University of California
Santa Barbara

Quasi-Polynomial Time Techniques for Independent Set and Beyond in Hereditary Graph Classes

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Peter Gartland

Committee in charge:

Professor Daniel Lokshtanov, Chair
Professor Subhash Suri
Professor Eric Vigoda

December 2023

The Dissertation of Peter Gartland is approved.

Professor Subhash Suri

Professor Eric Vigoda

Professor Daniel Lokshtanov, Committee Chair

December 2023

Quasi-Polynomial Time Techniques for Independent Set and Beyond in Hereditary Graph Classes

Copyright © 2023

by

Peter Gartland

To my family

Acknowledgements

I would like to thank my advisor, Daniel Lokshtanov, for introducing me to many problems I would pursue during my Ph.D. I am grateful for the technical knowledge he passed along to me and the insight he provided on these problems. I would also like to thank my other committee members, Subhash Suri and Eric Vigoda, for giving feedback on my dissertation and other Ph.D. milestones.

I am grateful to Michał Pilipczuk for hosting me at the University of Warsaw over the winter of 2023. During this visit, we, along with Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, and Paweł Rzażewski, were able to resolve one of the main results of this thesis, which appears in Chapter 4. I would also like to acknowledge all those involved in organizing the 2022 Dagstuhl Seminar on Vertex Partitioning, where we began our initial discussion on this problem.

Curriculum Vitæ

Peter Gartland

Education

2023	Ph.D. in Computer Science (Expected), University of California, Santa Barbara.
2019	M.A. in Mathematics, University of South Carolina.
2017	B.S. in Mathematics, The Catholic University of America

Publications

1. Maximum Weight Independent Set in Even-Hole-Free Graphs in Quasi-Polynomial Time. *Maria Chudnovsky, Peter Gartland, Daniel Lokshtanov.* In Preparation
2. Maximum Weight Independent Set in $\{Theta, Pyramid\}$-Free Graphs in Quasi-Polynomial Time. *Maria Chudnovsky, Peter Gartland, Daniel Lokshtanov.* In Preparation
3. On Induced Versions of Menger's Theorem on Sparse Graphs. *Peter Gartland, Tuukka Korhonen, Daniel Lokshtanov.* ArXiv 2023
4. Maximum Weight Independent Set in Graphs with no Long Claws in Quasi-Polynomial Time. *Peter Gartland, Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, Michał Pilipczuk, Paweł Rzażewski.* ArXiv 2023
5. Graph Classes with Few Minimal Separators. I. Finite Forbidden Induced Subgraphs. *Peter Gartland and Daniel Lokshtanov.* SODA 2023
6. Graph Classes with Few Minimal Separators. II. A Dichotomy. *Peter Gartland and Daniel Lokshtanov.* SODA 2023
7. Finding Large Induced Sparse Subgraphs in $C_{>t}$-Free Graphs in Quasi-Polynomial Time. *Peter Gartland, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, Paweł Rzażewski.* STOC 2021
8. Independent Set on P_k-Free Graphs in Quasi-Polynomial Time. *Peter Gartland and Daniel Lokshtanov.* FOCS 2020

Abstract

Quasi-Polynomial Time Techniques for Independent Set and Beyond in Hereditary Graph Classes

by

Peter Gartland

An independent set in a graph G is a collection of vertices with no edge between any two. The Independent Set problem is a classic NP-Hard problem that takes in a graph G and the task is to find an independent set in G of maximum size. An induced subgraph of G is another graph, H, where the vertex set of H is a subset $V(H) \subseteq V(G)$, and the edge set of H is all edges uv in $E(G)$ where $u, v \in V(H)$. A graph is said to be H-free if it does not contain H as an induced subgraph. Lastly, quasi-polynomial functions are functions that grow (slightly) faster than polynomial functions by allowing poly-log terms in their exponents, for example, $n^{log^2(n)}$ is a quasi-polynomial function.

In this thesis, we will provide a quasi-polynomial time algorithm for Independent Set on P_k-free graphs [1], where P_k denotes a path with k vertices. We will later extend this result in multiple ways. First, we give a quasi-polynomial time algorithm for Independent Set as well as related problems, such as Feedback Vertex Set on $C_{>k}$-free graphs [2], which are graphs that do not contain induced cycles with more than k vertices. Additionally, we will give a quasi-polynomial time algorithm for Independent Set on H-free graphs where H is a forest of paths and subdivided claws [3]. A subdivided claw is a graph formed by taking three paths of any desired length along with a vertex v that is a neighbor of exactly one end vertex in each of the three paths. This last result resolves an open problem first investigated by Alekseev [4] in 1982.

In a seemingly different research thread, we will characterize graph classes with few

minimal separators [5, 6]. A minimal separator is a set S such that there are two vertices u, v in different components of $G - S$ and for any proper subset $S' \subset S$, u and v are in the same component of $G - S'$. A class $\mathcal{F}$ of graphs is called tame if every graph in $\mathcal{F}$ on n vertices contains at most $n^{O(1)}$ minimal separators, *quasi-tame* if every graph in $\mathcal{F}$ on n vertices contains at most $n^{\log^{O(1)}(n)}$ minimal separators, and *feral* if there exists a constant $c > 1$ so that $\mathcal{F}$ contains n-vertex graphs with at least c^n minimal separators for arbitrarily large n. The classification of graph classes into (quasi) tame or feral has numerous algorithmic consequences and has recently received considerable attention. In particular, Independent Set and related problems can be solved in polynomial time on tame graph classes and quasi-polynomial time on quasi-tame graph classes [7].

An essential tool that both our characterization of quasi-tame graph classes and our quasi-polynomial time algorithms depend on are *dominated balanced separators*, which are vertex sets S of a graph G that are dominated by a constant number of vertices, and no component of $G - S$ has over, say, half of G's vertices. In the introductory chapter of this thesis, we give a conjecture on dominated balanced separators that we refer to as the Induced Grid Minor Conjecture. If true, it would unify many of the results in this thesis and give efficient algorithms for Independent Set and related problems on many graph classes for which such algorithms have long been sought with no success, such as even-hole-free graphs. Additionally, if this conjecture is true, we believe it could serve as a building block for additional interesting theorems for hereditary graph classes.

Contents

Chapter 1

Introduction

Many of the techniques used in this thesis build off tools developed in previous papers by other researchers. Additionally, many methods used in later chapters build off those developed in earlier chapters. So, for the reader to "get into our heads" and better understand how the fundamental ideas of this thesis were developed, in later sections of this introductory chapter, we will delve more into the techniques developed in previous works by other researchers as well as those that we create ourselves in this thesis in such a way that the reader may gain some understanding of how these tools came about. Readers who want more intuition and "story" behind this thesis can (and should) skip Section 1.1 and jump to Section 1.2 as much of what is said in Section 1.1 will appear in later sections of this chapter, with a more in-depth description. Readers who only wish to read a literature survey that merely describes the problems we resolve in this thesis and related publications on these problems only need to read Section 1.1.

1.1 Literature Survey

Independent Set An *independent set* (also known as a *stable set*) in a graph G is a vertex set S such that no two distinct vertices in S are adjacent in G. In the Independent Set problem the input is a graph G and an integer k, the task is to determine whether G contains an independent set S of size at least k. Up to polynomial time overhead, this task is equivalent to finding an independent set of maximum size in G, and we will often treat these two problems as equivalent. Independent Set is a well-studied and fundamental graph problem which is NP-complete [8, 9] and has a central role as a hard problem in many areas of computational complexity. One of the most well-known hardness results for Independent Set is that it was one of the very first problems shown to be NP-Hard to approximate [10, 11]; in fact, it is hard to approximate within a factor of $n^{1-\varepsilon}$ [12]. The Maximum Weight Independent Set problem represents a natural extension of this problem where the input graph is endowed with weights (real numbers) on its vertices. In this variant, the objective is to identify an independent set with maximum total weight. While our introductory chapter primarily concentrates on Independent Set, it's noteworthy that nearly all concepts discussed here readily apply to Maximum Weight Independent Set with only minor adjustments.

In light of the aforementioned complexity challenges of solving Independent Set on general graphs, researchers naturally focus on graph classes where solving the Independent Set problem becomes more manageable. Significant effort has been dedicated to identifying classes of graphs for which polynomial time algorithms exist for Independent Set. Some of the noteworthy results in this pursuit include polynomial time algorithms for Independent Set on Perfect Graphs [13], P_5-free graphs [14], and the development of a polynomial time approximation scheme for Independent Set on planar graphs [15] (Independent Set remains NP-Hard on planar graphs [8]).

Due to the vast and complex nature of all possible graph classes, which would include numerous peculiar and artificial counterexamples, it is necessary to impose restrictions on the classes considered to develop a rich theory regarding the computational complexity of Independent Set across these graph classes. A particularly interesting problem in this context, which has captivated the algorithmic graph theory community and will serve as a central theme in this thesis, revolves around the complexity of Independent Set on H-free graphs. H-free graphs are graphs that do not contain H as an induced subgraph, where an induced subgraph of a graph G is another graph, G', where the vertex set of G' is a subset $V(G') \subseteq V(G)$ and the edge set of G' is all edges uv in $E(G)$ where $u, v \in V(G')$. We will also be concerned with the slightly more general problem, Independent Set on $\mathcal{H}$-free graphs. Here, $\mathcal{H}$ is a finite set of graphs, and $\mathcal{H}$-free graphs are graphs that do not contain H as an induced subgraph for all $H \in \mathcal{H}$.

This problem was first investigated by Alekseev [4] in 1982. Alekseev observed a simple reduction that shows that Independent Set remains NP-Hard on $\mathcal{H}$-free graphs as long as $\mathcal{H}$ does not contain a graph H that is a forest of paths and subdivided claws (see Figure 1.1). The standard claw graph comprises of an independent set of size three and an additional vertex adjacent to the three independent vertices. A subdivided claw extends this concept, allowing for the three edges of the standard claw to be subdivided some number of times. The question of whether Independent Set on $\mathcal{H}$-free graphs, where $\mathcal{H}$ contains a graph H that is a forest of paths and subdivided claws, possesses a polynomial time algorithm for Independent Set has remained an open problem, which we conjecture to be true. It is straightforward to show that to prove this conjecture, it is both necessary and sufficient to show that H-free graphs possess a polynomial time algorithm for Independent Set when H is a forest of paths and subdivided claws. In light of this, we will now restrict our concern to the complexity of Independent Set on H-free graphs (where H is just a single graph, not a set of graphs). This thesis will

make substantial progress toward resolving this longstanding problem. In particular, we prove the following theorem in Chapter 3.

Theorem 1.1.1 ([16]). *For every positive integer k, there is an algorithm for* Independent Set *on P_k-free graphs running in quasi-polynomial time.*

Here, quasi-polynomial means poly-log functions are allowed in the exponent instead of just constants as in polynomial functions, for instance, $n^{log^2(n)}$ is a quasi-polynomial function. In Chapter 4, we extended the result of Theorem 1.1.1 to obtain the following theorem.

Theorem 1.1.2 ([3]). *For every graph H that is a forest of paths and subdivided claws, there is an algorithm for* Independent Set *on H-free graphs running in quasi-polynomial time.*

Our quasi-polynomial time algorithms have two significant implications. First, given the widespread belief that NP is not a subset of quasi-polynomial time (QP), our algorithms, under this assumption, imply that Independent Set on H-free graphs is not NP-Hard when H is a forest of paths and subdivided claws. Second, it is commonly observed that the discovery of a quasi-polynomial time algorithm often leads to the discovery of a polynomial time algorithm. Therefore, our quasi-polynomial time algorithms strongly support the conjecture that Independent Set on H-free graphs admits a polynomial time algorithm when H is a forest of paths and subdivided claws. Before we discuss these results further, let us look at previous work that has been done for this problem.

It has been known since 1980 that Independent Set is polynomial-time solvable in $S_{1,1,1}$-free graphs, by $S_{t,t,t}$ we mean a subdivided claw where each arm has t vertices (See Figure 1.1), so $S_{1,1,1}$ is just the standard claw graph. Around the same time, it

was shown that the class of P_4-free graphs, where P_t denotes a path with t vertices (see Figure 1.1), has powerful structural properties, thus allowing for efficient algorithms for Independent Set and many other combinatorial problems. Apart from results by Alekseev [17] (for the unweighted case) and Lozin and Milanič [18] (for the weighted case) extending the $S_{1,1,1}$-free case to $S_{1,1,2}$-free graphs the progress on polynomial time algorithms for Independent Set on H-free graphs where H is a forest of paths and subdivided claws was limited to various subclasses of P_t-free graphs and S_{t_1,t_2,t_3}-free graphs for small values of t, t_1, t_2, t_3 (for instance P_7, K_3-free graphs [19]). (see [20, 19, 21, 22, 23, 24, 25, 26, 27, 19, 28, 29, 30, 31, 32] for older and newer results of this kind) until around a decade ago.

Research in this area has gained significant momentum over the last decade, and progress can be categorized into two main threads. The first thread centers on the concept of *potential maximal cliques*, introduced by Bouchitte and Todinca [33]. This thread aims to devise polynomial-time algorithms for P_t-free graphs for small values of t. A groundbreaking result in this line of research is credited to Lokshtanov, Vatshelle, and Villanger [14], who were the first to demonstrate the effectiveness of the potential maximal clique framework in the context of P_t-free graphs by presenting a polynomial-time algorithm for Independent Set in P_5-free graphs. Subsequent extensions include algorithms for P_6-free graphs [34] and related graph classes [35].

The second thread attempts at treating P_t-free or $S_{t,t,t}$-free graphs in full generality, but relaxing the requirements on either the running time (by providing subexponential or quasi-polynomial-time algorithms) or the accuracy (by providing approximation algorithms, such as approximation schemes). Here, the starting point is the theorem of Gyárfás [36] (see also [37]).

Theorem 1.1.3. *[Gyárfás Path Growing Argument [36], [37]] Every vertex-weighted*

graph G contains an induced path Q such that every connected component of $G - N[V(Q)]$ has weight at most half of the weight of G.

Since induced paths in a P_t-free graph have less than t vertices, a P_t-free graph admits a balanced separator (in the sense of Theorem 1.1.3) consisting of the neighborhood of at most $t-1$ vertices. Chudnovsky et al. [38] observed that this easily gives a quasi-polynomial-time approximation scheme (QPTAS) for MWIS in P_t-free graphs, and they designed an elaborate argument involving the celebrated three-in-a-tree theorem of Chudnovsky and Seymour [39] to extend the result to the $S_{t,t,t}$-free case and H-free case where H is a forest of paths and subdivided claws. Abrishami et al. [40] also used the three-in-a-tree theorem to obtain a polynomial-time algorithm for MWIS for $S_{t,t,t}$-free graphs of bounded degree.

In Chapter 3, we show how to use Theorem 1.1.3 to design an exact quasi-polynomial time algorithm of Theorem 1.1.1. This algorithm was later simplified by Pilipczuk, Pilipczuk, and Rzażewski [41]. Chapter 4, where Theorem 1.1.2 is proven, provides the pinnacle of this thread of research by showing that Independent Set is quasi-polynomial time solvable on H-free graphs for all open cases, that is, when H is a forest of paths and subdivided claws.

Our last result in the direction of solving Independent set on some hereditary graph class will come in Chapter 5, the work in this chapter originally appeared in [2]. There, we will provide an algorithm that extends Theorem 1.1.1 in two ways. First, it broadens the graph class that it can run on from P_k-free graphs to $C_{>k}$-free graphs, $C_{>k}$-free graphs being graphs with no induced cycle of length more than k. Second, the algorithm can solve an extensive set of problems, which includes Independent Set, but also many other interesting problems such as Feedback Vertex Set, Maximum Weight Induced Planar Subgraph, Maximum Length Induced Path, and 3-Coloring. This

(substantially) generalizes a question asked by Pilipczuk, Pilipczuk, and Rzażewski [41], which asks if there exists a quasi-polynomial time algorithm for Feedback Vertex Set on P_k-free graphs. This additionally generalizes a paper of Abrishami et al. [35], which showed that $C_{>4}$-free graphs admit polynomial time algorithms for the same set of problems.

We now turn our attention to a different set of problems, in particular, we will be concerned about certain graph structures called minimal separators. At first, it will seem like we are dealing with a very different set of problems than we were in this subsection. But, in Section 1.4, we propose a conjecture which shows, in some sense, that these are really two different subproblems of the same underlying problem.

Minimal Separators In the last two chapters of this thesis, Chapters 6 and 7, we turn our attention to minimal separators. These two chapters correspond to the papers [5] and [6]. In a graph G with vertices u and v, a subset $S \subset V(G)$ is called a *u, v-separator* if u and v exist in different components of $G - S$. The set S is a *u, v-minimal separator* if no proper subset of S is a u, v-separator. S is a *minimal separator* if it is a u, v-minimal separator for some pair of vertices $u, v \in G$. A graph class $\mathcal{F}$ is called *tame* if there exists a polynomial function p such that for all $G \in \mathcal{F}$, G has at most $p(|G|)$ minimal separators. $\mathcal{F}$ is called *quasi-tame* if there exists a quasi-polynomial function p such that for all $G \in \mathcal{F}$, G has at most $p(|G|)$ minimal separators. On the other hand, a graph class is called *feral* if there exists a constant $c > 1$ such that for arbitrarily large graphs $G \in \mathcal{F}$, G contains at least $c^{|G|}$ minimal separators. A graph class is feral when an exponential bound on the number of minimal separators is the best possible. Despite the existence of functions with growth rates between (quasi) polynomial and exponential, in Chapter 7, we formally prove that the large majority of typical graph classes fall into either the (quasi) tame or feral category.

A sequence of noteworthy results, starting with a paper by Bouchitte and Todinca [33] and concluding with a paper by Fomin et al. [7], established that a large number of problems, the most interesting of which being Independent Set and Feedback Vertex Set, can be solved in time polynomial in the number of minimal separators in the graph. This proves that tame graph classes admit a polynomial time algorithm for these problems, while quasi-tame graph classes admit a quasi-polynomial time algorithm for the same problems. Numerous extensively studied graph classes fall into the category of tame graphs, including chordal graphs, permutation graphs, interval graphs, and P_4-free graphs (also known as cographs). These results on minimal separators contribute to a unified theory explaining why these graph classes exhibit polynomial time algorithms for problems like Independent Set and Feedback Vertex Set.

Recently, researchers have shown an interest in acquiring a broader understanding of the factors that force a graph class to be tame or feral. Milanic and Pivac undertook the initial systematic exploration of tame versus feral graph classes [42], where they systematically categorized all graph classes defined by forbidden induced subgraphs of size four or less into tame and feral classes. Another significant contribution in this direction was made by Abrishami et al. [43], who demonstrated that graphs avoiding structures known as thetas, prisms, pyramids, and turtles as induced subgraphs fall into the category of tame graphs. In their study, Abrishami et al. [43] identified the significance of the presence or absence of structures termed *k-creatures* in determining whether a graph class is tame or feral. Due to the pivotal role of k-creatures in explaining our results on this problem, we present the definition here.

Definition 1.1.4 (k-creatures (see Figure 1.3))**.** A graph G is said to be a *k-creature* if its vertices can be partitioned into sets A, $X = \{x_1, x_2, \ldots, x_k\}$, $Y = \{y_1, y_2, \ldots, y_k\}$, and B satisfying the following conditions:

1. $G[A]$ and $G[B]$ are connected.
2. A and $Y \cup B$ are anti-complete (i.e. $N[A] \cap (Y \cup B) = \emptyset$) and B and $A \cup X$ are anti-complete.
3. A dominates X (every vertex in X has a neighbor in A) and B dominates Y.
4. $x_i y_j$ is an edge if and only if $i = j$.

If a graph G does not contain an induced subgraph that is a k-creature, then we say that G is k-creature-free.

In a k-creature, it is easy to see that for a vertex $a \in A$ and $b \in B$, there are precisely 2^k minimal separators disjoint from A and B. Such a separator must pick precisely one vertex from each of $\{x_i, y_i\}$, and it can make each one of these k choices independently.

The aforementioned previous works essentially give different sufficient conditions for a graph to only have polynomially many minimal separators. This naturally leads to the question *is there a "right" sufficient condition for tameness?* That is - a condition that, on the one hand, is easy to state and verify, while on the other hand, captures all interesting tame graph classes. Abrishami et al. [43] conjectured that for every integer k, there exists a k' such that if an n-vertex graph G is k-creature-free, then G has at most $n^{k'}$ minimal separators. It turns out, we will show in Chapter 6 that this conjecture is false: for arbitrarily large n there exist n-vertex graphs that exclude 10-creatures and yet have $2^{\Omega(n)}$ minimal separators. Let us discuss, though, for a moment, why it would have been the "right" sufficient condition for polynomially many minimal separators if it had been true.

With only very weak assumptions, which the vast majority of interesting graph classes satisfy (which we formally describe in Chapter 7), we can show that if $\mathcal{F}$ is a graph class that satisfies these conditions and for arbitrarily large integers k, there is a $G \in \mathcal{F}$ such that G contains a k-creature, then for every k there is a graph $G' \in \mathcal{F}$ such that G'

contains a k-creature and $|G'| = O(k)$. It follows that G' contains $2^{\Omega(k)} = 2^{\Omega(|G'|)}$ minimal separators. Hence, $\mathcal{F}$ is feral. Thus, if the conjecture of Abrishami et al. were true, we would be able to say that if there exists a positive integer k such that no graph in $\mathcal{F}$ contains a k-creature, then $\mathcal{F}$ is tame, otherwise $\mathcal{F}$ is feral. In some sense, k-creatures would be the underlying reason graph classes are tame or feral.

As previously mentioned, we show in Chapter 6 that the conjecture of Abrishami et al. is false. In the same chapter, we show that a weaker version of the conjecture of Abrishami et al. is true. To state this result, we need a couple of definitions. First, a *k-skinny ladder* is a graph G consisting of two anti-complete paths $P_l = \ell_1\ell_2 \dots \ell_k$ and $P_r = r_1r_2 \dots r_k$ (subgraphs A and B are anti-complete if $V(A) \cap V(B) = \emptyset$ and there is no edge between A and B) and a set $\{s_1, s_2, \dots, s_k\}$ of vertices such that for every i, s_i is adjacent to ℓ_i and r_i and to no other vertices. Second, an *induced minor* of a graph G is a graph that can be obtained from G by deleting vertices and contracting edges (contracting an edge uv of a graph G results in a new graph G' where the vertices u and v has been removed and a new vertex w is added where w is made a neighbor of all vertices that was a neighbor of either u, v, or both u and v). The main result of Chapter 6 is the following weakening of the conjecture of Abrishami et al.

Theorem 1.1.5 ([5]). *For every integer k, the family of graphs that are k-creature free and exclude the k-skinny ladder as an induced minor is quasi-tame.*

Theorem 1.1.5 is strong enough to allow us to classify graph classes defined by a finite number of forbidden induced subgraphs. In particular, in Chapter 6 we will prove, using Theorem 1.1.5 the following theorem.

Theorem 1.1.6 ([5]). *Let $\mathcal{F}$ be a graph family defined by a finite number of forbidden induced subgraphs. If there exists a natural number k such that $\mathcal{F}$ forbids all k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, k-claw, and k-paw graphs (see Figure*

6.2), then $\mathcal{F}$ is quasi-tame. Otherwise, $\mathcal{F}$ is feral.

While Theorem 1.1.5 is strong enough to give a complete dichotomy for graph classes defined by a finite number of forbidden induced subgraphs, it fails to be the "right" theorem we seek for two reasons. The first problem is that the bounds are quasi-polynomial as opposed to polynomial. The second is that there are still very reasonable tame graph classes not captured by this theorem, such as the graph class made up of k-skinny-ladders for all natural numbers k.

The first issue was dealt with by Gajarsky et al. [44] who, building heavily upon the work that will be presented in Chapter 6, brought the bounds of Theorem 1.1.5 down from quasi-polynomial to polynomial. The second issue is dealt with in Chapter 7. To properly state our main result of Chapter 7, we must introduce another graph called *t-critters*. First, we need a small generalization of minimal separators that applies to vertex sets instead of just vertices. Given $A, B \subset V(G)$, we define S to be an *A, B-separator* if $A \cap S = B \cap S = \emptyset$ and no component of $G - S$ contains a vertex from both A and B. An *A,B-minimal separator* is an A, B-separator such that no proper subset of S is an A, B-separator.

Definition 1.1.7 (t-critter partition, t-critter)**.** A *t-critter partition* of a graph G is a partition of the vertex set of G into sets $A_1, A_2, \ldots A_{t+1}$, $B_1, B_2, \ldots, B_{t+1}, X_1, X_2, \ldots, X_t$, such that the following conditions are satisfied.

1. For all $1 \leqslant i, j \leqslant t+1$ with $i \neq j$, A_i is anti-complete with A_j, B_i is anti-complete with B_j, and A_i is anti-complete with B_j.
2. For all $1 \leqslant i \leqslant t+1$ A_i and B_i is connected.
3. The vertices of A_i, A_{i+1}, B_i, and B_{i+1} are the only vertices outside of X_i that have a neighbor in X_i.

4. There are (at least) two distinct $(A_i \cup A_{i+1})$, $(B_i \cup B_{i+1})$-minimal separators in $G[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i]$, S_1^i and S_2^i, such that there is a path from A_i to A_{i+1} through both $X_i - S_1^i$ and $X_i - S_2^i$ and there is a path from B_i to B_{i+1} through both $X_i - S_1^i$ and $X_i - S_2^i$.

A graph G is a *t-critter* if G has a t-critter partition. A graph G is *t-critter free* if G does not contain a t-critter as an induced subgraph.

The definition of t-critters is arguably technical and unappealing. A reader who is familiar with minimal separators should, after staring at the definition for a bit, be able to convince themselves that just as for k-creatures, for every vertex $a \in A = \bigcup_i A_i$ and vertex $b \in B = \bigcup_i B_i$ there are at least 2^t minimal a,b-separators in G disjoint from $A \cup B$. In particular, to separate a from b, we may, for every $i \leqslant t$, choose to delete either S_1^i or S_2^i, and for every i the choice between S_1^i or S_2^i can be made independently of the other choices. We are now ready to state the main theorem of Chapter 7 (See Chapter 7 for the formal statement of the theorem).

Theorem 1.1.8 ([6]). *Let $\mathcal{F}$ be a graph class that satisfies some weak assumptions (that almost all interesting graph classes satisfy). If there exist integers t and k such that for all $G \in \mathcal{F}$, G is k-creature free and t-critter free, then $\mathcal{F}$ is quasi-tame, else $\mathcal{F}$ is feral.*

This concludes our literature review section. Readers who wish to gain more insight into the works presented here are encouraged to read the other sections of this chapter.

1.2 Independent Set

An *independent set* (also known as a *stable set*) in a graph G is a vertex set S such that no two distinct vertices in S are adjacent in G. In the Independent Set problem, the input is a graph G and an integer k; the task is to determine whether

G contains an independent set S of size at least k. Up to polynomial time overhead, this task is equivalent to finding an independent set of maximum size in G, and we will often treat these two problems as equivalent. Independent Set is a well-studied and fundamental graph problem which is NP-complete [8, 9] and intractable within most frameworks for coping with NP-Hardness, playing a central role as a hard problem in many areas of computational complexity. Indeed, Independent Set was one of the very first problems shown to be NP-Hard to approximate [10, 11], intractable from the perspective of parameterized complexity [45], not to have a $2^{o(n)}$ time algorithm assuming the Exponential Time Hypothesis (ETH) [46], and whose hardness of parameterized approximation, assuming the Gap-ETH, was established [47]. The Maximum Weight Independent Set problem is a natural extension of Independent Set where the input graph now has weights on the vertices, and the problem asks to find an independent set of maximum total weight. In this introductory chapter, we will typically focus on Independent Set, but almost everything said here extends to Maximum Weight Independent Set with only minor modifications.

Given these hardness results above, researchers naturally turned their attention to graph classes for which Independent Set is tractable, and much work has been put into identifying classes for which there are polynomial time algorithms for Independent Set. Some of the most exciting results in this direction have been polynomial time algorithms for Independent Set on Perfect Graphs [13], P_5-free graphs [14], and a polynomial time approximation scheme for Independent Set on planar graphs [15] (Independent Set remains NP-Hard on planar graphs [8]).

As the space of all possible graph classes is too large and contains many strange and artificial counterexamples, we must restrict the graph classes we consider in some way if we wish to build an interesting theory around the computational complexity of Independent Set for various graph classes. One problem in this direction that has

generated much interest in the algorithmic graph theory community and a problem that will be a central focus in this thesis is the complexity of Independent Set on H-free graphs. H-free graphs are graphs that do not contain H as an induced subgraph, where an induced subgraph of a graph G is another graph, G', where the vertex set of G' is a subset $V(G') \subseteq V(G)$ and the edge set of G' is all edges uv in $E(G)$ where $u, v \in V(G')$. We will also be concerned with the slightly more general problem, Independent Set on $\mathcal{H}$-free graphs. Here, $\mathcal{H}$ is a finite set of graphs, and $\mathcal{H}$-free graphs are graphs that do not contain H as an induced subgraph for all $H \in \mathcal{H}$. This problem was first investigated by Alekseev [4] in 1982. Alekseev observed a simple reduction that shows that Independent Set remains NP-Hard on H-free graphs as long as H contains a vertex of degree four or more, a cycle, or two vertices of degree three in the same component. Let's quickly see how the reduction goes.

The reduction is from Independent Set on max degree three graphs, which is known to be NP-Hard [48], and goes as follows. Fix a graph H that has a vertex of degree four or more, a cycle, or two vertices of degree three in the same component. Let G be a graph of max degree three. It is easy to see that if we subdivide an edge of G exactly twice, the maximum independent set size increases by exactly one. Subdividing an edge uv of a graph G means we insert a new vertex w in the middle of the edge uv. More formally, we add the vertex w to the vertex set of G, we remove the edge uv from the edge set of G, and we add the edges uw and wv to the edge set of G. Now, we subdivide each edge of G exactly $2|H|$ times to get a new graph G'. So, $\alpha(G) + |E(G)||H| = \alpha(G')$, where $\alpha(G)$ denotes the maximum independent set size of a graph G. Furthermore, G' is H-free as G' has maximum degree three, all cycles in G' have length at least $2|H|$, and all vertices of degree three in the same component of G' are distance at least $2|H|$ apart and H was assumed to either have a vertex of degree at least four, a cycle, or two vertices of degree three in the same component. It follows that Independent Set is

NP-Hard on H-free graphs. The same reduction also shows that Independent Set remains NP-Hard on $\mathcal{H}$-free graphs when all graphs $H \in \mathcal{H}$ have a vertex of degree at least four, a cycle, or two vertices of degree three in the same component. Also, since Independent Set is hard to approximate (APX-Hard) on graphs of maximum degree three [49], Independent Set remains APX-Hard on $\mathcal{H}$-free graphs when all graphs $H \in \mathcal{H}$ have a vertex of degree at least four, a cycle, or two vertices of degree three in the same component.

So, what kind of graphs have max degree three, are acyclic, and have at most one vertex of degree three in each component? These graphs are forests of paths and subdivided claws (see Figure 1.1). The standard claw graph is an independent set of size three, and an additional vertex neighbors with the three independent vertices. A subdivided claw is one where the three edges of the standard claw may now be subdivided some number of times. We make the following conjecture about Independent Set when forests of paths and subdivided claws are forbidden. This conjecture appeared in the paper corresponding to Chapter 3 of this thesis and possibly elsewhere as well (this conjecture was a known and investigated open problem well before any of the work in this thesis was done, but before the work in this thesis there was little evidence of its truth, so few researchers would have referred to it as a conjecture).

Conjecture 1.2.1. *Let $\mathcal{H}$ be a finite set of graphs.* Independent Set *is polynomial-time solvable on $\mathcal{H}$-free graphs when $\mathcal{H}$ contains at least one graph H that is a forest of paths and subdivided claws, otherwise, it is* NP*-Hard.*

Notice that the NP-Hardness part of the conjecture is covered in the hardness proof we just presented. Additionally, if $H \in \mathcal{H}$, then the set of H-free graphs contains the set of $\mathcal{H}$-free graphs. So, this conjecture reduces to proving that Independent Set is polynomial-time solvable on H-free graphs where H is a forest of paths and subdivided

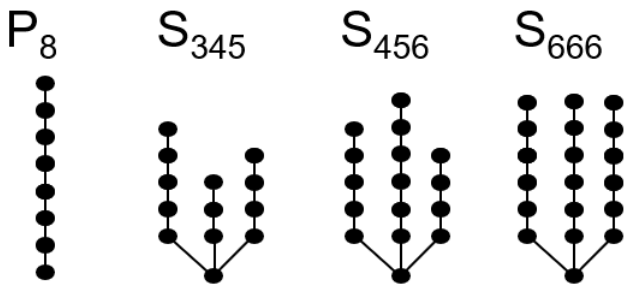


Figure 1.1: A forest of paths and subdivided claws made up of a path of length eight (P_8), a subdivided claw with arm lengths three, four, and five ($S_{3,4,5}$), subdivide claw with arm lengths four, five, and six ($S_{4,5,6}$), and a subdivided claw with all arms of length six ($S_{6,6,6}$).

claws. In this thesis, we will make substantial progress towards answering this conjecture. In particular, we prove the following theorem in Chapter 3.

Theorem 1.2.2 ([1])**.** *For every positive integer k, there is an algorithm for* Independent Set *on P_k-free graphs running in quasi-polynomial time.*

Here, quasi-polynomial means poly-log functions are allowed in the exponent instead of just constants as in polynomial functions, for instance, $n^{log^2(n)}$ is a quasi-polynomial function. In Chapter 4, we extended the result of Theorem 1.2.2 to obtain the following theorem.

Theorem 1.2.3 ([3])**.** *For every graph H that is a forest of paths and subdivided claws, there is an algorithm for* Independent Set *on H-free graphs running in quasi-polynomial time.*

The works in Chapters 3 and 4 appeared in [1] and [3] respectively.

Our quasi-polynomial time algorithms have two implications for Conjecture 1.2.1. The first is that it is widely believed that NP is not a subset of quasi-polynomial time (QP), so assuming this, our quasi-polynomial time algorithms allow us to conclude that Independent Set on H-free graphs is not NP-Hard when H is a forest of paths and

subdivided claws. The second is that most of the time when a quasi-polynomial time algorithm is found, researchers eventually find a polynomial time algorithm, so this tells us that Conjecture 1.2.1 is likely to be true. Before we discuss this result further, let us look at previous work that has been done for this problem.

It has been known since 1980 that Independent Set is polynomial-time solvable in $S_{1,1,1}$-free graphs, where $S_{t,t,t}$ denotes a claw where each arm has t vertices, so $S_{1,1,1}$ is just the standard claw graph, see Figure 1.1. Around the same time, it was shown that the class of P_4-free graphs, where P_t denotes a path with t vertices (see Figure 1.1) coincides with the class of *cographs* and has very strong structural properties (in modern terms, has bounded cliquewidth) thus allowing for efficient algorithms for Independent Set and many other combinatorial problems. Apart from results by Alekseev [17] (for the unweighted case) and Lozin and Milanič [18] (for the weighted case) extending the $S_{1,1,1}$-free case to $S_{1,1,2}$-free graphs the progress on Conjecture 4.1.1 was limited to various subclasses of P_t-free graphs and S_{t_1,t_2,t_3}-free graphs for small values of t, t_1, t_2, t_3 (for instance $\{P_7, K_3\}$-free graphs [19]). (see [20, 19, 21, 22, 23, 24, 25, 26, 27, 19, 28, 29, 30, 31, 32] for older and newer results of this kind) until around a decade ago.

Research in this area gained significant momentum in the last decade. The progress can be partitioned into two main threads. The first one focuses on the framework of *potential maximal cliques*, introduced by Bouchitté and Todinca [33], and tries to provide polynomial-time algorithms for P_t-free graphs for small values of t. A landmark result here is due to Lokshtanov, Vatshelle, and Villanger [14], who were the first to show the usability of the framework in the context of P_t-free graphs by providing a polynomial-time algorithm for Independent Set in P_5-free graphs. This has been later extended to P_6-free graphs [34] and related graph classes [35]. A notable property of this framework is that, in most cases, it not only provides algorithms for Independent Set, but for a wide range of problems asking for large induced subgraph of small treewidth, for example

Feedback Vertex Set.

The second thread attempts to treat P_t-free or $S_{t,t,t}$-free graphs in full generality, but relaxing the requirements on either the running time (by providing subexponential or quasi-polynomial-time algorithms) or the accuracy (by providing approximation algorithms, such as approximation schemes). This thread of research will be one of the main focuses of this thesis, in particular, our proof of Theorem 1.2.3 represents the pinnacle of this second line of research. Three papers by other authors on this second line of research have had significant influence over the development of this algorithm, the first is a $2^{\tilde{\mathcal{O}}(\sqrt{n})}$-time algorithm (here $\tilde{\mathcal{O}}$ suppresses poly-log factors) for Independent Set on P_k-free graphs [37], the second is a quasi-polynomial time approximation scheme (QPTAS) for Independent Set on H-free graphs where H is a forest of paths and subdivided claws [38], resolving all open cases in the QPTAS setting, and the third, which was concurrent with the work in this thesis, gave a more straightforward proof of the previously mentioned QPTAS using a nice structural property of $S_{t,t,t}$-free graphs [50] which will be an essential ingredient to some of our work. Because of the importance of these three papers to this thesis, we will briefly sketch the ideas of these papers here.

We begin with the $2^{\tilde{\mathcal{O}}(\sqrt{n})}$-time algorithm for Independent Set on P_k-free graphs of [37]. This paper introduced the notion of *dominated balanced separators*. A *balanced separator* S for a graph G is a set $S \subseteq V(G)$ such that no component of $G - S$ has over $|G|/2$ vertices. We say a set S is a *dominated balanced separator* if S can be dominated by a small number of vertices in G (either by some fixed constant or fixed function that is poly-logarithmic in the number of vertices of the graph). The reader should be aware that the definition of dominated balanced separators will be made more formal and change slightly in other chapters based on certain parameters, such as how large the components of $G - S$ are allowed to be, the definition being used in a given chapter will be made clear at the start of that chapter. We say that a hereditary graph class $\mathcal{F}$ has *dominated*

balanced separators if every $G \in \mathcal{F}$ has a dominated balanced separator. By hereditary, we mean that $\mathcal{F}$ is closed under vertex deletion, so if $G \in \mathcal{F}$ and $v \in G$, then $G - v \in \mathcal{F}$. So, some fixed poly-logarithmic function f exists, such that for every $G \in \mathcal{F}$, G has a balanced separator dominated by at most $f(|G|)$ vertices. The $2^{\tilde{\mathcal{O}}(\sqrt{n})}$-time algorithm for Independent Set on P_k-free graphs is based on two theorems which we will now sketch. The first is that P_k-free graphs have dominated balanced separators, based on the Gyárfás path growing argument, first used to prove χ-boundedness of P_k-free graphs [36]. The second is that any hereditary graph class with dominated balanced separators admits a $2^{\tilde{\mathcal{O}}(\sqrt{n})}$-time algorithm for Independent Set.

Theorem 1.2.4 (Gyárfás Path Growing Argument [36], [37])**.** *Let G be a P_k-free graph. There is a set S of size less than k such that $N[S]$ is a balanced separator.*

Proof: [sketch] Let G be a P_k-free graph, for simplicity assume that G is connected, and let v_1 be any vertex of G. We will find a sequence of vertices $P^j = \{v_1, v_2, \ldots, v_j\}$ such that P^j is an induced path in G (and hence $j < k$) and $N[P^j]$ is a balanced separator. To do this, we take an induced path $P^i = \{v_1, v_2, \ldots, v_i\}$ (hence $i < k$) and v_i has a neighbor, v, which no other vertex of P^i is a neighbor of and v has a neighbor in the large component of $G - N[P^i]$ (the unique component with over half of G's vertices - if there is no large component then we are done as we have found a dominated balanced separator). We show how to extend it to a path P^{i+1} with the same properties. Since G is P_k-free, this process cannot continue more than k iterations.

So, assume that we have such a $P^i = \{v_1, v_2, \ldots, v_i\}$ and let C be the large component of $G - N[P^i]$. Set $v_{i+1} = v$ and $P^{i+1} = \{v_1, v_2, \ldots, v_i, v_{i+1}\}$. P^{i+1} is a path since only v_i is neighbors with v_{i+1}. If $N[P^{i+1}]$ is a balanced separator, we are done, so we may assume there is a large component, C', of $G - N[P^{i+1}]$, this implies that C' is the large component of $C - N[v_{i+1}]$. Since C is connected and $C \cap N(v_{i+1})$ is non-empty by assumption, there

must be some vertex, v', in $C \cap N(v_{i+1})$ that has a neighbor in C' and since $v' \in C$, no vertex of P^i can be neighbors with v'. ■

Next, we sketch the $2^{\tilde{\mathcal{O}}(\sqrt{n})}$-time algorithm for Independent Set on any hereditary class with dominated balanced separators. We will use the term *branching* on a vertex in the following algorithm (and in many other places in this thesis). Branching on a vertex v in a graph G when trying to solve the Independent Set problem means making two recursive calls, one where you guess that v is in a maximum independent set of G and one where you guess it is not. If v is in a maximum independent set of G, then it is easy to verify that $\alpha(G) = \alpha(G - N[v]) + 1$, where $\alpha(G)$ is the size of a maximum independent set of G. If v is not in a maximum independent set of G, then $\alpha(G) = \alpha(G - v)$. Thus when we branch on v (in the context of solving Independent Set), we make two recursive calls, one solving Independent Set on $G - N[v]$ and one on $G - v$ and we return the maximum of $\alpha(G - N[v]) + 1$ and $\alpha(G - v)$.

Theorem 1.2.5 ([37])**.** *Let $\mathcal{F}$ be a hereditary graph class and f a quasi-polynomial function such that for every $G \in \mathcal{F}$, there is a balanced separator S of G dominated by at most $f(|G|)$ vertices in G. Then $\mathcal{F}$ admits a $2^{\tilde{\mathcal{O}}(\sqrt{n})}$-time algorithm for* Independent Set.

Proof: [sketch] Let $\mathcal{F}$ and f be as in the statement of this theorem, and let $G \in \mathcal{F}$ be a n-vertex graph. We begin by branching on all vertices of degree at least $\sqrt{n}$ in G until none are left. The runtime of this step is governed by the recurrence $T(n) \leqslant T(n-1) + T(n-\sqrt{n})$. Repeatedly applying the recurrence relationship $\sqrt{n} - 1$ times to the $T(n-1)$ term gives $T(n) \leqslant \sqrt{n}T(n-\sqrt{n})$ which solves to $2^{\tilde{\mathcal{O}}(\sqrt{n})}$, so this step takes $2^{\tilde{\mathcal{O}}(\sqrt{n})}$ time. We now may assume that we are working with a graph G' of max degree $\sqrt{n}$. We then find a balanced separator S dominated by at most $f(n)$ vertices. Since G' has max degree $\sqrt{n}$, and S is dominated by at most $f(n)$ vertices, $|S| \leqslant \sqrt{n}f(n)$.

We branch on each vertex of S, as there are at most $\sqrt{n}f(n)$ vertices in S, this leads to at most $2^{\sqrt{n}f(n)}$ cases to consider. Since no vertex of S is left after this branching, the size of the largest component after branching is at most $n/2$. Since the union of maximum independent sets of each component of a graph is a maximum independent set of the entire graph, we can recurse on each component independently and return the sum of the maximum independent set sizes found for each component. As there are at most n components in the graph, the runtime of this step is governed by the recurrence $T(n) \leqslant n \cdot 2^{\sqrt{n}f(n)}T(n/2)$ which solves to $2^{\tilde{\mathcal{O}}(\sqrt{n})}$. ■

Together Theorems 1.2.4 and 1.2.5 show that P_k-free graphs have a $2^{\tilde{\mathcal{O}}(\sqrt{n})}$-time algorithm for Independent Set. Next, we consider the paper of Chudnovsky et al. [38], which gives a QPTAS for Independent Set on H-free graphs where H is a forest of paths and subdivided claws. There are two important ideas to take away from this paper, the first is another application of dominated balanced separators, in particular, they show that any hereditary graph class with dominated balanced separators admits a QPTAS for Independent Set. We now sketch a proof of this.

Theorem 1.2.6 ([38])**.** *Let $\mathcal{F}$ be a hereditary graph class with dominated balanced separators. Then there exists a QPTAS for* Independent Set *on $\mathcal{F}$.*

Proof: [sketch] For simplicity, we assume there is a constant c such that for all graphs $G \in \mathcal{F}$, G has a balanced separator dominated by c vertices. Now, let $G \in \mathcal{F}$ be an n-vertex graph, ε the desired accuracy of the algorithm, I a maximum independent set of G, and $\beta = \varepsilon^{-1}c\log(n)$. We call a vertex $v \in G$ *I-heavy* if $N[v]$ contains at least β^{-1} fraction of the vertices of I. First, we show a small set X such that $N[X]$ contains all I-heavy vertices.

Let v be an I-heavy vertex and let w be a uniformly at random chosen vertex in I, then there is at most a $(1-\beta^{-1})$ chance that $v \notin N[w]$. It follows that if $X \subseteq I$ is a

set of size $\mathcal{O}(\beta \log(n))$ then there is less than $1/n$ chance that $v \notin N[X]$ and since there are at most n I-heavy vertices, using the union bound there is a non-zero chance that all I-heavy vertices are in $N[X]$. It follows there is some set X of size $\mathcal{O}(\beta \log(n))$ such that $N[X]$ contains all I-heavy vertices.

The algorithm works as follows. Investigate all $n^{\mathcal{O}(\beta \log(n))} = n^{\mathcal{O}(\varepsilon^{-1} c \log^2(n))}$ subcases corresponding to a possible choice for $X \subseteq I$ of size $\mathcal{O}(\beta \log(n))$ determined from the previous paragraph. Since $X \subseteq I$, we may remove $N[X]$ from G, and now no I-heavy vertices are left. We then take a set S of size at most c such that $N[S]$ is a balanced separator and remove $N[S]$ from G. Since there are no I-heavy vertices, we lose at most $\beta^{-1} c \leqslant \varepsilon / \log(n)$ of the weight of I in this step. As every component of $G - N[S]$ has at most $n/2$ vertices, the depth of this recursion is at most $\log(n)$. Hence, we lose at most $\varepsilon |I|$ fraction of the weight throughout this recursion. Additionally, it is straightforward to check that this recursion tree has at most $2^{\mathcal{O}(\varepsilon^{-1} c \log^4(n))}$ nodes. ■

Extended Strip Decompositions The second important idea of this paper [38] is more technical and makes use of a decomposition known as extended strip decompositions first introduced by Seymour and Chudnovsky to solve the Three-in-a-Tree problem [39]. For a detailed definition of extended strip decompositions, see Chapter 4; for now, we present an informal and simplified discussion of extended strip decompositions. An extended strip decomposition can, in some sense, be viewed as a generalization of the notion of line graphs (a line graph of a graph G is a new graph $L(G)$ where the vertex set of $L(G)$ is the edge set of G and two vertices in $L(G)$ are neighbors if and only if there corresponding edges in G are adjacent). While not every graph G can be realized as the line graph, $L(H)$, of some graph H, it is true that every graph G has an extended strip decomposition (H, η). Here, H is a graph, and η is a function that maps the vertices, edges, and triangles of H (where triangles of H are vertices x, y, and z such that

$xy, xz, yx \in E(H)$) onto vertex sets of G in such a way as to partition $V(G)$. This mapping must satisfy specific properties which resemble those of a line graph. For instance, let e and e' be edges in some graph K. Then e and e' are vertices in the line graph, $L(K)$, of K, and there is an edge between e and e' if and only if e and e' are adjacent edges in H. Similarly, if $x, y \in G$ and e, e' are now edges of H such that $x \in \eta(e)$ and $y \in \eta(e')$ then if e and e' are adjacent edges in H, there must be an edge between x and y in G, and if e and e' are not adjacent edges in H, then there can be no edge between x and y in G.

One additional structure of extended strip decompositions that will be useful are *particles*, which are more or less the image under η of small collections of vertices, edges, and triangles that are adjacent to each other. Without losing too much, you may think of a particle as the image of η with respect to some vertex, edge, or triangle of H. It is worth noting here that if G is the line graph of a graph H, then G has an extended strip decomposition (H, η) where η maps the vertex e in G to the corresponding edge e in H. Thus, G has an extended strip decomposition where each particle is very small, in fact, each particle is just a single vertex.

Recall the well-known fact that if H denotes a graph and $L(H)$ denotes the line graph of H, then Independent Set on $L(H)$ can be solved by applying a Maximum Matching algorithm to H, which can be done in polynomial time. Since there is some connection between extended strip decompositions and line graphs, there is some hope that these structures might help us solve Independent Set. Indeed, similar to how Independent Set on $L(H)$ can be solved by applying a Maximum Matching algorithm on H, the authors show that if a maximum independent set of every particle of (H, η) is known (or more precisely the graphs induced by each particle), then the maximum independent set of G can be computed in polynomial time by applying a Maximum Matching algorithm to a modified version of H. This implies that if there

is a hereditary graph class $\mathcal{F}$ such that for every $G \in \mathcal{F}$ we can find an extended strip decomposition (H, η) of G such that no particle of (H, η) contains over $|G|/c$ vertices for some fixed $c > 1$ (we will refer to such extended strip decompositions as *good*), then we can solve Independent Set in quasi-polynomial time on $\mathcal{F}$. This is done by first computing said extended strip decomposition, then recursively solving Independent Set on each particle (which now has at most $|G|/c$ vertices) and then applying an algorithm for Maximum Matching to find a maximum independent set of G. More formally, the authors implicitly prove the following theorem.

Theorem 1.2.7 ([38])**.** *Let $c > 1$ and $\mathcal{F}$ be a hereditary graph class such that for every $G \in \mathcal{F}$ there is an extended strip decomposition (H, η) of G, computable in polynomial time, such that no atom of (H, η) has over $|G|/c$ vertices. Then $\mathcal{F}$ admits a quasi-polynomial time algorithm for* Independent Set.

Note that this theorem gives an exact algorithm, not an approximation, the reason why the algorithm of [38] is a QTPAS and not a quasi-polynomial time algorithm is due to a different step of their algorithm that is similar to Theorem 1.2.6.

Of course, not every hereditary graph class satisfies the premise of Theorem 1.2.7. Recall, though, that line graphs always have an extended strip decomposition where each atom is just a single vertex. Furthermore, it is well known that line graphs are claw-free. Hence, claw-free graphs are a generalization of line graphs. Similarly, $S_{t,t,t}$-free graphs ($S_{t,t,t}$ denotes a claw where each edge has been subdivided $t - 1$ times, so each arm of the claw has t vertices) contain the set of claw-free graphs. So, it is not too much of a stretch to hope that $S_{t,t,t}$-free graphs might have good extended strip decompositions. Unfortunately, this is false, but it is close to being true. In both [38] and [50], it is shown that after removing a small collection of vertices, the resulting graph has a good extended strip decomposition. In particular, in [50], the authors show the following

structural theorem.

Theorem 1.2.8 ([50])**.** *Let G be an $S_{t,t,t}$-free graph. Then there is a set X of size $\mathcal{O}(\log(n))$ such that $G - N[X]$ has a good extended strip decomposition.*

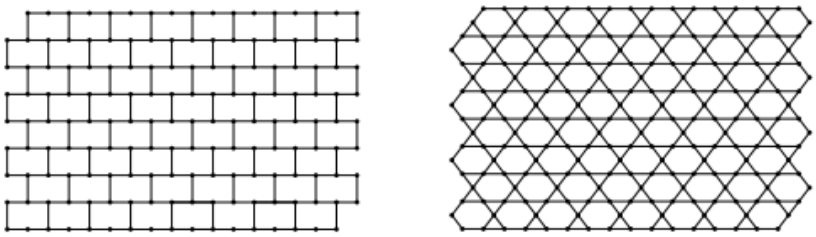
This theorem tells us that while $S_{t,t,t}$-free graphs do not have dominated balanced separators, they have something similar, which for Independent Set is just as good. In particular, in [38, 50], the following theorem is implicitly proved.

Theorem 1.2.9 ([38, 50])**.** *Let $\mathcal{F}$ be a graph class with a poly-log function f such that for every $G \in \mathcal{F}$, there is a set $|X| \leqslant f(|G|)$ such that $G - N[X]$ has a good extended strip decomposition. Then $\mathcal{F}$ admits a QPTAS for* Independent Set*.*

The proof of Theorem 1.2.9 essentially combines the proofs of Theorem 1.2.6 and 1.2.7.

Our path to giving a quasi-polynomial time algorithm for Independent Set on H-free graphs where H is a forest of paths and subdivided claws (Theorem 1.2.3) begins with our quasi-polynomial time algorithm for Independent Set on P_k-free graphs (Theorem 1.2.2). In particular, we will prove the following theorem.

This theorem will be presented in Chapter 3. The proof builds off the ideas from [37] that we sketched earlier, in particular, it uses the concept of dominated balanced separators to guide a branching process as was done in Theorem 1.2.5. But instead of being concerned with just a single dominated balanced separator as in the proof of Theorem 1.2.5, we must juggle around many dominated balanced separators at once and be more careful of our choices of vertices to branch on. Additionally, unlike Theorem 1.2.5, which works on any hereditary graph class with dominated balanced separators, our algorithm depends on the fact that the graph is P_k-free to run in quasi-polynomial time. Additionally, in Chapter 3 we show that to prove Conjecture 1.2.1 (up to quasi-polynomial



(a) (b)

Figure 1.2: (a) A wall graph. (b) The line graph of a wall graph.

time) it is sufficient to give a quasi-polynomial time algorithm for Independent Set on $S_{t,t,t}$-free graphs, so for the rest of this section we will focus more on $S_{t,t,t}$-free graphs and less on H-free graphs where H is a forest of paths and subdivided claws.

We face two roadblocks when trying to extend Theorem 1.2.2 to $S_{t,t,t}$-free graphs. The first and less serious roadblock we face is that the branching algorithm guided by dominated balanced separators for P_k-free graphs uses explicitly the fact that the graphs are P_k-free and does not immediately generalize to other graph classes with dominated balanced separators. However, with some new ideas, using dominated balanced separators to guide a branching algorithm that runs in quasi-polynomial time on $S_{t,t,t}$-free graphs is possible. A more significant roadblock, though, comes from the fact that $S_{t,t,t}$-free graphs do not necessarily *have* dominated balanced separators. In particular, the line graph of a wall (see Figure 1.2) is a claw-free graph (all line graphs are claw-free) and also does not have any balanced separator dominated by few vertices. Hence, an efficient branching algorithm guided by dominated balanced separators for $S_{t,t,t}$-free graphs seems useless as we may be unable to find dominated balanced separators. The results we recently discussed (in particular Theorem 1.2.7) along with the following theorem, which is more formally stated and proved in Chapter 4 saves us from this issue.

Theorem 1.2.10 ([3])**.** *For every fixed integer t, there exists an integer c_t and a polynomial-time algorithm that, given an n-vertex $S_{t,t,t}$-free graph G returns one of the following*

outcomes:

1. *a subset $X \subseteq V(G)$ of size at most $c_t \cdot \log(n)$ such that every component of $G - N[X]$ has at most $|G|/2$ vertices;*

2. *an extended strip decomposition of G where no atom has over $|G|/2$ vertices.*

Theorem 1.2.8 is a cornerstone in our proof of Theorem 1.2.10. At a very high level, the idea behind the proof of Theorem 1.2.10 is that if we have an $S_{t,t,t}$-free graph G with no dominated balanced separator then we apply Theorem 1.2.8 to get a set X of small size and a good extended strip decomposition, (H, η) of $G - N[X]$. Then, we incorporated the vertices of $N[X]$ one at a time into the extended strip decomposition, all while ensuring that it remains good.

Theorem 1.2.10 allows us to use the previously mentioned quasi-polynomial time algorithm for Independent Set on $S_{t,t,t}$-free graphs that have dominated balanced separators. The moment we come across an instance with a graph that does not have a dominated balanced separator, we can apply 1.2.10 to get a good extended strip decomposition, then use Theorem 1.2.7 to allow us to solve the problem on particles of substantially smaller size recursively. Thus allowing us to prove the following Theorem.

Theorem 1.2.11 ([3])**.** *$S_{t,t,t}$-free graphs admit a quasi-polynomial time algorithm for* Independent Set.

Our last result in the direction of solving Independent set on some hereditary graph class will come in Chapter 5, this work was done in [2]. There, we will provide an algorithm that extends Theorem 1.2.2 in two ways. First, it broadens the graph class that it can run on from P_k-free graphs to $C_{>k}$-free graphs (graphs with no induced cycles of length k or more). Second, the algorithm can solve an extensive set of problems, which includes Independent Set, but also many other interesting problems

such as Feedback Vertex Set, Maximum Weight Induced Planar Subgraph, Maximum Length Induced Path, and 3-Coloring. It was shown in [38] that $C_{>k}$-free graphs have dominated balanced separators, and our algorithm on $C_{>k}$-free graphs is still a branching-based algorithm that uses dominated balanced separators to guide the branching process, as was done for our algorithm for Independent Set on P_k-free graphs. Extending the branching algorithm for P_k-free graphs to work on $C_{>k}$-free graphs requires some new ideas, and getting the branching algorithm to also solve problems beyond just Independent Set requires substantially more work in the form of new ideas as well as using tools that lie in the intersection of logic and graph theory. In particular, we will prove the following Theorem.

Theorem 1.2.12 ([2])**.** *Fix two integers t and k and* CMSO_2 *sentence φ. Then there exists an algorithm that, given an n-vertex $C_{>t}$-free graph G, in quasi-polynomial time, finds a set $X \subseteq V(G)$ of maximum weight subject to the constraints that X has treewidth at most t and X satisfies φ. Additionally, $C_{>k}$-free graphs admit a quasi-polynomial time algorithm for* 3-Coloring.

Following the terminology of [51], in this Chapter, we will refer to the problem of finding an induced subgraph of maximum weight that satisfies the CMSO_2 statement, φ, and has treewidth at most k as the $(tw \leqslant k, \varphi)$-MWIS problem (where MWIS stands for maximum weight induced subgraph).

Treewidth and Monadic Second Order Logic The statement of Theorem 1.2.12 requires a few explanations, in particular, we need to define treewidth along with Monadic Second Order Logic (MSO_2) and is extension Counting Monadic Second Order Logic (CMSO_2). We begin with an informal discussion on treewidth.

Intuitively, the treewidth of a graph is a positive integer that measures how similar the graph is to a tree. A graph has treewidth one if and only if it is a forest (with at

least one edge). More formally a tree decomposition of a graph G is a pair (T, f) where T is a tree and f is a a function from the vertices of T to vertex sets of G that satisfies the following properties:

- $\bigcup_{v \in T} f(v) = V(G)$. That is, the image of f is $V(G)$.
- If $v \in G$ and $u, w \in T$ such that $v \in f(u)$ and $v \in f(w)$ then for any vertex z on the (unique) u, w path in T it holds that $v \in f(z)$.
- For every edge u, v of G, there is a vertex $w \in T$ such that $u, v \in f(w)$.

The treewidth of a tree decomposition (T, f) is defined to be $max_{v \in T}(|f(v)| - 1)$. The treewidth of a graph is the minimum treewidth taken over all tree decompositions of the graph. The -1 term is to ensure that the treewidth of a tree is one. Graphs of bounded treewidth have been extensively studied and are known to admit polynomial time algorithms for most problems researchers care about.

Next, we define MSO_2 and CMSO_2. MSO_2 and CMSO_2 are very closely related and only differ in a minor point in their definition, which will be pointed out at the appropriate time. CMSO_2 is a formal language for testing whether or not a graph (or a given substructure of a graph) has some property expressed by a CMSO_2 formula. Most standard graph properties can be expressed in CMSO_2 such as checking if a graph, G, is k-colorable for some fixed k, Hamiltonian, a given subset $S \subseteq V(G)$ is an independent set, if a graph is even-hole free, or if it is planar. The following is a CMSO_2 formula for determining if a subset $X \subseteq V(G)$ induces a connected subgraph. To verify the correctness of the following formula, we need a folkloric lemma; a set $X \subseteq V(G)$ induces a connected subset if and only if for all sets $Y \subseteq V(G)$ such that $X \cap Y \neq$ and $X \not\subseteq Y$ then there are vertices $x, y \in X$ such that $y \in Y$, $x \notin Y$ and $xy \in E(G)$. In other words, for every vertex set $Y \subseteq V(G)$ that contains some but not all of X, an edge of X crosses

the boundary of Y. Having to use roundabout definitions for standard graph properties like this is common with CMSO_2 formulas. We now present the CMSO_2 formula.

$$\mathbf{conn}(X) = \forall_{Y \subseteq V}[(\exists_{u \in X} u \in Y \wedge \exists_{v \in X} v \notin Y) \implies (\exists_{e \in E} \exists_{u \in X} \exists_{v \in X} \mathbf{inc}(u, e) \wedge \mathbf{inc}(v, e) \wedge u \in Y \wedge v \notin Y) \quad (1)$$

Let us unpack what is going on in this formula. Part of what makes up CMSO_2 formulas are *variables*. There are four types of variables: single vertices (in (1) this would be u and v), single edges (e), subsets of vertices (V, X, and Y), and subsets of edges (E). The variables given as part of the input are called *free variables* (a graph $G = (V, E)$ is always given as one of the free variables. However, it is not usually explicitly written as it is clear what G is from the context), in (1) X is a free variable. Variables that are not free are determined by the free variables, for example, the vertex set Y is determined by the vertex set of G.

CMSO_2 formulas are constructed inductively from smaller subformulas. The smallest building blocks are called *atomic formulas*, of which there are three. There is set membership checking, if u is a vertex (edge) variable and X is a vertex (edge) set variable, then we can write formula $u \in X$ which evaluates to true when u is an element of X. There is incidence checking, if u is a vertex variable and e is an edge variable, then we can write formula $\mathbf{inc}(u, e)$ which evaluates to true if the vertex u is incident to the edge e. There is equality checking, for any two variables x, y of the same type, we can write formula $x = y$ which evaluates to true when x and y are the same object. No equality checking occurs in the formula (1). Lastly, we can check if the size of a set is congruent to 0 modulo some number p. If X is a set and p is some number, we can write $|X| \equiv 0 \pmod{p}$, which evaluates to true when the size of X is congruent to 0 mod p. This last

atomic formula differentiates CMSO_2 from MSO_2 (it is allowed in CMSO_2 formulas, but not in MSO_2 formulas). No atomic formula of this type occurs in (1).

The four standard Boolean operators $\neg$, $\wedge$, $\vee$, and $\implies$ are used to assemble atomic formulas to make complex statements. $A \wedge B$ evaluates to true if and only if both A and B are true, $A \vee B$ evaluates to true as long as at least one of A and B is true. $\neg A$ evaluates to true if A is false and false if A is true. $v \notin Y$ is short hand for $\neg v \in Y$. $A \implies B$ evaluates to false when A is true and B is false, else it is true.

Lastly, we have quantifiers. CMSO_2 uses universal quantifiers (for all), $\forall$, and existential quantifiers (there exists), $\exists$. Supposed we have a formula $\forall_{Y \subseteq V} \Psi$ where Ψ is some subformula (this is precisely what occurs in (1)). This will evaluate to true if for all subsets $Y \subseteq V$, Ψ evaluates to true. Next, suppose we have $\exists_{u \in X} \Phi$. This evaluates to true if there exists some $u \in X$ such that Φ is true. For example, in (1), we have $\exists_{u \in X} u \in Y$, this evaluates to true when there exists $u \in X$ such that u also belongs to Y.

We can now fully understand what (1) is doing. Recall how we are determining if X induces a connected set. $X \subseteq V(G)$ induces a connected subset if and only if for all sets $Y \subseteq V(G)$ such that $X \cap Y \neq$ and $X \not\subseteq Y$ there are vertices $x, y \in X$ such that $y \in Y$, $x \notin Y$ and $xy \in E(G)$. (1) begins by stating $\forall_{Y \subseteq V} \Psi$, that is for all subsets Y of V the remainder of the formula, Ψ, must hold for (1) to be true. This matches the first part of our alternative definition for connectivity, where we state that for all $Y \subseteq V(G)$ "something" must hold. Next, we see an implication operator $\implies$. This means that for (1) to be true, it must hold that whenever

$$(\exists_{u \in X} u \in Y \wedge \exists_{v \in X} v \notin Y) \tag{2}$$

is true then

$$(\exists_{e\in E}\exists_{u\in X}\exists_{v\in X}\mathbf{inc}(u,e)\wedge\mathbf{inc}(v,e)\wedge u\in Y\wedge v\notin Y) \tag{3}$$

must be true.

Similarly, in our definition of connectivity, we require that if $X\cap Y\neq$ and $X\not\subseteq Y$ is true, then there must be vertices $x,y\in X$ such that $y\in Y$, $x\notin Y$ and $xy\in E(G)$ (here the terms "if" and "then there must be" plays the role of $\implies$). Now, (2) is true when there exists a $u\in X$ such that $u\in Y$ and a $v\in X$ such that $v\notin Y$. This is equivalent to the second part of our definition of connectivity, that $X\cap Y\neq\emptyset$ and $X\not\subseteq Y$. Lastly (3) is true when there is an edge $e\in E$ such that u and v are the endpoints of e (this is what $\exists_{e\in E}\exists_{u\in X}\exists_{v\in X}\mathbf{inc}(u,e)\wedge\mathbf{inc}(v,e)$ checks) and one of those endpoints, u is in Y and the other, v is not in Y (this is what $u\in Y\wedge v\notin Y$ checks). This is equivalent to the last part of our definition of connectivity, wherein there are vertices $x,y\in X$ such that $y\in Y$, $x\notin Y$ and $xy\in E(G)$.

We are now better positioned to understand the statement of Theorem 1.2.12. On $C_{>k}$-free graphs, we can solve the $(tw\leqslant k,\varphi)$-MWIS problem in quasi-polynomial time; that is, we can solve any problem that is expressible in CMSO_2 logic and has bounded treewidth in quasi-polynomial time. For instance since there are CMSO_2 formulas that can check if subset X of vertices of a graph G is an independent set, a forest, a planar graph, or an induced path and since independent sets have treewidth 0, forests have treewidth 1, planar graphs that are also $C_{>k}$-free have bounded treewidth, and induced paths have treewidth 1, Independent Set, Feedback Vertex Set (since Maximum Weight Forest is the compliment of Minimum Weight Feedback Vertex Set), Maximum Weight Induced Planar Subgraph, and Maximum Length Induced Path problems are all subcases of the $(tw\leqslant k,\varphi)$-MWIS problem.

We should note here that Theorem 1.2.9 can be extended to encompass not just

Independent Set but the *hereditary* $(tw \leqslant k, \varphi)$-MWIS problem (the hereditary $(tw \leqslant k, \varphi)$-MWIS problem requires that if φ evaluates to true when the input is a vertex set X, then φ also evaluates to true when the input is $X' \subseteq X$). This can be accomplished using similar techniques that we use to prove Theorem 1.2.12.

1.3 Minimal Separators

In the last two chapters of this thesis, Chapters 6 and 7, we turn our attention to *minimal separators*, these two chapters correspond to the papers [5] and [6]. Minimal separators are objects that can also be used to solve the $(tw \leqslant k, \varphi)$-MWIS problem. At first, it will seem like this is the only thing minimal separators have in common with dominated balanced separators, but we will later uncover a deeper connection between these two concepts.

Given a graph G and vertices $u, v \in G$, a set $S \subset V(G)$ is said to be a u, v-*separator* if u and v are in different components of $G - S$. S is a u, v-*minimal separator* if no proper subset of S is a u, v-separator. S is called a *minimal separator* if it is a u, v-minimal separator for some pair of vertices $u, v \in G$. We say that a graph class $\mathcal{F}$ is *tame* if there exists a polynomial p such that for all $G \in \mathcal{F}$, G has at most $p(|G|)$ minimal separators, quasi-tame if there exists a quasi-polynomial function p such that for all $G \in \mathcal{F}$, G has at most $p(|G|)$ minimal separators, and feral if there is a $c > 1$ such that there are arbitrarily large graphs $G \in \mathcal{F}$ such that G has at least $c^{|G|}$ minimal separators. A graph class is feral when an exponential bound on the number of minimal separators is the best that can be given. While many functions have growth rates between (quasi) polynomial and exponential, we will soon see that most typical graph classes are either (quasi) tame or feral.

A series of important results beginning with a paper from Bouchitte and Todinca [33]

and culminating with a paper from Fomin et al. [7] showed that the $(tw \leqslant k, \varphi)$-MWIS problem is solvable in time that is polynomial in the number of minimal separators of the graph. This means that tame graph classes admit a polynomial time algorithm for the $(tw \leqslant k, \varphi)$-MWIS problem, and quasi-tame graph classes admit a quasi-polynomial time algorithm for the $(tw \leqslant k, \varphi)$-MWIS problem. Many well-studied graph classes are tame, such as chordal graphs, permutation graphs, interval graphs, and P_4-free graphs, also known as cographs (we will soon sketch a proof that P_4-free graphs are tame). The results of Fomin et al. [7] on minimal separators provided a unifying theory as to why all of these graph classes admit polynomial time algorithms for problems like Independent Set, Feedback Vertex Set, and Maximum Length Induced Path.

Let us look at a quick proof that P_4-free graphs tame, in particular, we show they have $\mathcal{O}(n^2)$ minimal separators, although a more careful proof can show they have $\mathcal{O}(n)$ minimal separators. In this proof, we will use a folkloric lemma that a set S is a minimal separator if and only if at least two distinct components of $G - S$ dominate S (their neighborhood contains S). These components are called S-full components.

Let G be a P_4-free graph, and let S be any minimal separator of G. Let u be in one S-full component, call it C_u, and let v in a different S-full component, C_v. We claim that both u and v dominate S. Assume for a contradiction that u does not dominate S, let z be a non-neighbor of u in S. So there is an induced path P with its endpoints being u and z, all internal vertices are in C_u, and P contains at least three vertices. We can extend the induced path P by one more vertex by adding any neighbor of z in C_v (which must exist since C_v is S-full), this gives us a P_4, contrary to the assumption G is P_4-free. It follows that S is a subset of $N(u) \cap N(v)$, but if any vertex of $N(u) \cap N(v)$ is not in S, then C_u and C_v would not be two different components of $G - S$, hence $S = N(u) \cap N(v)$. So, every minimal separator of G is exactly the intersection of the neighborhood of some pair of vertices in G. Since there are $\mathcal{O}(n^2)$ pairs of vertices in G, G has $\mathcal{O}(n^2)$ minimal

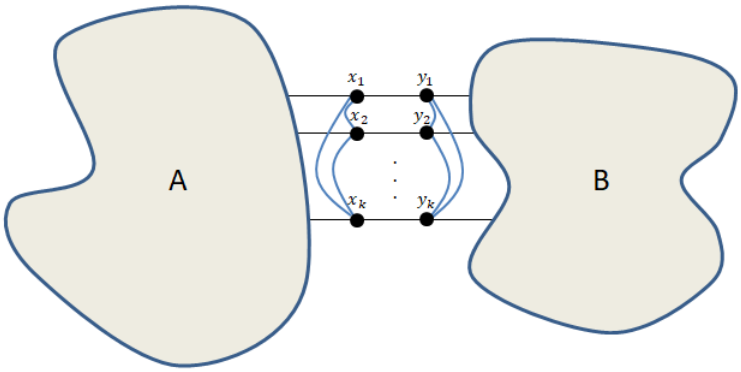


Figure 1.3: A k-creature. The blue edges indicate that x_i (y_i) may or may not be a neighbor of x_j (y_j)

separators.

More recently, researchers have become interested in gaining a more general understanding of what causes graph classes to be tame or feral. The first work to systemically study tame vs feral graph classes was by Milanic and Pivac [42], where they classified all graph classes that are defined by forbidden induced subgraphs of size four or less into tame vs feral. Another effort in this direction came from Abrishami et al. [43], where they proved that {theta, prism, pyramid, turtle}-free graphs are tame. In their paper, Abrishami et al. identified the presence/absence of a structure they called *k-creatures* as important in determining whether or not a graph class is tame or feral. Due to the importance of k-creatures in describing our results on this problem, we give the definition here.

Definition 1.3.1 (k-creatures (See Figure 1.3))**.** A graph G is said to be a *k-creature* if its vertices can be partitioned into sets A, $X = \{x_1, x_2, \ldots, x_k\}$, $Y = \{y_1, y_2, \ldots, y_k\}$, and B satisfying the following conditions:

1. $G[A]$ and $G[B]$ are connected.
2. A and $Y \cup B$ are anti-complete (i.e. $N[A] \cap (Y \cup B) = \emptyset$) and B and $A \cup X$ are anti-complete.

3. A dominates X (every vertex in X has a neighbor in A) and B dominates Y.
4. $x_i y_j$ is an edge if and only if $i = j$.

If a graph G does not contain an induced subgraph that is a k-creature, then we say that G is k-creature-free.

In a k-creature, it is easy to see that for a vertex $a \in A$ and $b \in B$, there are precisely 2^k minimal separators S which are disjoint from A and B. Such a separator S must pick exactly one vertex from each of $\{x_i, y_i\}$, and it can make each one of these k choices independently.

The aforementioned previous works essentially give different sufficient conditions for a graph to only have polynomially many separators. This naturally leads to the question *is there a "right" sufficient condition for tameness?* That is - a condition that, on one hand, is easy to state and verify, while on the other hand, captures all interesting tame graph classes. Abrishami et al. [43] conjectured that for every integer k, there exists a k' such that if an n-vertex graph G does not contain any k-creature as an induced subgraph, then G has at most $n^{k'}$ minimal separators. It turns out, we will show in Chapter 6 that this conjecture is false: for arbitrarily large n there exist n-vertex graphs that exclude 100-creatures and yet have $2^{\Omega(n)}$ minimal separators. Let us discuss, though, for a moment, why it would have been the "right" sufficient condition for polynomially many minimal separators if it had been true.

Towards this, we need to ask, which graph families $\mathcal{F}$ would have been tame, but whose tameness would not be captured by the conjecture of Abrishami et al. [43]? In simplest terms, it would be families $\mathcal{F}$ that are tame but that contain for every k an n-vertex graph G that contains a k-creature as an induced subgraph. Since k-creatures have at least 2^k minimal separators, and $\mathcal{F}$ is tame it must hold that $2^k \leqslant n^{\mathcal{O}(1)}$, meaning that $k = \mathcal{O}(\log n)$. In other words, the conjecture fails to capture tame graph classes

that contain k-creatures in graphs whose number of vertices is at least exponential in k.

This could happen for a few different reasons. One option, that we call Type 1, is that whenever a graph $G \in \mathcal{F}$ contains a k-creature then, G also contains some different piece of size exponential in k which is completely unrelated to the k-creature. The other option, which we call Type 2, is that whenever a graph $G \in \mathcal{F}$ contains a k-creature, then this k-creature itself has size exponential in k (meaning the sets A and B together have exponentially many vertices in k). Families of either one of these two types would have to be rather strange, although it is perfectly possible to construct artificial graph families of either type. For example, most interesting graph families are *hereditary*, that is, closed under vertex deletion. A hereditary family $\mathcal{F}$ cannot be Type 1 since whenever $\mathcal{F}$ contains a graph G that contains a k-creature, we can simply delete all the vertices not in the k-creature to obtain a k-creature which is in the family. Thus, if true, the conjecture of Abrishami et al. [43] would only fail to capture the tameness of hereditary classes of graphs whose every k-creature has size exponential in k.

As previously mentioned, we show in Chapter 6 that the conjecture of Abrishami et al. [43] is false. We will give a counterexample and show that a weaker version of the conjecture of Abrishami et al. is true. To state this result, we need a couple of definitions. First, a *k-skinny ladder* is a graph G consisting of two anti-complete paths $P_l = \ell_1\ell_2 \dots \ell_k$ and $P_r = r_1r_2 \dots r_k$ (subgraphs A and B are anti-complete if $V(A) \cap V(B) = \emptyset$ and there is no edge between A and B) and a set $\{s_1, s_2, \dots, s_k\}$ of vertices such that for every i, s_i is adjacent to ℓ_i and r_i and to no other vertices. Second, an *induced minor* of a graph G is a graph that can be obtained from G by deleting vertices and contracting edges (contracting an edge uv of a graph G results in a new graph G' where the vertices u and v has been removed and a new vertex w is added where w is made a neighbor of all vertices that was a neighbor of either u, v, or both u and v). The main result of Chapter 6 is the following weakening of the conjecture of Abrishami et al.

Theorem 1.3.2 ([5]). *For every integer k, the family of graphs that are k-creature free and exclude the k-skinny ladder as an induced minor is quasi-tame.*

Theorem 1.3.2 is strong enough to allow us to classify graph classes defined by a finite number of forbidden induced subgraphs. In particular, in Chapter 6, we will prove, using Theorem 1.3.2, the following theorem.

Theorem 1.3.3 ([5]). *Let $\mathcal{F}$ be a graph family defined by a finite number of forbidden induced subgraphs. If there exists a natural number k such that $\mathcal{F}$ forbids all k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, k-claw, and k-paw graphs, then $\mathcal{F}$ is quasi-tame. Otherwise, $\mathcal{F}$ is feral.*

While Theorem 1.3.2 is strong enough to give a complete dichotomy for graph classes defined by a finite number of forbidden induced subgraphs, it fails to be the "right" theorem we seek for two reasons. The first problem is that the bounds are quasi-polynomial as opposed to polynomial. The second is that there are still very reasonable tame graph classes not captured by this theorem, such as the graph class made up of k-skinny-ladders for all natural numbers k.

The first issue was recently dealt with by Gajarsky et al. [44] who, building heavily upon the work that will be presented in Chapter 6, the second issue is dealt with in Chapter 7. To properly state our main result of Chapter 7, we must introduce another graph called *t-critters*. First, we need a small generalization of minimal separators that applies to vertex sets instead of just vertices. Given $A, B \subset V(G)$, we define S to be an *A, B-separator* if $A \cap S = B \cap S = \emptyset$ and no component of $G - S$ contains a vertex from both A and B. An *A,B-minimal separator* is an A, B-separator such that no proper subset of S is an A, B-separator.

Definition 1.3.4 (t-critter partition, t-critter). A *t-critter partition* of a graph G is a

partition of the vertex set of G into sets $A_1, A_2, \dots A_{t+1}$, $B_1, B_2, \dots, B_{t+1}, X_1, X_2, \dots, X_t$, such that the following conditions are satisfied.

1. For all $1 \leqslant i, j \leqslant t+1$ with $i \neq j$, A_i is anti-complete with A_j, B_i is anti-complete with B_j, and A_i is anti-complete with B_j.
2. For all $1 \leqslant i \leqslant t+1$ A_i and B_i is connected.
3. The vertices of A_i, A_{i+1}, B_i, and B_{i+1} are the only vertices outside of X_i that have a neighbor in X_i.
4. There are (at least) two distinct $(A_i \cup A_{i+1})$, $(B_i \cup B_{i+1})$-minimal separators in $G[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i]$, S_1^i and S_2^i, such that there is a path from A_i to A_{i+1} through both $X_i - S_1^i$ and $X_i - S_2^i$ and there is a path from B_i to B_{i+1} through both $X_i - S_1^i$ and $X_i - S_2^i$.

A graph G is a *t-critter* if G has a t-critter partition. A graph G is *t-critter free* if G does not contain a t-critter as an induced subgraph.

The definition of t-critters is arguably technical and unappealing. A reader who is familiar with minimal separators should, after staring at the definition for a bit, be able to convince themselves that just as for k-creatures, for every vertex $a \in A = \bigcup_i A_i$ and vertex $b \in B = \bigcup_i B_i$ there are at least 2^t minimal a,b-separators in G disjoint from $A \cup B$. In particular, to separate a from b, we may, for every $i \leqslant t$, choose to delete either S_1^i or S_2^i, and for every i the choice between S_1^i or S_2^i can be made independently of the other choices (see Lemma 7.6.13). We are now ready to state the main theorem of Chapter 7.

Theorem 1.3.5 ([6]). *For every pair of integers t, k the family of k-creature free and t-critter free graphs is quasi-tame.*

The upper bound in Theorem 1.3.5 on the number of minimal separators is $n^{k' \log^{17}(n)}$ where k' is a constant that depends only on k and t. The proof of Theorem 1.3.5 contains some interesting ingredients, from a VC-dimension-based lemma from Chapter 6, to a "greedy branching" procedure inspired by our quasi-polynomial time algorithm for Independent Set on P_k-free graphs of Chapter 3, to covering-packing dualities [52] and Ramsey- and Erdös-Szekers [53] type arguments.

Theorem 1.3.5 is yet another sufficient condition for a graph to have few minimal separators. To boot, the condition is very technical, and the upper bound in the number of minimal separators is an ugly quasi-polynomial function. What makes *this* sufficient condition for an upper bound for the number of minimal separators special? What makes it special is that it is the *right* sufficient condition, in the way that the conjecture of Abrishami et al. [43] would have been right if only it had been correct. But don't take our word for it - we prove this in a precise and technical sense in Chapter 7.

Let us apply the same litmus test for Theorem 1.3.5 as we did for the conjecture of Abrishami et al. [43], and ask for which tame graph families $\mathcal{F}$ are not captured by Theorem 1.3.5? Again, there could be families of Type 1, namely families that do contain graphs G that contain k-critters or k-creatures for every k, but such graphs G always also have at least $2^{\Omega(k)}$ additional unrelated vertices. Such families cannot be hereditary, and so, if we restrict attention to hereditary families, the only tame families that Theorem 1.3.5 does not capture are Type 2 families that do contain k-critters or k-creatures for every k, but these k-critters or k-creatures have at least $2^{\Omega(k)}$ vertices.

It is, in fact, possible to construct such hereditary families. It is even possible to construct tame families that contain k-critters for every k, and yet are closed under induced minors, disproving a prior conjecture of ours. However, all such families are pretty artificial. The next theorem shows that they *have* to be artificial.

We previously discussed Monadic Second Order Logic (MSO) and its extension,

Counting Monadic Second Order Logic (CMSO). Their main claim to fame in graph algorithms is probably from Courcelle's Theorem [54], which states that every MSO-definable family of graphs can be recognized in linear time on graphs of bounded treewidth. The overwhelming majority of interesting graph families can be expressed in Counting Monadic Second Order Logic, this includes all graph classes with a finite number of forbidden minors, induced minors, topological minors, induced subgraphs or subgraphs, and several other classes such as bipartite, or perfect. We show that if we restrict attention to CMSO-definable hereditary properties, then the sufficient condition of Theorem 1.3.5 is also necessary.

Theorem 1.3.6 ([6])**.** *Let $\mathcal{F}$ be a CMSO-definable hereditary graph family. If there exists an integer k such that no graph in $\mathcal{F}$ contains a k-creature nor a k-critter, then $\mathcal{F}$ is quasi-tame. Otherwise, $\mathcal{F}$ is feral.*

The proof of Courcelle's theorem [54] establishes that CMSO-definable graph classes have many properties in common with regular languages. This has been exploited with great success in graph algorithms [45, 55], however, to the best of our knowledge, it has never been used to prove a purely structural result such as Theorem 1.3.6

The proof of Theorem 1.3.6 is based on a "pumping lemma" style argument that shows that a k-creature or k-critter on n vertices can be "pumped" to a $k \cdot x$-critter on $n \cdot x$ vertices, thereby demonstrating that in every CMSO-definable hereditary family that contains k-creatures or k-critters for arbitrarily large k, there exist k-creatures or k-critters in the family with only $O(k)$ vertices.

As we recently mentioned, part of the proof of Theorem 1.3.5 takes inspiration from our branching algorithm for Independent Set on P_k-free graphs. In the next section, we will see a bit more about why this works, and we will propose a conjecture that explains how dominated balanced separators generalize and extend minimal separators.

1.4 Induced Grid Minor Conjecture

Recall our definition of a *balanced separator*, it is a vertex set S of a graph G such that no component of $G - S$ contains over half of G's vertices, and S is a *dominated balanced separator* if it can be dominated by few vertices (poly-log in the size of G). We saw in Theorem 1.2.6 that hereditary graph classes with dominated balanced separators admit a QPTAS for Independent Set and more generally (as we noted at the end of the paragraph on CMSO_2), the hereditary $(tw \leqslant k, \varphi)$-MWIS problem. Additionally, we have mentioned that dominated balanced separators are a valuable tool for developing exact quasi-polynomial time algorithms for Independent Set and, more generally, the $(tw \leqslant k, \varphi)$-MWIS problem. This naturally leads to the question, which graph classes have dominated balanced separators? As well as the closely related question, which graph classes admit an efficient algorithm for Independent Set and the $(tw \leqslant k, \varphi)$-MWIS problem?

We formally define the $k \times k$ grid as the graph with vertex set $\{v(i,j) : 1 \leqslant i, j \leqslant k \text{ for } i, j \in \mathbb{N}\}$ and edge set $\{v(i,j)v(i',j') : |i - i'| + |j - j'| \leqslant 1\}$. Note that a $k \times k$ grid does not have balanced separators that are dominated by fewer than $k/3$ vertices, so hereditary graph classes that contain all $k \times k$ grids and an induced minor do not have dominated balanced separators. We believe that grids are the only induced minor obstruction to having dominated balanced separators. More specifically, we make the following conjecture.

Conjecture 1.4.1 ((Structural) Induced Grid Minor Conjecture)**.** *There exists a function f such that for all graphs G and integers k, if G does not contain a $k \times k$ grid as an induced minor then G has a balanced separator dominated by $f(k)$ vertices.*

Additionally, it can be shown that Independent Set is NP-Hard on the graph class that contains for all integers k, all subdivisions of the $k \times k$ grid. We believe that grids are

the only induced minor obstruction to efficient algorithms for not just Independent Set but for the $(tw \leqslant k, \varphi)$-MWIS problem, which includes Independent Set along with many other interesting problems like Feedback Vertex Set and Maximum Length Induced Path. This leads us to our second conjecture.

Conjecture 1.4.2 ((Algorithmic) Induced Grid Minor Conjecture)**.** *Let k be an integer and $\mathcal{F}$ a hereditary graph class such that no graph in $\mathcal{F}$ contains a $k \times k$ grid as an induced minor. Then $\mathcal{F}$ admits a polynomial time algorithm for* 3-Coloring *and the $(tw \leqslant k, \varphi)$-MWIS problem.*

It is interesting to note that whenever we have a $k \times k$ grid as an induced minor in a graph, we also have either a subdivision of a $\Omega(k) \times \Omega(k)$ wall or the line graph of subdivision of a $\Omega(k) \times \Omega(k)$ wall as an induced subgraph (See Figure 1.2). We could have even reworded these conjectures in terms of induced wall minors. Still, we chose to use induced grid minors because of these conjectures' natural connection with the classic Grid Minor Theorem.

We mentioned previously that as long as the balanced separators are dominated by some poly-log function of vertices that is usually enough to still give quasi-polynomial time results that we desire such as in Theorem 1.2.6 (and there will be a few places in this thesis where we are only able to prove poly-log bounds and not constant bounds), so a weakened but still interesting variation of Conjecture 1.4.1 would replace "has balanced separators dominated by $f(k)$ vertices" with "has balanced separators dominated by $f(k)\cdot$poly-log(n) vertices".

This is not the first time that it has been asked if Independent Set has an efficient algorithm on graphs that do not have some fixed grid as an induced minor, it appears to have been first asked by Dallard et al. [56]. But Conjecture 1.4.1 has, to the best of our knowledge, not been made by other researchers and is the first conjecture that tells

us *why* Independent Set should have an efficient algorithm on these graphs (because we believe they have dominated balanced separators). Additionally, it is, to the best of our knowledge, the first time that it has been conjectured that the $(tw \leqslant k, \varphi)$-MWIS problem has an efficient algorithm on graphs that do not have some fixed grid as an induced minor.

For multiple decades now, researchers have been studying problems like Independent Set, Feedback Vertex Set, and 3-Coloring on graph classes such as P_k-free graphs, $C_{>k}$-free graphs, graph with no induced cycles of even length, better known as even-hole-free graphs (and more general graph classes such as {theta, prism}-free graphs and graphs that forbid induced cycles of length 0 mod x where x is any positive integer) and tame graph classes. Except for tame graph classes, little progress had been made. Indeed, before this thesis, whether or not Independent Set on P_7-free graph was NP-Hard was still open. Conjectures 1.4.1 and 1.4.2 tie together these different areas of research as subproblems of the same overarching question as well as giving a promising line of attack for solving these problems (it might not be clear yet how tame graph classes are a subproblem of these conjectures, we will discuss this soon).

What evidence do we have of Conjectures 1.4.1 and 1.4.2? We have already seen a proof of Conjecture 1.4.1 when restricted to P_k-free graphs in this introduction. In Chapter 7, we will also see a proof of this conjecture for k-creature-free graphs. Additionally, it is known that this conjecture is true when restricted to $C_{>k}$-free graphs [38], graphs that are $S_{t,t,t}$-free and do not contain the line graph of any subdivision of the $t \times t$ wall for some fixed positive integer t as an induced subgraph [50], and there are two manuscripts in preparation which show that this conjecture holds for even-hole-free graphs as well as graphs that are {theta, pyramid}-free and do not contain the line graph of any subdivision of the $t \times t$ wall as an induced subgraph for some fixed positive integer t . For some of these results, Conjecture 1.4.1 is only known to hold with poly-log bounds

and not constant bounds as in the formal statement of the conjecture. For all of these cases where there are poofs of Conjecture 1.4.1, we have been able to use the existence of dominated balanced separators to prove Conjecture 1.4.2 for these graph classes (at least up to quasi-polynomial runtime). In this thesis, we will see algorithms solving the $(tw \leqslant k, \varphi)$-MWIS problem for P_k, $C_{>k}$, and k-creature free graph classes.

Let us now briefly discuss potential applications of Conjecture 1.4.1. In addition to giving a QPTAS for Independent Set on induced grid minor free graphs and being an essential ingredient to resolving 1.4.2, a positive resolution of Conjecture 1.4.1 may lead to other consequences as well. The well-known Grid Minor Theorem of Robertson and Seymour has played a central role in many interesting theorems about minor-closed graph classes, and we now have a very deep understanding of minor-closed graph classes, much of this do in some way or another to the Grid Minor Theorem. On the other hand, our knowledge of hereditary and induced minor-closed graph classes is much weaker. Readers familiar with the Grid Minor Theorem and treewidth should notice a very strong connection between the Grid Minor Theorem and our Induced Grid Minor Conjecture, so it isn't unreasonable to think that a positive resolution to Conjecture 1.4.1 could lead us to a better understanding of hereditary and induced minor-closed graph classes, as the Grid Minor Theorem did for minor-closed graph classes.

Let us give a few concrete examples of how a positive resolution of Conjecture 1.4.1 could help us gain a deeper understanding of hereditary and induced minor-closed graph classes.

The first is a sort of "induced" variation of Menger's Theorem, which has captured the interest of researchers recently [57, 16, 58]. There are many different variations that this can come in, the one we will concern ourselves with is the following. Given a graph class $\mathcal{F}$, we wish to prove there exists a function f such that for any graph $G \in \mathcal{F}$ and vertex sets $A, B \subseteq V(G)$ such that there are at most k pairwise anti-complete paths

from A and B (the paths are vertex disjoint and there is no edge between any two of the paths) there is a set X such that $|X| \leqslant f(k)$ and no component of $G - N[X]$ has vertices from both A and B. For graph classes with dominated balanced separators, we sketch a proof of a slight weakening of this "induced" Menger's Theorem. Note that in the following theorem, we require that we have dominated balanced separators for vertex sets in a graph, that is, for every graph G and every vertex set $C \subseteq V(G)$, we can find a set, X of small size (at most poly-log) such that no component of $G - N[X]$ has over $|C|/2$ vertices of C. We call X a dominated balanced separator for C in G. For every graph class we know of that has dominated balanced separators, weighted versions of dominated balanced separators like this have always come with only minor modifications to the proofs. Alternatively, we could add many leaves to the vertices of C to simulate giving the vertices of C "all of the weight" in the graph, assuming the graph class is closed under this leaf adding operation, then a dominated balanced separator for G would need to be a dominated balanced separator for C in G.

Theorem 1.4.3. *Let $\mathcal{F}$ be a hereditary graph class and f a function such that for all $G \in \mathcal{F}$, and all subsets $X \subseteq V(G)$, G has a balanced separator for X dominated by $f(|G|)$ vertices. Let $A, B \subseteq V(G)$ with at most k anti-complete paths from A to B. Then there is a vertex set $Y \subseteq V(G)$ of size at most $\mathcal{O}(f(|G|) \cdot k \cdot log(|G|))$ such that no component of $G - N[Y]$ has vertices from both A and B.*

Proof: [sketch] We prove by induction on $|A \cup B|$ that there is an A-B separator dominated by at most $\mathcal{O}(f(|G|) \cdot k \cdot \log(|A \cup B|))$ vertices. Take a balanced separator S for $A \cup B$ dominated by $f(|G|)$ vertices. Let $C_1, \ldots, C_t$ be the connected components of $G - S$ and let y_i be the maximum number of anti-complete A-B paths in C_i. Then the sum of y_i's can be at most k. In each component of $G - S$, $A \cup B$ is smaller by a factor of at least two. So, by induction on each component C_i we find an A-B

separator dominated by $\mathcal{O}(f(|G|) \cdot y_i \cdot \log(|A \cup B|/2))$ vertices. In total this is at most $\mathcal{O}(f(|G|) \cdot k \cdot \log(|A \cup B|) - f(|G|) \cdot k)$ vertices, so if we add in S (which was dominated by $f(|G|)$ vertices) then we found an A-B separator dominated by $\mathcal{O}(f(|G|) \cdot k \cdot \log(|A \cup B|))$ vertices as desired. ■

Another application of Conjecture 1.4.1 is as follows. Recently, there has been much work by several graph theorists in trying to understand what induced subgraphs must be forbidden to force graphs to have bounded treewidth [59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69]. In the bounded degree case, which has also generated interest recently, balanced separators dominated by a constant number of vertices are equivalent to balanced separators of constant size. Since having bounded treewidth is equivalent to having balanced separators of constant size (for any weighting of the vertices), a weighted variant of Conjecture 1.4.1 (which we also believe to be true) then predicts that in the bounded degree case, forbidding a $k \times k$ grid as an induced minor for some fixed value of k (or equivalently, forbidding all subdivisions of the $k \times k$ wall and the line graphs of all subdivisions of the $k \times k$ wall as an induced subgraph for some fixed k) forces the graph to have bounded treewidth. This was proven to be true by Korhonen [69].

Readers familiar with treewidth would notice immediately that to guarantee bounded treewidth, cliques and complete bipartite graphs must be forbidden (K_n and $K_{n,n}$), as well as all subdivisions of $k \times k$ walls and the line graphs of all subdivisions of $k \times k$ walls. It is reasonable to ask if forbidding these graphs as induced subgraphs would be sufficient to force graphs to have bounded treewidth, but several counterexamples to such a question have been found [68]. We have already seen that Conjecture 1.4.1 implies (if true) that forbidding all subdivision of $k \times k$ walls and the line graphs of all subdivisions of $k \times k$ walls as induced subgraphs for some fixed integer k implies the graph class has dominated balanced separators. So, if Conjecture 1.4.1 is proven to be true, the goal of understanding what induced subgraphs must be forbidden to force graphs to

have bounded treewidth can be reduced to understanding what graphs that forbid all subdivision of $k \times k$ walls and the line graphs of all subdivisions of $k \times k$ walls as induced subgraphs for some value of k (and therefore have dominated balanced separators) but also have high treewidth look like. This approach is similar to the one we took in Chapter 7 to characterize quasi-tame and feral graph classes. We proved that all k-creature free graph classes have dominated balanced separators (a structure that we know forces a graph to have exponentially minimal separators), then we asked, "what do k-creature free graphs (which have dominated balanced separators) with many minimal separators look like"? From there, we could show that such graph classes must contain k-critters for arbitrarily large values of k. This ends our discussion of potential applications of the Induced Grid Minor Conjecture.

We now look at how dominated balanced separators can be viewed as a generalization of minimal separators and that the key properties of minimal separators are captured in Conjectures 1.4.1 and 1.4.2. In particular, these two conjectures tell us that all tame graph classes have dominated balanced separators and efficient algorithms for the $(tw \leqslant k, \varphi)$-MWIS problem.

Recall that any graph that contains a k-creature as an induced subgraph has at least 2^k minimal separators. It follows that for every tame graph class $\mathcal{F}$ (or even quasi-tame graph class), there must be some quasi-polynomial function h such that no graph $G \in \mathcal{F}$ has an $h(|G|)$-creature. It is straightforward to verify that a $k \times k$ grid is a k-creature, and furthermore, any graph that contains a $k \times k$ grid as an induced minor contains a k-creature as an induced subgraph. Hence, no graph $G \in \mathcal{F}$ has a $h(|G|) \times h(|G|)$ grid as an induced minor. As long as the function f from Conjecture 1.4.1 is a polynomial (which would be consistent with our knowledge), then this conjecture would imply that (quasi) tame graph classes have dominated balanced separators. We do, in fact, show in Chapter 7 that every k-creature free graph contains a balanced separator dominated by

at most $2k$ vertices, so (quasi) tame graph classes have dominated balanced separators.

Identically, since for a (quasi) tame graph class $\mathcal{F}$, there must be some quasi-polynomial function h such that no graph $G \in \mathcal{F}$ has an $h(|G|) \times h(|G|)$ grid as an induced minor, Conjecture 1.4.2 tells us that we should be able to solve the $(tw \leqslant k, \varphi)$-MWIS problem in quasi-polynomial time. Admittedly, our current form of Conjecture 1.4.2 does not quite say this, but only a small modification would be needed, in particular only forbidding $h(|G|) \times h(|G|)$ grids as an induced minor as apposed to $k \times k$ grids for some fixed k and allowing the runtime to be quasi-polynomial now to compensate for the larger grid sizes permitted. In Chapter 7, we do use the presence of dominated balanced separators in $h(|G|)$-creature-free graphs to devise a branching algorithm that solves the $(tw \leqslant k, \varphi)$-MWIS problem in quasi-polynomial time for $h(|G|)$-creature-free graphs and hence for (quasi) tame graph classes.

Next, we turn our attention to two more related conjectures that apply to an even more general class of graphs. Recall one of the main results of this thesis is a quasi-polynomial time algorithm for Independent Set on H-free graphs where H is a forest of paths and subdivided claws. When we were discussing this result we noted that such graph classes do not have dominated balanced separators, our example was that these graph classes always contain the line graph of arbitrarily large walls (which have a large grid as an induced minor). It follows that this graph class does not satisfy the premise of Conjectures 1.4.1 and 1.4.2. This leads us to look for a natural extension of Conjectures 1.4.1 and 1.4.2 that can include graph classes like $S_{t,t,t}$-free graphs. Unfortunately, most problems, like Feedback Vertex Set, included in the $(tw \leqslant k, \varphi)$-MWIS problem are known to be NP-Hard even on claw-free graphs. So any extension of Conjecture 1.4.2 that includes claw-free graphs should only deal with the Independent Set problem, and not the more general $(tw \leqslant k, \varphi)$-MWIS problem.

Theorem 1.2.8 tells us that $S_{t,t,t}$-free graphs have a property that is not too far away

from having dominated balanced separators, in particular, it tells us that there exists a poly-log function f such that for every $S_{t,t,t}$-free graph G, there is a set $X \subseteq V(G)$ with $|X| \leqslant f(|G|)$ such that $G - N[X]$ has a good extended strip decomposition. As Theorem 1.2.9 states, this property is enough to get a QPTAS for Independent Set. It is simple to show that if S is a balanced separator, then $G - S$ has a good extended strip decomposition. Hence, graph classes with dominated balanced separators also have small sets X such that $G - N[X]$ has a good extended strip decomposition.

So, we search for reasonable conjectures that extend the graph classes Conjectures 1.4.1 and 1.4.2 to also include H-free graphs where H is a forest of paths and subdivided claws. From what we just discussed in the previous paragraph, it is intuitive that these conjectures would utilize extended strip decompositions somehow. According to Conjecture 1.4.1 (and the discussion immediately after the conjecture), if we forbid all subdivisions of a $k \times k$ wall and all the line graphs of all subdivisions of a $k \times k$ wall as induced subgraphs, the graphs should have dominated balanced separators. Furthermore, we already saw that line graphs always have good extended strip decompositions (recall an extended strip decomposition is good if each particle has at most say $n/2$ vertices), and more generally extended strip decompositions handle line graphs well. So, the presence of the line graph of a subdivision of some large wall should not pose an obstacle to finding a good extended strip decomposition. Hence, as long as we forbid all subdivisions of a $k \times k$ wall as an induced subgraph, it seems reasonable to expect that we can find a set X of small size such that $G - N[X]$ has a good extended strip decomposition. More precisely, we make the following two conjectures.

Conjecture 1.4.4. *There exists a function f such that given any integer k, if G does not contain a subdivision of a $k \times k$ wall as an induced subgraph, then there is a set $X \subseteq V(G)$ such that $|X| \leqslant f(k)$ and $G - N[X]$ has an extended strip decomposition*

where each particle has at most $|G|/2$ vertices and can be found in polynomial time.

Conjecture 1.4.5. *Let k be an integer and $\mathcal{F}$ be a hereditary graph class such that no graph in $\mathcal{F}$ contains a subdivision of a $k \times k$ wall as an induced subgraph. Then $\mathcal{F}$ admits a polynomial time algorithm for* Independent Set.

These two conjectures now include all of the graph classes that Conjectures 1.4.1 and 1.4.2 included along with H-free graphs where H is a forest of paths and subdivided claws, as well as other interesting graph classes such as theta-free graphs. We mentioned in the statement of Theorem 1.2.9 that as long as the set X from 1.4.4 is at most some poly-log function of the number of vertices of G, then that is enough to get a QPTAS for Independent Set, and there are in fact some cases like for $S_{t,t,t}$-free graphs where that bound is as good as we currently know how to do. So, a weakened but still useful variation on Conjecture 1.4.4 would replace "$|X| \leqslant f(k)$" with "$|X| \leqslant f(k) \cdot$ poly-log(n)".

What evidence do we have of Conjectures 1.4.4 and 1.4.5? Admittedly, it appears that we are further away from solving these conjectures than we are from solving Conjectures 1.4.1 and 1.4.2 (in fact, it can be shown without much trouble that a positive resolution of Conjecture 1.4.4 implies a positive resolution of Conjecture 1.2.1). We saw that Theorem 1.2.8 proved Conjecture 1.4.4 when restricted to $S_{t,t,t}$-free graphs, and it is easy to extend this result to prove this conjecture when limited to H-free graphs, where H is a forest of paths and subdivided claws (with quasi-polynomial bounds). This conjecture is also known to be true when restricted to {theta, pyramid}-free graphs (along with all graph classes Conjecture 1.4.1 is known to hold). In these graph classes Conjecture 1.4.2 also holds as well (with quasi-polynomial run time), with one of the most important results of this thesis being a proof of Conjecture 1.4.2 (with quasi-polynomial run time) when restricted to H-free graphs where H is a forest of paths and subdivided claws.

1.5 Organization of Chapters

Each chapter in this thesis will have a local preliminaries section. Additionally, there is a Preliminaries chapter (Chapter 2.4). Notation and definitions that are used in multiple chapters will be given in the Chapter 2.4, notations and definitions that are unique to a given chapter will go into the preliminaries section of that chapter.

In Chapter 3, we will give a quasi-polynomial time algorithm for Independent set on P_k-free graphs. Building off this work in Chapter 4, we give a quasi-polynomial time algorithm for Independent Set on H-free graphs, where H is a forest of paths and subdivided claws. In Chapter 5, we again build off the work of Chapter 3 to give a quasi-polynomial time algorithm for the $(tw \leqslant k, \varphi)$-MWIS problem on $C_{>k}$-free graphs.

In Chapters 6 and 7, we turn our attention to minimal separators. The main result of Chapter 6 is a characterization of quasi-tame and feral graph classed defined by a finite number of forbidden induced subgraphs. The main result of Chapter 7 is a characterization of all quasi-tame graph classes that meet some weak conditions. Specifically, we show that every graph which excludes certain graphs called *k-creatures* and *k-critters* as induced subgraphs has at most quasi-polynomially many minimal separators. We then demonstrate that this sufficient condition for having few minimal separators is the "right" one. In particular, we show that every hereditary graph class $\mathcal{F}$ definable in CMSO_2 logic that contains k-creatures or k-critters for every k is feral.

Chapter 2

Preliminaries

All graphs in this thesis are assumed to be simple, undirected graphs unless otherwise stated. We denote the edge set of a graph G by $E(G)$ and the vertex set of a graph by $V(G)$. For a vertex v, by $N_G(v)$ we denote the set of neighbors of v, and by $N_G[v]$ we denote the set $N_G(v) \cup \{v\}$. For a set $X \subseteq V(G)$, we also define $N_G(X) := \bigcup_{v \in X} N_G(v) - X$, and $N_G[X] = N_G(X) \cup X$. If it does not lead to confusion, we omit the subscript and write simply $N(\cdot)$ and $N[\cdot]$. Additionally, if G' is an induced subgraph of G, we use $N_G^{G'}(X)$ and $N_G^{G'}[X]$ to mean $N_G(X) \cap V(G')$ and $N_G[X] \cap V(G')$ respectively. We use $\mathcal{CC}(G)$ to denote the set of connected components of G. If G_1, G_2,..., G_m are graphs, then we use $G_1 + G_2 + ... + G_m$ to denote the graph that that has vertex set $V(G_1) \cup V(G_2) \cup ... \cup V(G_m)$, and edge set $E(G_1) \cup E(G_2) \cup ... \cup E(G_m)$.

Given a weight function $w : V(G) \to \mathbb{N}$ the weight of a vertex set S is defined as $w(S) = \sum_{v \in S} w(v)$. An *independent set* in G is a vertex set S such that no pair of vertices in S have an edge between them. We define $\mathsf{mwis}(G)$ to be the weight of the maximum weight independent set in G. The *length* of a path is the number of vertices in the path and we denote by P_k the path of length k. If $X \subseteq V(G)$ then we will use $G(X)$ to denote the the graph induced by the vertex set X, and if it is clear from the context we will use

$G - X$ to denote the graph $G(V(G) - X)$.

For a family $\mathcal{Q}$ of sets, by $\bigcup \mathcal{Q}$ we denote $\bigcup_{Q \in \mathcal{Q}} Q$. Let G be a graph. For $X \subseteq V(G)$, by $G[X]$ we denote the subgraph of G induced by X, i.e., $(X, \{uv \in E(G) : u, v \in X\})$. If the graph G is clear from the context, we will often identify induced subgraphs with their vertex sets. The sets $X, Y \subseteq V(G)$ are *complete* to each other if for every $x \in X$ and $y \in Y$ the edge xy is present in G. Note that this, in particular, implies that X and Y are disjoint. We say that two sets X, Y *touch* if $X \cap Y \neq \emptyset$ or there is an edge with one endpoint in X and another in Y. Finally, two disjoint sets are *anti-adjacent* or *anti-complete* if they do not touch.

Given a graph G, we say a vertex set $C \subseteq V(G)$ is a *connected vertex set* if $G[C]$ is a connected graph. A *walk* in a graph G is a sequence $v_1, v_2, \ldots, v_\ell$ of vertices in G such that each pair of consecutive vertices in the sequence are adjacent. The *length* of a walk $v_1, v_2, \ldots, v_\ell$ is the number ℓ of *vertices* in the walk. A walk whose first vertex is v_1 and last is v_ℓ is a walk *from* v_1 *to* v_ℓ. The vertex v_1 is called the first vertex of the walk, v_ℓ the last. All other vertices are *internal* vertices. A walk $v_1, v_2, \ldots, v_\ell$ where all vertices are distinct is a *path*. A path $P = v_1, v_2, \ldots, v_\ell$ is an *induced* path if there are no edges between v_i and v_j whenever $|i - j| > 1$. For three disjoint vertex sets A, B, C a walk (or path, or induced path) *from* A *to* B *through* C is a walk (or path, or induced path) whose first vertex is in A, last vertex is in B, and all internal vertices (if any) are in C.

The length of a path and cycle is the number of edges of the path. P_t denotes an induced path with t vertices (and $t - 1$ edges). C_t denotes an induced cycle of length t. A claw is a set of three independent vertices, v_1, v_2, and v_3 along with a a vertex u that is neighbors with each v_i. An $S_{t,t,t}$ is three anti-complete P_t's along with a vertex u that is neighbors with exactly one endpoint from each P_t and no other vertices, so a claw is $S_{1,1,1}$.

Given a graph G and a graph H, G is said to be *H-free* if G does not contain H as

an induced subgraph. If $\mathcal{H}$ is a set of graphs, then G is $\mathcal{H}$-free if for each $H \in \mathcal{H}$, G is H-free. By $C_{>t}$-free we mean a graph that does not have any induced cycle of length greater than t. By $T(G)$, we denote the set of all triangles in G. Similarly to writing $xy \in E(G)$, we write $xyz \in T(G)$ to indicate that $G[\{x, y, z\}] \simeq K_3$.

Minimal Separators Given a graph G, a non-empty set $S \subset V(G)$ is called a *separator* if there are at least two distinct components L and R of $G - S$. If $u \in L$ and $v \in R$ then we call S a *u-v-separator* or a *u, v-separator*. S is a *u, v-minimal separator* if S is a u, v-separator and no proper subset of S is a u, v-separator, or equivalently, if $N_G(L) = N_G(R) = S$. This equivalence is folkloric and easy to show. If C is a component of $G - S$ such that $N_G(C) = S$, then we say that C is an *S-full component*. Similarly, given $A, B \subset V(G)$ we define S to be an *A, B-separator* if $A \cap S = B \cap S = \emptyset$ and no component of $G - S$ contains a vertex from both A and B. An *A,B-minimal separator* is an A, B-separator such that no proper subset of S is an A, B-separator.

A family of graphs $\mathcal{F}$ is called *tame* if there exists a constant c such that for all $G \in \mathcal{F}$, G has at most $|V(G)|^c$ minimal separators. A family of graphs $\mathcal{F}$ is called *strongly-quasi-tame* if there exists a constant c such that for all $G \in \mathcal{F}$, G has at most $|V(G)|^{c \log(|V(G)|)}$ minimal separators. A family of graphs $\mathcal{F}$ is called $feral$ if there exists a constant $c > 1$ such that for all natural numbers N there exists a $G \in \mathcal{F}$, such that $|V(G)| = n > N$ and G has at least c^n minimal separators.

2.1 Extended strip decompositions.

Now let us define a certain graph decomposition which will play an important role in the paper. An *extended strip decomposition* of a graph G is a pair (H, η) that consists of:

- a simple graph H,
- a *vertex set* $\eta(x) \subseteq V(G)$ for every $x \in V(H)$,
- an *edge set* $\eta(xy) \subseteq V(G)$ for every $xy \in E(H)$, and its subsets $\eta(xy, x), \eta(xy, y) \subseteq \eta(xy)$,
- a *triangle set* $\eta(xyz) \subseteq V(G)$ for every $xyz \in T(H)$,

which satisfy the following properties (also see fig. 2.1):

1. The family $\{\eta(o) \mid o \in V(H) \cup E(H) \cup T(H)\}$ is a partition of $V(G)$.
2. For every $x \in V(H)$ and every distinct $y, z \in N_H(x)$, the set $\eta(xy, x)$ is complete to $\eta(xz, x)$.
3. Every $uv \in E(G)$ is contained in one of the sets $\eta(o)$ for $o \in V(H) \cup E(H) \cup T(H)$, or is as follows:
   - $u \in \eta(xy, x), v \in \eta(xz, x)$ for some $x \in V(H)$ and $y, z \in N_H(x)$, or
   - $u \in \eta(xy, x), v \in \eta(x)$ for some $xy \in E(H)$, or
   - $u \in \eta(xyz)$ and $v \in \eta(xy, x) \cap \eta(xy, y)$ for some $xyz \in T(H)$.

An extended strip decomposition (H, η) is *rigid* if for every $xy \in E(H)$, the sets $\eta(xy)$, $\eta(xy, x)$, and $\eta(xy, y)$ are nonempty, and for every isolated $x \in V(H)$, the set $\eta(x)$ is nonempty.

We say that a vertex $v \in V(G)$ is *peripheral* in (H, η) if there is a degree-one vertex x of H, such that $\eta(xy, x) = \{v\}$, where y is the (unique) neighbor of x in H. For a set $Z \subseteq V(G)$, we say that (H, η) is an *extended strip decomposition of* (G, Z) if H has $|Z|$ degree-one vertices and each vertex of Z is peripheral in (H, η).

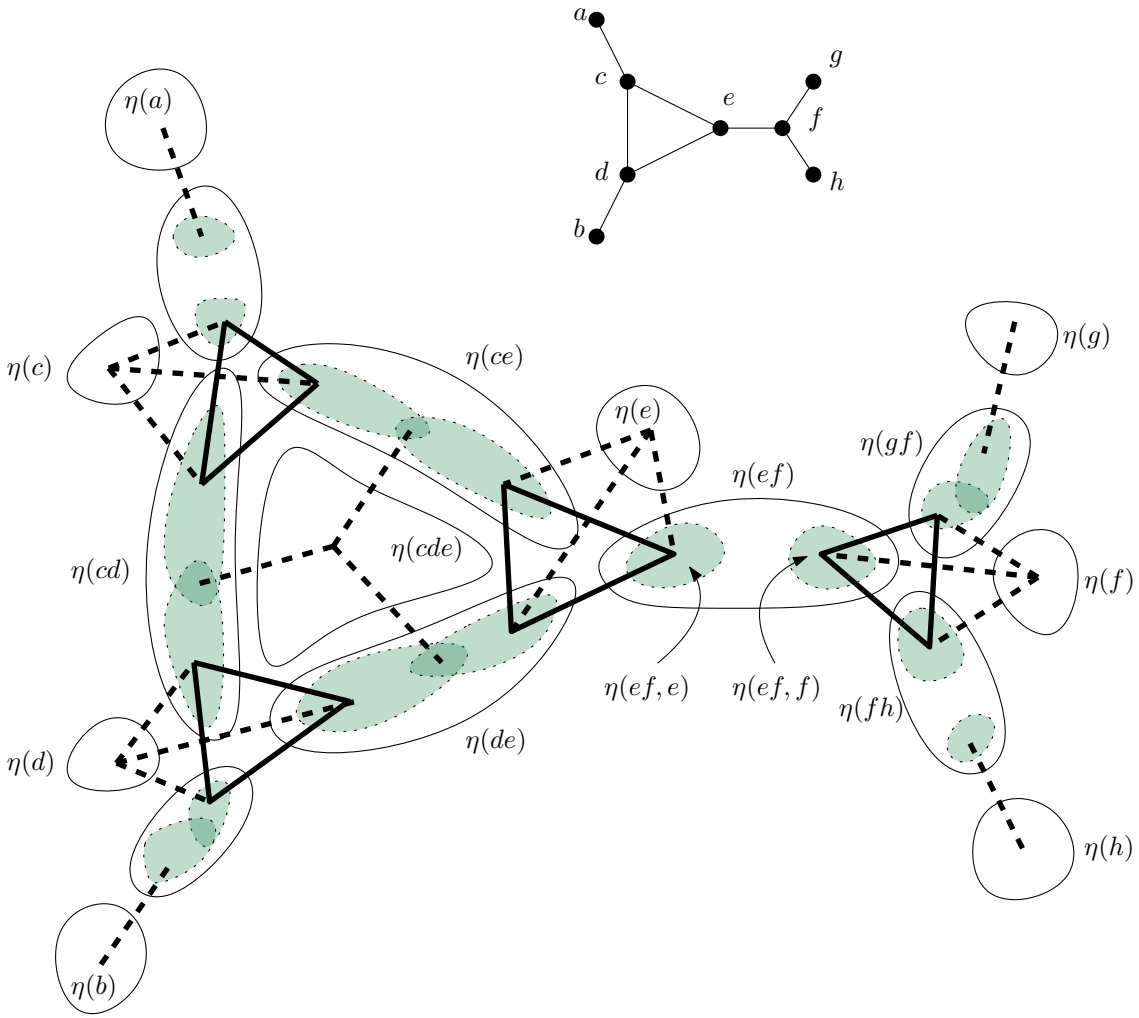


Figure 2.1: A graph H and an extended strip decomposition (H, η) of some graph G. Edges within sets $\eta(\cdot)$ are arbitrary. A solid edge across two sets indicates that there are complete to each other. A dashed edge means that edges across these sets are allowed but not mandatory. No edge means that the sets do not touch.

The following theorem by Chudnovsky and Seymour [39] is a slight strengthening of their celebrated solution of the famous *three-in-a-tree* problem. We will use it as a black-box to build extended strip decompositions.

Theorem 2.1.1 ([39, Section 6])**.** *Let G be an n-vertex graph and consider $Z \subseteq V(G)$ with $|Z| \geqslant 2$. There is an algorithm that runs in time $\mathcal{O}(n^5)$ and returns one of the following:*

- *an induced subtree of G containing at least three elements of Z, or*
- *a rigid extended strip decomposition (H, η) of (G, Z).*

Let us point out that actually, an extended strip decomposition produced by Theorem 2.1.1 satisfies more structural properties, but for our purpose, we will only use the fact that it is rigid.

Particles of extended strip decompositions. Let (H, η) be an extended strip decomposition of a graph G. We introduce some special subsets of $V(G)$ called *particles*, divided into five *types*.

$$\begin{aligned}
\text{vertex particle:} &\quad A_x := \eta(x) \text{ for each } x \in V(H),\\
\text{edge interior particle:} &\quad A^{\perp}_{xy} := \eta(xy) - (\eta(xy,x) \cup \eta(xy,y)) \text{ for each } xy \in E(H),\\
\text{half-edge particle:} &\quad A^{x}_{xy} := \eta(x) \cup \eta(xy) - \eta(xy,y) \text{ for each } xy \in E(H),\\
\text{full edge particle:} &\quad A^{xy}_{xy} := \eta(x) \cup \eta(y) \cup \eta(xy) \cup \bigcup_{z \,:\, xyz \in T(H)} \eta(xyz) \text{ for each } xy \in E(H),\\
\text{triangle particle:} &\quad A_{xyz} := \eta(xyz) \text{ for each } xyz \in T(H).
\end{aligned}$$

For an positive integer p, we write $[p] \coloneqq \{1, \ldots, p\}$. For a set A, $\binom{A}{p}$ denotes the set of all p-element subsets of A.

We say that two vertex subsets $X_1, X_2 \subseteq V(G)$ are *adjacent* if either $X_1 \cap X_2 \neq \emptyset$ or there is an edge $x_1x_2 \in E(G)$, such that $x_1 \in X_1$ and $x_2 \in X_2$. Otherwise, the sets are *nonadjacent*.

For a path P, the *length* of P is the number of edges of P. For a graph G, the *radius* of G is $\min_{v \in V(G)} \max_{u \in V(G)} \mathsf{dist}(u, v)$, where $\mathsf{dist}(u, v)$ denotes the *distance* between u and v, i.e., the length of a shortest u-v-path in G.

2.2 Graph minors

Let H be a graph. A *minor model* of H in a graph G is a mapping η that assigns to each $v \in V(H)$ a connected subgraph $\eta(v)$ of G so that:

- the subgraphs $\{\eta(v) \colon v \in V(H)\}$ are pairwise disjoint; and
- for every edge $v_1v_2 \in E(H)$, there is an edge in G with one endpoint in $\eta(v_1)$ and the other in $\eta(v_2)$.

Such a minor model η has *depth* t if every subgraph $\eta(v)$ has radius at most t. We say that G contains H as a (depth-t) minor if there is a (depth-t) minor model of H in G.

A *topological minor model* of H in G is a mapping ψ that assigns to each vertex $v \in V(H)$ a vertex $\psi(v)$ in G and to each edge $e \in E(H)$ a path $\psi(e)$ in G so that

- vertices $\psi(v)$ are pairwise different for different $v \in V(H)$;
- for each edge $v_1v_2 \in E(H)$, the path $\psi(v_1v_2)$ has endpoints $\psi(v_1)$ and $\psi(v_2)$ and does not pass through any of the vertices of $\{\psi(v) \colon v \in V(H)\}$ other than $\psi(v_1)$ and $\psi(v_2)$; and

- paths $\{\psi(e)\colon e \in E(H)\}$ are pairwise disjoint apart from possibly sharing endpoints.

The vertices $\{\psi(v)\colon v \in V(H)\}$ are the *roots* of the topological minor model ψ. We say that ψ has *depth* t if each path $\psi(e)$ for $e \in E(H)$ has length at most $2t+1$. We say that G contains H as a (depth-t) topological minor if there is a depth-t topological minor model of H in G. It is easy to see that if G contains H as a depth-t topological minor, then it also contains H as a depth-t minor.

2.3 Treewidth and tree decompositions

A *tree decomposition* of a graph G is a pair (T, β), where T is a tree and β is a function that maps every vertex of T to a subset of $V(G)$, such that the following properties hold:

- every edge of G is contained in $\beta(a)$ for some $a \in V(T)$, and
- for every $v \in V(G)$, the set $\{a \in V(T) \mid v \in \beta(a)\}$ is nonempty and induces a subtree of T.

The sets $\beta(a)$ for $a \in V(T)$ are called the *bags* of the decomposition (T, β). The *width* of the decomposition (T, β) is $\max_{a \in V(T)} |\beta(a)| - 1$ and the *treewidth* of a graph is the minimum possible width of its decomposition.

We will need the following well-known observation about graphs of bounded treewidth.

Lemma 2.3.1 (see Lemma 7.19 of [70])**.** *Let H be a graph of treewidth less than k and let $A \subseteq V(H)$. Then there exists a set $X \subseteq V(H)$ of size at most k such that every connected component of $H - X$ has at most $|A|/2$ vertices of A.*

2.4 MSO_2 and MSO_2 types

We assume the incidence encoding of graphs as relational structures: a graph G is encoded as a relational structure whose universe consists of vertices and edges of G (each distinguishable by a unary predicate), and there is one binary incidence relation binding each edge with its two endpoints. With this representation, the MSO_2 logic on graphs is the standard MSO logic on relational structures as above, which boils down to allowing the formulas to use vertex variables, edge variables, vertex set variables, and edge set variables, together with the ability of quantifying over them. For a positive integer p, the $\mathsf{C}_{\leqslant p}\mathsf{MSO}_2$ logic extends MSO_2 by allowing atomic formulas of the form $|X| \equiv a \bmod m$, where X is a set variable and $0 \leqslant a < m \leqslant p$ are integers; denote $\mathsf{CMSO}_2 \coloneqq \bigcup_{p>0} \mathsf{C}_{\leqslant p}\mathsf{MSO}_2$. The *quantifier rank* of a formula is the maximum number of nested quantifiers in it.

For a finite set Λ of labels, a Λ-*boundaried graph* is a pair (G, ι) where G is a graph and $\iota\colon B \to \Lambda$ is an injective function for some $B \subseteq V(G)$; then $B = \mathrm{dom}\iota$ is called the *boundary* of (G, ι). A k-*boundaried graph* is a shorthand for a $[k]$-boundaried graph, where we denote $[k] = \{1, \ldots, k\}$. For $v \in B$, $\iota(v)$ is the *label* of v. We define two operations on Λ-boundaried graphs.

If (G_1, ι_1) and (G_2, ι_2) are two Λ-boundaried graphs, then the result of *gluing* them is the boundaried graph $(G_1, \iota_1) \oplus_\Lambda (G_2, \iota_2)$ that is obtained from the disjoint union of (G_1, ι_1) and (G_2, ι_2) by identifying vertices of the same label (so that the resulting labelling is again injective).

If (G, ι) is a Λ-boundaried graph and $l \in \Lambda$, then the result of *forgetting* l is the Λ-boundaried graph $(G, \iota|_{\iota^{-1}(\Lambda - \{l\})})$. That is, if l belongs to the range of ι, we remove the label from the corresponding vertex (but we keep this vertex in G).

By $\mathsf{C}_{\leqslant p}\mathsf{MSO}_2$ on Λ-boundaried graphs we mean the $\mathsf{C}_{\leqslant p}\mathsf{MSO}_2$ logic over graphs en-

riched with $|\Lambda|$ unary predicates: for each label $l \in \Lambda$ we have a unary predicate that selects the only vertex with label l, if existent. Effectively, this boils down to the possibility of using elements of the boundary as constants in $\mathsf{C}_{\leqslant p}\mathsf{MSO}_2$ formulas.

The following folklore proposition explains the compositionality properties of CMSO_2 on boundaried graphs. The statement and the proof is standard, see e.g. [71, Lemma 6.1], so we only sketch it.

Proposition 2.4.1. *For every triple of integers k, p, q, there exists a finite set $\mathsf{Types}^{k,p,q}$ and a function that assigns to every k-boundaried graph (G, ι) a* type $\mathsf{type}^{k,p,q}(G, \iota) \in \mathsf{Types}^{k,p,q}$ *such that the following holds:*

1. *The types of isomorphic graphs are the same.*

2. *For every $\mathsf{C}_{\leqslant p}\mathsf{MSO}_2$ sentence φ on k-boundaried graphs, whether (G, ι) satisfies φ depends only on the type $\mathsf{type}^{k,p,q}(G, \iota)$, where q is the quantifier rank of φ. More precisely, there exists a subset $\mathsf{Types}^{k,p,q}[\varphi] \subseteq \mathsf{Types}^{k,p,q}$ such that for every k-boundaried graph (G, ι) we have*

$$(G, \iota) \models \varphi \qquad \text{if and only if} \qquad \mathsf{type}^{k,p,q}(G, \iota) \in \mathsf{Types}^{k,p,q}[\varphi].$$

3. *The types of ingredients determine the type of the result of the gluing operation. More precisely, for every two types $\tau_1, \tau_2 \in \mathsf{Types}^{k,p,q}$ there exists a type $\tau_1 \oplus_{k,p,q} \tau_2$ such that for every two k-boundaried graphs (G_1, ι_1), (G_2, ι_2), if $\mathsf{type}^{k,p,q}(G_i, \iota_i) = \tau_i$ for $i = 1, 2$, then*

$$\mathsf{type}^{k,p,q}((G_1, \iota_1) \oplus_{[k]} (G_2, \iota_2)) = \tau_1 \oplus_{k,p,q} \tau_2.$$

Also, the operation $\oplus_{k,p,q}$ is associative and commutative.

4. *The type of the ingredient determines the type of the result of the forget label operation. More precisely, for every type* $\tau \in \mathsf{Types}^{k,p,q}$ *and* $l \in [k]$ *there exists a type* $\tau_{\neg l}$ *such that for every* k*-boundaried graph* (G, ι)*, if* $\mathsf{type}^{k,p,q}(G, \iota) = \tau$ *and* (G', ι') *is the result of forgetting* l *in* (G, ι)*, then*

$$\mathsf{type}^{k,p,q}(G', \iota') = \tau_{\neg l}.$$

Proof: [sketch] It is well-known that there are only finitely many syntactically non-equivalent $\mathsf{C}_{\leqslant p}\mathsf{MSO}_2$ sentence over k-boundaried graphs and of quantifier rank at most q, so let $\mathsf{Sentences}^{k,p,q}$ be a set containing one such sentence from each equivalence class. We set $\mathsf{Types}^{k,p,q}$ as the power set (set of all subsets) of $\mathsf{Sentences}^{k,p,q}$. To each k-boundaried graph (G, ι) we define $\mathsf{type}^{k,p,q}(G, \iota) \subseteq \mathsf{Sentences}^{k,p,q}$ as the set of all sentences $\psi \in \mathsf{Sentences}^{k,p,q}$ satisfied in (G, ι). Thus, whether (G, ι) satisfies φ can be decided by verifying whether $\mathsf{type}^{k,p,q}(G, \iota)$ contains a sentence that is syntactically equivalent to φ. The remaining two assertions — about compositionality of the gluing and the forget label operations — follow from a standard argument using Ehrenfeucht-Fraïsse games. ■

In our algorithms, we shall work with a fixed CMSO_2 sentence φ over graphs with treewidth upper-bounded by a fixed constant k. Note that φ belongs to $\mathsf{C}_{\leqslant p}\mathsf{MSO}_2$ for a fixed constant p, and the quantifier rank of φ is a fixed constant q. Hence, when working in k-boundaried graphs, whether φ is satisfied in a k-boundaried graph (G, ι) can be read from its type $\mathsf{type}^{k,p,q}(G, \iota)$. To facilitate the computation of types, we shall assume that our algorithms have a hard-coded set of types $\mathsf{Types}^{k,p,q}$, together with the subset $\mathsf{Types}^{k,p,q}[\varphi] \subseteq \mathsf{Types}^{k,p,q}$ and functions

$$(\tau_1, \tau_2) \mapsto \tau_1 \oplus_{k,p,q} \tau_2 \qquad \text{and} \qquad (\tau, l) \mapsto \tau_{\neg l},$$

as described in proposition 2.4.1. Also, the algorithms have hard-coded the types of all k-boundaried graphs with $\mathcal{O}(k)$ vertices.

We will need also a simple observation that CMSO_2 types preserve connectivity properties, as being in the same connected component can be easily expressed in CMSO_2.

Lemma 2.4.2. *Fix integers $k \geqslant 0$, $p \geqslant 0$, and $q \geqslant 4$, and suppose that (G_1, ι_1) and (G_2, ι_2) are two k-boundaried graphs with $\mathsf{type}^{k,p,q}(G_1, \iota_1) = \mathsf{type}^{k,p,q}(G_2, \iota_2)$. Then the ranges of ι_1 and ι_2 are equal. Furthermore, for every two pairs $(u_1, u_2), (v_1, v_2) \in V(G_1) \times V(G_2)$ with $\iota_1(u_1) = \iota_2(u_2)$ and $\iota_1(v_1) = \iota_2(v_2)$, vertices u_1 and v_1 are in the same connected component of G_1 if and only if vertices u_2 and v_2 are in the same connected component of G_2.*

Proof: That the ranges of ι_1 and ι_2 are equal is clear: G_1 and G_2 satisfy the same sentences of the form "there exists a vertex with label $i \in [k]$". The assertion about having the same connectivity between boundary vertices follows from an analogous argument, supplied with the observation that being in the same connected component can be expressed by an MSO_2 formula of quantifier rank four:

$$\begin{aligned}\mathsf{Connected}(x,y) \;=\; & \neg\exists_{A\subseteq V(G)}(x \in A) \wedge (y \notin A) \wedge \\ & \big(\forall_{e\in E(G)}\forall_{u\in V(G)}\forall_{v\in V(G)}(\mathrm{inc}(u,e) \wedge \mathrm{inc}(v,e)) \Rightarrow ((u \in A) \Leftrightarrow (v \in A))\big).\end{aligned}$$

■

In our proofs, we will need to keep track of the exact value of the treewidth of a constructed induced subgraph of the given graph. For this, we use the following observation.

Lemma 2.4.3. *For every integer k there exists an MSO_2 sentence $\varphi_{\mathrm{tw}<k}$ such that $G \models \varphi_{\mathrm{tw}<k}$ if and only if the treewidth of G is less than k.*

Proof: Let $\mathcal{F}_{\mathrm{tw}<k}$ be the set of all minor-minimal graphs of treewidth at least k. By the Robertson-Seymour Theorem [72], $\mathcal{F}_{\mathrm{tw}<k}$ is finite. Define $\varphi_{\mathrm{tw}<k}$ to be the conjunction, over all $H \in \mathcal{F}_{\mathrm{tw}<k}$, of sentences asserting that H is not a minor of G. Note here that it is straightforward to express in MSO_2 that a fixed graph H is a minor of a given graph G. ■

Chapter 3

Independent Set on P_k-Free Graphs in Quasi-Polynomial Time

In this chapter we present an algorithm that takes as input a graph G with weights on the vertices, and computes a maximum weight independent set S of G. If the input graph G excludes a path P_k on k vertices as an induced subgraph, the algorithm runs in time $n^{O(k^2 \log^3 n)}$. Hence, for every fixed k our algorithm runs in quasi-polynomial time. This resolves in the affirmative an open problem of [Thomassé, SODA'20 invited presentation]. Previous to this work, polynomial time algorithms were only known for P_4-free graphs [Corneil et al., DAM'81], P_5-free graphs [Lokshtanov et al., SODA'14], and P_6-free graphs [Grzesik et al., SODA'19]. For larger values of t, only $2^{O(\sqrt{kn \log n})}$ time algorithms [Bascó et al., Algorithmica'19] and quasi-polynomial time approximation schemes [Chudnovsky et al., SODA'20] were known. Thus, our work is the first to offer conclusive evidence that Independent Set on P_k-free graphs is not NP-complete for any integer k.

Additionally we show that for every graph H, if there exists a quasi-polynomial time algorithm for Independent Set on C-free graphs for every connected component C of H, then there also exists a quasi-polynomial time algorithm for Independent Set on

H-free graphs. This lifts our quasi-polynomial time algorithm to T_k-free graphs, where T_k has one component that is a P_k, and $k-1$ components isomorphic to a *fork* (the unique 5-vertex tree with a degree 3 vertex).

3.1 Introduction

An *independent set* (also known as a *stable set*) in a graph G is a vertex set S such that no pair of distinct vertices in S are adjacent in G. In the Independent Set problem the input is a graph G on n vertices and integer k, the task is to determine whether G contains an independent set S of size at least k. Independent Set is a well-studied and fundamental graph problem which is NP-complete [8, 9] and intractable within most frameworks for coping with NP-hardness. Indeed, Independent Set was one of the very first problems to be shown to be NP-hard to approximate [10, 11], one of the first intractable problems from the perspective of parameterized complexity [45], one of the first problems to be shown not to have a $2^{o(n)}$ time algorithm assuming the Exponential Time Hypothesis (ETH) [46], and one of the very first problems whose hardness of parameterized approximation, assuming the Gap-ETH, was established [47].

With the above in mind, it is natural that a significant research effort has been devoted to identifying classes of input graphs for which the Independent Set problem is substantially easier than on general graphs. Of particular interest are the classes where Independent Set becomes polynomial time solvable. Most famously the problem becomes polynomial time solvable on Perfect graphs [13], other examples of polynomial time solvable cases include $k \times K_2$-free graphs [73] and graphs of bounded cliquewidth [74]. For an extensive list, see [75] and the companion website [76]. On the other hand the problem remains NP-complete even on planar graphs of maximum degree 3 [8], unit disc graphs [77], triangle-free graphs [78] and AT-free graphs [79].

This chapter fits in a long line of work to precisely classify the complexity of Independent Set on all *hereditary* graph classes defined by a single forbidden induced subgraph H (and more generally, by a finite set $\mathcal{H}$ of forbidden induced subgraphs). A graph G is said to be *H-free* if G does not contain a copy of H as an induced subgraph.

For a set $\mathcal{H}$ of graphs, G is $\mathcal{H}$-*free* if G is H-free for all $H \in \mathcal{H}$. The ultimate goal of this research direction is to establish a dichotomy theorem that for every finite set $\mathcal{H}$ of graphs determines whether Independent Set on $\mathcal{H}$-free graphs is polynomial time solvable, or NP-complete[1].

In 1982 Alekseev [4] observed that Independent Set remains NP-complete on the class of $\mathcal{H}$-free graphs for every finite set $\mathcal{H}$ that does not include a graph H whose every connected component is a path or a subdivision of the claw ($K_{1,3}$). Since then, no new NP-completeness results for Independent Set on $\mathcal{H}$-free graphs have been found for any other finite set $\mathcal{H}$. Thus, it is consistent with current knowledge that Independent Set is polynomial time solvable on $\mathcal{H}$-free graphs for all other finite sets $\mathcal{H}$. At the same time, progress on algorithms has been embarrassingly slow. The only connected graphs H for which NP-completness of Independent Set does not follow from Alekseev's result are paths and subdivisions of the claw. Polynomial time algorithms for Independent Set on claw-free graphs were found independently by Sbihi [80] and Minty [81] in 1980. A polynomial time algorithm on *fork*-free graphs (a fork is a claw with one subdivided edge) was found by Alekseev [17]. Subsequently, Lozin and Milanic [18] gave an algorithm for Weighted Independent Set on fork-free graphs. For paths, Independent Set on P_4-free graphs was shown to be polynomial time solvable by Corneil et al. [82] in 1981. After a series of papers giving polynomial time algorithms for various subclasses of P_5-free graphs [83, 84, 85, 28, 86, 87], in 2014 Lokshtanov et al. [14] gave a polynomial time algorithm on P_5 free graphs. Two years later, Lokshtanov et al. [88] devised a quasi-polynomial time algorithm on P_6-free graphs, before Grzesik et al. [89] designed a polynomial time algorithm for P_6-free graphs in 2019. This summarizes the state-of-the-art for polynomial time solvability of Independent Set on H-free graphs.

[1]There is of course the possibility that Independent Set on $\mathcal{H}$-free graphs has NP-intermediate complexity for some choice of $\mathcal{H}$. We believe this is unlikely, however that is pure speculation.

It appears that the currently known techniques are very far from being able to yield polynomial time algorithms for Independent Set on P_k-free graphs for $k = 8$, let alone for all values of k. More concretely, the polynomial time algorithms for P_5-free graphs of Lokshtanov et al. [14] and for P_6-free graphs of Grzesik et al. [89] are based on the same method. First, from a sample of two articles the complexity of applying this method seems to grow exponentially with k. Second, and more importantly, in a recent manuscript Grzesik et al. [90] show that a crucial component of this method fails completely on P_k-free graphs for $k \geqslant 8$.

The slow progress on polynomial time algorithms have prompted researchers to look for weaker forms of tractability of Independent Set on P_k-free graphs. Bacsó et al. [37] provided $2^{O(\sqrt{kn \log n})}$ time algorithms for Independent Set on P_k-free graphs (see also [91, 92]). Finally, Chudnovsky et al. [38] obtained quasi-polynomial time approximation schemes for P_k-free graphs for all k. In fact their result is much more general - they obtain quasi-polynomial time approximation schemes on $\mathcal{H}$-free graphs for all sets $\mathcal{H}$ for which NP-hardness does not follow from Alekseev's [4] observation. While the results above are general, they are consistent with Independent Set being NP-complete on all $\mathcal{H}$-free classes of graphs on which polynomial time algorithms are not already known. In this chapter we obtain a quasi-polynomial time algorithm for Weighted Independent Set on P_k-free graphs for every k. In particular we prove the following theorem.

Theorem 3.1.1. *There exists an algorithm that given a graph G and weight function $w : V(G) \to \mathbb{N}$ outputs the weight of a maximum weight independent set of G. If G is P_k-free then the algorithm runs in $n^{O(k^2 \log^3 n)}$ time.*

Theorem 3.1.1 implies that unless NP $\subseteq$ QP, Independent Set on P_k-free graphs is not NP-complete for any k. This is the first conclusive evidence against NP-completeness for any $k \geqslant 7$. The running time of the algorithm of Theorem 3.1.1 matches that of

Chudnovsky et al. [38], but computes optimal solutions instead of $(1-\varepsilon)$-approximate ones. It is also worth mentioning that our algorithm and analysis is substantially simpler than the quasi-polynomial time algorithm of Lokshtanov et al. [88] for the special case of P_6-free graphs. We have been unsuccessful in generalizing Theorem 3.1.1 to a quasi-polynomial time algorithms for H-free graphs where H is a subdivision of a claw. However, the techniques used to prove Theorem 3.1.1 can be used to show that such an algorithm would automatically generalize to all classes of $\mathcal{H}$-free graphs for which NP-hardness is not already known. More concretely, for a graph H let O_H be an oracle that takes as input an H-free graph G and outputs the weight of a maximum weight independent set in G. Further, let $\mathcal{CC}(H)$ denote the set of connected components of H. Our second result is the following.

Theorem 3.1.2. *There exists an algorithm that given as input a graph H, a graph G, and weight function $w : V(G) \to \mathbb{N}$ as well as access to oracles $O(H_i)$ for all $H_i \in \mathcal{CC}(H)$, outputs the weight of a maximum weight independent set of G. If G is H-free then the algorithm uses at most $n^{O(|H|^2|\mathcal{CC}(H)|\log^3(n))}$ operations and oracle calls on induced subgraphs of G.*

Theorem 3.1.2 has two immediate consequences. First, coupled with Theorem 3.1.1 and the polynomial time algorithm for Weighted Independent Set on fork-free graphs, Theorem 3.1.2 yields a quasi-polynomial time algorithm for Weighted Independent Set on T_k-free graphs, where T_k is the graph with k connected components the first of which is a P_k and each of the remaining $k-1$ are isomorphic to a fork. Second, Theorem 3.1.2 implies that if Weighted Independent Set has a quasi-polynomial time algorithm on H-free graphs for every subdivided claw H, then Weighted Independent Set also has a quasi-polynomial time algorithm on all $\mathcal{H}$-free classes of graphs, for finite sets $\mathcal{H}$, for which NP-hardness does not follow from Alekseev's result. Or, stated

more poetically, the buck stops at the (subdivided) claw.

Methods. The starting point for our algorithm is the $2^{O(\sqrt{n \log n})}$ time algorithm for P_k-free graphs of Bascó et al. [37]. The algorithm of Bascó et al. [37] is simple enough that we can give a quite detailed overview here. It combines two methods - "*degree reduction*" and "*balanced separation*".

The "degree reduction" approach can be summarised as follows. As long as the input graph G contains a vertex v of sufficiently high degree (degree $\geqslant d$) then *branch on* v. That is, find the best solution avoiding v by a recursive call on $G - v$, and the best solution containing v by adding v to the solution obtained from a recursive call on $G - N[v]$. Output the best of these two solutions. A simple recurrence analysis shows that this reduces the problem to solving $2^{O(\frac{n \log n}{d})}$ instances in which no vertex has degree at least d. Bascó et al. [37] set $d = \sqrt{n \log n}$ and obtain $2^{O(\sqrt{n \log n})}$ instances with maximum degree $\sqrt{n \log n}$.

The "balanced separation" technique is based on the classic "Gyárfás path" argument [36] for proving that P_k-free graphs are χ-bounded. A simple lemma (Lemma 2 of Bascó et al. [37]), whose proof spans less than a page, shows that in every P_k free graph G there exists a vertex set X_1 of size at most $k - 1$, such that every connected component of $G - N[X_1]$ has at most $n/2$ vertices. Bascó et al. [37] apply this result to instances output by the degree reduction procedure above. In such instances, $|N[X_1]| \leqslant O(\sqrt{n \log n})$, assuming k is a constant. Then, after guessing the intersection of the optimal solution with $N[X_1]$ (there are at most $2^{|N[X_1]|} \leqslant 2^{\sqrt{n \log n}}$ such guesses) the connected components of $G - N[X_1]$ become independent sub-instances of size at most $n/2$, on which the algorithm may be called recursively. Thus, solving a single instance on n vertices reduces to solving $2^{O(\sqrt{n \log n})}$ instances on at most $n/2$ vertices. Analyzing the corresponding recurrence shows that the total running time of the algorithm is upper

bounded by $2^{O(\sqrt{n \log n})}$.

If we wish to improve the running time from $2^{O(\sqrt{n \log n})}$ to quasi-polynomial, we may only apply degree reduction with $d = \Omega(\frac{n}{\log^{O(1)} n})$, and we can not afford to guess the intersection of the balanced separator $N[X_1]$ with an optimal solution. At this point we apply a slight generalization of degree reduction, to degree reduction relative to a vertex set S. Here we branch on vertices v that have at least d' neighbors in S (the vertex v itself does not have to be in S). A simple recursion analysis shows that this will reduce a single instance to $n^{|S|/d'}$ instances where every vertex has at most d' neighbors in S. We apply degree reduction on the balanced separator $N[X_1]$ with $d' = |N[X_1]|/c$ for some constant c (possibly depending on k). Thus, the initial degree reduction, followed by the degree reduction on $N[X_1]$, reduces the task of solving a single instance G to that of solving the problem on $2^{\log^{O(1)} n}$ instances in which every vertex has degree at most $n/\log^{O(1)} n$ and furthermore has at most $|N[X_1]|/c$ neighbors in the set $N[X_1]$. Here we are working with induced subgraphs of the original graph G, so when we say $N[X_1]$ we really mean what remains of the set $N[X_1]$ (with the neighborhood taken in the graph G) in the subgraph of G that is currently being considered.

The route above is perhaps the most natural one to try to obtain a quasi-polynomial time algorithm. Indeed, it is also the engine in the quasi-polynomial time algorithm of Lokshtanov et al. [88] for P_6-free graphs. However it is not at all clear how to deal with the instances output by this degree reduction. For P_6-free graphs, Lokshtanov et al. [88] (essentially) show that if the balanced separator $N[X_1]$ is chosen very carefully, then the degree reduction procedure never gets stuck: as long as $N[X_1]$ is non-empty some vertex is a neighbor to a constant fraction of $N[X_1]$. Thus the algorithm will make quasi-polynomially many calls on instances where the balanced separator $N[X_1]$ has been reduced to the empty set, in which case each connected component of the graph is substantially smaller than the original graph. This leads to a recurrence that solves

to quasi-polynomial time. We are not able to prove an analogous statement for P_k-free graphs for higher values of k, and so we are faced with the problem of how to deal with the degree-reduced instances described above.

The key insight of our algorithm is the following: *if we re-apply the "Gyárfás path" argument of Bascó et al. [37] on the degree-reduced instances to obtain a new balanced separator $N[X_2]$, then $N[X_2]$ can not have large intersection with $N[X_1]$.* This is because $N[X_2]$ is the neighbor set of a constant size set (X_2) and no vertex in the degree-reduced instance has many neighbors in $N[X_1]$. We now apply the degree reduction procedure again, this time on $N[X_2]$. If this reduction procedure completely reduces X_1 or X_2 to the empty set, or disconnects the graph into connected components so that the largest one has at most $0.9n$ vertices, then we have won, because the connected components of our instances are substantially smaller than on the original graph. If the procedure gets stuck then we obtain yet another balanced separator X_3, observe that X_3 has small intersection with X_2 and X_1, and do degree-reduction on X_3. And this keeps going, we keep adding new balanced separators into the mix until the degree-reduction procedure sufficiently disconnects the graph (i.e the largest connected component of the instances becomes sufficiently smaller than the original graph. The hard part of the analysis is to prove that the graph does become substantially disconnected by the time at most $O(\log n)$ separators have been added to the instance.

The actual final form of the algorithm is slightly different from what we describe above. Indeed, based on the ideas in the previous paragraph we can get an algorithm with running time $O(2^{n^\varepsilon})$ for every $\varepsilon > 0$, however to obtain quasi-polynomial time we need to be slightly more careful. The main difference is that we do not do degree reduction on each individual separator $N[X_i]$. Instead we define *level sets*. Level i is the set of all vertices that appear in at least i of the separators $N[X_1], \ldots, N[X_t]$ that we have constructed so far. We will maintain that throughout the course of the algorithm

the size of level i drops exponentially with i. Thus there will only be $O(\log n)$ levels, and we can afford to run degree reduction so that for each level, no vertex sees more than a $(\frac{1}{k \log n})^{O(1)}$ fraction of that level. Then, when we add a new separator, because it is the neighbor set of only a constant number of vertices, each level will increase by at most a factor of $1 + (\frac{1}{k \log n})^{O(1)}$ of the size of the previous level. Thus, such a process may continue to depth $(k \log n)^{O(1)}$ while maintaining the invariant that the size of the level i drops exponentially with i.

If recursion depth $\Omega(k \log n)$ is reached without sufficiently disconnecting the graph (i.e the largest connected component C of the graph still has size at least $N/2$, where N is the number of vertices in the original graph) this means that we have found $\Omega(k \log n)$ balanced separators for the graph such that no vertex is contained in more than $O(\log n)$ of them. A simple counting argument then shows that the average distance between pairs of vertices in the component C has to be at least $\frac{k \log n}{\log n} \geqslant k$, contradicting that G is P_k-free. This means that after recursion depth $O(k \log n)$, the graph has already been disconnected! At this point running the algorithm from scratch on each of the connected components yields at most n instances of size at most $n/2$ which solves to quasi-polynomial time.

Our algorithm for Theorem 3.1.2 follows the same template as the algorithm for Theorem 3.1.1. The key difference is that instead of growing a sequence of balanced separators we grow a sequence of (neighborhoods of) induced copies in G of connected components of H. Again the sequence has the property that the sets in the sequence do not overlap too much, so if we can grow the sequence to length $\Omega(|H|^{O(1)} \log n)$ then we can find an induced copy of H in G.

3.2 Preliminaries

Given a positive number k and a graph G, we call a set $S \subset V(G)$ a *c-balanced separator* if no connected component of $G - S$ has over c vertices.

A *vertex multi-family* $\mathcal{F}$ is a collection of vertex sets that allows for multiple instances of its vertex sets. If $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$ and X is a set of vertices, then $\mathcal{F} - X$ is the vertex multi-family $\{S_1 - X, S_2 - X, \ldots, S_n - X\}$. For two vertex multi-families $\mathcal{A}$ and $\mathcal{B}$ their *union* is denoted by $\mathcal{A} \cup \mathcal{B}$ and is defined by the vertex multi-family that contains all elements of $\mathcal{A}$ and of $\mathcal{B}$. The multiplicity of an element X in $\mathcal{A} \cup \mathcal{B}$ is its multiplicity in $\mathcal{A}$ plus its multiplicity in $\mathcal{B}$. We will use $\log(x)$ to denote the function $\lceil \log_2(x) \rceil$ throughout this paper.

3.3 Quasi-Polynomial Time Algorithm for P_k-Free Graphs

In this section we prove Theorem 3.1.1. We will make use of the following balanced separator lemma from Basco et al. [37].

Lemma 3.3.1. *[37] There exists an algorithm that given a graph G runs in polynomial time and outputs an induced path P in G such that $N(V(P))$ is a $\frac{|V(G)|}{2}$-balanced separator of G.*

We begin by proving a slight strengthening of Lemma 3.3.1.

Lemma 3.3.2. *There exists an algorithm that takes as input a graph G, and a positive integer i such that $2^i < |V(G)|$, runs in polynomial time and outputs a set X such that $N[X]$ is a $\frac{|V(G)|}{2^i}$-balanced separator in G. Furthermore, if G is P_k-free then $|X| \leqslant 2^{i+1} \cdot k$.*

Proof: Let G and i be as in the statement of the lemma, the proof is by induction on i. For $i = 1$ the algorithm calls the algorithm of Lemma 3.3.1 and obtains a path P. It then returns $X = V(P)$. Lemma 3.3.1 guarantees that in this case X satisfies the statement of this lemma. For $i > 1$ the algorithm first calls itself recursively on the input $(G, i-1)$ and obtains a set X' such that $N[X']$ is a $\frac{|V(G)|}{2^{i-1}}$-balanced separator in G, and furthermore, if G is P_k-free then $|X'| \leqslant 2^i \cdot k$. For each connected component C_j of $G - N[X']$ such that $|V(C_j)| > \frac{|V(G)|}{2^i}$ the algorithm calls itself recursively on $(C_j, 1)$ and obtains a set X_j such that $N[X_j]$ is a $\frac{|V(C_j)|}{2}$-balanced separator of C_j. If G is P_k-free it holds that $|X_j| \leqslant k$. The algorithms sets X as $X = X' \cup (\bigcup_j X_j)$ where the union is taken over all j such that $|V(C_j)| > \frac{|V(G)|}{2^i}$. Clearly the construction of X ensures that $N[X]$ is a $\frac{|V(G)|}{2^i}$-balanced separator of G. Furthermore if G is P_k-free then $|X| \leqslant |X'| + t \cdot k$ where t is the number of connected components of $G - X'$ whose size is more than $\frac{|V(G)|}{2^i}$. Since these components are disjoint we have that $t \leqslant 2^i$. Therefore $|X| \leqslant 2^i \cdot k + 2^i \cdot k = 2^{i+1} \cdot k$ as claimed.

To see that the run time is polynomial it suffices to show the number of times the algorithm makes a call to the algorithm of Lemma 3.3.1 is polynomial. To see this polynomial bound, note that on input (G, i) the algorithm makes at most 2^i calls to the algorithm of Lemma 3.3.1 plus the number calls it makes to the algorithm of Lemma 3.3.1 on input $(G, i-1)$. Since $2^i \leqslant |V(G)| = n$, the recurrence shows the algorithm makes at most $\Sigma_{i=0}^{\log(n)} n/2^i = O(n)$ calls to the algorithm of Lemma 3.3.1. ■

To describe the algorithm of Theorem 3.1.1 we first need to define the notion of *level sets* relative to a vertex multi-family $\mathcal{F}$.

Definition 3.3.3. Given a graph G and a vertex multi-family $\mathcal{F}$ consisting of vertex sets of G, for positive integers i, the i^{th} *level relative to* $\mathcal{F}$ is denoted by $L(\mathcal{F}, i)$ and defined

as follows

$$L(\mathcal{F}, i) = \{v \in V(G) \ : \ |\{S \in \mathcal{F} \ : \ v \in S\}| \geqslant i\}$$

In other words $L(\mathcal{F}, i)$ is a vertex set containing all vertices of G that are contained in at least i sets in $\mathcal{F}$. Our algorithm will also make use of a number N, this number will be approximately equal to the number of vertices in the input graph G.

Definition 3.3.4. The i^{th} *branch threshold* is denoted by Δ_i and is defined as $\Delta_i = N/2^i$. Given a multi-family $\mathcal{F}$, a vertex $v \in V(G)$ is a *branchable vertex* if there exists an $i \geqslant 1$ such that $|N[v] \cap L(\mathcal{F}, i)| \geqslant \Delta_i$.

In the following G is always a graph, w is a weight function $w : V(G) \to \mathbb{N}$, N is an integer, and $\mathcal{F}$ is a multi-family of subsets of $V(G)$. We now describe the main subroutine ALG_1 in the algorithm of Theorem 3.1.1. The algorithm takes as input G, w, N and $\mathcal{F}$ and (as we will prove) outputs the weight of a maximum weight independent set in G. The algorithm of Theorem 3.1.1 will call ALG_1 with parameters G, $N = |V(G)|$, w, and $\mathcal{F} = \emptyset$. ALG_1 is a recursive branching algorithm with only four rules. First, if G has at most one vertex, then return $V(G)$. Second, if the largest component of G has at most $|N|/2$ vertices then solve the problem recursively on each component and return the sum. Third, if there exists a branchable vertex v, then branch on v (i.e solve the problem with v forced in to the independent set, and v forced out). Finally, if none of the previous rules apply then *add* a new balanced separator X (obtained by Lemma 3.3.2) to $\mathcal{F}$. In other words, make a recursive call on the instance $(G, w, N, \mathcal{F} \cup \{N[X]\})$.

ALG_1 is very similar to well known exact exponential time branching algorithms for Independent Set [93]. The key differences are that we use the multi-family $\mathcal{F}$ of balanced separators to guide which vertex to branch on, that when no rules apply we add a separator to the family $\mathcal{F}$ (at a glance this appears to make no progress at all, but it increases the size of the level sets, making more vertices branchable), and that we wait

with recursing on connected components until the size of the largest component becomes less than $N/2$ (this is primarily to simplify the analysis).

ALG$_1$

1: **Input:** G, w, N, $\mathcal{F}$.
2: **Output:** $\mathsf{mwis}(G)$.
3: **if** $|V(G)| \leqslant 1$ **then**
4: **return** $w(V(G))$
5: **else if** $(\max_{C \in \mathcal{CC}(G)} |V(C)|) \leqslant N/2$ **then**
6: **return** $\sum_{C \in \mathcal{CC}(G)} \mathrm{ALG}_1(C, w, |V(C)|, \emptyset)$
7: **else if** exists branchable vertex v **then**
8: **return** $\max(\mathrm{ALG}_1(G - v, w, N, \mathcal{F} - \{v\}), \mathrm{ALG}_1(G - N[v], w, N, \mathcal{F} - N[v]) + w(v))$
9: **obtain** X by applying Lemma 3.3.2 on G with $i = 2$
10: **return** $\mathrm{ALG}_1(G, w, N, \mathcal{F} \cup \{N[X]\})$

We will distinguish between the three different kinds of recursive calls that ALG_1 can make. If the **else if** condition on line 5 holds, then the algorithm makes the recursive calls on line 6. In this case we say that ALG_1 *recurses on connected components*. If the **else if** condition on line 7 holds, then the algorithm makes the recursive calls on line 8. In this case we say that ALG_1 *branches on a branchable vertex*. Otherwise the algorithm makes the recursive call on line 10. In this case we say that ALG_1 *adds a balanced separator*. We will frequently need to refer to parts of the execution of the algorithm. For disambiguation, we collect the terminology here.

An *instance* is a four-tuple $(G, w, N, \mathcal{F})$. A *run* of the algorithm refers to the entire execution of the algorithm on an instance. A *call* $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ refers to the computation done in the root node of the recursion tree of the run $\mathrm{ALG}_1(G, w, N, \mathcal{F})$. We remark that parameters G, w, N, and $\mathcal{F}$ never change during the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$. When

a run or a call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ recursively calls ALG_1 on the instance $(G', w, N', \mathcal{F}')$ we say the run or the call *executes* a run or a call on $(G', w, N', \mathcal{F}')$. This will sometimes be referred to as *makes a recursive call* $\mathrm{ALG}_1(G', w, N', \mathcal{F}')$. A run of $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ is called a k-*fair run* if G is a P_k-free graph, $N = |V(G)|$, $\mathcal{F} = \emptyset$, and w is a weight function. A call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ is called a k-*fair call* if it is executed during the course of a k-fair run. An instance $(G, w, N, \mathcal{F})$ is called a k-*fair* instance if $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ is a k-fair call.

Lemma 3.3.5. *$ALG_1(G, w, N, \mathcal{F})$ terminates on every input.*

Proof: Consider a run of ALG_1 with initial input G, w, N, and $\mathcal{F}$. Whenever the algorithm makes a recursive call $\mathrm{ALG}_1(G', w, N', \mathcal{F}')$ we have that $|V(G')| \leqslant |V(G)|$ and $N' \leqslant N$. Furthermore, whenever the algorithm recurses on connected components or branches on a branchable vertex, then it executes $\mathrm{ALG}_1(G', w, N', \mathcal{F}')$ with either $|V(G')| < |V(G)|$ or $N' < N$. Finally, ALG_1 cannot add a balanced separator for over $|V(G)| \cdot \log(N)$ successive recursive calls since then a call $\mathrm{ALG}_1(G, w, N, \mathcal{F}'')$ with $\mathcal{F}'' = |V(G)| \cdot \log(N)$ would add a balanced separator. However, since each new balanced separator must be non-empty (since otherwise the algorithm would have recursed on connected components) we have that $L(\mathcal{F}'', \log(N)) \neq \emptyset$, and so the call $\mathrm{ALG}_1(G, w, N, \mathcal{F}'')$ would branch on a vertex. This contradicts that the call added a balanced separator, and proves that ALG_1 cannot add a balanced separator for over $|V(G)| \cdot \log(N)$ successive recursive calls. It follows by induction on $|V(G)| + N$ that ALG_1 always terminates. ■

Lemma 3.3.6. *A run $ALG_1(G, w, N, \mathcal{F})$ always returns the weight of a maximum weight independent set of G under the weight function w.*

Proof: Consider a run of ALG_1 with initial input G, w, N, and $\mathcal{F}$. It is clear from the algorithm that if each run $\mathrm{ALG}_1(G', w, N', \mathcal{F})$ that is executed by the call

$\mathrm{ALG}_1(G, w, N, \mathcal{F})$ returns the weight of a maximum weight independent set of G' with weight function w, then so would the run $\mathrm{ALG}_1(G, w, N, \mathcal{F})$. By Lemma 3.3.5 the height of the recursion tree is bounded, and the result is trivially true for the base case of $|V(G)| \leqslant 1$, so the result follows by induction on the height of the recursion tree of the run $\mathrm{ALG}_1(G, w, N, \mathcal{F})$. ■

We have now proved that ALG_1 always terminates and that it always outputs the correct answer. The remainder of the section is devoted to the running time analysis. We will now prove some lemmas to help us bound the run time of ALG_1 on k-fair runs. First, in Observation 3.3.7 we will prove that $\mathcal{F}$ remains a multi-family of balanced separators of G throughout the execution of the algorithm. In Observation 3.3.8 we will show that no vertex appears in many (more than $\log N$) sets in $\mathcal{F}$. This will ensure that $\mathcal{F}$ can never grow too large, because, as we will show in Lemma 3.3.9, a connected P_k-free graph can not contain a large fractional packing of balanced separators.

Observation 3.3.7. *Let $(G, w, N, \mathcal{F})$ be a k-fair instance. Then every set $S \in \mathcal{F}$ is a $\frac{N}{4}$-balanced separator of G.*

Proof: Consider a k-fair instance $(G, w, N, \mathcal{F})$. If $\mathcal{F} = \emptyset$ then the result is trivially true, so assume $\mathcal{F} \neq \emptyset$. It follows ALG_1 executes $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ during a k-fair call $\mathrm{ALG}_1(G', w, N, \mathcal{F}')$ by branching on a branchable vertex or ALG_1 executes $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ during a k-fair call $\mathrm{ALG}_1(G, w, N, \mathcal{F}'')$ by adding a balanced separator, X. In the first case, if all sets of $\mathcal{F}'$ are $\frac{N}{4}$-balanced separators for G', then since $G = G' - S$ for some vertex set S, and $\mathcal{F} = \mathcal{F}' - S$, all sets of $\mathcal{F}$ are $\frac{N}{4}$-balanced separators for G. In the second case, X is generated in such a way that it is guaranteed to be an $\frac{N}{4}$-balanced separator for G, so if all sets of $\mathcal{F}''$ are $\frac{N}{4}$-balanced separators for G, then all sets of $\mathcal{F}$ are $\frac{N}{4}$-balanced separators for G. The statement of the observation now follows by induction on the depth of the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ in the recursion tree. ■

Observation 3.3.8. *For every k-fair instance $(G, w, N, \mathcal{F})$, we have that $L(\mathcal{F}, \log(N) + 1) = \emptyset$.*

Proof: Consider a k-fair call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$. We will prove the statement by induction on the depth of the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ in the recursion tree of a run $\mathrm{ALG}_1(G^*, w, |V(G^*)|, \emptyset)$ which executes $\mathrm{ALG}_1(G, w, N, \mathcal{F})$.

If $\mathcal{F} = \emptyset$ then the result is trivially true. Suppose now that $\mathcal{F} \neq \emptyset$, it follows ALG_1 executes $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ during a k-fair call $\mathrm{ALG}_1(G', w, N, \mathcal{F}')$ by branching on a branchable vertex or by adding a balanced separator X. In the first case $\mathcal{F} = \mathcal{F}' - S$ for some vertex set S. By the induction hypothesis $L(\mathcal{F}', \log(N) + 1) = \emptyset$ and hence $L(\mathcal{F}, \log(N) + 1) = \emptyset$. In the second case, $\mathrm{ALG}_1(G, w, N, \mathcal{F}')$ does not branch on a branchable vertex, so we have that $L(\mathcal{F}', \log(N)) = \emptyset$ since every vertex in $L(\mathcal{F}', \log(N))$ is branchable. It follows that $L(\mathcal{F}, \log(N) + 1) = L(\mathcal{F}' \cup X, \log(N) + 1) = \emptyset$. ■

Lemma 3.3.9. *For every k-fair instance $(G, w, N, \mathcal{F})$ it holds that $|\mathcal{F}| \leqslant 10k \cdot \log(N)$.*

Proof: Consider a k-fair instance $(G, w, N, \mathcal{F})$. We will prove the result by induction on the depth of the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ in the recursion tree of a run $\mathrm{ALG}_1(G^*, w, |V(G^*)|, \emptyset)$ which executes $\mathrm{ALG}_1(G, w, N, \mathcal{F})$.

In the base case $\mathcal{F} = \emptyset$, and the claim of the lemma holds trivially, so assume $\mathcal{F} \neq \emptyset$. Thus the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ is executed by a k-fair call $\mathrm{ALG}_1(G', w, N', \mathcal{F}')$. By the induction hypothesis $|\mathcal{F}'| \leqslant 10k \cdot \log(N)$ ($N = N'$ since $\mathcal{F} \neq \emptyset$). Thus, unless $\mathrm{ALG}_1(G', w, N', \mathcal{F}')$ recurses by adding a balanced separator we have that $|\mathcal{F}| \leqslant 10k \cdot \log(N)$ as well. So assume that $\mathrm{ALG}_1(G', w, N', \mathcal{F}')$ adds a balanced separator X and that therefore $G' = G$, $N' = N$ and $\mathcal{F} = \mathcal{F}' \cup \{X\}$. We prove that $|\mathcal{F}'| < 10k \cdot \log(N)$, then the result follows since $|\mathcal{F}| = |\mathcal{F}'| + 1$.

Suppose for contradiction that $|\mathcal{F}'| \geqslant 10k \cdot \log(N)$, we will now produce an induced path of length k in G, contradicting that G is P_k-free. The call $\mathrm{ALG}_1(G', w, N', \mathcal{F}') =$

$\mathrm{ALG}_1(G, w, N, \mathcal{F}')$ added a balanced separator, and so the size of the largest connected component, C, in G is greater than $\frac{N}{2}$. This, together with Observation 3.3.7 then gives that every set $S \in \mathcal{F}$ is a $\frac{|V(C)|}{2}$-balanced separator for C. Consider the following random process. Uniformly at random, select vertices x and y in C. For all $S \in \mathcal{F}$, let X_S denote the random variable that is 1 if x and y are not in the same connected component of $C - S$ and 0 otherwise. Since S is a $\frac{|V(C)|}{2}$-balanced separator for C, the probability that x and y are in the same connected component of $C - S$ is at most $\frac{1}{2}$. Thus $X_S = 1$ with probability at least $\frac{1}{2}$. We denote by $\mathcal{F}_{x,y}$ all sets $S \in \mathcal{F}$ such that x and y are not in the same component of $C - S$, again including multiplicity. By linearity of expectation we have that

$$E[|\mathcal{F}_{x,y}|] = \sum_{S \in \mathcal{F}} E[X_S] \geqslant |\mathcal{F}|/2 > 5k \cdot \log(N).$$

It follows there exists vertices a and b in C such that $|\mathcal{F}_{a,b}| > 5k \cdot \log(N)$. Let P be a shortest path connecting a and b in C. Since G is P_k-free, P has at most $k - 1$ vertices. By Observation 3.3.8, each of these vertices is contained in at most $\log(N)$ sets in $\mathcal{F}_{a,b}$. But then there exists a set $S \in \mathcal{F}_{a,b}$ disjoint from $V(P)$ contradicting that a and b are not in the same component of $C - S$. ■

The following observation shows that the level sets do not grow a lot in each successive recursive call, and that they therefore never get very large. Note in particular that the size of level set i drops exponentially with i.

Observation 3.3.10. *For every k-fair call $ALG_1(G, w, N, \mathcal{F})$ that adds a balanced separator X and every i,*

$$|L(\mathcal{F} \cup X, i)| \leqslant \Delta_{i-1} \cdot 8k + |L(\mathcal{F}, i)|.$$

Furthermore, for every k-fair instance $(G', w, N', \mathcal{F}')$,

$$|L(\mathcal{F}', i)| \leqslant \Delta_{i-1} \cdot 8k \cdot |\mathcal{F}'|.$$

Proof: Consider a k-fair call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ that adds a balanced separator X. Let X_j denote the set of vertices in $L(\mathcal{F}, j) \cap X$, then we can see that $|L(\mathcal{F} \cup \{X\}, j)| \leqslant L(\mathcal{F}, j) + |X_{j-1}|$. Since the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ adds a balanced separator, X, there are no branchable vertices. So, we have that for all $v \in G$, $|N[v] \cap L(\mathcal{F}, j)| \leqslant \Delta_j$. Furthermore, by Lemma 3.3.2, since X is generated as an $\frac{N}{4}$-balanced separator and therefore a $\frac{|G|}{4}$-balanced separator for G, X is the neighborhood of at most $8k$ vertices, hence $|X_{j-1}| \leqslant \Delta_{j-1} \cdot 8k$ and the result $|L(\mathcal{F} \cup \{X\}, i)| \leqslant \Delta_{i-1} \cdot 8k + |L(\mathcal{F}, i)|$ follows.

The second statement follows by combining induction, the first part of this Observation, and the fact that if the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ executes $\mathrm{ALG}_1(G', w, N', \mathcal{F}')$, then $|\mathcal{F}| < |\mathcal{F}'|$ if and only if the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ adds a balanced separator.

■

For k-fair instances $(G, w, N, \mathcal{F})$ we define a measure:

$$\mu_k(G, w, N, \mathcal{F}) = 400k^2 \cdot \log^2(N) \cdot (N + |V(G)|) + \sum_i \left(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}} \right)$$
$$+ 16k \cdot N \cdot \log(N) \cdot (10k \cdot \log(N) - |\mathcal{F}|)$$

If $(G, w, N, \mathcal{F})$ is not a k-fair instance, then $\mu_k(G, w, N, \mathcal{F})$ is undefined. Note that $\mu_k(G, w, N, \mathcal{F})$ must always be an integer, and that it is independent of the weight function w. We will say that two instances $(G, w, N, \mathcal{F})$ and $(G', w', N', \mathcal{F}')$ are *essentially different* if $G' \neq G$, $N' \neq N$ or $\mathcal{F}' \neq \mathcal{F}$.

Lemma 3.3.11. *For every positive integer μ, the number of essentially different k-fair instances $(G, w, N, \mathcal{F})$ such that $\mu_k(G, w, N, \mathcal{F}) = \mu$ is finite. In addition, for every k-fair*

instance it holds that $\mu_k(G, w, N, \mathcal{F}) \geqslant 0$.

Proof: Consider a k-fair instance $(G, w, N, \mathcal{F})$ with $\mu_k(G, w, N, \mathcal{F}) = \mu$. Clearly, there are only a finite number of such instances with $N = 1$. We will show that if $N \geqslant 2$ then $|V(G)| \leqslant \mu$. If $|V(G)| \leqslant \mu$ then $|G|$, $|N|$, and $|\mathcal{F}|$ are all bounded in terms of μ, and the first part of the lemma follows.

By Lemma 3.3.9 we have that $|\mathcal{F}|$ is at most $10k \cdot \log(N)$. It follows that the terms $400k^2 \cdot \log^2(N) \cdot (N + |V(G)|)$, $\Sigma_i(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}})$, and $16k \cdot N \cdot \log(N) \cdot (10k \cdot \log(N) - |\mathcal{F}|)$ are all non negative. Hence $\mu_k(G, w, N, \mathcal{F}) \geqslant |V(G)|$. This also proves that $\mu_k(G, w, N, \mathcal{F}) \geqslant 0$. ■

Lemma 3.3.12. *For every k-fair instance $(G, w, N, \mathcal{F})$ it holds that* $\mu_k(G, w, N, \mathcal{F}) \leqslant 1050k^2 \cdot N \cdot \log^2(N)$

Proof: Consider a k-fair instance $(G, w, N, \mathcal{F})$. By Observation 3.3.10 and Lemma 3.3.9, we have that $|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}} < 8k \cdot N \cdot |\mathcal{F}| < 80k^2 \cdot N \cdot \log(N)$. Therefore, $\Sigma_i(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}}) < 80k^2 \cdot N \cdot \log^2(N)$. Also, since $N \geqslant |V(G)|$, we can see that

$$\begin{aligned}
\mu_k(G, w, N, \mathcal{F}) &= 400k^2 \cdot \log^2(N) \cdot (N + |V(G)|) + \Sigma_i(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}}) \\
&\quad + 16k \cdot N \cdot \log(N) \cdot (10k \cdot \log(N) - |\mathcal{F}|) \\
&< 800k^2 \cdot N \cdot \log^2(N) + 80k^2 \cdot N \cdot \log^2(N) + 160k^2 \cdot N \cdot \log^2(N) \\
&< 1050k^2 \cdot N \cdot \log^2(N)
\end{aligned}$$

■

We define $T_k(G, w, N, \mathcal{F})$ to be the running time of a k-fair run of ALG_1 starting with the inputs $(G, w, N, \mathcal{F})$. We also define

$$T_k(\mu) = max_{G,N,\mathcal{F} \,:\, \mu_k(G,w,N,\mathcal{F}) \leqslant \mu} \mathrm{T}_k(G, w, N, \mathcal{F}).$$

When we analyze run time we assume that arithmetic (addition, subtraction, comparisons) on weights of vertices and vertex sets is constant time. Thus, both the running time of ALG_1 and the measure of an instance $(G, w, N, \mathcal{F})$ are independent of the weight function w. Thus, by Lemma 3.3.11, $T_k(\mu)$ is well defined.

Lemma 3.3.13. *$T_k(\mu)$ satisfies the following recurrence:*

$$T_k(\mu) \leqslant \mu^{O(1)} + max \begin{cases} \mu T_k(.95\mu) \\ T_k(\mu - 1) + T_k(\mu[1 - 1/(2100k^2 \cdot \log^2(\mu))]) \\ T_k(\mu[1 - 1/(200k \cdot \log(\mu))]) \end{cases}$$

Proof: Let $(G, w, N, \mathcal{F})$ be a k-fair instance such that $\mu_k(G, w, N, \emptyset) = \mu$ and $T_k(\mu)$ is the run time of $\mathrm{ALG}_1(G, w, N, \mathcal{F})$. If the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ recurses on connected components, then it makes at most $|V(G)|$ recursive calls on instances of the form $(G', w, N', \emptyset)$, where $|V(G')| \leqslant |V(G)|$ and $N' \leqslant \frac{N}{2}$. It follows that for each of these recursive calls we have

$$\begin{aligned} \mu_k(G', w, N', \emptyset) &= 400k^2 \cdot \log^2(N') \cdot (N' + |V(G')|) + 160k^2 \cdot N' \cdot \log^2(N') \\ &\leqslant 400k^2 \cdot \log^2(N) \cdot (\frac{N}{2} + |V(G)|) + 80k^2 \cdot N \cdot \log^2(N) \\ &\leqslant 400k^2 \cdot \log^2(N) \cdot (N + |V(G)|) - 100k^2 \cdot N \cdot \log^2(N) \\ &\leqslant \mu - 100k^2 \cdot N \cdot \log^2(N) \\ &\leqslant .95\mu \qquad \text{(by Lemma 3.3.12)} \end{aligned}$$

Therefore, if the instance $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ recurses on connected components, we must have that $T_k(\mu) \leqslant |V(G)| \cdot T_k(.95\mu) \leqslant \mu \cdot T_k(.95\mu)$.

If the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ branches on a branchable vertex, v, then it makes two recursive calls, one execution $\mathrm{ALG}_1(G - \{v\}, w, N, \mathcal{F} - \{v\})$, where the instance $(G - \{v\}, w, N, \mathcal{F} - \{v\})$ has measure $\mu_k(G - \{v\}, w, N, \mathcal{F} - \{v\}) \leqslant \mu - 1$, and the other execution is $\mathrm{ALG}_1(G - N[v], w, N, \mathcal{F} - N[v])$. Note that for a branchable vertex, v, we have that

$$\sum_i (|L(\mathcal{F} - N[v], i)| \cdot \frac{N}{\Delta_{i-1}} \leqslant \sum_i (|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}}) - \frac{N}{2},$$

since for at least one level i we have that $|N[v] \cap L(\mathcal{F}, i)| \geqslant \Delta_i$ and $\frac{\Delta_i}{\Delta_{i-1}} = 1/2$. It follows that

$$\begin{aligned}
\mu_k(G - N[v], w, N, \mathcal{F} - N[v]) &= 400k^2 \cdot \log^2(N) \cdot (N + |V(G) - N[V]|) \\
&\quad + \sum_i \left(|L(\mathcal{F} - N[v], i)| \cdot \frac{N}{\Delta_{i-1}} \right) \\
&\quad + 16k \cdot N \cdot \log(N) \cdot (10k \cdot \log(N) - |\mathcal{F}|) \\
&\leqslant 400k^2 \log^2(N) \cdot (N + |V(G)|) + \sum_i \left(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}} \right) \\
&\quad + 16k \cdot N \cdot \log(N) \cdot (10k \cdot \log(N) - |\mathcal{F}|) - \frac{N}{2} \\
&\leqslant \mu - \frac{N}{2} \\
&\leqslant \mu \left(1 - \frac{1}{2100k^2 \cdot \log^2(N)} \right) \qquad \text{(by Lemma 3.3.12)} \\
&\leqslant \mu \left(1 - \frac{1}{2100k^2 \cdot \log^2(\mu)} \right)
\end{aligned}$$

Therefore, if the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ branches on a branchable vertex, then we have that $T_k(\mu) \leqslant T_k(\mu - 1) + T_k(\mu[1 - \frac{1}{2100k^2 \cdot \log^2(\mu)}])$.

Finally, if the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ adds a balanced separator, X, then it makes a single recursive call $\mathrm{ALG}_1(G, w, N, \mathcal{F} \cup X)$. By Observation 3.3.10 and Lemma 3.3.12 we

obtain the following.

$$\mu_k(G, w, N, \mathcal{F} \cup X) < \mu + 8k \cdot N \cdot \log(n) - 16k \cdot N \cdot \log(N) < \mu\left([1 - \frac{1}{200k \cdot \log(\mu)}]\right)$$

Therefore, if the call $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ adds a balanced separator, then $T_k(\mu) \leqslant T_k(\mu[1 - \frac{1}{200k \cdot \log(\mu)}])$.

The result now follows from the observation that $\mathrm{ALG}_1(G, w, N, \mathcal{F})$ only does $|V(G)|^{O(1)} = \mu^{O(1)}$ work in any given call and always recurses on connected components, branches on a branchable vertex, adds a balanced separator, or returns without making further recursive calls. ■

Since $T_k(\mu)$ is a non negative, non decreasing function, by adding the three possibilities in the max of Lemma 3.3.13 we immediately obtain the following simplified recurrence.

Corollary 3.3.14. $T_k(\mu) \leqslant \mu^{O(1)} + \mu T_k(.95\mu) + T_k(\mu - 1) + T_k(\mu[1 - \frac{1}{2100k^2 \cdot \log^2(\mu)}]) + T_k(\mu[1 - \frac{1}{200k \cdot \log(\mu)}]) < T_k(\mu - 1) + \mu^{O(1)} + 3\mu \cdot T_k(\mu[1 - \frac{1}{2100k^2 \cdot \log^2(\mu)}])$

Lemma 3.3.15. $T_k(\mu) = \mu^{O(k^2 \cdot \log^3(\mu))}$

Proof: The proof is by induction on μ. The base case is established by Lemma 3.3.11. By Corollary 3.3.14 we have the inequality $T_k(\mu) \leqslant T_k(\mu - 1) + \mu^{O(1)} + 3\mu T_k(\mu[1 - \frac{1}{2100k^2 \cdot \log^2(\mu)}])$ and repeatedly applying the inequality to the first term on the right hand side, gives $T_k(\mu) \leqslant \mu^{O(1)} + 3\mu^2 \cdot T_k(\mu[1 - \frac{1}{2100k^2 \cdot \log^2(\mu)}])$. By the inductive hypothesis then, there is some c such that

$$
\begin{aligned}
T_k(\mu) &\leqslant \mu^{O(1)} + 3\mu^2 \cdot (\mu[1 - \frac{1}{2100k^2 \cdot \log^2(\mu)}])^{ck^2 \cdot \log^3(\mu)} \\
&= \mu^{O(1)} + 3\mu^2 \cdot \mu^{ck^2 \cdot \log^3(\mu)} \cdot [1 - \frac{1}{2100k^2 \cdot \log^2(\mu)}]^{ck^2 \cdot \log^3(\mu)} \\
&\leqslant \mu^{O(1)} + 3\mu^2 \cdot \mu^{ck^2 \cdot \log^3(\mu)} \cdot e^{-\frac{ck^2 \cdot \log^3(\mu)}{2100k^2 \cdot \log^2(\mu)}} \qquad (\text{ since } 1 - x \leqslant e^{-x} \ ) \\
&\leqslant \mu^{O(1)} + 3\mu^2 \cdot \mu^{ck^2 \cdot \log^3(\mu)} \cdot e^{-\frac{c \log(\mu)}{2100}} \\
&\leqslant \mu^{ck^2 \cdot \log^3(\mu)} \qquad (\text{ for sufficiently large } c \ )
\end{aligned}
$$

■

We are now ready to prove Theorem 3.1.1.

Proof: [Proof of Theorem 3.1.1] The algorithm returns the answer of $\mathrm{ALG}_1(G, |V(G)|, w, \emptyset)$. By Lemma 3.3.5 ALG_1 terminates, by Lemma 3.3.6 ALG_1 returns return the weight of a maximum weighted independent set. For the running time, observe that $(G, w, N, \emptyset)$ is a k-fair instance and let $\mu = \mu_k(G, w, N, \emptyset)$. By Lemma 3.3.12 we have that $\mu < 1050k^2 \cdot N \cdot \log^2(N) = n^{O(1)}$. Hence, by Lemma 3.3.15 it follows that $T(G, w, N, \emptyset) \leqslant T(\mu) = \mu^{O(k^2 \cdot \log^3(\mu))} = n^{O(k^2 \cdot \log^3(n))}$. ■

3.4 Disconnected Forbidden Induced Subgraphs

Let H be a graph. We denote by O_H an oracle that takes an H-free graph G as input and outputs the weight of a maximum weight independent set in G. In this section we present a quasi-polynomial time algorithm for Maximum Weight Independent Set in H-free graphs, assuming we have access to the oracles O_C for all $C \in \mathcal{CC}(H)$. Specifically we will prove Theorem 3.1.2.

In the following, $H = H_0 + H_1 + ... + H_{c-1}$ is a graph, G is a graph, w is a weight function on the vertices of G, N is a positive integer, and $\mathcal{F}$ is a vertex multi-family of

subsets of $V(G)$. We now present the algorithm ALG_2 of Theorem 3.1.2. The algorithm is very similar to the algorithm ALG_1 for P_k free graphs, the main difference is that instead of packing balanced separators in the family $\mathcal{F}$, the algorithm "packs" (neighborhoods of) copies of induced H_i's.

ALG_2

1: **input:** $H, G, w, N, \mathcal{F}$.
2: **output:** $\mathsf{mwis}(G)$.
3: $i = |\mathcal{F}| \mod c$
4: **if** exists branchable vertex v **then**
5: **return** $\max(\text{ALG}_2(H, G - v, w, N, \mathcal{F} - \{v\}), \text{ALG}_2(H, G - N[v], w, N, \mathcal{F} - N[v]) + w(v))$
6: **else if** exists induced H_i **then**
7: **obtain** $X \leftarrow$ induced H_i in G
8: **return** $\text{ALG}_2(H, G, w, N, \mathcal{F} \cup \{N[X]\})$
9: **return** $O_{H_i}(G)$

The proof of correctness and running time analysis for ALG_2 closely follows that of ALG_1. The main difference is in the proof of why the family $\mathcal{F}$ can not grow beyond size $\log N$ (Lemmata 3.4.4 and 3.4.5). The other parts are just minor modifications of corresponding results from Section 3.3.

We will distinguish between the two different kinds of recursive calls that ALG_2 can make. If the **if** condition of line 4 holds, then the algorithm makes the recursive calls on line 5. In this case we say that ALG_2 *branches on a branchable vertex*. If the **else if** condition of line 6 holds, then the algorithm makes the recursive call in line 8. In this case we say that ALG_2 *adds a neighborhood*. We define instances, runs, calls, execution and making a recursive call similarly as for ALG_1. Just as for ALG_1, a run of

$\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ is called a *fair run* if G is an H-free graph, $N = |V(G)|$, $\mathcal{F} = \emptyset$, and w is a weight function. A call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ is called a *fair call* if it is executed during the course of a fair run. An instance $(H, G, w, N, \mathcal{F})$ is a *fair instance* if $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ is a fair call.

Lemma 3.4.1. *$ALG_2(H, G, w, N, \mathcal{F})$ terminates on every input.*

Proof: Consider a run of ALG_2 with initial input H, G, w, N, and $\mathcal{F}$. Whenever the algorithm makes a recursive call it does so with a call $\mathrm{ALG}_2(H, G', w, N, \mathcal{F}')$ where $|V(G')| \leqslant |V(G)|$. Furthermore, whenever the algorithm branches on a branchable vertex, then it recurses with a call $\mathrm{ALG}_2(H, G', w, N, \mathcal{F})$ where $|V(G')| < |V(G)|$. Furthermore ALG_2 can not add a neighborhood in over $|V(G)| \cdot \log(N)$ successive recursive calls. Suppose it does, then a call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F}'')$ with $\mathcal{F}'' = |V(G)| \cdot \log(N)$ adds a neighborhood, but $L(\mathcal{F}'', \log(N)) \neq \emptyset$ and thus there exists a branchable vertex, contradicting that $\mathrm{ALG}_2(H, G, w, N, \mathcal{F}'')$ with $\mathcal{F}'' = |V(G)| \cdot \log(N)$ adds a neighborhood. It follows by induction on $|V(G)|$ that ALG_2 always terminates. ■

Lemma 3.4.2. *A run $ALG_2(H, G, w, N, \mathcal{F})$ returns the weight of a maximum weight independent set of G.*

Proof: Consider an run of ALG_2 with initial input $(H, G, w, N, \mathcal{F})$. It is clear from the algorithm that if each run $\mathrm{ALG}_2(H, G', w, N, \mathcal{F}')$ that is executed by the call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ returns the weight of a maximum weight independent set of G' under the weight function w, then so would the run $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$. By Lemma 3.4.1 the height of the recursion tree is bounded, and the result is trivially true for the base case of $|V(G)| \leqslant 1$ so the result follows by induction on the depth of the recursion tree. ■

Observation 3.4.3. *For ever fair instance $(H, G, w, N, \mathcal{F})$, we have that $L(\mathcal{F}, \log(N) + 1) = \emptyset$.*

Proof: Consider a fair call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$. We will prove the result by induction on the depth of the call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ in the recursion tree of a run $\mathrm{ALG}_2(H, G^*, w, |V(G^*)|, \emptyset)$ which executes $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$.

If $\mathcal{F} = \emptyset$ then the result is trivially true. Suppose $\mathcal{F} \neq \emptyset$, it follows that ALG_2 executes $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ during a fair call $\mathrm{ALG}_2(H, G', w, N, \mathcal{F}')$ by branching on a branchable vertex or by adding a neighborhood, $N[X]$. In the first case, it is clear that since $\mathcal{F} = \mathcal{F}' - S$ for some vertex set S, if $L(\mathcal{F}', \log(N)+1) = \emptyset$, then $L(\mathcal{F}, \log(N)+1) = \emptyset$ in $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$. In the second case, since the instance $\mathrm{ALG}_2(H, G, w, N, \mathcal{F}')$ does not branch on a branchable vertex, we have that $L(\mathcal{F}', \log(N)) = \emptyset$ since every vertex in $L(\mathcal{F}', \log(N))$ is branchable. It follows that $L(\mathcal{F}, \log(N)+1) = L(\mathcal{F}' \cup \{N[X]\}, \log(N)+1) = \emptyset$. ■

Lemma 3.4.4. *Let G be a graph, N an integer greater than 1, and let $H = H_0 + H_1 + ... + H_{c-1}$ be a graph. If there exists a sequence of subsets of $V(G)$, $\{X_m\} = X_0, X_1, ..., X_{c\cdot|H|\cdot\log(N)-1}$ such that for all i, $X_i \subset V(G)$, the subgraph induced by X_i is isomorphic to $H_{i \pmod c}$, and for all $v \in X_i$ we have that $\{v\} \cap N[X_j] \neq \emptyset$ for at most $\log(N)$ X_j's where $j < i$, then there exists a subset $I \subseteq \{0, 1, 2, \ldots, c \cdot |H| \cdot \log(N) - 1\}$ such that $X_I = \bigcup_{i \in I} X_i$ forms an induced H in G.*

Proof: Let G and H be graphs, N an integer greater than 1, and $X_0, X_1, ..., X_{c\cdot|H|\cdot\log(N)-1}$ a sequence of sets of vertices with the properties given in the statement of the lemma. Given an X_j, set $i = j - (j \ (mod\ c)$. We will refer to the segment X_i, X_{i+1},..., X_{i+c-1} as X_j's block.

The proof is by induction on c. If $c = 1$ then the statement is trivially true. Assume now that $c > 1$ and that the statement is true for all smaller values. There are at most $|H_{c-1}| \cdot \log(N)$ X_j's such that some vertex of $X_{c\cdot|H|\cdot\log(N)-1}$ belongs to X_j, $j \neq c \cdot |H| \cdot \log(N) - 1$. Remove from the sequence each such X_j along with all other vertex

sets in X_j's block, as well as all X_t's such that $c-1 \equiv t \ (mod\ c)$. After these deletions, re-name the sets X_j in the updated sequence so that the index j of each set X_j is equal to the position of X_j in the sequence (starting with X_0).

Let $H' = H - H_{c-1}$. There are at least $\log(N) \cdot (c \cdot |H| - c \cdot |H_{c-1}| - |H| + |H_{c-1}|) - 1$ $= \log(N) \cdot (c-1) \cdot |H'| - 1$ remaining vertex sets in the updated sequence, and this new sequence along with H' and G satisfies the condition of the inductive hypothesis. It follows that there exists a set X'_I such that $G[X'_I] = H'$ and X'_I is the union of sets in the (updated) sequence. Since $X_{c\cdot|H|\cdot\log(N)-1}$ does not belong to the neighborhood of any of the vertex sets in the new sequence, $X_{c\cdot|H|\cdot\log(N)-1}$ is disjoint from $N[X'_I]$, and hence $X_I = X'_I \cup X_{c\cdot|H|\cdot\log(N)-1}$ induces H in G, completing the proof. ■

Lemma 3.4.5. *For every fair instance $(H, G, w, N, \mathcal{F})$ with $H = H_0 + H_1 + ... + H_{c-1}$, it holds that $|\mathcal{F}| < c \cdot |H| \cdot \log(N)$*

Proof: Let the fair instance $(H, G, w, N, \mathcal{F})$ be as in the statement of the lemma, furthermore let G' be the graph used in the initial input of ALG_2 of the fair run that produces the instance $(H, G, w, N, \mathcal{F})$. Assume to the contrary, that $|\mathcal{F}| \geqslant c \cdot |H| \cdot \log(N)$. In the fair run that executes the call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$, consider the sequence of recursive calls (ordered by when the call occurs) that lead to the call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$. In particular, consider the subsequence $\mathrm{ALG}_2^0(H, G^0, w, N, \mathcal{F}^0), \mathrm{ALG}_2^1(H, G^1, w, N, \mathcal{F}^1), ...$

$\mathrm{ALG}_2^{c\cdot|H|\cdot\log(N)-1}(H, G^{c\cdot|H|\cdot\log(N)-1}, w, N, \mathcal{F}^{c\cdot|H|\cdot\log(N)-1})$ such that the call $\mathrm{ALG}_2^i(H, G^i, w, N, \mathcal{F}^i)$ is the $(i+1)^{th}$ call to add a neighborhood $N[X_i]$. By Observation 3.4.3, we can see that for all X_i, and for all vertices $v \in X_i$, $\{v\} \cap N[X_j] \neq \emptyset$ for at most $\log(N)$ X_j's with $j < i$. The result follows now by observing that G', H, N, and the sequence $X_0, X_1, ..., X_{c\cdot|H|\cdot\log(N)-1}$ satisfy the hypothesis of Lemma 3.4.4, contradicting that G' is H-free.

■

Observation 3.4.6. *For every fair call $ALG_2(H, G, w, N, \mathcal{F})$ that recurses by adding a neighborhood $N[X]$ and for every i,*

$$|L(\mathcal{F} \cup N[X], i)| \leqslant \Delta_{i-1} \cdot |H| + |L(\mathcal{F}, i)|$$

Furthermore, for every fair instance $(H, G', w, N', \mathcal{F}')$,

$$|L(\mathcal{F}', i)| \leqslant \Delta_{i-1} \cdot |H| \cdot |\mathcal{F}'|$$

Proof: Consider a fair call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ that recurses by adding a neighborhood $N[X]$. Let X_j denote the set of vertices in $L(\mathcal{F}, j) \cap N[X]$, then we can see that $|L(\mathcal{F} \cup \{N[X]\}, j)| \leqslant L(\mathcal{F}, j) + |X_{j-1}|$. Since the call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ adds a neighborhood, $N[X]$, there are no branchable vertices. So, we have that for all $v \in G$, $|N[v] \cap L(\mathcal{F}, j)| < \Delta_j$. Hence $|X_{j-1}| < \Delta_{j-1} \cdot |H|$ and the result $|L(\mathcal{F} \cup \{N[X]\}, i)| \leqslant \Delta_{i-1} \cdot |H| + |L(\mathcal{F}, i)|$ follows.

The second inequality follows by combining induction, the first part the observation, and the fact that if the call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ executes $\mathrm{ALG}_2(H, G', w, N', \mathcal{F}')$, then $|\mathcal{F}| < |\mathcal{F}'|$ if and only if the call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ makes the call $\mathrm{ALG}_2(H, G', w, N', \mathcal{F}')$ by adding a neighborhood. ■

For fair instances $(H, G, w, N, \mathcal{F})$ we define the measure

$$\mu_H(H, G, w, N, \mathcal{F}) = |V(G)| + \sum_i \left(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}} \right) + 2|H| \cdot N \cdot \log(N) \cdot (|H| \cdot |\mathcal{CC}(H)| \cdot \log(N) - |\mathcal{F}|)$$

If $(H, G, w, N, \mathcal{F})$ is not a fair instance, then $\mu_H(H, G, w, N, \mathcal{F})$ is undefined. Note that $\mu_H(H, G, w, N, \mathcal{F})$ must always be an integer and that it is independent of the weight function w. We will say that two instances $(H, G, w, N, \mathcal{F})$ and $(H, G', w', N', \mathcal{F}')$ are

essentially different if $G' \neq G$, $N' \neq N$ or $\mathcal{F}' \neq \mathcal{F}$.

If $N = 1$ then a fair run $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ clearly terminates after a constant number of steps (since in a fair run, $|V(G)| \leqslant N$) regardless of the other inputs, so from now on we will assume $N > 1$.

Lemma 3.4.7. *For every positive integer μ, the number of essentially different fair instances $(H, G, w, N, \mathcal{F})$ such that $\mu_H(H, G, w, N, \mathcal{F}) = \mu$ is finite. In addition, for every fair instance $\mu(H, G, w, N, \mathcal{F}) \geqslant 0$.*

Proof: Consider a fair instance $(H, G, w, N, \mathcal{F})$. We will show that if $\mu_H(H, G, w, N, \mathcal{F}) = \mu$, then $|V(G)| \leqslant \mu$. If $|V(G)| \leqslant \mu$ then the range of inputs for G, N, and $\mathcal{F}$ are bounded in terms of μ and the first part of the Lemma follows.

By Lemma 3.4.5 we have that $|\mathcal{F}|$ is less than $|H| \cdot |\mathcal{CC}(H)| \cdot \log(N)$. It follows that the terms $|V(G)|$, $\Sigma_i(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}})$, and $2|H| \cdot N \cdot \log(N) \cdot (|H| \cdot |\mathcal{CC}(H)| \cdot \log(N) - |\mathcal{F}|)$ are all non negative. Hence $\mu(H, G, w, N, \mathcal{F}) \geqslant |V(G)|$. This also proves that $\mu_H(H, G, w, N, \mathcal{F}) \geqslant 0$. ■

Lemma 3.4.8. *$\mu_H(H, G, w, N, \mathcal{F}) \leqslant 4|H|^2 \cdot |\mathcal{CC}(H)| \cdot N \cdot \log^2(N)$ for every fair instance $(H, G, w, N, \mathcal{F})$.*

Proof: Consider a fair instance $(H, G, w, N, \mathcal{F})$. By Observation 3.4.6 and Lemma 3.4.5, we have that

$$|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}} < |H| \cdot N \cdot |\mathcal{CC}(H)| < |H|^2 \cdot |\mathcal{CC}(H)| \cdot N \cdot \log(N)$$

It follows that

$$\sum_i \left(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}} \right) < |H|^2 \cdot |\mathcal{CC}(H)| \cdot N \cdot \log^2(N)$$

Also, since $N \geqslant |V(G)|$, we have the following.

$$\begin{aligned}\mu_H(H,G,w,N,\mathcal{F}) &= |V(G)| + \Sigma_i(|L(\mathcal{F},i)| \cdot \frac{N}{\Delta_{i-1}}) \\ &\quad + 2|H| \cdot N \cdot \log(N) \cdot (|H| \cdot |\mathcal{CC}(H)| \cdot \log(N) - |\mathcal{F}|) \\ &< |V(G)| + |H|^2 \cdot |\mathcal{CC}(H)| \cdot N \cdot \log^2(N) + 2|H|^2 \cdot |\mathcal{CC}(H)| \cdot N \cdot \log^2(N) \\ &= 4|H|^2 \cdot |\mathcal{CC}(H)| \cdot N \cdot \log^2(N)\end{aligned}$$

■

We define $T_H(H,G,w,N,\mathcal{F})$ to be the running time (including the *number* of oracle calls) of ALG_2 starting with the inputs $(H,G,w,N,\mathcal{F})$. We also define

$$T_H(\mu) = \max_{\substack{G,N,\mathcal{F} \text{ s.t.} \\ \mu_H(H,G,w,N,\mathcal{F}) \leqslant \mu}} \mathrm{T}_H(H,G,w,N,\mathcal{F})$$

Just as for ALG_1, when we analyze run time we assume that arithmetic on weights takes constant time. Thus, both the running time of ALG_2 and the measure of an instance $(H,G,w,N,\mathcal{F})$ are independent of the weight function w, and so by Lemma 3.4.7, $T_H(\mu)$ is well defined.

Lemma 3.4.9. *$T_H(\mu)$ satisfies the following recurrence:*

$$T_H(\mu) \leqslant \mu^{O(1)} + max\begin{cases} T_H(\mu-1) + T_H(\mu[1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}]) \\ T_H(\mu[1 - \frac{1}{4|H| \cdot \log(\mu)}]) \end{cases}$$

Proof: Let $(H,G,w,N,\mathcal{F})$ be a fair instance such that $\mu_H(H,G,w,N,\mathcal{F})) = \mu$ and $T_H(\mu)$ is the run time of $\mathrm{ALG}_2(H,G,w,N,\mathcal{F})$. If the call $\mathrm{ALG}_2(H,G,w,N,\mathcal{F})$ branches on a branchable vertex, v, then it makes two recursive calls, one execution on $(H, G-\{v\}, w, N, \mathcal{F}-\{v\})$, which has measure at most $\mu - 1$. The other execution is

on the instance $(H, G - N[v], w, N, \mathcal{F} - N[v])$. Note that for a branchable vertex, v, we have that

$$\sum_i \left(|L(\mathcal{F} - N[v], i)| \cdot \frac{N}{\Delta_{i-1}} \right) \leqslant \sum_i \left(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}} \right) - \frac{N}{2}$$

since for at least one level i we have that $|N[v] \cap L(\mathcal{F}, i)| \geqslant \Delta_i$ and $\frac{\Delta_i}{\Delta_{i-1}} = 1/2$.

Hence,

$$\begin{aligned}
\mu_H(H, G - N[v], w, N, \mathcal{F} - N[v]) &= |V(G) - N[v]| + \Sigma_i(|L(\mathcal{F} - N[v], i)| \cdot \frac{N}{\Delta_{i-1}}) \\
&\quad + 2|H| \cdot N \cdot \log(N) \cdot (|H| \cdot |\mathcal{CC}(H)| \cdot \log(N) - |\mathcal{F} - N[v]|) \\
&\leqslant |V(G)| + \sum_i \left(|L(\mathcal{F}, i)| \cdot \frac{N}{\Delta_{i-1}} \right) \\
&\quad + 2|H| \cdot N \cdot \log(N) \cdot (|H| \cdot |\mathcal{CC}(H)| \cdot \log(N) - |\mathcal{F}|) - \frac{N}{2} \\
&= \mu - \frac{N}{2} \\
&\leqslant \mu \left(1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(N)} \right) \qquad \text{(by Lemma 3.4.8)} \\
&\leqslant \mu \left(1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)} \right)
\end{aligned}$$

Thus, if the call $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ branches on a branchable vertex, then we have that

$$T_H(\mu) \leqslant T_H(\mu - 1) + T_H\left(\mu[1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}] \right)$$

If $\mathrm{ALG}_2(H, G, w, N, \mathcal{F})$ adds a neighborhood, $N[X]$, it makes a single call $\mathrm{ALG}_2(H, G, w, N,$

$\mathcal{F} \cup \{N[X]\}$). By Observation 3.4.6 and Lemma 3.4.8 we get the following.

$$\mu(H, G, w, N, \mathcal{F} \cup \{N[X]\}) < \mu + |H| \cdot N \cdot \log(n) - 2|H| \cdot N \cdot \log(N)$$
$$< \mu\left([1 - \frac{1}{4|H| \cdot |\mathcal{CC}(H)| \cdot \log(\mu)}]\right) \quad (3.1)$$

Thus, if the call $\text{ALG}_2(H, G, w, N, \mathcal{F})$ adds a neighborhood, then $T_H(\mu) \leqslant T_H(\mu([1 - \frac{1}{4|H| \cdot |\mathcal{CC}(H)| \cdot \log(\mu)}])$. The result now follows from the observation that $\text{ALG}_2(H, G, w, N, \mathcal{F})$ only does $|V(G)|^{O(1)} = \mu^{O(1)}$ work in a given call and always branches on a branchable vertex, adds a balanced separator, or immediately returns a value without making further recursive calls. ■

Since $T_H(\mu)$ is a non negative, non decreasing function, by adding the two possibilities in the max of Lemma 3.4.9 we immediately obtain the following simplified recurrence.

Corollary 3.4.10. $T_H(\mu) \leqslant \mu^{O(1)} + T_H(\mu[1 - \frac{1}{4|H| \cdot \log(\mu)}]) + T_H(\mu - 1) + T_H(\mu[1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}]) < T_H(\mu - 1) + \mu^{O(1)} + 2T_H(\mu[1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}])$

Lemma 3.4.11. $T_H(\mu) = \mu^{O(|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu))}$

Proof: The proof is by induction on μ. The base case is established by Lemma 3.4.7. By Corollary 3.4.10 we have the inequality $T_H(\mu) \leqslant T_H(\mu - 1) + \mu^{O(1)} + 2T_H(\mu[1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}])$ and repeatedly applying the inequality to the first term on the right hand side, gives $T(\mu) \leqslant \mu^{O(1)} + 2\mu T_H(\mu[1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}])$. By the inductive hypothesis then, there is some constant c such that $T_H(\mu)$

$$
\begin{aligned}
&\leqslant \mu^{O(1)} + 2\mu(\mu[1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}])^{c|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu)} \\
&= \mu^{O(1)} + 2\mu\mu^{c|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu)} \left(1 - \frac{1}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}\right)^{c|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu)} \\
&\leqslant \mu^{O(1)} + 2\mu\mu^{c|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu)} \cdot e^{-\frac{c|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu)}{8|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^2(\mu)}} \qquad (\text{ since } (1-x) \leqslant e^{-x} \text{ }) \\
&\leqslant \mu^{O(1)} + 2\mu\mu^{c|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu)} \cdot e^{-\frac{c \log(\mu)}{8}} \\
&\leqslant \mu^{c|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu)} \qquad (\text{ for sufficiently large } c \text{ })
\end{aligned}
$$

■

We are now ready to prove Theorem 3.1.2.

Proof: [Proof of Theorem 3.1.2] The algorithm returns the answer of $\text{ALG}_2(H, G, w, |V(G)|, \emptyset)$. By Lemmata 3.4.1 and 3.4.2, ALG_2 will always terminate and return the weight of a maximum weight independent set in G. For the running time analysis, observe that $(H, G, w, |V(G)|, \emptyset)$ is a fair instance and let $\mu = \mu_H(H, G, w, N, \mathcal{F})$. We assume that $|H| \leqslant N$, since the run time bound follows trivially if $|H| > N$. By Lemma 3.4.8 we have that $\mu < 4|H|^2 \cdot |\mathcal{CC}(H)| \cdot N \cdot \log^2(N)$. Let $n = N = |V(G)|$, then it follows that

$$T_H(H, G, w, N, \mathcal{F}) \leqslant T_H(\mu) = \mu^{O(|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(\mu))} = n^{O(|H|^2 \cdot |\mathcal{CC}(H)| \cdot \log^3(n))}$$

This completes the proof. ■

Theorem 3.1.2 sligthly increases the current reach of Theorem 3.1.1. In particular, let T_k be the graph with k connected components the first of which is a path P_k on k vertices and the remaining $k-1$ are forks (a fork is a path on four vertices plus a single vertex adjacent to the second vertex of the path). Lozin and Milanic [18] gave a polynomial time algorithm for Weighted Independent Set on fork-free graphs. Theorem 3.1.2

implies that Weighted Independent Set on T_k free graphs can be solved by making $n^{O(k^3 \log^3(n))}$ oracle calls to the polynomial time algorithm of Lozin and Milanic [18] or the algorithm of Theorem 3.1.1. Thus we obtain the following result.

Theorem 3.4.12. *There exists an algorithm that given a T_k-free graph G and weight function $w : V(G) \to \mathbb{N}$, runs in $n^{O(k^3 \log^3 n)}$ time, and outputs the weight of a maximum weight independent set of G.*

3.5 Conclusion

In this chapter we gave a quasipolynomial time algorithm for Weighted Independent Set on P_k-free graphs for all integers k. The dependence on k in the exponent is $O(k^2)$ and so our algorithm is quasi-polynomial even for $k = \log^{O(1)} n$ and sub-exponential for $k = n^{\frac{1}{2}-\varepsilon}$ for $\varepsilon > 0$. In light of our algorithm it is tempting to conjecture that (Weighted) Independent Set on P_k-free graphs can be solved in polynomial time for every k. Given how dependent our algorithms are on branching on high degree vertices it looks unlikely that our techniques can lead to polynomial time algorithms for P_k-free graphs. Nevertheless it may be possible to extract structural insights from our algorithms that could eventually lead to polynomial time algorithms.

Our second main result (Theorem 3.1.2) implies that if there exists a quasi-polynomial time algorithm for H-free graphs for every subdivided claw H then there exists a quasi-polynomial time algorithm for every finite family $\mathcal{H}$ such that NP-completeness of Independent Set on $\mathcal{H}$-free graphs does not follow from Alekseev's result [4]. Thus, a quasi-polynomial time algorithm for subdivided-claw-free graphs would complete a dichotomy for the complexity of Independent Set on $\mathcal{H}$-free graphs for every finite family H: every case is either quasi-polynomial time solvable or NP-complete.

Chapter 4

Independent Set in Graphs with no Long Claws in Quasi-Polynomial Time

In this chapter we show that the Maximum Weight Independent Set problem (MWIS) can be solved in quasi-polynomial time on H-free graphs (graphs excluding a fixed graph H as an induced subgraph) for every H whose every connected component is a path or a subdivided claw (i.e., a tree with at most three leaves). This completes the dichotomy of the complexity of MWIS in $\mathcal{F}$-free graphs for any finite set $\mathcal{F}$ of graphs into NP-hard cases and cases solvable in quasi-polynomial time, and corroborates the conjecture that the cases not known to be NP-hard are actually polynomial-time solvable.

The key graph-theoretic ingredient in our result is as follows. Fix an integer $t \geqslant 1$. Let $S_{t,t,t}$ be the graph created from three paths on t edges by identifying one endpoint of each path into a single vertex. We show that, given a graph G, one can in polynomial time find either an induced $S_{t,t,t}$ in G, or a balanced separator consisting of $\mathcal{O}(\log |V(G)|)$ vertex

neighborhoods in G, or an extended strip decomposition of G (a decomposition almost as useful for recursion for MWIS as a partition into connected components) with each particle of weight multiplicatively smaller than the weight of G. This is a strengthening of a result of Majewski et al. [ICALP 2022] which provided such an extended strip decomposition only after the deletion of $\mathcal{O}(\log |V(G)|)$ vertex neighborhoods. To reach the final result, we employ an involved branching strategy that relies on the structural lemma presented above.

4.1 Introduction

The Maximum Weight Independent Set (MWIS) problem takes as input a graph G with vertex weights $\mathfrak{w} : V(G) \to \mathbb{Z}_{\geqslant 0}$ and asks for a set $X \subseteq V(G)$ of maximum possible weight that is *independent* (sometimes also called *stable*): no two vertices of X are adjacent. This classic combinatorial problem plays an important role as a central hard problem in several areas of computational complexity: it appears as one of the NP-hard problems on the celebrated list of Karp [9], it is the archetypical W[1]-hard problem in parameterized complexity [70], and is one of the classic problems difficult to approximate [12].

In the light of the hardness of MWIS within multiple paradigms, one may ask what assumptions on the input make the problem easier. More formally, we can ask for which graph classes $\mathcal{G}$, the assumption that the input graph comes from $\mathcal{G}$ allows for faster algorithms for MWIS. For example, if $\mathcal{G}$ is the class of planar graphs, MWIS remains NP-hard, but the classic layering approach of Baker [15] yields a polynomial-time approximation scheme and simple kernelization arguments give a parameterized algorithm [70].

This motivates a more methodological study of the complexity of MWIS depending on the graph class $\mathcal{G}$ the input comes from. As the space of all graph classes is too wide

and admits strange artificial examples, the arguably simplest regularization assumption is to restrict the attention to hereditary graph classes, i.e., graph classes closed under vertex deletion. Every hereditary graph class $\mathcal{G}$ can be characterized by *minimal forbidden induced subgraphs*: the (possibly infinite) set $\mathcal{F}$ of minimal (under vertex deletion) graphs that are not members of $\mathcal{G}$. Then, we have $G \in \mathcal{G}$ if and only if no member of $\mathcal{F}$ is an induced subgraph of G; when we want to emphasize the set $\mathcal{F}$, we refer to the graph class $\mathcal{G}$ as the class of *$\mathcal{F}$-free graphs* and shorten it to H-free graphs if $\mathcal{F} = \{H\}$.

If a problem turns out to be easier in a class of $\mathcal{F}$-free graphs, in many cases it is a single forbidden induced subgraph $H \in \mathcal{F}$ that is responsible for tractability, and the problem at hand is already easier in H-free graphs. A prime example of this phenomenon are the classes of line graphs and claw-free graphs. Recall that a *line graph* of a graph H is a graph G with $V(G) = E(H)$ where two vertices of G are adjacent if their corresponding edges in H are incident to the same vertex. Observe that MWIS in a line graph G of a graph H becomes the Maximum Weight Matching problem in the pre-image graph H; a problem solvable in polynomial time by deep combinatorial techniques [94]. It turns out that the tractability of MWIS in line graphs can be explained solely by one of the minimal forbidden induced subgraphs for the class of line graphs, namely the *claw* $S_{1,1,1}$. (For integers $a, b, c \geqslant 1$, by $S_{a,b,c}$ we denote the tree with exactly three leaves, within distance a, b, and c from the unique vertex of degree 3.) As proven in 1980, MWIS is polynomial-time solvable already in the class of $S_{1,1,1}$-free graphs [80, 81], called also the class of *claw-free graphs*.

Together with the vastness of the space of all hereditary graph classes, this motivates us to focus on $\mathcal{F}$-free graphs for finite sets $\mathcal{F}$, in particular on the case $|\mathcal{F}| = 1$. This turned out to be particularly interesting for MWIS. As observed by Alekseev [4], for the "overwhelming majority" of finite sets $\mathcal{F}$, MWIS remains NP-hard on $\mathcal{F}$-free graphs. More precisely Alekseev observed that MWIS remains NP-hard on $\mathcal{F}$-free graphs unless,

for at least one graph in $\mathcal{F}$, every connected component is a path or an $S_{a,b,c}$ for some integers a,b,c. Since the original NP-hardness proof of Alekseev [4] in 1982, no new finite sets $\mathcal{F}$ have been discovered such that MWIS remains NP-hard on $\mathcal{F}$-free graphs. We conjecture that this is because all of the remaining cases are actually solvable in polynomial time.

Conjecture 4.1.1. *For every H that is a forest whose every component has at most three leaves,* Maximum Weight Independent Set *is polynomial-time solvable when restricted to H-free graphs.*

Let us remark that Conjecture 4.1.1, if true, would yield a dichotomy for the computational complexity of MWIS on $\mathcal{F}$-free graphs for all finite sets $\mathcal{F}$. This is because for every $\mathcal{F}$ such that NP-hardness of MWIS on $\mathcal{F}$-free graphs does not follow from Alekseev's proof, the class of $\mathcal{F}$-free graphs is contained in the class of H-free graphs for some graph H for which polynomial time solvability of MWIS on H-free graphs follows from Conjecture 4.1.1.

From the positive side, as already mentioned, we know that MWIS is polynomial-time solvable in $S_{1,1,1}$-free graphs since 1980. Around the same time, it was shown that the class of P_4-free graphs (by P_t we denote the path on t vertices) coincides with the class of *cographs* and has very strong structural properties (in modern terms, has bounded cliquewidth) thus allowing efficient algorithms for MWIS and many other combinatorial problems. Apart from a result by Lozin and Milanič proving extending the $S_{1,1,1}$-free case to $S_{1,1,2}$-free graphs [18] (earlier the algorithm for the unweighted variant of the problem was provided by Alekseev [17]), the progress on Conjecture 4.1.1 was limited to various subclasses (see [20, 19, 21, 22, 23, 24, 25, 26, 27, 19, 28, 29, 30, 31, 32] for older and newer results of this kind) until around a decade ago.

The research in the area got significant momentum in the last decade. The progress

can be partitioned into two main threads. The first one focuses on the framework of *potential maximal cliques*, introduced by Bouchitté and Todinca [33], and focuses on providing polynomial-time algorithms for P_t-free graphs for small values of t. A landmark result here is due to Lokshtanov, Vatshelle, and Villanger [14] who were the first to show the usability of the framework in the context of P_t-free graphs by providing a polynomial-time algorithm for MWIS in P_5-free graphs. This has been later extended to P_6-free graphs [34] and related graph classes [35]. A notable property of this framework is that in most cases it not only provides algorithms for MWIS, but for a wide range of problems asking for large induced subgraph of small treewidth, for example Feedback Vertex Set.

The second thread attempts at treating P_t-free or $S_{t,t,t}$-free graphs in full generality, but relaxing the requirements on either the running time (by providing subexponential or quasi-polynomial-time algorithms) or the accuracy (by providing approximation algorithms, such as approximation schemes). Here, the starting point is the theorem of Gyárfás [95, 36] (see also [37]).

Theorem 4.1.2. *Every vertex-weighted graph G contains an induced path Q such that every connected component of $G - N[V(Q)]$ has weight at most half of the weight of G.*

As an induced path in a P_t-free graph has less than t vertices, a P_t-free graph admits a balanced separator (in the sense of Theorem 4.1.2) consisting of at most $t-1$ neighborhoods. Chudnovsky et al. [38] observed that this easily gives a quasi-polynomial-time approximation scheme (QPTAS) for MWIS in P_t-free graphs, and they designed an elaborate argument involving the celebrated three-in-a-tree theorem of Chudnovsky and Seymour [39] to extend the result to the $S_{t,t,t}$-free case and H-free case where H is a forest of trees with at most three leaves each. Abrishami et al. [40] used also the three-in-a-tree theorem to obtain a polynomial-time algorithm for MWIS for $S_{t,t,t}$-free graphs of

bounded degree. Gartland and Lokshtanov showed how to use the theorem of Gyárfás to design exact quasi-polynomial-time algorithm for MWIS in P_t-free graphs [1], for every fixed t. This algorithm was later simplified by Pilipczuk, Pilipczuk, and Rzażewski [41] and the union of the authors of these two papers showed that the approach works for a much wider class of problems and a slightly wider graph class [2]. Last year, Majewski et al. [50] gave a cleaner argument for an existence of a QPTAS for MWIS in $S_{t,t,t}$-free graphs.

This work provides the pinnacle of the second thread by showing that MWIS is quasi-polynomial-time solvable in all cases treated by Conjecture 4.1.1.

Theorem 4.1.3. *For every H that is a forest whose every component has at most three leaves, there is an algorithm for* Maximum Weight Independent Set *in H-free graphs running in time $n^{\mathcal{O}_H(\log^{19} n)}$.*

Here $\mathcal{O}_H$ denotes constants depending on $|H|$ being repressed. Theorem 4.1.3 provides strong evidence in favor of Conjecture 4.1.1, as it refutes the existence of an NP-hardness proof for MWIS for H-free graphs as in Conjecture 4.1.1, unless all problems in NP can be solved in quasi-polynomial time.

4.1.1 Our techniques

As discussed in [1] (in particular Theorem 2), to show Theorem 4.1.3 it suffices to focus on the case $H = S_{t,t,t}$ for a fixed integer $t \geqslant 1$. Together with a simple self-reducibility argument, it is enough to prove the following.

Theorem 4.1.4. *For every integer $t \geqslant 1$, the maximum possible weight of an independent set in a given n-vertex $S_{t,t,t}$-free graph can be found in $n^{\mathcal{O}_t(\log^{16}(n))}$ time.*

Here $\mathcal{O}_t$ denotes constants depending on t being repressed.

The key structural result

While Theorem 4.1.2 provides a balanced separator consisting of a few neighborhoods in a P_t-free graph, it does not seem to be directly usable for $S_{t,t,t}$-free graphs. The example of G being a line graph of a clique (which is $S_{1,1,1}$-free) shows that we cannot hope for merely a balanced separator consisting of a few neighborhoods in $S_{1,1,1}$-free graphs.

However, if G is a line graph, MWIS is solvable in polynomial-time by a very different reason than Theorem 4.1.2: because it corresponds to a matching problem in the preimage graph. Luckily, there is a known formalism capturing decompositions of a graph that are "like a line graph": extended strip decompositions.

For a graph G, a *strip decomposition* consists of a graph H (called the *host*) and a function η that assigns to every edge $e \in E(H)$ a subset $\eta(e) \subseteq V(G)$ such that $\{\eta(e) \mid e \in E(H)\}$ is a partition of $V(G)$ and a subset $\eta(e,x) \subseteq \eta(e)$ for every endpoint $x \in e$ such that the following holds: for every $v_1, v_2 \in V(G)$ with $v_1 \in \eta(e_1)$, $v_2 \in \eta(e_2)$ and $e_1 \neq e_2$ we have $v_1v_2 \in E(G)$ if and only if there is a common endpoint $x \in e_1 \cap e_2$ with $v_1 \in \eta(e_1,x)$ and $v_2 \in \eta(e_2,x)$. Note that if G is the line graph of H, then G has a strip decomposition with host H and $\eta(e) = \{e\}$ for every $e \in E(H) = V(G)$. The crucial observation is that if one provides a strip decomposition (H,η) of a graph G together with, for every $xy \in E(H)$, the maximum possible weight of an independent set in $G[\eta(xy)]$, $G[\eta(xy) - \eta(xy,x)]$, $G[\eta(xy) - \eta(y)]$, and $G[\eta(xy) - (\eta(xy,x) \cup \eta(xy,y)]$ (these graphs are henceforth called *particles*), then we can reduce computing the maximum weight of an independent set in G to the maximum weight matching problem in the graph H with some gadgets attached [38].

An *extended strip decomposition* also allows vertex sets $\eta(x)$ for $x \in V(H)$ and triangle sets $\eta(xyz)$ for triangles xyz in H; a precise definition can be found in preliminaries, but is irrelevant for this overview. Importantly, the notion of a particle generalizes and the

property that one can solve MWIS in G knowing the answers to MWIS in the particles is still true.

Extended strip decompositions come from the celebrated solution to the *three-in-a-tree* problem by Chudnovsky and Seymour.

Theorem 4.1.5 ([39, Section 6], simplified version)**.** *Let G be an n-vertex graph and Z be a subset of vertices with $|Z| \geqslant 2$. There is an algorithm that runs in time $\mathcal{O}(n^5)$ and returns one of the following:*

- *an induced subtree of G containing at least three elements of Z,*
- *an extended strip decomposition (H, η) of G where for every $z \in Z$ there exists a distinct degree-1 vertex $x_z \in V(H)$ with the unique incident edge $e_z \in E(H)$ and $\eta(e_z, x_z) = \{z\}$.*

In a sense, an extended strip decomposition as in Theorem 4.1.5 is a certificate that no three vertices of Z can be connected by an induced tree in G.

Chudnovsky et al. [38] combined Theorem 4.1.2 with Theorem 4.1.5 in a convoluted way to show a QPTAS for MWIS in $S_{t,t,t}$-free graphs; Thereom 4.1.5 is used here to construct an induced $S_{t,t,t}$ in the argumentation. Majewski et al. [50] provided a simpler argument for the existence of a QPTAS: they derived from Theorem 4.1.5 the following structural result.

Theorem 4.1.6 ([50, Theorem 2] in a weighted setting)**.** *For every fixed integer t, there exists a polynomial-time algorithm that, given an n-vertex graph G with nonnegative vertex weights, either:*

- *outputs an induced copy of $S_{t,t,t}$ in G, or*

- *outputs a set $\mathcal{P}$ consisting of at most $11\log n+6$ induced paths in G, each of length at most $t+1$, and a rigid extended strip decomposition of $G-N[\bigcup\mathcal{P}]$ with every particle of weight at most half of the total weight of $V(G)$.*

(Here, rigid means that the extended strip decomposition does not have some unnecessary empty sets; in a rigid decomposition the size of H is bounded linearly in the size of G. The formal statement of Theorem 4.1.6 in [50] is only for uniform weights in G, but as observed in the conclusions of [50], the proof works for arbitrary vertex weights.)

Majewski et al. [50] showed that Theorem 4.1.6 easily gives a QPTAS for MWIS in $S_{t,t,t}$-free graphs, along the same lines as how Chudnovsky et al. [38] showed that Theorem 4.1.2 easily gives a QPTAS for MWIS in P_t-free graphs.

However, it seems that the outcome of Theorem 4.1.6 is not very useful if one aims for an exact algorithm faster than a subexponential one. Our main graph-theoretic contribution is a strengthening of Theorem 4.1.6 to the following.

Theorem 4.1.7. *For every fixed integer t, there exists an integer c_t and a polynomial-time algorithm that, given an n-vertex graph G and a weight function $\mathfrak{w}: V(G)\to[0,+\infty)$, returns one of the following outcomes:*

1. *an induced copy of $S_{t,t,t}$ in G;*
2. *a subset $X\subseteq V(G)$ of size at most $c_t\cdot\log(n)$ such that every component of $G-N[X]$ has weight at most $0.99\mathfrak{w}(G)$;*
3. *a rigid extended strip decomposition of G where no particle is of weight larger than $0.5\mathfrak{w}(G)$.*

That is, we either provide an extended strip decomposition of the *whole* graph (not only after deleting a neighborhood of a small number of vertices as in Theorem 4.1.6)

or a small number of vertices such that deletion of their neighborhood breaks the graph into multiplicatively smaller (in terms of weight) components.

The proof of Theorem 4.1.7 is provided in Section 4.4. Let us briefly sketch it. We start by applying Theorem 4.1.6 to G; we are either already done or we have a set $Z := \bigcup_{P \in \mathcal{P}} V(P)$ of size $\mathcal{O}(\log n)$ and an extended strip decomposition (H, η) of $G - N[Z]$ with small particles. Our goal is now to add the vertices of $N[Z]$ one by one back to (H, η), possibly exhibiting one of the other outcomes of Theorem 4.1.7 along the way. That is, we want to prove the following lemma:

Lemma 4.1.8. *For every fixed integer t there exists an integer c_t and a polynomial-time algorithm that, given an n-vertex graph G, a weight function $\mathfrak{w} : V(G) \to [0, +\infty)$, a real $\tau \geqslant \mathfrak{w}(G)$, a vertex $v \in V(G)$, and a rigid extended strip decomposition (H, η) of $G - v$ with every particle of weight at most 0.5τ, returns one of the following:*

1. *an induced copy of $S_{t,t,t}$ in G;*
2. *a set $Z \subseteq V(G)$ of size at most c_t such that every connected component of $G - N[Z]$ has weight at most 0.99τ;*
3. *a rigid extended strip decomposition of G where no particle is of weight larger than 0.5τ.*

A simple yet important observation for Lemma 4.1.8 is that for $x \in V(H)$ of degree at least two, the set $\bigcup_{y \in N_H(x)} \eta(xy, x)$ can be dominated by at most two vertices, as the sets $\eta(xy, x)$ for $y \in N_H(x)$ are complete to each other. Consequently, if (A, B) is a separation in H of small order, then the part of G that is placed by η in $H[A]$ and the part of G that is placed by η in $H[B]$ can be separated by deleting at most $2|A \cap B|$ vertex neighborhoods in G. Hence, if there is a separation (A, B) in H of constant order

where both these sides have substantial weight (at least 0.01τ), we can provide the second outcome of Lemma 4.1.8.

As $N[v]$ is just one neighborhood, the same observation holds if, instead of looking at (H, η), we look at the inherited extended strip decomposition (H', η') of $G - N[v]$. Here, (H', η') is obtained from (H, η) by first deleting vertices of $N(v)$ from sets $\eta(\cdot)$ and then performing a cleanup operation that trims unnecessary empty sets and ensures that for every $xy \in E(H')$ there is a path in $G[\eta'(xy)]$ between $\eta'(xy, x)$ and $\eta'(xy, y)$. Hence, we can take all separations (A, B) in H' of order bounded by a large constant (depending on t) and orient them from the side that contains less than 0.01τ weight to the side containing almost all the weight of G. This orientation defines a tangle in H'. By classic results from the theory of graph minors, this tangle implies the existence of a large wall W in H' which is always mostly on the "large weight" side of any separation (A, B) of constant order. The cleaning operation ensures that the wall W is also present in (H, η).

An important observation now is that, because (H', η') is cleaned as described below, any family of vertex-disjoint paths in H' projects down to a family of induced, vertex-disjoint, and anti-adjacent paths in G of roughly the same length (or longer): for a path P in H, just follow paths from $\eta(xy, x)$ to $\eta(xy, y)$ in $G[\eta(xy)]$ for consecutive edges xy on P. Furthermore, a wall W is an excellent and robust source of long vertex-disjoint paths.

This allows us to prove that if the neighbors of v are well-connected to the wall W in (H, η) — either they are spread around the wall itself, or one can connect them to W via three vertex-disjoint paths in H — then G contains an induced $S_{t,t,t}$. Otherwise, we show that there is a separation (A, B) in H with the neighbors of v essentially all contained in the sets of $H[A]$, while W lies on the B-side of the separation. (Here, a large number of technical details are hidden in the phrase "essentially contained".) We construct a

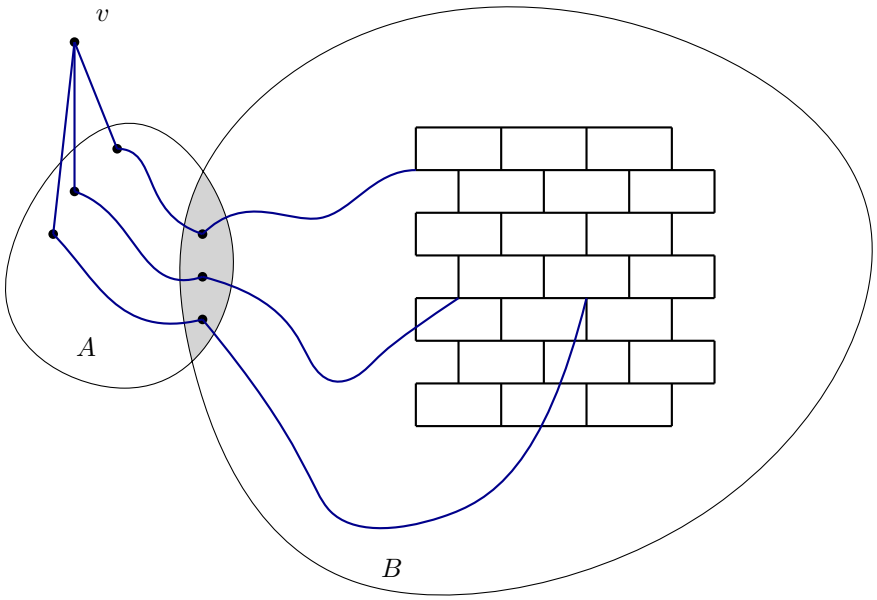


Figure 4.1: Extending a subdivided claw in G_A to an $S_{t,t,t}$ using the large wall W in B.

graph G_A being the subgraph of G induced by the vertices contained in the η sets of $H[A]$, augmented with a set Z of artificial vertices attached to $\bigcup_{y \in N_H(x) \cap A} \eta(xy, x)$ for $x \in A \cap B$; vertices of Z signify possible "escape paths" to the wall W. These "escape paths" allow us to show that any induced tree in G_A that contains at least three vertices of Z lifts to an induced $S_{t,t,t}$ in G, see fig. 4.1. Hence, the algorithm of Theorem 4.1.5 applied to G_A and Z can be used to rebuild $H[A]$ to accommodate v there as well, or to expose an induced $S_{t,t,t}$. This finishes the sketch of the proof of Lemma 4.1.8 and of Theorem 4.1.7.

We would like to highlight a significant difference between previous works [40, 38, 50] and our use of the three-in-a-tree theorem to exhibit an $S_{t,t,t}$ in a graph or obtain an extended strip decomposition. All aforementioned previous works essentially picked three anti-adjacent paths P_1, P_2, P_3 of length t each, with endpoints say x_i and y_i for $i = 1, 2, 3$, removed their neighborhood except for the neighbors of y_is, and called three-in-a-tree for the set $Z = \{x_1, x_2, x_3\}$; note that any induced tree in the obtained graph that

contains Z contains also an induced $S_{t,t,t}$. This method inherently produced extended strip decompositions not for the entire graph, but only for after removal of a number of neighborhoods. Furthermore, it used the assumption of being $S_{t,t,t}$-free only in a very local sense: there is no $S_{t,t,t}$ with paths extendable to the given three vertices of Z. In this work, in contrast, we apply the three-in-a-tree theorem to a potentially much bigger set Z, and use a subdivided wall in the host graph of the extended strip decomposition to extend any induced tree found to an induced $S_{t,t,t}$. In this way, we used the assumption of being $S_{t,t,t}$-free in a more global way than just merely asking for three particular leaves.

Branching

We now proceed with a sketch of our recursive branching algorithm. On a very high level, it is based on techniques used in the quasi-polynomial time algorithm for independent set on P_k-free graphs found in [1], though multiple new ideas are required to make the reasoning work in the setting of $S_{t,t,t}$-free graphs, making both the algorithm and its running time analysis quite a bit more technical. We will soon sketch the algorithm found in [1] and describe how to extend it to $S_{t,t,t}$-free graphs, but first we must address a major barrier. The fact that P_k-free graphs have balanced separators dominated by k vertices, as discussed after Theorem 4.1.2, is a crucial fact used in the algorithm of [1]. But, as mentioned previously, $S_{t,t,t}$-free graphs have no such property (take for instance the line graph of a clique). This is where Theorem 4.1.7 comes to the rescue.

When applying Theorem 4.1.7 to G (the input graph of the current call of the algorithm), since we assume that G is $S_{t,t,t}$-free, we are guaranteed that outcome (1) will not occur. If outcome (3) occurs then we get an extended strip decomposition (H, η) and, as previously mentioned, we can reduce finding a maximum independent set of G to finding a maximum independent set in each particle of (H, η). That is great news, as each particle has at most half of the weight of G, and we can easily employ a divide-and-conquer

strategy by recursively calling the algorithm on each particle of (H, η). So, since outcome (1) never happens and outcome (3) gives us an easy algorithm, we can always assume that outcome (2) happens, that is, that Theorem 4.1.7 gives us a balanced separator of G that is dominated by $\mathcal{O}(\log n)$ vertices, and now we can try to extend the techniques found in [1] to work for $S_{t,t,t}$-free graphs. Therefore, for the rest of this subsection we will focus on sketching an algorithm for independent set on an $S_{t,t,t}$-free graph G such that all induced subgraphs of G have a balanced separator dominated by some constant number of vertices (the stronger assumption of a constant number of vertices versus $\log n$ vertices does not change the algorithm very much and simplifies the discussion).

Before sketching the algorithm let us give a few short definitions around balanced separators for an $S_{t,t,t}$-free graph G (see Section 4.2 for formal definitions of balanced separators). For $n' > 0$, we say that a set $S \subseteq V(G)$ is a *n'-balanced separator* for G if no component of $G - S$ has more than n' vertices. If $A \subseteq V(G)$ and no component of $G - S$ contains over n' vertices of A, we say that S is a n'-balanced separator for (G, A). The outcome (2) of Theorem 4.1.7 gives us a $0.99|A|$-balanced separator for (G, A) dominated by $\mathcal{O}(\log n)$ vertices (again here for simplicity we will assume that these balanced separators are in fact dominated by a constant number of vertices). However, by picking a constant number of balanced separators as provided by Theorem 4.1.7 and taking their union, we can obtain $c|A|$-balanced separators for (G, A) dominated by a constant number of vertices for any fixed $c \in (0, 1)$, so we will assume we have access to such strengthened balanced separators.

Summary of the Quasi-Polynomial Time Algorithm for MWIS on P_k-free Graphs. The starting point for our algorithm is the algorithm for MWIS on P_k-free graphs by Gartland and Lokshtanov [1], who in turn build on an algorithm of Bacsó et al. [37]. We therefore give a brief summary of these algorithms.

We first consider the simple $n^{\mathcal{O}(k\sqrt{n}\log n)}$ time algorithm of Bacsó et al. [37] for MWIS on P_k-free graphs. We begin with an n-vertex P_k-free graph G and branch on all vertices of degree at least $\sqrt{n}$: we either exclude such a vertex from the solution (and thus remove it from the graph), or we include it (and then remove its whole neighborhood from the graph). After this we may assume that the graph in our current instance (we will still refer to this graph as G although some vertices of the original graph G have been removed) now has maximum degree at most $\sqrt{n}$. We solve this instance by finding an $n/2$-balanced separator, S, for G that is dominated by at most k vertices. Since G has maximum degree $\sqrt{n}$ and S is dominated by at most k vertices, S can have size at most $k\sqrt{n}$. We then branch on all $k\sqrt{n}$ vertices of S simultaneously, which then breaks up the graph into small connected components and we recurse on each component. A simple analysis shows that this runs in $n^{\mathcal{O}(k\sqrt{n}\log n)}$ time.

Now, let us try to improve it to an algorithm that runs in time $n^{\mathcal{O}(kn^{1/3}\log n)}$. We first state a modified form of a lemma that appears in [1].

Lemma 4.1.9. *Let G be an n-vertex P_k-free graph and $\mathcal{F}$ a multi-set of subsets of $V(G)$ such that for every $S \in \mathcal{F}$ no component of G has more than $n/2$ vertices. Assume that no vertex belongs to more than c sets of $\mathcal{F}$ counting multiplicity. Then provided $|\mathcal{F}| \geqslant 3ck$, no component of G contains more than $3n/4$ vertices.*

Proof: [Sketch of proof.] Let $S \in \mathcal{F}$ and assume for a contradiction that the largest component of G, call it C, has more than $3n/4$ vertices. Select vertices a, b uniformly at random from C. As $|C| > 3n/4$ the probability that a and b belong to different components of $G - S$ is at least $1/3$. If we let X_S be the random variable that is 1 if a and b are in different components of $G - S$ and 0 otherwise, then $\mathbb{E}[X_S] \geqslant \frac{1}{3}$. By the linearity of expectation, we have $\mathbb{E}[\sum_{S\in\mathcal{F}} X_S] \geqslant \frac{1}{3} \cdot 3ck \geqslant ck$. It follows that there exists vertices $a, b \in S$ such that for at least ck sets, S', in $\mathcal{F}$ (counting multiplicity) a and b

are in different components of $G - S'$. Let $\mathcal{F}'$ be the subset of $\mathcal{F}$ that contains these sets S'. It follows that for any induced path P with a and b as its endpoints, if $S' \in \mathcal{F}'$ then $V(P) \cap S' \neq \emptyset$. Since $\mathcal{F}'$ has at least ck sets and no vertex of P belongs to more than c sets in $\mathcal{F}'$, P must have at least k vertices, contradicting the assumption that G is P_k-free. ■

Now for the $n^{\mathcal{O}(kn^{1/3}\log n)}$ algorithm. We again begin by branching on vertices of high degree, but this time we set the threshold to vertices with degree at least $n^{2/3}$. After this we may assume the graph in our current instance, call it G^1, has maximum degree $n^{2/3}$. We then find a balanced separator, S^1, for G^1 that is dominated by k vertices, hence S has at most $kn^{2/3}$ vertices. We then branch on all vertices with at least $n^{1/3}$ neighbors in S^1. Now we assume the graph considered in our current instance, call it G^2, has maximum degree $n^{2/3}$ and a balanced separator S^1 such that no vertex of G^2 has more than $n^{1/3}$ neighbors in S^1. We then find a balanced separator, S^2, for G^2 that is dominated by k vertices, hence S has at most $kn^{2/3}$ vertices and $S^1 \cap S^2$ has size at most $kn^{1/3}$. We then branch on all vertices with at least $n^{1/3}$ vertices in S^2 and we branch on all vertices that belong to $S^1 \cap S^2$, so S^1 and S^2 "become disjoint". We repeat this $3k$ times until we are in an instance where we have a graph G^{3k} and $3k$ pairwise disjoint balanced separators $S^1, \ldots S^{3k}$. By Lemma 4.1.9, G^{3k} has no component with over $3n/4$ vertices and we then recurse on each component. A somewhat more involved, but still fairly simple analysis shows that this runs in $n^{\mathcal{O}(kn^{1/3}\log n)}$ time.

In the $n^{\mathcal{O}(kn^{1/3}\log n)}$-time algorithm, we branched on vertices that: had over $n^{2/3}$ neighbors, or had $n^{1/3}$ neighbors in any of the balanced separators we picked up, or belonged to two of the balanced separators we picked up. In order to modify this algorithm to run in quasi-polynomial time all that must be done is change the branching threshold. In particular, the algorithm collects balanced separators (each dominated by at most k vertices) and will branch on any vertex that has over $n/2^i$ neighbors that belong to i or

more of the collected balanced separators (the algorithm no longer branches on vertices that only have high degree). Any vertex that belongs to $\log n$ of the collected balanced separators will then be branched on, so no vertex will ever belong to more than $\log n$ of the collected balanced separators. So, by Lemma 4.1.9, after collecting $3k \log n$ of these balanced separators, the graph will not have any large component. A runtime analysis of this algorithm shows that it runs in quasi-polynomial time. Note that in all three algorithms discussed here (the $n^{\mathcal{O}(kn^{1/2}\log n)}$-time, $n^{\mathcal{O}(kn^{1/3}\log n)}$-time, and quasi-polynomial-time algorithm) it is crucial for efficient runtime that the balanced separators we use are dominated by few vertices (they were dominated by k vertices here, but being dominated by polylog(n) vertices would still be sufficent).

Back to $S_{t,t,t}$-free Graphs. Recall that we wish to get a quasi-polynomial time algorithm for MWIS on $S_{t,t,t}$-free graphs for the case where every induced subgraph of the input graph G has a set S of at most c_t vertices such that $N[S]$ is a $n/2$-balanced separator. Up to the bound on the set dominating the separator, this is precisely the case when we keep getting outcome (2) whenever we apply Theorem 4.1.7.

We want to mimic the algorithm for P_k-free graphs. This algorithm used that the input graph is P_k-free in precisely two places. The first is to keep getting constant size sets S such that $N[S]$ is an $n/2$-balanced separator. This is easily adapted to our new setting because we keep getting such sets whenever we apply Theorem 4.1.7.

The second place where P_k-freeness is used is in Lemma 4.1.9, which states that a P_k-free graph cannot have a set of $3k \log n$ balanced separators such that no vertex of G appears in at most $O(\log n)$ of them. If we could strengthen the statement of Lemma 4.1.9 to $S_{t,t,t}$-free graphs we would be done! Unfortunately such a strengthening is false, indeed a path is a counterexample (each vertex close to the middle of the path is a balanced separator).

Nevertheless, a subtle weakening of Lemma 4.1.9 does turn out to be true. In particular, in $S_{t,t,t}$-free graphs it is not possible to pack "very strong" balanced separators that are dominated by "very few" vertices. We will call such balanced separators *c-boosted balanced separators*. A somewhat simplified definition of a c-boosted balanced separator is a set $N[S]$ dominated by a set S of at most c vertices, such that no component of $G - N[S]$ has more than $|V(G)|/16c^2$ vertices (see Definition 4.3.1). It turns out that on $S_{t,t,t}$-free graphs Lemma 4.1.9 is true if "balanced separators" are replaced by "s-boosted balanced separators" for appropriately chosen integer s.

Lemma 4.1.10. *Let G be an n-vertex $S_{t,t,t}$-free graph, s an integer, and $\mathcal{F}$ a multi-set of subsets of $V(G)$ such that every set in $\mathcal{F}$ is an s-boosted balanced separator. Assume no vertex belongs to more than c sets of $\mathcal{F}$. Then, provided $|\mathcal{F}| \geqslant 80sct$, no component of G contains over $3n/4$ vertices.*

We skip sketching the proof of Lemma 4.1.10 here (see Section 4.3.2 for a formal statement and proof of this lemma), but we will remark that one of the key ingredients of the proof is a probabilistic argument akin to the proof of Lemma 4.1.9 (the proof of Lemma 4.3.10 is a bit more involved).

At this point we are one "disconnect" away from being able to utilize the strategy for P_k free graphs: Theorem 4.1.7 keeps giving us balanced separators, while Lemma 4.1.10 tells us that we can't pack *boosted* balanced separators. Indeed, if we assumed our $S_{t,t,t}$-free graphs always had, say, c_t-boosted balanced separators (where c_t is some constant that depends on t), then by the exact same reasoning as before, the strategy of iteratively collecting a c_t-boosted balanced separator and then branching (on all vertices that have over $n/2^i$ neighbors that belong to i or more of the collected c_t-boosted balanced separators) would work. Any vertex that belongs to $\log n$ of the collected c_t-boosted balanced separators will then be branched on, so no vertex will ever belong to over $\log n$

of the collected balanced separators. So, by Lemma 4.1.10, after collecting $80c_t t \log n$ of these c_t-boosted balanced separators, the graph will not have any large component. A running time analysis identical to the one for P_k-free graphs [1] would then show that this algorithm runs in quasi-polynomial time.

Is it possible to bridge the "disconnect" from the other side and keep getting *boosted* balanced separators? This looks difficult, but we are able to bridge the gap algorithmically, by branching in such a way that a "normal" balanced separator becomes boosted. We can then add this boosted balanced separator to our collection of previously created boosted balanced separators, and then apply Lemma 4.1.10 to this collection to conclude that the graph gets sufficiently disconnected before the collection grows too large. We now sketch how to "boost" a separator.

Boosting Separators. We begin with a balanced separator $N[S]$, dominated by a set S of at most c_t vertices, such that no component of $G - N[S]$ has more than $n/2$ vertices. (For technical reasons in the actual algorithm $N[S]$ is not a balanced separator, but rather a set given by Theorem 4.1.6 so that $G - N[S]$ has an extended strip decomposition with no large particles; from the viewpoint of efficient independent set algorithms this is just as useful.) We wish to turn $N[S]$ into a c_t-boosted balanced separator. In order to do this, we consider all vertices of $N[S]$ that have a neighbor in a large component of $G - N[S]$; we call this set $\mathsf{relevant}(G, S)$ (see Figure 4.2. This is a slight simplification of the actual definition of $\mathsf{relevant}(G, S)$ that we use in the algorithm, see Definition 4.3.2). By "large component" we mean any component of $G - N[S]$ that has more than $n/16c_t^2$ vertices (note that if there are no such components, then $N[S]$ is a c_t-boosted balanced separator). In order to branch in a way that turns $N[S]$ into a c_t-boosted balanced separator, we use the following lemma, similar to Lemmas 4.1.9 and 4.1.10.

Lemma 4.1.11. *Let G be an n-vertex $S_{t,t,t}$-free graph, let $N[S]$ be a balanced sepa-*

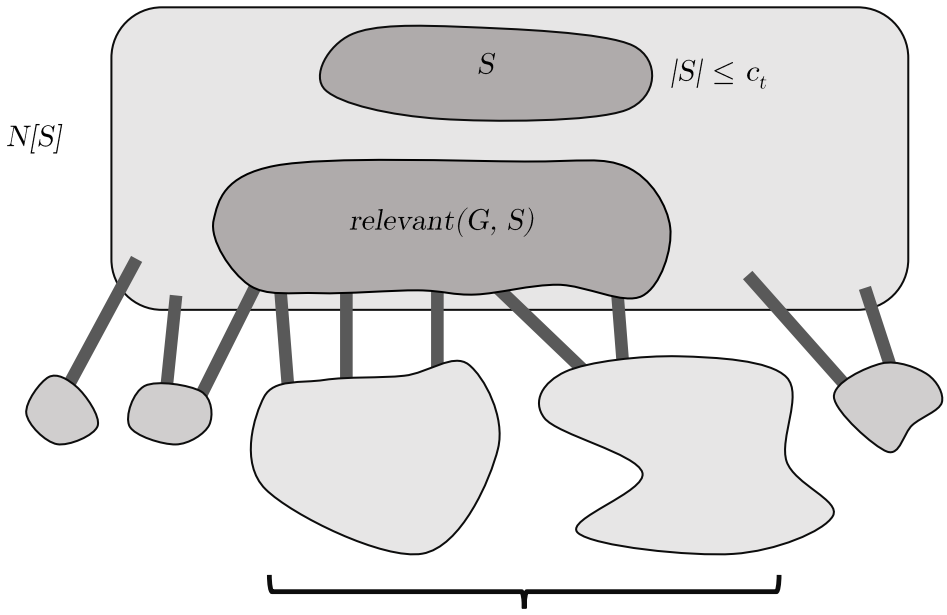


Figure 4.2: Illustration of how the set $\mathsf{relevant}(G, S)$ is obtained from S.

rator for G dominated by a set S of at most c_t vertices, and let $\mathcal{F}$ be a multi-set of $|\mathsf{relevant}(G,S)|/100c_t^3$-balanced separators for $(G, \mathsf{relevant}(G,S))$. Assume no vertex belongs to over c sets of $\mathcal{F}$. If $|\mathcal{F}| \geqslant 10ct$, either S is a c_t-boosted balanced separator or no component of G contains more than $3n/4$ vertices.

The proof of Lemma 4.1.11 follows a similar "expectation argument" that Lemma 4.1.9 uses, although it is a bit more involved. We do not sketch a proof of Lemma 4.1.11 here (this lemma statement is more or less a combination of Observation 4.3.6 and Lemma 4.3.9)

This lemma suggests the following branching strategy. We first pick up an $n/2$-balanced separator $N[S]$ dominated by a set S of c_t vertices, and we will try use Lemma 4.1.11 to turn $N[S]$ into a c_t-boosted balanced separator or break up G into small components. We use the same reasoning as before: iteratively collect $|\mathsf{relevant}(G,S)|/100c_t^3$-

balanced separators for $(G, \mathsf{relevant}(G, S))$ and branch (on all vertices that have over $n/2^i$ neighbors that belong to i or more of the collected balanced separators). Any vertex that belongs to $\log n$ of the collected balanced separators will then be branched on, so no vertex will ever belong to over $\log(n)$ of the collected balanced separators. So, by Lemma 4.1.11 after collecting $10t \log n$ of these $|\mathsf{relevant}(G, S)|/100c_t^3$-balanced separators for $(G, \mathsf{relevant}(G, S))$, either the graph will have no large component (and then we make large progress by calling the algorithm recursively on the components) or S is now a c_t-boosted balanced separator, which we then add to our collection of c_t-boosted balanced separators. By Lemma 4.1.10 this collection cannot grow larger than $80c_t t \log n$ before our graph no longer has large connected components.

The running time analysis of this algorithm essentially looks like this: if we could assume that boosting a single balanced separator to become a boosted balanced separator took constant time, then the analysis would be more or less identical to the analysis of the algorithm for MWIS on P_k-free graphs. However, now each individual "boosting" step is instead a branching algorithm whose analysis again is very similar to the analysis of the algorithm for MWIS on P_k-free graphs, so each boosting step corresponds to a recursive algorithm with quasi-polynomially many leaves. Since quasi-polynomial functions compose the entire running time is still quasi-polynomial. Finally we need to take into account what would happen if outcome (3) of Theorem 4.1.7 does occur, but this can fairly easily be shown to only be good for the progress of the algorithm.

4.2 Preliminaries

We define a *vertex list*, or more simple a *list*, to be an ordered multi-set of subsets $V(G)$. If $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$ is a list and $S \subseteq V(G)$ we define $\mathcal{F} \cup S$ to be the list $\mathcal{F}$ with S appended at the end, that is $\mathcal{F} \cup S = \{F_1, F_2, \ldots, F_k, S\}$. We define

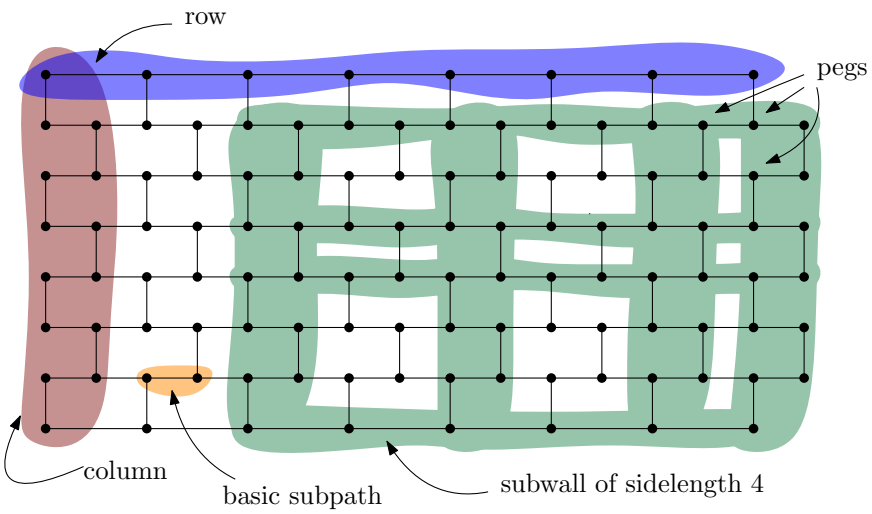


Figure 4.3: Wall of sidelength 8. The lines between pegs denote paths of arbitrary length.

$N_G^{G'}[\mathcal{F}] = \{N_G^{G'}[F_1], N_G^{G'}[F_2], \ldots, N_G^{G'}[F_k]\}$.

Let G be a graph, G' an induced subgraph of G, $Y \subseteq V(G')$, and $\mathfrak{w}$ a weight function for the vertices of G'. We say Y is a c-balanced separator for $(G', \mathfrak{w})$ if no component, C, of $G - Y$ has $\mathfrak{w}(C) > c$. Now let $Z \subseteq V(G')$ such that no component of $G' - Y$ contains over c vertices of Z. Then we say that Y is a *c-balanced separator for* (G', Z). If there is a set $X \subseteq V(G)$ such that $Y = N_G^{G'}[X]$ then we say that Y *has a core X originating in* G. When $Z = V(G')$ then we say that Y is a *c-balanced separator for G' with a core X originating in G* and when $G = G'$ then we say that Y is a *c-balanced separator for G' with a core X*.

Wall notation. A *wall* of *sidelength* ℓ is depicted in Figure 4.3; it consists of ℓ rows and ℓ columns as in the figure. A *peg* is a vertex of degree three in a wall. A path between two pegs that has no other peg as an internal vertex is called a *basic path* in a wall. We say that wall is *k-subdivided* if every basic path has length more than k. A *subwall* of a wall W is a wall whose rows and columns are subpaths of the rows and columns of W.

Separations and tangles. Let G be a graph. A *separation* in G is an ordered pair (A, B) of vertex sets $A, B \subseteq V(G)$ such that $A \cup B = V(G)$ and there is no edge of G with one endpoint in $A - B$ and the second endpoint in $B - A$. The *order* of the separation (A, B) is $|A \cap B|$.

A *tangle of order* k in a graph G is a family $\mathcal{T}$ of separations of order less than k such that:

- For every separation (A, B) of order less than k in G, exactly one of the separations (A, B) and (B, A) belongs to $\mathcal{T}$.
- For every triple $(A_1, B_1), (A_2, B_2), (A_3, B_3) \in \mathcal{T}$ we have $A_1 \cup A_2 \cup A_3 \neq V(G)$.

Observe that if $\mathcal{T}$ is a tangle of order k and $k' < k$, then the set $\mathcal{T}'$ consisting of all separations of $\mathcal{T}$ of order less than k' is a tangle of order k'. We call such $\mathcal{T}'$ the *restriction of* $\mathcal{T}$ *to order* k'.

Let W be a wall in G of sidelength k. Let (A, B) be a separation in G of order $k' < k$. Note that for exactly one $\Gamma \in \{A - B, B - A\}$, Γ contains at least $k - k'$ full rows and at least $k - k'$ full columns of W. Let $\mathcal{T}_W$ be the set of those separations (A, B) of order less than $\lceil k/3 \rceil$ such that $B - A$ contains at least $k - \lceil k/3 \rceil + 1$ full rows and at least $k - \lceil k/3 \rceil + 1$ full columns of W. It is straightforward to verify that $\mathcal{T}_W$ is a tangle of order $\lceil k/3 \rceil$; we call it the tangle *governed by* W.

We make the following simple but important observation.

Lemma 4.2.1. *If W is a wall in a graph G and W' is a subwall of W, then $\mathcal{T}_{W'} \subseteq \mathcal{T}_W$.*

Proof: Let k and k' be the sidelengths of W and W', respectively. Let $(A, B) \in \mathcal{T}_W$ be a separation of order less than $\lceil k'/3 \rceil$. Then, $B - A$ contains at least $k - \lceil k'/3 \rceil + 1$ full rows of W and at least $k - \lceil k'/3 \rceil + 1$ full columns of W. Since W' is a subwall of W, $B - A$ contains at least $k' - \lceil k'/3 \rceil + 1$ full rows of W' and at least $k' - \lceil k'/3 \rceil + 1$ full columns of W'. Hence, $(A, B) \in \mathcal{T}_{W'}$, as desired. ■

We will need the following result, which follows from the combination of the polynomial grid minor theorem [96, 97], the duality of tangles and branchwidth [98], and Lemma 14.6 of [99].

Theorem 4.2.2. *There exists a function $f_{\mathrm{KTW20}}(k) = \widetilde{O}(k^{18})$ such that if a graph G admits a tangle $\mathcal{T}$ of order $f_{\mathrm{KTW20}}(k)$ for an integer k, then G contains a wall W of sidelength $3k$ such that $\mathcal{T}_W$ is the restriction of $\mathcal{T}$ to order k.*

4.3 The Algorithm

In this section, we define $\log(n) = \max(2, \lceil \log_2(n) \rceil)$. Let t be a positive integer, throughout this section, we will use c_t to denote the constant given in Lemma 4.4.1 of the same name. In order to make dealing with constants easier (in particular the constants that arise from Definition 4.3.4), we will assume that $c_t \geqslant 34t$. Additionally, in this section we will assume that all graphs, G, come equipped with a weight function $\mathfrak{w} : V(G) \to [0, +\infty)$. If G' is an induced subgraph of G', then we assume that G inherits its weight function from G, that is the weight function for the vertices of G' is the weight function for the vertices of G when restricted to the vertices of G'. For a subset $X \subseteq V(G)$, $\mathfrak{w}(X) = \sum_{x \in X} \mathfrak{w}(x)$.

4.3.1 Definitions and Observations

In this subsection we collect most of the definitions we will use for this section and immediate observations about these definitions.

The interpretation of G and G' in the coming definitions will be as follows: we are running the algorithm in order to find the maximum size independent set of G. The algorithm is recursive, and only makes recursive calls on induced subgraphs of the input

graph. Suppose we want to analyze a recursive call in which the current induced subgraph of G that is being considered is G'. When arguing about the behavior of the algorithm on input G', it is useful to be able to conclude that the bigger graph G contains an $S_{t,t,t}$.

Throughout our algorithm, balanced separators as defined in Section 4.2 will often be readily available. However we will sometimes need balanced separators with even stronger properties; in particular we will need the amount that the separator disconnects the graph to depend super-linearly in the size of its core. We will call such balanced separators *boosted* (see Definition 4.3.1 below), and a substantial part of our algorithm will consist of trying to reduce the input graph G so that some vertex set becomes boosted (in the sense of Definition 4.3.1). Unlike for normal balanced separators, the following definition is always used with $Z = V(G')$, so reference to Z is dropped in the following definition.

Definition 4.3.1 (s-boosted balanced separator)**.** Let G be a graph, G' an induced subgraph of G, and s be a positive integer. A vertex set $Y \subseteq V(G')$ is an *s-boosted balanced separator for G' with a core X originating in G* if Y is a c-balanced separator for G' with core X originating in G, $|X| \leqslant s$, and $c \leqslant \frac{|C|}{16s^2}$, where C is a largest component of G'. When G and G' are clear from context, we may say that X is a core of the boosted balanced separator, Y.

The algorithm will often work with a graph G, a vertex set X in G and an induced subgraph G' of G. The aim is to ensure that X is a core of an s-boosted balanced separator (for an appropriately chosen s) of G'. If X doesn't already satisfy this, it is because $G' - Y$, where $Y = N_G^{G'}[X]$, has some connected components that are too big. The next definition zooms in on the neighborhood of these connected components into Y (the constants in the formal definition don't quite match the intuition above for book keeping reasons).

Definition 4.3.2 (relevant set)**.** Let t be a positive integer, G an $S_{t,t,t}$-free graph, $X \subseteq V(G)$, G' an induced subgraph of G, and N a positive integer. We define $\mathsf{relevant}_G(G', X, N)$ to be the subset of vertices of $N_G^{G'}[X]$ that have at least one neighbor in at least one component of $G' - N_G^{G'}[X]$ that contains over $\frac{N}{32c_t^2 \log^2(N)}$ vertices.

We make the following important observation about $\mathsf{relevant}_G(G', X, N)$ that follows directly from the fact that if G'' is an induced subgraph of G', then every component of $G'' - N_G^{G''}[X]$ of size at least $\frac{N}{32c_t^2 \log^2(N)}$ vertices is contained in some component of $G' - N_G^{G'}[X]$ of size at least $\frac{N}{32c_t^2 \log^2(N)}$.

Observation 4.3.3. *Let t be a positive integer, G an $S_{t,t,t}$-free graph, $X \subseteq V(G)$, G' an induced subgraph of G, G'' and induced subgraph of G', and N a positive integer. Then $\mathsf{relevant}_G(G'', X, N) \subseteq \mathsf{relevant}_G(G', X, N)$.*

Note that in an $S_{t,t,t}$-free graph, Theorem 4.1.6 will always return a family $\mathcal{F}$ satisfying the second bullet point of the theorem statement. This motivates the following definition.

Definition 4.3.4 (esd and inferred extended strip decomposition)**.** Let t be a positive integer and G an n-vertex $S_{t,t,t}$-free graph. We define $\mathsf{esd}(G)$ to be a subroutine that uses Theorem 4.1.6 to return a set $X \subseteq V(G)$, $|X| \leqslant (t+1)(11\log(n)+6) \leqslant 34t\log(n) \leqslant c_t \log(n)$ such that $G - N_G[X]$ has a rigid extended strip decomposition, (H, η), where no particle of (H, η) has over $|G|/2$ vertices. Furthermore, this subroutine runs in time polynomial time.

Additionally, if G' is an induced subgraph of G, we define the extended strip decomposition *inferred by* (X, G'), call it (H', η'). For each component, C, of G' that does not contain a vertex of $N_G^{G'}[X]$, H' contains an isolated copy H_c of H, and for all vertices, edges, and triangles, $\mathcal{R}_c$, of H_c let $\mathcal{R}$ be the corresponding vertex, edge, or triangle in H. We set $\eta'(\mathcal{R}_c) = \eta(\mathcal{R}) \cap C$. For each component C^* of G' that contain at least one vertex of $N_G^{G'}[X]$, H' contains an isolated vertex v_{c^*} and $\eta'(v_{c^*}) = C^*$.

It follows from the definition of extended strip decomposition that (H', η') is a valid extended strip decomposition of G'. Note that the extended strip decomposition inferred by (X, G') can be computed in polynomial time since we have access to the extended strip decomposition, (H, η), by Theorem 4.1.6 and since H has only $\mathcal{O}(n)$ vertices (because it is rigid, see the discussion after Theorem 4.1.6), H' has $n^{\mathcal{O}(1)}$ vertices and therefore $n^{\mathcal{O}(1)}$ particles. Furthermore, note that for any particle P of (H', η') either P is equal so some component C^* that contains at least one vertex of $N_G^{G'}[X]$ (when $P = \eta'(v_{c^*})$) or P is equal to to $P' \cap C$ where P' is a particle of (H, η) and C is a component of G' that does note contain any vertices of $N_G^{G'}[X]$. This leads to the next two observation. But we first give one additional definition related to extended strip decompositions which is meant to capture when each particle of our extended strip decomposition is "small enough" so that we make enough progress when we recursively call the algorithm on each particle.

Let t be a positive integer, G an $S_{t,t,t}$-free graph, N a natural number, and (H, η) an extended strip decomposition of G. We say that (H, η) is an *N-good* extended strip decomposition if no particle of (H, η) has over $(1 - \frac{1}{32c_t^2 \log^2(N)})N$ vertices of G. Note that these are the same constants used in the definition of relevant, the reason for this will become apparent in Observation 4.3.6.

The first observation follows from the fact that the size of the largest particle of the extended strip decomposition inferred by (X, G') is bounded by the size of the largest component of G'.

Observation 4.3.5. *Let t be a positive integer, N a natural number, G an $S_{t,t,t}$-free graph, G' an induced subgraph of G, and $X \subseteq V(G)$. If no component of G' has over $(1 - \frac{1}{32c_t^2 \log^2(N)})N$ vertices then the extended strip decomposition inferred by (X, G') is N-good.*

Observation 4.3.6. *Let t be a positive integer, G an n-vertex $S_{t,t,t}$-free graph, N a*

natural number, G' an induced subgraph of G, and $X \subseteq V(G)$ such that $G - N_G[X]$ has an extended strip decomposition, (H, η), where no particle contains over $N/2$ vertices. If $\mathsf{relevant}_G(G', X, N) = \emptyset$ then either $N_G^{G'}[X]$ is an $\frac{N}{32c_t^2 \log^2(N)}$-balanced separator for G' or the extended strip decomposition inferred by (X, G') is N-good.

Proof: Let t, N, G, G', X, and (H, η) be as in the statement of the lemma and (H', η') be the extended strip decomposition inferred by (X, G'). If there are no components of $G' - N_G^{G'}[X]$ that contain at least $\frac{N}{32c_t^2 \log^2(N)}$ of the vertices of G then we are done (as $N_G^{G'}[X]$ would be an $\frac{N}{32c_t^2 \log^2(N)}$ balanced separator), so we may assume that there is a component C of $G' - N_G^{G'}[X]$ that contains at least $\frac{N}{32c_t^2 \log^2(N)}$ vertices of G and $N_{G'}[C] \cap N_G^{G'}[X] = \emptyset$ by the assumption that $\mathsf{relevant}_G(G', X, N) = \emptyset$. It follows that C is a component of G' and therefore any component of G' that contains at least one vertex of $N_G^{G'}[X]$ has at most $(1 - \frac{1}{32c_t^2 \log^2(N)})N$ vertices. This combined with the note about particles made just after Definition 4.3.4 (that every particle of (H', η') is either a component, C, of G that contains at least one vertex of $N_G^{G'}[X]$ - which can have size at most $(1 - \frac{1}{32c_t^2 \log^2(N)})N$ - or a subset of a particle of (H, η) - which by assumption has at most $N/2$ vertices) proves the observation. ■

4.3.2 Preliminary Lemmas

We will now present a few lemmas that will be useful to have in hand before describing the algorithm. Given a graph G and an extended strip decomposition (H, η) for G the following lemma shows that solving independent set on G can be reduced to solving independent set on each particle of G. This reduction first appears in [38], the version we cite here is derived from [40] (Lemma 5.2).

Lemma 4.3.7 ([38, 40])**.** *Let G be an n-vertex graph and let (H, η) be an extended strip decomposition of G where H has N vertices. Furthermore, assume that for each particle,*

P, of (H, η), we know the weight of a maximum weight independent set of $G[P]$. Then in time polynomial in $n + N$ we can compute the weight of a maximum weight independent set of G.

Let G be a graph and (H, η) an extended strip decomposition for G such that for each particle, P, of (H, η) the weight of a maximum weight independent set of $G[P]$ is known. We use $\mathsf{matching}(H, \eta)$ to denote the output (the weight of a maximum weight independent set of G) of running the algorithm of Lemma 4.3.7.

Lemma 4.3.8. *Let t be an postive integer, G an n-vertex $S_{t,t,t}$-free graph, $A \subseteq V(G)$, and $i \leqslant \log(n)$ a natural number. Either G contains a set C such that $N_G[C]$ is an $(|A|/2^i)$-balanced separator for (G, A) and $|C| \leqslant (c_t)(70)2^{i+1}\log(n)$ or G has a rigid extended strip decomposition, (H, η), such that no particle contains over $(1 - 1/2^{i+2})|A|$ vertices of A. Furthermore, either C or (H, η) can be found in polynomial time.*

Proof: Let t, G, n, and A be a in the statement of this lemma. We first claim that either G contains a set C such that $N_G[C]$ is an $(|A|/2)$-balanced separator for (G, A) and $|C| \leqslant 70c_t \log(n)$ or G has an extended strip decomposition, (H, η), such that no particle contains over $(1 - 1/4)|A|$ vertices of A and either C or (H, η) can be found in polynomial time.

In order the prove this we will consider a process consisting of at most 70 step. At the j^{th} step we will assume we have a set C_j such that $N_G[C_j]$ is an $(|A| \cdot 0.99^j)$-balanced separator for (G, A) and $|C| \leqslant c_t j \log(n)$ (we have $C_0 = \emptyset$ for the base case). Given such a C_j, we show how to find C_{j+1} or find a rigid extended strip decomposition (H, η) of G such that no particle has over $(1 - 1/4)|A|$ vertices of A. If $N_G[C_j]$ is already an $|A|/2$-balanced separator for (G, A) then we are done, so assume this does not happen, let X be the component of $G - N_G[C_j]$ that contains over half the vertices of A and let $X_A = X \cap A$.

We apply Lemma 4.4.1 to G where all vertices of X_A have weight 1 and all other vertices have weight 0. Outcome (1) cannot occur as G is $S_{t,t,t}$-free. If outcome (2) occurs, then we get a set X_C such that $N_G[X_C]$ is an $(|X_A| \cdot 0.99)$-balanced separator for (G, X_A) and $|X_C| \leqslant c_t \log(n)$. We set $C_{j+1} = C_j \cup X_C$, it holds then that $N_G[C_{j+1}]$ is an $(|A| \cdot 0.99^{j+1})$-balanced separator for (G, A) and $|C_{j+1}| \leqslant |C_t| + |X_c| \leqslant c_t(j+1)\log(n)$, as desired. If outcome (3) occurs then we get a rigid extended strip decomposition (H, η) for G such that no particle of (H, η) contains over half of X_A. Since $|X_A| \geqslant |A|/2$ it follows that no particle of (H, η) contains over $(1 - 1/4)|A|$ vertices of A. Since $0.99^{70} < .5$ this process must end by the 70^{th} step. Since Lemma 4.4.1 runs in polynomial time, this process runs in polynomial time.

We now prove the full statement of this lemma in a similar manner. Fix some natural number $i \leqslant \log(n)$. In order the prove this we will consider a process consisting of at most i step. At the j^{th} step, $j < i$, we will assume we have a set C_j such that $N_G[C_j]$ is an $|A|/2^j$-balanced separator for (G, A) and $|C| \leqslant 70c_t \cdot 2^{j+1}\log(n)$ (we have $C_0 = \emptyset$ for the base case). So, assume C_j satisfies these properties, we show how to find C_{j+1} or find a rigid extended strip decomposition (H, η) of G such that no particle has over $(1 - 1/2^{j+2})|A|$ vertices of A (we have $C_0 = \emptyset$ for the base case).

Consider each component, X, of $G - N_G[C_j]$ that contain at least $|A|/2^{j+1}$ vertices of $|A|$, there are at most 2^{j+1} such components, set $X_A = X \cap A$. For each X and corresponding X_A we apply the claim from the first paragraph of this proof to G where all vertices of X_A have weight 1 and all other vertices have weight 0. The first possibility is for each X and X_A we get a set X_C such that $N_G[X_C]$ is an $|X_A|/2$-balanced separator for (G, X_A) and $|X_C| \leqslant 70c_t \log(n)$. Then we set $C_{j+1} = C_j \cup \bigcup_X X_C$, it holds then that

$N_G[C_{j+1}]$ is an $(|A|/2^{j+1})$-balanced separator for (G, A) and

$$|C_{j+1}| \leqslant |C_j| + \sum_X |X_C| \leqslant 70c_t \cdot 2^{j+1}\log(n) + 70c_t \cdot 2^{j+1}\log(n) = 70c_t \cdot 2^{j+2}\log(n),$$

as desired. The other possibility is that for at least one X we get a rigid extended strip decomposition (H, η) for G such that no particle of (H, η) contains over half of X_A. Since $|X_A| \geqslant |A|/2^{j+1}$ it follows that no particle of (H, η) contains over $(1 - 1/2^{j+3})|A|$ $\leqslant (1 - 1/2^{i+2})$ vertices of A (since $j < i$).

Repeating this $i \leqslant \log(n)$ times (or until we get a desired extended strip decomposition) then yields the result. Since each step applies Lemma 4.4.1 less than n time and Lemma 4.4.1 runs in polynomial time and there are at most $\log(n)$ steps, this process runs in polynomial time. ■

Cannot Pack Many Balanced Separators

Recall the following notation from Section 4.2: Let G be a graph, G' an induced subgraph of G, and $X \subseteq V(G)$. We define $N_G^{G'}[X]$ to mean $N_G[X] \cap V(G')$. Furthermore, let $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$ be a list. Then $N_G^{G'}[\mathcal{F}] = \{N_G^{G'}[F_1], N_G^{G'}[F_2], \ldots, N_G^{G'}[F_k]\}$.

Lemma 4.3.9. *Let t be a positive integer, G an n-vertex $S_{t,t,t}$-free graph, $N \leqslant |G|$ a natural number, G' an induced subgraph of G, $X \subseteq V(G)$, and $\mathcal{F}$ a list of vertex sets of G. Assume that $\mathsf{relevant}_G(G', X, N) \neq \emptyset$ and all sets of $N_G^{G'}[\mathcal{F}]$ are $\frac{|\mathsf{relevant}_G(G',X,N)|}{100c_t^2\log^2(N)|X|}$-balanced separators for $(G', \mathsf{relevant}_G(G', X, N))$ such that no vertex of G' belongs to over c sets of $N_G^{G'}[\mathcal{F}]$. If $|\mathcal{F}| \geqslant 10tc$ then G contains an induced $S_{t,t,t}$.*

Proof: Let t, G, G', N, X, and $\mathcal{F}$ have the same meaning as in the statement of this lemma. Among all components of $G' - N_G^{G'}[X]$ that have at least $\frac{|G'|}{32c_t^2\log^2(N)}$ vertices, let C denote the one such that the size of $C^* = N_{G'}[C] \cap \mathsf{relevant}_G(G', X, N)$ is maximized. Since

there are at most $32c_t^2 \log^2(N)$ components of $G' - N_G^{G'}[X]$ that have at least $\frac{|G'|}{32c_t^2 \log^2(N)}$ vertices and by definition all vertices of $\mathsf{relevant}_G(G', X, N)$ have at least one neighbor in a component of $G' - N_G^{G'}[X]$ of size at least $\frac{|G'|}{32c_t^2 \log^2(N)}$, it holds that $|C^*| \geqslant \frac{|\mathsf{relevant}_G(G',X,N)|}{32c_t^2 \log^2(N)}$. Next for all vertices in X let x be one such that $C^{*,x} = N_G^{G'}[x] \cap C^*$ is maximized, since $N_G^{G'}[X]$ dominates C^* it holds that $|C^{*,x}| \geqslant \frac{|C^*|}{|X|} \geqslant \frac{|\mathsf{relevant}_G(G',X,N)|}{32c_t^2 \log^2(N)|X|} > 0$ (the last inequality following from the assumption $\mathsf{relevant}_G(G', X, N) \neq \emptyset$).

Now let $f \in \mathcal{F}$ and let a, b, c be three independently and uniformly at random (with replacement) chosen vertices of $C^{*,x}$. We calculate the probability that no two vertices among a, b, c belong to the same component in $G' - N_G^{G'}[f]$. Since $|C^{*,x}| \geqslant \frac{|\mathsf{relevant}_G(G',X,N)|}{32c_t^2 \log^2(N)|X|}$ and $N_G^{G'}[f]$ is a $\frac{|\mathsf{relevant}_G(G',X,N)|}{100c_t^2 \log^2(N)|X|}$-balanced separator for $(G', \mathsf{relevant}_G(G', X, N))$ and $C^{*,x} \subseteq \mathsf{relevant}_G(G', X, N)$, we have that $N_G^{G'}[f]$ is a $\frac{|C^{*,x}|}{3}$-balanced separator for $(G', C^{*,x})$. So, since no component of $G - N_G^{G'}[f]$ has over $\frac{|C^{*,x}|}{3}$ vertices of $C^{*,x}$ there is at at least a $\frac{2}{3}$ probability that a and b are in different components in $G - N_G^{G'}[f]$ and there is at least a $\frac{1}{3}$ probability that c is in a different component from a and b conditioned on a and b being in different components. It follows that there is at least a $\frac{2}{3} \cdot \frac{1}{3} = \frac{2}{9}$ probability that no two vertices among a, b, c belong to the same component in $G' - N_G^{G'}[f]$.

Hence if X_f represents the random variable that is 1 if the independently and uniformly at random chosen $a, b, c \in C^{*,x}$ are in three different components in $G' - N_G^{G'}[f]$ and 0 otherwise, the expected value $\mathbb{E}[X_f] \geqslant \frac{2}{9}$. Then by the linearity of expectation, we have that $\mathbb{E}[\sum_{f \in \mathcal{F}} X_f] \geqslant \frac{2}{9} \cdot 10tc > 2tc$. Thus, there must exists a triple, $a, b, c \in C^{*,x}$, such that for a subset of $\mathcal{F}$, call it $\mathcal{F}^*$, of size greater than $2tc$, a, b, and c are in three different components in $G' - N_G^{G'}[f^*]$ for all $f^* \in \mathcal{F}^*$. This implies that any path P in G' with a and b as its endpoints must have over $2t$ vertices because for all $f^* \in \mathcal{F}^*$ $N_G^{G'}[f^*] \cap P \neq \emptyset$ (or else a and b would be in the same component of $G' - N_G^{G'}[f^*]$) and if P had at most $2t$ vertices, since $|\mathcal{F}^*| > 2tc$, that would force some vertex of P to belong to over c sets in $N_G^{G'}[\mathcal{F}^*]$, contrary to assumption. Similarly, all paths with a and c or b

and c with its endpoint must have over $2t$ vertices as well.

Now we show there exists three anti-complete induced paths, P_a, P_b and P_c, each with t vertices such that a, b, and c are one of the endpoints of P_a, P_b and P_c respectively, and all other vertices of these paths belong to C (recall from the first paragraph of this proof that C is the component of $G' - N_G^{G'}[X]$ such that $C^* = N_{G'}[C] \cap \mathsf{relevant}_G(G', X, N)$. Since $C^{*,x} \subseteq C^*$, all vertices of $C^{*,x}$, which includes a, b, and c, have a neighbor in C). To locate P_a, take a shortest path from a to b with all internal vertices in C (since a and b both have neighbors in C and C is connected, such a path exists). By the previous paragraph this path must have at least $2t$ vertices, so let P_a be the first t vertices of this path, so P_a is a path with t vertices such that a is one endpoint of the path and all other vertices are in C. Identical arguments show there are induced paths P_b and P_c which have b and c as their endpoints, respectively, and all other other vertices are in C. Furthermore, if P_a, P_b and P_c were not anti-complete, then that would imply that there exists paths between two vertices of $\{a, b, c\}$ with at most $2t$ vertices, which contradicts the conclusion of the previous paragraph.

Lastly, note that since C is a component of $G' - N_G^{G'}[X]$ and $x \in X$, x has no neighbors in C. So, since x is neighbors with a, b, and c and P_a, P_b and P_c are anti-complete, x along with P_a, P_b and P_c form an $S_{t,t,t}$. ■

Cannot Pack Many Boosted Balanced Separators

Lemma 4.3.10. *Let G be a graph and G' be an induced subgraph of G, and s, t, c be positive integers. If there exists a list $\mathcal{F}$ of s-boosted balanced separators of G' originating in G such that $|\mathcal{F}| \geqslant 80 \cdot s \cdot t \cdot c$, and no vertex of G' belongs to over c sets of $\mathcal{F}$, then G contains an $S_{t,t,t}$.*

This subsection is devoted to the proof of Lemma 4.3.10. Thus, within this sub-

section we will assume that the premise of Lemma 4.3.10 holds. Towards the proof of Lemma 4.3.10, we set $\mathcal{F} = \{Y_1, Y_2, \dots Y_\ell\}$. For every Y_i in $\mathcal{F}$ we let X_i be a core of Y_i originating in G. In other words, $Y_i = N_G^{G'}[X_i]$, and $|X_i| \leqslant s$. Additionally, C is the largest component of G', and $r = 4s$.

Lemma 4.3.11. *There exists a set $R \subseteq C$ of size r and subfamily $\mathcal{F}_1 \subseteq \mathcal{F}$ such that $|\mathcal{F}_1| > |\mathcal{F}|/2$, and for every $Y_i \in \mathcal{F}_1$, each connected component of $G' - Y_i$ contains at most one vertex of R.*

Proof: We pick a tuple $R = v_1, v_2, \dots, v_r$ of $r = 4s$ vertices from C uniformly at random (with repetition). Consider an arbitrary $Y_i \in \mathcal{F}$. For each choice of $1 \leqslant p < q \leqslant r$, the probability that v_q is in the same component of $G' - Y_i$ as v_p is at most $\frac{|C|}{16s^2}$. The union bound over all choices of p, q yields that the probability that no component of $G' - Y_i$ contains at least two vertices of R is at least $1 - \binom{r}{2} \cdot \frac{1}{16s^2} > 1/2$. Define $\mathcal{F}_1$ to be the set of all Y_i's in $\mathcal{F}$ such that no component of $G' - Y_i$ contains at least two vertices of R. The expected size of $\mathcal{F}_1$ is strictly larger than $|\mathcal{F}|/2$, and there exists at least one choice of R that achieves expectation, proving the statement of the lemma. ■

We will use the following well-known facts about trees, that we will state without proof.

Observation 4.3.12. *A tree with k leaves has at most $k-1$ vertices of degree at least 3.*

Observation 4.3.13. *Let T be a tree and P be a path in the tree such that all vertices on P have degree 2 in T. Then $T - V(P)$ has precisely two connected components.*

For the rest of the proof of Lemma 4.3.10, let R and $\mathcal{F}_1$ be as given in the statement of Lemma 4.3.11, G^* be an inclusion minimal connected induced subgraph of $G'[C]$ containing R, and T^* be a spanning tree of G^*. Define M as R plus all the vertices of T^* that have degree at least three in T^*. Finally, set $M^* = N_{T^*}[M]$. In the next lemma we collect a few simple observations about G^*, T^*, M, and M^*.

Lemma 4.3.14. *G^*, T^*, M and M^* have the following properties:*

1. *All leaves of T^* are in R,*
2. *$|M| \leqslant 2|R|$,*
3. *there are at most $|M| - 1$ connected components of $T^* - M$,*
4. *$|M^*| \leqslant 6|R|$,*
5. *each edge uv of G^* is an edge of T^* or has at least one endpoint in M^*.*

Proof: For (1), note that $G^* - v$ is connected for every leaf v of T^*. Thus $v \notin R$ would contradict minimality of G^*. For (2) we note that (1) implies that T^* has at most $|R|$ leaves, and therefore (by Observation 4.3.12) at most $|R| - 1$ vertices of degree at least 3.

For (3) and (4) we note that every vertex in $V(T^*) - M$ has degree precisely 2 in T^*. Thus every component P of $T^* - M$ is a path (on one or more vertices), only the endpoints of the path are neighbors (in T^*) of M, and $|N_{T^*}(P)| = 2$. Let κ be the number of connected components of $T^* - M$. Then κ applications of Observation 4.3.13 implies that $T^*[M]$ has at least $1+\kappa$ components. Hence $1+\kappa \leqslant |M|$, proving (3). Since each component of $T^* - M$ contains at most two neighbors (in T^*) of M it follows that $|N_{T^*}(M)| \leqslant 2|M|$, and hence $|M^*| \leqslant 3|M| \leqslant 6|R|$, proving (4).

For (5) suppose for contradiction that there exists an edge uv in G^* that is neither an edge in T^* nor has an endpoint in M^*. Let P be the path in T^* from u to v. Since uv is not an edge of T^* the path P has at least one internal vertex. Let u' be the vertex immediately after u on P. Since $u \notin M^*$ and $u \in N_{T^*}(u')$ it follows that $u' \notin M$. Thus $u' \notin R$ and u' has degree precisely 2 in T^*. But then $(T^* - u') \cup \{uv\}$ is connected (since we can go between u and the successor of u' on P via uv and then P) and contains R, contradicting the minimality of G^*. ■

We are now ready to conclude the proof of Lemma 4.3.10

Proof: [Proof of Lemma 4.3.10] We set M' to be the set of all vertices in T^* at distance (in T^*) at most t from M^*. By Lemma 4.3.14 (point 3) we have that $|M'| \leqslant |M^*| + 2t|M| \leqslant 6r + 4tr$. Since $|\mathcal{F}_1| > |\mathcal{F}/2| \geqslant r(4t+6)c$ there exists a $Y_i \in \mathcal{F}_1$ disjoint from M'.

Let z be the number of connected components of $T^* - M$ that have non-empty intersection with Y_i, and Z be the union of the vertex sets of all such components. Since each component of $T^* - M$ is a path of vertices of degree 2 in T^*, z applications of Observation 4.3.13 yield that $T^* - Z$ has precisely $z+1$ connected components. Since $Y_i \in \mathcal{F}_1$ we have that no connected component of $G^* - Y_i$ contains two vertices of R. Thus no connected component of $T^* - Y_i$ contains two vertices of R, and Y_i is disjoint from $M' \supseteq M$, so $Y_i \cap V(T^*) \subseteq Z$ and no connected component of $T^* - Z$ contains two vertices of R either. But then $T^* - Z$ has at least r connected components, implying $z + 1 \geqslant r$.

Since $Y_i \subseteq N_G[X_i]$ and $|X_i| \leqslant s$, and $z \geqslant r - 1 = 4s - 1 > 2|X_i|$, there exists an $x \in X_i$ such that $N_G[x]$ has non-empty intersection with three distinct components C_1, C_2 and C_3 of $T^* - M$. For each $j \in \{1, 2, 3\}$ define P_j to be a shortest path in $T^*[C_j]$ from $N_G[x]$ to $N_{T^*}[M^*]$. Since Y_i, and therefore $N_G[x]$, is disjoint from M' it follows that $|V(P_j)| \geqslant t$. Since P_j is *shortest* it follows that $x \notin V(P_j)$, that x is a neighbor (in G) of precisely one endpoint of P_j (and no other vertices of P_j), and that $V(P_j) \cap M^* = \emptyset$. Thus, Lemma 4.3.14 (point 5) yields that each P_j induces a path in G^* (and therefore in G) and that there are no edges in G between P_j and $P_{j'}$ for $j \neq j'$. But then $x \cup V(P_1) \cup V(P_2) \cup V(P_3)$ induces an S_{t_1,t_2,t_3} in G with $t_1, t_2, t_3 \geqslant t$. ■

4.3.3 Presentation of the Algorithm

We give one last definition before presenting the algorithm.

Definition 4.3.15 (level sets)**.** Let G be graph, G' and induced subgraph of G, and $\mathcal{F}$ a list of vertex sets of G. For all natural numbers j, the j^{th} *level set with respect to* G, G', *and* $\mathcal{F}$, denoted by $\mathcal{L}_j(G, G', \mathcal{F}, N)$, is defined as the set of vertices of G' that belong to at least j sets (counting multiplicity) of $N_G^{G'}[\mathcal{F}]$.

In the following recursive algorithm, the input will always consist of a natural number N, two level sets $\mathcal{F}_1$ and $\mathcal{F}_2$ and a graph G. Additionally, there will be a global variable $\mathcal{G}$ which is set to the very first graph the algorithm is called on, so G will always be an induced subgraph of $\mathcal{G}$. We will say that a vertex, v, in G is *N-branchable with respect to* $\mathcal{G}$, G, $\mathcal{F}_1$, *and* $\mathcal{F}_2$ (or more simply *branchable* when the values of N, $\mathcal{G}$, G, $\mathcal{F}_1$, and $\mathcal{F}_2$ are clear from the context) if there is an natural number j such that either $|N_G[v] \cap \mathcal{L}_j(\mathcal{G}, G, \mathcal{F}_1, N)| \geqslant \frac{N}{2^j}$ or $|N_G[v] \cap \mathcal{L}_j(\mathcal{G}, G, \mathcal{F}_2, N)| \geqslant \frac{N}{2^j}$.

We now present a quasi-polynomial time algorithm for independent set on $S_{t,t,t}$-free graphs which we will refer to as IND. We first give the high level ideas of how IND works, followed by a formal prose-style description of the algorithm, then we give the algorithm in pseudocode.

Overview. At the highest level IND is a recursive algorithm that does three basic operations. When there is a branchable vertex, v, IND will be recursively called on $G - v$ and $G - N_G[v]$, when there is an extended strip decomposition such that no particle contains to much weight, IND will be recursively called on each particle, and when there is a balanced separator that is dominated by few vertices, it adds the balanced separator to a list (either $\mathcal{F}_1$ or $\mathcal{F}_2$). The lists ($\mathcal{F}_1$ and $\mathcal{F}_2$) are what will guide the branching process, and the goal of branching is to (efficiently) reach an instance where the input graph has

a desirable extended strip decomposition. Both the "extended strip decomposition" and "add a balanced balanced separator" operations will come in two distinct flavors, type 1 and type 2.

Let us now be a little more detailed about how IND works. The algorithm is a recursive algorithm that takes as input an $S_{t,t,t}$-free graph G, a vertex set X (X may also be set to $\perp$, indicating that a new set for X must be found), an integer N, and two lists $\mathcal{F}_1$ and $\mathcal{F}_2$. If we wish to know the weight of a maximum weight independent set of the graph G, then IND is intended to be initially called on the inputs $G, X = \perp, N = |G|, \mathcal{F}_1 = \emptyset, \mathcal{F}_2 = \emptyset$. The algorithm sets a global variable $\mathcal{G}$ which is set to the first graph that the algorithm is called on, so that in all recursive calls, $\mathcal{G}$ refers to the initial graph the algorithm is called on. In any given call of IND, the vertex set X along with the vertex sets contained in $\mathcal{F}_1$ and $\mathcal{F}_2$ may not be subsets of $V(G)$, but they will always be subsets of $V(\mathcal{G})$. The integer N will be approximately equal to $|G|$ (and will always satisfy $|G| \leqslant N$) and is used for the sake of making the run time analysis easier.

The set X is obtained using $\mathsf{esd}(G)$ (see Definition 4.3.4) and will thus have the property that no particle of the corresponding extended strip decomposition of $G - N_G[X]$ will have more than $|G|/2 \leqslant N/2$ vertices and $|X| \leqslant c_t \log(N)$. One goal of the branching operation, "type 2 extended strip decomposition", and "type 2 balanced separator" operation are to efficiently reduce $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$ to the empty set. Observation 4.3.6 tells us that when $\mathsf{relevant}_{\mathcal{G}}(G, X, N) = \emptyset$ that either we will get an extended strip decomposition that is N-good (in which case we make a lot of progress as each particle now has much less than $N \approx |G|$ vertices, this is the "type 1 extended strip decomposition" operation) or we get that $N^G_{\mathcal{G}}[X]$ is a $c_t \log(N)$-boosted balanced separator (with X as a core). In either case we find a new X using Theorem 4.1.6 and repeat the process.

But how do we make progress in the case where $N^G_{\mathcal{G}}[X]$ is a $c_t \log(N)$-boosted balanced separator (and we do not have an N-good extended strip decomposition)? When $N^G_{\mathcal{G}}[X]$

is a $c_t \log(N)$-boosted balanced separator, we place X into $\mathcal{F}_1$ (this is the "type 1 balanced separator operation"). An analysis similar to that found in [1] (sketched in Section 4.1.1) shows that, because the size of X is at most $c_t \log(N)$), we can collect these cores of $c_t \log(N)$-boosted balanced separators into the list $\mathcal{F}_1$ and efficiently branch in a manner such that no vertex of $N_{\mathcal{G}}^{G}[\mathcal{F}_1]$ belongs to over $\log(N)$ of these sets, and therefore by Lemma 4.3.10, in an $S_{t,t,t}$-free graph, $\mathcal{F}_1$ cannot contain more than $80tc_t \log^2(N)$ sets. It follows that when we repeat the process from the previous paragraph, we can only get back that $N_{\mathcal{G}}^{G}[X]$ is $c_t \log(N)$-boosted balanced separator only a few times (at most $80tc_t \log^2(N)$ times) before get an extended strip decomposition that is N-good (or no component of G has many vertices, but by Observation 4.3.5 this implies the existence of an N-good extended strip decomposition), and we make good progress.

Next, let us briefly look at how the algorithm is able to efficiently reduce $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$ to the empty set using the branching, "type 2 extended strip decomposition", and "type 2 balanced separator" operations. The basic idea is based on a combination of Lemmas 4.3.8 and 4.3.9 and the techniques used in [1] (sketched in Section 4.1.1). IND applies Lemma 4.3.8 to G and $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$ (with $i = \log(200c_t^3 \log^3(N))$). If an extended strip decomposition, (H, η), is returned then since each particle, P, has much less than $|\mathsf{relevant}_{\mathcal{G}}(G, X, N)|$ vertices of $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$ (at most $(1 - \frac{1}{800c_t^3 \log^3(N)})|\mathsf{relevant}_{\mathcal{G}}(G, X, N)|$) and because $\mathsf{relevant}_{\mathcal{G}}(P, X, N) \subseteq \mathsf{relevant}_{\mathcal{G}}(G, X, N)$, it follows that $|\mathsf{relevant}_{\mathcal{G}}(P, X, N)| << |\mathsf{relevant}_{\mathcal{G}}(G, X, N)|$ and good progress is made in reducing the size of $\mathsf{relevant}_{\mathcal{G}}(P, X, N)$ (this is the type 2 extended strip decomposition operation). Otherwise the lemma returns a set C such that $N_G[C]$ is an $\frac{|\mathsf{relevant}_{\mathcal{G}}(G,X,N)|}{200c_t^3 \log^3(N)}$-balanced separator of $(G, |\mathsf{relevant}_{\mathcal{G}}(G, X, N)|)$ and $|C| \leqslant 28000c_t^4 \log^4(N)$.

So how do we make progress in efficiently reducing $|\mathsf{relevant}_{\mathcal{G}}(G, X, N)|$ when what we get back is a set C such that $N_G[C]$ is an $\frac{|\mathsf{relevant}_{\mathcal{G}}(G,X,N)|}{200c_t^3 \log^3(N)}$-balanced separator of G for $|\mathsf{relevant}_{\mathcal{G}}(G, X, N)|$? An analysis similar to that found in [1] shows that, because

$|C| \leqslant 28000c_t^4 \log^4(N)$, we can collect these sets C that we find into the list $\mathcal{F}_2$ (this is the "type 2 balanced separator operation") and efficiently branch in a manner such that no vertex of $N_{\mathcal{G}}^G[\mathcal{F}_2]$ belongs to over $\log(N)$ of these sets. Since $\mathcal{G}$ is $S_{t,t,t}$-free (and $|X| \leqslant c_t \log(N)$), it follows from Lemma 4.3.9 that $|\mathcal{F}_2|$ cannot grow larger than $10t \log(N)$. Hence, after applying Lemma 4.3.8 a few times (at most $10t \log(N)$ times), it must return an extended strip decomposition and good progress is made in decreasing $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$.

Formal description. We now give a formal description of our independent set algorithm for $S_{t,t,t}$-free graphs, which we will refer to as IND. The algorithm is a recursive algorithm that takes as input a graph G, a vertex set X (X may also be set to $\perp$), an integer N, and two lists $\mathcal{F}_1$ and $\mathcal{F}_2$. IND is intended to be initially called on the inputs $G, X = \perp, N = |G|, \mathcal{F}_1 = \emptyset, \mathcal{F}_2 = \emptyset$. The algorithm sets a global variable $\mathcal{G}$ which is set to the graph in the first set of input parameters, $G, X, N, \mathcal{F}_1, \mathcal{F}_2$, that the algorithm is called on so that on all recursive calls $\mathcal{G}$ refers to the initial graph the algorithm is called on. The vertex set X along with the vertex sets contained in $\mathcal{F}_1$ and $\mathcal{F}_2$ may not be subsets of $V(G)$, although they will always be subsets of $V(\mathcal{G})$. N will be approximately, but always greater than or equal to, the size of G. The function of N is to help in the runtime analysis.

When the algorithm makes a recursive call, some of the elements among the input parameters, $G, X, N, \mathcal{F}_1, \mathcal{F}_2$, will be the same in the recursive call as they are in the current instance, while the remaining parameters will be changed. In the following description of the algorithm, when describing the input to the recursive calls, we will only explicitly mention the parameters that are changed from the current call, unmentioned parameters are assumed to remain the same as in the current call. For instance if the graph that the recursive call is made on is different from the graph of the current call but all other

elements, $X, N, \mathcal{F}_1, \mathcal{F}_2$ remain the same as in the current call of the algorithm, we will only indicate what the new graph is the call is made on and not mention the unchanged elements, $X, N, \mathcal{F}_1, \mathcal{F}_2$.

In order to help the runtime analysis of the algorithm, we will label each call of IND that is made based on the first case it satisfies (which in turn determines the recursive calls it will make).

Base Case: For the Base Case (Label: base case call), if $|V(G)| \leqslant 1$ then IND returns $\mathfrak{w}(V(G))$.

Case 1: For Case 1 (Label: branch call), if there exists a branchable vertex $v \in G$, then IND is recursively called on two instances, the first instance on $G - v$ and the second on $G - N_G[v]$, and stores the numbers returned by these recursive calls as I_f and I_s respectively. The algorithm then returns the maximum of I_f and $I_s + \mathfrak{w}(v)$.

- If the algorithm has not returned at this point and X is equal to $\perp$, then the algorithm sets $X = \mathsf{esd}(G)$. No recursive call is made here, no label is given here, and the algorithm continues to see which case it satisfies. We say that the set X is *discovered* in this call.

Case 2: For Case 2 (Label: type 1 extended strip decomposition call) if the extended strip decomposition inferred by (X, G), call it (H, η), is an N-good extended strip decomposition, then for each particle P of (H, η) the algorithm recursively calls itself on $G = P$, $X = \perp$, $N = |P|$, $\mathcal{F}_1 = \emptyset, \mathcal{F}_2 = \emptyset$. Then IND returns $\mathsf{matching}(H, \eta)$.

Case 3: For Case 3 (Label: boosted balanced separator call), if $N_{\mathcal{G}}^G[X]$ is an $\frac{N}{32c_t^2 \log^2(N)}$-balanced separator for G then IND is recursively called with X added to $\mathcal{F}_1$, X set to $\perp$, and $\mathcal{F}_2$ set to $\emptyset$. Then the algorithm returns the value obtained from this recursive call. Here we say that X is the boosted balanced separator core added in this call.

- If IND has not returned at this point then note by Observation 4.3.6 and the fact that Case 2 and 3 do not hold implies $\mathsf{relevant}_{\mathcal{G}}(G, X, N) \neq \emptyset$. The algorithm then applies Lemma 4.3.8 (with $i = \log(200c_t^3 \log^3(N))$) to either obtain, in polynomial time, a rigid extended strip decomposition of G, call it (H, η), such no particle of (H, η) has over $(1 - \frac{1}{800c_t^3 \log^3(N)})|\mathsf{relevant}_{\mathcal{G}}(G, X, N)|$ vertices of $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$ or a $C \in G$ such that $N_G[C]$ is a $\frac{|\mathsf{relevant}_{\mathcal{G}}(G,X,N)|}{200c_t^3 \log^3(N)}$-balanced separator for $\mathsf{relevant}(G, X, N)$, and $|C| \leqslant 28000c_t^4 \log^4(|G|) \leqslant 28000c_t^4 \log^4(N)$. No recursive call is made here, no label is given, and the algorithm continues to see which case it satisfies.

Case 4: For Case 4 (Label: type 2 extended strip decomposition call), if Lemma 4.3.8 returned an extended strip decomposition, (H, η), then for each particle, P, of (H, η) IND is recursively called with the graph set to P and $\mathcal{F}_2$ set to the empty set. IND then returns $\mathsf{matching}(H, \eta)$.

Case 5: For Case 5 (Label: balanced separator call), if Lemma 4.3.8 returned a balanced separator, $N_G[C]$, for $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$ then IND is recursively called, adding C to $\mathcal{F}_2$. IND then returns the value obtained from this recursive call. Here we say the set C is the balanced separator core added in this call.

For completeness and ease of reference we give pseudocode for the algorithm IND below. The correctness proofs and running time analysis do not refer to the pseudocode, and so a reader may choose to skip it. Recall that IND sets a global variable $\mathcal{G}$ which is set to the graph in the first set of input parameters, $G, X, N, \mathcal{F}_1, \mathcal{F}_2$. This step is not explicitly mentioned in the pseudocode.

IND

1: **Input:** G, X, N, $\mathcal{F}_1$, $\mathcal{F}_2$

Output: $\mathsf{mwis}(G)$.
if $|V(G)| \leqslant 1$ **then**
  return $\mathfrak{w}(V(G))$
if exists branchable vertex, v, **then**
  return $\max(\mathrm{IND}(G - v, X, N, \mathcal{F}_1, \mathcal{F}_2), \mathrm{IND}(G - N_G[v], X, N, \mathcal{F}_1, \mathcal{F}_2) + \mathfrak{w}(v))$
if $X = \perp$ **then**
  Set $X = \mathsf{esd}(G)$
if the extended strip decomposition, (H, η), inferred by (X, G) is N-good **then**
  for all particles P in (H, η) **do**
    Get $\mathrm{IND}(P, \perp, |P|, \emptyset, \emptyset)$
  return $\mathsf{matching}(H, \eta)$
if $N_{\mathcal{G}}^G[X]$ is an $\frac{N}{32c_t^2 \log^2(N)}$-balanced separator for G **then**
  return $\mathrm{IND}(G, \perp, N, \mathcal{F}_1 \cup X, \emptyset)$
Use Lemma 4.3.8 with $i = \log(200c_t^3 \log^3(N))$ to obtain either a rigid extended strip decomposition, (H, η), such that no particle of (H, η) contains over $(1 - \frac{1}{800c_t^3 \log^3(N)})$ $|\mathsf{relevant}_{\mathcal{G}}(G, X, N)|$ vertices of $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$ or a set $C \subseteq V(G)$ such that $N_G[C]$ is a $\frac{|\mathsf{relevant}_{\mathcal{G}}(G,X,N)|}{200c_t^3 \log^3(N)}$-balanced separator for $\mathsf{relevant}_{\mathcal{G}}(G, X, N)$, and $|C| \leqslant 28000c_t^4 \log^4(|G|)$ $\leqslant 28000c_t^4 \log^4(N)$.
if Lemma 4.3.8 returns (H, η) **then**
  for all particles P in (H, η) **do**
    Get $\mathrm{IND}(P, X, N, \mathcal{F}_1, \emptyset)$
  return $\mathsf{matching}(H, \eta)$
if Lemma 4.3.8 returns C **then**
  return $\mathrm{IND}(G, X, N, \mathcal{F}_1, \mathcal{F}_2 \cup C)$

4.3.4 Correctness and Runtime Analysis

In order to analyze the runtime of the algorithm, we will find it useful to define the recursion tree generated by a run of the algorithm and prove that it has only a quasi-polynomial number of vertices. Because we have yet to prove the algorithm will terminate, the tree in the next definition may be infinite, but we will shortly prove that IND will terminate that the recursion tree for IND is finite.

Definition 4.3.16 (recursion tree)**.** The *recursion tree*, T, *generated* by $\text{IND}(G, \perp, |G|, \emptyset, \emptyset)$ is the directed rooted tree with a node for each call of IND made in the course of running IND on the initial input $(G, \perp, |G|, \emptyset, \emptyset)$, the root node corresponding to the initial call of IND on the input $(G, \perp, |G|, \emptyset, \emptyset)$. There is a directed edge from a node $p \in T$ to a node $c \in T$ when the call that corresponds to p invoked the call that corresponds to c. Furthermore, we label the vertices p and c as well as the edge pc as follows. The vertices p and c get the same label as the calls they correspond to respectively (replacing "call" now with "node"). If the call that corresponds to p is labeled with anything other than branch call, then the pc edge gets same label as the call that corresponds to p (replacing "call" now with "edge"). If p corresponds to a branch call then let v be the vertex that is branched on in that call and let G_p be the graph given in the input of that call. If c corresponds to the call where the graph $G_p - N_{G_p}[v]$ is used as the input then we label pc as a "success edge", and if c corresponds to the call where the graph $G_p - v$ is used as the input we label pc as a "failure edge".

Furthermore, let u be a node of T. Then we let $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$ denote the tuple that was used for the input of the call u corresponds to. We call this tuple the *parameters of* u.

Let G be a graph. We collect a set of observations about IND and the recursion tree generated by $\text{IND}(G, \perp, |G|, \emptyset, \emptyset)$ that follow directly from how IND has been defined.

We state these observations without a proof (as their proofs follow directly from how the algorithm was defined) and we will use them use in future proofs typically without reference to this observation.

Observation 4.3.17. *Let G be a graph and let T be the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$. Let p and c be nodes of T such that pc is an edge of T and let $(G_p, X_p, N_p, \mathcal{F}_{1,p}, \mathcal{F}_{2,p})$ and $(G_c, X_c, N_c, \mathcal{F}_{1,c}, \mathcal{F}_{2,c})$ be the parameters of p and c respectively. Then the following hold:*

1. *$N_p < N_c$ if pc is a type 1 recurse on particles edge and $N_p = N_c$ otherwise. Additionally, $N_p, N_c \geqslant |G|$.*

2. *G_c is an induced subgraph of G_p, G_c is a proper induced subgraph of G_p if pc is a success, failure, or type 2 recurse on particles edge, and $G_p = G_c$ if pc is a balanced separator or boosted balanced separator edge.*

3. *If $X_p = \perp$ and p is not a base case node nor a branch node, then there is a set X discovered in the call that corresponds to p.*

4. *Assume p is not a base case node nor a branch node. If $X_p \neq \perp$ then let $X = X_p$, else let X be the set that is discovered in the call that corresponds to p. Then if $\mathsf{relevant}_G(G_p, X, N_p) = \emptyset$ then (using Observation 4.3.6) p is either a type 1 extended strip decomposition node or a boosted balanced separator node.*

5. *If c is a base case node then c is a leaf of T.*

The next three lemmas show that IND$(G, \perp, |G|, \emptyset, \emptyset)$ terminates and returns the weight of a maximum weight independent set of G.

Lemma 4.3.18. *Let t be a positive integer, G an $S_{t,t,t}$-free graph, T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, $u \in T$ such that u is a balanced separator or boosted*

balanced separator node, and let $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$ *be the parameters of* u*. If* C *is the balanced separator or boosted balanced separator core added in the call that corresponds to* u*, then* $N_G^{G_u}[C] \neq \emptyset$*.*

Proof: Let t, G, T, u, $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$, and C be as in the statement of this lemma, and assume for a contradiction that $N_G^{G_u}[C] = \emptyset$.

First assume that u is a boosted balanced separator node, so either $C = X_u$ or $X_u = \perp$ and C was discovered in the call that corresponds to u. In either case, since u is a boosted balanced separator node, $N_G^{G_u}[C]$ is an $\frac{N_u}{32c_t^2\log^2(N_u)}$-balanced separator for G_u and by assumption $N_G^{G_u}[C] = \emptyset$, it follows that no component of G_u contains over $\frac{N_u}{32c_t^2\log^2(N_u)}$ vertices. It follows by Observation 4.3.5 that the extended strip decomposition inferred by (C, G_u) is N_u-good and hence u should have been a type 1 extended strip decomposition node and not a boosted balanced separator node.

Next, assume that u is a balanced separator node, so the empty set is a $\frac{|\mathsf{relevant}_G(G_u,X,N_u)|}{200c_t^3\log^3(N_u)}$-balanced separator of $(G_u, \mathsf{relevant}_G(G_u, X, N_u))$. If $X_u \neq \perp$ then let $X = X_u$, and if $X_u = \perp$ then let X be the the set discovered in the call that corresponds to u. Since u is not a type 1 extended strip decomposition node nor a boosted balanced separator node it follows from the 4 point of Observation 4.3.17 (or Observation 4.3.6) that $\mathsf{relevant}_G(G_u, X, N_u) \neq \emptyset$. Since $N_G^{G_u}[C] = \emptyset$ it follows that no component of G_u contains over $\frac{|\mathsf{relevant}_G(G_u,X,N_u)|}{200c_t^3\log^3(N_u)}$ vertices of $\mathsf{relevant}_G(G_u, X, N_u)$. Among all components of $G_u - N_G^{G_u}[X]$ that has over $\frac{N_u}{32c_t^2\log^2(N_u)}$ vertices (of which there are at most $32c_t^2\log^2(N_u)$), let B be one such that $|N_{G_u}[B] \cap \mathsf{relevant}_G(G_u, X, N_u)|$ is maximized, so since every vertex of $\mathsf{relevant}_G(G_u, X, N_u)$ has a neighbor in at least on component of $G_u - N_G^{G_u}[X]$ that has over $\frac{N_u}{32c_t^2\log^2(N_u)}$, it follows that $|N_{G_u}[B] \cap \mathsf{relevant}_G(G_u, X, N_u)| > \frac{|\mathsf{relevant}_G(G_u,X,N_u)|}{32c_t^2\log^2(N_u)}$. But then $N_{G_u}[B]$ is a connected set that contains at least $\frac{|\mathsf{relevant}_G(G_u,X,N_u)|}{32c_t^2\log^2(N_u)}$ vertices of $\mathsf{relevant}_G(G_u, X, N_u)$ and therefore the empty set cannot be a $\frac{|\mathsf{relevant}_G(G_u,X,N_u)|}{200c_t^3\log^3(N_u)}$-balanced

separator of $(G_u, \mathsf{relevant}_G(G_u, X, N_u))$. ■

Lemma 4.3.19. *Let t be a positive integer and, G an $S_{t,t,t}$-free n-vertex graph. Then IND$(G, \perp, |G|, \emptyset, \emptyset)$ terminated and the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$ is finite.*

Proof: Let t, G, and n be as in the statement of this lemma, T the recursion tree of IND$(G, \perp, |G|, \emptyset, \emptyset)$, and P some path in T. In order to show that IND$(G, \perp, |G|, \emptyset, \emptyset)$ terminates and that T is finite, it is sufficient to show that there is a bounded number of each of the 6 types of edges that can appear in P, type 1 and type 2 extended strip decomposition edges, balanced separator and boosted balanced separator edges, and success/failure edges. Let pc be an edge of T, and let $(G_p, X_p, N_p, \mathcal{F}_{1,p}, \mathcal{F}_{2,p})$ and $(G_c, X_c, N_c, \mathcal{F}_{1,c}, \mathcal{F}_{2,c})$ be the parameters of p and c respectively. If pc is a success edge, failure, edge, or type 2 extended strip decomposition edge, then G_c is a proper subgraph of G_p, hence there can be at most n of each type of these edges. If pc is a type 1 extended strip decomposition edge, then $N_c < N_p$, so again there can be at most n type 1 extended strip decomposition edges.

Now let u and w be two nodes of P with parameters $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$ and $(G_w, X_w, N_w, \mathcal{F}_{1,w}, \mathcal{F}_{2,w})$ respectively and let P' be the subpath of P that starts at u and ends at w. Assume that P' does not contain any type 1 or type 2 extended strip decomposition edges nor success/failure edges (so all edges are balanced separator and boosted balanced separator edges). It follows that $G_u = G_w$, $N_u = N_w = N$, and if P' has $N\log(N)$ boosted balanced separator edges, then $\mathcal{F}_{1,w}$ must have at least $N\log(N)$ sets $S \in \mathcal{F}_{1,w}$ (counting multiplicity) such that $N_G^{G_w}[S] \neq \emptyset$ (using Lemma 4.3.18 to ensure they are non-empty). Hence $\mathcal{L}_{\log(N)}(G, G_w, \mathcal{F}_{1,w}, N)$ is none empty and since any vertex $v \in \mathcal{L}_{\log(N)}(G, G_w, \mathcal{F}_{1,w}, N)$ is by definition a branchable vertex, w must be a branch node (or a base case node). It follows that P' can have at most $N\log(N)$ boosted

balanced separator edges. Since P has a bounded number of type 1 or type 2 extended strip decomposition edges and success/failure edges, there must be a bounded number of boosted balanced separator edges as well.

Now additionally assume that P' contains no boosted balanced separator edges as well, so all edges of P' are balanced separator edges, so if P' has $N\log(N)$ balanced separator edges, then $\mathcal{F}_{2,w}$ must have at least $N\log(N)$ sets $S \in \mathcal{F}_{2,w}$ (counting multiplicity) such that $N_G^{G_w}[S] \neq \emptyset$ (using Lemma 4.3.18 to ensure theyu are non-empty). Hence $\mathcal{L}_{\log(N)}(G, G_w, \mathcal{F}_{2,w}, N)$ is non-empty and since any vertex $v \in \mathcal{L}_{\log(N)}(G, G_w, \mathcal{F}_{2,w}, N)$ is by definition a branchable vertex, w must be a branch node (or a base case node). It follows that P' can have at most $N\log(N)$ balanced separator edges. Since P has a bounded number of all other edge types, there must be a bounded number of balanced separator edges as well, completing the proof. ■

Now that we have established the recursion tree is finite, we can prove that IND returns the correct answer.

Lemma 4.3.20. *Let G be a graph. Then IND$(G, \perp, |G|, \emptyset, \emptyset)$ returns the weight of a maximum weight independent set of G.*

Proof: Let G be a graph, T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, and u a node of T and assume that for all children, v, of u, that the call corresponding to v correctly returns the weight of a maximum weight independent set of G_v, where G_v is the graph used as input for the call corresponding to v. We show that the call corresponding to u then correctly returns the maximum weight independent set of G_u where G_u is the graph used as input for the call corresponding to u.

If u is balanced separator or boosted balanced separator node then $G_u = G_v$ and so u returns the weight of a maximum weight independent set of $G_v = G_u$. If u is a branch node which branches on the vertex $v \in G_u$, then if we set I_f and I_s to be the

weight of a maximum weight independent set of $G - v$ and $G - N_{G_u}[v]$ respectively, then u returns the maximum of I_f and $I_s + \mathfrak{w}(v)$ which is the weight of a maximum weight independent set of G_u. Lastly, if u is a type 1 or type 2 extended strip decomposition node, then u returns $\mathsf{matching}(H, \eta)$, which by Lemma 4.3.7 is the weight of a maximum weight independent set of G_u.

It now follows from induction that $\text{IND}(G, \perp, |G|, \emptyset, \emptyset)$ returns the weight of a maximum weight independent set of G. ■

Next, we observe that the degree of recursion trees is at most polynomial.

Observation 4.3.21. *There exists a constant c such that for any positive integer t and $S_{t,t,t}$-free n-vertex graph G, the recursion tree, T, of IND$(G, \perp, |G|, \emptyset, \emptyset)$ has a maximum degree of most n^c.*

Proof: Let t, G, and T be as in the statement of the lemma, and let $u \in T$ with parameters $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$. If u is any type of node other than a type 1 or type 2 extended strip decomposition node, then it is clear the degree of u is at most 2. If u is a type 1 extended strip decomposition node then (by the discussion immediately after Definition 4.3.4) the extended strip decomposition inferred by (X_u, G_u) has $n^{\mathcal{O}(1)}$ particles, hence u has degree $n^{\mathcal{O}(1)}$. If u is a type 2 extended strip decomposition, then each child of u corresponds to a particle of a rigid extended strip decomposition of G_u, which has $\mathcal{O}(n)$ vertices and therefore $n^{\mathcal{O}(1)}$ particles, and therefore u has degree $n^{\mathcal{O}(1)}$. ■

Let G be an n-vertex graph and T the recursion tree generated by $\text{IND}(G, \perp, |G|, \emptyset, \emptyset)$. Recall that the edges of T are labeled (see Definition 4.3.16). The next set of lemmas show that on any root to leaf path P of T, there are at most polylog(n) edges with any given label other than a failure label, of which there can be at most n. We will then be able to use this fact to bound the runtime of IND.

Lemma 4.3.22. *Let t be a positive integer, G an $S_{t,t,t}$-free n-vertex graph, T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, and P a path of T. Then P contains at most $64c_t^2 \log^3(n)$ type 1 extended strip decomposition edges.*

Proof: Let t, G, n, T, and P be as in the statement of this lemma. Let pc be a type 1 extended strip decomposition edge of P and let $(G_p, X_p, N_p, \mathcal{F}_{1,p}, \mathcal{F}_{2,p})$ and $(G_c, X_c, N_c, \mathcal{F}_{1,c}, \mathcal{F}_{2,c})$ be the parameters of p and c respectively. Then $N_c \leqslant (1 - \frac{1}{32c_t^2 \log^2(N_p)})N_p \leqslant (1 - \frac{1}{32c_t^2 \log^2(n)})N_p$.

Now, let u and w be two vertices of P with parameters $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$ and $(G_w, X_w, N_w, \mathcal{F}_{1,w}, \mathcal{F}_{2,w})$ respectively. Since for all $c \geqslant 2$, $(1-1/c)^{2c} \leqslant 1/2$ it follows that if there are $64c_t^2 \log^3(n)$ type 1 extended strip decomposition edges on the subpath of P that starts and u and ends at w then $N_w \leqslant (1-\frac{1}{32c_t^2 \log^2(n)})^{64c_t^2 \log^3(n)} N_u \leqslant (1-1/2)^{\log(n)} N_u \leqslant 1$. It follows that $|G_w| \leqslant 1$ and therefore w must be a base node and hence the last vertex of P. So, P cannot have over $64c_t^2 \log^3(N)$ type 1 extended strip decomposition edges. ■

Lemma 4.3.23. *Let t be a positive integer, G an $S_{t,t,t}$-free graph, T the recursion generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, and u a node of T with parameters $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$. Then no vertex of G_u belongs to over $\log(N_u)$ sets of $N_G^{G_u}[\mathcal{F}_{1,u}]$ and no vertex of G_u belongs to over $\log(N_u)$ sets of $N_G^{G_u}[\mathcal{F}_{2,u}]$, hence $\mathcal{L}_i(G, G_p, \mathcal{F}_{1,u}, N_u) = \mathcal{L}_i(G, G_p, \mathcal{F}_{2,u}, N_u) = \emptyset$ for $i > \log(N_u)$.*

Proof: Let t, G, T, u, and $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$ be as in the statement of this lemma. Assume for a contradiction that the statement of this lemma does not hold for u and that u is the first node on a path from the root to u such that the statement of this lemma does not hold. It follows there is some vertex, call it v, of G_u belongs to over $\log(N_u)$ sets of $N_G^{G_u}[\mathcal{F}_{1,u}]$ or $\log(N_u)$ sets of $N_G^{G_u}[\mathcal{F}_{2,u}]$. Let w be the parent of u in T and let $(G_w, X_w, N_w, \mathcal{F}_{1,w}, \mathcal{F}_{2,w})$ be the parameters of w. If the edge wu was a type 1 recurse

on particles edge, then $\mathcal{F}_{1,u} = \mathcal{F}_{2,u} = \emptyset$ but then v could not belong to over $\log(N_u)$ sets of $N_G[\mathcal{F}_{1,u}]$, so wu is not a type 1 recurse on particles edge, hence $N_w = N_u$.

As the arguments are nearly identical, with out loss of generality assume $v \in G_u$ belongs to over $\log(N_u)$ sets of $N_G[\mathcal{F}_{1,u}]$. Since (by how u was chosen) v does not belong to over $\log(N_w) = \log(N_u)$ sets of $N_G[\mathcal{F}_{1,w}]$ it follows that the edge wu must be a boosted balanced separator edge (hence $G_u = G_w$) and v must belong to $\log(N_u) = \log(N_w)$ sets of $N_G^{G_w}[\mathcal{F}_{1,w}]$ in order for v to belong to over $\log(N_u)$ sets of $N_G^{G_u}[\mathcal{F}_{1,u}]$. But then by definition of being a branchable vertex, since v belongs to $\log(N_w)$ sets of $N_G^{G_w}[\mathcal{F}_{1,w}]$ the call that corresponds to w should have branched on v (or some other branchable vertex) and the edge wu would therefore be a either a success or failure edge, which is a contradiction. Therefore, there can never be a "first node" in T such that the statement of this lemma does not hold. ■

Lemma 4.3.24. *Let t be a positive integer, G an $S_{t,t,t}$-free n-vertex graph, T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, u a node of T with parameters $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$, and C the largest component of G_u. Then either $|C| < N_u/2$ or all sets of $\mathcal{F}_{1,u}$ are cores of $c_t \log(N_u)$-boosted balanced separators of G_u originating in G.*

Proof: Let t, G, T, u, $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{1,u})$, and C be as in the statement of this lemma. Let S be in $\mathcal{F}_{1,u}$ and assume that $|C| \geqslant N_u/2$, we show that S is a core of a $c_t \log(N_u)$-boosted balanced separator for G_u originating in G.

Let w be the closest ancestor of u in T that corresponds to a call where S is the core of a boosted balanced separator added in that call. Let $(G_w, X_w, N_w, \mathcal{F}_{1,w}, \mathcal{F}_{2,w})$ be the parameters of w. Note that this implies $X_w = S$ or $X_w = \perp$ and S was discovered in the call that corresponds to w. Additionally, let a be the closest ancestor of w that corresponds to a call where S was discovered. Let $(G_a, X_a, N_a, \mathcal{F}_{1,a}, \mathcal{F}_{2,a})$ be the parameters of a, as S was discovered in the call the corresponds to a, it follows that $X_a = \perp$.

No edge on the path of T starting from w and ending at u can be a type 1 extended strip decomposition edge or else S would not be in $\mathcal{F}_{1,u}$ (since type 1 extended strip decompositions "resets $\mathcal{F}_1$" to $\emptyset$), and no edge on the path of T starting from a and ending at w can be a type 1 extended strip decomposition edge (since type 1 extended strip decompositions "resets X" to $\perp$). Hence $N_a = N_w = N_u$, so set $N = N_a = N_w = N_u$. So, $|S| \leqslant c_t \log(N)$ and since $N_G^{G_w}[S]$ is an $\frac{N}{32c_t^2 \log^2(N)}$-balanced separator for G_w and G_u is an induced subgraph of G_w, it follows that $N_G^{G_u}[S]$ is an $\frac{N}{32c_t^2 \log^2(N)}$-balanced separator for G_u. Since by assumption $|C| \geqslant N/2$, we have that $N_G^{G_u}[S]$ is an $\frac{|C|}{16c_t^2 \log^2(N)}$-balanced separator for G_u and therefore S is a core of a $c_t \log(N)$-boosted balanced separator for G_u originating in G. ■

Lemma 4.3.25. *Let t be a positive integer, G an n-vertex $S_{t,t,t}$-free graph, T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, and P a path of T. Then P contains at most $5200c_t^4 \log^5(n)$ boosted balanced separator edges.*

Proof: Let t, G, T, and P be as in the statement if this lemma. Let P' be a subpath of P that does not contain any edges that are type 1 extended strip decomposition edges. We first show that P' has less than $80tc_t \log^2(n)$ boosted balanced separator edges.

Assume for a contradiction that P' does have $80tc_t \log^2(n)$ boosted balanced separator edges, let uw be the $80tc_t \log^2(n))^{th}$ boosted balanced separator edge, and let $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$ and $(G_w, X_w, N_w, \mathcal{F}_{1,w}, \mathcal{F}_{2,w})$ be the parameters of u and w respectively. Since P' has no type 1 extended strip decomposition edges and $80tc_t \log^2(n)$ boosted balanced separator edges it follows that $|\mathcal{F}_{1,w}| \geqslant 80tc_t \log^2(n) \geqslant 80tc_t \log^2(N_w)$ and $N_u = N_w$.

Since uw is a boosted balanced separator edge and not a type 1 extended strip decomposition edge we can conclude that if C is the largest component of $G_u = G_w$, then $|C| \geqslant N_u/2 = N_w/2$ (or else by Observation 4.3.5 the extended strip decomposition

inferred by (X_u, G_u) would be N_u-good and uw would be a type 1 extended strip decomposition edge). It follows then from Lemma 4.3.24 that all $80tc_t \log^2(n) \geqslant 80tc_t \log^2(N_w)$ sets of $\mathcal{F}_{1,w}$ are cores of $c_t \log(N_w)$-boosted balanced separators for G_w originating in G. By Lemma 4.3.23 no vertex of G_w belongs to over $\log(N_w)$ vertex sets of $N_G^{G_w}[\mathcal{F}_{1,w}]$, it then follows from Lemma 4.3.10 that G contains an $S_{t,t,t}$, a contradiction.

Hence, P' has less than $80tc_t \log^2(n)$ boosted balanced separator edges. Since by Lemma 4.3.22 P has at most $64c_t^2 \log^3(n)$ type 1 extended strip decomposition edges, it follows that P contains at most $(80tc_t \log^2(n)) \cdot (64c_t^2 \log^3(n) + 1) \leqslant 5200tc_t^3 \log^5(n) \leqslant 5200c_t^4 \log^5(n)$ (recall by definition that $c_t \geqslant t$) boosted balanced separator edges. ■

Lemma 4.3.26. *Let t be a positive integer, G an n-vertex $S_{t,t,t}$-free graph, T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, and P a path of T. Then P contains at most $10^7 c_t^7 \log^9(n)$ type 2 extended strip decomposition edges.*

Proof: Let t, G, T, and P be as in the statement of this lemma. Let P' be a subpath of P that contains no type 1 extended strip decomposition nor boosted balanced separator edges. It then follows that there exists an integer N and a set $X \subseteq V(G)$ such that for any node $a \in P'$ with parameters $(G_a, X_a, N_a, \mathcal{F}_{1,a}, \mathcal{F}_{2,a})$ it holds that $N_a = N$ and $X_a = \perp$ or X.

We first show that P' has at most $1600c_t^3 \log^4(n) + 2$ type 2 extended strip decomposition edges. Let pc be a type 2 extended strip decomposition edge of P' and let $(G_p, X_p, N, \mathcal{F}_{1,p}, \mathcal{F}_{2,p})$ and $(G_c, X_c, N, \mathcal{F}_{1,c}, \mathcal{F}_{2,c})$ be the parameters of p and c respectively. Either $X_p = X_c = X$ or $X_p = \perp$, $X_c = X$ and X was discovered in the call p corresponds to. In either case we have $|\mathsf{relevant}_G(G_c, X, N)| \leqslant (1 - \frac{1}{800c_t^3 \log^3(N)})|\mathsf{relevant}_G(G_p, X, N)| \leqslant (1 - \frac{1}{800c_t^3 \log^3(n)})|\mathsf{relevant}_G(G_p, X, N)|$.

Now, let u and w be two vertices of P' with parameters $(G_u, X_u, N, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$ and $(G_w, X_w, N, \mathcal{F}_{1,w}, \mathcal{F}_{2,w})$ respectively. Since $(1 - 1/c)^{2c} \leqslant 1/2$ for $c \geqslant 2$, it follows that if

there are $1600c_t^3\log^4(n)+2$ type 2 extended strip decomposition edges on the subpath of P' that starts and u and ends at w then

$$\begin{aligned}|\mathsf{relevant}_G(G_w,X,N)| &\leqslant (1-\frac{1}{800c_t^3\log^3(n)})^{1600c_t^3\log^4(n)+2}|\mathsf{relevant}_G(G_u,X,N)|\\ &\leqslant (1-1/2)^{\log(n)+1}|\mathsf{relevant}_G(G_u,X,N)|<1.\end{aligned}$$

It follows that $|\mathsf{relevant}_G(G_w,X,N)|=0$ and therefore, for any node z in P' (with parameters say $(G_z,X,N,\mathcal{F}_{1,z},\mathcal{F}_{2,z})$) that comes after w it holds that $\mathsf{relevant}_G(G_z,X,N)=\emptyset$. It then follows from the 4^{th} point of Observation 4.3.17 that there cannot be any other type 2 extended strip decomposition edges after w in P'.

Since P' has at most $1600c_t^3\log^4(n)+2$ type 2 extended strip decomposition edges and by Lemmas 4.3.22 and 4.3.25, P has at most $64c_t^2\log^3(n)$ type 1 extended strip decomposition edges and at most $5200c_t^4\log(n)^5$ boosted balanced separator edges, it follows that P has at most $(1600c_t^3\log^4(n)+2)\ (64c_t^2\log^3(n)+5200c_t^4\log^5(n)+1)\leqslant 10^7c_t^7\log^9(n)$ type 2 extended strip decomposition edges. ■

Lemma 4.3.27. *Let t be a positive integer, G an n-vertex $S_{t,t,t}$-free graph, T the recursion tree generated by IND$(G,\perp,|G|,\emptyset,\emptyset)$, and P a path of T such that no edges of P are type 1 extended strip decomposition, boosted balanced separator, nor type 2 extended strip decomposition edges. Then P contains at most $10^9c_t^8\log^{11}(n)$ balanced separator edges.*

Proof: Let t, G, T, and P be as in the statement of this lemma. Let P' be a subpath of P that contains no type 1 extended strip decomposition, boosted balanced separator, nor type 2 extended strip decomposition edges. It then follows that there exists an integer N and a set $X\subseteq V(G)$ such that for any node $a\in P'$ with parameters $(G_a,X_a,N_a,\mathcal{F}_{1,a},\mathcal{F}_{2,a})$ it holds that $N_a=N$ and $X_a=\perp$ or X.

Assume for a contradiction that P' has $10t\log^2(n)$ balanced separator edge. Let

u_iw_i denote the $(i10t\log(n))^{th}$ balanced separator edge, u_0 be the first vertex of P', $(G_{u_i}, X_{u_i}, N, \mathcal{F}_{1,u_i}, \mathcal{F}_{2,u_i})$ be the parameters of u_i, $i \geqslant 0$, of P', and P^j denote the subpath of P' that starts at u_j and ends at u_{j+1}, $j \geqslant 0$.

For some P^j it must hold that $|\mathsf{relevant}_G(G_j, X, N)|/2 \leqslant |\mathsf{relevant}_G(G_{j+1}, X, N)|$ or else $\frac{|\mathsf{relevant}_G(G_0,X,N)|}{2^{\log(n)}} > |\mathsf{relevant}_G(G_{\log(n)}, X, N)|$ which implies that $|\mathsf{relevant}_G(G_{\log(n)}, X, N)| = 0$. Therefore, by the 4^{th} point of Observation 4.3.17 it follows that $u_{\log(n)}w_{\log(n)}$ cannot be a balanced separator edge. So, we conclude that for some P^j it must hold that $|\mathsf{relevant}_G(G_j, X, N)|/2 \leqslant |\mathsf{relevant}_G(G_{j+1}, X, N)|$, so fix this index as j.

For each balanced separator node, v, on the path P^j, let B_v denote the balanced separator core added in call v, $\mathcal{F}$ the list of these B_v's, and $(G_v, X_v, N, \mathcal{F}_{1,v}, \mathcal{F}_{2,v})$ the parameters of v. Note that $X = X_v$ (accept for possibly the first such v if $X_v = \perp$ and then X would be X discovered in the call the corresponds to v) and $|\mathsf{relevant}_G(G_v, X, N)|/2 \leqslant |\mathsf{relevant}_G(G_{u_j}, X, N)|/2 \leqslant |\mathsf{relevant}_G(G_{u_{j+1}}, X, N)|$ and $\mathsf{relevant}_G(G_{u_{j+1}}, X, N) \subseteq \mathsf{relevant}_G(G_v, X, N) \subseteq \mathsf{relevant}_G(G_{u_j}, X, N)$. So, since $N_G^{G_v}[B_v]$ is an $\frac{|\mathsf{relevant}_G(G_v,X,N)|}{200c_t^3\log^3(N)}$-balanced separator for $\mathsf{relevant}_G(G_v, X, N)$, it follows that $N_G^{G_{u_{j+1}}}[B_v]$ is an $\frac{|\mathsf{relevant}_G(G_{u_{j+1}},X,N)|}{100c_t^3\log^3(N)}$-balanced separator for $\mathsf{relevant}_G(G_{u_{j+1}}, X, N)$. By definition of $\mathcal{F}$, it follows that $|\mathcal{F}| \geqslant 10t\log(n) \geqslant 10t\log(N)$, by Lemma 4.3.23 no vertex of $G_{u_{j+1}}$ belongs to over $\log(N)$ set of $N_G^{G_{u_{j+1}}}[\mathcal{F}_{2,u_{j+1}}]$ and therefore of $N_G^{G_{u_{j+1}}}[\mathcal{F}]$ (as $\mathcal{F} \subseteq \mathcal{F}_{2,u_{j+1}}$), so by Lemma 4.3.9 G contains and $S_{t,t,t}$, a contradiction.

We conclude that P' has less that $10t\log^2(n) \leqslant 10c_t\log^2(n)$ (recall by definition $c_t \geqslant t$) balanced separators. Since by Lemmas 4.3.22, 4.3.25, and 4.3.26 P has at most $64c_t^2\log^3(n)$, $5200c_t^4\log^5(n)$, and $10^7c_t^7\log^9(n)$ type 1 extend strip decomposition edges, boosted balanced separator edges, and type 2 extended strip decomposition edges respectively, it follows that P has at most $10c_t\log^2(n)$ $(64c_t^2\log^3(n) + 5200c_t^4\log^5(n) + 10^7c_t^7\log^9(n) + 1) \leqslant 10^9c_t^8\log^{11}(n)$ balanced separator edges. ■

Let G be a graph, T the recursion tree generated by $\text{IND}(G, \perp, |G|, \emptyset, \emptyset)$, and $p, c \in T$

such that pc is an edge of T with parameters $(G_p, X_p, N_p, \mathcal{F}_{1,p}, \mathcal{F}_{2,p})$ and $(G_c, X_c, N_c, \mathcal{F}_{1,c}, \mathcal{F}_{2,c})$ respectively. We say that a vertex $v \in G$ is *added to level set i during the call corresponding to p* if $v \notin (\mathcal{L}_i(G, G_p, \mathcal{F}_{1,p}, N_p) \cup \mathcal{L}_i(G, G_p, \mathcal{F}_{2,p}, N_p))$ and $v \in (\mathcal{L}_i(G, G_c, \mathcal{F}_{1,c}, N_c) \cup \mathcal{L}_i(G, G_c, \mathcal{F}_{2,c}, N_c))$. Note that for a vertex v to be added to level set i during call p, the edge pc must be either a balanced separator edge or a boosted balanced separator edge and $v \in (\mathcal{L}_{i-1}(G, G_p, \mathcal{F}_{1,p}, N_p) \cup \mathcal{L}_{i-1}(G, G_p, \mathcal{F}_{2,p}, N_p))$. Given a path P in T we say that a vertex v is added to level set i in path P if there is at least one node u such that v is added to level set i during the call corresponding to u. Note that it is possible for a vertex v to be added to level set i in path P multiple times since the level sets $\mathcal{F}_1$ and $\mathcal{F}_2$ can get set to the empty set multiple times.

Additionally, we say that a vertex $v \in G$ is *removed from level set i during the call corresponding to p* if $v \in (\mathcal{L}_i(G, G_p, \mathcal{F}_{1,p}, N) \cup \mathcal{L}_i(G, G_p, \mathcal{F}_{2,p}, N))$ and $v \notin (\mathcal{L}_i(G, G_c, \mathcal{F}_{1,c}, N) \cup \mathcal{L}_i(G, G_c, \mathcal{F}_{2,c}, N))$. We say that a vertex v is added to level set i in path P if there is at least one node u such that v is removed from level set i during the call corresponding to u. Note that it is possible for a vertex v to be removed from level set i in path P multiple times since the level sets $\mathcal{F}_1$ and $\mathcal{F}_2$ can get set to the empty set multiple times. Furthermore, note that if the call that corresponds to the first vertex of P has the property that both lists $\mathcal{F}_1$ and $\mathcal{F}_2$ are the empty set, then the number of vertices added to level set i in path P, counting multiplicity, is at least as much as the number of vertices removed from level set i in path P. This holds because if for a node $u \in P$ a vertex v is removed from level set i during the call corresponding to u, the vertex v must have been first added to level set i during the call corresponding to w for some $w \in P$ that comes before u in P.

Lemma 4.3.28. *Let t be a positive integer, G be an n-vertex graph $S_{t,t,t}$-free graph, T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, and P a path of T such that no edges*

of P are type 1 extended strip decomposition edge and P contains c balanced separator and boosted balanced separator edges. Then there exists a natural number N such that for any vertex $v \in P$ with parameters $(G_v, X_v, N_v, \mathcal{F}_{1,v}, \mathcal{F}_{2,v})$ it holds that $N_v = N$ and at most $c \cdot 56000c_t^4 \log^4(N)N/2^i$ vertices are added to level set i in P.

Proof: Let t, G, T, c, and P be as in the statement of this lemma. The fact that there exists a natural number N such that for any vertex $v \in P$ with parameters $(G_v, X_v, N_v, \mathcal{F}_{1,v}, \mathcal{F}_{2,v})$ it holds that $N_v = N$ follows from the fact that P has no type 1 extended strip decomposition edges.

Now, let p be an node of P and let $(G_p, X_p, N, \mathcal{F}_{1,p}, \mathcal{F}_{2,p})$ be the parameters of p. Note that if there is a vertex v that is added to level set i during call p, it must be that the vertex v is in $\mathcal{L}_{i-1}(G, G_p, \mathcal{F}_{1,p}, N)$ or $\mathcal{L}_{i-1}(G, G_p, \mathcal{F}_{2,p}, N)$.

First, assume that p is a balanced separator node of P and the set B_p is the balanced separator core added in the call that corresponds to p, so $|B_p| \leqslant 28000c_t^4 \log^4(N)$. Since p is a balanced separator node, there is no branchable vertex in the call corresponding to p. Therefore each vertex of B_p has less than $N/2^{i-1}$ neighbors in level set $\mathcal{L}_{i-1}(G, G_p, \mathcal{F}_{2,p}, N)$ and therefore at most $|B_p|N/2^{i-1} \leqslant 56000c_t^4 \log^4(N)N/2^i$ vertices are added to level set i during call p.

Next, assume that p is a boosted balanced separator node of P, and let X be the boosted balanced separator core added in the call that corresponds to p (if $X_p \neq \perp$ then $X = X_p$, and if $X_p = \perp$ then the set X was discovered in the call that corresponds to p). Let p' be the first ancestor of p in T where X was discovered (possibly with $p' = p$) and let $(G_{p'}, X_{p'}, N_{p'}, \mathcal{F}_{1,p'}, \mathcal{F}_{2,p'})$ be the parameters of p', so $X \subseteq V(G_{p'})$ and $|X| \leqslant c_t \log(N_{p'})$. It follows that all nodes between p' and p are branch nodes, type 2 extended strip decomposition nodes, and balanced separator nodes (since all recursive calls of boosted balanced separator nodes and type 1 extended strip decomposition nodes

would reset X to $\perp$). It follows that $\mathcal{F}_{1,p'} = \mathcal{F}_{1,p}$ and $N_{p'} = N$ and so $|X| \leqslant c_t \log(N)$. Additionally, since X was discovered in the call that corresponds to p', p' cannot be a branch node so there is no branchable vertex in the call corresponding to p', in particular, there is no vertex in $G_{p'}$ that has at least $N/2^{i-1}$ neighbors in $\mathcal{L}_{i-1}(G, G_{p'}, \mathcal{L}_{1,p'}, N)$. Therefore each vertex of X has at most $N/2^{i-1}$ neighbors in level set $\mathcal{L}_{i-1}(G, G_{p'}, \mathcal{F}_{1,p'}, N)$ and therefore (since G_p is an induced subgraph of $G_{p'}$ and $\mathcal{F}_{1,p'} = \mathcal{F}_{1,p}$) in level set $\mathcal{L}_{i-1}(G, G_p, \mathcal{F}_{1,p}, N)$. Hence at most $|X|N/2^{i-1} \leqslant 2c_t \log(N)N/2^i$ vertices are added to level set i during call p.

In either case, we conclude that at most $56000c_t^4 \log^4(N)N/2^i$ vertices are added to level set i in the call p. Since by assumption there are c boosted balanced separator and balanced separator edges in P, it follows that at most $c \cdot 56000c_t^4 \log^4(N)N/2^i$ vertices are added to level set i in P. ■

Lemma 4.3.29. *Let t be a positive integer, G an $S_{t,t,t}$-free n-vertex graph, T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$, and P a path of T. Then P contains at most $10^{14}c_t^{12} \log^{16}(n)$ success edges.*

Proof: Let t, G, T, and P be as in the statement of this lemma. Let P' be a maximal subpath of P that contains no type 1 extended strip decomposition edges, c the number of balanced separator and boosted balanced separator edges in P', u the first vertex of P' (note by the maximality of P' the in-edge of u is either a type 1 extended strip decomposition or u is the root of T), and $(G_u, X_u, N_u, \mathcal{F}_{1,u}, \mathcal{F}_{2,u})$ the parameters of u.

By Lemma 4.3.28 there is a natural number N such that for any node $w \in P$ with parameters $(G_w, X_w, N_w, \mathcal{F}_{1,w}, \mathcal{F}_{2,w})$ it holds that $N_w = N$, hence $N_u = N$, and there are at most $c \cdot 56000c_t^4 \log^4(N)N/2^i$ vertices added to level set i in P' (counting multiplicity). Since the in-edge of u is a type 1 extended strip decomposition edge (or u is the root vertex of T), it follows that $\mathcal{F}_{1,u} = \mathcal{F}_{2,u} = \emptyset$. Hence if a vertex v is removed in P', the

vertex v must also have been added in P', so at most $c56000c_t^4 \log^4(N)N/2^i$ vertices can be removed from level set i in P' (counting multiplicity).

Let pc be a success edge of P' and let $(G_p, X_p, N, \mathcal{F}_{1,p}, \mathcal{F}_{2,p})$ and $(G_c, X_c, N, \mathcal{F}_{1,c}, \mathcal{F}_{2,c})$ be the parameters of p and c respectively. Then there is some vertex $v \in G_p$ such that $G_c = G_p - N_{G_p}[v]$. Furthermore there is some natural number i such that $N_{G_p}[v]$ contains at least $N/2^i$ vertices in either level set $\mathcal{L}_i(G, G_p, \mathcal{F}_{1,p}, N)$ or $\mathcal{L}_i(G, G_p, \mathcal{F}_{2,p}, N)$, call this set of vertices S. It follows that all of the vertices of S are removed from level set i in call p. Since at most $c \cdot 56000c_t^4 \log^4(N)N/2^i$ vertices are added to level set i in P' (therefore as noted in the first paragraph of this proof at most $c \cdot 56000c_t^4 \log^4(N)N/2^i$ vertices can be removed to level set i in P') this can happen to level set i in P' at most $c \cdot 56000c_t^4 \log^4(N)$ times before it is empty. Since by Lemma 4.3.23 for all $j > \log(N)$ $\mathcal{L}_j(G, G_p, \mathcal{F}_{2,p}, N)$ and $\mathcal{L}_j(G, G_p, \mathcal{F}_{2,p}, N)$ are already empty, it follows that $i \leqslant \log(N)$. Therefore there can be at most $c \cdot 56000c_t^4 \log^5(N)$ success edges before all level sets are empty. Hence there are at most $c \cdot 56000c_t^4 \log^5(N) \leqslant c \cdot 56000c_t^4 \log^5(n)$ success edges in P'.

Lastly, by Lemmas 4.3.25 and 4.3.27 there are at most $5200c_t^4 \log^5(n)$ and $10^9 c_t^8 \log^{11}(n)$ boosted balanced separator and balanced separator edges respectively in P. It then follows that there are at most $(5200c_t^4 \log(n)^5 + 10^9 c_t^8 \log^{11}(n) + 1)(56000c_t^4 \log^5(n)) \leqslant 10^{14} c_t^{12} \log^{16}(n)$ success edges in P. ■

Lemma 4.3.30. *Let t be a positive integer, G an $S_{t,t,t}$-free n-vertex graph and T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$. Then there are at most n failure edges on any root to leaf path in T.*

Proof: Let t, G, P, and T be as in the statement of this lemma, and let pc be a failure edge of T. Then $|G_c| = |G_p| - 1$. Hence it is impossible for P to have $n+1$ failure edges. ■

Let G be an n-vertex $S_{t,t,t}$-free graph. In order to bound the number of nodes in the recursion tree T generated by $\text{IND}(G, \perp, |G|, \emptyset, \emptyset)$ we will provide a sequence which will uniquely determine every node of T, then prove that there is at most $n^{\mathcal{O}(c_t^{12}\log^{16}(n))}$ such sequences. Let u be a node in T and let P be the path from the root node to u. The sequence we give is just the edges (given by their labels) taken on the path from the root to the vertex u. Let us assume that we have been given such a sequence of edge labels, S, that corresponds to the sequence of edge labelings of P. We show how to use S to reconstruct the path P from the root node to u (proving this sequence uniquely determines the vertex u). Assume we are currently at a vertex w, if w is a branch node, then the next label in S must either be a success or failure edge, and whichever one it is uniquely determines the next node in our path to u. Similarly, if w is a balanced separator or boosted balanced separator node, then the next label in S must be a balanced separator edge or boosted balanced separator edge and w has exactly one child, so again the next node in the path is uniquely determined. If w is a type 1 or type 2 extended strip decomposition node though, there exists some constant c (by Observation 4.3.21, independent of the choice of G) so that w can have up to n^c children, and all edges going to these children have the same label, hence the next node in the path is not uniquely determined. To fix this issue we define an *enriched recursion tree* to be a recursion tree T such that for every type 1 or type 2 extended strip decomposition node, each of its out edge labels are additionally given a unique number $1 - n^c$. It follows that with this enriching, the next label of S will uniquely determine a child of w and then this sequence uniquely determines the vertex u.

Lemma 4.3.31. *Let t be a positive integer, G an $S_{t,t,t}$-free n-vertex graph, and T the recursion tree generated by IND$(G, \perp, |G|, \emptyset, \emptyset)$. Then T has at most $n^{\mathcal{O}(c_t^{12}\log^{16}(n))}$ nodes.*

Proof: Let t, G and T be as in the statement of this lemma, and let c be the constant

from Observation 4.3.21. Furthermore, assume that T is an enriched recursion tree, that is for every $u \in T$ that is a type 1 or type 2 extended strip decomposition node, the labels of the out edges of u are given an additional unique integer between 1 and n^c, where c is the constant given in Observation 4.3.21.

We consider the set of all edge label sequences which contain at most $64c_t^2 \log^3(n)$ type 1 extended strip decomposition edges (by Lemma 4.3.22 that is as many as T can have), $5200c_t^4 \log^5(n)$ boosted balanced separator edges (by Lemma 4.3.25 that is as many as T can have), $10^7 c_t^7 \log^9(n)$ type 2 extended strip decomposition edges (by Lemma 4.3.26 that is as many as T can have), $10^9 c_t^8 \log^{11}(n)$ balanced separator edges (by Lemma 4.3.27 that is as many as T can have), $10^{14} c_t^{12} \log^{16}(n)$ (by Lemma 4.3.29 that is as many as T can have) success edges, and n failure edges (by Lemma 4.3.30 that is as many as T can have). To bound the number of such sequenced, first note that since there are six types of edges and none can appear over $n^{\mathcal{O}(1)}$ times there are at most $n^{\mathcal{O}(1)}$ choices for the number of each type of edge, call these choices n_1, n_2, n_3, n_4, n_5, and n_6 and let n_6 be the number which denotes the number of failure edges. The number of possible sequences with these number choices is then $\binom{n_1+n_2+n_3+n_4+n_5+n_6}{n_1,n_2,n_3,n_4,n_5} \leqslant (n_1+n_2+n_3+n_4+n_5+n_6)^{n_1+n_2+n_3+n_4+n_5}$. Since $n_6 \leqslant n$ and for $i < 6$, $n_i \leqslant 10^{14} c_t^{12} \log^{16}(n)$ this number is at most $n^{\mathcal{O}(c_t^{12} \log^{16}(n))}$. Since there are at most $n^{\mathcal{O}(1)}$ difference choices for values of the n_i's, it follows there are at most $n^{\mathcal{O}(1)} n^{\mathcal{O}(c_t^{12} \log^{16}(n))} = n^{\mathcal{O}(c_t^{12} \log^{16}(n))}$ sequences of this type.

Next we consider the "enriched" version of these sequences, that is, for each type 1 and type 2 extended strip decomposition edge in a sequence of S we give it some number between 1 and n^c. as there are at most $64c_t^2 \log^3(n)$ and $10^7 c_t^7 \log^9(n)$ type 1 and type 2 extended strip decomposition edges respectively there are at most $n^{c(64c_t^2 \log^3(n)+10^7 c_t^7 \log^9(n))}$ $n^{\mathcal{O}(c_t^{12} \log^{16}(n))} = n^{\mathcal{O}(c_t^{12} \log^{16}(n))}$ of these enriched sequences. It follows now that for any $u \in T$, the sequences of edges in the path from the root node to u is contained in S, and by the discussion just before the statement of this lemma, for each $u \in T$ is edge sequence

is unique. So, since $|S| = n^{\mathcal{O}(c_t^{12}\log^{16}(n))}$ it follows that T has at most $n^{\mathcal{O}(c_t^{12}\log^{16}(n))}$ nodes. ■

We are now ready to prove Theorem 4.1.4.

Proof: [Proof of Theorem 4.1.4] Let t be a positive integer, G an n-vertex $S_{t,t,t}$-free graph, and T the recursion tree generated by IND(G, $\perp$, $|G|, \emptyset, \emptyset$). By Lemma 4.3.20 IND($G$, $\perp$, $|G|, \emptyset, \emptyset$) returns the weight of a maximum weight independent set of G and by Lemma 4.3.31 $|T| = n^{\mathcal{O}(c_t^{12}\log^{16}(n))}$. All that must be verified then is for each $u \in T$ the amount of time spend in the call that correspond to u runs in polynomial time. We justify this runtime by discussing only the runtime of the steps of IND that do not clearly run polynomial time.

That the step of finding $X = \mathsf{esd}(G)$ runs in polynomial time was justified in Definition 4.3.4. That $\mathsf{matching}(H, \eta)$ runs in polynomial time follows from Lemma 4.3.7 and the facts that in type 2 extended strip decompositions calls, (H, η) is rigid, and therefore $|H| = \mathcal{O}(n)$, and in type 1 extended strip decomposition calls (H, η) is the extended strip decomposition inferred by some (X', G') and so $|H| = n^{\mathcal{O}(1)}$ (see the discussion after Definition 4.3.4), and by Lemma 4.3.7 $\mathsf{matching}(H, \eta)$ runs in time polynomial in $|G|$ and $|H|$. Lastly, that the second bullet point of IND, where Lemma 4.3.8 is applied runs in polynomial time is stated in Lemma 4.3.8. ■

4.4 Extended Strip Lemma

The main result of this section is the following:

Lemma 4.4.1 (Extended strip decomposition or small balanced separator)**.** *For every fixed integer t, there exists an integer c_t and a polynomial-time algorithm that, given an n-vertex graph G and a weight function $\mathfrak{w} : V(G) \to [0, +\infty)$, returns one of the following:*

1. *an induced copy of $S_{t,t,t}$ in G;*
2. *a $0.99\mathfrak{w}(G)$-balanced separator for $(G, \mathfrak{w})$ dominated by $c_t \cdot \log n$ vertices;*
3. *a rigid extended strip decomposition of G where no particle is of weight larger than $0.5\mathfrak{w}(G)$.*

The main difference between Lemma 4.4.1 and the main result of [50], namely Theorem 4.1.6, is that Lemma 4.4.1 promises in the last output an extended strip decomposition of the entire graph, not the graph with a small number of neighborhoods deleted.

The algorithm of Lemma 4.4.1 first applies Theorem 4.1.6 to find either an induced copy of $S_{t,t,t}$ (which can be immediately returned) or a set Z of size $\mathcal{O}(\log n)$ together with a rigid extended strip decomposition (H, η) of $G - N[Z]$ such that every particle of (H, η) has weight at most $0.5\mathfrak{w}(G)$. Then, we attempt to put back vertices of $N[Z]$ one-by-one to (H, η), maintaining the property that every particle of (H, η) has weight at most $0.5\mathfrak{w}(G)$. The following lemma, whose proof spans the remainder of this section, shows that in every such attempt, we can either succeed or obtain one of the first two outcomes of Lemma 4.4.1.

Lemma 4.4.2. *For every fixed integer t there exists an integer c_t and a polynomial-time algorithm that, given an n-vertex graph G, a weight function $\mathfrak{w} : V(G) \to [0, +\infty)$, a real $\tau \geqslant \mathfrak{w}(G)$, a vertex $v \in V(G)$, and a rigid extended strip decomposition (H, η) of $G - v$ with every particle of weight at most 0.5τ, returns one of the following:*

1. *an induced copy of $S_{t,t,t}$ in G;*
2. *a 0.99τ-balanced separator for $(G, \mathfrak{w})$ dominated by at most c_t vertices;*
3. *a rigid extended strip decomposition of G where no particle is of weight larger than 0.5τ.*

Let us formally prove Lemma 4.4.1 using Lemma 4.4.2. *Proof:* [Proof of Lemma 4.4.1.] Let $\tau \coloneqq \mathfrak{w}(G)$. Run Theorem 4.1.6 on $(G, \mathfrak{w})$. If an $S_{t,t,t}$ is returned, return it as well. Otherwise, we have a set Z of size $\mathcal{O}(\log n)$ together with a rigid extended strip decomposition (H, η) of $G - N[Z]$ such that every particle of (H, η) has weight at most 0.5τ.

Enumerate $N[Z]$ as $\{v_1, v_2, \ldots, v_k\}$. Let $G_i = G - \{v_1, \ldots, v_i\}$ for $0 \leqslant i \leqslant k$, so that $G_0 = G$ and $G_k = G - N[Z]$. Denote $(H_k, \eta_k) \coloneqq (H, \eta)$. We compute a sequence $(H_i, \eta_i)_{i=k}^{0}$ of rigid extended strip decompositions of graphs G_i whose every particle has weight at most 0.5τ as follows. For each $i = k, k-1, \ldots, 1$ apply Lemma 4.4.2 to G_{i-1}, v_i (recall that $G_i = G_{i-1} - v_{i-1}$), τ, and the rigid extended strip decomposition (H_i, η_i). If an $S_{t,t,t}$ is returned, terminate the algorithm and return it, too. If a 0.99τ-balanced separator X is returned, return $X \cup N[Z]$ as a 0.99τ-balanced separator of G dominated by $\mathcal{O}(\log n)$ vertices. Otherwise, denote the output rigid extended strip decomposition of G_{i-1} by (H_{i-1}, η_{i-1}) and continue with the next step. If we reach (H_0, η_0), we return it as the third output of Lemma 4.4.1. ■

The remainder of this section is devoted to the proof of Lemma 4.4.2.

4.4.1 Turning separations in H into separators in G

Let us make the following trivial observation.

Lemma 4.4.3. *If (H, η) is a rigid extended strip decomposition of a graph G and $x \in V(H)$ is of degree more than one, then $\bigcup_{y \in N_H(x)} \eta(xy, x)$ is dominated by two vertices.*

Proof: Pick two neighbors $y_1, y_2 \in N_H(x)$ and any $v_i \in \eta(xy_i, x)$ for $i = 1, 2$. (Recall that we mandate the interfaces $\eta(xy_i, x)$ to be nonempty in a rigid extended strip decomposition.) Then, v_i dominates $\bigcup_{y \in N_H(x) - \{y_i\}} \eta(xy, x)$, so $\{v_1, v_2\}$ dominates $\bigcup_{y \in N_H(x)} \eta(xy, x)$. ■

For an extended strip decomposition (H, η) of a graph G and a set $A \subseteq V(H)$, the *preimage* of A in G is the set $\overleftarrow{\eta}_{(H,\eta)}(A) \subseteq V(G)$ consisting of:

- all vertex sets $\eta(x)$ for $x \in A$;
- all edge sets $\eta(xy)$ for $|\{x, y\} \cap A| \geqslant 1$;
- all triangle sets $\eta(xyz)$ for $|\{x, y, z\} \cap A| \geqslant 2$.

We make the following two observations based on Lemma 4.4.3.

Lemma 4.4.4. *Let (H, η) be an extended strip decomposition of a graph G and let (A, B) be a separation in H. Let $X = \bigcup_{x \in A \cap B} \bigcup_{y \in N_H(x)} \eta(xy, x)$. Then, every connected component of $G - X$ is contained in one of the following sets: $\overleftarrow{\eta}_{(H,\eta)}(A - B)$, $\overleftarrow{\eta}_{(H,\eta)}(B - A)$, $\eta(x)$ for some $x \in A \cap B$, $\eta(xy)$ for some $xy \in E(H[A])$, or $\eta(xyz)$ for some triangle $xyz \in T(H)$ with $|\{x, y, z\} \cap A \cap B| \geqslant 2$. Furthermore, if (H, η) is rigid and every vertex of $A \cap B$ has degree at least 2, then X is dominated by at most $2|A \cap B|$ vertices.*

Proof: Observe that every set Γ being either $\eta(x)$ for $x \in A \cap B$, $\eta(xy)$ for some $xy \in E(H[A])$, or $\eta(xyz)$ for a triangle $xyz \in T(H)$ with $|\{x, y, z\} \cap A \cap B| \geqslant 2$ satisfies $N_G(\Gamma) \subseteq X$. Similarly, every edge that has exactly one endpoint on $\overleftarrow{\eta}_{(H,\eta)}(A - B) - X$ has its second endpoint in X and every edge that has exactly one endpoint on $\overleftarrow{\eta}_{(H,\eta)}(B - A) - X$ has its second endpoint in X. This proves the desired separation properties of X. The second part of the lemma follows directly from Lemma 4.4.3. ■

Lemma 4.4.5. *Let $0 < \delta < 0.5$ be a constant. Let (H, η) be an extended strip decomposition of a graph G with weight function $\mathfrak{w}$. Assume that no particle of (H, η) has weight more than $(1 - \delta)\mathfrak{w}(G)$, but there is a particle of (H, η) that has weight at least $\delta\mathfrak{w}(G)$. Then there exists a set $F \subseteq V(H)$ of size at most 2 such that $X \coloneqq \bigcup_{x \in F} \bigcup_{y \in N_H(x)} \eta(xy, x)$ is an $(1 - \delta)\mathfrak{w}(G)$-balanced separator in G. Furthermore, if (H, η) is rigid, then X is dominated by at most four vertices.*

Proof: Observe that inclusion-wise maximal particles are vertex particles for isolated vertices of H and full edge particles. Without loss of generality, we can assume that there is a particle of (H, η) of one of those two types that has weight at least $\delta\mathfrak{w}(G)$.

Assume first that $\mathfrak{w}(\eta(x)) \geqslant \delta\mathfrak{w}(G)$ for some isolated $x \in V(H)$. We also have $\mathfrak{w}(\eta(x)) \leqslant (1-\delta)\mathfrak{w}(G)$ by the assumptions of the lemma. Since $\eta(x)$ is the union of some connected components of G by the properties of an extended strip decomposition, $\emptyset$ is a $(1-\delta)\mathfrak{w}(G)$-balanced separator in G and the we are done.

Assume now that $\mathfrak{w}(A^{xy}_{xy}) \geqslant \delta\mathfrak{w}(G)$ for an edge $xy \in E(H)$. Again, by the assumptions of the lemma we have $\mathfrak{w}(A^{xy}_{xy}) \leqslant (1-\delta)\mathfrak{w}(G)$. Let F be the set of those vertices of $\{x, y\}$ that are of degree more than one in H and let $X \coloneqq \bigcup_{x' \in F} \bigcup_{y' \in N_H(x')} \eta(x'y', x')$. It follows from the properties of an extended strip decomposition that X separates A^{xy}_{xy} from $V(G) - A^{xy}_{xy}$, i.e., every path from A^{xy}_{xy} to $V(G) - A^{xy}_{xy}$ contains a vertex from X. If (H, η) is rigid, then Lemma 4.4.3 implies that X is dominated by at most $2|F| \leqslant 4$ vertices. Since $\delta\mathfrak{w}(G) \leqslant \mathfrak{w}(A^{xy}_{xy}) \leqslant (1-\delta)\mathfrak{w}(G)$, X is the desired $(1-\delta)\mathfrak{w}(G)$-balanced separator. ■

4.4.2 Locally cleaning an extended strip decomposition

We will need a few connectivity properties of an extended strip decomposition, a bit stronger than just being rigid. Luckily, they are easy to obtain via local modifications.

Let (H, η) be an extended strip decomposition of a graph G. A *local cleaning step* for (H, η) is one of the following modifications.

removing an isolated vertex with an empty set If $x \in V(H)$ is an isolated vertex satisfying $\eta(x) = \emptyset$, delete x from $V(H)$.

moving an isolated vertex set If $x \in V(H)$ is an isolated vertex with nonempty vertex set and $y \in V(H)$ is any other vertex that is not an isolated vertex with an

empty vertex set, we set $\eta(y) \coloneqq \eta(y) \cup \eta(x)$ and $\eta(x) \coloneqq \emptyset$.

moving a disconnected component of an edge set If for an edge $xy \in E(H)$ there exists a connected component C of $\eta(xy) - (\eta(xy, x) \cup \eta(xy, y))$ with no neighbors in $\eta(xy, y) - \eta(xy, x)$, we set $\eta(x) \coloneqq \eta(x) \cup C$ and $\eta(xy) \coloneqq \eta(xy) - C$.

moving a disconnected component of a triangle set If C is a connected component of a triangle $xyz \in T(H)$ with no neighbors in $\eta(yz, y) \cap \eta(yz, z)$, then we set $\eta(x) \coloneqq \eta(x) \cup C$ and $\eta(xyz) \coloneqq \eta(xyz) - C$.

moving a disconnected vertex of an interface If for an edge $xy \in E(H)$ there is a vertex $v \in \eta(xy, x) - \eta(xy, y)$ such that $N_G[v] \cap \eta(xy) \subseteq \eta(xy, x)$, set $\eta(x) \coloneqq \eta(x) \cup \{v\}$ and $\eta(xy) \coloneqq \eta(xy) - \{v\}$.

removing an edge with an empty interface If $xy \in E(H)$ is an edge with $\eta(xy, x) = \emptyset$, we set $\eta(y) \coloneqq \eta(y) \cup \eta(xy)$ and delete the edge xy from H.

suppressing a degree-1 vertex If $x \in V(H)$ is of degree 1 in H, with its unique neighbor y, set $\eta(y) \coloneqq \eta(y) \cup \eta(xy) \cup \eta(x)$, $\eta(x) \coloneqq \emptyset$ and delete the edge xy.

An extended strip decomposition is *locally cleaned* if no local cleaning step is applicable.

The following observations are immediate.

Lemma 4.4.6. *If (H, η) is an extended strip decomposition of G and (H', η') is a result of applying the first applicable local cleaning step to (H, η), then (H', η') is also an extended strip decomposition of G with $V(H') \subseteq V(H)$ and $E(H') \subseteq E(H)$. Furthermore, we have the following:*

- *For every $xy \in E(H')$ we have $\eta'(xy) \subseteq \eta(xy)$, $\eta'(xy, x) \subseteq \eta(xy, x)$, and $\eta'(xy, y) \subseteq \eta(xy, y)$.*

- *For every $xyz \in T(H')$, we have $\eta'(xyz) \subseteq \eta(xyz)$.*
- *The following potential strictly increases from (H, η) to (H', η'): the number of vertices of G in vertex sets of H/H', minus the number of vertices and edges of H/H', and additionally minus the number of vertices of H/H' that are not isolated vertices with empty vertex sets.*

Proof: The only nontrivial check is that whenever we delete an edge e of H, all triangles involving e already have empty sets. This follows for the "removing an edge with an empty interface" step due to inapplicability of the "moving a disconnected component of a triangle" step. ■

Note that the last property ensures that the local cleaning operation terminates and indeed produces a locally cleaned extended strip decomposition.

Lemma 4.4.7. *Let (H, η) be an extended strip decomposition of G that is locally cleaned. Then (H, η) is a rigid extended strip decomposition such that either H consists of a single vertex with the whole $V(G)$ in its vertex set, or every vertex of H has degree at least 2.*

Proof: If there was an isolated $x \in V(H)$ with nonempty $\eta(x)$, the "moving isolated vertex set" step would apply, unless already $\eta(x) = V(G)$. If there were an isolated vertex $x \in V(H)$ with $\eta(x) = \emptyset$, the "removing an isolated vertex with an empty set" step would apply. If there were a vertex $x \in V(H)$ of degree one, the "suppressing a degree-1 vertex" step would apply. If there were an edge $xy \in E(H)$ with empty $\eta(xy)$, $\eta(xy, x)$, or $\eta(xy, y)$, the "removing an edge with an empty interface" step would apply. This concludes the proof. ■

Recall that in the context of Lemma 4.4.2, we have access to a rigid extended strip decomposition (H, η) of $G - v$ with all particles of weight at most 0.5τ. We want to add v to the extended strip decomposition; on the way there we can identify an induced $S_{t,t,t}$ or a 0.99τ-balanced separator that is dominated by a small number of vertices.

If $\mathfrak{w}(V(G)-N_G[v]) \leqslant 0.99\tau$, then we can return $N_G[v]$ as the promised 0.99τ balanced separator. Hence, we assume $\mathfrak{w}(V(G) - N_G[v]) \geqslant 0.99\tau$, that is,

$$\tau \leqslant 0.99^{-1}\mathfrak{w}(V(G) - N_G[v]). \tag{4.1}$$

Let $S \subseteq N_G[v]$ with $v \in S$. A *local cleaning operation applied to* S consists of the following:

1. Computing an extended strip decomposition (H_S, η_S) of $G - S$ by restricting each set $\eta(\cdot)$ from (H, η) to $V(G) - S$. (Note that in this step (H_S, η_S) may not be rigid, as some sets $\eta_S(x)$, $\eta_S(xy)$ or interfaces $\eta_S(xy, x)$ may be empty.)

2. Iteratively, while possible, apply the first applicable local cleaning operation to (H_S, η_S).

3. If at any moment of the process there exists a particle of (H_S, η_S) whose weight is at least $0.01\mathfrak{w}(V(G) - S)$, apply Lemma 4.4.5 to it, obtaining a $0.99\mathfrak{w}(V(G) - S)$-balanced separator X of $G - S$ equal to $\bigcup_{x\in F}\bigcup_{y\in N_{H_S}(x)} \eta_S(xy, x)$ for some $F \subseteq V(H_S)$ of size at most 2. Since $V(H_S) \subseteq V(H)$ and $\eta_S(xy, x) \subseteq \eta(xy, x)$ for every $xy \in E(H_S)$ by Lemma 4.4.6, $X \subseteq \bigcup_{x\in F}\bigcup_{y\in N_H(x)} \eta(xy, x)$. Hence, by Lemma 4.4.3, X is dominated by at most four vertices in G (not necessarily in $G - S$). By (4.1) and since $\tau \geqslant \mathfrak{w}(G)$, every connected component of $G - S - X$ has weight at most

$$\mathfrak{w}(V(G) - S) - 0.01\mathfrak{w}(V(G) - S) = 0.99\mathfrak{w}(V(G) - S) \leqslant 0.99\tau.$$

Thus, we return $X \cup S$ as a 0.99τ-balanced separator of G dominated by at most five vertices.

Initially, every particle of (H_S, η_S) has weight at most 0.5τ which, by (4.1), is upper bounded by $\frac{0.5}{0.99}\mathfrak{w}(V(G) - N_G[v]) \leqslant 0.55\mathfrak{w}(V(G) - S)$. Thus, if Lemma 4.4.5 is triggered before the first cleaning operation, its assumptions are satisfied. Later in the process, we apply a local cleaning operation to an extended strip decomposition whose every particle is of weight at most $0.01\mathfrak{w}(V(G) - S)$. Since every local cleaning operation moves a subset of one set $\eta(\cdot)$ to another, after a single local cleaning operation every particle is of weight at most $0.02\mathfrak{w}(V(G) - S)$. This justifies the assumptions of Lemma 4.4.5 if triggered in later steps.

We conclude with the following straightforward summary of the properties of the result of local cleaning (cf. Lemma 4.4.7).

Lemma 4.4.8. *Let $S \subseteq V(G)$ with $v \in S$ and assume that the local cleaning operation applied to S finished with an extended strip decomposition (H_S, η_S) of $G - S$. Then:*

- *(H_S, η_S) is a rigid extended strip decomposition of $G - S$.*
- *$V(H_S) \subseteq V(H)$ and $E(H_S) \subseteq E(H)$.*
- *For every $x \in V(H_S)$ we have $\eta_S(x) \supseteq \eta(x)$.*
- *For every $xy \in E(H_S)$ we have $\eta_S(xy) \subseteq \eta(xy)$, $\eta_S(xy, x) \subseteq \eta(xy, x)$, and $\eta_S(xy, y) \subseteq \eta(xy, y)$.*
- *For every $xyz \in T(H_S)$ we have $\eta_S(xyz) \subseteq \eta(xyz)$.*
- *Every particle of (H_S, η_S) is of weight at most $0.01\mathfrak{w}(V(G) - S) \leqslant 0.01\tau$.*
- *Every vertex of H_S is of degree at least 2.*

We start the algorithm of Lemma 4.4.2 with applying the local cleaning operation to (H, η), that is, the case $S = \{v\}$. We either return a 0.99τ-balanced separator dominated

by at most five vertices or (we reuse the name (H, η) for the obtained extended strip decomposition, slightly abusing the notation) ensure the following two properties.

$$\text{Every particle of } (H, \eta) \text{ is of weight at most } 0.01\tau. \tag{4.2}$$

$$\text{Every } x \in V(H) \text{ is of degree at least 2.} \tag{4.3}$$

We henceforth proceed assuming (4.2) and (4.3). Note that Lemma 4.4.3 implies the following.

$$\text{For every } F \subseteq V(H) \text{ the set } \bigcup_{x \in F} \bigcup_{y \in N_H(x)} \eta(xy, x) \text{ is dominated by at most } 2|F| \text{ vertices in } G. \tag{4.4}$$

4.4.3 A wall avoiding $N[v]$

An edge $xy \in E(H)$ is *v-safe* if there exists a path in $G[\eta(xy)] - N[v]$ between a vertex of $\eta(xy, x)$ and a vertex of $\eta(xy, y)$. A subgraph of H is *v-safe* if all its edges are v-safe.

We now show the following.

Lemma 4.4.9. *For every constant σ there exists a constant σ' such that in polynomial time we can either find a 0.99τ-balanced separator dominated by at most σ' vertices in G or a v-safe wall W in H of sidelength 3σ with the following property:*

$$\forall_{(A,B) \in \mathcal{T}_W} \mathfrak{w}(\overleftarrow{\eta}_{(H,\eta)}(B - A)) \geqslant 0.99\tau. \tag{4.5}$$

Proof: Apply the local cleaning operation to (H_S, η_S) where $S \coloneqq N[v]$. If a 0.99τ-balanced separator X is found, return $X \cup S$ as the promised 0.99τ-balanced separator.

Otherwise, we have an extended strip decomposition (H_S, η_S) of $G - N[v]$ that satisfies the properties of Lemma 4.4.8.

Let $k \coloneqq f_{\mathrm{KTW20}}(\sigma)$ where f_{KTW20} comes from Theorem 4.2.2. Note that k is still a constant.

Assume that there exists a separation (A, B) in H_S of order less than k with both

$$\mathfrak{w}(\overleftarrow{\eta}_{(H_S,\eta_S)}(A - B)) \leqslant 0.99\tau \quad \text{and} \quad \mathfrak{w}(\overleftarrow{\eta}_{(H_S,\eta_S)}(B - A)) \leqslant 0.99\tau.$$

By Lemma 4.4.4, the set $X \coloneqq \bigcup_{x \in A \cap B} \bigcup_{y \in N_{H_S}(x)} \eta_S(xy, x)$ is a 0.99τ-balanced separator of $G - S$. Due to Lemma 4.4.8, we have $X \subseteq \bigcup_{x \in A \cap B} \bigcup_{y \in N_H(x)} \eta(xy, x)$, which is in turn dominated by at most $2|A \cap B| \leqslant 2k - 2$ vertices in G due to Lemma 4.4.3. Since $S = N[v]$, G admits a 0.99τ-balanced separator dominated by at most $2k - 1$ vertices. Since k is a constant, we can find such a separator in polynomial time and return it. Thus, henceforth we assume that such a separation (A, B) does not exist.

Let $\mathcal{T}'$ be a set consisting of every separation (A, B) in H_S of order less than k with

$$\mathfrak{w}(\overleftarrow{\eta}_{(H_S,\eta_S)}(B - A)) \geqslant 0.99\tau.$$

Since $\mathfrak{w}(G) \leqslant \tau$ and due to our assumption from the previous paragraph, for every separation (A, B) in H_S of order less than k, exactly one of (A, B) and (B, A) belongs to $\mathcal{T}'$. Also, for every $(A, B) \in \mathcal{T}'$ it holds that $\mathfrak{w}(\overleftarrow{\eta}_{(H_S,\eta_S)}(A - B)) \leqslant 0.01\tau$.

Assume that $\mathcal{T}'$ is not a tangle of order k. Then, there exist $(A_1, B_1), (A_2, B_2), (A_3, B_2) \in \mathcal{T}'$ with $A_1 \cup A_2 \cup A_3 = V(H_S)$. Let $F = (A_1 \cap B_1) \cup (A_2 \cap B_2) \cup (A_3 \cap B_3)$. We have $|F| \leqslant 3k - 3$. Let $X = \bigcup_{x \in F} \bigcup_{y \in N_{H_S}(x)} \eta_S(xy, y)$. Since $\mathfrak{w}(\overleftarrow{\eta}_{(H_S,\eta_S)}(A_i - B_i)) \leqslant 0.01\tau$ for $i = 1, 2, 3$ and every particle of (H_S, η_S) is of weight at most 0.01τ, X is a 0.99τ-balanced separator in $G - S$. Similarly as before, Lemmas 4.4.8 and 4.4.3 imply that X

is dominated by at most $6k-6$ vertices in G. Hence, $X \cup S$ is a 0.99τ-balanced separator in G dominated by at most $6k-5$ vertices in G. Since k is a constant, we can check in polynomial time if such a separator exists. Thus, henceforth we continue with the assumption that $\mathcal{T}'$ is a tangle of order k in H_S.

We now apply Theorem 4.2.2 to obtain a wall W in H_S of sidelength 3σ such that $\mathcal{T}'_W$, the tangle of order σ governed by W in H_S, is the restriction of $\mathcal{T}'$ to order σ. We observe that as σ is a constant, W can be computed in polynomial time for example by first guessing its pegs, and then applying an algorithm for Disjoint Paths of [100]. Since $E(H_S) \subseteq E(H)$, the wall W exists also in H. Observe that, due to the local cleaning operation, every edge $xy \in E(H_S)$ is v-safe in H. Hence, W is v-safe.

Let (A, B) be a separation in H of order less than k. Since $E(H_S) \subseteq E(H)$, we observe that $(A \cap V(H_S), B \cap V(H_S))$ is a separation in H_S of order less than k. Let then $\mathcal{T}$ be the set of all separations (A, B) of H of order less than k such that $(A \cap V(H_S), B \cap V(H_S))$ belongs to $\mathcal{T}'$. Similarly, let $\mathcal{T}_W$ be the set of all the separations (A, B) of H of order less than σ such that $(A \cap V(H_S), B \cap V(H_S))$ is in $\mathcal{T}'_W$. Then, $\mathcal{T}$ is a tangle of order k in H that satisfies

$$\forall_{(A,B)\in\mathcal{T}} \ \mathfrak{w}(\overleftarrow{\eta}_{(H_S,\eta_S)}(B-A)) \geqslant 0.99\tau.$$

Moreover, $\mathcal{T}_W$ is the tangle of order σ governed by W in H and is equal to the restriction of $\mathcal{T}$ to order σ. This proves (4.5). ■

We fix the wall W obtained via Lemma 4.4.9 for the remainder of the proof. In subsequent steps we are going to obtain more and more structural properties of (H, η) and W, at the cost of gradually shrinking W. The actual value of σ will be fixed at the end of the proof, so that the final remnants of W are still substantial.

4.4.4 Finding subdivided claws in a v-safe wall

In what follows, we will be thinking of every path P as a path with an orientation, so that P has a starting vertex $\overleftarrow{\mathbf{u}}(P)$ and an ending vertex $\overrightarrow{\mathbf{u}}(P)$, and for an integer $1 \leqslant i \leqslant |V(P)|$ we can speak of the i-th vertex $\mathbf{u}_i(P)$ of a path. Clearly, $\overleftarrow{\mathbf{u}}(P) = \mathbf{u}_1(P)$ and $\overrightarrow{\mathbf{u}}(P) = \mathbf{u}_{|V(P)|}(P)$.

For a moment, let us get out of the context of the proof of Lemma 4.4.2 and introduce an auxiliary tool for finding long disjoint paths in some parts of H and subwalls of W.

Lemma 4.4.10 (Finding paths of length t in a wall)**.** *Let $\widetilde{W}$ be a $2t$-subdivided wall of sidelength at least 3 in a graph $\widetilde{H}$. Let P_1, P_2, P_3 be vertex-disjoint paths such that $\overrightarrow{\mathbf{u}}(P_i)$ is a peg of $\widetilde{W}$ for $i = 1, 2, 3$. Let $\widetilde{H}'$ be the subgraph of $\widetilde{H}$ consisting of $\widetilde{W}$ and all paths P_i, $i = 1, 2, 3$.*

Then, there exist three vertex-disjoint paths P_1', P_2', P_3' in G' such that for every $i = 1, 2, 3$ there exists an integer $1 \leqslant k_i \leqslant \min(|V(P_i')|, |V(P_i)|)$ so that

- *the prefix of P_i' up to $\mathbf{u}_{k_i}(P_i')$ equals the prefix of P_i up to $\mathbf{u}_{k_i}(P_i)$; and*
- *the suffix of P_i' from $\mathbf{u}_{k_i}(P_i')$ is an induced path in $\widetilde{W}$ of length at least t.*

Proof: A *finishing touch* for a vertex v in $\widetilde{W}$ is a path defined as follows:

- if v is a peg of $\widetilde{H}$, then a finishing touch is a zero-length path consisting of v only;
- otherwise, if Q is the basic path of $\widetilde{W}$ containing v, then a finishing touch of v is a subpath of Q between v and one of its endpoints (which is always a peg).

Thus, a finishing touch connects v with a peg of $\widetilde{W}$, without any other peg on the way.

Consider the following modification step. Let $i \in \{1, 2, 3\}$ and assume P_i contains a vertex v such that the suffix of P_i starting in v is not a finishing touch for v, but there is a finishing touch of v whose intersection with $V(P_1) \cup V(P_2) \cup V(P_3)$ is contained in the

suffix of P_i starting at v. Then, modify P_i by replacing the suffix starting at v with the said finishing touch of v. Note that if we find the paths P_1', P_2', and P_3' for the modified paths, then they are also good for the original paths, as one can assume that $\mathbf{u}_{k_i}(P_i')$ is not later than v on P_i'.

As this modification either strictly decreases the number of edges of $\widetilde{H}'$ that are not in $\widetilde{W}$ or strictly decreases the number of pegs on the paths P_1, P_2, P_3 while keeping $\widetilde{H}'$ intact, without loss of generality we can assume that the modification step is not possible. This in particular implies that for $i = 1, 2, 3$, the only peg on P_i is $\overrightarrow{\mathbf{u}}(P_i)$.

Let Q be a basic path in $\widetilde{W}$. Assume that there is an internal vertex of Q which is on one of the paths P_i. We claim that either *both* endpoints of Q are endpoints of two out of three paths P_1, P_2, P_3, or the intersection of Q with the union of paths P_1, P_2, P_3 is a suffix of one of those paths being a finishing touch.

To this end, assume that an endpoint u of Q is not an endpoint of any of the paths P_i, $i = 1, 2, 3$. Since the endpoints are the only pegs on the paths P_i, $i = 1, 2, 3$, u does not lie on either of the paths P_i, $i = 1, 2, 3$. Let w be the closest to u vertex of Q that lies on one of the paths P_i, $i = 1, 2, 3$, and let $i \in \{1, 2, 3\}$ be such that w lies on P_i. By our assumption, w is an internal vertex of Q.

Note that the modification step is applicable and we could replace the suffix of P_i starting at w with the finishing touch being the subpath of Q from w to u. The only reason for this being an invalid modification is that actually the suffix of P_i starting from w is the second finishing touch of w, that is, the subpath of Q from w to the second endpoint. This proves the claim.

Since every peg in $\widetilde{W}$ has degree three, for every $i = 1, 2, 3$ we can find a basic path Q_i with one endpoint $\overrightarrow{\mathbf{u}}(P_i)$ and the second endpoint *not* being any of the endpoints of P_1, P_2, P_3. We have shown that the intersection of Q_i with the union of the paths P_1, P_2, P_3 is only a suffix of the path P_i (possibly of length 0). Denote this suffix by R_i.

For every $i = 1, 2, 3$, construct the path P'_i as follows. Define k_i so that $v_i := \mathbf{u}_{k_i}(P_i)$ is the starting vertex of R_i. If R_i is of length at least t, just pick $P'_i = P_i$. Otherwise, v_i is closer to $\overrightarrow{\mathbf{u}}(P_i)$ on Q_i than to the other endpoint of Q_i; obtain P'_i from P_i by replacing R_i with the subpath of Q_i from v_i to the second endpoint of Q_i, but without this endpoint (because it can be also an endpoint of Q_j for $j \neq i$). Since every basic path is of length at least $2t$, the obtained path P'_i is always of length at least t. This finishes the proof. ■

Now we get back to the context of the proof of Lemma 4.4.2, i.e., we work with an extended strip decomposition (H, η) of $G - v$. We make the following simple observation that we will later use multiple times to find $S_{t,t,t}$s within our reasoning.

Lemma 4.4.11. *Let $\widetilde{H}$ be a v-safe subgraph of H and let $\mathcal{P}$ be a family of vertex-disjoint paths in $\widetilde{H}$, each of length at least one. Then one can find a family $\mathcal{Q} = \{Q_P \mid P \in \mathcal{P}\}$ of disjoint anti-adjacent induced paths in $G - N[v]$ such that for every $P \in \mathcal{P}$, the path Q_P:*

- *is of length at least $|V(P)| - 2$;*
- *starts in a vertex of $\eta(\overleftarrow{\mathbf{u}}(P)\mathbf{u}_2(P), \overleftarrow{\mathbf{u}}(P))$;*
- *ends in a vertex of $\eta(\overrightarrow{\mathbf{u}}(P)\mathbf{u}_{|V(P)|-1}(P), \overrightarrow{\mathbf{u}}(P))$; and*
- *has all internal vertices contained in*

$$\bigcup_{e \in E(P)} \eta(e) - (\eta(\overleftarrow{\mathbf{u}}(P)\mathbf{u}_2(P), \overleftarrow{\mathbf{u}}(P)) \cup \eta(\overrightarrow{\mathbf{u}}(P)\mathbf{u}_{|V(P)|-1}(P), \overrightarrow{\mathbf{u}}(P))).$$

Proof: Fix $P \in \mathcal{P}$. Since $\widetilde{H}$ is v-safe, for every $1 \leqslant i < |V(P)|$ there exists a path $Q_{P,i}$ in $G[\eta(\mathbf{u}_i(P)\mathbf{u}_{i+1}(P))] - N[v]$ with endpoints in $\eta(\mathbf{u}_i(P)\mathbf{u}_{i+1}(P), \mathbf{u}_i(P))$ and in $\eta(\mathbf{u}_i(P)\mathbf{u}_{i+1}(P), \mathbf{u}_{i+1}(P))$ and all internal vertices in $\eta(\mathbf{u}_i(P)\mathbf{u}_{i+1}(P)) - (\eta(\mathbf{u}_i(P)\mathbf{u}_{i+1}(P), \mathbf{u}_i(P))$ $\cup \eta(\mathbf{u}_i(P)\mathbf{u}_{i+1}(P), \mathbf{u}_{i+1}(P)))$. By the properties of an extended strip decomposition, the

ending vertex of $Q_{P,i}$ is adjacent to the starting vertex of $Q_{P,i+1}$ for $1 \leqslant i < |V(P)| - 1$. Thus, the concatenation of those paths gives a path Q_P in $G - N[v]$ with the starting and ending vertices placed as desired and with at least $|V(P)| - 1$ vertices. Finally, the properties of an extended strip decomposition, together with the assumption that the paths $\mathcal{P}$ are vertex-disjoint imply that the paths $\{Q_P \mid P \in \mathcal{P}\}$ are induced, disjoint, and anti-adjacent. This completes the proof. ■

By the properties of an extended strip decomposition, every connected component of $G - v - \bigcup_{e \in E(H)} \eta(e)$ lies in a single set $\eta(x)$ for some $x \in V(H)$ or in a single set $\eta(xyz)$ for some $xyz \in T(H)$. We will be thinking of vertices that are reachable from v without visiting $\bigcup_{e \in E(H)} \eta(e)$ as vertices close to v in the following sense.

Definition 4.4.12 (projection)**.** The *projection of* v, denoted Π, is the set of those vertices $u \in \bigcup_{e \in E(H)} \eta(e)$ for which there exists a path P_u^{Π} with endpoints v and u and no internal vertex in $\bigcup_{e \in E(H)} \eta(e)$.

Note that P_u^{Π} is either a single edge, or a path whose all internal vertices lie in a single set $\eta(x)$ for some $x \in V(H)$, or in a single set $\eta(xyz)$ for some $xyz \in T(H)$. In particular, $\Pi \cap N[v] = \bigcup_{e \in E(H)} \eta(e) \cap N[v]$.

We need one more tool that will help us exhibit induced $S_{t,t,t}$s.

Lemma 4.4.13. *Let P be a path in H of length at least one with $x = \overleftarrow{\mathbf{u}}(P)$, $y = \mathbf{u}_2(P)$, and $\Pi \cap \eta(xy) \neq \emptyset$. Then there is a path Q in G that starts in v, ends in a vertex of $\eta(\overrightarrow{\mathbf{u}}(P)\mathbf{u}_{|V(P)|-1}(P), \overrightarrow{\mathbf{u}}(P))$, and has all internal vertices in*

$$\eta(x) \cup \eta(y) \cup \bigcup_{e \in E(P)} \eta(e) \cup \bigcup_{z \in V(H) \mid xyz \in T(H)} \eta(xyz).$$

Furthermore, if $\Pi \cap (\eta(xy) - \eta(xy, x)) \neq \emptyset$, then Q can be chosen with all internal vertices

in

$$\eta(y) \cup \bigcup_{e \in E(P)} \eta(e).$$

Proof: Pick $u \in \Pi \cap \eta(xy)$, preferably not in $\eta(xy, x)$ if possible. The path P_u^{Π} is either a direct edge, has all internal vertices in $\eta(y)$, all internal vertices in $\eta(x)$, or all internal vertices in $\eta(xyz)$ for some $z \in V(H)$ such that $xyz \in T(H)$. Furthermore, if $u \notin \eta(xy, x)$, then the last two options are impossible.

Since (H, η) is locally cleaned, in particular, the "moving a disconnected vertex of an interface" and "moving a disconnected component of an edge set" steps are inapplicable, there is a path Q_u in $\eta(xy)$ from u to a vertex $w \in \eta(xy, y)$ with all internal vertices not in $\eta(xy, x)$. By the properties of a locally cleaned extended strip decomposition, there is a path R_u from w to a vertex of $\eta(\overrightarrow{\mathbf{u}}(P)\mathbf{u}_{|V(P)|-1}(P), \overrightarrow{\mathbf{u}}(P))$ via $\bigcup_{e \in E(P)-\{xy\}} \eta(e)$. By concatenating P_u^{Π}, Q_u, and R_u, and possibly shortcuting it to an induced path we obtain the desired path Q. ■

We are ready to find our first $S_{t,t,t}$ of the proof.

Lemma 4.4.14. *For every constant t_1 there is a constant t_2 such that if W is a v-safe wall in H of sidelength at least t_2, then either G contains an induced $S_{t,t,t}$ or there exists a subwall W' of W of sidelength t_1 such that for every $e \in E(H)$ with $\Pi \cap \eta(e) \neq \emptyset$, at most one endpoint of e lies in W'.*

Proof: Consider the natural plane embedding of the wall W. We say that two vertices x, y of W are *radially close* if there exists a set F of at most $2t+4$ bounded faces of W such that the subgraph of W consisting of all vertices and edges lying on a face of F is connected and contains both x and y.

Consider the following process. First, all vertices of W are unmarked. As long as there exists an edge $xy \in E(H)$ with $\eta(xy) \cap \Pi \neq \emptyset$, $x, y \in V(W)$, and both x and y

unmarked, mark all vertices of W that are radially close to either x or y. Note that this in particular marks x and y.

Assume first that this process iterated for at least three steps; let x_1y_1, x_2y_2, and x_3y_3 be the three edges of H chosen in the first three steps. The *radially close* definition implies that W contains vertex-disjoint paths P_1, P_2, and P_3, each of length $t+1$ so that P_i starts in x_i for $i = 1, 2, 3$ and neither of the paths contains any of the vertices y_1, y_2, y_3.

For $i = 1, 2, 3$, proceed as follows. Apply Lemma 4.4.13 to a path consisting of the edge x_iy_i only, obtaining a path Q_i from v to a vertex $u_i \in \eta(x_iy_i, x_i)$ with all internal vertices in $\eta(x_i) \cup \eta(y_i) \cup (\eta(x_iy_i) - \eta(x_iy_i, x_i)) \cup \bigcup_{z \in V(H) \mid x_iy_iz \in T(H)} \eta(x_iy_iz)$. Since the vertices $x_1, y_1, x_2, y_2, x_3, y_3$ are pairwise distinct, the paths $\{Q_i - \{v\} \mid 1 \leqslant i \leqslant 3\}$ are anti-adjacent. Hence, $V(Q_1) \cup V(Q_2) \cup V(Q_3)$ induce a tree with three leaves u_1, u_2, u_3 and v being the unique vertex of degree 3. We now extend this tree with paths Q_{P_i} for $i = 1, 2, 3$, obtained from paths P_i, $i = 1, 2, 3$, using Lemma 4.4.11 for $\widetilde{H} = W$. This gives the desired $S_{t,t,t}$.

We are left with the case when the aforementioned process iterated for at most two steps. Then, if $t_2 > t_1 + 100(t+1)$, W contains a subwall W' of sidelength t_1 consisting of unmarked vertices only. This subwall satisfies the conditions of the lemma. ■

A wall W' in H is *v-pure* if it is v-safe, it is $2t$-subdivided, and for every $e \in E(H)$ with $\eta(e) \cap \Pi \neq \emptyset$, at most one endpoint of e lies in W'.

The following statement follows directly from Lemma 4.4.14 and the fact that by leaving only every $(2t+1)$-th column and row of a wall, we can extract a $2t$-subdivided subwall.

Lemma 4.4.15. *For every constant t_1 there exists a constant t_2 such that if W is a v-safe wall in H of sidelength at least t_2, then either G contains an induced $S_{t,t,t}$ or W contains a v-pure subwall W' of sidelength t_1.*

4.4.5 The case of Π being well-connected to a v-pure wall

Lemma 4.4.15 allows us to find a large v-pure wall W in H. We now observe that if there is a substantial connection between edges of H containing elements of Π in their sets and W, then G admits an induced $S_{t,t,t}$.

Lemma 4.4.16 (Claw rooted at v without a small cut)**.** *Let W be a v-pure wall in H of sidelength at least 3. Assume that H contains three vertex-disjoint paths P_1, P_2, and P_3 such that for every $i = 1, 2, 3$, the first edge e_i of P_i is such that $\eta(e_i) \cap \Pi \neq \emptyset$ and the ending vertex of P_i is a peg of W. Then G contains an induced $S_{t,t,t}$.*

Proof: Apply Lemma 4.4.10 to P_1, P_2, P_3, and W, obtaining paths P_1', P_2', P_3'. For every $i = 1, 2, 3$ define the path P_i'' as follows. If e_i is the first edge of P_i' (i.e., $k_i > 1$), then keep $P_i'' \coloneqq P_i'$. The other case is only possible if $e_i = x_i y_i$ with $x_i \in V(W)$, $y_i \notin V(W)$, $x_i = \overleftarrow{\mathbf{u}}(P_i)$, and the path P_i' is fully contained in W. Note that, because P_j and P_j' differ only inside W, the vertex x_i is not used by any other path P_j'. Define P_i'' to be the path P_i' prepended with the edge e_i (so now $\overleftarrow{\mathbf{u}}(P_i'') = y_i$). (Note that e_i cannot be an edge of W as $y_i \notin V(W)$.) In this manner, $(P_i'')_{i=1,2,3}$ are vertex-disjoint, each starts with e_i and contains a suffix of length at least t contained in the wall W.

Split each P_i'' into the said suffix R_i of length t and the remaining prefix Q_i. Lemma 4.4.11 applied to $\{R_i \mid 1 \leqslant i \leqslant 3\}$ and $\widetilde{H} = W$ gives pairwise disjoint, anti-adjacent, induced paths R_i' on t vertices each. Lemma 4.4.13 applied to Q_i gives a path Q_i' from v to a vertex of $\eta(\overrightarrow{\mathbf{u}}(Q_i)\mathbf{u}_{|V(Q_i)|-1}(Q_i), \overrightarrow{\mathbf{u}}(Q_i))$. Because the paths P_i'' for $i = 1, 2, 3$ are vertex-disjoint, the paths $Q_i' - \{v\}$ for $i = 1, 2, 3$ are anti-adjacent. Then, $Q_1' \cup Q_2' \cup Q_3' \cup R_1' \cup R_2' \cup R_3'$ contains an induced $S_{t,t,t}$ with the center in v. ■

Lemma 4.4.17 (Claw in wall rooted at v where paths in H start at a single vertex)**.** *Let W be a v-pure wall of sidelength at least 3. Assume that H contains three vertex-disjoint paths P_1, P_2, P_3 and a vertex $x \in V(H)$ such that for every $i = 1, 2, 3$ the ending vertex*

of P_i is a peg of W while $x\overleftarrow{\mathbf{u}}(P_i) \in E(H)$ and $\Pi \cap (\eta(x\overleftarrow{\mathbf{u}}(P_i)) - \eta(x\overleftarrow{\mathbf{u}}(P_i), x)) \neq \emptyset$. Then, G contains an induced $S_{t,t,t}$.

Proof: We start by applying Lemma 4.4.10 to P_1, P_2, and P_3, obtaining paths P_1', P_2', and P_3' and indices k_1, k_2, and k_3. We remark that x may appear on one of the paths P_i' and even one of those paths can start with the edge $x\overleftarrow{\mathbf{u}}(P_i)$.

Fix $i \in \{1,2,3\}$. Denote $y_i = \overleftarrow{\mathbf{u}}(P_i) = \overleftarrow{\mathbf{u}}(P_i')$. Lemma 4.4.13, applied to the single edge xy_i gives a path Q_i from v to $v_i \in \eta(y_i x, y_i)$ with all internal vertices in $\eta(y_i) \cup (\eta(xy_i) - (\eta(xy_i, x) \cup \eta(xy_i, y_i)))$. Observe that because vertices y_i are pairwise distinct, the paths $Q_i - \{v\}$ are pairwise disjoint and anti-adjacent for $i = 1, 2, 3$.

If $k_i > 1$, extend Q_i from v_i to a vertex of $\eta(\mathbf{u}_{k_i}(P_i')\mathbf{u}_{k_i-1}(P_i'), \mathbf{u}_{k_i}(P_i'))$ via sets $\eta(e)$ for e lying on the prefix of P_i' till $\mathbf{u}_{k_i}(P_i')$, and shorten the walk to an induced path in the end. Let Q_i' be the resulting path; set $Q_i' = Q_i$ if $k_i = 1$ and observe that if $k_i = 1$ then the first edge of P_i' is not xy_i as $xy_i \notin E(W)$ due to W being v-pure. To obtain the desired $S_{t,t,t}$ centered at v, extend every path Q_i' with a path obtained from Lemma 4.4.11 applied to the suffix of P_i' from $\mathbf{u}_{k_i}(P_i')$ and $\widetilde{H} = W$. ■

We will use Lemmas 4.4.16 and 4.4.17 to find a separation that separates W from Π. The precise meaning of "separate" is encapsulated in the following statement.

Definition 4.4.18 (capturing a projection)**.** Let (A, B) be a separation in H and let $Z \subseteq A \cap B$. We say that (A, B) *captures* Π *with backdoor set* Z if for every $e \in E(H)$ with $\Pi \cap \eta(e) \neq \emptyset$, either $e \subseteq A$ or there is an endpoint $x \in e$ such that $x \in Z$ and $\Pi \cap \eta(e) \subseteq \eta(e, x)$.

Lemma 4.4.19. *Let W be a v-pure wall in H of sidelength at least 55. Then either G contains an induced $S_{t,t,t}$ or there exists a separation $(A, B) \in \mathcal{T}_W$ of order less than 19 that captures Π with a backdoor set of size at most 2.*

Proof: Let

$$X_1 = \{x \in V(H) \mid \exists_{e_x = xy_x \in E(H)} \Pi \cap (\eta(e_x) - \eta(e_x, y_x)) \neq \emptyset\}.$$

For each $x \in X_1$ fix one edge e_x (and thus also the endpoint y_x) as in the definition above. Let

$$F = \{e = xy \in E(H) \mid x, y \notin X_1 \text{ and } \Pi \cap \eta(e, x) \cap \eta(e, y) \neq \emptyset\}.$$

Let F_2 be a maximal matching in F and let $X_2 \coloneqq V(F_2)$. For an edge $e = xy \in F_2$ we denote $e_x = e_y = e$. Furthermore, for every $x \in X_2$, if xy is the unique edge of F_2 containing x, then we denote $y_x = y$. Let $X \coloneqq X_1 \cup X_2$. Note that e_x and y_x has been defined for all $x \in X$. Observe that, by the definition of X,

$$\forall_{e \in E(H)} (\eta(e) \cap \Pi \neq \emptyset) \implies (e \subseteq X) \text{ or } (e \cap X = \{x\} \text{ and } \eta(e) \cap \Pi \subseteq \eta(e, x)). \quad (4.6)$$

Assume first that there exists a family $\mathcal{P}$ of 17 vertex-disjoint paths in H with starting points in X and ending points in pegs of W. Let $X' \subseteq X$ be the set of the starting points of the paths in $\mathcal{P}$ and for $x \in X'$ let $P_x \in \mathcal{P}$ be the path starting at x. We say that $x \in X'$ *kills* $x' \in X' - \{x\}$ if y_x lies on $P_{x'}$. Observe that x kills at most one $x' \in X'$ as paths in $\mathcal{P}$ are pairwise vertex-disjoint.

Consider an auxiliary bipartite graph K with sides being two copies of X' and an edge $(x, x') \in E(K)$ if x kills x'. We consider two subcases. In the first subcase, K has a matching of size 9. Then, K contains three edges (x_1, x_1'), (x_2, x_2'), and (x_3, x_3') such that $x_1, x_1', x_2, x_2', x_3, x_3'$ are six pairwise distinct vertices of X'. Then, the paths P_{x_i} prepended with the edge e_{x_i} are vertex-disjoint and satisfy the requirements of Lemma 4.4.16, giving an induced $S_{t,t,t}$ in G. In the other case, K has a vertex cover of size at most 8. By

deleting these vertices from X', we obtain a subset $X'' \subseteq X'$ of size 9 where no vertex kills another one.

Consider now a graph L on the vertex set X'' where $xx' \in E(L)$ if $y_x = y_{x'}$. By Ramsey's Theorem, L has an independent set of size 3 or a clique of size 4. If I is an independent set of size 3 in L, then for every $x \in I$ obtain a path P'_x as follows: if y_x does not lie on P_x, set P'_x to be P_x prepended with the edge xy_x, and otherwise set P'_x to be the edge xy_x with the suffix of P_x starting in y_x; note that $\{P'_x \mid x \in I\}$ satisfy the requirements of Lemma 4.4.16. If I is a clique of size 4 in L, denote by z the vertex that is equal to y_x for every $x \in I$. Note that because F_2 is a matching, at most one element of I is in X_2. Hence, z and $\{P_x \mid x \in I - X_2\}$ satisfy the requirements of Lemma 4.4.17. This finishes the case where the family of paths $\mathcal{P}$ exists.

In the other case, by Menger's theorem, there is a separation (A, B) in H of order less than 17 such that $X \subseteq A$ but all pegs of W lie in B. By (4.6), (A, B) captures Π, but the set of backdoors can be as large as $A \cap B$. Our goal is now to modify (A, B) a bit to restrict the set of backdoors. To this end, consider a subgraph H' of H with $V(H') = B$ and $e \in E(H')$ if $e \subseteq B - A$ or $|e \cap A \cap B| = 1$ and $\eta(e) \cap \Pi \neq \emptyset$.

We consider two subcases. In the first subcase, H' contains a family $\mathcal{Q}$ of three vertex-disjoint paths from $A \cap B$ to the set of pegs of W that are in $B - A$. By the construction of H', each path $Q \in \mathcal{Q}$ starts with a vertex $x \in A \cap B$ and an edge $e_x = xy_x$, $y_x \in B - A$, and $\Pi \cap \eta(e_x) \neq \emptyset$. Then, Lemma 4.4.16 applied to $\mathcal{Q}$ yields an induced $S_{t,t,t}$ in G.

In the second subcase, there is a separation (A', B') in H' of order less than 3 such that $A \cap B \subseteq A'$ while all pegs of W that lie in $B - A$ belong to B'. Let $A'' \coloneqq A \cup A'$ and $B'' \coloneqq B' \cup (A \cap B)$. Then, (A'', B'') is a separation in H with $X \subseteq A''$ and all pegs of W lying in B''. Furthermore, $A'' \cap B'' = (A' \cap B') \cup (A \cap B)$. In particular, the order of (A'', B'') is less than 19 and hence $(A'', B'') \in \mathcal{T}_W$ as W has sidelength at least 55 (so $\mathcal{T}_W$ has order at least 19) and all pegs of W lie in B''.

Consider now $z \in A'' \cap B''$ such that there exists an edge $zy \in E(H)$ with $y \in B'' - A''$ and $\Pi \cap \eta(zy) \neq \emptyset$. By (4.6), $\{y, z\} \cap X \neq \emptyset$. However, as $X \subseteq A$ while $y \notin A$, we have $z \in X$, $z \in A$, and, by (4.6) again, $\Pi \cap \eta(zy) \subseteq \eta(zy, z)$. The edge zy belongs to H'. Since $y \in B'' - A'' \subseteq B' - A'$, we have $z \in B'$. Since $z \in A \cap B \subseteq A'$, we have $z \in A' \cap B'$.

Hence, $(A'', B'') \in \mathcal{T}_W$ is a separation of order less than 19 that captures Π with backdoor set $Z \subseteq A' \cap B'$, which is of size at most 2. This finishes the proof. ■

4.4.6 Cleaning the backdoors

Lemma 4.4.19 allows us to find a separation of small order that captures Π with at most two backdoors. Our goal in this section is to further clean the situation with regards to how exactly the neighbors of v can appear around $\eta(z)$ and $\eta(zx, x)$ for a backdoor vertex z. On the way there, we will need to sacrifice small parts of the wall W defining the tangle or slightly increase the size of the allowed separation.

We start with the following straightforward observation from the definition of capturing Π and the fact that (H, η) is locally cleaned.

Lemma 4.4.20. *Let (A, B) be a separation in H that captures Π with backdoor set Z. Suppose $xyz \in T(H)$ is such that $\eta(xyz) \cap N(v) \neq \emptyset$. Then either $x, y, z \in A$, or two of the vertices x, y, z belong to Z and the third one is in $B - A$.*

We need a few definitions. Let (A, B) be a separation in H that captures Π with backdoor set Z. For $z \in Z$, let $\mathfrak{P}_z := \bigcup_{x \in N_H(z) \cap (B-A)} \eta(zx, z)$. We say that (A, B) is *triangle-safe* if for every triangle $T = xyz \in T(H)$ with $\eta(T) \cap N(v) \neq \emptyset$ either $x, y, z \in A$ or, assuming without loss of generality $y, z \in A \cap B$ and $x \in B - A$, we have that v is complete to both $\eta(xy, y)$ and $\eta(xz, z)$.

Assume (A, B) is triangle-safe. Fix a backdoor vertex $z \in Z$. Observe that for every $u \in \Pi \cap \mathfrak{P}_z$, the path P_u^{Π} is either a direct edge or goes via $\eta(z)$: it cannot go through

$\eta(xyz)$ for a triangle with say $x \in B-A$ as, thanks to triangle-safeness, v is then complete to $\eta(xz, z)$, where u resides. An *entry point* is a vertex $v' \in \{v\} \cup \eta(z)$ that has a neighbor in $\mathfrak{P}_z$ and admits a path $Q^{\Pi}_{v'}$ from v to v' (it can be of zero length when $v = v'$) whose internal vertices have no neighbors in $\mathfrak{P}_z$. The backdoor z is *pure* if every entry point is complete to $\mathfrak{P}_z$.

In the next two lemmas we first ensure that we have a separation in H that captures Π and is triangle-safe, and then we ensure that all backdoors are pure.

Lemma 4.4.21 (triangle cleanup)**.** *Let W be a v-pure wall in G and let $(A, B) \in \mathcal{T}_W$ be a separation capturing Π with a set of backdoors Z of size at most 2 such that the sidelength of W is $k_W \geqslant 4|A \cap B| + 7$. Then, either G admits an induced $S_{t,t,t}$ or there exists a subwall W' of W of sidelength at least $k_W - |A \cap B|$ and a separation $(A', B') \in \mathcal{T}_{W'}$ in H of order at most $|A \cap B| + 2$ that captures Π with a backdoor set of size at most 2, $A \subseteq A'$, $B' \subseteq B$, and is triangle-safe.*

Proof: If (A, B) is already triangle-safe, we can return $W = W'$ and $(A', B') = (A, B)$, so assume otherwise. By Lemma 4.4.20, this is only possible if $|Z| = 2$, say $Z = \{z_1, z_2\}$ and there is a triangle $z_1z_2x \in T(H)$ with $\eta(xyz) \cap N(v) \neq \emptyset$ and $x \in B-A$, but v is not complete to $\eta(zx, z)$ for some $z \in \{z_1, z_2\}$. Let us call such a triangle a *violating triangle*. Let X be the set of those $x \in B - A$ for which z_1z_2x is a violating triangle.

Since $(A, B) \in \mathcal{T}_W$, $B - A$ contains $k_W - |A \cap B|$ full rows and $k_W - |A \cap B|$ full columns of W; let W' be a subwall of W completely contained in $B - A$ of sidelength at least $k_W - |A \cap B|$.

We consider two cases. In the first case, there are three vertex-disjoint paths P_1, P_2, P_3 from X to the pegs of W' in the graph $H[B - A]$. Let $x_i = \overleftarrow{\mathbf{u}}(P_i)$ for $i = 1, 2, 3$. In this case, we exhibit an induced $S_{t,t,t}$ in G.

To this end, apply Lemma 4.4.10 to P_1, P_2, P_3, and W', obtaining paths P_1', P_2', and P_3'. We note that as all paths P_1, P_2, and P_3, as well as the wall W' are in $H[B-A]$, the paths P_1', P_2', and P_3' are also contained in $H[B-A]$; in particular, they do not contain vertices z_1 nor z_2. Since (A,B) captures Π, $H[B-A]$ is v-safe. Thus, Lemma 4.4.11 applied to $\{P_1', P_2', P_3'\}$ in $H[B-A]$ yields disjoint anti-adjacent induced paths R_1, R_2, R_3.

For $i = 1, 2, 3$, let C_i be a component of $G[\eta(x_i z_1 z_2)]$ that contains a neighbor of v. Since z_1 and z_2 are symmetric so far, as $z_1 z_2 x_2$ is a violating triangle, we can assume that v is not complete to $\eta(z_1 x_2, z_1)$; pick $y_2 \in \eta(z_1 x_2, z_1)$ nonadjacent to v. Since (H, η) is locally cleaned, C_1 has a neighbor $y_1 \in \eta(z_1 x_1, z_1) \cap \eta(z_1 x_1, x_1)$ and C_3 has a neighbor $y_3 \in \eta(z_2 x_3, z_2) \cap \eta(z_2 x_3, x_3)$. (We emphasize the intended lack of symmetry in the choice of z_1 vs z_2 in the last two sentences.) By the properties of an extended strip decomposition, $y_1 y_2 \in E(G)$, $y_1 y_3, y_2 y_3 \notin E(G)$, and C_1 and C_3 are anti-adjacent to y_2. Let Q be a shortest path from y_1 to y_3 via C_1, v, and C_3 (note that it may go via direct edges vy_1 or vy_3 if they exist). Let R_2' be a shortest path from y_2 to a vertex of $\eta(z_1 x_2, x_2)$ with all internal vertices in $\eta(z_1 x_2) - (\eta(z_1 x_2, z_1) \cup \eta(z_1 x_2, x_2))$. By appending R_1 to y_1, R_2' and R_2 to y_2, and R_3 to y_3 and connecting y_1 and y_3 via Q, we obtain an induced $S_{t,t,t}$ in G with center in y_1.

In the second case, there is a separation (A_1, B_1) in $H[B-A]$ of order less than 3 with $X \subseteq A_1$ and all pegs of W' lying in B_1. Define $A' = A \cup A_1$ and $B' = B_1 \cup (A \cap B)$. Clearly, (A', B') is a separation of order at most $|A \cap B| + 2$ in H with $A \subseteq A'$ and $B' \subseteq B$. Since the sidelength of W' is at least $3|A \cap B| + 7$ and all pegs of W' lie in B', we have $(A', B') \in \mathcal{T}_{W'}$. Since $A \subseteq A'$, any backdoor vertex of (A', B') is also a backdoor vertex of (A, B), and thus is in the set Z of size 2. Finally, since $X \subseteq A_1$, for every $xyz \in T(H)$ with $\eta(xyz) \cap N(v) \neq \emptyset$ we have $x, y, z \in A'$. Thus, the wall W' and the separation (A', B') is the desired outcome. ■

Lemma 4.4.22 (backdoor cleanup)**.** *Let W be a v-pure wall in G and let $(A, B) \in \mathcal{T}_W$ be a separation capturing Π with a set of backdoors Z of size at most 2 that is triangle-safe and such that the sidelength of W is $k_W \geqslant 4|A \cap B| + 7$.*

Suppose there exists a backdoor vertex $z \in Z$ that is not pure. Then, one of the following holds:

- *G admits an induced $S_{t,t,t}$;*
- *H admits a separation $(A_0, B_0) \in \mathcal{T}_W$ of order 1 that captures Π with backdoor set $A \cap B$;*
- *H admits a subwall W' of W of sidelength at least $k_W - |A \cap B|$ and a separation $(A', B') \in \mathcal{T}_{W'}$ of order at most $|A \cap B| + 2$ with $A \subseteq A'$ and $B' \subseteq B$ capturing Π with backdoor set contained in $Z - \{z\}$; or*
- *H admits a separation $(A^*, B^*) \in \mathcal{T}_W$ of order at most $|A \cap B|$ with $A \subseteq A^*$, $B^* \subseteq B$, and $(A, B) \neq (A^*, B^*)$.*

Proof: Recall that for $z \in A \cap B$, $\mathfrak{P}_z = \bigcup_{x \in N_H(z) \cap (B-A)} \eta(zx, z)$. Fix an entry point $v' \in \{v\} \cup \eta(z)$ that causes z not to be pure: v' has some neighbors in $\mathfrak{P}_z$, but is not complete to $\mathfrak{P}_z$.

Since $(A, B) \in \mathcal{T}_W$ while the sidelength of W is at least $4|A \cap B| + 7$, we define W' to be a subwall of W of sidelength at least $k_W - |A \cap B| \geqslant 3|A \cap B| + 7$ that is fully contained in $H[B - A]$.

We define X_1 to be the set of those vertices $x \in A$ for which either $xz \in E(H)$ and $\Pi \cap (\eta(xz) - \eta(xz, z)) \neq \emptyset$, or there exists a neighbor $y \in N_H(x) \cap (A - \{z\})$ with $\Pi \cap \eta(xy) \neq \emptyset$. Let $X_\perp$ be the set of those vertices $y \in B - A$ for which there exists $x_y \in X_1$ and a path P_y in H from x_y to y that avoids z and whose only vertex outside A is y; denote the last edge of P_y by e_y and the penultimate vertex of P_y by z_y (so that

$e_y = z_y y$). Note that Lemma 4.4.13 implies that for every $y \in X_\perp$ there exists an induced path R_y from v to a vertex $\eta(e_y, y)$ whose all internal vertices belong to

$$(\eta(e_y) - \eta(e_y, y)) \cup \bigcup_{x \in N_H(z) \cap A} (\eta(zx) - \eta(zx, x)) \cup \bigcup_{e \in E(H[A - \{z\}])} \eta(e) \cup \bigcup_{x \in A - \{z\}} \eta(x) \cup \bigcup_{T \in T(H[A])} \eta(T). \tag{4.7}$$

We construct an auxiliary graph H_1 as follows. Start with $H_1 \coloneqq H[B - A]$. Add all vertices of $\mathfrak{P}_z$ to H_1 and for a vertex $u \in \mathfrak{P}_z$, if $u \in \eta(zy, z)$ for $y \in B - A$, make u adjacent to y in H_1. Add three new vertices $a_\circ$, $a_\bullet$, and $a_\perp$. Make $a_\perp$ adjacent to all vertices of $X_\perp$. Make $a_\circ$ adjacent to all vertices of $\mathfrak{P}_z$ that are nonadjacent to v' in G and make $a_\bullet$ adjacent to all vertices of $\mathfrak{P}_z$ that are adjacent to v' in G.

We consider two cases. In the first case, there are three vertex-disjoint paths $P_\circ$, $P_\bullet$, and $P_\perp$ in H_1 from $a_\circ$, $a_\bullet$, and $a_\perp$, respectively, to the set of pegs of W'. In this case, we will exhibit an induced $S_{t,t,t}$ in G.

Since every vertex of $\mathfrak{P}_z$ is of degree 2 in H_1, the path $P_\circ$ starts in $a_\circ$, continues via $u_\circ \in \eta(zx_\circ, z)$ to $x_\circ \in B - A$ and then stays in $B - A$ till the end. Similarly, the path $P_\bullet$ starts in $a_\bullet$, continues via $u_\bullet \in \eta(zx_\bullet, z)$ to $x_\bullet \in B - A$ and then stays in $B - A$ till the end. By construction, $v'u_\circ \notin E(G)$ while $v'u_\bullet \in E(G)$. Since $P_\circ$ and $P_\bullet$ are vertex-disjoint, $x_\circ \neq x_\bullet$ and thus $u_\circ u_\bullet \in E(G)$ by the properties of an extended strip decomposition. Let $a_\perp x_\perp$ be the first edge of $P_\perp$ for some $x_\perp \in X_\perp$.

Apply Lemma 4.4.10 to $P_\circ$, $P_\bullet$, $P_\perp$, and the wall W', obtaining paths $P'_\circ$, $P'_\bullet$, and $P'_\perp$. Note that $a_\circ u_\circ$ and $u_\circ x_\circ$ remain the first two edges of $P'_\circ$, $a_\bullet u_\bullet$ and $u_\bullet x_\bullet$ remain the first two edges of $P'_\bullet$, and $a_\perp x_\perp$ remains the first edge of $P'_\perp$.

The suffixes of paths $P'_\circ$, $P'_\bullet$, and $P'_\perp$ from $x_\circ$, $x_\bullet$, and $x_\perp$, respectively, stay in $H[B-A]$, which is v-safe due to the assumption that (A, B) captures Π. Apply Lemma 4.4.11 to these three suffixes and $H[B-A]$, obtaining disjoint anti-adjacent induced paths $R_\circ$, $R_\bullet$, and $R_\perp$.

We now exhibit an induced $S_{t,t,t}$ in G with the center in $u_\bullet$. For the first leg, as (H, η) is locally cleaned (in particular, the "moving a disconnected vertex of an interface" and "moving a disconnected component of an edge set" operations are inapplicable), there is a path in $\eta(zx_\bullet)$ (possibly of zero length) from $u_\bullet$ to a vertex of $\eta(zx_\bullet, x_\bullet)$ whose only vertex of $\eta(zx_\bullet, z)$ is $u_\bullet$; concatenate this path with $R_\bullet$. For the second leg, start with an edge $u_\bullet u_\circ$, continue via an analogous path in $\eta(zx_\circ)$ from $u_\circ$ to a vertex of $\eta(zx_\circ, x_\circ)$ whose only vertex of $\eta(zx_\circ, z)$ is $u_\circ$ and append $R_\circ$ at the end. For the third leg, go with a shortest path from $u_\bullet$ to the starting vertex of $R_\perp$ via v', $Q^\Pi_{v'}$, v, $R_{x_\perp}$, and finish with $R_\perp$. Condition (4.7) ensures that this is indeed an induced $S_{t,t,t}$ in G.

We are left with the second case where H_1 contains a separation (A_1, B_1) of order less than three with $a_\circ, a_\bullet, a_\perp \in A_1$ but all pegs of W' in B_1. Pick such (A_1, B_1) with B_1 inclusion-wise minimal.

Observe that no vertex of $\mathfrak{P}_z$ is in $A_1 \cap B_1$, as if some $u \in A_1 \cap B_1$ with $u \in \eta(zx, z)$, $x \in B - A$, then as $a_\circ, a_\bullet \in A_1$, $(A_1 \cup \{x\}, B_1 - \{u\})$ is also a separation of order less than 3, contradicting the choice of (A_1, B_1). Let B' consist of the pegs of W' and $N_H[(B - A) \cap (B_1 - A_1)]$ and $A' = V(H) - ((B - A) \cap (B_1 - A_1))$. Clearly, (A', B') is a separation in H. Observe that $A' \cap B' \subseteq (A \cap B) \cup (A_1 \cap B_1)$, and thus the order of (A', B') is at most $|A \cap B| + |A_1 \cap B_1| \leqslant |A \cap B| + 2$. As all pegs of W' are in B', $(A', B') \in \mathcal{T}_{W'} \subseteq \mathcal{T}_W$. We have $B' \subseteq B$ and thus $A \subseteq A'$, and hence (A', B') is triangle-safe and captures Π with a backdoor set being a subset of Z.

If there is no path in $H[B - A]$ from $X_\perp$ to a peg of W', then define a separation as follows. Let B_0 consist of z and all vertices of H reachable in $H - \{z\}$ from a peg of W', and let $A_0 = (V(H) - B_0) \cup \{z\}$. Clearly, (A_0, B_0) is a separation of order 1 in H. Since all pegs of W' lie in B_0, we have $(A_0, B_0) \in \mathcal{T}_{W'} \subseteq \mathcal{T}_W$. By the assumption, $X_\perp \subseteq A_0$. Hence, (A_0, B_0) is Π-capturing. This gives the second outcome of the lemma.

If for some $a \in \{a_\circ, a_\bullet\}$ there is no path in H_1 from a to a peg of W', define a

separation (A^*, B^*) as follow. Let C be all vertices of $B - A$ reachable from a in H_1. Since v' has at least one neighbor in $\mathfrak{P}_z$, but is not complete to $\mathfrak{P}_z$, we have that a is not an isolated vertex of H_1 and thus $C \neq \emptyset$. Let $A^* = A \cup C$ and $B^* = B - C$. Then, (A^*, B^*) is also a separation in H with $A \cap B = A^* \cap B^*$ and all pegs in W' lying in B^* (thus $(A^*, B^*) \in \mathcal{T}_{W'} \subseteq \mathcal{T}_W$) but with $A \subsetneq A^*$, $B^* \subsetneq B$. This gives the last outcome of the lemma.

If z is not a backdoor of (A', B'), then we are done with the penultimate outcome. Otherwise, there exists $x \in B' - A'$ with $\Pi \cap \eta(zx, z) \neq \emptyset$. We have $x \in B_1 - A_1$ and thus either $a_\circ$ or $a_\bullet$ belongs to $A_1 \cap B_1$. Since $|A_1 \cap B_1| < 3$ and for every $a \in \{a_\circ, a_\bullet, a_\perp\}$ there is a path from a to a peg of W' in H_1, $A_1 \cap B_1$ is of size 2 and consists of $a \in \{a_\circ, a_\bullet\}$ and a vertex $z' \in B - A$. In particular, $a_\perp \in A_1 - B_1$ hence $X_\perp \subseteq A_1$. Observe that for every $y \in X_\perp$, we have $N_H(z_y) \cap (B - A) \subseteq X_\perp$. Hence, for every $y \in X_\perp$, the vertex z_y lies in $A' - B'$. As there is a path from $X_\perp$ to a peg of W' in $H[B - A]$, in particular we have $X_\perp \neq \emptyset$, so at least one z_y in $(A' - B') \cap (A \cap B)$ exists. As $A_1 \cap B_1 = \{a, z'\}$, we have $A' \cap B' \subseteq (A \cap B) \cup \{z'\}$. As $(A' - B') \cap (A \cap B) \neq \emptyset$, we have $|A' \cap B'| \leqslant |A \cap B|$ and $(A', B') \neq (A, B)$. This proves that (A', B') satisfies the properties of the last outcome of the lemma. ■

We conclude this section with a lemma summarizing what we obtained so far.

Lemma 4.4.23. *For every constant t, there exists a constant c_t and a polynomial-time algorithm that, given input as in Lemma 4.4.2, either returns one of the promised outputs of Lemma 4.4.2 or outputs all the following objects:*

- *a locally clean extended strip decomposition (H, η) of $G - v$ with every vertex of degree at least 2 and every particle of weight at most 0.01τ,*
- *a separation (A, B) in H of order at most c_t that is triangle-safe, captures Π with backdoor set Z of size at most 2 such that every $z \in Z$ is pure, and satisfies*

$\mathfrak{w}(\overleftarrow{\eta}_{(H,\eta)}(B-A)) \geqslant 0.99\tau$, *and*

- *for every* $X \subseteq A \cap B$ *of size three, a family* $\mathcal{Q}^X = \{Q^X_x \mid x \in X\}$ *of three disjoint anti-adjacent induced paths in* G *on* t *vertices each, where for every* $x \in X$ *there exists* $y_x \in N_H(x) \cap (B-A)$ *such that the path* Q^X_x *starts in a vertex of* $\eta(xy_x, x)$ *and has all remaining vertices in* $\bigcup_{e \in E(G[B-A])} \eta(e)$ *or in* $\eta(xy_x) - \eta(xy_x, x)$.

Proof: We compute (H, η) as discussed in Section 4.4.2, by exhaustively applying the local cleaning operation to the input extended strip decomposition of $G-v$ and discarding isolated vertices of H with empty vertex sets. Either a 0.99τ-balanced separator dominated by a constant number of vertices is returned, or (H, η) satisfies properties (4.2), (4.3), and (4.4).

We fix σ to be large enough constant depending on t, emerging from the proof. We start searching for W and (A, B) by applying Lemma 4.4.9. Thus, we can either already conclude, or get a v-safe wall W of sidelength 3σ satisfying (4.5).

We proceed with W with a series of lemmas that each either concludes the reasoning, or exhibit a subwall W' of W with some additional property. The new subwall will be large in the following sense: if for every constant σ' there exists a constant σ such that if W is of sidelength at least σ, then W' is guaranteed to be of sidelength at least σ'. After each step, we rename W' back to W. For the sake of clarity, we will not follow the exact computation of the dependencies of σ on σ', but only refer to them as a "decrease" or "losing on the sidelength".

We start with applying Lemma 4.4.15 to W; by losing a bit on the sidelength of W, we can assume that W is actually v-pure. We then apply Lemma 4.4.19. This yields either an induced $S_{t,t,t}$ in G (which we can return) or a separation $(A, B) \in \mathcal{T}_W$ of order less than 19 that captures Π and has backdoor set of size at most 2. We pass it to Lemma 4.4.21 that either finds $S_{t,t,t}$ or, at the cost of a slight decrease (of at most

$|A \cap B|$) of the sidelength of W and an increase of the order of (A, B) by at most 2, upgrades (A, B) to be triangle-safe. (Here, we slightly abuse the notation by denoting the output of Lemma 4.4.21 by W and (A, B), again.)

We now iterate on improving W and (A, B) using Lemma 4.4.22. While (A, B) has a backdoor vertex z that is not pure, apply Lemma 4.4.22 to W, (A, B), and z. If the third outcome happens, replace W with W' and (A, B) with (A', B') and repeat; note that this can happen only twice as initially (A, B) has at most two backdoors. If the fourth outcome happens, replace (A, B) with (A^*, B^*) and repeat; note that this step can happen $\mathcal{O}(|V(H)|)$ times, but does not degrade the order of (A, B) nor the size of the wall W.

If we reach (A, B) with all backdoor vertices being pure, we proceed as follows. First, note that we can assume that there is no $(A', B') \in \mathcal{T}_W$ with $A \subseteq A'$ but $|A' \cap B'| < |A \cap B|$, as then we can replace (A, B) with (A', B'), because such an operation cannot turn a pure backdoor into a non-pure backdoor. We would like to return (H, η) and (A, B) as the third outcome of the lemma; to this end, we need to construct the path families $\mathcal{Q}^X$. Fix a subwall W' of W contained in $H[B - A]$; assuming σ is large enough, we can choose W' of sidelength at least $3|A \cap B| + 4$. Since $A \cap B$ and the pegs of W' cannot be separated in $H[B]$ by a separation of order less than $|A \cap B|$, by Menger's theorem, there is a family $\mathcal{P} = \{P_x \mid x \in A \cap B\}$ of vertex-disjoint paths in $H[B]$ such that every $P_x \in \mathcal{P}$ starts in x and ends in a peg of W'. For every $X \subseteq A \cap B$ of size three, construct $\mathcal{Q}^X$ as follows: Apply Lemma 4.4.10 to $\{P_x \mid x \in X\}$ and W', obtaining paths $\{P'_x \mid x \in X\}$. Let y_x be the second vertex of a path P'_x for $x \in X$. As $H[B - A]$ is v-pure, apply Lemma 4.4.11 to the paths P'_x with the first edges removed, obtaining paths R_x for $x \in X$. Prepend every path R_x with a shortest path in $\eta(xy_x)$ from $\eta(xy_x, x)$ to $\eta(xy_x, y_x)$, obtaining the desired path Q^X_x.

The remaining case is when one of the applications of Lemma 4.4.22 returns a separa-

tion $(A_0, B_0) \in \mathcal{T}_W$ of order 1 that captures Π. Let $\{z\} = A_0 \cap B_0$. Compute an extended strip decomposition (H', η') of G as follows. Start with $H' \coloneqq H[B_0]$ and $\eta'(\alpha) \coloneqq \eta(\alpha)$ for every $\alpha \in V(H') \cup E(H') \cup T(H')$. Let V' be the set of vertices of G that are already in some set of η'. Add every vertex of $V(G) - V'$ to the vertex set $\eta'(z)$.

We first observe that (H', η') is indeed an extended strip decomposition of G. Indeed, since z is a cutvertex of H, every vertex of $V(G) - (V' \cup \{v\})$ has only neighbors in $V(G) - V'$ and $\bigcup_{y \in N_H(z) \cap B_0} \eta(zy, z)$. Furthermore, since (A_0, B_0) captures Π, all neighbors of v are also only in $V(G) - V'$ and $\bigcup_{y \in N_H(z) \cap B_0} \eta(zy, z)$. Since $(A_0, B_0) \in \mathcal{T}_W$, we have $\mathfrak{w}(\overleftarrow{\eta}_{(H,\eta)}(B_0 - A_0)) \geqslant 0.99\tau$ by (4.5). Hence, as $\mathfrak{w}(G) \leqslant \tau$, we have $\mathfrak{w}(\eta'(z)) \leqslant 0.01\tau$. Since every particle of (H, η) is of weight at most 0.01τ by (4.2), every particle of (H', η') is of weight at most $(0.01 + 0.01)\tau = 0.02\tau$. Hence, we can output (H', η') as the third output of Lemma 4.4.2.

To finish the proof, we observe that after getting the first wall W from Lemma 4.4.9, all further steps tackle separations of constant order and subwalls of a wall of constant sidelength. Hence, all computations in later steps can be done in polynomial time naively.

■

4.4.7 Applying three-in-a-tree

Lemma 4.4.23 allows us to conclude the proof of Lemma 4.4.2 by applying the three-in-a-tree theorem (Theorem 2.1.1) in the following way.

Given the input to Lemma 4.4.2, we apply Lemma 4.4.23 for the constant t. Unless we are already done, we obtained the last output, consisting of (H, η), (A, B), and W. Recall that (A, B) captures Π with a backdoor set Z of size at most 2, it is triangle-safe, and all $z \in Z$ are pure backdoors.

We will need the following notation. Fix $z \in Z$. Recall that $\mathfrak{P}_z = \bigcup_{x \in N_H(z) \cap (B - A)} \eta(zx, z)$.

Let $V_z \subseteq \eta(z)$ be the set of those vertices $u \in \eta(z)$ that are reachable from v by a path contained in $G[\{v\} \cup \eta(z)]$ whose all vertices, except for possibly the last one, are anti-adjacent to $\mathfrak{P}_z$. Note that as z is a pure backdoor, every vertex $u \in V_z$ is either completely adjacent to $\mathfrak{P}_z$ or completely anti-adjacent to $\mathfrak{P}_z$; we denote by $V_z^\bullet$ and V_z° the sets of vertices of $u \in V_z$ that are completely adjacent and completely anti-adjacent to $\mathfrak{P}_z$, respectively. Furthermore, observe that as z is a pure backdoor, v is either completely adjacent to $\mathfrak{P}_z$ or completely anti-adjacent to $\mathfrak{P}_z$ and, in the first case, $V_z = \emptyset$.

We construct an auxiliary graph G_A as follows. Let $X_A \subseteq V(G)$ consist of the following:

- the vertex v;
- every $\eta(e)$ for $e \in E(G[A])$;
- every $\eta(xyz)$ for $xyz \in T(G[A])$;
- every $\eta(x)$ for $x \in A - B$; and
- for every $z \in Z$, the set V_z.

Observe that X_A is disjoint with $\overleftarrow{\eta}_{(H,\eta)}(B - A)$ and thus $\mathfrak{w}(X_A) \leqslant 0.01\tau$.

We start with $G_A \coloneqq G[X_A]$ and then, for every $x \in A \cap B$ we add two new adjacent vertices $a_{x,1}$ and $a_{x,2}$ to G_A, adjacent to every vertex of $\bigcup_{y \in N_H(x) \cap A} \eta(xy, x)$. Furthermore, if $x \in Z$, we make $a_{x,1}$ and $a_{x,2}$ fully adjacent to $V_z^\bullet$ and, if v is complete to $\mathfrak{P}_z$, also adjacent to v. Observe that for $x \in A \cap B$, the vertices $a_{x,1}$ and $a_{x,2}$ are true twins in G_A. Let $\mathcal{Z} = \{a_{x,i} \mid x \in A \cap B, i \in \{1,2\}\}$.

We apply Theorem 2.1.1 to G_A and the set $\mathcal{Z}$. There are two possible outcomes: either an induced tree in G_A containing at least three elements of $\mathcal{Z}$ or a rigid extended strip decomposition (H_A, η_A) of $(G_A, \mathcal{Z})$. We deal with these cases separately.

An induced tree in G_A. Let K be an induced tree in G_A containing at least three elements of $\mathcal{Z}$. Without loss of generality, we can assume that K contains exactly three elements of $\mathcal{Z}$, and K is either a path with two endpoints in $\mathcal{Z}$ or a tree with exactly three leaves being the elements of $\mathcal{Z}$. For $x \in A \cap B$, as $a_{x,1}$ and $a_{x,2}$ are true twins while K contains three elements of $\mathcal{Z}$ and is an induced tree, K can contain only one of $a_{x,1}$ and $a_{x,2}$. Without loss of generality, we can assume that $V(K) \cap \mathcal{Z} = \{a_{x,1} \mid x \in X\}$ for some $X \subseteq A \cap B$ of size three.

Consider the family $\mathcal{Q}_X = \{Q_x^X \mid x \in X\}$ promised by Lemma 4.4.23 and let u_x be the starting vertex of Q_x^X for $x \in X$. Observe that u_x has exactly the same neighbors in X_A as $a_{x,1}$, while all other vertices of Q_x^X are anti-adjacent to X_A. Hence, $(V(K) - \{a_{x_1} \mid x \in X\}) \cup \{V(Q_x^X) \mid x \in X\}$ induces a tree in G that contains an $S_{t,t,t}$, as desired.

An extended strip decomposition of G_A. We will now show that one can merge (H_A, η_A) and (H, η) into an extended strip decomposition (H^*, η^*) of the whole graph G.

Recall that for every $a_{x,i} \in \mathcal{Z}$ there is a degree-1 vertex $\xi_{x,i} \in V(H_A)$ with a unique neighbor $\zeta_{x,i} \in V(H_A)$ and $\eta_A(\xi_{x,i}\zeta_{x,i}, \xi_{x,i}) = \{a_{x,i}\}$. For every $x \in A \cap B$, since $a_{x,1}$ and $a_{x,2}$ are adjacent, we have $\zeta_{x,1} = \zeta_{x,2}$ (and we henceforth denote this vertex by ζ_x) and $a_{x,i} \in \eta_A(\xi_{x,i}\zeta_x, \zeta_x)$ for $i = 1, 2$. Thus, we can assume without loss of generality that actually $\eta_A(\xi_{x,i}\zeta_x) = \{a_{x,i}\}$ and $\eta_A(\xi_{x,i}) = \emptyset$, as all vertices of $\eta_A(\xi_{x,i}\zeta_x) \cup \eta_A(\xi_{x,i})$ except for $a_{x,i}$ can be moved to $\eta_A(\zeta_x)$. Furthermore, since $a_{x,i}$ and $a_{y,j}$ are nonadjacent for $x \neq y$, $i, j \in \{1, 2\}$, we have that ζ_x and ζ_y are distinct.

Denote $H'_A \coloneqq H_A - \{\xi_{x,i} \mid x \in A \cap B, i = 1, 2\}$ and observe that H'_A with η'_A defined as η_A restricted to the vertices, edges, and triangles present in H'_A is an extended strip decomposition of $G[X_A]$.

Note that every vertex of $V(G) - X_A$ appears in (H, η) either in $\eta(x)$ for some $x \in B$, in $\eta(e)$ for an edge e with at least one endpoint in $B - A$ and both endpoints in B,

or in $\eta(xyz)$ for a triangle xyz with $x, y, z \in B$ and at least one vertex in $B - A$. Hence, $H'_B := H[B]$ with η'_B defined as η with the domain restricted to the objects present in $H[B]$ and every value restricted to vertices of $V(G) - X_A$ is an extended strip decomposition of $G - X_A$. We note that $\eta'_B(e) = \emptyset$ for an edge e with both endpoints in $A \cap B$, but we retain the edge e in H'_B as there may be triangles of $H[B]$ involving it.

We construct H^* by taking a disjoint union of H'_A and H'_B, discarding edges e of H'_B that have both endpoints in $A \cap B$, and identifying ζ_x with x for every $x \in A \cap B$. We define η^* as follows:

- $\eta^*(e)$ for $e \in E(H^*)$ equals $\eta'_A(e)$ or $\eta'_B(e)$, depending on whether e came from H'_A or H'_B;
- $\eta^*(xyz)$ for $xyz \in T(H^*)$ equals $\eta'_A(xyz)$, $\eta'_B(xyz)$, or $\emptyset$, depending on whether the triangle xyz is present in H'_A, or at least one of x, y, z is in $B - A$, or none of these options happen;
- $\eta^*(x)$ for $x \in V(H^*)$ equals $\eta'_A(x)$ for $x \in V(H'_A) - \{\zeta_x \mid x \in A \cap B\}$, equals $\eta'_B(x)$ for $x \in B - A$, equals $\eta'_A(\zeta_x) \cup \eta'_B(x)$ for $x \in A \cap B$.

Finally, we discard from H^* every edge e for which $\eta^*(e) = \emptyset$ and e does not participate in any triangle with a nonempty set in η^*, and every isolated vertex with an empty vertex set.

We now check that (H^*, η^*) is indeed a rigid extended strip decomposition of G. Recall that (H'_A, η'_A) and (H'_B, η'_B) are extended strip decompositions of $G[X_A]$ and $G - X_A$, respectively. Furthermore, (H'_A, η'_A) is rigid; for (H'_B, η'_B), we have $\eta'_B(e) = \eta(e)$ and $\eta'_B(e, x) = \eta(e, x)$ for every e with at least one endpoint in $B - A$ and any $x \in e$, so (H'_B, η'_B) can violate the requirements of being rigid only at vertex sets and edges contained in $A \cap B$. We infer that it remains to check the following three properties:

1. For every $uw \in E(G)$ with $u \in X_A$ and $w \notin X_A$, u and w are placed in (H^*, η^*) in a way allowing the edge uw, that is, both u and w either

   - belong to one set $\eta^*(e)$ for some $e \in E(H^*)$, or
   - belong to one set $\eta^*(x) \cup \bigcup_{y \in N_{H^*}(x)} \eta(xy, x)$ for some $x \in V(H^*)$, or
   - belong to one set $\eta^*(xyz) \cup (\eta^*(xy, x) \cap \eta^*(xy, y)) \cup (\eta^*(yz, y) \cap \eta^*(yz, z)) \cup (\eta^*(zx, z) \cap \eta^*(zx, x))$ for some triangle $xyz \in T(H^*)$.

2. For every $u \in X_A$ and $w \in V(G) - X_A$ such that for some $x \in A \cap B$ it holds that $u, w \in \bigcup_{y \in N_{H^*}(x)} \eta^*(xy, x)$, we have $uw \in E(G)$.

3. For every edge $xy \in E(H)$ with $x, y \subseteq A \cap B$, if there exists a triangle xyz with $z \in B - A$ and $\eta(xyz) = \eta'_B(xyz) \neq \emptyset$, the edge $\zeta_x\zeta_y$ exists in H'_A.

For the first property, let $u \in X_A$ and $w \notin X_A$ be adjacent. We observe that, since (H, η) is an extended strip decomposition of $G - v$, we can break into the following subcases.

- $u = v$. Then, as (A, B) is triangle-safe and captures Π, we have two options when w exists:

  - There exists $z \in Z$ with v complete to $\mathfrak{P}_z$ and $w \in \mathfrak{P}_z$. Then, v is adjacent to $a_{z,1}$ and $a_{z,2}$. Hence, in (H'_A, η'_A) we have that $v \in \eta'_A(\zeta_z) \cup \bigcup_{y \in N_{H'_A}(\zeta_z)} \eta(\zeta_z y, \zeta_z)$. Consequently, in (H^*, η^*) we have that both $u = v$ and w belong to $\eta^*(z) \cup \bigcup_{y \in N_{H^*}(z)} \eta^*(zy, z)$, as desired.
  - There exists a triangle $xyz \in T(H)$ with $x, y \in Z$ and $z \in B - A$ with $w \in \eta(xyz)$. Then, as (A, B) is triangle-safe, we have that v is complete to $\eta(xz, x) \cup \eta(yz, y)$. Hence, v is adjacent to $a_{x,1}$, $a_{x,2}$, $a_{y,1}$, and $a_{y,2}$ in G_A. The only way how (H_A, η_A) can accommodate this is if $\zeta_x\zeta_y \in E(H_A)$

and $v \in \eta_A(\zeta_x\zeta_y, \zeta_x) \cap \eta_A(\zeta_x\zeta_y, \zeta_y)$. Hence, $v \in \eta^*(xy, x) \cap \eta^*(xy, y)$ while $w \in \eta^*(xyz)$, as desired.

- $u \in \eta(z)$ for some $z \in V(H)$. By the definition of X_A, we have actually $z \in A$. By the existence of w, we have $z \in A \cap B$. Since the only parts of sets $\eta(x)$ for $x \in A \cap B$ that are in X_A are sets V_x for $x \in Z$, we have $z \in Z$ and v is anticomplete to $\mathfrak{P}_z$. Furthermore, as V_z° is nonadjacent to $\mathfrak{P}_z$ and to $\eta(z) - V_z$, we have $u \in V_z^\bullet$ and $w \in \mathfrak{P}_z \cup (\eta(z) - V_z)$. Then, u is adjacent to $a_{z,1}$ and $a_{z,2}$ in G_A. Thus, $u \in \eta'_A(\zeta_z) \cup \bigcup_{y \in N_{H'_A}(z)} \eta(\zeta_z y, \zeta_z)$. Hence, $u, w \in \eta^*(z) \cup \bigcup_{y \in N_{H^*}(z)} \eta^*(zy, z)$, as desired.

- $u \in \eta(e)$ for some $e \in E(H)$. By the definition of X_A, both endpoints of e are in A. Due to the existence of the vertex $w \in N_G(u) - X_A$, we have the following options:

  - There exists an endpoint x of e that lies in $A \cap B$, $u \in \eta(e, x)$, and $w \in \eta(x)$ or $w \in \bigcup_{y \in N_H(x) \cap (B-A)} \eta(xy, x)$. Then, u is adjacent to $a_{x,1}$ and $a_{x,2}$ in G_A, and therefore $u \in \eta'_A(\zeta_x) \cup \bigcup_{y \in N_{H'_A}(\zeta_x)} \eta'_A(\zeta_x y, \zeta_x)$. Thus, both u and w belong to $\eta^*(x) \cup \bigcup_{y \in N_{H^*}(x)} \eta^*(xy, x)$.
  - Both endpoints x, y of e lie in $A \cap B$, $u \in \eta(xy, x) \cap \eta(xy, y)$ and there exists $z \in B - A$ with $xz, yz \in E(H)$ and $w \in \eta(xyz)$. We infer that u is adjacent to $a_{x,1}$, $a_{x,2}$, $a_{y,1}$, and $a_{y,2}$ in G_A. The only way how u is accommodated in (H_A, η_A) is that $\zeta_x\zeta_y \in E(H_A)$ and $u \in \eta_A(\zeta_x\zeta_y, \zeta_x) \cap \eta_A(\zeta_x\zeta_y, \zeta_y)$. Hence, $u \in \eta^*(xy, x) \cap \eta^*(xy, y)$ while $w \in \eta^*(xyz)$.

- $u \in \eta(xyz)$ for some $xyz \in T(H)$. By the definition of X_A, we have $x, y, z \in A$. Then, $N_G(\eta(xyz)) \subseteq X_A$, contradicting the existence of w. Hence, this case is impossible.

For the second property, let $x \in A \cap B$ and $u, w \in \bigcup_{y \in N_{H^*}(x)} \eta^*(xy, x)$ with $u \in X_A$

and $w \notin X_A$. By the way we obtained (H^*, η^*), there exists $y_u \in N_{H'_A}(\zeta_x)$ with $u \in \eta_A(\zeta_x y_u, \zeta_x)$ and $y_w \in N_H(x) \cap (B - A)$ with $w \in \eta(xy_w, x)$. In particular, u is adjacent to $a_{x,1}$ and $a_{x,2}$ in G_A. Therefore, by the way we constructed G_A, we have that either $u \in \bigcup_{y \in N_H(x) \cap A} \eta(xy, x)$, or for $z \in Z$, either $u \in V_z^\bullet$, or $u = v$ and v is completely adjacent to $\mathfrak{P}_z$. On all cases, u is completely adjacent to $\bigcup_{y \in N_H(x) \cap (B-A)} \eta(xy, x)$, where w resides. This proves $uw \in E(G)$, as desired.

For the third property, consider $xyz \in T(H)$ with $x, y \in A \cap B$, $z \in B - A$, and $\eta(xy) \neq \emptyset$. Since (H, η) is locally cleaned, there is an edge $uw \in E(G)$ with $w \in \eta(xyz)$ and $u \in \eta(xy, x) \cap \eta(xy, y)$. Then, $u \in X_A$ and u is adjacent to $a_{x,1}$, $a_{x,2}$, $a_{y,1}$, and $a_{y,2}$ in G_A. Hence, the only option to accommodate u in (H_A, η_A) is that $\zeta_x \zeta_y \in E(H_A)$ and $u \in \eta_A(\zeta_x \zeta_y, \zeta_x) \cap \eta_A(\zeta_x \zeta_y, \zeta_y)$. This proves the third property.

Hence, (H^*, η^*) is indeed a rigid extended strip decomposition of G. Since $\mathfrak{w}(X_A) \leqslant 0.01\tau$ and every particle of (H, η) has weight at most 0.01τ, every particle of (H^*, η^*) has weight at most 0.5τ. This proves that (H^*, η^*) satisfies the requirements for the last outcome of Lemma 4.4.2. This concludes the proof of Lemma 4.4.2, and thus also the proof of Lemma 4.4.1.

4.5 Conclusion

Let us point out some possible directions for future research. First, on the structural side, we believe that theorem 4.1.7 could be improved so that in the second outcome the balanced separator is dominated by a constant (depending on t) number of vertices. The only reason why the current statement has the logarithmic bound is that in theorem 4.1.6 the number of deleted neighborhoods is logarithmic. Majewski et al. [50] conjectured that theorem 4.1.6 can actually be improved so that the number of deleted neighborhoods is constant. Proving this conjecture would immediately yield an improved version of our

theorem 4.1.7. However, such a stronger version, while being more elegant, would not give any essentially new algorithmic result: the running time of our algorithms would still be quasi-polynomial (though a bit faster).

On the algorithmic side, an obvious natural problem is to provide a polynomial-time algorithm for MWIS in $S_{t,t,t}$-free graphs, for all t. While we believe that extended strip decompositions are the right tool to use towards this goal, it seems that decompositions like the ones obtained by theorem 4.1.7 would not lead to such a statement. This is because recursing into a polynomial number of multiplicatively smaller particles inherently leads to a quasi-polynomial running time. We believe the ultimate goal would be to build an extended strip decomposition where each particle induces a graph from some "simple" class. In particular, so that we can solve MWIS for each particle in polynomial time without using recursion. Such decompositions for the simplest case, i.e., claw-free graphs, are provided by a deep structural result of Chudnovsky and Seymour [101].

An important milestone on the way towards obtaining a polynomial-time algorithm for MWIS in $S_{t,t,t}$-free graphs is to solve the case of P_t-free graphs, which is already a very ambitious goal.

Chapter 5

Finding Sparse Induced Subgraphs in $C_{>k}$-Free Graphs in Quasi-Polynomial Time

For an integer t, a graph G is called *$C_{>t}$-free* if G does not contain any induced cycle on more than t vertices. We prove the following statement: for every pair of integers d and t and a CMSO_2 statement φ, there exists an algorithm that, given an n-vertex $C_{>t}$-free graph G with weights on vertices, finds in time $n^{\mathcal{O}(\log^4 n)}$ a maximum-weight vertex subset S such that $G[S]$ has degeneracy at most d and satisfies φ. The running time can be improved to $n^{\mathcal{O}(\log^2 n)}$ assuming G is P_t-free, that is, G does not contain an induced path on t vertices. This expands the recent results of the authors [to appear at FOCS 2020 and SOSA 2021] on the Maximum Weight Independent Set problem on P_t-free graphs in two directions: by encompassing the more general setting of $C_{>t}$-free graphs, and by being applicable to a much wider variety of problems, such as Maximum Weight Induced Forest or Maximum Weight Induced Planar Graph.

5.1 Introduction

Consider the Maximum Weight Independent Set (MWIS) problem: given a vertex-weighted graph G, find an independent set in G that has the largest possible weight. While NP-hard in general, the problem becomes more tractable when structural restrictions are imposed on the input graph G. In this work we consider restricting G to come from a fixed *hereditary* (closed under taking induced subgraphs) class $\mathcal{C}$. The goal is to understand how the complexity of MWIS, and of related problems, changes with the class $\mathcal{C}$. A concrete instance of this question is to consider *H-free graphs* — graphs that exclude a fixed graph H as an induced subgraph — and classify for which H, MWIS becomes polynomial-time solvable in H-free graphs.

Somewhat surprisingly, we still do not know the complete answer to this question. A classic argument of Alekseev [4] shows that MWIS is NP-hard in H-free graphs, unless H is a forest of paths and *subdivided claws*: graphs obtained from the claw $K_{1,3}$ by subdividing each of its edges an arbitrary number of times. The remaining cases are still open apart from several small ones: of P_5-free graphs [14], P_6-free graphs [89], claw-free graphs [80, 81], and fork-free graphs [102, 18]. Here and further on, P_t denotes a path on t vertices.

On the other hand, there are multiple indications that MWIS indeed has a much lower complexity in H-free graphs, whenever H is a forest of paths and subdivided claws, than in general graphs. Concretely, as we saw in Chapter 4, in this setting the problem is known to admit a quasi-polynomial time algorithm; note that the existence of such algorithms for general graphs is excluded under standard complexity assumptions. We saw in Chapter 3 an algorithm for MWIS in P_t-free graphs, for every fixed t. The running time was $n^{\mathcal{O}(\log^3 n)}$, which was subsequently improved to $n^{\mathcal{O}(\log^2 n)}$ by Pilipczuk et al. [41].

A key fact that underlies the algorithms for P_t-free graphs is the following balanced

separator theorem (see theorem 5.2.1): In every P_t-free graph, we can find a connected set X consisting of at most t vertices, such that the number of vertices in every connected component of $G-N[X]$ is at most half of the number of vertices of G. It has been observed by Chudnovsky et al. [38] that the same statement is true also in the class of *$C_{>t}$-free graphs*: graphs that do not contain an induced cycle on more than t vertices. Note here that, on one hand, every P_t-free graph is $C_{>t}$-free as well, and, on the other hand, $C_{>t}$-free graphs generalize the well-studied class of *chordal graphs*, which are exactly $C_{>3}$-free. Using the separator theorem, Chudnovsky et al. [38, 50] gave a subexponential-time algorithm and a QPTAS for MWIS on $C_{>t}$-free graphs, for every fixed t.

The basic toolbox developed for MWIS can also be applied to other problems of similar nature. Consider, for instance, the Maximum Weight Induced Forest problem: in a given vertex-weighted graph G, find a maximum-weight vertex subset that induces a forest; note that by duality, this problem is equivalent to Feedback Vertex Set. By lifting techniques used to solve MWIS in polynomial time in P_5-free and P_6-free graphs [14, 89], Abrishami et al. [35] showed that Maximum Weight Induced Forest is polynomial-time solvable both in P_5-free and in $C_{>4}$-free graphs. In fact, the result is even more general: it applies to every problem of the form "find a maximum-weight induced subgraph of treewidth at most k"; MWIS and Maximum Weight Induced Forest are particular instantiations for $k = 0$ and $k = 1$, respectively.

As far as subexponential-time algorithms are concerned, Novotná et al. [103] showed how to use separator theorems to get subexponential-time algorithms for any problem of the form "find the largest induced subgraph belonging to $\mathcal{C}$", where $\mathcal{C}$ is a fixed hereditary class of graphs that have a linear number of edges. The technique applies both to P_t-free and $C_{>t}$-free graphs under the condition that the problem in question admits an algorithm which is single-exponential in the treewidth of the instance graph.

Our results. We extend the recent results on quasipolynomial-time algorithms for MWIS in P_t-free graphs [1, 41] in two directions:

(a) We expand the area of applicability of the techniques to $C_{>t}$-free graphs.

(b) We show how to solve in quasipolynomial time not only the MWIS problems, but a whole family of problems that can be, roughly speaking, described as finding a maximum-weight induced subgraph that is sparse and satisfies a prescribed property.

Both of these extensions require a significant number of new ideas. Formally, we prove the following.

Theorem 5.1.1. *Fix a pair of integers d and t and a CMSO_2 sentence φ. Then there exists an algorithm that, given a $C_{>t}$-free n-vertex graph G and a weight function $\mathfrak{w}\colon V(G) \to \mathbb{N}$, in time $n^{\mathcal{O}(\log^4 n)}$ finds a subset S of vertices such that $G[S]$ is d-degenerate, $G[S]$ satisfies φ, and, subject to the above, $\mathfrak{w}(S)$ is maximum possible; the algorithm may also conclude that no such vertex subset exists. The running time can be improved to $n^{\mathcal{O}(\log^2 n)}$ if G is P_t-free.*

Recall here that a graph G is *d-degenerate* if every subgraph of G contains a vertex of degree at most d; for instance, 1-degenerate graphs are exactly forests and every planar graph is 5-degenerate. Also, CMSO_2 is the *Monadic Second Order* logic of graphs with quantification over edge subsets and modular predicates, which is a standard logical language for formulating graph properties. In essence, the logic allows quantification over single vertices and edges as well as over subsets of vertices and of edges. In atomic expressions one can check whether an edge is incident to a vertex, whether a vertex/edge belongs to a vertex/edge subset, and whether the cardinality of some set is divisible by a fixed modulus. We refer to [104] for a broader introduction.

Corollaries. By applying theorem 5.1.1 for different sentences φ, we can model various problems of interest. For instance, as 1-degenerate graphs are exactly forests, we immediately obtain a quasipolynomial-time algorithm for the Maximum Weight Induced Forest problem in $C_{>t}$-free graphs. Further, as being planar is expressible in CMSO_2 and planar graphs are 5-degenerate, we can conclude that the problem of finding a maximum-weight induced planar subgraph can be solved in quasipolynomial time on $C_{>t}$-free graphs. In section 5.7.5 we give a generalization of theorem 5.1.1 that allows counting the weights only on a subset of S. From this generalization it follows that for instance the following problem can be solved in quasipolynomial time on $C_{>t}$-free graphs: find the largest collection of pairwise nonadjacent induced cycles.

Let us point out a particular corollary of theorem 5.1.1 of a more general nature. It is known that for every pair of integers d and t there exists $\ell = \ell(d,t)$ such that every graph that contains P_ℓ *as a subgraph*, contains either K_{d+2}, or $K_{d+1,d+1}$, or P_t as an induced subgraph [105]. Since the degeneracy of K_{d+2} and $K_{d+1,d+1}$ is larger than d, we conclude that every P_t-free graph of degeneracy at most d does not contain P_ℓ as a subgraph. On the other hand, for every integer ℓ, the class of graphs that do not contain P_ℓ as a subgraph is well-quasi-ordered by the induced subgraph relation [106]. It follows that for every pair of integers t and d and every hereditary class $\mathcal{C}_d$ such that every graph in $\mathcal{C}_d$ has degeneracy at most d, the class $\mathcal{C}_d \cap (P_t\text{-free})$ of P_t-free graphs from $\mathcal{C}_d$ is characterized by a finite number of forbidden induced subgraphs: there exists a finite list $\mathcal{F}$ of graphs such that a graph G belongs to $\mathcal{C}_d \cap (P_t\text{-free})$ if and only if G does not contain any graph from $\mathcal{F}$ as an induced subgraph. As admitting a graph from $\mathcal{F}$ as an induced subgraph can be expressed by a CMSO_2 sentence, from theorem 5.1.1 we can conclude the following.

Theorem 5.1.2. *Let $\mathcal{C}$ be a hereditary graph class such that each member of $\mathcal{C}$ is d-degenerate, for some integer d. Then for every integer t there exists algorithm that, given*

a P_t-free n-vertex graph G and a weight function $\mathfrak{w}\colon V(G) \to \mathbb{N}$, in time $n^{\mathcal{O}(\log^2 n)}$ finds a subset S of vertices such that $G[S] \in \mathcal{C}$ and, subject to this, $\mathfrak{w}(S)$ is maximum possible.

Degeneracy and treewidth. Readers familiar with the literature on algorithmic results for CMSO_2 logic might be slightly surprised by the statement of theorem 5.1.1. Namely, CMSO_2 is usually associated with graphs of bounded treewidth, where the tractability of problems expressible in this logic is asserted by Courcelle's Theorem [54]. theorem 5.1.1, however, speaks about CMSO_2-expressible properties of graphs of bounded degeneracy. While degeneracy is upper-bounded by treewidth, in general there are graphs that have bounded degeneracy and arbitrarily high treewidth. However, we prove that in the case of $C_{>t}$-free graphs, the two notions are functionally equivalent.

Theorem 5.1.3. *For every pair of integers d and t, there exists an integer $k = (dt)^{\mathcal{O}(t)}$ such that every $C_{>t}$-free graph of degeneracy at most d has treewidth at most k.*

As the properties of having treewidth at most k and having degeneracy at most d are expressible in CMSO_2, from theorem 5.1.3 it follows that in the statement of theorem 5.1.1, assumptions "$G[S]$ has degeneracy at most d" and "$G[S]$ has treewidth at most k" could be replaced by one another. Actually, both ways of thinking will become useful in the proof.

Simple QPTASes. As an auxiliary result, we also show a simple technique for turning algorithms for MWIS in P_t-free and $C_{>t}$-free graphs into approximation schemes for (unweighted) problems of the following form: in a given graph, find the largest induced subgraph belonging to $\mathcal{C}$, where $\mathcal{C}$ is a fixed graph class that is closed under taking disjoint unions and induced subgraphs and is *weakly hyperfinite* [107, Section 16.2]. This last property is formally defined as follows: for every $\varepsilon > 0$, there exists a constant $c(\varepsilon)$ such that from every graph $G \in \mathcal{C}$ one can remove an ε-fraction of vertices so that

every connected component of the remaining graph has at most $c(\varepsilon)$ vertices. Weak hyperfiniteness is essentially equivalent to admitting sublinear balanced separators, so all the well-known classes of sparse graphs, e.g. planar graphs or all proper minor-closed classes, are weakly hyperfinite. We present these results in section 5.8.

3-Coloring. In [41], it is shown how to modify the quasipolynomial algorithm for MWIS in P_t-free graphs to obtain an algorithm for 3-Coloring with the same asymptotic running time bound in the same graph class. We remark here that the same modification can be applied to the algorithm of theorem 5.1.1, obtaining the following:

Theorem 5.1.4. *For every integer t there exists an algorithm that, given an n-vertex $C_{>t}$-free graph G, runs in time $n^{\mathcal{O}(\log^4 n)}$ and verifies whether G is 3-colorable.*

5.2 Overview

In this section we present an overview of the proof of our main result, theorem 5.1.1. We try to keep the description non-technical, focusing on explaining the main ideas and intuitions. Complete and formal proofs follow in subsequent sections.

5.2.1 Approach for P_t-free graphs

We need to start by recalling the basic idea of the quasipolynomial-time algorithm for MWIS in P_t-free graphs [1, 41]; we choose to follow the exposition of [41]. The main idea is to exploit the following balanced separator theorem.

Theorem 5.2.1 (Gyárfás [36], Bacsó et al. [37])**.** *Let G be an n-vertex P_t-free graph. Then there exists a set X consisting of at most t vertices of G such that $G[X]$ is connected and every connected component of $G - N[X]$ has at most $n/2$ vertices. Furthermore, such a set can be found in polynomial time.*

In the MWIS problem, there is a natural branching strategy that can be applied on any vertex u. Namely, branch into two subproblems: in one subproblem — *success branch* — assume that u is included in an optimal solution, and in the other — *failure branch* — assume it is not. In the success branch we can remove both u and all its neighbors from the consideration, while in the failure branch only u can be removed. Hence, theorem 5.2.1 suggests the following naive Divide&Conquer strategy: find a set X as provided by the Theorem and branch on all the vertices of X as above in order to try to disconnect the graph. This strategy does *not* lead to any reasonable algorithm, because the graph would get shattered only in the subproblem corresponding to success branches for all $x \in X$. However, there is an intuition that elements of X are reasonable candidates for *branching pivots*: vertices such that branching on them leads to a significant progress of the algorithm.

The main idea presented in [41] is to perform branching while measuring the progress in disconnecting the graph in an indirect way. Let G be the currently considered graph. For a pair of vertices u and v, let the *bucket* of u and v be defined as:

$$\mathcal{B}^G_{u,v} \coloneqq \{\, P \,:\, P \text{ is an induced path in } G \text{ with endpoints } u \text{ and } v\}.$$

Observe that since G is P_t-free, every element of $\mathcal{B}^G_{u,v}$ is a path on fewer than t vertices, hence $\mathcal{B}^G_{u,v}$ has always at most n^{t-1} elements and can be computed in polynomial time (for a fixed t). On the other hand, $\mathcal{B}^G_{u,v}$ is nonempty if and only if u and v are in the same connected component of G.

Let X be a set whose existence is asserted by theorem 5.2.1. Observe that if u and v are in different components of $G - N[X]$, then *all* the paths of $\mathcal{B}^G_{u,v}$ are intersected by $N[X]$. Moreover, as every connected component of $G - N[X]$ has at most $n/2$ elements, this happens for at least half of the pairs $\{u, v\} \in \binom{V(G)}{2}$. Since X has only at most t

vertices, by a simple averaging argument we conclude the following.

Claim 5.2.2. *There is a vertex x such that $N[x]$ intersects at least a $\frac{1}{t}$-fraction of paths in at least $\frac{1}{2t}$-fraction of buckets.*

A vertex x having the property mentioned in claim 5.2.2 shall be called $\frac{1}{2t}$-*heavy*, or just *heavy*. Then claim 5.2.2 asserts that there is always a heavy vertex; note that such a vertex can be found in polynomial time by inspecting the vertices of G one by one.

We may now present the algorithm:

1. If G is disconnected, then apply the algorithm to every connected component of G separately.
2. Otherwise, find a heavy vertex in G and branch on it.

We now sketch a proof of the following claim: on each root-to-leaf path in the recursion tree, this algorithm may execute only $\mathcal{O}(\log^2 n)$ success branches. By claim 5.2.2, in each success branch a constant fraction of buckets get their sizes reduced by a constant multiplicative factor. Since buckets are of polynomial size in the first place, after $\Omega(\log n)$ success branches a $\frac{1}{10}$-fraction of the initial buckets must become empty. Since in a connected graph all the buckets are nonempty, it follows that after $\Omega(\log n)$ success branches, the vertex count of the connected graph we are working on must have decreased by at least a multiplicative factor of 0.01 with respect to the initial graph. As this can happen only $\mathcal{O}(\log n)$ times, the claim follows.

Now the recursion tree has depth at most n and each root-to-leaf path contains at most $\mathcal{O}(\log^2 n)$ success branches. Therefore, the total size of the recursion tree is $n^{\mathcal{O}(\log^2 n)}$, which implies the same bound on the running time. This concludes the description of the algorithm for P_t-free graphs; let us recall that this algorithm was already presented in [41].

5.2.2 Lifting the technique to $C_{>t}$-free graphs

We now explain how to lift the technique presented in the previous section to the setting of $C_{>t}$-free graphs. As we mentioned before, the main ingredient — the balanced separator theorem — remains true.

Theorem 5.2.3 (Gyárfás [36], Chudnovsky et al. [38])**.** *Let G be an n-vertex $C_{>t}$-free graph. Then there is a set X consisting of at most t vertices of G such that $G[X]$ is connected and every connected component of $G - N[X]$ has at most $n/2$ vertices. Furthermore, such a set can be found in polynomial time.*

However, in the previous section we used the P_t-freeness of the graph in question also in one other place: to argue that the buckets $\mathcal{B}^G_{u,v}$ are of polynomial size. This was crucial for the argument that $\Omega(\log n)$ success branches on heavy vertices lead to emptying a significant fraction of the buckets. Solving this issue requires reworking the concept of buckets.

The idea is that in the $C_{>t}$-free case, the objects placed in buckets will connect triples of vertices, rather than pairs. Formally, a *connector* is a graph formed from three disjoint paths Q_1, Q_2, Q_3 by picking one endpoint a_i of Q_i, for each $i = 1, 2, 3$, and either identifying vertices a_1, a_2, a_3 into one vertex, or turning a_1, a_2, a_3 into a triangle; see fig. 5.1. The paths Q_i are the *legs* of the connector, the other endpoints of the legs are the *tips*, and the (identified or not) vertices a_1, a_2, a_3 are the *center* of the connector. We remark that we allow the degenerate case when one or more paths Q_1, Q_2, Q_3 has only one vertex, but we require the tips to be pairwise distinct.

The following claim is easy to prove by considering any inclusion-wise minimal connected induced subgraph containing u, v, w.

Claim 5.2.4. *If vertices u, v, w belong to the same connected component of a graph G, then in G there is an induced connector with tips u, v, w.*

A *tripod* is a connector in which every leg has length at most $t/2+1$ (w.l.o.g. t is even). Every connector contains a *core*: the tripod induced by the vertices at distance at most $t/2$ from the center. The next claim is the key observation that justifies looking at connectors and tripods.

Claim 5.2.5. *Let G be a $C_{>t}$-free graph, let T be an induced connector in G, and let X be a subset of vertices such that $G[X]$ is connected and no two tips of T are in the same connected component of $G - N[X]$. Then $N[X]$ intersects the core of T.*

Proof of Claim 1. *Since no two tips of T lie in the same component of $G - N[X]$, it follows that $N[X]$ intersects at least two legs of T, say Q_1 and Q_2 at vertices q_1 and q_2, respectively. We may choose q_1 and q_2 among $N[X] \cap V(Q_1)$ and $N[X] \cap V(Q_2)$ so that they are as close in T as possible to the center of T. Since $G[X]$ is connected, there exists a path P with endpoints q_1 and q_2 such that all the internal vertices of P belong to X. Now P together with the shortest q_1-q_2 path within T form an induced cycle in G. As this cycle must have at most t vertices, we conclude that q_1 or q_2 belongs to the core of T.*

claim 5.2.5 suggests that in $C_{>t}$-free graphs, cores of connectors are objects likely to be hit by balanced separators provided by theorem 5.2.3, similarly as in P_t-free graphs, induced paths were likely to be hit by balanced separators given by theorem 5.2.1. Let us then use cores as objects for defining buckets.

Let G be a $C_{>t}$-free graph. For an unordered triple $\{u,v,w\} \in \binom{V(G)}{3}$ of distinct vertices, we define the bucket $\mathcal{B}^G_{u,v,w}$ as the set of all cores of all induced connectors with tips u,v,w. Let us stress here that $\mathcal{B}^G_{u,v,w}$ is a set, not a multiset, of tripods: even if some tripod is the core of multiple connectors with tips u,v,w, it is included in $\mathcal{B}^G_{u,v,w}$ only once. Therefore, as each tripod has $\mathcal{O}(t)$ vertices, the buckets are again of size $n^{\mathcal{O}(t)}$ and can be enumerated in polynomial time. By claim 5.2.4, the bucket $\mathcal{B}^G_{u,v,w}$ is nonempty if

and only if u, v, w are in the same connected component of G. Moreover, from claim 5.2.5 we infer the following.

Claim 5.2.6. *Let $\{u, v, w\} \in \binom{V(G)}{3}$ be a triple of vertices of G and let X be a vertex subset such that $G[X]$ is connected and no two vertices out of u, v, w belong to the same connected component of $G - N[X]$. Then $N[X]$ intersects all the tripods in the bucket $\mathcal{B}^G_{u,v,w}$.*

We now would like to obtain an analogue of claim 5.2.2, that is, find a vertex x such that $N[x]$ intersects a significant fraction of tripods in a significant fraction of buckets. Let then X be a set provided by theorem 5.2.3 for G. For a moment, let us assume optimistically that each connected component of $G-N[X]$ contains at most $n/10$ vertices, instead of $n/2$ as promised by theorem 5.2.3. Observe that if we choose a triple of distinct vertices uniformly at random, then with probability at least $\frac{1}{2}$ no two of these vertices will lie in the same connected component of $G - N[X]$. By claim 5.2.5, this implies that $N[X]$ intersects all the tripods in at least half of the buckets. By the same averaging argument as before, we get the following.

Claim 5.2.7. *Suppose that in G there is a set X consisting of at most t vertices such that $G[X]$ is connected and every connected component of $G - N[X]$ has at most $n/10$ vertices. Then there is a heavy vertex in G.*

Here, we define a heavy vertex as before: it is a vertex x such that $N[x]$ intersects at least a $\frac{1}{t}$-fraction of tripods in at least a $\frac{1}{2t}$-fraction of buckets.

Unfortunately, our assumption that every component of $G - N[X]$ contains at most $n/10$ vertices, instead of at most $n/2$ vertices, is too optimistic. Consider the following example: G is a path on n vertices. The cores of connectors degenerate to subpaths consisting of at most t consecutive vertices of the path, and for every vertex x, the set

$N[x]$ intersects any tripod in only an $\mathcal{O}(t/n)$-fraction of the buckets. Therefore, in this example there is no heavy vertex at all. We need to resort to a different strategy.

Secondary branching. So let us assume that the currently considered graph G is connected and has no heavy vertex — otherwise we may either recurse into connected components or branch on the heavy vertex (detectable in polynomial time). We may even assume that there is no $\frac{1}{4t}$-*heavy* vertex: a vertex x such that $N[x]$ intersects at least a $\frac{1}{2t}$-fraction of tripods in at least a $\frac{1}{4t}$-fraction of buckets buckets. Indeed, branching on such vertices also leads to quasipolynomial running time (with all factors in the analysis appropriately scaled).

Let us fix a set X provided by theorem 5.2.3 for G; then $G[X]$ is connected and each connected component of $G - N[X]$ has at most $n/2$ vertices. By claim 5.2.7, there must be some components of $G - N[X]$ that have more than $n/10$ vertices, for otherwise there would be a heavy vertex. The idea is that we perform some additional branching in those large components in order to shatter them into smaller pieces, so that a heavy vertex must appear due to claim 5.2.7.

A connected component C of $G - N[X]$ shall be called a *chip* if C has at least $n/100$ vertices and C has a neighbor in $N[X]$. Our goal is to get rid of all the chips by performing branching on vertices of G. Indeed, if we achieve this, then either the vertex count of every connected component of G drops to below $0.99n$ — which is a huge progress for the algorithm — or G still contains a connected component D that constitutes more than 99% of the original graph G. However, in the latter case, every component of $D - N[X]$ has at most $n/100$ vertices, as otherwise it would be a chip. So by applying claim 5.2.7 we may find a $\frac{1}{2t}$-heavy vertex in D. We will later argue what shall be done with such a heavy vertex, but for now let us focus on the goal of getting rid of all the chips.

Consider a chip C. A *C-link* is a path in G with endpoints in $N[X] \cap N(C)$ and all

internal vertices in C; this path should be induced, except that we allow the existence of an edge between the endpoints. Observe the following:

Claim 5.2.8. *Every C-link has at most t vertices.*

Proof of Claim 2. *Let P be a C-link. Since the endpoints of P are in $N[X]$ and $G[X]$ is connected, there exists an induced path Q with same endpoints as P such that all the internal vertices of P are in X. Then $P \cup Q$ is an induced cycle in G, hence both P and Q must have at most t vertices.*

The idea is that in order to shatter the chips, we perform a secondary branching procedure, but this time we use C-links as objects that are hit by neighborhoods of vertices. Formally, for a pair $(w, \{u, v\})$, where $\{u, v\} \in \binom{N[X]}{2}$, we consider the *secondary bucket* $\mathcal{L}^G_{w;u,v}$ consisting of all C-links with endpoints u and v, where C is the chip that contains w. Again, by claim 5.2.8, each secondary bucket is of size at most n^t and can be enumerated in polynomial time. Note that $\mathcal{L}^G_{w;u,v}$ is nonempty if and only if there is a chip C such that $u, v \in N(C)$ and $w \in C$.

We shall say that a vertex z of G is *secondary-heavy* if $N[z]$ intersects at least a $\frac{1}{100t}$-fraction of links in at least a $\frac{1}{200t}$-fraction of nonempty secondary buckets.

Claim 5.2.9. *If there is a nonempty bucket, then there is a secondary-heavy vertex.*

Proof of Claim 3 (Sketch)**.** *Since every chip has at least $n/100$ vertices, there are at most 100 chips, so there is a chip C which gives rise to at least a $\frac{1}{100}$-fraction of nonempty buckets. We apply a weighted variant of theorem 5.2.3 to the graph $G[N[C]]$ in order to find a set $Z \subseteq N[C]$ of size at most t such that every connected component of $C - N[Z]$ contains at most half of the vertices of $N(C)$. Then $N[Z]$ intersects all the links in at least half of the buckets. The same averaging argument as used before shows that one of vertices of Z is secondary-heavy.*

The secondary branching procedure now branches on a secondary-heavy vertex (detectable in polynomial time) as long as there are nonempty secondary buckets. This is always possible by claim 5.2.9. Note that during this branching, the set of chips changes — some components of $G-N[X]$ may stop being chips due to either getting disconnected from $N[X]$ or their sizes being reduced to below $n/100$.

The same analysis as in section 5.2.1 shows that branching on secondary-heavy vertices results in a recursion tree with $n^{\mathcal{O}(\log^2 n)}$ leaves. In each of these leaves all secondary buckets are empty, so no chip has more than one neighbor in $N[X]$.

Let us focus on one leaf of the recursion tree described above. Suppose for a moment that there is still some chip C left. Then $N(C)$ consists of only one vertex, say z. Note that z is a cutvertex in the current graph and if we branch on z, in both branches C becomes a separate connected component of the graph, not connected to $N[X]$. Since C has at least $n/100$ and at most $n/2$ vertices, we conclude that after branching on z, in both the success branch and in the failure branch, every connected component has at most $0.99n$ vertices. This is a huge progress for the algorithm, so let us perform this branching at the end.

Thus, in each of the $n^{\mathcal{O}(\log^2 n)}$ leaves of the obtained recursion tree we have either an instance where every connected component has at most $0.99n$ vertices, or an instance that has no chips, and therefore there exists a heavy vertex (for the primary branching). It is tempting now to say that in the latter instances we simply branch on the obtained heavy vertex. However, this would *not* lead to a sound complexity analysis, because in total we would end up with not one, but $n^{\mathcal{O}(\log^2 n)}$ failure branches, in which essentially no progress can be measured.

The final observation is as follows: we do not need to perform branching on the obtained heavy vertices, as already their appearance witnesses that the (primary) branching noted a substantial progress. More precisely, consider an instance in a leaf of the recur-

sion tree of the secondary branching, and let x be a $\frac{1}{2t}$-heavy vertex in this instance. This vertex was not $\frac{1}{4t}$-heavy before the secondary branching, because we assumed that there were no $\frac{1}{4t}$-heavy vertices at that point. Obviously, all the tripods that $N[x]$ hits after the secondary branching were also hit by $N[x]$ before the secondary branching was executed. This means that for at least a $\frac{1}{2t}$-fraction of all the buckets, the number of tripods in them must have at least halved during the secondary branching — so that the at most $\frac{1}{4t}$-fraction of hit tripods before the branching could become the at least $\frac{1}{2t}$-fraction of hit tripods after the branching. This is a progress comparable to making a success branch when branching on a heavy vertex.

To summarize, we perform branching on $\frac{1}{4t}$-heavy vertices and recursing on connected components as long as a $\frac{1}{4t}$-heavy vertex can be found. When this ceases to be the case, we resort to secondary branching. Such an application of secondary branching results in producing $n^{\mathcal{O}(\log^2 n)}$ subinstances to solve, where in each subinstance either the vertex count of the graph drops by 1%, or we note a progress in emptying the buckets comparable to making a success branch on a heavy vertex. Combining this with the complexity analysis presented in section 5.2.1, we infer that the running time is $n^{\mathcal{O}(\log^4 n)}$. This concludes the description of an $n^{\mathcal{O}(\log^4 n)}$-time algorithm for MWIS on $C_{>t}$-free graphs.

5.2.3 Degeneracy branching

Our goal in this section is to generalize the approach presented in the previous section to an algorithm solving the following problem: given a vertex-weighted $C_{>t}$-free graph G, find a maximum-weight subset of vertices S such that $G[S]$ is d-degenerate. Here d and t are considered fixed constants. Thus we allow the solution to be just sparse instead of independent, but, compared to theorem 5.1.1, so far we do not introduce CMSO_2-expressible properties. Let us call the considered problem Maximum Weight

Induced d-Degenerate Graph (MWID). The algorithm for MWID that we are going to sketch is formally presented in section 5.4 and section 5.5, and is the subject of theorem 5.4.1 there.

Recall that a graph G is d-degenerate if every subgraph of G has a vertex of degree at most d. We will rely on the following characterization of degeneracy, which is easy to prove.

Claim 5.2.10. *A graph G is d-degenerate if and only if there exists a function $\eta\colon V(G) \to \mathbb{N}$ such that for every $uv \in E(G)$ we have $\eta(u) \neq \eta(v)$ and for each $u \in V(G)$, u has at most d neighbors v with $\eta(v) < \eta(u)$.*

A function $\eta(\cdot)$ satisfying the premise of claim 5.2.10 shall be called a *degeneracy ordering*. Note that we only require that a degeneracy ordering is injective on every edge of the graph, and not necessarily on the whole vertex set. For a vertex u, the value $\eta(u)$ is the *position* of u and the set neighbors of u with smaller positions is the *left neighborhood* of u.

We shall now present a branching algorithm for the MWID problem. For convenience of exposition, let us fix the given $C_{>t}$-free graph G, an optimum solution $S^\star$ in G, and a degeneracy ordering $\eta^\star$ of $G[S^\star]$. We may assume that the co-domain of $\eta^\star$ is $[n] \coloneqq \{1, \ldots, n\}$.

Recall that when performing branching for the MWIS problem, say on a vertex x, in the failure branch we were removing x from the graph, while in the success branch we were removing both x and its neighbors. When working with MWID, we cannot proceed in the same way in the second case, because the neighbors of x can be still included in the solution. Therefore, instead of modifying the graph G along the recursion, we keep track of two disjoint sets A and W: A consists of vertices already decided to be included in the solution, while W is the set of vertices that are still allowed to be taken to the

solution in further steps. Initially, $A = \emptyset$ and $W = V(G)$. We shall always branch on a vertex $x \in W$: in the failure branch we remove x from W, while in the success branch we move x from W to A. The intuition is that moving x to A puts more restrictions on the neighbors of x that are still in W. This is because they are now adjacent to one more vertex in A, and they cannot be adjacent to too many, at least as far as vertices with smaller positions are concerned.

For the positions, during branching we will maintain the following two pieces of information:

- a function $\eta\colon A \to [n]$ that is our guess on the restriction of $\eta^\star$ to A; and
- a function $\zeta\colon W \to [n]$ which signifies a *lower bound* on the position of each vertex of W, assuming it is to be included in the solution.

Initially, we set $\zeta(v) = 1$ for each $v \in V(G)$. The quadruple (A, W, η, ζ) as above describes a subproblem solved during the recursion. We will say that such a subproblem is *lucky* if all the choices made so far are compliant with $S^\star$ and $\eta^\star$, that is,

$$A \subseteq S^\star \subseteq A \cup W, \qquad \eta = \eta^\star|_A, \qquad \text{and} \qquad \eta^\star(u) \geqslant \zeta(u) \text{ for each } u \in S^\star \cap W.$$

Additionally to the above, from a lucky subproblem we also require the following property:

$$\text{for each } v \in A \text{ and } u \in N(v) \cap W \text{ such that } \zeta(u) \leqslant \eta^\star(v), \text{ we have } u \in S^\star \text{ and } \eta^\star(u) < \eta^\star(v). \tag{5.1}$$

In other words, all the neighbors of a vertex $v \in A$ should have their lower bounds larger than the guessed position of v, unless they will be actually included in the solution at positions smaller than that of v. The significance of this property will become clear in a moment.

First, observe that if $G[W]$ is disconnected, then we can treat the different connected components of $G[W]$ separately: for each component D of $G[W]$ we solve the subproblem $(A, D, \eta, \zeta|_D)$ obtaining a solution S_D, and we return $\bigcup_D S_D$ as the solution to (A, W, η, ζ). Property (5.1) is used to guarantee the correctness of this step: it implies that when taking the union of solutions S_D, the vertices of A do not end up with too many left neighbors.

Thus, we may assume that $G[W]$ is connected. In this case we execute branching on a vertex of W. For the choice of the branching pivot x we use exactly the same strategy as described in the previous section: having defined the buckets in exactly the same way, we always pick x to be a heavy vertex in $G[W]$, or resort to secondary branching in $G[W]$ (which picks secondary-heavy pivots) in the absence of heavy vertices.

An important observation is that in the success branch — when the vertex $x \in W$ is moved to A — the algorithm notes a significant progress that allows room for additional guessing (by branching). More precisely, on every root-to-leaf path in the recursion tree there are only $\mathcal{O}(\log^4 n)$ success branches, which means that following each success branch we can branch further into $n^{\mathcal{O}(1)}$ options, and the size of the recursion tree will be still $n^{\mathcal{O}(\log^4 n)}$. We use this power to guess (by branching) the following objects when deciding that x should be included in the solution $S^\star$ (here, we assume that the current subproblem is lucky):

- the position $\eta^\star(x)$;
- the set of left neighbors $L = \{v \in W \cap N(x) \mid \eta^\star(v) < \eta^\star(x)\}$;
- the positions $(\eta^\star(v) \colon v \in L)$; and
- for each $v \in L$, its left neighbors $L_v \coloneqq \{u \in W \cap N(v) \mid \eta^\star(u) < \eta^\star(v)\}$.

This guess is reflected by the following clean-up operations in the subproblem:

- Move $\{x\} \cup L$ from W to A and set their positions in $\eta(\cdot)$ as the guess prescribes. Note that the vertices of $\bigcup_{v \in L} L_v$ are *not* being moved to A.

- For each $w \in (N(x) \cap W) - L$, increase $\zeta(w)$ to $\max(\zeta(w), \eta(x) + 1)$.
- For each $v \in L$ and $w \in (N(v) \cap W) - L_v$, increase $\zeta(w)$ to $\max(\zeta(w), \eta(v) + 1)$.

It is easy to see that if (A, W, η, ζ) was lucky, then at least one of the guesses leads to considering a lucky subproblem. In particular, property (5.1) is satisfied in this subproblem. This completes the description of a branching step.

It remains to argue why it is still true that on every root-to-leaf path in the recursion tree there are at most $\mathcal{O}(\log^4 n)$ success branches. Before, the key argument was that when a success branch is executed, a constant fraction of buckets (either primary or secondary) loses a constant fraction of elements. Now, the progress is explained by the following claim, which follows easily from the way we perform branching.

Claim 5.2.11. *Suppose (A, W, η, ζ) is a lucky subproblem in which branching on x is executed, and let (A', W', η', ζ') be any of the obtained child subproblems. Then for every $y \in N(x) \cap W$, we either have*

$$y \notin W' \qquad \text{or} \qquad |\{z \in A \cap N(y) \mid \eta(z) < \zeta(y)\}| < |\{z \in A' \cap N(y) \mid \eta'(z) < \zeta'(y)\}|.$$

Note that for $y \in W$, if y gets included in the solution, then the whole set

$$M_y \coloneqq \{z \in A \cap N(y) \mid \eta(z) < \zeta(y)\}$$

must become the left neighbors of y. So if the size of M_y exceeds d, then we can conclude that y cannot be included in the solution and we can safely remove y from W. Thus, the increase of the cardinality of M_y for all neighbors y of x that do not get excluded from consideration is the progress achieved by the algorithm.

Formally, we do as follows. Recall that before, we measured the progress in emptying

a bucket $\mathcal{B}^G_{u,v,w}$ by monitoring its size. Now, we monitor the *potential* of $\mathcal{B}^G_{u,v,w}$ defined as

$$\Phi(\mathcal{B}^G_{u,v,w}) \coloneqq \sum_{T \in \mathcal{B}^G(u,v,w)} \sum_{y \in V(T)} (d - |M_y|).$$

Thus, $\Phi(\mathcal{B}^G(u,v,w))$ measures how much the vertices of tripods of $\mathcal{B}^G_{u,v,w}$ have left till saturating their "quotas" for the number of left neighbors. From claim 5.2.11 it can be easily inferred that when branching on a heavy vertex, a constant fraction of buckets lose a constant fraction of their potential, and the same complexity analysis as before goes through.

5.2.4 CMSO_2 properties

We now extend the approach presented in the previous section to a sketch of a proof of theorem 5.1.1 in full generality. That is, we also take into account CMSO_2-expressible properties.

Degeneracy and treewidth. The first step is to argue that degeneracy and treewidth are functionally equivalent in $C_{>t}$-free graphs, i.e., to prove theorem 5.1.3. This part of the reasoning is presented in section 5.6.

The argument goes roughly as follows. Suppose, for contradiction, that G is a $C_{>t}$-free d-degenerate graph that has huge treewidth (in terms of d and t). Using known results [108], in G we can find a huge *bramble* $\mathcal{B}$ — a family of connected subgraphs that pairwise either intersect or are adjacent — such that every vertex of G is in at most two elements of $\mathcal{B}$. This property means that $\mathcal{B}$ gives rise to a huge clique minor in G', the graph obtained from G by adding a copy of every vertex (the copy is a true twin of the original). Note that G' is still $C_{>t}$-free and is $2d+1$-degenerate. Now, we can easily prove that the obtained clique minor in G' can be assumed to have *depth* at most

t: every branch set induces a subgraph of radius at most t. Using known facts about bounded-depth minors [109, Lemma 2.19 and Corollary 2.20], it follows that G' contains a topological minor model of a large clique that has depth at most $3t+1$: every path representing an edge has length at most $6t+3$. Finally, we show that if we pick at random $t+1$ roots $v_0, \ldots, v_t$ of this topological minor model, and we connect them in order into a cycle in G' using the paths from the model, then with high probability this cycle will be induced. This is because G' is $(2d+1)$-degenerate, so two paths of the model chosen uniformly at random are with high probability nonadjacent, due to their shortness. Thus, we uncovered an induced cycle on more than t vertices in G', a contradiction.

Boundaried graphs and types. We proceed to the proof of theorem 5.1.1. By theorem 5.1.3, the subgraph $G[S]$ induced by the solution has treewidth smaller than k, where k is a constant that depends only on d and t. Therefore, we will use known compositionality properties of CMSO_2 logic on graphs of bounded treewidth.

For an integer ℓ, an *ℓ-boundaried graph* is a pair (H, ι), where H is a graph and ι is an injective partial function from $V(H)$ to $[\ell]$, called the *labelling*. The domain of ι is the *boundary* of (H, ι) and if $\iota(u) = \alpha$, then u is a boundary vertex with label α. On ℓ-boundaried graphs we have two natural operations: *forgetting* a label — removing a vertex with this label from the domain of ι — and *gluing* two boundaried graphs — taking their disjoint union and fusing boundary vertices with the same labels. It is not hard to see that a graph has treewidth less than ℓ if and only if it can be constructed from two-vertex ℓ-boundaried graphs by means of these operations.

The crucial, well-known fact about CMSO_2 is that this logic behaves in a compositional way under the operations on boundaried graphs. Precisely, for each fixed ℓ and CMSO_2 sentence φ there is a finite set Types of *types* such that to every ℓ-boundaried graph (H, ι) we can assign $\mathsf{type}(H, \iota) \in \mathsf{Types}$ so that:

- Whether $H \models \varphi$ can be uniquely determined by examining $\mathsf{type}(H, \iota)$.
- The type of the result of gluing two ℓ-boundaried graphs depends only on the types of those graphs.
- The type of the result of forgetting a label in an ℓ-boundaried graph depends only on the label in question and the type of this graph.

See proposition 2.4.1 for a formal statement. In our proof we will use $\ell := 6k$, that is, the boundaries will by a bit larger than the promised bound on the treewidth.

Enriching branching with types. We now sketch how to enrich the algorithm from the previous section to the final branching procedure; this part of the reasoning is presented in section 5.7. The idea is that we perform branching as in the previous section (with significant augmentations, as will be described in a moment), but in order to make sure that the constructed induced subgraph $G[S]$ satisfies φ, we enrich each subproblem with the following information:

- A rooted tree decomposition (T, β) of $G[A]$ of width at most ℓ ($\beta \colon V(T) \to A$ is the bag function).
- For each node a of T, a *projected type* $\mathsf{type}_a \in \mathsf{Types}$.

Again, we fix some optimum solution $S^\star$ together with a d-degeneracy ordering $\eta^\star$ of $G[S^\star]$. Compared to the approach of the previous section, we extend the definition of a subproblem being lucky as follows:

- For each connected component D of $G[W \cap S^\star]$, we require that $N(D) \cap A$ is a set of size at most $4k$ such that there exists a bag of (T, β) that entirely contains it. For such a component D, let $a(D)$ be the topmost node of T satisfying $N(D) \cap A \subseteq \beta(a(D))$.
- For each node a of T, consider the graph H_a induced by $\beta(a)$ and the union of all those components D of $G[W \cap S^\star]$ for which $a(D) = a$. Then the type of H_a with

$\beta(a)$ as the boundary is equal to type_a.

Thus, one can imagine the solution $S^\star$ as A plus several extensions into W — vertex sets of components of $G[W \cap S^\star]$. Each of such extensions D is hanged under a single node $a(D)$ of (T, β) and is attached to it through a neighborhood of size at most $4k$. For each node a of T, we store the projected combined type type_a of all the extensions D for which a is the topmost node to which D attaches. Note that as since the algorithm will always make only $\log^{\mathcal{O}(1)} n$ success branches along each root-to-leaf path, we maintain the invariant that $|A| \leqslant \log^{\mathcal{O}(1)} n$, which implies the same bound on the number of nodes of T.

Recall that in the algorithm presented in the previous section, two basic operations were performed: (a) recursing on connected components of $G[W]$ once this graph becomes disconnected; and (b) branching on a node $x \in W$.

Lifting (a) to the new setting is conceptually simple. Namely, each graph H_a is correspondingly split among the components of $G[W]$, so we guess the projected types of those parts of H_a so that they compose to type_a. The caveat is that in order to make the time complexity analysis sound, we can perform such guessing only when a significant progress is achieved by the algorithm. This is done by performing (a) *only* when each connected component of $G[W]$ contains at most 99% of all the vertices of W, which means that the number of active vertices after this step will drop by 1% in each branch. This requires technical care.

More substantial changes have to be applied to lift operation (b), branching on a node $x \in W$. Failure branch works the same way as before: we just remove x from W. As for success branches, recall that in each of them together with x we move to A the whole set L of left neighbors of x in W. Clearly, the vertices of $L \cup \{x\}$ belong to the same component of $G[A \cap S^\star]$, say D. It would be natural to reflect the move of $L \cup \{x\}$ to A in the decomposition (T, β) as follows: create a new node b with $\beta(b) = (L \cup \{x\}) \cup (N(D) \cap A)$

and make it a child of $a(D)$ in T. Note that thus, $|\beta(b)| \leqslant d+1+4k \leqslant 5k$, so the bound on the width of (T, β) is maintained (even with a margin). However, there is a problem: if by A' we denote the updated A, i.e., $A' = A \cup (L \cup \{x\})$, then the removal of $L \cup \{x\}$ breaks D into several connected components whose neighborhoods in A' are contained in $(N(D) \cap A) \cup (L \cup \{x\})$. This set, however, may have size as large as $4k + d + 1$, so we do not maintain the invariant that every connected component of $G[W \cap S^\star]$ has at most $4k$ neighbors in A.

We remedy this issue using a trick that is standard in the analysis of bounded-treewidth graphs. Since the graph $G[D \cup (N(D) \cap W)]$ has treewidth smaller than k, there exists a set K consisting of at most k vertices of $D \cup (N(D) \cap W)$ such that every connected component of $D - K$ contains at most $|N(D) \cap W|/2$ vertices of $N(D) \cap W$ (see lemma 2.3.1). The algorithm guesses K along with L, moves K to W along with x and L, and and sets $\beta(b) \coloneqq (N(D) \cap A) \cup (L \cup \{x\}) \cup K$; thus $|\beta(b)| \leqslant 4k + (d+1) + k \leqslant 6k$. Now it is easy to see that due to the inclusion of K, every connected component of $D - (K \cup L \cup \{x\})$ has only at most $2k + k + (d+1) \leqslant 4k$ neighbors in $\beta(b)$, and the problematic invariant is maintained.

This concludes the overview of the proof of theorem 5.1.1.

5.3 Preliminaries

We use standard graph notation. For an positive integer p, we write $[p] \coloneqq \{1, \ldots, p\}$. For a set A, $\binom{A}{p}$ denotes the set of all p-element subsets of A.

5.4 Branching framework

In this section we prove theorem 5.4.1 stated below, which is a weaker variant of theorem 5.1.1 that does not speak about CMSO_2 properties.

Theorem 5.4.1. *Fix a pair of integers d and t. Then there exists an algorithm that, given a $C_{>t}$-free n-vertex graph G and a weight function $\mathfrak{w}\colon V(G) \to \mathbb{N}$, in time $n^{\mathcal{O}(\log^4 n)}$ finds a subset of vertices S such that $G[S]$ is d-degenerate and, subject to the above, $\mathfrak{w}(S)$ is maximum possible. The running time can be improved to $n^{\mathcal{O}(\log^2 n)}$ if G is P_t-free.*

We present the strategy for our branching algorithm in a quite general fashion, so that it can be reused later for the proof of theorem 5.1.1. For the description, we fix a positive integer d that is the bound on the degeneracy of the sought induced subgraph.

We will rely on the following characterization of graphs of bounded degeneracy. As the statement of lemma 5.4.2 is a bit non-standard, we include a sketch of a proof.

Lemma 5.4.2. *A graph G has degeneracy at most d if and only if there exists a function $\eta\colon V(G) \to [|V(G)|]$ such that for every $uv \in E(G)$ we have $\eta(u) \neq \eta(v)$, and for every $v \in V(G)$, the set $\{u \in N_G(v) \mid \eta(u) < \eta(v)\}$ has size at most d.*

Proof: [sketch] If G is d-degenerate, then we can construct an ordering η by removing a vertex v with minimum degree, inductively ordering the vertices of $G-v$, and appending v at the last position.

On the other hand, consider any subgraph G' of G and let η' be the ordering η restricted to the vertices of G'. Let $v \in V(G')$ be a vertex at the last position in η'; if there is more than one such vertex, we choose one arbitrarily. Note that all neighbors of v in G' precede it in η', so there are at most d of them. ■

Function η as in lemma 5.4.2 shall be called a *d-degeneracy ordering* of G, and the value $\eta(v)$ is the *position* of v. We remark here that, contrary to the usual definition, we do not

require η to be injective, but only to give different positions to endpoints of a single edge. Henceforth, we say that a function $\eta\colon Z \to [|V(G)|]$ for some $Z \subseteq V(G)$ is *edge-injective* if $\eta(u) \neq \eta(v)$ for every $uv \in E(G[Z])$.

Let G be an n-vertex $C_{>t}$-free graph and $\mathfrak{w}\colon V(G) \to \mathbb{N}$ be a weight function. Without loss of generality, assume $t \geqslant 6$. Fix a subset $S^* \subseteq V(G)$ such that $G[S^*]$ has degeneracy at most d and fix a d-degeneracy ordering $\eta^*\colon S^* \to [|S^*|]$ of $G[S^*]$. We think of S^* as of the intended optimum solution.

5.4.1 Recursion structure

A *subproblem* $\mathcal{R}$ consists of

- Two disjoint vertex sets $A^{\mathcal{R}}, X^{\mathcal{R}} \subseteq V(G)$. We additionally denote $R^{\mathcal{R}} \coloneqq V(G) - (A^{\mathcal{R}} \cup X^{\mathcal{R}})$ and call $R^{\mathcal{R}}$ the *free vertices*.
- An integer $\ell^{\mathcal{R}} \geqslant 0$, called the *level* of the subproblem.
- A set $W^{\mathcal{R}} \subseteq R^{\mathcal{R}}$ that is nonadjacent to $R^{\mathcal{R}} - W^{\mathcal{R}}$ and is of size less than $0.99^{-\ell^{\mathcal{R}}}$. The vertices of $W^{\mathcal{R}}$ are called the *active vertices*;
- An edge-injective function $\eta^{\mathcal{R}}\colon A^{\mathcal{R}} \to [n]$.
- A function $\zeta^{\mathcal{R}}\colon W^{\mathcal{R}} \to [n+1]$.

The superscript can be omitted if it is clear from the context.

In our recursive branching algorithms, one recursive call will treat one subproblem. The intended meaning of the components of the subproblem is as follows:

- $A^{\mathcal{R}}$ corresponds to the vertices already decided to be in the partial solution. The value $\eta^{\mathcal{R}}(v)$ for $v \in A^{\mathcal{R}}$ indicates the final position in the degeneracy ordering.
- $X^{\mathcal{R}}$ corresponds to the vertices already decided to be not in the partial solution.

- $R^{\mathcal{R}}$ are vertices yet to be decided.

Relying on some limited dependence between the connected components of $G[R^{\mathcal{R}}]$ in the studied problems, one recursive call focuses only on a group of these components, whose union of vertex sets is denoted by $W^{\mathcal{R}}$, and seeks for a best way to extend the current partial solution into $W^{\mathcal{R}}$. The integer $\zeta^{\mathcal{R}}(v)$ for $v \in W^{\mathcal{R}}$ indicates the minimum position at which the vertex v can be placed in the final degeneracy ordering.

This intuition motivates the following definition. A subproblem $\mathcal{R}$ is *lucky* if

- $A^{\mathcal{R}} \subseteq S^*$, $X^{\mathcal{R}} \cap S^* = \emptyset$;
- $\eta^{\mathcal{R}} = \eta^*|_{A^{\mathcal{R}}}$;
- for every $u \in S^* \cap W^{\mathcal{R}}$ we have $\zeta^{\mathcal{R}}(u) \leqslant \eta^*(u)$; and
- for every $u \in W^{\mathcal{R}}$ and every $v \in N_G(u) \cap A^{\mathcal{R}}$, if $\zeta^{\mathcal{R}}(u) \leqslant \eta^*(v)$, then $u \in S^*$ and $\eta^*(u) < \eta^*(v)$.

We will also need the following notion: for an edge-injective function $\eta\colon Z \to [n]$ for some $Z \subseteq V(G)$, $v \in V(G)$, and an integer p, the *quota* of v in η at position p is defined as

$$\gamma(\eta, v, p) = d - |\{u \in N_G(v) \cap Z \mid \eta(u) < p\}|.$$

If $v \in Z$, then $\gamma(\eta, v)$ is a shorthand for $\gamma(\eta, v, \eta(v))$. Intuitively, $\gamma(\eta, v)$ measures the number of "free slots" for the neighbors of v that precede it in η. Note that an edge-injective function $\eta : V(G) \to [n]$ is a d-degeneracy ordering of G if and only if $\gamma(\eta, v) \geqslant 0$ for every $v \in V(G)$. Furthermore, if $\eta' : Z' \to [n]$ is an extension of $\eta\colon Z \to [n]$ to some $Z' \supseteq Z$, then for every $v \in V(G)$ and $p \in [n]$, it holds that $\gamma(\eta, v, p) \geqslant \gamma(\eta', v, p)$.

Extending. For subproblems $\mathcal{R}_1$ and $\mathcal{R}_2$, we say that

- $\mathcal{R}_2$ *extends* $\mathcal{R}_1$ if
  - $A^{\mathcal{R}_1} \subseteq A^{\mathcal{R}_2}$, $X^{\mathcal{R}_1} \subseteq X^{\mathcal{R}_2}$ (and hence $R^{\mathcal{R}_1} \supseteq R^{\mathcal{R}_2}$);
  - $A^{\mathcal{R}_2} - A^{\mathcal{R}_1} \subseteq W^{\mathcal{R}_1}$ and $X^{\mathcal{R}_2} - X^{\mathcal{R}_1} \subseteq W^{\mathcal{R}_1}$;
  - $W^{\mathcal{R}_1} \supseteq W^{\mathcal{R}_2}$;
  - $\eta^{\mathcal{R}_1} = \eta^{\mathcal{R}_2}|_{A^{\mathcal{R}_1}}$;
  - for every $v \in W^{\mathcal{R}_2}$ we have $\zeta^{\mathcal{R}_2}(v) \geqslant \zeta^{\mathcal{R}_1}(v)$; and
  - for every $v \in A^{\mathcal{R}_2} \cap W^{\mathcal{R}_1}$ we have $\eta^{\mathcal{R}_2}(v) \geqslant \zeta^{\mathcal{R}_1}(v)$.
- $\mathcal{R}_2$ *extends* $\mathcal{R}_1$ *completely* if additionally $(A^{\mathcal{R}_2} - A^{\mathcal{R}_1}) \cup (X^{\mathcal{R}_2} - X^{\mathcal{R}_1}) = W^{\mathcal{R}_1}$ and $\ell^{\mathcal{R}_2} = 0$ (in particular, $W^{\mathcal{R}_2} = \emptyset$).

One easy way to extend a subproblem $\mathcal{R}$ is to select a set $Z \subseteq W^{\mathcal{R}}$ and move it to X; formally, the operation of *deleting* Z creates a new subproblem $\mathcal{R}'$ extending $\mathcal{R}$ by keeping all the ingredients the same, except for $X^{\mathcal{R}'} = X^{\mathcal{R}} \cup Z$, $W^{\mathcal{R}'} = W^{\mathcal{R}} - Z$, and $\zeta^{\mathcal{R}'} = \zeta^{\mathcal{R}}|_{W^{\mathcal{R}'}}$. Clearly, if $\mathcal{R}$ is lucky and $Z \cap S^* = \emptyset$, then $\mathcal{R}'$ is lucky as well.

A second (a bit more complicated way) to extend a subproblem $\mathcal{R}$ is the following. First, select a set $Z \subseteq W^{\mathcal{R}}$. Second, select an edge-injective function $\eta\colon A^{\mathcal{R}} \cup Z \to [n]$ extending $\eta^{\mathcal{R}}$ such that $\eta(u) \geqslant \zeta^{\mathcal{R}}(u)$ for every $u \in Z$; we henceforth call such a function a *position guess* for Z and $\mathcal{R}$. Third, for every $u \in Z$ select a set $D_u \subseteq W^{\mathcal{R}} - Z$ of size at most $\gamma(\eta, u)$. The family $(D_u)_{u \in Z}$ is called the *left neighbor guess* for Z, $\mathcal{R}$, and η. Finally, define the operation of *taking Z at positions η with left neighbors $(D_u)_{u \in Z}$* as constructing a new subproblem $\mathcal{R}'$ created from $\mathcal{R}$ by keeping all the ingredients the same, except for $A^{\mathcal{R}'} = A^{\mathcal{R}} \cup Z$, $W^{\mathcal{R}'} = W^{\mathcal{R}} - Z$, $\eta^{\mathcal{R}'} = \eta$, and, for every $w \in W^{\mathcal{R}'}$, taking

$$\zeta^{\mathcal{R}'}(w) = \max(\zeta^{\mathcal{R}}(w), \max\{1 + \eta(u)\colon u \in Z \wedge w \in N_G(u) - D_u\}).$$

We have the following immediate observation:

Lemma 5.4.3. *Let $\mathcal{R}$ be lucky and $Z \subseteq S^* \cap W^{\mathcal{R}}$. Then $\eta|_{A^{\mathcal{R}} \cup Z} \coloneqq \eta^*|_{A^{\mathcal{R}} \cup Z}$ is a position guess for $\mathcal{R}$ and Z and $(D_u)_{u \in Z}$ defined as*

$$D_u = \{w \in N_G(u) \cap S^* \cap W^{\mathcal{R}} \mid \eta^*(w) < \eta^*(u)\}$$

is a left neighbor guess for $\mathcal{R}$, Z, and η. Furthermore, the result of taking Z at positions η and left neighbors $(D_u)_{u \in Z}$ is lucky as well.

Filtering. We will need a simple filtering step. Consider a subproblem $\mathcal{R}$.

- A vertex $v \in A^{\mathcal{R}}$ is *offending* if there are more than d vertices $u \in N_G(v)$ such that either $u \in A^{\mathcal{R}}$ and $\eta^{\mathcal{R}}(u) < \eta^{\mathcal{R}}(v)$, or $u \in W^{\mathcal{R}}$ and $\zeta^{\mathcal{R}}(u) \leqslant \eta^{\mathcal{R}}(v)$.
- A vertex $v \in W^{\mathcal{R}}$ is *offending* if $\zeta^{\mathcal{R}}(v) > n$ or $\gamma(\eta^{\mathcal{R}}, v, \zeta^{\mathcal{R}}(v)) < 0$, where γ is defined w.r.t. $Z = A^{\mathcal{R}}$.
- The subproblem $\mathcal{R}$ is *clean* if there are no offending vertices.

We observe the following.

Lemma 5.4.4. *If in a subproblem $\mathcal{R}$ there is an offending vertex $v \in A^{\mathcal{R}}$ or $v \in W^{\mathcal{R}} \cap S^*$, then $\mathcal{R}$ is not lucky.*

Proof: For contradiction, suppose $\mathcal{R}$ is lucky. Assume first there is an offending vertex $v \in W^{\mathcal{R}} \cap S^*$. Since $\mathcal{R}$ is lucky, we have $\zeta^{\mathcal{R}}(v) \leqslant \eta^*(v) \leqslant n$. Furthermore,

$$\begin{aligned} d &\geqslant |\{u \in N_G(v) \cap S^* \mid \eta^*(u) < \eta^*(v)\}| \\ &\geqslant |\{u \in N_G(v) \cap A^{\mathcal{R}} \mid \eta^*(u) < \eta^*(v)\}| \\ &\geqslant |\{u \in N_G(v) \cap A^{\mathcal{R}} \mid \eta^*(u) < \zeta^{\mathcal{R}}(v)\}| \\ &= d - \gamma(\eta^{\mathcal{R}}, v, \zeta^{\mathcal{R}}(v)), \end{aligned}$$

which implies that $\gamma(\eta^{\mathcal{R}}, v, \zeta^{\mathcal{R}}(v)) \geqslant 0$. This contradicts the assumption that v is offending.

Assume now that there is an offending vertex $v \in A^{\mathcal{R}}$. Note that if $u \in N_G(v) \cap A^{\mathcal{R}}$ satisfies $\eta^{\mathcal{R}}(u) < \eta^{\mathcal{R}}(v)$, then $u \in S^*$ and $\eta^*(u) < \eta^*(v)$. Further, if $u \in N_G(v) \cap W^{\mathcal{R}}$ satisfies $\zeta^{\mathcal{R}}(u) \leqslant \eta^{\mathcal{R}}(v)$, then by the last item of the definition of being lucky we infer that $u \in S^*$ and $\eta^*(u) < \eta^*(v)$. Consequently, if v is offending, then $|\{u \in N_G(v) \cap S^* \mid \eta^*(u) < \eta^*(v)\}| > d$, a contradiction. ■

A *filtering step* applied to a subproblem $\mathcal{R}$ creates a subproblem $\mathcal{R}'$ that is the result of deleting all offending vertices $v \in W^{\mathcal{R}}$ from $\mathcal{R}$. By lemma 5.4.4 we infer that if $\mathcal{R}$ is lucky, then $\mathcal{R}'$ is lucky, too.

5.4.2 Subproblem tree

Note that subproblems of level 0 have necessarily $W^{\mathcal{R}} = \emptyset$. These subproblems shall correspond to the leaves of the recursion. Let us now proceed to the description of the recursion tree.

A *subproblem tree* is a rooted tree where every node x is labelled with a subproblem $\mathcal{R}(x)$ and is one of the following five types: leaf node, filter node, split node, branch node, and free node. We require that the root of the tree is labelled with a clean subproblem and that for every x and its child y, $\mathcal{R}(y)$ extends $\mathcal{R}(x)$.

For brevity, we say that x is clean if $\mathcal{R}(x)$ is clean. Similarly, we say that y (completely) extends x if $\mathcal{R}(y)$ (completely) extends $\mathcal{R}(x)$. Also, the level of x is the level of $\mathcal{R}(x)$.

Leaf node. A leaf node x has no children, is of level $\ell^{\mathcal{R}(x)} = 0$, and hence has $W^{\mathcal{R}(x)} = \emptyset$.

Filter node. A filter node x has no or one child. If $\mathcal{R}(x)$ contains an offending vertex $v \in A^{\mathcal{R}(x)}$, then x has no children. Otherwise, x contains a single child y with $\mathcal{R}(y)$ being the result of the filtering step applied to $\mathcal{R}(x)$. Note that the child of a filter node is always clean.

We additionally require that a parent of a filter node is not a filter node.

Split node. We say that a subproblem $\mathcal{R}$ is *splittable* if $\ell^{\mathcal{R}} \geqslant 1$ and every connected component of $G[W^{\mathcal{R}}]$ has size less than $0.99^{-\ell^{\mathcal{R}}+1}$. We observe that it is straightforward to split a splittable subproblem into a constant number of subproblems of smaller level.

Lemma 5.4.5. *If $\mathcal{R}$ is a splittable subproblem, then there exists a family $\mathcal{F}$ of one or two subproblems of level $\ell^{\mathcal{R}}-1$ that all extend $\mathcal{R}$, so that for every $\mathcal{Q} \in \mathcal{F}$ we have $A^{\mathcal{Q}} = A^{\mathcal{R}}$, $X^{\mathcal{Q}} = X^{\mathcal{R}}$, and $\eta^{\mathcal{Q}} = \eta^{\mathcal{R}}$, and furthermore $\{W^{\mathcal{Q}} \colon \mathcal{Q} \in \mathcal{F}\}$ is a partition of $W^{\mathcal{R}}$ with $\zeta^{\mathcal{Q}} = \zeta^{\mathcal{R}}|_{W^{\mathcal{Q}}}$ for every $\mathcal{Q} \in \mathcal{F}$.*

Proof: Let $C_1, \ldots, C_p$ be connected components of $G[W^{\mathcal{R}}]$ in a non-increasing order of their number of vertices. Since $\mathcal{R}$ is of level $\ell^{\mathcal{R}}$, it holds that $\sum_{i=1}^{p} |C_i| < 0.99^{-\ell^{\mathcal{R}}}$. Let j be the maximum index for which it holds that $\sum_{i=1}^{j} |C_i| < 0.99^{-\ell^{\mathcal{R}}+1}$. Then the assumption that $\mathcal{R}$ is splittable implies $j \geqslant 1$.

If $j = p$, then we can set $\mathcal{F}$ to be a singleton that contains a copy of $\mathcal{R}$. In particular, the set of active vertices remains unchanged. So now assume $j < p$. Since we ordered C_is in a non-increasing order of the number of vertices, the maximality of j implies that $\sum_{i=1}^{j} |C_i| \geqslant \frac{1}{2} \cdot 0.99^{-\ell^{\mathcal{R}}+1}$. Consequently,

$$\sum_{i=j+1}^{p} |C_i| < (1 - 0.495) \cdot 0.99^{-\ell^{\mathcal{R}}} < 0.99^{-\ell^{\mathcal{R}}+1}.$$

Hence, we can split $W^{\mathcal{R}}$ into $\bigcup_{i=1}^{j} C_i$ and $\bigcup_{i=j+1}^{p} C_i$ for the active sets of the subproblems of $\mathcal{F}$. Note that in this case $\mathcal{F}$ is of size two. ■

A split node x satisfies the following properties: $\mathcal{R}(x)$ is a splittable subproblem and the family of subproblems of children of x satisfy the properties promised by lemma 5.4.5 for the family $\mathcal{F}$. Moreover, we require that all the children of a split node are free nodes.

We remark that if a split node x is clean, then all the children of x are clean as well.

We also make the following observation.

Lemma 5.4.6. *Let x be a clean split node with children $\mathcal{Y}$. For every $y \in \mathcal{Y}$, let $\mathcal{R}'_y$ be a clean subproblem extending $\mathcal{R}(y)$ completely. Define a subproblem $\mathcal{R}'$ as*

$$A^{\mathcal{R}'} = \bigcup_{y\in\mathcal{Y}} A^{\mathcal{R}'_y}, \qquad X^{\mathcal{R}'} = \bigcup_{y\in\mathcal{Y}} X^{\mathcal{R}'_y}, \qquad \eta^{\mathcal{R}'} = \bigcup_{y\in\mathcal{Y}} \eta^{\mathcal{R}'_y},$$
$$W^{\mathcal{R}'} = \emptyset, \qquad \zeta^{\mathcal{R}'} = \emptyset, \qquad \ell^{\mathcal{R}'} = 0.$$

Then, $\mathcal{R}'$ is clean and extends completely $\mathcal{R}(x)$.

Proof: Most of the asserted properties follow from the definitions in a straightforward manner; here we only discuss the nontrivial ones.

First, note that $\eta^{\mathcal{R}'}$ is a well-defined function with domain $A^{\mathcal{R}'}$. This is because each $\eta^{\mathcal{R}'_y}$ extends $\eta^{\mathcal{R}(x)}$, and the sets $\{A^{\mathcal{R}'_y} - A^{\mathcal{R}(x)} \colon y \in \mathcal{Y}\}$ are pairwise disjoint, because $\{W^{\mathcal{R}(y)} \colon y \in \mathcal{Y}\}$ is a partition of $W^{\mathcal{R}(x)}$. Thus, $\mathcal{R}'$ is indeed a subproblem extending completely $\mathcal{R}(x)$. It remains to show that it is clean, that is, there are no offending vertices in $A^{\mathcal{R}'}$ (note here that $W^{\mathcal{R}'}$ is empty).

Consider first a vertex $v \in A^{\mathcal{R}'_y} - A^{\mathcal{R}(x)}$ for some $y \in \mathcal{Y}$. Then, since $W^{\mathcal{R}(y)}$ is nonadjacent to $R^{\mathcal{R}(y)} - W^{\mathcal{R}(y)}$, we have that $N_G(v) \subseteq W^{\mathcal{R}(y)} \cup A^{\mathcal{R}(x)} \cup X^{\mathcal{R}(x)}$. We infer that since v is not offending in $\mathcal{R}'_y$, due to this subproblem being clean, it is also not offending in $\mathcal{R}'$.

Consider now a vertex $v \in A^{\mathcal{R}(x)}$. Let $u \in N_G(v) \cap A^{\mathcal{R}'}$ be such that $\eta^{\mathcal{R}'}(u) < \eta^{\mathcal{R}'}(v)$. Then either $u \in A^{\mathcal{R}(x)}$ and $\eta^{\mathcal{R}(x)}(u) < \eta^{\mathcal{R}(x)}(v)$, or $u \in W^{\mathcal{R}(x)}$ and $\zeta^{\mathcal{R}(x)}(u) < \eta^{\mathcal{R}(x)}(v)$.

Since v is not offending in $\mathcal{R}(x)$, the number of such vertices u is bounded by d. Hence, v is not offending in $\mathcal{R}'$. This completes the proof of the lemma. ■

Branch node. Branch node x is additionally labelled with a *pivot* $\nu^x \in W^{\mathcal{R}(x)}$. Intuitively, in the branching we decide whether ν^x belongs to the constructed solution or not. In case we decide to include ν^x in A, we also guess its left neighbors and their positions in our fixed degeneracy ordering, as well as several other useful pieces of information.

Formally, a branch node x has a child y^x such that $\mathcal{R}(y^x)$ is the result of deleting ν^x from $\mathcal{R}(x)$. We call the edge pair xy^x of the subproblem tree the *failure branch* at node x, while y^x is the *failure child* of x. Note that if x is clean, then the failure child y^x is also clean.

Additionally, for every tuple $\mathcal{D} = (D, \eta, (D_u)_{u \in D'})$, where

- $D \subseteq W^{\mathcal{R}(x)} \cap N_G(\nu^x)$ is of size at most $\gamma(\eta^{\mathcal{R}(x)}, \nu^x, \zeta^{\mathcal{R}(x)}(\nu^x))$,
- η is a position guess for $\mathcal{R}$ and D', where $D' \coloneqq D \cup \{\nu^x\}$, and
- $(D_u)_{u \in D'}$ is a left neighbor guess for $\mathcal{R}$ and D' such that $D_{\nu^x} = \emptyset$,

the branch node x has a child $z^x_{\mathcal{D}}$. This child is associated with the subproblem $\mathcal{R}(z^x_{\mathcal{D}})$ that is the result of taking D' at positions η with left neighbors $(D_u)_{u \in D'}$ in the subproblem $\mathcal{R}$. We call each edge $xz^x_{\mathcal{D}}$ of the subproblem tree a *success branch* at node x, and $z^x_{\mathcal{D}}$ is a *success child* of x.

We remark that the children $z^x_{\mathcal{D}}$ may not be clean even if x is clean. Therefore, we require that every child $z^x_{\mathcal{D}}$ is a filter node. The child of $z^x_{\mathcal{D}}$, if present, is denoted as $s^x_{\mathcal{D}}$ and is called a *success grandchild* of x. We require that all the success grandchildren of x are free nodes.

Observe that for success children of a branch node, there are at most n^d choices for D, at most n^{d+1} choices for η, and at most n^d choices for each D_u, $u \in D$. This gives at

most $n^{d^2+2d+1} = n^{(d+1)^2}$ choices for the tuple $\mathcal{D}$, and consequently this is an upper bound on the number of success children.

Finally, we make the following observation that follows immediately from lemma 5.4.3:

Lemma 5.4.7. *Assume that a branch node x is lucky. If $\nu^x \notin S^*$, then the failure child y^x is lucky. Otherwise, if $\nu^x \in S^*$, then for*

$$D = \{u \in N_G(\nu^x) \cap W^{\mathcal{R}(x)} \cap S^* \mid \eta^*(u) < \eta^*(\nu^x)\},$$
$$\eta = \eta^*|_{A^{\mathcal{R}(x)} \cup D'},$$
$$D_u = \{w \in N_G(u) \cap W^{\mathcal{R}(x)} \cap S^* - D' \mid \eta^*(w) < \eta^*(u)\},$$

we have $|D| \leqslant d$, η is a position guess for $\mathcal{R}(x)$ and D', $(D_u)_{u \in D'}$ is a left neighbor guess for $\mathcal{R}(x)$, D', and η satisfying $D_{\nu^x} = \emptyset$, and $z^x_{\mathcal{D}}$ is a lucky success child of x.

Free node. For free nodes, we put three restrictions:

- a free node is a success grandchild of a branch node or a child of a split node;
- the children of a free node have the same level as the free node itself; and
- a child of a free node is clean or is a filter node.

Recall also the requirements stated in the above sections:

- Every child of a split node is a free node.
- Every success grandchild of a branch node is a free node.

Free nodes are essentially not used in the proof of theorem 5.4.1. Precisely, we use a trivial subroutine of handling them that just passes the same instance to the child, which

can be a node of a different type. Free nodes will play a role in further results, precisely in the proof of theorem 5.1.1, where they will be used as placeholders for additional branching steps, implemented using non-trivial subroutines for handling free nodes. The requirements stated above boil down to placing free nodes as children of split nodes and as success grandchildren of branch nodes, which means that the algorithm is allowed to perform the discussed additional guessing at these points. As we will see in the analysis, both passing through a split node and through a success branch correspond to some substantial progress in the recursion, which gives space for those extra branching steps.

Final remarks. The requirements that the root node is clean, that success children of a branch node are filter nodes, and that a child of a free node is either clean or a filter node, ensure the following property: every node of the subproblem tree is clean unless it is a filter node.

In a subproblem tree, the level of a child node equals the level of the parent, unless the parent node is a split node, in which case the level of a child is one less than the level of a parent. In particular, on a root-to-leaf path in a subproblem tree the levels do not increase.

Also, if y is a child of x, then $W^{\mathcal{R}(y)} \subseteq W^{\mathcal{R}(x)}$. Furthermore, $W^{\mathcal{R}(y)} = W^{\mathcal{R}(x)}$ can happen only when x is a split node, a filter node, or a free node. Taking into account the restrictions on the parents of filter and free nodes, we infer the following:

Lemma 5.4.8. *The depth of a subproblem tree rooted in a node x is at most $\mathcal{O}(|W^{\mathcal{R}(x)}| + \ell^{\mathcal{R}(x)})$.*

5.4.3 Branching strategy

A *branching strategy* is a recursive algorithm that, given a subproblem $\mathcal{R}$, creates a node x of a subproblem tree with $\mathcal{R}(x) = \mathcal{R}$, decides the type of x, appropriately

constructs subproblems for the children of x, and recurses on those children subproblems. The subproblem trees returned by the recursive calls are then attached by making their roots children of x. A branching strategy can pass some additional information to the child subcalls; for instance, after creating a split node, the call should ask all the subcalls to create free nodes, while a branch node should tell all its success children to be filter nodes with free nodes as their children. We require that the subproblem tree constructed by a branching strategy is a subproblem tree, as defined in the previous section.

A few remarks are in place. If the level of $\mathcal{R}$ is 0, the branching strategy has no option but to create a leaf node and stop (unless the parent or grandparent asks it to perform a filter or free node first). If a branching strategy makes a decision to create a filter node or a split node, there are no more decisions to make: the filtering step works deterministically, and for the split node we always create child subproblems using lemma 5.4.5.

If a branching strategy decides to make a branch node, the only remaining decision is to choose the pivot ν^x; after this, the failure branch and the success branches are defined deterministically. The method of choosing the pivot is the cornerstone of our combinatorial analysis, and is presented in the remainder of this section. (We will also use the freedom of sometimes *not* choosing to create a split node, even if the current subproblem is splittable.)

Finally, the definition of a subproblem tree allows only a limited freedom of creating free nodes: they have to be children of a split node or success grandchildren of a branch node. On the other hand, we would like to leave the freedom of how to handle free nodes to applications. Thus, a branching strategy is parameterized by a subroutine that handles free nodes: the subroutine is called by the branching strategy when handling a child of a split node or a success grandchild of a branch node, and asked to create the family of subproblems for children.

For theorem 5.4.1, we do not use the possibility of creating free nodes. Formally, the subroutine handling free nodes, invoked at a subproblem $\mathcal{R}$, returns a one-element family $\{\mathcal{R}\}$, that is, creates a dummy free node x with a single child y such that $\mathcal{R}(x) = \mathcal{R}(y)$.

An application of a branching strategy to a graph G is a shorthand for applying the branching strategy to a subproblem $\mathcal{R}$ defined by:

- $A^{\mathcal{R}} = X^{\mathcal{R}} = \emptyset$;
- $W^{\mathcal{R}} = V(G)$;
- $\ell^{\mathcal{R}} = \lceil -\log_{0.99}(|V(G)| + 1) \rceil$, and
- $\zeta^{\mathcal{R}}(v) = 1$ for every $v \in V(G)$.

Note that such a subproblem $\mathcal{R}$ is clean and lucky, regardless of the choice of S^* and η^*.

section 5.5 is devoted to branching strategies that lead to a quasipolynomial running time bounds. Formally, we prove there the following:

Lemma 5.4.9. *For every fixed pair of integers d and t, and every subroutine handling free nodes, there exists a branching strategy that, when applied to an n-vertex $C_{>t}$-free graph G, creates a recursion tree that is a subproblem tree such that on every root-to-leaf path there are $\mathcal{O}(\log n)$ split nodes and $\mathcal{O}(\log^4 n)$ success branches. If the input graph is P_t-free, the bound on the number of success branches on a single root-to-leaf path improves to $\mathcal{O}(\log^2 n)$.*

Furthermore, the branching strategy takes polynomial time to decide on the type of the node and on the choice of the branching pivot (in case the node is decided to be a branch node).

Let us now show how lemma 5.4.9 implies theorem 5.4.1 *Proof:* [of theorem 5.4.1.] Fix a set $S^* \subseteq V(G)$ maximizing $\mathfrak{w}(S^*)$ subject to $G[S^*]$ being of degeneracy at most d. Fix also a degeneracy ordering η^* of $G[S^*]$. This allows us to speak about lucky nodes.

Consider the branching strategy provided by lemma 5.4.9 with the discussed dummy subroutine handling free nodes.

By lemma 5.4.8, the depth of the subproblem tree is $\mathcal{O}(n)$. Furthermore, every node of the subproblem tree has at most $1 + n^{(d+1)^2}$ children, and the only nodes that may have more than one child are branch nodes and split nodes. Since each branch node has only one failure child, while on every root-to-leaf path there are $\mathcal{O}(\log n)$ split nodes and $\mathcal{O}(\log^4 n)$ success branches, it follows that the subproblem tree has $n^{\mathcal{O}(\log^4 n)}$ nodes.

For every clean node x in the generated subproblem tree, we would like to compute a clean subproblem $\mathcal{R}'(x)$ that extends $\mathcal{R}(x)$ completely and, subject to that, maximizes $\mathfrak{w}(A^{\mathcal{R}'(x)})$. Observe that for the answer to $\mathcal{R}'(r)$ at the root node r we can return $A^{\mathcal{R}'(r)}$ as the sought subset S: the cleanness of $\mathcal{R}'(r)$ implies that $G[A^{\mathcal{R}'(r)}]$ is d-degenerate, while a subproblem of level 0 with $A = S^*$, $X = V(G) - S^*$, and $\eta = \eta^*$ extends completely $\mathcal{R}(r)$.

For a leaf node x, the only option is to return $\mathcal{R}(x)$. For a split node x with children $\mathcal{Y}$, we apply lemma 5.4.6 to $\{\mathcal{R}'(y) \mid y \in \mathcal{Y}\}$, thus obtaining a clean subproblem $\mathcal{R}'(x)$. For a branch node x, we take $\mathcal{R}'(x)$ to be the subproblem with the maximum weight of A among subproblem $\mathcal{R}'(y^x)$ for the failure child y^x and subproblems $\mathcal{R}'(s^x_{\mathcal{D}})$ for success grandchildren $s^x_{\mathcal{D}}$. Note that both $\mathcal{R}'(y^x)$ and every $\mathcal{R}'(s^x_{\mathcal{D}})$ are clean subproblems that extending completely $\mathcal{R}(x)$. Finally, for a dummy free node x with a child y, $\mathcal{R}'(x)$ equals $\mathcal{R}'(y)$.

To finish the proof, it suffices to argue that for every lucky node x, we have

$$\mathfrak{w}(A^{\mathcal{R}'(x)}) \geqslant \mathfrak{w}(S^* \cap (A^{\mathcal{R}(x)} \cup W^{\mathcal{R}(x)})). \tag{5.2}$$

We prove this fact by a bottom-up induction on the subproblem tree.

For a lucky leaf node the claim is straightforward, as being lucky implies that $A^{\mathcal{R}(x)} \subseteq$

S^* while being a leaf node implies $W^{\mathcal{R}(x)} = \emptyset$. For a lucky split node, note that all children are also lucky; the claim is immediate by the inductive assumption for the children and the construction of lemma 5.4.6. For a lucky branch node x, lemma 5.4.7 implies that x has a lucky failure child or a lucky success grandchild; the claim follows from the inductive assumption of the said lucky (grand)child. Finally, for a lucky dummy free node, the claim is immediate by the inductive assumption on its child.

This finishes the proof of theorem 5.4.1. ■

5.5 Branching strategies: choosing pivots in P_t-free and $C_{>t}$-free graphs

This section is devoted to the proof of lemma 5.4.9. Recall that a branching strategy is essentially responsible for choosing whether the current node of the tree is a split node or a branch node and, in the latter case, choosing the branching pivot.

The following observation, immediate from the definition of a success branch, will be the basic source of progress.

Lemma 5.5.1. *Let x be a branch node and $y := s^x_{\mathcal{D}}$ be a successful grandchild of x, where $\mathcal{D} = (D, \eta, (D_u)_{u \in D'})$ and $D' = D \cup \{\nu^x\}$. Then, for every $u \in N_G(\nu^x)$, we have either $u \notin W^{\mathcal{R}(y)}$ or $\zeta^{\mathcal{R}(y)}(u) > \eta^{\mathcal{R}(y)}(\nu^x)$. Consequently, for every $u \in N_G(\nu^x) \cap W^{\mathcal{R}(y)}$, it holds that*

$$\gamma(\eta^{\mathcal{R}(y)}, u, \zeta^{\mathcal{R}(y)}(u)) \leqslant \gamma(\eta^{\mathcal{R}(x)}, u, \zeta^{\mathcal{R}(x)}(u)) - 1.$$

Proof: If $u \in W^{\mathcal{R}(y)}$, then in particular $y \notin D$. Then, in the definition of $\zeta^{\mathcal{R}(z^x_{\mathcal{D}})}(u)$, one term over which the maximum is taken is equal to $1 + \eta(\nu^x)$ and $\eta(\nu^x) = \eta^{\mathcal{R}(y)}(\nu^x)$. Thus, ν^x, which is in $W^{\mathcal{R}(x)} \cap A^{\mathcal{R}(y)}$, is taken into account in $\gamma(\eta^{\mathcal{R}(y)}, u, \zeta^{\mathcal{R}(y)}(u))$, but does not contribute to $\gamma(\eta^{\mathcal{R}(x)}, u, \zeta^{\mathcal{R}(x)}(u))$ due to being not included in $A^{\mathcal{R}(x)}$. ■

5.5.1 Quasipolynomial branching strategy in P_t-free graphs

To obtain the promised bounds for P_t-free graphs, we closely follow the arguments for MWIS from [41].

Let us fix d, t, and a subroutine handling free nodes.

The branching strategy chooses a leaf node whenever possible (the level of the subproblem is zero) and a split node whenever possible (the subproblem is splittable). Given a subproblem of positive level that is not splittable, the branching strategy creates a branch node.

For a P_t-free graph G and a pair $\{u,v\} \in \binom{V(G)}{2}$, the *bucket* of $\{u,v\}$ is a set $\mathcal{B}^G_{u,v}$ that consists of all induced paths with endpoints u and v. Two remarks are in place. First, $\mathcal{B}^G_{u,v} \neq \emptyset$ if and only if u and v are in the same connected component of G. Second, an n-vertex P_t-free graph has at most n^{t-1} induced paths; hence, all buckets can be enumerated in polynomial time.

For $\varepsilon > 0$, a vertex $x \in V(G)$ is *ε-heavy* if the neighborhood $N[x]$ intersects more that ε fraction of paths from more than ε fraction of buckets, that is,

$$\left|\left\{\{u,v\} \in \binom{V(G)}{2} \;\middle|\; \left|\{P \in \mathcal{B}^G_{u,v} \mid N[x] \cap V(P) \neq \emptyset\}\right| > \varepsilon |\mathcal{B}^G_{u,v}|\right\}\right| > \varepsilon \binom{|V(G)|}{2}.$$

Note that empty buckets cannot contribute towards the left hand side of the inequality above.

We make use of the following lemma.

Lemma 5.5.2 ([41]). *A connected P_t-free graph contains a $\frac{1}{2t}$-heavy vertex.*

In our case, a subproblem $\mathcal{R}$ at a branch node may not have $G[W^{\mathcal{R}}]$ connected, but $G[W^{\mathcal{R}}]$ contains a large connected component if $\mathcal{R}$ is not splittable.

Corollary 5.5.3. *If $\mathcal{R}$ is a not splittable subproblem of positive level, then $G[W^{\mathcal{R}}]$ contains a $\frac{1}{3t}$-heavy vertex.*

Proof: Recall that $n^{\mathcal{R}} := |W^{\mathcal{R}}| < 0.99^{-\ell^{\mathcal{R}}}$. Since $\mathcal{R}$ is of positive level but not splittable, there is a connected component D of $G[W^{\mathcal{R}}|$ of size at least $0.99^{-\ell^{\mathcal{R}}+1}$. That is, $|D| \geqslant 0.99n^{\mathcal{R}}$.

lemma 5.5.2 asserts there is a $\frac{1}{2t}$-heavy vertex in $G[D]$. We have

$$\binom{|D|}{2} \geqslant \binom{\lceil 0.99n^{\mathcal{R}} \rceil}{2} \geqslant 0.9\binom{n^{\mathcal{R}}}{2}.$$

Thus, x is $\frac{0.9}{2t}$-heavy in G and $\frac{0.9}{2t} > \frac{1}{3t}$. ■

The branching strategy chooses a $\frac{1}{3t}$-heavy vertex of $G[W^{\mathcal{R}(x)}]$ as a branching pivot ν^x at a branch node x. Recall that at a branch node x the subproblem $\mathcal{R}(x)$ is of positive level and not splittable. Hence, corollary 5.5.3 asserts the existence of such a pivot. As already discussed, all buckets can be enumerated in polynomial time and thus such a heavy vertex can be identified.

The bound on the number of split nodes on any root-to-leaf path in the subproblem tree is immediate from the fact that there are $\mathcal{O}(\log n)$ levels. It remains to argue that any root-to-leaf path has $\mathcal{O}(\log^2 n)$ success branches.

To this end, let Q be a maximal upward path in the subproblem tree that consists of nodes of the same level ℓ. As there are $\mathcal{O}(\log n)$ possible levels, it suffices to prove that Q contains $\mathcal{O}(\log n)$ success branches.

Consider then a success branch on Q: a branch node x and its success grandchild $y := s^x_{\mathcal{D}}$ for $\mathcal{D} = (D, \eta, (D_u)_{u \in D'})$, $D' = D \cup \{\nu^x\}$. lemma 5.5.1 suggests the following potential at node x:

$$\mu(x) \coloneqq \sum_{\{u,v\} \in \binom{W^{\mathcal{R}(x)}}{2}} \log_2 \left[1 + \sum_{P \in \mathcal{B}_{u,v}^{G[W^{\mathcal{R}(x)}]}} \sum_{u \in V(P)} (1 + \gamma(\eta^{\mathcal{R}(x)}, u, \zeta^{\mathcal{R}(x)}(u))) \right].$$

Since we branch on a $\frac{1}{3t}$-heavy vertex and the innermost sums in the definition above are bounded by $\mathcal{O}(dt)$, we have that for some universal constant c:

$$\mu(x) - \mu(y) \geqslant \frac{c}{dt^3} \binom{|W^{\mathcal{R}(x)}|}{3}.$$

Let x_0 be the topmost node of Q and $n_Q = |W^{\mathcal{R}(x_0)}|$. Since all nodes of x_0 are of the same level, we have $|W^{\mathcal{R}(x)}| \geqslant 0.99 n_Q$ for every branch node x on Q. We infer that

$$\mu(x) - \mu(y) = \Omega(n_P^2).$$

We have $\mu(x_0) = \mathcal{O}(n_P^2 \log n_P)$ while $\mu(x) \geqslant 0$ for any node x. Consequently, Q may contain $\mathcal{O}(\log n_P) = \mathcal{O}(\log n)$ success branches, as desired.

This finishes the proof of lemma 5.4.9 for P_t-free graphs.

5.5.2 Quasipolynomial branching strategy in $C_{>t}$-free graphs

We now prove lemma 5.4.9 for $C_{>t}$-free graphs. Hence, let us fix d, t, and a subroutine handling free nodes. W.l.o.g. assume t is even and $t \geqslant 6$. Recall that our goal is to design a branching strategy, which given a subproblem $\mathcal{R}$ should decide the type of the node created for $\mathcal{R}$ and, in case this type is the branch node, choose a suitable branching pivot. We will measure the progress of our algorithm by keeping track of the number of some suitably defined objects in the graph.

A *connector* is a graph with three designated vertices, called *tips*, obtained in the fol-

lowing way. Take three induced paths Q_1, Q_2, Q_3; here we allow degenerated, one-vertex paths. The paths Q_1, Q_2, Q_3 will be called the *legs* of the connector. The endvertices of Q_i are called a_i and b_i. Now, join these paths in one of the following ways:

a) identify a_1, a_2, and a_3 into a single vertex, i.e., $a_1 = a_2 = a_3$, or

b) add edges a_1a_2, a_2a_3, and a_1a_3.

Furthermore, if the endvertices are identified, then at most one leg may be degenerated; thus, b_1, b_2, b_3 are pairwise different after the joining. There are no other edges between the legs of the connector. The vertices b_1, b_2, and b_3 are the *tips* of the connector, and the set $\{a_1, a_2, a_3\}$ is called the *center* (this set can have either three or one element); see fig. 5.1. If one of the paths forming the connector has only one vertex, and the endvertices were identified, then the connector is just an induced path with the tips being the endpoints of the path plus one internal vertex of the path. Note that, given a connector as a graph and its tips, the legs and the center of the connector are defined uniquely.

We will need the following folklore observation.

Lemma 5.5.4. *Let G be a graph, $A \subseteq V(G)$ be a set consisting of exactly three vertices in the same connected component of G, and let $A \subseteq B \subseteq V(G)$ be an inclusion-wise minimal set such that $G[B]$ is connected. Then the graph $G[B]$ with the set A as tips is a connector.*

Proof: Let $A = \{u, v, w\}$, let P_{uv} be a shortest path from u to v in $G[B]$ and let P_w be a shortest path from w to $V(P_{uv})$ in $G[B]$. By minimality of B, we have $B = V(P_{uv}) \cup V(P_w)$.

If $w \in V(P_{uv})$ (equivalently, $|V(P_w)| = 1$), then $G[B]$ is a path and we are done. Otherwise, let $q \in V(P_w) \cap V(P_{uv})$ be the endpoint of P_w distinct than w and let p be

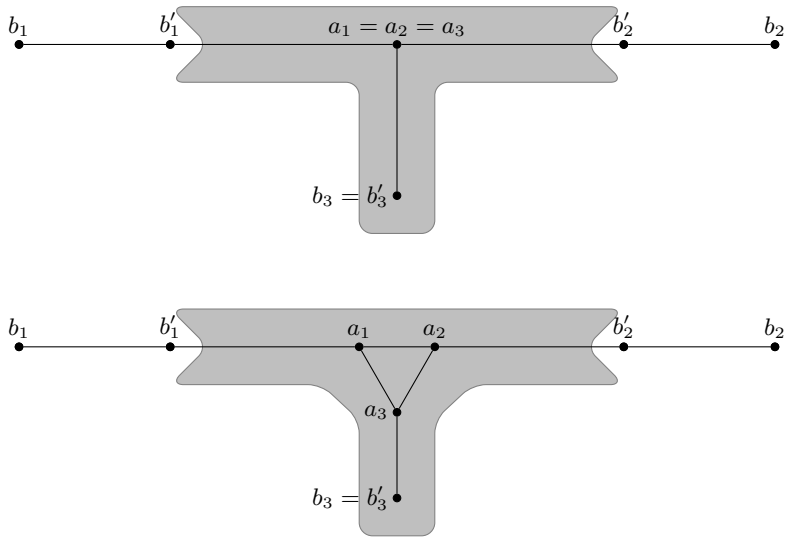


Figure 5.1: Two connectors with two long legs and one short leg; one connector with a_is identified and one with a_is forming a triangle. Vertices b'_i are the tips of the core T of the connector. The gray area depicts the set T^*.

the unique neighbor of q on P_w. By the minimality of P_w, p and q are the only vertices of P_w that may have neighbors on P_{uv}. If p has two neighbors $x, y \in V(P_{uv})$ that are not consecutive on P_{uv}, then $G[B]$ remains connected after the deletion from B of all vertices on P_{uv} between x and y (exclusive), a contradiction to the choice of B. Thus, $N(p) \cap V(P_{uv})$ consists of q and possibly one neighbor of q on P_{uv}. We infer that $G[B]$ is a connector with tips u, v, and w, as desired. ■

A *tripod* is a connector where each of the paths Q_1, Q_2, Q_3 has at most $t/2+1$ vertices. A leg of a tripod is *long* if it contains exactly $t/2+1$ vertices, and *short* otherwise. The *core* of a connector C with legs Q_1, Q_2, Q_3, is the tripod consisting of the first $t/2+1$ (or all of them, if the corresponding path Q_i is shorter) vertices of each path Q_i, starting from a_i. A tripod *in* G is a tripod that is an induced subgraph of G. Note that each tripod has at most $3t/2+3$ vertices, hence given G, we can enumerate all tripods in G in time $n^{\mathcal{O}(t)}$.

Let T be a tripod in G with legs Q_1, Q_2, Q_3. Let $L(T)$ denote the tips of T which are

endvertices of the long legs. We denote

$$T^* \coloneqq N_G[V(T) - L(T)] - L(T).$$

In other words, T^* is the closed neighborhood of the tripod T in G, except that we do not necessarily include the neighbors of tips of long legs and we exclude those tips themselves. Note that tips of short legs together with their neighborhoods are included in T^*.

We have the following simple observation.

Lemma 5.5.5. *For every tripod T in G and for every connected component C of $G - T^*$, the component C contains at most one tip of $L(T)$ and no tip of a short leg.*

Proof: First, note that tips of short legs are contained in T^*, hence they are not contained in $G - T^*$.

For contradiction, without loss of generality assume that $b_1, b_2 \in C \cap L(T)$. Then, Q_1, Q_2, and a shortest path from b_1 to b_2 in C yield an induced cycle on more than t vertices in G, a contradiction. ■

Suppose T is a tripod in G. With each tip b_i of T we associate a *bag* B_i defined as follows:

- if b_i is the endpoint of a long leg, then B_i is the vertex set of the connected component of $G - T^*$ that contains b_i; and
- otherwise, $B_i = \{b_i\}$.

Note that lemma 5.5.5 implies that the bags B_1, B_2, B_3 are pairwise disjoint and nonadjacent in G, except for the corner case of two adjacent tips of short legs.

We now define *buckets* that group tripods. Each bucket will be indexed by an unordered triple of distinct vertices of G. Every tripod T in G with bags B_1, B_2, B_3 belongs

to the bucket $\mathcal{B}^G_{u,v,w}$ for all triples u, v, w such that $u \in B_1$, $v \in B_2$, and $w \in B_3$. Note that thus, the buckets do not have to be pairwise disjoint. The superscript G can be omitted if the graph is clear from the context.

Observe the following.

Lemma 5.5.6. *For every $\{u, v, w\} \in \binom{V(G)}{3}$ and every $T \in \mathcal{B}_{u,v,w}$, there exists a connector T' with tips u, v, w whose core equals T. Consequently, the bucket $\mathcal{B}_{u,v,w}$ is nonempty if and only if u, v, w lie in the same connected component of G.*

Proof: Let Q_u, Q_v, Q_w be the legs of T with tips b_u, b_v, b_w and bags B_u, B_v, and B_w, respectively, such that $u \in B_u$, $v \in B_v$, $w \in B_w$. Let Q'_u be the concatenation of Q_u and a shortest path from b_u to u in $G[B_u]$. Similarly define Q'_v and Q'_w.

Recall that the bags B_u, B_v, and B_w are pairwise distinct and nonadjacent (except for the case of two adjacent tips of short legs). Hence, Q'_u, Q'_v, and Q'_w form a connector T' with tips u, v, and w. Since $B_u \neq \{b_u\}$ only if the leg Q_u is long, T is the core of T'. ■

The next combinatorial observation is critical to the complexity analysis.

Lemma 5.5.7. *Let G be a connected $C_{>t}$-free graph, and let u, v, w be three distinct vertices of G. Let $X \subseteq V(G)$ be such that $G[X]$ is connected and no two of u, v, w are in the same connected component of $G - N[X]$. Then $N[X]$ intersects all tripods in $\mathcal{B}_{u,v,w}$.*

Proof: Let $T \in \mathcal{B}_{u,v,w}$. Let Q_u, Q_v, Q_w be the legs of T with tips b_u, b_v, b_w and bags B_u, B_v, and B_w, respectively. Let T' be the connector for T given by lemma 5.5.6 with legs Q'_u, Q'_v, and Q'_w. Since no two of u, v, w lie in the same connected component of $G - N[X]$, the set $N[X]$ intersects at least two legs of Q'_u, Q'_v, and Q'_w. Without loss of generality, assume that $N[X]$ intersects Q'_u and Q'_v. Let $u' \in N[X] \cap V(Q'_u)$ be the vertex of $N[X] \cap V(Q'_u)$ that is farthest from u on Q'_u and similarly define $v' \in N[X] \cap V(Q'_v)$.

Then, the subpath of Q'_u from u' to the center of T', the subpath of Q'_v from v' to the center of T', and a shortest path from u' to v' with all internal vertices in X form an induced cycle in G. This cycle has more than t vertices unless $u' \in V(Q_u)$ or $v' \in V(Q_v)$. Hence, $V(T) \cap N[X] \neq \emptyset$, as desired. ■

For $\varepsilon > 0$, we will say that a vertex v is ε-heavy (or just *heavy*, if ε is clear from the context) if $N[v]$ intersects strictly more than ε-fraction of tripods in at least ε-fraction of buckets. Note that the "strictly more" part makes empty buckets not count toward the ε-fraction of the buckets hit.

Intuitively, our branching strategy chooses as the pivot a $\frac{1}{10t}$-heavy vertex of $G[W^{\mathcal{R}}]$. lemma 5.5.1 captures the source of the gain in a success branch: the quotas of the neighbors of the pivot get reduced.

Unfortunately, it may happen that there is no $\frac{1}{10t}$-heavy vertex. For an example, consider the case when $H \coloneqq G[W^{\mathcal{R}}]$ is a long path. Then $\mathcal{B}_{u,v,w}$ consists of subpaths of H of length at most t containing the middle vertex of $\{u, v, w\}$, and an arbitrary neighborhood $N[x]$ intersects tripods in roughly $t/|V(H)|$ fraction of all buckets. However, lemma 5.5.8 below shows a setting where a heavy vertex is guaranteed to exist.

Lemma 5.5.8. *Let G be a connected $C_{>t}$-free graph and let $X \subseteq V(G)$ be a set of size at most t such that $G[X]$ is connected and every connected component of $G - N[X]$ has at most $0.1 \cdot |V(G)|$ vertices. Then there exists a $1/(2t)$-heavy vertex in G.*

Proof: Let $n = |V(G)|$. Assume $n \geqslant 5$, as otherwise $N[X] = V(G)$ and the statement is trivial. The number of buckets $\mathcal{B}_{u,v,w}$ such that at least two of u, v, w are in the same

connected component of $G - N[X]$ is upper bounded by

$$\begin{aligned}
&\sum_{C\text{: component of } G-N[X]} \binom{|V(C)|}{3} + \binom{|V(C)|}{2} \cdot (n - |V(C)|) \\
&\leqslant n \cdot \left(\frac{(0.1n-1)(0.1n-2)}{6} + \frac{(0.1n-1)\cdot 0.9n}{2} \right) \\
&\leqslant n \cdot 0.1 \cdot \frac{n-1}{2} \cdot \frac{2.8n-2}{3} \\
&= \frac{1}{2} \cdot n \cdot \frac{n-1}{2} \cdot \frac{0.56n-0.4}{3} \\
&< \frac{1}{2}\binom{n}{3}.
\end{aligned}$$

Here, the last inequality uses $n \geqslant 5$. Hence, by lemma 5.5.7, the set $N[X]$ intersects all tripods in at least half of the buckets. Since $|X| \leqslant t$, there exists $w \in X$ such that $N[w]$ intersects at least $1/t$ fraction of tripods in at least $1/(2t)$ fraction of the buckets. ■

Unfortunately, theorem 5.2.3 for $A = V(G)$ gives us only a connected set X of size at most t such that every component of $G - N[X]$ has at most $|V(G)|/2$ vertices. The example of a long path shows that the fraction $1/2$ cannot be improved while keeping X both connected and of constant size.

In the absence of a heavy vertex, we shift to a secondary branching strategy. To describe the properties of this strategy, let us first make the following definition. For a subproblem $\mathcal{R}$ and a subset $K \subseteq W^{\mathcal{R}}$, we say that a subproblem $\mathcal{R}'$ *considerably extends* $(\mathcal{R}, K)$ if $\mathcal{R}'$ extends $\mathcal{R}$, is of the same level, and at least one of the following properties holds:

- every connected component of $H' \coloneqq G[W^{\mathcal{R}'}]$ is of size at most $0.99|W^{\mathcal{R}}|$, or
- every connected component C of $H' - K$ with more than $0.01|W^{\mathcal{R}}|$ vertices satisfies $N_{H'}[C] \cap K = \emptyset$.

The secondary branching strategy:

(1) is initiated with a subproblem $\mathcal{R}$ and a set $X \subseteq W^{\mathcal{R}}$ such that $G[X]$ is connected; for convenience, denote $n^{\mathcal{R}} \coloneqq |W^{\mathcal{R}}|$, $H \coloneqq G[W^{\mathcal{R}}]$, and $K \coloneqq N_H[X]$;

(2) terminates (i.e., falls back to the primary branching strategy) if and only if when called at a subproblem $\mathcal{R}'$ that considerably extends $(\mathcal{R}, K)$;

(3) never forms a split node (that is, at every node either terminates or selects a branching pivot, makes a branch node and successive filter and free nodes for success branches);

(4) on every root-to-leaf path in the subproblem tree created by the recursion there are $\mathcal{O}(\log^2 n^{\mathcal{R}})$ success branches.

Note that, in particular, all subproblems in a subproblem tree created by the secondary branching strategy are of the same level.

We postpone the description of the secondary branching strategy to section 5.5.2. Now, using it as a blackbox, we describe our primary strategy.

For a subproblem $\mathcal{R}$, the primary branching strategy makes the following decisions:

1. If $\mathcal{R}$ is of level 0, make a leaf node and terminate.

2. If $\mathcal{R}$ is splittable, make a split node using lemma 5.4.5 and recurse on the constructed children.

3. If $G[W^{\mathcal{R}}]$ contains a $1/(10t)$-heavy vertex w, create a branch node x and choose w as the branching pivot ν^x.

4. Otherwise, construct a set X from theorem 5.2.3 for the graph $G[W^{\mathcal{R}}]$ and invoke the secondary branching strategy on $\mathcal{R}$ and X.

The crucial observation is the following.

Lemma 5.5.9. *Let $\mathcal{R}$ be a subproblem such that $H \coloneqq G[W^{\mathcal{R}}]$ does not contain a $1/(10t)$-heavy vertex. Let X be a result of an application of theorem 5.2.3 to H and let $K = N_H[X]$. Let $\mathcal{R}'$ be a subproblem considerably extending $(\mathcal{R}, K)$ and let $H' \coloneqq G[W^{\mathcal{R}'}]$. Then:*

a) If every connected component of H' has less than $0.99|W^{\mathcal{R}}|$ vertices, then $\mathcal{R}'$ is splittable.

b) Otherwise, there is subset $F \subseteq \binom{V(H')}{3}$ of size at least $\frac{1}{10t} \cdot \binom{|V(H)|}{3}$ such that for every $\{u, v, w\} \in F$, the size of the bucket $\mathcal{B}^{H'}_{u,v,w}$ is of size less than one-fifth of the size of $\mathcal{B}^{H}_{u,v,w}$ (in particular, $\mathcal{B}^{H}_{u,v,w} \neq \emptyset$).

Proof: Let $n^{\mathcal{R}} \coloneqq |W^{\mathcal{R}}|$. Assume first that every connected component of H' has fewer than $0.99n^{\mathcal{R}}$ vertices. By assumption, $n^{\mathcal{R}} < 0.99^{-\ell^{\mathcal{R}}}$. Hence, every connected component of H' has fewer than $0.99^{-\ell^{\mathcal{R}}+1}$ vertices. As $\ell^{\mathcal{R}} = \ell^{\mathcal{R}'}$, $\mathcal{R}'$ is splittable.

Assume now that there is a connected component D' of H' that has at least $0.99n^{\mathcal{R}}$ vertices. Since $\mathcal{R}'$ considerably extends $(\mathcal{R}, K)$, every connected component C of $H' - K$ that has more than $0.01n^{\mathcal{R}}$ vertices satisfies $N_{H'}[C] \cap K = \emptyset$.

Let $H'' = G[W^{\mathcal{R}'} \cup X] = G[V(H') \cup X]$. Since every connected component of $H - K$ has at most $n^{\mathcal{R}}/2$ vertices, $H[X]$ is connected, while D' has at least $0.99n^{\mathcal{R}}$ vertices, we infer that there is a connected component D of H'' that contains both X and D'. Observe that since D is a connected component of H'' and H'' is an induced subgraph of H, it follows that $\mathcal{B}^{D}_{u,v,w} = \mathcal{B}^{H''}_{u,v,w} \subseteq \mathcal{B}^{H}_{u,v,w}$.

We have $K \cap V(H'') = N_{H''}[X]$, $H' - K = H'' - N_{H''}[X]$, and for every connected component C of $H'' - N_{H''}[X]$ we have $N_{H'}[C] = N_{H''}[C]$. Hence, every connected component C of $H'' - N_{H''}[X]$ that has more than $0.01n^{\mathcal{R}}$ vertices satisfies $N_{H''}[C] \cap N_{H''}[X] = \emptyset$, Consequently, every connected component C of $D - N_{H''}[X]$ contains at most $0.01n^{\mathcal{R}}$ vertices. Since $0.01n^{\mathcal{R}} < 0.1|V(D)|$, lemma 5.5.8 implies that there is a set

$F_0 \subseteq \binom{V(D)}{3}$ of size at least $\frac{1}{2t}\binom{|V(D)|}{3}$ and a vertex $x \in V(D)$ such that $N_D[x]$ intersects at least a fraction of $\frac{1}{2t}$ tripods from every bucket $\mathcal{B}^D_{u,v,w} = \mathcal{B}^{H''}_{u,v,w}$ for $\{u,v,w\} \in F_0$. As $|V(D)| \geqslant 0.99n^{\mathcal{R}}$, we have that $|F_0| \geqslant \frac{1}{4t}\binom{n^{\mathcal{R}}}{3}$.

Note that $N_D[x] = N_{H''}[x] \subseteq N_H[x]$. Hence, for fixed $\{u,v,w\} \in F_0$, every tripod hit by $N_{H''}[x]$ in $\mathcal{B}^{H''}_{u,v,w}$ is also hit by $N_H[x]$ in $\mathcal{B}^H_{u,v,w}$. However, x is not $1/(10t)$-heavy in H, because we assumed that H has no $1/(10t)$-heavy vertices. This implies that there is a set $F \subseteq F_0$ of size at least

$$\left(\frac{1}{4t} - \frac{1}{10t}\right)\binom{n^{\mathcal{R}}}{3} \geqslant \frac{1}{10t}\binom{n^{\mathcal{R}}}{3}$$

such that for every $\{u,v,w\} \in F$, the neighborhood $N_H[x]$ hits less than a $\frac{1}{10t}$-fraction of tripods in $\mathcal{B}^H_{u,v,w}$. However, for $\{u,v,w\} \in F$, $N_{H''}[x]$ hits a $\frac{1}{2t}$-fraction of tripods in $\mathcal{B}^{H''}_{u,v,w}$ and every tripod hit by $N_{H''}[x]$ in H'' is hit by $N_H[x]$ in H. Consequently, for every $\{u,v,w\} \in F$ we have $|\mathcal{B}^H_{u,v,w}| > 5|\mathcal{B}^{H''}_{u,v,w}| \geqslant 5|\mathcal{B}^{H'}_{u,v,w}|$, as desired. ■

With lemma 5.5.9 established, we can proceed with the analysis. Consider the subproblem tree of the algorithm applied to the graph G. The claim that every root-to-leaf path contains $\mathcal{O}(\log n)$ split nodes is straightforward, because the root node has level $\lceil -\log_{0.99}(n+1)\rceil = \mathcal{O}(\log n)$ and the level of the split node is one higher than the level of its children. To show the more difficult claim that every root-to-leaf path contains $\mathcal{O}(\log^4 n)$ success branches, it suffices to show that any upward path in the subproblem tree consisting of nodes of the same level ℓ contains $\mathcal{O}(\log^3 n)$ success branches.

Consider such a maximal upward path P with all nodes of level ℓ. On the path P we can distinguish subpaths that are maximal subpaths contained in a subtree of the subproblem tree corresponding to one use of the secondary branching strategy; we henceforth call them *secondary subpaths*. Because of the maximality of P, for every secondary subpath Q, the topmost vertex of Q is a node where the secondary branching

strategy has been invoked, whereas the bottom-most vertex of Q is where the secondary branching strategy decided to terminate.

Recall that we promised that every secondary subpath contains $\mathcal{O}(\log^2 n)$ edges that are success branches (this promise will be fulfilled in the next section). Hence, it suffices to show that P contains:

- $\mathcal{O}(\log n)$ edges that are success branches at branch nodes of the primary branching strategy, and
- $\mathcal{O}(\log n)$ secondary subpaths.

Let $n_P = |W^{\mathcal{R}(r)}|$ where r is the topmost node of P. By the threshold at which we use split node in the primary strategy and the termination condition for the secondary strategy, for every x on P except for possibly the bottom-most one, we have

$$|W^{\mathcal{R}(x)}| \geqslant 0.99 n_P. \tag{5.3}$$

Let $\mathcal{X}$ be the family of pairs (x, y) of nodes of P such that either y is a success grandchild of x or the secondary branching strategy has been invoked at x and terminated at y (i.e., x and y are endpoints of a secondary subpath). Our goal is to show that $|\mathcal{X}| = \mathcal{O}(\log n)$.

Consider a pair $(x, y) \in \mathcal{X}$ that corresponds to a secondary subpath such that $\mathcal{R}(y)$ considerably extends $(\mathcal{R}(x), K)$, because every connected component of $G[W^{\mathcal{R}(y)}]$ is of size at most $0.99|W^{\mathcal{R}(x)}|$. By lemma 5.5.9, $\mathcal{R}(y)$ is splittable. Consequently, all children of y are of lower level and y is the bottom-most node of P. Hence, $\mathcal{X}$ contains at most one such pair; let us exclude it from $\mathcal{X}$ and further consideration.

For every remaining pair $(x, y) \in \mathcal{X}$ that corresponds to a secondary subpath, lemma 5.5.9 implies that there are at least $\frac{1}{10t}\binom{|W^{\mathcal{R}(x)}|}{3}$ nonempty buckets of $G[W^{\mathcal{R}(x)}]$ that in $G[W^{\mathcal{R}(y)}]$ have size decreased by a multiplicative factor of at least $1/5$.

On the other hand, consider any pair $(x, y) \in \mathcal{X}$ that does not correspond to a secondary path, i.e. x is a branch node and y is its success grandchild, and recall that this corresponds to branching on a $\frac{1}{10t}$-heavy pivot ν^x. Motivated by lemma 5.5.1, we measure the progress through the following potential:

$$\mu(x) \coloneqq \sum_{\{u,v,w\} \in \binom{W^{\mathcal{R}(x)}}{3}} \log_2 \left[1 + \sum_{T \in \mathcal{B}_{u,v,w}^{G[W^{\mathcal{R}(x)}]}} \sum_{u \in V(T)} (1 + \gamma(\eta^{\mathcal{R}(x)}, u, \zeta^{\mathcal{R}(x)}(u))) \right].$$

Observe that when y is a child of x on P, we have $\mu(x) \geqslant \mu(y)$. Let us now analyze the drop of this potential for each $(x, y) \in \mathcal{X}$.

Consider first the case when (x, y) does not correspond to a secondary path, i.e., x is a branch node and y is its success grandchild. As the branching is performed on a $\frac{1}{10t}$-heavy pivot, while each of the innermost sums is upper bounded by $\mathcal{O}(dt)$, by lemma 5.5.1 we may infer that

$$\mu(x) - \mu(y) \geqslant \frac{c_1}{dt^3} \binom{|W^{\mathcal{R}(x)}|}{3},$$

for some universal constant $c_1 > 0$.

Consider now the case when (x, y) corresponds to a secondary path. As argued, then there are at least $\frac{1}{10t}\binom{|W^{\mathcal{R}(x)}|}{3}$ nonempty buckets of $G[C^{\mathcal{R}(x)}]$ whose sizes are decreased by at least one-fifth in $G[C^{\mathcal{R}(y)}]$. Note that if this happens to a bucket corresponding to $\{u, v, w\} \in \binom{W^{\mathcal{R}(x)}}{3}$, then the corresponding double-sum under the logarithm decreases by a multiplicative fraction of at least $\frac{1}{\mathcal{O}(dt)}$, because the innermost sums are upper bounded by $\mathcal{O}(dt)$ and lower bounded by 1. We again infer that in this case,

$$\mu(x) - \mu(y) \geqslant \frac{c_2}{dt^2} \binom{|W^{\mathcal{R}(x)}|}{3},$$

for some universal constant $c_2 > 0$.

By (5.3), we conclude that in both cases, $\mu(x) - \mu(y) = \Omega(n_P^3)$. On the other hand, since every bucket is of size $n^{\mathcal{O}(t)}$, we have that $\mu(x) = \mathcal{O}(n_P^3 \log n_P)$ for every x on P. Hence $|\mathcal{X}| = \mathcal{O}(\log n_P)$, as desired.

Secondary branching strategy

We now move to the explanation of the secondary branching strategy.

For a graph H, a set $C \subseteq V(H)$ such that $H[C]$ is connected, and distinct vertices $u, v \in N_H(C)$, a *C-link* between u and v is a path P in H with the following properties:

- P has endpoints u and v and length at least 2;
- all internal vertices of P belong to C; and
- P is an induced path in $H - E(H[N_H(C)])$ (i.e. P is an induced path in H, except that we allow the existence of the edge uv).

We now make a combinatorial observation that is critical to the analysis:

Lemma 5.5.10. *Let H be a $C_{>t}$-free graph, let $X \subseteq V(H)$ be such that $H[X]$ is connected, and let C be a connected component of $H - N[X]$. Then every C-link has at most t vertices.*

Proof: Let P be the C-link in question, let u, v be its endpoints, and let Q be a shortest path with endpoints u, v and all internal vertices in X; such Q exists because $u, v \in N[X]$ and $H[X]$ is connected. Then $P \cup Q$ is an induced cycle in H. Thus, both P and Q have at most t vertices. ■

Recall that the setting of the secondary branching strategy is as follows: we have a subproblem $\mathcal{R}$ and a set $X \subseteq W^{\mathcal{R}}$ such that $G[X]$ is connected. Let $n^{\mathcal{R}} = |W^{\mathcal{R}}|$ and $K = N_{G[W^{\mathcal{R}}]}[X]$.

Recall also that the secondary branching strategy is allowed only to make branch nodes (and subsequent filter and free nodes at success branches) and it always terminates whenever the current subproblem considerably extends $(\mathcal{R}, K)$. The crux is to describe the choice of the branching pivot when the current subproblem does not considerably extend $\mathcal{R}$.

We fix a threshold $\tau \coloneqq 0.01n^{\mathcal{R}}$. For an induced subgraph H of $G[W^{\mathcal{R}}]$, a *chip* in H is the vertex set C of a connected component of $H - K$ satisfying the following properties: C contains more than τ vertices and $N_H[C] \cap K \neq \emptyset$.

Assume that we are now considering a subproblem $\mathcal{R}'$ extending $\mathcal{R}$. Observe that if $H \coloneqq G[W^{\mathcal{R}'}]$ has no chip, then $\mathcal{R}'$ considerably extends $(\mathcal{R}, K)$ and the secondary branching strategy terminates.

If H contains a chip C with $|N_H(C)| = 1$, we choose the unique element of $N_H(C)$ as the branching pivot. Note that after branching on such a pivot, for every child z of the current branching node (both in the failure and in the success branches) (the remainder of) the chip C in $G[W^{\mathcal{R}(z)}]$ is in a different connected component than any vertex of K. If $|C| \leqslant 0.99n^{\mathcal{R}}$, then every connected component in $G[W^{\mathcal{R}(z)}]$ has at most $0.99n^{\mathcal{R}}$ vertices due to $|C| \geqslant \tau = 0.01n^{\mathcal{R}}$, and if $|C| > 0.99n^{\mathcal{R}}$, then $G[W^{\mathcal{R}(z)}]$ contains no chips. Hence, the secondary branching strategy terminates both at the failure child, and at all success grandchildren of the current branch node.

We are left with the following case: in H, there is at least one chip and every chip C in H satisfies $|N_H(C)| \geqslant 2$. We define (secondary branching) buckets as follows. The buckets are indexed by a pair consisting of a vertex $w \in V(H)$ and an unordered pair $\{u, v\} \in \binom{K \cap V(H)}{2}$. For such a choice of w, u, v, the bucket $\mathcal{L}_{w;u,v}$ contains all C-links with endpoints u and v where C is a chip in H and $w \in V(C)$. lemma 5.5.10 ensures that every link in a bucket has at most t vertices and, consequently, every bucket has size $\mathcal{O}(n^t)$ and can be enumerated in polynomial time. Note that $\mathcal{L}_{w;u,v}$ is nonempty if

and only if there exists a chip C with $w \in V(C)$ and $u, v \in N_H(C)$.

For $\varepsilon > 0$, a vertex $x \in V(H)$ is *ε-heavy* if $N_H[x]$ intersects strictly more than an ε fraction of links in at least an ε fraction of nonempty buckets. We prove the following.

Lemma 5.5.11. *If there exists a nonempty bucket, then there exists a $\frac{1}{200t}$-heavy vertex.*

Proof: Since there are at most $n^{\mathcal{R}}/\tau = 100$ chips, pick a chip C such that for at least a fraction of 0.01 nonempty buckets $\mathcal{L}_{w;u,v}$ we have $w \in V(C)$. Let $H_C \coloneqq H[N_H[C]]$; note that $C \subseteq N_H[C] \subseteq C \cup K$. Apply theorem 5.2.3 to H_C with $A = N_H(C)$, obtaining a set Y_C of size at most t such that every connected component of $H_C - N[Y_C]$ contains at most $|N_H(C)|/2$ vertices of $N_H(C)$. Consequently, $N_{H_C}[Y_C]$ intersects all links in at least half of the buckets $\mathcal{L}_{w;u,v}$ with $w \in V(C)$. We infer that there is $y \in Y_C$ such that $N_{H_C}[y]$ intersects at least a $\frac{1}{t}$ fraction of links in at least $\frac{1}{200t}$ fraction of all nonempty buckets. This completes the proof. ■

If there exists a chip C with $|N_H(C)| \geqslant 2$, then there exists a nonempty bucket, as every bucket $\mathcal{L}_{w;u,v}$ for $w \in V(C)$ and $\{u, v\} \in \binom{N_H(C)}{2}$ is nonempty. Hence, lemma 5.5.11 allows us to choose a $\frac{1}{200t}$-heavy vertex as the branching pivot.

It remains to show that with this choice of the branching pivot, the subproblem tree generated by the secondary branching strategy has $\mathcal{O}(\log^2 n)$ success branches on any root-to-leaf path.

As argued, for a branch node x, if there exists a chip C in $H \coloneqq G[W^{\mathcal{R}(x)}]$ with $|N_H(C)| = 1$, then the pivot is the unique element of $N_H(C)$ and the secondary branching strategy terminates both in the failure child and in all success grandchildren. Thus, on any root-to-leaf path there is at most one such branch node.

For every other branch node x, the pivot is a $\frac{1}{200t}$-heavy vertex. The *secondary level*

of a node x is defined as

$$\lambda(x) \coloneqq \lfloor \log_2(1 + |\{(w, \{u, v\}) \mid \mathcal{L}_{w;u,v}^{G[W^{\mathcal{R}(x)}]} \neq \emptyset\}|)\rfloor.$$

Note that the secondary level is positive if and only if there is a nonempty bucket. Since at the root node there are $\mathcal{O}((n^{\mathcal{R}})^3)$ nonempty buckets, there are $\mathcal{O}(\log n^{\mathcal{R}})$ possible secondary levels. Furthermore, during the recursion the buckets can only shrink, so the secondary level of an ancestor is never lower than the secondary level of a descendant. Hence, it suffices to prove that any upward path in the subproblem tree on which all branching nodes have the same secondary level contains $\mathcal{O}(\log n^{\mathcal{R}})$ success branches.

To this end, consider the following potential for a node x.

$$\widehat{\mu}(x) \coloneqq \sum_{(w,\{u,v\}):\ \mathcal{L}_{w;u,v}^{G[W^{\mathcal{R}(x)}]} \neq \emptyset} \log_2 \left[\sum_{Q \in \mathcal{L}_{w;u,v}^{G[W^{\mathcal{R}(x)}]}} \sum_{q \in Q} (1 + \gamma(\eta^{\mathcal{R}(x)}, q, \zeta^{\mathcal{R}(x)}(q))) \right].$$

For every node x of secondary level λ, there are between $2^\lambda - 1$ and $2^{\lambda+1} - 2$ nonempty buckets of $G[W^{\mathcal{R}(x)}]$. Hence, for node x of secondary level λ,

$$\widehat{\mu}(x) \leqslant (2^{\lambda+1} - 2) \cdot \mathcal{O}(\log n^{\mathcal{R}}) = 2(2^\lambda - 1) \cdot \mathcal{O}(\log n^{\mathcal{R}}).$$

Let x be a branch node and y its successful grandchild, both with the same secondary level. Noting that the innermost sums in the definition of $\widehat{\mu}$ are lower-bounded by 1 and upper-bounded by $(d+1)t$, from the definition of a $\frac{1}{200t}$-heavy vertex and lemma 5.5.1 we infer that

$$\widehat{\mu}(x) - \widehat{\mu}(y) \geqslant \frac{c}{dt^3} \cdot (2^\lambda - 1),$$

for some universal constant $c > 0$. Since the potential μ never becomes negative, it can

decrease only $\mathcal{O}(\log n^{\mathcal{R}})$ times at a successful branch when the secondary level λ is fixed. Since there are $\mathcal{O}(\log n^{\mathcal{R}})$ secondary levels in total, we conclude that every root-to-leaf path contains $\mathcal{O}(\log^2 n^{\mathcal{R}})$ success branches.

This completes the proof of the properties of the secondary branching strategy and thus of lemma 5.4.9.

5.6 $C_{>t}$-free graphs of bounded degeneracy have bounded treewidth

It is well known that if a graph G has treewidth k, then its degeneracy it at most k. However, these parameters can be arbitrarily far away from each other: for instance, 3-regular expanders have degeneracy 3 and treewidth linear in the number of vertices [110]. In this section we prove that if we restrict our attention to $C_{>t}$-free graphs, the treewidth is bounded by a function of degeneracy. In particular, we show theorem 5.1.3.

Theorem 5.1.3. *For every pair of integers d and t, there exists an integer $k = (dt)^{\mathcal{O}(t)}$ such that every $C_{>t}$-free graph of degeneracy at most d has treewidth at most k.*

Before we proceed to the proof of theorem 5.1.3, let us recall the notion of brambles. Recall that two sets A, b are adjacent if either $A \cap B \neq \emptyset$ or there is an edge with one endpoint in A and the other in B. For brevity, we say that a set A is adjacent to a vertex v if A is adjacent to $\{v\}$, i.e., either $v \in A$ or v is adjacent to some vertex of A. A *bramble* of size p in a graph G is a collection $\mathcal{B} = (B_1, B_2, \ldots, B_p)$ of nonempty vertex subsets such that each B_i induces a connected graph and all B_is are pairwise adjacent. The sets B_i are called *branch sets*. The *order* of a bramble $\mathcal{B}$ is the size of a smallest set of vertices that hits all branch sets. Observe that the size of a bramble is always at least its order. We will use the following result of Hatzel *et al.* [108], which in a graph

of large treewidth constructs a bramble of large order in which no vertex participates in more than two branch sets.

Theorem 5.6.1 (Hatzel *et al.* [108])**.** *There exists a polynomial* $\mathsf{p}(\cdot)$ *such that for every positive integer* k*, every graph* G *of treewidth at least* k *contains a bramble* $\mathcal{B}$ *of order at least* $\sqrt{k}/\mathsf{p}(\log k)$ *such that each vertex of* G *is in at most two branch sets of* $\mathcal{B}$*.*

We now proceed to the proof of theorem 5.1.3. Without loss of generality we may assume that t is even, $t \geqslant 4$, and $d \geqslant 2$. For contradiction, suppose that G is a $C_{>t}$-free graph with degeneracy at most d and treewidth larger that

$$k \coloneqq \left(500\ 000 \cdot d^2 t^5\right)^{4t+4} \cdot \left[\mathsf{p}\left(\log\left((500\ 000 \cdot d^2 t^5)^{4t+4}\right)\right)\right]^4,$$

where $\mathsf{p}(\cdot)$ is the polynomial provided by theorem 5.6.1. Thus, by applying theorem 5.6.1 to G we obtain a bramble $\mathcal{B} = (B_1, B_2, \ldots, B_p)$ of order

$$p > \frac{\sqrt{k}}{\mathsf{p}(\log k)} \geqslant \left(500\ 000 \cdot d^2 t^5\right)^{2t+2}.$$

Note that we can assume that each branch set of $\mathcal{B}$ is inclusion-wise minimal (subject to $\mathcal{B}$ being a bramble), as otherwise we can remove some vertices from branch sets. Therefore, for each branch set B_i and each vertex v of B_i, either there is some branch set B_j which is adjacent to v but nonadjacent to $B_i - \{v\}$, or v is a cutvertex in $G[B_i]$ and its role is to keep the branch set connected.

Claim 5.6.2. *For each* $i \in [p]$*, and all* $u, v \in B_i$*, the distance between* u *and* v *in* $G[B_i]$ *is at most* t*.*

Proof of Claim 4. *For contradiction, suppose that there is* B_i *violating the claim. Let* u, v *be the vertices at maximum distance in* $G[B_i]$*, by assumption this distance is at*

least $t+1$. As u and v are the ends of a maximal path in $G[B_i]$, none of them is a cutvertex in $G[B_i]$. Thus there is a branch set B_u which is adjacent only to u in B_i, and another branch set which is adjacent only to v in B_i. Recall that $B_u \cup B_v$ is connected and nonadjacent to $B_i - \{u, v\}$. So by concatenating a shortest u-v-path in B_i and a shortest u-v-path in $B_u \cup B_v$, we obtain an induced cycle with at least $t+1$ vertices, a contradiction.

Let G' be the lexicographic product $G \bullet K_2$: the graph obtained from G by introducing, for each $x \in V(G)$, a copy x' of x and making it adjacent to x, all neighbors of x, and all their copies. Note that in G', the copy x' is a true twin of x. Observe also that the degeneracy of G' is at most $2d+1$: we can modify a d-degeneracy ordering of G into a $(2d+1)$-degeneracy ordering of G' by inserting each vertex x' immediately after x.

Claim 5.6.3. *The graph G' contains K_p as a depth-t minor.*

Proof of Claim 5. *We construct a family $\mathcal{B}' = (B'_1, B'_2, \ldots, B'_p)$ as follows. We start with $B'_i := B_i$ for all $i \in [p]$ and we iteratively inspect every vertex x of G. If x belongs to more than one of the sets $\{B_1, \ldots, B_p\}$, then, by the properties given by theorem 5.6.1, x must belong to exactly two of them, say $x \in B_i \cap B_j$ for some $i \neq j$. Then replace x with x' in B'_j, thus making B'_i and B'_j not overlap on x.*

It is clear that once this operation is applied to each vertex of G, the resulting sets of $\mathcal{B}'$ are pairwise disjoint and pairwise adjacent. Further, for each $i \in [p]$ the graph $G'[B'_i]$ is isomorphic to $G[B_i]$, as we only replaced some vertices by their true twins, so in particular $G'[B'_i]$ is connected. Therefore, $\mathcal{B}'$ is a minor model of a clique of order p in G'. By claim 5.6.2, the radius of each graph $G'[B'_i]$ is at most t, hence this model has depth at most t.

The next result binds the maximum size of a bounded-depth clique minor and the maximum size of a bounded-depth topological clique minor that can be found in a graph.

It is a fairly standard fact used in the sparsity theory; for the proof, see e.g. [109, Lemma 2.19 and Corollary 2.20].

Proposition 5.6.4. *Let G be a graph and let t, p, p' be integers such that $p \geqslant 1 + (p' + 1)^{2t+2}$. If G contains K_p as a depth-t minor, then G contains $K_{p'}$ as a depth-$(3t+1)$ topological minor.*

By combining claim 5.6.3 and proposition 5.6.4, we conclude that G'' contains $K_{p'}$ as a topological depth-$(3t+1)$ minor, where

$$p' \coloneqq \left\lfloor \frac{p^{\frac{1}{2t+2}}}{4} \right\rfloor \geqslant 100\ 000 \cdot d^2 \cdot t^5.$$

Fix some topological depth-$(3t+1)$ minor model of $K_{p'}$ in G'. Let R be the set of roots of the minor model and consider the graph $G'[R]$. It has p' vertices and, as a subgraph of G', is $(2d+1)$-degenerate. Therefore, there is an independent set R' in $G'[R]$ of size at least

$$p'' \coloneqq \left\lceil \frac{p'}{2d+2} \right\rceil \geqslant \frac{100\ 000 \cdot d^2 \cdot t^5}{2d+2} \geqslant 20\ 000 \cdot d \cdot t^5.$$

Observe that restricting our minor model only to the roots that are in R' and paths incident to them gives us a topological depth-$(3t+1)$ minor model of $K_{p''}$ with the additional property that the roots are pairwise nonadjacent.

Let H be the subgraph of G' induced by the vertices used by the topological minor model obtained in the previous step. Let X be the set of vertices of H with degree larger than $200 \cdot d \cdot t^2$, which are not roots. Since H is $(2d+1)$-degenerate, we observe that

$$|X| \leqslant \frac{(2d+1)|V(H)|}{100 \cdot dt^2} \leqslant \frac{(2d+1)(6t+3)\binom{p''}{2}}{100 \cdot dt^2} \leqslant \frac{20}{100t}\binom{p''}{2} = \varepsilon \cdot \binom{p''}{2}, \text{ where } \varepsilon \coloneqq \frac{1}{5t}.$$

Let H' be obtained from H by removing all vertices in X, along with all paths from the topological minor model which contain a vertex from X. Note that thus, we have removed at most $\varepsilon \cdot \binom{p''}{2}$ paths.

Observe that H' still contains a depth-$(3t+1)$ topological minor model of some graph Z with p'' vertices and at least

$$\binom{p''}{2} - |X| \geqslant \binom{p''}{2} - \varepsilon\binom{p''}{2} = (1-\varepsilon)\binom{p''}{2}$$

edges. Thus, the average degree of a vertex in Z is at least $(1-\varepsilon)(p''-1)$.

Let $\mathcal{W} = (v_0, v_1, \ldots, v_{t/2})$ be a sequence of vertices of Z, chosen independently and uniformly at random. In what follows, all arithmetic operations on the indices of the vertices v_i are computed modulo $t/2+1$, in particular $v_{t/2+1} = v_0$.

We prove that with positive probability, $\mathcal{W}$ has the following four properties:

(P1) The vertices v_i are pairwise distinct.

(P2) For every $0 \leqslant i \leqslant t/2$, v_iv_{i+1} is an edge of Z; let P_i be the corresponding path in H'.

(P3) For every $0 \leqslant i \leqslant t/2$ and $0 \leqslant j \leqslant t/2$ such that $j \notin \{i, i+1\}$, the internal vertices on the path P_i are anti-adjacent to v_j.

(P4) For all $0 \leqslant i < j \leqslant t/2$, the internal vertices of P_i are anti-adjacent to the internal vertices of P_j.

Observe that these four properties imply that the concatenation of all paths P_i is a hole of length more than t in G' (recall here that the roots of the minor model are independent in H'). The assumption that G is $C_{>t}$-free implies that G' is $C_{>t}$-free as well, hence this will be a contradiction.

For (P1), since $p'' \geqslant 20\,000 \cdot d \cdot t^5$, by the union bound the probability that $v_i = v_j$ for some $i \neq j$ is at most $\binom{t}{2}/p'' < 0.1$.

For (P2), since v_i and v_{i+1} are independently chosen vertices, and Z has at least $(1-\varepsilon)\binom{p''}{2}$ edges, the probability that $v_i v_{i+1}$ is not an edge of Z is bounded by $\varepsilon = \frac{1}{5t}$. By the union bound, the probability that (P2) does not hold is bounded by $\varepsilon \cdot (t/2+1) \leqslant 0.2$.

For (P3), fix $0 \leqslant i \leqslant t/2$ and assume $v_i v_{i+1} \in E(Z)$ so that P_i is defined. Then, the total number of neighbors of the internal vertices of P_i is bounded by $(6t+3) \cdot 200 \cdot d \cdot t^2 \leqslant 2000 \cdot d \cdot t^3$. Since v_j is a vertex of $V(Z)$ chosen at random independently of the choice of v_i and v_{i+1}, the probability that v_j is among these neighbors is bounded by $2000 \cdot dt^3/p'' \leqslant 0.1/t^2$. By the union bound, (P2) holds but (P3) does not hold with probability at most $t(t-2) \cdot \frac{0.1}{t^2} \leqslant 0.1$.

For (P4), fix $0 \leqslant i < j \leqslant t/2$. Note that it may be possible that $i+1 = j$ or $j+1 = i$ (cyclically modulo $t/2+1$), but not both. Hence, by symmetry between i and j, assume that the choice of v_{j+1} is independent of the choices of v_i, v_{i+1}, and v_j. Assume that $v_i v_{i+1} \in E(Z)$ so that P_i is defined. As in the previous paragraph, there are at most $2000dt^3$ neighbors in H' of the internal vertices of P_i. There are $p'' = |V(Z)|$ choices for v_{j+1}, all of them leading to either $v_j v_{j+1} \notin E(Z)$ or to vertex-disjoint (except for v_j) choices of the path P_j. Hence, for at most $2000dt^3$ of these choices, we have $v_j v_{j+1} \in E(Z)$ but there is an edge between an internal vertex of P_j and an internal vertex of P_i. By the union bound, (P2) holds but (P4) does not hold with probability less than $\binom{t/2+1}{2} \cdot \frac{2000dt^3}{p''} \leqslant \binom{t/2+1}{2} \cdot \frac{2000dt^3}{20\,000dt^5} \leqslant 0.1$.

By the union bound over all the above cases, $\mathcal{W}$ satisfies all properties (P1)–(P4) with probability at least $1 - 0.1 - 0.2 - 0.1 - 0.1 = 0.5$. This gives the desired contradiction and completes the proof.

5.7 MSO_2 and $C_{>t}$-free graphs

In this section we prove theorem 5.1.1 using the branching strategy described in section 5.4.

5.7.1 Extending subproblems

Fix integers $k \geqslant 1$ and $t \geqslant 3$, CMSO_2 sentence φ, and a $C_{>t}$-free graph G. Let q be the maximum of the quantifier rank of φ, the quantifier rank of $\varphi_{\mathrm{tw}<k}$, where $\varphi_{\mathrm{tw}<k}$ comes from lemma 2.4.3, and 4 (so that we can use lemma 2.4.2). Let p be the largest modulus used in φ, or 0 if φ does not contain any modular atomic expression. We also denote $k' := 6k$. In what follows, we will be using CMSO_2 types $\mathsf{Types}^{k',p,q}$, as given by proposition 2.4.1. Hence, by "CMSO_2 types" we mean elements of $\mathsf{Types}^{k',p,q}$, and we drop the super- or subscript k', p, q when it is clear from the context.

Recall that every graph of treewidth less than k is $(k-1)$-degenerate. Hence, we will use the branching strategy provided lemma 5.4.9 for $d := k-1$ and t. More precisely, the algorithm executes the branching, i.e. decides on the types of nodes, chooses branching pivots, etc., exactly as prescribed by the strategy given by lemma 5.4.9. However, we enrich the subproblems with some additional piece of information, which intuitively encodes a skeleton of a decomposition of the subgraph induced by the constructed solution. From the recursive subcalls, we expect returning quite an elaborate result: intuitively, optimum solutions to the subproblem of every possible CMSO_2 type. As we will check at the end, the properties asserted by lemma 5.4.9 ensure a quasipolynomial running time bound.

Let us start by formally augmenting the notion of a subproblem with extra information. At every leaf, split, and branch node x, the subproblem $\mathcal{R} := \mathcal{R}(x)$ contains additionally the following:

1. a tree decomposition $(T^{\mathcal{R}}, \beta^{\mathcal{R}})$ of $G[A^{\mathcal{R}}]$ with maximum bag size at most $k' = 6k$ and the root node having an empty bag;
2. a labelling $\iota^{\mathcal{R}} \colon A^{\mathcal{R}} \to [k']$ such that for every $a \in V(T^{\mathcal{R}})$, $\iota^{\mathcal{R}}$ restricted to $\beta^{\mathcal{R}}(a)$ is injective.

We call such a subproblem an *extended subproblem*. Leaf, split, and branch nodes are called *extendable nodes*.

For such an extended subproblem $\mathcal{R}$, a set $S \subseteq W^{\mathcal{R}}$ is a *feasible solution* if for every connected component C of $G[S]$, we have $|N_G(C) \cap A^{\mathcal{R}}| \leqslant 4k$ and there exists a node $a \in V(T^{\mathcal{R}})$ such that $N_G(C) \cap A^{\mathcal{R}} \subseteq \beta^{\mathcal{R}}(a)$. Observe that from the properties of a tree decomposition it follows that the family of those nodes $a \in V(T^{\mathcal{R}})$ for which $N_G(C) \cap A^{\mathcal{R}} \subseteq \beta^{\mathcal{R}}(a)$ is a connected subtree of T. By $a^{\mathcal{R}}(C)$ we denote the highest such node a.

For an extended subproblem $\mathcal{R}$, a *type assignment* is a function $\mathsf{TypeTree} \colon V(T^{\mathcal{R}}) \to \mathsf{Types}$. A feasible solution S is *of type* $\mathsf{TypeTree}$ if for every $a \in V(T^{\mathcal{R}})$, the subgraph of G induced by

$$\mathsf{extS}^{\mathcal{R}}(a, S) \coloneqq \beta^{\mathcal{R}}(a) \cup \bigcup \{C \in \mathrm{cc}(G[S]) \mid a^{\mathcal{R}}(C) = a\}$$

equipped with the labelling $\iota^{\mathcal{R}}|_{\beta^{\mathcal{R}}(a)}$ on boundary $\beta^{\mathcal{R}}(a)$ is of CMSO_2 type $\mathsf{TypeTree}(a)$.

The branching strategy will return, at every extendable node x, for every type assignment $\mathsf{TypeTree}$ at x, a feasible solution $\mathsf{S}[x, \mathsf{TypeTree}]$ of type $\mathsf{TypeTree}$. We allow $\mathsf{S}[x, \mathsf{TypeTree}] = \bot$, indicating that no such feasible solution was found, and we use the convention that the weight of $\bot$ is $-\infty$. In the algorithm description the following operation will be useful when defining $\mathsf{S}[x, \cdot]$ for a fixed node x: given a current state of the table $\mathsf{S}[x, \cdot]$ and a feasible solution S of type $\mathsf{TypeTree}$, *updating* $\mathsf{S}[x, \mathsf{TypeTree}]$ *with* S is

an operation that sets $\mathsf{S}[x, \mathsf{TypeTree}] \coloneqq S$ if the weight of S is larger than the weight of the former value of $\mathsf{S}[x, \mathsf{TypeTree}]$.

In the root of the recursion r, the considered extension is the trivial one: the tree decomposition $(T^{\mathcal{R}(r)}, \beta^{\mathcal{R}(r)})$ consists of a single node a^r with an empty bag, and labelling $\iota^{\mathcal{R}(r)}$ is empty. After the computation if finished, we iterate over all type assignments $\mathsf{TypeTree}$ for r for which $\mathsf{TypeTree}(a^r) \in \mathsf{type}[\varphi] \cap \mathsf{type}[\varphi_{\mathrm{tw}<k}]$ and return the set $\mathsf{S}[r, \mathsf{TypeTree}]$ of maximum weight found (ignoring values $\perp$). If no such set $\mathsf{S}[r, \mathsf{TypeTree}]$ is found (all values are $\perp$), the algorithm returns that there is no such set S.

Note that, assuming the recursive strategy indeed maintains the invariant that $\mathsf{S}[x, \mathsf{TypeTree}]$ is of type $\mathsf{TypeTree}$, for the returned set S we have $G[S] \models \varphi$ and $G[S]$ is of treewidth less than k.

5.7.2 Extended computation at nodes of the subproblem tree

Recall that now, the setting is that each extendable node (lead, split, or branch node) is assigned an extended subproblem. We need to describe (a) how at each node we handle the extended subproblem and what extended subproblems are passed down the subproblem tree; (b) what is the subroutine for handling free nodes; and (c) how the tables $\mathsf{S}[\cdot, \cdot]$ are computed along the recursion. As before, each type of a node is handled differently.

Leaf nodes

For a leaf node x, there is little choice the algorithm could do.

We have $W^{\mathcal{R}(x)} = \emptyset$, so the only feasible solution is $S = \emptyset$. There is exactly one type assignment $\mathsf{TypeTree}$ such that for every $a \in V(T^{\mathcal{R}(x)})$ we have that $G[\mathsf{extS}^{\mathcal{R}(x)}(a, \emptyset)]$ with the labelling $\iota^{\mathcal{R}}|_{\beta^{\mathcal{R}}(a)}$ is of type $\mathsf{TypeTree}(a)$. For this labelling, we set $\mathsf{S}[x, \mathsf{TypeTree}] = \emptyset$

and for every other type assignment $\mathsf{TypeTree}'$ we set $\mathsf{S}[x, \mathsf{TypeTree}'] = \bot$.

Split nodes

Let x be a split node. We do not use the power of free nodes below split nodes; formally, they are dummy free nodes as in the proof of theorem 5.4.1 that only pass the subproblem to their single child. Therefore, in what follows we speak about grandchildren of a split node.

If x has one grandchild y, then $\mathcal{R}(y)$ differs from $\mathcal{R}(x)$ only by its level. Therefore, given an extension of $\mathcal{R}(x)$, we pass the same extension to the grandchild y and for the return value at x, we copy the result returned by the grandchild y.

The situation is more interesting if x has two grandchildren y_1 and y_2. Recall that then $W^{\mathcal{R}(y_1)}$ and $W^{\mathcal{R}(y_2)}$ is a partition of $W^{\mathcal{R}(x)}$, while $A^{\mathcal{R}(y_1)} = A^{\mathcal{R}(y_2)} = A^{\mathcal{R}(x)}$. Given a subproblem extension of $\mathcal{R}(x)$, we pass it without modifications to both y_1 and y_2. To compute $\mathsf{S}[x, \cdot]$ based on $\mathsf{S}[y_1, \cdot]$ and $\mathsf{S}[y_2, \cdot]$, we proceed as follows.

First, we initiate $\mathsf{S}[x, \mathsf{TypeTree}] = \bot$ for every tree assignment $\mathsf{TypeTree}$ at x. Then, we iterate over all pairs $\mathsf{TypeTree}_1$ and $\mathsf{TypeTree}_2$ of type assignments for $\mathcal{R}(x)$ such that both $\mathsf{S}[y_1, \mathsf{TypeTree}_1] \neq \bot$ and $\mathsf{S}[y_2, \mathsf{TypeTree}_2] \neq \bot$. Let $\mathsf{TypeTree}$ be the type assignment for $\mathcal{R}(x)$ defined as

$$\mathsf{TypeTree}(a) = \mathsf{TypeTree}_1(a) \oplus \mathsf{TypeTree}_2(a) \qquad \text{for every } a \in T^{\mathcal{R}(x)}.$$

Then observe that as $W^{\mathcal{R}(y_1)}$ is nonadjacent to $W^{\mathcal{R}(y_2)}$, $\mathsf{S}[y_1, \mathsf{TypeTree}_1] \cup \mathsf{S}[y_2, \mathsf{TypeTree}_2]$ is a feasible solution for x of type $\mathsf{TypeTree}$. We update $\mathsf{S}[x, \mathsf{TypeTree}]$ with $\mathsf{S}[y_1, \mathsf{TypeTree}_1] \cup \mathsf{S}[y_2, \mathsf{TypeTree}_2]$.

Branch and subsequent free nodes

Let x be a branch node and recall that we are given a subproblem extension of $\mathcal{R}(x)$. For the failure child y^x of x, we pass this subproblem extension to y^x without modifications.

Consider now a success grandchild $s := s^x_{\mathcal{D}}$ for $\mathcal{D} = (D, \eta, (D_u)_{u \in D'})$, $D' = D \cup \{\nu^x\}$. We now use the power of the free node $s^x_{\mathcal{D}}$ to guess how the extension at x should be enhanced.

Formally, we iterate over every possibility of:

- a partition $\mathcal{B}$ of D' into nonempty subsets;
- a node $a_B \in V(T^{\mathcal{R}(x)})$ and a subset $N_B \subseteq \beta^{\mathcal{R}(x)}(a_B)$ of size at most $4k$ for every $B \in \mathcal{B}$;
- a set $X_B \subseteq W^{\mathcal{R}(s)}$ of size at most k for every $B \in \mathcal{B}$ so that the sets $(X_B)_{B \in \mathcal{B}}$ are pairwise vertex-disjoint; we denote $X_{\mathcal{B}} := \bigcup_{B \in \mathcal{B}} X_B$;
- a position guess $\eta^{\mathcal{B}}$ for $X_{\mathcal{B}}$ and a left neighbor guess $(D^{\mathcal{B}}_u)_{u \in X_{\mathcal{B}}}$ for $X_{\mathcal{B}}$ and $\eta^{\mathcal{B}}$.

For every choice $\mathcal{C} = (\mathcal{B}, (a_B, N_B, X_B)_{B \in \mathcal{B}}, \eta^{\mathcal{B}}, (D^{\mathcal{B}}_u)_{u \in X_{\mathcal{B}}})$ as above, we construct a child $\hat{s}^x_{\mathcal{D},\mathcal{C}}$ of $s^x_{\mathcal{D}}$ that is created from $s^x_{\mathcal{D}}$ by taking $X_{\mathcal{B}}$ at positions $\eta^{\mathcal{B}}$ with left neighbors $(D^{\mathcal{B}}_u)_{u \in X_{\mathcal{B}}}$. Every created child $\hat{s}^x_{\mathcal{D},\mathcal{C}}$ is a filter node; we denote the resulting grandchild (if present) as $\tilde{s}^x_{\mathcal{D},\mathcal{C}}$.

Denoting $\tilde{s} = \tilde{s}^x_{\mathcal{D},\mathcal{C}}$ for brevity, the grandchild's subproblem is extended as follows:

- create $(T^{\mathcal{R}(\tilde{s})}, \beta^{\mathcal{R}(\tilde{s})})$ from $(T^{\mathcal{R}(x)}, \beta^{\mathcal{R}(x)})$ by adding, for every $B \in \mathcal{B}$, a new node $\tilde{a}_B$ with bag $N_B \cup B \cup X_B$ that is a child of a_B;
- create $\iota^{\mathcal{R}(\tilde{s})}$ by extending $\iota^{\mathcal{R}(x)}$ to the new elements of $W^{\mathcal{R}(\tilde{s})}$ in any manner that is injective on the newly created bags.

Observe that every newly created bag is of size bounded by $4k+k+k=k'$. Since $\mathcal{B}$ is a partition of D' and we assume that the sets $(X_B)_{B\in\mathcal{B}}$ are pairwise disjoint, for every two newly created nodes a'_{B_1} and a'_{B_2}, we have $\beta^{\mathcal{R}(\tilde{s})}(a'_{B_1})\cap\beta^{\mathcal{R}(\tilde{s})}(a'_{B_2})\subseteq N_{B_1}\cap N_{B_2}\subseteq A^{\mathcal{R}(x)}$, the pair $(T^{\mathcal{R}(\tilde{s})},\beta^{\mathcal{R}(\tilde{s})})$ is indeed a tree decomposition of $G[A^{\mathcal{R}(\tilde{s})}]$ with maximum bag size at most k', and it is straightforward to extend $\iota^{\mathcal{R}(x)}$ to obtain $\iota^{\mathcal{R}(\tilde{s})}$.

To complete the description of the algorithm, it remains to show how to assemble the table $\mathsf{S}[x,\cdot]$ at the branch node x from the tables computed for its great-great-grandchildren $\tilde{s}^x_{\mathcal{D},\mathcal{C}}$ and the failure child y^x.

To this end, we initiate $\mathsf{S}[x,\mathsf{TypeTree}]=\mathsf{S}[y^x,\mathsf{TypeTree}]$ for every type assignment $\mathsf{TypeTree}$ at x (note that every feasible solution at y^x is also a feasible solution at x and is of the same type). Then, for every great-great-grandchild $\tilde{s}:=\tilde{s}^x_{\mathcal{D},\mathcal{C}}$ and type assignment $\widetilde{\mathsf{TypeTree}}$ at $\tilde{s}$ such that $\tilde{S}:=\mathsf{S}[\tilde{s},\widetilde{\mathsf{TypeTree}}]\neq\bot$, we proceed as follows. Let $\mathcal{D}=(D,\eta,(D_u)_{u\in D'})$, $D'=D\cup\{\nu^x\}$, $\mathcal{C}=(\mathcal{B},(a_B,N_B,X_B)_{B\in\mathcal{B}},\eta^{\mathcal{B}},(D^{\mathcal{B}}_u)_{u\in X_{\mathcal{B}}})$, and $X_{\mathcal{B}}=\bigcup_{B\in\mathcal{B}}X_B$. Observe that $A^{\mathcal{R}(\tilde{s})}-A^{\mathcal{R}(x)}=D'\cup X_{\mathcal{B}}$. Let $S:=\tilde{S}\cup D'\cup X_{\mathcal{B}}$.

We say that the pair $(\tilde{s},\tilde{S})$ is *liftable* if the set S obtained as above is a feasible solution at x and, furthermore, for every $B\in\mathcal{B}$ there exists a connected component $\tilde{C}_B$ of $G[S]$ such that

$$\tilde{C}_B=B\cup X_B\cup\bigcup\{\tilde{C}\in\mathrm{cc}(G[\tilde{S}])\mid a^{\tilde{s}}(\tilde{C})=\tilde{a}_B\},$$

the components $(\tilde{C}_B)_{B\in\mathcal{B}}$ are pairwise distinct, and $a^x(\tilde{C}_B)=a_B$ for every $B\in\mathcal{B}$. For a liftable pair $(\tilde{s},\tilde{S})$, the set S is called the *lift of* $(\tilde{s},\tilde{S})$.

If $(\tilde{s},\tilde{S})$ is liftable, then we can use proposition 2.4.1 to compute the type $\mathsf{TypeTree}_S$ of the lift S at x as follows. First, for every $B\in\mathcal{B}$, the CMSO_2 type of $G[C_B\cup\beta^{\mathcal{R}(x)}(a_B)]$ with labelling $\iota^{\mathcal{R}(x)}|_{\beta^{\mathcal{R}(x)}}(a_B)$ can be computed from $\widetilde{\mathsf{TypeTree}}(\tilde{a}_B)$ by forgetting the labels of $B\cup X_B$. Second, for each $a\in V(T^{\mathcal{R}(x)})$, the type $\mathsf{TypeTree}_S(a)$ is the composition of

$\widetilde{\mathsf{TypeTree}}(a)$ and the types of all graphs $(G[C_B \cup \beta^{\mathcal{R}(x)}(a_B)], \iota^{\mathcal{R}(x)}|_{\beta^{\mathcal{R}(x)}(a_B)})$ for those B for which $a_B = a$.

For every liftable pair $(\tilde{s}, \tilde{S})$, we compute the lift S and its type $\mathsf{TypeTree}_S$ at x as above and we update $\mathsf{S}[x, \mathsf{TypeTree}_S]$ with S. This finishes the description of the algorithm at branch nodes.

We conclude this section with an immediate, yet important corollary of the way how we compute the type $\mathsf{TypeTree}_S$.

Lemma 5.7.1. *Let x be a branch node, $\tilde{s}$ be its great-great-grandchild, and $\tilde{S}_1$ and $\tilde{S}_2$ be two feasible solutions at $\tilde{s}$ of the same type such that both $(\tilde{s}, \tilde{S}_1)$ and $(\tilde{s}, \tilde{S}_2)$ are liftable. Let S_i be the lift of $(\tilde{s}, \tilde{S}_i)$ for $i = 1, 2$. Then S_1 and S_2 are of the same type at x.*

Proof: The aforementioned algorithm to compute the type of the lift of $(\tilde{s}, \tilde{S}_i)$ uses only the type of $\tilde{S}_i$ at $\tilde{s}$ and the subproblems $\mathcal{R}(x)$ and $\mathcal{R}(\tilde{s})$. The claim follows. ■

5.7.3 Correctness

Fix a subset $S^* \subseteq V(G)$ such that $G[S^*]$ is of treewidth less than k and $G[S^*] \models \varphi$. We would like to show that the algorithm returns a set S of weight at least the weight of S^* (not necessarily S^*). Clearly, since $G[S^*]$ is $(k-1)$-degenerate, we can speak about lucky nodes of the subproblem tree, defined in the same manner as in section 5.4.

A lucky extendable node x is called a *gander* if $S^* \cap W^{\mathcal{R}(x)}$ is a feasible solution for $\mathcal{R}(x)$. For a gander x, define the type assignment $\mathsf{TypeTree}^x$ as the type of the feasible solution $S^* \cap W^{\mathcal{R}(x)}$.

Note that the root r of the subproblem tree is a gander and $\mathsf{TypeTree}^r(a^r) \in \mathsf{type}[\varphi] \cap \mathsf{type}[\varphi_{\mathrm{tw}<k}]$. Thus, it suffices to show the following:

Lemma 5.7.2. *For every gander x, we have that $\mathsf{S}[x, \mathsf{TypeTree}^x] \neq \bot$ and the weight of $\mathsf{S}[x, \mathsf{TypeTree}^x]$ is at least the weight of $S^* \cap W^{\mathcal{R}(x)}$.*

Proof: The proof proceeds by a bottom-up induction on the subproblem tree.

For a leaf gander x, since x is lucky, from the definition of $\mathsf{TypeTree}^x$ the only type assignment $\mathsf{TypeTree}$ for which $\mathsf{S}[x, \mathsf{TypeTree}] \neq \bot$ is exactly $\mathsf{TypeTree}^x$. Consequently, $\mathsf{S}[x, \mathsf{TypeTree}^x] = \emptyset$ and the claim is proven.

For a split gander x, the claim is straightforward if x has one grandchild. Assume then x has two grandchildren y_1 and y_2. Since $W^{\mathcal{R}(y_1)}$ is nonadjacent to $W^{\mathcal{R}(y_2)}$, every connected component of $G[S^* \cap W^{\mathcal{R}(x)}]$ is contained either in $W^{\mathcal{R}(y_1)}$ or in $W^{\mathcal{R}(y_2)}$. It follows that both y_1 and y_2 are ganders, too.

By induction, for $i = 1, 2$ the weight of $\mathsf{S}[y_i, \mathsf{TypeTree}^{y_i}]$ is at least the weight of $S^* \cap W^{\mathcal{R}(y_i)}$. Denote

$$\mathsf{TypeTree}(a) \coloneqq \mathsf{TypeTree}^{y_1}(a) \oplus \mathsf{TypeTree}^{y_2}(a) \qquad \text{for } a \in V(T^{\mathcal{R}(x)}).$$

Then, on one hand from proposition 2.4.1 we have that $\mathsf{TypeTree} = \mathsf{TypeTree}^x$, and on the other hand the handling of split nodes will update $\mathsf{S}[x, \mathsf{TypeTree}]$ with $\mathsf{S}[y_1, \mathsf{TypeTree}^{y_1}] \cup \mathsf{S}[y_2, \mathsf{TypeTree}^{y_2}]$. The claim for split ganders follows.

It remains to analyse branch ganders. Let x be a branch gander. If $\nu^x \notin S^*$, then y^x is lucky and it is immediate that it is a gander, too. By the inductive assumption, $\mathfrak{w}(\mathsf{S}[y^x, \mathsf{TypeTree}^{y^x}]) \geqslant \mathfrak{w}(S^* \cap W^{\mathcal{R}(y^x)})$. Since in this case $\mathsf{TypeTree}^x = \mathsf{TypeTree}^{y^x}$ and we initiated $\mathsf{S}[x, \mathsf{TypeTree}^x]$ with $\mathsf{S}[y^x, \mathsf{TypeTree}^x]$, the inductive claim follows.

Assume then that $\nu^x \in S^*$. Define D, η, and $(D_u)_{u \in D'}$ for $D' = D \cup \{\nu^x\}$ as in lemma 5.4.7, that is:

$$\begin{aligned} D &= \{u \in N_G(\nu^x) \cap W^{\mathcal{R}(x)} \cap S^* \mid \eta^*(u) < \eta^*(\nu^x)\}, \\ \eta &= \eta^*|_{A^{\mathcal{R}(x)} \cup D'}, \\ D_u &= \{w \in N_G(u) \cap W^{\mathcal{R}(x)} \cap S^* - D' \mid \eta^*(w) < \eta^*(u)\}. \end{aligned}$$

Then, by lemma 5.4.7, the branch node x creates a child $z^x_{\mathcal{D}}$ for $\mathcal{D} = (D, \eta, (D_u)_{u \in D'})$ and $z^x_{\mathcal{D}}$ is lucky. Hence, the grandchild $s^x_{\mathcal{D}}$ exists and is lucky as well.

We now want to look at some particular child of the free node $s^x_{\mathcal{D}}$. Define the partition $\mathcal{B}$ of D' as the partition induced by the connected components of $G[S^* \cap W^{\mathcal{R}(x)}]$ on D', that is:

$$\mathcal{B} = \{C \cap D' \colon C \in \mathrm{cc}(G[S^* \cap W^{\mathcal{R}(x)}]) \text{ such that } C \cap D' \neq \emptyset\}.$$

For every $B \in \mathcal{B}$, let C_B be the connected component of $G[S^* \cap W^{\mathcal{R}(x)}]$ that contains B. Define $a_B \coloneqq a^x(C_B)$ for $B \in \mathcal{B}$ and $N_B \coloneqq N_G(C_B) \cap A^{\mathcal{R}(x)}$. Since x is a gander, we have $|N_B| \leqslant 4k$ and $N_B \subseteq \beta^{\mathcal{R}(x)}(a_B)$. Let $X^\circ_B \subseteq C_B \cup N_B$ be a set of size at most k promised by lemma 2.3.1 for the graph $G[C_B \cup N_B]$ and $A = N_B$; note here that $G[C_B \cup N_B]$ is of treewidth less than k, because $C_B, N_B \subseteq S^*$. Let $X_B \coloneqq X^\circ_B \cap C_B$. Finally, define $\eta^{\mathcal{B}}$ and $(D^{\mathcal{B}}_u)_{u \in X_{\mathcal{B}}}$ similarly as for the grandchildren of x:

$$\eta^{\mathcal{B}} = \eta^*|_{A^{\mathcal{R}(x)} \cup D' \cup X_{\mathcal{B}}},$$
$$D^{\mathcal{B}}_u = \{w \in N_G(u) \cap W^{\mathcal{R}(x)} \cap S^* - (D' \cup X_{\mathcal{B}}) \mid \eta^*(w) < \eta^*(u)\}.$$

Let $\mathcal{C} = (\mathcal{B}, (a_B, N_B, X_B)_{B \in \mathcal{B}}, \eta^{\mathcal{B}}, (D^{\mathcal{B}}_u)_{u \in X_{\mathcal{B}}})$. Observe that $\hat{s} \coloneqq \hat{s}^x_{\mathcal{D},\mathcal{C}}$ is lucky by lemma 5.4.3, as $X_{\mathcal{B}} \subseteq S^*$. Hence, $\tilde{s} \coloneqq \tilde{s}^x_{\mathcal{D},\mathcal{C}}$ exists and is lucky as well. We claim that $\tilde{s}$ is a gander.

To this end, consider a connected component $\tilde{C}$ of $G[S^* \cap W^{\mathcal{R}(\tilde{s})}]$. We want to show that $|N_G(\tilde{C}) \cap S^*| \leqslant 4k$ and there exists a node $\tilde{a} \in V(T^{\mathcal{R}(\tilde{s})})$ such that $N_G(\tilde{C}) \cap S^* \subseteq \beta^{\mathcal{R}(\tilde{s})}(\tilde{a})$. The claim is straightforward if $\tilde{C}$ is also a connected component of $G[S^* \cap W^{\mathcal{R}(x)}]$, because x is a gander. Otherwise, $\tilde{C} \subsetneq C$ for some connected component C of $G[S^* \cap W^{\mathcal{R}(x)}]$. Note that there exists $B \in \mathcal{B}$ such that $B \subseteq C$, that is, $C = C_B$ for some $B \in \mathcal{B}$.

By the choice of X°_B, the connected component of $G[C_B \cup N_B] - X^\circ_B$ that contains $\tilde{C}$

contains at most $|N_B|/2 \leqslant 2k$ vertices of N_B. Consequently,

$$|N_G(\tilde{C}) \cap S^*| \leqslant |X_B| + |B| + |N_B|/2 \leqslant k + k + 2k = 4k.$$

Furthermore,

$$N_G(\tilde{C}) \cap S^* \subseteq X_B \cup B \cup N_B \subseteq \beta^{\mathcal{R}(\tilde{s})}(\tilde{a}_B).$$

We infer that $\tilde{s}$ is indeed a gander.

Since $S^* \cap W^{\mathcal{R}(\tilde{s})}$ is a feasible solution at $\tilde{s}$, it follows immediately from the definition of $\mathcal{C}$ that $(\tilde{s}, S^* \cap W^{\mathcal{R}(\tilde{s})})$ is liftable with the lift $S^* \cap W^{\mathcal{R}(x)}$.

Let now $\tilde{S} = \mathsf{S}[\tilde{s}, \mathsf{TypeTree}^{\tilde{s}}]$ and $S = \tilde{S} \cup D' \cup X_{\mathcal{B}}$; note that $A^{\mathcal{R}(\tilde{s})} - A^{\mathcal{R}(x)} = D' \cup X_{\mathcal{B}}$.

We claim that $(\tilde{s}, \tilde{S})$ is liftable. To this end, fix $B \in \mathcal{B}$. Observe that:

$$\mathsf{extS}^{\tilde{s}}(\tilde{a}_B, S^* \cap W^{\mathcal{R}(\tilde{s})}) = \beta^{\mathcal{R}(\tilde{s})}(a) \cup C_B.$$

Consider the connected component C_B of $G[S^* \cap C^{\mathcal{R}(x)}]$: it contains $X_B \cup B = \beta^{\mathcal{R}(\tilde{s})}(a_B) - \beta^{\mathcal{R}(\tilde{s})}(\tilde{a}_B)$, which is nonempty due to $B \neq \emptyset$, and its neighborhood in $A^{\mathcal{R}(x)}$ is the set $N_B \subseteq \beta^{\mathcal{R}(x)}$. Observe that the graph induced by $\mathsf{extS}^{\tilde{s}}(\tilde{a}_B, \tilde{S})$ is of the same type as the graph induced by $\mathsf{extS}^{\tilde{s}}(\tilde{a}_B, S^* \cap W^{\mathcal{R}(\tilde{s})})$ (both with the boundary labelling $\iota^{\tilde{s}}|_{\beta^{\mathcal{R}(\tilde{s})}(\tilde{a}_B)}$). Therefore, due to $X_B \cup B$ being nonempty, by lemma 2.4.2 we infer that there exists a connected component $\tilde{C}_B$ of $G[S]$ such that

$$\tilde{C}_B = X_B \cup B \cup \bigcup \{C' \in \mathrm{cc}(G[\tilde{S}]) \mid a^{\tilde{s}}(C') = \tilde{a}_B\}$$

and

$$N_G(\tilde{C}_B) \cap A^{\mathcal{R}(x)} = N_B = N_G(C_B) \cap A^{\mathcal{R}(x)}. \tag{5.4}$$

Consequently, every connected component C of $G[S]$ that is disjoint with D' is also a

connected component of $G[\tilde{S}]$, and hence $N_G(C) \cap A^{\mathcal{R}(\tilde{s})} = N_G(C) \cap A^{\mathcal{R}(x)}$. Since $\tilde{S}$ is a feasible solution at $\tilde{s}$, we infer that S is a feasible solution at x. Furthermore, from (5.4) if follows that for every $B \in \mathcal{B}$ we have $a^x(\tilde{C}_B) = a$. Hence, $(\tilde{s}, \tilde{S})$ is liftable and S is the lift.

By induction, the weight of $\tilde{S}$ is not smaller than the weight of $S^* \cap W^{\mathcal{R}(\tilde{s})}$. As $D' \cup X_{\mathcal{B}} \subseteq S^*$, the weight of S is not smaller than the weight of $S^* \cap W^{\mathcal{R}(x)}$. Since S is liftable, the algorithm updates $\mathsf{S}[x, \mathsf{TypeTree}_S]$ with S, where $\mathsf{TypeTree}_S$ is the type assignment of S at node x.

Since $\tilde{S}$ and $S^* \cap W^{\mathcal{R}(\tilde{s})}$ are of the same type $\mathsf{TypeTree}^{\tilde{s}}$ at $\tilde{s}$, it follows from lemma 5.7.1 that S and $S^* \cap W^{\mathcal{R}(x)}$ are of the same type at x, that is, $\mathsf{TypeTree}_S = \mathsf{TypeTree}^x$. This finishes the induction step for branch ganders and completes the proof of the lemma. ■

5.7.4 Complexity analysis

We are left with arguing that the time complexity is as promised. By lemma 5.4.9, the subproblem tree generated by the recursion has depth $\mathcal{O}(n)$, $\mathcal{O}(\log^2 n)$ or $\mathcal{O}(\log^4 n)$ success branches on any root-to-leaf path depending on whether we work in P_t-free or $C_{>t}$-free regime, and $\mathcal{O}(\log n)$ split nodes on any root-to-leaf path.

Note that success branches are the only places where we add nodes to the tree decomposition $(T^{\mathcal{R}}, \beta^{\mathcal{R}})$. Furthermore, a success branch adds at most k nodes to the tree decomposition, one for each element of $\mathcal{B}$. Hence, at every node x we have $|V(T^{\mathcal{R}(x)})| = \mathcal{O}(\log^4 n)$ and $|V(T^{\mathcal{R}(x)})| = \mathcal{O}(\log^2 n)$ if G is P_t-free. As $|\mathsf{Types}| = \mathcal{O}(1)$, there are $2^{\mathcal{O}(\log^4 n)}$ type assignments to consider ($2^{\mathcal{O}(\log^2 n)}$ if G is P_t-free).

At a free node that is a grandchild of a branch node, the sets D', $\mathcal{B}$, X_B, N_B are of constant size. Consequently, every free node has a number of children bounded polynomially in n.

We infer that the whole subproblem tree has size $n^{\mathcal{O}(\log^4 n)}$. At every node, the algorithm spends time $n^{\mathcal{O}(\log^4 n)}$ inspecting all type assignments (or pairs of type assignments in the case of a split node). Both bounds improve to $n^{\mathcal{O}(\log^2 n)}$ if G is P_t-free. The running time bound follows, and hence the proof of theorem 5.1.1 is complete.

5.7.5 A generalization

We now give a slight generalization of theorem 5.1.1 that can be useful for expressing some problems that do not fall directly under its regime. The idea is that together with the solution S we would like to distinguish a subset $M \subseteq S$ that satisfies some CMSO_2-expressible predicate, and only the vertices of M contribute to the weight of the solution. The proof is a simple gadget reduction to theorem 5.1.1.

Theorem 5.7.3. *Fix a pair of integers d and t and a CMSO_2 formula $\varphi(X)$ with one free vertex subset variable. Then there exists an algorithm that, given a $C_{>t}$-free n-vertex graph G and a weight function $\mathfrak{w}\colon V(G) \to \mathbb{N}$, in time $n^{\mathcal{O}(\log^4 n)}$ finds subsets of vertices $M \subseteq S$ such that $G[S]$ is d-degenerate, $G[S] \models \varphi(M)$, and, subject to the above, $\mathfrak{w}(M)$ is maximum possible; the algorithm may also conclude that no such vertex subsets exist. The running time can be improved to $n^{\mathcal{O}(\log^2 n)}$ if G is P_t-free.*

Proof: For a graph G and a set $M \subseteq V(G)$, the *forked version* of (G, M) is the graph $\breve{G}^M$ created from G by attaching three degree-one neighbors to every vertex of $V(G)$ and, additionally, a two-edge path to every vertex of M. If G is weighted, then we assign weights to the vertices of $\breve{G}^M$ so that all of them are zero, except that for every $v \in M$ the other endpoint of the attached two-edge path inherits the weight of v.

Note that the vertices of $V(\breve{G}^M) - V(G)$ are exactly the vertices of degree one or two in $\breve{G}^M$; the vertices of $V(G)$ are of degree at least three in $\breve{G}^M$. This implies that if H' is a forked version of some other graph H and $M \subseteq V(H)$, then the pair (H, M) is defined

uniquely and is easy to decode:

- The vertices of H are exactly the vertices of H' that are of degree at least three, and H is the induced subgraph of H' induced by those vertices.
- Every vertex of H needs to be adjacent to exactly three vertices of degree 1 or to three vertices of degree 1 and one vertex of degree 2, which in turn has another neighbor of degree 1. The vertices of the latter category are exactly the vertices of M.

Note that if G is $C_{>t}$-free for some $t \geqslant 3$, then $\breve{G}^M$ is $C_{>t}$-free as well. Moreover, $\mathrm{tw}(\breve{G}^M) \leqslant \max(\mathrm{tw}(G), 1)$.

Let k, t, and φ be as in theorem 5.1.1. Construct a CMSO_2 sentence $\breve{\varphi}$ that for a graph H' behaves as follows:

- If H' is a forked version of some pair (H, M), then $H' \models \breve{\varphi}$ if and only if $H \models \varphi(M)$.
- Otherwise, $H' \not\models \breve{\varphi}$.

Writing $\breve{\varphi}$ in CMSO_2 is straightforward: we distinguish vertices of H and M as described above, and then apply φ relativized to those subsets of vertices.

Let $\varphi' \coloneqq \breve{\varphi}$ and $k' = k$ if $k \geqslant 2$, and $\varphi' \coloneqq \breve{\varphi} \wedge \varphi_{\mathrm{tw}<k}$ and $k' = 2$ if $k < 2$ (where the sentence $\varphi_{\mathrm{tw}<k}$ comes from lemma 2.4.3). We apply theorem 5.7.3 to k', t, and φ', and, given a graph G with weight function $\mathfrak{w}$, apply the asserted algorithm to $G' \coloneqq \breve{G}^{V(G)}$ (with weight function $\mathfrak{w}'$), obtaining a set S'. By the construction of G' and φ', $G'[S']$ must be a forked version of $(G[S], M)$ for some $M \subseteq S \subseteq V(G)$ such that $G[S] \models \varphi(M)$ and $G[S]$ has treewidth less than k. Moreover, $\mathfrak{w}'(S') = \mathfrak{w}(M)$.

We return (S, M). To see the correctness of this output, note that for every $M \subseteq S \subseteq V(G)$ such that $G[S]$ is of treewidth less than k and $G[S] \models \varphi(M)$, if we denote $H = G[S]$,

then $\breve{H}^M$ is an induced subgraph of G' of treewidth less than k', $\mathfrak{w}'(V(\breve{H}^M)) = \mathfrak{w}(M)$, and $\breve{H}^M \models \varphi'$. ■

For an example application of theorem 5.7.3, consider the Maximum Induced Cycle Packing problem: given an (unweighted) graph G, find the largest (in terms of cardinality) collection of pairwise non-adjacent induced cycles in G. Consider the following property of a graph G and a vertex subset $M \subseteq V(G)$: G is a disjoint union of cycles and every connected component of G contains exactly one vertex of M. It is straightforward to write a CMSO_2 formula $\varphi(M)$ such that $G \models \varphi(M)$ if and only if G and M have this property. Noting that disjoint unions of cycles are 2-degenerate, we can apply theorem 5.7.3 for the formula $\varphi(X)$ to conclude that the Maximum Induced Cycle Packing problem admits a $n^{\mathcal{O}(\log^4 n)}$-time algorithm on $C_{>t}$-free graphs, for every fixed t. Here, we endow the input graph with a weight function that assigns a unit weight to every vertex.

5.8 A simple technique for approximation schemes

In this final section we present a simple technique for turning polynomial-time and quasipolynomial-time algorithms for MWIS on P_t-free and $C_{>t}$-free graphs into PTASes and QPTASes for more general problem, definable as looking for the largest induced subgraph that belongs to some weakly hyperfinite class. Let us stress that this technique works only for unweighted problems.

We define the *blob graph* of a graph G, denoted G°, as the graph defined as follows:

$$V(G^\circ) \coloneqq \{X \subseteq V(G) \mid G[X] \text{ is connected}\},$$
$$E(G^\circ) \coloneqq \{X_1X_2 \mid X_1 \text{ and } X_2 \text{ are adjacent}\}.$$

The main combinatorial insight of this section is the following combinatorial property of G°. Let us point out that a similar result could be derived from the work of Cameron and Hell [111], although it is not stated there explicitly.

Theorem 5.8.1. *Let G be a graph. The following hold.*

(S1) The length of a longest induced path in G° is equal to the length of a longest induced path in G.

(S2) The length of a longest induced cycle in G° is equal to the length of a longest induced cycle in G, with the exception that if G has no cycle at all (G is a forest), then G° may contain triangles, but it has no induced cycles of length larger than 3 (i.e. it is a chordal graph).

Proof: Note that since G is an induced subgraph of G° (as witnessed by the mapping $u \mapsto \{u\}$), we only need to upper-bound the length of a longest induced path (resp., cycle) in G° by the length of a longest induced path (resp. cycle) in G.

Let $P^\circ = X_1, X_2, \ldots, X_t$ be an induced path in G°. We observe that the graph $G[\bigcup_{j=1}^{t} X_j]$ is connected and for each $j' \in [t-2]$ the sets $\bigcup_{j=1}^{j'} X_j$ and $\bigcup_{j=j'+2}^{t} X_j$ are nonadjacent.

Fix an induced path $P = v_1, v_2, \ldots, v_p$ in $G[\bigcup_{i=1}^{t} X_i]$. We define

$$\mathrm{set}(i) \coloneqq \max\{j \mid \{v_1, v_2, \ldots, v_i\} \cap X_j \neq \emptyset\}.$$

Claim 5.8.2. *For all $i \in [p-1]$ it holds that $\mathrm{set}(i+1) \in \{\mathrm{set}(i), \mathrm{set}(i)+1\}$.*

Proof of Claim 6. *It is clear that $\mathrm{set}(i+1) \geqslant \mathrm{set}(i)$, so suppose $\mathrm{set}(i+1) \geqslant \mathrm{set}(i)+2$. Since $v_i v_{i+1}$ is an edge of G, we conclude that there is an edge in G° between the sets $\{X_j \mid j \leqslant \mathrm{set}(i)\}$ and $\{X_j \mid j \geqslant \mathrm{set}(i)+2\}$, a contradiction with P° being induced.*

The following claim encapsulates the main idea of the proof.

Claim 5.8.3. *Let $P^\circ = X_1, X_2, \ldots, X_t$ be an induced path in G° such that $X_1 \not\subseteq X_2$. Let $X_1' \subseteq X_1 - X_2$ and $X_t' \subseteq X_t$ be nonempty sets. Let $P = v_1, v_2, \ldots, v_p$ be a shortest path in $G[\bigcup_{j=1}^{t} X_j]$ such that $v_1 \in X_1'$ and $v_p \in X_t'$. Then P is induced, $p \geqslant t$, and $\{v_2, v_3, \ldots, v_{p-1}\} \cap (X_1' \cup X_t') = \emptyset$.*

Proof of Claim 7. *The path P is induced and $\{v_2, v_3, \ldots, v_{p-1}\} \cap (X_1' \cup X_t') = \emptyset$ by the minimality assumption. Recall that X_1 must be disjoint with $\bigcup_{j=3}^{t} X_j$. Thus* $\mathrm{set}(1) = 1$ *and* $\mathrm{set}(p) = t$*, so the claim follows from claim 5.8.2.*

Now we are ready to prove (S1). Our goal is to prove that if G° contains an induced path on t vertices, then so does G. If $t = 1$, then the statement is trivial, so assume that $t \geqslant 2$ and let $P^\circ = X_1, X_2, \ldots, X_t$ be an induced path in G°.

If $X_1 \not\subseteq X_2$, then we are done by claim 5.8.3 applied to P° for $X_1' = X_1 - X_2$ and $X_t' = X_t$. So assume that $X_1 \subseteq X_2$ and note that $X_2 \not\subseteq X_1$, for X_1 and X_2 are two different vertices of P°. If $t = 2$, then any edge from X_1 to $X_2 - X_1$ is an induced path in G with two vertices; such an edge exists as $G[X_2]$ is connected. So from now on we may assume $t \geqslant 3$.

Let $X_2' \subseteq X_2 - X_1$ be such that $G[X_2']$ is a connected component of $G[X_2 - X_1]$ and X_2' and X_3 are adjacent. Such a set exists as X_3 is adjacent to X_2, but nonadjacent to X_1. Note that $G[X_2]$ being connected implies that there exists a nonempty set $X_2'' \subseteq X_2'$, such that every vertex from X_2'' has a neighbor in X_1. Furthermore, $X_2'' \cap X_3 = \emptyset$, as X_1 is nonadjacent to X_3. Observe that $\widehat{P}^\circ \coloneqq X_2', X_3, \ldots, X_t$ is an induced path in G° with at least $t - 1 \geqslant 2$ vertices, such that $X_2' \not\subseteq X_3$. Let $P' = v_2, v_3, \ldots, v_p$ be the induced path in G with at least $t-1$ vertices obtained by claim 5.8.3 applied to $\widehat{P}^\circ$, X_2'', and X_t. Now recall that $v_2 \in X_2''$, so there is $v_1 \in X_1$ adjacent to v_2. Note that v_1 is nonadjacent to every v_i for $i > 2$, because $v_i \notin X_2''$ for $i > 2$. Thus $P \coloneqq v_1, v_2, \ldots, v_p$ is an induced

path in G with at least t vertices.

Now let us prove (S2). We proceed similarly to the proof of (S1). If G° is chordal (every induced cycle is of length 3), then we are done by the exceptional case of the statement. Otherwise, let $C^\circ = X_1, X_2, \ldots, X_t$ be an induced cycle in G° for some $t \geqslant 4$; we want to find an induced cycle of length at least t in G. Note that $X_t \not\subseteq X_{t-1}$ and $X_t \not\subseteq X_1$, as otherwise C° is not induced. We observe that there are nonempty sets $X_t^1 \subseteq X_t$ and $X_t^{t-1} \subseteq X_t$, such that every vertex from X_t^1 has a neighbor in X_1 and every vertex from X_t^{t-1} has a neighbor in X_{t-1}. Let Q be a shortest path contained in X_t whose one endvertex, say x^1 is in X_t^1 and the other endvertex, say x^{t-1} is in X_t^{t-1}. Note that it is possible that $x^1 = x^{t-1}$. The minimality of Q implies that no vertex of Q, except for x^1, x^{t-1}, has a neighbor in $\bigcup_{j=1}^{t-1} X_j$.

Let P° be the induced path $X_1, X_2, \ldots, X_{t-1}$. Denote $X_1' \coloneqq N(x^1) \cap X_1$ and $X_{t-1}' \coloneqq N(x^{t-1}) \cap X_{t-1}$. Recall that both these sets are nonempty and $X_1' \cap X_2 = \emptyset$ and $X_{t-1}' \cap X_{t-2} = \emptyset$. Let $P = v_1, v_2, \ldots, v_p$ be the induced path given by claim 5.8.3 for P°, X_1', and X_{t-1}'. Recall that $p \geqslant t-1$. Now let C be the cycle obtained by concatenating P and Q, and observe that the cycle C is induced. Furthermore, as P has at least $t-1$ vertices and Q has at least one vertex, C has at least t vertices, which completes the proof. ■

Let us define an auxiliary problem called Maximum Induced Packing. An instance of Maximum Induced Packing is a triple $(G, \mathcal{F}, \mathfrak{w})$, where G is a graph, $\mathcal{F}$ is a family of connected induced subgraph of G, and $\mathfrak{w}\colon \mathcal{F} \to \mathbb{R}_+$ is a weight function. A *solution* to $(G, \mathcal{F}, \mathfrak{w})$ is a set $X \subseteq V(G)$, such that

- each connected component of $G[X]$ belongs to $\mathcal{F}$; and
- $\sum_{C\colon \text{ component of } G[X]} \mathfrak{w}(C)$ is maximized.

We observe the following.

Theorem 5.8.4. *Let $(G, \mathcal{F}, \mathfrak{w})$ be an instance of* Maximum Induced Packing*, where $|\mathcal{F}| = N$.*

1. *If G is P_t-free for some integer t, then the instance $(G, \mathcal{F}, \mathfrak{w})$ can be solved in time $N^{\mathcal{O}(\log^2 N)}$.*

2. *If G is $C_{>t}$-free for some integer t, then the instance $(G, \mathcal{F}, \mathfrak{w})$ can be solved in time $N^{\mathcal{O}(\log^4 N)}$.*

3. *If G is P_6-free or $C_{>4}$-free, then the instance $(G, \mathcal{F}, \mathfrak{w})$ can be solved in time $N^{\mathcal{O}(1)}$.*

Proof: Let G' be the subgraph of G° induced by $\mathcal{F}$. Clearly, G' has N vertices. We observe that solving the instance $(G, \mathcal{F}, \mathfrak{w})$ of Maximum Induced Packing is equivalent to solving the instance $(G', \mathfrak{w})$ of MWIS. Now the theorem follows from theorem 5.8.1 and the fact that MWIS can be solved in time $n^{\mathcal{O}(\log^2 n)}$ in n-vertex P_t-free graphs [1, 41], in time $n^{\mathcal{O}(\log^4 n)}$ in n-vertex $C_{>t}$-free graphs, using theorem 5.1.1 only for MWIS, and in polynomial time in P_6-free [89] or $C_{>4}$-free graphs [35]. ■

As an example of an application of theorem 5.8.4, we obtain the following corollary.

Corollary 5.8.5. *For every fixed d and t, given an n-vertex P_t-free graph G, in time $n^{\mathcal{O}(\log^2 n)}$ we can find the largest induced subgraph of G with maximum degree at most d.*

Proof: Note that every connected P_t-free graph with maximum degree at most d has at most d^t vertices. Thus, the family $\mathcal{F}$ of all connected induced subgraphs of G with maximum degree at most d has size at most $N \coloneqq n^{d^t}$ and can be enumerated in polynomial time. For each $F \in \mathcal{F}$ set $\mathfrak{w}(F) \coloneqq |V(F)|$. We may now apply theorem 5.8.4 to solve the instance $(G, \mathcal{F}, \mathfrak{w})$ of Maximum Induced Packing in time $N^{\mathcal{O}(\log^2 N)} = n^{\mathcal{O}(\log^2 n)}$. ■

Note that the strategy we used to prove theorem 5.8.4 cannot be used to solve Max Induced Forest in quasipolynomial time, as there can be arbitrarily larger P_t-free

tree; consider, for instance, the family of stars. However, it is sufficient to obtain a simple QPTAS for the unweighted version of the problem.

A class of graphs $\mathcal{C}$ is called *weakly hyperfinite* if for every $\varepsilon > 0$ there is $c(\varepsilon) \in \mathbb{N}$, such that in every graph $F \in \mathcal{C}$ there is a subset X of at least $(1-\varepsilon)|V(F)|$ vertices such that every connected component of $F[X]$ has at most $c(\varepsilon)$ vertices [107, Section 16.2]. Weakly hyperfinite classes are also known under the name *fragmentable* [112]. Every class closed under edge and vertex deletion which has sublinear separators is weakly hyperfinite [107, Theorem 16.5], hence well-known classes of sparse graphs, such as planar graphs, graphs of bounded genus, or in fact all proper minor-closed classes, are weakly hyperfinite.

For a class $\mathcal{C}$ of graphs, by Largest Induced $\mathcal{C}$-Graph we denote the following problem: given a graph G, find a largest induced subgraph of G, which belongs to $\mathcal{C}$. To make the problem well defined, we will always assume that $K_1 \in \mathcal{C}$. We can now conclude the following.

Theorem 5.8.6. *Let $\mathcal{C}$ be a nonempty, weakly hyperfinite class of graphs, which is closed under vertex deletion and disjoint union operations. Then, the* Largest Induced $\mathcal{C}$-Graph *problem*

1. *has a QPTAS in $C_{>t}$-free graphs, for every fixed t; and*
2. *has a PTAS in P_6-free graphs and in $C_{>4}$-free graphs.*

Proof: Let n be the number of vertices of the given graph G and let ε be the desired accuracy, i.e., the goal is to find a solution whose size is at least a $(1-\varepsilon)$-fraction of the optimum. Let $c \coloneqq c(\varepsilon)$.

Let X^* be the vertex set of an optimum solution. By the properties of $\mathcal{C}$, there exists $X' \subseteq X^*$ of size at least $(1-\varepsilon)|X^*|$ such that each connected component of $G[X']$ has at most c vertices. Let $\mathcal{F}$ be the set of all connected induced subgraphs of G that have

at most c vertices and belong to $\mathcal{C}$. Clearly $|\mathcal{F}| \leqslant n^c$ and $\mathcal{F}$ can be enumerated in polynomial time. For each $F \in \mathcal{F}$, we set $\mathfrak{w}(F) \coloneqq |V(F)|$.

Apply the algorithm of theorem 5.8.4 to solve the instance $(G, \mathcal{F}, \mathfrak{w})$ of Maximum Induced Packing in time $n^{\mathcal{O}(\log^4 n^c)} = n^{\mathcal{O}(\log^4 n)}$ if G is $C_{>t}$-free, or in polynomial time if G is P_6-free or $C_{>4}$-free. Let X be the optimum solution found by the algorithm. As $\mathcal{C}$ is closed under the disjoint union operation, we observe that $G[X]$ is a feasible solution to Largest Induced $\mathcal{C}$-Graph. Moreover we have $|X| \geqslant |X'| \geqslant (1-\varepsilon)|X^*|$. ■

Chapter 6

Graph Classes with Few Minimal Separators. I. Finite Forbidden Induced Subgraphs

A vertex set S in a graph G is a *minimal separator* if there exist vertices u and v that are in distinct connected components of $G - S$, but in the same connected component of $G - S'$ for every $S' \subset S$. A class $\mathcal{F}$ of graphs is called *tame* if there exists a constant c so that every graph in $\mathcal{F}$ on n vertices contains at most $O(n^c)$ minimal separators. If there exists a constant c so that every graph in $\mathcal{F}$ on n vertices contains at most $O(n^{c \log n})$ minimal separators the class is *strongly-quasi-tame*. If there exists a constant $c > 1$ so that $\mathcal{F}$ contains n-vertex graphs with at least c^n minimal separators for arbitrarily large n then $\mathcal{F}$ is called *feral*. The classification of graph classes into tame or feral has numerous algorithmic consequences, and has recently received considerable attention.

A key graph-theoretic object in the quest for such a classification is the notion of a *k-creature*. A k-creature consists of 4 disjoint vertex sets $A, B, X = \{x_1, \ldots, x_k\}, Y = \{y_1, \ldots y_k\}$ such that: (a) A and B are connected, (b) there are no edges from A to $Y \cup B$

and no edges from B to $X \cup A$, (c) A dominates X (every vertex in X has a neighbor in A) and B dominates Y and (d) $x_i y_j$ is an edge if and only if $i = j$. It is easy to verify that a k-creature contains at least 2^k minimal separators. On the other hand, in a recent article Abrishami et al. [43] conjecture that every hereditary class $\mathcal{F}$ that excludes k-creatures for some fixed constant k is tame.

In this paper we first give a counterexample to the conjecture of Abrishami et al. Our main result is a proof of a weaker form of their conjecture. More concretely, we prove that a hereditary class $\mathcal{F}$ is strongly quasi-tame if it excludes k-creatures for some fixed constant k and additionally every minimal separator can be dominated by another fixed constant k' number of vertices. The tools developed on the way lead to a number of additional results of independent interest.

(i) We obtain a complete classification of all hereditary graph classes defined by a finite set of forbidden induced subgraphs into strongly quasi-tame or feral. This substantially generalizes a recent result of Milanič and Pivač [113] , who classified all hereditary graph classes defined by a finite set of forbidden induced subgraphs on at most 4 vertices into tame or feral. **(ii)** We show that every hereditary class that excludes k-creatures and additionally excludes all cycles of length at least c, for some constant c, is tame. This generalizes the result of Chudnovsky et al. [114] who obtained the same statement for $c = 5$. **(iii)** We show that every hereditary class that excludes k-creatures and additionally excludes a complete graph on c vertices for some fixed constant c is tame.

6.1 Introduction

Let G be a graph and u and v be distinct vertices in G. A vertex set S is a *u,v-separator* if u and v are in distinct components of $G - S$. The set S is a *u,v-minimal separator* if S is a u,v-separator, but no proper subset of S is a u,v-separator. Finally,

S is a *minimal separator* if S is a u,v-minimal separator for some pair of vertices u and v. Minimal separators have a tremendous role in the design of graph algorithms, both directly, such as in the structural characterization of chordal graphs [75] but also indirectly in optimization algorithms for graph separation and routing problems (for example [115, 116, 100]). The theory of potential maximal cliques, developed by Bouchitté and Todinca [33] implies that a several fundamental graph problems, such as computing the *treewidth* and *minimum fill in* of a graph G can be done in time polynomial in the number of vertices of G and the number of minimal separators in G. Lokshtanov [117] showed that the same result holds for computing the *tree-length* of the graph G, while Fomin et al. [7] proved a general result that showed that a whole class of problems (including e.g. *maximum independent set* and *minimum feedback vertex set*) can be solved in time polynomial in the number of vertices and minimal separators of the graph. All of these algorithms require a list of all the minimal separators of G to be provided as input. However, the listing algorithms for minimal separators of Kloks and Kratsch [118] or Berry et al. [119] can be used to compute such a list in time polynomial in the number of vertices times a factor linear in the number of minimal separators of G.

This brings to the forefront the main question asked in this Chapter — *which classes of graphs have polynomially many minimal separators?* We will say that a graph class $\mathcal{F}$ is *tame* if there exists an integer c such that every graph in $\mathcal{F}$ on n vertices has at most $O(n^c)$ minimal separators. A number of important graph classes have been shown to be tame, such as Chordal [75] (and more generally Weakly Chordal [33]), Permutation (and, more generally d-Trapezoid [120]), Circular Arc [121] and Polygon Circle graphs [122]. Most of these results date back to the late 1990s and early 2000s. Much more recently [43, 123, 114, 42], research has started to focus on a more systematic classification of which graph classes are tame and which are not. Indeed the term *tame* for graph classes with polynomially many minimal separators was defined by Milanič and

Pivač [42], who classified all hereditary (closed under vertex deletion) classes defined by a set of forbidden induced subgraphs, all of which have at most 4 vertices, as tame or not tame.

Building on the terminology of Milanič and Pivač [42], we will say that a class of graphs $\mathcal{F}$ is *quasi-tame* if there exist constants c, c' such that every n-vertex graph in the family contains at most $O(n^{c \log^{c'} n})$ minimal separators. Further, $\mathcal{F}$ is *strongly quasi-tame* if it is quasi-tame with $c' \leqslant 1$. On the opposite side of the spectrum, we will say that $\mathcal{F}$ is *feral* if there exists a constant c such that for every $N \geqslant 0$ there exists an $n \geqslant N$ such that $\mathcal{F}$ contains an n-vertex graph with at least c^n minimal separators.

Abrishami et al. [43] define a structure, called a k-creature, the presence of which appears to control, to a large extent, whether a graph has many or few (quasi-polynomially many) separators. A *k-creature* in a graph G is a four-tuple $(A, B, X = \{x_1, x_2, \ldots, x_k\}, Y = \{y_1, y_2, \ldots, y_k\})$ of mutually disjoint vertex subsets of $V(G)$, satisfying the following conditions (see Figure 6.1).

1. A and B induce connected subgraphs of G,
2. A and $Y \cup B$ are *anti-complete* (i.e., no vertex in A is adjacent to a vertex in $B \cup Y$) and B is anti-complete with $X \cup A$.
3. A dominates X (every vertex in X has a neighbor in A) and B dominates Y, and
4. $x_i y_j$ is an edge if and only if $i = j$.

A graph G is *k-creature-free* if there does not exists a 4-tuple of vertex sets of $V(G)$ that form a k-creature. It is easy to see that a k-creature contains at least 2^k minimal separators (select precisely one of $\{x_i, y_i\}$ for every $i \leqslant k$). Because deleting a vertex cannot increase the number of minimal separators, a graph G that contains a k-creature contains at least 2^k minimal separators. Thus, a graph family $\mathcal{F}$ that contains n-vertex graphs with k-creatures for arbitrarily large n and with $k = \Omega(n)$ is feral. For $\mathcal{F}$ to not

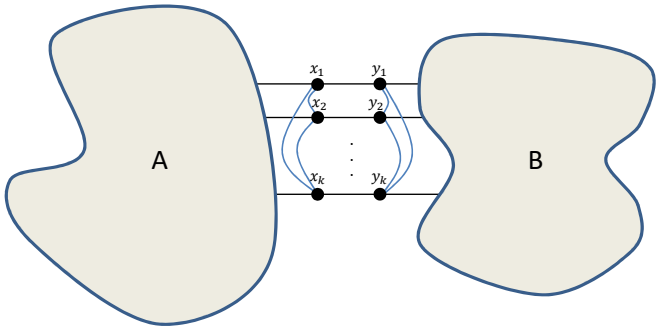


Figure 6.1: A graph induced by the vertices of a k-creature. The blue edges indicate that x_i (y_i) may or may not be a neighbor of x_j (y_j)

be tame it is sufficient for k to grow super-logarithmically with n (i.e $n \in 2^{o(k)}$). A sort of converse to this observation was conjectured in [43].

Conjecture 6.1.1. *[43] For every fixed natural number k, the family of graphs that are k-creature-free is tame.*

Even if Conjecture 6.1.1 were to be true, it would still not give a complete characterization of hereditary graph classes into tame or non-tame. In particular Abrishami et al. [43] give an example of a tame hereditary class $\mathcal{F}$ that contains k-creatures for arbitrarily large k. Their example can also be slightly modified to show that there exist hereditary families that are neither tame nor feral. This makes it appear that, at least for hereditary classes in their full generality, the boundary between tame and non-tame graph classes is so "strange-looking" that a complete dichotomy may be out of reach, and that we therefore have to settle for sufficient conditions for tameness / non-tameness, and possibly complete characterizations for more well-behaved sub-classes of hereditary families. For an example, Conjecture 6.1.1, if true, would have yielded a complete dichotomy into tame or feral for all classes of graphs closed under *induced minors* (i.e closed under vertex deletion and edge contraction).

Unfortunately it turns out that **Conjecture 6.1.1 is false**. In particular we give (in Section 6.4) an example of a feral family $\mathcal{F}$ that excludes 100-creatures. The family

$\mathcal{F}$ consists of all *k-twisted ladders* (see Section 6.4 for a definition). Our main result is nevertheless that Conjecture 6.1.1 is true "in spirit", in the sense that for large classes of hereditary families, excluding k-creatures does imply few minimal separators. To state Theorem 6.1.2 we need to define *k-skinny-ladders*. A *k-skinny-ladder* is a graph G consisting of two anti-complete paths $P_l = \ell_1\ell_2 \dots \ell_k$ and $P_r = r_1r_2 \dots r_k$ and a set $\{s_1, s_2, \dots, s_k\}$ of vertices such that for every i, s_i is adjacent to ℓ_i and r_i and to no other vertices.

Theorem 6.1.2. *For every natural number k, the family of graphs that are k-creature-free and do not contain a k-skinny-ladder as an induced minor is strongly-quasi-tame.*

Theorem 6.1.2 suggests that other counterexamples to Conjecture 6.1.1 should resemble the counterexample we provide in Section 6.4. Furthermore, we do not have an an example of a non-tame class for which strong quasi-tameness follows from Theorem 6.1.2. Therefore we conjecture that the statement of Theorem 6.1.2 remains true even if strongly quasi-tame is replaced by tame.

Excluding the k-skinny-ladder is closely tied to *domination* of minimal separators. A vertex set X *dominates* S if every vertex in S is either in X or has a neighbor in X. An important ingredient in the proof of Theorem 6.1.2 (see Lemma 6.5.15) is that for every k there exists a k' such that if G excludes k-creatures and excludes k-skinny-ladders as an induced minor then every minimal separator S in G is dominated by a set X on at most k' vertices. In fact, because a k-skinny-ladder is itself 5-creature-free and contains a minimal separator (namely the set $\{s_1, s_2, \dots, s_k\}$) which cannot be dominated by $k-1$ vertices, among the hereditary classes $\mathcal{F}$ that exclude k-creatures, the presence or absence of k-skinny-ladders (as induced minors) precisely characterizes whether every minimal separator of every graph in $\mathcal{F}$ can be dominated by a constant size set of vertices.

While the statement of Theorem 6.1.2 is concise, it is not immediately clear which

graph families it applies to. Which families are k-creature-free? What does it mean in terms of forbidden induced subgraphs to exclude a k-skinny-ladder as an induced minor? In the second half of this chapter we obtain an equivalent characterization of the premise of Theorem 6.1.2 in terms of forbidden induced subgraphs. Specifically, we first show that for every $k \geqslant 1$ there exists a k' such that if G contains a k'-creature then G contains a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, or a k-ladder as an induced subgraph (see Figure 6.2, formal definitions in Section 6.6). Additionally, it is easy to see that every graph that contains a $2k$-skinny-ladder as an induced minor either contains a k-ladder or a *k-contracted-ladder* as an induced subgraph. Here a *k-contracted-ladder* is a graph obtained from a k-ladder by contracting all of the horizontal paths into single vertices (see Section 6.6 for a formal definition). This leads to the following variant of Theorem 6.1.2.

Theorem 6.1.3. *For every natural number k, the family of graphs that exclude the k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, k-ladder, and the k-contracted-ladder as induced subgraphs, is strongly-quasi-tame.*

Theorems 6.1.2 and 6.1.3 are equivalent in the sense that for every k there exists a k' such that the graph family that satisfies the premise of Theorem 6.1.2 with k also satisfies the premise of Theorem 6.1.3 with k', and the graph family that satisfies the premise of Theorem 6.1.3 with k also satisfies the premise of Theorem 6.1.2 with k'.

To demonstrate the power of Theorem 6.1.2 (or equivalently, Theorem 6.1.3) we show that it gives, as a pretty direct consequence, a complete classification of all hereditary graph classes defined by a finite set of forbidden induced subgraphs into strongly quasi-tame or feral. Indeed, it is an easy exercise to show that if a family $\mathcal{F}$ is defined by a finite set of forbidden induced subgraphs and contains k-skinny-ladders for arbitrarily large k as induced minors, then there exists a constant p such that $\mathcal{F}$ either contains

all p-subdivisions of 3-regular graphs (an *p-subdivision* of G is the graph obtained from G by replacing each edge of G by a path on $p+1$ edges) or all line graphs (see [124] for a definition) of p-subdivisions of 3-regular graphs. In this case $\mathcal{F}$ is feral. Therefore, Theorem 6.1.2 proves Conjecture 6.1.1 for hereditary graph classes defined by a finite set of forbidden induced subgraphs, albeit with strongly quasi-tame instead of tame.

The "strongly quasi-tame" part of the classification of families $\mathcal{F}$ defined by a finite set of forbidden induced subgraphs into strongly quasi-tame or feral follows directly by inspecting the graphs in the statement of Theorem 6.1.3. The "feral" part follows by observing that if $\mathcal{F}$ contains some of the graphs in the premise of Theorem 6.1.3 for arbitrarily large k, then $\mathcal{F}$ must also contain such graphs with only $O(k)$ vertices. This part of the proof crucially depends on $\mathcal{F}$ being defined by a *finite* set of forbidden induced subgraphs.

Theorem 6.1.4. *Let $\mathcal{F}$ be a graph family defined by a finite number of forbidden induced subgraphs. If there exists a natural number k such that $\mathcal{F}$ forbids all k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, k-claw, and k-paw graphs, then $\mathcal{F}$ is strongly-quasi-tame. Otherwise $\mathcal{F}$ is feral.*

Note that some of the graphs of Figure 6.2 share a name with graphs that appear in the work of Abrishami et al. [43], but the definitions given here are slightly different. In particular, in some of the places where they require single edges we allow arbitrarily long paths. Abrishami et al. [43] prove that the family of (what they define to be) theta-free, pyramid-free, prism-free, and turtle-free graphs is tame. We remark that our results are incomparable to theirs, in the sense that there are classes of graphs whose tameness follows from their work, but not ours, and vice versa.

Theorem 6.1.4 substantially generalizes the main result of Milanič and Pivač [42], who obtained a complete classification into tame or feral of hereditary graph classes

characterized by forbidden induced subgraphs on at most 4 vertices. The generalization comes at a price - as our upper bounds on the number of minimal separators are quasi-polynomial instead of polynomial.

Next we explore for which classes we are able to improve our quasi-polynomial upper bounds to polynomial ones. Here, again, domination plays a crucial role. We show that for every pair k, k' of integers, every class of graphs that excludes k-creatures and additionally has the property that every minimal separator S is dominated by a vertex set X of size at most k' and *disjoint from* S is tame. We then proceed to show that graphs that exclude k-creatures and all cycles of length at least r for any choice of natural numbers k and r have this property, leading to Theorem 6.1.5.

Theorem 6.1.5. *For every pair of natural numbers k and r, the family of graphs that are $C_{\geqslant r}$-free, k-theta-free, k-prism-free, and k-pyramid-free is tame.*

Here a graph G is *$C_{\geqslant r}$-free* if it contains no induced cycles of length at least r. Theorem 6.1.5 is optimal in the sense that k-theta, k-prism, and k-pyramid graphs have at least 2^{k-2} minimal separators and therefore can have exponentially many minimal separators. Further, it substantially strengthens the results of Chudnovsky et al. [114], who prove the same statement but only for $r = 5$.

Finally we show that graph classes that exclude k-creatures, k-skinny-ladders, as well as k-cliques satisfy the property that every minimal separator S can be dominated by a constant size set X disjoint from S. This implies that this family of graphs is tame as well.

Theorem 6.1.6. *For any fixed natural number k, the family of graphs that are k-creature-free, contain no k-skinny-ladder as an induced minor, and contain no minimal separator that has a clique of size k is tame.*

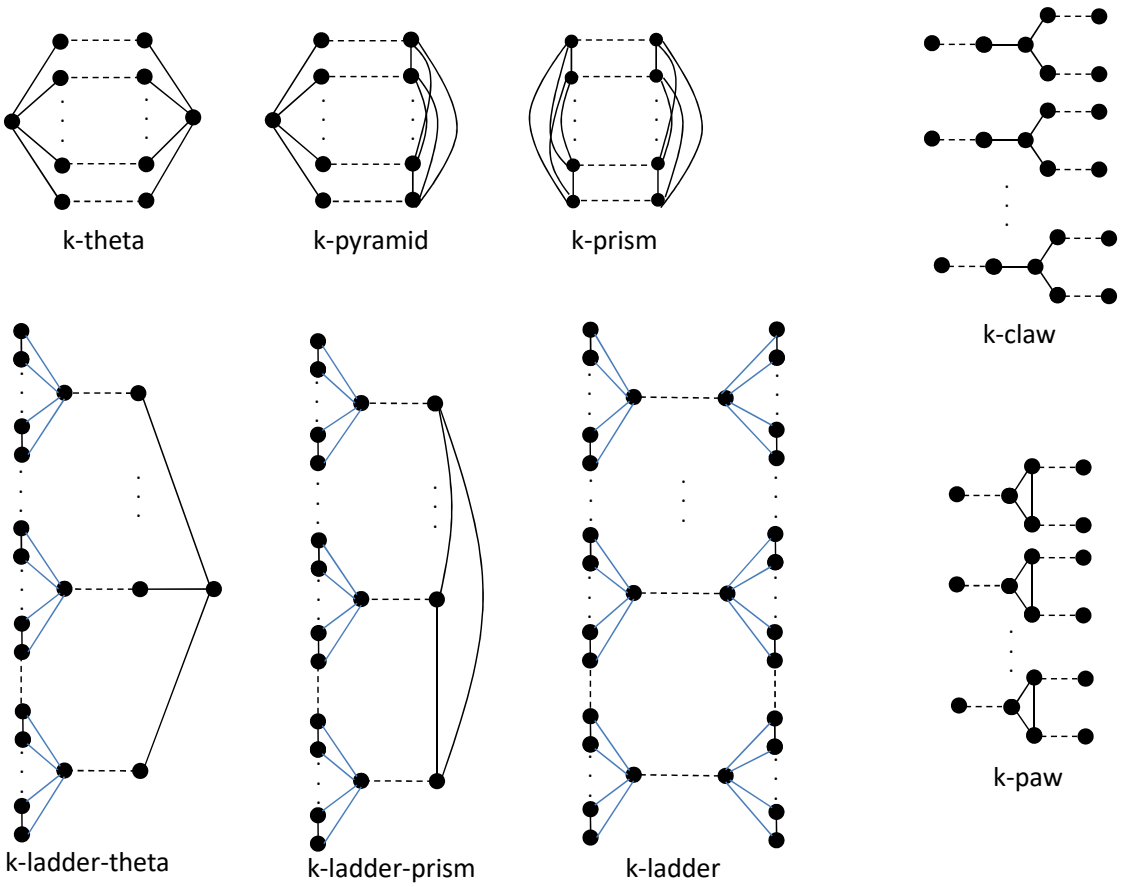


Figure 6.2: Dashed lines represent the option of having an arbitrary length path or just an edge (except for k-claw and k-paw graphs which the dotted line is always a path of length k.) The blue lines used in the k-ladder-theta, k-ladder-prism, and k-ladder graphs represents the option of either having or not having that edge, but for each vertex incident to more than one of the blue edges, at least one of those blue edges must belong to the graph.

Theorem 6.1.4 provided a classification of all hereditary graph classes defined by a finite set of forbidden induced subgraphs into strongly quasi-tame or feral. In the same way that Theorem 6.1.4 is a fairly direct consequence of Theorem 6.1.3, we can obtain from Theorem 6.1.6 a complete dichotomy of all hereditary graph classes defined by a finite set of forbidden induced subgraphs, and additionally exclude at least one clique, into *tame* or feral.

Theorem 6.1.7. *Let $\mathcal{F}$ be a graph family defined by a finite number of forbidden induced subgraphs. If there exists a natural number k such that $\mathcal{F}$ forbids all k-clique, k-theta, k-ladder-theta, k-claw, and k-paw graphs then $\mathcal{F}$ is tame. Otherwise, $\mathcal{F}$ contains all cliques or $\mathcal{F}$ is feral.*

Subsequent Work. There have been two significant developments since the first version of this manuscript. The first is a manuscript by Gajarský et al. [44] which answers Conjectures 6.9.1 and 6.9.2 in the affirmative, that is, they prove that "strongly-quasi-tame" can be replaced by "tame" in the statements of Theorems 6.1.2, 6.1.3, and 6.1.4, respectively.

The proof of Gajarský et al. builds heavily on top of the results in this chapter. In particular, their proof of Conjecture 6.9.1 (i.e the strenthening of our Theorem 6.1.2 from quasi-tame to tame) requires all the tools that we develop in the proof of Theorem 6.1.2, except that the last crucial piece of our proof, namely Lemma 6.5.16, is replaced by a remarkably elegant argument that yields a polynomial upper bound on the number of minimal separators, rather than a quasi-polynomial one. Given Theorem 6.1.2, the proof of Theorem 6.1.3, which is given in Section 6.6, amounts to characterizing the inclusion minimal hereditary families $\mathcal{F}$ that contain a k-creature for every integer k. Thus, given that Conjecture 6.9.1 is true, the truth of Conjecture 6.9.2 follows directly from our proof of Theorem 6.1.3 as given in Section 6.6.

Our Theorems 6.1.5, 6.1.6 and 6.1.7 all give polynomial upper bounds for a subset of the graph classes covered by Theorems 6.1.2 and 6.1.3. Thus the proof of Gajarský et al. [44] that Conjectures 6.9.1 and 6.9.2 are true completely subsumes Theorems 6.1.5, 6.1.6 and 6.1.7. We nevertheless keep their statements and proofs in this chapter, both because their proofs pre-dates the proof of Gajarský et al., and because they retain some (if arguably small) value. Specifically the proofs of Theorems 6.1.5, 6.1.6 and 6.1.7 all work a manner similar to the proof of Conjecture 6.9.1 by Gajarský et al., namely by replacing Lemma 6.5.16 by a polynomial upper bound. Our "replacements of Lemma 6.5.16" are slightly simpler than the one by Gajarský et al. Of course our proofs of Theorems 6.1.5, 6.1.6 and 6.1.7 only replace the quasi-polynomial bound of Lemma 6.5.16 by a polynomial upper bound for different special cases, while Gajarský et al., (essentially) do it for Lemma 6.5.16 in its full generality.

The second development concerns Conjecture 6.9.3, which conjectures that every induced-minor-closed class $\mathcal{F}$ is either tame or feral. It turns out that Conjecture 6.9.3 is false by a counterexample that combines the features of the counterexample to Conjecture 6.1.1 given in section 6.4 of this chapter with the construction that shows that there exist hereditary families that are neither feral nor tame.

However, a much more general statement (that avoids the special cases which make Conjecture 6.9.3 false) is true. In Chapter 7 we show that every hereditary graph class which is definable in Monodic Second Order Logic (CMSO_2 Logic) is either quasi-tame or feral.

In terms of generality the result of Chapter 7 completely subsumes Theorems 6.1.2 and Theorem 6.1.3, at a cost of the quasi-polynomial bound on the number of minimal separators being much worse (about $n^{O(\log^{17} n)}$, as opposed to $n^{O(\log n)}$). The proof of the main result in Chapter 7 requires some, but far from all, tools in the present chapter (namely Lemma 6.5.6 and the entire characterization of the inclusion minimal hereditary

families $\mathcal{F}$ that contain a k-creature for every integer k, given in Section 6.6).

More importantly, the proof of the main result in Chapter 7 is very complex (spanning close to 200 pages of this thesis), and appears to be very difficult to strengthen to a polynomial upper bound, leaving the polynomial bound of Gajarský et al. [44] as highly relevant. Therefore all of the main contributions of the present chapter (the proofs of Theorems 6.1.2 and 6.1.3, with exception of Lemma 6.5.16) are crucial to either the proof Gajarský et al. [44] of Conjectures 6.9.1 and 6.9.2, or the proof of the dichotomy for CMSO-definable hereditary classes of Chapter 7, or both.

Outline of the chapter. In Section 6.2 we give a high level overview of our proofs. In Section 6.3 we set up the standard definitions and notations used in the chapter. In Section 6.4 we give the counterexample to Conjecture 6.1.1. In Section 6.5 we prove our main result, Theorem 6.1.2. In Section 6.6 we characterize the premise in the statement of Theorem 6.1.2 (being k-creature-free and k-skinny-ladder induced minor-free) in terms of forbidden induced subgraphs, and use this characterization to prove Theorems 6.1.3 and 6.1.4. In Sections 6.7 and 6.8 we prove the polynomial bounds on the number of minimal separators in graphs that are both k-creature-free and long cycle-free, and in graphs that are k-creature-free, k-skinny-ladder induced minor-free, and k-clique-free. We conclude with some open problems in Section 6.9.

6.2 Overview

In this section we provide high level overview of our proofs. We will give quite detailed proof sketches of some of the pivotal steps, while skipping technical details of the more cumbersome parts. We start with the main ideas behind the proof of Theorem 6.1.2.

6.2.1 Overview of the Proof of Theorem 6.1.2.

Recall that Theorem 6.1.2 states that for every natural number k, the family of graphs that are k-creature-free and do not contain a k-skinny-ladder as an induced minor is strongly-quasi-tame. There are three key lemmas that lie at the heart of the proof of Theorem 6.1.2. The first of these states that for a k-creature-free graph G, there are at most n^k distinct ways for the neighborhood of a vertex v to intersect the minimal separators S of G, where $n = |V(G)|$.

Claim 6.2.1. *Let G be a k-creature-free graph with $n = |V(G)|$, $v \in G$, and let $S^v = \{N(v) \cap S : v \notin S$ and S is a minimal separator of $G\}$. Then $|S^v| \leqslant n^k$.*

Claim 6.2.1 is stated as Lemma 6.5.6 in the formal proof. Note that Claim 6.2.1 on its own does not imply that the number of minimal separators of G is bounded, only that the number of ways the neighborhood of a vertex can intersect the minimal separators of G is polynomial. In fact the counterexample given in Section 6.4 shows that the number of minimal separators of a k-creature-free graph of maximum degree at most 3 can be exponential.

The proof of Claim 6.2.1 is based on VC-dimension (see Definition 6.5.3) and follows from the Sauer-Shelah Lemma (see Lemma 6.5.4). In particular, if $|S^v|$ is large then there exists a vertex u not adjacent to v such that there are at least n^{k-1} distinct intersections $N(v) \cap S'$, where S' is a u, v-minimal separator. By the Sauer-Shelah Lemma there is a subset of $N(v)$ of size k that is shattered by the sets of the form $N(v) \cap S'$.

From the definition of shattering it follows that there are vertices $X = \{v_1, \ldots, v_k\}$ in $N(v)$ such that each v_i belongs to a private u, v-minimal separator S_i, i.e., $X \cap S_i = \{v_i\}$. Now, let C_i be the component that u belongs to in $G - S_i$, let v_i' be a neighbor of v_i in C_i with minimum distance to u, and let P_i be a shortest path from v_i' to u in C_i. Notice that no vertex of P_i can be neighbors with v_j for $i \neq j$ or else there would be a u, v

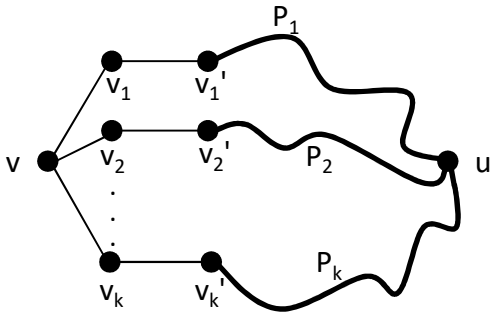


Figure 6.3: The k-creature formed in Claim 6.2.1 and Lemma 6.5.6. Note there may or may not be edges between the v_i's and the P_i's may overlap or have edges between them as well.

path in $G - S_i$, and v_i cannot be neighbors with u, or else there would be a u, v path in $G - S_j$ for $j \neq i$. It then follows that v together with the set X and the P_i's make a k-creature (See Figure 6.3). So, for any fixed u, v, there are at most n^{k-1} unique sets of the form $N(v) \cap S$ where S is a u, v-minimal separator. Finally, it is easy to check that for every u-w minimal separator S there exists some v-w minimal separator or v-u minimal separator S' such that $N(v) \cap S = N(v) \cap S'$, proving the claim.

The second ingredient in the proof of Theorem 6.1.2 states that the minimal separators of graphs that are k-creature-free and have no k-skinny-ladder as an induced minor can be dominated by few vertices.

Claim 6.2.2. *Let $\mathcal{F}$ be a graph family that forbids k-creatures and has no k-skinny-ladder as an induced minor, then there exists a constant c such that for all graphs $G \in \mathcal{F}$, every minimal separator of G can be dominated by c vertices.*

The proof of Claim 6.2.2 (re-stated as Lemma 6.5.15 in the formal proof) is substantially more involved than the proof of Claim 6.2.1. Indeed the full proof of Lemma 6.5.15 takes up the bulk of Section 6.5. The overall strategy of the proof of Claim 6.2.2 is to start with the assumption that G is a graph and S is a minimal separator in G that cannot be dominated by c vertices and use this assumption to show the existence of either a

k-creature or a k-skinny-ladder in G for sufficiently large k. Here sufficiently large means that k tends to infinity when c tends to infinity.

The proof is carried out in a sequence of steps, where each step "zooms in" on a more structured induced subgraph of G which still has a minimal separator (which is a subset of the original separator S) that cannot be dominated by c' vertices for some sufficiently large c'. As an example step let C_u and C_v be two full components of $G - S$ (a *full component* of $G - S$ is a component C such that $N(C) = S$). Without loss of generality $V(G) = C_u \cup S \cup C_v$, because if $G - S$ also contains some other component C then S is still a minimal separator in $(G - C) - S$ and S still cannot be dominated by c vertices in $G - C$.

The first step of the proof of Claim 6.2.2 is to reduce to the case where $G - S$ has precisely two components P_L and P_R and both P_L and P_R induce paths. While this sounds like a pretty strong claim this is actually one the less technical steps in the proof of Claim 6.2.2. The idea is to look for a k-creature (A, X, Y, B) where $A \cup X \subseteq C_u$, $Y \subseteq S$ and $B = C_v$. If we fail to find a k-creature of this form then one can find $k - 1$ induced paths $P_1 \dots P_{k-1}$ in C_u that together dominate S (see Lemma 6.5.8). A symmetric argument shows the existence of induced paths $Q_1 \dots Q_{k-1}$ in C_v that together dominate S. Since S is completely covered by at most k^2 sets on the form $N(P_i) \cap N(Q_j)$, it follows that there must exist some pair i, j such that $S \cap N(P_i) \cap N(Q_j)$ cannot be dominated by c/k^2 vertices. We now consider $G[P_i \cup (S \cap N(P_i) \cap N(Q_j)) \cup Q_j]$, in this graph the minimal separator $(S \cap N(P_i) \cap N(Q_j))$ cannot be dominated by c/k^2 vertices and the two full components are $P_L = P_i$ and $P_R = Q_j$.

The next sequence of steps (Lemmas 6.5.9, 6.5.10, and ultimately 6.5.11) show that it is possible to zoom in on a sub-path P'_L of P_L, a sub-path P'_R of P_R and a subset $I \subseteq S$ of size at least c' (where c' is lower bounded by an unbounded function of c) such that both P'_L and P'_R dominate I, I is an independent set (no pair of vertices in I are

adjacent) and furthermore no vertex in P'_L or P'_R have more than one neighbor in I. Note that I is now a minimal separator in $G[P'_L \cup I \cup P'_R]$. The additional properties of I witness that I cannot be dominated by less than $|I| \geqslant c'$ vertices, because no vertex of $G[P'_L \cup I \cup P'_R]$ can dominate more than one vertex in I. Thus, Lemmas 6.5.9, 6.5.10, and 6.5.11 allow us to reduce the proof of Claim 6.2.2 from the general case where S cannot be dominated by few vertices, but we do not know why, to the special case where S cannot be dominated by few vertices because no vertex in G dominates more than one vertex of S. The proofs of Lemmas 6.5.9, 6.5.10, and 6.5.11 are fairly technical, and we skip them in this overview.

Assuming Lemma 6.5.11 we are in the following setting. Our graph G consists of an independent set S of size at least c, which is much larger than k and two paths P_L and P_R that both dominate S. Further, no two vertices in S have any common neighbor. Our goal is to find a k-skinny-ladder as an induced minor in G. Observe that the graph G already kind of looks like a skinny-ladder. The main problem is that each of the vertices of S can have many neighbors in P_L and in P_R and that these neighbors can "interleave" a lot (see e.g. Figure 6.8). The next series of lemmas — namely Lemmas 6.5.12, 6.5.13, and 6.5.14, culminating with 6.5.15 — show that if the neighbors of the vertices in S interleave "too much", then we can find a k-creature in G, while if they do not then G contains a k-skinny-ladder.

Claim 6.2.1 together with Claim 6.2.2 are almost enough to prove Theorem 6.1.2. Suppose that instead of Claim 6.2.2 we had the stronger statement that for every minimal separator S in every k-creature-free, k-skinny-ladder induced minor-free graph there is a dominating set D of size c such that D is disjoint from S. [1] In this hypothetical scenario we can give a simple proof of a statement stronger than Theorem 6.1.2 — a

[1] This claim is actually false: for any $k \geqslant 1$ start with a k-skinny-ladder for and turn $\{s_1, \ldots, s_k\}$ into a clique. It is easy to check that this graph does not contain a 5-creature or a 5-skinny-ladder (as an induced minor), while no set of size less than k disjoint from S can dominate S.

polynomial upper bound on the number of minimal separators of k-creature-free, k-skinny-ladder induced minor-free graphs. Suppose for contradiction that the number of minimal separators is super-polynomial. By the dream-claim there is some constant size set D such that there are super-polynomially many minimal separators S that are disjoint from D and dominated by D. By Claim 6.2.1 each vertex $v \in D$ has only polynomially many options for the intersection $N(v) \cap S$. But then there must be two distinct minimal separators S_1 and S_2 that are disjoint from D, dominated by D, and that satisfy $S_1 \cap N(v) = S_2 \cap N(v)$ for every vertex v in D. But D dominates S_1 and S_2, and therefore we have

$$S_1 = \bigcup_{v \in D} N(v) \cap S_1 = \bigcup_{v \in D} N(v) \cap S_2 = S_2$$

contradicting that S_1 and S_2 are different minimal separators.

The final ingredient of the proof of Theorem 6.1.2 is a strengthening of this argument that also works for the case when the dominating set D is not necessarily disjoint from S. This strengthening comes at the cost that we are only able to prove a quasi-polynomial upper bound on the number of minimal separators.

Claim 6.2.3. *There exists a function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that the following holds. Let G be a graph with n vertices and let k and k' be integers such that for all induced subgraphs G' of G and for all $v \in G'$, if $S^v_{G'} = \{N(v) \cap S : v \notin S$ and S is a minimal separator of $G'\}$, then $|S^v_{G'}| \leqslant n^k$ and every minimal separator of every induced subgraph of G can be dominated by k' vertices. Then G has at most $n^{f(k,k')\log(n)}$ minimal separators where $n = |V(G)|$.*

Claim 6.2.3 is stated as Lemma 6.5.16 in the formal proof (we note that for technical reasons the statement of Lemma 6.5.16 slightly differs from that of Claim 6.2.3). Note

that by Claims 6.2.1 and 6.2.2, graphs that are k-creature-free and forbid k-skinny-ladders as an induced minor satisfy the premise of Claim 6.2.3.

The basic idea of the proof is to use a recursive branching algorithm that outputs all of the minimal separators of the graph, and upper bound the total number of sets output by this algorithm. The algorithm takes a tuple (G, X) where G is a graph that satisfies the conditions of Claim 6.2.3, and $X \subseteq V(G)$ and returns all minimal separators of G contained in X (and possibly other sets as well). Initially the algorithm is called with $X = V(G)$. We will measure the running time of the algorithm in terms of n and an upper bound x on the size of X. Initially we have $x = n$.

Fix some minimal separator S of G that is contained in X. We set Q to be the set of vertices of G that have at least $\frac{1}{2k'}|X|$ neighbors in X. The reason for choosing this particular fraction will become apparent shortly. By assumption, for each $q \in Q$ if $q \notin S$ then there are at most n^k options as to what $N[q] \cap S$ is. For each option $Y \in S_G^q$ we call the algorithm on $(G - Y, X - N[Y])$, and for each set S' that is returned by the call $(G - Y, X - N[q])$, we add $S' \cup Y$ to our collection of sets that we will return. If $Y' \in S_G^v$ is equal to $N[q] \cap S$, then $S - Y'$ is a minimal separator of $G - Y'$ contained in $X - N[q]$ and so $(S - Y') \cup Y' = S$ will be included in the list of sets that we return. In each of our branches we are calling the algorithm on $X' = X - N[q]$, and since q has at least $\frac{1}{2k'}|X|$ neighbors in X, X' is a constant fraction smaller than X. Thus the running time of (and the number of sets output by) the algorithm is governed by the recurrence $T(n, x) \leqslant n^{k+O(1)}T(n, x(1 - \frac{1}{2k'}))$, which solves to $n^{O(\log x)} \leqslant n^{O(\log n)}$ for fixed k and k'.

But what if $Q \subseteq S$? To handle this case we use that fact that $S - Q$ must then be a minimal separator of $G - Q$ and by assumption there are at most k' vertices of $G - Q$ that dominate $S - Q$. We can now see why the fraction $\frac{1}{2k'}$ was used to define Q; the neighborhood of these k' vertices contain at most $1/2$ the vertices of X. Thus, for every set R of k' vertices of G, we call the algorithm on $(G - Q, (X - Q) \cap N(R))$. For each

set S' that is returned from the call $(G - Q, (X - Q) \cap N(R))$, we add $S' \cup Q$ to the list of sets output by the algorithm. Since there is some set R' of k' vertices in $G - Q$ such that R' dominates $S - Q$, S will get added to the list. Each of the recursive calls invoke the algorithm on $X' = X \cap N[R]$, and $|X'| \leqslant .5|X|$. In this case the running time of (and the number of sets output by) the algorithm is governed by the recurrence $T(n, x) \leqslant n^{k'+k}T(n, \frac{x}{2})$, which also solves to $n^{O(\log x)} \leqslant n^{O(\log n)}$ for fixed k and k'. This completes the sketch of the proof of Claim 6.2.3 and therefore also of Theorem 6.1.2.

6.2.2 Overview of the Proof of Theorems 6.1.3 and 6.1.4.

The conclusion of Theorem 6.1.2 is simple - G only has a quasi-polynomial number of minimal separators. On the other hand the premise is somewhat opaque. It is not immediately obvious which graphs contain k-creatures for arbitrarily large k, and which graphs contain a k-skinny-ladder as an induced minor. In the second part of the paper we re-formulate the premise of Theorem 6.1.2 in terms of forbidden induced subgraphs. More concretely, the bulk of the work in Section 6.6 goes into proving the following statement.

Claim 6.2.4. *For every natural number k, there is a number k' such that if a graph G contains a k'-creature, then G contains a k-theta, k-pyramid, k-prism, k-ladder, k-ladder-theta, or k-ladder-prism as an induced subgraph.*

See Figure 6.2 for a depiction of the graphs in Claim 6.2.4, and see Section 6.6 for definitions. This statement appears as Lemma 6.6.10 in the formal proof. Claim 6.2.4 is best possible in the sense that each one of the k-theta, k-pyramid, k-prism, k-ladder, k-ladder-theta, or k-ladder-prism contains a k-creature, and that dropping any one of them from the list would make the conclusion of Claim 6.2.4 false. The contrapositive of the statement of Claim 6.2.4 implies that if a hereditary graph family $\mathcal{F}$ excludes the k-theta,

k-pyramid, k-prism, k-ladder, k-ladder-theta, and k-ladder-prism as induced subgraphs, then there exists a k' depending only on k such that $\mathcal{F}$ is k'-creature-free. Therefore, Claim 6.2.4 together with Theorem 6.1.2 and the observation that a $2k$-skinny-ladder induced minor either yields a k-creature or a k-contracted ladder as an induced subgraph implies Theorem 6.1.3.

So, how do we prove Claim 6.2.4? At a very high level it is just a sequence of structural lemmas, each on the form "if G contains a k-creature that additionally has some property X, then G also contains a k'-creature for some k' which is much smaller than k, but still tends to infinity with k, and the k'-creature has some stronger structural property Y". The next lemma now has as premise "if G contains a k creature that additionally has property Y", continuing the chain. Since we are looking for highly symmetric induced subgraphs in fairly general graphs it should come as no surprise that this sequence of arguments makes frequent use of Ramsey's Theorem.

Slightly more concretely, the proof of Claim 6.2.4 considers the two "sides" of the k-creature separately. Specifically, suppose that G contains a k'-creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$), then we focus in on just one half of the k'-creature, say the side the consists of A and $\{x_1, x_2, \ldots, x_{k'}\}$. Here we can find what amounts to a k-half-theta, k-half-prism, or k-half-ladder (imagine cutting a k-theta, k-prism, or k-ladder in half vertically, see Figure 6.9). Then on the other side that consists of B and $\{y_1, y_2, \ldots, y_{k'}\}$ we find a k-half-theta, k-half-prism, or k-half-ladder in the same way and we can merge them together to make either a k-theta, k-pyramid, k-prism, k-ladder, k-ladder-theta, or a k-ladder-prism.

Let us now see how to derive Theorem 6.1.4 from Theorem 6.1.3. We can see that for every natural number k there is a k' large enough such that if a graph contains a k'-ladder as an induced subgraph or a k'-skinny-ladder as an induced minor, then it contains a k-paw or k-claw as an induced subgraph. Hence, putting this together with Theorem

6.1.3 leads us to one half of Theorem 6.1.4, that if there is a natural number k such that a family of graphs $\mathcal{F}$ forbids k-theta, k-pyramid, k-prism, k-ladder-theta, k-ladder-prism, k-claw, and k-paw graphs, then $\mathcal{F}$ is strongly-quasi-tame.

Next, let's see how we prove the second part of the statement of Theorem 6.1.4. This states that if $\mathcal{F}$ is a family of graphs defined by a finite number of forbidden induced subgraphs, then if $\mathcal{F}$ contains k-theta, k-pyramid, k-prism, k-ladder-theta, k-ladder-prism, k-claw, or k-paw graphs for arbitrarily large k, then $\mathcal{F}$ is feral. If $\mathcal{F}$ contains k-theta, k-pyramid, k-prism, k-ladder-theta, or k-ladder-prism graphs for arbitrarily large k, then we take some k-theta, k-pyramid, k-prism, k-ladder-theta, or k-ladder-prism graph that is contained in $\mathcal{F}$ and show that if we contract the correct edges, then we can end up with a k-theta, k-pyramid, k-prism, k-ladder-theta, or k-ladder-prism graph that is still contained in $\mathcal{F}$ but now has $O(k)$ vertices. It follows that in this case $\mathcal{F}$ is feral.

It is critical here that our graph family is defined by a *finite* set of forbidden induced subgraphs. In general one cannot just contract an edge of a graph G that belongs to a family $\mathcal{F}$ and expect G to still belong to $\mathcal{F}$ after contraction. However, for a family that is defined by a finite set of forbidden induced subgraphs we can contract edges that are in the middle of sufficiently long paths consisting of vertices of degree 2. In our proofs we only contract such edges. Now, if $\mathcal{F}$ contains k-paw or k-claw graphs for arbitrarily large k, then we show that for large enough k we can essentially glue the claws or paws together on top of each other an create a graph with $O(k)$ vertices, exponentially many minimal separators, and avoid any of the forbidden subgraphs of $\mathcal{F}$ (see Figure 6.10 of Section 6.6 for a picture of this. Again it is crucial here that that our graph family is defined by a *finite* set of forbidden induced subgraphs). This concludes our sketch of the proof of Theorem 6.1.4.

6.2.3 Overview of the Proofs of Theorems 6.1.5, 6.1.6, and 6.1.7

Recall that Theorems 6.1.5 and 6.1.6 give *polynomial* upper bounds on the number of minimal separators for graphs that exclude k-creatures, k-skinny-ladders as induced minors, and additionally long cycles (in the case of Theorem 6.1.5) or large cliques (in the case of Theorem 6.1.6).

Given the tools developed on the way to proving Theorems 6.1.2 and 6.1.4, Theorems 6.1.5 and 6.1.6 follow almost for free. Recall the "dream strengthening" of Claim 6.2.2 from the proof sketch of Theorem 6.1.2; every minimal separator S in a graph that excludes a k-creature and a k-skinny-ladder as an induced minor is dominated by a set D of constant size k', disjoint from S. This dream strengthening is false in general, but it turns out to be true (and fairly easy to prove) in graphs that additionally exclude either all long cycles or all sufficiently large cliques. Now Theorems 6.1.5 and 6.1.6 follow directly from the argument in the failed proof attempt in Section 6.2.1 for Theorem 6.1.2 based on the dream claim. Theorem 6.1.7 is "extracted" from Theorem 6.1.6 in exactly the same way Theorem 6.1.4 is derived from Theorem 6.1.2.

6.3 Preliminaries

Let $\mathcal{F}$ be a family of graphs. We say that $\mathcal{F}$ is a *family of graphs defined by a finite number of forbidden induced subgraphs* if there exists a finite set of graphs $\mathcal{H}$ such that $G \in \mathcal{F}$ if and only if G if G is $\mathcal{H}$-free. We say that $\mathcal{H}$ is a set of forbidden subgraphs that define $\mathcal{F}$.

6.4 A k-Creature-Free Feral Graph Family

In this section we will show that the graph of Figure 6.4, which we will refer to as the *k-twisted-ladder*, is a counterexample to Conjecture 6.1.1. We begin the next paragraph by giving a few definitions, then in the following paragraph we will observe that the k-twisted-ladder has 2^k minimal separators, and finally Lemma 6.4.1 completes the counterexample by showing that the k-twisted-ladder does not contain a large k-creature.

We define a partition of the vertices as follows, let S denote the set of labeled vertices of the k-twisted ladder that have 1 as their superscript. If we remove S from the k-twisted-ladder we get two induced paths, one on the left side which we will refer to as L and one on the right side which we will refer to as R. We also define the i^{th} *block* of the k-twisted-ladder to be the set of vertices that contains the vertices of the subpath of L that has c_{i+1}^L and c_i^L as its endpoints, the vertices of the subpath of R that has c_{i+1}^R and c_i^R as its endpoints, and the vertices a_i^1 and b_i^1. So, the i^{th} block and the $(i+1)^{th}$ block overlap at the vertices c_{i+1}^R and c_{i+1}^L.

To see that the k-twisted-ladder has at least 2^k minimal separators we make the following set, X. For each i with $1 \leqslant i \leqslant k$ we choose $j \in \{1, 2\}$ and add a_i^j and b_i^j to X. X is then an x, y-minimal separator, and there are 2^k different choices we had when making X, so the k-twisted-ladder has at least 2^k minimal separators.

To complete the counterexample, we show in the following lemma that this structure does not have a large k-creature. To make the result as easy as possible to verify, we show no k-twisted-ladder has a 100-creature, although a significantly smaller upper bound exists.

Lemma 6.4.1. *k-twisted-ladders are 100-creature-free for all k.*

Proof:

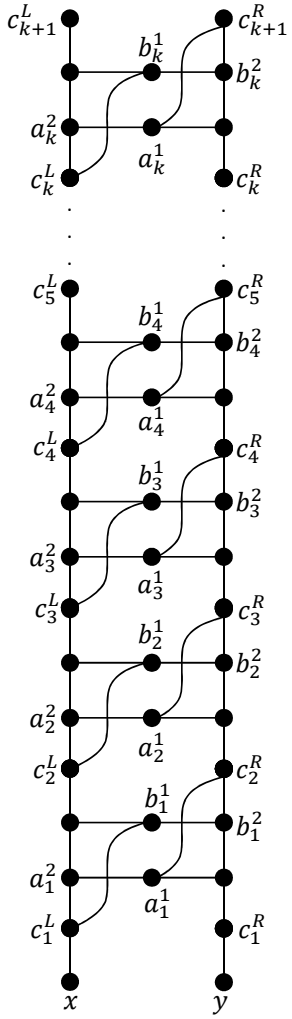


Figure 6.4: The k-twisted-ladder.

Let H be a k-twisted-ladder. Assume for a contradiction that H contains a 100-creature $(A, B, \{x_1, x_2, \ldots, x_{100}\}, \{y_1, y_2, \ldots, y_{100}\})$.

Let X_A and X_B denote the highest numbered block that A and B have a vertex in respectively, and let Y_A and Y_B denote the lowest numbered block that A and B have a vertex in respectively. Let $i = max(Y_A, Y_B) + 1$ and let $j = min(X_A, X_B) - 1$. Let r be an integer such that $i \leqslant r \leqslant j$ (if no such r exists, then the only blocks that can contain vertices from both A and B must be two adjacent blocks. Since each block only has 10 vertices, A has size at most 20 and since the max degree of the twisted-ladder is 3, A cannot dominate all 100 vertices of $\{x_1, x_2, \ldots, x_{100}\}$, a contradiction to the definition of k-creature). Then since A and B are connected and both contain vertices in blocks above and below block r we can see by inspection that A must contain one vertex from $\{c_r^L, c_r^R\}$ and one from $\{c_{r+1}^L, c_{r+1}^R\}$ and B must contain one vertex from $\{c_r^L, c_r^R\}$ and one from $\{c_{r+1}^L, c_{r+1}^R\}$. Furthermore, since A is anti-complete with B, we can again see from inspection that if $c_r^L \in A$ then we must have $c_{r+1}^L \in A$, $c_r^R \in B$, and $c_{r+1}^R \in B$ (the removal of the closed neighborhoods of c_r^L and c_{r+1}^R would separate blocks numbered greater than r from blocks numbered less than r, so both c_r^L and c_{r+1}^R cannot belong to A since B is connected and has vertices in blocks above and below r). Similarly if $c_r^R \in A$ then we must have $c_{r+1}^R \in A$, $c_r^L \in B$, and $c_{r+1}^L \in B$.

Therefore, without loss of generality we may assume that for all r with $i \leqslant r \leqslant j$ that $c_r^L \in A$ and $c_r^R \in B$. It then follows from this assumption and the fact that A is anti-complete with B that there are only two possibilities for the restriction of A and B to the r^{th} block. Either we have that both the restriction of A to the r^{th} block is the subpath of L with endpoints c_r^L and c_{r+1}^L and the restriction of B is the subpath of R with endpoints c_r^R and c_{r+1}^R or the restriction of A is the induced path made up of c_{r+1}^L along with b_r^1 and b_r^1's two neighbors in L and the restriction of B is the induced path made up of c_r^R along with a_r^1 and a_r^1's two neighbors in R. Note that in either case, we

may conclude by inspection of the twisted-ladder that every vertex in block r is either in A, B or $N(A) \cap N(B)$.

We now show that it is impossible for the vertices of $\{x_1, x_2, \ldots, x_{100}\}$ to be within distance two of both A and B, which would contradict the definition of a k-creature. By the definition of a k-creature, no vertex of $\{x_1, x_2, \ldots, x_{100}\}$ can belong to $N(A) \cap N(B)$. Hence, by the last sentence of the previous paragraph, no vertex of $\{x_1, x_2, \ldots, x_{100}\}$ belongs to blocks i through j. Since no vertex of $\{x_1, x_2, \ldots, x_{100}\}$ belongs to blocks i through j, $i - 1 = max(Y_A, Y_B)$ and $j + 1 = min(X_A, X_B)$, and the vertices of $\{x_1, x_2, \ldots, x_{100}\}$ must be within distance two of both A and B it follows that all vertices of $\{x_1, x_2, \ldots, x_{100}\}$ must be with distance two of blocks $i - 1$ and $j + 1$. But we can see by inspection of the twisted-ladder that there do not exists that many vertices within distance two of these two blocks. We can conclude that $(A, B, \{x_1, x_2, \ldots, x_{100}\}, \{y_1, y_2, \ldots, y_{100}\})$ cannot be a 100-creature. ■

6.5 *k*-Creature and *k*-Skinny-Ladder Induced Minor Free Graphs

In this section we will provide all the lemmas needed for a proof of Theorem 6.1.2 and conclude this section with a proof of Theorem 6.1.2. We begin this section by stating some well known results which we will need later on. The three key ideas of this section are Lemma 6.5.6 which shows that the neighborhood of a vertex v of a k-creature-free graph G can intersect the minimal separators of G that do not contain v in at most n^k different ways, Lemma 6.5.15 which shows that all minimal separators of graphs that are k-creature-free and do not contain a k-skinny-ladder as an induced minor can be dominated by a constant number vertices, and Lemma 6.5.16 which uses a branching

algorithm to list all minimal separators of its input graph assuming the input graph satisfies certain properties and proves a bound on the number of minimal separators produced by this algorithm. An easy proof combining Lemma 6.5.6, and Lemma 6.5.16 is then used to establish Theorem 6.1.2. Most of the work of this section goes into proving lemmas needed for the proof of Lemma 6.5.15, in particular, Lemmas 6.5.8 through 6.5.14 build up to a proof of Lemma 6.5.15. In this section and the rest of the paper when we refer to k-creatures and k-skinny-ladders we will assume $k > 1$.

Lemma 6.5.1 (**Ramsey's Theorem**). *[125]*

For every pair of positive integer k and ℓ there is a least positive integer $R(k, \ell)$ such that every graph with at least $R(k, \ell)$ vertices contains a clique of size k or an independent set of size ℓ.

Throughout this paper we will us the notation $R(k, \ell)$ to denote the least positive integer such that every graph with at least $R(k, \ell)$ vertices contains a clique of size k or an independent set of size ℓ.

Lemma 6.5.2 (**Erdös-Szekeres Theorem**). *[53]*

For every pair on positive integers r and s, any sequence of distinct real numbers of length at least (r-1)(s-1) + 1 contains a monotone increasing subsequence of length r or a monotone decreasing subsequence of length s.

Definition 6.5.3 (**VC-Dimension**). Let $\mathcal{F} = \{S_1, S_2, \ldots\}$ be a finite family of finite sets and let H be a set. $\mathcal{F}$ is said to *shatter* H if for every subset $H' \subseteq H$ there is a $S_i \in \mathcal{F}$ such that $H' = S_i \cap H$. The *VC-dimension* of $\mathcal{F}$ is the cardinality of the largest set that it shatters.

Lemma 6.5.4 (**Sauer-Shelah Lemma**). *[126] Let $\mathcal{F}$ be a finite family of finite sets such that the VC-dimension of $\mathcal{F}$ is $k > 1$, and let $n = |\bigcup_{S_i \in \mathcal{F}} S_i|$, so n is the number*

of distinct elements contained in the sets of $\mathcal{F}$. Then the number of sets of $\mathcal{F}$ is at most $\Sigma_{i=0}^{k}\binom{n}{i} \leqslant n^k$.

Lemma 6.5.5. *Let S be a u, v-minimal separator and a u, w-separator and let $S' \subset S$ be a u, w-minimal separator. Then $N(w) \cap S' = N(w) \cap S$.*

Proof: Let S be a u, v-minimal separator and a u, w-separator, let $S' \subset S$ be a u, w-minimal separator, and let C_u and C'_u be the connected components that u lies in in $G - S$ and $G - S'$ respectively. Note that $C_u \subseteq C'_u$. Clearly $N(w) \cap S' \subseteq N(w) \cap S$. Now let $y \in N(w) \cap S$. Then there is a path from y to u such that all internal vertices of this path are contained in C_u. This same path has its internal vertices in C'_u, so if $y \notin S'$ then S' does not separate u from w. The result follows. ■

Lemma 6.5.6. *Let G be a k-creature-free graph and let $\mathcal{S}$ be a set of minimal separators of G. Then for every $v \in G$, if $S^v = \{N(v) \cap S|\ S \in \mathcal{S}$ and $v \notin S\}$ then $|S^v| \leqslant |V(G)|^k$.*

Proof: Let G be a k-creature-free graph with n vertices, let $\mathcal{S}$ be a set of minimal separators of G, and fix two non-adjacent vertices u, v. Let $S^{v,u} = \{N(v) \cap S | S \in \mathcal{S}$ and S is a u, v-minimal separator of $G\}$. We first show that $|S^{v,u}| \leqslant n^{k-1}$. Assume for a contradiction that $|S^{v,u}| > n^{k-1}$, then by the Sauer-Shelah Lemma there is a subset of size k of $N(v)$ that is shattered by $S^{v,u}$. It follows that there are vertices $V = \{v_1, \ldots, v_k\}$ in $N(v)$ such that each v_i belongs to a private u, v-minimal separator S_i, i.e., $V \cap S_i = \{v_i\}$. Now, let C_i be the component that u belongs to in $G - S_i$, C_i dominates S_i since S_i is a u, v-minimal separator. Let v'_i be a neighbor of v_i in C_i with minimum distance to u (note that v_i is not a neighbor of u or else there would be a u, v path in $G - S_j$ for $j \neq i$, hence $v'_i \neq u$), and let P_i be a shortest path from v'_i to u in C_i. Let $P = \bigcup_{i=1} (V(P_i) - \{v'_i\})$.

We claim that $(v,\ P,\ \{v_1, v_2, \ldots, v_k\},\ \{v'_1, v'_2, \ldots, v'_k\})$ is a k-creature (see Figure 6.3 to for a visual description of this step of the proof). To see this note that (1) $G[P]$ is

connected since it is a set of paths which all contain u. (2) v is anti-complete with each P_i since v and P_i are both contained in two different S_i-full components hence v is anti-complete with P and $\{v'_1, v'_2, \ldots, v'_k\}$. To see that P is anti-complete with $\{v_1, v_2, \ldots, v_k\}$ note that each $P_i - v'_i$ is anti-complete with v_i since v'_i was chosen to be a neighbor of v_i in C_i that is as close as possible to u and P_i is a shortest path from v'_i to u in C_i. Furthermore, if P_i was not anti-complete with v_j for $i \neq j$ then S_i would not be a u, v-minimal separator since by assumption $v_j \notin S_i$ and no vertex of P_i is in S_i (since $V(P_i) \subseteq C_i$), hence there would be a path from v to u in $G - S_i$. It then follows that P is anti-complete with $\{v_1, v_2, \ldots, v_k\}$. (3) P dominates $\{v'_1, v'_2, \ldots, v'_k\}$ follows from how we defined P and $\{v_1, v_2, \ldots, v_k\} \subseteq N(v)$. (4) v_i is anti-complete with P_j for $i \neq j$ was showing when establishing (2), hence v_i is a neighbor of v'_j only if $i = j$. So, (v, P, $\{v_1, v_2, \ldots, v_k\}$, $\{v'_1, v'_2, \ldots, v'_k\}$) is a k-creature, a contradiction to G being k-creature-free. It follows that $|S^{v,u}| \leqslant n^{k-1}$.

Now, let $S \in \mathcal{S}$ such that $v \notin S$ and assume that S is an x, y-separator. S must be either a v, x-separator or a v, y-separator, let us assume without loss of generality that S is a v, x-separator, so there is an $S' \subseteq S$ that is a v, x-minimal separator. By Lemma 6.5.5 we have that $N(v) \cap S' = N(v) \cap S$, hence $N(v) \cap S \in S^{v,x}$, where $S^{v,x} = \{N(v) \cap S'' | S'' \in \mathcal{S}$ and S'' is a v, x-minimal separator of $G\}$. It follows that if $S^v = \{N(v) \cap S'' |\ S'' \in \mathcal{S}$ and $v \notin S''\}$ then $S^v = \bigcup_{x \in V(G)} S^{v,x}$. Then by the conclusion of the previous paragraph, it follows that $|S^v| \leqslant n^k$. ■

The following corollary will be needed in Sections 6.7 and 6.8.

Corollary 6.5.7. *If G is a k-creature-free graph and every minimal separator, S, of G can be dominated by k' vertices of G not in S, then G has at most $|V(G)|^{kk'+k'}$ minimal separators.*

Proof: Assume G is a k-creature-free graph and every minimal separator, S, of G

can be dominated by k' vertices of G not in S. For every $v \in G$ let $S^v = \{N(v) \cap S | v \notin S$ and S is a minimal separator of $G\}$. By Lemma 6.5.6 it holds that $|S^v| \leqslant |V(G)|^k$. Let $X = \bigcup_{v \in G} S^v$. Then $|X| = |V(G)|^{k+1}$ and the assumption that all minimal separators, S, of G can be dominated by k' vertices in G not in S implies that S is the union of at most k' sets in X. It follows there are at most $|V(G)|^{kk'+k'}$ minimal separators in G. ■

We remark that it is possible to generalize Lemma 6.5.6 and Corollary 6.5.7 to the r^{th} neighborhood of a vertex for any fixed positive integer r while still maintaining polynomial bounds by using the fact the family of k-creature-free graphs are closed under contracting edges.

The following lemmas will be building towards a proof of Lemma 6.5.15, that all minimal separators of a graph that is k-creature-free and has no k-skinny-ladder as an induced minor can be dominated by few vertices. We begin with a proof that minimal separators can be dominated by a few induced paths in k-creature-free graphs (the paths may have edges between them).

Lemma 6.5.8. *Let G be a graph that is k-creature-free, let S be a minimal separator of G, and let A be an S-full component of $G - S$. Then S is dominated by a set of less than k induced paths of A.*

Proof: Let G, S, and A be as in the statement of this lemma, and let A' be a minimally connected induced subgraph of A such that S is dominated by A'. Let T be a breadth first search tree of A' rooted at some vertex $v \in A'$, and let $L = \{\ell_1, \ell_2, \ldots, \ell_c\}$ be the set of leaves of T. Since A' is minimal each leaf, $\ell_i \in L$, must have a neighbor $s_i \in S$ such that no other vertex of A' is a neighbor of s_i, else $A' - s_i$ would still be connected and dominate S. Then if K is another S-full component different from A we claim that the tuple $(V(A') - L, V(K), L = \{\ell_1, \ell_2, \ldots, \ell_c\}, \{s_1, s_2, \ldots, s_c\})$ forms a c-creature. To see this, note that (1) $G[V(A') - L]$ is still connected since L is a set of leaves of T, hence

$T - L$ is a spanning tree of $G[V(A') - L]$. Additionally, K is connected by definition. (2) $V(A') - L$ is anti-complete with $\{s_1, s_2, \ldots, s_c\}$ since ℓ_i is the only vertex of A' that is neighbors of s_i, $V(A')$ is anti-complete with $V(K)$ since they are contained in two different S-full components hence $V(A' - L)$ and $V(K')$ are anti-complete and $V(K')$ and L are anti-complete. (3) That $V(A') - L$ dominates L is straight forward, and K dominates $\{s_1, s_2, \ldots, s_c\}$ since K is an S-full component. (4) By how we chose the s_i's we have have ℓ_i is a neighbor of s_j if and only if $i = j$.

It follows that if G is k-creature-free, then T has at most $k - 1$ leaves. Since T is a breadth first search tree of A', a root to leaf path in T is also an induced path in A', therefore A' is the union of at most $k - 1$ induced paths and the result follows. ■

A key step to proving Lemma 6.5.15 is to show that if a k-creature-free graph G has a minimal separator, S, that cannot be dominated by $f(k)$ vertices, then using Lemma 6.5.8 we can find an induced path in an S-full, call the induced path P_L, and another induced paths from another S-full component, call this induced path P_R, such that we can find a large independent set, I, of S where every vertex in I has at least one neighbor in P_L and one neighbor in P_R and furthermore, no pair of vertices in I share a neighbor in either P_L or P_R. This is proven in Lemma 6.5.11.

The paths P_L and P_R are paths obtained by selecting induced paths such that the set $S_1 = N(P_L) \cap S \cap N(P_R)$ takes at least $f(k)/k^2$ vertices to dominate, such paths must exits by Lemma 6.5.8. The idea of the proof of Lemma 6.5.11 is to find a special vertex $v_1 \in S_1$, such that we can find a small set of vertices X_1 such that no vertex in $S_1 - N[X_1]$ shares a neighbor with v_1 in P_L and we find a small set of vertices Y_1 such that no vertex in $S_1 - N[Y_1]$ shares a neighbor with v_1 in P_R. We then add v_1 to I and set $S_2 = S_1 - (N[X_1] \cup N[Y_1] \cup \{v_1\})$. Then we find a special vertex $v_2 \in S_2$ and repeat. If the X_i's and Y_i's are small in size and S_1 cannot be dominated by few vertices then we can create a large set I in this way so that no vertices in P_L and P_R have more than

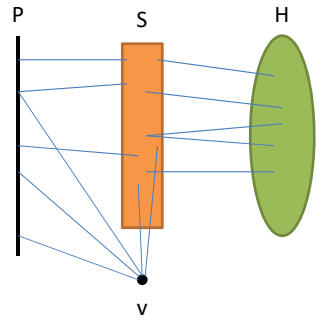


Figure 6.5: P, S, and H and the vertex v of Lemma 6.5.9

one neighbor in I, so we get our desired P_L, P_R, and I.

The following lemma shows us how to locate X_i given the vertex v_i. The specially chosen vertex v_i will have the property that there exists a connected graph H such that all vertices of S_i have at least one neighbor in H except for v_i, and H is anti-complete with P_L (see Figure 6.5, where $P_L = P, S_i = S$, and $v_i = v$). In this situation, Lemma 6.5.9 shows how to obtain the desired set X_i, such that no vertex of $S_i - N[X_i]$ shares a neighbor with v_i in the path P_L.

Lemma 6.5.9. *Let G be a k-creature-free graph and let (S, H, P, v) be a tuple of disjoint subsets of $V(G)$ with the following properties (see Figure 6.5): $G[H]$ is connected, $G[P]$ is an induced path, H is anti-complete with P and v, and H dominates S. Then there is a set, X, of size at most k such that $N(S - N[X]) \cap N(v) \cap P = \emptyset$ and no vertex of $S - N[X]$ is a neighbor of v.*

Proof:

Let G, S, H, P, and v be as in the statement of this lemma. Let $P' = P \cap N(v)$, let $S' = (S \cap N(P')) - N(v)$, and let $P'' = \{p_1, p_2, \ldots, p_c\}$ be a minimal subset of P' that dominates S'. Since P'' is a minimal dominating set each element of P'' has a private neighbor in S', in other words for each $p_i \in P''$ there is an $s_i \in S'$ such that p_i is a neighbor of s_i and p_j is not a neighbor of s_i if $i \neq j$. We claim that the tuple

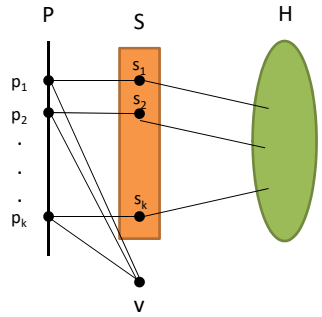


Figure 6.6: The k-creature formed in Lemma 6.5.9

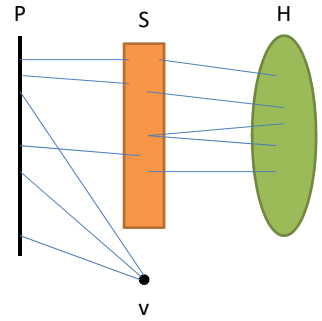


Figure 6.7: P, S, and H and the vertex v of Lemma 6.5.10

$(\{v\}, H, P'' = \{p_1, p_2, \ldots, p_c\}, \{s_1, s_2, \ldots, s_c\})$ forms a c-creature (see figure 6.6). To see this note that (1) by assumption $G[H]$ is connected. (2) By assumption v and H are anti-complete, v and $\{s_1, s_2, \ldots, s_c\} \subseteq S'$ are anti-complete by how S' was defined, and by assumption H and $\{p_1, p_2, \ldots, p_c\} \subseteq P$ are anti-complete. (3) v dominates $\{p_1, p_2, \ldots, p_c\} \subseteq P'$ by how P' is defined and H dominates $\{s_1, s_2, \ldots, s_c\}$ since H dominates S by assumption. (4) By how the s_i's were chosen, s_i is a neighbor of p_j if and only if $i = j$.

Hence, since G is k-creature free we may assume that $c \leqslant k - 1$. By how P'' was chosen then we have that $N(S' - N[P'']) \cap N(v) \cup P = \emptyset$. Then setting $X = P'' \cup \{v\}$ is a set of size at most k that satisfies the conclusion of this lemma.

■

While Lemma 6.5.9 works for finding X_i such that no vertex of $S_i - N[X_i]$ has a

common neighbor with v_i in P_L, it will not work to obtain a corresponding Y_i for P_R, the problem will lie in finding a suitable H with respect to P_R, S_i, and v_i (v_i is chosen specifically so we can find a suitable H with respect to P_L, S_i, and v_i, the path P_R is not taken into consideration in this selection). This issue is taken care of by Lemma 6.5.10, which shows that there is a set of at most $k-1$ connected components, $C_1, C_2, \ldots, C_{k-1}$ of $P_L - N(v_i)$ that collectively dominated S_i. Then for each C_j clearly all vertices of $S_i \cap N(C_j)$ have a neighbor in C_j, and P_R and v_i are anti-complete with C_j, so we can apply Lemma 6.5.9 to ($S_i \cap N(C_j)$, C_i, P_R, v_i) to get a set $Y_{i,j}$ such that no vertex of $(S_i \cap N(C_j)) - N[Y_{i,j}]$ shares a neighbor with v_i in P_R. Since collectively all the C_j's dominate S_i, if we take $Y_i = \bigcup_j Y_{i,j}$ then no vertex of $S_i - N[Y_i]$ shares a neighbor with v_i in P_R.

Lemma 6.5.10. *Let G be a k-creature-free graph and let (S, H, P, v) be a tuple of disjoint subsets of $V(G)$ with the following properties (see Figure 6.7): $G[H]$ is connected, $G[P]$ is an induced path, H is anti-complete with P and v, v is anti-complete with S, S is dominated by H and S is dominated by P, and $N(S) \cap N(v) \cap P = \emptyset$. Then there is a set of at most $k-1$ connected components of $G[P] - N(v)$ such that every vertex of S has a neighbor in at least one of these connected components.*

Proof: Let G, S, H, P, and v be as in the statement of this lemma. Assume for a contradiction that there does not exists a set of at most $k-1$ connected components of $G[P] - N(v)$ such that every vertex of S has a neighbor in at least one of these connected components. It follows then there is a set of k connected components of $G[P] - N(v)$, say $C_1, C_2, \ldots, C_k$, such that there exists $s_1, s_2, \ldots, s_k$ in S where $N(s_i) \cap V(C_j) \neq \emptyset$ if and only if $i = j$. Since $G[P]$ is connected, for every C_i there exists a vertex $c_i \in N(v) \cap P$ such that $c_i \in N(C_i)$ (the c_i's may not be unique even though the C_i's are). Now, for each s_i, let s_i' be the vertex in C_i that s_i is a neighbor of such that there exists an induced

path P_i from s'_i to c_i with internal vertices in C_i such that s'_i is the only neighbor of s_i on the path P_i (recall by assumption that $N(S) \cap N(v) \cap V(P) = \emptyset$ so s_i cannot be a neighbor of c_i).

We claim the tuple $(\{v\} \cup \bigcup V(P_i - s'_i), H, \{s'_1, s'_2 \ldots, s'_k\}, \{s_1, s_2 \ldots, s_k\})$ is a k-creature, contradicting the assumption G is k-creature-free. To see this note that (1) H is connected by assumption and for all $i, 1 \leqslant i \leqslant k$, c_i belongs to $P_i - s'_i$ which is a neighbor of v, so $\{v\} \cup \bigcup V(P_i - s'_i)$ induces a connected graph, (2) $\{v\} \cup \bigcup V(P_i - s'_i)$ is anti-complete with H by definition, $\{v\} \cup \bigcup V(P_i - s'_i)$ is anti-complete with $\{s_1, s_2 \ldots, s_k\}$ by how the paths P_i were chosen and by the assumption that v is anti-complete with S, and H is anti-complete with $\{s'_1, s'_2 \ldots, s'_k\} \subseteq V(P)$ by assumption. (3) $\{v\} \cup \bigcup V(P_i - s'_i)$ dominates $\{s'_1, s'_2 \ldots, s'_k\}$ since $s'_i \in P_i$ and H dominates $\{s_1, s_2 \ldots, s_k\} \subseteq S$ by assumption. (4) s_i is a neighbor of s'_i by how s'_i was chosen and for $i \neq j$ s_i is not a neighbor of $s'_j \in C_j$ because s_i has no neighbor in C_j by how the C_j's where chosen. ■

We are now ready to prove Lemma 6.5.11.

Lemma 6.5.11. *Let S be a minimal separator of a k-creature-free graph G such that S cannot be dominated by $2k^4x$ vertices. Then there exists there exists an independent subset I of S of size x such that there exists two induced paths, P_L and P_R, in two different components of $G - S$ that dominate the vertices of I and no vertex of P_L nor P_R has more than one neighbor in I (see Figure 6.8).*

Proof: Assume that G is a k-creature-free graph, and let S be a minimal separator of G that cannot be dominated by $2k^4x$ vertices of G, and let L and R be two different S'-full components of G. It follows from Lemma 6.5.8 that there is a set of less than k induced paths in L that together dominated S and there is a set of less than k induced paths in R that together dominate S. So, since there are less than k^2 pairs of these induced paths with one from R and one from L, it follows there exists two induced paths

P_L in L and P_R in R such that $(N(P_L) \cap S \cap N(P_R))$ cannot be dominated by $2k^2x$ vertices of G. Let $S' = (N(P_L) \cap S \cap (N(P_R))$. Number the vertices of P_R sequentially 1 through $|V(P_R)|$.

Assume that we have an independent set of vertices I_{i-1} of size $i-1$, $1 \leqslant i \leqslant x$, and a vertex set Z_{i-1} of size at most $2k^2(i-1)$, with the properties that no vertex of $S'-N[Z_{i-1}]$ is a neighbor of a vertex in I_{i-1}, and for all $v \in I_{i-1}$ if $w \in I_{i-1} \cup (S'-N[Z_{i-1}])$, $v \neq w$, then $N(v) \cap N(w) \cap (V(P_L) \cup V(P_R)) = \emptyset$, that is for all $v \in I_{i-1}$, no $w \in I_{i-1} \cup (S' - N[Z_{i-1}])$, $w \neq v$, shares a neighbor with v in P_L nor P_R. We will show how to produce a set I_i of size i and Z_i of size at most $2k^2i$ with the same properties, assuming $i \leqslant x$. Note that for the base case the empty set satisfies the conditions required of S_0 and Z_0.

Let $S'' = S' - N[Z_{i-1}]$, since $i \leqslant x$ and since S' cannot be dominated by $4k^2x$ vertices S'' must be non-empty. Label the vertices of S'' according to the lowest numbered neighbor it has in P_R. Let v be a highest labeled vertex in S''. Let w be the lowest numbered neighbor v has in P_R and assume w (and therefore v) is labeled with the number p. Let H denote the subpath of P_R that is made up of the vertices labeled 1 through $p-1$, hence H is anti-complete with v and H dominates all vertices of $S'' - N(w)$ (since v is a highest labeled vertex of S'' and has label p, all vertices of $S'' - N(w)$ must have a neighbor that has a label lower than p and hence in H).

We now wish to apply Lemma 6.5.9 using $(S'' - N(w), V(H), V(P_L), v)$. To see that this tuple satisfies the assumption of Lemma 6.5.9 note that H and P_L are paths and therefore connected, H is anti-complete with P_L since they are contained in two different S-full components, that H is anti-complete with v was noted at the end of the previous paragraph, and that H dominates $S'' - N(w)$ was also noted at the end of the previous paragraph. Hence, we may apply Lemma 6.5.9 using $(S'' - N(w), V(H), V(P_L), v)$ to get a set X of size at most k such that $N((S'' - N[w]) - N[X]) \cap N(v) \cap V(P_L) = \emptyset$ and no vertex of $(S'' - N[w]) - N[X]$ is a neighbor of v. This implies if we set $X' = X \cup \{w\}$

then $N(S'' - N[X']) \cap N(v) \cap V(P_L) = \emptyset$, no vertex of $S'' - N[X']$ is a neighbor of v, and $v \notin S'' - N[X']$.

We now wish to find a set Y of size less than k^2 such that no vertex of $S'' - (N[X'] \cup N[Y])$ shares a neighbor with v in either P_L or P_R. For ease of notation, set $S''' = S'' - N[X']$. It is tempting to try to use Lemma 6.5.9 on something like (S''', P_L, P_R, v), but this lemma requires that v not have any neighbors in P_L. Instead, we first use Lemma 6.5.10 on $(S''', V(H), V(P_L), v)$. To see that we can apply this lemma, note that both H and P_L are paths, H is anti-complete with P_L since they are contained in two different S-full components, that H is anti-complete with v was noted at in the last sentence two paragraphs ago as was the fact that H dominates $S''' \subseteq S'' - N(w)$ (recall $w \in X$), that v is anti-complete with $S''' = S'' - N[X']$ was noted at the end of the previous paragraph, and P_L dominates S''' because P_L dominates S' by assumption and $S''' \subseteq S'$.

So, we use Lemma 6.5.10 on $(S''', V(H), V(P_L), v)$ to get connected components $C_1, C_2, \ldots, C_c$, $c < k$, of $P_L - N(v)$ (hence v has no neighbors in each C_i) such that all vertices of S''' have a neighbor in at least one C_i. Now, for each C_i we apply Lemma 6.5.9 on $(S''' \cap N(C_i),\ C_i,\ P_R,\ v)$ to get a set Y_i of size at most k such that $N((S''' \cap V(C_i)) - N[Y_i]) \cap N(v) \cap V(P_R) = \emptyset$.

Since the C_i's dominate S''', it follows that if we set $Y = \bigcup Y_i$ then $N(S''' - N[Y]) \cap N(v) \cap V(P_R) = \emptyset$. Since $S''' = S'' - N[X']$ and $N(S'' - N[X']) \cap N(v) \cap V(P_L) = \emptyset$ it follows that if we set $Z_i = Z_{i-1} \cup X' \cup Y$ then no vertex $S' - N[Z_i]$ shares a neighbor with v in P_L nor P_R. We may set $I_i = I_{i-1} \cup \{v\}$ and $Z_i = Z_{i-1} \cup X' \cup Y$. Since each Y_i has at most k vertices we have that $|Y| \leqslant k^2$ and $|X'| \leqslant k$ so $|Z_i| \leqslant |Z_{i-1}| + k + k^2 \leqslant 2k^2 i$ as required.

The statement of the lemma now follows from the fact that S cannot be dominated by $2k^2x$ vertices so this process can continue until we attain the set I_x, which has the property that for any pair $v, w \in I_x$ $v \neq w$ it holds that $N(v) \cap N(w) \cap (V(P_L) \cup V(P_R)) = \emptyset$. So

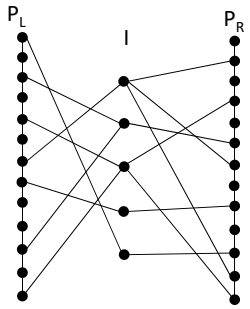


Figure 6.8: An example of a graph produced by Lemma 6.5.11

I_x which is the desired set, along with the paths P_L and P_R.

■

We now present three straightforward lemmas that will help us in the proof of Lemma 6.5.15. This first lemma is essentially a quick application of the Erdös-Szekeres Theorem. First we must give the following definition.

We call a graph G a *k-almost-skinny-ladder* if the following conditions hold:

- $V(G) = L \cup S \cup R$ with L, S, and R mutually disjoint and $|S| = k$.
- $G[L]$ and $G[R]$ form induced paths of G and L is anti-complete with R.
- Each $s \in S$ has at least one neighbor in L and at least one neighbor in R.
- For all pairs $x, y \in S$, if a, b are neighbors of x in L, then y has no neighbors on the subpath of $G[L]$ that has a and b as its endpoints. Similarly, if a, b are neighbors of x in R, then y has no neighbors on the subpath of $G[R]$ that has a and b as its endpoints.

The last condition of almost-skinny-ladders requires that no vertex of L or R has more than one neighbor in S. It is a straight forward application of the Erdös-Szekeres Theorem to show that a k-almost-skinny-ladder contains a k-skinny-ladder as an induced minor, as the next lemma shows.

Lemma 6.5.12. *Let G be a graph that contains a k^2-almost-skinny-ladder as an induced subgraph. Then G contains a k-skinny-ladder as an induced minor.*

Proof: Let G be a graph that has a k^2-almost-skinny-ladder, H, as an induced subgraph. $V(H) = L \cup S \cup R$ where L, S, R each have the same meaning as in the definition of an almost-skinny-ladder. Number the vertices of L sequentially 1 through $|V(L)|$, and similarly, number the vertices of R sequentially 1 through $|V(R)|$.

Next we label each vertex in S with a number 1 through $|S|$ such that for all $s_i, s_j \in S$ $i > j$ if and only if all of s_i's neighbors in L have a higher number than all of s_j's neighbors in L (by the definition of an almost-skinny-ladder such a numbering exists). Let $n(s_i)$ be the number of the highest numbered neighbor s_i has is R. We now apply the Erdös-Szekeres Theorem to the sequence $n(s_1), n(s_2) \ldots, n(s_{k^2})$ to get an increasing or decreasing subsequence of length at least k and set S^* to be the subset of S that corresponds to the subsequence obtained from our application of the Erdös-Szekeres Theorem. If the Erdös-Szekeres Theorem returned a decreasing subsequence then reverse the numbering of R, else leave it unchanged. Then for every $s_i, s_j \in S^*$, if $i > j$ then all of s_i's neighbors in L have a higher number than all of s_j's neighbors in L and all of s_i's neighbors in R have a higher number than all of s_j's neighbors in R. We can now apply the obvious edge contractions to L and R to form a k-skinny-ladder. ■

Lemma 6.5.13. *Let G be a graph, let $a, b \in G$ be two non adjacent vertices of G, and let $P_1, P_2, \ldots, P_k$ be k mutually anti-complete induced paths. Assume for all P_i that both a and b have a neighbor in P_i and no vertex of P_i is a neighbor of both a and b. Then G contains a k-creature.*

Proof: Let G, a, b, $P_1, P_2, \ldots, P_k$ be as in the statement of the lemma. For each P_i we can then, by assumption, find a subpath of P_i, call it P_i^*, such that P_i^* has endpoints a_i, b_i where a_i is a neighbor of a, b_i is a neighbor of b, no internal vertex is a neighbor of

a or b. Since a and b do not share any neighbors in P_i we can see that P_i^* has at least 2 vertices and since the P_i^*'s are anti-complete by assumption, together the P_i^*'s along with a and b make a k-creature. ■

Lemma 6.5.14. *Let G be a directed graph with maximum out-degree or maximum in-degree at most c, $c > 0$. Then G has an independent set (no vertex is an in-neighbor or out-neighbor of any other vertex in this set) of size at least $\frac{|V(G)|}{2c+1}$. Furthermore, if $|V(G)| \geqslant 2t$ and the maximum out-degree or maximum in-degree of G is at most $\frac{1}{4t}|V(G)|$, then G has an independent set of size at least t.*

Proof: Let G be a directed graph. We will prove the statements for bounded maximum out-degree (for maximum in-degree the proof is nearly identical). If the maximum out-degree of G is c, $c > 0$, then as long as G has at least one vertex, there must exists a vertex $v \in G$ with in-degree at most c. If we let G' be the subgraph induced by all vertices of $G - v$ that do not have v as an in-neighbor or an out-neighbor, then the size of G' is at least $|V(G)| - 2c - 1$, and G' has maximum out-degree c. It follows by an inductive argument that we can find an independent set of size at least $\frac{|V(G)|}{2c+1}$.

To prove the furthermore statement, assume the maximum out-degree of G is at most $\frac{1}{4t}|V(G)|$, $t > 0$, and $|V(G)| \geqslant 2t$, so we have that $\frac{|V(G)|}{2t} + 1 \leqslant \frac{|V(G)|}{t}$. From the first paragraph we have that G contains an independent set of size at least $\frac{|V(G)|}{\frac{2|V(G)|}{4t}+1} = \frac{|V(G)|}{\frac{|V(G)|}{2t}+1} \geqslant \frac{|V(G)|}{\frac{|V(G)|}{t}} = t$. ■

We are now in a position to prove Lemma 6.5.15, which states that if our graph G is k-creature-free and has a minimal separator that cannot be dominated by few vertices, then G contains a k-skinny-ladder as an induced minor. How do we show this? By Lemma 6.5.11 we know that if our graph G is k-creature-free and has a minimal separator that cannot be dominated by few vertices, then we can find a large independent set I and paths P_L and P_R such that all vertices of I have neighbors in P_L and P_R and no vertex in

P_L and P_R has over one neighbor in I. This is the starting point for the proof of Lemma 6.5.15

Heuristically, the idea of the proof is that we set $I_1 = I, L_1 = P_L$, and $R_1 = P_R$ and we can either find a large subset $A \subset I_1$ that satisfies certain properties, in which case we use Lemmas 6.5.14 and 6.5.12 to show that L_1, R_1 and A contain a k-skinny-ladder as an induced minor, or there is a large subset $I_2 \subset I_1$ such there is a way to divide either L_1 or R_2 into two "halves" such that both halves dominate I_2, for simplicity let us say we can do this with L_1. We then set P_1 to be one half of L_1, we set L_2 to be the other half, and we set $R_2 = R_1$. We now repeat this process with L_2, R_2 and I_2 and so on. In the end we either end up with our desired k-skinny-ladder, or we end up with k anti-complete induced paths all of which dominate some independent set I_k, and no vertices in I_k have a common neighbor in any of these anti-complete paths. But Lemma 6.5.13 shows this implies the existence of a k-creature in G which is a contradiction, so we must be in the case where this process produces a k-skinny-ladder as an induced minor.

Lemma 6.5.15. *Let S be a minimal separator of a k-creature-free graph G such that S cannot be dominated by $2k^4[(8k^2)^{k+1}]$ vertices. Then G contains a k-skinny-ladder as an induced minor.*

Proof: Assume that G is k-creature-free and S is a minimal separator of G such that S cannot be dominated by $2k^4[(8k^2)^{k+1}]$ vertices. It follows from Lemma 6.5.11 that there is an independent set $I \subseteq S$ of $(8k^2)^{k+1}$ vertices and two induced paths P_L and P_R that dominate I, P_L anti-complete with P_R, and every vertex in $v \in V(P_L) \cup V(P_R)$ has at most one neighbor in I.

Number the vertices of P_L sequentially 1 through $|V(P_L)|$ and number the vertices of P_R sequentially 1 through $|V(P_R)|$. For a vertex x in P_L or P_R we will use the notation $n(x)$ to denote the number it has been given in P_L or P_R. For every $v \in I$ let $\ell(v)$ and

$r(v)$ denote the highest numbered vertex v is a neighbor of in P_L and P_R respectively. We now set $L_1 = P_L, R_1 = P_R$, and $I_1 = I$. We will consider the following process to produce a k^2-almost-skinny-ladder. We will show this process cannot go past k iterations do to the fact that G is k-creature-free.

We will ensure the following properties are met at the end of the i^{th} step (and we will assume that these properties hold for the previous steps). At the i^{th} step we produce L_{i+1} which is a subpath of L_i, R_{i+1} which is a subpath of R_i, $I_{i+1} \subseteq I_i$, $|I_{i+1}| \geqslant (8k^2)^{k-i+1}$, and for every $v \in I_{i+1}$ it holds that $\ell(v) \in L_{i+1}$ and $r(v) \in R_{i+1}$. We will also produce P_i which will be either a subpath of L_i or R_i and is anti-complete with L_{i+1}, anti-complete with R_{i+1}, and anti-complete with P_j for $j < i$, and P_i will dominate I_j if $i < j$ (note by Lemma 6.5.13 that if we have k such paths then G would have a k-creature, so this process cannot go past k steps).

At the i^{th} step, $1 \leqslant i \leqslant k$, we do as follows. Create an auxiliary directed graph, A_i, whose vertex set is I_i and there is an edge from $v \in I_i$ to $w \in I_i$ if at least one of the following two cases hold

1. $n(\ell(v)) > n(\ell(w))$ and v has a neighbor x in L_i such that $n(x) < n(\ell(w))$
2. $n(r(v)) > n(r(w))$ and v has a neighbor x in R_i such that $n(x) < n(r(w))$

If the maximum in-degree of A_i is at most $\frac{1}{4k^2}|I_i|$ then we stop. Since by assumption $|I_i| \geqslant (8k^2)^{k-i+2}$ this gives an independent set of size at least k^2 by Lemma 6.5.14. If there is an $s_t \in I_i$ with in-degree over $\frac{1}{4k^2}|I_i|$ then either case 1 or case 2 is satisfied for at least half of s_t's in-neighbors. This means that for at least $\frac{1}{8k^2}$ fraction of the vertices of I_i, call this subset of vertices I_{i+1}, all vertices $s \in I_{i+1}$ must satisfy case 1 with s playing the role of v and s_t playing the role of w, or all vertices $s \in I_{i+1}$ must satisfy case 2 again with s playing the role of v and s_t playing the role of w. For both case 1 and case 2 we now describe what to do if all the vertices of I_{i+1} satisfy that case (if all vertices of I_{i+1}

happen to satisfy both cases, then we go with case 1). Each number here corresponds with what to do in that case.

1. In case 1, set P_i to the subpath of L_i that is made up of vertices numbered less than $n(\ell(s_t))$, set $R_{i+1} = R_i$, and set L_{i+1} to be the vertices of L_i numbered greater than $n(\ell(s_t))$.

2. In case 2, set P_i to the subpath of R_i that is made up of vertices numbered less than $n(r(s_t))$, set $L_{i+1} = L_i$, and set R_{i+1} to be the vertices of R_i numbered greater than $n(r(s_t))$.

This concludes the i^{th} step. We now show that in case 1, all properties required of L_{i+1}, R_{i+1}, I_{i+1}, and P_i are met (the argument is identical when we are in case 2). It is straight forward to see that L_{i+1} is a subpath of L_i, R_{i+1} is a subpath of R_i, and $I_{i+1} \subseteq I_i$. Since I_i was assumed to have size at least $(8k^2)^{k-i+2}$ and $|I_{i+1}| \geqslant \frac{1}{8k}|I_i|$, it holds that $|I_{i+1}| \geqslant (8k^2)^{k-i+1}$. Since L_{i+1} is made up of vertices of L_i numbered greater than $n(\ell(s_t))$ and for every vertex $s \in I_{i+1}$ it holds that $n(\ell(s)) \geqslant n(\ell(s_t))$ it follows that $\ell(s) \in L_{i+1}$. Also, since $R_{i+1} = R_i$ it follows that $r(s) \in R_i$. Lastly, we can see that P_i is a subpath of L_i and is anti-complete with L_{i+1} and R_{i+1}. By definition every vertex $s \in I_{i+1}$ must have a neighbor $x \in L_i$ such that $n(x) \leqslant n(\ell(s_t))$ and therefore $x \in P_i$ so P_i dominates I_{i+1}.

Now we observe that for $a < b$ that P_a dominates I_b. This follows from the fact that P_a dominated I_{a+1} and that $I_b \subseteq I_{a+1}$ (since $I_a \subseteq I_{a+1} \subseteq \ldots I_b$). Also observe that P_a is anti-complete with P_b since P_a is anti-complete with L_{a+1} and R_{a+1} and therefore L_b and R_b, which P_b is a subpath of. Hence the P_a's are pairwise anti-complete. Lastly, observe that if $x, y \in I_{i+1}$ then $x, y \in I$ which means x and y share no neighbors in P_L and P_R. We can now see that the conditions of Lemma 6.5.13 are satisfied, therefore this process cannot go past the k^{th} step without producing a k-creature.

We conclude there is some step $j \leqslant k$ such that the auxiliary graph A_j has max in-degree less than $\frac{1}{4k^2}|I_j|$, and since $|I_j| \geqslant (8k^2)^{k-i+2} \geqslant 8k^2$ it therefore has an independent set of size k^2 by Lemma 6.5.14. Let I^* denote such an independent set, we claim that $G[V(L_j) \cup I^* \cup V(R_j)]$ makes an k^2-almost-skinny-ladder. Let $x, y \in I^*$ and let a, b be the highest and lowest numbered neighbors of x in L_j respectively, and assume that y has a neighbor c on the induced path of L_j that has a and b as its endpoints. If y's highest numbered neighbor in L_j is greater than $n(a)$ then y has an edge to x in A_j by case 1. If y's highest numbered neighbor is L_j is less than $n(a)$, then x has an edge to y again by case 1. Both cases yield a contradiction to I^* being an independent set in A_j. A symmetric argument show that if a', b' are x's highest and lowest numbered neighbors R_j respectively, then y cannot have a neighbor in the induced subpath of R that has a', b' as its endpoints. It follows that $G[V(L_j) \cup I^* \cup V(R_j)]$ is a k^2-almost-skinny-ladder. Applying Lemma 6.5.12 shows that G contains a k-skinny-ladder as an induced minor.

■

The following lemma uses a branching algorithm to produce all of the minimal separators of a graph G and proves a bound on the number of minimal separators produced by this algorithm. See Claim 6.2.3 of Section 6.2 and the discussion following the claim for a high level description of Lemma 6.5.16.

Lemma 6.5.16. *There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that the following holds. Let G be a graph and let k and c be integers such that for all induced subgraphs G' of G and for all $v \in G'$, if $S^v_{G'} = \{N(v) \cap S : v \notin S \text{ and } S \text{ is a minimal separator of } G'\}$, then $|S^v_{G'}| \leqslant c$ and every minimal separator of any induced subgraph of G can be dominated by k vertices. Then G has a most $(c + n^k)^{f(k)\log(n)}$ minimal separators where $n = |V(G)|$.*

Proof: Let G, S^v_G, k, c, and n be as in the statement of this lemma. The proof of the bound makes use of a branching algorithm. The algorithm takes as input G and

$X \subseteq V(G)$ and the algorithm will use the set K_{ret} to store the vertex sets it will return. It will return K_{ret} which will contain all minimal separators of G contained in X (possibly along with other vertex sets which are not minimal separators). We have no concern about the runtime of the algorithm, but we care about the size of the final set it returns. The algorithm is intended to be used initially on the input $(G, V(G))$.

Assume the the input to the algorithm is (G, X). If X is empty, then the algorithm returns $\{\emptyset\}$ (if G is disconnected then $\emptyset$ is a minimal separator of G). Else, the algorithm determines the set $Q \subseteq V(G)$ where Q contains all vertices $v \in G$ such that $|N[v] \cap X| \geqslant \frac{1}{2k}|X|$. The algorithm then initializes K_{ret} to $\emptyset$ then branches in the following two ways:

1. For every $q \in Q$ and every $Y \in S_G^q$ the algorithm recursively calls itself on $(G - Y,$ $X - N_G[q])$. The recursive call $(G - Y,\ X - N_G[q])$ returns the collection K' of vertex sets and for each set S in K', the algorithm adds the set $S \cup Y$ to K_{ret}.

2. For every set R of k vertices of G such that $R \cap Q = \emptyset$, the algorithm recursively calls itself on $(G-Q,\ (X-Q) \cap N_G(R))$. The recursive call $(G-Q,\ (X-Q) \cap N_G(R))$ returns a collection K' of vertex sets. Then for each set, S in K' the algorithm adds the set $S \cup Q$ to K_{ret}.

After completing this, the algorithm then returns the set K_{ret}.

This algorithm will terminate since each recursive call is on input $(G',\ X')$ where the size of X' is strictly less than X. Note that since the set R has no vertex in Q and $|R| = k$, the neighborhood of R contains at most $\frac{1}{2}$ of the vertices of X, so in (2) each recursive call made is on input $(G',\ X')$ where $|X| \geqslant \frac{1}{2}|X'|$, additionally note by how the vertices of Q were chosen, in (1) each recursive call is made on input $(G',\ X')$ where is made on $|X| \geqslant \frac{1}{2k}|X'|$.

We now show that if this algorithm is called on an instance (G, X) the set returned from this algorithm contains all minimal separators of G contained in X. Let S be a

minimal separator of G contained in X. Assume all of the recursive calls (G', X') the algorithm makes returns a set that contains all minimal separators of G' contained in X', possibly along with additional vertex sets (note that the base case for when $X = \emptyset$ is handled by returning $\{\emptyset\}$). If $Y = N_G(q) \cap S$ for some $q \in G$ and $q \notin S$, then $S - Y$ is a minimal separator of $G - Y$ that is contained in $X - N_G[q]$. So if there is a $q \in Q$ such that $q \notin S$, then S gets added to K_{ret} in (1). If $Q \subseteq S$, then $S - Q$ is a minimal separator of $G - Q$, and by assumption there exists some collection of at most k vertices, R, in $G - Q$ such that $S - Q \subseteq N_{G-Q}(R)$ and therefore $S - Q \subseteq (X - Q) \cap N_G(R)$. It follows that in this case we also have S gets added to K_{ret} in (2). Induction on the the depth of the recursive call now shows that this algorithm returns all minimal separators.

Let $T(n, x)$ denote the maximum number of minimal separators that a vertex set X of size at most x can contains for any graph G with $|V(G)| \leqslant n$ and $X \subset V(G)$, such that the graph G satisfies the conditions of the lemma. The algorithm just shown makes at most cn recursive calls in (1) and n^k recursive calls in (2), each on an instance (G', X') where $|X| \geqslant \frac{1}{2k}|X'|$. Hence, $T(n, x) \leqslant (cn + n^k)T(n, [1 - \frac{1}{2k}]x)$. Using the fact that $(1 - \frac{1}{y})^y \leqslant \frac{1}{e} < \frac{1}{2}$ we expand the inequality $T(n, x) \leqslant (cn + n^k)T(n, [1 - \frac{1}{2k}]x)$ out $2k$ times to get $T(n, x) \leqslant (cn + n^k)^{2k}T(n, \frac{1}{2}x)$, then expanding this inequality $\log(x)$ times gives $T(n, x) \leqslant (cn + n^k)^{2k \log(x)}T(n, 1)$. Since $T(n, 1) = 2$ it follows that $T(n, x) \leqslant 2(cn + n^k)^{2k \log(x)}$. By taking the initial set X to be $V(G)$, it follows that G then contains at most $2(cn + n^k)^{2k \log(n)}$ minimal separator. ■

We are now ready to prove Theorem 6.1.2.

Proof: [Proof of Theorem 6.1.2] Let G be a graph that is k-creature-free and has no k-skinny-ladder as an induced minor and let $n = |V(G)|$. For every induced subgraph G' of G and for every $v \in G'$, let $S^v_{G'} = \{N(v) \cap S : v \notin S$ and S is a minimal separator of $G'\}$. Then $|S^v_{G'}| \leqslant n^k$ for by Lemma 6.5.6. By Lemma 6.5.15, since G is k-creature-free and has no k-skinny-ladder as an induced minor every minimal separator of any induced

subgraph of G' is dominated by $k' = 2k^4[(8k^2)^{k+1}]$ vertices. Lemma 6.5.16 then implies that G has at most $2(n^{k+1} + n^{k'})^{2k' \log(n)}$ minimal separators. It follows that the family of graphs that are k-creature-free and do not contain a k-skinny-ladder as an induced minor are strongly-quasi-tame. ■

6.6 Finite Forbidden Induced Subgraphs

In this section we will provide the lemmas needed in the proofs of Theorems 6.1.3 and 6.1.4 as well give a proof of these theorems. The majority of the work of this section goes into proving that given an integer k, if G contains a k'-creature for large enough k', then G must contain a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, or k-ladder as an induced subgraph.

We will require a number of new graph definitions for this section. The following graphs, except for k-ladder graphs, appear in the statement of Theorem 6.1.4. Figure 6.2 depicts these graphs. It can be seen that all graphs here except for k-claw and k-paw graphs contains at least 2^{k-2} minimal separators.

- A graph G is a *k-theta* if G consist of two vertices a, b and k induced paths $P_1, P_2, \ldots P_k$. For $1 \leqslant i \leqslant k$ the end points of P_i are a and b, every P_i is anti-complete with P_j, and every P_i has length at least 4.

- A graph G is a *k-prism* if G consist of two disjoint cliques $a_1, a_2, \ldots, a_k$ and $b_1, b_2, \ldots, b_k$ along with k induced paths $P_1, P_2, \ldots P_k$ each of length at least 2. For $1 \leqslant i \leqslant k$ the end points of P_i are a_i and b_i, every $P_i - \{a_i, b_i\}$ is anti complete with P_j, and a_i is neighbors with b_j if and only if $i = j$ and P_i is a path of length 2.

- A graph G is a *k-pyramid* if G consist of a vertex a and a clique $b_1, b_2, \ldots, b_k$, where

a is anti-complete with $b_1, b_2, \ldots, b_k$, along with k induced paths $P_1, P_2, \ldots P_k$ each of length at least 3. For $1 \leqslant i \leqslant k$ the end points of P_i are a and b_i and every $P_i - \{a, b_i\}$ is anti complete with $P_j - \{a\}$.

- A graph G is a *k-ladder* if G consists of two anti-complete paths L and R with the vertices of L and R are numbered sequentially from 1 to $|V(L)|$ and 1 to $|V(R)|$, along with k anti-complete induced paths $P_1, P_2, \ldots P_k$ that are also disjoint from L and R, each of length at least 2. For $1 \leqslant i \leqslant k$ the end points of P_i are a_i and b_i. Every a_i has at least one neighbor in L, every b_i has at least one neighbor in R. Furthermore if $i > j$, then every neighbor of a_i in L has a higher number then every neighbor of b_j in L and every neighbor of b_i in R has a higher number then every neighbor of b_j in R.

- A graph G is a *k-contracted-ladder* if can be obtained from a k-ladder by contracting each of the the paths $P_1, P_2, \ldots P_k$ into single vertices.

- A graph G is a *k-ladder-theta* if G consists of an induced path L and a vertex b anti-complete with L, along with k induced paths $P_1, P_2, \ldots P_k$ that are also disjoint from L, each of length at least 3. For $1 \leqslant i \leqslant k$ the end points of P_i are a_i and b, every $P_i - \{b\}$ is anti-complete with $P_j - \{b\}$, $P_i - \{a_i\}$ is anti-complete with L, every a_i has at least one neighbor in L, and if x, y are neighbors with a_i in L, then no a_j with $i \neq j$ has a neighbor in the induced subpath of L that has x and y as its endpoints.

- A graph G is a *k-ladder-prism* if G consists of an induced path L and clique $b_1, b_2, \ldots, b_k$ where L is anti-complete with $b_1, b_2, \ldots, b_k$, along with k induced paths $P_1, P_2, \ldots P_k$ that are also disjoint from L, each of length at least 2. For $1 \leqslant i \leqslant k$ the end points of P_i are a_i and b_i, every $P_i - \{b_i\}$ is anti-complete with P_j, $P_i - \{a_i\}$

is anti-complete with L, every a_i has at least one neighbor in L, and if x, y are neighbors with a_i in L, then no a_j with $i \neq j$ has a neighbor in the induced subpath of L that has x and y as its endpoints.

- A graph G is a *k-claw* if G consists of k anti-complete copies of the following graph which we call a *long-claw of arm length k*: let v be a vertex and P_1, P_2, P_3 be three paths of length k each with v as one of its endpoints and $P_i - \{v\}$ is anti-complete with $P_j - \{v\}$ (i.e., the graph is a claw with each edge subdivide $k-2$ times)

- A graph G is a *k-paw* if G consists of k anti-complete copies of the following graph which we call a *long-paw of arm length k*: let v_1, v_2, v_3 be a triangle and P_1, P_2, P_3 be three disjoint induced paths of length k each such that P_i has v_i as one of its endpoints and $P_i - \{v_i\}$ is anti-complete with P_j for $1 \leqslant i \neq j \leqslant 3$.

It will be useful in this section to define the following graphs as well. These graphs are depicted in Figure 6.9.

- A graph G is a *k-half-theta* if G consists of a vertex v and k induced paths P_1, $P_2, \ldots, P_k$ of G such that each path has length at least 2, for $1 \leqslant i \leqslant k$ it holds that v is one endpoint of P_i, and for $j \neq i$ it hold that $P_i - v$ is anti-complete with $P_j - v$. Let x_i denote the endpoint of P_i that is not v. Then we say the vertices $x_1, x_2, \ldots, x_k$ are the endpoints of the k-half-theta. If X is a vertex set and $x_i \in X$ for all i with $1 \leqslant i \leqslant k$, then we say G is a k-half-theta ending in X.

- A graph G is a *k-half-prism* if G consists of a clique of vertices $v_1, v_2, \ldots, v_k$ and k induced paths P_1, P_2, …, P_k of G such that each path has length at least 1, for $1 \leqslant i \leqslant k$ it holds that v_i is one endpoint of P_i, and for $j \neq i$ it hold that $P_i - v_i$ is anti-complete with P_j. If the length of P_i is greater than 1 then let x_i denote the endpoint of P_i that is not v_i, and if the length of P_i is 1 then let $x_i = v_i$. We say

the vertices $x_1, x_2, \ldots, x_k$ are the endpoints of the k-half-prism. If X is a vertex set and $x_i \in X$ for all i with $1 \leqslant i \leqslant k$, then we say G is a k-half-prism ending in X.

- A graph G is a *k-half-ladder* if G consists of a path P of G along with k additional paths $P_1, P_2, \ldots, P_k$ of G such that each path has length at least 1. For $1 \leqslant i \leqslant k$ let P_i's endpoints be v_i and x_i (with v_i possibly equal to x_i). We call P the backbone path and the P_i's the auxiliary paths. We require that v_i has at least one neighbor in P, P is anti-complete with $P_i - v_i$, and for $j \neq i$ P_i is anti-complete with P_j. Lastly, we also require that if a and b are two neighbors of some v_i in P, then there is no v_j, $i \neq j$ such that v_j has a neighbor in the induced subpath of P with endpoint a and b. We say the vertices $x_1, x_2, \ldots, x_k$ are the endpoints of the k-half-ladder. If X is a vertex set and $x_i \in X$ for all i with $1 \leqslant i \leqslant k$, then we say G is a k-half-ladder ending in X.

- A graph G is a *k-half-quasi-ladder* if G consists of a path P of G along with k additional paths $P_1, P_2, \ldots, P_k$ of G such that each path has length at least 1. For $1 \leqslant i \leqslant k$ let P_i's endpoints be v_i and x_i (with v_i possibly equal to x_i). We call P the backbone path and the P_i's the auxiliary paths. We require that v_i has at least one neighbor in P, P is anti-complete with $P_i - v_i$, and for $j \neq i$ P_i is anti-complete with P_j. We say the vertices $x_1, x_2, \ldots, x_k$ are the endpoints of the k-half-quasi-ladder. If X is a vertex set and $x_i \in X$ for all i with $1 \leqslant i \leqslant k$, then we say G is a k-half-ladder ending in X. Note that a k-half-quasi-ladder is almost the same as a k-half-ladder, but we drop the requirement that if a and b are two neighbors of some v_i in P, then there is no v_j, $i \neq j$ such that v_j has a neighbor in the subpath of P with endpoint a and b.

The following lemmas, culminating with Lemma 6.6.10, work towards proving that given an integer k, if G contains a k'-creature for large enough k', then G must contain

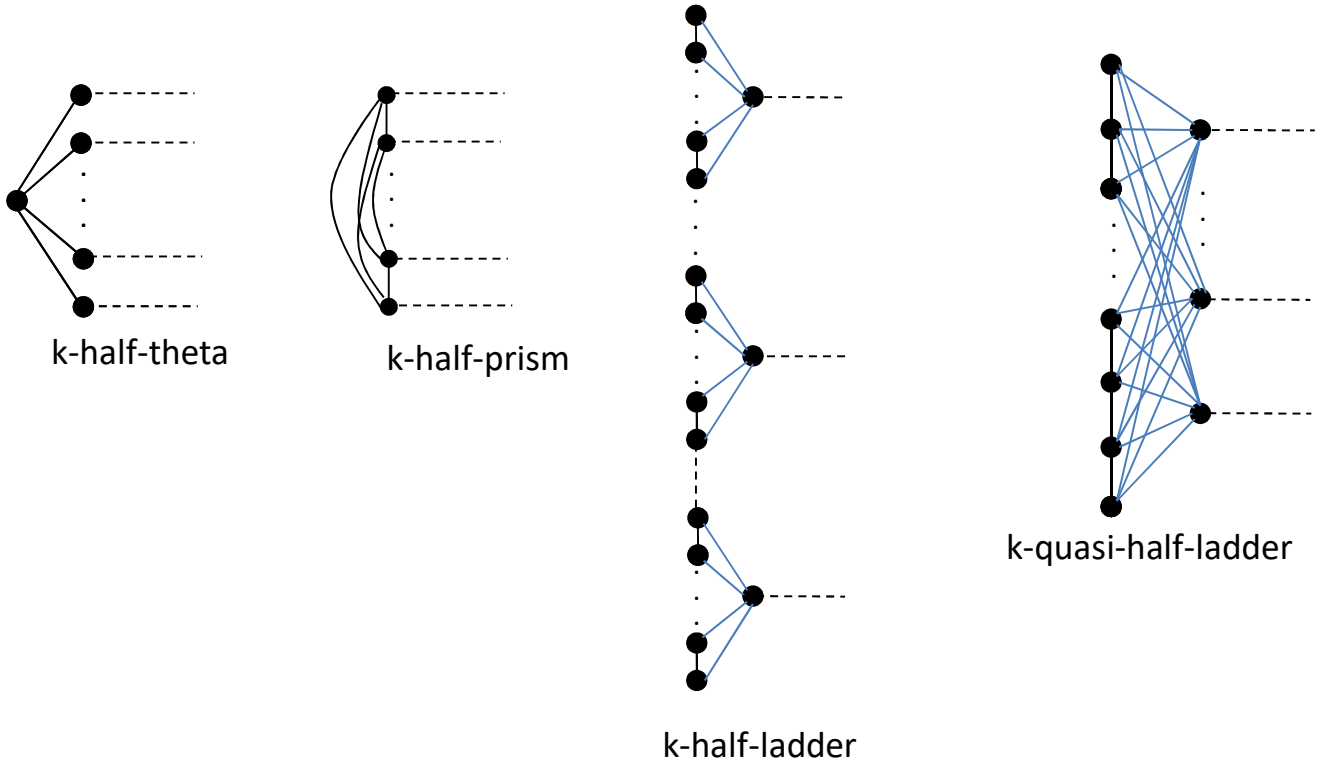


Figure 6.9: Dashed lines represent the option of having an arbitrary length path (possibly of length 0). The blue lines used in the k-half-ladder and k-almost-half-ladder graphs represents the option of either having or not having that edge, but for each vertex not on the backbone path that is adjacent at least one blue edges, at least one of those blue edges must belong to the graph.

an induced k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, or k-ladder.

Our first goal is to show that if we have a k' creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$) for large enough k', then if we focus in on one half, say that half with A and $\{x_1, x_2, \ldots, x_{k'}\}$, then we can find a k-half-theta, k-half-prism, or k-half-quasi-ladder that that ends in $\{x_1, x_2, \ldots, x_{k'}\}$. This goal is accomplished with Lemmas 6.6.1 through 6.6.4. Since if we have a k-clique in the set $\{x_1, x_2, \ldots, x_{k'}\}$ then we have a k-half-prism ending in $\{x_1, x_2, \ldots, x_{k'}\}$ we may assume, by Ramsey's Theorem, that $\{x_1, x_2, \ldots, x_{k'}\}$ is an independent set.

Lemma 6.6.1. *Let G be a graph that contains a k-creature (A, B, $\{x_1, x_2, \ldots, x_k\}$, $\{y_1, y_2, \ldots, y_k\}$) where $\{x_1, x_2, \ldots, x_k\}$ is an independent set of G. Let A' be a minimally connected induced subgraph of $G[A]$ such that $\{x_1, x_2, \ldots, x_k\} \subset N(A')$. If A' contains a vertex with degree at least $R(d, d)$ in A', then $G[A \cup \{x_1, x_2, \ldots, x_k\}]$ contains a d-half theta or a d-half-prism ending in $\{x_1, x_2, \ldots, x_k\}$.*

Proof: Let G be a graph that contains a k-creature (A, B, $\{x_1, x_2, \ldots, x_k\}$, $\{y_1, y_2, \ldots, y_k\}$). Let A' be a minimally connected induced subgraph of $G[A]$ such that $\{x_1, x_2, \ldots, x_k\} \subset N_G(A')$. Assume $v \in A'$ has degree at least $\mathrm{R}(d, d)$ in A'. Let $v_1, v_2, \ldots, v_{R(d,d)}$ be distinct neighbors of v in A'. By the minimality of A', for each v_i there must be a vertex x_{v_i} such that every path starting from v and ending at x_{v_i} with internal vertices contained A' must contain v_i, since if this does not happen for some given v_i then the connected component of $A' - v_i$ that contains v would be a proper induced subgraph of A' that is connected and whose open neighborhood contains $\{x_1, x_2, \ldots, x_k\}$. It follows there must exist induced paths $P_1, P_2, \ldots, P_{R(d,d)}$ such that $v_i \in P_i$, P_i's endpoints are v_i and x_{v_i}, and $P_i - v_i$ is anti-complete with P_j. We then apply Ramsey's Theorem to the v_i's get a subset of size d of the P_i's that along with v form a d-half theta that ends in $\{x_1, x_2, \ldots, x_k\}$ (if Ramsey's Theorem provides an independent set of size d) or a subset of size d of the P_i's

that form a d-half prism that ends in $\{x_1, x_2, \ldots, x_k\}$ (if Ramsey's Theorem provides a clique of size d) and the result now follows. ■

Lemma 6.6.2. *Let G be connected graph with maximum degree d and contains at least d^k vertices with degree greater than 2. Then there exists an induced path of G that contains at least k vertices of degree greater than 2.*

Proof: Let G be a connected graph with maximum degree d and contains at least d^k vertices with degree greater than 2. Let T be a breadth first search tree of G rooted at some vertex $v \in G$. We create the desired path as follows. Let v_1 be the first descendent of v in T that has degree greater than 2 in G (v_1 could be v). We begin our path at v_1. We will grow the path $P_i = \{x_1, x_2, \ldots x_m\}$ where $x_1 = v_1$, x_j is the parent of x_{j+1} in T, P_i contains at least i vertices of G with degree greater than 2 in G, and the subtree of T rooted at x_m contains at least d^{k-i+1} vertices of G with degree greater than 2 in G.

Assume that we have such a path $P_i = \{x_1, x_2, \ldots x_m\}$, $i < k$ (the vertex v_1 satisfies the conditions of P_1). We will show how to attain P_{i+1}. Since the maximum degree in G is d, x_m has at most d children in T, and by assumption the subtree of T rooted at x_m has at least d^{k-i+1} vertices of degree greater than 2 in G, it follows that for at least one child, call it x_{m+1}, the subtree rooted at x_{m+1} has at least d^{k-i} vertices of G with degree greater than 2 in G. Now let v_{i+1} be the first descendant of x_{m+1} with degree different from 2 in G (v_{i+1} could be x_{m+1}) and let P_{i+1} be the path P_i along with the induced path in T from x_{m+1} to v_{i+1}. It follows P_{i+1} satisfies the required conditions.

Hence we can produce a P_k that satisfies the conditions stated before, and we can then see that P_k is an induced path in G with at least k vertices of degree greater than 2. ■

Lemma 6.6.3. *Let G be a graph that contains a k-creature (A, B, $\{x_1, x_2, \ldots, x_k\}$, $\{y_1, y_2, \ldots, y_k\}$) where $\{x_1, x_2, \ldots, x_k\}$ is an independent set. Let A' be a minimally*

connected subgraph of $G[A]$ such that $\{x_1, x_2, \ldots, x_k\} \subset N(A')$. If A' contains an induced path, P, with at least $R(d,d)$ vertices of degree greater than 2 in A', then there is a d-half-quasi-ladder or a d-half-prism in $G[A \cup \{x_1, x_2, \ldots, x_k\}]$ that ends in $\{x_1, x_2, \ldots, x_k\}$.

Proof: Let G, A', $\{x_1, x_2, \ldots, x_k\}$, and P be as in the statement of the lemma, let $v_1, v_2, \ldots, v_{R(d,d)}$ be vertices of P that have degree greater than 2 in A', and for each v_i let v_i' be a neighbor of v_i in A' that is not in P. By the minimality of A', for each v_i' there must exist a vertex x_{v_i} such that every path from v_i to x_{v_i} with internal vertices contains in A' must contain v_i', since if this does not happen for some given v_i' then the component of $A' - v_i'$ that contains v_i would be a proper induced subgraph of A' that is connected and whose open neighborhood contains $\{x_1, x_2, \ldots, x_k\}$. It follows there must exists induced paths $P_1, P_2, \ldots, P_{R(d,d)}$ disjoint from P with internal vertices contained in A', P_i's endpoints are v_i' and x_{v_i}, and $P_i - v_i'$ is anti-complete with P_j. We then apply Ramsey's Theorem to the v_i''s to get a subset of size d of the P_i's along with P that form a d-half-quasi-ladder that ends in $\{x_1, x_2, \ldots, x_k\}$ (if Ramsey's Theorem provides an independent set of size d) or a subset of size d of the P_i's that yield a d-half-prism that ends in $\{x_1, x_2, \ldots, x_k\}$ (if Ramsey's Theorem provides a clique of size d). ■

Lemma 6.6.4. *Let G be a graph that contains a $k \cdot (d^{c+1}+d)$-creature (A, B, $\{x_1, x_2, \ldots, x_{k\cdot(d^{c+1}+d)}\}$, $\{y_1, y_2, \ldots, y_{k\cdot(d^{c+1}+d)}\}$). Let A' be a minimally connected subgraph of $G[A]$ such that $\{x_1, x_2, \ldots, x_{k\cdot(d^{c+1}+d)}\} \subset N(A')$. Assume the max degree in A' is d and that A' contains less than d^c vertices of degree greater than 2 in A'. Then $G[A \cup \{x_1, x_2, \ldots, x_{k\cdot(d^{c+1}+d)}\}]$ contains a k-half-quasi-ladder ending in $\{x_1, x_2, \ldots, x_{k\cdot(d^{c+1}+d)}\}$.*

Proof: Let G, A' and $\{x_1, x_2, \ldots, x_{k\cdot(d^{c+1}+d)}\}$ be as in the statement of the lemma. Let T be a breadth first search tree of A' rooted at some vertex v. Then T is a tree in which every vertex except for the root can have at most $d-1$ children, hence there are at most $d^c + 1$ vertices that have more than one descendent, and the maximum number

of descendants any vertex from this set can have is d. It follows that there are at most $d^{c+1}+d$ leaves of T, and therefore A' is the union of at most $d^{c+1}+d$ induced paths in A'. Hence, there exists some induced path P in A' such that P's open neighborhood contains at least k vertices in $\{x_1, x_2, \ldots, x_{k\cdot(d^{c+1}+d)}\}$, which gives us a k-half-quasi-ladder ending in $\{x_1, x_2, \ldots, x_{k\cdot(d^{c+1}+d)}\}$. ■

Lemma 6.6.5. *Let $k' = k \cdot R(k,k)^{R(k,k)+1} + R(k,k)$, and let G be a graph that contains an $R(k',k')$-creature (A, B, $\{x_1, x_2, \ldots, x_{R(k',k')}\}$, $\{y_1, y_2, \ldots, y_{R(k',k')}\}$). Then $G[A \cup \{x_1, x_2, \ldots, x_{R(k',k')}\}]$ contains an induced k-half-theta, k-half-prism, or a k-half-quasi-ladder, ending in $\{x_1, x_2, \ldots, x_{R(k',k')}\}$.*

Proof: Let $k' = k \cdot R(k,k)^{R(k,k)+1} + R(k,k)$. Assume that G contains a $R(k',k')$-creature (A, B, $\{x_1, x_2, \ldots, x_{R(k',k')}\}$, $\{y_1, y_2, \ldots, y_{R(k',k')}\}$). Apply Ramsey's Theorem to $\{x_1, x_2, \ldots, x_{R(k',k')}\}$. If Ramsey's Theorem returns a clique of size k' or more then we have that $G[A \cup \{x_1, x_2, \ldots, x_{R(k',k')}\}]$ contains a k-half-prism ending in $\{x_1, x_2, \ldots, x_{R(k',k')}\}$, so we can assume that Ramseys theorem returns an independent set of size at least k'. By relabeling the x_i's and y_i's if follows that G contains a k'-creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$) where $\{x_1, x_2, \ldots, x_{k'}\}$ is an independent set.

Let A' be a minimally connected induced subgraph of $G[A]$ such that $\{x_1, x_2, \ldots, x_{k'}\} \subset N(A')$. If A' contains a vertex of degree $\mathrm{R}(k,k)$ in A', then by Lemma 6.6.1 $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains a k-half-theta ending in $\{x_1, x_2, \ldots, x_{k'}\}$. So we may assume max degree of A' is $\mathrm{R}(k,k)$.

If A' contains $\mathrm{R}(k,k)^{R(k,k)}$ vertices of degree greater than two, then there is an induced path of A' that contains $\mathrm{R}(k,k)$ vertices of degree greater than two by Lemma 6.6.2. Then by Lemma 6.6.3 $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains a k-half-quasi-ladder or a k-half-prism ending in $\{x_1, x_2, \ldots, x_{k'}\}$. So we may assume that A' has maximum degree $\mathrm{R}(k,k)$ and contains fewer than $\mathrm{R}(k,k)^{R(k,k)}$ vertices of degree greater than two. It then follows

from Lemma 6.6.4 that $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains a k-half-quasi-ladder ending in $\{x_1, x_2, \ldots, x_{k'}\}$. ■

If in the statement of Lemma 6.6.5 we would replace k-half-quasi-ladder with k-half-ladder, then we would basically be done. If we had a k''-creature (A, B, $\{x_1, x_2, \ldots, x_{k''}\}$, $\{y_1, y_2, \ldots, y_{k''}\}$) for large enough k'' then we could find a k'-half-theta, k'-half-prism, or k'-half-ladder in $A \cup \{x_1, x_2, \ldots, x_{k''}\}$. We could then switch over to the other side with B and $\{y_1, y_2, \ldots, y_{k''}\}$ and restricting our self to the k' vertices of $\{y_1, y_2, \ldots, y_{k''}\}$ that match up with end endpoints of the k'-half-theta, k'-half-prism, or k'-half-ladder in $\{x_1, x_2, \ldots, x_{k''}\}$ we just found, repeat the same process in B and $\{y_1, y_2, \ldots, y_{k''}\}$ to find a k-theta, k-pyramid, k-prism, k-ladder, k-ladder-theta, k-ladder-prism. Our goal then now is clear, we must clean up a k'-half-quasi-ladder to give use a k-half-ladder (or possible a k-half theta or even a k-theta).

The next three lemmas show how to clean up a half-quasi-ladder into a half-ladder, half-theta, or theta. Their proofs are similar to those of lemmas 6.5.9 6.5.11, and 6.5.15 respectively, although the conclusions we draw from them are somewhat different.

So, we are now in a situation where we have have found a k'-half-quasi-ladder. Let us say that P is the backbone path, and $P_1, P_2, \ldots, P_{k'}$ are the auxiliary paths of the k'-half-quasi-ladder where P_i has endpoints s_i and x_i, and let the x_i's be the endpoint of our half-quasi-ladder. Set S equal to the set of s_i's. To turn our k'-half-quasi-ladder into a k-half-ladder, we first want to find a subset $S' \subset S$ such that no vertices of S' share a neighbor in P. Notice that if any vertex of P has k neighbors in S, then we have a k-half-theta ending in $\{x_1, x_2, \ldots, x_{k'}\}$ and we are done, so we can assume that the vertices of S cannot be dominated by a small set of vertices of P. So, what the next lemma shows that if we take some $v \in S$ then we can either find a k-half-theta ending in V (and therefore W) or find a small set $X \subset V(P)$ such that no vertex of $S - N[X]$ shares a neighbor with v in P. As S cannot be dominated by a small subset of P, this

lemma can be repeatedly used to give us a large subset S' of S such that no two vertices of S' share a neighbor in P, which is precisely what we do in Lemma 6.6.7.

Lemma 6.6.6. *Let (G, S, P, v) be a tuple where G is a graph, $v \in G$, $S \subset V(G)$, and P is an induced path of G such that $(S \cup \{v\})$ and $V(P)$ are disjoint. Assume $G[V(P) \cup S \cup \{v\}]$ does not have a k-half-theta ending in S, then there is a set $X \subset S \cup V(P) \cup \{v\}$ of size at most $4k-1$ such that $N(S - N[X]) \cap N(v) \cap V(P) = \emptyset$, and no vertex of $S - N[X]$ is neighbors with v.*

Proof: Let G, S, P, and v be as in the statement of this lemma. Number the vertices of P 1 through $|V(P)|$ such that the vertex numbered i is neighbors with the vertices numbers $i-1$ and $i+1$. We now consider the following process to build the set desired set X such that $N(S - N[X]) \cap N(v) \cap V(P) = \emptyset$ and $X \subset S \cup V(P) \cup \{v\}$.

We do the following for the first step of the process. Let $X_1 = \{v\}$, and let $S_1 = \{s : s \in S - N(X_1)$ and $N(s) \cap N(v) \cap V(P) \neq \emptyset\}$ (i.e., S_1 is the set of vertices of $S - N(X_1)$ that share a neighbor with v in P). Label the vertices of S_1 by the lowest numbered vertex it is neighbors with in $V(P) \cap N(v)$. Let s_1 be a highest labeled vertex in S_1, and let p_1 be s_1's lowest numbered neighbor in $N(v) \cap V(P)$. This completes the first step.

For the i^{th} step we do the following. Let $X_i = X_{i-1} \cup \{s_{i-1}, p_{i-1}\}$, and let $S_i = S_{i-1} - N[X_i]$ and label the vertices of S_i by the lowest vertex it sees in $V(P) \cap N(v)$ (the vertices of S_i inherit their labels from their labels in S_{i-1}). Let s_i be a highest labeled vertex in S_i and let p_i be s_i's lowest neighbor in $N(v) \cap V(P)$. Note by how we selected v, s_1, p_1, s_2, p_2, ... s_i, p_i that s_a, $1 \leqslant a \leqslant i$, cannot be neighbors with p_b if $a > b$ since p_b would be in X_a and therefore s_a would not be in S_a, and s_a cannot have a neighbor with p_b if $a < b$ since that would contradict either p_a being s_a's lowest numbered neighbor in $N(v) \cap P$ or s_a being a highest labeled vertex in S_a. Hence, we then have that among these vertices s_j is only neighbors with p_j for $1 \leqslant j \leqslant i$, and v is

only neighbors with p_j for $1 \leqslant j \leqslant i$. p_{2i} could be neighbors with p_{2i+1} and/or p_{2i-1} since they could be consecutive vertices on the path P, but p_{2i} cannot be neighbors with p_{2j}. It follows that the set $\{v\} \cup \{p_2, p_4, \ldots, p_{2c}\} \cup \{s_2, s_4, \ldots, p_{2c}\}$, $2c \leqslant i$, forms a c-half-theta in $G[V(P) \cup S\{v\}]$ ending in S.

We continue this process until we reach an S_j that is empty. By what we noted in the previous paragraph, this process cannot go past the $2k^{th}$ step if $G[V(P) \cup S \cup \{v\}]$ does not contain a k-half-theta ending in S. Set X to be X_j. Since S_j is empty, it follows $N(S - N[X]) \cap N(v) \cap V(P) = \emptyset$. We also have that no vertex of $S - N[X]$ is neighbors with v since $v \in X$ and $|X| \leqslant 4k - 1$ since $j \leqslant 2k$ and since the first step adds a single vertex and each step after that only adds two vertices. ■

Lemma 6.6.7. *Let (G, S, P) be a tuple such that G is a graph, $S \subset V(G)$ such that S cannot be dominated by $4kx$ vertices and P is an induced path disjoint from S that dominates S. Assume $G[V(P) \cup S]$ does not contain a k-half-theta ending in S. Then there exists a subset S' of S of size x such that no vertex of P has more than one neighbor in S'.*

Proof:

Let G, S, and P be as in the statement of the lemma. Assume that we have an independent set of vertices vertices S_{i-1} of size $i - 1$, $i \leqslant k$, and a set Z_{i-1} of size at most $4k(i-1)$, with the properties that no vertex $S - N[Z_{i-1}]$ is neighbors with a vertex in S_{i-1}, and any vertex in P that is neighbor with some vertex in S_{i-1} has no other neighbors in S_{i-1} nor in $S - N[Z_{i-1}]$. We will use this to produce a set S_i of size i and Z_i of size at most $4k^2i$ with the same properties. Note that the empty set satisfies the conditions of S_0.

Let $S' = S - N[Z_{i-1}]$. Let s be some vertex in S', since $i \leqslant k$ and S cannot be dominated by $4kx$ vertices, such an s must exists. We can then apply Lemma 6.6.6 using

(G, S', P, s) and to get a set X of size at most $4k-1$ such that $(S'-N[X])\cap N(s)\cap V(P) = \emptyset$ and no vertex of $S' - N[X]$ is neighbors with s. We then set $S_i = S_{i-1} \cup \{s\}$ and $Z_i = Z_{i-1} \cup X$ and we can see these sets satisfies the required properties.

Since the empty set satisfies the properties of S_0 and S cannot be dominated by $4kx$ vertices, we can continue the process until we generate the set S_x which has size x and no vertex of P has more than one neighbor in S_x. ■

The previous two lemmas now give us a k'-half-quasi-ladder with backbone path P, auxiliary paths $P_1, P_2, \ldots, P_{k'}$ where P_i has endpoints s_i and w_i such that no vertex of P is neighbors with more than one of the s_i's. The next lemma now show use how to take such a k'-half-quasi-ladder and produce a k-half-ladder.

Lemma 6.6.8. *Let T be an induced $4k[2(4k)^{k+1}]^2$-half-quasi-ladder of a graph G ending in X. Assume T does not have an induced k-half-theta ending in X and assume that G does not contain an induced k-theta. Then T contains a k-half-ladder ending in X.*

Proof: Let G, T, and X be as in the statement of the lemma. Let P be the backbone path of T and $P_1, P_2, \ldots, P_{4k[2(4k)^{k+1}]^2}$ be its auxiliary paths, where the endpoints of P_i are v_i and x_i, and the x_i's are the endpoints of T, so $x_i \in X$. Let $S = \{v_1, v_2, \ldots, v_{4k[2(4k)^{k+1}]^2}\}$. Clearly, if any vertex of P is neighbors with k distinct v_i's, then T contains a k-half-theta ending in X. It follows that since T does not have a k-half-theta ending in X, the vertices of S cannot be dominated by less than $4[2(4k)^{k+1}]^2$ vertices in T. Also, if $G[S \cup V(P)]$ contain a k-half-theta ending in S, then it contains a k-half-theta ending in X, so we can apply Lemma 6.6.7 with (G, P, S) to get a set $S' \subset S$ of size $2(4k)^{k+1}$ such that no vertex of P is neighbors with more than one vertex in S'. It follows that by only taking the paths P_i such that $v_i \in S'$, that these P_i's together with P, form a $2(4k)^{k+1}$-half-quasi-ladder where no vertex of P has a neighbor with more than one vertex in any of the P_i's. We will call this $2(4k)^{k+1}$-half-quasi-ladder T', we will call its backbone path P' so $P' = P$,

and we will call the auxiliary paths $P'_1, P'_2, \ldots, P'_{2(4k)^{k+1}}$ where the endpoints of P'_i are v'_i and x'_i, and the x'_i's are the endpoints of T', so $x'_i \in X$. We use S' as before to denote the set of v'_i's.

Now, number the vertices of P' 1 through $|V(P')|$ such that the vertex numbered i is neighbors with the vertices numbers $i-1$ and $i+1$. For a vertex x in P' we will use the notation $n(x)$ to denote the number it has been given in P'. For every $s_j \in S'$ let $p_j \in P'$ be the highest numbered neighbor s_j has in P. We now set $P_1 = P'$ and $S_1 = S'$. We will consider the following process, where we will try to produce a large independent set in an auxiliary graph related to some P_i and S_i which we will then use to produce a k-half-ladder. We will show this process cannot go past k iterations if T does not have a k-half-theta ending in X. We will ensure that at the i^{th} step that $V(P_i) \subset V(P')$, $S_i \subset S'$, $|S_i| \geqslant 2(4k)^{k-i+2}$, P_i is an induced path, and if $s_j \in S_i$ then $p_j \in P_i$. We will also produce induced subpaths D_i of P such that the D_i's are anti-complete with respect to one another and the vertices of D_i will dominate S_j if $i < j$.

At the i^{th} step we do as follows. Create an auxiliary directed graph, AUX_i, whose vertex set is S_i and there is an edge from $s_a \in S_i$ to $s_b \in S_i$ if the following condition holds

1. $n(p_a) > n(p_b)$ and s_a has a neighbor x in P' such that $n(x) < n(p_b)$

If the maximum in degree of AUX_i is at most $\frac{1}{4k}|S_i|$ then we stop. If $i \leqslant k$ (which we will show must happen) then since $|S_i| \geqslant 2(4k)^{k-i+2}$ this gives an independent set of size at least k by Lemma 6.5.14. If there is an $s_j \in S_i$ with in degree at least $\frac{1}{4k}|S_i|$ then for at least $\frac{1}{4k}$ fraction of the vertices of S_i must satisfy (1) playing the role of s_a while s_j plays the role of s_b. Call this set of vertices S_{i+1}. If $s_j \in S_i$ with in degree at least $\frac{1}{4k}|S_i|$ then we do as follows. Define D_i to be the subpath of P_i that is made up of vertices with numbers less than $n(p_j)$. Set P_{i+1} to be the vertices of P_i with numbers

greater than $n(p_j)$. This concludes the i^{th} step.

It can then be seen that $V(P_{i+1}) \subset V(P)$, $S_{i+1} \subset S$, $|S_{i+1}| \geqslant 2(4k)^{k-i+1}$, P_{i+1} is an induced path, and if $s_j \in S_{i+1}$ then $p_j \in P_{i+1}$ as required. Furthermore, it can be seen that any of the previously D_j's that have been produced in this process ($j \leqslant i$) dominate all vertices of S_{i+1}. Since the D_j's are disjoint and anti complete, By Lemma 6.5.13 then, this process cannot go past the k^{th} iteration without producing a k-theta in G.

We conclude there is some step $j \leqslant k$ such that the auxiliary graph AUX_j has max in-degree less than $\frac{1}{4k}|S_j|$, and since $|S_j| \geqslant 8k$ it therefore has an independent set of size k by Lemma 6.5.14. Let S^* denote such an independent set.

We claim by only taking the paths P_i' such that $v_i' \in S^*$, that these P_i''s together with P', form a k-half-ladder. Let $x, y \in S^*$ and let a, b be the highest and lowest numbered neighbors of x in L respectively, and assume that y has a neighbor c on the induced path of L that has a and b as its endpoints. If y's highest numbered neighbor in L is greater than $n(a)$ then y has an edge to x in AUX_j. If y's highest numbered neighbor in L is less than $n(a)$, then x has an edge to y. It follows that taking the P_i' such that $v_i' \in S^*$ together with P', form a k-half-ladder. ■

Corollary 6.6.9. *Let k be a natural number. There exists a natural number k' large enough such that if G is a graph that contains a k'-creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$), then $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains an induced k-half-theta, k-half-prism, or k-half-ladder ending in $\{x_1, x_2, \ldots, x_{k'}\}$ or G contains an induced k-theta.*

Proof: By Lemma 6.6.5 there exists a k' large enough such that if G contains a k'-creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$) then $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains an induced $4k[2(4k)^{k+1}]^2$-half-theta, $4k[2(4k)^{k+1}]^2$-half-prism, or a $4k[2(4k)^{k+1}]^2$-half-quasi-ladder, ending in $\{x_1, x_2, \ldots, x_{k'}\}$. If $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains a $4k[2(4k)^{k+1}]^2$-half-theta or a $4k[2(4k)^{k+1}]^2$-half-prism ending in $\{x_1, x_2, \ldots, x_{k'}\}$ then we are done. If $G[A \cup$

$\{x_1, x_2, \ldots, x_{k'}\}]$ contains a $4k[2(4k)^{k+1}]^2$-half-quasi-ladder ending in $\{x_1, x_2, \ldots, x_{k'}\}$ then we may apply Lemma 6.6.8 to get that either $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains a k-half-ladder ending in $\{x_1, x_2, \ldots, x_{k'}\}$ or G contains a k-theta. ■

We now know that given a k'-creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$) for large enough k' in each half, $A \cup \{x_1, x_2, \ldots, x_{k'}\}$ and $B \cup \{y_1, y_2, \ldots, y_{k'}\}$ we can find a k-half-theta, k-half-prism, or k-half-ladder, and we can combine them together to make a k-theta, k-prism, k-pyramid, k-ladder, k-ladder-theta, or a k-ladder-prism. The next lemma formalizes this.

Lemma 6.6.10. *Let k be a natural number. Then there exists a natural number k' large enough such that if G is a graph that contains a k'-creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$), then G contains an induced k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, or a k-ladder.*

Proof: Let k be a natural number. By Corollary 6.6.9 there exists a k' large enough such that if G is a graph that contains a k'-creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$), then $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains an induced k^2-half-theta, k^2-half-prism, or k^2-half-ladder ending in $\{x_1, x_2, \ldots, x_{k'}\}$ or G contains an induced k^2-theta. It then also follows from Corollary 6.6.9 there exists a k'' large enough such that if G is a graph that contains a k''-creature (A, B, $\{x_1, x_2, \ldots, x_{k''}\}$, $\{y_1, y_2, \ldots, y_{k''}\}$), then $G[B \cup \{y_1, y_2, \ldots, y_{k''}\}]$ contains an induced k'-half-theta, k'-half-prism, or k'-half-ladder ending in $\{y_1, y_2, \ldots, y_{k''}\}$ or G contains an induced k'-theta.

So, assume that G is a graph that contains an k''-creature (A, B, $\{x_1, x_2, \ldots, x_{k''}\}$, $\{y_1, y_2, \ldots, y_{k''}\}$). If G contains an induced k'-theta then we are done, so assume that $G[B \cup \{y_1, y_2, \ldots, y_{k''}\}]$ contains an induced k'-half-theta, k'-half-prism, or k'-half-ladder ending in $\{y_1, y_2, \ldots, y_{k''}\}$. By relabeling the x_i's and y_i's we can then assume that G contains a k' creature (A', B', $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$) such that $G[B' \cup$

$\{y_1, y_2, \ldots, y_{k'}\}]$ is a k'-half-theta, k'-half-prism, or k'-half-ladder. Then applying Corollary 6.6.9 gives us that $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains an induced k^2-half-theta, k^2-half-prism, or k^2-half-ladder ending in $\{x_1, x_2, \ldots, x_{k'}\}$. If $G[B' \cup \{y_1, y_2, \ldots, y_{k'}\}]$ is a k'-half-ladder and $G[A \cup \{x_1, x_2, \ldots, x_{k'}\}]$ contains a k^2-half-ladder, then an application of the Erdös-Szekeres Theorem gives us a k-ladder. Otherwise, it follows that G must contain a k^2-theta, a k^2-prism, k^2-pyramid, k^2-ladder-theta, or k^2-ladder-prism. ■

With Lemma 6.6.10 in hand we can now provide a proof of Theorem 6.1.3.

Proof: [Proof of Theorem 6.1.3] Let G be a graph, $|V(G)| = n$, where G forbids all k-theta, k-pyramid, k-prism, k-ladder, k-ladder-theta, and k-ladder-prism graphs as well as k-contracted-ladder graphs. By Lemma 6.6.10 there exists a function $f : \mathbb{N} \to \mathbb{N}$ (f in independent of the choice of k or G) such that G is $f(k)$-creature-free. Furthermore, we can see that any graph that contains a $2k$-skinny-ladder as an induced minor must either contain a k-ladder or a k-contracted-ladder as an induced subgraph, therefore G contains no $2k$-skinny-ladder as an induced minor. Hence by Theorem 6.1.2 there is a function $f^* : \mathbb{N} \to \mathbb{N}$ (f^* is independent of the choice of k or G) such that G has at most $n^{f^*(k)\log(n)}$ minimal separators. It follows that the family of graphs that forbid all k-theta, k-pyramid, k-prism, k-ladder, k-ladder-theta, and k-ladder-prism graphs as well as k-contracted-ladder graphs are strongly-quasi-tame. ■

The following two lemmas will be used in Lemma 6.6.13 to establish that if $\mathcal{F}$ is a family of graphs defined by a finite number of forbidden induced subgraphs and $\mathcal{F}$ allows for at least one of k-thetas, k-prisms, k-pyramids, k-ladder-thetas, or k-ladder-prisms, for arbitrarily large k, then we can ensure it contains these graphs where their number of vertices only grow linearly with respect to k, and therefore have exponentially many minimal separators. These two lemmas achieve this by showing that a graph in $\mathcal{F}$ has certain paths that are too long, then we can contract part of those paths and maintain that the resulting graph remains in $\mathcal{F}$.

Lemma 6.6.11. *Let G be a graph and let H be a graph with $|V(H)| \leqslant h$, where $h > 5$. Assume that G contains an induced path P of length at least $5h$ where all internal vertices of P have degree 2 in G. Then there exists an edge e in G such that if G^e contains H as an induced subgraph, then so does G.*

Proof: Let G be a graph, let H be a graph with $|V(H)| \leqslant h$ where $h > 5$, and let P be an induced path of G of length at least $5h$ where all internal vertices of P have degree 2, say $P = p_1, p_2, \ldots, p_{5h}$. Let e be the edge between $p_{\lceil \frac{5h-1}{2} \rceil}$ and $p_{\lceil \frac{5h+1}{2} \rceil}$. Let v denote the new vertex $p_{\lceil \frac{5h-1}{2} \rceil}$ and $p_{\lceil \frac{5h+1}{2} \rceil}$ create when e is contracted in G to make G^e, and let P' be what the path P becomes after contracting e in G, so $P' = p_1, p_2, \ldots, p_{\lceil \frac{5h-1}{2} \rceil - 1}, v, p_{\lceil \frac{5h+1}{2} \rceil + 1}, \ldots, p_{5k}$. Assume that G^e contains H as an induced subgraph. We will show that there exists a set $X \subset V(G^e)$ that induces H such that $v \notin X$. It will then follows that G contains an induced H.

Any component of H that is not an induced path can only contain vertices outside of P' or within distance h of either the endpoints of P' since all internal vertices of P' have degree 2 in G^e. For the components of H that are paths, since there are at most h vertices among these components, we can ensure that the vertices of X that we use to induce these components either do not belong to P' or only contain vertices from the subpaths $p_{h+2}, p_{h+3}, \ldots, p_{\lceil \frac{5h-1}{2} \rceil - 1}$ and $p_{\lceil \frac{5h+1}{2} \rceil + 1}, p_{\lceil \frac{5h+1}{2} \rceil}, \ldots, p_{4h-2}$. It follows that $v \notin X$. ■

Lemma 6.6.12. *Let G be a graph and let H be a graph with $|V(H)| \leqslant h$, where $h > 5$. Assume that G contains an induced path P of length $5h[(h+1)(5h)^{2h+2} + 1]$ such that the only neighbor the vertices of P might have outside of P is a single vertex v. Then there exists a subpath P' of P such that if $G^{P'}$ contains H as an induced subgraph, then so does G.*

Proof: Let G be a graph and let H be a graph with $|V(H)| \leqslant h$, where $h > 5$.

Assume that G contains an induced path P of length $5h[(h+1)(5h)^{2h+2}+1]$ such that the only neighbor the vertices of P might have outside of P is a single vertex v. Let a, b be the endpoints of P. Now divide P into a sequence of subpaths $P_1, P_2, \ldots, P_k$ each of length at least 2 so that all internal vertices of P_i have degree 2 in G, all endpoints of P_i are either a vertex of degree 3 or a or b, P_1 has a and one of its endpoints, P_k has b as one of its endpoints, and P_i shares one of its endpoints with P_{i+1} (i.e., these are subpaths that whose endpoints are a, b, or the vertices that are neighbors with v and are sequenced going from one end of P to the other). We define a second sequence $a_1, a_2, \ldots a_k$ where $a_i = |E(P_i)|$. If any $a_i \geqslant 5h$ then the result follows from Lemma 6.6.11, so we can assume for all i that $a_i \leqslant 5h$. It then follows that k is at least $(h+1)(5h)^{2h+2}+1$, and therefore by the pigeonhole principle there must be a continuous subsequence of length $2h+2$ that is repeated at least $h+2$ times, where none of these continuous subsequences overlap with each other. Let $S = s_0, s_1, \ldots, s_{2h+1}$ be this repeated subsequence. So we have $h+2$ sequences for $1 \leqslant i \leqslant h+2$, $A_i = a_{j_i}, a_{j_i+1}, \ldots, a_{j_i+2h+1}$ where for $1 \leqslant m \leqslant h+2$ and c, $0 \leqslant c \leqslant 2h+1$, $a_{j_m+c} = s_c$ and no part of A_m overlaps with some other A_n (so $|j_n - j_m| \geqslant 2h+2$) and $j_m > j_n$ if $m > n$. Fix the values denoted by j_m for $1 \leqslant m \leqslant h+2$.

We wish to combine the first half of A_1 with the second half of A_2 by contracting a path in P. Let x be the endpoint of P_{j_1+h+1} that it shares with P_{j_1+h}, and let y be the endpoint P_{j_2+h+1} shares with P_{j_2+h}. Let P' be the subpath of P that has x and y as its endpoints. Let w be the vertex that gets created when contracting the path P' in G to get $G^{P'}$ and let all the subpaths P_i of P in G that were not contained in P' retain their labels in $G^{P'}$, so P_{j_1+h} and P_{j_2+h+1} share w as an endpoint, and let the a_i's retain their same meaning as long as P_i was not a subpath of P'. It follows that $G^{P'}$ has h sequences for $3 \leqslant i \leqslant h+2$, $A_i = a_{j_i}, a_{j_i+1}, \ldots, a_{j_i+2h+1}$ where for $1 \leqslant m \leqslant h+2$ and c, $0 \leqslant c \leqslant 2h+1$, $a_{j_m+c} = s_c$ and no part of A_m overlaps with some other A_n (so $|j_n - j_m| \geqslant 2h+2$) and $j_m > j_n$ if $m > n$. Furthermore, A_1 and A_2 have now been combined to give

$A' = a_{j_1}, a_{j_1+1}, \ldots, a_{j_1+h}, a_{j_2+h+1}, a_{j_2+h+2}, \ldots, a_{j_2+2h+1}$ so that $a_{j_1+c} = s_c$ for $0 \leqslant c \leqslant h$ and $a_{j_2+c} = s_c$ for $h+1 \leqslant c \leqslant 2h+1$. We will show that if there exists a set $X \subset V(G^{P'})$ that induces H in $G^{P'}$ then we can require $w \notin X$. The result then follows since if $w \notin X$ then the vertices that correspond to X in G induced an H in G.

So, assume $X \subset V(G^{P'})$ and induces H. If $w \notin X$ then we are done, so assume $w \in X'$ for some connected component X' of X. For i with $3 \leqslant i \leqslant h+1$, let P_i^* denote the path induced by $V(P_{j_i}), V(P_{j_i+1}), \ldots, V(P_{j_i+2h+1})$ in $G^{P'}$, so P_i^* is the path that naturally corresponds to S_i, and let P_1^* denote the path induced by

$$V(P_{j_1}), V(P_{j_1+1}), \ldots, V(P_{j_1+h}), V(P_{j_2+h+1}), V(P_{j_2+h+2}), \ldots, V(P_{j_2+2h+1}),$$

so P_1^* naturally corresponds with A'. Then since X' has at most h vertices there is at least one P_i^* that contains no vertex of X and since X' is connected and contains w, all vertices of $X' \cap P$ must be completely contained in $V(P_1^*)$ since w is at least distance h from either endpoint of P_1^*. It follows that we can replace the vertices of $X' \cap P$, which must be completely contained in the interal vertices of P_1^*, with the corresponding vertices in a P_i^* that contains no vertices of X and still maintain that the vertices of X induce H. Now $w \notin X$ and the result then follows. ■

Lemma 6.6.13. *Let $\mathcal{F}$ be a family of graphs determined by a finite number of forbidden induced subgraphs. Then if $\mathcal{F}$ does not forbid all k-thetas, k-prisms, k-pyramids, k-ladder-thetas, k-ladder-prisms, and k-ladders for arbitrarily large k, then $\mathcal{F}$ is feral.*

Proof: Let $\mathcal{F}$ be a family of graphs determined by a finite number of forbidden induced subgraphs, and let $\mathcal{H}$ be a set of forbidden subgraphs that define $\mathcal{F}$. Let let $h > 5$ be a number such that for any $H \in \mathcal{H}$, $|V(H)| \leqslant h$. First assume that $\mathcal{F}$ allows for either k-thetas k-prisms, or k-pyramids for arbitrarily large k. Then by Lemma 6.6.11 we can ensure that all paths with internal vertices all having degree 2 of the k-thetas

k-prisms, or k-pyramids are at most $5h$ (we keep on contracting the appropriate edges given by Lemma 6.6.11 until no path where all internal vertices have degree 2 have length more than $5h$) and therefore $\mathcal{F}$ contains a k-theta k-prism, or k-pyramid with at most $5h \cdot k$ vertices. Since a k-theta, k-prism, or k-pyramid must have at least 2^k minimal separators, it follows that there exists a $c > 1$ such that for every natural number N ther exists a $G \in \mathcal{F}$ such that $|V(G)| = n > N$ and the number of minimal separators in G is at least c^n.

Now assume that $\mathcal{F}$ allows for k-ladder-thetas or k-ladder-prisms for arbitrarily large k. Every k-ladder-theta and k-ladder-prism contains a k-half-ladder and by Lemma 6.6.11 we can ensure that all paths with internal vertices all having degree 2 of the k-ladder-theta or k-ladder-prism are at most $5h$ and by Lemma 6.6.12 we can ensure that the backbone path of the corresponding k-half-ladder has length at most $[5h(h+1)(5h)^{2h+1}+1] \cdot k$ by contracting the appropriate edges and paths if necessary while still guaranteeing the resulting graph belongs to $\mathcal{F}$ (Lemma 6.6.12 gives us that if there is a subpath of length over $[5h(h+1)(5h)^{2h+1}+1]$ of the backbone path that only has one neighbor outside of the backbone path, there there exists a subpath of the backbone path that we can contract and still maintain that the resulting graph is a k-ladder-theta or k-ladder-prism contained in $\mathcal{F}$). Since k-ladder-thetas and k-ladder-prisms have at least 2^k minimal separators it follows that there exists a contains $c > 1$ such that for every natural number N there exists a $G \in \mathcal{F}$ such that the number of minimal separators in G is at least c^n. It follows that $\mathcal{F}$ is feral. ■

The following lemma shows why it is necessary to forbid k-paw and k-claw graphs for a family of graphs defined by a finite number of forbidden induced subgraphs to be strongly-quasi-tame. Figure 6.10 gives a picture of the two graphs constructed in the following lemma.

Lemma 6.6.14. *Let $\mathcal{F}$ be a family of graphs determined by a finite number of forbidden induced subgraphs. Then if $\mathcal{F}$ does not forbid k-claws and k-paws for some natural number k, then $\mathcal{F}$ is feral.*

Proof: Let $\mathcal{F}$ be a family of graphs determined by a finite number of forbidden induced subgraphs, and let $\mathcal{H}$ be a set of forbidden subgraphs that define $\mathcal{F}$. Let $h > 5$ be a number such that for any $H \in \mathcal{H}$, $|V(H)| \leqslant h$. First we assume that $\mathcal{F}$ allows k-claw for arbitrarily large k. We will construct a graph with many minimal separators. Assume that we have two sets of $2^c - 1$ long-claws, $C_1^1, C_2^1, \ldots C_{2^c}^1$, and $C_1^2, C_2^2, \ldots C_{2^c}^2$ where in both sets each long claw has arm length h. We label the leaves of C_i^1 as a_i^1, b_i^1, c_i^1 and we label the endpoints of C_i^2 as a_i^2, b_i^2, c_i^2. Then for $1 \leqslant i \leqslant 2^{c-1} - 1$ we glue a_{2i}^1 to b_i^1, a_{2i+1}^1 to c_i^1, a_{2i}^2 to b_i^2, and a_{2i+1}^2 to c_i^2. Furthermore, for $2^{c-1} \leqslant i \leqslant 2^c - 1$ we add an edge between b_i^1 and b_i^2 and between c_i^1 and c_i^2. Note that any collection of $b_i^{j_i}$ and $c_i^{\ell_i}$ with $2^{c-1} \leqslant i \leqslant 2^c - 1$ and $j_i, \ell_i = 1$ or 2 is a minimal separator, so there are at least 2^{2^c} minimal separators in this construction. Since the arm length of each long-claw is h, the total number of vertices in this construction is less than $3h \cdot 2^{c+1}$.

If $\mathcal{F}$ allows for k-claws, then forest of paths and subdivided claws cannot be forbidden in $\mathcal{F}$, and it can be seen that any induced subgraph of size at most h of the construction just given is a forest of paths and subdivided claws (i.e., three anti-complete paths where one endpoint of each path are glued together). It follows that this construction must belong to $\mathcal{F}$ and since this construction has at least 2^{2^c} minimal separators and less than $3h \cdot 2^{c+1}$ vertices, the statement of the lemma follows for the case where k-claw graphs for arbitrarily large k are not forbidden.

Now we assume that $\mathcal{F}$ allows k-paw graphs for arbitrarily large k. The construction and analysis we make in this case is nearly identical to the k-claw case. We present it here for completeness. Assume that we have two set of $2^c - 1$ long-paws, $C_1^1, C_2^1, \ldots C_{2^c}^1$, and

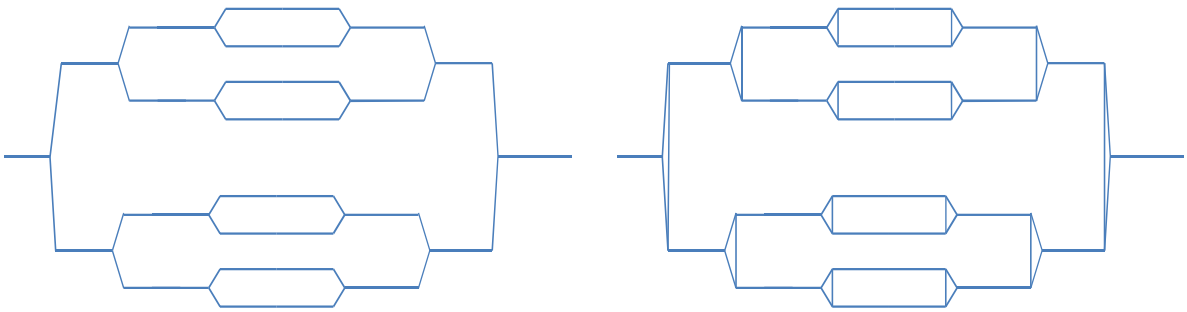


Figure 6.10: The two graphs in this figure are small versions of the constructions of the graphs given in Lemma 6.6.14, explicit vertices are omitted in this graph. The left side graph is the construction provided when when the k-claw is not forbidden for arbitrarily large k. The right hand side graph is the construction provided when when the k-paw is not forbidden for arbitrarily large k.

$C_1^2, C_2^2, \ldots C_{2^c}^2$ where in both sets each long-paw has arm length h. We label the endpoints of C_i^1 as a_i^1, b_i^1, c_i^1 and we label the endpoints of C_i^2 as a_i^2, b_i^2, c_i^2. Then for $1 \leqslant i \leqslant 2^{c-1} - 1$ we glue a_{2i}^1 to b_i^1, a_{2i+1}^1 to c_i^1, a_{2i}^2 to b_i^2, and a_{2i+1}^2 to c_i^2. Lastly, for $2^{c-1} \leqslant i \leqslant 2^c - 1$ we add an edge between a_i^1 and a_i^2 and between b_i^1 and b_i^2. Note that any collection of $b_i^{j_i}$ and $c_i^{\ell_i}$ with $2^{c-1} \leqslant i \leqslant 2^c - 1$ and $j_i, \ell_i = 1$ or 2 is a minimal separator, so there are at least 2^{2^c} minimal separators in this construction. Since the arm length of each long-claw is h, the total number of vertices in this construction is less than $3h \cdot 2^{c+1}$.

Since $\mathcal{F}$ allows for k-paws, a forest of paths and subdivided paws cannot be forbidden in $\mathcal{F}$, and it can be seen that any induced subgraph of size at most h of the construction just given is a forest of paths and subdivided paws. It follows that this construction must belong to $\mathcal{F}$ and since this construction has at least 2^{2^c} minimal separators and less than $3h \cdot 2^{c+1}$ vertices, the statement of the lemma follows for the case where k-paw graphs for arbitrarily large k are not forbidden. ■

We are now ready to prove Theorem 6.1.4

Proof: [Proof of Theorem 6.1.4] Let $\mathcal{F}$ be a family of graphs defined by a finite number of forbidden induced subgraphs. It follows from Lemmas 6.6.13 and 6.6.14 that

if $\mathcal{F}$ allows for any k-thetas, k-prisms, k-pyramids, k-ladder-thetas, k-ladder-prisms, k-claws, or k-paws for arbitrarily large k, $\mathcal{F}$ is feral.

Now assume that there exists a natural number k such that $\mathcal{F}$ forbids k-thetas, k-prisms, k-pyramids, k-ladder-thetas, k-ladder-prisms, k-claws, and k-paws. Observe that there exists a k' large enough such that if G contains an induced k'-ladder, then G contains an induced k-claw or k-paw graph, therefore $\mathcal{F}$ forbids k'-ladders. It then follows from Lemma 6.6.10 there exists a k'' such that no $G \in \mathcal{F}$ can contain a k''-creature, where the minimum value of k'' is a function of k. Furthermore, it is clear that there exists a k''' large enough such that if G contains a k'''-skinny-ladder as an induced minor, then G contains a k-claw or a k-paw as an induced subgraph. Hence $\mathcal{F}$ forbids k'''-skinny-ladders as an induced minor. It then follows from Theorem 6.1.2 that there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for all $G \in \mathcal{F}$ the number of minimal separators of G is at most $n^{f(k)\log(n)}$. Hence $\mathcal{F}$ is strongly quasi-tame. ■

6.7 Long Cycle-free Graphs

Here we present a proof of Theorem 6.1.5 which is based on an easy application of Corollary 6.5.7. We will need the following lemma in order to apply Corollary 6.5.7.

Lemma 6.7.1. *Let G be a $C_{\geqslant r}$-free graph and assume G does not contain a k-creature. Then every minimal separator, S, can be dominated by $r \cdot k^2$ vertices of G not in S.*

Proof: Let G be a $C_{\geqslant r}$-free graph and assume G does not contain a k-creature. Assume for a contradiction that there exists a minimal separator, S, of G such that S cannot be dominated by $r \cdot k^2$ vertices in G and not in S. Let H be an S-full component of $G - S$, then by Lemma 6.5.8, S is dominated a subset of H that is the union of k^2 induced paths in H. It follows there must exists some induced path P in H such that

$S_P = N(P) \cap S$ cannot be dominated by r vertices in P. There then exists a subpath P' of P such that there are vertices $a, b \in S_P$ that have no neighbor in P', both component of $P - P'$ have vertices that are neighbors with a and/or b. It follows that we can extend the path P' to have endpoints x_a and x_b such that the only neighbors of a in P' is x_a and possible x_b and the only neighbors of b in P' is x_b and possibly x_a. If x_a and x_b are both neighbors with a then P' and a form a cycle of length r, and if x_a and x_b are both neighbors with b then P' and b form a cycle of length r so assume neither of these cases occur. If a and b are neighbors then P' a, b make a cycle of length more than r. Else, there is an induced path, T between a and b with all of its internal vertices contained in some S-full component other than H. It follows that P', and T makes a cycle of length more than r, a contradiction. ■

Proof: [Proof of Theorem 6.1.5] Let G be a $C_{\geqslant k}$-free graph that is k-theta, k-prism, and k-pyramid free. Since G is $C_{\geqslant k}$-free this implies that G is also k-ladder-theta, k-ladder-prism, and k-ladder free. Lemma 6.6.10 then implies that there exists a function $f : \mathbb{N} \to \mathbb{N}$ (independent of the choice of k or G) such that G is $f(k)$-creature-free. Lemma 6.7.1 gives that every minimal separator S of G can be dominated by $kf(k)^2$ vertices not in S. Hence, by Corollary 6.5.7 G has at most $|V(G)|^{(kf(k)^2)^2+2kf(k)^2}$ minimal separators. It follows that the family of graphs that are $C_{\geqslant k}$-free, k-theta, k-prism, and k-pyramid free is tame. ■

6.8 Graph With Bounded Clique Size

Here we present a proof of Theorems 6.1.6 and 6.1.7 which are based on an easy application of Corollary 6.5.7. We will need the following lemma in order to apply Corollary 6.5.7.

Lemma 6.8.1. *Let $k' = 4[(8k^2)^{k+1}]^7$. If G is k-creature-free, G does not contain a k-*

skinny-ladder as an induced minor, and no minimal separator of G contains a clique of size k, then every minimal separator S of G can be dominated by at most $(k')^{k+1}$ vertices of $G - S$.

Proof: Let k', k, and G be as in the statement of the lemma. Let G' be an induced subgraph of G and let S' be a minimal separator of G'. Then G' must be k-creature-free and k-ladder free, so it follow from Lemma 6.5.15 that S' can be dominated by k' vertices of $G' - S'$.

We will produce a set of $(k')^{k+1}$ vertices of $G - S$ that dominate S by considering the following recursive algorithm. The input to the algorithm is (G',S') where G' is a subgraph of G and S' is a minimal separator of G', and the algorithm returns a set of vertices which will be described shortly. The algorithm finds two vertex sets A and B such that $|A| + |B| \leqslant k'$, $A \subset V(G')$, $B \subset S'$, and $A \cup B$ dominate S' (such a set must exists by what was established in the previous paragraph). Let B' be a set of vertices in $G' - S'$ such that $|B'| \leqslant |B|$ and B' dominates B. For each $b \in B$ we recursively call the algorithm on $(G' - (S' - [S' \cap N(b)]), S' \cap N(b))$ (note that $S' \cap N(b)$ is a minimal separator of $G' - (S' - [S' \cap N(b)])$). Let X be the union of the sets returned by each recursive call. Then algorithm then returns $X \cup A \cup B'$.

If we initially call this algorithm on (G, S) for some minimal separator S of G, then it is clear that the set this algorithm returns is a subset of vertices of $G-S$ that dominate S. We can also see the depth of this recursive algorithm cannot go past k without producing a clique of size k in S since the minimal separator we recursively call this algorithm on is always dominated by the open neighborhood of some vertex v of S. So, the depth of the recursion tree is at most $k - 1$ and each node has at most k' children since $|B| \leqslant k'$. It follows that since each recursive call of the algorithm adds at most k' vertices to the set it returns, the size of the final returned set cannot exceed $k' \cdot k'^k$ ■

Proof: [Proof of Theorem 6.1.6] Let G be a graph that is k-creature-free and does not contain a k-skinny-ladder as an induced minor, and furthermore assume that no minimal separator of G has a clique of size k. By Lemma 6.8.1 there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that all minimal separators, S, of any graph that is k-creature-free, does not contain a k-skinny-ladder as an induced minor, and has no minimal separator that contains a clique of size k, can be dominated by $f(k)$ vertices outside of S. It then follows from Corollary 6.5.7 that G has at most $|V(G)|^{f(k)^2+2f(k)}$ minimal separators. Hence, the family of graphs that are k-creature-free, do not contain a k-skinny-ladder as an induced minor, and have no minimal separator has a clique of size k is tame. ■

Proof: [Proof of Theorem 6.1.7] Let $\mathcal{F}$ be a family of graphs defined by a finite number of forbidden induced subgraphs. Assume that $\mathcal{F}$ forbids the complete graph on k vertices for some natural number k. It follows from Lemmas 6.6.13 and 6.6.14 that if $\mathcal{F}$ allows for any k'-thetas, k'-ladder-thetas, k'-claws, or k'-paws for arbitrarily large k', then $\mathcal{F}$ is feral.

Now assume that for some integer k that $\mathcal{F}$ forbids k-thetas, k-ladder-thetas, k-claws, and k-claws. Since $\mathcal{F}$ forbids k-cliques as well, it follows that $\mathcal{F}$ forbids k-prisms, k-pyramids, and k-ladder-prisms. Observe that there exists a k' large enough such that if G contains an induced k'-ladder, then G contains an induced k-claw or k-paw, therefore G does not contain a k'-ladder. It follows from Lemma 6.6.10 there exists a k'' such that no $G \in \mathcal{F}$ can contain a k''-creature, where the minimum value of k'' is a function of k. Furthermore, it is clear that there exists a k''' large enough such that if G contains a k'''-skinny-ladder as an induced minor, then G contains a k-claw or a k-paw as an induced subgraph. Hence $\mathcal{F}$ forbids k'''-skinny-ladders as an induced minor. Now, if no graph of $\mathcal{F}$ contains a minimal separator with a clique of size k, then it follows by Lemma 6.8.1 there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that for all $G \in \mathcal{F}$ it holds that all minimal separators S of G can be bounded by $f(k)$ vertices in $G - S$. It then follows

from Corollary 6.5.7 that for all $G \in \mathcal{F}$ has at most $|V(G)|^{f(k)^2+2f(k)}$ minimal separators. Therefore $\mathcal{F}$ is tame. ■

6.9 Conclusion

In this Chapter we disproved a conjecture of Abrishami et al. [43] that for any natural number k, the family of graphs that exclude k-creatures is tame. On the other hand, we proved a weakened form of the conjecture, that every family of graphs that excludes k-creatures and also excludes k-skinny-ladders as induced minors is strongly-quasi-tame. This led to a complete classification of graph families defined by a finite number of forbidden induced subgraphs into strongly-quasi-tame and feral, substantially generalizing the main result of Milanič and Pivač [42]. The tools we develop on the way to prove our main results yield with some additional effort polynomial upper bounds instead of quasi-polynomial, proving tameness instead of strong quasi-tameness, for two interesting special cases. In particular we show that the conjecture of Abrishami et al. [43] is true for $C_{\geqslant r}$-free graphs for every integer r, as well as for K_r-free graphs excluding an r-skinny-ladder for every integer r. The first of these results generalizes work of Chudnovsky et al. [114], who proved that $C_{\geqslant 5}$-free, k-creature-free graphs are tame,

Although Theorems 6.1.2 and 6.1.4 provide a strongly-quasi-tame bound we have no examples of non-tame families that exclude k-creatures and k-skinny-ladders for some k. We conjecture that these classes of graphs are actually tame.

Conjecture 6.9.1. *For every natural number k, the family of graphs that are k-creature-free and do not contain a k-skinny-ladder as an induced minor is tame.*

Conjecture 6.9.1, if true, put together with the proof of Theorem 6.1.4 would lead to the following classification of hereditary families defined by a finite set of forbidden induced subgraphs.

Conjecture 6.9.2. *Let $\mathcal{F}$ be a graph family defined by a finite number of forbidden induced subgraphs. If there exists a natural number k such that $\mathcal{F}$ forbids all k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, k-claw, and k-paw graphs, then $\mathcal{F}$ is tame. Otherwise $\mathcal{F}$ is feral.*

We remark that Conjecture 6.9.1 implies Conjecture 6.9.2, but not the other way around. In particular Conjecture 6.9.2 might be easier to prove.

We have so far been unsuccessful in identifying other counterexamples to Conjecture 6.1.1 that look "substantially different" from the k-twisted ladders constructed in Section 6.4. For this reason it is tempting to conjecture that at least for induced minor closed classes, a "clean" classification of all classes into tame or feral is possible.

Conjecture 6.9.3. *Every induced-minor-closed class $\mathcal{F}$ is either tame or feral.*

Since removing vertices and contracting edges can not increase the number of minimal separators, Conjecture 6.9.3, would show (in an informal sense) that both the brittleness of the boundary between tame and non-tame hereditary classes, as well as the existence of non-tame hereditary classes that are not feral is primarily due to "number fiddling" effects such as in the example of Abrishami et al. [43] of a tame family containing k-creatures for arbitrarily large k.

Remark: As mentioned in the introduction, subsequent work([44] and Chapter 7 of this thesis), has confirmed that Conjectures 6.9.1 and 6.9.2 are true, while Conjecture 6.9.3 is false. We nevertheless keep the statements of these conjectures here, both because they provided guidance and motivation for the subsequent works.

Chapter 7

Graph Classes with Few Minimal Separators. II. A Dichotomy

A class $\mathcal{F}$ of graphs is called tame if every graph in $\mathcal{F}$ on n vertices contains at most $n^{O(1)}$ minimal separators, *quasi-tame* if every graph in $\mathcal{F}$ on n vertices contains at most $2^{\log^{O(1)}(n)}$ minimal separators, and *feral* if there exists a constant $c > 1$ so that F contains n-vertex graphs with at least c^n minimal separators for arbitrarily large n. The classification of graph classes into (quasi-) tame or feral has numerous algorithmic consequences, and has recently received considerable attention.

In this paper we precisely characterize the structure of graphs which have few minimal separators. Specifically we show that every graph which excludes certain graphs called *k-creatures* and *k-critters* as induced subgraphs has at most quasi-polynomially many minimal separators. We then demonstrate that this sufficient condition for having few minimal separators is the "right" one. In particular we show that every hereditary graph class $\mathcal{F}$ definable in CMSO logic that contains k-creatures or k-critters for every k is feral.

7.1 Introduction

Let G be a graph and u and v be distinct vertices in G. A vertex set S is a *u,v-separator* if u and v are in distinct components of $G - S$. The set S is a *u,v-minimal separator* if S is a u,v-separator, but no proper subset of S is a u,v-separator. Finally, S is a *minimal separator* if S is a u,v-minimal separator for some pair of vertices u and v. Building on the terminology of Milanič and Pivač [42], we will say that a graph class $\mathcal{F}$ is *tame* if there exists an integer c so that every graph in $\mathcal{F}$ on n vertices has at most $O(n^c)$ minimal separators.

Minimal separators are a fundamental combinatorial object that show up naturally both in structural arguments [75, 115], as well as in algorithmic applications [116]. Many graph problems including Treewidth, Minimum Fill In, Treelength, Independent Set, Feedback Vertex Set, and others [33, 7, 117] can be solved in time polynomial in the number n of vertices plus the number of minimal separators in the input graph. These algorithms run in time polynomial in n precisely for the graphs that have at most $n^{O(1)}$ minimal separators, motivating the question we address in this paper - *which graph classes are tame?*

Since this paper is the second in a series we will refrain from a more in-depth discussion on the importance of minimal separators, or the history of study of tame graph classes. Such a discussion may be found in the first paper in the series [5] We will simply mention that a substantial body of work has been devoted to identifying tame graph classes [43, 123, 114, 33, 75, 121, 120, 42, 122, 44].

All of the aforementioned previous work essentially gives different sufficient conditions for a graph to only have polynomially many separators. This naturally leads to the question *is there a "right" sufficient condition for tameness?* That is - a condition that on one hand is easy to state and verify, while on the other hand captures all interesting

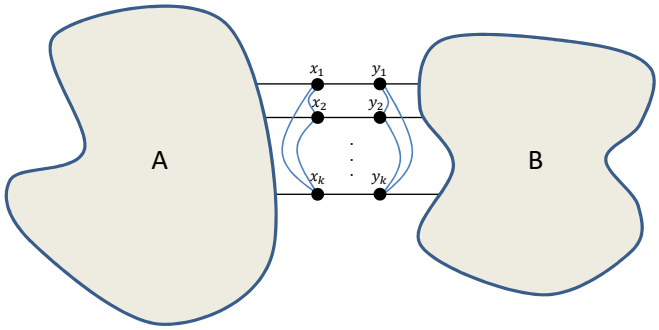


Figure 7.1: A graph induced by the vertices of a k-creature. The blue edges indicate that x_i (y_i) may or may not be a neighbor of x_j (y_j)

tame graph classes. One Theorem to tame them all, so to speak.

Abrishami et al. [43] conjectured that the presence or absence of *k-creatures* (more or less) completely dictates whether a graph has many or few minimal separators. To properly state their conjecture we first need to define k-creatures.

Definition 7.1.1 (k-creatures)**.** (see Figure 7.1) A graph G is said to be a *k-creature* if its vertices can be partitioned into sets A, $X = \{x_1, x_2, \ldots, x_k\}$, $Y = \{y_1, y_2, \ldots, y_k\}$, and B satisfying the following conditions:

(i) $G[A]$ and $G[B]$ are connected.

(ii) A and $Y \cup B$ are anti-complete (i.e. $N[A] \cap (Y \cup B) = \emptyset$) and B and $A \cup X$ are anti-complete.

(iii) A dominates X (every vertex in X has a neighbor in A) and B dominates Y.

(iv) $x_i y_j$ is an edge if and only if $i = j$.

When identifying a k-creature in a graph, we will typically denote it as a tuple of vertex sets (V_1, V_2, V_3, V_4) such that the graph $G[V_1 \cup V_2 \cup V_3 \cup V_4]$ is a k-creature with the set V_1 corresponding to A, V_2 corresponding to X, V_3 corresponding to Y, and V_4 corresponding to B.

We will say that a graph G is *k-creature free* if G does not contain a k-creature as an induced subgraph. It is quite easy to see that for a vertex $a \in A$ and $b \in B$ there are precisely 2^k minimal separators S which are disjoint from A and B. Such a separator S must pick precisely one vertex from each of $\{x_i, y_i\}$, and it can make each one of these k choices independently.

Abrishami et al. [43] conjectured that for every integer k there exists a k' such that if an n-vertex graph G does not contain any k-creature as an induced subgraph, then G has at most $n^{k'}$ minimal separators. It turns out, as we showed in the first paper in this series [5], that this conjecture is false: for arbitrarily large n there exist n-vertex graphs that exclude 100-creatures and yet have $2^{\Omega(n)}$ minimal separators. However, before discarding the conjecture of Abrishami et al. [43], let us discuss why it would have been the "right" sufficient condition for polynomially many minimal separators if it had been true.

Towards this we need to ask, which graph families $\mathcal{F}$ would have been tame, but whose tameness would not be captured by the conjecture? It would be precisely families $\mathcal{F}$ that are tame, but that contain for every k an n-vertex graph G that contains a k-creature as an induced subgraph. Since k-creatures have 2^k minimal separators, and $\mathcal{F}$ is tame it must hold that $2^k \leqslant n^{O(1)}$, meaning that $k = O(\log n)$. In other words the conjecture fails to capture tame graph classes that contain k-creatures in graphs whose number of vertices is at least exponential in k.

This could manifest itself in two different ways. One option, that we call Type 1, is that whenever a graph $G \in \mathcal{F}$ contains a k-creature then G also contains some different piece of size exponential in k which is completely unrelated to the k-creature. The other option, which we call Type 2, is that whenever a graph $G \in \mathcal{F}$ contains a k-creature, then this k-creature itself has size exponential in k. Families of either one of these two types would have to be rather strange, although it is perfectly possible to construct

artificial graph families of either type. For an example, most interesting graph families are *hereditary*, that is, closed under vertex deletion. A hereditary family $\mathcal{F}$ cannot possibly be Type 1. Indeed whenever $\mathcal{F}$ contains a graph G that contains a k-creature we can simply delete all the vertices not in the k-creature to obtain a k-creature which is in the family. Thus, the conjecture of Abrishami et al. [43] if true, would only fail to capture tameness of hereditary classes of graphs whose every k-creature has size exponential in k.

As previously mentioned, the conjecture of Abrishami et al. [43] is false. In [5] the authors gave a counterexample, and showed that a weaker version of the conjecture of Abrishami et al. is true. To state this result we need three definitions. First, a family $\mathcal{F}$ is *quasi-tame* if every n-vertex graph in the family has at most $2^{\log^{O(1)} n}$ minimal separators. Second, a *k-skinny ladder* is a graph G consisting of two anti-complete paths $P_l = \ell_1\ell_2 \dots \ell_k$ and $P_r = r_1r_2 \dots r_k$ and a set $\{s_1, s_2, \dots, s_k\}$ of vertices such that for every i, s_i is adjacent to ℓ_i and r_i and to no other vertices. Third, an *induced minor* of a graph G is a graph that can be obtained from G by deleting vertices and contracting edges. The main result of [5] is the following weakening of the conjecture of Abrishami et al.

Theorem 7.1.2 ([5])**.** *For every integer k, the family of k-creature free graphs that exclude k-skinny ladders as induced minors is quasi-tame.*

Theorem 7.1.2 fails to be "the one theorem to tame them all" in two different ways. The first is the quasi-polynomial upper bound on the number of minimal separators. The second is that Theorem 7.1.2 fails to capture the tameness of some perfectly reasonable hereditary graph classes. For an example, the class of all induced subgraphs of k-skinny ladders (for all $k \in \mathbb{N}$) can easily be checked to be tame, yet Theorem 7.1.2 fails to conclude anything at all about this family.

The first shortcoming of Theorem 7.1.2 was very recently rectified by Gajarsky et al. [44] who, building heavily upon the work of [5], showed a version of Theorem 7.1.2, but with the conclusion of quasi-tame replaced by tame. In this paper we rectify the second shortcoming. To properly state our main result we need to introduce another kind of graph, called *t-critters*. We first need a small generalization of minimal separators which applies to vertex sets instead of just vertices. Given $A, B \subset V(G)$ we define S to be an *A, B-separator* if $A \cap S = B \cap S = \emptyset$ and no component of $G - S$ contains a vertex from both A and B. An *A,B-minimal separator* is an A, B-separator such that no proper subset of S is an A, B-separator.

Definition 7.1.3 (t-critter partition, t-critter)**.** (see Figure 7.2) A *t-critter partition* of a graph G is a partition of the vertex set of G into sets $A_1, A_2, \ldots A_{t+1}$, $B_1, B_2, \ldots, B_{t+1}$, $X_1, X_2, \ldots, X_t$, such that the following conditions are satisfied.

(i) For all $1 \leqslant i, j \leqslant t+1$ with $i \neq j$, A_i is anti-complete with A_j, B_i is anti-complete with B_j, and A_i is anti-complete with B_j.

(ii) For all $1 \leqslant i \leqslant t+1$ A_i and B_i is connected.

(iii) The vertices of A_i, A_{i+1}, B_i, and B_{i+1} are the only vertices outside of X_i that have a neighbor in X_i.

(iv) There are (at least) two distinct $(A_i \cup A_{i+1})$, $(B_i \cup B_{i+1})$-minimal separators in $G[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i]$, S_1^i and S_2^i, such that there is a path from A_i to A_{i+1} through both $X_i - S_1^i$ and $X_i - S_2^i$ and there is a path from B_i to B_{i+1} through both $X_i - S_1^i$ and $X_i - S_2^i$.

A graph G is a *t-critter* if G has a t-critter partition. A graph G is *t-critter free* if G does not contain a t-critter as an induced subgraph.

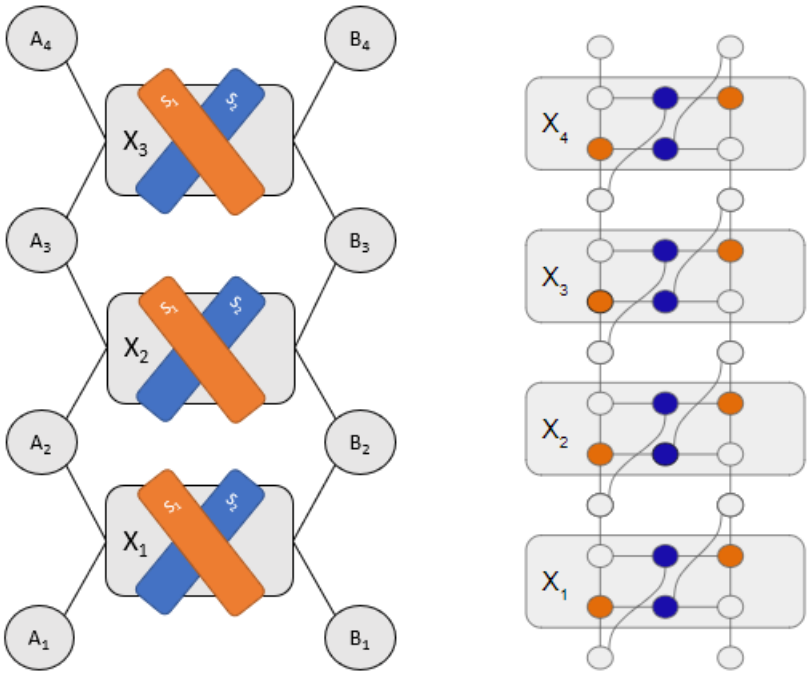


Figure 7.2: The left hand graph is a general visualization of a 3-critter, the orange blocks representing S_1^i and the blue blocks representing S_2^i. The right hand graph is specific instance of a 4-critter. The orange vertices represent the sets S_1^i and the blue vertices represent the sets S_2^i

The definition of t-critters is arguably technical and unappealing. After staring at the definition for a few minutes the reader should be able to convince themselves that, just as for k-creatures, for every vertex $a \in A = \bigcup_i A_i$ and vertex $b \in B = \bigcup_i B_i$ there are at least 2^t minimal a,b-separators in G disjoint from $A \cup B$. In particular, in order to separate a from b we may for every $i \leqslant t$ choose to delete either S_1^i or S_2^i, and for every i the choice between S_1^i or S_2^i can be made independently of the other choices (see Lemma 7.6.13). We are now ready to state our main theorem.

Theorem 7.1.4. *For every pair of integers t, k the family of k-creature free and t-critter free graphs is quasi-tame.*

The upper bound in Theorem 7.1.4 on the number of minimal separators is $n^{k' \log^{17} n}$ where k' is a constant that depends only on k and t. The proof of Theorem 7.1.4 contains some interesting ingredients, from the VC-dimension-based lemma of [5], to a "greedy branching" procedure inspired by the recent quasi-polynomial time algorithm for Independent Set on P_k-free graphs [1], to covering-packing dualities [52] and Ramsey- and Erdös-Szekers [53] type arguments.

Theorem 7.1.4 is yet another sufficient condition for a graph to have few minimal separators. To boot, the condition is very technical and the upper bound in the number of minimal separators is an ugly quasi-polynomial function. What makes *this* sufficient condition for an upper bound for the number of minimal separators special? What makes it special is that it is the *right* sufficient condition, in the way that the conjecture of Abrishami et al. [43] would have been right if only it had been correct. But don't take our word for it - we actually prove this in a precise and technical sense.

Let us apply the same litmus test for Theorem 7.1.4 as we did for the conjecture of Abrishami et al. [43], and ask for which tame graph families $\mathcal{F}$ are not captured by Theorem 7.1.4? Again there could be families of Type 1, namely families that do contain

graphs G that contain k-critters or k-creatures for every k, but such graphs G always also contain at least $2^{\Omega(k)}$ additional unrelated vertices. Such families cannot be hereditary, and so, if we restrict attention to hereditary families the only tame families that are not captured by Theorem 7.1.4 are Type 2 families, that do contain k-critters or k-creatures for every k, but these k-critters or k-creatures have at least $2^{\Omega(k)}$ vertices.

It is in fact possible to construct such hereditary families. It is even possible to construct tame families that contain k-critters for every k, and yet are closed under induced minors, disproving Conjecture 4 of the authors [5] in the process[1] However all such families are pretty artificial. The next theorem shows that they *have* to be artificial.

Monadic Second Order Logic (MSO) and their extension, Counting Monadic Second Order Logic (CMSO), (see Section 7.6 for a definition) can be viewed as formal languages to express families of graphs. In graph algorithms their main claim to fame probably comes from from Courcelle's Theorem [54], which states that every MSO-definable family of graphs can be recognized in linear time on graphs of bounded treewidth. The overwhelming majority of interesting graph families can be expressed in Counting Monadic Second Order Logic, this includes all graph classes with a finite number of forbidden minors, induced minors, topological minors, induced subgraphs or subgraphs, and a number of other classes such as bipartite, or perfect. We show that if we restrict attention to CMSO-definable hereditary properties then the sufficient condition of Theorem 7.1.4 is also necessary.

Theorem 7.1.5. *Let $\mathcal{F}$ be a CMSO-definable hereditary graph family. If there exists an integer k such that $\mathcal{F}$ neither contains a k-creature nor a k-critter then $\mathcal{F}$ is quasi-tame. Otherwise $\mathcal{F}$ is feral.*

The proof of Courcelle's theorem [54] establishes that CMSO-definable graph classes

[1]We do not give such a construction in this paper, as it is long enough as it is.

have many properties in common with regular languages. This has been exploited with great success in graph algorithms [45, 55], however, to the best of our knowledge, it has never been used to prove a purely structural result such as Theorem 7.1.5

The proof of Theorem 7.1.5 is based on a "pumping lemma" style argument that shows that a k-creature or k-critter on n vertices can be"pumped" to a $k \cdot x$-critter on $n \cdot x$ vertices, thereby demonstrating that in every CMSO-definable hereditary family that contains k-creatures or k-critters for arbitrarily large k, there exist k-creatures or k-critters in the family with only $O(k)$ vertices.

Paper Organization. In Section 7.2 we review basic definitions and notations. In Section 7.3 we give an overview of our proofs of Theorems 7.1.4 and 7.1.5.

The proof of Theorem 7.1.4 cleanly breaks into two parts, where the output of the first part is then fed into the second part. The first part of the proof is given in Section 7.4, while the second part of the proof is given in Section 7.5. Finally we prove Theoerm 7.1.5 in Section 7.6. It is worth noting that while the proof of Theorem 7.1.4 is a bit of a monstrosity, the proof of Theorem 7.1.5 is relatively short and slick. If the reader is willing to assume the statement of Theorem 7.1.4 on face value, the proof of Theorem 7.1.5 may be read independently of the proof of Theorem 7.1.4.

7.2 Preliminaries

Unless otherwise stated, graphs in this paper are assumed to be simple, undirected graphs. We denote the edge set of a graph G by $E(G)$ and the vertex set of a graph by $V(G)$. If $v \in V(G)$, then we use $N_G[v]$ to denote the closed neighborhood of v in the graph G, i.e. the set of all neighbors v has in G together with v itself. We use $N_G(v)$ to denote the set $N_G[v] - \{v\}$. If $X \subseteq V(G)$, then $N_G[X] = \bigcup_{x \in X} N_G[x]$ and $N_G(X) =$

$N_G[X] - X$. When the graph G is clear from the context, we will use $N[v]$, $N(v)$, $N[X]$, and $N(X)$. If $X \subseteq V(G)$, then we use $G[X]$ to denote the induced subgraph of G with vertex set X and $G - X$ denotes $G[V(G) - X]$. Additionally, for a natural number i, we inductively define $N^i_G[X]$ to equal $N_G[X]$ if $i = 1$ and $N_G[N^{i-1}_G[X]]$ for $i > 1$. Given a graph G and disjoint sets $X, Y \subseteq V(G)$ we define the *distance* between X and Y to be the lowest integer i such that $N^i_G[X] \cap Y \neq \emptyset$.

Given a graph G, a non-empty set $S \subset V(G)$ is called a *separator* if there are at least two distinct components L and R of $G - S$. If $u \in L$ and $v \in R$ then we call S a *u-v-separator* or a *u, v-separator*. S is a *u, v-minimal separator* if S is a u, v-separator and no proper subset of S is a u, v-separator, or equivalently, if $N_G(L) = N_G(R) = S$. This equivalence is folkloric and easy to show. If C is a component of $G - S$ such that $N_G(C) = S$, then we say that C is an *S-full component*. Similarly, given $A, B \subset V(G)$ we define S to be an *A, B-separator* if $A \cap S = B \cap S = \emptyset$ and no component of $G - S$ contains a vertex from both A and B. An *A,B-minimal separator* is an A, B-separator such that no proper subset of S is an A, B-separator.

Let G be a graph. A *vertex list* (or simply a *list*), $\mathcal{S}$, is an ordered tuple of vertex sets of G, that is, $\mathcal{S}$ is a collection of vertex sets which allow multiple instances of its vertex sets and gives an index to each element it contains. Let $\mathcal{S} = \{S_1, S_2, \ldots S_m\}$ be a vertex list and let S be a vertex set. We define $\mathcal{S} \cup S$ to be the list $\{S_1, S_2, \ldots S_m, S_{m+1}\}$ where $S_{m+1} = S$. Given a vertex list $\mathcal{S}$ we define $N_G[\mathcal{S}]$ to be the vertex list where each element $S \in \mathcal{S}$ is replaced with $N_G[S]$. Lastly, for a set $A \subseteq V(G)$, we define $\mathcal{S} - A$ to be the vertex list $\{S_1 - A, S_2 - A, \ldots S_m - A\}$

Given a graph G with n vertices, a set $S \subseteq V(G)$ is called a *δ-balanced separator* if no component of $G - S$ contains over δ vertices.

Given a graph G, we say a vertex set $C \subseteq V(G)$ is a *connected vertex set* if $G[C]$ is a connected graph. A *walk* in a graph G is a sequence $v_1, v_2, \ldots, v_\ell$ of vertices in G such

that each pair of consecutive vertices in the sequence are adjacent. The *length* of a walk $v_1, v_2, \ldots, v_\ell$ is the number ℓ of *vertices* in the walk. A walk whose first vertex is v_1 and last is v_ℓ is a walk *from* v_1 *to* v_ℓ. The vertex v_1 is called the first vertex of the walk, v_ℓ the last. All other vertices are *internal* vertices. A walk $v_1, v_2, \ldots, v_\ell$ where all vertices are distinct is a *path*. A path $P = v_1, v_2, \ldots, v_\ell$ is an *induced* path if there are no edges between v_i and v_j whenever $|i - j| > 1$. For three disjoint vertex sets A, B, C a walk (or path, or induced path) *from* A *to* B *through* C is a walk (or path, or induced path) whose first vertex is in A, last vertex is in B, and all internal vertices (if any) are in C.

We define contracting an edge. Let G be a graph. For an edge $uv \in E(G)$ we define the *contraction of* uv to result in a new graph G' with vertex set $V(G') = (V(G) - \{u, v\}) \cup \{w\}$ and there is an edge between two vertices $x, y \in V(G') - \{w\}$ in G' if there is an edge between x and y in G, and there is an edge between $x \in V(G') - \{w\}$ and w in G' if there is an edge x and u in G or an edge between x and v in G. Given a connected vertex set $A \subset V(G)$ we define the *contraction of* A to be the graph that results from contracting all edges between every pair of vertices of A. It can easily be seen that the contractions may be performed in any order without changing the final result.

Given a graph G we define a function $\mathcal{CC}(G)$ that returns a set which contains the vertex sets of all connected components of G.

Given a graph G we say that two vertex sets $A, B \subseteq V(G)$ are *anti-complete* if A and B are disjoint and there is no edge in G between any vertex of A and any vertex of B. Let $X, Y \subseteq V(G)$ be disjoint sets. We define a *semi-induced matching* in G of size m between X and Y to be two subsets $X' \subseteq X, Y' \subseteq Y$ with $X' = \{x'_1, x'_2, \ldots, x'_m\}$ and $Y' = \{y'_1, y'_2, \ldots, y'_m\}$ where x'_i is neighbors with y'_j in G if and only if $i = j$.

Let A and B be sets. The Cartesian product of A and B, denoted as $A \times B$, is the set with consists of all ordered pairs of the form (a, b) where $a \in A$ and $b \in B$.

7.3 Overview

7.3.1 Generalized ω-Creatures

In order to prove Theorem 7.1.4 we will show that a graph that is k-creature free but still has a large number of minimal separators must have a t-critter. t-critters are highly structured objects and thus it is difficult to extract such objects from a graph directly, so in order to find a t-critter in a k-creature free graph with many minimal separators, we first construct an intermediate structure that we call a *generalized ω-creature* which we will define shortly. Generalized ω-creatures are less structured than t-critters and are thus easier to find in a graph, but they share some essential features with t-critters. In fact, the last main component of our proof is that for any natural numbers t, every generalized ω-creature, for ω large enough, contains a t-critter.

It is not necessary to have a full understanding of the definition of generalized ω-creatures until the end Section 7.4.2. We nevertheless give the definition here as it can help put the prior work of Section 7.4 into context which prepares the ground for extracting a generalized ω-creature from a k-creature free graph with many minimal separators.

Bistars, Bistar Partitions, and Generalized ω-Creatures. We define an *ω-bistar*, H, to be a graph with two *central vertices*, denoted by c_A and c_B, and ω independent vertices, called the *peripheral vertices*, that are neighbors with c_A and c_B (so H is a complete bipartite graph with 2 vertices on one side and ω on the other).

Let G be a graph. We define an *ω-bistar partition* of G to be an ω'-bistar graph H, with $\omega' \geqslant \omega$, along with a function φ from $V(G)$ to $V(H)$ such that for every edge $uv \in V(G)$ either $\varphi(u) = \varphi(v)$ or one of $\varphi(u)$ and $\varphi(v)$ is either c_A or c_B and the other is a peripheral vertex. We will use A_φ to denote the vertices of G in $\varphi^{-1}(c_A)$ and B_φ to

denote the vertices of G in $\varphi^{-1}(c_B)$.

We now give the definition for generalized ω-creatures.

Definition 7.3.1 (Generalized ω-Creature)**.** A *generalized ω-creature* is a tuple $W = (G, H, \varphi, S_1, S_2)$ where G is a graph, (H, φ) is an ω-bistar partition for G, and $S_1, S_2 \subseteq V(G)$ with the following properties:

(i) There exists $S_1^\star \subseteq S_1$ and $S_2^\star \subseteq S_2$ such that for every peripheral vertex, u, of H, $\varphi^{-1}(u) \cap S_1^\star$ and $\varphi^{-1}(u) \cap S_2^\star$ are distinct A_φ, B_φ-minimal separators in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)]$.

(ii) A_φ is entirely contained in one component of $G - S_1$ and B_φ is entirely contained in a different component of $G - S_1$ and similarly A_φ is entirely contained in one component of $G - S_2$ and B_φ is entirely contained in a different component of $G - S_2$.

(iii) For every peripheral vertex u of H and all pair of components C_1 and C_2 of $G[A_\varphi \cup B_\varphi]$ there is a path from C_1 to C_2 through $\varphi^{-1}(u) - S_1$ in G if and only if there is a path from C_1 to C_2 through $\varphi^{-1}(u) - S_2$ in G.

(iv) For all peripheral vertices, u, of H, if a component, X_A, of $G[A_\varphi]$ has a neighbor in $\varphi^{-1}(u)$ then there is at least one component, X_B, of $G[B_\varphi]$ such that X_A has a path through $\varphi^{-1}(u)$ to X_B in G. Similarly, if a component, X_B, of $G[B_\varphi]$ has a neighbor in $\varphi^{-1}(u)$ then there is at least one component X_A of $G[A_\varphi]$ such that X_B has a path through $\varphi^{-1}(u)$ to X_A in G.

Note that by property *(i)* that for every $u \in H$, $\varphi^{-1}(u)$ must be non-empty. Sets $S_1^\star \subseteq S_1$ and $S_2^\star \subseteq S_1$ satisfying property *(i)* of W are called *witness separators* for W.

For a generalized ω-creature $W = (G, H, \varphi, S_1, S_2)$ we define the sets $A_1(W)$, $A_2(W)$, $B_1(W)$ and $B_2(W)$ as the component of $G - S_1$ that contains A_φ, the component of $G - S_2$ that contains A_φ, the component of $G - S_1$ that contains B_φ, and the component of $G - S_2$ that contains B_φ respectively. The vertex sets $A_1^\star(W)$, $A_2^\star(W)$, $B_1^\star(W)$ and $B_2^\star(W)$ are defined analogously, but with $S_1^\star$ and $S_2^\star$ in place of S_1 and S_2

Just like how for each X_i of a t-critter we can independently pick either S_i^1 or S_i^2 to build a minimal separator for the t-critter (giving 2^t minimal separators), it can be shown that conditions *(i)-(iii)* together allow us, for each peripheral vertex $u \in H$, to independently choose the vertex set $S_1 \cap \varphi^{-1}(u)$ or $S_2 \cap \varphi^{-1}(u)$ in order to build a minimal separator, S, for the graph G of the generalized ω-creature $W = (G, H, \varphi, S_1, S_2)$, giving 2^ω such minimal separators. In such a minimal separator, the vertex sets A_φ will be in one S-full component of $G - S$ and B_φ will be in a different S-full component. We will prove this later on in the paper.

7.3.2 Proof of Theorem 7.1.4

The following two lemmas show how generalized ω-creatures help us to prove Theorem 7.1.4. Proving the first lemma is the main goal of Section 7.4 and proving the second lemma is the main goal of Section 7.5. We will actually need a generalized ω-creature with a some additional structure, in particular we will need the generalized ω-creature to be a connected, good, full generalized ω-creature. We will introduce these definitions later in the paper and the reader does not need to concern themselves with them at the moment.

Lemma 7.3.2. *Let G be a k-creature free graph with n vertices, let $\omega > 1$ and $\delta = 3\omega$, let c be an integer large enough so that $400k^3\delta^2 \log^4(c) < c/6$, let $x = 400k^3\delta^2 \log^4(n)$, and let G have at least $2^c(12n)^{6k^2x^4 \log(n)}$ minimal separators. Then there exists an induced*

minor G' of G, an ω-bistar partition H and φ of G', and sets $S_1, S_2 \subseteq V(G')$ such that $(G', H, \varphi, S_1, S_2)$ is a connected, good, full generalized ω-creature.

Lemma 7.3.3. *Let $k \geqslant 2$ and G be a k-creature free graph, and $W = (G, H, \varphi, S_1, S_2)$ be a connected, good, full generalized ω-creature. Then there exists an induced subgraph G' of G which is a t-critter for $t \geqslant \frac{(\log\log(\omega))^{1/4k}}{96k^4} - 4$.*

This paper natural breaks up into two parts, first part where we prove Lemma 7.3.2 and the second where we prove Lemma 7.3.3. We will provide an outline of how we prove Lemmas 7.3.2 and 7.3.3 later on in this section, right now we give a proof of Theorem 7.1.4 using Lemmas 7.3.2 and 7.3.3. Additionally, since Lemma 7.3.2 deals with an induced minor G' of G we need to show that k-creature free graphs and t-critter free graphs are closed under taking induced minors. We state the lemmas here and prove them shortly.

Lemma 7.3.4. *Let G be graph and let G' be an induced minor of G. If G is k-creature free then G' is k-creature free.*

Lemma 7.3.5. *Let G be a graph and let G' be an induced minor of G. If G is t-critter free then G' is t-critter free.*

Proof: [proof of Theorem 7.1.4] Let G be a k-creature free graph with n vertices, let ω be large enough to satisfy the inequality $t \leqslant \frac{(\log\log(\omega))^{1/4k}}{96k^4} - 4$ (note that the size of ω only depends on t and k), let $\delta = 3\omega$, let c be large enough to satisfy the inequality $400k^3\delta^2\log^4(c) < c/6$, and let $x = 400k^3\delta^2\log^4(n)$, so $x = k'\log^4(n)$ where k' only depends on k and t. We will show that if G has at least $2^c(12n)^{6k^2x^4\log(n)} = n^{O(\log(n)^{17})}$ minimal separators then G must contain a t-critter. It follows from Lemma 7.3.2 that there is an induced minor G' of G, an ω-bistar partition H and φ of G', and sets $S_1, S_2 \subseteq V(G')$ such that $W = (G', H, \varphi, S_1, S_2)$ is a connected, good, full generalized ω-creature. By Lemma 7.3.4 G' is k-creature free. Then by Lemma 7.3.3 it follows that G' contains a

t'-critter for $t' \geqslant \frac{(\log\log(\omega))^{1/4k}}{96k^4} - 4$. By how ω was chosen we have $t' \geqslant t$, so G' contains a t-critter. Hence, by (the contra-positive of) Lemma 7.3.5, G contains a t-critter. ■

We now give proofs for Lemmas 7.3.4 and 7.3.5. They are fairly straightforward and just involve checking a few cases.

Proof: [proof of Lemma 7.3.4] Let G be a k-creature free graph. We first show that the deletion of a single vertex or the contraction of a single edge leaves us with a graph that is also k-creature free. Let $v \in G$. If $G - v$ contains a k-creature, (A, X, Y, B), then clearly G contains the same k-creature, (A, X, Y, B), hence $G - v$ is k-creature free. Now let uv be an edge of G, let G' be the graph that results from contracting uv, and let w denote the vertex that is creature from this contraction, so $V(G') = (V(G) - \{u, v\}) \cup \{w\}$. Assume for a contradiction that G' contains a k-creature, (A, X, Y, B). We will break the proof into cases when w either belongs to A, X, Y, B or none of these sets. By the symmetry of k-creatures, the cases where $w \in A$ and $w \in B$ are identical and the cases where $w \in X$ and $w \in Y$ are identical, so we will only prove the cases for $w \in A$, $w \in X$ and $w \notin A \cup X \cup Y \cup B$.

If $w \notin A, X, Y, B$, then we can see that (A, X, Y, B) is a k-creature in G, a contradiction. If $w \in A$, then we can see that $((A - \{w\}) \cup \{u, v\}, X, Y, B)$ is a k-creature in G, a contradiction. Lastly, if $w \in X$, say $w = x_i$, then, in G, at least one of u, v is neighbors with y_i and at least one of u, v has a neighbor in A. If one of these two vertices, say u, is both neighbors with y_i and has a neighbor in A in G, then $(A, (X - \{w\}) \cup \{u\}, Y, B)$ is a k-creature in G. So we may assume that in G exactly one of u, v is neighbors with y_i, say u, and does not have a neighbor in A, and that the other one, v, has a neighbor in A (but is not neighbors with y_i). It then follows that $(A \cup \{v\}, (X - \{w\}) \cup \{u\}, Y, B)$ is a k-creature in G.

A straightforward application of induction then shows that if G'' is any induced minor of G then since G is k-creature free, so is G''. ■

Proof: [proof of Lemma 7.3.5] Let G be a graph that is t-critter free. We first show that the deletion of a single vertex or the contraction of a single edge leaves us with a graph that is also t-critter free. Let $v \in G$. If $G - v$ contains a t-critter, then clearly G contains the same t-critter, hence $G - v$ is t-critter free. Now, let uv be an edge in G, let G' be the graph that results in from contracting uv, and let w denote the vertex that is creature from this contraction, so $V(G') = (V(G) - \{u, v\}) \cup \{w\}$. Assume for a contradiction that G' contains a t-critter, T, and let $A_1, A_2, \ldots A_{t+1}$, $B_1, B_2, \ldots B_{t+1}, X_1, X_2, \ldots, X_t$ be the partitioning of the vertices of T given in Definition 7.1.3. We will break the proof into cases when w either belongs to one of the A_i's, one of the B_i's, one of the X_i's, or none of these sets. By the symmetry of k-critters, the cases when $w \in A_i$ for some i and when $w \in B_i$ for some i are identical, so we will only prove the cases where $w \in A_i$ for some i, $w \in X_i$ for some i, and $w \notin T$.

If $w \notin T$ then T is an induced subgraph of G and hence G contains a t-critter as an induced subgraph, a contradiction. If $w \in X_i$ for some i, then let T' be the induced subgraph of G with vertex set $(T - \{w\}) \cup \{u, v\}$. It is straightforward to verify that the partitioning $A_1, A_2, \ldots, A_{t+1}$, $B_1, B_2, \ldots, B_{t+1}$, $X_1, X_2, \ldots, (X_i - \{w\}) \cup \{u, v\}, \ldots X_{t+1}$ of $V(T')$ satisfies properties *(i)-(iii)* of Definition 7.1.3. Now, since T is a t-critter, there are subsets S_1^i and S_2^i of X_i that satisfy property *(iv)* from Definition 7.1.3. Then, in T', it can be seen that there exists $Z_1, Z_2 \subseteq \{u, v\}$ such that $(S_1^i - \{w\}) \cup Z_1$ and $(S_2^i - \{w\}) \cup Z_2$ satisfy property *(iv)* and for all other sets X_j, $j \neq i$ it can be seen (using the same S_1^j and S_2^j as in T) that property *(iv)* is satisfied in T'. This again contradicts the assumption that G is t-critter free. Lastly, if $w \in A_i$ for some i, it is straightforward to verify that the partitioning $A_1, A_2, \ldots, (A_i - \{w\}) \cup \{u, v\}, \ldots, A_{t+1}$, $B_1, B_2, \ldots, B_{t+1}$, $X_1, X_2, \ldots, X_{t+1}$ satisfies properties *(i)-(iv)* of Definition 7.1.3. This contradicts the assumption that G is t-critter free.

A straightforward application of induction then show show that if G'' is an induced

minor of G, then G'' is t-critter free. ■

7.3.3 An Overview of Lemma 7.3.2

In order to find a generalized ω-creature in a k-creature free graph G with many minimal separators, we will "grow" sets $A, B, C \subseteq V(G)$, where initially A and B only contain a single vertex and C is the empty set. The goal is that in the end we will be able to find a connected, good, full generalized ω-creature, $W = (G', H, \varphi, S_1, S_2)$, where A and B correspond to the sets A_φ and B_φ and $G' = G - C$. Intuitively, we use the minimal separators of G as a resource in order to "grow" the sets A, B, and C and slowly obtain all of the properties we will require of them. We try to motivate the key properties of A, B, and C now.

We have, from the definition of an ω-bistar partition, that for any two peripheral vertices $u, v \in H$ there can be no edge between the sets $\varphi^{-1}(u)$ and $\varphi^{-1}(v)$. Since, as noted as a consequence of property *(i)* in the definition of ω-creatures, for all $u \in H$, $\varphi^{-1}(u) \neq \emptyset$, the first property that we will need A, B, and C to satisfy is that $G - (A \cup B \cup C)$ has at least ω components (actually for technical reasons we will need that no component of $G - (A \cup B \cup C)$ contains many vertices which will imply $G - (A \cup B \cup C)$ has at least ω components).

We now try to motivate the second property that we will require of A, B, and C. As we mentioned before (when we introduced the definition of a generalized ω-creature) that since W is a generalized ω-creature we can construct a minimal separator, S, of G' by selecting sets of the form $S_1 \cap \varphi^{-1}(u)$ or $S_2 \cap \varphi^{-1}(u)$ for each peripheral vertex u in H and unioning them together and S will have the property that A_φ and B_φ will be contained in different S-full components of $G - S$. It turns out that in order to be able to find sets S_1 and S_2 that have this nice property, we must be very careful to ensure that

as we grow A, B, and C that G has many minimal separators (at least $\frac{\mu}{n^{polylog(n)}}$ where μ is the number of minimal separators of G) that are *consistent with A, B, and C*. We will formally define this property later, but roughly speaking, a minimal separator S is consistent with A, B, and C if A is contained in one S-full component of $G - S$ and B is contained in a different S-full component of $G - S$. This requirement isn't surprising, since as we just saw, the minimal separator S that was previously described has the property that A_φ and B_φ will be contained in different S-full components of $G - S$.

A third and closely related property of A, B, and C that we will require is that $G[A]$ and $G[B]$ have few components (where few means $\log^{O(1)}(n)$). This requirement is partially (but though not entirely) due to property *(iii)* of generalized ω-creatures. Intuitively, to get property *(iii)* to work out, we filter the minimal separators we consider in such a way so that for all remaining separators under consideration, if we take any pair, say S_1 and S_2, it holds that for every component X of $G - (A \cup B \cup C)$ and every pair of components P, Q of $G[A \cup B]$ there is a path from P to Q through $X - S_1$ if and only if there is a path from P to Q through $X - S_2$. As long as the number of components of $G[A]$ and $G[B]$ stay small then by the pigeon hole principle this filtering step will not lose too many minimal separators.

Thus, we have three apposing requirements we want A, B, and C to meet. On the one hand, we want $G - (A \cup B \cup C)$ to have no components with many vertices (and therefore $G - (A \cup B \cup C)$ will have at least ω components). This can be trivially satisfied by just picking A, B, and C to be really large sets. But, our second and third properties require us to be very careful with how we grow A, B, and C. Picking A, B, and C without much care would probably result in G not having many minimal separators that are consistent with A, B, and C and/or $G[A]$ and $G[B]$ having too many components.

It turns out that if we can find sets A, B, and C with these properties, then we would be able to construct a generalized ω-creature. Finding A, B, and C with these

three properties is the focus of Subsection 7.4.1. Recall though that we do not just need a generalized ω-creature, but a connected, good, full generalized ω-creature. Further enriching A, B, and C to allow us to make a connected, good, full generalized ω-creature is the goal of Subsection 7.4.2.

So, how do we find these sets A, B, and C that have the three previously described properties? It involves a rather surprising combination and extension of two ideas.

The first idea comes from a quasi-polynomial time branching algorithm for independent set on P_k-free graphs of Gartland and Lokshtanov [1]. In the algorithm used in [1], when a vertex, v, is branched on the algorithm is recursively called on the inputs $G - v$ and on $G - N[v]$. The algorithm works by using $n/2$-balanced separators dominated by few vertices (it is a critical property of P_k-free graph that they always have $n/2$-balanced separators dominated by few vertices, we prove that k-creature free graphs also have this property in Lemma 7.4.7) to guide the selection of a vertex v to branch on. This branching "efficiently" breaks up the graph into small connected components (similar to what want the set A, B, and C, to do, but where "efficient" now does not refer to run time, but the fact that G has many minimal separators that are consistent with A, B, and C. It turns out that these two notions of "efficient" are strongly connected). Because we are working with k-creature free graphs though instead of P_k-free graphs, our process of selecting which vertex v to branch on becomes more complicated and requires some new ideas.

The second idea we use allows us to bridge this gap between independent set branching used in [1] and our goal of "growing" the sets A, B, and C. This idea is a lemma which appears in [5] where the authors prove that if G is a k-creature free graph with n vertices, then for every $v \in V(G)$, if $S^v = \{N(v) \cap S |\ S \in \mathcal{S} \text{ and } v \notin S\}$ then $|S^v| \leqslant n^k$. Intuitively, when a vertex, v, has been selected, instead of branching on the vertex and removing the set $N[v]$ from G, this lemma allows us to to allocate the vertices of $N[v]$ to the set A, B,

and C in such a way that many minimal separators of G remain consistent with A, B, and C.

7.3.4 An Overview of Lemma 7.3.3

In the second part of the paper we extract a k-critter from an ω-creature as long as ω is sufficiently large compared to k, and satisfies some additional structural properties (namely is *good*, *connected*, and *full*). For the purposes of this overview we will not need the precise definition of these notions.

The first thing a reader should notice is that a k-critter *is* a generalized k-creature. Consider a k-critter partition $(A_1, \ldots, A_{k+1}, B_1, \ldots, B_{k+1}, X_1, \ldots, X_k)$ of a graph G. Set $A = \bigcup_i A_i$ and $B = \bigcup_i B_i$. Letting H be a k-bistar we define the function φ that maps all vertices in A to c_A, all vertices in B to c_B and each X_i to the i^{th} peripheral vertex of H. We then set $S_1 = \bigcup_i S_1^i$ and $S_2 = \bigcup_i S_2^i$. It can now be verified that $(G, H, \varphi, S_1, S_2)$ is in fact a generalized k-creature.

This fact is never stated or used explicitly in the proof, but it should help the reader understand what is going on. We have at hand a generalized ω-creature for some gigantic value of ω. Our goal is to extract from it a highly structured generalized k-creature. Here gigantic means that ω is bigger than any given function of k, but does not depend on n.

The key difference between a k-critter and a generalized ω-creature is that in a k-critter, every X_i has neighbors in precisely 4 components of $G[A \cup B]$, namely A_i, A_{i+1}, B_i and B_{i+1}. On the other hand, the (pre-image $\varphi^{-1}(v)$ of) peripheral vertices v of a generalized ω-creature (which correspond to the X_i's of a k-critter) may have neighbors in any number of components of $G[A \cup B]$.

Now the proof for extracting a k-critter from the generalized-ω creature goes like a typical "Ramsey Theory" style proof. We are given a large and unstructured object (here

the largeness of the object is the number ω of peripheral vertices) and the goal is to extract from it a smaller, but still large, structured sub-object. We proceed in several steps. First we extract a generalized ω-creature in which the *adhesion size* is upper bounded by $2k$. The adhesion size is the maximum number of components of $G[A \cup B]$ that a peripheral vertex $\varphi^{-1}(v)$ can have neighbors into. From the generalized ω-creature of bounded adhesion size we then extract a "path-like" generalized ω'-creature, for ω' roughly equal to $\log \omega$. After some additional steps we extract a "critter-like" generalized ω'-creature, which basically is an ω'-critter.

It would seem that we are done. But no! We are not able to actually execute the arguments in the way we described above. Instead we are able to get each of the properties listed above on "one side", say the A side, of the generalized ω'-creature. After the A-side is critter-like we need to turn around and re-do all of the arguments again for the B-side. However, this time when we are cleaning the B-side we need to make sure that we don't break the nice properties that we worked so hard for on the A side. After both sides of the generalized ω'-creature are critter-like, we inspect its properties and observe that it is in fact an ω'-critter. In several places of the argument we may stumble on a k-creature instead of the more structured generalized ω-creature that we are looking for. In that case we halt and declare a win.

In order to execute this Ramsey-Theory style argument we need to define ways in which we can remove a few pieces of our generalized ω-creature while still keeping it a generalized ω-creature. For this we need to define three operations – dissolve, absorb and erase. One can think of these operations as analogues standard graph operations like vertex deletion, edge deletion and edge contraction for graphs, but for operating on generalized ω-creatures instead.

Because generalized ω-creatures are somewhat brittle and have a long and technical definition, every change to them requires us to go over and verify that the definitions are

maintained. This makes many of these proofs excruciatingly long and wordy.

7.3.5 An Overview of Theorem 7.1.5

Let $\mathcal{F}$ be a CMSO-definable hereditary graph family. Theorem 7.1.5 consists of two main statements regarding $\mathcal{F}$ namely *(i)* If there exists an integer k such that $\mathcal{F}$ neither contains a k-creature nor a k-critter then $\mathcal{F}$ is quasi-tame, and *(ii)* otherwise $\mathcal{F}$ is feral. Statement *(i)* follows directly from Theorem 7.1.4. The premise of the "otherwise" case of Theorem 7.1.5 is that the family $\mathcal{F}$ contains, for every k, a k-creature or a k-critter. Because a k-creature contains a $(k-1)$-creature as an induced subgraph, and a k-critter contains a $(k-1)$-critter as an induced subgraph, it follows that $\mathcal{F}$ either contains a k-creature for every k, or a k-critter for every k (or both). Thus, to prove Theorem 7.1.5 we are left with proving two implications, namely

(i) if $\mathcal{F}$ contains a k-creature for every k then $\mathcal{F}$ is feral, and

(ii) if $\mathcal{F}$ contains a k-critter for every k then $\mathcal{F}$ is feral.

The approach for proving both implications is to use the fact that $\mathcal{F}$ is hereditary and CMSO-definable to show that if $\mathcal{F}$ contains a k-creature (k-critter) for every k then $\mathcal{F}$ contains for every k a k-creature (k-critter) with $O(k)$ vertices. Here the constant hidden in the big-oh depends on the CMSO formula describing $\mathcal{F}$. Since k-creatures and k-critters both contain 2^k minimal separators this proves that $\mathcal{F}$ is feral.

We now sketch the proof of *(i)*. The authors showed in [5] that for every k there exists a k' such that every k'-creature contains one of six highly structured k-creatures as an induced subgraph. Since our premise is that $\mathcal{F}$ contains a k-creature for *every* k, we may now assume without loss of generality that $\mathcal{F}$ contains one of these six structured k-creatures for every k.

All of the the six structured k-creatures have the property that they can be partitioned into at most $5k$ paths, such that only the endpoints of these paths have neighbours on the outside (this is not quite true, but close enough to the truth for this overview, see Section 7.6). Therefore, if the structured k-creature $G \in \mathcal{F}$ has more than $c \cdot 5k$ vertices, then it contains an induced path on at least c vertices, such that only the endpoints of the path have neighbors on the outside of the path.

We now use the "regular-like" properties of CMSO-definable families: From the perspective of being a member of $\mathcal{F}$ there exists some universal constant γ, such that for every path P of length more than γ there exists a path P' of length at most γ such that "replacing" P with P' in G (which amounts to shortening the relevant path by $|V(P)| - |V(P')|$ vertices) does not affect whether the graph is in $\mathcal{F}$ or not.

In particular if G was in $\mathcal{F}$ before the path shortening, then it is also in $\mathcal{F}$ after. But shortening a path in G does not destroy the property of G being a k-creature. Thus, as long as $c > \gamma$ and the number of vertices in the k-creature is at least $c \cdot 5k$ we can prove that $\mathcal{F}$ contains a structured k-creature strictly smaller than G. Hence $\mathcal{F}$ must contain a k-creature on at most $5ck$ vertices.

The proof for k-critters is based on the same idea, but now things are more technical because k-critters do not necessarily contain long paths. Instead we prove that a k-critter contains a cut of size 3 in each X_i, and use these cuts to drive our "pumping" argument.

7.4 Finding A Generalized ω-Creature

In order to find a generalized ω-creature in a k-creature free graph G with many minimal separators, we will "grow" sets $A, B, C \subseteq V(G)$, where initially A and B only contain a single vertex and C is the empty set. The goal is that in the end we will be able to find a generalized ω-creature, $W = (G', H, \varphi, S_1, S_2)$ where A and B correspond

to the sets A_φ and B_φ and $G' = G - C$. Intuitively, we use the minimal separators of G as a resource in order to "grow" the sets A, B, and C and slowly obtain all of the properties we require of them. As we mentioned before, when we construct a minimal separator, S, by selecting sets of the form $S_1 \cap \varphi^{-1}(u)$ or $S_2 \cap \varphi^{-1}(u)$ for $u \in H$, A_φ and B_φ will be contained in different S-full components of $G - S$. Hence, as we are growing the sets A, B, and C we should try and keep tract of the minimal separators, S, such that A is contained in one S-full component of $G - S$, B is contained in some different S-full component, and the vertices of C are "irrelavant" in some sense (as we wind up just removing the vertices of C from G to get G'). It turns out the the correct notion of "irrelavant" is that these vertices are in neither of the S-full components A and B are contained in. This motivates the following definition.

Definition 7.4.1. Let G be a graph, let $A, B, C \subseteq V(G)$, and let S be a minimal separator of G. We say that S is *consistent with A, B, and C* if the vertices of A all belong to one S-full component of $G - S$, the vertices of B all belong to a different S-full component of $G - S$, and no vertex of C belongs to the same component as a vertex in $A \cup B$ does in $G - S$.

We make the following two observations about this definition which we will use frequently without explicit reference.

Observation 7.4.2. *Let G be a graph and let $A, B, C \subseteq V(G)$. If there is at least one minimal separator of G that is consistent with A, B, and C, then A and B are anti-complete and C is disjoint from $A \cup B$.*

Proof: Let G be a graph and let $A, B, C \subseteq V(G)$ and assume S is a minimal separator of G that is consist with A, B, and C. Since A and B are contained in different components of $G - S$, then A and B must be anti-complete. Since no vertex

of C belongs the the same component that a vertex in $A \cup B$ does in $G - S$, C must be disjoint from $A \cup B$. ■

Observation 7.4.3. *Let G be a graph, let $A, B, C \subseteq V(G)$, and let S be a minimal separator that is consistent with A, B, and C. If $v \in S$, then S is consistent with A, B, and $C \cup \{v\}$.*

Proof: Let G be a graph, let $A, B, C \subseteq V(G)$, and let S be a minimal separator that is consistent with A, B, and C. If $v \in S$, then v will not belong to the same component that a vertex in $A \cup B$ does in $G - S$, hence S is consistent with A, B, and $C \cup \{v\}$. ■

We now formalize this notion of C being a set of "irrelevant" vertices in the following lemma.

Lemma 7.4.4. *Let G be a graph, let $A, B, C \subseteq V(G)$ with $A, B \neq \emptyset$, and let S be a minimal separator that is consistent with A, B, and C. Then $S - C$ is a minimal separator of $G - C$ that is consistent with A, B, and $\emptyset$. Furthermore, if S' is a minimal separator of G that is consistent with A, B, and C such that $S \neq S'$, then $S - C \neq S' - C$.*

Proof: Let G be a graph, let $A, B, C \subseteq V(G)$ with $A, B \neq \emptyset$, and let S be a minimal separator of G that is consistent with A, B, and C. Let A_S and B_S be the S-full components of $G - S$ that contain A and B respectively, so $C \cap (A_S \cup B_S) = \emptyset$. It follows that A_S and B_S are then $(S - C)$-full components of $(G - C) - (S - C)$, hence $S - C$ is a minimal separator of $G - C$ that is consistent with A, B, and $\emptyset$.

Now let S' be another minimal separator of G that is consistent with A, B, and C such that $S' \neq S$. By the preceding paragraph, if $A_{S'}$ and $B_{S'}$ are the S'-full components of $G - S'$ that contain A and B, then $A_{S'}$ and $B_{S'}$ are then $(S' - C)$-full components of $(G - C) - (S' - C)$. Now, if $S' - C = S - C$, then the $(S - C)$-full components are the same as the $(S' - C)$-full components, hence $A_S = A_{S'}$ and $B_S = B_{S'}$, but this then implies that $S = S'$. ■

The previous lemma more or less shows that we can remove the vertices of C from G without having much of an effect on the minimal separators of G that are consistent with A, B, and C, so the reader may intuitively think of these vertices as being irrelevant.

Lastly, we prove a lemma which shows that as far as minimal separators of G that are consistent with A, B, and C are concerned, we can intuitively think of the components of $G[A]$ and $G[B]$ as actually being just single vertices of the graph.

Lemma 7.4.5. *Let G be a graph, let $A, B, C \subseteq V(G)$, let S be a minimal separator of G that is consistent with A, B, and C, and let G' be the graph that results from contracting each component of $G[A]$ and $G[B]$ in G. Then S is a minimal separator of G'.*

Proof: Let G be a graph, let $A, B, C \subseteq V(G)$ and let S be a minimal separator of G that is consistent with A, B, and C, it follows that $S \cap (A \cup B) = \emptyset$. Let A_S and B_S be the S-full components of $G - S$ that contain A and B respectively. Let G' be the graph that results from contracting each component of $G[A]$ and $G[B]$ in G, and let A'_S and B'_S be the vertex sets of G' that correspond to A_S and B_S. Then A'_S and B'_S are S-full components of $G' - S$, hence S in a minimal separator in G'. ■

We could have actually strengthened the result of the previous lemma by also showing that if A' and B' are the vertices of G' that correspond to the components of $G[A]$ and $G[B]$ in G, then S is a minimal separator of G' that is consistent with A', B', and C. We have no need for such a strengthening though so we do not prove this. In most cases, we will not explicitly preform the contractions that Lemma 7.4.5 allows us to do as this would create some additional technical complications in our proofs. But, the reader should note that since the components of $G[A]$ and $G[B]$ can all be contracted to single vertices without destroying much structure in G (as far as minimal separators that are consistent with A, B, and C are concerned), these components can often intuitively be thought of as if they were single vertices.

In order to construct a generalized ω-creature from a k-creature free graph G with many minimal separators we will combine structural properties that k-creature free graphs have and use the minimal separators of G as a resource, which if given enough of (more than some amount that is quasi-polynomial in the number of vertices) will allow us to make a generalized ω-creature for some large value of ω. Our goal of creating a generalized ω-creature can be split into two main smaller goals.

The first goal is to find (or "grow") vertex sets A, B and C so that no component in $G - (A \cup B \cup C)$ has many vertices (say no more that n/δ for some large value of δ) while maintaining that a large fraction of the minimal separators of G are consistent with A, B, and C as well as that $G[A]$ and $G[B]$ have few (poly-log(n)) components. This process of growing A, B and C shares many similarities with the quasi-polynomial time algorithm for independent set on P_k-free graphs by [1], although more complex as P_k-free graphs are in many ways more structurally simple when compared to k-creature free graphs. Additionally, we borrow tools from [5] which allow us to translate independent set branching techniques into branching techniques that can be applied to minimal separators.

The second goal is to leverage the facts that $G[A]$ and $G[B]$ have few components (which from our previous discussion we can pretend each component is just a single vertex), $G - (A \cup B \cup C)$ has many small components, and G still has many minimal separators that are consistent with A, B, and C in order to "grow" A, B, and C into sets A', B', and C' and find a generalized ω-creature, $W = (G', H, \varphi, S_1, S_2)$, where G' is $G - C'$, $A_\varphi = A'$, $B_\varphi = B'$, and for each peripheral vertex $u \in H$ the set $\varphi^{-1}(u)$ corresponds to a component of $G - (A' \cup B' \cup C')$. Additionally, the generalized ω-creature we will build will have a little bit more structure than is required from a regular generalized ω-creature, which will allow us to prove in Section 7.5.1 that for all peripheral vertices $u \in H$, $\varphi^{-1}(u)$ will have neighbors in less than k components of $G'[A_\varphi]$ and

$G'[B_\varphi]$.

7.4.1 Breaking up G

Lemma 7.4.6. *Let G be a k-creature free graph (assume $k \geqslant 2$) with $n \geqslant 2$ vertices and μ minimal separators and let $\delta > 1$. Then there exists sets $A, B, C \subseteq V(G)$ with $A, B \neq \emptyset$ where the following properties hold:*

(i) No component of $G - (A \cup B \cup C)$ contains over n/δ vertices.

(ii) G has at least $\mu/(12n^{(k+1)})^{400k^3\delta^2 \log^4(n)}$ minimal separators that are consistent with A, B, and C.

(iii) The number of components of $G[A \cup B]$ is at most $400k^3\delta^2 \log^4(n)$.

This section is devoted to proving Lemma 7.4.6 and Lemma 7.4.6 is the only lemma from this section that is used outside of this section.

We find (or "grow") A, B, and C of Lemma 7.4.6 using techniques similar to those used by Gartland and Lokshtanov [1] where they give a quasi-polynomial time branching algorithm for independent set on P_k-free graphs. A crucial step of the algorithm is proving that P_k-free graphs have balanced separators that are dominated by few vertices. As we now show, k-creature free graphs also have balanced separators dominated by few vertices.

Let G be a graph with n vertices and let $S \subseteq V(G)$ such that S is *not* an $n/2$-balanced separator, hence $G - S$ has a unique component with over $n/2$ vertices. In the following proof we will refer to this component as the *large component* of $G - S$, all other components of $G - S$ will be referred to as *small components*.

Lemma 7.4.7. *Let G be a k-creature free graph with n vertices, then there is a set $S \subseteq V(G)$ such that S is an $n/2$-balanced separator and S can be dominated by $2k$ vertices.*

Proof: Let G be a k-creature free graph with n vertices. Assume for a contradiction that G does not contain an $n/2$-balanced separator dominated by at most $2k$ vertices. Consider all vertex sets, S, such that S is dominated by at most k vertices (hence S is not an $n/2$-balanced separator and therefore $G - S$ has a large component) and S is dominated by some small component, A, of $G - S$. The open neighborhood of a single vertex meets these conditions, so at least one set satisfies this property. Now, among all such sets, let S' be a set such that the size of the large component of $G - S'$ is smallest. Let C denote the large component of $G - S'$, let A denote some small component of $G - S'$ that dominates S', and let X be a vertex set of size at most k that dominates S'. Note by how S' was chosen (to minimize the size of the large component, C), S' is not anti-complete with C, since if it was, C would be a component of G. But then $c \in C$ would have the property that the largest component of $G - N_G(c)$ would have fewer vertices than C, which would contradict how S' was chosen.

Let $Y \subseteq V(G)$ be a set of size at most k. Since C is the only component of $G - S'$ that has over $n/2$ vertices, if no component of $G[C - N_G[Y]]$ has over $n/2$ vertices then no component of $G - (S' \cup N_G[Y])$ has over $n/2$ vertices. Since X dominates S' it follows that no component of $G - (N_G[X] \cup N_G[Y])$ has over $n/2$ vertices, but since X and Y both have size at most k this implies G has an $n/2$ balanced separator of size at most $2k$, contrary to assumption. So, for any set $Y \subseteq V(G)$, the largest component of $G[C - N_G[Y]]$ has over $n/2$ vertices and hence this largest component is unique.

For all sets $Z \subseteq S'$ of size at most k, let C_Z denote the component of $G[C - N_G[Z]]$ $= G[C - (N_G[Z] \cap C)]$ that has over $n/2$ vertices (which must exists by the previous paragraph), let γ_Z denote the number of neighbors that C_Z has in $S' - Z$, and let γ denote the minimum over all γ_Z. Let $Z' \subseteq S'$ be a set of size at most k where $C_{Z'}$ has exactly γ neighbors in $S' - Z'$. We study two cases now, one where $\gamma = 0$ and the other where $\gamma > 0$ and get a contradiction in both cases, which will allow us to conclude that

our assumption that G does not contain an $n/2$-balanced separator dominated by $2k$ vertices is impossible.

Case 1: Assume $\gamma = 0$. We will show that $N_G[Z'] \cap C$ is dominated by at most k vertices (namely the vertices of Z'), $N_G[Z'] \cap C$ is dominated by a small component of $G - (N_G[Z'] \cap C)$, and the large component of $G - (N_G[Z'] \cap C)$, is strictly smaller than the large component of $G - S'$, contradicting how S' was chosen. Since $|Z'| \leqslant k$, clearly any subset of $N_G[Z'] \cap C$ is dominated by at most k vertices, which establishes the first condition we must show.

Now, since C is a component of $G - S'$ and $C_{Z'}$ is a component of $G[C - (N_G[Z'] \cap C)]$, $C_{Z'}$ is a component of $G - (S' \cup (N_G[Z'] \cap C))$. Because $C_{Z'}$ is a component of $G - (S' \cup (N_G[Z'] \cap C))$, $N_G(C_{Z'}) \subseteq S' \cup (N_G[Z'] \cap C)$, furthermore by assumption $\gamma = 0$ so no vertex of $C_{Z'}$ has a neighbor in $S' - Z'$, it follows that $N_G(C_{Z'}) \subseteq N_G[Z'] \cap C$. Hence $C_{Z'}$ is a component of $G - (N_G[Z'] \cap C)$, and therefore the large component of $G - (N_G[Z'] \cap C)$ since by definition (two paragraphs above), $C_{Z'}$ has over $n/2$ vertices.

Now, if $N_G[Z'] \cap C = \emptyset$ then since $\gamma = 0$ this implies S' is anti-complete with C, but as noted in the first paragraph, S' is not anti-complete with C, so $N_G[Z'] \cap C \neq \emptyset$. Since $N_G[Z'] \cap C \neq \emptyset$, it holds that C'_Z is a strict subset of C. Lastly, recall that A is a small component of $G - S'$ that dominates S', so since A and S' are both disjoint from C (and hence no vertices from $A \cup S'$ belong to $C_{Z'}$) and $Z' \subseteq S'$, the vertices of $A \cup S'$ will belong to a small component of $G - (N_G[Z'] \cap C)$ that dominates $N_G[Z'] \cap C$. Hence, $N_G[Z'] \cap C$ is dominated by a set of at most k vertices (namely Z'), $N_G[Z'] \cap C$ is dominated by a small component of $G - (N_G[Z'] \cap C)$, and the large component of $G - (N_G[Z'] \cap C)$, specifically $C_{Z'}$, is strictly smaller than the large component of $G - S'$, specifically C, contradicting how S' was chosen. We may then conclude that $\gamma = 0$ is impossible.

Case 2: Assume $\gamma > 0$. Among all sets $Z \subseteq S'$ of size at most k such that number of neighbors C_Z has in $S' - Z$ is γ, let $\hat{Z}$ be one of smallest size (fewest vertices). If $|\hat{Z}| < k$ then γ must be 0 or else we could add an element of $S' - \hat{Z}$ that has a neighbor in $C_{\hat{Z}}$ to $\hat{Z}$ to get a set $\hat{Z}' \subseteq S'$ of size at most k and $C_{\hat{Z}'}$ would have less than γ neighbors in $S' - \hat{Z}'$ which contradicts the how we chose γ. So, since by assumption $\gamma \neq 0$, we have that $|\hat{Z}| = k$.

Next, we claim that every vertex $z_i \in \hat{Z}$ has a private neighbor $w_i \in N_G[\hat{Z}] \cap C$ which in turn has a neighbor in the large component of $C - N_G[\hat{Z}]$, $C_{\hat{Z}}$. Assume for a contradiction that this claim is false, so there is a vertex $z \in \hat{Z}$ such that for every neighbor $w \in N_G[\hat{Z}] \cap C$ that z has, where w also has a neighbor in $C_{\hat{Z}}$, there is another vertex in $\hat{Z} - z$ that is a neighbor of w. We claim that by how z was chosen, $C_{\hat{Z}} = C_{\hat{Z}-z}$. It follows immediately from the definition of these sets that $C_{\hat{Z}} \subseteq C_{\hat{Z}-z}$.

Next note that since $C_{\hat{Z}}$ is a component of $C - (N_G[\hat{Z}] \cap C)$, $N_C(C_{\hat{Z}}) \subseteq N_G[\hat{Z}] \cap C$. So, if $C_{\hat{Z}-z} \not\subseteq C_{\hat{Z}}$ then since both sets are connected and $C_{\hat{Z}} \subset C_{\hat{Z}-z} \subseteq C$, it must be that there is some $v \in C_{\hat{Z}-z}$ such that $v \in N_C(C_{\hat{Z}}) \subseteq N_G[\hat{Z}] \cap C$. Since no vertex of $N_G[\hat{Z} - z] \cap C$ is in $C_{\hat{Z}-z}$, it must be that $v \in N_G(z) \cap C$ and $v \notin N_G[\hat{Z} - z] \cap C$. Therefore $v \in N_G[\hat{Z}] \cap C$, is a neighbor of z, v has a neighbor in $C_{\hat{Z}}$, and v is not a neighbor of any vertex in $Z - z$, contrary to how z was chosen. Hence $C_{\hat{Z}} = C_{\hat{Z}-z}$.

So, there is no vertex in $C_{\hat{Z}-z}$ that is a neighbor of z, hence number of neighbors that $C_{\hat{Z}-z}$ has in $S' - (\hat{Z} - z)$ is γ, which contradicts that $\hat{Z}$ was chosen to be as small as possible since the number of neighbors $C_{\hat{Z}-z}$ has in $S' - (\hat{Z} - z)$ is also γ. We may then let W denote the set that contains these w_i's from the claim we just proved.

We now show that $(A, \hat{Z}, W, C_{\hat{Z}})$ is a k-creature. By how A and $C_{\hat{Z}}$ were chosen, we have that $G[A]$ and $G[C_{\hat{Z}}]$ are connected. Next, A and C are both vertex sets of components of $G - S'$, so A and $C_{\hat{Z}}$ are anti-complete since $C_{\hat{Z}} \subseteq C$, A and W are

anti-complete since $W \subseteq C$, and $\hat{Z}$ and $C_{\hat{Z}}$ are anti-complete since $C_{\hat{Z}} \subseteq C - N_G[\hat{Z}]$. Next, by the definition of A, A dominates S' and therefore dominates $\hat{Z} \subseteq S'$ and by definition of $C_{\hat{Z}}$ we can see that $C_{\hat{Z}}$ dominates W. Lastly, by how the vertices of W were selected, there is a semi-induced matching between $\hat{Z}$ and W and $|\hat{Z}| = |W| = k$. It follows that $(A, \hat{Z}, W, C_{\hat{Z}})$ is a k-creature, a contradiction to the assumption that G is k-creature free.

It now follows that the original assumption that G does not contain an $n/2$-balanced separator dominated by $2k$ vertices is impossible. ■

Lemma 7.4.8. *Let G be a k-creature free graph with n vertices and let $\delta > 1$, then there is a set $S \subseteq V(G)$ such that S is an n/δ-balanced separator and S can be dominated by $8k\delta$ vertices.*

Proof: Let G be a k-creature free graph with n vertices and let $\delta > 1$. We will prove by induction on i that G has an $n/2^i$-balanced separator dominated by $4k2^i$ vertices. By rounding to the nearest multiple of 2, this proves that G has an n/δ-balanced separator that is dominated by $8k\delta$ vertices.

Lemma 7.4.7 handles the base case where $i = 1$. Assume that for all i less than $j > 1$, G has a $n/2^i$-balanced separator dominated by $4k2^i$ vertices. We show that G has a $n/2^j$-balanced separator dominated by $4k2^j$ vertices. By the inductive hypothesis, G has an $n/2^{j-1}$-balanced separator, S, dominated by $4k2^{j-1}$ vertices. There are at most 2^j components of $G - S$ that have over $n/2^j$ vertices, so for each such component, C_r, apply Lemma 7.4.7 to get an $|C_r|/2$-balanced separator S_r for $G[C_r]$ that is dominated by $2k$ vertices. Setting $S' = S \cup (\bigcup_r S_r)$ it follows that S' is an $n/2^j$-balanced separator for G. Furthermore, since there are at most 2^j S_r's, each of which is dominated by $2k$ vertices, S' is dominated by $4k2^{j-1} + 2k2^j = 4k2^j$ vertices. ■

In the branching algorithm used in [1], balanced separators for a P_k-free graph, G,

are collected in order to guide in the selection of a vertex, v, to branch on, the algorithm is recursively called on the input $G - v$ and on $G - N[v]$ and it can be shown that with well chosen v, the graph G will be efficiently broken up into small components. In our methods here, we do not wish to remove vertices from the graph, but either add v or $N[v]$ to the sets A, B and C to get new A, B, and C sets. We actually will not be branching either, but greedily choosing the branch that roughly corresponds to the one that contains a new A, B, and C that have the most minimal separators that are consistent with the new A, B, and C. We will use the term *refining* on a vertex v when refer to this "greedy branching" process. At each refining step, some vertices are added to A, B, and C, so the number of minimal separators that are consistent with A, B, and C drops by some factor, but because this refining process efficiently breaks up the graph into small components, we will be able to grow A, B, and C into sets where all components of $G - (A \cup B \cup C)$ are small, and additionally there will still be some large fraction of minimal separators of G that are consistent with A, B, and C.

We now discuss refining in a little more in depth. In general, given $v \in G$, there are $3^{|N_G[v]|}$ way to allocate the vertices of $N[v]$ into the sets A, B, and C (and therefore $3^{|N[v]|}$ different refinement options that must be considered), so we cannot guarantee that any of the refinement options correspond to a new A, B, and C such that a large fraction of minimal separators are consistent with A, B, and C. But, if G is k-creature free, then the next three lemmas, the first of which is taken from [5] and the second of which is a slight strengthening of the first, shows that we only need to consider roughly n^k refinement options and this allows us to guarantee that at least one option corresponds to a new A, B, and C such that a large fraction of minimal separators are consistent with A, B, and C.

Lemma 7.4.9 ([5]). *Let G be a k-creature-free graph with n vertices and let $\mathcal{S}$ be a set*

of minimal separators of G. Then for every $v \in V(G)$, if $S^v = \{N(v) \cap S|\ S \in \mathcal{S}$ and $v \notin S\}$ then $|S^v| \leqslant n^k$.

Lemma 7.4.10. *Let G be a k-creature free graph with n vertices, let $U \subseteq V(G)$ be a vertex set such that $G[U]$ has c components, let $\mathcal{S}$ be a set of minimal separators of G, and let $\mathcal{S}^U = \{N(U) \cap S | S \in \mathcal{S}$ and $U \cap S = \emptyset\}$. Then $|\mathcal{S}^U| \leqslant n^{kc}$.*

Proof: Let G be a k-creature free graph with n vertices, let $U \subseteq V(G)$ be a vertex set such that $G[U]$ has c components, let G' be the graph that results from contracting each component of $G[U]$ in G, and let U' be the vertices of G' that correspond to the components of $G[U]$, so U' has c vertices and by Lemma 7.3.4 G' is k-creature free. Observe that if S is a minimal separator of G such that $U \cap S = \emptyset$, then S is also a minimal separator of G' such that $S \cap U' = \emptyset$ and furthermore $N_G(U) \cap S = N_{G'}(U') \cap S$. So to prove this lemma it is sufficient to prove that if $\mathcal{S}^{U'} = \{N(U') \cap S | S \in \mathcal{S}$ and $U \cap S = \emptyset\}$ then $|\mathcal{S}^{U'}| \leqslant n^{kc}$.

For $u_i \in U'$ let $S^{u_i} = \{N(u_i) \cap S | S \in \mathcal{S}$ and $u_i \notin S\}$ and let $V \in S^{U'}$. Then we can see that for each $u_i \in U'$ we can select a $V_i \in S^{u_i}$ such that $V = \bigcup V_i$. Since by Lemma 7.4.9 S^{u_i} has at most n^k elements and U' has c elements, it follows that $S^{U'}$ has at most n^{kc} elements. ■

Lemma 7.4.11. *Let G be a k-creature free graph with n vertices, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, and let $v \in G$. Then at least one of the following cases apply:*

(i) At least $(1 - 1/n)\mu$ minimal separators of G are consistent with A, B, and $C \cup \{v\}$.

(ii) There exists $A', B', C' \subseteq V(G)$ such that at least $(1/3)(1/n^{k+1})\mu$ minimal separators of G are consistent with A', B', and C' where $A \subseteq A'$, $B \subseteq B'$, and $C \subseteq C'$, $N_G[v] \subseteq (A' \cup B' \cup C')$, and the number of components in $G[A' \cup B']$ is at most one more than the number of components in $G[A \cup B]$.

Proof: Let G be a k-creature free graph with n vertices, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, and let $\mathcal{S}$ denote this set of minimal separators. Let $v \in G$, if at least $(1 - 1/n)\mu$ minimal separators of $\mathcal{S}$ contain v, then since S is consistent with A, B, and $C \cup \{v\}$, case (i) is satisfied. So, we may assume that at least μ/n minimal separators of $\mathcal{S}$ do not contain v. It then follows that for at least $\mu/3n$ minimal separators $S \in \mathcal{S}$, either v belongs to the same component that A does in $G - S$, v belongs to the same component B does in $G - S$, or v does not belong to the same component A does in $G - S$ nor the same component B does in $G - S$.

First, assume that for at least $\mu/3n$ minimal separators $S \in \mathcal{S}$, v belongs to the same component that A does in $G - S$, denote this subset of $\mathcal{S}$ by $\mathcal{S}_A$. By Lemma 7.4.10 there are at most n^k sets of the form $N_G(v) \cap S$ for $S \in \mathcal{S}_A$, hence for some set, X, there are at least $\mu/3n^{k+1}$ minimal separators, S, of $\mathcal{S}_A$ such that $N_G(v) \cap S = X$, denote this subset of $\mathcal{S}_A$ as $\mathcal{S}_A^X$. Since for all $S \in \mathcal{S}_A^X$ v belongs to the same component that A does in $G - S$, it follows that all vertices of $N_G[v] - X$ belong to the same component that A does in $G - S$, hence S is consistent with $A \cup N_G[v] - X, B$ and C. Furthermore, since $X \subset S$, it then holds that S is consistent with $A \cup N_G[v] - X, B$, and $C \cup X$. It then follows that case (ii) of the lemma statement is satisfied in this case (the additional new component of $G[A' \cup B']$ arises if $N_G[v] - X$ is anti-complete with A, making $N_G[v] - X$ the new component of $G[A' \cup B']$).

The case where for at least $\mu/3n$ minimal separators $S \in \mathcal{S}$, v belongs to the same component that B does in $G - S$ is handled in the exact same was as in the previous paragraph. So, we now consider the case where for at least $\mu/3n$ minimal separators $S \in \mathcal{S}$, v does not belong to S, nor does v belong the same component A does in $G - S$, nor the same component B does in $G - S$. Denote this subset of $\mathcal{S}$ by $\mathcal{S}'$. Since for $S \in \mathcal{S}'$ $v \notin S$, it follows that no vertex of $N_G(v)$ belongs to the same component A does in $G - S$

nor the same component B does in $G - S$. Hence, for $S \in \mathcal{S}'$, S is consistent with A, B, and $C \cup N_G[v]$. It then follows that case (ii) of the lemma statement is satisfied in this case, completing the proof. ■

We now define a function which formalizes this refinement process (greedily choosing the the best branch). This function will play a central role in this section.

Definition 7.4.12. Let G be a k-creature free graph with n vertices, let $v \in G$, and let $A, B, C \subseteq V(G)$. We define a function **Refine**(G, v, A, B, C). We match the output of **Refine**(G, v, A, B, C) according to the first case of Lemma 7.4.11 that is met by G, v, A, B, and C. **Refine**(G, v, A, B, C) returns:

(i) $(A, B, C \cup \{v\})$ if case (i) of Lemma 7.4.11 is the first case satisfied.

(ii) (A', B', C') if case (ii) of Lemma 7.4.11 is the first case satisfied, where A', B' and C' are the sets whose existence is given by case (ii) of Lemma 7.4.11.

If the return of **Refine**(G, v, A, B, C) comes from case (i) then we call **Refine**(G, v, A, B, C) a $failure$ refinement, else we call **Refine**(G, v, A, B, C) a *success* refinement.

A simple application of Lemma 7.4.11 then gives us the following lemma.

Lemma 7.4.13. *Let G be a k-creature free graph with n vertices, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, let $v \in G$, and let $A', B', C' =$ **Refine**(G, v, A, B, C). Then the following conditions hold:*

*(i) If **Refine**(G, v, A, B, C) is a failure refinement then G has at least $(1 - 1/n)\mu$ minimal separators that are consistent with A', B', and C' and $A' = A$, $B' = B$, and $C' = C \cup \{v\}$.*

*(ii) If **Refine**(G, v, A, B, C) is a success refinement then G has at least $(1/3)(1/n^{k+1})\mu$ minimal separators that are consistent with A', B', and C', $A \subseteq A'$, $B \subseteq B'$, and*

$C \subseteq C'$, $N_G[v] \subseteq (A' \cup B' \cup C')$, and the number of components of $G[A' \cup B']$ is at most one more than the number of components of $G[A \cup B]$.

Collecting Separators with Anti-Complete Cores.

This subsection is devoted to proving Lemma 7.4.14, which is the only lemma from this subsection that is used outside of this subsection. Before we state Lemma 7.4.14 we give a short definition pertaining to vertex lists. Let G be a graph and let $\mathcal{S}$ be a list that contains vertices from G. We will say that $\mathcal{S}$ is an *anti-complete vertex list* if for every two elements $S_i, S_j \in \mathcal{S}$, S_i is anti-complete with S_j. Note that this implies that if $S_i = S_j$ then $S_i = S_j = \emptyset$.

Lemma 7.4.14. *Let G be a k-creature free graph with $n \geqslant 2$ vertices and μ minimal separators and let $\delta > 1$. Without loss of generality assume $k \geqslant 2$. Then there exists sets $A, B, C \in V(G)$ with $A, B \neq \emptyset$ and an anti-complete vertex list $\mathcal{S}$ of size $log(n) + 1$ such that the following properties hold:*

(i) Let $G' = G - (A \cup B \cup C)$. For all $S_i \in \mathcal{S}$, $N_G[S_i] - (A \cup B \cup C)$ is an n/δ-balanced separator of G'.

(ii) G has at least $\mu/(4(3n^{(k+1)})^{160k^2\delta^2 \log^3(n)+2})$ minimal separators that are consistent with A, B, and C.

(iii) $G[A \cup B]$ has at most $160k^2\delta^2 \log^3(n) + 2$ components.

(iv) No vertex v in $G' = G - (A \cup B \cup C)$ belongs to all sets of $N_{G'}[\mathcal{S}]$.

The algorithm from [1] (which the techniques of subsubsection 7.4.1 is based on) works by collecting balanced separators dominated by few vertices while branching on vertices who have a sufficient number of neighbors into the collected balanced separators.

Eventually this leads to having a large collection of balanced separators such that no vertex of the remaining graph has neighbors in more than $\log(n)$ of the collected balanced separators. Since the graphs the algorithm of [1] is run on are P_k-free, this is enough to guarantee that the graph now has been efficiently broken up into small components, and the algorithm is then called on each component.

Lemma 7.4.14 is proved using very similar techniques to the algorithm described in the previous paragraph, even up to determining what vertex to refine (branch) on. This process of collecting balanced separators and refining allows us to gather a vertex list, $\mathcal{S}$, of balanced separators as well as vertex sets A, B, and C which satisfy the properties stated in Lemma 7.4.14. A key difference is that when working with P_k-free graphs, this would be enough to ensure that $G - (A \cup B \cup C)$ has no large component, thus proving Lemma 7.4.6, but when we are dealing with graphs that are k-creature free we must do more in order to break up the graph into small components. We cover this additional process in subsubsection 7.4.1. A second key difference here is that in the independent set algorithm for P_k-free graphs of [1], the algorithm continues to recurse on connected components after breaking up the graph. Here in this paper, we stop our process after finding A, B, and C such that $G - (A \cup B \cup C)$ has no large components, we do not do any recursion on the components.

In order to prove Lemma 7.4.14, we will study a sequence produced from a k-creature free graph G and a number $\delta > 1$, which we denote by $\mathsf{seq}_1(G, \delta)$. $\mathsf{seq}_1(G, \delta)$ is a sequence of tuples $(A_i, B_i, C_i, \mathcal{S}_i)$ where $A_i, B_i, C_i \subseteq V(G)$ and $\mathcal{S}_i$ is a vertex list. Before we can describe how the sequence is created, we need to provide the following definitions.

Definition 7.4.15. Let G be a k-creature free graph and let $(A_i, B_i, C_i, \mathcal{S}_i)$ be the i^{th} element of $\mathsf{seq}_1(G, \delta)$. For all natural numbers j, the j^{th} *level set with respect to* A_i, B_i, C_i *and* $\mathcal{S}_i$, denoted by $\mathcal{L}_j(A_i, B_i, C_i, \mathcal{S}_i)$, is defined as the set of vertices that belong to at

least j sets (counting multiplicity) of $N_G[\mathcal{S}_i] - (A_i \cup B_i \cup C_i)$.

Definition 7.4.16. Let G be a k-creature free graph with n vertices and let $(A_i, B_i, C_i, \mathcal{S}_i)$ be the i^{th} element of $\mathsf{seq}_1(G, \delta)$. A vertex $v \in G'$, $G' = G - (A_i \cup B_i \cup C_i)$, is *good for refining* with respect to $(A_i, B_i, C_i, \mathcal{S}_i)$ if there exists a j such that $|N_{G'}[v] \cap \mathcal{L}_j(A_i, B_i, C_i, \mathcal{S}_i)| \geqslant n/2^j$.

We shall use the notion of "good for refining" when determining what the $i+1^{th}$ tuple of the sequence will be. Note that this definition implies that any vertex of $G - (A_i \cup B_i \cup C_i)$ that belongs to $\mathcal{L}_{\log(n)}(A_i, B_i, C_i, \mathcal{S}_i)$ is good for refining, meaning any vertex that belongs to $\log(n)$ sets of $N_G[\mathcal{S}_i] - (A_i \cup B_i \cup C_i)$ is good for refining.

Let G be a k-creature free graph with n vertices and μ minimal separators and let $\delta > 1$. We now define $\mathsf{seq}_1(G, \delta)$. Let $a, b \in G$ be two vertices such that at least μ/n^2 minimal separators of G are a, b-minimal separators (since there are $n(n-1)/2$ pairs of vertices in G and every minimal separator is a u, v-minimal separator for some pair $u, v \in G$ such a pair a, b must exist). For the base case of this sequence we define $A_1 = \{a\}$, $B_1 = \{b\}$, $C_1 = \emptyset$, and $\mathcal{S}_1 = \emptyset$. We will maintain throughout the sequence that $A_i \subseteq A_{i+1}$, $B_i \subseteq B_{i+1}$, $C_i \subseteq C_{i+1}$, and $\mathcal{S}_i \subseteq \mathcal{S}_{i+1}$. We recursively define this sequence as follows.

Assume we are given $(A_i, B_i, C_i, \mathcal{S}_i)$, we will refer to the following process of determine the next tuple $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ as the i^{th} *step of* $\mathsf{seq}_1(G, \delta)$. If there is a vertex $v \in G - (A_i \cup B_i \cup C_i)$ that is good for refining, then we set $A_{i+1}, B_{i+1}, C_{i+1} = \mathbf{Refine}(G, v, A_i, B_i, C_i)$ and $\mathcal{S}_{i+1} = \mathcal{S}_i$ (hence, by the definition of **Refine**, $A_i \subseteq A_{i+1}$, $B_i \subseteq B_{i+1}$, and $C_i \subseteq C_{i+1}$). If $\mathbf{Refine}(G, v, A_i, B_i, C_i)$ is a failure refinement then we call $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ a *failure tuple*, else $\mathbf{Refine}(G, v, A_i, B_i, C_i)$ is a success refinement and we call $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ a *success tuple*.

If there is no vertex that is good for refining, then let $G' = G - (A_i \cup B_i \cup C_i)$. Since

G' is a subgraph of G, G' has at most n vertices and by Lemma 7.3.4 G' is a k-creature free, so by Lemma 7.4.8 there exists a (possibly empty) set of vertices $S \subseteq V(G')$ where $|S| \leqslant 8k\delta$ and $N_{G'}[S]$ is an n/δ-balanced separator for G'. Set $A_{i+1} = A_i$, $B_{i+1} = B_i$, $C_{i+1} = C_i$, and $\mathcal{S}_{i+1} = \mathcal{S}_i \cup \{S\}$. In this case we call $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ a *separator tuple*.

The sequence terminates once we reach a tuple $(A_j, B_j, C_j, \mathcal{S}_j)$ such that $|\mathcal{S}_j| = 10k\delta \log^2(n)$.

The following observation was noted just before defining $\mathsf{seq}_1(G, \delta)$ and follows directly from the definition of $\mathsf{seq}_1(G, \delta)$. We will use it frequently without explicit reference.

Observation 7.4.17. *Let G be a k-creature free graph, let $\delta > 1$, and let $(A_i, B_i, C_i, \mathcal{S}_i)$ and $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ be the i^{th} and $i + 1^{th}$ elements of the sequence $\mathsf{seq}_1(G, \delta)$. Then $A_i \subseteq A_{i+1}$, $B_i \subseteq B_{i+1}$, $C_i \subseteq C_{i+1}$, and $\mathcal{S}_i \subseteq \mathcal{S}_{i+1}$.*

Lemma 7.4.18. *Let G be a k-creature free graph and let $\delta > 1$. Then $\mathsf{seq}_1(G, \delta)$ is finite.*

Proof: Let G be a k-creature free graph. Consider the i^{th} and $i + 1^{th}$ tuples of $\mathsf{seq}_1(G, \delta)$, $(A_i, B_i, C_i, \mathcal{S}_i)$ and $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$. If $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a success or failure tuple then $\mathcal{S}_i = \mathcal{S}_{i+1}$, and if $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a separator tuple, then $|\mathcal{S}_i| + 1 = |\mathcal{S}_{i+1}|$. Since the sequence ends once we reach a tuple with $(A_j, B_j, C_j, \mathcal{S}_j)$ where $\mathcal{S}_j = 10k\delta \log^2(n)$, the sequence will terminate after the $10k\delta \log^2(n)^{th}$ separator tuple. So all we must show is there there is a finite number of success and failure tuples. This follows from three facts. The first is that regardless if $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a success, failure, or separator tuple, $A_i \subseteq A_{i+1}$, $B_i \subseteq B_{i+1}$, $C_i \subseteq C_{i+1}$. The second is that if $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a success or failure tuple, then at the i^{th} step, **Refine** is called on a vertex that belongs to $G - (A_i \cup B_i \cup C_i)$ and it follows that $(A_i \cup B_i \cup C_i) \subset (A_{i+1} \cup B_{i+1} \cup C_{i+1})$ where the containment is strict. Hence, if there are

$|V(G)|$ failure or success tuples that precede $(A_i, B_i, C_i, \mathcal{S}_i)$, then $A_i \cup B_i \cup C_i = V(G)$. The third fact is that if $A_i \cup B_i \cup C_i = V(G)$, then there are no vertices that are good for refining at this step (or any future step) and therefore $(A_{j+1}, B_{j+1}, C_{j+1}, \mathcal{S}_{j+1})$ is a separator tuple. Hence there are at most $|V(G)|$ success and failure tuples. ■

It now follows from 7.4.18 that we may assume there is a last element of $\mathsf{seq}_1(G, \delta)$.

Lemma 7.4.19. *Let G be a k-creature free graph with n vertices, let $\delta > 1$, and let $(A, B, C, \mathcal{S})$ denote the last tuple of $\mathsf{seq}_1(G, \delta)$. Let $G' = G - (A \cup B \cup C)$, then for every $S \in \mathcal{S}$, $N_G[S] - (A \cup B \cup C)$ is an n/δ-balanced separator for G'.*

Proof: Let G be a k-creature free graph with n vertices, let $\delta > 1$, and let $(A, B, C, \mathcal{S})$ denote the last tuple of $\mathsf{seq}_1(G, \delta)$. Let S_j denote the j^{th} element in the list $\mathcal{S}$. Then there is some tuple $(A_i, B_i, C_i, \mathcal{S}_i)$ of $\mathsf{seq}_1(G, \delta)$ such that if $G' = G - (A_i \cup B_i \cup C_i)$ then $S_j \subseteq V(G')$ and $N_{G'}[S_j] = N_G[S_j] - (A_i \cup B_i \cup C_i)$ is an n/δ-balanced separator for G'. Since $A_i \subseteq A$, $B_i \subseteq B$, and $C_i \subseteq C$, it follows that $N_G[S_j] - (A \cup B \cup C)$ is a balanced separator for G'' where $G'' = G - (A \cup B \cup C)$. ■

Lemma 7.4.20. *Let G be a k-creature free graph with n vertices, let $\delta > 1$, and let $(A_i, B_i, C_i, \mathcal{S}_i)$ denote the i^{th} tuple of $\mathsf{seq}_1(G, \delta)$. Then no vertex of $G - (A_i \cup B_i \cup C_i)$ belongs to over $\log(n)$ sets (counting multiplicity) of $N_G[\mathcal{S}_i] - (A_i \cup B_i \cup C_i)$.*

Proof: Let G be a k-creature free graph with n vertices, let $\delta > 1$, and let $(A_i, B_i, C_i, \mathcal{S}_i)$ denote the i^{th} tuple of $\mathsf{seq}_1(G, \delta)$. We will prove by induction on i that no vertex of $G' = G - (A_i \cup B_i \cup C_i)$ belongs to over $\log(n)$ sets of $N_G[\mathcal{S}_i] - (A_i \cup B_i \cup C_i)$. If $i = 1$ then $(A_i, B_i, C_i, \mathcal{S}_i)$ is the first element of the sequence, and by definition $\mathcal{S}_i = \emptyset$, so the result holds for the base case. Assume the result holds for all i less than $j > 1$, we prove it holds for $i = j$.

Consider the $j - 1^{th}$ and j^{th} elements of $\mathsf{seq}_1(G, \delta)$, $(A_{j-1}, B_{j-1}, C_{j-1}, \mathcal{S}_{j-1})$ and $(A_j, B_j, C_j, \mathcal{S}_j)$. If $(A_j, B_j, C_j, \mathcal{S}_j)$ is a success or failure tuple then $A_i \subseteq A_{i+1}$, $B_i \subseteq$

B_{i+1}, $C_i \subseteq C_{i+1}$, and $\mathcal{S}_{j-1} = \mathcal{S}_j$. Then by the induction hypothesis no vertex of $G - (A_{j-1} \cup B_{j-1} \cup C_{j-1})$ belongs to over $\log(n)$ vertices of $N_G[\mathcal{S}_{j-1}] - (A_{j-1} \cup B_{j-1} \cup C_{j-1})$ which implies no vertex of $G - (A_j \cup B_j \cup C_j)$ belongs to over $\log(n)$ vertices of $N_G[\mathcal{S}_j] - (A_j \cup B_j \cup C_j)$. If $(A_j, B_j, C_j, \mathcal{S}_j)$ is a separator tuple then this implies that no vertex of $G - (A_{j-1} \cup B_{j-1} \cup C_{j-1})$ belongs to $\log(n)$ or more set (counting multiplicity) of $N_G[\mathcal{S}_{j-1}] - (A_{j-1} \cup B_{j-1} \cup C_{j-1})$ since such a vertex would good for refining. Hence, no vertex of $G - (A_j \cup B_j \cup C_j)$ belongs to over $\log(n)$ vertices of $N_G[\mathcal{S}_j] - (A_j \cup B_j \cup C_j)$.

■

Let d be a natural number and let G be a graph with n vertices. G is said to be *d-degenerate* if there is a bijective function $f : V(G) \to [n]$ where for each vertex $v \in G$, v has at most d neighbors $u \in G$ such that $f(u) < f(v)$. The function f is called a *degeneracy ordering* of G. We will need the following easy to prove lemma, which is folklore, and so we omit the proof.

Lemma 7.4.21 (folklore). *Let G be a d-degenerate graph with n vertices. Then G has an independent set of size $\lceil n/(d+1) \rceil$.*

Let $(A, B, C, \mathcal{S})$ be the last tuple of $\mathsf{seq}_1(G, \delta)$, let $(A_{i-1}, B_{i-1}, C_{i-1}, \mathcal{S}_{i-1})$ and $(A_i, B_i, C_i, \mathcal{S}_i)$ be the $i-1^{th}$ and i^{th} elements of $\mathsf{seq}_1(G, \delta)$, and let $S \in \mathcal{S}$. We say S was added at step $i-1$ if $S_{i-1} \cup \{S\} = S_i$.

Lemma 7.4.22. *Let G be a k-creature free graph (assume $k \geqslant 2$) with n vertices, let $\delta > 1$, and let $(A, B, C, \mathcal{S})$ be the last tuple of $\mathsf{seq}_1(G, \delta)$. There exists an anti-complete sub-list $\mathcal{S}' \subseteq \mathcal{S}$ of size at least $\log(n) + 1$.*

Proof: Let G be a k-creature free graph (assume $k \geqslant 2$) with n vertices, let $\delta > 1$, and let $(A, B, C, \mathcal{S})$ be the last tuple of $\mathsf{seq}_1(G, \delta)$. Let $G_{\mathcal{S}}$ be a graph with $|\mathcal{S}|$ vertices and let f be a bijective function $f : V(G_{\mathcal{S}}) \to [n]$. The vertex $v \in G_{\mathcal{S}}$ will correspond to

the $f(v)^{th}$ element of $\mathcal{S}$. We now define the edges of $G_{\mathcal{S}}$. Let $u, v \in G_{\mathcal{S}}$, the edge uv is in $E(G_{\mathcal{S}})$ if and only if the $f(v)^{th}$ element and $f(u)^{th}$ element of $G_{\mathcal{S}}$ are not anti-complete. The statement of this lemma is then equivalent to $G_{\mathcal{S}}$ having an independent set of size at least $\log(n) + 1$. In order to establish this, we show $G_{\mathcal{S}}$ is $8k\delta \log(n)$-degenerate with degeneracy ordering f and apply Lemma 7.4.21.

Let $v \in G_{\mathcal{S}}$ and assume that $S_{f(v)}$, the $f(v)^{th}$ element of $\mathcal{S}$, was added at the $i - 1^{th}$ step of $\mathsf{seq}_1(G, \delta)$. Let $(A_{i-1}, B_{i-1}, C_{i-1}, \mathcal{S}_{i-1})$ and $(A_i, B_i, C_i, \mathcal{S}_i)$ be the $(i-1)^{th}$ and i^{th} elements of $\mathsf{seq}_1(G, \delta)$. It follows that if $G' = G - (A_{i-1} \cup B_{i-1} \cup C_{i-1})$ then $S_{f(v)} \subseteq V(G')$ and for $u \in G_{\mathcal{S}}$ with $f(u) < f(v)$ the $f(u)^{th}$ element of $\mathcal{S}$ is also the $f(u)^{th}$ element of $\mathcal{S}_{i-1}$. By Lemma 7.4.20 each vertex of $S_{f(v)}$ belongs to at most $\log(n)$ sets of $N_{G'}[\mathcal{S}_{i-1}]$, and since there are at most $8k\delta$ vertices in $S_{f(v)}$ this implies v has at most $8k\delta \log(n)$ neighbors u where $f(u) < f(v)$, proving $G_{\mathcal{S}}$ is $8k\delta \log(n)$-degenerate with degeneracy ordering f.

Now since $G_{\mathcal{S}}$ is $8k\delta \log(n)$-degenerate, and $G_{\mathcal{S}}$ has $|\mathcal{S}| = 10k\delta \log(n)^2$ vertices (by the definition of $\mathsf{seq}_1(G, \delta)$), it follows from Lemma 7.4.21 that $G_{\mathcal{S}}$ has an independent set of size at least $\log(n) + 1$. ■

We now wish to show that $\mathsf{seq}_1(G, \delta)$ does not contain many success tuples. Toward this end we track the sizes of the level sets. Let i be a natural number, we will say that a vertex $v \in G$ is *added to level set $\mathcal{L}_i$ at step $j - 1$* if for the j^{th} tuple, $(A_j, B_j, C_j, \mathcal{S}_j)$, of $\mathsf{seq}_1(G, \delta)$ v is in $\mathcal{L}_i(A_j, B_j, C_j, \mathcal{S}_j)$, but v is not in $\mathcal{L}_i(A_{j-1}, B_{j-1}, C_{j-1}, \mathcal{S}_{j-1})$. We say that $v \in G$ is added to level set $\mathcal{L}_i$ if it is added to level set $\mathcal{L}_i$ at step $j - 1$ for some j.

Similarly, we will say that a vertex $v \in G$ is *removed from level set $\mathcal{L}_i$ at step j* if for the j^{th} tuple, $(A_j, B_j, C_j, \mathcal{S}_j)$, of $\mathsf{seq}_1(G, A, B, C)$, v is in $\mathcal{L}_i(A_j, B_j, C_j, \mathcal{S}_j)$, but v is not in $\mathcal{L}_i(A_{j+1}, B_{j+1}, C_{j+1}, \mathcal{S}_{j+1})$. We say that $v \in G$ is removed from level set $\mathcal{L}_i$ if it is removed from level set $\mathcal{L}_i$ at step j for some j.

Lemma 7.4.23. *Let G be a k-creature free graph with n vertices and let $\delta > 1$. In the sequence $\mathsf{seq}_1(G, \delta)$, for all natural numbers i, at most $160k^2\delta^2 n \log^2(n)/2^i$ vertices are added to level set $\mathcal{L}_i$.*

Proof: Let G be a k-creature free graph with n vertices and let $\delta > 1$. Consider the $j-1^{th}$ and j^{th} element of $\mathsf{seq}_1(G, \delta)$, $(A_{j-1}, B_{j-1}, C_{j-1}, \mathcal{S}_{j-1})$ and $(A_j, B_j, C_j, \mathcal{S}_j)$. First, assume that $(A_j, B_j, C_j, \mathcal{S}_j)$ is a separator tuple. Since $(A_j, B_j, C_j, \mathcal{S}_j)$ is a separator tuple, there was no vertex that was good for refining in step $j-1$, in particular this implies that no vertex of $G' = G - (A_{j-1} \cup B_{j-1} \cup C_{j-1})$ has over $n/2^{i-1}$ neighbors in $\mathcal{L}_{i-1}(A_{j-1}, B_{j-1}, C_{j-1}, \mathcal{S}_{j-1})$. If a vertex v is added to level set $\mathcal{L}_i$ at step $j-1$ then since $\mathcal{S}_j$ only has one additional set, call it S, that $\mathcal{S}_{j-1}$ does not have, it follows that $v \in \mathcal{L}_{i-1}(A_{j-1}, B_{j-1}, C_{j-1}, \mathcal{S}_{j-1})$ and $v \in N_G[S] - (A_{j-1} \cup B_{j-1} \cup C_{j-1})$. Since $|S| \leqslant 8k\delta$ and no vertex of S has over $n/2^{i-1}$ neighbors in $\mathcal{L}_{i-1}(A_{j-1}, B_{j-1}, C_{j-1}, \mathcal{S}_{j-1})$, it follows that $|\mathcal{L}_{i-1}(A_{j-1}, B_{j-1}, C_{j-1}, \mathcal{S}_{j-1}) \cap (N_G[S] - (A_{j-1} \cup B_{j-1} \cup C_{j-1}))| \leqslant 8k\delta n/2^{i-1}$ $= 16k\delta n/2^i$. Hence, at most $16k\delta n/2^i$ vertices are added to level set $\mathcal{L}_i$ at step j when $(A_j, B_j, C_j, \mathcal{S}_j)$ is a separator tuple.

Now, if $(A_j, B_j, C_j, \mathcal{S}_j)$ is a success or failure tuple, then $\mathcal{S}_{j-1} = \mathcal{S}_j$, so no vertices are added to level set $\mathcal{L}_i$ at step j. Since by how $\mathsf{seq}_1(G, \delta)$ was defined, there are $10k\delta \log^2(n)$ separator tuples, therefore at most $160k\delta n \log^2(n)/2^i$ vertices are added to level set $\mathcal{L}_i$. ■

Since a vertex v must be added to level set $\mathcal{L}_i$ before it can be removed from level set $\mathcal{L}_i$, and by Lemma 7.4.23 at most $160k^2\delta^2 n \log^2(n)/2^i$ vertices are added to level set $\mathcal{L}_i$, we get the following corollary, at most $160k^2\delta^2 n \log^2(n)/2^i$ are removed from level set $\mathcal{L}_i$

Corollary 7.4.24. *Let G be a k-creature free graph with n vertices and let $\delta > 1$. In the sequence $\mathsf{seq}_1(G, \delta)$, for all natural numbers i, at most $160k^2\delta^2 n \log^2(n)/2^i$ vertices are removed from level set $\mathcal{L}_i$.*

Let G be a graph with n vertices, let $\delta > 1$, and let $(A_i, B_i, C_i, \mathcal{S}_i)$ be the i^{th} tuple of $\mathsf{seq}_1(G, \delta)$. If there are at least $n/2^c$ vertices that are removed from level set $\mathcal{L}_c$ at step $i-1$, then we say the tuple $(A_i, B_i, C_i, \mathcal{S}_i)$ *drains* level set $\mathcal{L}_c$.

Lemma 7.4.25. *Let G be a graph with n vertices and let $\delta > 1$. If $(A_i, B_i, C_i, \mathcal{S}_i)$ is a success tuple of $\mathsf{seq}_1(G, \delta)$. Then $(A_i, B_i, C_i, \mathcal{S}_i)$ drains at least one level set $\mathcal{L}_j$.*

Proof: Let G be a graph with n vertices, let $\delta > 1$, let $(A_{i-1}, B_{i-1}, C_{i-1}, \mathcal{S}_{i-1})$ and $(A_i, B_i, C_i, \mathcal{S}_i)$ be the $i-1^{th}$ and i^{th} tuples of $\mathsf{seq}_1(G, \delta)$, and assume $(A_i, B_i, C_i, \mathcal{S}_i)$ is a success tuple. There is then some vertex v in $G' = G - (A_{i-1} \cup B_{i-1} \cup C_{i-1})$ that **Refine** was called on to get sets A_i, B_i, and C_i. By the definition of a vertex being good for refining there must be at least one level set, say $\mathcal{L}_j(A_{i-1}, B_{i-1}, C_{i-1}, \mathcal{S}_{i-1})$, such that $|\mathcal{L}_j(A_{i-1}, B_{i-1}, C_{i-1}, \mathcal{S}_{i-1}) \cap N_{G'}[v]| \geqslant n/2^j$. Hence, since $(A_i, B_i, C_i, \mathcal{S}_i)$ is a success tuple and therefore $N_{G'}[v] \subset A_i \cup B_i \cup C_i$, there are at least $n/2^j$ vertices from level set $\mathcal{L}_j$ that are removed at step $i-1$. Therefore $(A_i, B_i, C_i, \mathcal{S}_i)$ drains level set $\mathcal{L}_j$. ■

Lemma 7.4.26. *Let G be a k-creature free graph with n vertices and let $\delta > 1$. Then $\mathsf{seq}_1(G, \delta)$ contains at most n failure tuples and at most $160k^2\delta^2 \log^3(n)$ success tuples.*

Proof: Let G be a k-creature free graph with n vertices and let $\delta > 1$. By Corollary 7.4.24 at most $160k^2\delta^2 n \log^2(n)/2^j$ vertices are removed from level set $\mathcal{L}_j$, hence at most $160k^2\delta^2 \log^2(n)$ tuples of $\mathsf{seq}_1(G, \delta)$ drain level set $\mathcal{L}_j$. By Lemma 7.4.20, for any tuple $(A_i, B_i, C_i, \mathcal{S}_i)$ of $\mathsf{seq}_1(G, \delta)$, no vertex of $G - (A_i \cup B_i \cup C_i)$ belongs to over $\log(n)$ sets of $N_G[\mathcal{S}_i] - (A_i \cup B_i \cup C_i)$, hence no vertices are ever added to level set $\mathcal{L}_j$ for $j > \log(n)$, and therefore no tuple of $\mathsf{seq}_1(G, \delta)$ will ever drain a level set $\mathcal{L}_j$ for $j > \log(n)$. Furthermore, By Lemma 7.4.25 every success tuple of $\mathsf{seq}_1(G, \delta)$ drains at least one level set $\mathcal{L}_i$. It follows that there are at most $160k^2\delta^2 n \log^3(n)$ success tuples of $\mathsf{seq}_1(G, \delta)$.

Now, consider the i^{th} and $i+1^{th}$ elements of $\mathsf{seq}_1(G, \delta)$, $(A_i, B_i, C_i, \mathcal{S}_i)$ and $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ and assume that $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a failure tuple. Let v be the vertex

of $G' = G-(A_i \cup B_i \cup C_i)$ that **Refine** was called on to get the tuple $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$. It follows that $v \notin (A_i \cup B_i \cup C_i)$, but $v \in (A_{i+1} \cup B_{i+1} \cup C_{i+1})$. Additionally, if $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is the n^{th} failure tuple, then we must have that $(A_{i+1} \cup B_{i+1} \cup C_{i+1}) = V(G)$. in this case, this forces any tuples after $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ in $\mathsf{seq}_1(G, \delta)$ to be separator tuples. ■

Corollary 7.4.27. *Let G be a k-creature free graph with $n \geqslant 2$ vertices and μ minimal separators, let $\delta > 1$, and let $(A, B, C, \mathcal{S})$ be the last tuple of $\mathsf{seq}_1(G, \delta)$. Then G has at least $\mu/(4(3n^{(k+1)})^{160k^2\delta^2 \log^3(n)+2})$ minimal separators that are consistent with A, B, and C.*

Proof: Let G be a k-creature free graph with $n \geqslant 2$ vertices and μ minimal separators, let $\delta > 1$, and let $(A, B, C, \mathcal{S})$ be the last tuple of $\mathsf{seq}_1(G, \delta)$. Consider the i^{th} and $i + 1^{th}$ elements of $\mathsf{seq}_1(G, \delta)$, $(A_i, B_i, C_i, \mathcal{S}_i)$ and $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$. Assume that there are μ' minimal separators of G that are consistent with A_i, B_i, and C_i. By Lemma 7.4.13 it follows that if $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a failure tuple, then there are at least $\mu'(1 - 1/n)$ minimal separators of G that are consistent with A_{i+1}, B_{i+1}, and C_{i+1} and if $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a success tuple, then there are at least $\mu'(1/3n^{k+1})$ minimal separators of G that are consistent with A_{i+1}, B_{i+1}, and C_{i+1}. Furthermore, if $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a separator tuple, then $A_i = A_{i+1}$, $B_i = B_{i+1}$ and $C_i = C_{i+1}$, and so there are μ' minimal separators of G that are consistent with A_{i+1}, B_{i+1} and C_{i+1}.

Now, the first tuple, $(A_1, B_1, C_1, \mathcal{S}_1)$, of $\mathsf{seq}_1(G, \delta)$ was chosen so that at least μ/n^2 minimal separators of G agree with A_1, B_1, and C_1. Furthermore, by Lemma 7.4.26 there are at most n failure tuples and at most $160k^2\delta^2 \log^3(n)$ success tuples in $\mathsf{seq}_1(G, \delta)$. It then follows that there are at least $\frac{\mu(1-1/n)^n}{n^2(3n^{(k+1)})^{(160k^2\delta^2 \log^3(n))}} \geqslant \frac{\mu}{4(3n^{(k+1)})^{160k^2\delta^2 \log^3(n)+2}}$ (using the fact that $(1 - 1/n)^n > 1/4$ when $n \geqslant 2$) minimal separators of G that are consistent with A, B, and C. ■

Corollary 7.4.28. *Let G be a k-creature free graph with n vertices, let $\delta > 1$, and let $(A, B, C, \mathcal{S})$ be the last tuple of $\mathsf{seq}_1(G, \delta)$. Then $G[A \cup B]$ has at most $160k^2\delta^2 \log^3(n)+2$ components.*

Proof: Let G be a k-creature free graph with n vertices, let $\delta > 1$, and let $(A_1, B_1, C_1, \mathcal{S}_1)$, $(A_i, B_i, C_i, \mathcal{S}_i)$, $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$, and $(A, B, C, \mathcal{S})$ be the 1^{st}, i^{th}, $i+1^{th}$, and last elements of $\mathsf{seq}_1(G, \delta)$ respectively. If $(A_{i+1}, B_{i+1}, C_{i+1}, \mathcal{S}_{i+1})$ is a separator tuple then $A_{i+1} = A_i$ and $B_{i+1} = B_i$, if it is a failure tuple then $A_{i+1} = A_i$ and $B_{i+1} = B_i$ by Lemma 7.4.13, and if it is a success tuple then $G[A_{i+1} \cup B_{i+1}]$ has at most one more component then $G[A_i \cup B_i]$ by Lemma 7.4.13. Since, by Lemma 7.4.26 there are at most $160k^2\delta^2 \log^3(n)$ success tuples in $\mathsf{seq}_1(G, \delta)$ and since A_1 and B_1 both have one component, we conclude that $G[A \cup B]$ has at most $160k^2\delta^2 \log^3(n) + 2$ components. ■

We are now ready to prove Lemma 7.4.14.

Proof: [Proof of Lemma 7.4.14] Let G be a k-creature free graph (assume $k \geqslant 2$) with $n \geqslant 2$ vertices and μ minimal separators, let $\delta > 1$, and let $(A, B, C, \mathcal{S})$ be the last tuple of $\mathsf{seq}_1(G, \delta)$. Let the sub-list $\mathcal{S}' \subset \mathcal{S}$ be the anti-complete sub-list of size $\log(n)+1$ promised by Lemma 7.4.22, we will show that A, B, C, and $\mathcal{S}'$ satisfy the conclusions of Lemma 7.4.14. Property (i) is established by Lemma 7.4.19, property (ii) is established by Corollary 7.4.27, property (iii) is established by Corollary 7.4.28, and property (iv) is established by Lemma 7.4.20 combined with the fact that $|\mathcal{S}'| = \log(n) + 1$. ■

Separating Balanced Separators

Recall that our final goal of this section is to prove Lemma 7.4.6, that is, to take a k-creature free graph G with n vertices and to find vertex sets A, B, and C such that no component of $G - (A \cup B \cup C)$ contains over n/δ vertices and a large fraction of minimal

separators of G are consistent with A, B, and C. We achieve this in this subsubsection by taking the output from Lemma 7.4.14, $\mathcal{S}, A, B$, and C, and enriching A, B, and C to get A', B', and C' so that for each $S_i, S_j \in \mathcal{S}$ and for each vertex $v_i \in S_i$ and $v_j \in S_j$, no component of $G' = G-(A' \cup B' \cup C' \cup V)$ contains both a vertex from $N_{G'}[v_i]$ and $N_{G'}[v_j]$ where $V = N_G[v_i] \cap N_G[v_j]$. We will show later on using Lemma 7.4.37 that this property is enough to ensure that no component of $G - (A' \cup B' \cup C')$ has over n/δ vertices and prove Lemma 7.4.6.

We begin by showing that sets A, B, and C can be enriched so that $N_G[v_i]-(A \cup B \cup C \cup V)$ and $N_G[v_j]-(A \cup B \cup C \cup V)$ are anti-complete (where $V = N_G[v_i] \cap N_G[v_j]$), then they can be further enriched so that $N_G[v_i]-(A \cup B \cup C \cup V)$ and $N_G[v_j]-(A \cup B \cup C \cup V)$ are far apart in $G-(A \cup B \cup C \cup V)$, then finally they can be enriched so that $N_G[v_i]-(A \cup B \cup C \cup V)$ and $N_G[v_j] - (A \cup B \cup C \cup V)$ are in different components of $G - (A \cup B \cup C \cup V)$, which finally gives us the sets A', B', and C' from the previous paragraph.

Since v_i and v_j are anti-complete, if we set $V = N_G[v_i] \cap N_G[v_j]$, then notice that $(v_i, N_G(v_i) - V, N_G(v_j) - V, v_j)$ is nearly a k-creature, it only lacks the requirement of a semi-induced matching of size k between $N_G(v_i) - V, N_G(v_j) - V$. This is an important observation which we use in our "enrichment" process of A, B, and C and motivates the following definition of a *pre-creature*, (Y_1, X_1, X_2, Y_2), which is just a k-creature but without the requirement of a semi-induced matching of size k between X_1 and X_2.

Definition 7.4.29. Let G be a graph. A four-tuple (Y_1, X_1, X_2, Y_2) of vertex sets in G is called a *pre-creature* if the following conditions are satisfied:

(i) $G[Y_1]$ and $G[Y_2]$ are connected.

(ii) Y_1 is anti-complete with $X_2 \cup Y_2$ and Y_2 is anti-complete with $X_1 \cup Y_1$.

(iii) Y_1 dominates and is disjoint from X_1 and Y_2 dominates and is disjoint from X_2.

Similar to the methods of subsubsection 7.4.1, the "enrichment" process of A, B, and C used to get the sets A', B', and C' which satisfy Lemma 7.4.6 works by taking a carefully chosen vertex and using the **Refine** function on it. We will use the next lemma to guide us on picking a vertex that will be a good choice to use the **Refine** function on.

Lemma 7.4.30. *Let G be a k-creature free graph and let (Y_1, X_1, X_2, Y_2) be a pre-creature of G where $X_1 \neq \emptyset$. Then there exists a vertex $v \in X_1$ such that v dominates at least $(1/k)|N_G(X_1) \cap X_2|$ vertices of $N_G(X_1) \cap X_2$.*

Proof: Let G be a k-creature free graph and let (Y_1, X_1, X_2, Y_2) be a pre-creature of G where $X_1 \neq \emptyset$. Let $D_1 \subset X_1$ be a minimal subset of X_1 that dominates $N_G(X_1) \cap X_2$. Assume for a contradiction that $|D_1| \geqslant k$. Since D_1 is minimal, it follows that for each $d_i^1 \in D_1$ there is a $d_i^2 \in N_G(X_1) \cap X_2$ such that d_i^1 is the only vertex of D_1 that is neighbors with d_i^2. Let D_2 denote this set of d_i^2's. Then since (Y_1, X_1, X_2, Y_2) is a pre-creature, (Y_1, D_1, D_2, Y_2) is a k'-creature for some $k' \geqslant k$, a contradiction.

Thus, we may assume that $|D_1| < k$. Hence, at least one vertex $v \in D_1$ must have the property that v dominates at least $(1/k)|N_G(X_1) \cap X_2|$ vertices of $N_G(X_1) \cap X_2$. ■

Let v_i and v_j be anti-complete vertices. We now prove a lemma that shows we can enrich A, B, and C so that $N_G[v_i] - (A \cup B \cup C \cup V)$ and $N_G[v_j] - (A \cup B \cup C \cup V)$ are anti-complete (where $V = N_G[v_i] \cap N_G[v_j]$). We will actually need to prove a slightly more general lemma that is stated in terms of pre-creatures so that it can be used for induction in a later lemma.

Lemma 7.4.31. *Let G be a k-creature free graph with n vertices, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, and let (Y_1, X_1, X_2, Y_2) be a pre-creature of G. Then there exists sets $A', B', C' \subseteq V(G)$ with the following properties:*

(i) G *has at least* $\mu/(4(3n^{(k+1)})^{k\log(n)})$ *minimal separators that are consistent with* $A', B',$ *and* C'.

(ii) The number of components of $G[A' \cup B']$ *is at most the number of components of* $G[A \cup B]$ *plus* $k\log(n)$.

(iii) $X_1 - (A' \cup B' \cup C')$ *is anti-complete with* $X_2 - (A' \cup B' \cup C')$ *in* G.

(iv) $A \subseteq A'$, $B \subseteq B'$, *and* $C \subseteq C'$.

In order to prove Lemma 7.4.31, we will study the following sequence produced from a k-creature free graph G with n vertices along with sets $A, B, C \subseteq V(G)$ and pre-creature $P = (Y_1, X_1, X_2, Y_2)$, which we denote by $\mathsf{seq}_2(G, A, B, C, P)$. $\mathsf{seq}_2(G, A, B, C, P)$ is a sequence of tuples (A_i, B_i, C_i) with $A_i, B_i, C_i \subseteq V(G)$.

We now define the sequence $\mathsf{seq}_2(G, A, B, C, P)$. For the base case of this sequence we set $A_1 = A$, $B_1 = B$, and $C_1 = C$. We recursively define this sequence as follows. Given (A_i, B_i, C_i) let $X_1' = X_1 - (A_i \cup B_i \cup C_i)$, $X_2' = X_2 - (A_i \cup B_i \cup C_i)$, and $G' = G - (A_i \cup B_i \cup C_i)$. If $N_{G'}(X_1') \cap X_2' = \emptyset$ then we terminate the sequence (so (A_i, B_i, C_i) is the last tuple of the sequence), else since (Y_1, X_1, X_2, Y_2) is a pre-creature of G, (Y_1, X_1', X_2', Y_2) is a pre-creature of G (and $X_1' \neq \emptyset$ since $N_{G'}(X_1') \cap X_2' \neq \emptyset$) and so Lemma 7.4.30 guarantees the existence of $v \in X_1'$ such that v dominates at least $(1/k)|N_{G'}(X_1') \cap X_2'|$ vertices of $N_{G'}(X_1') \cap X_2'$ (Note that $N_{G'}(X_1') \cap X_2' = N_G(X_1') \cap X_2'$). We then set $A_{i+1}, B_{i+1}, C_{i+1} = \mathbf{Refine}(G, v, A_i, B_i, C_i)$. If $\mathbf{Refine}(G, v, A_i, B_i, C_i)$ is a failure refinement then we call $(A_{i+1}, B_{i+1}, C_{i+1})$ a *failure tuple*, else $\mathbf{Refine}(G, v, A_i, B_i, C_i)$ is a success refinement and we call $(A_{i+1}, B_{i+1}, C_{i+1})$ a *success tuple*.

Lemma 7.4.32. *Let* G *be a* k*-creature free graph with* n *vertices, let* $A, B, C \subset V(G)$, *and let* P *be a pre-creature of* G. $\mathsf{seq}_2(G, A, B, C, P)$ *has at most* n *failure tuples and at most* $k\log(n)$ *success tuples.*

Proof: Let G be a k-creature free graph with n vertices, let $A, B, C \subset V(G)$, and let $P = (Y_1, X_1, X_2, Y_2)$ be a pre-creature of G. Let (A_i, B_i, C_i) and $(A_{i+1}, B_{i+1}, C_{i+1})$ be the i^{th} and $i + 1^{th}$ tuples of $\mathsf{seq}_2(G, A, B, C, P)$. We can see from the definition of **Refine** and seq_2 that $(A_i \cup B_i \cup C_i) \subset (A_{i+1} \cup B_{i+1} \cup C_{i+1})$ where the containment is strict, hence if $i = n$ then $(A_{i+1} \cup B_{i+1} \cup C_{i+1}) = V(G)$. Hence, by the definition of seq_2 $(A_{i+1} \cup B_{i+1} \cup C_{i+1})$ would be the final tuple of $\mathsf{seq}_2(G, A, B, C, P)$. Therefore, there are at most $n + 1$ tuples of $\mathsf{seq}_2(G, A, B, C, P)$, hence there are at most n failure tuples in $\mathsf{seq}_2(G, A, B, C, P)$ (recall the first tuple it neither a failure nor success tuple).

Now, assume $(A_{i+1}, B_{i+1}, C_{i+1})$ is a success tuple, let $X_1^i = X_1 - (A_i \cup B_i \cup C_i)$, $X_2^i = X_2 - (A_i \cup B_i \cup C_i)$, $G^i = G - (A_i \cup B_i \cup C_i)$, $X_1^{i+1} = X_1 - (A_{i+1} \cup B_{i+1} \cup C_{i+1})$, $X_2^{i+1} = X_2 - (A_{i+1} \cup B_{i+1} \cup C_{i+1})$, and $G^{i+1} = G - (A_{i+1} \cup B_{i+1} \cup C_{i+1})$. If v is the vertex of X_1^i that refine is called on to get the sets A_{i+1}, B_{i+1}, and C_{i+1}, then by how v was selected, v dominates at least $(1 - 1/k)|N_{G^i}(X_1^i) \cap X_2^i|$ vertices of $|N_{G^i}(X_1^i) \cap X_2^i|$. Then by Lemma 7.4.13 $N_G[v] \subseteq A_{i+1} \cup B_{i+1} \cup C_{i+1}$ and it follows that $(1 - 1/k)|N_{G^i}(X_1^i) \cap X_2^i|$ $\geqslant |N_{G^{i+1}}(X_1^{i+1}) \cap X_2^{i+1}|$. Hence if $(A_{i+1}, B_{i+1}, C_{i+1})$ is the $(k \log(n))^{th}$ success tuple, then $|N_{G^{i+1}}(X_1^{i+1}) \cap X_2^{i+1}|$ is at most $n(1 - 1/k)^{k \log(n)} \leqslant n/e^{log(n)} < 1$ (recall for our definition of $\log(n)$ that $\log(n) \geqslant \ln(n)$), which implies $|N_{G^{i+1}}(X_1^{i+1}) \cap X_2^{i+1}| = 0$. ■

Corollary 7.4.33. *Let G be a k-creature free graph with n vertices, let $A, B, C \subset V(G)$, let G have μ minimal separators that are consistent with A, B, and C, and let P be a pre-creature of G. Let $(A'$, B', $C')$ be the final tuple of $\mathsf{seq}_2(G, A, B, C, P)$. Then G has at least $\mu/(4(3n^{(k+1)})^{k \log(n)})$ minimal separators that are consistent with A', B', and C'.*

Proof: Let G be a k-creature free graph with n vertices, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, and let P be a pre-creature of G. Let $(A'$, B', $C')$ be the final tuple of $\mathsf{seq}_2(G, A, B, C, P)$. Consider the i^{th} and $i + 1^{th}$ tuples of $\mathsf{seq}_2(G, A, B, C, P)$, $(A_i$, B_i, $C_i)$ and $(A_{i+1}$, B_{i+1}, $C_{i+1})$, and

assume there are μ' minimal separators of G that are consistent with A_i, B_i, and C_i. If $(A_{i+1}, B_{i+1}, C_{i+1})$ is a failure tuple, then by Lemma 7.4.13 there are at least $(1-1/n)\mu'$ minimal separators that are consistent with A_{i+1}, B_{i+1}, and C_{i+1}, and if $(A_{i+1}, B_{i+1}, C_{i+1})$ is a success tuple, then by Lemma 7.4.13 there are at least $(1/3n^{k+1})\mu'$ minimal separators that are consistent with A_{i+1}, B_{i+1}, and C_{i+1}. Since by Lemma 7.4.32 there are at most n failure tuples and $k\log(n)$ success tuples, it follows that there are at least $\frac{\mu(1-1/n)^n}{(3n^{k+1})^{k\log(n)}} \leqslant \frac{\mu}{4(3n^{k+1})^{k\log(n)}}$ minimal separators of G that are consistent with A', B', and C' (using the fact that $(1-1/n)^n \geqslant 1/4$ for $n \geqslant 2$). ■

Corollary 7.4.34. *Let G be a k-creature free graph with n vertices, let $A, B, C \subset V(G)$, let P be a pre-creature of G, and let (A', B', C') be the final tuple of $\mathsf{seq}_2(G, A, B, C, P)$. Then $G[A' \cup B']$ has at most $k\log(n)$ more components then $G[A \cup B]$.*

Proof: Let G be a k-creature free graph with n vertices, let $A, B, C \subset V(G)$, let P be a pre-creature of G, and let (A', B', C') be the final tuple of $\mathsf{seq}_2(G, A, B, C, P)$. Consider the i^{th} and $i+1^{th}$ tuples of $\mathsf{seq}_2(G, A, B, C, P)$, (A_i, B_i, C_i) and $(A_{i+1}, B_{i+1}, C_{i+1})$. If $(A_{i+1}, B_{i+1}, C_{i+1})$ is a failure tuple, then by Lemma 7.4.13 $A_i = A_{i+1}$ and $B_i = B_{i+1}$. If $(A_{i+1}, B_{i+1}, C_{i+1})$ is a success sequence, then by Lemma 7.4.13 $G[A_{i+1} \cup B_{i+1}]$ has at most one more component than $G[A_i \cup B_i]$. Since by Lemma 7.4.32 there are at most $k\log(n)$ success tuples, it follows that $G[A' \cup B']$ has at most $k\log(n)$ more components than $G[A \cup B]$. ■

We are now ready to prove Lemma 7.4.31.

Proof: [Proof of Lemma 7.4.31] Let G be a k-creature free graph with n vertices, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, and let $P = (Y_1, X_1, X_2, Y_2)$ be a pre-creature of G. Let (A', B', C') be the final tuple of $\mathsf{seq}_2(G, A, B, C, P)$, we will show the set A', B', and C' satisfy the conclusions of Lemma 7.4.31. Property (i) is established by Corollary 7.4.33, property (ii) is es-

tablished by Corollary 7.4.34, property (iii) is established by the termination condition of $\mathsf{seq}_2(G, A, B, C, P)$, and property (iv) is established by the facts in the first tuple of $\mathsf{seq}_2(G, A, B, C, P)$, (A_1, B_1, C_1), that $A_1 = A$, $B_1 = B$, and $C_1 = C$ and that for the i^{th} and $i + 1^{th}$ tuples of $\mathsf{seq}_2(G, A, B, C, P)$, $(A_i,\ B_i,\ C_i)$ and $(A_{i+1},\ B_{i+1},\ C_{i+1})$, that $A_i \subseteq A_{i+1}$, $B_i \subseteq B_{i+1}$, and $C_i \subseteq C_{i+1}$. ■

Let v_i and v_j be anti-complete vertices. Lemma 7.4.31 showed how we can enrich A, B, and C so that $N_G[v_i] - (A \cup B \cup C \cup V)$ and $N_G[v_j] - (A \cup B \cup C \cup V)$ are anti-complete (where $V = N_G[v_i] \cap N_G[v_j]$). In the next lemma we show how to further enrich A, B, and C so that $N_G[v_i] - (A \cup B \cup C \cup V)$ and $N_G[v_j] - (A \cup B \cup C \cup V)$ are far apart in $G - (A \cup B \cup C \cup V)$. Because we will need to apply this lemma in an inductive proof later on, we will need to prove something slightly more general that what was just stated.

Lemma 7.4.35. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, let i be a natural number, let Y_1 and Y_2 be anti-complete connected subsets of $V(G)$, and let $Z \subseteq V(G)$ be disjoint from $N(Y_1) \cap N(Y_2)$. Then there exists sets $A', B', C' \subseteq V(G)$ with the following properties:*

(i) G has at least $\mu/(12n^{(k+1)})^{ik\log(n)}$ minimal separators that are consistent with A', B', and C'.

(ii) The number of components of $G[A' \cup B']$ is at most $ik\log(n)$ more than the number of components of $G[A \cup B]$.

(iii) Every path from Y_1 to Y_2 in $G[Z \cup Y_1 \cup Y_2]$ of length less than $i + 4$ contains an internal vertex from $(A' \cup B' \cup C')$.

(iv) $A \subseteq A'$, $B \subseteq B'$, and $C \subseteq C'$.

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, let i be a natural number, let Y_1 and Y_2

be anti-complete connected subsets of $V(G)$, and let Z be a subset of $V(G)$ disjoint from $V = N_G(Y_1) \cap N_G(Y_2)$. We now prove by induction on i that there exists sets A', B', C' that satisfy properties (i)-(iv) of this lemma. For the base case, when $i = 0$, since Y_1 and Y_1 are anti-complete and Z is disjoint from $N_G(Y_1) \cap N_G(Y_2)$, every path from Y_1 to Y_2 in $G[Y_1 \cup Y_2 \cup Z]$ must be of length at least 4, so $A' = A$, $B' = B$ and $C' = C$ satisfy the conclusion of the lemma. Assume now that $0 < i$ and that the conclusion of the lemma holds for all $i' < i$. We prove that it also holds for i.

We apply the inductive hypothesis to G, A, B, C, Y_1, Y_2, Z, and $i - 1$ to get sets A', B', and C' that satisfy properties (i)-(iv) of this lemma (for the natural number $i-1$). In particular, by property (iii) every path from Y_1 to Y_2 in $G[Y_1 \cup Y_2 \cup Z]$ of length less than $(i-1)+4$ contains an internal vertex from $(A' \cup B' \cup C')$. Now, let ℓ be the length of a shortest path from Y_1 to Y_2 in $G[Y_1 \cup Y_2 \cup Z]$ that does not contain an internal vertex from $(A' \cup B' \cup C')$ and let $\mathcal{P}$ denote the set of all such shortest paths. If $\ell \geqslant i + 4$ then the sets A', B', and C' satisfy properties (i)-(iv) of the lemma (now for the natural number i) and we are done. So we may assume $\ell = (i-1)+4$. Since $i \geqslant 1$, each path in $\mathcal{P}$ contains at least two internal vertices. Let X_1 denote the set of vertices that occur as the first internal vertex (the vertex closest to Y_1) in a path of $\mathcal{P}$, let X_2 denote the set of vertices that occur as the second internal vertex in a path of $\mathcal{P}$, and let $X_{\geqslant 3}$ denote the set of vertices that occur as the third or later internal vertex of some path in $\mathcal{P}$. It is straightforward to verify that since $\mathcal{P}$ is a set of shortest paths it holds that X_1, X_2, and $X_{\geqslant 3}$ are disjoint and furthermore $P = (Y_1, X_1, X_2, Y_2 \cup X_{\geqslant 3})$ is a pre-creature of G. In particular one can verify that $G[Y_1]$ and $G[Y_2 \cup X_{\geqslant 3}]$ are connected, Y_1 is anti-complete with $X_2 \cup (Y_2 \cup X_{\geqslant 3})$ and $(Y_2 \cup X_{\geqslant 3})$ is anti-complete with Y_1, and Y_1 dominates X_1 and $(Y_2 \cup X_{\geqslant 3})$ dominates X_2, which establishes that P is a pre-creature.

Since P is a pre-creature of G we may apply Lemma 7.4.31 to G, A', B', C', and P to get sets A'', B'', and C'' that satisfy properties (i)-(iv) of Lemma 7.4.31. We now verify

that the sets A'', B'', and C'' satisfy conditions (i)-(iv) of this lemma.

We first verify property (iii) holds. Let Q be a shortest path from from Y_1 to Y_2 in $G[Y_1 \cup Y_2 \cup Z]$ that has no internal vertex contained in $(A'' \cup B'' \cup C'')$, which implies it has no internal vertex contained in $(A' \cup B' \cup C')$. If this path is of length $(i-1)+4$ then Q is a path of $\mathcal{P}$, but since $X_1 - (A'' \cup B'' \cup C'')$ is anti-complete with $X_2 - (A'' \cup B'' \cup C'')$ this is impossible. So it must be that Q has length at least $i+4$ which establishes property (iii).

To verify property (i), note that by the inductive hypothesis G has at least $\mu' = \mu/(12n^{(k+1)})^{(i-1)k\log(n)}$ minimal separators that are consistent with A', B', and C', and by property (i) of Lemma 7.4.31 G has at least $\mu'/4(3n^{(k+1)})^{k\log(n)} > \mu'/(12n^{(k+1)})^{k\log(n)}$ minimal separators that are consistent with A'', B'' and C''. It follows that G has at least $\mu/(12n^{(k+1)})^{ik\log(n)}$ minimal separators that are consistent with A'', B'', and C'' which established property (i).

To verify property (ii) note that by the inductive hypothesis $G[A' \cup B']$ has at most $(i-1)k\log(n)$ more components than $G[A \cup B]$ and by property (ii) of Lemma 7.4.31 $G[A'' \cup B'']$ has at most $k\log(n)$ more components than $G[A' \cup B']$. It follows that $G[A'' \cup B'']$ has at most $ik\log(n)$ more components than $G[A \cup B]$ which established property (ii).

Lastly, to verify property (iv) note that by the inductive hypothesis $A \subseteq A'$, $B \subseteq B'$ and $C \subseteq C'$ and by Lemma 7.4.31 $A' \subseteq A''$, $B' \subseteq B''$ and $C' \subseteq C''$. It then follows that $A \subseteq A''$, $B \subseteq B''$ and $C \subseteq C''$ which established property (iv). ■

Let v_i and v_j be anti-complete vertices. Lemma 7.4.35 showed how we can enrich A, B, and C so that $N_G[v_i] - (A \cup B \cup C \cup V)$ and $N_G[v_j] - (A \cup B \cup C \cup V)$ are far apart in $G - (A \cup B \cup C \cup V)$. This next lemma shows how we can further enrich A, B, and C so that no component of $G - (A \cup B \cup C \cup V)$ contains both a vertex from $N_G[v_i] - (A \cup B \cup C \cup V)$ and from $N_G[v_j] - (A \cup B \cup C \cup V)$. To make the proof easier,

we will prove something slightly more general that what was just stated.

Lemma 7.4.36. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, let Y_1 and Y_2 be anti-complete connected subsets of $V(G)$, and let $Z \subseteq V(G)$ be disjoint from $N_G(Y_1) \cap N_G(Y_2)$. Then there exists sets $A', B', C' \subseteq V(G)$ with the following properties:*

(i) G has at least $\mu/(12n^{(k+1)})^{2k\log(n)\log(|Z|)}$ minimal separators that are consistent with A', B', and C'.

(ii) The number of components of $G[A' \cup B']$ is at most $2k\log(n)\log(|Z|)$ more than the number of components of $G[A \cup B]$.

(iii) Every path from Y_1 to Y_2 in $G[Y_1 \cup Y_2 \cup Z]$ has an internal vertex that belongs to $(A' \cup B' \cup C')$.

(iv) $A \subseteq A'$, $B \subseteq B'$, and $C \subseteq C'$.

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$, let G have μ minimal separators that are consistent with A, B, and C, let Y_1 and Y_2 be anti-complete connected subsets of $V(G)$, and let $Z \subseteq V(G)$ be disjoint from $N_G(Y_1) \cap N_G(Y_2)$.

We prove the statement of the lemma by induction on $|Z|$. If $Z = \emptyset$ then A, B and C already satisfy the conclusion of the lemma. Therefore, assume that $Z \neq \emptyset$ and that the statement of the lemma holds for sets Z' such that $|Z'| < |Z|$. We apply Lemma 7.4.35 to G, A, B, C, $i = 2$, Y_1, Y_2 and Z to get sets A', B' and C' that satisfy properties *(i)-(iv)* of Lemma 7.4.35. We set $Z' = Z - (A' \cup B' \cup C')$ and $G' = G[Y_1 \cup Y_2 \cup Z']$. From property (iii) of Lemma 7.4.35 it follows that every path from Y_1 to Y_2 in G' has length at least six. If there does not exist a path from Y_1 to Y_2 in G' then using properties *(i), (ii)* and *(iv)* of Lemma 7.4.35 it can be verified that sets A', B', and C' satisfy the conclusions of this lemma. Therefore we may assume that Y_1 and Y_2 are in the same component of G'.

We define $X_1 = N_{G'}(Y_1)$, $X_2 = N_{G'}(Y_1 \cup X_1)$, $X_1' = N_{G'}(Y_2)$, and $X_2' = N_{G'}(Y_2 \cup X_1')$, since every path from Y_1 to Y_2 in G' has length at least six, X_1, X_2, X_1', and X_2' are mutually disjoint. Furthermore, observe that $X_2 = N_{G'}(X_1) - Y_1$ and similarly that $X_2' = N_{G'}(X_1') - Y_2$. By definition of X_1, X_2, X_1', and X_2' it holds that every connected component C of $G' - (Y_1 \cup X_1 \cup X_2 \cup Y_2 \cup X_1' \cup X_2')$ satisfies $N_{G'}(C) \subseteq X_2 \cup X_2'$ (any vertex outside of Y_1 that has a neighbor in Y_1 would belong to X_1 and any vertex outside of $Y_1 \cup X_1$ that has a neighbor in X_1 would belong to X_2, similar statements hold for X_1' and X_2'). Let Z_1 contain $X_1 \cup X_2$ as well as the union of the vertex sets of all connected components, C, of $G' - (Y_1 \cup X_1 \cup X_2 \cup Y_2 \cup X_1' \cup X_2')$ that satisfy $\emptyset \subset N_{G'}(C) \subseteq X_2$. Similarly, let Z_2 contain $X_1' \cup X_2'$ as well as the union of the vertex sets of all connected components, C, of $G' - (Y_1 \cup X_1 \cup X_2 \cup Y_2 \cup X_1' \cup X_2')$ that satisfy $\emptyset \subset N_{G'}(C) \subseteq X_2'$. Observe that no component of $G' - (Y_1 \cup X_1 \cup X_2 \cup Y_2 \cup X_1' \cup X_2')$ is added to both Z_1 and to Z_2. Since every path from Y_1 to Y_2 in G' has length at least six, it follows that $X_1 \cup X_2$ and $X_1' \cup X_2'$ are disjoint. Therefore $Z_1 \cap Z_2 = \emptyset$. Without loss of generality $|Z_1| \leqslant |Z_2|$. Hence, since $Z_1 \cup Z_2 \subseteq Z' \subseteq Z$ and it follows that $|Z_1| \leqslant |Z|/2 < |Z|$.

We define Y_2' to be the connected component of $G' - (Y_1 \cup Z_1)$ that contains Y_2. We claim that $Q = Y_1 \cup Z_1 \cup Y_2'$ is equal to the component, T, of G' that contains $Y_1 \cup Y_2$ (recall that by the discussion of the second paragraph, we can assume Y_1 and Y_2 belong to the same component of G'). First, to see that that every vertex of Q belongs to T note that Y_2' to be the connected component of $G' - (Y_1 \cup Z_1)$ that contains Y_2, hence $Y_2' \subseteq C$, Z_1 is a connected set and since $X_1 \subseteq Z_1$, Z_1 has neighbors in Y_1, hence $Z_1 \subseteq Y_1$, and clearly $Y_1 \subseteq T$. Next, we verify that no vertex of G' outside of Q belongs to T. Let v be a vertex of G' outside of Q had a neighbor in T. As noted before, Y_2' contains X_1' and X_2' and by definition Z_1 contains X_1 and X_2, so we may assume that v belongs to a component of $G' - (Y_1 \cup X_1 \cup X_2 \cup Y_2 \cup X_1' \cup X_2')$, hence, by the definition of Z_1, we can see that this implies that if v has a neighbor in Z_1 then v must belong to Z_1. Next, if v has

a neighbor in Y_2', then, by definition of Y_2', v would be apart of Y_2', so this is impossible. Lastly, if v had a neighbor with Y_1 then v would be in $X_1 \subseteq Z_1$ and therefore in T, so this is impossible. It now follows that $Q = Y_1 \cup Z_1 \cup Y_2'$ is equal to the component, T, of G' that contains $Y_1 \cup Y_2$.

Since Y_1, Z_1, and Y_2' are disjoint, it follows that these sets partition the component that $Y_1 \cup Y_2$ belong to in G'. We have that Y_2' is connected, that Y_1 and Y_2' are anti-complete (because $X_1 \subseteq Z_1$), and that Y_1 and Y_2' do not have common neighbors in Z_1 (because $X_1 \subseteq Z_1$ and $X_2 \subseteq Z_2$). We may therefore apply the induction hypothesis to G, A', B', C', Y_1, Y_2' and Z_1. Let A'', B'' and C'' be the sets that satisfy properties *(i)-(iv)* of this lemma when applied to G, A', B', C', Y_1, Y_2' and Z_1. We prove that A'', B'' and C'' satisfy the conclusion of the lemma (when applied to G, A, B, C, Y_1, Y_2 and Z).

We first prove property *(iii)*, that every path P from Y_1 to Y_2 in $G[Y_1 \cup Y_2 \cup Z]$ contains an internal vertex in $(A'' \cup B'' \cup C'')$. Suppose for contradiction that this is not the case, and let P be a path from Y_1 to Y_2 in $G[Y_1 \cup Y_2 \cup Z]$ with internal vertices disjoint from $(A'' \cup B'' \cup C'')$. Without loss of generality we assume the first vertex of P is in Y_1, the last is in Y_2, and all internal vertices of P are in $Z - (Y_1 \cup Y_2)$. Since P is internally vertex disjoint from $(A'' \cup B'' \cup C'')$ and by the induction hypothesis $A' \subseteq A''$, $B' \subseteq B''$, and $C' \subseteq C''$ we have that P is internally vertex disjoint from $(A' \cup B' \cup C')$, and that therefore P is a path from Y_1 to Y_2 in G'.

The first vertex of P lies in Y_1, while the last vertex of P is in Y_2, which is in in Y_2'. Define v to be the first vertex on P in Y_2' and let P' be the sub-path of P that starts in Y_1 and ends in v. We have that P' lies in the component of G' that contains Y_1 and Y_2. We argue that all internal vertices of P' lie in Z_1. Indeed, none of the internal vertices of P' lie in Y_1 (since only the first vertex of P is in Y_1), and none of the internal vertices of P' lie in Y_2' by the choice of v. But, as we claimed just after the definition of Y_2', we have that Y_1, Y_2' and Z_1 partition the component of G' that contain $Y_1 \cup Y_2$ (and therefore P')

and hence all internal vertices of P' are in Z_1. But then P' contradicts property *(iii)* of the inductive hypothesis, namely that every path from Y_1 to Y_2' in $G[Y_1 \cup Y_2' \cup Z_1]$ has an internal vertex that belongs to $(A'' \cup B'' \cup C'')$. We conclude that every path P from Y_1 to Y_2 in $G[Y_1 \cup Y_2 \cup Z]$ contains an internal vertex in $(A'' \cup B'' \cup C'')$, which proves property *(iii)* in the statement of the lemma.

We check the remaining properties of the statement of the Lemma. Property *(iv)* follows from the fact that $A \subseteq A'$, $B \subseteq B'$, $C \subseteq C'$ (by Lemma 7.4.35) and that $A' \subseteq A''$, $B' \subseteq B''$, $C' \subseteq C''$ by the inductive hypothesis.

For property *(i)*, Lemma 7.4.35 yields that the number of separators in G consistent with A', B' and C' in G is at least $\mu/(12n^{(k+1)})^{2k\log(n)}$. The induction hypothesis now yields that there are at least

$$\frac{\mu/(12n^{(k+1)})^{2k\log(n)}}{(12n^{(k+1)})^{2k\log(n)\log(|Z_1|)}} \geqslant \frac{\mu/(12n^{(k+1)})^{2k\log(n)}}{(12n^{(k+1)})^{2k\log(n)\log(|Z|/2)}} = \frac{\mu}{(12n^{(k+1)})^{2k\log(n)\log(|Z|)}}$$

minimal separators in G consistent with A'', B'' and C''.

For property *(ii)* we have that $G[A' \cup B']$ has at most $2k\log n$ more components than $G[A \cup B]$. By the inductive hypothesis $G[A'' \cup B'']$ has at most $2k\log n \log|Z_1|$ more components than $G[A' \cup B']$. However $|Z_1| \leqslant |Z|$ so $\log(|Z_1|) \leqslant \log|Z| - 1$ and therefore $G[A'' \cup B'']$ has at most $2k\log n\log|Z|$ more components than $G[A \cup B]$. This concludes the proof. ■

If A, B, and C and $\mathcal{S}$ are the outputs of Lemma 7.4.14 then Lemma 7.4.36 shows us how we can enrich A, B, and C so that for each $S_i, S_j \in \mathcal{S}$ and for each vertex $v_i \in S_i$ and $v_j \in S_j$, no component of $G' = G - (A' \cup B' \cup C' \cup V)$ contains both a vertex from $N_{G'}[v_i]$ and $N_{G'}[v_j]$ where $V = N_G[v_i] \cap N_G[v_j]$ (A, B, and C are enriched by applying Lemma 7.4.36 for each pair of vertices v_i and v_j). After doing this the next lemma shows us why this is sufficient to guarantee that $G - (A \cup B \cup C)$ has no large component.

Lemma 7.4.37. *Let G be a graph with n vertices, let $\delta > 1$, and let $\mathcal{S}$ be a vertex list of n/δ-balanced separators of G such that no vertex of G belongs to every balanced separator of $\mathcal{S}$. If for every pair $S_i, S_j \in \mathcal{S}$ it holds that no component of $G - (S_i \cap S_j)$ contains a vertex from both $S_i - (S_i \cap S_j)$ and $S_j - (S_i \cap S_j)$, then G has no component of size greater than n/δ.*

Proof: Let G be a graph with n vertices, let $\delta > 1$, let $\mathcal{S}$ be a list of n/δ-balanced separators of G such that no vertex of G belongs to every balanced separator of $\mathcal{S}$, and for every pair $S_i, S_j \in \mathcal{S}$ it holds that no component of $G - (S_i \cap S_j)$ contains a vertex from both $S_i - (S_i \cap S_j)$ and $S_j - (S_i \cap S_j)$. Assume, for a contradiction then G has a component, X, of size greater than n/δ.

Since X is a component of size greater than n/δ, every n/δ-balanced separator must have at least one vertex in X. Let v be a vertex of X that belongs the largest number of sets of $\mathcal{S}$ as possible. By how $\mathcal{S}$ was defined, there is some $S \in \mathcal{S}$ such that $v \notin S$. Let P be a shortest path from v to S and let s be the endpoint of P that belongs to S.

Since v was chosen to be a vertex of X that belongs the largest number of sets of $\mathcal{S}$ as possible, there must be a set $S' \in \mathcal{S}$ such that $v \in S'$ and $s \notin S'$. Since we also have that $v \notin S$, there must be a subpath P^* of P with endpoints s and v' where $v' \in S'$, $v' \neq s$, and no internal vertex of P^* belongs to S' or S (recall P is a shortest path from v to S). It follows that P^* is a path in $G - (S \cap S')$ and therefore since neither v' nor s belong to $S \cap S'$, v' and s are in the same connected component in $G - (S \cap S')$, a contradiction to how $\mathcal{S}$ was defined. ■

We are now ready to prove Lemma 7.4.6. As indicated before, the proof works by taking A, B, and C and $\mathcal{S}$, the outputs of Lemma 7.4.14, then for each $S_i, S_j \in \mathcal{S}$ and for each vertex $v_i \in S_i$ and $v_j \in S_j$, applying Lemma 7.4.36 to v_i and v_j. The graph $G - (A \cup B \cup C)$ along with the vertex list $\mathcal{S}$ will then satisfy the conditions of

Lemma 7.4.37.

Proof: [Proof of Lemma 7.4.6] Let G be a k-creature free graph, $k \geqslant 2$, with $n \geqslant 2$ vertices and μ minimal separators and let $\delta > 1$. Apply Lemma 7.4.14 to G and δ to get sets A, B, and C and an anti-complete list $\mathcal{S}$ that satisfy properties *(i)-(vi)* of Lemma 7.4.14.

We wish to "enrich" A, B, and C so that for each unordered pair of vertices x and y where x belongs to some set of $\mathcal{S}$ and y belong to a different set of $\mathcal{S}$, it will hold that no vertex of $N_G[x]$ and $N_G[y]$ will be in the same component of $G - (A \cup B \cup C \cup (N_G[x] \cap N_G[y])$. To do this we go through each pair of vertices, x, y, and after considering the i^{th} pair we will have vertex sets A^i, B^i and C^i such that for any pair x, y that have previously been considered it holds that no vertex of $N_G[x]$ and $N_G[y]$ will be in the same component of $G - (A \cup B \cup C \cup (N_G[x] \cap N_G[y])$.

More formally, we set $A^0 = A$, $B^0 = B$ and $C^0 = C$. Assume that we have already considered $i-1$ unordered vertex pairs x and y where x belongs to some set of $\mathcal{S}$ and y belong to a different set of $\mathcal{S}$ and we have sets A^{i-1}, B^{i-1} and C^{i-1}. Since $\mathcal{S}$ is an anti-complete vertex list, x and y are anti-complete, so we set $Y_1 = x$, $Y_2 = y$, and $Z = V(G) - (N_G[x] \cap N_G[y])$ and apply Lemma 7.4.36 to A^{i-1}, B^{i-1}, C^{i-1}, Y_1, Y_2, and Z to get the sets A^i, B^i, and C^i which satisfy properties *(i)-(iv)* of Lemma 7.4.36.

Assume that there are ℓ unordered vertex pairs x and y where x belongs to some set of $\mathcal{S}$ and y belong to a different set of $\mathcal{S}$. We show that the sets A^ℓ, B^ℓ, and C^ℓ satisfy properties *(i)-(iii)* of Lemma 7.4.6.

We first show property *(i)* is satisfied. We can see that by properties *(iii)* and *(iv)* of Lemma 7.4.36 that for any pair of vertices x and y where x belongs to some set of $\mathcal{S}$ and y belong to a different set of $\mathcal{S}$, it holds that no vertex of $N_G[x] - (A^\ell \cup B^\ell \cup C^\ell \cup V)$ and $N_G[y] - (A^\ell \cup B^\ell \cup C^\ell \cup V)$ will be in the same component of $G - (A^\ell \cup B^\ell \cup C^\ell \cup V)$ where $V = N_G[x] \cap N_G[y]$. It then follows that for any two distinct sets $S_i, S_j \in N_G(\mathcal{S})$

that no vertex of $S_i - (A^\ell \cup B^\ell \cup C^\ell \cup V')$ and $S_j - (A^\ell \cup B^\ell \cup C^\ell \cup V')$ belong to the same component in $G - (A^\ell \cup B^\ell \cup C^\ell \cup V')$ where $V' = S_i \cap S_j$. Since $A \subseteq A^\ell$, $B \subseteq B^\ell$, and $C \subseteq C^\ell$, if $G' = G - (A^\ell \cup B^\ell \cup C^\ell)$ then it follows from property *(i)* of Lemma 7.4.14 that for all $S \in \mathcal{S}$ $N_G[S] - (A^\ell \cup B^\ell \cup C^\ell)$ is an n/δ-balanced separator of G' and no vertex of G' belongs to all sets of $N_{G'}[\mathcal{S}]$. It follows from Lemma 7.4.37 that no vertex of G' has over n/δ components.

Next we show property *(ii)* and *(iii)* are satisfied. By properties *(ii)* and *(iii)* of Lemma 7.4.14 there are at least $\mu/(4(3n^{(k+1)})^{160k^2\delta^2\log^3(n)+2})$ minimal separators that are consistent with A, B, and C and $G[A \cup B]$ contains at most $260k^2\delta^2\log(n)^3 + 2$ components. Furthermore, by properties *(i)* and *(ii)* of Lemma 7.4.36 if there are μ' minimal separators that are consistent with A^{i-1}, B^{i-1} and C^{i-1} then there are at least $\mu'/(12n^{(k+1)})^{2k\log^2(n)}$ minimal separators that are consistent with A^i, B^i, and C^i and there are at most $2k\log(n)^2$ more minimal separators in $G[A^i \cup B^i]$ than in $G[A^{i-1} \cup B^{i-1}]$. Since each $S \in \mathcal{S}$ has size at most $8k\delta$ and $\mathcal{S}$ has size $\log(n)+1$, ℓ must be less than $(8k\delta\log(n))^2$. Hence G has at least

$$\frac{\mu}{4(3n^{(k+1)})^{160k^2\delta^2\log^3(n)+2})(12n^{(k+1)})^{(2k\log^2(n))(8k\delta\log(n))^2}}$$

$$\geqslant \frac{\mu}{((12n^{(k+1)})^{160k^2\delta^2\log^3(n)+2})(12n^{(k+1)})^{(128k^3\log^4(n))})} \geqslant \frac{\mu}{(12n^{(k+1)})^{400k^3\delta^2\log^4(n)}}$$

minimal separators that are consistent with A^ℓ, B^ℓ, C^ℓ, and $G[A^\ell \cup B^\ell]$ has at most

$$260k^2\delta^2\log(n)^3 + 2 + (2k\log^2(n))(8k\delta\log(n))^2 \leqslant 400k^3\delta^2\log(n)^4$$

components. ■

7.4.2 Constructing the Generalized ω-Creature

In this section we will prove that if G is a k-creature free graph with n vertices and a sufficient number of minimal separators, then we can find a generalized ω-creature in G. We do this by taking the output of Lemma 7.4.6 from the previous subsection giving us A, B, and C such that $G - (A \cup B \cup C)$ has no component with over $\approx n/2\omega$ vertices and a large fraction of minimal separators of G are consistent with A, B, and C. This already gives us something close to a generalized ω-creature. Setting H to be an ω-bistar and φ to be a function that maps A and B to c_A and c_B (the two central vertices of H) respectively and ω of the components of $G - (A \cup B \cup C)$ to the ω peripheral vertices of H gives us something close to a generalized ω-creature, we are just missing the special sets S_1 and S_2. If there are at least ω components, X, of $G - (A \cup B \cup C)$ such that at least two minimal separators, S_X and S'_X (S_X and S'_X can depend on the specific X chosen), of G are consistent A, B, and C and $S_X \cap X \neq S'_X \cap X$ (plus an additional property which will be describe later on), then we can in fact show that we can construct a generalized ω-creature.

On the other hand, if we cannot find such a set of ω components of $G - (A \cup B \cup C)$ then this implies all minimal separators S of G that are consistent with A, B, and C intersect all but ω components $G - (A \cup B \cup C)$ in the exact same "unique" way. We will show that we can allocate the vertices of these "uniquely" intersected components of $G - (A \cup B \cup C)$ to makes sets A', B', and C' such that any minimal separator of G that was consistent with A, B, and C will be consistent with A', B' and C'. Since each component of $G - (A \cup B \cup C)$ has at most $n/2\omega$, this implies that over half of the vertices of G belong to A', B', or C'. Lemmas 7.4.4 and 7.4.5 then allow us to making an induced minor, G', of G with at most half the vertices of G but still maintaining a large fraction of G's minimal separators. Since we only sacrificed a small fraction of minimal

separators to drop the number of vertices in the graph by a factor of 2, repeating this process at most $\log(n)$ times then must result in us finding a generalized ω-creature for some large value of ω (or else we would end up with an empty graph with a supposedly large number of minimal separators).

Unfortunately, this isn't quite enough for our purposes. We will require not just any generalized ω-creature, but one with a bit more structure, which will force us to do some more pre-processing before constructing our generalized ω-creature. We call this more structured object a *connected good* generalized ω-creature, which we now define.

Good Components and Connected, Good, Full Generalized ω-Creatures.

Definition 7.4.38 (Full generalized ω-creatures)**.** A generalized ω-creature is *full* if, for every peripheral vertex, u, of H, $\varphi^{-1}(u) \cap S_1$ and $\varphi^{-1}(u) \cap S_2$ are distinct A_φ, B_φ-minimal separators in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)]$.

Note that a generalized ω-creature being full just means that the witness separators $S_1^\star$ and $S_2^\star$ in property *(i)* of generalized ω-creatures are precisely S_1 and S_2.

Let G be a graph, let $A, B, \subseteq V(G)$, and let X be a component of $G - (A \cup B)$. X is called a *non-leaf component of $G - (A \cup B)$ with respect to A and B* if it has at least two distinct neighbors in $G[A \cup B]$. A component of $G[X - N_G^2[A \cup B]]$ is said to be a *sub-component of X with respect to A and B*. A sub-component, Y, of X with respect to A and B is called a *non-leaf sub-component with respect to A and B* if Y has neighbors in at least two distinct components of $G[N_G^2[A \cup B]]$. X is said to be *good with respect to A and B* if X has at most one non-leaf sub-component with respect to A and B and for every pair of components P and Q of $G[A \cup B]$ it holds that $N_G^2[P] \cap X$ is anti-complete with $N_G^2[Q] \cap X$.

Definition 7.4.39 (Connected, Good)**.** Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature. If for all peripheral vertices, u, of H it holds that $\varphi^{-1}(u)$ is a connected vertex set (which implies that the vertex sets $\varphi^{-1}(u)$ are the components of $G-(A_\varphi \cup B_\varphi)$) then we call W a *connected generalized ω-creature*. If W is a connected generalized ω-creature and all components of $G-(A_\varphi \cup B_\varphi)$ are good with respect to A_φ and B_φ, then we call W a *connected good generalized ω-creature*.

We will find it useful to make the definitions just given slightly more flexible by allowing us to incorporate an additional set C into these definitions. Let G be a graph, let A, B, C, X, Y be vertex sets, and let $G' = G - C$. We will say that X is a non-leaf component of $G-(A \cup B \cup C)$ if X is a non-leaf component of $G'-(A \cup B)$ with respect to A and B. We say Y is a sub-component of X with respect to A, B and C if Y is a sub-component of X with respect to A and B in G'. We say Y is a non-leaf sub-component of X with respect to A, B and C, if Y is a non-leaf sub-component of X with respect to A and B in G'. We say X is good with respect to A, B, and C if X is good with respect to A and B in G'.

Our main result this subsection is to prove that any graph with a sufficient number of minimal separators will contain a connected, good, full generalized ω-creature for large ω. In particular, we will prove Lemma 7.3.2, which is the only lemma from this subsection that will be used outside of this subsection. We repeat the statement of Lemma 7.3.2 here for convenience.

Lemma 7.3.2. *Let G be a k-creature free graph with n vertices, let $\omega > 1$ and $\delta = 3\omega$, let c be an integer large enough so that $400k^3\delta^2 \log^4(c) < c/6$, let $x = 400k^3\delta^2 \log^4(n)$, and let G have at least $2^c(12n)^{6k^2x^4 \log(n)}$ minimal separators. Then there exists an induced minor G' of G such that $(G', H, \varphi, S_1, S_2)$ is a connected, good, full generalized ω-creature.*

Making the Components of $G - (A \cup B \cup C)$ Good

Let G be a graph with n vertices and a large number of minimal separators. Our first step toward proving Lemma 7.3.2 is to apply Lemma 7.4.6 to G with $\delta \approx \omega$ to get the sets A, B, and C. We then enrich A, B, and C to ensure that all components of $G - (A \cup B \cup C)$ are good components. In particular, we prove the following lemma.

Lemma 7.4.40. *Let G be a k-creature free graph (assume $k \geqslant 2$) with $n \geqslant 2$ vertices, let $\delta > 1$, and let G have μ minimal separators. Then there exist $A, B, C \subseteq V(G)$ such that the following conditions hold:*

(i) No component in $G - (A \cup B \cup C)$ has over n/δ vertices.

(ii) Let $x = 400k^3\delta^2 \log^4(n)$. Then G has at least $\mu/(12n)^{5k^2x^4}$ minimal separators that are consistent with $A, B,$ and C.

(iii) $G[A \cup B]$ has at most $400k^3\delta^2 \log^4(n)$ components.

(iv) All components, X, of $G - (A \cup B \cup C)$ are good with respect to A, B, and C.

(v) There are at most $k(400k^3\delta^2 \log^4(n))^2$ components of $G - (A \cup B \cup C)$.

Notice that properties *(i)-(iii)* are similar to what Lemma 7.4.6 can guarantee us. So, in order to prove Lemma 7.4.40 we assume that we have a k-creature free graph, G, and that we have been given sets A, B, and C (which will come from Lemma 7.4.6) and we want to find sets A', B', and C' which contain A, B, and C respectively and satisfy the properties of Lemma 7.4.40, in particular, some effort is required in order to satisfy properties *(iv)* and *(v)*. In order to do this we must study what happens as we grow A and B by successively taking their neighborhoods in a specially chosen induced subgraph of G. We will need the following lemma to determine this induced subgraph of G (which will end up being $G - C'$, where C' is a set produced by the following lemma).

Lemma 7.4.41. *Let G be a graph, let $A, B, C \subseteq V(G)$ such that G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, and let r be a natural number. Then there exists $A', B', C' \subseteq V(G)$ where the following conditions hold:*

(i) Let $G[A \cup B]$ have c components, then G has at least μ/n^{rkc} minimal separators that are consistent with A', B', and C'.

(ii) $C' \cap A = \emptyset$ and $C' \cap B = \emptyset$.

(iii) Let $G' = G - C'$ then $A' = N^r_{G'}[A]$ and $B' = N^r_{G'}[B]$.

Proof: Let G be a graph, let $A, B, C \subseteq V(G)$ such that G has $\mu > 0$ minimal separators that are consistent with A, B, and C, let $G[A \cup B]$ have c components, and let r be a natural number. We will show by induction on r that there exists sets A', B', and C' that satisfy conditions *(i)-(iii)* of this lemma. Recall for a set $X \subseteq V(G)$ that we define $N^0_G[X] = X$. It follows that taking $A' = A$, $B' = B$ and $C' = C$ satisfies conditions *(i)-(iii)* of this lemma for the base case when $r = 0$. Now assume this holds for all r less than some $r' > 0$. We will show it holds for $r = r'$.

We use the inductive hypothesis to find sets A', B', and C' such that *(i)* G has at least $\mu/n^{(r-1)kc}$ minimal separators that are consistent with A', B' and C' (let $\mathcal{S}$ be the set of these minimal separators) *(ii)* $C' \cap A = \emptyset$ and $C' \cap B = \emptyset$, and *(iii)* if $G' = G - C'$ then $A' = N^{r-1}_{G'}[A]$ and $B' = N^{r-1}_{G'}[B]$. Since $G[A \cup B]$ has c components, condition *(iii)* shows that $G[A' \cup B']$ has at most c components. By definition of consistent, for all $S \in \mathcal{S}$ it holds that $A' \cap S = \emptyset$ and $B' \cap S = \emptyset$, so we may then apply Lemma 7.4.10 with $U = A' \cup B'$ (and using the fact that $U \cap S = \emptyset$ for $S \in \mathcal{S}$ to conclude the set $\mathcal{S}^U = \{N_G(U) \cap S | S \in \mathcal{S}\}$ has size $|\mathcal{S}^U| \leqslant n^{kc}$. Since $|\mathcal{S}^U| \leqslant n^{kc}$ there is an $X \in \mathcal{S}^U$ such that at least $|\mathcal{S}|/n^{kc} \geqslant \mu/n^{rkc}$ (by the inductive hypothesis) minimal separators $S \in \mathcal{S}$ have the property that $N_G(U) \cap S = X$. Denote this subset of $\mathcal{S}$ as $\mathcal{S}^X$, so $|\mathcal{S}^X| \geqslant \mu/n^{rkc}$.

We will show that $N_{G'}[A'] - X$, $N_{G'}[B'] - X$, and $C \cup X$ satisfy the properties of this lemma.

We first establish property *(i)*. We have that for all $S \in \mathcal{S}^X$, since S is consistent with A', B', and C' and $N_G(A' \cup B') \cap S = X$, it follows that S is consistent with $N_G[A'] - X$, $N_G[B'] - X$, and $C \cup X$ and therefore, since $N_{G'}[A'] \subseteq N_G[A']$ and $N_{G'}[B'] \subseteq N_G[B']$, S is consistent with $N_{G'}[A'] - X$, $N_{G'}[B'] - X$, and $C \cup X$. Since $|\mathcal{S}^X| \geqslant \mu/n^{rkc}$ this established condition (i).

To see that property *(ii)* holds note that since A' and B' are both disjoint from $S \in \mathcal{S}$ and from C' we have that $X \cap A' = X \cap B' = \emptyset$, it holds that $(C' \cup X) \cap (N_{G'}[A'] - X) = (C' \cup X) \cap (N_{G'}[B'] - X) = \emptyset$. This establishes condition (ii).

Lastly, we establish condition *(iii)*. Since, by the inductive hypothesis, $A' = N_{G'}^{r-1}[A]$, $B' = N_{G'}^{r-1}[B]$, and $X \cap (A' \cup B') = \emptyset$ it follows that if $G'' = G' - X$ then $N_{G'}[A'] - X = N_{G''}^{r}[A]$ and $N_{G'}[B'] - X = N_{G''}^{r}[B]$ which establishes condition (iii). ■

We now define a sequence which we will study in order to prove Lemma 7.4.40. Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let A', B', and C' be the sets returned by applying Lemma 7.4.41 to G, A, B, C, and $2r = 2(kc^3 + c)$ and let $G' = G - C'$. We define a sequence, $\mathsf{seq}_3(G, A, B, C)$, to be a sequence of tuples where the i^{th} tuple, $1 \leqslant i \leqslant r$, is defined to be (A_i, B_i) where $A_i = N_{G'}^{2i}[A]$ and $B_i = N_{G'}^{2i}[B]$. The sets A', B', and C' will be referred to as the *core sets* of $\mathsf{seq}_3(G, A, B, C)$ and the graph G' will be referred to as the *core graph* of $\mathsf{seq}_3(G, A, B, C)$.

We now prove a series of lemmas which will eventually allow us to show that there exists some tuple (A_j, B_j) of $\mathsf{seq}_3(G, A, B, C)$ such that A_j, B_j, and C' satisfy the properties of Lemma 7.4.40, assume that the sets A, B, and C were obtained from Lemma 7.4.6.

Lemma 7.4.42. *Let G be a k-creature free graph and let $A, B, C \subseteq V(G)$ where G has*

$\mu \geqslant 1$ minimal separators that are consistent with A, B, and C. Then for a tuple (A_i, B_i) of $\mathsf{seq}_3(G, A, B, C)$ it holds that A_i is anti-complete with B_i.

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, and let A', B', and C' be the core sets of $\mathsf{seq}_3(G, A, B, C)$. If $G[A \cup B]$ has c components then by property *(i)* of Lemma 7.4.41 G has at least $\mu/n^{2(kc^3+c)kc} > 0$ minimal separators that are consistent with A', B', and C', hence G has at least one minimal separator that is consistent with A', B', and C'. It follows that A' and B' must be anti-complete. By property *(iii)* of Lemma 7.4.41 we can see that for a tuple (A_i, B_i) of $\mathsf{seq}_3(G, A, B, C)$ it holds that $A_i \subseteq A'$ and $B_i \subseteq B'$, hence A_i is anti-complete with B_i. ■

Lemma 7.4.43. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, and let (A_i, B_i) be a tuple of $\mathsf{seq}_3(G, A, B, C)$. Then $|\mathcal{CC}(G[A_i \cup B_i])|$ is a non-increasing sequence in i.*

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let (A_i, B_i) and (A_{i+1}, B_{i+1}) be tuples of $\mathsf{seq}_3(G, A, B, C)$, and let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$. Then $A_{i+1} = N^2_{G'}[A_i]$ and $B_{i+1} = N^2_{G'}[B_i]$ and it follows that $G[A_{i+1} \cup B_{i+1}]$ has at most as many components as $G[A_i \cup B_i]$ which proves the lemma. ■

Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C. We define the *i^{th} critical number*, c_i, of $\mathsf{seq}_3(G, A, B, C)$ to be the i^{th} number such that the number of components of $G[A_{c_i} \cup B_{c_i}]$ is strictly less than the number of components of $G[A_{(c_i)+1} \cup B_{(c_i)+1}]$ for the tuples (A_{c_i}, B_{c_i}) and $(A_{(c_i)+1} \cup B_{(c_i)+1})$ of $\mathsf{seq}_3(G, A, B, C)$. In other words, the critical numbers denote indexes of $\mathsf{seq}_3(G, A, B, C)$ where the number of components strictly decreases. For convenience, we also let 0 be a critical number of $\mathsf{seq}_3(G, A, B, C)$ as well

as the index, r, of the last tuple of $\mathsf{seq}_3(G, A, B, C)$ in order to have the fact that every index of the sequence $\mathsf{seq}_3(G, A, B, C)$ is either a critical number or lies between two critical numbers. Let $\mathsf{seq}_3(G, A, B, C)$ have t critical numbers other than 0 and r. We will call 0 the 0^{th} critical number and will be denoted by c_0 and we will call r the $t+1^{th}$ critical number and will be denoted by c_{t+1}. The following corollary follows from the fact that if there are c components of $G[A \cup B]$ then $\mathsf{seq}_3(G, A, B, C)$ has $r = kc^3 + c$ elements, Lemma 7.4.43, and the fact that no graph can have a negative number of components.

Corollary 7.4.44. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, and let there be c components of $G[A \cup B]$. Then there exists critical numbers c_i and c_{i+1} of $\mathsf{seq}_3(G, A, B, C)$ such that $c_{i+1} - c_i > kc^2 + 1$.*

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, and let there be c components of $G[A \cup B]$. Since for each tuple (A_i, B_i) of $\mathsf{seq}_3(G, A, B, C)$ we have that $A_i, B_i \neq \emptyset$ and A_i and B_i are anti-complete by Lemma 7.4.42 it follows that $G[A_i \cup B_i]$ must have at least two components. Hence there can be at most c critical numbers (including 0 and $r = kc^3 + c$) of $\mathsf{seq}_3(G, A, B, C)$. Since $\mathsf{seq}_3(G, A, B, C)$ has $kc^3 + c$ elements, it follows that there must be some critical numbers c_i and c_{i+1} such that $c_{i+1} - c_i > kc^2 + 1$. ■

Lemma 7.4.45. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let c_i and c_{i+1} be critical numbers of $\mathsf{seq}_3(G, A, B, C)$, and let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$. If (A_j, B_j) is a tuple of $\mathsf{seq}_3(G, A, B, C)$ where $c_i < j < c_{i+1}$ then for all pairs of components P, Q of $G[A_j \cup B_j]$ it holds that $N^2_{G'}[P]$ is anti-complete with $N^2_{G'}[Q]$.*

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$

minimal separators that are consistent with A, B, and C, let c_i and c_{i+1} be critical numbers of $\mathsf{seq}_3(G, A, B, C)$, let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$, and let (A_j, B_j) and (A_{j+1}, B_{j+1}) be tuples of $\mathsf{seq}_3(G, A, B, C)$ where $c_i < j < c_{i+1}$ (in particular, this imples that j is not a critical number). Assume for a contradiction that there are components P, Q of $G[A_j \cup B_j]$ such that $P' = N^2_{G'}[P]$ is not anti-complete with $Q' = N^2_{G'}[Q]$. It follows that $G[P' \cup Q']$ is connected and therefore only contains one component. Since $A_{j+1} = N^2_{G'}[A_j]$ and $B_{j+1} = N^2_{G'}[B_j]$, this implies that the number of components of $G[A_{j+1} \cup B_{j+1}]$ is strickly less that the number of components of $G[A_j \cup B_j]$. But this contradicts the fact that j is not a critical number. ■

The following lemma will be needed to prove property *(v)* of Lemma 7.4.40.

Lemma 7.4.46. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let c denote the number of components of $G[A \cup B]$, let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$, and let c_i and c_{i+1} be critical numbers of $\mathsf{seq}_3(G, A, B, C)$. If (A_j, B_j) is a tuple of $\mathsf{seq}_3(G, A, B, C)$ where $c_i < j < c_{i+1}$. Then there are less than kc^2 non-leaf components of $G' - (A_j \cup B_j)$ with respect to A_j and B_j.*

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let c denote the number of components of $G[A \cup B]$, let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$, let c_i and c_{i+1} be critical numbers of $\mathsf{seq}_3(G, A, B, C)$, and let (A_j, B_j) be a tuple of $\mathsf{seq}_3(G, A, B, C)$ where $c_i < j < c_{i+1}$. Assume for a contradiction that there are at least kc^2 non-leaf components of $G' - (A_j \cup B_j)$ with respect to A_j and B_j. Since there are c components of $G[A \cup B]$, by Lemma 7.4.43 there are at most c components of $G[A_j \cup B_j]$. It follows there are $(c^2 - c)/2$ pairs of distinct components of $G[A_j \cup B_j]$, so by the pigeon hole principle, there exists components P and Q of $G[A_j \cup B_j]$ such that there are at least k non-leaf

components, $X_1, X_2, \ldots, X_k$, of $G' - (A_j \cup B_j)$ that have neighbors in both P and Q. Let Y_i denote an induced path from P to Q whose internal vertices are contained in X_i, so the internal vertices of Y_t are anti-complete with the internal vertices of $Y_{t'}$ for $t \neq t'$. By Lemma 7.4.45 each Y_i must have at least six vertices. So, since P and Q are connected, P is anti-complete with Q, the internal vertices of Y_t are anti-complete with the internal vertices of $Y_{t'}$ for $t \neq t'$, and the Y_i's are induced paths from P to Q of length at least six, we can see that this implies that G contains a k-creature, which is a contradiction.

■

Lemma 7.4.47. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$, let c_i and c_{i+1} be critical numbers of $\mathsf{seq}_3(G, A, B, C)$, and let (A_j, B_j) and (A_{j+1}, B_{j+1}) be tuples of $\mathsf{seq}_3(G, A, B, C)$ where $c_i < j < c_{i+1}$. Then every non-leaf component of $G' - (A_j \cup B_j)$ with respect to A_j and B_j contains a non-leaf sub-component with respect to A_j and B_j. Furthermore, if X is a component of $G' - (A_j \cup B_j)$ then every non-leaf sub-component of X with respect to A_j and B_j is a non-leaf component of $G' - (A_{j+1} \cup B_{j+1})$ with respect to A_{j+1} and B_{j+1}.*

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$, let c_i and c_{i+1} be critical numbers of $\mathsf{seq}_3(G, A, B, C)$, and let (A_j, B_j) and (A_{j+1}, B_{j+1}) be tuples of $\mathsf{seq}_3(G, A, B, C)$ where $c_i < j < c_{i+1}$. Assume X is a component of $G' - (A_j \cup B_j)$ and let Y be a non-leaf sub-component of X with respect to A_j and B_j.

First we show that every non-leaf sub-component of X with respect to A_j and B_j is a non-leaf component of $G' - (A_{j+1} \cup B_{j+1})$ with respect to A_{j+1} and B_{j+1}. By definition, Y is a component of $G[X - N^2_{G'}[A_j \cup B_j]] = G[X - (N^2_{G'}[A_j] \cup N^2_{G'}[B_j])] = G[X - (A_{j+1} \cup B_{j+1})]$,

and therefore Y is a component of $G' - (A_{j+1} \cup B_{j+1})$. Furthermore, by definition, Y has neighbors in two components of $G[N^2_{G'}[A_j \cup B_j]] = G[N^2_{G'}[A_j] \cup N^2_{G'}[B_j]] = G[A_{j+1} \cup B_{j+1}]$, so Y is a non-leaf component of $G' - (A_{j+1} \cup B_{j+1})$ with respect to A_{j+1} and B_{j+1}.

Next, we show that if X is a non-leaf component of $G' - (A_j \cup B_j)$ then X must contain a non-leaf sub-component with respect to A_j and B_j. Let Z denote the set of vertices that belong to a component of $G[A_j \cup B_j]$ that has at least one neighbor in X, so $G[Z]$ has at least two components since X is a non-leaf component and $G[X - (N^2_{G'}[A_i] \cup N^2_{G'}[B_i])] = G[X - N^2_{G'}[Z]]$. Since $G[X \cup Z]$ is connected, $G[(X - N^2_{G'}[Z]) \cup N^2_{G'}[Z]] = G[X \cup N^2_{G'}[Z]]$ is connected and it follows from Lemma 7.4.45 that since $G[Z]$ has at least two components, $G[N^2_{G'}[Z]]$ has at least two components. It follows there must be at least one component of $G[X - N^2_{G'}[Z]] = G[X - N^2_{G'}[A_j \cup B_j]]$ that has at least one neighbor in two components of $G[N^2_{G'}[Z]]$ (or else $G[(X - N^2_{G'}[Z]) \cup N^2_{G'}[Z]]$ would not be connected) and therefore has at least one neighbor in two components of $G[N^2_{G'}[A_{j+1} \cup B_{j+1}]]$. ■

The next corollary follows immediately from Lemma 7.4.47

Corollary 7.4.48. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$ where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$, let c_i and c_{i+1} be critical numbers of $\mathsf{seq}_3(G, A, B, C)$, and let (A_j, B_j) be a tuple of $\mathsf{seq}_3(G, A, B, C)$ where $c_i < j < c_{i+1}$. Then the number of non-leaf components of $G' - (A_j \cup B_j)$ with respect to A_j and B_j is a non-decreasing sequence in j (when the possible values of j are restricted to lie between c_i and c_{i+1}).*

The following lemma is the main lemma that we need to prove Lemma 7.4.40. Namely, that we can find an (A_j, B_j) of $\mathsf{seq}_3(G, A, B, C)$ has the property that all components of $G' - (A_j \cup B_j)$ are good, where G' is the core graph of $\mathsf{seq}_3(G, A, B, C)$.

Lemma 7.4.49. *Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, and let G' be the*

core graph of $\mathsf{seq}_3(G, A, B, C)$. *Then there exists an integer* j *such that the tuple* (A_j, B_j) *of* $\mathsf{seq}_3(G, A, B, C)$ *has the property that all components of* $G' - (A_j \cup B_j)$ *are good with respect to* A_j *and* B_j.

Proof: Let G be a k-creature free graph, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, where G has $\mu \geqslant 1$ minimal separators that are consistent with A, B, and C, let c denote the number of components of $G[A \cup B]$, and let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$. By Corollary 7.4.44 there exists two critical numbers c_i and c_{i+1} of $\mathsf{seq}_3(G, A, B, C)$ such that $c_{i+1} - c_i > kc^2 + 1$. By Lemma 7.4.46 for any (A_j, B_j) in $\mathsf{seq}_3(G, A, B, C)$ with $c_i < j < c_{i+1}$ there are less than kc^2 non-leaf components of $G' - (A_j \cup B_j)$ with respect to A_j and B_j, and by Corollary 7.4.48 the number of non-leaf components of $G' - (A_j \cup B_j)$ with respect to A_j and B_j is a non-decreasing sequence in j when $c_i < j < c_{i+1}$. It follows that there must exists some j', $c_i < j' < c_{i+1}$ such that the number of non-leaf components of $G' - (A_{j'} \cup B_{j'})$ is the same as the number of non-leaf components of $G' - (A_{j'+1} \cup B_{j'+1})$.

We now show that all components, X, of $G' - (A_{j'} \cup B_{j'})$ are good with respect to $A_{j'}$, and $B_{j'}$. Let P and Q be two components of $G[A_{j'} \cup B_{j'}]$. By Lemma 7.4.45 it holds that $N^2_{G'}[P] \cap X$ is anti-complete with $N^2_{G'}[Q] \cap X$. Next, it follows from Lemma 7.4.47 that if X has two non-leaf sub-components with respect to $A_{j'}$ and $B_{j'}$, then the number of non-leaf components of $G' - (A_{j'} \cup B_{j'})$ with respect to $A_{j'}$ and $B_{j'}$ is strictly less than the number of non-leaf components of $G' - (A_{j'+1} \cup B_{j'+1})$ with respect to $A_{j'+1}$ and $B_{j'+1}$, which is a contradiction. It follows that X is a good component. ■

Let G be a k-creature free graph with many minimal separators, and let A, B, and C be the sets obtained from Lemma 7.4.6 applied to G. We will show that the tuple (A_j, B_j) obtained from Lemma 7.4.49 almost satisfies the properties of Lemma 7.4.40. The problem is that $G' - (A_j \cup B_j)$ could have a large number of components, so property

(v) of Lemma 7.4.40 might not be satisfied. But Lemma 7.4.46 shows that not many components of $G' - (A_j \cup B_j)$ have neighbors in both A_j and B_j. The next lemma show how we get rid of the components of $G' - (A_j \cup B_j)$ that do not have neighbors in both A_j and B_j.

Lemma 7.4.50. *Let G be a graph, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, and let S be a minimal separator of G that is consistent with A, B, and C. Let A^*, B^*, and C^* be the sets of vertices that belong to a component of $G-(A\cup B\cup C)$ that has neighbors in A and not B, that has neighbors in B and not A, and that does not have neighbor $B \cup A$ respectively. Then S is consistent with $A \cup A^*$, $B \cup B^*$, and $C \cup C^*$.*

Proof: Let G be a graph, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, and let S be a minimal separator of G that is consistent with A, B, and C. Let A^*, B^*, and C^* be the sets of vertices that belong to a component of $G-(A\cup B\cup C)$ that has neighbors in A and not B, that has neighbors in B and not A, and that does not have neighbor $B \cup A$ respectively.

First, we show that $S \cap A^* = \emptyset$. Assume for a contradiction that there is an $s \in S \cap A^*$. Since S is consistent with A, B, and C, B belongs to an S-full component, call it B', of $G - S$ which does not contain any vertex from A nor C. So, there must be some path, P from B to s such that the internal vertices of P are contained in B' and therefore disjoint from A and C. But, since s lies in a component, Q, of $G - (A \cup B \cup C)$ such that $N_G(Q) \subseteq A \cup C$, any path from B to Q must contain a vertex from $A \cup C$, hence such a path P cannot exists.

A symmetric argument shows that $S \cap B^* = \emptyset$. Now since S does not contain any vertex from $A^* \cup B^*$, we can see that A^* and A belong to the same component of $G - S$ and B^* and B belong to the same component of $G - S$. Hence S is consistent with $A \cup A^*$, $B \cup B^*$ and C. All that needs to be shown to complete the proof is that no vertex of C^* belongs to the same component that A nor B belongs to $G - S$.

So, assume for a contradiction that there is a $c \in C^* - S$ that belongs to the same component that either A or B does in $G - S$, without loss of generality assume that s belongs to the same component A does in $G - S$. So, there must be some path, P from A to c such that the internal vertices of P are contained in the component A belongs to in $G - S$ and therefore is disjoint from B and C. But, since c lies in a component, Q, of $G - (A \cup B \cup C)$ such that $N_G(Q) \subseteq C$, any path from A to Q must contain a vertex from C, hence such a path P cannot exists. ■

We are now ready to prove Lemma 7.4.40

Proof: [Proof of Lemma 7.4.40] Let G be a k-creature free graph, $k \geqslant 2$, with $n \geqslant 2$ vertices, let $\delta > 1$, and let G have μ minimal separators. Apply Lemma 7.4.6 to G and δ to get sets A, B, and C that satisfy properties *(i)-(iii)* of Lemma 7.4.6. Let G' be the core graph of $\mathsf{seq}_3(G, A, B, C)$ and let A', B', and C' be the core sets of $\mathsf{seq}_3(G, A, B, C)$. By Lemma 7.4.49 there exists a tuple (A_j, B_j) of $\mathsf{seq}_3(G, A, B, C)$ such that all components of $G' - (A_j \cup B_j)$ are good with respect to A_j and B_j, hence all components of $G - (A_j \cup B_j \cup C')$ are good with respect to A_j, B_j, and C'. Let A^* be the set of all vertices that belong to a component, X, of $G - (A_j \cup B_j \cup C')$ such that X has neighbors in A and no neighbors in B, let B^* be the set of all vertices that belong to a component, X, of $G - (A_j \cup B_j \cup C')$ such that X has neighbors in B and no neighbors in A, and let C^* be the set of all vertices that belong to a component, X, of $G - (A_j \cup B_j \cup C')$ such that X has no neighbors $A \cup B$. We will show that the sets $A'' = A_j \cup A^*$, $B'' = B_j \cup B^*$, and $C'' = C' \cup C^*$ satisfy properties *(i)-(v)* of this lemma.

We first establish property *(i)* of this lemma. By property *(i)* of Lemma 7.4.6 no component of $G - (A \cup B \cup C)$ has over n/δ vertices. By property *(iii)* of Lemma 7.4.41 and the definition of $\mathsf{seq}_3(G, A, B, C)$ it follows that $A \subseteq A_j$, $B \subseteq B_j$, and $C \subseteq C'$, hence no component of $G - (A'' \cup B'' \cup C'')$ has over n/δ vertices. This establishes property *(i)* of this lemma.

Next, we establish properties *(ii)*. Let $x = 400k^3\delta^2\log^4(n)$. Then by property *(ii)* of Lemma 7.4.6, G has at most $\mu/(12n^{(k+1)})^x$ minimal separators that are consistent with A, B, and C, and property *(iii)* of Lemma 7.4.6 $G[A \cup B]$ has at most x components. So by property *(i)* of Lemma 7.4.41 and by how A', B' and C' were defined, G has at least

$$\frac{\mu}{(12n^{(k+1)})^x n^{2(kx^3+x)kx}} \geqslant \frac{\mu}{(12n)^{(k+1)x+2(kx^3+x)kx}} \geqslant \frac{\mu}{(12n)^{5k^2x^4}}$$

minimal separators that are consistent with A', B', and C'. Since $A_j \subseteq A'$ and $B_j \subseteq B'$, G has at least $\mu/(12n)^{5k^2x^4}$ minimal separators that are consistent with A_j, B_j, and C'. Lastly, by Lemma 7.4.50 G has at least $\mu/(12n)^{5k^2x^4}$ minimal separators that are consistent with $A'' = A_j \cup A^*$, $B'' = B_j \cup B^*$, and $C'' = C' \cup C^*$. Hence property *(ii)* of this lemma is satisfied.

Now, we establish property *(iii)*. By property *(iii)* of Lemma 7.4.6, $G[A \cup B]$ has at most $400k^3\delta^2\log^4(n)$ components. By how A_j and B_j are defined, we have that $A_j = N^{2j}_{G'}[A]$ and $B_j = N^{2j}_{G'}[B]$, so it follows that $G[A_j \cup B_j]$ has less than or equal to $400k^3\delta^2\log^4(n)$ components. It can then be seen by the definition of A^* and B^* that $G[A'' \cup B'']$ has less than or equal to $400k^3\delta^2\log^4(n)$ components. This established property *(iii)*.

Next, we prove property *(iv)*. It follows from the definitions of A^*, B^*, and C^* that $\mathcal{CC}(G - (A'' \cup B'' \cup C''))$ is a subset of $\mathcal{CC}(G - (A_j \cup B_j \cup C'))$, and $N_G(A^*) \subseteq A_j \cup C'$ and $N_G(B^*) \subseteq B_j \cup C'$. Let $G'' = G - C''$. Since $C'' \subseteq C'$ we have that $N_{G''}(A^*) \subseteq A_j$ and $N_{G''}(B^*) \subseteq B_j$. Then for a component, X, of $G - (A'' \cup B'' \cup C'')$, we have that $N^2_{G''}[A'' \cup B''] \cap X = N^2_{G''}[A_j \cup B_j] \cap X$, and since X is a component of $G'' - (A_j \cup B_j)$ and of $G' - (A_j \cup B_j)$ it follows that $N^2_{G''}[A_j \cup B_j] \cap X = N^2_{G'}[A_j \cup B_j] \cap X$. Hence $N^2_{G''}[A'' \cup B''] \cap X = N^2_{G'}[A_j \cup B_j] \cap X$.

So, in G, any non-leaf sub-component of X with respect to A'', B'', and C'' must be

a non-leaf sub-component of X with respect to A_j, B_j, and C' and for any components P, Q of $G[A'' \cup B'']$ it holds that $N^2_{G''}[P] \cap X$ is anti-complete with $N^2_{G''}[Q] \cap X$ (or else we could find components P', Q' of $G[A_j \cup B_j]$ such that $N^2_{G'}[P'] \cap X$ is not anti-complete with $N^2_{G'}[Q'] \cap X$, contradicting the fact that X is good with respect to A_j, B_j, and C'). It follows that all components of $G - (A'' \cup B'' \cup C'')$ are good with respect to A'', B'', and C''.

Lastly, we verify property *(v)*. Note that by the definition of A^*, B^* and C^*, the components of $G - (A'' \cup B'' \cup C'')$ are precisely the components of $G - (A_j \cup B_j \cup C')$ that have neighbors in both A_j and B_j, and therefore in A'' and B''. This implies that every component of $G-(A''\cup B''\cup C'')$ is a non-leaf component of $G'-(A_j\cup B_j)$ in G' with respect to A_j and B_j, of which there are at most $k|\mathcal{CC}(G[A_j \cup B_j])|^2$ such component by Lemma 7.4.46. As noted when proving property *(iii)* of this lemma, $G[A_j \cup B_j]$ has at most $400k^3\delta^2 \log^4(n)$ components, hence there are at most $k(400k^3\delta^2 \log^4(n))^2$ components of $G - (A'' \cup B'' \cup C'')$, which establishes property *(v)*. ■

Finding the Sets S_1 and S_2 for a Generalized ω-Creature

Let G be a k-creature free graph with many minimal separators and let A, B, and C be the output of applying Lemma 7.4.40 to G. We outlined at the start of 7.4.2 how we can use A, B, and C to construct a connected, good, full generalized ω-creature. The pieces right now that we are missing are the special sets S_1 and S_2. We claimed that if there are at least ω components, X, of $G - (A \cup B \cup C)$ such that at least two minimal separators, S_X and S'_X (S_X and S'_X can depend on the specific X chosen), of G are consistent A, B, and C and $S_X \cap X \neq S'_X \cap X$ (plus an additional property), then we can in fact show that we can construct a good connected generalized ω-creature. The following definition is the additional property that we need the minimal separators to satisfy, it essentially forces all pairs of minimal separators that we consider to satisfy

property *(iii)* of Definition 7.3.1.

Signatures Let G be a graph, let $A, B, C \subseteq V(G)$, and let $\mathcal{M} \subseteq (\mathcal{CC}(G[A]) \times \mathcal{CC}(G[A]) \cup \mathcal{CC}(G[B]) \times \mathcal{CC}(G[B]))$. We call $\mathcal{M}$ a *mark for A and B*. Let S be a minimal separator of G that is consistent with A, B, and C and let X be a component of $G - (A \cup B \cup C)$. If $\mathcal{M}$ is the mark with the property that the pair (U, V) of $(\mathcal{CC}(G[A]) \times \mathcal{CC}(G[A]) \cup \mathcal{CC}(G[B]) \times \mathcal{CC}(G[B]))$ is in $\mathcal{M}$ if and only if $U \neq V$ and there is a path from U to V through $X - S$, then we call $\mathcal{M}$ the *mark of S with respect to X, A, and B*.

Definition 7.4.51. Let G be a graph, and let $A, B, C \subseteq V(G)$. Define a function T from the set of components of $G - (A \cup B \cup C)$ to the set of marks for A and B, so for each component X of $G - (A \cup B \cup C)$, $T(X)$ is a mark for A and B. We call T a *signature for G, A, B, and C*. We say that a minimal separator S of G *agrees with* A, B, C, and T if S is consistent with A, B, and C, and for all components X of $G - (A \cup B \cup C)$ it holds that $T(X)$ is equal to the mark of S with respect to X, A, and B.

Let G be a graph, let $A, B, C \subseteq V(G)$, let T be a signature of G, A, B, and C, and let S_1 and S_2 be two minimal separators of G that agree with A, B, C, and T. Notice that for any component X of $G - (A \cup B \cup C)$, for all pair of components C_1 and C_2 of $G[A \cup B]$ there is a path from C_1 to C_2 through $X - S_1$ in G if and only if there is a path from C_1 to C_2 through $X - S_2$ in G which is what property *(iii)* of Definition 7.3.1 requires of generalized ω-creatures.

The next few lemmas show that if G has many minimal separators that are consistent with A, B, and C, then there exists a signature T for G, A, B, and C so that G has many minimal separators that agree with A, B, C, and T. We begin with the following observation about signature functions which we will use without explicitly reference. The proof follows easily from the definition of signatures.

Observation 7.4.52. *Let G be a graph, let $A, B, C \subseteq V(G)$, and let S be a minimal separator of G that is consistent with A, B, and C. Then there exists exactly one signature, T, for G, A, B, and C such that S agrees with A, B, C, and T.*

Lemma 7.4.53. *Let G be a graph, let $A, B, C \subseteq V(G)$ where A is anti-complete with B and let there be x components of $G[A \cup B]$ and y components of $G - (A \cup B \cup C)$. Then there are $2^{x^2 y}$ functions that are signatures for G, A, B, and C.*

Proof: Let G be a graph, let $A, B, C \subseteq V(G)$ where A is anti-complete with B and let there be x components of $G[A \cup B]$ and y components of $G - (A \cup B \cup C)$. Let T be a signature for G, A, B, and C. The domain of T is the set of components of $G - (A \cup B \cup C)$, which by assumption has y elements, and the range of T is the power set of $(\mathcal{CC}(G[A]) \times \mathcal{CC}(G[A]) \cup \mathcal{CC}(G[B]) \times \mathcal{CC}(G[B]))$, which by assumption has at most 2^{x^2} elements. So for each of the y elements of the domain there is a choice of 2^{x^2} elements to map it to, hence there are at most $2^{x^2 y}$ possible signatures for G, A, B, and C. ■

Lemma 7.4.54. *Let G be a k-creature free graph (assume $k \geqslant 2$) with $n \geqslant 2$ vertices, let $\delta > 1$, and let G have μ minimal separators. Then there exist $A, B, C \subseteq V(G)$ and signature T for G, A, B, and C such that the following conditions hold:*

(i) No component of $G - (A \cup B \cup C)$ has over n/δ vertices.

(ii) Let $x = 400k^3\delta^2 \log^4(n)$. Then G has at least $\mu/(12n)^{6k^2x^4}$ minimal separators that agree with A, B, C, and T.

(iii) $G[A \cup B]$ has at most $400k^3\delta^2 \log^4(n)$ components.

(iv) All components of $G - (A \cup B \cup C)$ are good with respect to A, B, and C.

Proof: Let G be a k-creature free graph (assume $k \geqslant 2$) with $n \geqslant 2$ vertices, let $\delta > 1$, and let G have μ minimal separators. We apply Lemma 7.4.40 using G and δ to

gets sets A, B, and C that satisfy properties *(i)-(v)* of Lemma 7.4.40. We can see that this implies the sets A, B, and C satisfy properties *(i), (iii)*, and *(iv)* of this lemma, so we are left with finding a suitable signature T to satisfy property *(iii)*.

Let $x = 400k^3\delta^2\log^4(n)$, then by property (i) of Lemma 7.4.40, there are at least $\mu/(12n)^{5k^2x^4}$ minimal separators that are consistent with A, B, and C. Furthermore, by properties (iii) and (iv) of Lemma 7.4.40 there are x components of $G[A \cup B]$ and kx^2 components of $G-(A\cup B\cup C)$, hence, by Lemma 7.4.53 there are $2^{x^2kx^2} = 2^{x^4k}$ possible signature functions for G, A, B, and C. It follows that there is some signature function T such that there are at least

$$\frac{\mu}{(12n)^{5k^2x^4}}\frac{1}{2^{x^4k}} \geqslant \frac{\mu}{(12n)^{6k^2x^4}}$$

minimal separators of G that agrees with A, B, C, and T. ■

If G is a k-creature free graph with many minimal separators, then by Lemma 7.4.54 there are sets $A, B, C \subseteq V(G)$ and signature T for G, A, B, and C such that G has many minimal separtors that agree with A, B, C, and T. Let S_1 and S_2 be two minimal separators of G that agree with A, B, C, and T. As stated previously, for any component, X, of $G-(A\cup B\cup C)$, for all pair of components C_1 and C_2 of $G[A\cup B]$ there is a path from C_1 to C_2 through $X - S_1$ in G if and only if there is a path from C_1 to C_2 through $X - S_2$ in G which is what property *(iii)* of Definition 7.3.1 requires of generalized ω-creatures. The problem we have is that there may be components X of $G-(A\cup B\cup C)$ such that $S_1 \cap X = S_2 \cap X$, and so S_1 and S_2 will fail property *(i)* of Definition 7.3.1. The following lemma and corollary will help us fix this problem.

Lemma 7.4.55. *Let G be a graph, let $A, B \subseteq V(G)$, let T be a signature for G, A, B, and $C = \emptyset$, let X be a component of $G-(A\cup B)$, and let S_1, S_2 be two minimal separators of G that agree with A, B, C, and T. Let $S_1 \cap X = S_1'$ and $S_2 \cap X = S_2'$. Then $(S_1 - S_1') \cup S_2'$*

is a minimal separator that agrees with $A, B, C,$ and T.

Proof: Let G be a graph, let $A, B \subseteq V(G)$, let T be a signature for G, A, B, and $C = \emptyset$, let X be a component of $G-(A\cup B)$, and let S_1, S_2 be two minimal separators of G that agree with A, B, C, and T. Let $S_1 \cap X = S_1'$ and $S_2 \cap X = S_2'$. Let $S = (S_1 - S_1') \cup S_2'$.

We first show that there is some component of $G - S$ that contains A. An identical argument shows that there is some component of $G - S$ that contains B. Let A_1 and A_2 be two components of $G[A]$, and let P be a path from A_1 to A_2 in $G - S_1$. We show how to get a path P' from A_1 to A_2 in $G - S$ by replacing portions of P. Let P^* be a maximal subpath such that no internal vertex of P^* is contained in A, let A_1' and A_2' be the components of $G[A]$ that the endpoints of P^* belong to. Since S_1 is an A, B-separator, we have that all internal vertices of P^* are contained in $G - (A \cup B)$. It follows that the internal vertices of P^* belong to some component X' of $G - (A \cup B)$. If $X' = X$ by assumption $T(X)$ is the mark of both S_1 and S_2 for X, A, and B, so P^* can be replaced with some other path from A_1' to A_2' with internal vertices contained in $X' - S_2$. If $X' \neq X$, then it follows that P^* is also a path in $G - S$ so it does not need to be replaced. We can see that we can replace the portions of P that do not belong to A in this manner to get a path P' from A_1 to A_2 in $G - S$.

We now show that S separates A from B. Assume for a contradiction that there is a path from A to B in $G - S$, let P be a shortest such path so we may assume that the end points of P are in A and B and all internal vertices are in $G - (S \cup A \cup B)$. It follows that all the internal vertices of P must be contained in some component X' of $G - (A \cup B)$, but if $X' = X$ then such a path must contain a vertex from S_2, else S_2 would not be consistent with A, B, and C, and if $X' \neq X$ then such a path must contain a vertex from S_1, else S_1 would not be consistent with A, B, and C. From the definition of S it follows that no such path can exist, hence A and B are contained in different

components of $G - S$.

Now we show that A and B belong to two S-full components of $G - S$. We show that the component A is contained in in $G - S$ dominates S, an identical argument proves the component B is contained in dominates S. Let $s \in S$ and let X' be the component that s belongs to in $G - (A \cup B)$. If $X' = X$, then since A is an S_2-full component of $G - S_2$ there is path P from A to s in $G - S_2$ whose internal vertices belong to $X' - S_2$. If $X' \neq X$, then since A is an S_1-full component of $G - S_1$ there must be a path from A to s in $G - S_1$ whose internal vertices belong to $X' - S_1$. It follows that the component A belongs to in $G - S$ dominates S.

Together, this proves that S is a minimal separator of G that is consistent with A, B, and $C = \emptyset$. To show that S agrees with A, B, C, and T, we must show that for all components, X', of $G - (A \cup B)$ the mark of S with respect to X, A, and B is $T(X')$. This follows from the fact that if $X' = X$, then the mark of S with respect to X', A, and B is the same as the mark of S_2 with respect to X', A, and B, which is $T(X')$, and if $X' \neq X$, then the mark of S with respect to X', A, and B is the same as the mark of S_1 with respect to X', A, and B, which is again $T(X')$. ■

Corollary 7.4.56. *Let G be a graph, let $A, B \subseteq V(G)$, and let T be a signature for G, A, B, and $C = \emptyset$. If for all components, X, of $G - (A \cup B)$ there are minimal separators S_X and S'_X (S_X and S'_X can depend on the choice of X) of G that agree with A, B, C, and T, where $S_X \cap X \neq S_2 \cap X$, then there exist minimal separators S_1 and S_2 of G that agree with A, B, C, and T, such that for all components, X, of $G - (A \cup B)$ $S_1 \cap X \neq S_2 \cap X$.*

Proof: Let G be a graph, let $A, B \subseteq V(G)$, and let T be a signature for G, A, B, and $C = \emptyset$. Assume for all components, X, of $G - (A \cup B)$ there are minimal separators S_X and S'_X (S_X and S'_X can depend on the choice of X) of G that agree with A, B, C, and

T, where $S_X \cap X \neq S_2 \cap X$.

Let S_1 and S_2 be two minimal separators of G that agree with A, B, C, and T that maximize the number of components, X, of $G - (A \cup B)$ such that $S_1 \cap X \neq S_2 \cap X$. Assume for a contradiction that there is an X, of $G-(A\cup B)$ such that $S_1 \cap X = S_2 \cap X$. By Assumption, there is an S_3 that agrees with A, B, C, and T and if $S_3' = S_3 \cap X$ and $S_2' = S_2 \cap X$ then $S_3' \neq S_2'$. By Lemma 7.4.55 $S_2'' = (S_2 - S_2') \cup S_3'$ is a minimal separator that agrees with A, B, C, and T. But then the number of components, X, of $G-(A\cup B)$ such that $S_1 \cap X \neq S_2'' \cap X$ is greater than the number of components, X, of $G-(A\cup B)$ such that $S_1 \cap X \neq S_2 \cap X$, a contradiction. ■

Lemma 7.4.57. *Let G be a graph, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, let $\mathcal{Y}$ be a set of components of $G - (A \cup B \cup C)$, let $\hat{Y} = \bigcup_{Y \in \mathcal{Y}} Y$, let T be a signature for G, A, B, and C, and let S and S' be a minimal separators of G that agrees with A, B, C, and T such that $S \cap \hat{Y} = S' \cap \hat{Y}$. Furthermore, let A^* be the set of all vertices that belong to a component of $G[\hat{Y} - S] = G[\hat{Y} - S']$ that has a neighbor in A, let B^* be the set of all vertices that belong to a component of $G[\hat{Y} - S] = G[\hat{Y} - S']$ that has a neighbor in B, and let $C^* = \hat{Y} - (A^* \cup B^*)$. Then there is a signature T' for G, $A' = A \cup A^*$, $B' = B \cup B^*$, and $C' = C \cup C^*$ such that S and S' agree with A', B', C', and T'.*

Proof: Let G be a graph, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, let $\mathcal{Y}$ be a set of components of $G - (A \cup B \cup C)$, let $\hat{Y} = \bigcup_{Y \in \mathcal{Y}} Y$, let T be a signature for G, A, B, and C, and let S and S' be a minimal separators of G that agrees with A, B, C, and T such that $S \cap \hat{Y} = S' \cap \hat{Y}$. Furthermore, let A^* be the set of all vertices that belong to a component of $G[\hat{Y} - S] = G[\hat{Y} - S']$ that has a neighbor in A, let B^* be the set of all vertices that belong to a component of $G[\hat{Y} - S] = G[\hat{Y} - S']$ that has a neighbor in B, and let $C^* = \hat{Y} - (A^* \cup B^*)$.

Now, let $C_Y = S \cap \hat{Y} = S' \cap \hat{Y}$, so S and S' are consistent with A, B, and $C \cup C_Y$ and

observe that $\mathcal{CC}(G[\hat{Y} - S]) = \mathcal{CC}(G[\hat{Y} - S'])$ is a subset of $\mathcal{CC}(G - (A \cup B \cup (C \cup C_Y)))$. So, if we let A^{**}, B^{**}, and C^{**} be the sets of vertices that belong to a component of $G - (A \cup B \cup (C \cup C_Y))$ that has neighbors in A and not B, that has neighbors in B and not A, and that does not have neighbor $B \cup A$ respectively, then it holds that $A^* \subseteq A^{**}$, $B^* \subseteq B^{**}$, and $C^* \subseteq C^{**} \cup C_Y$. Furthermore, by Lemma 7.4.50, S and S' are both consistent with $A \cup A^{**}$, $B \cup B^{**}$, and $C \cup C_Y \cup C^{**}$, hence S and S' are consistent with $A' = A \cup A^*$, $B' = B \cup B^*$, and $C' = C \cup C^*$.

Lastly, we show for each component, X, of $G - (A' \cup B' \cup C')$ that S and S' have the same mark with respect to X, A', and B'. It will then follow that there is a signature T' for G, A', B', and C' such that S and S' agree with A', B', C', and T'.

Observe that since $A^* \cup B^* \cup C^* = \hat{Y}$ we have that $\mathcal{CC}(G - (A' \cup B' \cup C')) = \mathcal{CC}(G - (A \cup B \cup C)) - \mathcal{Y}$. This implies that A^*, B^*, and C^* must be anti-complete with the component, X, of $G - (A' \cup B' \cup C')$. Hence $N_G[A'] \cap X = N_G[A] \cap X$ and $N_G[B'] \cap X = N_G[B] \cap X$ and that X must also be a component of $G - (A \cup B \cup C)$. It follows that if there are components P, Q of $G[A' \cup B']$ such that there is a path from P to Q through $X - S$, then there are components P', Q' of $G[A \cup B]$ such that $P' \subset P$ and $Q' \subset Q$ and there is a path from P' to Q' through $X - S$. Since S and S' have the same mark with respect to X, A, and B, namely $T(X)$, it follows that there is a path from P' to Q' through $X - S'$. Therefore, there is a path from P to Q through $X - S'$. Hence S and S' have the same mark with respect to X, A', and B'. ■

Lemma 7.4.58. *Let G be a graph with n vertices, let $\delta > 1$, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, such that no component of $G - (A \cup B \cup C)$ has over $n/3\delta$ vertices and $G[A \cup B]$ has at most $n/6$ components, and let G have μ minimal separators that are consistent with A, B, and C. Furthermore, let $\mathcal{X}$ be the set that contains all components, X, of $G - (A \cup B \cup C)$ such that there exists at least two minimal separators S_X and S'_X*

that are consistent with A, B, and C and $S \cap X \neq S' \cap X$ (S_X and S'_X may depend on the component X). If $|\mathcal{X}| < \delta$ then there is an induced minor G' of G with at least μ minimal separators and G' has at most $n/2$ vertices.

Proof: Let G be a graph with n vertices, let $\delta > 1$, let $A, B, C \subseteq V(G)$, $A, B \neq \emptyset$, such that no component of $G-(A\cup B\cup C)$ has over $n/3\delta$ vertices and $G[A\cup B]$ has at most $n/6$ components, and let G have μ minimal separators that are consistent with A, B, and C. Furthermore, let $\mathcal{X}$ be the set that contains all components, X, of $G - (A \cup B \cup C)$ such that there exists at least two minimal separators S_X and S'_X that are consistent with A, B, and C and $S_X \cap X \neq S'_X \cap X$, and assume that $|\mathcal{X}| < \delta$.

Let $\hat{X} = \bigcup_{X \in \mathcal{X}} X$. We first show that we can find sets $A \subseteq A'$, $B \subseteq B'$, and $C \subseteq C'$ such that $\hat{X} = V(G) - (A' \cup B' \cup C')$ and all minimal separators of G that are consistent with A, B, and C are consistent with A', B', and C'. Let $\mathcal{Y} = \mathcal{CC}(G - (A \cup B \cup C)) - \mathcal{X}$ and let $\hat{Y} = \bigcup_{Y \in \mathcal{Y}} Y$. It follows by assumption that for any two minimal separators S and S' that are consistent with A, B, and C we have that $S \cap \hat{Y} = S' \cap \hat{Y}$. Let A^* denote the set of all vertices that belong to a component of $G[\hat{Y} - S] = G[\hat{Y} - S']$ that has at least one neighbor in A, let B^* denote the set of all vertices that belong to a component of $G[\hat{Y} - S]$ $= G[\hat{Y} - S']$ that has at least one neighbor in B, and let $C^* = \hat{Y} - (A^* \cup B^*)$. Then by Lemma 7.4.57 and the fact that $G[\hat{Y} - S] = G[\hat{Y} - S']$, the μ minimal separators of G that are consistent with A, B, and C are consistent with $A' = A \cup A^*$, $B' = B \cup B^*$, and $C' = C \cup C^*$. Additionally, $\hat{Y} = (A^* \cup B^* \cup C^*)$, so we have that $\hat{X} = G - (A' \cup B' \cup C')$, as desired.

Now, by Lemma 7.4.4 $G - C'$ has μ minimal separators that are consistent with A', B', and $\emptyset$ and by Lemma 7.4.5, if G' is the graph that results from contracting each component of $G[A]$ and $G[B]$ in G, then G' has at least μ minimal separators.

Since $\hat{X} = G - (A' \cup B' \cup C')$, $|\hat{X}| \leqslant n/3$, and $G[A]$ and $G[B]$ together have at most

$n/6$ components, it follows that G' has at most $n/3+n/6 = n/2$ vertices. This completes the lemma. ■

The following lemma is a slight strengthening of Lemma 7.4.4 that we will require.

Lemma 7.4.59. *Let G be a graph, let $A, B, C \subseteq V(G)$ with $A, B \neq \emptyset$, let T be a signature for G, A, B, and C, and let S be a minimal separator that agrees with T, A, B, and C. Then $S - C$ is a minimal separator of $G - C$ that agrees with A, B, $\emptyset$, and T.*

Proof: Let G be a graph, let $A, B, C \subseteq V(G)$ with $A, B \neq \emptyset$, let T be a signature for G, A, B, and C, and let S be a minimal separator that agrees with T, A, B, and C.

By Lemma 7.4.4 we have that $S-C$ is a minimal separator of $G-C$ that is consistent with A, B, and $\emptyset$. Let $G' = G - C$. Since $G - (A \cup B \cup C) = G' - (A \cup B)$ we have that for a component, X, of $G' - (A \cup B)$, the mark of S with respect to X, A, and B in G is $T(X)$. Since $X \cap C = \emptyset$ the mark of $S - C$ with respect to X, A, and B is also $T(X)$. ■

The following lemma comes from [127], where the authors prove a tighter bound, for our purposes the following bound is easier to use and sufficient.

Lemma 7.4.60 ([127])**.** *Every graph G on n vertices has at most 2^n minimal separators.*

Lemma 7.4.61. *Let G be a k-creature free graph with n vertices, let $\omega \geqslant 1$, let $\delta = 3\omega$, let c be a natural number large enough to satisfy the inequality $400k^3\delta^2 \log^4(c) < c/6$, let $x = 400k^3\delta^2 \log^4(n)$, and let G have at least $2^c(12n)^{6k^2x^4 \log(n)}$ minimal separators. Then there exists an induced minor G' of G, $A, B \subseteq V(G')$, and a signature T for G', A, B, and $C = \emptyset$, such that the following properties hold:*

(i) There are at least ω components of $G' - (A \cup B)$.

(ii) For all components, X, of $G' - (A \cup B)$ there are two minimal separators S_X and S'_X, $S_X \cap X \neq S'_X \cap X$ (S_X and S'_X may depend on the choice of X), that agree with A, B, $C = \emptyset$, and T.

(iii) All components of $G' - (A \cup B)$ *are good with respect to* A *and* B*.*

Proof: Let $\omega > 1$, let $\delta = 3\omega$, and let c be an natural number large enough to the satisfies the inequality $400k^3\delta^2 \log^4(c) < c/6$. Note that for all $c' > c$ that $400k^3\delta^2 \log^4(c')$ $< c'/6$ also holds. We first show, by a proof by contradiction, that for any k-creature free graph G with n vertices and at least $2^c(12n)^{6k^2x^4 \log(n)}$ minimal separators, where x $= 400k^3\delta^2 \log^4(n)$, that there exists an induced minor G' of G, sets $A, B, C \subseteq V(G')$, and signature T for G', A, B, and C such that (1) there are at least ω components, X, of $G' - (A \cup B \cup C)$ such that there are minimal separators S_X and S'_X that agree with A, B, C, and T (the minimal separators may depend on X) where $S_X \cap X \neq S'_X \cap X$, and (2) all components of $G' - (A \cup B \cup C)$ are good with respect to A, B, and C.

So, assume for a contradiction, that G is a k-creature free graph with n vertices and at least $2^c(12n)^{6k^2x^4 \log(n)}$ minimal separators where G is chosen with as few vertices as possible so that no induced minor G' of G and sets $A, B, C \subseteq V(G')$, and signature T for G', A, B, and C satisfy (1) and (2). Since G has at least 2^c minimal separators, by Lemma 7.4.60 we have that $n \geqslant c$, hence n is large enough to satisfy the inequality $400k^3\delta^2 \log^4(n) < n/6$.

We now apply Lemma 7.4.54 to G and $\delta = 3\omega$ to gets sets $A, B, C \subseteq V(G)$ and signature T for G, A, B, and C that satisfy properties $(i) - (iv)$ of Lemma 7.4.54. Let $\mathcal{X}$ be the set of components, X, of $G - (A \cup B \cup C)$ such that there exists at least two minimal separators S and S' that agree with A, B, C, and T and $S \cap X \neq S' \cap X$. Since (2) holds by property *(iv)* of Lemma 7.4.54 it must be that property (1) fails, hence $|\mathcal{X}| < \omega$. So, since $400k^3\delta^2 \log^4(n) < n/6$, by property *(iii)* of Lemma 7.4.54 $G[A \cup B]$ has less than $n/6$ vertices so we may apply Lemma 7.4.58 to find an induced minor G'

of G which has $n' \leqslant n/2$ vertices at and least

$$\frac{2^c(12n)^{6k^2x^4\log(n)}}{(12n)^{6k^2x^4}} = 2^c(12n)^{6k^2x^4(\log(n)-1)} \geqslant 2^c(12n')^{6k^2x^4\log(n')}$$

minimal separators (because any minimal separator that agrees with T, A, B, and C is consistent with A, B, and C). Since G was chosen as small as possible so that no induced minor of G satisfies (1) and (2), $|V(G')| < |V(G)|$, and G' has at least $2^c(12n'x)^{2k^2x^4\log(n')}$ minimal separators, it follows that there must be an induced minor of G' that satisfies (1) and (2). But then this implies that there is an induced minor of G that satisfied (1) and (2), a contradiction.

We may then assume that G has an induced minor G' and sets $A, B, C \subseteq V(G')$, and signature T for G', A, B, and C such that (1) and (2) hold. Again, let $\mathcal{X}$ be the set of components, X, of $G' - (A \cup B \cup C)$ such that there exists at least two minimal separators S and S' of G' that agree with A, B, C, and T and $S \cap X \neq S' \cap X$. By assumption $|\mathcal{X}| \geqslant \omega$. Let $\hat{Y}$ denote the set of vertices that belong to a component, Y, of $G' - (A \cup B \cup C)$ such that $Y \notin \mathcal{X}$, hence for all minimal separators S_Y and S'_Y that agree with A, B, C, and T it holds that $S_Y \cap Y = S'_Y \cap Y$. It follows that we may apply Lemma 7.4.57 to show that there exists a partition of $\hat{Y}$ into sets A^*, B^*, and C^* and a signature T' for G', $A' = A \cup A^*$, $B' = B \cup B^*$, and $C' = C \cup C^*$ such that if S agrees with A, B, C, and T, then S agrees with A', B', C', and T'. Let $G'' = G' - C'$. We now show that G'', A', B', and T' satisfies properties *(i)* - *(iii)* of this lemma.

First, we make two needed observations which follow from how we defined $\hat{Y}$ and the fact that A^*, B^*, and C^* are a partition of $\hat{Y}$. The first is that $\mathcal{X} = \mathcal{CC}(G' - (A' \cup B' \cup C'))$ $= \mathcal{CC}(G'' - (A' \cup B'))$. The second is that for all components, X, of $G'' - (A' \cup B')$ it holds that $N^2_{G''}[A' \cup B'] \cap X = N^2_{G''}[A \cup B] \cap X = N^2_{G'}[A \cup B] \cap X$ (the first equality holds because $N_{G'}(A^*)$ and $N_{G'}(B^*)$ are contained in $A \cup B \cup C$, and $C' \subseteq C$ hence $N_{G'}(A^*)$ and

$N_{G'}(B^*)$ are contained in $A \cup B$. The second equality holds because X is a component of both $G'' - (A \cup B)$ and of $G' - (A \cup B)$) .

We now verify properties *(i)-(iii)*. Property *(i)* holds since by assumption, $|\mathcal{X}| \geqslant \omega$.

Next, we establish property *(ii)*. Let X be a component of $G'' - (A' \cup B')$, so $X \in \mathcal{X}$. It follows that in G' there are two minimal separators, S_X and S'_X where $S_X \cap X \neq S'_X \cap X$ and S_X and S'_X agree with A, B, C, and T and therefore S_X and S'_X agree with A', B', C', an T'. By Lemma 7.4.59 $S_X - C'$ and $S'_X - C'$ are minimal separators of G'' that agree with A', B', $\emptyset$, and T'. Since $X \subseteq V(G'')$ we have that $C' \cap X = \emptyset$ and it follows that $(S_X - C') \cap X \neq (S'_X - C') \cap X$. This established property *(ii)*.

Lastly, we show property *(iii)* holds. Let X be a component of $G'' - (A' \cup B')$, so X is a component of $G' - (A \cup B \cup C)$ and therefore X is good with respect to A, B, and C in G'. Since $N^2_{G''}[A' \cup B'] \cap X = N_{G'}[A \cup B] \cap X$ it follows that X is good with respect to A' and B' in G''. ■

Lemma 7.4.62. *Let G be a graph with n vertices and let $A, B, C \subseteq V(G)$. If $S \subset V(G)$ is a minimal separator that is consistent with A, B, and C then for each component, X, of $G - (A \cup B \cup C)$, $S \cap X$ is a minimal A, B-separator of $G[A \cup B \cup X]$.*

Proof: Let G be a graph with n vertices and let $A, B, C \subseteq V(G)$, let S be a minimal separator of G that is consistent with A, B, and C, and let X be a component of $G - (A \cup B \cup C)$. If $S \cap X$ is not an A, B-separator of $G[A \cup B \cup X]$, then there is a path, P from A to B through $X - S$, hence there is a path from A to B in $G - S$, so which contradicts the assumption that S is consistent with A, B, and C, hence $S \cap X$ is an A, B-separator of $G[A \cup B \cup X]$.

Now let $s \in S \cap X$ and assume for a contradiction that $(S \cap X) - s$ is an A,B-separator of $G[A \cup B \cup X]$. Let A' and B' be the S-full components of $G - S$ that contain A and B respectively. Let P_A and P_B be shortest paths from A to s and B to s such that

all internal vertices are contained in A' and B' respectively. It follows that all internal vertices of P_A and P_B belong to $G - (A \cup B \cup C \cup S)$. Therefore all internal vertices of P_A and P_B belong to $X - S$. It follows that $(S \cap X) - s$ is not an A,B-separator of $G[A \cup B \cup X]$. Hence $S \cap X$ is an A,B-minimal separator of $G[A \cup B \cup X]$. ■

We are now ready to prove Lemma 7.3.2.

Proof: [Proof of Lemma 7.3.2] Let G be a k-creature free graph with n vertices, let $\omega > 1$ and $\delta = 3\omega$, let c be an integer large enough so that $400k^3\delta^2 \log^4(c) < c/6$, let $x = 400k^3\delta^2 \log^4(n)$, and let G have at least $2^c(12n)^{6k^2x^4 \log(n)}$ minimal separators. We may then apply Lemma 7.4.61 to G and ω to get an induced minor G' of G sets $A, B \subseteq V(G')$ an signature T for G', A, B, and $C = \emptyset$ that satisfy properties *(i)-(iv)* of Lemma 7.4.61. Let ω' denote the number of components of $G' - (A \cup B)$, so $\omega' \geqslant \omega$ by property *(i)* of Lemma 7.4.61. Let H be an ω'-bistar with central vertices c_A and c_B and exactly ω' peripheral vertices, and for each component X of $G' - (A \cup B)$, let v_x denote a unique peripheral vertex of H. Let φ be the function that maps the vertices of A to c_A, the vertices of B to c_B and the vertices of $X \in \mathcal{CC}(G' - (A \cup B))$ to v_x. Lastly, for each component X of $G' - (A \cup B)$, by property *(ii)* of Lemma 7.4.61 there are two minimal separators S_X, S'_X, $S_X \cap X \neq S'_X \cap X$ that agree with A, B, $C = \emptyset$ and T. Hence, by Lemma 7.4.55 there exists minimal separators S_1 and S_2 that agree with A, B, $C = \emptyset$, and T and for each component, X, of $G' - (A \cup B)$, it holds that $S_1 \cap X \neq S_2 \cap X$. We will show that $W = (G', H, \varphi, S_1, S_2)$ is a connected, good, full generalized ω-creature.

To see that property *(i)* of Definition 7.3.1 holds let $S_1^* = S_1$ and $S_2^* = S_2$ and let u be a peripheral vertex of H. By how S_1 and S_2 were defined $\varphi^{-1}(u) \cap S_1 \neq \varphi^{-1}(u) \cap S_2$ and by Lemma 7.4.62 they are A_φ, B_φ-minimal separators in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)]$. Note that this implies that W is a full generalized ω-creature.

Property *(ii)* of Definition 7.3.1 follows from the fact that S_1 and S_2 are minimal separators that agree with A_φ, B_φ, and $C = \emptyset$.

That property *(iii)* of Definition 7.3.1 holds follows directly from the fact that S_1 and S_2 are both agree with A, B, $C = \emptyset$, and T, in particular, for every component X of $G' - (A \cup B) = G' - (A_\varphi \cup B - \varphi)$, the mark of S_1 and S_2 with respect to X, A_φ and B_φ is $T(X)$.

Next, we show that property *(iv)* of Definition 7.3.1 is satisfied. If there was a peripheral vertex, u, such that $\varphi^{-1}(u)$ did not have neighbors in both $A_\varphi = A$ and $B_\varphi = B$, then we can see that no minimal separator that is consistent with A, B, and $C = \emptyset$ would contain a vertex from $\varphi^{-1}(u)$. But S_1 and S_2 are minimal separators that are consistent with A, B, and $C = \emptyset$ and for every peripheral vertex, u, of H, we have by property *(i)* that $\varphi^{-1}(u) \cap S_1 \neq \varphi^{-1}(u) \cap S_2$, therefore $\varphi^{-1}(u) \cap S_1$ and $\varphi^{-1}(u) \cap S_2$ are not both empty sets. It follows that $\varphi^{-1}(u)$ must have neighbors in both A_φ and B_φ.

Lastly, that for every peripheral vertex $u \in H$, $\varphi^{-1}(u)$ is a connected vertex set follows directly from how we defined φ, hence W is connected. The fact that each component of $G' - (A_\varphi \cup B_\varphi) = G' - (A \cup B)$ is good follows from property *(iii)* of Lemma 7.4.61, hence W is good. That W is full was observed when proving property *(i)*.

It now follows that W is a connected, good, full generalized ω-creature. ■

7.5 Extracting Critters from Generalized Creatures

7.5.1 Generalized ω-Creatures and Their Properties

We have already seen the definition of generalized ω-creatures, and three properties that they might or might not have; being *full*, *connected* and *good*. In this subsection we introduce two more properties of generalized ω-creatures, being *disjoint* and *adhesion size* α for some integer $\alpha > 0$. We will also prove some basic properties of generalized ω-creatures that will be used in order to extract a sufficiently large critter from them.

Definition 7.5.1. (Adhesion Size) Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature. For each peripheral vertex u of H the *adhesion size* of u is the number of distinct connected components of $G[A_\varphi \cup B_\varphi]$ containing at least one neighbor of $\varphi^{-1}(u)$. The *adhesion size* of the generalized ω-creature $(G, H, \varphi, S_1, S_2)$ is the maximum adhesion size of its peripheral vertices.

Definition 7.5.2. (Disjoint Generalized ω-Creatures) Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature. If $S_1 \cap S_2 = \emptyset$ then W is called a *disjoint* generalized ω-creature.

The definition of generalized ω-creatures would lead a reader to have some expectation of what a "typical" generalized ω-creature would look like, and also imagine some strange corner cases that could occur. Over the next few lemmas we rule out some corner cases; e.g. we show (lemmas that imply) that in a generalzied ω-creature A_φ and B_φ are non-empty and that every peripheral vertex u of H satisfies that there exists a path from A_φ to B_φ through $\varphi^{-1}(u)$.

Observation 7.5.3. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature. Then, S_1 and S_2 are disjoint from A_φ and B_φ.*

Proof: The statement follows directly from property *(ii)* of generalized ω-creatures. ■

Lemma 7.5.4. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature. Then, for every peripheral vertex $u \in H$, $\varphi^{-1}(u)$ has a neighbor in A_φ and a neighbor in B_φ.*

Proof: By property *(i)* exists $S_1^\star \subseteq S_1$ and $S_2^\star \subseteq S_2$ such that $\varphi^{-1}(u) \cap S_1^\star$ and $\varphi^{-1}(u) \cap S_2^\star$ are distinct A_φ, B_φ-minimal separators in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)]$. Thus both $\varphi^{-1}(u) \cap S_1^\star$ and $\varphi^{-1}(u) \cap S_2^\star$ are non-empty. Since $\varphi^{-1}(u) \cap S_1^\star \neq \emptyset$, minimality of $\varphi^{-1}(u) \cap S_1^\star$ implies that there is a path from A_φ to B_φ in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)]$. Let P be a shortest such path, then the first vertex of P is in A_φ, the last is in B_φ, and all internal

vertices of P are in $\varphi^{-1}(u)$. Since $S_1^\star$ is disjoint from $A_\varphi \cup B_\varphi$ it follows that P has at least one internal vertex. Thus the first and last vertices of P are neighbors of $\varphi^{-1}(u)$ in A_φ and B_φ respectively. ■

Observation 7.5.5. *For every generalized ω-creature $W = (G, H, \varphi, S_1, S_2)$, peripheral vertex u of H, and component C of $G[\varphi^{-1}(u)] - (S_1 \cup S_2)$, either $N(C) \cap A_\varphi$ is empty or $N(C) \cap B_\varphi$ is empty.*

Proof: Suppose that $N(C) \cap A_\varphi \neq \emptyset$ and $N(C) \cap B_\varphi \neq \emptyset$ for some component C of $G[\varphi^{-1}(u)] - (S_1 \cup S_2)$. This contradicts property *(i)* of generalized ω-creatures. ■

Lemma 7.5.6. *Let G be a graph and (H, φ) be an ω-bistar partition of G. Let X be a vertex set in G disjoint from $A_\varphi \cup B_\varphi$. Then X is a A_φ-B_φ-separator in G if and only if $X \cap \varphi^{-1}(u)$ is a A_φ-B_φ-separator in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)]$ for every peripheral vertex u of H.*

Proof: We prove instead the equivalence of the negations: that there is a path from A_φ to B_φ in $G - X$ if and only if there exists a peripheral vertex u of H and a path from A_φ to B_φ in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)] - X$.

The backward direction is trivial. For the forward direction, let P be a shortest path from A_φ to B_φ in $G - X$. Then none of the internal vertices of P lie in $A_\varphi \cup B_\varphi$. Since $N_G(\varphi^{-1}(u)) \subseteq A_\varphi \cup B_\varphi$ for every peripheral vertex u, it follows that there exists a peripheral vertex u such that $V(P) \subseteq A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)$. This concludes the proof. ■

Definition 7.5.7 (Flipping)**.** Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature, and v be a peripheral vertex of H. *Flipping W at v* results in the tuple $W' = (G, H, \varphi, S_1', S_2')$ where

$$S_1' = (S_1 - \varphi^{-1}(v)) \cup (S_2 \cap \varphi^{-1}(v)) \text{ and } S_2' = (S_2 - \varphi^{-1}(v)) \cup (S_1 \cap \varphi^{-1}(v)).$$

Let $S_1^\star$ and $S_2^\star$ be witness separators for W. *Flipping* $S_1^\star$ and $S_2^\star$ at v results in

$$S_1'^\star = (S_1^\star - \varphi^{-1}(v)) \cup (S_2^\star \cap \varphi^{-1}(v)) \text{ and } S_2'^\star = (S_2^\star - \varphi^{-1}(v)) \cup (S_1^\star \cap \varphi^{-1}(v)).$$

Lemma 7.5.8. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature, v be a peripheral vertex of H, and $W' = (G, H, \varphi, S_1', S_2')$ be the result of flipping W at v. Then: Furthermore,*

- *W' is a generalized ω-creature.*
- *If $S_1^\star$ and $S_2^\star$ are witness separators for W and $S_1'^\star$ and $S_2'^\star$ are the result of flipping $S_1^\star$ and $S_2^\star$ at v, then $S_1'^\star$ and $S_2'^\star$ are witness separators for W'.*
- *If W is a disjoint generalized ω-creature then W' is a disjoint generalized ω-creature.*

Proof: Let $S_1^\star \subseteq S_1$ and $S_2^\star \subseteq S_2$ be witness separators for W and $S_1'^\star$ and $S_2'^\star$ be the result of flipping $S_1^\star$ and $S_2^\star$ at v.

We observe that for every peripheral vertex u of H we have that

$$\{\varphi^{-1}(u) \cap S_1, \varphi^{-1}(u) \cap S_2\} = \{\varphi^{-1}(u) \cap S_1', \varphi^{-1}(u) \cap S_2'\}$$

and that

$$\{\varphi^{-1}(u) \cap S_1^\star, \varphi^{-1}(u) \cap S_2^\star\} = \{\varphi^{-1}(u) \cap S_1'^\star, \varphi^{-1}(u) \cap S_2'^\star\}.$$

From this it immediately follows that $S_1'^\star$ and $S_2'^\star$ satisfy property *(i)* for W', and that W' additionally satisfies properties *(iii)*, and *(iv)*. For property *(ii)*, Lemma 7.5.6 implies that S_1' and S_2' are both A_φ,B_φ-separators.

We argue that all of A_φ is in the same connected component of $G - S_1'$. By assumption all of A_φ is in the same connected component of $G - S_1$. Consider an arbitrary pair a, a' of vertices in A. Since all of A_φ is in the same connected component of $G - S_1$ there

exists a sequence $a_1, a_2, \ldots a_t$ of vertices in A such that $a_1 = a$, $a_t = a'$, and for every pair a_i, a_{i+1} of consecutive vertices in the sequence it holds that $a_i a_{i+1}$ is an edge or there exists a peripheral vertex u of H such that there is a path from a_i to a_{i+1} through $\varphi^{-1}(u) - S_1$. In the second case, by property *(iii)* it holds that $a_i a_{i+1}$ there is a path from a_i to a_{i+1} through $\varphi^{-1}(u) - S_2$. Thus, since $\varphi^{-1}(u) \cap S'_1 \in \{\varphi^{-1}(u) \cap S_1, \varphi^{-1}(u) \cap S_2\}$ we have that there is a path from a_i to a_{i+1} through $\varphi^{-1}(u) - S'_1$. It follows that a and a' are in the same component of $G - S'_1$. But then all of A_φ is in the same component of $G - S'_1$. Identical proofs show that A_φ is in the same component of $G - S'_2$, and that B_φ is in one component of $G - S'_1$ and of $G - S'_2$. Hence W' satisfies property *(ii)* and is a generalized ω-creature.

Finally we show that if W is a disjoint generalized ω-creature then W' is a disjoint generalized ω-creature. Suppose that W is a disjoint generalized ω-creature. We then have that

$$S'_1 \cap S'_2 = \big((S_1 - \varphi^{-1}(v)) \cup (S_2 \cap \varphi^{-1}(v))\big) \cap \big((S_2 - \varphi^{-1}(v)) \cup (S_1 \cap \varphi^{-1}(v))\big)$$

But $(S_1 - \varphi^{-1}(v))$ is disjoint with $(S_2 - \varphi^{-1}(v)$ and $(S_2 \cap \varphi^{-1}(v))$ is disjoint with $(S_1 \cap \varphi^{-1}(v))$ because S_1 is disjoint with S_2, and $(S_1 - \varphi^{-1}(v))$ is disjoint with $(S_1 \cap \varphi^{-1}(v))$ and $(S_2 \cap \varphi^{-1}(v))$ is disjoint with $(S_2 - \varphi^{-1}(v))$ because $\varphi^{-1}(v)$ is disjoint from its complement. So S'_1 and S'_2 are disjoint and hence W' is a disjoint generalized ω-creature. ■

7.5.2 Properties of Good Connected Generalized ω-Creatures

Let $(G, H, \varphi, S_1, S_2)$ be a good connected generalized ω-creature and let X be a component of $G - (A_\varphi \cup B_\varphi)$. Since X is good with respect to A and B we have that X has at most one non-leaf sub-component. We now show that X has precisely one non-leaf sub-component.

Lemma 7.5.9. *Let $W = (G, H, \varphi, S_1, S_2)$ be a good connected generalized ω-creature and let X be a component of $G-(A_\varphi \cup B_\varphi)$. Then X has precisely one non-leaf sub-component Y.*

Proof: By definition of good, X has at most one non-leaf sub-component. By Lemma 7.5.4 X has a neighbor in A_φ and a neighbor in B_φ. Let $X_A = N_G^2[A_\varphi] \cap X$ and $X_B = N_G^2[B_\varphi] \cap X$. We have that X_A and X_B are non-empty, and since W is good it follows that X_A and X_B are anti-complete. Since W is connected there exists a path P from X_A to X_B through $X - (X_A \cup X_B)$. Let P' be the sub-path of P obtained by removing the endpoints of P. Then P' is a connected set in $X - N_G^2[A_\varphi \cup B_\varphi]$ with neighbors in two distinct components of $G[X \cap N_G^2[A_\varphi \cup B_\varphi]]$. Thus P' is contained in a non-leaf sub-component of X. ■

In light of Lemma 7.5.9 we can give the unique non-leaf sub-component of X a name.

Definition 7.5.10. Let $(G, H, \varphi, S_1, S_2)$ be a good connected generalized ω-creature and let X be a component of $G - (A_\varphi \cup B_\varphi)$. Then the *kernel* of X is the unique non-leaf sub-component of X.

Lemma 7.5.11. *Let $(G, H, \varphi, S_1, S_2)$ be a good generalized ω-creature, and X be a component of $G-(A_\varphi \cup B_\varphi)$. Then, for each connected component C of $G[X \cap N_G^2[A_\varphi \cup B_\varphi]]$ there is precisely one component Y in $G[A_\varphi \cup B_\varphi]$ such that $N_G[X] \cap Y \neq \emptyset$.*

Proof: We first argue that there exist at least one component Y in $G[A_\varphi \cup B_\varphi]$ such that $N_G[X] \cap Y \neq \emptyset$. Let $x \in X$. Since $x \notin (A_\varphi \cup B_\varphi)$ we have that $x \in N_G(A_\varphi \cup B_\varphi)$ or $x \in N_G^2(A_\varphi \cup B_\varphi)$. If $x \in N_G^2(A_\varphi \cup B_\varphi)$ then x has a neighbor y in $N_G(A_\varphi \cup B_\varphi)$, and $y \in X$. We may then choose y as x instead, and therefore, without loss of generality $x \in N_G(A_\varphi \cup B_\varphi)$. x has a neighbor in $A_\varphi \cup B_\varphi$ establishing the existence of at least one component Y in $G[A_\varphi \cup B_\varphi]$ such that $N_G[X] \cap Y \neq \emptyset$.

We now prove that Y is unique. Suppose for contradiction that there exists a component $Y' \neq Y$ of $G[A_\varphi \cup B_\varphi]$ such that $N_G[X] \cap Y' \neq \emptyset$. Let P be a path from $N_G(Y) \cap X$ to $N_G(Y') \cap X$. Let q be the first vertex on P such that $q \in N_G^2[Y'']$ for some component $Y'' \neq Y$ of $G[A_\varphi \cup B_\varphi]$. If q is the first vertex of P then $q \in N_G(Y)$ contradicting that $N_G^2[Y] \cap N_G^2[Y''] \cap X$ is empty (because the generalized ω-creature is good). Otherwise, let p be the predecessor of q on P. We have that $p \in N_G^2[Y]$ because $p \in N_G^2[A_\varphi \cup B_\varphi]$ and q is the *first* on P such that $q \in N_G^2[Y'']$ for some component $Y'' \neq Y$ of $G[A_\varphi \cup B_\varphi]$. But then p and q contradict that there is no edge from $N_G^2[Y] \cap X$ to $N_G^2[Y''] \cap X$ (which should have been true, because the generalized ω-creature is good). ■

Lemma 7.5.12. *Let $(G, H, \varphi, S_1, S_2)$ be a good generalized ω-creature and X be a component of $G - (A_\varphi \cup B_\varphi)$. For every pair C_1, C_2 of distinct components of $G[A_\varphi \cup B_\varphi]$ and path P from C_1 to C_2 through X, P contains an internal vertex in the kernel of X.*

Proof: Let s and t be the first and last vertex of P, respectively. Since $N_2^G[C_1]$ and $N_2^G[C_2]$ are disjoint and anti-complete, P has at least 7 vertices. Let s' and t' be the successor of s on P and the predecessor of t on P, respectively. Let x be the last vertex of P in $N_2^G[C_1] \cap X$. The vertex x is well defined because s' is in $N_2^G[C_1] \cap X$. Let y be the first vertex in $N_G^2[A_\varphi \cup B_\varphi - C_1] \cap X$ on the sub-path of P from x to t. Since $t' \in N_G^2[C_2] \cap X$, y is well defined. Since $N_2^G[C_1] \cap X$ and $N_2^G[A_\varphi \cup B_\varphi - C_1] \cap X$ are anti-complete we have $y \neq x$ and xy is not an edge of G. Let P' be the subpath of P from x to y. Since xy is not an edge, P' contains at least one internal vertex. Furthemore P' is a path from $N_G^2[C_1] \cap X$ to $N_G^2[A_\varphi \cup B_\varphi - C_1] \cap X$ through $X - N_G^2[A_\varphi \cup B_\varphi]$. Thus all the internal vertices of P' are contained in a non-leaf sub-component of X, namely the kernel of X, as claimed. ■

Lemma 7.5.13. *Let G be a k-creature free graph, $(G, H, \varphi, S_1, S_2)$ be a good generalized ω-creature, and X be a component of $G - (A_\varphi \cup B_\varphi)$. Let C be the kernel of X. Then*

each connected component Y of $G[X \cap N_G^2[A_\varphi \cup B_\varphi]]$ has at least one neighbor in C.

Proof: Since $G[X]$ is connected there exists a path in $G[X]$ that starts in Y and ends in C. Let P be a shortest such path. In particular, only the first vertex of P is in Y and only the last is in C. If P has no internal vertices then the first and last vertex of P are adjacent, proving the statement of the lemma. We now show that P has no internal vertices.

Suppose for contradiction that P has an internal vertex, let a be the endpoint of P which is in Y, and let q be the internal vertex in P which is adjacent to a. Since $q \in X - Y$ and Y is a connected component of $G[X \cap N_G^2[A_\varphi \cup B_\varphi]]$ it follows that $q \notin N_G^2[A_\varphi \cup B_\varphi]$. Then q is in a connected component Z of $G[X - N_G^2[A_\varphi \cup B_\varphi]]$. Since $q \notin C$ we have that $Z \neq C$ and so Z is not the kernel of X. But then $N_G(Z) \subseteq Y$ and the first vertex on P outside of Z is in Y, contradicting that only the first vertex of P is in Y. We conclude that P has no internal vertices, and this shows the statement of the lemma. ■

7.5.3 Dissolving a Peripheral Vertex

Definition 7.5.14. Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature, and let u be a peripheral vertex of H. *Dissolving* u in H produces a tuple $W' = (G', H', \varphi', S_1', S_2')$ where:

- $G' = G - (\varphi^{-1}(u) - (A_1(W) \cup B_1(W)))$,
- $H' = H - u$.

- For all $v \in (V(G')$,

$$\varphi'(v) = \begin{cases} c_A \text{ if } v \in \varphi^{-1}(u) \cap A_1(W) \\ c_B \text{ if } v \in \varphi^{-1}(u) \cap B_1(W) \\ \varphi'(v) \text{ otherwise} \end{cases}$$

- $S_1' = S_1 - \varphi^{-1}(u)$ and $S_2' = S_2 - \varphi^{-1}(u)$.

Lemma 7.5.15. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature and let u be a peripheral vertex of H. Let $W' = (G', H', \varphi', S_1', S_2')$ be the result of dissolving u in W. Then H' is a $(w-1)$-bistar partition of G'. Furthermore, for every peripheral vertex v of H' it holds that*

- $N_{G'}[\varphi'^{-1}(v)] = N_G[\varphi^{-1}(v)]$,
- $N_{G'}[\varphi'^{-1}(v)] \cap A_{\varphi'} = N_G[\varphi^{-1}(v)] \cap A_\varphi$,
- *and* $N_{G'}[\varphi'^{-1}(v)] \cap B_{\varphi'} = N_G[\varphi^{-1}(v)] \cap B_\varphi$.

Proof: We first show that H' is a $(w-1)$-bistar partition of G'. Let $xy \in E(G')$. If neither $\varphi(x)$ nor $\varphi(y)$ is equal to u then $\varphi'(x) = \varphi(x)$ and $\varphi'(y) = \varphi(y)$ so $\varphi'(x) = \varphi'(y)$ or $\varphi'(x)\varphi'(y) \in E(H')$. If both $\varphi(x)$ and $\varphi(y)$ are equal to u then $\{x, y\} \subseteq (A_1(W) \cup B_1(w)) \cap \varphi^{-1}(u)$. But $A_1(W)$ and $B_1(w)$ are disjoint and anticomplete so $\{x, y\} \subseteq A_1(W) \cap \varphi^{-1}(u)$ or $\{x, y\} \subseteq B_1(W) \cap \varphi^{-1}(u)$. In the first case $\varphi'(x) = \varphi'(y) = c_A$, in the second $\varphi'(x) = \varphi'(y) = c_B$. If $\varphi(x) = u$ and $\varphi(y) \neq u$ then $\varphi(y) \in \{c_A, c_B\}$. If $\varphi(y) = c_A$ then $y \in A_\varphi$ and hence, since $xy \in E(G)$, $x \in A_1(W)$. But then $\varphi'(x) = \varphi'(y) = c_A$. If $\varphi(y) = c_B$ then $y \in B_\varphi$ and hence, since $xy \in E(G)$, $x \in B_1(W)$. But then $\varphi'(x) = \varphi'(y) = c_B$. Finally, $V(H') = V(H) - \{u\}$. Thus we conclude that H' is a $(w-1)$-bistar partition of G'.

For the second part of the statement, We have that $\varphi'^{-1}(v) = \varphi^{-1}(v)$ and that $\varphi^{-1}(u) = \varphi^{-1}(v)$ are disjoint and anticomplete. Furthermore, since $(H'\varphi')$ is an $(\omega-1)$-bistar partition of G' we have that $N_{G'}(\varphi'^{-1}(v)) \subseteq A_{\varphi'} \cup B_{\varphi'}$. Since $A_{\varphi'} \subseteq A_\varphi \cup \varphi^{-1}(u)$ and $B_{\varphi'} \subseteq B_\varphi \cup \varphi^{-1}(u)$ it follows that $N'_G(\varphi'^{-1}(v)) \subseteq A_\varphi \cup B_\varphi$. But then $N_{G'}[\varphi'^{-1}(v)] = N_G[\varphi^{-1}(v)]$, $N_{G'}[\varphi'^{-1}(v)] \cap A_{\varphi'} = N_G[\varphi^{-1}(v)] \cap A_\varphi$, and $N_{G'}[\varphi'^{-1}(v)] \cap B_{\varphi'} = N_G[\varphi^{-1}(v)] \cap B_\varphi$. This completes the proof. ■

Lemma 7.5.16. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature and let u be a peripheral vertex of H. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the result of dissolving u in W. Then*

- *W' is a generalized $(\omega-1)$-creature. Furthermore,*
- *if W is a full generalized ω-creature then W' is a full generalized $(\omega-1)$-creature,*
- *if W is a good generalized ω-creature then W' is a good generalized $(\omega-1)$-creature,*
- *if W is a disjoint generalized ω-creature then W' is a disjoint generalized $(\omega-1)$-creature,*
- *if W is a good connected generalized ω-creature then W' is a good connected generalized $(\omega-1)$-creature,*
- *if W has adhesion size α then W' has adhesion size α. For every peripheral vertex v of H' its adhesion size in W' is at most its adhesion size in W.*

Proof: By Lemma 7.5.15 we have that H' is a $(w-1)$-bistar partition of G'. We now check the properties of generalized $(\omega-1)$ creatures for W'.

- For property *(i)* let $S_1^\star, S_2^\star$ be witness separators for W, and set $S_1'^\star = S_1^\star - \varphi^{-1}(u)$ and $S_2'^\star = S_2^\star - \varphi^{-1}(u)$. We claim that $S_1'^\star$ and $S_2'^\star$ are witness separators for W'. Let v' be a peripheral vertex of H'. Let $Z = N_G[\varphi^{-1}(v)]$. By Lemma 7.5.15 we have

$Z = N_{G'}[\varphi'^{-1}(v)]$. Since $S_1^\star$ and $S_2^\star$ are witness separators for W it follows that $S_1^\star \cap \varphi^{-1}(v)$ and $S_2^\star \cap \varphi^{-1}(v)$ are distinct minimal A_φ, B_φ-separators in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(v)]$. Then $S_1^\star \cap \varphi^{-1}(v)$ and $S_2^\star \cap \varphi^{-1}(v)$ are distinct minimal $A_\varphi \cap Z$,$B_\varphi \cap Z$-separators in $G[(A_\varphi \cup B_\varphi \cup \varphi^{-1}(v)) \cap Z]$. Since $Z = N_{G'}[\varphi'^{-1}(v)]$ and $S_1'^\star \cap Z = S_1^\star \cap Z$ and $S_2'^\star \cap Z = S_2^\star \cap Z$ it follows that $S_1'^\star \cap \varphi'^{-1}(v)$ and $S_2'^\star \cap \varphi'^{-1}(v)$ are distinct minimal $A_{\varphi'} \cap Z$,$B_{\varphi'} \cap Z$-separators in $G'[(A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)) \cap Z]$. Since every path from $A_{\varphi'}$ to $B_{\varphi'}$ through $\varphi'^{-1}(v)$ is a path from $A_{\varphi'} \cap Z$ to $B_{\varphi'} \cap Z$ through $\varphi'^{-1}(v) \cap Z$ it follows that $S_1'^\star \cap \varphi'^{-1}(v)$ and $S_2'^\star \cap \varphi'^{-1}(v)$ are distinct minimal $A_{\varphi'}$,$B_{\varphi'}$-separators in $G'[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v) \cap Z]$.

- For property *(ii)* we observe that

$$A_1(W') \cap \varphi^{-1}(u) = A_{\varphi'} \cap \varphi^{-1}(u) = A_1(W) \cap \varphi^{-1}(u),$$

$$B_1(W') \cap \varphi^{-1}(u) = B_{\varphi'} \cap \varphi^{-1}(u) = B_1(W) \cap \varphi^{-1}(u),$$

  $A_{\varphi'} - \varphi^{-1}(u) = A_\varphi - \varphi^{-1}(u)$, $B_{\varphi'} - \varphi^{-1}(u) = B_\varphi - \varphi^{-1}(u)$, and $G' - \varphi^{-1}(u) = G - \varphi^{-1}(u)$. Thus $A_1(W') = A_1(W)$ and $B_1(W') = B_1(W)$. Since $A_{\varphi'} \subseteq A_1(W)$ and $B_{\varphi'} \subseteq B_1(W)$ we conclude that W' satisfies property *(ii)*.

- For property *(iii)* let C_1 and C_2 be components of $G[A'_\varphi \cup B'_\varphi]$ and let v be a peripheral vertex of H'. Let $Z = N'_G[\varphi' {-} 1(v)]$, by Lemma 7.5.15 we have $Z = N_G[\varphi {-} 1(v)]$. Suppose there is a path P in G' from C_1 to C_2 through $\varphi'^{-1}(v) - S'_1$. Then P is a path in G' from $C_1 \cap Z$ to $C_2 \cap Z$ through $(\varphi'^{-1}(v) - S'_1) \cap Z$. Hence P is a path in G from $C_1 \cap Z$ to $C_2 \cap Z$ through $(\varphi^{-1}(v) - S'_1) \cap Z$. But $\varphi^{-1}(v) \cap S'_1 = \varphi^{-1}(v) = \cap S_1$, so P is a path in G from C_1 to C_2 through $(\varphi^{-1}(v) - S_1)$.

  By Property *(iii)* applied to W there exists a path P' from C_1 to C_2 through $(\varphi^{-1}(v) {-} S_2)$. Then P' is a path in G from $C_1 \cap Z$ to $C_2 \cap Z$ through $(\varphi^{-1}(v) {-} S_2) \cap Z$.

Hence P' is a path in G' from $C_1 \cap Z$ to $C_2 \cap Z$ through $(\varphi'^{-1}(v) - S_2) \cap Z$. But $\varphi'^{-1}(v) \cap S_2 = \varphi'^{-1}(v) = \cap S'_2$, so P' is a path in G' from C_1 to C_2 through $(\varphi'^{-1}(v) - S'_2)$.

The proof that if there is a path P in G' from C_1 to C_2 through $\varphi'^{-1}(v) - S'_2$ then there exists a path P' in G' from C_1 to C_2 through $\varphi'^{-1}(v) - S'_1$ is symmetric.

- For property *(iv)* consider a peripheral vertex v of H' and a component C_A of $G[A_{\varphi'}]$ that has a neighbor in $\varphi'^{-1}(v)$ in G'. Let x be a vertex in C_A that has a neighbor in $\varphi'^{-1}(v)$ in G'. Since $\varphi'^{-1}(v)$ and $\varphi^{-1}(u)$ are disjoint and anticomplete, and $x \in A_\varphi \cup \varphi^{-1}(u)$ it follows that $x \in A_\varphi$. Let C'_A be the component of $G[A_\varphi]$ that contains x. Since $\varphi^{-1}(v) = \varphi'^{-1}(v)$ the component C'_A has a neighbor in $\varphi^{-1}(v)$ in G. By Property *(iv)* applied to W there is a path P from C'_A to B_φ through $\varphi^{-1}(v)$.

  Since $C'_A \subseteq A_\varphi \subseteq A_{\varphi'}$, $B_\varphi \subseteq B_{\varphi'}$, and $\varphi^{-1}(v) = \varphi'^{-1}(v)$, it follows that P is a path in G' from C'_A to $B_{\varphi'}$ through $\varphi'^{-1}(v)$. But $C'_A \subseteq A_\varphi \subseteq A_{\varphi'}$ implies that $C'_A \subseteq C_A$ so P is a path in G' from C_A to $B_{\varphi'}$ through $\varphi'^{-1}(v)$.

  The proof that if a component C_B of $G[B_{\varphi'}]$ has a neighbor in $\varphi'^{-1}(v)$ in G', then there exists a path in G' from C_B to $A_{\varphi'}$ through $\varphi'^{-1}(v)$ is symmetric.

Next we verify that whenever W is full, or good, or disjoint, or connected, or has adhesion size α then W' has the same property.

- First, suppose that W is full and let $S_1^\star = S_1, S_2^\star = S_2$ be witness separators for W. In the proof that W' had property *(i)* we showed that $S_1'^\star = S_1^\star - \varphi^{-1}(u)$ and $S_2'^\star = S_2^\star - \varphi^{-1}(u)$ are witness separators for W'. Then $S_1'^\star = S_1 - \varphi^{-1}(u) = S'_1$ and $S_2'^\star = S_2 - \varphi^{-1}(u) = S'_2$, so W' is full as well.

- Suppose now that W is disjoint. Then $S_1 \cap S_2 = \emptyset$. Since $S'_1 \subseteq S_1$ and $S'_2 \subseteq S_2$, W' is also disjoint.

- If W is connected then, for every peripheral vertex v of H', $G'[\varphi'^{-1}(v)] = G[\varphi^{-1}(v)]$ so W' is connected as well.

- For bounding the adhesion size of W' let v be a peripheral vertex of H'. By Lemma 7.5.15 we have that $N_{G'}(\varphi'^{-1}(v)) = N_G(\varphi^{-1}(v)) \subseteq A_\varphi \cup B_\varphi$. Since $A_\varphi \cup B_\varphi \subseteq A_{\varphi'} \cup B_{\varphi'}$, every connected component of $A_\varphi \cup B_\varphi$ is contained in some connected component of $A_{\varphi'} \cup B_{\varphi'}$. Hence the adhesion size of v in W' is at most v's adhesion size in W. Therefore, if W has adhesion size α then the adhesion size of W' is at most α.s

- If W is a good connected generalized ω-creature we have already shown that W' is connected. We now show that W' is also good. Let v be a peripheral vertex of H' and $X = \varphi'^{-1}(v)$. Since $X = \varphi'^{-1}(v) = \varphi^{-1}(v)$ and W is connected it follows that X is a component of $G - (A_\varphi \cup B_\varphi)$. Since W' is connected X is a component of $G' - (A_{\varphi'} \cup B_{\varphi'})$. By Lemma 7.5.15 we have that $N_{G'}[X] = N_G[X]$, $N_{G'}X] \cap A_{\varphi'} = N_G[X] \cap A_\varphi$, and $N_{G'}[X] \cap B_{\varphi'} = N_G[X] \cap B_\varphi$. Thus $N^2_{G'}[A_{\varphi'} \cup B_{\varphi'}] \cap X = N^2_G[A_\varphi \cup B_\varphi] \cap X$. Hence every sub-component Y of X in G' with respect to $A_{\varphi'}$ and $B_{\varphi'}$ is also a sub-component of X in G with respect to A_φ and B_φ. Since W is good it follows that X has at most one non-leaf sub-component in G with respect to A_φ and B_φ. Thus X has at most one non-leaf sub-component in G' with respect to $A_{\varphi'}$ and $B_{\varphi'}$

  Suppose now for contradiction that there are two components C_1 and C_2 of $G'[A_{\varphi'} \cup B_{\varphi'}]$ such that $N^2_{G'}[C_1] \cap X$ is not anticomplete with $N^2_{G'}[C_2] \cap X$. Then there is a path P on at most 6 vertices from C_1 to C_2 through X. Let x be the first vertex of

P and y be the last vertex of P. By Lemma 7.5.15 both x and y are in $A_\varphi \cup B_\varphi$. But then P is a path on at most 6 vertices from x to y through X in G. Since x and y are in different components of $G'[A_{\varphi'} \cup B_{\varphi'}]$, they are also in different components C'_1 and C'_2 of $G[A_\varphi \cup B_\varphi]$. But then P is a path on at most 6 vertices from C'_1 to C'_2 through X in G, contradicting that $N^2_G[C'_1] \cap X$ is anticomplete with $N^2_G[C'_2] \cap X$.

■

7.5.4 Absorbing a Component

Definition 7.5.17. Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature with $\omega \geqslant 2$. We say a component C of $G[A_\varphi \cup B_\varphi]$ is *absorbable* if there exists a peripheral vertex, u, of H where $N_G(C) \subseteq \varphi^{-1}(u)$.

Let C be an absorbable component of $G[A_\varphi \cup B_\varphi]$ and let $u \in H$ be the vertex such that $N_G(C) \subseteq \varphi^{-1}(u)$. *Absorbing* C in W produces a tuple $W' = (G, H, \varphi', S_1, S_2)$ where $\varphi'(x) = \varphi(x)$ for all $x \in G - C$ and $\varphi'(x) = v$ for all $x \in C$.

Lemma 7.5.18. *Let $\omega \geqslant 2$ and $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature, and C be an absorbable component of $G[A_\varphi \cup B_\varphi]$. Let $W' = (G, H, \varphi', S_1, S_2)$ be the result of absorbing U in W. Then*

- *W' is a generalized ω-creature. Furthermore,*
- *if W is a full generalized ω-creature then W' is a full generalized ω-creature,*
- *if W is a good generalized ω-creature then W' is a good generalized ω-creature,*
- *if W is a disjoint generalized ω-creature then W' is a disjoint generalized ω-creature,*
- *if W is a connected generalized ω-creature then W' is a connected generalized ω-creature,*

- *if W has adhesion size α then W' has adhesion size α. For every peripheral vertex v of H its adhesion size in W' is at most its adhesion size in W.*

Proof: We prove the statement of the lemma for C being a component of $G[A_\varphi]$. We show that (H, φ') is an ω-bistar partition of G. Let $xy \in E(G)$. If x and y are both in C then $\varphi'(x) = \varphi'(y) = v$. If neither x nor y are in C then $\varphi'(x) = \varphi(x)$, $\varphi'(y) = \varphi(y)$ and therefore either $\varphi'(x)\varphi'(y)$ is an edge of H or $\varphi'(x) = \varphi'(y)$. If $x \in C$ and $y \notin C$ then $\varphi'(y) = \varphi(y) = v$ and $\varphi'(x) = v$. Hence (H, φ') is an ω-bistar partition of G.

Let u be the peripheral vertex of H such that $N_G(C) \subseteq \varphi^{-1}(u)$. Before proving that W' satisfies the properties of generalized ω-creatures we show that $A_{\varphi'}$ is non-empty. Since $\omega \geqslant 2$ there exists a peripheral vertex $v \neq u$ of H. By Lemma 7.5.4 v has a neighbor in A_φ. Since $v \neq u$ this neighbor is not in C, and hence $A_{\varphi'}$ is non-empty.

Let $S_1^\star \subseteq S_1$ and $S_2^\star \subseteq S_2$ be witness separators for W. We now proceed to verify that W' satisfies the properties of generalized ω-creatures.

- For property *(i)* let v be a peripheral vertex of H. Suppose first $v \neq u$. Then $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)] = G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(v)] - C$. Further, C is a connected component of $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(v)]$ since $N_G(C) \subseteq \varphi^{-1}(u)$. So $S_1^\star$ and $S_2^\star$ are minimal $A_{\varphi'}$,$B_{\varphi'}$-separators in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)]$.

  We now consider the case that $v = u$. We have that $S_1^\star$ and $S_2^\star$ separate A_φ from B_φ. Additionally $A_{\varphi'} \subseteq A_\varphi$ and $B_{\varphi'} \subseteq B_\varphi$. So $S_1^\star$ and $S_2^\star$ are $A_{\varphi'}$,$B_{\varphi'}$ separators in G and hence $S_1^\star \cap \varphi'^{-1}(v)$ and $S_2^\star \cap \varphi'^{-1}(v)$ are $A_{\varphi'}$,$B_{\varphi'}$-separators in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)]$.

  We prove that $S_1^\star \cap \varphi^{-1}(v)$ is a *minimal* $A_{\varphi'}$,$B_{\varphi'}$-separator in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)]$. Suppose for contradiction that a proper subset $\hat{S}_1$ of $S_1^\star \cap \varphi^{-1}(v)$ is also an $A_{\varphi'}$,$B_{\varphi'}$-separator in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)]$. Then there exists a partition of $A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)$ into L, R, and $\hat{S}_1$ such that $A_{\varphi'} \subseteq L$, $B_{\varphi'} \subseteq R$, and there are no edges from L to R. Since $G[C]$ is connected we have that $C \subseteq L$ or $C \subseteq R$. If $C \subseteq L$ then $\hat{S}_1$ is

separates $A_{\varphi'} \cup C = A_\varphi$ from $B_{\varphi'} = B_\varphi$, contradicting that $S_1^\star \cap \varphi^{-1}(v)$ is a minimal A_φ,B_φ-separator in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(v)]$.

Thus $C \subseteq R$. However, since $A_{\varphi'}$ is non-empty and A_φ is contained in a connected component of $G - S_1$ (by property *(ii)* applied to W), there is a path P' from $A_\varphi - C = A_{\varphi'}$ to C in $G[A_1(W)]$. Since $A_1(W)$ is disjoint from B_φ and $N_G(C) \subseteq \varphi^{-1}(u) = \varphi^{-1}(v)$ the path P' is a path from $A_{\varphi'}$ to C through $\varphi^{-1}(v) - S_1$. Since $V(P')$ is disjoint from S_1 and $\hat{S}_1 \subseteq S_1$ it follows that all of P' must lie in R. But then the endpoint of P' in $A_{\varphi'}$ must be in R, contradicting that $A_{\varphi'} \subseteq L$. Hence $S_1^\star \cap \varphi^{-1}(v)$ is a minimal $A_{\varphi'}$,$B_{\varphi'}$-separator in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)]$.

The proof that $S_2^\star \cap \varphi^{-1}(v)$ is a minimal $A_{\varphi'}$,$B_{\varphi'}$-separator in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)]$ is symmetric.

- For property *(ii)* it is sufficient to observe that

$$A_{\varphi'} = A_\varphi - C \subseteq A_1(W) \text{ and } A_{\varphi'} = A_\varphi - C \subseteq A_2(W),$$

$$B_{\varphi'} = B_\varphi \subseteq B_1(W) \text{ and } B_{\varphi'} = B_\varphi \subseteq B_2(W).$$

Further $A_1(W)$ and $B_1(W)$ are components of $G - S_1$, while $A_2(W)$ and $B_2(W)$ are components of $G - S_2$.

- For property *(iii)* let v be a peripheral vertex of H, and C_1 and C_2 be components of $G[A_{\varphi'(v)} \cup B_{\varphi'}]$ such that there is a path P from C_1 to C_2 through $\varphi'^{-1}(v) - S_1$. Note that C_1 and C_2 are also components of $G[A_{\varphi(v)} \cup B_\varphi]$ and that $C_1 \neq C$ and $C_2 \neq C$ because they are components of $G[A_{\varphi'} \cup B_{\varphi'}]$.

If $V(P)$ does not intersect C then P is a path in G from C_1 to C_2 through $\varphi^{-1}(v) - S_1$. By Property *(iii)* applied to W there exists a path P' in G from C_1 to C_2

through $\varphi^{-1}(v) - S_2 \subseteq \varphi'^{-1}(v) - S_2$.

If $V(P)$ intersects C then $C \subseteq \varphi'^{-1}(v)$ so $v = u$. We have that P contains a path P_1 from C_1 to C through $\varphi'^{-1}(v) - (S_1 \cup C)$ and a path P_2 from C to C_2 through $\varphi'^{-1}(v) - (S_1 \cup C)$. But $\varphi'^{-1}(v) - C = \varphi^{-1}(v)$ so P_1 is a path from C_1 to C through $\varphi^{-1}(v) - S_1$ and P_2 is a path from C to C_2 through $\varphi^{-1}(v) - S_1$. By Property *(iii)* applied to W there exist paths P_1' from C_1 to C through $\varphi^{-1}(v) - S_2$ and P_2' from C to C_2 through $\varphi^{-1}(v) - S_2$. But then C_1 and C are in the same component of $G[C_1 \cup C \cup C_2 \cup \varphi^{-1}(v)] - S_2$ and C and C_2 are in the same component of $G[C_1 \cup C \cup C_2 \cup \varphi^{-1}(v)] - S_2$, so C_1 and C_2 are in the same component of $G[C_1 \cup C \cup C_2 \cup \varphi^{-1}(v)] - S_2$. We have that $C \cup \varphi^{-1}(v) = \varphi'^{-1}(v)$ so there exists a path P' from C_1 to C_2 through $\varphi'^{-1}(v) - S_2$.

The proof that if there exists a path from C_1 to C_2 through $\varphi'^{-1}(v) - S_2$ then there exists a path from C_1 to C_2 through $\varphi'^{-1}(v) - S_1$ is symmetric.

- For property *(iv)* let C_B be a component of $G[A_{\varphi'} \cup B_{\varphi'}]$ and v be a peripheral vertex of H such that C_B has a neighbor in $\varphi'^{-1}(v)$.

  We first prove that if C_B is a component of $G[A_{\varphi'}]$ then there exists a path from C_B to $B_{\varphi'}$ through $\varphi'^{-1}(v)$. We have that C_B is also a component of $G[A_\varphi]$ and, and that C and C_B are anti-complete. Since $\varphi^{-1}(v) = \varphi'^{-1}(v) - C$ it follows that C_B has a neighbor in $\varphi^{-1}(v)$. By property *(iv)* applied to W there is a path P from C_B to B_φ through $\varphi^{-1}(v)$. Since $B_{\varphi'} = B_\varphi$ this path P is also a path from C_B to B_φ through $\varphi'^{-1}(v)$.

  We prove that if C_B is a component of $G[B_{\varphi'}]$ then there exists a path from C_B to $A_{\varphi'}$ through $\varphi'^{-1}(v)$. We have that C_B is also a component of $G[B_\varphi]$, and that C and C_B are anti-complete. Since $\varphi^{-1}(v) = \varphi'^{-1}(v) - C$ it follows that C_B has a neighbor in $\varphi^{-1}(v)$. By property *(iv)* applied to W there is a path P from C_B to

A_φ through $\varphi^{-1}(v)$. Let x be the endpoint of P in A_φ. If $x \notin C$ then $x \in A_{\varphi'}$ and thus P is a path from C_B to $A_{\varphi'}$ through $\varphi'^{-1}(v)$.

If $x \in C$ then $v = u$ (since C has a neighbor in $\varphi^{-1}(v)$) and $C \subseteq \varphi'^{-1}(v)$. Since $A_{\varphi'}$ is non-empty and A_φ is contained in a connected component of $G - S_1$ (by property *(ii)* applied to W), there is a path P' from C to $A_\varphi - C = A_{\varphi'}$ in $G[A_1(W)]$. Since $A_1(W)$ is disjoint from B_φ and $N_G(C) \subseteq \varphi^{-1}(u) = \varphi^{-1}(v)$ the path P' is a path from C to $A_{\varphi'}$ through $\varphi^{-1}(v)$.

Consider now the walk P'' that starts in C_B, follows P to C, goes through C to the startpoint of P' and then follows P' to $A_{\varphi'}$. Since all internal vertices of P are in $\varphi^{-1}(v) \subseteq \varphi'^{-1}(v)$, C is a subset of $\varphi'^{-1}(v)$, and all internal vertices of P' are in $\varphi^{-1}(v) \subseteq \varphi'^{-1}(v)$ we have that P'' is a walk from C_B to $A_{\varphi'}$ through $\varphi'^{-1}(v)$. Then $V(P'')$ contains a path from C_B to $A_{\varphi'}$ through $\varphi'^{-1}(v)$.

Next we verify that whenever W is full, or good, or disjoint, or connected, or has adhesion size α then W' has the same property.

- If W is a full generalized ω-creature, then $S_1^\star = S_1$ and $S_2^\star = S_2$, so W' is also a full generalized ω-creature.

- If W is a disjoint generalized ω-creature, then $S_1 \cap S_2 = \emptyset$ and therefore W' is also disjoint generalized ω-creature.

- If W is a connected generalized ω-creature, then for every peripheral vertex $v \neq u$, $G[\varphi'^{-1}(v)] = G[\varphi^{-1}(v)]$ is connected. Furthermore, $G[\varphi'^{-1}(u)] = G[\varphi^{-1}(u) \cup C]$, $G[\varphi^{-1}(u)]$ is connected because W is connected, $G[C]$ is connected because it is a connected component. Finally C has a neighbor in $G[\varphi^{-1}(u)]$ because $N_G(C) \subseteq \varphi^{-1}(u)$, $A_\varphi - C$ is non-empty, and $A_1(W)$ is a connected subgraph of G that contains

A_φ. Hence $G[\varphi'^{-1}(u)]$ is connected and therefore then W' is a connected generalized ω-creature.

- Suppose that W is a good connected generalized ω-creature. Let v be a peripheral vertex of H. We show that $\varphi'^{-1}(v)$ good with respect to $A_{\varphi'}$ and $B_{\varphi'}$. If $v \neq u$ then $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(v)] = G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(v)] - C$. Further, C is a connected component of $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(v)]$ since $N_G(C) \subseteq \varphi^{-1}(u)$. Thus, since $\varphi^{-1}(v)$ is good with respect to A_φ and B_φ, $\varphi'^{-1}(v) = \varphi^{-1}(v)$ is also good with respect to $A_{\varphi'}$ and $B_{\varphi'}$.

  We now consider the case when $v = u$. We have that $\varphi'^{-1}(u) = \varphi^{-1}(u) \cup C$. We have already shown that $G[\varphi'^{-1}(u)]$ is connected. First, suppose for contradiction that there exist two components C_1 and C_2 of $G[A_{\varphi'} \cup B_{\varphi'}]$ such that $N^2_G[C_1] \cap \varphi'^{-1}(u)$ is not anti-complete with $N^2_G[C_2] \cap \varphi'^{-1}(u)$. Then there exists a path P on at most six vertices from C_1 to C_2 through $\varphi'^{-1}(u)$. If P does not contain any internal vertices in C then all internal vertices in P are in $\varphi^{-1}(u)$. This contradicts that $N^2_G[C_1] \cap \varphi^{-1}(u)$ is anti-complete with $N^2_G[C_2] \cap \varphi^{-1}(u)$. So P contains an internal vertex in C. But then P contains a sub-path on at most 5 vertices from C_1 to C through $\varphi^{-1}(u)$, contradicting that $N^2_G[C_1] \cap \varphi^{-1}(u)$ is anti-complete with $N^2_G[C] \cap \varphi^{-1}(u)$. Hence every pair C_1 and C_2 of distinct components of $G[A_{\varphi'} \cup B_{\varphi'}]$ satisfy that $N^2_G[C_1] \cap \varphi'^{-1}(u)$ is anti-complete with $N^2_G[C_2] \cap \varphi'^{-1}(u)$.

  Suppose now for contradiction that $\varphi'^{-1}(u)$ contains two distinct non-leaf sub-components Z_1 and Z_2 with respect to $A_{\varphi'}$ and $B_{\varphi'}$. Since $A_{\varphi'} \cup B_{\varphi'} \subseteq A_\varphi \cup B_\varphi$ it follows that $N^2_G[A_{\varphi'} \cup B_{\varphi'}] \subseteq N^2_G[A_\varphi \cup B_\varphi]$. Thus $\varphi^{-1}(u) - N^2_G[A_\varphi \cup B_\varphi] \subseteq \varphi'^{-1}(u) - N^2_G[A_{\varphi'} \cup B_{\varphi'}]$. Hence every sub-component of $\varphi^{-1}(u)$ with respect to A_φ and B_φ is contained in a sub-component of $\varphi'^{-1}(u)$ with respect to $A_{\varphi'}$ and $B_{\varphi'}$. Since W is good, Lemma 7.5.9 yields that $\varphi^{-1}(u)$ has a kernel, namely a unique

non-leaf sub-component Z with respect to A_φ and B_φ. Since Z is fully contained in a sub-component of $\varphi'^{-1}(u)$ with respect to $A_{\varphi'}$ and $B_{\varphi'}$, at least one of Z_1 and Z_2 is disjoint from Z. Without loss of generality, $Z_1 \cap Z = \emptyset$.

Since Z_1 is a non-leaf sub-component of $\varphi'^{-1}(u)$ with respect to $A_{\varphi'}$ and $B_{\varphi'}$, there exist two components C_1 and C_2 of $G[A_{\varphi'} \cup B_{\varphi'}]$ such that there is a path P from C_1 to C_2 through $(N_G^2[C_1] \cap \varphi'^{-1}(u)) \cup Z_1 \cup (N_G^2[C_2] \cap \varphi'^{-1}(u))$. Notably, P is disjoint from the kernel Z of $\varphi^{-1}(u)$ with respect to A_φ and B_φ. If P does not contain any vertices of C then P is a path from C_1 to C_2 through $\varphi^{-1}(u)$, contradicting Lemma 7.5.12 which states that every path from C_1 to C_2 through $\varphi^{-1}(u)$ must intersect the kernel Z of $\varphi^{-1}(u)$. If P does contain a vertex of C then P contains a sub-path P' from C_1 to C through $\varphi^{-1}(u)$, again contradicting Lemma 7.5.12 which states that every path from C_1 to C through $\varphi^{-1}(u)$ must intersect the kernel Z of $\varphi^{-1}(u)$. Thus W' is a good connected generalized ω-creature

- We now bound the adhesion size of (every peripheral vertex of) W'. For every peripheral vertex v of H with $v \neq u$ we have $N(\varphi'^{-1}(v)) = N(\varphi^{-1}(v))$. For u we have $N(\varphi'^{-1}(u)) = N(\varphi^{-1}(v)) - C$. Since every connected component of $G[A_{\varphi'} \cup B_{\varphi'}]$ is a component of $G[A_\varphi \cup B_\varphi]$ it follows that the adhesion size of every peripheral vertex v of H in W' is at most its adhesion size in W. Hence, if W has adhesion size α then W' has adhesion size α.

The proof for the case when C is a component of $G[B_\varphi]$ is symmetric. ■

7.5.5 Extracting a Generalized ω-Creature with Bounded Adhesion Size.

Let G be a k-creature free graph, and $W = (G, H, \varphi, S_1, S_2)$ be a connected good full generalized ω-creature. Our goal this subsection is to show that we can extract from W a connected good full $(\omega/2)$-creature with adhesion size $2k$. This result is encapsulated in Lemma 7.5.20, which is the only lemma that will be used outside of this section.

Lemma 7.5.19. *Let $W = (G, H, \varphi, S_1, S_2)$ be a good generalized ω-creature, let X be a component of $G - (A_\varphi \cup B_\varphi)$, and let C be a component of $G - X$. Then there do not exist k distinct components D_1, D_2, ..., D_k of $G[A_\varphi \cup B_\varphi]$ such that $D_1 \cup D_2 \cup \ldots \cup D_k \subseteq C$ and $N(X) \cap D_i \neq \emptyset$ for every i.*

Proof: Suppose for contradiction that D_1, D_2, ..., D_k exist. Let $\hat{A}$ be the kernel of X.

For every $i \leqslant k$ let Y_i be a connected component of $G[X \cap N_G^2[A_\varphi \cup B_\varphi]]$ that has a neighbor in D_i. By Lemma 7.5.11 the components $Y_1, \ldots, Y_k$ are distinct. By Lemma 7.5.13 each Y_i has a neighbor a_i in $\hat{A}$. The vertices $a_1, \ldots, a_k$ need not be distinct.

For every i, a_i has a neighbor x_i in Y_i. We have that x_i is either in the first or second neighborhood of $A_\varphi \cup B_\varphi$. However, x_i can't be in the first neighborhood of $A_\varphi \cup B_\varphi$ since then a_i would be in the second, and it is not (since $a_i \in \hat{A}$). Thus x_i is in the second neighborhood of $A_\varphi \cup B_\varphi$. Then x_i has a neighbor y_i in $N_G(A_\varphi \cup B_\varphi)$, and therefore $y_i \in Y_i$. Since $y_i \in N_G(A_\varphi \cup B_\varphi)$ it follows that y_i has a neighbor $d_i \in A_\varphi \cup B_\varphi$. By Lemma 7.5.11, $d_i \in D_i$.

We show that $(\hat{A}, \{x_1, \ldots, x_k\}, \{y_1, \ldots, y_k\}, C)$ is a k-creature. The sets $G[\hat{A}]$ and $G[C]$ are connected (by definition of $\hat{A}$ and C) and anti-complete. Further $\hat{A}$ and $\{y_1, \ldots, y_k\}$ are anti-complete because $\{y_1, \ldots, y_k\} \subseteq N_G[A_\varphi \cup B_\varphi]$ while $\hat{A} \cap N_G[A_\varphi \cup$

$B_\varphi] = \emptyset$. Similarly C and $\{x_1, \ldots, x_k\}$ are anti-complete because $\{x_1, \ldots, x_k\} \subseteq N_G^2(A_\varphi \cup B_\varphi)$ while $C \subseteq (A_\varphi \cup B_\varphi)$. Every x_1 has a neighbor a_i in $\hat{A}$, and every y_i has a neighbor $d_i \in C$. Finally $x_i y_i$ is an edge and $x_i y_j$ is not an edge for $i \neq j$ because $x_i \in Y_i$ while $y_j \in Y_j$ there are no edges from Y_i to Y_j since they are distinct components of $N_G^2[A_\varphi \cup B_\varphi]]$. Thus $(\hat{A}, \{x_1, \ldots, x_k\}, \{y_1, \ldots, y_k\}, C)$ is a k-creature, contradicting that G is k-creature free. The statement of the lemma follows. ■

Lemma 7.5.20. *Let $W = (G, H, \varphi, S_1, S_2)$ be a connected, good, and full generalized ω-creature. Then there exists an induced subgraph G' of G and connected, good and full generalized $(\omega/2)$-creature $W' = (G', H', \varphi', S_1', S_2')$ with max adhesion size $2k$.*

Proof: Without loss of generality, W has no absorbable components. If W has absorbable components, let $\hat{W}$ be the result of absorbing all absorbable components in W. By Lemma 7.5.18, $\hat{W}$ is a connected, good and full generalized ω-creature, and $\hat{W}$ has no absorbable components. Then $G, \hat{W}$ also satisfy the premise of the lemma. We may therefore assume that W has no absorbable components.

Claim 7.5.21. *For every peripheral vertex u of H and every component C of $G - \varphi^{-1}(u)$, there exists a peripheral vertex v in H such that $\varphi^{-1}(v) \subseteq C$.*

Proof: Suppose for contradiction that there exists a peripheral vertex u of H and a component C of $G - \varphi^{-1}(u)$ such that there does not exist a peripheral vertex v satisfying $\varphi^{-1}(v) \subseteq C$.

For every peripheral vertex v of H we have that $\varphi^{-1}(v) \cap \varphi^{-1}(u) = \emptyset$ and that $G[\varphi^{-1}(v)]$ is connected (since W is connected). Thus $\varphi^{-1}(v) \subseteq C$ or $\varphi^{-1}(v) \cap C = \emptyset$. By our assumption $\varphi^{-1}(v) \cap C = \emptyset$ for every peripheral vertex v other than u. But then C is a component of $G[A \cup B]$ and $N_G(C) \subseteq \varphi^{-1}(u)$, so C is absorbable, contradicting the assumption that W has no absorbable components. ■

Let S be the set of peripheral vertices u of H such that $G - \varphi^{-1}(u)$ has at most two connected components.

Claim 7.5.22. $|S| \geqslant \omega/2$.

Proof: Let $\hat{G}$ be the (bipartite) graph that has, on one side, a vertex x_u for every peripheral vertex u of H, and on the other side a vertex v_C for every connected component C of $G[A \cup B]$. There is an edge from x_u to x_C in $\hat{G}$ if and only if there is an edge from $\varphi^{-1}(u)$ to C in G. Let $\hat{P} = \{x_u \ : \ u \in V(G) - \{c_A, c_B\}\}$. That is, $\hat{P}$ is the set of vertices of $\hat{G}$ corresponding to peripheral vertices of H.

Note that $\hat{G}$ is obtained from G by contracting every edge $uv \in E(G)$ such that $\varphi^{-1}(u) = \varphi^{-1}(v)$. Thus $\hat{G}$ is connected. Let $\hat{T}$ be an inclusion minimal connected subgraph of G such that $\hat{P} \subseteq V(\hat{T})$. We have that $\hat{T}$ is a tree (since removing an edge of a cycle preserves connectivity) and that every leaf of $\hat{T}$ is in $\hat{P}$ (since removing a leaf from a tree preserves connectivity).

We claim that for every peripheral vertex u if H which is *not* in S, the degree of x_u in $\hat{T}$ is at least 3. Suppose not, and let C_1, C_2, C_3 be distinct components of $G - \varphi^{-1}(u)$ (the components C_1, C_2, C_3 are well defined because $u \notin S$). Let v_1, v_2, and v_3 be peripheral vertices of H such that $\varphi^{-1}(v_1) \subseteq C_1$, $\varphi^{-1}(v_2) \subseteq C_2$, and $\varphi^{-1}(v_3) \subseteq C_3$. The vertices v_1, v_2, v_3 exist by Claim 7.5.21.

Since the degree of x_u in $\hat{T}$ is at most two, $\hat{T} - x_u$ has at most two connected components. By the pigeon hole principle two of the vertices $\{x_{v_1}, x_{v_2}, x_{v_3}\}$ appear in the same component of $\hat{T} - x_u$. Without loss of generality, this is x_{v_1} and x_{v_2}. But then there is a path from x_{v_1} to x_{v_2} in $\hat{G}$ avoiding x_u, and therefore a path from $\varphi^{-1}(v_1)$ to $\varphi^{-1}(v_2)$ in G avoiding $\varphi^{-1}(u)$. But this contradicts that $\varphi^{-1}(v_1)$ and $\varphi^{-1}(v_2)$ are subsets of different components of $G - \varphi^{-1}(u)$. We conclude that for every peripheral vertex u if H which is not in S the degree of x_u in $\hat{T}$ is at least 3.

Let $\hat{P}_3$ be the set of vertices in $\hat{P}$ that have degree at least 3 in $\hat{T}$. A well known fact about trees is that every tree has more leaves than vertices of degree at least 3 (See e.g. [124], Chapter 1, Exercise 17). Therefore, $\hat{T}$ has at least $|\hat{P}_3|$ leaves. For every leaf x_ℓ of $\hat{T}$, we have that ℓ is a peripheral vertex of H and that $\ell \in S$ because the degree of x_ℓ in $\hat{T}$ is 1. It follows that $|S| \geqslant |\hat{P}_3|$, while $|\hat{P}_3|$ is at least the number of peripheral vertices of H which are not in S. It follows that $|S| \geqslant \omega/2$. ■

We can now finish the proof of the Lemma. By Lemma 7.5.19, the adhesion size (in W) of every $u \in S$ is at most $2k$. Let W' be the generalized $|S|$-creature obtained from W by dissolving every peripheral vertex not in S. By Lemma 7.5.16 W' is in fact a connected, good and full generalized $|S|$-creature, by Claim 7.5.22 we have $|S| \geqslant \omega/2$, and finally (again by Lemma 7.5.16) the adhesion size (in W') of every $u \in S$ is at most $2k$. The statement of the lemma follows. ■

7.5.6 Extracting a Disjoint Generalized ω-Creature.

Lemma 7.5.23. *Let $W = (G, H, \varphi, S_1, S_2)$ be a full generalized ω-creature, u be a peripheral vertex of H, and C be a component of $G[\varphi^{-1}(u)] - (S_1 \cup S_2)$ such that $N(C) \cap (A_\varphi \cup B_\varphi) \neq \emptyset$ and $N(C) \cap \varphi^{-1}(u) \subseteq S_1 \cap S_2$. Define $\varphi'(v) = \varphi(v)$ for every $v \in V(G) - C$. If $N(C) \cap A_\varphi \neq \emptyset$ let $\varphi'(v) = c_A$ for every $v \in C$. If $N(C) \cap B_\varphi \neq \emptyset$ set $\varphi'(v) = c_B$ for every $v \in C$. Then $W' = (G, H, \varphi', S_1, S_2)$ is a full generalized ω-creature. Further the max adhesion size of W' is at most the max adhesion size of W.*

Before proving Lemma 7.5.23, observe that by Observation 7.5.5 precisely one of the cases $N(C) \cap A_\varphi \neq \emptyset$ and $N(C) \cap B_\varphi \neq \emptyset$ in the statement of Lemma 7.5.23 will apply.

Proof: [Proof of Lemma 7.5.23] We prove the statement of the lemma for the case that $N(C) \cap A_\varphi \neq \emptyset$. Then, by Observation 7.5.5 we have $N(C) \cap A_\varphi = \emptyset$. First we show that (H, φ') is an ω-bistar partition of G. Let xy be an edge of G. If neither x nor y are

in C then $\varphi'(x) = \varphi(y)$ and $\varphi'(x) = \varphi(y)$, and therefore $\varphi'(x) = \varphi'(y)$ or $\varphi'(x)\varphi'(y)$ is an edge of H. If both x and y are in C then $\varphi'(x) = \varphi'(y) = c_A$. Thus, suppose that $x \in C$ and $y \notin C$. We have that $N(C) \subseteq \varphi^{-1}(u) \cup A_\varphi$ and therefore, $y \in A_\varphi$ or $y \in \varphi^{-1}(u)$. In the former case we have $\varphi'(x) = \varphi'(y) = c_A$, while in the latter case we have $\varphi'(x) = c_A$ while $\varphi'(y) = u$, and $c_A u$ is an edge of H. Next we check that W' satisfies the properties of full generalized ω-creatures.

- Property *(i)* clearly holds for all peripheral vertices of H other than u (with $S_1^\star = S_1$ and $S_2^\star = S_2$). For u observe that $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(u)] = [A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)]$ and that $\varphi'^{-1}(u) \cap S_1 = \varphi^{-1}(u) \cap S_1$ and $\varphi'^{-1}(u) \cap S_2 = \varphi^{-1}(u) \cap S_2$. Thus $\varphi'^{-1}(u) \cap S_1$ and $\varphi'^{-1}(u) \cap S_2$ are distinct minimal $A_{\varphi'}$,$B_{\varphi'}$ separators in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(u)]$.

- For property *(ii)* observe that G, S_1, S_2 are the same for W and W'. Further, $B_{\varphi'} = B_\varphi$ and $A_{\varphi'} = A_\varphi \cup C$. Thus $B_{\varphi'}$ is entirely contained in a component of $G - S_1$, and A_φ is entirely contained in a different component of $G - S_1$. Since C is disjoint from S_1 and has a neighbor in A_φ, all of $A_{\varphi'}$ is contained in the same component of $G - S_1$ as A_φ. An identical argument shows that $B_{\varphi'}$ is entirely contained in a component of $G - S_2$, and that $A_{\varphi'}$ is entirely contained in a different component of $G - S_2$.

- For property *(iii)*, let C_1 and C_2 be components of $G[A_{\varphi'} \cup B_{\varphi'}]$, and v be a peripheral vertex of H such that there is a path P from C_1 to C_2 through $\varphi'^{-1}(v) - S_1$. Observe now that C does not have any neighbors in $\varphi'^{-1}(v) - S_1$. Indeed, if $v \neq u$ then $\varphi'^{-1}(v) = \varphi^{-1}(v)$ is anticomplete with $\varphi^{-1}(u)$, and $C \subseteq \varphi^{-1}(u)$. If $v = u$ then $N(C) \cap \varphi'^{-1}(u) \subseteq S_1 \cap S_2$ by assumption. Thus C does not have any neighbors in $\varphi'^{-1}(v) - S_1$. Thus the path P starts and ends in $G[A_\varphi \cup B_\varphi]$. Let C_1' and C_2' be the components of $G[A_\varphi \cup B_\varphi]$ containing the first and last vertex of P, respectively. Note that $C_1' \subseteq C_1$ and that $C_2' \subseteq C_2$.

Since $\varphi'^{-1}(v) \subseteq \varphi^{-1}(v)$ we have that P is a path from C'_1 to C'_2 through $\varphi^{-1}(v) - S_1$. Therefore (by property *(iii)* applied to W) there is a path P' from C'_1 to C'_2 through $\varphi^{-1}(v) - S_2$. Since C is a component of $G[\varphi^{-1}(u)] - S_2$. we have that the internal vertices of P' are either entirely inside C or disjoint from C. However all internal vertices of P' cannot be inside C, because then the endpoints of P' are in the same component of $G[A_{\varphi'}]$ contradicting that C_1 and C_2 are distinct components of $G[A_{\varphi'}]$. Hence P' is a path from C'_1 to C'_2 through $\varphi'^{-1}(v) - S_2$.

The proof for the reverse direction of the equivalence, namely that if there is a path P from C_1 to C_2 through $\varphi'^{-1}(v) - S_2$, then there also is a path from C_1 to C_2 through $\varphi'^{-1}(v) - S_1$, is identical.

- For property *(iv)*, let X_A be a component of $G[A_{\varphi'}]$ and v be a peripheral vertex of H such that X_A has a neighbor in $\varphi'^{-1}(v)$. Let xy be an edge of G with $x \in X_A$ and $y \in \varphi'^{-1}(v)$. There are two cases, either $x \in C$ or $x \in A_\varphi$. If $x \in C$ then $v = u$ and $y \in S_1$. By property *(i)* applied to W and $S_1^\star = S_1$, there is a path P from y to B_φ through $\varphi^{-1}(u) \cap B_1(u)$. But $C \subseteq A_1(u)$ and $\varphi'^{-1}(u) = \varphi^{-1}(u) - C$, and therefore P is a path from y to $B_\varphi = B_{\varphi'}$ through $\varphi'^{-1}(u)$.

  If $x \in A_\varphi$, let X'_A be the component of $G[A_\varphi]$ that contains x. We have that $X'_A \subseteq X_A$. Furthermore, $y \in \varphi^{-1}(v)$ because $\varphi'^{-1}(v) \subseteq \varphi^{-1}(v)$. Thus X'_A has a neighbor in $\varphi^{-1}(v)$, and hence, by property *(iv)* applied to W, there is a path P from X'_A to B_φ through $\varphi^{-1}(v)$.

  If P has no internal vertices in C then P is a path from X'_A to $B_{\varphi'} = B_\varphi$ through $\varphi'^{-1}(v)$. If P has internal vertices in C, then $v = u$. Since $x \in X_A$, $y \in C$ and xy is an edge, we have that $C \subseteq X_A$. Let x' be the last vertex of X_A on P. The sub-path of P from x' to $B_{\varphi'} = B_\varphi$ is a path from X_A to $B_{\varphi'}$ through $\varphi'^{-1}(v)$.

  Consider now a component X_B of $G[B_{\varphi'}]$ and let v be a peripheral vertex of H

such that X_B has a neighbor in $\varphi'^{-1}(v)$. X_B is also a component of $G[B_\varphi]$ and $\varphi'^{-1}(v) \subseteq \varphi^{-1}(v)$, so by property *(iv)* applied to W there exists a path P from X_B to A_φ through $\varphi^{-1}(v)$. Let x' be the first vertex on P from $A_{\varphi'}$. The sub-path of P that ends in x' is a path from X_B to $A_{\varphi'}$ through $\varphi'^{-1}(v)$.

Finally we upper bound the max adhesion size of W. Note that every component of $G[A'_\varphi \cup B'_\varphi]$ is a component of $G[A_\varphi \cup B_\varphi]$, with the exception of the unique component X of $G[A'_\varphi \cup B'_\varphi]$ which contains C. The component X is equal to C plus the union of all components of $G[A_\varphi \cup B_\varphi]$ which contain a neighbor of C. By assumption there exists at least one component of $G[A_\varphi \cup B_\varphi]$ which contain a neighbor of C.

Since C does not have any neighbors in $\varphi^{-1}(v)$ for any peripheral vertex $v \neq u$, the adhesion size of every peripheral vertex $v \neq u$ in W' is at most its adhesion size in W. For u, every component of $G[A'_\varphi \cup B'_\varphi]$ that contains a neighbor of $\varphi'^{-1}(u)$ contains a component of $G[A_\varphi \cup B_\varphi]$ that contains a neighbor of $\varphi^{-1}(u)$. Thus the adhesion size of u in W' is at most its adhesion size in W. This completes the proof for the case that $N(C) \cap A \neq \emptyset$. The proof for the case where $N(C) \cap B \neq \emptyset$ is symmetric. ■

Lemma 7.5.24 (Extract Disjoint ω-Creature)**.** *Let $W = (G, H, \varphi, S_1, S_2)$ be a full generalized ω-creature of max adhesion size α. Then there exists a full disjoint generalized ω-creature, $W' = (G', H, \varphi', S'_1, S'_2)$, of max adhesion size α such that G' is an induced subgraph of G.*

Proof: Without loss of generality W satisfies the following additional property: there does not exist a peripheral vertex u, and component C of $G[\varphi^{-1}(u)] - (S_1 \cup S_2)$ such that $N(C) \cap (A_\varphi \cup B_\varphi) \neq \emptyset$ and $N(C) \cap \varphi^{-1}(u) \subseteq S_1 \cap S_2$. Indeed, if such a u and C exists then Lemma 7.5.23 yields a full generalized ω-creature with the same graph G, adhesion size at most α, and strictly larger $|A_\varphi \cup B_\varphi|$. Since $|A_\varphi \cup B_\varphi| \leqslant |V(G)|$ there must exist a full generalized ω-creature $W^\star = (G, H, \varphi^\star, S_1, S_2)$ on the same graph G

that additionally satisfies the additional property. Since $W^\star$ satisfies the premise of the lemma we may assume that $W = W^\star$.

We set $G' = G - (S_1 \cap S_2)$, $\varphi'(v) = \varphi(v)$ for every $v \in V(G')$, $S'_1 = S_1 - S_2$ and $S'_2 = S_2 - S_1$. We claim that $W' = (G', H, \varphi', S'_1, S'_2)$ satisfies the conclusion of the lemma. We first prove that it is a full sugeneralized ω-creature. Indeed, G' is an induced subgraph of G, and (H, φ') is an ω-bistar partition of G. Note that $A_{\varphi'} = A_\varphi$ and $B_{\varphi'} = B_\varphi$.

- For property *(i)* we have that for every peripheral vertex, u, of H, $G'[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(u)] - (\varphi'^{-1}(u) \cap S'_1)$ is equal to $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(u)] - (\varphi^{-1}(u) \cap S_1)$, and so S'_1 is a minimal separator in $G'[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(u)]$. Similarly, $\varphi'^{-1}(u) \cap S'_2$ a minimal separator in $G'[A_{\varphi'} \cup B_{\varphi'} \cup \varphi'^{-1}(u)]$. Finally, $\varphi'^{-1}(u) \cap S'_1$ and $\varphi'^{-1}(u) \cap S'_2$ are distinct because $\varphi^{-1}(u) \cap S_1$ and $\varphi'^{-1}(u) \cap S_2$ are distinct.

- Property *(ii)* is satisfied by W' because $G' - S'_1 = G - S_1$, and $G' - S'_2 = G - S_2$.

- Property *(iii)* is satisfied by W' because every pair of components C_1 and C_2 of $G'[A_{\varphi'} \cup B_{\varphi'}]$ are components of $G[A_\varphi \cup B_\varphi]$, and for every peripheral vertex u of H we have $\varphi'^{-1}(u) - S'_1 = \varphi^{-1}(u) - S_1$ and $\varphi'^{-1}(u) - S'_2 = \varphi^{-1}(u) - S_2$.

- For property *(iv)*, let u be a peripheral vertex and X_A be a component of $G'[A_{\varphi'}]$ that has a neighbor in $\varphi'^{-1}(u)$.

  Let xy be an edge of G' with $x \in X_A$ and $y \in \varphi'^{-1}(u)$. We claim that there exists a path in G' from X_A to $S_1 \cup S_2$ through $\varphi'^{-1}(u)$. If $y \in S_1 \cup S_2$ then xy is the desired path, so suppose $y \in \varphi'^{-1}(u) - S_1 \cup S_2$. Let C be the connected component of $\varphi^{-1}(u) - (S_1 \cup S_2)$ that contains u. The property of W discussed at the start of the proof ensures that C has at least one neighbor z in $\varphi'^{-1}(u) \cap (S_1 \cup S_2) - (S_1 \cap S_2)$. Let P now be a path from x to z through C. P does not contain any vertices of $S_1 \cap S_2$ and hence it is a path in G' from X_A to $S_1 \cup S_2$ through $\varphi'^{-1}(u)$.

Let z be the endpoint of the path P in $S_1 \cup S_2$. If z is in S_1 then, by minimality of $S_1 \cap \varphi^{-1}(u)$ (here we use that W is full), there exists a path P' in G from z to B_φ through $\varphi^{-1}(u) - S_1$. But P' does not contain any vertices of $S_1 \cap S_2$ and hence P' is a path from z to $B_{\varphi'}$ through $\varphi'^{-1}(u)$. An identical argument shows that if z is in S_2 then there exists a path P' from z to $B_{\varphi'}$ through $\varphi'^{-1}(u)$. But then P, followed by P' is a path in G' from X_A to $B_{\varphi'}$ through $\varphi'^{-1}(u)$.

The proof of the analogous statement for peripheral vertex u and component X_B of $G'[B_{\varphi'}]$ that has a neighbor in $\varphi'^{-1}(u)$ is identical.

Having shown that $W' = (G', H, \varphi', S'_1, S'_2)$ is a full generalized ω-creature, note that S'_1 and S'_2 are disjoint. Further the adhesion size of every peripheral vertex v of H in W' is at most its adhesion size in W, since $A_{\varphi'} = A_\varphi$, $B_{\varphi'} = V_\varphi$, and $\varphi'^{-1}(v) \subseteq \varphi^{-1}(v)$. This concludes the proof of the lemma. ■

7.5.7 Connectivity Graphs and Long Induced Paths in them

Definition 7.5.25 (Realize)**.** Let W be a generalized ω-creature $W = (G, H, \varphi, S_1, S_2)$. A peripheral vertex u of H *realizes* an (unordered) pair of distinct components $\{C_1, C_2\}$ of $G[A_\varphi \cup B_\varphi]$ if there is a path in G from C_1 to C_2 through $\varphi^{-1}(u) - S_1$.

Note that by property *(i)* of generalized ω-creatures, if u realizes $\{C_1, C_2\}$, then C_1 and C_2 are either both components of $G[A_\varphi]$ or both components of $\mathrm{G}[B_\varphi]$.

Definition 7.5.26 (Connectivity Graph)**.** The *A-connectivity graph* of a generalized ω-creature $W = (G, H, \varphi, S_1, S_2)$ is a graph $\mathcal{C}_A$. The vertices of $\mathcal{C}_A$ are the connected components of $G[A_\varphi]$. Two components C_1 and C_2 of $G[A_\varphi]$ are connected by an edge in $\mathcal{C}_A$ if there exists a peripheral vertex u of H that realizes $\{C_1, C_2\}$.

The *B-connectivity graph* $\mathcal{C}_B$ of W is defined similarly, with vertices of $\mathcal{C}_B$ being components of $G[B_\varphi]$, and two components C_1 and C_2 are connected by an edge in $\mathcal{C}_B$ if there exists a peripheral vertex u of H that realizes $\{C_1, C_2\}$.

The A-connectivity graphs tracks which pairs $\{C_1, C_2\}$ of components of $G[A_\varphi]$ are realized by some peripheral vertex of H. We will (towards the end of the proof) also be interested in precisely which peripheral vertices realize a given pair. We encapsulate this in the notion of labeled connectivity graphs.

Definition 7.5.27 (Labeled Connectivity Graph)**.** The *labeled A-connectivity graph* of a generalized ω-creature $W = (G, H, \varphi, S_1, S_2)$ is a pair $(\mathcal{C}_A, \lambda)$ where $\mathcal{C}_A$ is the A-connectivity graph of W, and $\lambda : E(\mathcal{C}_A) \to 2^{V(H)}$ takes as input an edge $\{C_1, C_2\}$ and outputs the subset of peripheral vertices u of H that realize $\{C_1, C_2\}$.

The labeled B-connectivity graph W is a pair $(\mathcal{C}_B, \lambda)$ where $\mathcal{C}_B$ is the B-connectivity graph of W, and $\lambda : E(\mathcal{C}_B) \to 2^{V(H)}$ takes as input an edge $\{C_1, C_2\}$ and outputs the subset of peripheral vertices u of H that realize $\{C_1, C_2\}$.

We will occasionally be interested in the subgraph of the A-connectivity graph $\mathcal{C}_A$ of a generalized ω-creature W induced by a vertex set Z. Just as for normal graphs, we will denote the induced subgraph by $\mathcal{C}_A[Z]$. We can treat such an induced subgraph as a labeled induced subgraph by dropping from the domain of the labeling λ all edges that do not appear in the considered induced subgraph. We will denote this labeled induced subgraph by $(\mathcal{C}_A, \lambda)[Z]$.

Lemma 7.5.28. *Let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature. Then the A-connectivity graph $\mathcal{C}_A$ and the B-connectivity graph $\mathcal{C}_B$ of W are connected.*

Proof: This follows immediately from Property *(ii)* of generalized ω-creatures. ■

Lemma 7.5.29. *Let G be k-creature free, $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature and C be a component of $G[A_\varphi \cup B_\varphi]$. Let Z be the set of peripheral vertices u of H such that $\varphi^{-1}(u)$ contains a neighbor of C in G. Then $|Z| < k$.*

Proof: Suppose for contradiction that $|Z| \geqslant k$, and let $z_1, \ldots, z_k$ be k distinct vertices in Z. Let $S_1^\star \subseteq S_1$ and $S_2^\star \subseteq S_2$ be witness separators for W. For each $i \leqslant k$ proceed as follows. Since C has a neighbor in $\varphi^{-1}(z_i)$, by property (iv) of generalized ω-creatures, there is a path from C to B_φ through $\varphi^{-1}(z_i)$. By property (i) this path contains at least one vertex of $S_1^\star$ (and at least one vertex of $S_2^\star$). Thus, there exists a path from C to $S_1^\star \cup S_2^\star$ in $G[C \cup \varphi^{-1}(z_i)]$. Let P_i^1 be a shortest such path.

Let x_i be the last vertex of P_i^1, without loss of generality x_i is in $S_1^\star$. If it is in $S_2^\star$ we instead consider the generalized ω-creature resulting from flipping W (and $S_1^\star$ and $S_2^\star$) at z_i (see Lemma 7.5.8). Since this changes $S_1 \cap \varphi^{-1}(z_i)$ to $S_2 \cap \varphi^{-1}(z_i)$, and $S_1^\star \cap \varphi^{-1}(z_i)$ to $S_2^\star \cap \varphi^{-1}(z_i)$, and vice versa (and changes nothing else), we may now assume that x_i is in $S_1^\star$.

By property *(i)* of generalized ω-creatures we have that $\varphi^{-1}(z_i) \cap S_1^\star$ is an A_φ, B_φ-minimal separator in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(z_i)]$. Thus there exists an induced path P_i^2 from x_i to B_φ in $G[B_\varphi \cup \varphi^{-1}(z_i)]$ that does not contain any vertices of $S_1^\star$. We select P_i^2 such that only the last vertex of P_i^2 is in B_φ.

Notice now that P_1^i followed by P_2^i is a path from A_φ to B_φ in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(z_i)]$. Let y_i be the vertex immediately after x_i on the path P_2^i. The vertex y_i can not be the last vertex of P_2^i, since then $P_1^i P_2^i$ is a path from A_φ to B_φ in $G[A_\varphi \cup B_\varphi \cup \varphi^{-1}(z_i)]$ disjoint from $S_2^\star$.

We define

$$A = C \cup \bigcup_{i \leqslant k} P_1^i - \{x_i\}$$

$$B = B_2^\star(W) \cup \bigcup_{i \leqslant k} P_2^i - \{x_i, y_i\}$$

and claim that $(A, B, \{x_1, \ldots, x_k\}, \{y_1, \ldots, y_k\})$ forms a k-creature. We check the properties of k-creatures one by one:

- A is disjoint from $\{x_1, \ldots, x_k\} \cup \{y_1, \ldots, y_k\} \cup B$ because $A \subseteq A_1^\star(W)$ while $\{x_1, \ldots, x_k\} \cup \{y_1, \ldots, y_k\} \cup B \subseteq S_1^\star \cup B_1^\star(W)$.

- $\{x_1, \ldots, x_k\}$ is disjoint from $\{y_1, \ldots, y_k\}$ because y_i is the successor of x_i on P_2^i. For $j \neq i$ we have that $x_i \in \varphi^{-1}(z_i)$ while $y_j \in \varphi^{-1}(z_j)$, which are disjoint and anticomplete. This shows not only that $\{x_1, \ldots, x_k\}$ is disjoint from $\{y_1, \ldots, y_k\}$, but also that $x_i y_j$ is an edge if and only if $i = j$.

- $\{y_1, \ldots, y_k\}$ is disjoint from B because $\{y_1, \ldots, y_k\} \subseteq A_2^\star(W) \cup S_2$, while $B = B_2^\star(W) \cup \bigcup_{i \leqslant k} P_2^i - \{x_i, y_i\}$.

- $G[A]$ is connected because C is connected and each P_1^i is a path that starts from C.

- $G[B]$ is connected because $B_2^\star(W)$ is connected and each $P_2^i - \{x_i, y_i\}$ is a path that ends in $B_\varphi \subseteq B_2^\star(W)$.

- A and $B \cup \{y_1, \ldots, y_k\}$ are anticomplete: indeed $A \subseteq A_1^\star(W)$ while $B \cup \{y_1, \ldots, y_k\} \subseteq B_1^\star(W)$, and $A_1^\star(W)$ and $B_1^\star(W)$ are anticomplete.

- B and $A \cup \{x_1, \ldots, x_k\}$ are anticomplete: since we have already shown that A and B are anticomplete it suffices to show that B and $\{x_1, \ldots, x_k\}$ are anticomplete. Suppose for contradiction that $x_j b$ is an edge with $b \in B$. We have that $\{x_1, \ldots, x_k\} \subseteq A_2^\star(W)$, so $\{x_1, \ldots, x_k\}$ is anticomplete with $B_2^\star(W)$. We conclude that $b \in (P_2^i - \{x_i, y_i\})$. Since $b \notin B_2^\star(W)$ we have that b cannot be the last vertex

of $P_2^i - \{x_i, y_i\}$, and therefore $b \in \varphi^{-1}(z_i)$. But $x_j \in \varphi^{-1}(z_j)$ and $\varphi^{-1}(z_i)$ and $\varphi^{-1}(z_j)$ are anticomplete unless $i = j$, contradicting that $x_i b$ is an edge. So $i = j$ and b is a vertex on $P_2^i - \{x_i, y_i\}$. However y_i is the only vertex on P_2^i adjacent to x_i, and $b \neq y_i$ yielding a contradiction. Thus B and $\{x_1, \ldots, x_k\}$ are anticomplete.

- Each x_i has a neighbor in A, namely its predecessor in P_1^i, and each y_i has a neighbor in B, namely its successor in P_1^i.

We conclude that $(A, B, \{x_1, \ldots, x_k\}, \{y_1, \ldots, y_k\})$ forms a k-creature in G, contradicting that G is k-creature free.

■

Lemma 7.5.30. *Let G be a k-creature free graph and $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature with adhesion size at most α. Then the the A-connectivity graph $\mathcal{C}_A$ of W (and the B-connectivity graph $\mathcal{C}_B$ of W) contains an induced path of length at least $\log_{k\alpha}(\omega/k) - 1$.*

Proof: We prove the statement for the A-connectivity graph $\mathcal{C}_A$ of W. First, by Lemma 7.5.28 $\mathcal{C}_A$ is connected. Next we show that $\mathcal{C}_A$ has at least $\omega/k\omega$ vertices. By Lemma 7.5.4 each peripheral vertex v of H satisfies that $\varphi^{-1}(v)$ has a neighbor in A_φ. On the other hand, for every component C of $G[A_\varphi]$, Lemma 7.5.29 yields that there are at most $k-1$ peripheral vertices v such that $\varphi^{-1}(v)$ has a neighbor in C. Thus, $G[A_\varphi]$ has at least ω/k components, and so the A-connectivity graph $\mathcal{C}_A$ of W has at least ω/k vertices.

Next, we show that the maximum degree of $\mathcal{C}_A$ is at most $k\alpha$. Indeed, consider a component C of $G[A_\varphi]$, and another component C' of $G[A_\varphi]$ such that C and C' are adjacent in $\mathcal{C}_A$. Then there exists a peripheral vertex v of H such that $\varphi^{-1}(v)$ has a neighbor both in C and in C'. By Lemma 7.5.29 there are at most $k-1$ vertices u such

that $\varphi^{-1}(u)$ has a neighbor in C. For each such peripheral vertex u, there are at most α components C'' of $G[A_\varphi]$ that have a neighbor in $\varphi^{-1}(u)$. Thus there are at most $k\alpha$ components C' adjacent to C in $\mathcal{C}_A$.

We have that $\mathcal{C}_A$ is a connected graph with at least ω/k vertices and maximum degree at most $k\alpha$. Pick any vertex C of $\mathcal{C}_A$. For every $d \geqslant 1$, the number of vertices at distance exactly d from C in $\mathcal{C}_A$ is at most $(k\alpha)^d$, and therefore the number of vertices at distance at most d is at most $(k\alpha)^{d+1}$. Let C' be the vertex of $\mathcal{C}_A$ furthest away from C in $\mathcal{C}_A$, and let d be the distance from C to C' in $\mathcal{C}_A$. We have that $\omega/k \leqslant (k\alpha)^{d+1}$, and therefore $\log_{k\alpha}(\omega/k) \leqslant d+1$. Thus a shortest path from C to C' in $\mathcal{C}_A$ satisfies the conclusion of the lemma. The proof for the B-connectivity graph $\mathcal{C}_B$ of W is identical. ■

7.5.8 Erasing Components

We will work towards extracting from W a generalized ω'-creature whose A-connectivity graph is a path. Towards this we will identify a long induced path P in the A-connectivity graph of G, and delete all components of $G[A_\varphi]$ that are not on the path P. However, when we delete components of $G[A_\varphi]$ we need to appropriately modify the generalized ω-creature in the peripheral vertices in order to ensure that the result of path-filtering is still a generalized ω-creature.

Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature. A component C of $G[A_\varphi]$ is *erasable* if $\mathcal{C}_A - \{C\}$ is connected. Similarly, a component C of $G[B_\varphi]$ is *erasable* if $\mathcal{C}_B - \{C\}$ is connected. For an induced path P in the A-connectivity graph $\mathcal{C}_A$ of W, or the B-connectivity graph $\mathcal{C}_B$ of W, a component C of $G[A_\varphi \cup B_\varphi]$ is *P-erasable* if C is erasable and $C \notin V(P)$.

Lemma 7.5.31. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature. If P is an induced path in the A-connectivity graph $\mathcal{C}_A$ of W and there exists a component C of $G[A_\varphi]$ not*

in $V(P)$ then there exists a P-erasable component C' of $G[A_\varphi]$. If P is an induced path in the B-connectivity graph $\mathcal{C}_B$ of W and there exists a component C of $G[B_\varphi]$ not in $V(P)$ then there exists a P-erasable component C' of $G[B_\varphi]$.

Proof: We prove the statement for P being a path in $\mathcal{C}_A$. By Lemma 7.5.28 $C_\mathcal{A}$ is connected. Therefore it has a spanning tree T that contains all the edges of P. Since there exists a component C of $G[A_\varphi]$ not in $V(P)$, T has a leaf C' not in $V(P)$. Since $T - \{C'\}$ is connected, $\mathcal{C}_A - \{C'\}$ is connected as well. Hence C' is erasable. The proof for statement for P being a path in $\mathcal{C}_B$ is identical. ■

We are aiming at an operation that will delete all of the vertices of a P-erasable component C from the graph G of a generalized ω-creature. Simply deleting all vertices of C from G does not immediately work, because some peripheral vertices of H might violate property *(iv)* of generalized ω-creatures after such a deletion. The next definitions aim to highlight the peripheral vertices for which such a problem could occur.

Definition 7.5.32 (Chunk)**.** Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature and v be a peripheral vertex of H. A *chunk* of v is a connected component of $G[\varphi^{-1}(v)]$. A chunk of W is a chunk of v for some peripheral vertex v of H.

Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature, and C be a component of $G[A_\varphi \cup B_\varphi]$. A peripheral vertex u of H is *C-dependent* if $C \subseteq A_\varphi$ and C contains $N(\varphi^{-1}(u)) \cap A_\varphi$ or $C \subseteq B_\varphi$ and C contains $N(\varphi^{-1}(u)) \cap B_\varphi$. The set $D(C)$ denotes the set of all C-dependent peripheral vertices in H. Similarly, a chunk Z of W is C-dependent if $C \subseteq A_\varphi$ and C contains $N(\varphi^{-1}(u)) \cap A_\varphi$ or $C \subseteq B_\varphi$ and C contains $N(\varphi^{-1}(u)) \cap B_\varphi$. The set $\mathcal{D}_S(C)$ denotes the set of all C-dependent chunks of W. Note that despite the similar names $D(C)$ and $\mathcal{D}_S(C)$ are objects of different types. More concretely $D(C)$ is a set of vertices of H while $\mathcal{D}_S(C)$ is a set of vertex sets. We are now ready to define the operation of erasing a component.

Definition 7.5.33 (Erasing Component)**.** Let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature, and let C be an erasable component of $G[A_\varphi \cup B_\varphi]$. We define X to be the union of all C-dependent chunks of W, If C is a component of $G[A_\varphi]$ we define Y to be the union of all C-dependent chunks of W that have at least one neighbor in B_φ. If C is a component of $G[B_\varphi]$ we define Y to be the the union of all C-dependent chunks of W that have at least one neighbor in B_φ. We set

- $G' = (G - C) - (X - Y)$, and
- $H' = H - D(C)$.
- For every $v \in Y$ we set $\varphi'(v) = c_B$ if $C \subseteq A_\varphi$ and $\varphi'(v) = c_A$ if $C \subseteq B_\varphi$. For every $v \in V(G') - Y$ we set $\varphi'(v) = \varphi(v)$.
- We set $S_1' = S_1 - X$ and $S_2' = S_2 - X$.

Lemma 7.5.34. *Let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature, and let C be an erasable component of $G[A_\varphi]$ or of $G[B_\varphi]$. Let $W' = (G', H', \varphi', S_1', S_2')$ be obtained from W by erasing C. Then W' is a disjoint generalized ω'-creature, where $\omega' = \omega - |D(C)|$. Further for every peripheral vertex u of H' its adhesion size in W' is at most its adhesion size in W.*

Proof: We prove the statement of the lemma for the case when C is a component of $G[A_\varphi]$. First we show that (H', φ') is a ω'-bistar partition of G'. Note that every peripheral vertex of H, except the vertices in $D(C)$, is a peripheral vertex of H'. Thus H' has ω' peripheral vertices. Consider an arbitrary edge $uv \in E(G')$. If $\varphi(u) \notin Y$ and $\varphi(v) \notin Y$ then $\varphi'(u) = \varphi(u)$, $\varphi'(v) = \varphi(v)$, so $\varphi'(u) = \varphi'(v)$ or $\varphi'(u)\varphi'(v) \in E(H')$. If $\varphi(u) \in Y$ and $\varphi(v) \in Y$ then $\varphi'(u) = \varphi'(v) = c_B$. Suppose now that $\varphi(u) \in Y$ and $\varphi(v) \notin Y$. But then u is in a C-dependent chunk of W, and therefore $v \in B_\varphi$. But then $\varphi'(u) = \varphi'(v) = c_B$. Thus (H', φ') is an ω'-bistar partition of G'.

Next observe that S'_1 and S'_2 are disjoint because $S'_1 \subseteq S_1$, $S'_2 \subseteq S_2$, and S_1 and S_2 are disjoint. Both S'_1 and S'_2 are disjoint from $A_{\varphi'} \cup B_{\varphi'}$ because $A_{\varphi'} \cup B_{\varphi'} \subseteq A_\varphi \cup B_\varphi \cup X$ while $S'_1 = S_1 - X$ and $S'_2 = S_2 - X$. We now verify that $W' = (G', H', \varphi', S'_1, S'_2)$ satisfies the properties of generalized ω'-creatures.

- For property *(i)* we first show that S'_1 and S'_2 separate $A_{\varphi'}$ from $B_{\varphi'}$ in G'. Suppose for contradiction that there exists a path P from $A_{\varphi'}$ to $B_{\varphi'}$ in $G' - S'_1$. Since (H', φ') is an ω'-bistar partition of G' the path P has at least three vertices. Further we may select P such that none of the internal vertices of P are in $A_{\varphi'} \cup B_{\varphi'}$. Thus, since (H', φ') is an ω'-bistar partition of G' there exists a peripheral vertex u of H' such that P is a path in G' from $A_{\varphi'}$ to $B_{\varphi'}$ through $\varphi'^{-1}(u)$. But $\varphi'^{-1}(u) \subseteq \varphi^{-1}(u)$ and G' is an induced subgraph of G, so

$$N_{G'}(\varphi'^{-1}(u)) \subseteq N_G(\varphi^{-1}(u)) \subseteq A_\varphi \cup B_\varphi$$

  It follows that P is a path in G from A_φ to B_φ through $\varphi^{-1}(u)$. Let Q be the chunk of u that contains P. Since P is a path in G' its last endpoint is in $A_\varphi - C$, and therefore Q is not C-dependent. But then $Q \cap S_1 = Q \cap S'_1$ and thus P is a path in G from A_φ to B_φ through $\varphi^{-1}(u) - S_1$, contradicting property *(i)* applied to W. The proof that S'_2 is a $A_{\varphi'}$, $B_{\varphi'}$-separator in G' is symmetric.

  Next we show that for every peripheral vertex $u \in H'$ there exists a path from $A_{\varphi'}$ to $B_{\varphi'}$ through $\varphi'^{-1}(u)$. Since u is a peripheral vertex of H' it is not C-dependent in A, and therefore there exists a component C' of $G[A_\varphi]$ other than C with a neighbor in A_φ. By property *(iv)* applied to W there is a path P from C' to B_φ through $\varphi^{-1}(u)$. The internal vertices of P are contained in a chunk Q of u, and this chunk is not C-dependent because Q has a neighbor in C'. But then $Q \subseteq \varphi'^{-1}(u)$ so P is a path in G' from $A_{\varphi'}$ to $B_{\varphi'}$ through $\varphi'^{-1}(u)$. This implies that every minimal

$A_{\varphi'}$,$B_{\varphi'}$ separator in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi^{-1}(u)]$ is non-empty.

Since S'_1 is an $A_{\varphi'}, B_{\varphi'}$-separator in G it follows that for every peripheral vertex u of H', $S'_1 \cap \varphi^{-1}(u)$ is an $A_{\varphi'}, B_{\varphi'}$-separator in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi^{-1}(u)]$. Thus $S'_1 \cap \varphi^{-1}(u)$ contains a minimal $A_{\varphi'}, B_{\varphi'}$-separator $S'^{\star}_{1,u}$ in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi^{-1}(u)]$. Similarly, $S'_2 \cap \varphi^{-1}(u)$ contains a minimal $A_{\varphi'}, B_{\varphi'}$-separator $S'^{\star}_{2,u}$ in $G[A_{\varphi'} \cup B_{\varphi'} \cup \varphi^{-1}(u)]$. Since S'_1 and S'_2 are disjoint while $S'^{\star}_{1,u} \subseteq S'_1$ and $S'^{\star}_{2,u} \subseteq S'_2$ are nonempty, $S'^{\star}_{1,u}$ and $S'^{\star}_{2,u}$ are distinct. Let $S'^{\star}_1 = \bigcup_u S'^{\star}_{1,u}$ and $S'^{\star}_2 = \bigcup_u S'^{\star}_{2,u}$, where the union is taken over all peripheral vertices u of H'. Then $S'^{\star}_1$ and $S'^{\star}_2$ are witness separators for W'.

- For property *(ii)* we first prove that $A_{\varphi'}$ is contained in a connected component of $G' - S'_1$.

  Observe that the A-connectivity graph $\mathcal{C}_A$ of W satisfies that $\mathcal{C}_A - \{C\}$ is connected, and that $A_{\varphi'} = A_{\varphi} - C$. Thus, for every pair of components C_1, C_2 of $G'[A_{\varphi'}]$ there is a sequence $R_1, R_2, \ldots, R_t$ of components of $G'[A_{\varphi'}]$, such that $C_1 = R_1$, $C_2 = R_t$, and for every $i < t$ there exists a peripheral vertex u of H a path P in G from R_i to R_{i+1} through $\varphi^{-1}(u) - S_1$. The internal vertices of the path P are contained in a chunk $Q \subseteq \varphi^{-1}(u)$ of W. Since Q has a neighbor in R_i it is not C-dependent, and therefore $Q \subseteq \varphi'^{-1}(u)$. But then P is a path from R_i to R_{i+1} through $\varphi'^{-1}(u) - S_1$. Further $S'_1 \subseteq S_1$ and therefore P is a path from R_i to R_{i+1} through $\varphi'^{-1}(u) - S'_1$. So R_i and R_{i+1} are contained in the same component of $G' - S'_1$, and therefore so are C_1 and C_2. But C_1 and C_2 were arbitrarily chosen components in $G'[A_{\varphi'}]$, hence $A_{\varphi'}$ is contained in a connected component of $G' - S'_1$. The proof that $A_{\varphi'}$ is contained in a connected component of $G' - S'_2$ is identical.

  We now prove that $B_{\varphi'}$ is contained in a connected component of $G' - S'_1$. First note that $B_{\varphi} \subseteq B_{\varphi'}$ and therefore every connected component of $G'[B_{\varphi'}]$ contains a connected component of $G[B_{\varphi}]$. Further, by Lemma 7.5.28 for every pair of

components C_1, C_2 of $G[B_\varphi]$ there is a sequence $R_1, R_2, \ldots, R_t$ of components of $G[B_\varphi]$, such that $R_1 = C_1$, $R_t = C_2$, and for every $i < t$ there exists a peripheral vertex u of H and a path P in G from R_i to R_{i+1} through $\varphi^{-1}(u) - S_1$. The internal vertices of the path P are contained in a chunk $Q \subseteq \varphi^{-1}(u)$ of W. Since Q has a neighbor in R_i, Q is not C-dependent, and therefore $Q \subseteq \varphi'^{-1}(u)$. But then P is a path from R_i to R_{i+1} through $\varphi'^{-1}(u) - S_1$. Further $S'_1 \subseteq S_1$ and therefore P is a path from R_i to R_{i+1} through $\varphi'^{-1}(u) - S'_1$. So R_i and R_{i+1} are contained in the same component of $G' - S'_1$, and therefore so are C_1 and C_2. But C_1 and C_2 were arbitrarily chosen components in $G[B_{\varphi'}]$, hence B_φ is contained in a connected component of $G' - S'_1$. But every component of $G[B_\varphi]$ contains a component of $G[B_{\varphi'}]$, so all of $B_{\varphi'}$ is contained in one connected component of $G' - S'_1$. The proof that $B_{\varphi'}$ is contained in one connected component of $G' - S'_2$ is identical.

- For property *(iii)* let u be a peripheral vertex of H' and let C_1 and C_2 be components of $G'[A_{\varphi'} \cup B_{\varphi'}]$. Suppose there is a path P from C_1 to C_2 through $\varphi'^{-1}(u) - S'_1$ in G' Since $\varphi'^{-1}(u) \subseteq \varphi^{-1}(u)$ and $N_{G'}(\varphi'^{-1}(u)) \subseteq N_G(\varphi^{-1}(u)) \subseteq A_\varphi \cup B_\varphi$ the first and last vertices of P are in $A_\varphi \cup B_\varphi$. Let C'_1 be the connected component of $G[A_\varphi \cup B_\varphi]$ that contains the first vertex of P and C'_2 be the connected component of $G[A_\varphi \cup B_\varphi]$ that contains the last vertex of P. Note that every component of $G[A_\varphi \cup B_\varphi]$, with the exception of C, is contained in a component of $G[A_{\varphi'} \cup B_{\varphi'}]$, and that therefore $C'_1 \subseteq C_1$ and $C'_2 \subseteq C_2$. By property *(iii)* applied to W there is a path P' in G from C'_1 to C'_2 through $\varphi^{-1}(u) - S_2$ in G. The internal vertices of the path P' are contained in a chunk $Q \subseteq \varphi^{-1}(u)$ of W. Since Q has a neighbor in C'_1, Q is not C-dependent, and therefore $Q \subseteq \varphi'^{-1}(u)$. Thus P' is a path in G' from C'_1 to C'_2 through $\varphi'^{-1}(u) - S_2$. Since $S'_2 \subseteq S_2$ it follows that P' is a path in G' from C'_1 through C'_2 in $\varphi'^{-1}(u) - S'_2$. Since $C'_1 \subseteq C_1$ and $C'_2 \subseteq C_2$, P' is a path

in G' from C_1 to C_2 through $\varphi'^{-1}(u) - S'_2$. An identical proof shows that if there is a path in G' from C_1 and C_2 through $\varphi'^{-1}(u) - S'_2$, then there is also a path in G' from C_1 and C_2 through $\varphi'^{-1}(u) - S'_1$.

- For property *(iv)* let u be a peripheral vertex of H' and C_A be a component of $G'[A_{\varphi'}]$ with a neighbor in $\varphi'^{-1}(u)$. Then, by property *(iv)* applied to W there is a path P in G from C_A to B_φ through $\varphi^{-1}(u)$. The internal vertices of the path P are contained in a chunk $Q \subseteq \varphi^{-1}(u)$ of W. Since Q has a neighbor in C_A, Q is not C-dependent, and therefore $Q \subseteq \varphi'^{-1}(u)$. Since $B_\varphi \subseteq B_{\varphi'}$ it follows that P is a path in G' from C_A to $B_{\varphi'}$ through $\varphi'^{-1}(u)$.

  Consider now a peripheral vertex u of H' and a component C_B of $G'[B_{\varphi'}]$ with a neighbor y in $\varphi'^{-1}(u)$. Let Q be the connected component of $G'[\varphi'^{-1}(u)]$ that contains y. Then Q is a chunk of u in W, and furthermore, because $Q \subseteq \varphi'^{-1}(u)$, Q is not C-dependent. Therefore Q has a neighbor in $A_\varphi - C = A_{\varphi'}$, and so there is a path in G' from C_B to $A_{\varphi'}$ through $\varphi'^{-1}(u)$.

The proof for the case where C is a component of $G[B_\varphi]$ is symmetric.

For the upper bound on the adhesion size of every peripheral vertex u it is sufficient to observe that every connected component of $G'[A_{\varphi'} \cup B_{\varphi'}]$ that contains a neighbor of $\varphi' - 1(u)$ contains a connected component of $G[A_{\varphi'} \cup B_{\varphi'}]$ that contains a neighbor of $\varphi - 1(u)$. ■

Next we track what erasing a component of $G[A_\varphi \cup B_\varphi]$ does to the labeled connectivity graph. We will only track the effect on the connectivity graph for the side of the component C that we erase.

Lemma 7.5.35. *Let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature, and let C be an erasable component of $G[A_\varphi]$. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the generalized ω'-*

creature resulting from erasing C. Let $(\mathcal{C}_A, \lambda)$ be the labeled A-connectivity graph of W, and let $(\mathcal{C}'_A, \lambda')$ be the labeled A-connectivity graph of W'. Then $(\mathcal{C}'_A, \lambda') = (\mathcal{C}_A, \lambda) - \{C\}$.

Proof: The components of $G'[A_{\varphi'}]$ are precisely the components of $G[A_\varphi]$, except for C. Thus the vertex sets of $(\mathcal{C}'_A, \lambda')$ and $(\mathcal{C}_A, \lambda)$ are equal.

We first prove that for every pair of components C_1, C_2 of $G'[A_{\varphi'}]$, if u is a peripheral vertex of H that realizes $\{C_1, C_2\}$ (in W) then u is also a peripheral vertex of H' that realizes $\{C_1, C_2\}$ in W'. Towards this aim, suppose that there exists a path P in G from C_1 to C_2 through $\varphi^{-1}(u) - S_1$. Then the internal vertices of P are contained in a chunk Q of u. The chunk Q has a neighbor in C_1 and therefore it is not C-dependent. Thus $u \notin D(C)$, so u is a peripheral vertex in H' and P is a path from C_1 to C_2 through $\varphi'^{-1}(u)$. Since $S'_1 \subseteq S_1$ we conclude that P is a path from C_1 to C_2 through $\varphi'^{-1}(u) - S'_1$, and u realizes $\{C_1, C_2\}$ in W'.

Next we show that for every pair of components C_1, C_2 of $G'[A_{\varphi'}]$, if u is a peripheral vertex of H that realizes $\{C_1, C_2\}$ in W' then u also realizes $\{C_1, C_2\}$ in W. Towards this goal suppose that there exists a path P in G' from C_1 to C_2 through $\varphi'^{-1}(u')$. Then all internal vertices of P lie in a chunk Q of u'. Then Q is also a chunk of W, and since Q has a neighbor in C_1 it is not C-dependent. Thus $S_1 \cap Q = S'_1 \cap Q$ and so P is a path in G from C_1 to C_2 through $\varphi^{-1}(u')$. Hence u realizes $\{C_1, C_2\}$ in W. This concludes the proof. ■

Finally we track how erasing a component in $G[A_\varphi]$ affects which peripheral vertices have neighbors in which components of $G[A_\varphi]$.

Lemma 7.5.36. *Let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature, and let C be an erasable component of $G[A_\varphi]$. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the generalized ω'-creature resulting from erasing C. Then, for every peripheral vertex u in H and every component C' in $G[A_\varphi]$ with $C' \neq C$ it holds C' has a neighbor in $\varphi^{-1}(u)$ in G if and*

only if u is a peripheral vertex of H' and C' has a neighbor in $\varphi'^{-1}(u)$ in G'.

Proof: For the forward direction, suppose $\varphi'^{-1}(u)$ has a neighbor in C' in G'. Then u is not C-dependent and therefore u is a peripheral vertex of H'. Furthermore, $\varphi^{-1}(u)$ contains a chunk Q that has a neighbor in C'. Q is not C-dependent, and therefore $Q \subseteq \varphi'^{-1}(u)$. Thus $\varphi'^{-1}(u)$ has a neighbor of C' in G'.

For the reverse direction suppose that u is a peripheral vertex of H' and C' has a neighbor in $\varphi'^{-1}(u)$ in G'. Since $\varphi'^{-1}(u) \subseteq \varphi^{-1}(u)$ it follows that C' has a neighbor in $\varphi^{-1}(u)$ in G. ■

7.5.9 Path Filtering: Extracting an A-Path-Like ω-creature.

Definition 7.5.37. We will say that a generalized ω-creature W is A-path-like if the A-connectivity graph of W is a path.

Definition 7.5.38. A disjoint ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ is an *A-descendant* of a disjoint generalized ω-creature $W = (G, H, \varphi, S_1, S_2)$ if W' can be obtained from W by any sequence of dissolving peripheral vertices, erasing erasable components of $G[\varphi_A]$, absorbing absorbable components of $G[\varphi_A]$.

At a later stage in the proof we will show that if W has some nice properties (to be defined later) and W' is a descendant of W, then W' also has these properties. Note that the erasing component operation only applies to disjoint generalized ω-creatures, so the notion of descendant is only well defined for disjoint generalized ω-creatures. For now we will make the following simple observation.

Lemma 7.5.39. *If a disjoint ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ is an A-descendant of a disjoint ω'-creature $W = (G, H, \varphi, S_1, S_2)$ then G' is an induced subgraph of G. Furthermore, if W has adhesion size α then W' has adhesion size α*

Proof: Dissolving peripheral vertices and erasing erasable components of a disjoint generalized ω-creature $W = (G, H, \varphi, S_1, S_2)$ produces a generalized $\hat{\omega}$-creature $\hat{W} = (\hat{G}, \hat{H}, \hat{\varphi}, \hat{S}_1, \hat{S}_2)$ where $\hat{G}$ is an induced subgraph of G. Absorbing an absorbable component of $G[\varphi_A]$ produces a generalized $\hat{\omega}$-creature $\hat{W} = (\hat{G}, \hat{H}, \hat{\varphi}, \hat{S}_1, \hat{S}_2)$ where $\hat{G}$ is equal to G. The fact that G' is an induced subgraph of G now follows by induction on the number of operations used to obtain W' from W.

For the bound on the adhesion size, by Lemma 7.5.16 dissolving a peripheral vertex does not increase adhesion size, by Lemma 7.5.18 absorbing a component does not increase adhesion size, and by Lemma 7.5.34 erasing an erasable component does not increase adhesion size. The bound on the adhesion size of W' now follows by induction on the number of operations used to obtain W' from W. ■

Lemma 7.5.40 (Path Filtering)**.** *Let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature with adhesion size α, let $(\mathcal{C}_A, \lambda)$ be the labeled A-connectivity graph of W. Let (P, λ) be an induced path in $(\mathcal{C}_A, \lambda)$ Then there exists a disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ such that*

- *W' is A-path-like.*
- *W' is an A-descendant of W.*
- *$A_{\varphi'} = \bigcup_{C \in V(P)} C$*
- *Every peripheral vertex u of H' is a peripheral vertex of H.*
- *The labeled A-connectivity graph of W' is equal to (P, λ).*
- *For every peripheral vertex u of H and component $C' \in V(P)$ it holds that C' has a neighbor in $\varphi^{-1}(u)$ in G if and only if u is a peripheral vertex of H' and C' has a neighbor in $\varphi'^{-1}(u)$ in G'.*

Proof: We prove the lemma by induction on the number of components in $G[A_\varphi]$. For the base case, if $P = \mathcal{C}_A$, then W satisfies the conclusion of the lemma. So suppose that $|V(\mathcal{C}_A)| > |V(P)|$. Then, by Lemma 7.5.31, $G[A_\varphi]$ has a P-erasable component C. By Lemma 7.5.34, erasing C from W yields a disjoint generalized $\hat{\omega}$-creature $\hat{W} = (\hat{G}, \hat{H}, \hat{\varphi}, \hat{S}_1, \hat{S}_2)$ of adhesion size at most α. By Lemma 7.5.35 the labeled A-connectivity graph $\hat{\mathcal{C}}_A$ of $\hat{W}$ is equal to $(\hat{\mathcal{C}}_A, \lambda) - \{C\}$. Therefore (P, λ) is an induced subgraph of $(\hat{\mathcal{C}}_A, \lambda)$. Since the number of components of $\hat{G}[A_{\hat{\varphi}}]$ is less than the number of components in $G[A_\varphi]$, the induction hypothesis implies that there exists a disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ such that

- W' is A-path-like,
- W' is an A-descendant of $\hat{W}$ (and therefore also of W),
- $A_{\varphi'} = \bigcup_{C \in V(P)} C$,
- every peripheral vertex u of H' is a peripheral vertex of $\hat{H}$ (and thererfore also of H),
- the labeled A-connectivity graph of W' is equal to (P, λ), and
- for every peripheral vertex u of $\hat{H}$ and component $C' \in V(P)$ it holds that C' has a neighbor in $\hat{\varphi}^{-1}(u)$ in $\hat{G}$ if and only if u is a peripheral vertex of H' and C' has a neighbor in $\varphi'^{-1}(u)$ in G'.

For the last point, by By Lemma 7.5.36 we have that for every peripheral vertex u of H and component $C' \in V(P)$ it holds that C' has a neighbor in $\varphi^{-1}(u)$ in G if and only if u is a peripheral vertex of $\hat{H}$ and C' has a neighbor in $\hat{\varphi}^{-1}(u)$ in $\hat{G}$. We conclude that For every peripheral vertex u of H and component $C' \in V(P)$ it holds that C' has a neighbor in $\varphi^{-1}(u)$ in G if and only if u is a peripheral vertex of H' and C' has a neighbor in $\varphi'^{-1}(u)$ in G'. Thus W' satisfies the conclusion of the lemma. ■

Lemma 7.5.41. *Let G be a k-creature free graph and $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature with adhesion size α. Then there exists an A-path-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ with adhesion size α, such that W' is an A-descendant of G. Further, $\omega' \geqslant \frac{\log_{k\alpha}(\omega)-3}{\alpha}$.*

Proof: By Lemma 7.5.30, the $\mathcal{A}$-connectivity graph of W contains a path P on at least $\log_{k\alpha}(\omega/k) - 1$ vertices, and thus at least $\log_{k\alpha}(\omega/k) - 2$ edges. By Lemma 7.5.40 there exists an A-path-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ with adhesion size α, such that W' is an A-descendant of G. For every edge C_iC_j of P there is a peripheral vertex u of H' that realizes the edge. Then $\varphi'^{-1}(u)$ has neighbors in C_i and C_j (in G'). Hence, for every integer x, if u realizes x edges of P then u has neighbors in at least $x+1$ vertices of P (recall that vertices of P are components of $G'[A_{\varphi'}]$). But W' has adhesion size α and therefore each peripheral vertex of H' realizes at most $\alpha - 1$ edges of P. But then the number of peripheral vertices of H' is at least $\frac{\log_{k\alpha}(\omega/k)-2}{\alpha-1} \geqslant \frac{\log_{k\alpha}(\omega)-3}{\alpha}$.

■

7.5.10 Effect of Dissolve on the Connectivity Graph

We will use Lemma 7.5.41 to extract an A-path-like generalized ω-creature W. This gets us quite far towards making W a critter, but there are still many irregularities to clean up. For this we will use the "dissolve" operation, but now we need to be careful not to destroy the progress that we have already made. Since this progress is in the connectivity graph (in particular the connectivity graph is a path), we need to track how dissolving a peripheral vertex affects the connectivity graph.

At this point we will need to make a small detour and analyze how dissolving a peripheral vertex affects the (labeled) connectivity graph.

Lemma 7.5.42. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature with labeled A-connectivity graph $(\mathcal{C}_A, \lambda_A)$ and labeled B-connectivity graph $(\mathcal{C}_B, \lambda_B)$, and let u be a peripheral vertex of H. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the generalized $(\omega - 1)$-creature resulting from dissolving u in W. Then,*

(i) for every connected component C of $G[A_\varphi]$ there is a connected component C' of $G[A_{\varphi'}]$ such that $C \subseteq C'$.

(ii) Every connected component C' of $G[A_{\varphi'}]$ contains a connected component C of $G[A_\varphi]$.

(iii) For every pair of connected components C_1, C_2 of $G[A_\varphi]$ there exists a connected component C' of $G[A_{\varphi'}]$ such that $C_1 \cup C_2 \subseteq C'$ if and only if there exists a path P from C_1 to C_2 in $(\mathcal{C}_A, \lambda_A)$ such that every edge $\{Z, Z'\}$ of P satisfies $u \in \lambda_A(\{Z, Z'\})$.

(iv) For every pair of distinct connected components C'_1 and C'_2 of $G[A'_\varphi]$ and peripheral vertex v of H', v realizes $\{C'_1, C'_2\}$ in W' if and only if there exist components C_1 and C_2 of $G[A_\varphi]$ such that $C_1 \subseteq C'_1$, $C_2 \subseteq C'_2$ and v realizes $\{C_1, C_2\}$ in W.

(v) For every connected component C' of $G[A_{\varphi'}]$ and peripheral vertex v of H', $\varphi'^{-1}(v)$ has a neighbor in C' in G' if and only if C' contains a component C of $G[A_\varphi]$ such that $\varphi^{-1}(v)$ has a neighbor in C.

Furthermore, all of the above statements hold with A replaced by B (and A_φ by B_φ and $A_{\varphi'}$ by B_φ).

Proof: Since $A_\varphi \subseteq A'_\varphi$ it follows immediately that for every connected component C of $G[A_\varphi]$ there is a connected component C' of $G[A'_\varphi]$ such that $C \subseteq C'$.

Consider now a connected component C' of $G[A_{\varphi'}]$. If C' does not contain a vertex from $A'_\varphi - A_\varphi$ then $C' \subseteq A_\varphi$ and therefore C' is a component of $G[A_\varphi]$. If C' contains a

vertex a of $A'_\varphi - A_\varphi$, then $a \in A_1(W) \cap \varphi^{-1}(u)$. Since $G[A_1(W)]$ is connected, contains A_φ, and is disjoint from B_φ, the connected component of $G[A_1(W) \cap \varphi^{-1}(u)]$ that contains a has a neighbor a' in A_φ. But then C' contains a' and therefore also the connected component C of $G[A_\varphi]$ that contains a'

Let C_1 and C_2 be connected components of $G[A_\varphi]$. Suppose there exists a connected component C' of $G[A'_\varphi]$ such that $C_1 \cup C_2 \subseteq C'$. Let Q be a path from C_1 to C_2 through $C' - (C_1 \cup C_2)$. Let $q_1, q_2, \ldots, q_\ell$ be the vertices of Q that are also vertices of A_φ, in the order that they appear on Q. For every vertex $q_i \in \{q_1, \ldots, q_\ell\}$ define Z_i to be the connected component of $G[A_\varphi]$ that contains q. Then, for every $i < \ell$ it holds that either $Z_i = Z_{i+1}$ or the subpath of Q from q_i to q_{i+1} is a path from Z_i to Z_{i+1} through $A_{\varphi'} - A_\varphi$. But $A_{\varphi'} - A_\varphi \subseteq \varphi^{-1}(u) - S_1$. Thus, if $Z_i \neq Z_{i+1}$ then $\{Z_i, Z_{i+1}\}$ is an edge of $\mathcal{C}_A$ and u realizes $\{Z_i, Z_{i+1}\}$, so $u \in \lambda_A(\{Z_i, Z_{i+1}\})$. Hence there exists a walk from $Z_1 = C_1$ to $Z_\ell = C_2$ in $\mathcal{C}_A$ such that every edge $\{Z_i, Z_{i+1}\}$ of the walk satisfies $u \in \{Z_i, Z_{i+1}\}$. But then there also exists such a path from C_1 to C_2.

For the reverse direction let C_1 and C_2 be connected components of $G[A_\varphi]$ such that there exists a path P in $\mathcal{C}_A$ from C_1 to C_2 such that every edge $\{Z, Z'\}$ of P satisfies $u \in \lambda_A(\{Z, Z'\})$ For each edge $\{Z, Z'\}$ of P there exists a path Q from Z to Z' through $\varphi^{-1}(u) - S_1$. The component of $G[\varphi^{-1}(u) - S_1]$ that contains Q is in $A_1(W)$ and therefore $Q \subseteq A_{\varphi'}$. Hence Z and Z' are contained in the same component of $G[A_{\varphi'}]$. But then all vertices of P (including C_1 and C_2) are contained in the same component of $G[A_{\varphi'}]$.

Next, let C'_1 and C'_2 be a pair of distinct components of $G[A'_\varphi]$, and v be a peripheral vertex of H'. We prove the fourth statement.

For the forward direction, suppose v realizes $\{C'_1, C'_2\}$ in W'. Let P be a path in G' from C'_1 to C'_2 through $\varphi'^{-1}(v) - S'_1$. Let s be the first vertex of P and t be the last vertex of P. Since $\varphi'^{-1}(v) = \varphi^{-1}(v)$, $A'_\varphi \subseteq A_\varphi \cup \varphi^{-1}(u)$ and $\varphi^{-1}(v)$ and $\varphi^{-1}(u)$ are

anti-complete, it follows that $N_{G'}(\varphi'^{-1}(v)) \cap A_{\varphi'} \subseteq N_G(\varphi^{-1}(v)) \cap A_\varphi$. Hence s and t are both elements of A_φ. Let C_1 and C_2 be the components of $G[A_\varphi]$ that contain s and t respectively. We have that $C_1 \subseteq C'_1$ and $C_2 \subseteq C'_2$. Then P is a path in G from C_1 to C_2 through $\varphi'^{-1}(v) - S'_1 = \varphi^{-1}(v) - S_1$, so v realizes $\{C_1, C_2\}$ in W.

For the reverse direction, suppose there exist components C_1 and C_2 of $G[A_\varphi]$ such that $C_1 \subseteq C'_1$, $C_2 \subseteq C'_2$ and v realizes $\{C_1, C_2\}$ in W. Let P be a path in G from C_1 to C_2 through $\varphi^{-1}(v) - S_1$. Then P is a path in G' from C_1 to C_2 through $\varphi^{-1}(v) - S_1 = \varphi'^{-1}(v) - S'_1$. Since $C_1 \subseteq C'_1$, $C_2 \subseteq C'_2$ it follows that v realizes $\{C'_1, C'_2\}$ in W'.

Now we show the fifth property. Let C' be a connected component of $G[A_{\varphi'}]$ and v be peripheral vertex v of H'. For the forward direction suppose that $\varphi'^{-1}(v)$ contains a neighbor x in C'. We have that $C' \subseteq A_\varphi \cup \varphi^{-1}(u)$. On the other hand $\varphi'^{-1}(v) = \varphi^{-1}(v)$ and $\varphi^{-1}(u)$ are anticomplete, so $x \in A_\varphi$. Let C be the component of $G[A_\varphi]$ that contains x. By property *(i)* $C \subseteq C'$ and C.

For the reverse direction suppose that C' contains a component C of $G[A_\varphi]$ such that $\varphi^{-1}(v)$ has a neighbor in C. Then $\varphi'^{-1}(v) = \varphi^{-1}(v)$ has a neighbor in $C \subseteq C'$ in G'.

The proofs of the corresponding statements for B_φ and B'_φ are symmetric. ■

Next we need a lemma that tracks the effect on the connectivity graph if we dissolve many peripheral vertices instead of just one. To avoid a (slightly) technical induction we do not fully generalize Lemma 7.5.42 to dissolving sets of peripheral vertices, and instead prove a slightly weaker set of statements that are still sufficient for our needs.

Lemma 7.5.43. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature with labeled A-connectivity graph $(\mathcal{C}_A, \lambda_A)$ and labeled B-connectivity graph $(\mathcal{C}_B, \lambda_B)$, and let U be a set of peripheral vertices of H. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the generalized $(\omega - |U|)$-creature resulting from dissolving all peripheral vertices $u \in U$ in W. Then,*

(i) for every connected component C of $G[A_\varphi]$ there is a connected component C' of

$G[A_{\varphi'}]$ *such that* $C \subseteq C'$.

(ii) Every connected component C' *of* $G[A_{\varphi'}]$ *contains a connected component* C *of* $G[A_\varphi]$.

(iii) For every pair of connected components C_1, C_2 *of* $G[A_\varphi]$ *if there exists a path* P *from* C_1 *to* C_2 *in* $(\mathcal{C}_A, \lambda_A)$ *such that for every edge* $\{Z, Z'\}$ *of* P *there exists a* $u \in U$ *such that* $u \in \lambda_A(\{Z, Z'\})$, *then there exists a connected component* C' *of* $G[A_{\varphi'}]$ *such that* $C_1 \cup C_2 \subseteq C'$.

Furthermore, all of the above statements hold with A *replaced by* B *(and* A_φ *by* B_φ *and* $A_{\varphi'}$ *by* B_φ*).*

Proof: We prove the statements *(i)* and *(ii)* of the lemma by induction on $|U|$. If $|U| = 1$ the statements follows by Lemma 7.5.42. So suppose that the $|U| \geqslant 2$ and that the statement of the lemma holds for all smaller values of $|U|$.

Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω-creature with labeled A-connectivity graph $(\mathcal{C}_A, \lambda_A)$ and labeled B-connectivity graph $(\mathcal{C}_B, \lambda_B)$, and let U be a set of peripheral vertices of H. Let v be a vertex in U and $U' = U - \{v\}$. Let $\hat{W} = (\hat{G}, \hat{H}, \hat{\varphi}, \hat{S}_1, \hat{S}_2)$ be the generalized $(\omega-1)$-creature resulting from dissolving u in W. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be generalized $(\omega - |U|)$-creature resulting from dissolving all peripheral vertices $u \in U$ in W. Observe that W' is also the $(\omega - |U|)$-creature resulting from dissolving all peripheral vertices $u \in U'$ in $\hat{W}$.

Statement *(i)*: Let C be a connected component C of $G[A_\varphi]$. By Lemma 7.5.42 there exists connected component $\hat{C}$ of $\hat{G}[A_{\hat{\varphi}}]$ such that $C \subseteq \hat{C}$. By the induction hypothesis there exists a connected component C' of $G[A_{\varphi'}]$ such that $\hat{C} \subseteq C'$. But then $C \subseteq C'$.

Statement *(ii)*: Let C' be a connected component of $G[A_{\varphi'}]$. By the induction hypothesis C' contains a connected component $\hat{C}$ of $\hat{G}[A_{\hat{\varphi}}]$. By Lemma 7.5.42 $\hat{C}$ contains a connected component C of $G[A_\varphi]$. But then $C \subseteq C'$.

Statement *(iii)*: Let C_1 and C_2 be connected components of $G[A_\varphi]$ such that there exists a path P in $\mathcal{C}_A$ from C_1 to C_2 such that every edge $\{Z, Z'\}$ of P satisfies $u \in \lambda_A(\{Z, Z'\})$ For each edge $\{Z, Z'\}$ of P there exists a $u \in U$ and a path Q from Z to Z' through $\varphi^{-1}(u) - S_1$. The component of $G[\varphi^{-1}(u) - S_1]$ that contains Q is in $A_1(W)$ and therefore (since $u \in U$) we have that $Q \subseteq A_{\varphi'}$. Hence Z and Z' are contained in the same component of $G[A_{\varphi'}]$. But then all vertices of P (including C_1 and C_2) are contained in the same component of $G[A_{\varphi'}]$.

The proofs of the corresponding statements for B_φ and B'_φ are symmetric. ■

The next lemma shows that if we have already made an A-path-like or B-path-like creature then dissolving peripheral vertices will not break this property.

Lemma 7.5.44. *Let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature, U be a set of peripheral vertices of H, and $W' = (G, H, \varphi, S_1, S_2)$ be the disjoint generalized $(\omega - 1)$-creature resulting from dissolving u in W. If W is A-path-like then W' is A-path-like. If W is B-path-like then W' is B-path-like.*

Proof: We show that if W is A-path-like then W' is A-path-like. We only show the statement for $U = \{u\}$. The full statement for arbitrary sets U then follows immediately by induction on $|U|$.

Let $\mathcal{C}_A$ and $\mathcal{C}'_A$ be the A-connectivity graphs of W and W' respectively. We define a function f that assigns to every component C' of $G[A'_\varphi]$ the set of components C of $G[A_\varphi]$ such that $C \subseteq C'$. By statement *(i)* of Lemma 7.5.42 $f(C')$ is non-empty. The definition of f immediately implies that for two distinct components C'_1 and C'_2 of $G[A'_\varphi]$ the output of $f(C'_1)$ and of $f(C'_2)$ are disjoint. By statement *(iii)* of Lemma 7.5.42, $f(C')$ induces a connected subgraph of $\mathcal{C}_A$. By statement *(iv)* of Lemma 7.5.42, if $\{C'_1, C'_2\}$ is an edge of $\mathcal{C}'_A$ then there exists an edge from some vertex $C_1 \in f(C'_1)$ to a vertex $C_2 \in f(C'_2)$ in $\mathcal{C}'_A$. Hence f is a minor model of $\mathcal{C}'_A$ in $\mathcal{C}_A$. Since W is path-like, $\mathcal{C}_A$ is a path. Since $\mathcal{C}'_A$ is a

minor of $\mathcal{C}_A$, every connected component of $\mathcal{C}'_A$ is a path as well. Since $\mathcal{C}'_A$ is connected (by Lemma 7.5.28), $\mathcal{C}'_A$ is a path, and we conclude that W' is A-path-like.

The proof that if W is B-path-like then W' is B-path-like is symmetric. ■

7.5.11 From Path-Like to Irreducible

Let $W = (G, H, \varphi, S_1, S_2)$ be an A-path-like, disjoint generalized ω-creature, and $(\mathcal{C}_A, \lambda)$ be the labeled A-connectivity graph of W. Recall that vertices in the connectivity graph $\mathcal{C}_A$ are components of $G[A_\varphi]$. We may therefore talk about vertices of $\mathcal{C}_A$ being adjacent to vertices in G, or having neighbors in vertex sets in G. A path P in $\mathcal{C}_A$ *reduces* a peripheral vertex u of H if P has at least two vertices, $\varphi^{-1}(u)$ has a neighbor in the first vertex of P and a neighbor in the last vertex of P, and for every edge $\{C_1, C_2\}$ of P there is a peripheral vertex $y \neq x$ that realizes $\{C_1, C_2\}$. An *A-reduction pair* of W is a pair (u, P) where u is a peripheral vertex of H and P is a path in $\mathcal{C}_A$ that reduces u. When W is clear from context we will simply say that (u, P) is a A-reduction pair.

We say that a path-like, disjoint generalized ω-creature is *A-irreducible* if there does not exist a reduction pair (u, P) of W An *A-reduction packing* is a set $\mathcal{P}$ of reduction pairs such that for every pair (u_1, P_1), (u_2, P_2) of distinct A-reduction pairs in $\mathcal{P}$, $V(P_1) \cap V(P_2) = \emptyset$. In other words an A-reduction packing is a set of A-reduction pairs whose paths are pairwise vertex-disjoint. An *A-reduction hitting set* for W is a set $X \subseteq V(\mathcal{C}_A)$ such that for every A-reduction pair (u, P) of W we have $X \cap V(P) \neq \emptyset$.

Lemma 7.5.45. *Let $W = (G, H, \varphi, S_1, S_2)$ be a path-like, disjoint generalized ω-creature, and $(\mathcal{C}_A, \lambda)$ be the labeled A-connectivity graph of W. For every integer $p \geqslant 1$ there either exists an A-reduction packing $\mathcal{P}$ of size p or an A-reduction hitting set X of size $p - 1$.*

Proof: The lemma follows directly from the well-known fact (see e.g. [52]) that for every set $\mathcal{S}$ of intervals on the real line and integer p, there either exists a subset $\mathcal{S}' \subseteq \mathcal{S}$

of p pairwise disjoint intervals, or there exists a set $X \subseteq \mathbb{R}$ such that $|X| < p$ and every interval in $\mathcal{S}$ contains an element of X.

Indeed we associate every vertex of $\mathcal{C}_A$ with an integer, namely the position of this vertex in the path $\mathcal{C}_A$. Every A-reduction pair (u, P) is associated with the interval from the first to the last vertex of P. A set of disjoint intervals now corresponds to an A-reduction packing of the same cardinality. Similarly, since every interval starts and ends at an integer position, every set of reals that intersect all intervals can be assumed without loss of generality to be a set of integers, and therefore corresponds to an A-reduction hitting set of the same size. ■

An A-reduction pair (u_1, P_1) *conflicts* with another A-reduction pair (u_2, P_2) if $\varphi^{-1}(u_1)$ has a neighbor in a vertex of P_2 or $\varphi^{-1}(u_2)$ has a neighbor in a vertex of P_1. Note that the conflict relation is symmetric - if (u_1, P_1) conflicts with (u_2, P_2) then (u_2, P_2) conflicts with (u_1, P_1).

Lemma 7.5.46. *Let $W = (G, H, \varphi, S_1, S_2)$ be a path-like, disjoint generalized ω-creature with adhesion size α, and $\mathcal{P}$ be an A-reduction packing of W. Then there exists an A-reduction pair (u, P) in $\mathcal{P}$ that conflicts with at most $2\alpha - 1$ A-reduction pairs in $\mathcal{P}$.*

Proof: Given the A-reduction packing $\mathcal{P}$, define for every peripheral vertex u of H the set

$$Q(u) = \{(u', P') \in \mathcal{P} \ : \ \varphi^{-1}(u) \text{ has a neighbor in a vertex of } P'\}$$

Note that in an A-reduction packing $\mathcal{P}$ the paths in the A-reduction pairs are all vertex disjoint. Therefore, since u has adhesion size at most α it follows that $|Q(u)| \leqslant \alpha$ for every peripheral vertex u of H. For every path P in the A-connectivity graph $\mathcal{C}_A$ of W, define

$$Q(P) = \{(u', P') \in \mathcal{P} \ : \ \varphi^{-1}(u') \text{ has a neighbor in a vertex of } P\}$$

We have that

$$\begin{aligned}\sum_{(u,P)\in\mathcal{P}} |Q(P)| &= \sum_{(u,P)\in\mathcal{P}} \sum_{(u',P')\in\mathcal{P}} \begin{cases} 1 \text{ if } \varphi^{-1}(u') \text{ has a neighbor in a vertex of } P \\ 0 \text{ otherwise} \end{cases} \\ &= \sum_{(u',P')\in\mathcal{P}} |Q(u')| \\ &\leqslant |\mathcal{P}|\alpha\end{aligned}$$

Thus there exists a $(u,P) \in \mathcal{P}$ such that $|Q(P)| \leqslant \alpha$. Every A-reduction pair (u',P') that conflicts with (u,P) is in $Q(u) \cup Q(P)$. Indeed, if $\varphi^{-1}(u)$ has a neighbor in a vertex of P' then $(u',P') \in Q(u)$. If $\varphi^{-1}(u')$ has a neighbor in a vertex of P then $(u',P') \in Q(P)$. But $|Q(u) \cup Q(P)| \leqslant 2\alpha$, and $(u,P) \in Q(u)$, so (u,P) conflicts with at most $2\alpha - 1$ A-reduction pairs in $\mathcal{P}$, as claimed. ■

An A-reduction packing $\mathcal{P}$ is *conflict free* if no A-reduction pair in $\mathcal{P}$ conflicts with another A-reduction pair in $\mathcal{P}$.

Lemma 7.5.47. *Let $W = (G, H, \varphi, S_1, S_2)$ be a path-like, disjoint generalized ω-creature of adhesion size α and $\mathcal{P}$ be an A-reduction packing of W. Then there exists a conflict free A-reduction packing $\mathcal{P}' \subseteq \mathcal{P}$ such that $|\mathcal{P}'| \geqslant \mathcal{P}/2\alpha$.*

Proof: We prove the lemma by induction on $|\mathcal{P}|$. For $|\mathcal{P}| = 0$ the statement trivially holds, so suppose that $|\mathcal{P}| > 0$. By Lemma 7.5.46 there exists an A-reduction pair $(u,P) \in \mathcal{P}$ that conflicts with at most $2\alpha - 1$ A-reduction pairs in $\mathcal{P}$. Let $\mathcal{P}^\star$ be the subset of all A-reduction pairs in $\mathcal{P} - \{(u,P)\}$ that do not conflict with (u,P). We have that $|\mathcal{P}| - 2\alpha \leqslant |\mathcal{P}^\star| < |\mathcal{P}|$. By the induction hypothesis $\mathcal{P}^\star$ contains a conflict free A-reduction packing $\mathcal{P}'$ of size at least $|\mathcal{P}^\star|/2\alpha \geqslant (|\mathcal{P}|/2\alpha) - 1$. Since (u,P) does not conflict with any A-reduction pair in $\mathcal{P}'$ it follows that $\mathcal{P}' \cup \{(u,P)\}$ is a conflict packing of size at least $\mathcal{P}/2\alpha$, and that $\mathcal{P}' \cup \{(u,P)\} \subseteq \mathcal{P}$. This concludes the proof. ■

Lemma 7.5.48. *Let $W = (G, H, \varphi, S_1, S_2)$ be a path-like, disjoint generalized ω-creature of adhesion size α and $\mathcal{P}$ be a conflict free A-reduction packing of W. Then there exists a path-like disjoint, generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ with adhesion size $\alpha - 1$ and $\omega' = |\mathcal{P}|$. Furthermore, W' is an A-descendant of W.*

Proof: Define R to be the set of all peripheral vertices u of H such that there exists an A-reduction pair $(u, P) \in \mathcal{P}$. Since $\mathcal{P}$ is conflict free we have that for every $u \in R$ there is precisely one A-reduction pair (u, P) in $\mathcal{P}$. Indeed, if there were two such pairs (u, P_1) and (u, P_2) then (u, P_1) would conflict with (u, P_2) because $\varphi^{-1}(u)$ has a neighbor in the first vertex of P_2. Hence $|R| = \mathcal{P}$ We set U to be the set of all peripheral vertices of H that are not in R, and obtain $W' = (G', H', \varphi', S'_1, S'_2)$ by dissolving all peripheral vertices in u. By Lemma 7.5.16 W' is a disjoint generalized $|R|$-creature. By definition of A-descendants, W' is an A-descendant of W. By Lemma 7.5.44 W' is A-path-like.

It remains to show that the adhesion size of W' is at most $\alpha - 1$. Let v be a peripheral vertex of H' with maximum adhesion size in W', and let $(\mathcal{C}_\mathcal{A}, \lambda)$ be the labeled A-connectivity graph of W. Let $C_1, C_2, \ldots, C_\ell$ be the connected components of $G[A_\varphi \cup B_\varphi]$ that contain $N_G[\varphi^{-1}(v)]$. Since W has adhesion size α we have that $\ell \leqslant \alpha$. By Lemma 7.5.43 there exist components $C'_1, C'_2 \ldots, C'_\ell$ of $G'[A_{\varphi'} \cup B_{\varphi'}]$ such that $C_i \subseteq C'_i$ for every i. Furthermore $N_G[\varphi^{-1}(v)] \subseteq A_\varphi \cup B_\varphi$, G' is an induced subgraph of G, and $\varphi'^{-1}(v) = \varphi^{-1}(v)$. Thus $N_{G'}[\varphi'^{-1}(v)] \subseteq N_G[\varphi^{-1}(v)]$. Hence $C'_1, C'_2 \ldots, C'_\ell$ contain $N_{G'}[\varphi'^{-1}(v)]$.

Since $v \in R$ there is a A-reduction pair $(v, P) \in \mathcal{P}$. The endpoints of P are components of $G[A_\varphi]$ that contain neighbors of $\varphi^{-1}(v)$ in G. Without loss of generality, C_1 and C_2 are the two endpoints of P. Since (v, P) is a A-reduction pair we have that for every edge $\{Z, Z'\}$ of P there exists a peripheral vertex $u \neq v$ of H such that $u \in \lambda(\{Z, Z'\})$.

We claim that $u \in U$. Suppose not, then there exists an A-reduction pair $(u, P') \in \mathcal{P}$.

But $\varphi^{-1}(u)$ has a neighbor in Z (since u realizes $\{Z, Z'\}$) and therefore (u, P') conflicts with (v, P), contradicting that $\mathcal{P}$ is conflict free. We conclude that $u \in U$.

But then, for every edge $\{Z, Z'\}$ of P there exists a peripheral vertex $u \in U$ such that $u \in \lambda(\{Z, Z'\})$. By statement *(iii)* of Lemma 7.5.43 C_1 and C_2 are contained in the same component of $G[A_{\varphi'}]$. But then $C'_1 = C'_2$ and therefore there are at most $\ell - 1 = \alpha - 1$ components of $G[A_{\varphi'} \cup B_{\varphi'}]$ that contain neighbors of $\varphi'^{-1}(v)$ in G'. Hence the adhesion size of v in W' is at most $\alpha - 1$. Since v was the vertex in H' with maximum adhesion size, the adhesion size of W' is at most $\alpha - 1$ as claimed. ■

Lemma 7.5.49. *Let $W = (G, H, \varphi, S_1, S_2)$ be a path-like, disjoint generalized ω-creature of adhesion size α, $X \subseteq V(P)$, X be an A-reduction hitting set for W and P be a path in the A-connectivity graph $\mathcal{C}_A$ of W such that $V(P)$ is disjoint from X. Then there exists an A-irreducible path-like disjoint generalized ω' creature W' which is A-descendant of W, with adhesion size α and $\omega' \geqslant |V(P)|/\alpha$.*

Proof: Let $W' = (G', H', \varphi', S'_1, S'_2)$ be disjoint the ω'-creature obtained from W and P by the Path Filtering Lemma (Lemma 7.5.40). We claim that W' satisfies the conclusion of the lemma. From Lemma 7.5.40 we have that:

- W' is A-path-like.
- W' is an A-descendant of W.
- $A_{\varphi'} = \bigcup_{C \in V(P)} C$
- Every peripheral vertex u of H' is a peripheral vertex of H.
- The labeled A-connectivity graph of W' is equal to (P, λ), where $(\mathcal{C}_A, \lambda)$ is the labeled A-connectivity graph of W.
- For every peripheral vertex u of H and component $C' \in V(P)$ it holds that C' has a neighbor in $\varphi^{-1}(u)$ in G if and only if u is a peripheral vertex of H' and C' has

a neighbor in $\varphi'^{-1}(u)$ in G'.

By Lemma 7.5.39, the adhesion size of W' is at most α.

All that remains to show is that W' is A-irreducible, and that $\omega' \geqslant |V(P)|/\alpha$. To see that W' is A-irreducible, suppose for contradiction that u, P' is an A-reduction pair in W'. Then u is also a peripheral vertex of H, P' is also an induced path in G, $\varphi^{-1}(u)$ has a neighbor in the first and last vertex of P' in G, and every edge $\{C_1, C_2\}$ of p satisfies that $\lambda(\{C_1, C_2\}) - \{u\}$ is non-empty. But then (u, P') is an A-reduction pair in W. Since P' is a sub-path of P and $V(P)$ is disjoint from X it follows that $V(P')$ is disjoint from X. This contradicts that X is an A-reduction hitting set for W.

We now show that $\omega' \geqslant |V(P)|/\alpha$. Indeed, for every edge $\{C_1, C_2\}$ of P some peripheral vertex u of H' realizes $\{C_1, C_2\}$ in W'. Then $\varphi^{-1}u$ has a neighbor both in C_1 and in C_2.

Since every vertex of P is incident to an edge of P it holds that for every vertex C of P there exists a peripheral vertex u of H' such that $\varphi^{-1}u$ has a neighbor in C. Thus, since W' has adhesion size α, the number of peripheral vertices of H' is at least $|V(P)|/\alpha$. Hence W' is a generalized ω'-creature with $\omega' \geqslant |V(P)|/\alpha$. ■

Lemma 7.5.50. *Let G be k-creature free, and $W = (G, H, \varphi, S_1, S_2)$ be a path-like, disjoint generalized ω-creature of adhesion size α. Then there exists a path-like disjoint, A-irreducible generalized ω' creature $W' = (G', H', \varphi', S'_1, S'_2)$ with adhesion size α, such that $\omega' \geqslant \frac{\omega^{1/\alpha}}{4k\alpha^2}$. Furthermore W' is an A-descendant of W.*

Proof: We prove the statement of the lemma by induction on α. If $\alpha \leqslant 1$ then no A-reduction pair (u, P) can exist. In particular in every A-reduction pair (u, P), $\varphi^{-1}(u)$ has a neighbor in the first and last vertex of P, and these two vertices are distinct. Thus u has adhesion size at least 2. Thus, if $\alpha \leqslant 1$ then W already satisfies the conclusion of the lemma.

Suppose now that $\alpha \geqslant 2$ and that the statement of the lemma holds for all lower values of α. Let $W = (G, H, \varphi, S_1, S_2)$ be a path-like, disjoint generalized ω-creature of adhesion size α. If W is A-irreducible it already satisfies the conclusion of the lemma. We consider the case where W is not A-irreducible, and set $p = \omega^{1-\frac{1}{\alpha}} \cdot 2\alpha$. By Lemma 7.5.45 there either exists an A-reduction packing $\mathcal{P}$ of size p or an A-reduction hitting set X of size $p-1$.

We fist consider the case that there exists an A-reduction packing $\mathcal{P}$ of size p. Then, by Lemma 7.5.47 there exists a conflict-free A-reduction packing $\mathcal{P}' \subseteq \mathcal{P}$ of size at least $|\mathcal{P}|/2\alpha$. Then, by Lemma 7.5.48 there exists a path-like disjoint generalized $\hat{\omega}$-creature $\hat{W}$, A-descendant of W, with adhesion size $\alpha - 1$ and $\hat{\omega} = |\mathcal{P}'| \geqslant p/2\alpha \geqslant \omega^{1-\frac{1}{\alpha}}$. By the induction hypothesis applied to $\hat{W}$ there exists an A-irreducible path-like disjoint, generalized ω' creature $W' = (G', H', \varphi', S_1', S_2')$, A-descendant of $\hat{W}$ (and therefore of W) with adhesion size $\alpha - 1$ and

$$\omega' \geqslant \frac{\hat{\omega}^{\frac{1}{\alpha-1}}}{4k(\alpha-1)^2} \geqslant \frac{(\omega^{1-\frac{1}{\alpha}})^{\frac{1}{\alpha-1}}}{4k\alpha^2} = \frac{\omega^{\frac{1}{\alpha}}}{4k\alpha^2}$$

We now consider the case that there exists an A-reduction hitting set X of size p. On one hand, by Lemma 7.5.4, for every peripheral vertex u of H, $\varphi^{-1}(u)$ has a neighbor in A_φ. On the other hand, By Lemma 7.5.29, for every a component C of $G[A_\varphi]$ there are fewer than k peripheral vertices u of H such that $\varphi^{-1}(u)$ has a neighbor in C. Therefore there are at least ω/k vertices in $\mathcal{C}_A$. Since $\mathcal{C}_A$ is a path, it contains a sub-path P on at least $\frac{(\omega/k)-|X|}{|X|+1} > \frac{\omega}{k(p+1)} - 1$ vertices disjoint from X. Since the number of vertices is an integer, P has at least $\frac{\omega}{k(p+1)}$ vertices. Thus, by Lemma 7.5.49 there exists an A-irreducible, path-like disjoint generalized ω' creature W' which is A-descendant of W, with adhesion size α and $\omega' \geqslant \left(\frac{\omega}{k(p+1)} - 1\right)/\alpha > \frac{\omega}{k(p+1)\alpha} - 1$. Since ω' is an integer it

follows that

$$\omega' \geqslant \frac{\omega}{k(p+1)\alpha} \geqslant \frac{\omega}{2kp\alpha} \geqslant \frac{\omega^{\frac{1}{\alpha}}}{4k\alpha^2}.$$

This concludes the proof. ■

Lemma 7.5.51. *Let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature with adhesion size α. Then there exists a path-like disjoint, A-irreducible generalized ω' creature $W' = (G', H', \varphi', S_1', S_2')$ with adhesion size α, such that $\omega' \geqslant \frac{(\log_{k\alpha}(\omega)-3)^{1/\alpha}}{8k\alpha^2}$. Furthermore W' is an A-descendant of W.*

Proof: Let G be a k-creature free graph and $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature with adhesion size α. Then, by Lemma 7.5.41 there exists an A-path-like disjoint generalized ω''-creature $W'' = (G'', H'', \varphi'', S_1'', S_2'')$ with adhesion size α, such that W'' is an A-descendant of G. Further, $\omega'' \geqslant \frac{\log_{k\alpha}(\omega/k)-2}{\alpha-1}$. By Lemma 7.5.39, G'' is an induced subgraph of G and therefore also k-creature free.

By Lemma 7.5.50 applied to W'' there exists a path-like disjoint, A-irreducible generalized ω' creature $W' = (G', H', \varphi', S_1', S_2')$ with adhesion size α, such that

$$\omega' \geqslant \frac{\omega''^{1/\alpha}}{4k\alpha^2} \geqslant \frac{(\frac{\log_{k\alpha}(\omega)-3}{\alpha})^{1/\alpha}}{4k\alpha^2} \geqslant \frac{(\log_{k\alpha}(\omega)-3)^{1/\alpha}}{8k\alpha^2}.$$

Thus W' satisfies the conclusion of the lemma. ■

7.5.12 From A-Path-like and A-Irreducible to A-Critter-like

Definition 7.5.52 (A-critter-like and B-critter-like)**.** Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized disjoint ω-creature, and let $(\mathcal{C}_A, \lambda_A)$ and $(\mathcal{C}_B, \lambda_B)$ be the labeled A-connectivity graph and the labeled B-connectivity graph of W, respectively. W is A-critter-like if:

- $\mathcal{C}_A$ is a path,

- Every edge $\{C_1, C_2\}$ of $\mathcal{C}_A$ satisfies $|\lambda_A(\{C_1, C_2\})| = 1$.
- For every peripheral vertex u of H and pair $\{C_1, C_2\}$ of components of $G[\varphi_A]$, if $\varphi^{-1}(u)$ has a neighbor in C_1 and a neighbor in C_2 then u realizes $\{C_1, C_2\}$ in W.

Similarly, W is B-critter-like if:

- $\mathcal{C}_B$ is a path,
- Every edge $\{C_1, C_2\}$ of $\mathcal{C}_B$ satisfies $|\lambda_B(\{C_1, C_2\})| = 1$.
- For every peripheral vertex u of H and pair $\{C_1, C_2\}$ of components of $G[\varphi_B]$, if $\varphi^{-1}(u)$ has a neighbor in C_1 and a neighbor in C_2 then u realizes $\{C_1, C_2\}$ in W.

In this section we show how to extract an A-critter-like disjoint generalized ω'-creature from an A-path-like disjoint generalized ω'-creature.

We will need to get rid of peripheral vertices of H that do not realize any edges of $\mathcal{C}_A$. More formally, let $W = (G, H, \varphi, S_1, S_2)$ be an A-path-like generalized ω-creature, and let $\mathcal{C}_A$ be the A-connectivity graph of W. We say that a peripheral vertex u of H is A-*useless* if u does not realize any pair $\{C_1, C_2\}$ of distinct components of $G[A_\varphi]$, and $\varphi^{-1}(u)$ has no neighbors in the endpoints of $\mathcal{C}_A$. We would like to dissolve such peripheral vertices, thus we need to track what dissolving them does to the connectivity graph.

Lemma 7.5.53. *Let $W = (G, H, \varphi, S_1, S_2)$ be an A-path-like, A-irreducible disjoint generalized ω-creature of adhesion size α, and let $(\mathcal{C}_A, \lambda)$ be the labeled A-connectivity graph of W. Let u be an A-useless peripheral vertex of H. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the generalized ω-creature resulting from dissolving u in W. Let $(\mathcal{C}'_A, \lambda')$ be the labeled A-connectivity graph of W'. Then W' is an A-path-like, A-irreducible, and disjoint generalized $(\omega - 1)$-creature of adhesion size α. Furthermore, for every peripheral vertex v of H, if v is not A-useless in W then v is not A-useless in W'.*

Proof: By Lemma 7.5.16 we have that W' is a disjoint generalized $(\omega-1)$-creature of adhesion size α. By Lemma 7.5.42 statement *(i)* every connected component of $G[A_\varphi]$ is contained in a component of $G[A_{\varphi'}]$. By Lemma 7.5.42 statements *(ii)* and *(iii)*, every connected component C' of $G[A_{\varphi'}]$ contains precisely one component of $G[A_\varphi]$. By Lemma 7.5.42 statement *(iv)* a peripheral vertex v of H' realizes a pair $\{C'_1, C'_2\}$ of components of $G[A_{\varphi'}]$ in W' if and only if v realizes $\{C_1, C_2\}$ in W, where C_1 and C_2 are the unique components of $G[A_\varphi]$ contained in C'_1 and C'_2 respectively.

Hence the bijection ψ that maps a component of $G[A_\varphi]$ to the unique component of $G[A_{\varphi'}]$ that contains it is an isomorphism from $(\mathcal{C}_A, \lambda)$ to $(\mathcal{C}'_A, \lambda')$, in the sense that $\{C, C'\} \in E(\mathcal{C}_A)$ if and only if $\{\psi(C), \psi(C')\} \in E(\mathcal{C}_A)$ and $\lambda(\{C, C'\}) = \lambda'(\{\psi(C), \psi(C')\})$.

Hence W' is A-path-like, and every peripheral vertex v which realizes an edge of $\mathcal{C}_A$ also realizes the same edge of $\mathcal{C}'_A$. Similarly, every peripheral vertex v with a neighbor in an endpoint Q of $\mathcal{C}_A$ has a neighbor in $\psi(Q)$. Hence, if v is not A-useless in W it is not A-useless in W' either. Furthermore, if W' has an A-reduction pair (v, P) then $(v, \psi^{-1}(P))$ is an A-reduction pair in W. Thus W' is A-irreducible. ■

For this we will absorb absorbable components of $G[A_\varphi]$. We therefore need to track how absorbing a component of $G[A_\varphi]$ affects the labeled A-connectivity graph.

Lemma 7.5.54. *Let $\omega \geqslant 2$ and $W = (G, H, \varphi, S_1, S_2)$ be an A-path-like, A-irreducible disjoint generalized ω-creature of adhesion size α such that H has no A-useless peripheral vertices. Let C be an absorbable component of $G[\varphi_A]$, and u be the peripheral vertex of H such that $N(C) \subseteq \varphi^{-1}(u)$. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the generalized ω-creature resulting from absorbing C in W. Then W' is an A-path-like, A-irreducible, and disjoint generalized ω-creature of adhesion size α, and H' has no A-useless peripheral vertices.*

Proof: By Lemma 7.5.18 W' is a disjoint generalized ω-creature of adhesion size α. By definition of absorb, $A_{\varphi'} = A_\varphi - C$ so the set of connected components of $G[A_{\varphi'}]$ is

precisely the set of connected components of $G[A_\varphi]$ minus $\{C\}$.

We observe that $G[A_\varphi]$ has at least 2 components. Otherwise C is the only component of $G[A_\varphi]$. Then Lemma 7.5.4 yields that for every peripheral vertex v of H, $\varphi^{-1}(v)$ has a neighbor in C, contradicting that C is absorbable.

Let $(\mathcal{C}_A, \lambda)$ be the labeled A-connectivity graph of W and $(\mathcal{C}'_A, \lambda')$ be the labeled A-connectivity graph of W'. By assumption $\mathcal{C}_A$ is a path. We have that $S'_1 = S_1$ and for every peripheral vertex $v \neq u$ of H' we have that $\varphi'^{-1}(v) = \varphi^{-1}(v)$. Hence for every pair of components C_1, C_2 of $G[A_{\varphi'}]$ and peripheral vertex $v \neq u$ we have that v realizes $\{C_1, C_2\}$ in W' if and only if v realizes $\{C_1, C_2\}$ in W.

There are two cases, C is an endpoint of $\mathcal{C}_A$ or not. Suppose C is an endpoint of $\mathcal{C}_A$. We claim that for every pair $\{C_1, C_2\}$ of components of $G[A_{\varphi'}]$, u realizes $\{C_1, C_2\}$ in W' if and only if u realizes $\{C_1, C_2\}$ in W. For the reverse direction, let P be a path from C_1 to C_2 through $\varphi^{-1}(u) - S_1$. Then P is also a path from C_1 to C_2 through $\varphi'^{-1}(u) - S'_1$, because $\varphi'^{-1}(u) = \varphi^{-1}(u) \cup C$, $S'_1 = S'_1$ and S_1 is disjoint from C. Hence u realizes $\{C_1, C_2\}$ in W'.

For the forward direction, suppose that u realizes an $\{C_1, C_2\}$ in W' Then there exists a path P from C_1 to C_2 though $\varphi'^{-1}(u) - S'_1 = (\varphi^{-1}(u) \cup C) - S_1$. Since C only has one neighbor in $\mathcal{C}_A$, without loss of generality C_1 is not a neighbor of C in $\mathcal{C}_A$. If P contains a vertex of C, let P' be a shortest sub-path of P with one endpoint in C_1 and the other in C. Then P' is a path from C to C_1 through $\varphi^{-1}(u) - S_1$, contradicting that C_1 is non-neighbor of C in $\mathcal{C}_A$. Therefore P is disjoint from C. Then P is a path from C_1 to C_2 though $\varphi^{-1}(u) - S_1$, so u realizes $\{C_1, C_2\}$ in W.

Thus, when C is an endpoint of $\mathcal{C}_A$ we get that $(\mathcal{C}'_A, \lambda') = (\mathcal{C}_A - \{C\}, \lambda)$. Hence W' is path-like. Furthermore every A-reduction pair (v, P) in W' is an A-reduction pair also in W. Hence W' is irreducible. It remains to show that no vertices of H' are A-useless. Consider now a peripheral vertex $v \neq u$ of H. Since v is not A-useless in W, v

realizes an edge of $\mathcal{C}_A$ or $\varphi^{-1}(v)$ has a neighbor in an endpoint of $\mathcal{C}_A$. Since $v \neq u$ and $N(C) \subseteq \varphi^{-1}(u)$ we have that v has no neighbors in C. Thus if v realizes an edge $\{C_1, C_2\}$ of $\mathcal{C}_A$ in W then v realizes $\{C_1, C_2\}$ in W'. If $\varphi^{-1}(v)$ has a neighbor in an endpoint C' of $\mathcal{C}_A$ then $C' \neq C$ and v has a neighbor in C' in W' as well. Hence v is not A-useless in W' Finally, let C_1 be the unique neighbor of C in $\mathcal{C}_A$. Since u realizes the edge $\{C, C_1\}$ in $\mathcal{C}_A$, $\varphi^{-1}(u)$ has a neighbor in C_1, and C_1 is an endpoint of $\mathcal{C}'_A$. Therefore u is not A-useless in W'. We conclude that in the case when u is an endpoint of $\mathcal{C}_A$ the statement of the lemma holds.

Suppose now that C is not an endpoint of $\mathcal{C}_A$ and let X and Y be the predecessor and successor of C on the path $\mathcal{C}_A$, respectively. We claim that u realizes a pair $\{C_1, C_2\}$ of components of $G[A_{\varphi'}]$ if and only if u realizes $\{C_1, C_2\}$ in W or $\{C_1, C_2\} = \{X, Y\}$. For the reverse direction, suppose u realizes $\{C_1, C_2\}$ in W. Let P be a path from C_1 to C_2 through $\varphi^{-1}(u) - S_1$. Then P is also a path from C_1 to C_2 through $\varphi'^{-1}(u) - S'_1$, because $\varphi'^{-1}(u) = \varphi^{-1}(u) \cup C$, $S'_1 = S'_1$ and S_1 is disjoint from C. Hence u realizes $\{C_1, C_2\}$ in W'.

Suppose now that $C_1 = X$ and $C_2 = Y$. Then there is a path P_1 from C_1 to C through $\varphi^{-1}(u) - S_1$, and a path P_2 from C to C_2 through $\varphi^{-1}(u) - S_1$. Since $S_1 = S'_1$ and $\varphi^{-1}(u) \cup C = \varphi'^{-1}(u)$ there is a walk from C_1 to C_2 through $\varphi'^{-1}(u)$, namely P_1, followed by a walk from the end of P_1 to the start of P_2 in $G[C]$ and then by P_2 from C to C_2. Hence u realizes $\{C_1, C_2\} = \{X, Y\}$ in W'.

For the forward direction, suppose that u realizes an $\{C_1, C_2\}$ in W' Then there exists a path P from C_1 to C_2 though $\varphi'^{-1}(u) - S'_1 = (\varphi^{-1}(u) \cup C) - S_1$. If P is disjoint from C then P is a path from C_1 to C_2 though $\varphi^{-1}(u) - S_1$, so u realizes $\{C_1, C_2\}$ in W. If $V(P)$ intersects with C then P contains a sub-path P_1 from C_1 to C through $\varphi'^{-1}(u) - C = \varphi^{-1}(u)$ and a path P_2 from C_2 to C through $\varphi'^{-1}(u) - C = \varphi^{-1}(u)$. But then C_1 and C_2 are both adjacent to C in $\mathcal{C}_A$ and therefore $\{C_1, C_2\} = \{X, Y\}$. Thus,

when C is not an endpoint of $\mathcal{C}_A$ we get that $(\mathcal{C}'_A, \lambda')$ and $(\mathcal{C}_A - \{C\}, \lambda)$ are equal, except that $\{X, Y\}$ is a non-edge of $\mathcal{C}_A - \{C\}$ and an edge of $\mathcal{C}'_A$ with label $\lambda'(\{X, Y\}) = \{u\}$. Hence W' is path-like.

We show that W' is A-irreducible. For every A-reduction pair (v, P) in W', if $\{X, Y\}$ is not an edge of P then (v, P) is also an A-reduction pair in W. If $\{X, Y\}$ is an edge of P then (v, P') is an A-reduction pair in W, where P' is the path obtained from P by removing the edge $\{X, Y\}$, adding the vertex C and the edges $\{X, C\}$ and $\{C, Y\}$. In particular $\lambda(\{X, C\}) = \lambda\{C, Y\} = \lambda'\{X, Y\} = \{u\}$ Thus, since W is A-irreducible, W' is A-irreducible as well.

It remains to show that no vertices of H' are A-useless. Consider now a peripheral vertex $v \neq u$ of H. Since v is not A-useless in W, v realizes an edge of $\mathcal{C}_A$ or $\varphi^{-1}(v)$ has a neighbor in an endpoint of $\mathcal{C}_A$. Since $v \neq u$ and $N(C) \subseteq \varphi^{-1}(u)$ we have that v has no neighbors in C. Thus if v realizes an edge $\{C_1, C_2\}$ of $\mathcal{C}_A$ in W then v realizes $\{C_1, C_2\}$ in W'. If $\varphi^{-1}(v)$ has a neighbor in an endpoint C' of $\mathcal{C}_A$ then $C' \neq C$ and v has a neighbor in C' in W' as well. Hence v is not A-useless in W' Finally observe that u realizes $\{X, Y\}$ in W' and therefore is not A-useless in W'. This concludes the proof. ■

Lemma 7.5.55. *If W is an A-path-like, A-irreducible, and disjoint generalized ω-creature that does not have A-useless peripheral vertices or absorbable components in $G[A_\varphi]$, then W is A-critter-like.*

Proof: We prove that W satisfies each of the three properties of A-critter-like generalized ω-creatures. Let $(\mathcal{C}_A, \lambda)$ be the A-connectivity graph of W. First, $\mathcal{C}_A$ is a path because W is A-path-like. For the second property, let $\{C_1, C_2\}$ be an edge of the A-connectivity graph $\mathcal{C}_A$. Since every edge of $\mathcal{C}_A$ is realized by at least one peripheral vertex u of H we have that $|\lambda_A(\{C_1, C_2\})| \geqslant 1$. We now show that $|\lambda_A(\{C_1, C_2\})| \leqslant 1$. Suppose for contradiction that there exists a peripheral vertex $v \neq u$ such that $v \in \lambda_A(\{C_1, C_2\})$.

Since u realizes $\{C_1, C_2\}$ we have that $\varphi^{-1}(u)$ has neighbors both in C_1 and in C_2. But then (u, C_1C_2) is an A-reduction pair, contradicting that W is A-irreducible.

For the third property, suppose for contradiction that there exists a peripheral vertex u such that $\varphi^{-1}(u)$ has neighbors in two distinct components C_1, C_2 of $G[A_\varphi]$ and u does not realize $\{C_1, C_2\}$. Let P be the path from C_1 to C_2 in $\mathcal{C}_A$. We consider two cases, either at least one edge of P is labeled with $\{v\}$ for some peripheral vertex $v \neq u$, or all edges of P are labeled with $\{u\}$.

Suppose first that at least one edge $\{Z, Z'\}$ of P is labeled with $\{v\}$ for some peripheral vertex $v \neq u$. Let P' a the shortest sub-path of P that contains the edge $\{Z, Z'\}$ such that the first component C_1' of P' and the last component C_2' have a neighbor in $\varphi^{-1}(u)$. Since P itself satisfies the two properties, P' is well defined. Since P' *shortest*, no internal vertex of P' can have a neighbor in $\varphi^{-1}(u)$. But then u does not realize any edge of P' and therefore (u, P') is an A-reduction pair, contradicting that W is A-irreducible.

Suppose now that all edges of P are labeled with $\{u\}$. Since u does not realize $\{C_1, C_2\}$ the path P has at least one internal vertex C. Since C is not absorbable there exists a peripheral vertex $v \neq u$ such that C has a neighbor in $\varphi^{-1}(v)$. Since v is not A-useless, v realizes at least one edge of $\mathcal{C}_A$ or $\varphi^{-1}(v)$ has a neighbor in an endpoint of $\mathcal{C}_A$. If v realizes at an edge of $\mathcal{C}_A$ then this edge can not be incident to C, because all edges incident to C are labeled $\{u\}$. Hence $\varphi^{-1}(v)$ has a neighbor in at least one vertex of $\mathcal{C}_A$ other than C. If $\varphi^{-1}(v)$ has a neighbor in an endpoint of $\mathcal{C}_A$ then this endpoint is not equal to C, because C is an internal vertex of P. In either case $\varphi^{-1}(v)$ has a neighbor in at least one vertex of $\mathcal{C}_A$ other than C.

Let P' be a shortest path in $\mathcal{C}_A$ from C to another vertex of $\mathcal{C}_A$ that contains a neighbor of $\varphi^{-1}(v)$. None of the edges of P' are labeled $\{v\}$ because only the endpoints of P' contain neighbors of $\varphi^{-1}(v)$, and one of the endpoints, namely C, is not incident to any edges labeled $\{v\}$. But then (v, P') is an A-reduction pair, contradicting that W

is A-irreducible. We conclude that W is A-critter-like. ■

Lemma 7.5.56. *Let G be k-creature free, and $W = (G, H, \varphi, S_1, S_2)$ be an A-path-like, A-irreducible disjoint generalized ω creature of adhesion size α. Then there exists an A-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ with adhesion size α, such that $\omega' \geqslant \frac{\omega}{k\alpha} - 1$. Furthermore W' is an A-descendant of W.*

Proof: Let G be k-creature free, and $W = (G, H, \varphi, S_1, S_2)$ be an A-path-like, A-irreducible disjoint generalized ω creature of adhesion size α.

We claim that $G[A_\varphi]$ has at least ω/k components. Indeed, by Lemma 7.5.4 for every peripheral vertex v of H, $\varphi^{-1}(v)$ has a neighbor in A_φ, and by Lemma 7.5.29 for each component C of $G[A_\varphi]$ there are at most $k-1$ peripheral vertices v such that $\varphi^{-1}(v)$ has a neighbor in C. Thus $G[A_\varphi]$ has at least ω/k components.

Let Z be the set of all peripheral vertices of H that are not A-useless. We claim that $|Z| \geqslant \omega/k\alpha - 1$. Indeed, $\mathcal{C}_A$ has at least $\omega/k - 1$ edges, and each of these edges is realized by a peripheral vertex $v \in Z$. Since each peripheral vertex v can realize no more than α edges of $\mathcal{C}_A$ it follows that $|Z| \geqslant \frac{\omega}{k\alpha} - 1$.

Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the result of repeatedly dissolving A-useless peripheral vertices of H, as long as any are present. By Lemma 7.5.53 no vertices of Z get dissolved by this process. Furthermore, by Lemma 7.5.53 W' is an A-path-like, A-irreducible, and disjoint generalized ω'-creature of adhesion size α, where $\omega' = |Z| \geqslant \frac{\omega}{k\alpha} - 1$.

Let $W'' = (G'', H'', \varphi'', S''_1, S''_2)$ be the result of repeatedly absorbing absorbable components of $G'[A_{\varphi'}]$, as long as any are present. By Lemma 7.5.54 W'' is an A-path-like, A-irreducible, and disjoint generalized ω'-creature of adhesion size α, and H'' has no A-useless peripheral vertices. Furthermore, $G''[A_{\varphi''}]$ has no absorbable components. Since W'' was obtained by dissolving peripheral vertices and absorbing absorbable components of $G[A_\varphi]$, W'' is an A-descendant of W. By Lemma 7.5.55 W'' satisfies the conclusion of

the lemma. ■

Lemma 7.5.57. *Let G be k-creature free, and $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω creature of adhesion size α. Then there exists an A-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ with adhesion size α, such that $\omega' \geqslant \frac{(\log_{k\alpha}(\omega)-3)^{1/\alpha}}{8k^2\alpha^3} - 1$. Furthermore W' is an A-descendant of W.*

Proof: By Lemma 7.5.51 applied to W there exists an A-path-like, A-irreducible disjoint generalized ω' creature $W'' = (G'', H'', \varphi'', S''_1, S''_2)$ with adhesion size α, such that $\omega'' \geqslant \frac{(\log_{k\alpha}(\omega)-3)^{1/\alpha}}{8k\alpha^2}$. Furthermore W'' is an A-descendant of W. By Lemma 7.5.56 applied to W'' there exists an A-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ with adhesion size α, such that $\omega' \geqslant \frac{\omega''}{k\alpha} - 1 \geqslant \frac{(\log_{k\alpha}(\omega)-3)^{1/\alpha}}{8k^2\alpha^3} - 1$. Furthermore W' is an A-descendant of W'', and hence of W. This completes the proof. ■

7.5.13 A-Descendents stay B-Critter-Like

Lemma 7.5.58. *Let $W = (G, H, \varphi, S_1, S_2)$ be a B-critter-like disjoint generalized ω-creature and u be a peripheral vertex in H. Then either there exists a unique component C of $G[B_\varphi]$ such that $\varphi^{-1}(u)$ has a neighbor in C, or there exist precisely two components C_1, C_2 of $G[B_\varphi]$ such that u realizes $\{C_1, C_2\}$. In this case no other peripheral vertices of H realize $\{C_1, C_2\}$. Furthermore all of the statements above hold with B replaced by A.*

Proof: Let $(\mathcal{C}_B, \lambda)$ be the B-connectivity graph of W. By Lemma 7.5.4 there exists at least one component C_1 such that $\varphi^{-1}(u)$ has a neighbor in C_1. If there exist two components C_1, C_2 such that $\varphi^{-1}(u)$ has a neighbor in C_1 and a neighbor in C_2, then by the third property of critter-like generalized ω-creatures u realizes $\{C_1, C_2\}$. Then $\{C_1, C_2\}$ is an edge of $\mathcal{C}_B$ Further, if another peripheral vertex v also realizes $\{C_1, C_2\}$ then $\{u, v\} \subseteq \lambda(\{C_1, C_2\})$, contradicting the second property of critter-like generalized

ω-creatures. If there are exist three components C_1, C_2, C_3 such that $\varphi^{-1}(u)$ has a neighbor in each of them then, by the second property, $\{C_1, C_2\}$, $\{C_2, C_3\}$ and $\{C_1, C_3\}$ are all edges of $\mathcal{C}_B$. But this contradicts the first property, namely that $\mathcal{C}_B$ is a path. ■

Lemma 7.5.59. *Let $W = (G, H, \varphi, S_1, S_2)$ be a B-critter-like disjoint generalized ω-creature of adhesion size α, and u be a peripheral vertex of H. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the result of dissolving u in W. Then W' is a B-critter-like disjoint generalized $(\omega-1)$-creature of adhesion size α. Furthermore, if W is A-critter-like then W' is A-critter-like.*

Proof: By Lemma 7.5.16 W' is a disjoint generalized $(\omega - 1)$-creature of adhesion size α. It remains to show that it is B-critter-like. Let $(\mathcal{C}_B, \lambda)$ and $(\mathcal{C}'_B, \lambda')$ be the labeled B-connectivity graphs of W and W' respectively. Let $C_1, C_2, \ldots, C_t$ be the components of $G[B_\varphi]$ in the order they are visited by the path $\mathcal{C}_B$. By Lemma 7.5.42, statement *(i)* there exists a sequence of (not necessarily distinct) components $C'_1, C'_2, \ldots, C'_t$ of $G'[B_{\varphi'}]$ such that $C_i \subseteq C'_i$ for every i. Lemma 7.5.42, statement *(ii)* every component of $G'[B_{\varphi'}]$ appears in the sequence $C'_1, C'_2, \ldots, C'_t$ at least once. By Lemma 7.5.58 there are two cases. Either there exists precisely one component C_r such that $\varphi^{-1}(u)$ has a neighbor in C_r, or there exists precisely one component C_r such that $\varphi^{-1}(u)$ has a neighbor in C_r and in C_{r+1}

We first consider the case that there exists precisely one component C_r such that $\varphi^{-1}(u)$ has a neighbor in C_r. Because u does not realize any edge of $\mathcal{C}_B$, Lemma 7.5.42, statement *(iii)* yields that $C'_i \neq C'_j$ for every pair of distinct integers i, j. Lemma 7.5.42, statement *(iv)* then yields that the edge set of $\mathcal{C}'_B$ is equal to $\{\{C'_i, C'_{i+1}\} \ : \ i < t\}$ and every $\{C'_i, C'_{i+1}\}$ satisfies that $\lambda'(\{C'_i, C'_{i+1}\}) = \lambda(\{C'_i, C'_{i+1}\})$. Thus $\mathcal{C}_B$ is path-like and $|\lambda'(\{C'_i, C'_{i+1})| \leqslant 1$ for every i By Lemma 7.5.42, statement *(v)* every peripheral vertex v of H' and every component C'_i satisfies that $\varphi'^{-1}(v)$ has a neighbor in C'_i if and only if $\varphi^{-1}(v)$ has a neighbor in C_i. Therefore, if $\varphi'^{-1}(v)$ has a neighbor in C'_i and in C'_j then

$\{C'_i, C'_j\}$ in an edge of $\mathcal{C}'_B$ and $\lambda'(\{C'_i, C'_j\}) = \{v\}$.

Next we consider the case that there exists precisely one r such that $\varphi^{-1}(u)$ has a neighbor in C_r and in C_{r+1} Because u realizes the edge $\{C_r, C_{r+1}\}$ in $\mathcal{C}_B$ and no other edges, Lemma 7.5.42, statement *(iii)* yields that $C'_r = C'_{r+1}$ and $C'_i \neq C'_j$ for every pair of distinct integers i, j such that $\{i, j\} \neq \{r, r+1\}$.

Lemma 7.5.42, statement *(iv)* then yields that the edge set of $\mathcal{C}'_B$ is equal to $\{\{C'_i, C'_{i+1}\} : i < t \text{ and } i \neq r\}$. Note here that the edge $\{C'_{r+1}, C'_{r+2}\}$, if $r+2 \leqslant t$, is equal to $\{C'_r, C'_{r+2}\}$, because $C_r = C'_{r+1}$. Thus $\mathcal{C}'_B$ is path-like. Furthermore Lemma 7.5.42, statement *(iv)* implies that $\lambda'(\{C'_i, C'_{i+1}\}) = \lambda(\{C'_i, C'_{i+1}\})$ for every $i \neq r$ (for $i = r$ we have $C'_i = C'_{i+1}$ and there is no self loop in $\mathcal{C}'_B$). Thus $|\lambda'(\{C'_i, C'_j\})| \leqslant 1$ for every edge $\{C'_i, C'_j\}$ of $\mathcal{C}'_B$.

By Lemma 7.5.42, statement *(v)* every peripheral vertex v of H' and every component C'_i with $i \notin \{r, r+1\}$ satisfies that $\varphi'^{-1}(v)$ has a neighbor in C'_i if and only if $\varphi^{-1}(v)$ has a neighbor in C_i. Furthermore Lemma 7.5.42, statement *(v)* yields that for every peripheral vertex v of H', $\varphi'^{-1}(v)$ has a neighbor in C'_r if and only if $\varphi^{-1}(v)$ has a neighbor in C_r or in C_{r+1}.

For the final property of B-critter-like generalized $(\omega-1)$-creatures, suppose that $\varphi'^{-1}(v)$ has a neighbor in two distinct components C'_i and in C'_j of $G'[B_{\varphi'}]$. Without loss of generality $j \notin \{r, r+1\}$. Then $\varphi^{-1}(v)$ has a neighbor in C_j. If $C'_i \neq C'_r$ then $\varphi^{-1}(v)$ has a neighbor in C_i in G. Then, since W is B-critter-like it follows that v realizes $\{C_i, C_j\}$ in W. But by Lemma 7.5.42, statement *(iv)* v realizes $\{C'_i, C'_j\}$ in W'. Suppose now that $C'_i \neq C'_r$. Then exists $q \in \{r, r+1\}$ such that $\varphi^{-1}(v)$ has a neighbor in C_q in G. Since $q \neq j$ and W is B-critter-like it follows that v realizes $\{C_q, C_j\}$ in W. But then Lemma 7.5.42, statement *(iv)* implies that v realizes $\{C'_r, C'_j\}$ in W', completing the proof.

The proof that if W is A-critter-like then W' is A-critter-like is symmetric. ■

Lemma 7.5.60. *Let $W = (G, H, \varphi, S_1, S_2)$ be a B-critter-like disjoint generalized ω-creature of adhesion size α, and C be an absorbable component of $G[A_\varphi]$. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the result of absorbing C in W. Then W' is a B-critter-like disjoint generalized ω-creature of adhesion size α.*

Proof: By Lemma 7.5.18 W' is a disjoint generalized $(\omega - 1)$-creature of adhesion size α. It remains to show that it is B-critter-like. Let $(\mathcal{C}_B, \lambda)$ and $(\mathcal{C}'_B, \lambda')$ be the labeled B-connectivity graphs of W and W' respectively. Let $C_1, C_2, \ldots, C_t$ be the components of $G[B_\varphi]$ in the order they are visited by the path $\mathcal{C}_B$. By Lemma 7.5.58 for every $i < t$ there exists a unique peripheral vertex v_i that realizes $\{C_i, C_{i+1}\}$ in W. Again by Lemma 7.5.58 $\varphi^{-1}(v_i)$ has no neighbors in any components of $G[B_\varphi]$ other than C_i, C_{i+1}, and therefore the vertices $v_1, \ldots, v_{t-1}$ are all distinct. Let $Q = \{v_1, \ldots, v_{t-1}\}$. By Lemma 7.5.58 every peripheral vertex u of H not in Q satisfies that $\varphi^{-1}(u)$ has neighbors in precisely one component of $G[B_\varphi]$.

By definition of absorbing, $B_{\varphi'} = B_\varphi$ so $C_1, C_2, \ldots, C_t$ are also the components of $G[B_\varphi]$. Since $C \in A_\varphi$, and A_φ and B_φ are anti-complete, it follows that for every peripheral vertex v of H and component C_i, $\varphi'^{-1}(v)$ has a neighbor in C_i if and only if $\varphi^{-1}(v)$ has a neighbor in C_i.

Consider an edge $\{C_i, C_j\}$ of $\mathcal{C}'_B$, and let v be the peripheral vertex that realizes this edge. Then $\varphi'^{-1}(v)$ has neighbors in C_i and C_j, and therefore $\varphi^{-1}(v)$ has neighbors in C_i and C_j. Since W is B-critter-like it follows that v realizes $\{C_i, C_j\}$ in $\mathcal{C}_B$.

Hence $\mathcal{C}'_B$ is a sub-graph of $\mathcal{C}_B$. However $\mathcal{C}_B$ is a path and $\mathcal{C}'_B$ is connected by Lemma 7.5.28, and therefore $\mathcal{C}'_B = \mathcal{C}_B$. Further, for every edge $\{C_i, C_{i+1}\}$ of $\mathcal{C}_B$ the peripheral vertex v_i is the only peripheral vertex of $H' = H$ such that $\varphi'^{-1}(v)$ has neighbors in $\{C_i, C_{i+1}\}$. Since every edge of $\mathcal{C}_B$ is realized by at least one peripheral vertex, $\{C_i, C_{i+1}\}$ is realized by v_i and no other peripheral vertices in W'.

But then W' is path-like, $|\lambda'(\{C_i, C_{i+1}\})| \leqslant 1$ for every i and every peripheral vertex v such that $\varphi'^{-1}(v)$ has neighbors in two distinct components $\{C_i, C_j\}$ realizes the pair $\{C_i, C_j\}$ in W'. Hence W' is B-critter-like. ■

Lemma 7.5.61. *Let $W = (G, H, \varphi, S_1, S_2)$ be a B-critter-like disjoint generalized ω-creature of adhesion size α, and C be an erasable component of $G[A_\varphi]$. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the result of erasing C in W. Then W' is a B-critter-like disjoint generalized ω'-creature of adhesion size α.*

Proof: By Lemma 7.5.34 W' is a disjoint generalized ω'-creature of adhesion size α. It remains to show that it is B-critter-like. Let $(\mathcal{C}_B, \lambda)$ and $(\mathcal{C}'_B, \lambda')$ be the labeled B-connectivity graphs of W and W' respectively. Let $C_1, C_2, \ldots, C_t$ be the components of $G[B_\varphi]$ in the order they are visited by the path $\mathcal{C}_B$. By Lemma 7.5.58 for every $i < t$ there exists a unique peripheral vertex v_i that realizes $\{C_i, C_{i+1}\}$ in W. Again by Lemma 7.5.58 $\varphi^{-1}(v_i)$ has no neighbors in any components of $G[B_\varphi]$ other than C_i, C_{i+1}, and therefore the vertices $v_1, \ldots, v_{t-1}$ are all distinct. Let $Q = \{v_1, \ldots, v_{t-1}\}$. By Lemma 7.5.58 every peripheral vertex u of H not in Q satisfies that $\varphi^{-1}(u)$ has neighbors in precisely one component of $G[B_\varphi]$. We conclude that $G - A_\varphi$ satisfies the following property: every path from C_i to C_j (with $i \leqslant j$) in $G - A_\varphi$ intersects all components $C_i, C_{i+1}, \ldots, C_j$.

We will make use of the following properties of the erase operation, all of which are easily observed directly from the definition of erase.

(i) $B_\varphi \subseteq B_{\varphi'} \subseteq V(G) - A_\varphi$,

(ii) Every connected component of $G'[B_{\varphi'} - B_\varphi]$ has a neighbor in B_φ

(iii) Every peripheral vertex v of H' satisfies $\varphi'^{-1}(v) \subseteq \varphi^{-1}(v)$

By *(i)* every connected component of $G[B_\varphi]$ is contained in a connected component of $G'[B_{\varphi'}]$. By *(ii)* every connected component of $G'[B_{\varphi'}]$ contains at least one connected component of $G[B_{\varphi'}]$.

We claim that if a peripheral vertex v of H' satisfies that $\varphi'^{-1}(v)$ has a neighbor in a component C' of $G'[B_{\varphi'}]$, then C' contains a component C_i of $G[B_\varphi]$ such that $\varphi^{-1}(v)$ has a neighbor in C_i. Indeed, suppose that $\varphi'^{-1}(v)$ has a neighbor x in a component C' of $G'[B_{\varphi'}]$. Then, by *(iii)* $x \in B_\varphi$ or $x \in \varphi^{-1}(v)$. If $x \in B_\varphi$ then C' contains the component C_i of $G[B_\varphi]$ that contains x, and $\varphi^{-1}(v)$ has a neighbor in C_i, namely x. If $x \in \varphi^{-1}(v)$, let Z be the component of $C' - B_\varphi$ that contains x. Since $x \in \varphi^{-1}(v)$ and Z is disjoint from $A_\varphi \cup B_\varphi$ it follows that $Z \subseteq \varphi^{-1}(v)$. By *(ii)*, Z has a neighbor y in B_φ. Then C' contains the component C_i of $G[B_\varphi]$ that contains y, and $\varphi^{-1}(v)$ has a neighbor in C_i, namely y.

Let C' be a component of $G'[B_{\varphi'}]$. If C' contains C_i and C_j it also contains a path from C_i to C_j in $G - A_\varphi$. But this path intersects all components $C_i, C_{i+1}, \ldots, C_j$, and therefore

$$C_i \cup C_{i+1} \cup \ldots \cup C_j \subseteq C'.$$

Thus there exists an ordering of the components of $G'[B_{\varphi'}]$ into $C'_1, C'_2, \ldots C'_{t'}$ and sequence $0 = j_0 < j_1 < j_2 < \ldots j_{t'} = t$ of integers such that for every $i \in \{1, \ldots, t\}$ it holds that

$$C_{j_{i-1}+1} \cup C_{j_{i-1}+2}, \ldots \cup C_{j_i} \subseteq C'_i$$

Consider now a peripheral vertex v of H' and suppose that $\varphi'^{-1}(v)$ has neighbors both in C'_i and C'_j. Without loss of generality $i < j$. Then C'_i contains a component $C_{i'}$ of $G[B_\varphi]$ and C'_j contains a component $C_{j'}$ of $G[B_\varphi]$ such that $\varphi^{-1}(v)$ has a neighbor in $C_{i'}$ and in $C_{j'}$. But then $i' = j_{i'}$, $j' = j_{i'} + 1$ and $v = v_{j_{i'}}$. Hence $j = i + 1$. Thus, every edge of the B-connectivity graph $\mathcal{C}'_B$ of W' goes from a component C'_i to a component C'_{i+1}, and if the edge $\{C'_i, C'_{i+1}\}$ is present then $\lambda'(\{C'_i, C'_{i+1}\}) = v_{j_{i'}}$. By Lemma 7.5.28 $\mathcal{C}'_B$ is connected and therefore every pair $\{C'_i, C'_{i+1}\}$ is an edge of $\mathcal{C}'_B$.

Hence we have proved that if, for a peripheral vertex v of H', $\varphi'^{-1}(v)$ has neighbors

both in C'_i and C'_j, then $j = i+1$ and $\{C'_i, C'_{i+1}\}$ is an edge of $\mathcal{C}'_B$ (so v realizes $\{C'_i, C'_j\}$ and W' is B-path-like), and $v = v_{j_{i'}}$ (so $|\lambda'(\{C'_i, C'_{i+1}\})| = 1$). This concludes the proof. ■

Lemma 7.5.62. *Let $W = (G, H, \varphi, S_1, S_2)$ be a B-critter-like disjoint generalized ω-creature of adhesion size α, and $W' = (G', H', \varphi', S'_1, S'_2)$ be an A-descendant of W. Then W' is a B-critter-like disjoint generalized ω'-creature of adhesion size α.*

Proof: Since $W' = (G', H', \varphi', S'_1, S'_2)$ is an A-descendant of W, W' is obtained from G by a sequence of dissolving peripheral vertices, absorbing absorbable components of $G[A_\varphi]$, and erasing components of $G[A_\varphi]$. By Lemma 7.5.59, Lemma 7.5.60, and Lemma 7.5.61, dissolving peripheral vertices, absorbing absorbable components of $G[A_\varphi]$, and erasing components of $G[A_\varphi]$ in a a B-critter-like disjoint generalized ω-creature of adhesion size α results in a B-critter-like disjoint generalized ω'-creature of adhesion size α. The statement of the lemma now follows by induction on the number of operations in the sequence used to obtain W' from W. ■

7.5.14 Making the Generalized Creature Critter-Like on Both Sides

Lemma 7.5.63. *Let G be a k-creature free graph and $W = (G, H, \varphi, S_1, S_2)$ be an A-critter-like disjoint generalized ω-creature with adhesion size α. Then there exists a B-critter-like disjoint generalized ω-creature $W' = (G, H, \varphi', S_1, S_2)$ with adhesion size α.*

Proof: We set

$$\varphi'(v) = \begin{cases} c_B \text{ if } \varphi(v) = c_A \\ c_A \text{ if } \varphi(v) = c_B \\ \varphi(v) \text{ otherwise.} \end{cases}$$

Then $A_{\varphi'} = B_\varphi$ and $B_{\varphi'} = A_\varphi$. Since A and B are interchangeable in the definitions of (disjoint) generalized ω-creatures, the A- and B-connectivity graphs $(\mathcal{C}_A, \lambda_A)$ and $(\mathcal{C}_B, \lambda_B)$, A/B-critter-like and adhesion size, W' is a B-critter-like disjoint generalized ω-creature with adhesion size α. ■

We are now ready to extract a generalized ω-creature that is both A-critter-like and B-critter-like.

Lemma 7.5.64. *Let $k \geqslant 2$ and G be k-creature free, and $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω creature of adhesion size α. Then there exists an induced subgraph G' of G and an A-critter-like and B-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$, where $\omega' \geqslant \frac{(\log_{k\alpha} \log_{k\alpha}(\omega))^{1/\alpha}}{16k^2\alpha^3} - 12$.*

Proof: By Lemma 7.5.57 there exists an A-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$ with adhesion size α, such that $\omega' \geqslant \frac{(\log_{k\alpha}(\omega)-3)^{1/\alpha}}{8k^2\alpha^3} - 1$. Furthermore W' is an A-descendant of W, so G' is an induced subgraph of G. By Lemma 7.5.63 there exists a B-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi^\star, S'_1, S'_2)$ with adhesion size α. By Lemma 7.5.57 applied to W' there exists an A-critter-like disjoint generalized ω''-creature $W'' = (G'', H'', \varphi'', S''_1, S''_2)$ with adhesion size α, such that

$$\begin{aligned}
\omega'' &\geqslant \frac{(\log_{k\alpha}(\omega') - 3)^{1/\alpha}}{8k^2\alpha^3} - 1 \\
&\geqslant \frac{(\log_{k\alpha}\left(\frac{(\log_{k\alpha}(\omega)-3)^{1/\alpha}}{8k^2\alpha^3} - 1\right) - 3)^{1/\alpha}}{8k^2\alpha^3} \\
&\geqslant \frac{(\log_{k\alpha} \log_{k\alpha}(\omega))^{1/\alpha}}{16k^2\alpha^3} - 12
\end{aligned}$$

Furthermore W'' is an A-descendant of W', so G'' is an induced subgraph of G', and therefore of G. Since W'' is an A-descendant of W' and W' is B-critter-like, by Lemma 7.5.62 W'' is B-critter-like. This concludes the proof. ■

7.5.15 Coordinating The Orderings

We are almost done, but not quite. In particular the A and B side of W are now both critter-like, but the peripheral vertices may come in different order on the two sides. We fix this by finding a large common sub-sequence using the famous Erdös-Szekers theorem [53], and dissolve all of the peripheral vertices that are not in the sequence. By Lemma 7.5.59 this maintains the critter-like property both on the A and the B side. However we do still need to prove that when we dissolve the peripheral vertices that are out of order, we don't re-order the ones that are in order.

$W = (G, H, \varphi, S_1, S_2)$ be an A-critter-like a disjoint generalized ω creature of adhesion size α.

An ordering $v_1, v_2, \ldots, v_\omega$ of the peripheral vertices of H is an *A-critter ordering* if the following condition is satisfied: if $\varphi^{-1}(v_i)$ and $\varphi^{-1}(v_j)$ both have a neighbor in a component C of $G[A_\varphi]$, then all $v_r \in \{v_i, v_{i+1} \ldots v_j\}$ also satisfy that $\varphi^{-1}(v_r)$ has a neighbor in C. A *B-critter ordering* is defined similarly, using components C of $G[B_\varphi]$.

Lemma 7.5.65. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω creature. If W is A-critter-like then W has an A-critter ordering. If W is B-critter-like then W has a B-critter ordering.*

Proof: Suppose W is A-critter-like. Let $(\mathcal{C}_A, \lambda)$ be the labeled A-connectivity graph of W. Let $C_1, C_2, \ldots C_t$ be the order in which the components of $G[A_\varphi]$ appear on the path $\mathcal{C}_A$.

By Lemma 7.5.58 every peripheral vertex v of H satisfies that either there is a unique C_i such that $\varphi^{-1}(u)$ has a neighbor in C_i and in no other components, or there is a unique C_i such that $\varphi^{-1}(u)$ has a neighbor in C_i and C_{i+1} and in no other components. In the second case no other peripheral vertex v also satisfies that $\varphi^{-1}(v)$ has a neighbor both in C_i and C_{i+1}.

Consider the following ordering: first take all peripheral vertices u such that C_1 is the only component such that $\varphi^{-1}(u)$ has a neighbor in it, then take the unique peripheral vertex u such that $\varphi^{-1}(u)$ has a neighbor in C_1 and C_2, then take all peripheral vertices u such that C_2 is the only component such that $\varphi^{-1}(u)$ has a neighbor in it, and so on. It is easily verified that this ordering is an A-critter ordering. The proof that if W is B-critter-like then W has a B-critter ordering is symmetric. ■

Let X be a set and $\sigma = x_1, x_2, \ldots x_t$ be an ordering of X. Let Y be a subset of X. Then the sub-ordering of σ induced by Y is denoted by $\sigma[Y]$ and is the ordering of Y in which all elements of Y come in the same order as in σ.

Lemma 7.5.66. *Let $W = (G, H, \varphi, S_1, S_2)$ be an A-critter-like disjoint generalized ω creature and u be a peripheral vertex in H. Let σ be an A-critter ordering of W. Let $W' = (G, H, \varphi, S_1, S_2)$ be the A-critter-like disjoint generalized ω creature resulting from dissolving u. Then $\sigma[V(H') - \{c_A, c_B\}]$ is an A-critter ordering of W'. Similarly, if W is B-critter-like and σ is a B-critter ordering of W then $\sigma[V(H') - \{c_A, c_B\}]$ is a B-critter ordering of W'.*

Proof: By Lemma 7.5.59 W' is an A-critter-like generalized $(\omega - 1)$-creature. Let $(\mathcal{C}_A, \lambda)$ and $(\mathcal{C}'_A, \lambda')$ be the labeled A-connectivity graphs of W and W' respectively. Let $C_1, C_2, \ldots, C_t$ be the components of $G[A_\varphi]$ in the order they are visited by the path $\mathcal{C}_A$. By Lemma 7.5.42, statement *(i)* there exists a sequence of (not necessarily distinct) components $C'_1, C'_2, \ldots, C'_t$ of $G'[A_{\varphi'}]$ such that $C_i \subseteq C'_i$ for every i. Lemma 7.5.42, statement *(ii)* every component of $G'[A_{\varphi'}]$ appears in the sequence $C'_1, C'_2, \ldots, C'_t$ at least once. By Lemma 7.5.58 there are two cases. Either there exists precisely one component C_r such that $\varphi^{-1}(u)$ has a neighbor in C_r, or there exists precisely one component C_r such that $\varphi^{-1}(u)$ has a neighbor in C_r and in C_{r+1}

We first consider the case that there exists precisely one component C_r such that

$\varphi^{-1}(u)$ has a neighbor in C_r. Because u does not realize any edge of $\mathcal{C}_B$, Lemma 7.5.42, statement *(iii)* yields that $C'_i \neq C'_j$ for every pair of distinct integers i, j. By Lemma 7.5.42, statement *(v)* every peripheral vertex v of H' and every component C'_i satisfies that $\varphi'^{-1}(v)$ has a neighbor in C'_i if and only if $\varphi^{-1}(v)$ has a neighbor in C_i. Therefore $\sigma[V(H) - \{u\}]$ is an A-critter ordering of W.

Next we consider the case that there exists precisely one r such that $\varphi^{-1}(u)$ has a neighbor in C_r and in C_{r+1} Because u realizes the edge $\{C_r, C_{r+1}\}$ in $\mathcal{C}_B$ and no other edges, Lemma 7.5.42, statement *(iii)* yields that $C'_r = C'_{r+1}$ and $C'_i \neq C'_j$ for every pair of distinct integers i, j such that $\{i, j\} \neq \{r, r+1\}$.

By Lemma 7.5.42, statement *(v)* every peripheral vertex v of H' and every component C'_i with $i \notin \{r, r+1\}$ satisfies that $\varphi'^{-1}(v)$ has a neighbor in C'_i if and only if $\varphi^{-1}(v)$ has a neighbor in C_i. Furthermore Lemma 7.5.42, statement *(v)* yields that for every peripheral vertex v of H', $\varphi'^{-1}(v)$ has a neighbor in C'_r if and only if $\varphi^{-1}(v)$ has a neighbor in C_r or in C_{r+1}. Therefore $\sigma[V(H) - \{u\}]$ is an A-critter ordering of W.

The proof that if W is B-critter-like and σ is a B-critter ordering of W then $\sigma[V(H') - \{c_A, c_B\}]$ is a B-critter ordering of W' is symmetric. ■

For an ordering $\sigma = x_1, x_2, \ldots, x_t$ of a set X the *reverse* ordering of σ is the ordering denoted by σ^R and defined as $\sigma^R = x_t, x_{t-1}, \ldots, x_1$.

Observation 7.5.67. *Let $W = (G, H, \varphi, S_1, S_2)$ be a generalized ω creature and σ be an A-critter ordering of W. Then σ^R is an A-critter ordering of W.*

Proof: Let $\sigma = v_1, \ldots v_\omega$ and $\sigma^R = v_t, \ldots v_1$. If $\varphi^{-1}(v_i)$ and $\varphi^{-1}(b)$ both have a neighbor in a component C of $G[A_\varphi]$, then all $v_r \in \{v_i, v_{i+1} \ldots v\}$ also satisfy that $\varphi^{-1}(v_r)$ has a neighbor in C. These are precisely the peripheral vertices between v_i and v_j both in the ordering σ and in the ordering σ^R. ■

Given two orderings $\sigma_A = x_1^A, x_2^A, \ldots, x_t^A$ and $\sigma_B = x_1^B, x_2^B, \ldots, x_t^B$ of a set X we say

that σ_A and σ_B *agree* on a subset Y of X if, for every pair of elements $y, y' \in Y$, if $y = y_i^A = y_{i'}^B$ and $y' = y_j^A = y_{j'}^B$ then $i < j$ if and only if $i' < j'$. The following theorem is a re-formulation of the well-known Erdös-Szekers Theorem [53] in terms of orderings.

Theorem 7.5.68 (Erdös-Szekers Theorem [53])**.** *For any two orderings σ_A, σ_B of a set U there exists a subset X of U such that $|X|^2 \geqslant |U|$ and either σ_A and σ_B agree on X or the reverse σ_A^R of σ_A agrees with σ_B on X.*

Lemma 7.5.69. *Let $W = (G, H, \varphi, S_1, S_2)$ be an A-critter-like and B-critter-like disjoint generalized ω-creature. Then there exist an A-critter-like and B-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S_1', S_2')$ and ordering σ of $V(H')$ such that G' is an induced subgraph of G, $\omega' \geqslant \sqrt{\omega}$ and σ is both an A-critter ordering and a B-critter ordering of W'.*

Proof: By Lemma 7.5.65 there exists an A-critter ordering $\sigma_A = v_1^A, v_2^A, \ldots v_\omega^A$ and a B-critter ordering $\sigma_B = v_1^B, v_2^B, \ldots v_\omega^B$ of W. Let σ_A^R be the reverse of σ_A. By Observation 7.5.67 σ_A^R is also an A-critter order of W. By Theorem 7.5.68 there exists a subset X of the peripheral vertices such that $|X| \geqslant \omega$ and either σ_A or σ_A^R agree with σ_B on X. Without loss of generality σ_A agrees with σ_B on X (otherwise we can simply exchange σ_A with σ_A^R). Let $\omega' = |X|$, we have that $\omega' \geqslant \sqrt{\omega}$. Let W' be the generalized ω'-creature resulting from dissolving all peripheral vertices of H not in X. By Lemma 7.5.59, W' is a disjoint A-critter-like and B-critter-like generalized ω-creature. By Lemma 7.5.66, $\sigma_A[X]$ is an A-critter-order of W' and $\sigma_B[X]$ is a B-critter-order of W'. But σ_A and σ_B agree on X, so $\sigma_A[X] = \sigma_B[X]$. ■

Lemma 7.5.70. *Let $k \geqslant 2$ and G be k-creature free, and $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω creature of adhesion size α. Then there exists an induced subgraph G' of G and an A-critter-like and B-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S_1', S_2')$, where $\omega' \geqslant \frac{(\log_{k\alpha} \log_{k\alpha}(\omega))^{1/2\alpha}}{4k\alpha^2} - 12$.*

Furthermore there exists an ordering σ of the peripheral vertices such that σ is both an A-critter ordering and a B-critter ordering of W.

Proof: By Lemma 7.5.64 there exists an induced subgraph G' of G and an A-critter-like and B-critter-like disjoint generalized ω'-creature $W' = (G', H', \varphi', S'_1, S'_2)$, where $\omega' \geqslant \frac{(\log_{k\alpha} \log_{k\alpha}(\omega))^{1/\alpha}}{16k^2\alpha^3} - 12$. By Lemma 7.5.69 applied to W' there exist an A-critter-like and B-critter-like disjoint generalized ω''-creature $W'' = (G'', H'', \varphi'', S''_1, S''_2)$ and ordering σ of $V(H'')$ such that G'' is an induced subgraph of G' (and therefore of G),

$$\omega'' \geqslant \sqrt{\omega'} \geqslant \frac{(\log_{k\alpha} \log_{k\alpha}(\omega))^{1/2\alpha}}{4k\alpha^2} - 12,$$

and σ is both an A-critter ordering and a B-critter ordering of W''. Hence W'' satisfies the conclusion of the lemma. ■

7.5.16 Extracting a Critter

The conclusion of Lemma 7.5.69 is almost sufficient for us to directly extract an ω-critter from it. There is still one complication, namely peripheral vertices that on at least one side do not realize any edge of the connectivity graph.

Lemma 7.5.71. *Let G be a k-creature free graph, and let $W = (G, H, \varphi, S_1, S_2)$ be a disjoint generalized ω-creature Let Z_A be the set of all peripheral vertices of H such that there exists a component C_A of $G[A_\varphi]$ such that $N(\varphi^{-1}(v)) \cap A_\varphi \subseteq C_A$, and Z_B be the set of all peripheral vertices of H such that there exists a component C_B of $G[B_\varphi]$ such that $N(\varphi^{-1}(v)) \cap B_\varphi \subseteq C_B$. Then $|Z_A| < k$ and $|Z_B| < k$.*

Proof: We prove that $|Z_A| < k$. Suppose for contradiction that $|Z_A| \geqslant k$. Let $W' = (G', H', \varphi', S'_1, S'_2)$ be the result of dissolving all peripheral vertices not in Z_A. Then, by Lemma 7.5.16 W' is a disjoint generalized ω'-creature, where $\omega' = |Z_A| \geqslant k$,

and G' is an induced subgraph of G. By Lemma 7.5.28 the connectivity graph $\mathcal{C}_A$ of W is connected. Further no vertex of Z_A realizes an edge of $\mathcal{C}_A$. Therefore, by Lemma 7.5.43, statement *(iii)*, there exists a component C of $G[A'_\varphi]$ such that $A_\varphi \subseteq C$. But then, by Lemma 7.5.43, statement *(ii)*, $G[A'_\varphi] = C$. This yields a contradiction. On the one hand, Lemma 7.5.4 shows that each vertex $v \in Z_A$ satisfies that $\varphi'^{-1}(v)$ has a neighbor in $A_{\varphi'}$. On the other hand, G' is k-creature free and W' is a disjoint generalized ω-creature, and so by Lemma 7.5.29, there can not be k peripheral vertices v such that $\varphi'^{-1}(v)$ has a neighbor in $C' = A_{\varphi'}$. This contradicts the assumption that $|Z_A| \geqslant k$. The proof of the upper bound for $|Z_B|$ is symmetric. ■

With Lemma 7.5.71 in hand we are ready to extract a critter!

Lemma 7.5.72. *Let G be a k-creature free graph and $W = (G, H, \varphi, S_1, S_2)$ be an A-critter-like and B-critter-like disjoint generalized ω-creature, such that there exists an ordering σ of the peripheral vertices such that σ is both an A-critter ordering and a B-critter ordering of W. Then there exists an induced subgraph G' in G, such that G' is an ω'-critter, where $\omega' \geqslant \frac{\omega-2k}{2k+1}$.*

Proof: Let $(\mathcal{C}_A, \lambda_A)$ and $(\mathcal{C}_B, \lambda_B)$ be the labeled A-connectivity graph and labeled B-connectivity graph of W respectively. Let $A_1, A_2, \ldots A_p$ be the components of $G[A_\varphi]$ in the order that the appear in on the path $\mathcal{C}_A$, and $B_1, B_2, \ldots B_q$ be the components of $G[B_\varphi]$ in the order that the appear in on the path $\mathcal{C}_B$. Let $v_1, \ldots, v_\omega$ be an ordering of the peripheral vertices such that σ is both an A-critter ordering and a B-critter ordering of W. Let X_A be the subset of peripheral vertices such that every $v_i \in X_A$ realizes a pair $\{A_j, A_{j+1}\}$. Select integers $i_1 < i_2 < \ldots < i_{p-1}$ such that $X_A = \{v_{i_1}, v_{i_2}, \ldots v_{i_{p-1}}\}$.

We claim that either, for every $j \in \{1, \ldots, p-1\}$ we have that v_{i_j} realizes $\{A_j, A_{j+1}\}$, or for every $j \in \{1, \ldots, p-1\}$ we have that v_{i_j} realizes $\{A_{p-j}, A_{p+1-1}\}$ Suppose v_{i_a} realizes $\{A_j, A_{j+1}\}$ and v_{i_b} realizes $\{A_{j+1}, A_{j+2}\}$. If $|a-b| > 1$ then there exists a v_{i_c} in

X_A between v_{i_a} and v_{i_b} in the ordering σ. Since σ is an A-critter ordering $\varphi^{-1}(v_{i_c})$ has a neighbor in A_{j+1}. But then $\varphi^{-1}(v_{i_a})$, $\varphi^{-1}(v_{i_b})$ and $\varphi^{-1}(v_{i_c})$ all have a neighbor in A_{j+1}. But $\{\varphi^{-1}(v_{i_a}), \varphi^{-1}(v_{i_b}), \varphi^{-1}(v_{i_c})\} \subseteq X_A\}$ so each of them realize an edge of $\mathcal{C}_A$. But then each of these edges is incident to A_{j+1}, contradicting that W is A-critter-like. We conclude that if v_{i_a} realizes $\{A_j, A_{j+1}\}$ and v_{i_b} realizes $\{A_{j+1}, A_{j+2}\}$ then $|a-b| \leqslant 1$. In other words, the vertices of X_A that realize consecutive edges of $\mathcal{C}_A$ come consecutively in the ordering $v_{i_1}, v_{i_2}, \ldots v_{i_{p-1}}$. Hence for every $j \in \{1, \ldots, p-1\}$ we have that v_{i_j} realizes $\{A_j, A_{j+1}\}$, or for every $j \in \{1, \ldots, p-1\}$ we have that v_{i_j} realizes $\{A_{p-j}, A_{p+1-1}\}$, as claimed.

Without loss of generality the first of these two cases holds. In the second case we may re-name for every i the component A_i to A_{p+1-i} and vice versa (this corresponds to traversing the path $\mathcal{C}_A$ from right to left, rather than from left to right). After re-naming for every $j \in \{1, \ldots, p-1\}$ we have that v_{i_j} realizes $\{A_j, A_{j+1}\}$. Note that this does not affect that σ is an A-critter ordering, because the definition of A-critter orderings is independent of orderings of the components of A. Let X_B be the subset of peripheral vertices such that every $v_i \in X_B$ realizes a pair $\{B_j, B_{j+1}\}$. Select integers $i'_1 < i'_2 < \ldots < i'_{q-1}$ such that $X_B = \{v_{i'_1}, v_{i'_2}, \ldots v_{i_{p-1'}}\}$. An identical argument to the one for X_A shows that without loss of generality for every $j \in \{1, \ldots, q-1\}$ we have that $v_{i'_j}$ realizes $\{B_j, B_{j+1}\}$,

Let Z_A be the set of all peripheral vertices of H such that there exists a component C_A of $G[A_\varphi]$ such that $N(\varphi^{-1}(v)) \cap A_\varphi \subseteq C_A$, and Z_B be the set of all peripheral vertices of H such that there exists a component C_B of $G[B_\varphi]$ such that $N(\varphi^{-1}(v)) \cap B_\varphi \subseteq C_B$. Then, by Lemma 7.5.71 we have $|Z_A| < k$ and $|Z_B| < k$.

Note that all peripheral vertices that are not in $Z_A \cup Z_B$ realize an edge in $\mathcal{C}_A$ and an edge in $\mathcal{C}_B$ and therefore are in $X_A \cap X_B$. Since $|Z_A \cup Z_B| < 2k$ it follows that there exists a consecutive sub-sequence $v_\ell, v_{\ell+1}, \ldots, v_\rho$ of at least $\frac{\omega-2k}{2k+1}$ vertices of $X_A \cap X_B$.

Further, for every $j \leqslant p-1$ we have that v_{i_j} realizes $\{A_j, A_{j+1}\}$, and for every $j \leqslant q-1$ we have that $v_{i'_j}$ realizes $\{B_j, B_{j+1}\}$. Thus there exist integers a and b such that for every $1 \leqslant c \leqslant \rho + 1 - \ell$ it holds that $v_{\ell+c-1}$ realizes $\{A_{a+c-1}, A_{a+c}\}$ and $\{B_{b+c-1}, B_{b+c}\}$. Said more plainly the peripheral vertices $v_\ell, v_{\ell+1}, \ldots, v_\rho$ realize the subpath $A_a, \ldots A_{a+\rho+1-\ell}$ of $\mathcal{C}_A$ and the subpath $B_b, \ldots B_{b+\rho+1-\ell}$ of $\mathcal{C}_B$.

We now construct a critter. We set $\hat{t} = \rho + 1 - \ell$ and for every $1 \leqslant i \leqslant \hat{t}+1$ we set $\hat{A}_i = A_{a+i-1}$ and $\hat{B}_i = B_{b+i-1}$. For every $1 \leqslant i \leqslant \hat{t}$ we set $\hat{X}_i = \varphi^{-1}(v_{\ell+i-1})$. We now verify that $\hat{A}_1, \ldots \hat{A}_{\hat{t}+1}$, $\hat{B}_1, \ldots \hat{B}_{\hat{t}+1}$, $\hat{X}_1, \ldots \hat{X}_{\hat{t}}$ satisfy the properties of a $\hat{t}$-critter. For property *(i)*, $\hat{A}_i$ is anticomplete with $\hat{A}_j$ for $j \neq i$ because $\hat{A}_i$ and $\hat{A}_j$ are distinct components of $G[A_\varphi]$. Similarly $\hat{B}_i$ is anticomplete with $\hat{B}_j$ for $i \neq j$. Finally, $\hat{A}_i$ is anticomplete with $\hat{B}_j$ (even for the case $i = j$) because A_φ is anticomplete with B_φ. For property *(ii)* every set $\hat{A}_i$ and $\hat{B}_i$ is connected in G because they are components of $G[A_\varphi]$ and $G[B_\varphi]$ respectively.

For property *(iii)* we observe that for every $1 \leqslant i \leqslant t$, $\hat{X}_i = \varphi^{-1}(v_{\ell+i-1})$, and ($v_{\ell+i-1}$ realizes the edges

$$\{A_{a+i-1}, A_{a+i}\} = \{\hat{A}_i, \hat{A}_{i+1}\} \text{ and}$$

$$\{B_{b+i-1}, B_{b+i}\} = \{\hat{B}_i, \hat{B}_{i+1}\}$$

in $\mathcal{C}_A$ and $\mathcal{C}_B$, respectively. Since W is A-critter-like and B-critter-like it follows that $N(\hat{X}_i) \subseteq \hat{A}_i \cup \hat{A}_{i+1} \cup \hat{B}_i \cup \hat{B}_{i+1}$.

For property *(iv)*, let $S_1^\star$, $S_2^\star$ be witness separators for W. By property *(i)* of generalized ω-creatures we have that $S_1^\star \cap \hat{X}_i$ and $S_1^\star \cap \hat{X}_i$ are distinct minimal A_φ, B_φ-separators in $G[A_\varphi \cup B_\varphi \cup \hat{X}_i]$. Since $N_G(\hat{X}_i) \subseteq \hat{A}_i \cup \hat{A}_{i+1} \cup \hat{B}_i \cup \hat{B}_{i+1}$ it follows that $S_1^\star \cap \hat{X}_i$ and $S_1^\star \cap \hat{X}_i$ are distinct minimal $(\hat{A}_i \cup \hat{A}_{i+1})$, $(\hat{B}_i \cup \hat{B}_{i+1})$-separators in $G[\hat{X}_i \cup \hat{A}_i \cup \hat{A}_{i+1} \cup \hat{B}_i \cup \hat{B}_{i+1}]$.

For the last part of property *(iv)* note that that $\hat{X}_i = \varphi^{-1}(v_{\ell+i-1})$ and that $v_{\ell+i-1}$ realizes the pairs $\{\hat{A}_i, \hat{A}_{i+1}\}$ and $\{\hat{B}_i, \hat{B}_{i+1}\}$. Thus (from the definition of A-connectivity

graphs and B-connectivity graphs) there is a path from $\hat{A}_i$ to $\hat{A}_{i+1}$ through $\hat{X}_i - S_1$ and from $\hat{B}_i$ to $\hat{B}_{i+1}$ through $\hat{X}_i - S_1$. By property *(iii)* of generalized ω-creatures there is also a path from $\hat{A}_i$ to $\hat{A}_{i+1}$ through $\hat{X}_i - S_2$ and from $\hat{B}_i$ to $\hat{B}_{i+1}$ through $\hat{X}_i - S_2$. Since $S_1^\star \subseteq S_1$, and $S_2^\star \subseteq S_2$ it follows that there exist paths from $\hat{A}_i$ to $\hat{A}_{i+1}$ both through $\hat{X}_i - S_1^\star$ and through $\hat{X}_i - S_2^\star$ and from $\hat{B}_i$ to $\hat{B}_{i+1}$ both through $\hat{X}_i - S_1^\star$ and through $\hat{X}_i - S_2^\star$.

We conclude that $\hat{A}_1, \dots \hat{A}_{\hat{t}+1}$, $\hat{B}_1, \dots \hat{B}_{\hat{t}+1}$, $\hat{X}_1, \dots \hat{X}_{\hat{t}}$ is a $\hat{t}$-critter and $\hat{t} \geqslant \frac{\omega-2k}{2k+1}$ ■

We are now ready to prove Lemma 7.3.3

Lemma 7.3.3 *Let $k \geqslant 2$ and G be a k-creature free graph, and $W = (G, H, \varphi, S_1, S_2)$ be a connected, good, full generalized ω-creature. Then there exists an induced subgraph G' of G which is a t-critter for $t \geqslant \frac{(\log\log(\omega))^{1/4k}}{96k^4} - 4$.*

Proof: Since W is a connected, good full generalized ω-creature, by Lemma 7.5.20 there exists an induced subgraph G' of G and a good, full, connected generalized ω'-creature W' with adhesion size $\alpha \leqslant 2k$ and $\omega' \geqslant \omega/2$.

Since W' is a full generalized ω'-creature with adhesion size $2k$, by Lemma 7.5.24 applied to W' there exists a full disjoint generalized ω''-creature, $W'' = (G'', H'', \varphi'', S_1'', S_2'')$, of adhesion size $2k$ such that G'' is an induced subgraph of G' (and therefore of G), and $\omega'' = \omega' \geqslant \omega/2k$.

Since G'' is k-creature free and W'' is a disjoint generalized ω''-creature with adhesion size $2k$, by Lemma 7.5.70 applied to W'' there exists an induced subgraph G''' of G'' and an A-critter-like and B-critter-like disjoint generalized ω'''-creature $W''' = (G''', H''', \varphi''', S_1''', S_2''')$, where $\omega''' \geqslant \frac{(\log_{2k^2}\log_{2k^2}(\omega/2k))^{1/4k}}{16k^3} - 12$. Furthermore there exists an ordering σ of the peripheral vertices such that σ is both an A-critter ordering and a B-critter ordering of W'''.

Therefore, by Lemma 7.5.72 there exists an induced subgraph $G^\star$ of G''' (and therefore

of G) that is a t-critter for

$$t \geqslant \frac{\omega''' - 2k}{2k+1} \geqslant \frac{(\log\log(\omega))^{1/4k}}{96k^4} - 4$$

This concludes the proof. ■

7.6 Families with Creatures or Critters are Feral

7.6.1 Boundaried Graphs, Monadic Second Order Logic, and Finite State

Towards the proof of Theorem 7.1.5 we first review the definitions of CMSO logic, boundaried graphs, gluing and finite state.

Definition 7.6.1. [**Graph Family**] A *graph family* is a set $\mathcal{F}$ of graphs.

Definition 7.6.2. [**Boundaried graph**] A *boundaried graph* is a graph G with a set $\delta(G) \subseteq V(G)$ of distinguished vertices called *boundary vertices*, and an injective labeling $\lambda_G : \delta(G) \to \mathbb{N}$. The set $\delta(G)$ is the *boundary* of G, and the *label set* of G is $\Lambda(G) = \{\lambda_G(v) \mid v \in \delta(G)\}$.

For ease of presentation, we sometimes abuse notation and treat equally-labeled vertices of different boundaried graphs, as well as the vertex that is the result of the identification of two such vertices, as the same vertex. Given a finite set $I \subseteq \mathbb{N}$, $\mathcal{G}_I$ denotes the class of all boundaried graphs whose label set is I, and $\mathcal{G}_{\subseteq I} = \bigcup_{I' \subseteq I} \mathcal{G}_{I'}$. A boundaried graph in $\mathcal{G}_{\subseteq [t]}$ is called a *t-boundaried* graph. Finally, $\mathcal{G}$ denotes the class of all boundaried graphs. The main operation employed to unite two boundaried graphs is the one that glues their boundary vertices together. Formally,

Definition 7.6.3. [**Gluing by** $\oplus$] Let G_1 and G_2 be two boundaried graphs. Then, $G_1 \oplus G_2$ is the (not-boundaried) graph obtained from the disjoint union of G_1 and G_2 by identifying equally-labeled vertices in $\delta(G_1)$ and $\delta(G_2)$.

Counting Monadic Second Order Logic The syntax of Monadic Second Order Logic (MSO) of graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Leftrightarrow$, $\Rightarrow$, variables for vertices, edges, sets of vertices and sets of edges, the quantifiers $\forall$ and $\exists$, which can be applied to these variables, and five binary relations:

1. $u \in U$, where u is a vertex variable and U is a vertex set variable;
2. $d \in D$, where d is an edge variable and D is an edge set variable;
3. $\mathbf{inc}(d, u)$, where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident to u;
4. $\mathbf{adj}(u, v)$, where u and v are vertex variables, and the interpretation is that u and v are adjacent;
5. equality of variables representing vertices, edges, vertex sets and edge sets.

Counting Monadic Second Order Logic (CMSO) extends MSO by including atomic sentences testing whether the cardinality of a set is equal to q modulo r, where q and r are integers such that $0 \leqslant q < r$ and $r \geqslant 2$. That is, CMSO is MSO with the following atomic sentence: $\mathbf{card}_{q,r}(S) = \mathbf{true}$ if and only if $|S| \equiv q \pmod{r}$, where S is a set. We refer to [128, 54, 129] for a detailed introduction to CMSO.

Definition 7.6.4. [**Family** $\mathcal{F}_\psi$] Given a CMSO-formula ψ, the family $\mathcal{F}_\psi$ is defined as the set of all graphs G such that $G \models \psi$.

For an example the formula

$$\psi = \exists X_1 \subseteq V(G)\ \exists X_2 \subseteq V(G)\ \Big[\ \big(\forall u \in V(G)\ u \in X_1 \vee u \in X_2\big) \wedge \forall u \in V(G)\ \forall v \in V(G) \big(\neg \mathbf{adj}(u,v) \vee (u \in X_1 \wedge v \in X_2) \vee (u \in X_2 \wedge v \in X_1)\big)\Big]$$

yields the family $\mathcal{F}_\psi$ of all bipartite graphs.

Definition 7.6.5. [**CMSO-definable family**] A family $\mathcal{F}$ is *CMSO-definable* if there exists a CMSO-formula ψ such that $\mathcal{F} = \mathcal{F}_\psi$. In this case, we say that ψ *defines* σ.

Finite State The goal of this subsection is to recall a variant of the classical Courcelle's Theorem [54, 130, 129] (see also [104]), which is a central component in the proof of our main result. This statement essentially says that the canonical equivalence relation over boundaried graphs defined below has finite index.

Definition 7.6.6. [**Canonical equivalence**] Given a graph family $\mathcal{F}$, the *canonical equivalence relation* $\equiv_{\mathcal{F}}$ on boundaried graphs is defined as follows. For two boundaried graphs G_α and G_β, we say that $G_\alpha \equiv_{\mathcal{F}} G_\beta$ if **(i)** $\Lambda(G_\alpha) = \Lambda(G_\beta)$ and **(ii)** for all boundaried graphs G_γ we have

$$G_\alpha \oplus G_\gamma \in \mathcal{F} \Leftrightarrow G_\beta \oplus G_\gamma \in \mathcal{F}$$

It is easy to verify that $\equiv_{\mathcal{F}}$ is indeed an equivalence relation. Given a family $\mathcal{F}$ of graphs and $I \subseteq \mathbb{N}$, we let $\mathcal{E}_{\equiv_\sigma}[\mathcal{G}_{\subseteq I}]$ denote the set of equivalence classes of $\equiv_{\mathcal{F}}$ when restricted to $\mathcal{G}_{\subseteq I}$.

Definition 7.6.7. [**Finite state**] A graph property σ has *finite state* if, for every $I \subseteq \mathbb{N}$, $\mathcal{E}_{\equiv_{\mathcal{F}}}[\mathcal{G}_{\subseteq I}]$ is finite.

Given a CMSO sentence ψ, the canonical equivalence relation associated with ψ is $\equiv_{\mathcal{F}_\psi}$, and for the sake of simplicity, we denote this relation by $\equiv_\psi$. We are now ready to state the variant of Courcelle's Theorem which was proven by Bodlaender et al. [131] (see also [54, 130, 129]) and which we use in this paper.

Theorem 7.6.8 ([131])**.** *Every CMSO-definable graph property has finite state.*

Remark 1. Theorem 7.6.8 is stated for graphs here, while it is stated and proved for more general structures by Bodlaender et al.[131]. Because we do not need the full power of the theorem of [131], and stating the theorem in its full generality requires an extra page of definitions we only state it here for the special case of graphs.

Remark 2. We would like to remark that neither the notion of "finite state" nor the statement of Theorem 7.6.8 should in any way be attributed to Bodlaender et al. [131].

The notion of finite state and a theorem very similar to the statement of Theorem 7.6.8 was stated and proved explicitly by Downey and Fellows [45]. For technical reasons the precise statement of the theorem(s) of Downey and Fellows [45] does not adequately suit our needs (or the needs of Bodlaender et al. [131]), nor is it obvious how to derive Theorem 7.6.8 as a corollary from the results of Downey and Fellows [45]. However the proof of Theorem 7.6.8 very closely follows proofs of analogous statements by Downey and Fellows [45].

The fact that every MSO-definable or CMSO-property on graphs has finite state is implicitly used, if (to the best of our knowledge) never explicitly stated, in every proof of (variants of) Courcelle's Theorem [132, 54, 130, 129].

7.6.2 Pumping Proof

Let G be a t-critter with t-critter partition $(A_1, \ldots, A_{t+1}, B_1, \ldots, B_{t+1}, X_1, \ldots X_t)$. The pair (G, W) is called a *witness-minimal t-critter* if there does not exist a proper

induced subgraph G' of G such that

$$W' = \{A_i \cap V(G') : i \leqslant t+1\}, \{B_i \cap V(G') : i \leqslant t+1\}, \{X_i \cap V(G') : i \leqslant t\}$$

is a t-critter partition of G'. A pair (S_1^i, S_2^i) of vertex subsets of X_i *satisfy* X_i if S_1^i, S_2^i satisfy property *(iv)* of t-critters for X_i.

Lemma 7.6.9. *Let G be a t-critter and $W = (A_1, \ldots, A_{t+1}, B_1, \ldots, B_{t+1}, X_1, \ldots X_t)$ be a t-critter partition of G such that (G, W) is a witness minimal t-critter. Then, for every $i \leqslant t$ and pair (S_1^i, S_2^i) that satisfy X_i we have $S_1^i \cap S_2^i = \emptyset$.*

Proof: Suppose for contradiction that $S_1^i \cap S_2^i$ contains a vertex v. We claim that

$$W' = (A_1, \ldots, A_{t+1}, B_1, \ldots, B_{t+1}, X_1, \ldots, X_i - \{v\}, X_{i+1}, \ldots, X_t)$$

is a t-critter partition of $G - v$. Since W satisfies properties *(i)*, *(ii)*, *(iii)* for G, we have that W' satisfies properties *(i)*, *(ii)*, *(iii)* for $G - v$. Thus it is sufficient to argue that $(S_1^i - \{v\}, S_2^i - \{v\})$ satisfy $X_i - \{v\}$ in $G - v$. Since S_1^i is a minimal $(A_i \cup A_{i+1})$,$(B_i \cup B_{i+1})$-separator in $G[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i]$ it follows that $S_1^i - \{v\}$ is a minimal $(A_i \cup A_{i+1})$,$(B_i \cup B_{i+1})$-separator in $(G-v)[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i - \{v\}]$. Further, since $(G-v)[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i - \{v\}] - (S_1^i - \{v\}) = G[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i] - S_1^i$ it follows that there is a path from A_i to A_{i+1} through $(X_i - \{v\}) - (S_1^i - \{v\})$ and from B_i to B_{i+1} through $(X_i - \{v\}) - (S_1^i - \{v\})$. A symmetric argument shows that $S_2^i - \{v\}$ is a minimal $(A_i \cup A_{i+1})$,$(B_i \cup B_{i+1})$-separator in $(G-v)[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i - \{v\}]$ and that there is a path from A_i to A_{i+1} through $(X_i - \{v\}) - (S_2^i - \{v\})$ and from B_i to B_{i+1} through $(X_i - \{v\}) - (S_1^2 - \{v\})$. But then $G - v$ is a t-critter, contradicting witness-minimality of (G, W). ■

An immediate corollary to Lemma 7.6.9 is that A and B can not have any common

neighbors.

Lemma 7.6.10. *Let G be a t-critter and $W = (A_1, \ldots, A_{t+1}, B_1, \ldots, B_{t+1}, X_1, \ldots X_t)$ be a t-critter partition of G such that (G, W) is a witness-minimal t-critter. Then, $N(A_1 \cup A_2 \ldots, A_{t+1}) \cap N(B_1 \cup B_2 \ldots, B_{t+1}) = \emptyset$.*

Proof: Suppose for contradiction that there exists a vertex v in the common neighborhood of $A_1 \cup A_2 \ldots, A_{t+1}$ and of $B_1 \cup B_2 \ldots, B_{t+1}$. Then $x \in X_i$ for some i, so $x \in N(A_i \cup A_{i+1}) \cap N(B_i \cup B_{i+1}) \cap X_i$. Let (S_1^i, S_2^i) satisfy X_i. Since both S_1^i, S_2^i separate $A_i \cup A_{i+1}$ from $B_i \cup B_{i+1}$ it follows that $x \in S_1^i$ and $x \in S_2^i$. This contradicts the conclusion of Lemma 7.6.9 that S_1^i and S_2^i are disjoint. ■

Lemma 7.6.11. *Let G be a t-critter and $W = (A_1, \ldots, A_{t+1}, B_1, \ldots, B_{t+1}, X_1, \ldots X_t)$ be a t-critter partition of G, such that (G, W) is a witness-minimal t-critter. Then, for every $i \leqslant t$ we have $|N(A_i) \cap X_i| \leqslant 3$, $|N(B_i) \cap X_i| \leqslant 3$, $|N(A_{i+1}) \cap X_i| \leqslant 3$, $|N(B_{i+1}) \cap X_i| \leqslant 3$.*

Proof: Let (S_1^i, S_2^i) be a pair that satisfies X_i. Since S_1^i, S_2^i are disjoint minimal $(A_i \cup A_{i+1})$, $(B_i \cup B_{i+1})$-separators in $G[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i]$ there exists a path P_{AB} from A to B through X_i. We select P_{AB} to be the shortest such path, in particular P_{AB} contains precisely one vertex in $N(A_i \cup A_{i+1})$ and precisely one vertex in $N(B_i \cup B_{i+1})$.

Since (S_1^i, S_2^i) satisfy X_i there exists a path P_{A1} from A_i to A_{i+1} through $X_i - S_1^i$, a path P_{A2} from A_i to A_{i+1} through $X_i - S_2^i$, a path P_{B1} from B_i to B_{i+1} through $X_i - S_1^i$, and a path P_{B2} from B_i to B_{i+1} through $X_i - S_2^i$. We select P_{A1}, P_{A2}, P_{B1} and P_{B2} to be the shortest such paths, specifically each of P_{A1} and P_{A2} contain precisely one vertex in $N(A_i)$ and precisely one in $N(A_{i+1})$. and each of P_{B1} and P_{B2} contain precisely one vertex in $N(B_i)$ and precisely one in $N(B_{i+1})$. Note that P_{A1} and P_{A2} are disjoint from $N(B_i \cup B_{i+1})$, and Note that P_{B1} and P_{B2} are disjoint from $N(A_i \cup A_{i+1})$ because each of these paths is disjoint from at least one $(A_i \cup A_{i+1})$, $(B_i \cup B_{i+1})$-separator.

We prove that $|N(A_i) \cap X_i| \leqslant 3$. Let x be the unique vertex in $N(A_i \cup A_{i+1})$ on P_{AB}, y be the unique vertex in $N(A_i)$ on P_{A1}, and z be the unique vertex in $N(A_i)$ on P_{A2}. Suppose for contradiction that $|N(A_i) \cap X_i| > 3$, and select a vertex $v \in (N(A_i) \cap X_i) - \{x, y, z\}$. We claim that

$$W' = (A_1, \ldots, A_{t+1}, B_1, \ldots, B_{t+1}, X_1, \ldots, X_i - \{v\}, X_{i+1}, \ldots, X_t)$$

is a t-critter partition of $G - v$.

Since G is a minimal t-critter, by Lemma 7.6.9 S_1^i and S_2^i are disjoint. Since they are distinct (by property *(iv)* of t-critters) they are also non-empty. If $v \in S_1^i$ then let $\hat{S}_1^i = S_1^i - \{v\}$. We have that $\hat{S}_1^i$ is a minimal $(A_i \cup A_{i+1})$,$(B_i \cup B_{i+1})$-separator in $(G - v)[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i - \{v\}]$. If $v \notin S_1^i$ then $S_1^i - \{v\}$ is a (not necessarily minimal) $(A_i \cup A_{i+1})$,$(B_i \cup B_{i+1})$-separator in $(G - v)[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i - \{v\}]$. Thus S_1^i contains a minimal $(A_i \cup A_{i+1})$,$(B_i \cup B_{i+1})$-separator $\hat{S}_1^i$ in $(G - v)[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i - \{v\}]$. Since $\hat{S}_1^i$ contains a vertex of P_{AB} we have that $\hat{S}_1^i$ is non-empty. In either case S_1^i contains a non-empty minimal $(A_i \cup A_{i+1})$,$(B_i \cup B_{i+1})$-separator $\hat{S}_1^i$ in $(G - v)[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i - \{v\}]$. An identical argument shows that S_2^i contains a non-empty minimal $(A_i \cup A_{i+1})$,$(B_i \cup B_{i+1})$-separator $\hat{S}_2^i$ in $(G - v)[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i - \{v\}]$. Since S_1^i and S_2^i are disjoint, so are $\hat{S}_1^i$ and $\hat{S}_2^i$.

Finally, P_{A1} is a path in $G - v$ from A_i to A_{i+1} through $(X_i - \{v\}) - \hat{S}_1^i$, P_{A2} is a path in $G - v$ from A_i to A_{i+1} through $(X_i - \{v\}) - \hat{S}_2^i$, P_{B1} is a path in $G - v$ from B_i to B_{i+1} through $(X_i - \{v\}) - \hat{S}_1^i$, and P_{B2} is a path in $G - v$ from B_i to B_{i+1} through $(X_i - \{v\}) - \hat{S}_2^i$. Hence W' is a t-critter partition of $G - v$ contradiciting minimality of (G, W). The proofs that $|N(B_i) \cap X_i| \leqslant 3$, $|N(A_{i+1}) \cap X_i| \leqslant 3$, and $|N(B_{i+1}) \cap X_i| \leqslant 3$ are symmetric. ■

Let s be a positive integer. A t-critter partition $W = (A_1, \ldots, A_{t+1}, B_1, \ldots, B_{t+1}, X_1, \ldots X_t)$

of a graph G is said to be *s-size-bounded* if $|A_i| \leqslant s$ and $|B_i| \leqslant s$ for $i \leqslant t+1$ and $|X_i| \leqslant s$ for $i \leqslant t$. In other words a A t-critter partition W is s-size bounded if all parts of the partition have size at most s. An s-size bounded t-critter is a graph G that has an s-size bounded t-critter partition W.

Lemma 7.6.12. *For every CMSO-definable hereditary graph family $\mathcal{F}$ there exists an integer T, such that for every integer s, if there exists an s-size-bounded t-critter $G \in \mathcal{F}$ and $t > T$ then there exists an s-size-bounded t-critter $G' \in \mathcal{F}$ and $t' > t$.*

Proof: Let $\mathcal{F}$ be a CMSO-definable graph family, and G be a minimal t-critter such that $G \in \mathcal{F}$. Set $I = \{1, \ldots, 6\}$. Since $\mathcal{F}$ is a CMSO-definable graph property, by Theorem 7.6.8 we have that the number of equivalence classes in $\mathcal{E}_{\equiv_{\mathcal{F}}}[\mathcal{G}_{\subseteq I}]$ is finite. Let γ be the number of equivalence classes in $\mathcal{E}_{\equiv_{\mathcal{F}}}[\mathcal{G}_{\subseteq I}]$. We prove that $T = 2^6\gamma$ satisfies the conclusion of the lemma.

Let G be a graph and $W = (A_1, \ldots, A_{t+1}, B_1, \ldots, B_{t+1}, X_1, \ldots X_t)$ be an s-size bounded t-critter partition for G. Because σ is hereditary we may assume without loss of generality that (G, W) is a witness-minimal t-critter. For every $1 \leqslant i \leqslant t$ we define

$$\begin{aligned} D_i &= N(A_i \cup B_i) \cap X_i, \\ Z_i &= D_i \cup \bigcup_{j \leqslant i}(A_j \cup B_j) \cup \bigcup_{j<i} X_j, \text{ and} \\ Q_i &= (V(G) - Z_i) \cup D_i. \end{aligned}$$

By Lemma 7.6.11 $|D_i| \leqslant 6$ for every i, while Lemma 7.6.10 yields that $D_i \cap N(A_i)$ and $D_i \cap N(B_i)$ are disjoint. For each i we define an injective labeling $\lambda_i : D_i \to \{1, \ldots, 6\}$, such that every $v \in D_i \cap N(A_i)$ satisfies $\lambda_i(v) \in \{1, 2, 3\}$, while every $v \in D_i \cap N(B_i)$ satisfies $\lambda_i(v) \in \{4, 5, 6\}$.

For every $i \leqslant t$ we define the boundaried graph $G_i^{pre} = G[Z_i]$ with boundary D_i

and labeling λ_i and $G_i^{post} = G[Q_i]$ with boundary D_i and labeling λ_i. Note that for every i we have $G = G_i^{pre} \oplus G_i^{post}$. If $t > T = 2^6\gamma$ then there exist $i < j \leqslant t$ such that $\Lambda(G_i^{pre}) = \Lambda(G_j^{pre})$ and $G_i^{pre} \equiv_{\mathcal{F}} G_j^{pre}$. Let $G' = G_j^{pre} \oplus G_i^{post}$. We claim that G' satisfies the conclusion of the Lemma. First, note that $G \in \mathcal{F}$ by assumption, so $G = (G_i^{pre} \oplus G_i^{post}) \in \mathcal{F}$. Since $G_j^{pre} \equiv_{\mathcal{F}} G_i^{pre}$ it follows that $(G_j^{pre} \oplus G_i^{post}) \in \mathcal{F}$. However, $G' = G_j^{pre} \oplus G_i^{post}$ and so $G' \in \mathcal{F}$.

We set $\hat{t} = (t+j-i)$ and show that that G' is a $\hat{t}$-critter by giving a $\hat{t}$-critter partition $\hat{W} = \hat{A}_1, \dots \hat{A}_{t\hat{+}1}$, $\hat{B}_1, \dots \hat{B}_{t\hat{+}1}$, $\hat{X}_1, \dots \hat{X}_{\hat{t}}$, of G' For $p \leqslant j$ we set $\hat{A}_p = A_p$, $\hat{B}_p = B_p$, and $\hat{X}_p = X_p$, or rather the copies of A_p, B_p and X_p respectively, in G_j^{pre}. For p from $j+1$ and up to $\hat{t}+1$ we set $\hat{A}_p = A_{p+i-j}$ and $\hat{B}_p = B_{p+i-j}$, more specifically the copies of A_{p+i-j} and B_{p+i-j} in G_i^{post}. For p from $j+1$ and up to p we set $\hat{X}_p$ to be the copy of X_{p+i-j} in G_i^{post}. It follows directly from their definitions that

$$\{\hat{A}_p, \hat{B}_p \ : \ 1 \leqslant p \leqslant t \hat{+} 1\} \cup \{\hat{X}_p \ : \ 1 \leqslant p \leqslant \hat{t}\}$$

is a partition of $V(G)$ and that it satisfies properties *(i)*, *(ii)* and *(iii)* of $\hat{t}$-critter partitions. We now check property (iv).

For property (iv) we note that for every $p < j$ we have that $G'[\hat{A}_p \cup \hat{A}_{p+1} \cup \hat{B}_p \cup \hat{B}_{p+1} \cup \hat{X}_p] = G[A_p \cup A_{p+1} \cup B_p \cup B_{p+1} \cup X_i]$, and that therefore (iv) is satisfied for all $p < j$. Similarly, for $p > j$ we have that $G'[\hat{A}_p \cup \hat{A}_{p+1} \cup \hat{B}_p \cup \hat{B}_{p+1} \cup \hat{X}_i] = G[A_{p+i-j} \cup A_{p+i-j+1} \cup B_{p+i-j} \cup B_{p+i-j+1} \cup X_{p+i-j}]$, and that therefore (iv) is satisfied for all $p > j$.

We are left with verifying property (iv) for $p = j$. We have that $G'[\hat{A}_j \cup \hat{B}_j] = G[A_j \cup B_j]$ and that $G'[\hat{A}_{j+1} \cup \hat{B}_{j+1} \cup \hat{X}_j] = G[A_{i+1} \cup B_{i+1} \cup X_i]$. Additionally, for every edge $uv \in E(G_j^{pre})$ such that $u \in A_j \cup B_j$ and $v \in D_j$ the copy u' of u in $\hat{A}_j \cup \hat{B}_j$ and vertex v' of $\hat{X}_j$ corresponding to v are adjacent in G'.

Let $\hat{S}_1^j$ and $\hat{S}_2^j$ be the copies in $\hat{X}_j$ of S_1^i and S_2^i respectively. We claim that $\hat{S}_1^j$ is a

$(\hat{A}_j \cup \hat{A}_{j+1})$, $(\hat{B}_j \cup \hat{B}_{j+1})$-separator in $G'[\hat{A}_j \cup \hat{A}_{j+1} \cup \hat{B}_j \cup \hat{B}_{j+1} \cup \hat{X}_j]$. Indeed, suppose there was a path $\hat{P}$ in $G'[\hat{X}_j] - \hat{S}_1^j$ that starts in a neighbor of $\hat{A}_j \cup \hat{A}_{j+1}$ and ends in a neighbor of $\hat{B}_j \cup \hat{B}_{j+1}$. Then the copy P of $\hat{P}$ in X_i would be a path that starts in a neighbor of $\hat{A}_i \cup \hat{A}_{i+1}$ and ends in a neighbor of $\hat{B}_i \cup \hat{B}_{i+1}$, contradicting that S_1^i separates $\hat{A}_i \cup \hat{A}_{i+1}$ from $\hat{B}_i \cup \hat{B}_{i+1}$ in $G[A_i \cup A_{i+1} \cup B_j \cup B_{i+1} \cup X_i]$. For an identical reason $\hat{S}_2^j$ is a $(\hat{A}_j \cup \hat{A}_{j+1})$, $(\hat{B}_j \cup \hat{B}_{j+1})$-separator in $G'[\hat{A}_j \cup \hat{A}_{j+1} \cup \hat{B}_j \cup \hat{B}_{j+1} \cup \hat{X}_j]$.

Since S_1^i and S_2^i are disjoint, so are $\hat{S}_1^i$ and $\hat{S}_2^i$. Since X_i contains a path from a neighbor of $A_i \cup A_{i+1}$ to a neighbor of $B_i \cup B_{i+1}$ in G, it follows that $\hat{X}_j$ contains a path from a neighbor of $\hat{A}_j \cup \hat{A}_{j+1}$ to a neighbor of $\hat{B}_k \cup \hat{B}_{j+1}$ in G. Therefore each of $\hat{S}_1^i$ and $\hat{S}_2^i$ contains a non-empty minimal $(\hat{A}_j \cup \hat{A}_{j+1})$, $(\hat{B}_j \cup \hat{B}_{j+1})$-separator in $G'[\hat{A}_j \cup \hat{A}_{j+1} \cup \hat{B}_j \cup \hat{B}_{j+1} \cup \hat{X}_j]$. Since $\hat{S}_1^i$ and $\hat{S}_2^i$ are disjoint these minimal separators are distinct.

Finally, $X_i - S_1^i$ contains a path P from a neighbor of A_i to a neighbor of A_{i+1}. The copy of P in $\hat{X}$ is a path from a neighbor of $\hat{A}_j$ to a neighbor of $\hat{A}_{j+1}$ in $\hat{X}_j - \hat{S}_1^j$. Identical arguments yield the existence of a paths from a neighbor of $\hat{A}_j$ to a neighbor of $\hat{A}_{j+1}$ in $\hat{X}_j - \hat{S}_2^j$, from a neighbor of $\hat{B}_j$ to a neighbor of $\hat{B}_{j+1}$ in $\hat{X}_j - \hat{S}_1^j$, and from a neighbor of $\hat{B}_j$ to a neighbor of $\hat{B}_{j+1}$ in $\hat{X}_j - \hat{S}_2^j$. We conclude that $\hat{W}$ is a $t + j - i$-critter partition of G'. Since every part of $\hat{W}$ is a copy of a part of W and W is s-size-bounded, so is $\hat{W}$.

■

Lemma 7.6.13. *Let G be a t-critter, then G has at least 2^t minimal separators.*

Proof: Let G be a t-critter and let $A_1, A_2, \ldots A_{t+1}$, $B_1, B_2, \ldots B_{t+1}, X_1, X_2, \ldots, X_t$ be the partitioning of its vertices given in Definition 7.1.3. Let S initially be an empty set, and for each i, $1 \leqslant i \leqslant t$, choose either S_1^i or S_2^i and add this set to S. Since each choice of adding S_1^i or S_2^i is made independently, there are 2^t choices for S. Let $a_1 \in A_1$ and $b_1 \in B_1$, we claim that S is a a_1,b_1-minimal separator.

It follows from properties *(ii)* and *(iv)* of Definition 7.1.3 that the vertices of the A_i's

all belong to one component, say A, of $G - S$ and that the vertices of the B_i's all belong to one component, say B, of $G - S$. It follows from properties *(i)*, *(iii)*, and *(iv)* that $A \neq B$. Hence S is an a_1, b_1-separator. To see that it is minimal, take some vertex $v \in S$, say v belongs to S^i where S^i is either S_1^i or S_2^i for $1 \leqslant i \leqslant t$. Then since S^i is a minimal $(A_i \cup A_{i+1})$, (B_i, B_{i+1})-separator in $G[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i]$, there will be a path from either A_i or A_{i+1} to either B_i or B_{i+1} in $G[A_i \cup A_{i+1} \cup B_i \cup B_{i+1} \cup X_i] - (S^i - v)$. Since all vertices of the A_i's belong to one component of $G - S$ and all vertices of the B_i's belong to a different component of $G - S$, there will be a path from a_1 to b_1 in $G - (S - v)$. It follows that S is a minimal separator. ■

Lemma 7.6.14. *For every hereditary CMSO-definable family $\mathcal{F}$, if for every t there exists a t-critter $G \in \mathcal{F}$ then σ is feral.*

Proof: Let $\mathcal{F}$ be a hereditary CMSO-definable family. By Lemma 7.6.12 there exists an integer T, such that for every integer s, if there exists an s-size-bounded t-critter $G \in \mathcal{F}$ with $t > T$ then there exists an s-size-bounded t-critter $G' \in \mathcal{F}$ with and $t' > t$. Select $t = T + 1$, then by assumption there exists a t-critter $G \in \mathcal{F}$. Let $s = |V(G)|$, then G is an s-bounded t-critter with $t > T$. By Lemma 7.6.12 there exists an s-bounded t'-critter $G' \in \mathcal{F}$, and $t' \geqslant t + 1$. Induction then yields that for every q there exists an s-bounded n-critter $G' \in \mathcal{F}$, and $n \geqslant q$. Such an n-critter G' has at least n and at most $5sn$ vertices, and at least 2^n minimal separators (by Lemma 7.6.13). Thus for every n there exists a graph $G' \in \mathcal{F}$, at least n vertices, and at least $(2^{1/5s})^n$ minimal separators. Therefore $\mathcal{F}$ is feral. ■

Lemma 7.6.14 handle graph properties that contain arbitrarily large critters. We now need to handle properties that contain arbitrarily large creatures.

The authors [5] showed that if G contains a k-creature for sufficiently large k, then G must contain a k'-creature which falls into one out of 6 very structured graph families.

Specifically, The authors proved the following.

Lemma 7.6.15 ([5]). *Let k be a natural number. Then there exists a natural number k' large enough so that if G is a graph that contains a k'-creature (A, B, $\{x_1, x_2, \ldots, x_{k'}\}$, $\{y_1, y_2, \ldots, y_{k'}\}$), then G contains an induced k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, or a k-ladder.*

We do not re-define the graphs listed in Lemma 7.6.15 in this paper, because everything we need to know about them is encapsulated in a simple observation which can easily be derived by inspection.

To phrase this observation we need a few definitions. A *boundaried path* is a 2-boundaried graph G such that G is a path and the endpoints of the path are the boundary vertices. An *apex path* is a graph G such that there exists a vertex v such that $G - v$ is a path and v is adjacent to both endpoints of the path $G - v$. The vertex v is then called an *apex*. The apex may or may not have edges to the internal verties of the path. A *boundaried apex path* is a 3-boundaried graph G such that G is an apex path, and the boundary of G is the apex v as well as the two endpoints of the path $G - v$. A *shortening* of a boundaried path is a boundaried path on fewer vertices. A *shortening* of a boundaried apex path is a boundaried apex path on fewer vertices.

An inspection k-thetas, k-prisms, k-pyramids, k-ladder-thetas, k-ladder-prisms, and k-ladders shows that if G is one of these graphs and G has $n >> k$ vertices, then G either contains a long induced path P or a long induced apex path $\hat{P}$, such that the internal vertices of P (or $\hat{P}$) do not have any neighbors outside of P (or $\hat{P}$). Further, shortening this path does not destroy the property of G being a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism, and k-ladder. We now formalize this observation in the language of boundaried graphs.

Lemma 7.6.16. *Let G be a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder on n vertices. Then $G = P \oplus R$ where P is either a boundaried path or a boundaried apex path on at least $\frac{n}{5k}$ vertices. Furthermore, for every shortening P' of P, $P' \oplus R$ is a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder.*

Proof: [Proof sketch.] The statement of the lemma immediately follows from the observation that for each of the listed graphs, the vertex set can be partitioned into at most $5k$ induced paths and apex paths, such that for each path/apex path in the partition only the endpoints and apex have neighbours outside the path/apex path. Details omitted. ■

Lemma 7.6.17. *Let $\mathcal{F}$ be a CMSO-definable graph family. Then there exists a constant c such that for every k, if there exists a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder G such that $\sigma(G) = \mathsf{true}$ then there exists a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder $G' \in \mathcal{F}$ such that $|V(G')| \leqslant ck$*

Proof: Set $I = \{1, 2, 3\}$ and consider the equivalence classes $\mathcal{E}_{\equiv_{\mathcal{F}}}[\mathcal{G}_{\subseteq I}]$ of $\equiv_\sigma$ restricted to I-boundaried graphs. Let γ_1 be the maximum, taken over all equivalence classes in $\mathcal{E}_{\equiv_{\mathcal{F}}}[\mathcal{G}_{\subseteq I}]$ that contain at least one boundaried path, of the *minimum* number of vertices of a boundaried path in that equivalence class. Similarly, let γ_2 be the maximumm taken over all equivalence classes in $\mathcal{E}_{\equiv_{\mathcal{F}}}[\mathcal{G}_{\subseteq I}]$ that contain at least one boundaried apex path, of the minimum number of vertices of a boundaried apex path in that equivalence class.

We set $\gamma = \max(\gamma_1, \gamma_2)$. From the choice of γ it follows that for every boundaried path P, if $V(P) > \gamma$ then there exists a boundaried path P' such that $P' \equiv_{\mathcal{F}} P$ and $|V(P')| \leqslant \gamma < |V(P)|$. Similarly, for every boundaried apex path P, if $V(P) > \gamma$ then there exists a boundaried apex path P' such that $P' \equiv_{\mathcal{F}} P$ and $|V(P')| \leqslant \gamma < |V(P)|$.

We set $c = 5\gamma$ and claim that c satisfies the conclusion of the lemma. Let k be an integer and suppose that there exists a graph $G \in \mathcal{F}$ such that G is a k-theta, k-prism,

k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder. Of all such graphs pick G with the minimum number of vertices. We claim $n = |V(G)| \leqslant ck = 5\gamma k$.

Suppose not, then by Lemma 7.6.16 we have that $G = P \oplus R$, where P is either a boundaried path or a boundaried apex path on at least $\frac{n}{5k} > \frac{5\gamma k}{5k} \geqslant \gamma$ vertices.

By the choice of γ there exists a shortening P' of P such that $P' \equiv_{\mathcal{F}} P$ and $|V(P')| \leqslant \gamma < |V(P)|$. Since P and P' are both boundaried paths or both boundaried apex paths it follows that the sizes of their boundaries are equal, namely $|\delta(P)| = |\delta(P')|$. We set $G' = P' \oplus R$. We have that

$$|V(G')| = |V(P')| + |V(R)| - |\delta(P')| < |V(P)| + |V(R)| - |\delta(P)| = |V(G)|.$$

By Lemma 7.6.16, since P' is a shortening of P we have that G' is a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder. Further, since $P' \equiv_{\mathcal{F}} P$ and $\sigma(P \oplus R) \in \mathcal{F}$ it follows that $G' = (P' \oplus R) \in \mathcal{F}$. But that contradicts the choice of G as the k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder with fewest vertices such that $G \in \mathcal{F}$. ■

Lemma 7.6.18. *Let G be a k-creature. Then G has at least 2^k minimal separators.*

Proof: Let G be a k-creature, and let A, $X = \{x_1, x_2, \ldots, x_k\}$, $Y = \{y_1, y_2, \ldots, y_k\}$, and B be the partition of $V(G)$ given in Definition 7.1.1. Let $a \in A$ and $b \in B$. We can make a minimal a, b-separator by selecting exactly one vertex from each pair x_i and y_i, $1 \leqslant i \leqslant k$. There are 2^k choices for such a minimal separator, which proves the lemma. ■

Lemma 7.6.19. *For every hereditary CMSO-definable graph family $\mathcal{F}$, if for every t there exists a t-creature G such that $G \in \mathcal{F}$ then $\mathcal{F}$ is feral.*

Proof: Let $\mathcal{F}$ be a hereditary CMSO-definable graph family such that for every t

there exists a t-creature G such that $G \in \mathcal{F}$.

We first claim that for every integer k there exists a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder G such that $G \in \mathcal{F}$. Towards a proof of this claim let k be given. By Lemma 7.6.15 there exists a k' such that every k'-creature G contains a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder as an induced subgraph. By our assumption on σ there exists a k'-creature G such that $G \in \mathcal{F}$. Let G' be an induced k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder in G. Since G' is an induced subgraph of G and $\mathcal{F}$ is hereditary it follows that $G' \in \mathcal{F}$. This proves the claim.

The claim, together with Lemma 7.6.17 yields that there exists a constant c such that for every k there exists a k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder G such that $G \in \mathcal{F}$ and $|V(G)| \leqslant ck$. However each of k-theta, k-prism, k-pyramid, k-ladder-theta, k-ladder-prism or k-ladder is a k-creature and therefore has at least 2^k minimal separators by Lemma 7.6.18. Hence, for every n there exists a graph G on at least n and at most cn vertices such that $G \in \mathcal{F}$ and G has at least $2^{n/c}$ minimal separators. Hence $\mathcal{F}$ is feral, as claimed. ■

Theorem 7.1.4, together with Lemmas 7.6.14 and 7.6.19 together imply Theorem 7.1.5

Proof: [Proof of Theorem 7.1.5] Let $\mathcal{F}$ be a CMSO-definable hereditary graph family. If there exists an integer k such that $\mathcal{F}$ neither contains a k-creature nor a k-critter then, by Theorem 7.1.4 $\mathcal{F}$ is quasi-tame. If no such integer k exists it follows that $\mathcal{F}$ either contains a t-critter for every t, or a t-creature for every t. In the first case $\mathcal{F}$ is feral by Lemma 7.6.14, in the second case $\mathcal{F}$ is feral by Lemma 7.6.19. ■

Bibliography

[1] P. Gartland and D. Lokshtanov, *Independent set on P_k-free graphs in quasi-polynomial time*, in *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 613–624, 2020.

[2] P. Gartland, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and P. Rzażewski, *Finding large induced sparse subgraphs in $C_{>t}$-free graphs in quasipolynomial time*, in *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021* (S. Khuller and V. V. Williams, eds.), pp. 330–341, ACM, 2021.

[3] P. Gartland, D. Lokshtanov, T. Masařík, M. Pilipczuk, M. Pilipczuk, and P. Rzażewski, *Maximum weight independent set in graphs with no long claws in quasi-polynomial time*, *arXiv preprint arXiv:2305.15738* (2023).

[4] V. Alekseev, *The effect of local constraints on the complexity of determination of the graph independence number*, *Combinatorial-algebraic methods in applied mathematics* (1982) 3–13. (in Russian).

[5] P. Gartland and D. Lokshtanov, *Graph classes with few minimal separators. I. finite forbidden induced subgraphs*, in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 3063–3097.

[6] P. Gartland and D. Lokshtanov, *Graph classes with few minimal separators. ii. a dichotomy*, in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 3098–3178.

[7] F. V. Fomin, I. Todinca, and Y. Villanger, *Large induced subgraphs via triangulations and CMSO*, *SIAM J. Comput.* **44** (2015), no. 1 54–87.

[8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Series of Books in the Mathematical Sciences. W. H. Freeman and Co., 1979.

[9] R. M. Karp, *Reducibility among combinatorial problems*, in *Complexity of Computer Computations*, pp. 85–103, 1972.

[10] U. Feige, S. Goldwasser, L. Lovász, and S. S. andF Mario Szegedy, *Interactive proofs and the hardness of approximating cliques*, *J. ACM* **43** (1996), no. 2 268–292.

[11] D. Zuckerman, *Linear degree extractors and the inapproximability of max clique and chromatic number*, *Theory of Computing* **3** (2007), no. 1 103–128.

[12] J. Håstad, *Clique is hard to approximate within* $n^{1-\varepsilon}$, *Acta Math.* **182** (1999), no. 1 105–142.

[13] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization.*, *Combinatorica* **1** (1981) 169–197.

[14] D. Lokshtanov, M. Vatshelle, and Y. Villanger, *Independent set in* P_5*-free graphs in polynomial time*, in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5–7, 2014* (C. Chekuri, ed.), pp. 570–581, SIAM, 2014.

[15] B. S. Baker, *Approximation algorithms for np-complete problems on planar graphs*, *J. ACM* **41** (1994), no. 1 153–180.

[16] P. Gartland, T. Korhonen, and D. Lokshtanov, *On induced versions of menger's theorem on sparse graphs*, 2023.

[17] V. E. Alekseev, *Polynomial algorithm for finding the largest independent sets in graphs without forks*, *Discrete Applied Mathematics* **135** (2004), no. 1-3 3–16.

[18] V. V. Lozin and M. Milanič, *A polynomial algorithm to find an independent set of maximum weight in a fork-free graph*, *J. Discrete Algorithms* **6** (2008), no. 4 595–604.

[19] A. Brandstädt and R. Mosca, *Maximum weight independent sets for* (P_7*, triangle)-free graphs in polynomial time*, *Discret. Appl. Math.* **236** (2018) 57–65.

[20] N. C. Lê, C. Brause, and I. Schiermeyer, *The Maximum Independent Set Problem in subclasses of* $S_{i,j,k}$*-free graphs*, *Electron. Notes Discret. Math.* **49** (2015) 43–49.

[21] M. U. Gerber, A. Hertz, and V. V. Lozin, *Stable sets in two subclasses of banner-free graphs*, *Discrete Applied Mathematics* **132** (2003), no. 1-3 121–136.

[22] V. V. Lozin and D. Rautenbach, *Some results on graphs without long induced paths*, *Inf. Process. Lett.* **88** (2003), no. 4 167–171.

[23] F. Maffray and L. Pastor, *Maximum weight stable set in* (P_7*, bull)-free graphs*, *CoRR* **abs/1611.09663** (2016).

[24] R. Mosca, *Stable sets of maximum weight in (P_7, banner)-free graphs*, *Discrete Mathematics* **308** (2008), no. 1 20–33.

[25] V. V. Lozin, M. Milanic, and C. Purcell, *Graphs without large apples and the Maximum Weight Independent Set problem*, *Graphs Comb.* **30** (2014), no. 2 395–410.

[26] V. V. Lozin, J. Monnot, and B. Ries, *On the maximum independent set problem in subclasses of subcubic graphs*, *J. Discrete Algorithms* **31** (2015) 104–112.

[27] A. Harutyunyan, M. Lampis, V. V. Lozin, and J. Monnot, *Maximum independent sets in subcubic graphs: New results*, *Theor. Comput. Sci.* **846** (2020) 14–26.

[28] V. V. Lozin and R. Mosca, *Independent sets in extensions of $2K_2$-free graphs*, *Discret. Appl. Math.* **146** (2005), no. 1 74–80.

[29] R. Mosca, *Stable sets in certain P_6-free graphs*, *Discret. Appl. Math.* **92** (1999), no. 2-3 177–191.

[30] R. Mosca, *Independent sets in (P_6, diamond)-free graphs*, *Discret. Math. Theor. Comput. Sci.* **11** (2009), no. 1 125–140.

[31] R. Mosca, *Maximum weight independent sets in (P_6, co-banner)-free graphs*, *Inf. Process. Lett.* **113** (2013), no. 3 89–93.

[32] R. Mosca, *Independent sets in (P_{4+4}, triangle)-free graphs*, *Graphs Comb.* **37** (2021), no. 6 2173–2189.

[33] V. Bouchitté and I. Todinca, *Treewidth and minimum fill-in: Grouping the minimal separators*, *SIAM J. Comput.* **31** (2001), no. 1 212–232.

[34] A. Grzesik, T. Klimosová, M. Pilipczuk, and M. Pilipczuk, *Polynomial-time algorithm for maximum weight independent set on P_6-free graphs*, *ACM Trans. Algorithms* **18** (2022), no. 1 4:1–4:57.

[35] T. Abrishami, M. Chudnovsky, M. Pilipczuk, P. Rzażewski, and P. D. Seymour, *Induced subgraphs of bounded treewidth and the container method*, in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10–13, 2021* (D. Marx, ed.), pp. 1948–1964, SIAM, 2021.

[36] A. Gyárfás, *Problems from the world surrounding perfect graphs*, *Applicationes Mathematicae* **3** (1987), no. 19 413–441.

[37] G. Bacsó, D. Lokshtanov, D. Marx, M. Pilipczuk, Z. Tuza, and E. J. van Leeuwen, *Subexponential-time algorithms for maximum independent set in P_t-free and broom-free graphs*, *Algorithmica* **81** (2019), no. 2 421–438.

[38] M. Chudnovsky, M. Pilipczuk, M. Pilipczuk, and S. Thomassé, *Quasi-polynomial time approximation schemes for the maximum weight independent set problem in* H*-free graphs*, in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020* (S. Chawla, ed.), pp. 2260–2278, SIAM, 2020.

[39] M. Chudnovsky and P. D. Seymour, *The three-in-a-tree problem*, *Combinatorica* **30** (2010), no. 4 387–417.

[40] T. Abrishami, M. Chudnovsky, C. Dibek, and P. Rzażewski, *Polynomial-time algorithm for maximum independent set in bounded-degree graphs with no long induced claws*, in *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference, January 9–12, 2022* (N. B. Joseph (Seffi) Naor, ed.), pp. 1448–1470, SIAM, 2022.

[41] M. Pilipczuk, M. Pilipczuk, and P. Rzażewski, *Quasi-polynomial-time algorithm for independent set in P_t-free graphs via shrinking the space of induced paths*, in *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11–12, 2021* (H. V. Le and V. King, eds.), pp. 204–209, SIAM, 2021.

[42] M. Milanic and N. Pivac, *Polynomially bounding the number of minimal separators in graphs: Reductions, sufficient conditions, and a dichotomy theorem*, *Electron. J. Comb.* **28** (2021), no. 1 1.

[43] T. Abrishami, M. Chudnovsky, C. Dibek, S. Thomassé, N. Trotignon, and K. Vušković, *Graphs with polynomially many minimal separators*, *J. Comb. Theory, Ser. B* **152** (2022) 248–280.

[44] J. Gajarskỳ, L. Jaffke, P. T. Lima, J. Novotná, M. Pilipczuk, P. Rzażewski, and U. S. Souza, *Taming graphs with no large creatures and skinny ladders*, *arXiv preprint arXiv:2205.01191* (2022).

[45] R. G. Downey and M. R. Fellows, *Parameterized Complexity*. Springer-Verlag, New York, 1999.

[46] R. Impagliazzo, R. Paturi, and F. Zane, *Which problems have strongly exponential complexity?*, *J. Comput. Syst. Sci.* **63** (2001), no. 4 512–530.

[47] P. Chalermsook, M. Cygan, G. Kortsarz, B. Laekhanukit, P. Manurangsi, D. Nanongkai, and L. Trevisan, *From gap-eth to fpt-inapproximability: Clique, dominating set, and more*, in *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017* (C. Umans, ed.), pp. 743–754, IEEE Computer Society, 2017.

[48] M. R. Garey and D. S. Johnson, *The rectilinear steiner tree problem is np-complete*, *SIAM Journal on Applied Mathematics* **32** (1977), no. 4 826–834.

[49] V. Guruswami and A. K. Sinop, *The complexity of finding independent sets in bounded degree (hyper)graphs of low chromatic number*, pp. 1615–1626. https://epubs.siam.org/doi/pdf/10.1137/1.9781611973082.125.

[50] K. Majewski, T. Masařík, J. Novotná, K. Okrasa, M. Pilipczuk, P. Rzażewski, and M. Sokołowski, *Max Weight Independent Set in Graphs with No Long Claws: An Analog of the Gyárfás' Path Argument*, in *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)* (M. Bojańczyk, E. Merelli, and D. P. Woodruff, eds.), vol. 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, (Dagstuhl, Germany), pp. 93:1–93:19, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[51] M. Chudnovsky, R. McCarty, M. Pilipczuk, M. Pilipczuk, and P. Rzażewski, *Sparse induced subgraphs in p_6-free graphs*, *arXiv preprint arXiv:2307.07330* (2023).

[52] M. C. Golumbic, *Algorithmic graph theory and perfect graphs*. Elsevier, 2004.

[53] P. Erdös and G. Szekeres, *A combinatorial problem in geometry*, *Compositio mathematica* **2** (1935) 463–470.

[54] B. Courcelle, *The monadic second-order logic of graphs I: Recognizable sets of finite graphs*, *Inform. and Comput.* **85** (1990) 12–75.

[55] M. Grohe, K.-i. Kawarabayashi, and B. Reed, *A simple algorithm for the graph minor decomposition- logic meets structural graph theory–*, in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 414–431, SIAM, 2013.

[56] C. Dallard, M. Milanič, and K. Štorgel, *Treewidth versus clique number. ii. tree-independence number*, *arXiv preprint arXiv:2111.04543v1* (2021).

[57] S. Albrechtsen, T. Huynh, R. W. Jacobs, P. Knappe, and P. Wollan, *The induced two paths problem*, *arXiv preprint arXiv:2305.04721* (2023).

[58] K. Hendrey, S. Norin, R. Steiner, and J. Turcotte, *On an induced version of menger's theorem*, 2023.

[59] T. Abrishami, M. Chudnovsky, and K. Vušković, *Induced subgraphs and tree decompositions i. even-hole-free graphs of bounded degree*, *Journal of Combinatorial Theory, Series B* **157** (2022) 144–175.

[60] T. Abrishami, M. Chudnovsky, C. Dibek, S. Hajebi, P. Rzazewski, S. Spirkl, and K. Vušković, *Induced subgraphs and tree decompositions ii. toward walls and their line graphs in graphs of bounded degree*, 2021.

[61] T. Abrishami, M. Chudnovsky, S. Hajebi, and S. Spirkl, *Induced subgraphs and tree decompositions iii. three-path-configurations and logarithmic treewidth*, *Advances in Combinatorics* (9, 2022).

[62] T. Abrishami, M. Chudnovsky, S. Hajebi, and S. T. Spirkl, *Induced subgraphs and tree decompositions iv. (even hole, diamond, pyramid)-free graphs*, *Electron. J. Comb.* **30** (2022).

[63] T. Abrishami, M. Chudnovsky, S. Hajebi, and S. Spirkl, *Induced subgraphs and tree decompositions vi. one neighbor in a hole*, *arXiv preprint arXiv:2205.04420* (2022).

[64] T. Abrishami, M. Chudnovsky, S. Hajebi, and S. Spirkl, *Induced subgraphs and tree decompositions vi. graphs with 2-cutsets*, *arXiv preprint arXiv:2207.05538* (2022).

[65] T. Abrishami, B. Alecu, M. Chudnovsky, S. Hajebi, and S. Spirkl, *Induced subgraphs and tree-decompositions vii. basic obstructions in h-free graphs*, *arXiv preprint arXiv:2212.02737* (2022).

[66] T. Abrishami, B. Alecu, M. Chudnovsky, S. Hajebi, and S. Spirkl, *Induced subgraphs and tree decompositions viii. excluding a forest in (theta, prism)-free graphs*, *arXiv preprint arXiv:2301.02138* (2023).

[67] B. Alecu, M. Chudnovsky, S. Hajebi, and S. Spirkl, *Induced subgraphs and tree decompositions ix. grid theorem for perforated graphs*, *arXiv preprint arXiv:2305.15615* (2023).

[68] N. L. D. Sintiari and N. Trotignon, *(theta, triangle)-free and (even hole, k 4)-free graphs—part 1: Layered wheels*, *Journal of Graph Theory* **97** no. 4 475–509, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/jgt.22666].

[69] T. Korhonen, *Grid induced minor theorem for graphs of small degree*, *Journal of Combinatorial Theory, Series B* **160** (2023) 206–214.

[70] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*. Springer, 2015.

[71] M. Grohe and S. Kreutzer, *Methods for algorithmic meta theorems*, in *Model Theoretic Methods in Finite Combinatorics - AMS-ASL Joint Special Session, Washington, DC, USA, January 5-8, 2009*, vol. 558 of *Contemporary Mathematics*, pp. 181–206, American Mathematical Society, 2009.

[72] N. Robertson and P. D. Seymour, *Graph minors. XX. Wagner's conjecture*, *J. Comb. Theory, Ser. B* **92** (2004), no. 2 325–357.

[73] E. Balas and C. S. Yu, *On graphs with polynomially solvable maximal-weight clique problem*, *Networks* **19** (1989) 247—-253.

[74] B. Courcelle, J. A. Makowsky, and U. Rotics, *Linear time solvable optimization problems on graphs of bounded clique-width*, *Theory Comput. Syst.* **33** (2000), no. 2 125–150.

[75] A. Brandstädt, J. P. Spinrad, *et. al.*, *Graph classes: a survey*. No. 3. Siam, 1999.

[76] H. De Ridder *et. al.*, *Information system on graph classes and their inclusions)*, *www.graphclasses.org* (2016).

[77] B. N. Clark, C. J. Colbourn, and D. S. Johnson, *Unit disk graphs*, *Discrete Math.* **86** (1990), no. 1 – 3 165 – 177.

[78] S. Poljak, *A note on stable sets and colorings of graphs*, *Commentationes Mathematicae Universitatis Carolinae* **15** (1974), no. 2 307 – 309.

[79] H. Broersma, T. Kloks, D. Kratsch, and H. Müller, *Independent sets in asteroidal triple-free graphs*, *SIAM J. Discrete Math.* **12** (1999), no. 2 276–287.

[80] N. Sbihi, *Algorithme de recherche d'un stable de cardinalite maximum dans un graphe sans etoile*, *Discrete Mathematics* **29** (1980), no. 1 53–76. (in French).

[81] G. J. Minty, *On maximal independent sets of vertices in claw-free graphs*, *Journal of Combinatorial Theory, Series B* **28** (1980), no. 3 284 – 304.

[82] D. Corneil, H. Lerchs, and L. Burlingham, *Complement reducible graphs*, *Discrete Applied Mathematics* **3** (1981), no. 3 163 – 174.

[83] R. Boliac and V. V. Lozin, *An augmenting graph approach to the stable set problem in P_5-free graphs*, *Discrete Applied Mathematics* **131** (2003), no. 3 567 – 575.

[84] A. Brandstädt and R. Mosca, *On the structure and stability number of P_5- and co-chair-free graphs*, *Discrete Applied Mathematics* **132** (2003), no. 1–3 47 – 65. Stability in Graphs and Related Topics.

[85] M. U. Gerber and V. V. Lozin, *On the stable set problem in special P_5-free graphs*, *Discrete Applied Mathematics* **125** (2003), no. 2-3 215–224.

[86] R. Mosca, *Some observations on maximum weight stable sets in certain P_5-free graphs*, *European Journal of Operational Research* **184** (2008), no. 3 849 – 859.

[87] B. Randerath and I. Schiermeyer, *On maximum independent sets in P_5-free graphs*, *Discrete Applied Mathematics* **158** (2010) 1041–1044.

[88] D. Lokshtanov, M. Pilipczuk, and E. J. van Leeuwen, *Independence and efficient domination on P_6-free graphs*, *ACM Trans. Algorithms* **14** (2018), no. 1 3:1–3:30.

[89] A. Grzesik, T. Klimosova, M. Pilipczuk, and M. Pilipczuk, *Polynomial-time algorithm for maximum weight independent set on p6-free graphs*, in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019* (T. M. Chan, ed.), pp. 1257–1271, SIAM, 2019.

[90] A. Grzesik, T. Klimosová, M. Pilipczuk, and M. Pilipczuk, *Covering minimal separators and potential maximal cliques in P_t-free graphs*, *CoRR* **abs/2003.12345** (2020).

[91] C. Brause, *A subexponential-time algorithm for the maximum independent set problem in p_t-free graphs*, *Discret. Appl. Math.* **231** (2017) 113–118.

[92] C. Groenland, K. Okrasa, P. Rzazewski, A. D. Scott, P. D. Seymour, and S. Spirkl, *H-colouring p_t-free graphs in subexponential time*, *Discret. Appl. Math.* **267** (2019) 184–189.

[93] F. V. Fomin and D. Kratsch, *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010.

[94] J. Edmonds, *Paths, trees, and flowers*, *Canadian Journal of Mathematics* **17** (1965) 449–467.

[95] A. Gyárfás, *On Ramsey covering-numbers*, in *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II*, no. 10 in Colloq. Math. Soc. Janos Bolyai, pp. 801–816. North-Holland, Amsterdam, 1975.

[96] C. Chekuri and J. Chuzhoy, *Polynomial bounds for the grid-minor theorem*, *J. ACM* **63** (2016), no. 5 40:1–40:65.

[97] J. Chuzhoy and Z. Tan, *Towards tight(er) bounds for the excluded grid theorem*, *J. Comb. Theory, Ser. B* **146** (2021) 219–265.

[98] N. Robertson and P. D. Seymour, *Graph minors. X. obstructions to tree-decomposition*, *J. Comb. Theory, Ser. B* **52** (1991), no. 2 153–190.

[99] K. Kawarabayashi, R. Thomas, and P. Wollan, *Quickly excluding a non-planar graph*, *CoRR* **abs/2010.12397** (2020) [arXiv:2010.1239].

[100] N. Robertson and P. D. Seymour, *Graph minors XIII. the disjoint paths problem*, *J. Comb. Theory, Ser. B* **63** (1995), no. 1 65–110.

[101] M. Chudnovsky and P. D. Seymour, *Claw-free graphs. V. Global structure*, *J. Comb. Theory, Ser. B* **98** (2008), no. 6 1373–1410.

[102] V. E. Alekseev and V. V. Lozin, *Augmenting graphs for independent sets*, *Discrete Applied Mathematics* **145** (2004), no. 1 3 – 10.

[103] J. Novotná, K. Okrasa, M. Pilipczuk, P. Rzążewski, E. J. van Leeuwen, and B. Walczak, *Subexponential-time algorithms for finding large induced sparse subgraphs*, *Algorithmica* (July, 2020).

[104] B. Courcelle and J. Engelfriet, *Graph Structure and Monadic Second-Order Logic — A Language-Theoretic Approach*, vol. 138 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2012.

[105] A. Atminas, V. V. Lozin, and I. Razgon, *Linear time algorithm for computing a small biclique in graphs without long induced paths*, in *Proceedings of the 13th Scandinavian Symposium and Workshops Algorithm Theory, SWAT 2012*, vol. 7357 of *Lecture Notes in Computer Science*, pp. 142–152, Springer, 2012.

[106] G. Ding, *Subgraphs and well-quasi-ordering*, *J. Graph Theory* **16** (1992), no. 5 489–502.

[107] J. Nešetřil and P. Ossona de Mendez, *Sparsity — Graphs, Structures, and Algorithms*, vol. 28 of *Algorithms and combinatorics*. Springer, 2012.

[108] M. Hatzel, P. Komosa, M. Pilipczuk, and M. Sorge, *Constant congestion brambles*, *CoRR* **abs/2008.02133** (2020) [arXiv:2008.0213].

[109] M. Pilipczuk, M. Pilipczuk, and S. Siebertz, *Lecture notes for the course Sparsity*, Winter semester 2019/20. available at: `https://www.mimuw.edu.pl/~mp248287/sparsity2/`.

[110] M. Grohe and D. Marx, *On tree width, bramble size, and expansion*, *J. Comb. Theory, Ser. B* **99** (2009), no. 1 218–228.

[111] K. Cameron and P. Hell, *Independent packings in structured graphs*, *Math. Program.* **105** (2006), no. 2-3 201–213.

[112] K. Edwards and G. Farr, *Fragmentability of graphs*, *Journal of Combinatorial Theory, Series B* **82** (2001), no. 1 30 – 37.

[113] M. Milanič and N. Pivač, *Minimal separators in graph classes defined by small forbidden induced subgraphs*, in *International Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 379–391, Springer, 2019.

[114] M. Chudnovsky, M. Pilipczuk, M. Pilipczuk, and S. Thomassé, *On the maximum weight independent set problem in graphs without induced cycles of length at least five*, *SIAM J. Discret. Math.* **34** (2020), no. 2 1472–1483.

[115] K. Menger, *Zur allgemeinen kurventheorie*, *Fundamenta Mathematicae* **10** (1927), no. 1 96–115.

[116] D. Marx, *Parameterized graph separation problems*, *Theor. Comput. Sci.* **351** (2006), no. 3 394–406.

[117] D. Lokshtanov, *On the complexity of computing treelength*, *Discret. Appl. Math.* **158** (2010), no. 7 820–827.

[118] T. Kloks and D. Kratsch, *Finding all minimal separators of a graph*, in *STACS 94, 11th Annual Symposium on Theoretical Aspects of Computer Science*, vol. 775 of *Lecture Notes in Computer Science*, pp. 759–768, Springer, 1994.

[119] A. Berry, J. P. Bordat, and O. Cogis, *Generating all the minimal separators of a graph*, *Int. J. Found. Comput. Sci.* **11** (2000), no. 3 397–403.

[120] D. Kratsch, *The structure of graphs and the design of efficient algorithms, habilitation*, *Friedrich-Schiller-University of Jena, Germany* (1996).

[121] T. Kloks, D. Kratsch, and C. K. Wong, *Minimum fill-in on circle and circular-arc graphs*, *J. Algorithms* **28** (1998), no. 2 272–289.

[122] K. Suchan, *Minimal separators in intersection graphs*, *Master's thesis, Akademia Gorniczo-Hutnicza im. Stanislawa Staszica w Krakowie* (2003).

[123] M. Chudnovsky, S. Thomassé, N. Trotignon, and K. Vuskovic, *Maximum independent sets in (pyramid, even hole)-free graphs*, *CoRR* **abs/1912.11246** (2019) [arXiv:1912.1124].

[124] R. Diestel, *Graph Theory, 4th Edition*, vol. 173 of *Graduate texts in mathematics*. Springer, 2012.

[125] F. Ramsey, *On a problem of formal logic*, *Proceedings of the London Mathematical Society* **s2-30** (1930), no. 1 264–286, [https://londmathsoc.onlinelibrary.wiley.com/doi/pdf/10.1112/plms/s2-30.1.264].

[126] N. Sauer, *On the density of families of sets*, *Journal of Combinatorial Theory, Series A* **13** (1972), no. 1 145 – 147.

[127] F. V. Fomin and Y. Villanger, *Treewidth computation and extremal combinatorics*, *Combinatorica* **32** (2012), no. 3 289–308.

[128] S. Arnborg, J. Lagergren, and D. Seese, *Easy problems for tree-decomposable graphs*, *Journal of Algorithms* **12** (1991) 308–340.

[129] B. Courcelle, *The expression of graph properties and graph transformations in monadic second-order logic*, in *Handbook of graph grammars and computing by graph transformation, Vol. 1*, pp. 313–400. World Sci. Publ, 1997.

[130] B. Courcelle, *The monadic second-order logic of graphs. III. Tree-decompositions, minors and complexity issues*, *RAIRO Inform. Théor. Appl.* **26** (1992), no. 3 257–286.

[131] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos, *(meta) kernelization*, *J. ACM* **63** (2016), no. 5 44:1–44:69.

[132] R. B. Borie, R. G. Parker, and C. A. Tovey, *Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families*, *Algorithmica* **7** (1992), no. 5&6 555–581.