UNIVERSITY OF CALIFORNIA, SAN DIEGO

The proof and search complexity of three combinatorial principles

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

in

Mathematics

by

James Aisenberg

Committee in charge:

Professor Sam Buss, Chair
Professor Russell Impagliazzo
Professor Shachar Lovett
Professor Jeffrey Remmel
Professor Jacques Verstraete

2016

The dissertation of James Aisenberg is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2016

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my committee: Russell Impagliazzo, Shachar Lovett, Jeffrey Remmel, Jacques Verstraete, and especially my advisor Sam Buss. I am also grateful for the support of my family; Kathy and Robert Aisenberg, as well as Jon, Katie and Lucia Aisenberg. I am also grateful for the support of my friends, especially Matt van Duyn, Andrew Kirwin, and Marisa Brandt.

Chapter 2, in full, is a reprint of material that will appear in the *Journal of Symbolic Logic*. Aisenberg, James; Bonet, Maria L.; Buss, Sam. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of material that has been submitted for publication. Aisenberg, James; Bonet, Maria L.; Buss, Sam; Crăciun, Adrian; Istrate, Gabriel. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of material that has been submitted for publication. Aisenberg, James; Bonet, Maria L.; Buss, Sam. The dissertation author was the primary investigator and author of this paper.

VITA

2009	B.A. in Mathematics and Physics, Wesleyan University. Middletown, CT.
2010	Graduate Studies, University of Cambridge. Cambridge, UK.
2016	Ph.D. in Mathematics, University of California, San Diego. San Diego, CA.

PUBLICATIONS

James Aisenberg, Maria Luisa Bonet and Sam Buss. "2-D Tucker is PPA complete." Submitted for publication.

James Aisenberg, Maria Luisa Bonet, Sam Buss, Adrian Crăciun, and Gabriel Istrate. "Short proofs of the Kneser-Lovász coloring principle." Submitted for publication.

James Aisenberg, Maria Luisa Bonet, Sam Buss, Adrian Crăciun, and Gabriel Istrate. "Short proofs of the Kneser-Lovász coloring principle." In *Proc. 42th International Colloquium on Automata, Languages, and Programming (ICALP'15)*, Lecture Notes in Computer Science 9135 (2015): 44–55.

James Aisenberg, Maria Luisa Bonet, and Sam Buss. "Quasi-polynomial size Frege proofs of Frankl's theorem on the trace of finite sets." To appear in *J. of Symbolic Logic*.

Itamar Sela, James Aisenberg, Tsampikos Kottos, Alex Elgart, and Doron Cohen. "Anomalous decay of a prepared state due to non-Ohmic coupling to the continuum." *Phys. Rev. E* 81.3 (2010): 036219.

James Aisenberg, Itamar Sela, Tsampikos Kottos, Doron Cohen, and Alex Elgart. "Quantum decay into a non-flat continuum." *J. Phys. A: Math. Theo.* 43.9 (2010): 095301.

Itamar Sela, James Aisenberg, Tsampikos Kottos, and Doron Cohen. "Quantum anomalies and linear response theory." *J. Phys. A: Math. Theo.* 43.33 (2010): 332001.

ABSTRACT OF THE DISSERTATION

The proof and search complexity of three combinatorial principles

by

James Aisenberg
Doctor of Philosophy in Mathematics
University of California San Diego, 2016
Professor Sam Buss, Chair

This work concerns the propositional proof complexity and computational complexity of Frankl's theorem on the trace of sets, the Kneser-Lovász theorem, and the Tucker lemma.

We show that propositional translations of Frankl's theorem on the trace of sets has quasi-polynomial size Frege proofs. For constant values of the parameter t, we prove that Frankl's theorem has polynomial size AC^0-Frege proofs from instances of the pigeonhole principle.

We prove that propositional translations of the Kneser-Lovász theorem have polynomial size extended Frege proofs and quasi-polynomial size Frege proofs for all fixed values of k. We present a new counting-based combinatorial proof of the Kneser-Lovász theorem that avoids the topological arguments of prior proofs for all but finitely many base cases. We introduce new "truncated Tucker lemma" principles, which are miniaturizations of the octahedral Tucker lemma. The truncated Tucker lemma implies the Kneser-Lovász theorem. We show that the $k = 1$ case of the truncated Tucker lemma has polynomial size extended Frege proofs.

We show that the 2-D Tucker search problem is PPA-hard under many-one reductions; therefore it is complete for PPA. The same holds for k-D Tucker for all $k \geq 2$. This corrects a claim in the literature that the Tucker search problem is in PPAD.

Frankl's theorem, the Kneser-Lovász theorem, and the truncated Tucker lemma are all shown to give total NP search problems. These problems are all shown to be PPP-hard under many-one reductions.

Chapter 1

Introduction

This dissertation is about algorithms and proofs associated with three combinatorial principles: Frankl's theorem on the trace of finite sets [29], Lovász's theorem on the chromatic number of Kneser graphs [48], and Tucker's lemma about triangulations of the n-ball [62]. Careful analysis of the algorithms involved in the proofs of these principles yields applications to the separation problem between Frege and extended Frege systems in propositional proof complexity [9], and in the classification of "inefficient proofs of existence" [53] in the domain of total NP search problems.

We begin by stating the three principles, giving an overview of their proofs without going into too many details, and briefly highlighting the algorithmic content of their proofs that we will explore later. In Section 1.1, we discuss the relevant background material in propositional proof complexity and total NP search problems. Section 1.2 summarizes the main results in this dissertation.

Frankl's theorem (Theorem 2.1) states:

Theorem 1.1. (Frankl [29]) *Let t be a positive integer and $m \leq n^{\frac{2^t-1}{t}}$. Then for any $m \times n$ 0/1 matrix with distinct rows, there is a column such that if this column is deleted, the resulting $m \times (n-1)$ matrix will contain fewer than 2^{t-1} pairs of equal rows.*

The proof of Frankl's theorem begins by showing that it suffices to consider the case of hereditary matrices. A 0/1 matrix with distinct rows is *hereditary* if replacing any 1 with a 0 in the matrix identifies two rows. The hereditary case is then handled by appealing to the Kruskal–Katona theorem [43, 47]. The algorithmic content of this proof that we are interested in is the reduction to the hereditary case. This reduction is done by giving a polynomial time algorithm that transforms an arbitrary counterexample to

Frankl's theorem into a hereditary counterexample to Frankl's theorem. The algorithm consists of successive applications of the down-shift operation to the given matrix. The *down-shift operation* considers some 1 in the matrix: if attempting to replace this 1 with a 0 identifies two rows, then no change is made, otherwise the matrix is updated with the 1 replaced by 0. This process is repeated until there are no 1's that can be replaced by 0's without identifying two rows. At this stage, the resulting matrix is hereditary. It is straightforward to check that being a counterexample to Frankl's theorem is an invariant of the down-shift operation. Chapter 2 investigates the complexity of the down-shift operation, where we give a more efficient algorithm for recognizing successive applications of the down-shift operation. This algorithm has implications for the separation problem between Frege and extended Frege proof systems. In Chapter 5, Frankl's theorem is shown to give a total NP search problem.

Tucker's lemma (Theorem 4.5) states:

Theorem 1.2 (Tucker [62])**.** *Let T be a triangulation of the n-ball in the ℓ_1-norm with the property that if $\sigma \in T$ is on the boundary of the n-ball, then $-\sigma \in T$. Let $V(T)$ denote the 0-simplices (vertices) of T. If $\lambda : V(T) \to \{\pm 1, \ldots, \pm n\}$ is a map with the property that for each vertex v on the boundary of T, $\lambda(v) = -\lambda(-v)$, then there is some 1-simplex $\{v_1, v_2\} \in T$ with $\lambda(v_1) = -\lambda(v_2)$.*

Tucker's lemma has a constructive proof [30, 31, 32, 50, 49] that inputs λ and outputs a 1-simplex $\{v_1, v_2\}$ with $\lambda(v_1) = -\lambda(v_2)$ by defining a walk whose nodes are the simplices in the triangulation T. The search procedure for Tucker's lemma runs in polynomial time in the number of simplices in T. Since the input to the search procedure is a labelling of 0-simplices, and the output is a particular 1-simplex, it is an interesting feature of this search procedure that its runtime is bound by the total number of simplices in the triangulation (and not just the number of 0- and 1-simplices). The fact that there are triangulations where the total number of simplices is exponentially larger than the number of 0- and 1-simplices is explored in the context of propositional proof complexity in Chapter 3, and in the context of total NP search problems in Chapter 5 in the form of the Truncated Tucker lemma. An unrelated question about what problems are reducible to the Tucker search problem is explored in Chapter 4.

The Kneser–Lovász theorem (Theorem 3.1) states:

Theorem 1.3 (Lovász [48])**.** *Let $n \geq 2k > 1$, and let $\binom{n}{k}$ denote the k-subsets of $\{1, \ldots, n\}$. If $c : \binom{n}{k} \to \{2k, \ldots, n\}$ is a map, then there are $A, B \in \binom{n}{k}$ with $A \cap B = \emptyset$*

such that $c(A) = c(B)$.

Lovász pioneered the use of topological methods in combinatorics by proving Kneser's conjecture by means of the Borsuk–Ulam theorem [11] about continuous maps from the n-sphere to $\mathbb{R}^n$. Matoušek [49] gave a more combinatorial proof of the Kneser–Lovász theorem that involves a reduction to Tucker's lemma. For fixed k, this reduction is very inefficient. To find disjoint A and B with $c(A) = c(B)$ for a map $c : \binom{n}{k} \to \{2k, \ldots, n\}$, Matoušek's construction reduces to an instance of the Tucker lemma on an exponentially large (in n) triangulation of the n-ball. In Chapter 3, we give a different proof of the Kneser–Lovász theorem that is much more efficient, with implications to the separation problem between Frege and extended Frege systems. In Chapter 5, we define total NP search problems based on the Kneser–Lovász theorem, and relate them to the search problems based on the Truncated Tucker lemma.

1.1 Background

1.1.1 Propositional proof complexity

Propositional proof complexity takes proof systems for establishing tautologies as its objects of study. A tautology is a Boolean formula that evaluates to True under any assignment to its variables. Fix some suitable encoding of Boolean formulas as binary strings, and let TAUT be the set of all tautologies under that fixed encoding. Following Cook–Reckhow [25], we define a propositional proof system so that the correctness of a proof can be verified in polynomial time, and every tautology has a proof (i.e., the proof system is complete).

Definition 1.4. A *propositional proof system* is a polynomial time computable surjective function $f : \{0,1\}^* \to \text{TAUT}$. If $f(x) = y$, then we say that x is an *f-proof* of y.

Unless otherwise specified, all proof systems will be taken to be propositional proof systems. A proof system f is said to be *super* if every tautology has a polynomial length proof. In other words, there exists a polynomial p such that for every tautology y there exists an f-proof x of y with $|x| \leq p(|y|)$. Whether or not super propositional proof systems exists is a fundamental question in computational complexity theory:

Theorem 1.5 (Cook–Reckhow [25])**.** *Super propositional proof systems exist iff* NP $=$ co$-$NP.

A function has a *quasipolynomial growth rate* if it is in $2^{O((\log x)^c)}$ for some constant c. We say that one proof system is stronger than other if the former can efficiently simulate proofs in the latter:

Definition 1.6. If f and g are propositional proof systems, then g *(quasi)polynomially simulates* f if there is some (quasi)polynomial time computable function h such that $f(x) = g(h(x))$ for all x.

A proof system is *optimal* if it can polynomially simulate any other proof system. Krajíček and Pudlák [46] show that EXP = NEXP implies that optimal proof systems exist.

Many proof systems have been shown to have good lower bounds. That is, they are known not to be super: resolution [36], constant-depth Frege [5], cutting plane [10, 56], the polynomial calculus [57], and others (see [59] for a survey). By contrast, Frege systems do not currently have any known superpolynomial lower bounds.

Frege systems are the usual "textbook style" proof system for propositional logic [25]. A Frege proof is a sequence of formulas. Each forumula in the sequence is either an axiom or has been inferred by a rule of inference from previously derived formulas. Axioms are substitution instances from a finite set of axiom schemes. Inference rules are also substitution instances from a finite set of inference rule schemes. A Frege system is fully specified by the axiom schemes, inference rule schemes, and formula basis (the allowed Boolean connectives). The notion of a Frege system is robust. For a fixed formula basis, every Frege system can polynomially simulate any other Frege system [25]. For Frege systems over different bases, it is still true that every Frege system can polynomially simulate every other Frege system, but care must used to make this precise [58].

Extended Frege systems are Frege systems with an additional non-schematic rule of inference, the extension rule. The extension rule allows extended Frege proofs to define new variables that abbreviate formulas. The extension rule allows a formula like the one that follows to appear as a line in an extended Frege proof:

$$x \equiv \phi$$

where ϕ is a formula, and x is a variable that has not been previously mentioned. If "$\equiv$" is not in the formula basis, then the line above is replaced with something logically

equivalent, such as

$$(x \to \phi) \wedge (\phi \to x).$$

Extended Frege systems, like Frege systems are robust. They can polynomially simulate each other over the same basis [25] or over different bases [58]. Any Frege proof is trivially an extended Frege proof, so extended Frege systems polynomially simulate Frege systems. A fundamental question in propositonal proof complexity is:

Question 1.7. *Can Frege systems polynomially simulate extended Frege systems?*

The extension rule allows extended Frege proofs to reason using Boolean circuits, whereas Frege proofs can only reason using Boolean formulas. Since it is conjectured that there are Boolean circuits that can only be expressed as exponentially longer Boolean formulas, it is also conjectured that there is an exponential separation between Frege proof length and extended Frege proof length. Currently there are no known results that make this connection precise, however.

If there is an exponential separation between Frege and extended Frege systems, then there ought to be a combinatorial principle whose propositional translations have polynomial size extended Frege proofs and require exponential size Frege proofs. Over the past 30 years, a number of combinatorial principles have been considered as candidates to provide such a separation, but none have born fruit. This history is described in detain in Section 2.1. In this section we will only discuss the pigeonhole principle. The $n+1 \to n$ pigeonhole principle states that there is no injection from a set of size $n+1$ (the pigeons) to a set of size n (the holes). To encode this statement in propositional logic, we use $(n+1)n$ propositional variables p_{ij} where $i = 1, \ldots, n+1$ and $j = 1, \ldots, n$. The interpretation of these variables it that they define a mapping $f : X \to Y$ where $X = \{1, \ldots, n+1\}$, $Y = \{1, \ldots, n\}$ and $f(i) = j$ iff p_{ij} is true. The pigeonhole principle tautology PHP is as follows:

$$\left(\bigwedge_i \bigvee_j p_{ij} \right) \to \bigvee_j \bigvee_{i \neq i'} p_{ij} \wedge p_{i'j}$$

The left-hand side of the conditional states that the p_{ij}'s define a total function. The right-hand side of the conditional states that the function defined by the variables p_{ij} is not injective.

The pigeonhole principle has polynomial size extended Frege proofs [25]. The argument is as follows: Suppose $f : \{1, \ldots, n+1\} \to \{1, \ldots, n\}$ is an injection. Define

$g : \{1, \ldots, n\} \to \{1, \ldots, n-1\}$ as follows:

$$g(x) = \begin{cases} f(x) & f(x) \neq n \\ f(n+1) & f(x) = n \end{cases}$$

It is straightforward to check that g is an injection if f is. The extension rule allows extended Frege proofs to introduce propositional variables defining g from the propositional variables defining f. Let q_{ij} be new propositional variables for $i = 1, \ldots n$, and $j = 1, \ldots, n-1$. The extension rule is used to define

$$q_{ij} \equiv p_{ij} \vee (p_{in} \wedge p_{(n+1)j}).$$

A straightforward argument shows that if the p's define an injection, then so do the q's. This process is iterated $n-1$ times (introducing new variables each time) until we reach an injection from $\{1, 2\}$ to $\{1\}$. A short exhaustive search verifies that there is no such injection, which completes the proof. This argument is readily translated into polynomial size extended Frege proofs. These extended Frege proofs use the extension rule to reason about linear depth circuits. Naïve translations of these circuits into formulas give exponential size formulas. In other words, naïve translations of these polynomial size extended Frege proofs into Frege proofs by replacing each extension variable with the formula it abbreviates yields exponential size Frege proofs.

For a number of years, it was thought that the pigeonhole principle would provide an exponential separation between Frege systems and extended Frege systems. However, Buss [16] showed that the pigeonhole principle has polynomial size Frege proofs. The main idea of the proof is to show that Frege systems can "count." In other words, for a collection of formulas $\phi_1, \ldots, \phi_k$, there are polynomial size formulas (in the size of $\phi_1, \ldots, \phi_k$) that express, in binary, the number of ϕ_i's that are true. These binary numbers can then be compared to other binary numbers, again using polynomial size formulas. The counting formulas make use of carry-save addition, which allows these formulas to be defined using log depth circuits. It is not enough to simply show that counting can be expressed by polynomial size formulas, crucial to the argument in [16] is that Frege systems can prove that counting formulas behave the way they are supposed to.

The inductive polynomial size extended Frege proof of the pigeonhole principle and the counting-based polynomial size Frege proof of the pigeonhole principle are quite different. Some have pointed to this fact as evidence that Frege systems do not

polynomially simulate extended Frege systems. If they did, then there must be a polynomial time procedure that inputs the inductive extended Frege proof, and outputs the counting-based Frege proof. This seems implausible. However, subsequently Buss [14] found quasipolynomial size Frege proofs of the pigeonhole principle that closely follow the inductive extended Frege proofs. This argument replaces the linear depth circuits of the extended Frege proof with a reduction to st-connectivity and hence $\log^2$ depth circuits, yielding quasipolynomial size Frege proofs. Thus the difference between the inductive proof and the counting proof of the pigeonhole principle cannot be taken as evidence against a quasipolynomial simulation of extended Frege systems by Frege systems.

1.1.2 Total NP search problems

Usually the problems studied in computational complexity theory are decision problems. For example, given a Boolean formula, decide if it has a satisfying assignment. It is also interesting to consider the complexity of search problems. For example, given a formula, find a satisfying assignment if it exists or state that there is no such assignment. By the self-reducibility of satisfiability, the search problem and decision problem for satisfiability are equivalent to each other, but in general search and decision problems could be different. For the satisfiability problem, a solution can be verified in polynomial time (in the length of the formula). Since a formula may be unsatisfiable, the search for a satisfying assignment may fail, i.e., the satisfiability problem is not a total search problem.

More formally, we define the class FNP to be the set of search problems associated with polynomial time, polynomially balanced relations $R(x, y)$. *Polynomially balanced* means that there exists some polynomial p such that if $R(x, y)$ holds, then $|y| \leq p(|x|)$. For a relation R, *the search problem associated with* R is: given x, find a y such that $R(x, y)$ holds or state that no such y exists. A relation is *total* if every x has a y such that $R(x, y)$ holds. The class of total NP search problems [53], that is the total problems in FNP, is denoted TFNP.

Returning to the previous example, satisfiability is in FNP because the relation R can treat x as an encoding of a CNF, y as an encoding of an assignment to the variables of x, and can evaluate x according to the assignment of the variables given by y in polynomial time. It is not, however, in TFNP because some CNFs are unsatisfiable.

Consider the problem Pigeonhole-Circuit: the input is a boolean circuit

with n inputs and n outputs. A solution is either two different assignments to the variables that give the same output, or one assignment of the variables that outputs $\vec{0}$ (that is, every output bit is 0). By the pigeonhole principle, every circuit has such solutions. Therefore Pigeonhole-Circuit is in TFNP.

Let FP denote the subclass of problems in TFNP with the property that a solution can be found in polynomial time. By definition FP $\subseteq$ TFNP $\subseteq$ FNP. We relate FP, TFNP, and FNP to more traditional decision problems in complexity. It's straightforward to see the following:

Theorem 1.8 ([53])**.** P = NP *iff* FP = FNP*.*

Theorem 1.9 ([53])**.** FP = TFNP *implies that* $P = \text{NP} \cap \text{co}-\text{NP}$*.*

To show that a relation is in TFNP requires a proof that the relation is total. There seems to be many reasons why relations are total, and so this makes it unlikely that TFNP has any complete problems. This makes TFNP what is called a semantic class. The notion of a semantic class is difficult to make precise, so for our purposes, a *semantic class* is a class that complexity theorists generally conjecture does not any have complete problems in it. Semantic classes are often contrasted with syntactic classes. For our purposes, a *syntactic class* is a class that is known to have complete problems in it. For example, $\text{NP} \cap \text{co}-\text{NP}$ is currently a semantic class and not a syntactic class. However, if it is shown that $\text{P} = \text{NP} \cap \text{co}-\text{NP}$, then $\text{NP} \cap \text{co}-\text{NP}$ will become a syntactic class and not a semantic class. Papadimitriou [53] defined several interesting syntactic subclasses of TFNP. Each subclass is associated with a combinatorial lemma that guarantees that relations in the subclass are total. For example, PPP is defined to be the class of problems proved total by the pigeonhole principle. The problem Pigeonhole-Circuit described above is complete for that class.

We will now shift to the oracle (type 2) setting, following the treatment in [7]. We do this both so that we can discuss separation results between subclasses of TFNP (absolute separations are difficult, in light of Theorem 1.8), and also because it simplifies the presentation.

Define the class FNP^2 to be the set of search problems on relations $R(\alpha, x, y)$ where α is an oracle, R is computable in polynomial time with oracle access to α, and R is polynomially balanced. A solution to the search problem associated with R on input (α, x) is any y such that $R(\alpha, x, y)$ holds. The class TFNP^2 is defined to be the subclass of FNP^2 where every input (α, x) has a y such that $R(\alpha, x, y)$.

Each subclass of TFNP^2 that we will define will have a standard problem in it. The class is formed by taking the smallest class that contains the problem, and is closed under many-one reductions.

A problem R_1 is *many-one reducible* to R_2 if there are polynomial time type 2 functions F, G, and H such that $H(\alpha, x, y)$ is a solution to R_1 on input (α, x) for any y that is a solution to R_2 on input $(G[\alpha, x], F(\alpha, x))$, where $G[\alpha, x]$ is a function that takes a string z as input, and outputs $G(\alpha, x, z)$.

We are ready to define the subclasses of TFNP^2. PPP is the subclass of TFNP^2 whose problems are proved total by the pigeonhole principle [53]. The standard problem for PPP is Pigeon.

Definition 1.10. An instance of Pigeon is a function $f : \{1, \ldots, n\} \to \{1, \ldots, n-1\}$. A solution to such an instance is one of the following:

1. A pair $a, b \in \{1, \ldots, n\}$ such that $a \neq b$ and $f(a) = f(b)$, or
2. an $a \in \{1, \ldots, n\}$ such that $f(a) \notin \{1, \ldots, n-1\}$.

Formally, an instance of Pigeon is (α, x) where α is a function oracle that computes f and $n = 2^{|x|}$. A solution to Pigeon is guaranteed to exist by the pigeonhole principle. In the definition of Pigeon, solutions of type 2 seem redundant, after all f was specified to be a map into $\{1, \ldots, n-1\}$, so of course every $f(a)$ will be in $\{1, \ldots, n-1\}$. The reason to include a solution of type 2 is that we do not assume that the input is valid. In other words, Pigeon is not a promise problem. Solutions of type 2 allow invalid inputs to be considered.

PPA is the subclass of TFNP^2 whose problems are proved total by the parity principle on undirected graphs [53]. The parity principle states that every undirected graph has an even number of nodes with odd degree. This principle is simplified as follows: every undirected graph where each nodes has degree 2 or less that has a leaf must have another. The standard problem for PPA is Leaf.

Definition 1.11. An instance of Leaf is an undirected graph $G = (V, E)$ on n many nodes, where each node has at most two neighbors, and there is a standard leaf $\vec{0}$ with exactly one neighbor. The edges E of the graph G are determined by given by a neighbor set N: for a node $v \in V$, the set of neighbors of v is $N(v)$. A solution to G as an instance of Leaf is one of the following:

1. a $v \neq \vec{0} \in V$ with $|N(v)| = 1$,
2. a $v_1, v_2 \in V$ such that $v_2 \in N(v_1)$ but $v_1 \notin N(v_2)$,
3. a $v \in V$ such that $|N(v)| > 2$, or
4. $\vec{0}$ with $|N(\vec{0})| \neq 1$.

Formally, an instance of LEAF is (α, x) where α is a function oracle computing the neighbor relation $N(v)$, and $n = 2^{|x|}$. Solutions of type 1 are leaves that are not the standard leaf. Solutions of type 2-4 are for rejecting invalid inputs. Solutions are guaranteed to exist by the parity argument.

PPAD is the subclass of TFNP^2 whose problems are proved total by the parity argument on directed graphs [53]. The parity argument on directed graphs states that every directed graph where each node has in-degree ≤ 1 and out-degree ≤ 1 with a source must have another node that is either a source or a sink. The standard problem for PPAD is LEAFD.

Definition 1.12. An instance of LEAFD is a directed graph $G = (V, E)$ on n many nodes, where each node has in-degree ≤ 1 and out-degree ≤ 1 with a standard source $\vec{0}$. For a node $v \in V$ in the graph, the set of outgoing neighbors of v is $N_{\text{out}}(v)$ and the set of incoming neighbors of v is $N_{\text{in}}(v)$. A solution to G as an instance of LEAFD is one of the following:

1. a $v \neq \vec{0} \in V$ with $|N_{\text{in}}(v)| \neq |N_{\text{out}}(v)|$,
2. a $v_1, v_2 \in V$ such that $v_2 \in N_{\text{out}}(v_1)$ and $v_1 \notin N_{\text{in}}(v_2)$,
3. a $v_1, v_2 \in V$ such that $v_2 \in N_{\text{in}}(v_1)$ and $v_1 \notin N_{\text{out}}(v_2)$,
4. a $v \in V$ such that $|N_{\text{out}}(v)| > 1$ or $|N_{\text{in}}(v)| > 1$, or
5. $\vec{0}$ with $|N_{\text{out}}(\vec{0})| \neq 1$ or $|N_{\text{in}}(\vec{0})| \neq 0$.

Formally, an instance of LEAFD is (α, x) where α is a function oracle computing the incoming and outgoing neighbor sets $N_{\text{in}}(v)$ and $N_{\text{out}}(v)$, and $n = 2^{|x|}$. Solutions of type 1 are unbalanced nodes that are not the standard source. Solutions of type 2–5 are for rejecting invalid inputs.

Since LEAFD is many-one reducible to LEAF (by ignoring the direction information), it follows that PPAD $\subseteq$ PPA. A straightforward argument also shows that

PPAD $\subseteq$ PPP [53]. [7] shows that relative to a fixed generic oracle, these inclusions are proper, and also that relative to a fixed generic oracle PPA and PPP are incomparable. We will not define generic oracles here, but only remark that these claims are equivalent to showing that LEAF is not many-one reducible to LEAFD, that PIGEON is not many-one reducible to LEAFD, and that LEAF and PIGEON are not many-one reducible to each other.

We describe how the type 2 classes PPP, PPA and PPAD defined above relate to the usual type 1 classes. These type 1 classes are closed under type 1 many-one reductions. If R_1 and R_2 are type 1 problems in FNP, then we say that R_1 is many-one reducible to R_2 if there are polynomial time computable functions f and g such that if x is an input to R_1, then $R_2(f(x), y)$ implies that $R_1(x, g(y))$. That is, an input x to R_1 can be converted into an input $f(x)$ to R_2 in polynomial time in such a way that a solution y to R_2 can be converted back to a solution $g(y)$ to R_1 in polynomial time.

Each standard problem PIGEON, LEAF, and LEAFD takes an input (α, x) where α is an oracle and x is a string. If we insist that α is computable in polynomial time with access to x, then we obtain a type 1 search problem. Taking the closure under type 1 many-one reductions yields the usual classes PPP, PPA and PPAD defined by Papadimitriou [53].

1.2 Summary of main results

Chapters 2 and 3 concern the proof complexity of Frankl's theorem and the Kneser-Lovász theorem. As mentioned before, an important open question in propositional proof complexity is whether Frege systems (quasi)polynomially simulate extended Frege systems. It is widely believed that they do not; however, there are few candidates of combinatorial principles that could witness such a separation [9]. Chapter 2, which is reproduced with permission from [1], discusses a long standing candidate for separating Frege and extended Frege: Frankl's theorem on the trace of finite sets (Theorem 2.1).

Bonet, Buss, and Pitassi [9] showed that Frankl's theorem has polynomial size extended Frege proofs. The extension rule was used to construct a hereditary counterexample to Frankl's theorem from an arbitrary counterexample by carrying out a polynomial time procedure. Chapter 2 shows that the polynomial time procedure can be replaced by a procedure computable by AC^1 circuits. This allows us to show (Theorems 2.8 and 2.9, respectively):

Theorem 1.13. *Propositional translations of Frankl's theorem have quasipolynomial size Frege proofs.*

Theorem 1.14. *Fix $t > 0$. The propositional translations of Frankl's theorem have polynomial size proofs in constant-depth Frege systems where the pigeonhole principle tautologies are taken as additional axioms.*

Chapter 3, which is reproduced with permission from [3], discusses the proof complexity of the Kneser-Lovász theorem, a more recent [40] candidate for separation Frege and extended Frege. Prior work [40] showed that propositional translations of the Kneser-Lovász theorem have polynomial size Frege proofs for $k = 2$, and polynomial size extended Frege proofs for $k = 3$. It was left open if the $k = 3$ case could separate Frege and extended Frege, and whether the fixed $k > 3$ case could show that extended Frege was not super. The main contribution of Chapter 3 eliminates these possibilities. We show that (Theorems 3.4 and 3.5, respectively):

Theorem 1.15. *For fixed parameter $k \geq 1$, the propositional translations of the Kneser-Lovász theorem have polynomial size extended Frege proofs.*

Theorem 1.16. *For fixed parameter $k \geq 1$, the propositional translations of the Kneser-Lovász theorem have quasipolynomial size Frege proofs.*

The key ingredient behind the proofs of both these theorems is a new proof of the Kneser-Lovász theorem based on counting that mostly avoids the topological reasoning of prior proofs.

Because our argument circumvented the topological reasoning of prior proofs, it does not address the question of whether or not Frege systems can carry out such arguments. To clarify this, we define a family of tautologies based on a version of the Tucker lemma that we call the truncated Tucker lemma. The truncated Tucker lemma has parameter k that matches the k of the Kneser-Lovász theorem. In other words, there are short Frege proofs of the Kneser-Lovász tautologies for fixed parameter k taking the truncated Tucker tautologies with fixed parameter k as additional axioms. We show (Theorem 3.26):

Theorem 1.17. *The $k = 1$ case of the truncated Tucker lemma has polynomial size extended Frege proofs.*

It is open whether Frege systems have subexponential size proofs of the $k = 1$ truncated Tucker tautologies. Thus, they are a candidate for separating Frege and extended Frege systems. We also leave open whether the $k > 1$ truncated Tucker tautologies have subexponential size extended Frege proofs.

Chapters 4 and 5 concern the computational complexity of Frankl's theorem, the Kneser-Lovász theorem, the Tucker lemma, and the truncated Tucker lemma in the context of total NP search problems. Chapter 4, which is reproduced with permission from [2], discusses the complexity of the search problem associated with the Tucker lemma. The search problem associated with the Tucker lemma was erroneously claimed to be PPAD-complete [53]. We show instead that (Theorem 4.1):

Theorem 1.18. 2-D Tucker *is* PPA*-complete under many-one reductions.*

It was known that 2-D Tucker is in PPA [53]. The proof of Theorem 4.1 involves showing that 2-D Tucker is PPA-hard. This construction builds off of Pálvölgyi's construction for showing that 2-D Tucker is PPAD-hard [52], while taking advantage of the boundary.

Chapter 5 defines a number of total NP search problems based on the combinatorial principles considered in earlier chapter, discusses some basic relationships between them, and poses a number of open questions. In particular, Frankl's theorem, the truncated Tucker lemma, and the Kneser-Lovász theorem all give rise to total NP search problems, and these search problems are all shown to be PPP-hard.

Chapter 2

Quasi-polynomial size Frege proofs of Frankl's Theorem on the trace of sets

2.1 Introduction

This paper extends results of Bonet, Buss, and Pitassi [9] and Nozaki, Arai, and Arai [51] by proving that Frankl's Theorem [29] has quasi-polynomial size Frege proofs. A Frege system is a "textbook" style proof system for propositional logic based on schematic axioms and inferences such as *modus ponens*. An extended Frege system is a Frege system augmented with the extension rule allowing the introduction of abbreviations, cf. Cook-Reckhow [25]. Lines in a Frege proof are Boolean formulas, whereas lines in an extended Frege proof can express Boolean circuits. It is generally conjectured that some Boolean circuits can only be expressed by exponentially larger Boolean formulas. For this reason, it is also generally conjectured that Frege proofs cannot polynomially simulate extended Frege proofs; however this is an open question.

Bonet, Buss, and Pitassi [9] looked for examples of tautologies that might be conjectured to provide exponential separations between the Frege and extended Frege proof systems. They found only a small number of examples other than partial consistency statements. The first type of examples were based on linear algebra, and included the Oddtown Theorem, the Graham-Pollack Theorem, the Fisher Inequality, and the Ray-Chaudhuri-Wilson Theorem. The remaining example was Frankl's Theorem on the

trace of sets.

The four principles based on linear algebra all have short extended Frege proofs using facts about determinants and eigenvalues. The same is true for the "$AB{=}I \Rightarrow BA{=}I$" tautologies about square matrices A and B over GF_2 that was subsequently suggested by S. Cook. Recently, Hrubeš and Tzameret [38] showed that determinant identities such as $\det(A)\det(B) = \det(AB)$ and $AB = I \Rightarrow BA = I$ have quasi-polynomial size Frege proofs. Thus it seems highly likely (as was already conjectured by [9]) that all these principles have quasi-polynomial size Frege proofs.

The remaining principle, Frankl's Theorem, was shown to have polynomial size extended Frege proofs by [9]. The main result of the present paper, Theorem 2.8, shows that the propositional formulations of Frankl's Theorem also have quasi-polynomial size Frege proofs.

Very few other other candidates (other than partial consistency principles) for exponentially separating Frege and extended Frege systems have been proposed. Kołodziejczyk, Nguyen, and Thapen [44] suggested the propositional translations of various local improvement principles LI, $\mathrm{LI}_{\log}$ and LLI as candidates, motivated by results on their provability in the bounded arithmetic theory V^1_2. They proved the LI principle is equivalent to partial consistency statements for extended Frege systems, but the other two remained as candidates. However, Beckmann and Buss [8] subsequently proved that $\mathrm{LI}_{\log}$ is provably equivalent (in S^1_2) to LI and that the linear local improvement principle LLI is provable in U^1_2. Therefore the former is equivalent to a partial consistency statement, and the latter has quasi-polynomial size Frege proofs. Thus neither of these provide good candidates for exponentially separating Frege and extended Frege systems. The rectangular local improvement principles RLI_k ([44, 8] for $k \geq 2$ are possible candidates for separation, as they are neither known to be provable in U^1_2 nor known to be many-complete for the provably total NP search problems of V^1_2.

Another family of propositional tautologies based on the Kneser-Lovász Theorem was recently proposed by Istrate and Crăciun [40]. They showed that the $k = 3$ versions of these tautologies have polynomial size extended Frege proofs, but left open whether they have (quasi-)polynomial size Frege proofs. However, subsequent work of Aisenberg, Bonet, Buss, Crăciun, and Istrate [3] has established that the Kneser-Lovász tautologies have polynomial size extended Frege proofs and quasi-polynomial size Frege proofs.

We thus lack many good candidates for super-quasipolynomially separating Frege and extended Frege systems, apart from partial consistency principles (cf., [25, 17]) or principles such as LI and $\mathrm{LI}_{\log}$ which are equivalent to partial consistency principles. This raises the question of whether Frege systems can quasi-polynomially simulate extended Frege systems. This seems very unlikely since none of the cases where Frege proofs (quasi-)polynomially simulate extended Frege proofs use methods that generalize to simulate arbitrary extended Frege proofs. The known simulations, such as the results of the present paper, may instead be useful to help show what kinds of techniques will be needed to separate Frege and extended Frege proofs.

The two restricted cases of Frankl's Theorem (Theorem 2.1) where the parameter t is equal to 1 or 2 have already been shown to have polynomial size Frege proofs. The $t = 1$ case is Bondy's Theorem, which Bonet, Buss, and Pitassi [9] proved to have polynomial size Frege proofs. They proved more than this in fact; namely, Bondy's Theorem is equivalent over AC^0-Frege to the pigeonhole principle PHP_n^{n+1}. Their proof involved showing that the bounded arithmetic theories $I\Delta_0 + \Delta_0$-PHP and $I\Delta_0 + \Delta_0$-BONDY are equivalent. Nozaki, Arai, and Arai [51] improved this by showing that the $t = 2$ case of Frankl's Theorem (known as Bollobás' Theorem) also has polynomial size Frege proofs. They did not explicitly address the question of AC^0-Frege reducibility to the pigeonhole principle, but it is easy to see that their constructions give such a reduction. In other words, their proof shows that there are polynomial size AC^0-Frege proofs of the propositional translations of Bollabás' Theorem from instances of the pigeonhole principle, and that Bollobás' Theorem is provable in $I\Delta_0 + \Delta_0$-PHP.

We extend these results to general t. Theorem 2.9 states that, for any fixed value of t, Frankl's Theorem has polynomial size Frege proofs. In fact, for a fixed value of t, Frankl's Theorem has polynomial size AC^0-Frege proofs from the Δ_0-PHP formulas. Likewise, for fixed values of t, Frankl's Theorem is provable in $I\Delta_0 + \Delta_0$-PHP.

Our proof methods substantially extend the constructions of [29, 9]. Like the original proof of Frankl [29], we reduce from the general case of Frankl's Theorem to the case where the matrix is hereditary. However, the direct transformation to a hereditary matrix as described by Frankl does not yield quasi-polynomial size propositional formulas. Thus, we need to use a different, more complicated construction that builds a hereditary matrix that is AC^1-definable. This construction can be translated into quasi-polynomial size Frege proofs and is the main new contribution of the present paper. The

prior constuction of [29, 9] could only be translated to polynomial size extended Frege proofs, but required exponential size Frege proofs. Surprisingly, our more complicated construction produces the same hereditary matrix as the prior construction, at least if the Frankl construction is carried out column by column.

Once the general case of Frankl's Theorem has been reduced to the case of hereditary matrices, the remainder of the proof of Frankl's Theorem is carried out by using the Kruskal-Katona Theorem [43, 47] in the same way as was done by both Frankl and Bonet-Buss-Pitassi. Additional work is need for the case of constant t, where we show that Frankl's Theorem has AC^0-Frege + PHP proofs. For this, we use a sharpened "functional" form (Theorem 2.7) of the Kruskal-Katona Theorem, which is based on AC^0-definable bijections. For constant values of t, we show that the functional form of the Kruskal-Katona Theorem has polynomial size AC^0-Frege proofs, and this allows us to construct the needed AC^0 reduction to the pigeonhole principle.

2.1.1 Frankl's Theorem and the Kruskal-Katona Theorem

Throughout the paper, A is an $m \times n$ 0/1 matrix with m distinct rows. We identify rows r of A with strings in $\{0,1\}^n$.

Theorem 2.1. (Frankl [29]) *Let t be a positive integer and $m \leq n^{\frac{2^t-1}{t}}$. Then for any $m \times n$ 0/1 matrix with distinct rows, there is a column such that if this column is deleted, the resulting $m \times (n-1)$ matrix will contain fewer than 2^{t-1} pairs of equal rows.*

We can rephrase this theorem using the following terminology.

Definition 2.2. Let r_1 and r_2 be two rows of A, and $j \in \{0, \ldots, n-1\}$. Row r_1 is *equivalent modulo column j to* row r_2 if r_1 and r_2 differ in exactly column j. We define P_j to be the set of rows r_1 for which there exists such a row r_2.

Note that $j \in \{0, \ldots, n-1\}$; columns are numbered from left to right, starting with $j = 0$. Since the rows of A are distinct, there can be at most one row equivalent to r_1 modulo column j; thus, $|P_j|$ is even. When column j is deleted, there are $|P_j|/2$ pairs of equal rows in the resulting $m \times (n-1)$ matrix. Frankl's Theorem can be rephrased as follows.

Theorem 2.3. *Let t be a positive integer, and let $m \leq n^{\frac{2^t-1}{t}}$. Then for any $m \times n$ 0/1 matrix with distinct rows, there is a j such that $|P_j| < 2^t$.*

Theorem 2.3 is trivial if $m < 2^t$ since $|P_j| \leq m$. Also, if $m \leq n$, we can take $t = 1$ and then Theorem 2.3 follows from Bondy's Theorem; and we already know Bondy's theorem has polynomial size Frege proofs. Thus we may assume that $m \geq 2^t$ and $m > n$.

Our proof, like the usual proof of Frankl's Theorem, goes through hereditary matrices and the Kruskal-Katona Theorem.

Definition 2.4. Let $\mathcal{F} = \{S_1, \ldots, S_m\}$ be a family of subsets of $\{0, \ldots, n-1\}$. The *incidence matrix* for $\mathcal{F}$ is an $m \times n$ 0/1 matrix with matrix element $a_{i,j} = 1$ iff $j \in S_i$. The family $\mathcal{F}$ is *hereditary* if $X \subset Y \in \mathcal{F}$ implies $X \in \mathcal{F}$. A 0/1 matrix is *hereditary* if it is the incidence matrix of some hereditary family.

Equivalently, a 0/1 matrix A is hereditary provided that, for any row r, changing any entry 1 in r to 0 yields another row of A.

Definition 2.5. If $r \in \{0,1\}^n$, we write $|r|_1$ to denote the number of ones in r. If A is an $m \times n$ 0/1 matrix and $k \geq 0$, we write $|A_{\leq k}|$ to denote the number of rows r of A such that $|r|_1 \leq k$.

For $r \in \mathbb{N}$, we let $|r|_1$ denote the number of 1's in the binary representation of r. For X a set of natural numbers, we write $|X_{\leq k}|$ to denote the number of $r \in X$ such that $|r|_1 \leq k$.

We next state the Kruskal-Katona Theorem needed for the proof of Frankl's Theorem. This is actually only a corollary to the Kruskal-Katona Theorem, see [29, 9], but we henceforth refer to it as the "Kruskal-Katona Theorem".

Theorem 2.6. *Let A be an $m \times n$ 0/1 hereditary matrix with distinct rows, and $k \geq 0$. Then*

$$|A_{\leq k}| \;\geq\; |\{0, 1, 2, \ldots, m-1\}_{\leq k}|. \tag{2.1}$$

Theorem 2.6 was shown to have polynomial size Frege proofs by [9]. When discussing AC^0-Frege proofs of Frankl's Theorem, we need the following functional form of the Kruskal-Katona Theorem.

Theorem 2.7. *Let A be an $m \times n$ 0/1 hereditary matrix with distinct rows. Then there is a bijection f from $\{0, 1, 2, \ldots, m-1\}$ onto the rows of A such that for every i, $|i|_1 \geq |f(i)|_1$.*

Theorem 2.7 is an immediate consequence of Theorem 2.6. Its advantage is that, for constant values of m, the bijection f can be defined with a constant depth formula.

2.1.2 Frege, extended Frege, and the main theorems

Frege proof systems are implicationally sound and complete propositional proof systems formalized with a finite set of schematic axioms and the inference rule *modus ponens* using, without loss of generality, the connectives $\neg$, $\wedge$, $\vee$, and $\rightarrow$. The length of a Frege proof is defined to be the total number of symbols in the proof. *Extended Frege systems* can be defined to be the same as Frege systems, but with proof length equal to the number of formulas (lines) in the proof instead of the number of symbols. An AC^0-Frege proof is a Frege proof in which all lines have alternation depth $O(1)$. For more information on Frege and extended Frege systems, see [25] or [9, 16, 45].

Frankl's Theorem, in the form of Theorem 2.3, is formalized as an infinite family of propositional tautologies as follows. Fix positive values n, m and t such that $m \leq n \cdot (2^t - 1)/t$. For $0 \leq i < m$ and $0 \leq j < n$, let $p_{i,j}$ be a propositional variable with the intended interpretation that $p_{i,j}$ is true iff the (i, j) entry of A is equal to 1. For $i \neq i'$, the formula $\textsc{Eq}(i, i', j)$ expresses that rows i and i' differ only in column j as

$$\textsc{Eq}(i, i', j) \ := \ \bigwedge_{j' \neq j} (p_{i,j'} \leftrightarrow p_{i',j'}).$$

By [16], there are polynomial size formulas expressing counting which allow polynomial size Frege proofs to reason about sizes of sets. This enables us to define the cardinality of P_j as

$$\textsc{CardP}(j) \ := \ \Big|\big\{i : 0 \leq i < m \text{ and } \bigvee_{i' \neq i} \textsc{Eq}(i, i', j)\big\}\Big|.$$

The size of $\textsc{CardP}(j)$ is polynomially bounded by the total size of the m many formulas $\bigvee_{i'} \textsc{Eq}(i, i', j)$; hence polynomially bounded by m and n. Letting $\textsc{DistinctRows}$ be the formula $\bigwedge_{i \neq i'} \bigvee_j (\neg p_{i,j} \leftrightarrow p_{i',j})$, Frankl's Theorem (for these values of m, n, t) can be expressed by the polynomial size propositional formula

$$\textsc{DistinctRows} \rightarrow \bigvee_j (\textsc{CardP}(j) < 2^t).$$

This formula has size polynomially bounded by m, n and t. We next state our two main results precisely. A proof is said to be *quasi-polynomially bounded* if it is quasi-polynomially bounded by the size of the formula that is proved.

Theorem 2.8. *There are quasi-polynomial size Frege proofs $P_{m,n,t}$ of the propositional translations of Frankl's Theorem.*

As already remarked, Theorem 2.8 is trivial if $m < 2^t$, and is known (via Bondy's Theorem) for $m \leq n$. In other cases, the Frege proof $P_{m,n,t}$ will have quasi-polynomially (in m) many steps, and each formula in $P_{m,n,t}$ will be equivalent to an AC^1-circuit. Namely, each formula will have only polynomially many distinct subformulas, and will have only $O(\log m)$ many alternations of $\wedge$'s and $\vee$'s.

For the next theorem, we assume t is constant. In this case, there are polynomial size formulas with $O(1)$ alternations of $\wedge$'s and $\vee$'s (that is, AC^0-circuits) that express the condition "$\textsc{CardP}(j) < 2^t$". To see this, note that its negation "$\textsc{CardP}(j) \geq 2^t$" can be expressed as the disjunction over all 2^t-tuples $i_1 < i_2 < \cdots < i_{2^t}$ of the assertions that every $i_\ell \in P_j$. Thus, for a constant value for t, the propositional translations of Frankl's Theorem can be expressed as constant depth, polynomial size formulas.

As is customary (cf. [23]), we let AC^0-Frege + PHP denote the Frege proof system augmented with all substitution instances of the $n+1$ into n pigeonhole principle for all $n \geq 1$, and restricted so that all formulas have alternation depth $O(1)$.

Theorem 2.9. *Fix $t > 0$. There are AC^0-Frege + PHP proofs $P^t_{m,n}$ of the propositional translations of Frankl's Theorem which have polynomial size (in m, n) and in which all formulas have alternation depth $O(t) = O(1)$.*

The outline of the paper is as follows. Sections 2.2.1 through 2.2.3 give our new reduction to the hereditary case of Frankl's Theorem. The general strategy of the proof is as follows. Given a 0/1 matrix A, we let T be the prefix tree for the rows of A. The nodes of T are sets of rows of A that share a common prefix, and the ancestor relation for T is set inclusion. We define a function χ that takes as input a node of T and a list of column indices, and produces another node in T. This χ function is used to define another $m \times n$ 0/1 matrix A', which is hereditary. Furthermore, if A violates the conditions of Frankl's Thoerem, then so does A', From here, we are in the situation for the usual proof of Frankl's Theorem, and we conclude our proof by using the Kruskal-Katona Theorem. Section 2.2.4 describes the functional form of the Kruskal-Katona Theorem which will be needed for polynomial size Frege proofs of the constant t case.

Section 2.3.1 discusses how to formalize this proof of Frankl's Theorem in propositional logic. The key point is that (the graph of) the χ function can be defined with

AC^1-circuits, and that the properties of the χ function can be established with quasi-polynomial size Frege proofs. Section 2.3.2 discusses the formalization of the constant t case of Frankl's Theorem with AC^0-Frege + PHP proofs. The key new tool is that the bijective form of the Kruskal-Katona Theorem can be formulated and proved in AC^0-Frege.

Section 2.4 shows that the matrix A' is identical to the hereditary counterexample produced in the usual proof of Frankl's Theorem when the reduction to a hereditary matrix is carried out column by column.

2.2 Proof of Frankl's Theorem

This section gives our reduction from the general case of Frankl's Theorem to the hereditary case. We define the reduction and prove its correctness in detail, so that it will be clear in Section 2.3 that the arguments can be formalized with quasi-polynomial size Frege proofs. Section 2.2.1 builds the prefix tree for the rows of A, Section 2.2.2 defines the χ function and establishes its properties. Section 2.2.3 uses the χ function to construct hereditary matrix, culminating with Theorem 2.25. Section 2.2.4 proves the bijective version of the Kruskal-Katona Theorem as will be needed for the AC^0-Frege + PHP proofs. We assume henceforth that A is an $m \times n$ 0/1 matrix with distinct rows and $m \leq n^{\frac{2^t-1}{t}}$.

2.2.1 The prefix tree for A

Recall that a row r is identified with a string in $\{0,1\}^n$. A binary string x is a *prefix* of r when r equals the concatenation xy for some y.

Definition 2.10. Let $x \in \{0,1\}^*$. Then $[\![x]\!]$ denotes the collection of the rows of A that have prefix x:

$$[\![x]\!] \;=\; \{r : r \text{ is a row of } A,\, x \text{ is a prefix of } r\}.$$

We call x the *maximal length representative* for $[\![x]\!]$ if there is no y with $|y| > |x|$ and $[\![y]\!] = [\![x]\!]$. The notation $[x]$ is used to denote $[\![x]\!]$ in this case.

Of course, every non-empty $[\![x]\!]$ has a unique maximal representative. Whenever we use the notation $[x]$, it is (implicitly) required that $[\![x]\!] \neq \emptyset$ and x is its maximal representative. For $|x| < n$, we have $[\![x]\!] = [\![x0]\!] \cup [\![x1]\!]$. The string x is a maximal

representative for $[\![x]\!]$ iff $[\![x]\!] \neq \emptyset$ and either $|x| = n$ or both $[\![x0]\!]$ and $[\![x1]\!]$ are non-empty.

The classes $[x]$ are the nodes of a binary tree T called the *prefix tree* of A. The root of T is $[\![\epsilon]\!]$, where ϵ is the empty string and thus $[\![\epsilon]\!]$ is the set of all rows of A. The root $[\![\epsilon]\!]$ is equal to $[y]$ for y the longest common initial substring of the rows. The leaves of T are the singleton nodes $[r]$, where $r \in \{0,1\}^n$ is a row of A.

The parent-child relationships of T are defined so that $[x]$ is an ancestor of $[y]$ in T precisely when $[x] \neq [y]$ and x is a prefix of y. In more detail, if $[x]$ is not a leaf node (in other words $|x| < n$) then the only two children of $[x]$ are its left child $[\![x0]\!]$ and its right child $[\![x1]\!]$. Thus T is an ordered binary tree, and since T has m leaves, it has $m-1$ internal nodes.

As an example, Figure 2.1 shows the prefix tree for the matrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Our single/double bracket notation means, for instance, that the rightmost leaf [11000] of the tree is also equal to $[\![11]\!] = [\![110]\!] = [\![1100]\!]$. The sets P_j of rows which are equivalent modulo column j were defined in Section 2.1.1. In this example, the sets P_j are:

$$\begin{aligned} P_0 &= \{00000, 10000, 00001, 10001, 01000, 11000\} \\ P_1 &= \{00000, 01000, 10000, 11000\} \\ P_2 = P_3 &= \{00000, 00100, 00010, 00110\} \\ P_4 &= \{00000, 00001, 10000, 10001\}. \end{aligned}$$

Each set P_j has prefix tree T_j. Formally, the nodes of T_j will identified with nodes of T, making it an induced subtree of T.

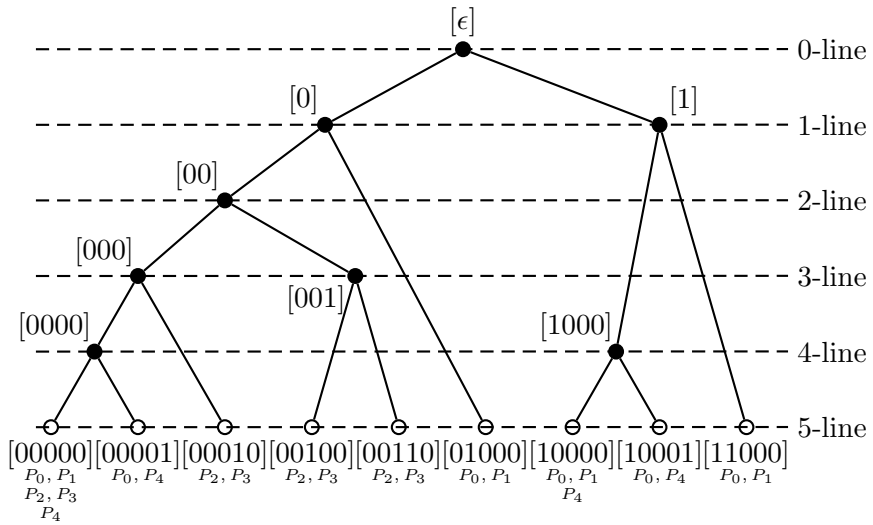


Figure 2.1: The prefix tree T of A.

Definition 2.11. Let $j \in \{0, \ldots, n-1\}$. The tree T_j has leaves $[r]$ for $r \in P_j$, and has as internal nodes the least common ancestors of every pair of $[r]$'s. The ancestor relationship is inherited from T.

Definition 2.12. Let $j \in \{0, \ldots, n-1\}$. The *j-line* through the tree T is defined to be

$$\{[x] : [x] \in T \text{ and } |x| = j\}.$$

In other words, the j-line is the set of nodes $[x]$ in T such that $[\![x0]\!] \neq [\![x1]\!]$ with $|x| = j$. In the above, $[\![10]\!] = [1000]$ is on the 4-line. The j-line corresponds to column j of the matrix, in that two rows of A which differ first in column j give rise to a node on the j-line. Note that any node on the j-line is in the tree T_j, but T_j has other nodes as well. We picture the tree T with root at the top and j-lines ordered accordingly, and say that the j-line and its nodes in T are *above* the j'-line if $j < j'$.

In Figure 2.2, the tree T_0 has one node on the 0-line, $[\epsilon]$. Its two children are roots of isomorphic subtrees of T_0. The next lemma shows this property always holds.

Definition 2.13. Let S be T or one of its induced subtrees T_j. Let $[x]$ be an internal node of S. The *left subtree of* $[x]$ *in* S is the subtree of S rooted at the left child of $[x]$ in S. The right subtree of $[x]$ is defined similarly.

Lemma 2.14. *Let $[x] \in T_j$ lie on the j-line. Then the right and left subtrees of $[x]$ in T_j are isomorphic in the following strong sense: For each node $[x0u]$ in the left subtree,*

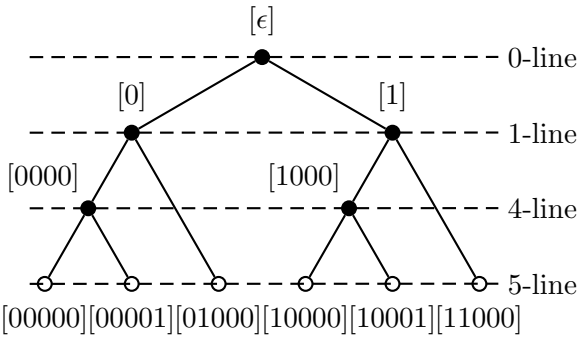


Figure 2.2: The prefix tree T_0 associated with P_0.

there is a corresponding node $[x1u]$ *in the right subtree; and conversely, for each node* $[x1u]$ *in the right subtree, there is a corresponding node* $[x0u]$ *in the left subtree.*

Proof. The leaves of the left (resp., right), subtrees of $[x]$ in T_j are the classes $[r]$ for r a row of A of the form $r = x0y$ (resp., $r = x1y$). In fact, $[x0y]$ is in P_j if and only if $[x1y]$ is in P_j. The internal nodes of these two subtrees are least common ancestors of these leaves. From this, the lemma follows. □

A consequence of Lemma 2.14 is that every leaf node of T_j has a ancestor on the j-line. Indeed, every node of T_j below the j-line has an ancestor on the j-line. This is because every node $[x0u]$ of T_j has a corresponding node $[x1u]$ in T_j, and their least common ancestor is $[x]$ on the j-line.

2.2.2 The χ function

The χ_S function takes a node $[x]$ of a tree S and a sequence of columns, and produces a node in the subtree of S rooted at $[x]$:

Definition 2.15. Let S be either T or one of its induced subtrees T_j. Let $[x]$ be an internal node of S and let $j_1 < j_2 < \cdots < j_\ell$ be a (possibly empty) sequence of columns with $\ell \geq 0$. The function $\chi_S([x], j_1, j_2, \ldots, j_\ell)$, with $\ell + 1$ arguments, is defined by induction on ℓ, and will equal either $[x]$ or a node below $[x]$ in S. For the base case $\ell = 0$, define $\chi_S([x]) = [x]$.

Now let $\ell \geq 1$. Suppose $[x]$ has the property that its left and right subtrees in S each contain a node $[y]$ on the j_1-line for which $\chi_S([y], j_2, \ldots, j_\ell)$ is defined. Let $[y]$ be the leftmost such node in the right subtree of $[x]$ in S. Then $\chi_S([x], j_1, \ldots, j_\ell)$ is defined (written $\chi_S([x], j_1, \ldots, j_\ell)\!\downarrow$) and

$$\chi_S([x], j_1, \ldots, j_\ell) \;=\; \chi_S([y], j_2, \ldots, j_\ell).$$

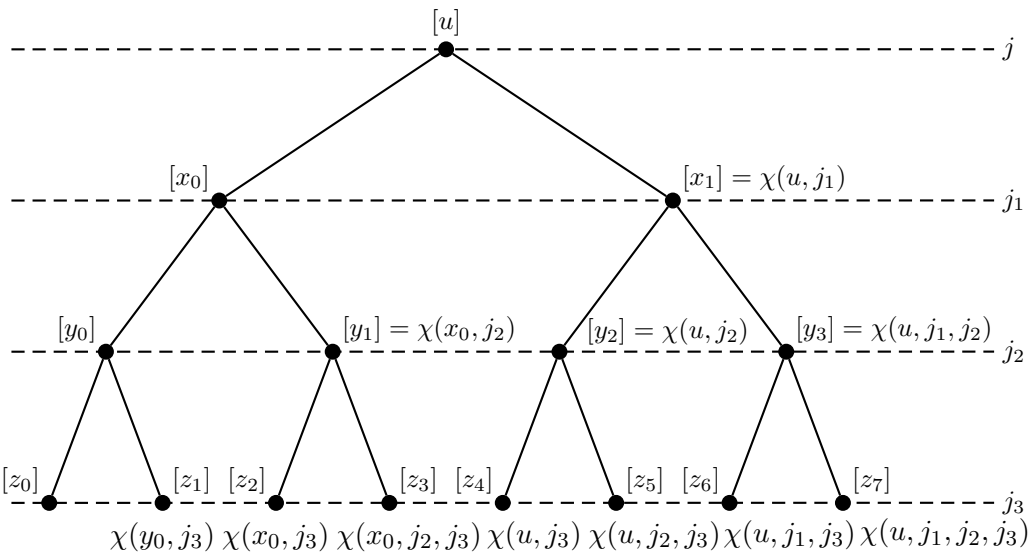


Figure 2.3: An example of a tree T with χ values specified.

In all other cases, $\chi_S([x], j_1, \ldots, j_\ell)$ is undefined.

When $S = T$, we write $\chi([x], j_1, \ldots, j_\ell)$ instead of $\chi_T([x], j_1, \ldots, j_\ell)$. Additionally, to simplify the notation, $\chi([x], j_1, \ldots, j_\ell) = [z]$ will be written as $\chi(x, j_1, \ldots, j_\ell) = z$.

We use the notation $\vec{j}$ to stand for an increasing sequence $j_1, \ldots, j_\ell$. Additionally, $|\vec{j}| = \ell$ is the length of the sequence $\vec{j}$. Finally, we write $\vec{j}' \subseteq \vec{j}$ to denote that the sequence $\vec{j}'$ is a subsequence of $\vec{j}$. Note that $\chi_S(x, \vec{j})$ is defined only for *internal* nodes $[x]$, and its value is also an internal node of S.

Later, Lemma 2.36 will describe the meaning of the χ function when A is hereditary. (The reader may skip ahead to read the statement and proof of Lemma 2.36 if desired.) The general intuition is that when $\chi(x, j_1, \ldots, j_\ell)\!\downarrow$ then the subtree rooted at $[x]$ contains a complete binary subtree of height ℓ as an induced subtree; the internal nodes of this binary tree lie on the j_i-lines for $i = 1, \ldots, \ell$.

Lemma 2.16. *Let S be T or one of its induced subtrees T_j. For fixed $\ell \geq 0$, the map $(x, j_1, \ldots, j_\ell) \mapsto \chi_S(x, j_1, \ldots, j_\ell)$ is injective.*

Proof. We will suppress the subscript S from χ_S in what follows. First we prove the following subclaim: For fixed $j_1, \ldots, j_\ell$, the map $x \mapsto \chi(x, j_1, \ldots, j_\ell)$ is injective. We prove this by induction. The base case $\ell = 0$ is the injectivity of the identity function. For the induction step, suppose $\ell \geq 1$ and the map $x \mapsto \chi(x, j_2, \ldots, j_\ell)$ is injective. Suppose $[x] \neq [x']$ and that $\chi(x, j_1, \ldots, j_\ell) = \chi(x', j_1, \ldots, j_\ell)$ and these quantities are

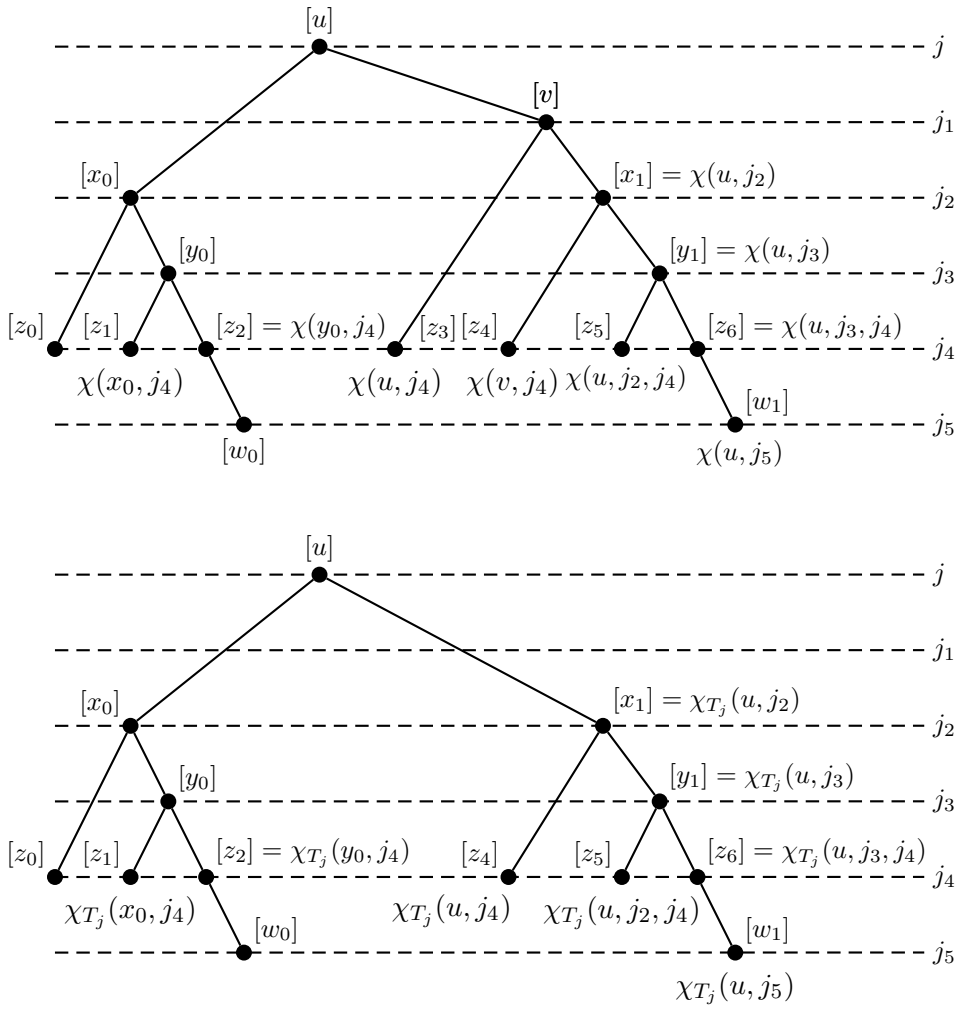


Figure 2.4: An example of a tree T (top) and T_j (bottom) with χ values specified. Each node is an internal node; the leaf nodes are not drawn.

defined. This means that $\chi(y, j_2, \ldots, j_\ell) = \chi(y', j_2, \ldots, j_\ell)$, where $[y]$ is the leftmost node on the j_1-line in $[x]$'s right subtree for which $\chi(y, j_2, \ldots, j_\ell)\downarrow$, and similarly for $[y']$ in $[x']$'s right subtree. By the induction hypothesis, $z \mapsto \chi(z, j_2, \ldots, j_\ell)$ is injective. Therefore $[y] = [y']$, and $[y]$ is in the right subtrees of both $[x]$ and $[x']$. Thus, one of $[x]$ and $[x']$ is an ancestor of the other, say $[x]$ is an ancestor of $[x']$. Since $\chi(x', j_1, \ldots, j_\ell)$ is defined, there must be some element $[u]$ on the j_1-line in $[x']$'s left subtree for which $\chi(u, j_2, \ldots, j_\ell)\downarrow$. This element is to the left of $[y]$ on the j_1-line, and, since it is in the left subtree of $[x']$, it is in $[x]$'s right subtree. This is a contradiction, because $[y]$ is the leftmost node on the j_1-line in $[x]$'s right subtree for which $\chi(y, j_2, \ldots, j_\ell)$ is defined. This completes the proof of the subclaim.

To prove the lemma from the subclaim, we again argue by induction. The base case $\ell = 0$ is again the injectivity of the identity map. For the induction step, suppose that $(x, j_2, \ldots, j_\ell) \mapsto \chi(x, j_2, \ldots, j_\ell)$ is injective. Suppose $\chi(x, j_1, \ldots, j_\ell) = \chi(x', j'_1, \ldots, j'_\ell)$ (and are defined). Let $[y]$ be the leftmost node on the j_1-line in $[x]$'s right subtree such that $\chi(y, j_2, \ldots, j_\ell)\downarrow$ and $[y']$ be the leftmost node in on the j'_1-line in $[x']$'s right subtree such that $\chi(y', j'_2, \ldots, j'_\ell)\downarrow$. So,

$$\chi(y, j_2, \ldots, j_\ell) = \chi(x, j_1, \ldots, j_\ell) = \chi(x', j'_1, \ldots, j'_\ell) = \chi(y', j'_2, \ldots, j'_\ell).$$

By the induction hypothesis, $[y] = [y']$, and $j_k = j'_k$ for $k = 2, \ldots, \ell$. Since $[y] = [y']$ and these are on the j_1- and j'_1-lines, it follows that $j_1 = j'_1$. Therefore, $\chi(x, j_1, \ldots, j_\ell) = \chi(x', j_1, \ldots, j_\ell)$. By the subclaim, it follows that $[x] = [x']$. □

Lemma 2.17. *Let S be T or one of its induced subtrees T_j.*

1. *Suppose $\chi_S(x, j_1, \ldots, j_\ell) = z$, and $0 \leq k \leq \ell$. Then there is a $[y]$ such that $\chi_S(y, j_{k+1}, \ldots, j_\ell) = z$.*
2. *For fixed $[x]$, the map $\vec{j} \mapsto \chi_S(x, \vec{j})$ is injective.*
3. *Suppose $\chi_S(x, \vec{j})\downarrow$ and that $\vec{j}' \subseteq \vec{j}$. Then $\chi_S(x, \vec{j}')\downarrow$.*
4. *Suppose $\chi_S(x, j_1, \ldots, j_\ell) = \chi_S(y, j'_1, \ldots, j'_{\ell'})$ with $[x]$ on the j_0-line, and $\ell < \ell'$. Then $j_i = j'_{i+\ell'-\ell}$ for $0 \leq i \leq \ell$; in other words, $j_0, \ldots, j_\ell$ is a suffix of $j'_1, \ldots, j'_{\ell'}$.*

Proof. In what follows, we suppress the subscript from χ_S.

Part 1. is proved by induction on k. When $k = 0$, just use $[y] = [x]$. The induction step is immediate from the definition of χ. Note that the $k = \ell$ case corresponds to $[y] = [z]$.

Suppose part 2. fails with $\chi(x, j_1, \ldots, j_\ell) = z$ and $\chi(x, j'_1, \ldots, j'_{\ell'}) = z$. By Lemma 2.16, $\ell \neq \ell'$; w.l.o.g., $\ell > \ell'$. By part 1., there is a $[y]$ on the $j_{\ell-\ell'}$-line such that $\chi(y, j_{\ell-\ell'+1}, \ldots, j_\ell) = z$. By Lemma 2.16, $[y] = [x]$, which is a contradiction.

Part 3. is proved by induction on $|\vec{j}'|$. If $\vec{j}'$ is the empty sequence, $\chi(x, \vec{j}')$ is equal to $[x]$ and hence defined. Otherwise, let k be such that j_k is the first entry in $\vec{j}'$, namely $\vec{j}'$ is the sequence $j_k, \vec{j}''$. The value $\chi(x, j_k, \vec{j}'')$ is defined if and only if there are nodes $[u]$ and $[v]$, on the j_k-line in the left and right subtrees of $[x]$ respectively, such that both $\chi(u, \vec{j}'')\downarrow$ and $\chi(v, \vec{j}'')\downarrow$. By part 1. and since $\chi(x, \vec{j})\downarrow$, there are nodes $[u']$ and $[v']$ on the j_k-line such that $\chi(u', j_k, j_{k+1}, \ldots, j_\ell)\downarrow$ and $\chi(v', j_k, j_{k+1}, \ldots, j_\ell)\downarrow$. Thus, applying the the induction hypothesis twice, $\chi(u', \vec{j}'')\downarrow$ and $\chi(v', \vec{j}'')\downarrow$. Letting $u = u'$ and $v = v'$, this proves part 3.

Part 4. follows from part 1. and Lemma 2.16. □

It is an immediate consequence of parts 2. and 3. of Lemma 2.17 that if $\chi(x, \vec{j})\downarrow$ then $|\vec{j}| \leq \log m$. In particular, we need only consider values of ℓ which are $\leq \log m$. This is because there are 2^ℓ many $\vec{j}' \subseteq \vec{j}$ and each value $\chi(x, \vec{j}')$ maps to a distinct node of the tree T, and T has only $m - 1$ internal nodes.

Lemma 2.18. *Let S be T or one of its induced subtrees T_j. For $[y]$ a node in S, let $\ell_S(y)$ be the largest value ℓ such that $y = \chi_S(x, j_1, \ldots, j_\ell)$ for some $[x], j_1, \ldots, j_\ell$.*

1. *If $y = \chi_S(x, j_1, \ldots, j_\ell)$ and $[x]$ is the leftmost node on the j-line such that $\chi(x, j_1, \ldots, j_\ell)\downarrow$, then $\ell = \ell_S(y)$.*

2. *Conversely, if $\chi_S(x, j_1, \ldots, j_{\ell_S(y)}) = y$ with $[x]$ on the j-line, then $[x]$ is the leftmost node on the j-line such that $\chi_S(x, j_1, \ldots, j_{\ell_S(y)})\downarrow$.*

Proof. To prove part 1., suppose there are $[x']$ and $j'_1, \ldots, j'_{\ell'}$ with $\ell' > \ell$ such that $\chi_S(x', j'_1, \ldots, j'_{\ell'}) = y$. By Lemma 2.17, part 4., $j_1, \ldots, j_\ell$ is a proper suffix of $j'_1, \ldots, j'_{\ell'}$ By the definition of χ, there is a $[z]$ in the left subtree of $[x']$ such that $\chi_S(z, j'_2, \ldots, j'_{\ell'})\downarrow$. Thus, by Lemma 2.17, part 1., and the suffix property, there is a node $[v]$ in the left subtree of $[x]$ such that $\chi_S(v, j_1, \ldots, j_\ell)\downarrow$. This $[v]$ is on the same j-line as $[x]$, and it is to the left of $[x]$.

For part 2., suppose there is a node $[x']$ on the j-line to the left of $[x]$ such that $\chi_S(x', j_1, \ldots, j_{\ell_S(y)})\downarrow$. Pick $[x']$ to be the rightmost such node to the left of $[x]$. Let $[z]$ be the least common ancestor of $[x]$ and $[x']$. Then $\chi_S([z], j, j_1, \ldots, j_{\ell_{S(y)}}) = y$, and this contradicts the definition of $\ell_S(y)$. □

Lemma 2.19. *For fixed $[x] \in T_j$, $[x]$ on the j-line, the function $\vec{\jmath} \mapsto \chi_{T_j}(x, \vec{\jmath})$ maps surjectively onto the internal nodes of the right subtree of T_j rooted at $[x]$.*

Proof. The left and right subtrees of $[x]$ in T_j are isomorphic by Lemma 2.14. For each $[y] \in T_j$ in the right subtree of $[x]$, let $[\widetilde{y}] \in T_j$ denote the corresponding node in the left subtree. Recall that $\widetilde{y}$ is the same as y except the $(j+1)$-st bit is changed from "1" to "0".

Fix an internal node $[z]$ in the right subtree of $[x]$. Let ℓ be the maximum value such that there exists $[y]$ in the subtree rooted at $[x]$ and exists $j_1, \ldots, j_\ell$ so that $\chi_{T_j}(y, j_1, \ldots, j_\ell) = z$. We claim that $[y] = [x]$ for the maximum value of ℓ. Suppose $[y] \neq [x]$. The node $[y]$ is on some line $j_0 < j_1$. Since $[y] \neq [x]$, $[y]$ is in $[x]$'s right subtree. Furthermore, $[\widetilde{y}]$ is on the j_0-line in $[x]$'s left subtree, and by Lemma 2.14, $\chi_{T_j}(\widetilde{y}, j_1, \ldots, j_\ell)\!\downarrow$. Let $[u]$ be the rightmost node on the j_0-line to the left of $[y]$ such that $\chi_{T_j}(u, j_1, \ldots, j_\ell)\!\downarrow$. There must exist such a $[u]$ since $[\widetilde{y}]$ has these properties. Let $[v]$ be the least common ancestor of $[u]$ and $[y]$. From the choice of $[u]$, it follows that $\chi_{T_j}(v, j_0, j_1, \ldots, j_\ell) = z$. This contradicts the maximality of ℓ. □

An example of Lemma 2.19 can be seen in Figure 2.4. Observe that every node in the right subtree of $[u]$ in the tree T_j (bottom) is of the form $\chi(u, \ldots)$.

Lemma 2.20. *If $[x] \in T_j$ and $\chi_{T_j}(x, j_1, \ldots, j_\ell)$ is defined, then $\chi(x, j_1, \ldots, j_\ell)$ is defined (in T).*

Proof. The claim is proved by induction on ℓ. The base case is trivial, since $\chi_{T_j}(x) = \chi(x) = x$. For the induction step, suppose $\chi_{T_j}(x, j_1, \ldots, j_\ell)$ is defined. The left and right subtrees of $[x]$ in T_j both contain nodes $[y]$ on the j_1-line such that $\chi_{T_j}(y, j_2, \ldots, j_\ell)\!\downarrow$. By the induction hypothesis, $\chi(y, j_2, \ldots, j_\ell)\!\downarrow$ for both $[y]$'s. Thus $\chi(x, j_1, \ldots, j_\ell)$ is defined. □

An example of Lemma 2.20 can be seen in Figure 2.4. Observe that $\chi_{T_j}(u, j_4)$ is defined, and equals z_4. So the lemma guarantees that $\chi(u, j_4)$ is defined. However, $\chi(u, j_4) = z_3 \neq \chi_{T_j}(u, j_4)$.

2.2.3 The hereditary matrix A'

We use the χ function to define a hereditary matrix associated with A.

Definition 2.21. The *hereditary matrix A' associated with A* is the 0/1 matrix with n columns such that:

- For all $x, j_0, \ldots, j_\ell$, if $[x]$ is on the j_0-line, and $\chi(x, j_1, \ldots, j_\ell)$ is defined, then there is a row in A' with 1's in columns $j_0, \ldots, j_\ell$ and 0's elsewhere.
- A' consists only of these rows, together with the zero row.

Later, Corollary 2.37 will show that if A is hereditary, then $A' = A$. For general A, we have:

Lemma 2.22. *If A' is the hereditary matrix associated with A, then A' is hereditary. Moreover, A' has the same dimensions as A.*

Proof. Let r be a row of A', with 1's in the $\ell + 1$ columns $j_0 < j_1 < \cdots < j_\ell$, and 0's in all other columns. We must show that the row obtained by replacing any 1 in r with a 0 is also in A'. This holds for the 1's in any of the columns $j_1, \ldots, j_\ell$ by part 3. of Lemma 2.17. So, consider replacing the leftmost 1, in column j_0, with a 0. By definition of A', $\chi(x, j_1, \ldots, j_\ell)$ is defined for some $[x]$ on the j_0-line. Therefore, there is a node $[y]$ on the j_1-line such that $\chi(y, j_2, \ldots, j_\ell)\downarrow$, and thus A' contains a row with 1's in columns $j_1, \ldots, j_\ell$ and 0's elsewhere.

To prove that A' has m rows, we define a bijection Θ from the non-zero rows of A' onto the internal nodes of T. Let r be a row of A' with 1's in (only) columns $j_0, \ldots, j_\ell$. Let $[x]$ be the leftmost node on the j_0-line for which $\chi(x, j_1, \ldots, j_\ell)$ is defined. Then Θ is defined by $\Theta(r) = \chi(x, j_1, \ldots, j_\ell)$.

To prove that Θ is a bijection, we show it has an inverse. Let $[y]$ be an internal node of T. Then there are $[x]$ on the j-line and $j_1, \ldots, j_{\ell(S)}$ which satisfy all the properties of Lemma 2.18. Thus, A' contains a row r with 1's in (only) columns $j, j_1, \ldots, j_{\ell_S(y)}$, and $\Theta(r) = y$. By Lemmas 2.17 and 2.18, r is the only row with $\Theta(r) = y$. □

Definition 2.23. For $0 \leq j < n$, let X_j denote the set of rows of A' with a 1 in column j.

Lemma 2.24. $|X_j| \geq |P_j|/2$.

Proof. Recall the bijection Θ defined in the proof of Lemma 2.22, which maps rows of A' to internal nodes of T. By part 4. of Lemma 2.17, if $[x]$ is on the j-line, and $\chi(x, j_1, \ldots, j_\ell)\downarrow$, then $\Theta^{-1}(\chi(x, j_1, \ldots, j_\ell)) \in X_j$. So it suffices to show that there are at least $|P_j|/2$ many nodes $[z]$ such that $\chi(x, \vec{j}) = z$ for some $[x]$ on the j-line and some sequence $\vec{j}$.

Let $[x]$ be an internal node of T on the j-line, and let S be the subtree of T rooted at $[x]$. We claim that there are at least $|P_j \cap S|/2$ many distinct nodes of the form $\chi(x, \vec{\jmath})$. This will prove the lemma, because P_j is the union over all such S's of $P_j \cap S$.

The claim is trivial if $P_j \cap S = \emptyset$. Otherwise, we have $|P_j \cap S| \geq 2$. The subtree of T_j rooted at $[x]$ has $|P_j \cap S| - 1$ many internal nodes. Thus, by Lemma 2.14, the right subtree has $(|P_j \cap S| - 2)/2 = |P_j \cap S|/2 - 1$ many internal nodes. By Lemma 2.19, it follows that there are $|P_j \cap S|/2 - 1$ many $\vec{\jmath}$'s for which $\chi_{T_j}(x, \vec{\jmath})$ is defined. By Lemma 2.20, it follows that there are at least that many $\vec{\jmath}$'s for which $\chi(x, \vec{\jmath})$ is defined (in T). Furthermore, the node $\chi(x)$ is also defined, so there are at least $|P_j \cap S|/2$ many nodes of the form $\chi(x, \vec{\jmath})$. □

The results above are summarized in the following lemma. An $m \times n$ counterexample to Frankl's Theorem for t is an $m \times n$ 0/1 matrix A of distinct rows such that $|P_j| \geq 2^t$ for all j.

Theorem 2.25. *If A is an $m \times n$ counterexample to Frankl's Theorem for t, then A' is an $m \times n$ hereditary counterexample to Frankl's Theorem for t.*

Proof. We have already shown that A' is an $m \times n$ hereditary matrix. Define P'_j for A' in the same way that P_j was defined for A. Since A' is hereditary, $|P'_j| = 2|X_j|$. That A' is a counterexample to Frankl's theorem for t follows immediately from Lemma 2.24 and the hypothesis that A is a counterexample. □

Theorem 2.25 brings us back to the usual proof of Frankl's Theorem. Namely the usual proof of Frankl's Theorem is by contradiction and constructs a hereditary matrix violating the conditions of Frankl's Theorem and then gives a simple argument based on the Kruskal-Katona Theorem to show that no such hereditary matrix exists.

We are interested in quasi-polynomial size Frege proofs of Frankl's Theorem. Section 2.3.1 will argue that Theorem 2.25 can be expressed and proved with quasi-polynomial size Frege proofs. Furthermore, Bonet, Buss, and Pitassi [9] showed that there are polynomial size Frege proofs of the Kruskal-Katona Theorem (in the form of Theorem 2.6), and from this, that there are polynomial size Frege proofs of Frankl's Theorem for hereditary matrices. These constructions, with Theorem 2.25, suffice to prove Theorem 2.8.

2.2.4 The functional Kruskal-Katona Theorem

To prove Theorem 2.9 with t constant we need to use the functional form of the Kruskal-Katona Theorem (Theorem 2.7). This allows proving Theorem 2.7 with an argument that that can be formalized with constant depth Frege proofs. In addition, we restructure the proof of Frankl's Theorem to use the pigeonhole principle instead of a counting argument; this will allow us to prove Frankl's Theorem from the Kruskal-Katona Theorem with arguments that can be formalized with constant depth Frege proofs.

We next prove Theorem 2.7. Our argument will be somewhat circular: for $m = m_0 + m_1 > 1$ with $m_0 \geq m_1$, we will assume the existence of a function

$$g_{m_0,m_1}(x) : \{0, \ldots, m-1\} \to (\{0\} \times \{0, \ldots, m_0 - 1\}) \cup (\{1\} \times \{0, \ldots, m_1 - 1\})$$

such that $g_{m_0,m_1}(a) = \langle 0, b\rangle$ implies $|a|_1 \geq |b|_1$ and such that $g_{m_0,m_1}(a) = \langle 1, b\rangle$ implies $|a|_1 \geq |b|_1 + 1$. We claim that the fact that the Kruskal-Katona Theorem is true implies that g_{m_0,m_1} exists. The range of g_{m_0,m_1} is

$$(\{0\} \times \{0, \ldots, m_0 - 1\}) \cup (\{1\} \times \{0, \ldots, m_1 - 1\})$$

and can be viewed as the set of rows of a hereditary matrix. The inequality (2.1) of the Kruskal-Katona Theorem thus implies the existence of g_{m_0,m_1}.

This circularity of using the Kruskal-Katona Theorem for its own proof should not be too disturbing however. The point is that we know the Kruskal-Katona Theorem is true. As it turns out, we only need the Kruskal-Katona Theorem for small values of m, namely the parameter m of the Kruskal-Katona Theorem will be equal to the value 2^{t-1} of Frankl's Theorem (not the value m of Frankl's Theorem!). Thus, we only need to appeal to constantly many of the functions g_{m_0,m_1}, and these can just be hard-coded into the Frege proofs.

Proof of Theorem 2.7. We argue by induction on m. Let j be the leftmost column in A with a 1 appearing column j. Let A_0 be the set of rows in A with a 0 in column j. Let A_1 be all the other rows in A. Let A_1^* be the strings in $\{0,1\}^n$ which are obtained from rows of A_1 by replacing the 1's in column j with 0's.

Let $m_0 = |A_0|$ and $m_1 = |A_1^*| = |A_1|$. By choice of j and the fact that A is hereditary, $m > m_0 \geq m_1$. By two applications of the induction hypothesis, there are

maps

$$f_0 : \{0, \ldots, m_0 - 1\} \to A_0 \text{ and } f_1 : \{0, \ldots, m_1 - 1\} \to A_1^*$$

with the property that $f_i(b) = a$ implies $|a|_1 \leq |b|_1$.

To define the function $f : \{0, \ldots, m-1\} \to A$, set

$$f(b) \;=\; \begin{cases} f_0(x) & \text{if } g_{m_0,m_1}(b) = \langle 0, x\rangle \\ f_1(x) + 2^j & \text{if } g_{m_0,m_1}(b) = \langle 1, x\rangle \end{cases}$$

where $f_1(x) + 2^j$ denotes the row $f_1(x)$, with a 1 replacing the 0 in column j. As before, columns are numbered from left to right, beginning with column $j = 0$.

To finish the proof, we claim that $f(b) = a$ implies $|a|_1 \leq |b|_1$. If $g_{m_0,m_1}(b) = \langle 0, x\rangle$, then $|a|_1 = |f_0(x)|_1 \leq |x|_1 \leq |b|_1$. And if $g_{m_0,m_1}(b) = \langle 1, x\rangle$, then $|a|_1 = |f_1(x)|_1 + 1 \leq |x|_1 + 1 \leq |b|_1$. □

Frankl's Theorem for hereditary matrices follows as an immediate consequence of the next lemma and the pigeonhole principle.

Lemma 2.26. *Let A be an $m \times n$ 0/1 hereditary matrix with distinct rows and with $|P_j| \geq 2^t$ for all j. Let D be the least common multiple of the integers $1, 2, 3, \ldots, t$. Then there is an injection from a set of size $\frac{2^t-1}{t} \cdot D \cdot n$ to a set of size $(m-1) \cdot D$.*

The least common multiple $D = D(t)$ of $1, 2, \ldots, t$ satisfies $D = O(t)$, see e.g. [37, Thm. 414].

Proof. Let Y_j be the set of rows in A with a 1 in column j. Let Y_j^* be the strings $r \in \{0,1\}^n$ obtained from rows of Y_j by replacing the 1's in column j with 0's. By hypothesis, $|Y_j^*| \geq 2^{t-1}$. The set Y_j^* is hereditary since A is. Let $Z_j \subset Y_j^*$ be a hereditary subset with $|Z_j| = 2^{t-1}$, for example the least 2^{t-1} elements of Y_j^* in the lexicographic ordering. Let $B = \{0, \ldots, 2^{t-1} - 1\}$. Define A^+ and B^+ as follows:

$$\begin{aligned} A^+ &= \{\langle a, k\rangle : a \neq \vec{0} \text{ is a row of } A \text{ and } 0 \leq k < D\} \\ B^+ &= \{\langle b, k\rangle : b \in B \text{ and } 0 \leq k < \tfrac{D}{|b|_1+1}\}. \end{aligned}$$

The matrix A is hereditary with m distinct rows, $\vec{0}$ is a row of A, and so,

$$|A^+| = (m-1) \cdot D.$$

Since B can be viewed as the set of all strings in $\{0,1\}^{t-1}$,

$$|B^+| \;=\; \sum_{i=0}^{t-1} \binom{t-1}{i} \frac{D}{i+1} \;=\; \sum_{i=1}^{t} \binom{t}{i} \frac{D}{t} \;=\; \frac{(2^t-1)\cdot D}{t}.$$

We show there is an injection from $\{0,\ldots,n-1\} \times B^+$ to A^+. By Theorem 2.7, now with $m = |B| = 2^{t-1}$, there are bijections $f_j : B \to Z_j$ so that $|f_j(b)|_1 \leq |b|_1$ for all $b \in B$.

Define $\Phi : \{0,\ldots,n-1\} \times B^+ \to A^+$ by

$$\langle j, b, k\rangle \mapsto \Big\langle f_j(b) + 2^j,\; \frac{D}{|f_j(b)|_1 + 1} \cdot j' + k \Big\rangle,$$

where j' is the number of 1's to the left of column j in $f_j(b)$. Note that the fraction is always an integer by choice of D. To see that Φ maps into A^+, observe that $f_j(b)+2^j \neq \vec{0}$ and

$$\frac{D}{|f_j(b)|_1 + 1} \cdot j' + k \;<\; D.$$

since $j' \leq |f_j(b)|_1$ and $k < \frac{D}{|b|_1+1} \leq \frac{D}{|f_j(b)|_1+1}$.

We show that Φ is injective by showing that it has an inverse. Given $\Phi(j,b,k) = \langle a, k'\rangle$, we show how to recover j, b and k. We have $a = f_j(b) + 2^j$, and $k' = \frac{D}{|a|_1} j' + k$ with j' the number of 1's to the left of column j in a.

From a, we compute $\frac{D}{|a|_1}$. Since $k < \frac{D}{|a|_1}$, we can obtain j' and k using $k' = \frac{D}{|a|_1} j' + k$. Then, from j' and a, we can recover j; and from j and a, we can recover $b = f_j^{-1}(a - 2^j)$. □

2.3 Formalization in the Frege system

This section sketches the proofs of Theorems 2.8 and 2.9 by showing how to transform the above proofs of Theorem 2.25 and Lemma 2.26 into families of quasi-polynomial size Frege proofs (respectively, polynomial size, constant depth Frege proofs).

2.3.1 Quasi-polynomial size Frege proofs

Recall that an $m \times n$ 0/1 matrix A is represented by propositional variables $p_{i,j}$ where $0 \leq i < m$ and $0 \leq j < n$. Section 2.1.2 already introduced the formulas $\textsc{Eq}(i,i',j)$, $\textsc{CardP}(j)$, and $\textsc{DistinctRows}$. We shall argue that the other concepts used in the proof of Theorem 2.3 can all be expressed by polynomial or quasi-polynomial size Boolean formulas.

First, we need formulas that define the tree T. The leaves of T are just the rows of A. Accordingly, a leaf is specified by a value i with $0 \leq i < m$. An internal node $[x]$ of T will be specified by giving a pair (i, i') of leaves, one in each of the two subtrees of $[x]$ in T. In order to make the choices for i and i' unique, we always use the least values i and i'. Accordingly, we define

$$\begin{aligned} \textsc{EqTo}(i, i', j) &:= \bigwedge_{j'=0}^{j-1} (p_{i,j'} \leftrightarrow p_{i',j'}) \\ \textsc{FirstEqTo}(i, j) &:= \bigwedge_{i'=0}^{i-1} \neg \textsc{EqTo}(i, i', j). \end{aligned}$$

For $i \neq i'$, we define $\textsc{TNodeLn}(i, i', j)$ to mean that the rows i and i' define a node $[x] \in T$ on the j-line, as:

$$\textsc{EqTo}(i, i', j) \wedge \textsc{FirstEqTo}(i, j{+}1) \wedge \textsc{FirstEqTo}(i', j{+}1) \wedge \neg p_{i,j} \wedge p_{i',j}.$$

For $i = i'$, $\textsc{TNodeLn}(i, i, n)$ is defined to be the constant *True*. For $j < n$, $\textsc{TNodeLn}(i, i, j)$ is the constant *False*. Finally, the nodes of T are defined by the pairs (i, i') satisfying

$$\textsc{TNode}(i, i') := \bigvee_{j=0}^{n} \textsc{TNodeLn}(i, i', j).$$

It is straightforward to give formulas defining structural properties of T. For instance, the node (i_2, i_2') is in the left subtree below the node (i_1, i_1') iff

$$\begin{aligned} \textsc{InLeft}(i_1, i_1'; i_2, i_2') := \\ \bigvee_{j_1 < j_2} \Big(& \textsc{TNodeLn}(i_1, i_1', j_1) \wedge \textsc{TNodeLn}(i_2, i_2', j_2) \\ & \wedge \textsc{EqTo}(i_1, i_2, j_1) \Big) \wedge \neg p_{i_2,j_1} \Big). \end{aligned}$$

$\textsc{InRight}$ is defined similarly, but with $\neg p_{i_2,j_1}$ replaced with p_{i_2,j_1}.

The rows of A are ordered by

$$\textsc{LeftOf}(i, i') := \bigvee_{j=0}^{n-1} \Big(\neg p_{i,j} \wedge p_{i',j} \wedge \textsc{EqTo}(i, i', j) \Big)$$

which expresses that row i precedes row i' in lexicographic order. Since nodes of T correspond to (prefixes of) rows of A, $\textsc{LeftOf}$ also induces a left to right ordering on T.

We now give quasi-polynomial size formulas defining the graph of the χ functions. $\textsc{Chi}(i_1, i_1'; j_1, \ldots, j_\ell; i_2, i_2')$ defines the property $\chi(x, j_1, \ldots, j_k) = z$ where (i_1, i_1')

and (i_2, i'_2) represent nodes $[x]$ and $[z]$ in T. For $\ell = 0$, $\textsc{Chi}(i_1, i'_1; i_2, i'_2)$ is true iff $i_1 = i_2$, $i'_1 = i'_2$, $i_1 \neq i'_1$, and $\textsc{TNode}(i_1, i'_1)$. Then, inductively for $\ell \geq 1$, define (the indices $k, k', k_1, k'_1, \ldots$ range over rows, i.e., are in $\{0, \ldots, m-1\}$):

$$
\begin{aligned}
&\textsc{Chi}(i_1, i'_1; j_1, \ldots, j_\ell; i, i') \ := \\
&\quad \bigvee_{k_1<k'_1} \bigvee_{k_2<k'_2} \Bigg[\textsc{TNodeLn}(k_1, k'_1, j_1) \wedge \textsc{InLeft}(i_1, i'_1; k_1, k'_1) \\
&\qquad \wedge\ \textsc{TNodeLn}(k_2, k'_2, j_1) \wedge \textsc{InRight}(i_1, i'_1; k_2, k'_2) \\
&\qquad \wedge\ \textsc{Chi}(k_2, k'_2; j_2, \ldots, j_\ell; i, i') \wedge \bigvee_{k<k'} \textsc{Chi}(k_1, k'_1; j_2, \ldots, j_\ell; k, k') \\
&\qquad \wedge\ \neg\Big(\bigvee_{k_3<k'_3} \big[\textsc{TNodeLn}(k_3, k'_3, j_1) \wedge \textsc{InRight}(i_1, i'_1; k_3, k'_3) \\
&\qquad\qquad \wedge \textsc{LeftOf}(k_3, k_2) \wedge \bigvee_{k<k'} \textsc{Chi}(k_3, k'_3; j_2, \ldots, j_\ell; k, k') \big] \Big) \Bigg].
\end{aligned}
$$

The Chi formulas are readily modified to define the functions χ_T, for $T = T_j$. The leaves of T that are in P_j are definable by letting $\textsc{Pj}(i, j)$ be $\bigvee_{i' \neq i} \textsc{Eq}(i, i', j)$. The formula $\textsc{TjNode}(i, i', j)$ that defines the property of (i, i') being a node in T_j can be defined similarly to $\textsc{TNode}(i, i')$ but restricting to leaves that lie in T_j. The χ_{T_j} function can be defined similarly to the χ function by a formula $\textsc{ChiTj}(i_1, i'_1; j_1, \ldots, j_\ell; i, i'; j)$ which has j as an extra parameter. We leave the details of formalizing TjNode and ChiTj to the reader.

All of the formulas defined above except Chi and ChiTj are constant depth and have polynomial size (in m, n). The formulas Chi and ChiTj, however, are defined inductively on ℓ, and have depth $O(\ell)$ using AND and OR gates with fan-in as large as n or m^2 (for example, the AND gate in FirstEqTo and the big OR gates in the definition of Chi, respectively). Thus, these formulas have size bounded by $(m+n)^{O(\ell)} = (m+n)^{O(\log m)}$. In other words, Chi and ChiTj are quasi-polynomial size formulas, and the χ function is NC^2-definable. In fact, since the values of $j_1, \ldots, j_\ell$ are fixed, the Chi and ChiTj have polynomial size, unbounded fan-in *circuits* of depth $O(\ell)$, so (the graph of) the function χ is even in AC^1.

The number of different formulas Chi and ChiTj that need to be constructed is bounded by $m^4 n^{O(\log m)}$. This is because the Chi formula has four parameters i_1, i'_1, i, i' that range over the m rows of A and ℓ many parameters $j_1, \ldots, j_\ell$ ($\ell + 1$ many for ChiTj) that range over the n columns of A. The value ℓ is bounded by $\log m$ by part 3.

of Lemma 2.17 and the injectivity of the χ function (Lemma 2.16). This means there are quasi-polynomially many formulas $\textsc{Chi}(\cdots)$ and $\textsc{ChiTj}(\cdots)$.

We have shown how to express concepts such as the trees T and T_j and the χ and χ_j functions with quasi-polynomial size formulas. It is now straightforward to formulate and prove the propositional translations of Lemmas 2.14-2.24 and Theorem 2.25 with quasi-polynomial size Frege proofs. Indeed the proofs of these lemmas are all very concrete and constructive, and they are readily translated into propositional logic.

Although it is left to the reader to verify that the translations to propositional logic can be carried out straightforwardly, we do mention a couple points. First, as usual, the propositional proofs replace the use of induction with a "brute-force induction" or "exhaustive" enumeration of cases. For example, the propositional translation of Lemma 2.16 becomes the propositional formulas

$$\neg\big(\textsc{Chi}(i_1, i_1'; j_1, \ldots, j_\ell; i_3, i_3') \wedge \textsc{Chi}(i_2, i_2'; j_1', \ldots, j_\ell'; i_3, i_3')\big)$$

for all choices of sequences $i_1, i_1', j_1, \ldots, j_\ell$ not identical to $i_2, i_2', j_1', \ldots, j_\ell'$. The propositional proof derives all these statements, for all such values, successively for ℓ equal to 0 up to $\log m$. Second, note that the hereditary matrix A', as defined in Definition 2.21 has quasi-polynomially many possible rows. The proof of Theorem 2.25 gives an injection from the rows of A' to the rows of A, and, with this injection, propositional proofs can be used to bound the number of rows of A'.

As already discussed, [9] showed that polynomial size Frege proofs can prove the hereditary case of Frankl's Theorem. This completes the proof of Theorem 2.8 that the propositional translations of Frankl's Theorem have quasi-polynomial size Frege proofs.

2.3.2 Polynomial size constant depth proofs

For t fixed, Theorem 2.9 asserts the existence of polynomial size, constant depth Frege proofs of Frankl's Theorem. The first difficulty is that the predicates $\textsc{Chi}$ and $\textsc{ChiTj}$ are defined with formulas of depth $O(\log m)$, since the function $\chi(x, j_1, \ldots, j_\ell)$ is invoked with ℓ as large as $\log m$. To avoid this, we modify Definition 2.21 of the hereditary matrix A' to restrict attention to rows that have at most t many 1's, and we prove an analogue of Lemma 2.22.

Definition 2.27. The matrix $A'_{\leq t}$ is the 0/1 matrix that contains as rows exactly those rows of A' with no more than t many 1's.

Lemma 2.28. *$A'_{\leq t}$ is an $m' \times n$ hereditary matrix, where $m' \leq m$ and $m' < n^t$.*

Proof. The fact that $A'_{\leq t}$ is hereditary follows immediately by the same argument that showed A' is hereditary. The fact that $m < n^t$ follows from the fact that there are fewer than n^t many subsets of $\{1, \ldots, n\}$ of size $\leq t$. Finally, $m' \leq m$ is proved by showing, as in the proof of Lemma 2.22, that the function Θ is an injective map from the nonzero rows of $A'_{\leq t}$ into the internal nodes of T. (It may not be surjective, however.) □

We also need to modify the definition of X_j, and prove an analogue of Lemma 2.24.

Definition 2.29. For $0 \leq j < n$, let $X_{j,\leq t}$ denote the set of rows of $A'_{\leq t}$ with a 1 in column j.

Lemma 2.30. $|X_{j,\leq t}| \geq \min\{|P_j|/2, 2^{t-1}\}$.

Proof. This is similar to the proof of Lemma 2.24, but now we reason with only the rows of $A'_{\leq t}$, not the rows of A'. The argument splits into two cases. First suppose there is some row r of $X_{j,\leq t}$ that contains t many 1's. There are 2^{t-1} many rows that can be obtained from r by deleting 1's from columns other than column j. These all lie in $X_{j,\leq t}$, so $|X_{j,\leq t}| \geq 2^{t-1}$.

Second suppose that all rows in $X_{j,\leq t}$ contain fewer than t many 1's. Then the argument used in the proof of Lemma 2.24 applies to show that $|X_{j,\leq t}| \geq |P_j|/2$. □

Similarly to Theorem 2.25, we obtain the following.

Theorem 2.31. *If A is an $m \times n$ counterexample to Frankl's Theorem for t. Then $A'_{\leq t}$ is an $m' \times n$ hereditary counterexample to Frankl's Theorem for t with $m' \leq m$.*

We claim that, using Lemmas 2.28 and 2.30 and Theorem 2.31, the entire proof of Frankl's Theorem for constant t can by formalized by constant depth, polynomial size Frege proofs in which all formulas have depth $O(t)$. We sketch the proof of this claim below.

First, the basic properties of the tree T, using formulas TNodeLn, TNode, InRight, etc., can be expressed with constant depth, polynomial size formulas. Second, counting sets up to a constant cardinality, say $s = O(t)$ or $s = O(2^t)$, can be done with polynomial size formulas (for fixed t). To see this, let $\phi_1, \ldots, \phi_n$ be formulas. The condition that at least s of the ϕ_i's are true can be expressed by letting $\mathcal{I}$ range

over subsets of $\{1, \ldots, n\}$ of size exactly s, and writing $\bigvee_{\mathcal{I}} \bigwedge_{i \in \mathcal{I}} \phi_i$. This allows the statement $\textsc{CardP}(j) < 2^t$ to be expressed by a constant depth, polynomial size formula. Therefore, for fixed t, Frankl's Theorem can be stated with constant depth, polynomial size formulas.

Thirdly, as can be straightforwardly checked, the predicates $\textsc{Chi}$ and $\textsc{ChiTj}$, when retricted to $\ell \leq t$ can be expressed by Boolean formulas of depth $O(t)$ and size $n^{O(t)}$.

These considerations allow Lemmas 2.14-2.20, 2.28 and 2.30 and Theorem 2.31 to be expressed with constant depth, polynomial size Boolean formulas, and proved with constant depth, polynomial size Frege proofs. The assertion "$m' \leq m$" of Lemma 2.28 and Theorem 2.31 cannot be expressed explicitly as constant depth polynomial size formulas. Instead, it is formalized by defining an injection from the rows of $A'_{\leq t}$ into the rows of A. Recall that Θ is an injection from the nonzero rows of $A'_{\leq t}$ into the internal nodes of T. The rows of A are the same as the leaves of T, and it is easy to explicitly define an injection between the internal nodes of T and the leaves of T, omitting one leaf (say, the leftmost leaf). By composition, there is an injection from the rows of $A'_{\leq t}$ into the rows of A. Constant depth, polynomial size Frege proofs can define this injection and prove its properties.

Finally, we need to argue that the arguments in Section 2.2.4 can be formalized as polynomial size, constant depth Frege proofs.

We sketch how to formalize Section 2.2.4's proof of Theorem 2.7 as polynomial size, constant depth Frege proofs, when m is a constant.[1] The difficulty is that the proof given above defines the function f by induction in a way that is not readily formalizable with constant depth formulas. However, the key point is that f is a map from $\{0, \ldots, m-1\}$ onto the rows of A, and since m is constant, there are only finitely many possibilities for f. It is now convenient to work with the inverse of f, which we denote F. Theorem 2.7 is proved by using "brute-force" induction, for ℓ ranging from n down to 1 to prove the following assertion. We let $E_{\ell,i}$ denote the set of rows of A that agree with row i in their first ℓ entries. We let $r_{\ell,i}$ be the last $n - \ell$ columns of row i (that is, discarding the first ℓ columns).

There is a function F_ℓ (not necessarily injective) from the m many rows of A

[1] Recall that the variable m is used in different ways for Frankl's Theorem and the Kruskal-Katona Theorem. In our applications, the value for the Kruskal-Katona Theorem is $m = 2^{t-1}$, and this is constant since t is.

into $\{0, \ldots, m-1\}$ such that: for each row i, $0 \leq i < m-1$, (a) $|F(i)|_1 \geq |r_{\ell,i}|_1$, and (b) F_ℓ restricted to $E_{\ell,i}$ is a bijection onto $\{0, \ldots, |E_{\ell,i}| - 1\}$.

This assertion is expressible as a polynomial size, constant depth formula, since m is constant and there are only finitely many possibilities for F_ℓ. Furthermore, the argument from the proof of Theorem 2.7 readily shows that the existence of F_ℓ follows from the existence of the F_k's for $k > \ell$ (and from the finitely many functions g_{m_0,m_1}). Finally, the f of Theorem 2.7 is just the inverse of F_0.

The proof of Lemma 2.26 is straightforward to formalize with polynomial size, constant depth Frege proofs. This follows from the facts that, since t is constant, the value $D = D(t)$ is a fixed constant, and that the proof of Lemma 2.26 gives an explicit construction of the injection and only involves counting up to a constant. This completes the proof of Theorem 2.9.

2.4 Equivalent definitions of the hereditary matrix

The usual proof of Frankl's Theorem uses a much simpler construction of a hereditary counterexample matrix than the χ function procedure of Definition 2.21. The construction starts with a matrix A which, by hypothesis, violates Frankl's Theorem. If A is not hereditary, there is some entry 1 in A such that if this 1 is replaced with a 0 the matrix still contains distinct rows. A hereditary counterexample matrix is formed by iteratively replacing such 1's with 0's until a hereditary matrix is obtained. It is easy to verify that this process preserves the property that the matrix violates Frankl's Thoerem. This construction as described in [29, 9] did not specify the order in which 1's are to be replaced with 0's. We shall prove that there is some order for changing 1's to 0's such that this construction yields the same matrix as our matrix A' from Section 2.2.3.

The next definition describes the effect of replacing all 1's in column j with 0's which do not identify any pair of rows. Recall that if $r \in \{0,1\}^n$ is a row with a 1 in column j, then $r - 2^j$ represents the same row but with that 1 replaced with 0. Throughout this section, let A be an $m \times n$ 0/1 matrix with distinct rows.

Definition 2.32. Let $0 \leq j < n$, and let A_0, respectively A_1, denote the set of rows of A with a 0, respectively a 1, in column j. The *downshift* of A in column j is the matrix $\mathrm{DownShift}(A, j)$ containing the rows

$$A_0 \cup \{r : r \in A_1,\, r - 2^j \in A_0\} \cup \{r - 2^j : r \in A_1,\, r - 2^j \notin A_0\}.$$

Definition 2.33. Let $0 \leq j < n$. Then A is *hereditary in column j* if, for any row r of A with a 1 in column j, $r - 2^j$ is also a row in A.

By definition, the matrix DownShift(A, j) is hereditary in column j.

Definition 2.34. Define the sequence of matrices $A^{(n)}$, $A^{(n-1)}, \ldots, A^{(1)}, A^{(0)}$ by letting $A^{(n)}$ equal A, and $A^{(j)}$ equal DownShift$(A^{(j+1)}, j)$ for each $j < n$.

Lemma 2.35. *The matrix $A^{(j)}$ is hereditary in columns $j, j+1, \ldots, n-1$. In particular, $A^{(0)}$ is hereditary.*

Proof. The proof is by induction on $j = n, \ldots, 1, 0$. The base case of $j = n$ is trivial. For the induction step, suppose $A^{(j+1)}$ is hereditary in columns $j+1, \ldots, n-1$. By the definition of DownShift, $A^{(j)}$ is hereditary in column j, so we need to prove that it is hereditary in all columns $k > j$. Consider a row $w = u1z$ is in $A^{(j)}$, where $|u| = k > j$. We need to prove that $u0z$ is a row of $A^{(j)}$.

Write u in the form xiy where $|x| = j$ and $i \in \{0, 1\}$ and $|y| = k-j-1$. Thus w is equal to $xiy1z$. First suppose $i = 1$ and $w = x1y1z$. Since $x1y1z$ is a row of $A^{(j)}$ and has a 1 in column j, both $x1y1z$ and $x0y1z$ are present as rows in $A^{(j+1)}$. Since $A^{(j+1)}$ is hereditary in column k, $x1y0z$ and $x0y0z$ are rows of $A^{(j+1)}$. Thus, by the definition of DownShift, $x1y0z = u0z$ is also a row of $A^{(j)}$.

Otherwise, $i = 0$ and $w = x0y1z$. If w is also a row of $A^{(j+1)}$, then since $A^{(j+1)}$ is hereditary in column k, $x0y0z$ is also a row of $A^{(j+1)}$. Therefore, $x0y0z = u0z$ is a row of $A^{(j)}$. Otherwise, $x1y1z$ is a row of $A^{(j+1)}$, but $x0y1z$ is not. Since $A^{(j+1)}$ is hereditary in column k, $x1y0z$ is a row of $A^{(j+1)}$. Therefore, by the definition of DownShift, $x0y0z = u0z$ is a row of $A^{(j)}$. □

Lemma 2.36. *Let A be hereditary in columns $j, \ldots, n-1$, let $[x]$ be a node of T on the j_0-line, $j \leq j_0$, and let u be the string*

$$x10^{j_1-j_0-1}10^{j_2-j_1-1}1 \cdots 10^{j_\ell-j_{\ell-1}-1}10^{n-j_\ell-1}. \tag{2.2}$$

In other words, u is x plus 1's in columns $j_0, \ldots, j_\ell$. Then $\chi(x, j_1, \ldots, j_\ell)\downarrow$ iff u is a row of A.

Proof. Suppose $\chi(x, j_1, \ldots, j_\ell)\downarrow$. We argue by induction on ℓ. For the base base, $\ell = 0$, we have u equal to $x10^{n-j_0-1}$ and since x is a maximal representative for $[x]$, A has a row $x1w$ for some $w \in \{0, 1\}^{n-j_0-1}$ By the hereditary property, u is also a row of A.

For the induction step, suppose $\ell > 0$. Then there is a $[y]$ in the right subtree of $[x]$ on the j_1-line such that $\chi(y, j_2, \ldots, j_\ell)\downarrow$. We have $y = x1w$ for some $w \in \{0,1\}^{j_1 - j_0 - 1}$. By the induction hypothesis,

$$x1w10^{j_2 - j_1 - 1}1 \cdots 10^{j_\ell - j_{\ell-1} - 1}10^{n - j_\ell - 1}$$

is a row of A. Thus, by the hereditary property, u is also a row of A.

For the converse, suppose u is a row of A. We again argue by induction on ℓ. First suppose $\ell = 0$. By the hereditary property, $x0^{n-j_0-1}$ is a row of A. Thus, $[x]$ exists as an internal node of T, and we have $\chi(x)\downarrow$. Second, suppose $\ell > 0$. Let $y_0 = x0^{j_1 - j_0}$ and $y_1 = x10^{j_1 - j_0 - 1}$. Using the hereditary property of A, both $[y_0]$ and $[y_1]$ exist as nodes of A. Using the hereditary property of A with respect to the row u, and applying the induction hypothesis twice, both $\chi(y_0, j_2, \ldots, j_\ell)\downarrow$ and $\chi(y_1, j_2, \ldots, j_\ell)\downarrow$. Since $[y_0]$ and $[y_1]$ lie on the j_1-line in the left and right subtrees of $[x]$, respectively, $\chi(x, j_1, \ldots, j_\ell)\downarrow$. □

Corollary 2.37. *If A is hereditary, and A' is the hereditary matrix associated with A, then $A' = A$.*

Proof. If v is a non-zero row of A' with 1's in columns $j_0, \ldots, j_\ell$ and 0's elsewhere, then by the definition of A', there is a node $[x]$ on the j_0-line such that $\chi(x, j_1, \ldots, j_\ell)\downarrow$. By Lemma 2.36, A contains a row of the form (2.2) with 1's in columns $j_0, \ldots, j_\ell$. Since A is hereditary, v is also a row of A. Therefore every row of A' is a row of A, and since the matrices have the same number of rows, $A' = A$. □

Lemma 2.38. *Let $T^{(j+1)}$ and $T^{(j)}$ be the prefix trees for $A^{(j+1)}$ and $A^{(j)}$. Let $[x]$ be a node of $T^{(j+1)}$ on the j_0-line with $\chi_{T^{(j+1)}}(x, j_1, \ldots, j_\ell)\downarrow$. Then there exists a node $[x']$ of $T^{(j)}$ on the j_0-line such that $\chi_{T^{(j)}}(x', j_1, \ldots, j_\ell)\downarrow$. Moreover, if $j_0 \leq j$, then we can take $[x'] = [x]$.*

Proof. If $\ell = 0$, then the claim is trivial, so assume that $\ell > 0$. The proof is by induction on the number of elements of $j_0, \ldots, j_\ell$ that are less than or equal to j. For the first base case (when $j_0 > j$), we have $j_0 \geq j + 1$, so Lemmas 2.35 and 2.36 and the fact that $\chi_{T^{(j+1)}}(x, j_1, \ldots, j_\ell)\downarrow$ imply that the u of Equation (2.2) is a row of $A^{(j+1)}$. Let x' be x, except modified to have a 0 in column j. By definition of DownShift,

$$x'10^{j_1 - j_0 - 1}10^{j_2 - j_1 - 1}1 \cdots 10^{j_\ell - j_{\ell-1} - 1}10^{n - j_\ell - 1}$$

is a row of $A^{(j)}$. By Lemmas 2.35 and 2.36, $\chi_{T^{(j)}}(x', j_1, \ldots, j_\ell)\downarrow$.

The second base case is when $j_0 = j$. Since $\chi_{T^{(j+1)}}(x, j_1, \ldots, j_\ell)\downarrow$, there are nodes $[y_0]$ and $[y_1]$ in $[x]$'s left and right subtrees on the j_1-line in $T^{(j+1)}$ such that $\chi_{T^{(j+1)}}(y_i, j_2, \ldots, j_\ell)\downarrow$ for $i = 0, 1$. We have $y_0 = x0w_0$ and $y_1 = x1w_1$ for some strings w_0, w_1 of length $j_1 - j_0 - 1$. By Lemma 2.36, $A^{(j+1)}$ contains the rows $u_i = xiw_i1\vec{0}\cdots\vec{0}1\vec{0}$ for $i = 0, 1$, where the indicated 1's are in columns $j_1, \ldots, j_\ell$. $A^{(j+1)}$ is hereditary in columns $j+1, \ldots, n-1$, therefore the presence of the row u_1 implies that $v = x1\vec{0}1\vec{0}1\cdots\vec{0}1\vec{0}$ with the indicated 1's in columns $j_0, \ldots, j_\ell$ is a row of $A^{(j+1)}$. Similarly the presence of u_0 implies that $v - 2^j$ is a row of $A^{(j+1)}$. Because v and $v - 2^j$ are rows of $A^{(j+1)}$, by definition of DownShift, v is a row of $A^{(j)}$. So by Lemmas 2.35 and 2.36, $\chi_{T^{(j)}}(x, j_1, \ldots, j_\ell)\downarrow$.

In the final base case, $j_0 < j < j_1$. Since $\chi_{T^{(j+1)}}(x, j_1, \ldots, j_\ell)\downarrow$, there are nodes $[y_0]$ and $[y_1]$ in $[x]$'s left and right subtrees on the j_1-line in $T^{(j+1)}$ such that $\chi_{T^{(j+1)}}(y_i, j_2, \ldots, j_\ell)\downarrow$ for $i = 0, 1$. So by Lemmas 2.35 and 2.36,

$$y_i 10^{j_2 - j_1 - 1} 10^{j_3 - j_2 - 1} 1 \cdots 10^{j_\ell - j_{\ell-1} - 1} 10^{n - j_\ell - 1}$$

for $i = 0, 1$ are rows of $A^{(j+1)}$. Let y_i' be y_i modified to have a 0 in column j. By definition of DownShift,

$$y_i' 10^{j_2 - j_1 - 1} 10^{j_3 - j_2 - 1} 1 \cdots 10^{j_\ell - j_{\ell-1} - 1} 10^{n - j_\ell - 1}$$

for $i = 0, 1$ are elements of $A^{(j)}$. By Lemmas 2.35 and 2.36 again, $\chi_{T^{(j)}}(y_i', j_2, \ldots, j_\ell)\downarrow$ for $i = 0, 1$. Since $j_0 < j$, it follows that $[y_0']$ and $[y_1']$ are in the left and right subtrees of $[x]$, therefore $\chi_{T^{(j)}}(x, j_1, \ldots, j_\ell)\downarrow$.

For the induction step we have $j_0 < j_1 < j$. Since $\chi_{T^{(j+1)}}(x, j_1, \ldots, j_\ell)\downarrow$, it follows that $T^{(j+1)}$ has nodes $[y_0]$ and $[y_1]$ on the j_1-line in $[x]$'s left and right subtrees such that $\chi_{T^{(j+1)}}(y_i, j_2, \ldots, j_\ell)\downarrow$ for $i = 0, 1$. By the "moreover" clause of the induction hypothesis, $\chi_{T^{(j)}}(y_i, j_2, \ldots, j_\ell)\downarrow$ for $i = 0, 1$. Thus $\chi_{T^{(j)}}(x, j_1, \ldots, j_\ell)\downarrow$. □

Recall that Definition 2.21 defined the matrix A' associated with A.

Theorem 2.39. $A^{(0)} = A'$.

Proof. Define $(A^{(j)})'$ to be the hereditary matrix associated with $A^{(j)}$ in the sense of Definition 2.21. By Lemma 2.38, Definition 2.21, and the fact that $(A^{(j+1)})'$ and $(A^{(j)})'$ both have m rows, $(A^{(j+1)})' = (A^{(j)})'$. Therefore, $(A^{(0)})' = (A^{(n)})' = A'$. Moreover, by Corollary 2.37, since $A^{(0)}$ is hereditary, $A^{(0)} = (A^{(0)})' = A'$. □

Chapter 2, in full, is a reprint of material that will appear in the *Journal of Symbolic Logic*. Aisenberg, James; Bonet, Maria L.; Buss, Sam. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Short Proofs of the Kneser-Lovász Coloring Principle

3.1 Introduction

This paper discusses proofs of Lovász's theorem about the chromatic number of Kneser graphs and the proof complexity of propositional translations of the Kneser-Lovász theorem. Our main results give a new proof of the Kneser-Lovász theorem, which, for fixed parameter k, uses a simple counting argument instead of the topological arguments used in prior proofs, for all but finitely many cases. These arguments can be formalized in propositional logic to give polynomial size extended Frege proofs and quasi-polynomial size Frege proofs.

The proof complexity of Frege and extended Frege systems was first studied by Cook and Reckhow [24, 25] and Statman [60]. Frege systems (denoted $\mathcal{F}$) are sound and complete proof systems for propositional logic with a finite set of schemes for axioms and inference rules. The typical example is a "textbook style" propositional proof system using modus ponens as its only rule of inference. In fact, all Frege systems are equivalent to this system [25]. Extended Frege systems (denoted $e\mathcal{F}$) are Frege systems augmented with the extension rule, which allows variables to abbreviate complex formulas. The reader unfamiliar with Frege systems can consult the surveys [9, 18, 19, 25, 45, 59] for more information.

The *size* of a Frege or extended Frege proof is the number of symbols in the proof. A proof system $\mathcal{P}_1$ *simulates* a proof system $\mathcal{P}_2$ if and only if there is a polynomial

$p(n)$ such that, for any propositional formula φ, if φ has a $\mathcal{P}_2$-proof of size n, then φ has a $\mathcal{P}_1$-proof of size $\leq p(n)$. Also, $\mathcal{P}_1$ *quasi-polynomially simulates* $\mathcal{P}_2$ if and only if there is a $k > 0$ such that, if φ has a $\mathcal{P}_2$-proof of size n then φ has a $\mathcal{P}_1$-proof of size $2^{(\log n)^k}$. It is trivial that extended Frege systems simulate Frege systems.

It is generally conjectured that the extension rule can provide substantial shortening of proof length, and therefore that Frege systems do not (quasi-polynomially) simulate extended Frege systems. The intuition is that Frege proofs are able to reason using Boolean formulas; whereas extended Frege proofs can reason using Boolean circuits. (See [41] for a formalization of this intuition.) Boolean formulas are conjectured to require exponential size to simulate Boolean circuits. There is no known direct connection to proof complexity, but it is generally conjectured by analogy that there is an exponential separation between the sizes of Frege proofs and extended Frege proofs, and thus that Frege systems do not (quasi-polynomially) simulate extended Frege systems.

Bonet, Buss, and Pitassi [9] systematically looked for combinatorial tautologies that could be candidates for exponentially separating proof sizes for Frege and extended Frege systems. Surprisingly, they found only a small number. The first candidates were based on linear algebra, including the Oddtown theorem, the Graham–Pollack theorem, the Fisher Inequality, the Ray-Chaudhuri–Wilson theorem, and the $AB = I \Rightarrow BA = I$ tautology (the last was suggested by S. Cook). The remaining candidate was Frankl's theorem on the trace of sets. All of these principles were shown to have polynomial size extended Frege proofs, but it was open whether they had polynomial size Frege proofs.

Hrubeš and Tzameret [39] recently showed that the five tautologies based on linear algebra have quasi-polynomial size Frege proofs by showing that there are quasi-polynomial size definitions of determinants whose properties can be established by quasi-polynomial Frege proofs (as was conjectured by [9]). Subsequently, Aisenberg, Bonet, and Buss [1] showed that Frankl's theorem also has quasi-polynomial size Frege proofs. With these results, none of the principles considered by Bonet-Buss-Pitassi provide an exponential separation of Frege and extended Frege systems.

An earlier combinatorial candidate was the pigeonhole principle, introduced by Cook and Reckhow [25]. They showed this has polynomial size extended Frege proofs. Buss [16] later showed this also has polynomial size Frege proofs. Buss's proof was based on "counting", and showed that Frege proofs can use polynomial size formulas (based on carry-save addition) to define sizes of sets, and can reason about sizes effectively.

Carry-save addition also allows Frege systems to reason about integer multiplication and about adding vectors of integers. The ability of Frege proofs to "count" and to reason about sizes of sets will be important for our Frege proofs of the Kneser-Lovász theorem. The counting proofs were quite different than Cook and Reckhow's inductive proofs of the pigeonhole principle, so these were sometimes taken as evidence that Frege systems do not (quasi-polynomially) simulate Frege proofs. However, [14] recently showed that Cook and Reckhow's inductive proofs can be reformulated as quasi-polynomial size Frege proofs.

Another class of candidates is based on consistency statements. We write $\text{Con}_{\mathcal{P}}(n)$ for the propositional statement expressing the condition that the proof system $\mathcal{P}$ does not have a proof of $p \wedge \neg p$ of size $\leq n$. For "natural" systems $\mathcal{P}$ (including Frege and extended Frege systems), the formula $\text{Con}_{\mathcal{P}}(n)$ has size polynomially bounded by n (e.g., [22, 17]). Propositional consistency statements have been studied for first-order systems by Pudlák [54, 55] and Friedman [unpublished]. Pudlák showed that axiomatizable theories of arithmetic have polynomial size (first-order) proofs of their partial consistency statements; Pudlák and Friedman independently proved polynomial lower bounds as well. Cook [22] showed that an extended Frege system has polynomial size proofs of its own partial consistency statements $\text{Con}_{e\mathcal{F}}(n)$. Buss [17] proved similarly that a Frege system has polynomial size proofs of its partial consistency statements $\text{Con}_{\mathcal{F}}(n)$.

It also follows from [17] that Frege systems (quasi-)polynomially simulate extended Frege systems iff there are (quasi-)polynomial size Frege proofs of $\text{Con}_{e\mathcal{F}}(n)$. In addition, $\text{Con}_{e\mathcal{F}}(n)$ is a "logical" principle not really a "combinatorial" principle.[1] For these reasons, partial consistency statements such as $\text{Con}_{e\mathcal{F}}(n)$ do not serve as the kinds of candidates for separating Frege and extended Frege system that we are seeking.

Other candidates for exponentially separating Frege and extended Frege systems arose from the work of Kołodziejczyk, Nguyen, and Thapen [44] in the setting of bounded arithmetic [15]. These include various forms of the local improvement principles LI, $\text{LI}_{\log}$ and LLI. The results of [44] showed that the LI principle is many-one complete for the NP search problems of V_2^1; it follows that LI is equivalent to partial consistency statements for extended Frege systems. Beckmann and Buss [8] subsequently proved that $\text{LI}_{\log}$ is provably equivalent (in S_2^1) to LI and that the linear local improvement

[1] However, see Avigad [6] for a combinatorial version of $\text{Con}_{e\mathcal{F}}(n)$.

principle LLI is provable in U^1_2. The LLI principle thus has quasi-polynomial size Frege proofs. Combining the results of [8, 44] shows that $\mathrm{LI}_{\log}$ and LLI are many-one complete for the NP search problems of V^1_2 and U^1_2, respectively, and thus equivalent to partial consistency statements for extended Frege and Frege systems, respectively.

Thus, apart from partial consistency statement, none of the above principles serve as combinatorial candidates for showing that Frege systems do not quasi-polynomially simulate extended Frege systems.

A new candidate based on the Kneser-Lovász theorem was recently proposed by Istrate and Crãciun [40]. As defined below, the Kneser-Lovász theorem gives a lower bound on the chromatic of the (n, k)-Kneser graphs. Istrate and Crãciun showed that the $k = 3$ case of these tautologies have polynomial size extended Frege proofs, but left open whether they have (quasi-)polynomial size Frege proofs. However, the main results of the present paper show that, for any fixed $k \geq 1$, the Kneser-Lovász tautologies have quasi-polynomial size Frege proofs. Thus these also do not give an exponential separation of Frege from extended Frege systems.

With these last results, we have few remaining combinatorial candidates for showing Frege systems do not quasi-polynomially simulate extended Frege systems. One remaining candidate is tautologies based on the Rectangular Local Improvement principles, RLI_k, of Beckmann-Buss [8] for fixed $k \geq 2$. The only other combinatorial candidate we know of is introduced in Section 3.6 below. This is the $k = 1$ case of the "truncated Tucker lemma". Theorem 3.26 shows it has polynomial size extended Frege proofs; however, we have been unable to show that it has quasi-polynomial size Frege proofs.

The outline of the paper is as follows. First, in Section 3.2 we define the (n, k)-Kneser graphs and state Lovász's theorem about their chromatic numbers. Theorems 3.4 and 3.5 state our main results about Frege and extended Frege proofs of that theorem. Section 3.3 gives an informal ("mathematical") proof of the Kneser-Lovász theorem using a new proof method based on a simple counting argument. Prior proofs used, at least implicitly, a topological fixed-point lemma. The most combinatorial proof is by Matoušek [49] and is inspired by the octahedral Tucker lemma; see also Ziegler [63]. Our new proofs mostly avoid topological arguments and use a counting argument instead. The counting arguments are used to prove the existence of "star-shaped" color classes. These counting arguments can be formalized with Frege proofs. For the Kneser-Lovász

theorem, the counting arguments reduce the general case to "small" instances of size $n \leq 2k^4$. For fixed k, there are only finitely many small instances, and they can be verified by exhaustive enumeration. As we shall see, this leads to polynomial size extended Frege proofs, and quasi-polynomial size Frege proofs for the Kneser-Lovász principles. Sections 3.3.1 and 3.3.2 give two "mathematical" versions of the counting proofs, which will be formalized as extended Frege proofs and Frege proofs (respectively). Section 3.3.3 is a short diversion and considers whether there are colorings of the Kneser graphs with many non-star-shaped color classes.

Section 3.4 discusses some of the details of formalizing the arguments in Section 3.3 in the Frege and extended Frege systems, establishing our two main theorems. We focus on expressing the concepts described in Section 3.3 in propositional logic, and we only sketch some of the details of how Frege systems can prove properties of these concepts.

The proofs of the Kneser-Lovász theorem in Sections 3.3 and 3.4 reduce the general case of the Kneser-Lovász theorem to finitely many base cases, which are then handled by exhaustive enumeration. It would be interesting to give a uniform proof that does not need to handle the base cases in this way. Motivated by this, Section 3.5 defines new "truncated" forms of the Tucker lemma. These truncated Tucker lemmas can be be expressed as families of polynomial size propositional tautologies. The octahedral Tucker lemma, on the other hand, can only be expressed by exponential size formulas. Matoušek showed that the Kneser-Lovász theorem follows from the Tucker lemma. We refine this by showing that the Tucker lemma implies the two truncated Tucker lemmas, the two versions of the truncated Tucker lemma are equivalent, and that the truncated Tucker lemmas imply the Kneser-Lovász theorem. Since the truncated Tucker lemmas can be expressed as polynomial size tautologies, it is natural to ask about their proof complexity in (extended) Frege systems. Section 3.6 proves that the $k = 1$ cases of the truncated Tucker lemmas have polynomial size extended Frege proofs. It is open whether these have (quasi-)polynomial size Frege proofs. Thus, this is a candidate for separating the Frege and extended Frege systems. Likewise, it is open whether the truncated Tucker lemmas for $k > 1$ have subexponential size extended Frege proofs. It is tempting to try to modify the combinatorial proof of the Tucker lemma by Freund and Todd [32] (see also Matoušek [49]), but we have been unable to express this argument with polynomial size extended Frege proofs. Freund and Todd's argument uses a version of the parity

principle PPA [53]. The difficulty with translating these arguments to extended Frege proofs is that they apply the parity principle on exponentially large graphs.

3.2 The Kneser-Lovász Principle and Statement of the Main Theorems

The (n, k)-Kneser graph is defined to be the undirected graph whose vertices are the k-subsets of $\{1, \ldots, n\}$; there is an edge between two vertices iff those vertices have empty intersection. The Kneser-Lovász theorem states that Kneser graphs have a large chromatic number:

Theorem 3.1 (Lovász [48])**.** *Let $n \geq 2k > 1$. The (n, k)-Kneser graph has no coloring with $n - 2k + 1$ colors.*

It is well-known that the (n, k)-Kneser graph has a coloring with $n-2k+2$ colors (see Section 3.3.3), so the bound $n - 2k + 1$ is optimal. For $k = 1$, the Kneser-Lovász theorem is just the pigeonhole principle.

Istrate and Crăciun [40] noted that, for fixed values of k, the propositional translations of the Kneser-Lovász theorem have polynomial size in n. They presented proofs that can be formalized by polynomial size Frege proofs for $k = 2$, and by polynomial size extended Frege proofs for $k = 3$. This left open the possibility that the $k = 3$ case could exponentially separate the Frege and extended Frege systems. It was also left open whether the $k > 3$ case of the Kneser-Lovász theorem gave tautologies that require exponential size extended Frege proofs. As discussed above, the present paper refutes these possibilities. Theorems 3.4 and 3.5 summarize these results.

Let $[n]$ be the set $\{1, \ldots, n\}$; members of $[n]$ are called *nodes*. We identify $\binom{n}{k}$ with the set of k-subsets of $[n]$, the *vertices* of the (n, k)-Kneser graph.

Definition 3.2. An *m-coloring* of the (n, k)-Kneser graph is a map c from $\binom{n}{k}$ to $[m]$, such that for $S, T \in \binom{n}{k}$, if $S \cap T = \emptyset$, then $c(S) \neq c(T)$. If $\ell \in [m]$, then the *color class* P_ℓ is the set of vertices assigned the color ℓ by c.

The formulas Kneser_k^n are the natural propositional translations of the statement that there is no $(n - 2k + 1)$-coloring of the (n, k)-Kneser graph:

Definition 3.3. Let $n \geq 2k > 1$, and $m = n - 2k + 1$. For $S \in \binom{n}{k}$ and $i \in [m]$, the propositional variable $p_{S,i}$ has the intended meaning that vertex S of the Kneser graph is assigned the color i. The formula Kneser_k^n is

$$\bigwedge_{S \in \binom{n}{k}} \bigvee_{i \in [m]} p_{S,i} \;\rightarrow \bigvee_{\substack{S,T \in \binom{n}{k} \\ S \cap T = \emptyset}} \bigvee_{i \in [m]} (p_{S,i} \wedge p_{T,i}) \,.$$

Theorem 3.4. *For fixed parameter $k \geq 1$, the propositional translations Kneser_k^n of the Kneser-Lovász theorem have polynomial size extended Frege proofs.*

Theorem 3.5. *For fixed parameter $k \geq 1$, the propositional translations Kneser_k^n of the Kneser-Lovász theorem have quasi-polynomial size Frege proofs.*

When both k and n are allowed to vary, it is open whether the Kneser_k^n tautologies have quasi-polynomial size (extended) Frege proofs, or equivalently, have proofs with size quasi-polynomially bounded in terms of n^k.

3.3 Mathematical Arguments

Section 3.3.1 gives the new proof of the Kneser-Lovász theorem; this is later shown to be formalizable with polynomial size extended Frege proofs. Section 3.3.2 gives a slightly more complicated but more efficient proof, later shown to be formalizable with quasi-polynomial size Frege proofs. The next definition and lemma are crucial for Sections 3.3.1 and 3.3.2.

Any two vertices in a color class P_ℓ have nonempty intersection. One way this can happen is for the color class to be "star-shaped":

Definition 3.6. A color class P_ℓ is *star-shaped* if $\bigcap P_\ell$ is nonempty. If P_ℓ is star-shaped, then any $i \in \bigcap P_\ell$ is called a *central node* of P_ℓ.

The next lemma bounds the size of color classes that are not star-shaped. It will be used in our proof of the Kneser-Lovász theorem to establish the existence of star-shaped color classes. The idea is that non-star-shaped color classes are too small to cover all $\binom{n}{k}$ vertices.

Lemma 3.7. *Let c be a coloring of $\binom{n}{k}$. If P_ℓ is not star-shaped, then*

$$|P_\ell| \;\leq\; k^2 \binom{n-2}{k-2}.$$

Proof. Suppose P_ℓ is not star-shaped. If P_ℓ is empty, the claim is trivial. So suppose $P_\ell \neq \emptyset$, and let $S_0 = \{a_1, \ldots, a_k\}$ be some element of P_ℓ. Since P_ℓ is not star-shaped, there must be sets $S_1, \ldots, S_k \in P_\ell$ with $a_i \notin S_i$ for $i = 1, \ldots, k$.

To specify an arbitrary element S of P_ℓ, we do the following. Since S and S_0 have the same color, $S \cap S_0$ is nonempty. We first specify some $a_i \in S \cap S_0$. Likewise, $S \cap S_i$ is nonempty; we second specify some $b \in S \cap S_i$. By construction, $a_i \neq b$, so S is fully specified by the k possible values for a_i, the k possible values for b, and the $\binom{n-2}{k-2}$ possible values for the remaining members of S. Therefore, $|P_\ell| \leq k^2\binom{n-2}{k-2}$. □

3.3.1 Argument for Extended Frege Proofs

Let $k > 1$ be fixed. We prove the Kneser-Lovász theorem by induction on n. The base cases for the induction are $n = 2k, \ldots, N(k)$ where $N(k)$ is the constant depending on k specified in Lemma 3.8. We shall show that $N(k)$ is no greater than k^4. Since k is fixed, there are only finitely many base cases. Since the Kneser-Lovász theorem is true, these base cases can all be proved by a fixed Frege proof of finite size (depending on k). Therefore, in our proof below, we only show the induction step.

Lemma 3.8. *Fix $k > 1$. There is an $N(k)$ so that, for $n > N(k)$, any $(n - 2k + 1)$-coloring of $\binom{n}{k}$ has at least one star-shaped color class.*

Proof. Suppose that a coloring c has no star-shaped color class. Since there are $n-2k+1$ many color classes, Lemma 3.7 implies that

$$(n - 2k + 1) \cdot k^2 \binom{n-2}{k-2} \;\geq\; \binom{n}{k}. \tag{3.1}$$

For fixed k, the left-hand side of (3.1) is $\Theta(n^{k-1})$ and the right-hand side is $\Theta(n^k)$. Thus, there exists an $N(k)$ such that (3.1) fails for all $n > N(k)$. Hence for $n > N(k)$, there must be at least one star-shaped color class. □

To obtain an upper bound on the value of $N(k)$, note that (3.1) is equivalent to

$$(n - 2k + 1)k^3(k - 1) \;\geq\; n(n - 1). \tag{3.2}$$

Since $2k - 1 \geq 1$, (3.2) implies that $(n - 1)k^4 > n(n - 1)$ and thus that $n < k^4$. Thus, (3.1) will be false if $n \geq k^4$; so $N(k) < k^4$.

We are now ready to give our first proof of the Kneser-Lovász theorem.

Proof of Theorem 3.1, except for base cases. Fix $k > 1$. By Lemma 3.8, there is some $N(k)$ such that for $n > N(k)$, any $(n-2k+1)$-coloring c of $\binom{n}{k}$ has a star-shaped color class. As discussed above, the cases where $n \leq N(k)$ are handled by exhaustive search and the truth of the Kneser-Lovász theorem. For $n > N(k)$, we prove Theorem 3.1 by infinite descent. In other words, we show that if c is an $(n-2k+1)$-coloring of $\binom{n}{k}$, then there is some c' that is an $((n-1)-2k+1)$-coloring of $\binom{n-1}{k}$.

By Lemma 3.8, the coloring c has some star-shaped color class P_ℓ with central node i. Without loss of generality, $i = n$ and $\ell = n-2k+1$. Let

$$c' \;=\; c \restriction \tbinom{n-1}{k}$$

be the restriction of c to the domain $\binom{n-1}{k}$. This discards the central node n of P_ℓ, and thus all vertices with color ℓ. Therefore, c' is an $((n-1)-2k+1)$-coloring of $\binom{n-1}{k}$. This completes the proof. □

3.3.2 Argument for Frege Proofs

We now give a second proof of the Kneser-Lovász theorem. The proof above required $n - N(k)$ rounds of infinite descent to transform a Kneser graph on n nodes to one on $N(k)$ nodes. Our second proof replaces this with only $O(\log n)$ many rounds, and this efficiency will be key for formalizing this proof with quasi-polynomial size Frege proofs in Section 3.4.2.

We refine Lemma 3.8 to show that for n sufficiently large, there are many (i.e., a constant fraction) star-shaped color classes. The idea is to combine the upper bound of Lemma 3.7 on the size of non-star-shaped color classes with the trivial upper bound of $\binom{n-1}{k-1}$ on the size of star-shaped color classes.

Lemma 3.9. *Fix $k > 1$ and $0 < \beta < 1$. Then there exists an $N(k,\beta)$ such that for $n > N(k,\beta)$, if c is an $(n-2k+1)$-coloring of $\binom{n}{k}$, then c has at least $\frac{n}{k}\beta$ many star-shaped color classes.*

Proof. The value of $N(k,\beta)$ can be set equal to $\frac{k^3(k-\beta)}{1-\beta}$. Let $n > \frac{k^3(k-\beta)}{1-\beta}$, and suppose c is an $(n-2k+1)$-coloring of $\binom{n}{k}$. Let α be the number of star-shaped color classes of c. It is clear that an upper bound on the size of each star-shaped color class is $\binom{n-1}{k-1}$. There are $n-\alpha-2k+1$ many non-star-shaped classes, and Lemma 3.7 bounds their size by $k^2\binom{n-2}{k-2}$. This implies that

$$\binom{n-1}{k-1}\alpha + k^2\binom{n-2}{k-2}(n-\alpha-2k+1) \;\geq\; \binom{n}{k}. \tag{3.3}$$

Assume for a contradiction that $\alpha < \frac{n}{k}\beta$. Since $n > \frac{k^3(k-\beta)}{1-\beta}$, $0 < \beta < 1$, and $k \geq 2$, we have $n-1 > k^3(k-1) > k^2(k-1)$. Therefore, $\binom{n-1}{k-1} > k^2\binom{n-2}{k-2}$, and if α is replaced by the larger value $\frac{n}{k}\beta$, the left hand side of (3.3) increases. Thus,

$$\binom{n-1}{k-1}\frac{n}{k}\beta + k^2\binom{n-2}{k-2}\Big(n - \frac{n}{k}\beta - 2k + 1\Big) \;>\; \binom{n}{k}.$$

Since $\binom{n-1}{k-1}\frac{n}{k} = \binom{n}{k}$ and $n - \frac{n}{k}\beta - 2k + 1 = \frac{k-\beta}{k}n - 2k + 1$,

$$k^2\binom{n-2}{k-2}\Big(\frac{k-\beta}{k}n - 2k + 1\Big) \;>\; (1-\beta)\binom{n}{k}.$$

Expanding the binomial coefficients yields

$$k^3(k-1)\Big(\frac{k-\beta}{k}n - 2k + 1\Big) \;>\; (1-\beta)n(n-1).$$

We have $\frac{k-\beta}{k}(n-1) > \frac{k-\beta}{k}n - 2k + 1$. Therefore,

$$k^3(k-1)\frac{k-\beta}{k}(n-1) \;>\; (1-\beta)n(n-1).$$

Dividing by $n-1$ gives $k^3(k-\beta) \;>\; (1-\beta)n$, contradicting $n > \frac{k^3(k-\beta)}{1-\beta}$. □

We now give our second proof of the Kneser-Lovász theorem.

Proof of Theorem 3.1, except for base cases. Fix $k > 1$. By Lemma 3.9 with $\beta = 1/2$, if $n > N(k, 1/2)$ and c is an $(n-2k+1)$-coloring of $\binom{n}{k}$, then c has at least $n/2k$ many star-shaped color classes. We prove the Kneser-Lovász theorem by induction on n. The base cases are where $2k \leq n \leq N(k, 1/2)$, and there are only finitely of these, so they can be exhaustively proven. For $n > N(k, 1/2)$, we structure the induction proof as an infinite descent. In other words, we show that if c is an $(n-2k+1)$-coloring of $\binom{n}{k}$, then there is some c' that is an $((n - \frac{n}{2k}) - 2k + 1)$-coloring of $\binom{n-\frac{n}{2k}}{k}$. For simplicity of notation, we assume $\frac{n}{2k}$ is an integer. If this is not the case, we really mean to round up to the nearest integer $\lceil \frac{n}{2k} \rceil$.

By permuting the color classes and the nodes, we can assume w.l.o.g. that the $\frac{n}{2k}$ color classes P_ℓ for $\ell = n - \frac{n}{2k} - 2k + 2, \ldots, n - 2k + 1$ are star-shaped, and each such P_ℓ has a central node in $\{n - (n/2k) + 1, \ldots, n\}$. That is, the last $\frac{n}{2k}$ many color classes are star-shaped, and they all have a central node among the last $\frac{n}{2k}$ nodes in $[n]$. We shall discard these $n/2k$ many star-shaped color classes, and the topmost $n/2k$ many nodes. This discards the central nodes of the discarded color classes, thereby removing all the vertices of the Kneser graph which are assigned discarded color classes. (It is possible

that some star-shaped color classes share central nodes. We only need to be sure to discard at least one central node for each color classes, and thus, in this case, additional nodes can be discarded so that $n/2k$ are discarded in all.)

More formally, define c' to be the coloring of $\binom{n-n/2k}{k}$ which assigns the same colors as c. The map c' is a $(\frac{2k-1}{2k}n - 2k + 1)$-coloring of $\binom{\frac{2k-1}{2k}n}{k}$, since $n - \frac{n}{2k} = \frac{2k-1}{2k}n$. This completes the proof of the induction step. □

When formalizing the above argument with quasi-polynomial size Frege proofs, it will be important to know how many iterations of the procedure are required to reach the base cases, so let us calculate this.

After s iterations of this procedure, we have a $((\frac{2k-1}{2k})^s n - 2k + 1)$-coloring of $\binom{(\frac{2k-1}{2k})^s n}{k}$. We pick s large enough so that $(\frac{2k-1}{2k})^s n$ is less than $N(k, 1/2)$. In other words, since k is constant,

$$s \;=\; \log_{\frac{2k}{2k-1}}\Big(\frac{n}{k^3(2k-1)}\Big) \;=\; O(\log n)$$

will suffice, and only $O(\log n)$ many rounds of the procedure are required.

3.3.3 Optimal Colorings of Kneser Graphs

This section is a brief diversion motivated by the question of whether Lemma 3.9 about the number of non-star-shaped colors is optimal.

It is well-known that $\binom{n}{k}$ has an $(n-2k+2)$-coloring [48]. A simple construction of such a coloring, which we call c_1, is given here for completeness as follows. For $S \in \binom{n}{k}$, define $c_1(S)$ by:

(1) If $S \not\subseteq [2k-1]$, let $c_1(S) = \max(S) - (2k-2)$. Clearly $1 < c_1(S) \leq n - 2k + 2$.

(2) If $S \subseteq [2k-1]$, let $c_1(S) = 1$.

We claim that c_1 defines a proper coloring. By construction, if $c_1(S) > 1$, then $c_1(S) + (2k-2) \in S$. Thus, if $c_1(S) = c_1(S') > 1$, then $S \cap S' \neq \emptyset$ and S and S' are not joined by an edge in the Kneser graph. On the other hand, if $c_1(S) = 1$, then S contains k elements from the set $[2k-1]$. Any two such subsets have nonempty intersection, and therefore if $c_1(S) = c_1(S') = 1$, then again $S \cap S' \neq \emptyset$. Note that c_1 contains $n - 2k + 1$ many star-shaped color classes, and only one non-star-shaped color class.

In view of Lemma 3.9, it is interesting to ask whether it is possible to give $(n-2k+2)$-colorings with fewer star-shaped color classes and more non-star-shaped color classes. The next theorem gives the best construction we know.

Theorem 3.10. *Let $k \geq 1$ and $n \geq 3k+3$. There is an $(n-2k+2)$ coloring c_{k-1} of $\binom{n}{k}$ which has $k-1$ many non-star-shaped color classes and only $n-3k+3$ many star-shaped color classes.*

Proof. To construct c_{k-1}, partition the set $[n]$ into $n-2k+2$ many subsets $T_1, \ldots, T_{n-2k+2}$ as follows. For $i \leq n-3k+3$, T_i is chosen to be a singleton set, say $T_i = \{n-i+1\}$. The remaining $k-1$ many T_i's are subsets of size 3, say $T_i = \{j-2, j-1, j\}$ where $j = 3(i-(n-3k+3))$. Since $n = (n-3k+3)+3(k-1)$, the sets T_i partition $[n]$, and each T_i has cardinality either 1 or 3. For S a subset of n of cardinality k, define the color $c_{k-1}(S)$ to equal the least i such that

$$|S \cap T_i| \ > \ \frac{1}{2}|T_i|.$$

We claim there must exist such an i. If not, then S contains no members of the singleton subsets T_i and at most one member of each of the subsets T_i of size three. But there are only $k-1$ many subsets of size three, contradicting $|S| = k$.

It is easy to check that if $c_{k-1}(S) = c_{k-1}(S')$ then $S \cap S' \neq \emptyset$. Thus c_{k-1} is a coloring. Furthermore, c_{k-1} has $k-1$ many non-star-shaped color classes and $n-3k+3$ many star-shaped color classes. □

Theorem 3.10 can be extended to show that when $n < 3k+3$, there is a $n-2k+2$ coloring with no star-shaped color class. The proof construction uses a similar idea, based on the fact that $[n]$ can be partitioned into $n-2k+2$ many subsets, each of odd cardinality ≥ 3. We leave the details to the reader.

Question 3.11. *Do there exist $(n-2k+2)$-colorings of the (n,k)-Kneser graphs with more than $k-1$ many non-star-shaped color classes?*

3.4 Formalization in Propositional Logic

3.4.1 Polynomial Size Extended Frege Proofs

We sketch the formalization of the argument in Section 3.3.1 as a polynomial size extended Frege proof, establishing Theorem 3.4. We concentrate on showing how

to express concepts such as "star-shaped color class" with polynomial size propositional formulas. For expository reasons, we omit the straightforward details of how (extended) Frege proofs can prove properties of these concepts.

Fix values for k and n with $n > N(k)$. We describe an extended Frege proof of Kneser_k^n. We have variables $p_{S,j}$ (recall Definition 3.3), collectively denoted $\vec{p}$. The proof assumes $\text{Kneser}_k^n(\vec{p})$ is false, and proceeds by contradiction. The main step is to define new variables $\vec{p}'$ with the extension rule and prove that $\text{Kneser}_k^{n-1}(\vec{p}')$ fails. This will be repeated until reaching a Kneser graph over only $N(k)$ nodes.

For this, let $\text{Star}(i,\ell)$ be a formula that is true when $i \in [n]$ is a central node of the color class P_ℓ; namely,

$$\text{Star}(i,\ell) \;:=\; \bigwedge_{S\in\binom{n}{k}, i\notin S} \neg p_{S,\ell}.$$

Note that P_ℓ may have more than one central node. Conversely, a node i may be a central node for more than one color class.

We use $\text{Star}(\ell) := \bigvee_i \text{Star}(i,\ell)$ to express that P_ℓ is star-shaped.

The extended Frege proof defines an instance of the Kneser-Lovász principle Kneser_k^{n-1} by discarding one node and one color. The first star-shaped color class P_ℓ is discarded; accordingly, we let

$$\text{DiscardColor}(\ell) \;:=\; \text{Star}(\ell) \wedge \bigwedge_{\ell'<\ell} \neg\text{Star}(\ell').$$

The node to be discarded is the least central node of the discarded P_ℓ:

$$\text{DiscardNode}(i) := \bigvee_\ell \Big[\text{DiscardColor}(\ell) \wedge \text{Star}(i,\ell) \wedge \bigwedge_{i'<i} \neg\text{Star}(i',\ell)\Big].$$

After discarding the node i and the color ℓ, the remaining nodes and colors are renumbered to the ranges $[n-1]$ and $[n-2k]$, respectively. In particular, the "new" color j (in the instance of Kneser_k^{n-1}) corresponds to the "old" color $j^{-\ell}$ (in the instance of Kneser_k^n) where

$$j^{-\ell} = \begin{cases} j & \text{if } j < \ell \\ j+1 & \text{if } j \geq \ell. \end{cases}$$

And, if $S = \{i_1,\ldots,i_k\} \in \binom{n-1}{k}$ is a "new" vertex (for the Kneser_k^{n-1} instance), then it corresponds to the "old" vertex $S^{-i} \in \binom{n}{k}$ (for the instance of Kneser_k^n), where $S^{-i} =$

$\{i'_1, i'_2, \ldots, i'_k\}$ with

$$i'_t = \begin{cases} i_t & \text{if } i_t < i \\ i_t + 1 & \text{if } i_t \geq i. \end{cases}$$

For each $S \in \binom{n-1}{k}$ and $j \in [n-2k]$, the extended Frege proof uses the extension rule to introduce a new variable $p'_{S,j}$ defined as follows

$$p'_{S,j} \equiv \bigvee_{i,\ell} \left(\text{DiscardNode}(i) \wedge \text{DiscardColor}(\ell) \wedge p_{S^{-i},j^{-\ell}} \right).$$

As seen in the definition by extension, $p'_{S,j}$ is defined by cases, one for each possible pair i, ℓ of nodes and colors such that the node i is the least central node of the P_ℓ color class, where P_ℓ is the first star-shaped color class. The extended Frege proof then shows that $\neg\text{Kneser}^n_k(\vec{p})$ implies $\neg\text{Kneser}^{n-1}_k(\vec{p}')$, i.e., that if the variables $p_{S,j}$ define a coloring, then the variables $p'_{S,j}$ also define a coloring. The first step for the extended Frege proof is to show that there is at least one star-shaped color class, and then there is a unique ℓ such that DiscardColor(ℓ) holds. In fact, we claim there are polynomial size Frege proofs of

$$\bigvee_{\ell} \text{DiscardColor}(\ell)$$

and

$$\bigwedge_{\ell_1 < \ell_2} \left(\neg\text{DiscardColor}(\ell_1) \vee \neg\text{DiscardColor}(\ell_2) \right).$$

These assertions are proved using the proof of Lemma 3.8, and the counting techniques which can be formalized in Frege proofs. Note that we only need to count numbers of vertices in $\binom{n}{k}$; hence, for fixed k, we are only counting sets of polynomial size. Therefore, polynomial size Frege proofs can carry out the proof of Lemma 3.8. For similar reasons, there are polynomial size Frege proofs that there is a unique value $i \in [n-2k+1]$ which satisfies DiscardNode(i).

For fixed values of ℓ and i, a polynomial size Frege proof now establishes

$$\text{DiscardColor}(\ell) \wedge \text{DiscardNode}(i) \wedge \text{Kneser}^{n-1}_k(\vec{p}') \rightarrow \text{Kneser}^n_k(\vec{p}).$$

This Frege proof argues as follows, assuming DiscardColor(ℓ) and DiscardNode(i) and $\text{Kneser}^{n-1}_k(\vec{p}')$. Since $\text{Kneser}^{n-1}_k(\vec{p}')$ is true, either (a) its hypothesis is false and we have $\bigwedge_{j=1}^{n-2k} \neg p'_{S,j}$ for some $S \in \binom{n}{k}$ or (b) its conclusion is true and there are $S, T \in \binom{n}{k}$ and j such that $S \cap T = \emptyset$ and $p'_{S,j}$ and $p'_{T,j}$. If (a) holds then $\neg p_{S^{-i},j^{-\ell}}$ for all $j \in [n-2k]$

and this together with the fact that $i \notin S^{-i}$ and i and ℓ were discarded further implies that the hypothesis of $\text{Kneser}_k^n(\vec{p})$ is false so $\text{Kneser}_k^n(\vec{p})$ is true. Likewise, if (b) holds, then using S^{-i} and T^{-i} and $j^{-\ell}$ shows that that the conclusion of Kneser_k^n is true.

Putting all these arguments together gives the desired Frege proof of

$$\neg\text{Kneser}_k^n(\vec{p}) \to \neg\text{Kneser}_k^{n-1}(\vec{p}').$$

The extended Frege proof iterates this process of removing one node and one color until it is shown that there is a coloring of $\binom{N(k)}{k}$. This is then refuted by exhaustively considering all graphs with $\leq N(k)$ nodes. □

3.4.2 Quasi-polynomial Size Frege Proofs

This section discusses some of the details of the formalization of the argument in Section 3.3.2 as quasi-polynomial size Frege proofs, establishing Theorem 3.5. First we will form an extended Frege proof, then modify it to become a Frege proof. As before, the proof starts with the assumption that $\text{Kneser}_k^n(\vec{p})$ is false. As we describe next, the extended Frege proof then introduces variables $\vec{p}'$ by extension so that $\text{Kneser}_k^{n-n/2k}(\vec{p}')$ is false. This process will be repeated $O(\log n)$ times. The final Frege proof is obtained by unwinding the definitions by extension.

For a set X of formulas and $t > 0$, we now use the notation "$|X| \leq t$" to denote a formula that is true when the number of true formulas in X is less than or equal to t. As already discussed, "$|X| \leq t$" can be expressed by a formula of size polynomially bounded by the total size of the formulas in X, using the construction in [16]. "$|X| = t$" is defined similarly.

The formulas $\text{Star}(i, \ell)$ and $\text{Star}(\ell)$ are the same as in Section 3.4.1. A color ℓ is now discarded if it is among the least $n/2k$ star-shaped color classes.

$$\text{DiscardColor}(\ell) \;:=\; \text{Star}(\ell) \wedge \big(|\{\text{Star}(\ell') : \ell' \leq \ell\}| \leq n/2k\big)$$

The discarded nodes are the least central nodes of the discarded color classes.

$$\text{DiscardNode}(i) \;:=\; \bigvee_{\ell} \Big[\text{DiscardColor}(\ell) \wedge \text{Star}(i, \ell) \wedge \bigwedge_{i'<i} \neg\text{Star}(i', \ell)\Big].$$

$\text{DiscardNode}(i)$ will hold for at most $n/2k$ many nodes i, since there are only $n/2k$ many discarded colors. We could modify the definition of DiscardNode to discard exactly $n/2k$ many nodes; however, this is not strictly necessary, as the only use of DiscardNode is

to define the predicate RenumNode(i', i) below, and that definition effectively discards exactly $n/2k$ many nodes even if DiscardNode(i) picks out fewer than $n/2k$ many nodes to be discarded.

The remaining, non-discarded colors and nodes are renumbered to form an instance of $\text{Kneser}_k^{n-n/2k}$. For this, the formula RenumNode(i', i) is true when the node i' is the ith node that is not discarded; similarly RenumColor(j', j) is true when the color j' is the jth color that is not discarded.

$$\begin{aligned}\text{RenumNode}(i', i) &:= \big(|\{\neg\text{DiscardNode}(i'') : i''\leq i'\}| = i\big) \wedge \neg\text{DiscardNode}(i')\\ \text{RenumColor}(j', j) &:= \big(|\{\neg\text{DiscardColor}(j'') : j''\leq j'\}| = j\big) \wedge \neg\text{DiscardColor}(j')\end{aligned}$$

The predicate RenumNode(i', i) defines a bijection between the sets $[n-n/2k]$ and the non-discarded nodes of $[n]$. Likewise, the predicate RenumColor(j', j) defines a bijection between $[(n-n/2k)-2k+1]$ and the non-discarded colors.

For each $S = \{i_1, \ldots, i_k\} \in \binom{n-n/2k}{k}$ and $j \in [(n - n/2k) - 2k + 1]$, we define by extension

$$p'_{S,j} \equiv \bigvee_{i'_1,\ldots i'_k,j'} \left(\bigwedge_{t=1}^{k} \big(\text{RenumNode}(i'_t, i_t)\big) \wedge \text{RenumColor}(j', j) \wedge p_{\{i'_1,\ldots,i'_k\},j'} \right).$$

The Frege proof then argues that if the variables $p_{S,j}$ define a coloring, then the variables $p'_{S,j}$ define a coloring, i.e., that $\neg\text{Kneser}_k^n(\vec{p}) \to \neg\text{Kneser}_k^{n-n/2k}(\vec{p}')$. The first step for this is proving that there are at least $n/2k$ star-shaped color classes by formalizing the proofs of Lemmas 3.7 and 3.9. Those proofs were "counting" arguments: they involved counting the number of members of $\binom{n}{k}$ that are contained in the color classes P_ℓ. Since $\binom{n}{k} < n^k$, there are only polynomially many members of $\binom{n}{k}$. Likewise there are $< n$ color classes. The proofs of Lemmas 3.7 and 3.9 used binonomial coefficients $\binom{n'}{k'}$, but only with $n' \leq n$ and $k' \leq k$, thus the proofs only used counting for polynomial size sets. Therefore, all these counting arguments can be carried out using polynomial size Frege proofs with the techniques from [16]. From this, the fact that RenumNode(i', i) and RenumColor(j', j) define bijections follows easily.

After that, it is straightforward to prove that, for each $S \in \binom{n-n/2k}{k}$ and $j \in [(n - n/2k) - 2k + 1]$, the variable $p'_{S,j}$ is well-defined. In addition, a polynomial size Frege proof can prove that that if $\text{Kneser}_k^n(\vec{p})$ is false, then $\text{Kneser}_k^{n-n/2k}(\vec{p}')$ is false.

This is iterated $O(\log n)$ times until fewer than $N(k, 1/2)$ nodes remain. The proof concludes with a hard-coded proof that there are no such colorings of the finitely

many small Kneser graphs.

To form the quasi-polynomial size Frege proof, we unwind the definitions by extension. Each definition by extension was polynomial size; they are nested to a depth of $O(\log n)$. So the resulting Frege proof is quasi-polynomial size. □

3.5 The Tucker Lemma and the Truncated Tucker Lemmas

A natural question arising from the previous sections is the possibility of giving short uniform Frege proofs of the Kneser-Lovász theorem for fixed k, namely, proofs that avoid handling finitely many base cases separately. A possible approach to this problem is formalizing the proof of Matoušek [49] in the Frege system. A significant obstacle in carrying this out is that Matoušek's proof goes through the octahedral Tucker lemma, and, as will be discussed below, naïve propositional translations of the octahedral Tucker lemma are exponential size. To overcome this, we describe two miniaturizations of the octahedral Tucker lemma, called the truncated Tucker lemmas. The truncated Tucker lemmas have polynomial size propositional translations, and are strong enough to imply the Kneser-Lovász theorem with polynomial size, constant depth Frege proofs.

Our definitions and proofs below borrow techniques and notation from Matoušek [49].

Definition 3.12. Let $n \geq 1$. The *octahedral ball* $\mathcal{B}^n$ is:

$$\mathcal{B}^n := \{(A, B) : A, B \subseteq [n] \text{ and } A \cap B = \emptyset\}.$$

Definition 3.13. Let $n > 1$. A mapping $\lambda : \mathcal{B}^n \to \{1, \pm 2, \ldots, \pm n\}$ is *antipodal* if $\lambda(\emptyset, \emptyset) = 1$, and for all other pairs $(A, B) \in \mathcal{B}^n$, $\lambda(A, B) = -\lambda(B, A)$.

Note that -1 is not in the range of λ, and $(\emptyset, \emptyset)$ is the only member of $\mathcal{B}^n$ that is mapped to 1 by λ.

Definition 3.14. Two pairs (A_1, B_1) and (A_2, B_2) in $\mathcal{B}^n$ are *complementary* with respect to an antipodal map λ on $\mathcal{B}^n$ if $A_1 \subseteq A_2$, $B_1 \subseteq B_2$ and $\lambda(A_1, B_1) = -\lambda(A_2, B_2)$.

Theorem 3.15 (Octahedral Tucker lemma)**.** *If $\lambda : \mathcal{B}^n \to \{1, \pm 2, \ldots, \pm n\}$ is antipodal, then there are two elements in $\mathcal{B}^n$ that are complementary.*

For a proof of Theorem 3.15, see [49].

Definition 3.16. Let $1 \leq k \leq n$. The *truncated octahedral ball* $\mathcal{B}^n_{\leq k}$ is:

$$\mathcal{B}^n_{\leq k} := \Big\{(A, B) \in \mathcal{B}^n : |A| \leq k, |B| \leq k\}.$$

We write $\binom{n}{\leq k}$ for $\{A \subseteq [n] : |A| \leq k\}$.

The octahedral Tucker lemma used the subset relation $\subseteq$ on $[n]$ to define complementary. The truncated Tucker lemma uses an analogous partial order $\preceq$ to define k-complementary. For $A \subseteq [n]$, let $A_{\leq k}$ denote the least k elements of A. By convention, if $|A| < k$, then $A_{\leq k} = A$.

Definition 3.17. Let $\preceq$ be the partial order on sets in $\binom{n}{\leq k}$ defined by $A \preceq B$ iff $(A \cup B)_{\leq k} = B$.

Remark: Note that when $n = k$, $\mathcal{B}^n = \mathcal{B}^n_{\leq k}$, and the $\preceq$ relation is identical to the subset relation.

Lemma 3.18. *The relation $\preceq$ is a partial order with $\emptyset$ its least element.*

Proof. It is clearly reflexive. For anti-symmetry, $A_1 \preceq A_2$ and $A_2 \preceq A_1$ imply that $A_1 = (A_1 \cup A_2)_{\leq k} = (A_2 \cup A_1)_{\leq k} = A_2$. For transitivity, suppose $A_1 \preceq A_2$ and $A_2 \preceq A_3$. Then $(A_1 \cup A_2)_{\leq k} = A_2$ and $(A_2 \cup A_3)_{\leq k} = A_3$. This implies that

$$A_3 = (A_2 \cup A_3)_{\leq k} = ((A_1 \cup A_2)_{\leq k} \cup A_3)_{\leq k} = (A_1 \cup (A_2 \cup A_3)_{\leq k})_{\leq k} = (A_1 \cup A_3)_{\leq k}.$$

Therefore $A_1 \preceq A_3$. That $\emptyset$ is the least element is clear from the definition. □

Definition 3.19. For (A_1, B_1) and (A_2, B_2) in $\mathcal{B}^n_{\leq k}$, write $(A_1, B_1) \preceq (A_2, B_2)$ when $A_1 \preceq A_2$, $B_1 \preceq B_2$, and $A_i \cap B_j = \emptyset$ for $i, j \in \{1, 2\}$. The pairs (A_1, B_1) and (A_2, B_2) are *k-complementary* with respect to an antipodal map λ on $\mathcal{B}^n_k$ if $(A_1, B_1) \preceq (A_2, B_2)$ and $\lambda(A_1, B_1) = -\lambda(A_2, B_2)$.

We are ready to state the version of the truncated Tucker lemma for $\mathcal{B}^n_{\leq k}$.

Theorem 3.20 (Truncated Tucker lemma on $\mathcal{B}^n_{\leq k}$)**.** *Let $n \geq k \geq 1$. If $\lambda : \mathcal{B}^n_{\leq k} \to \{1, \pm 2 \ldots, \pm n\}$ is antipodal, then there are two elements in $\mathcal{B}^n_{\leq k}$ that are k-complementary.*

When $k = n$, this is equivalent to the octahedral Tucker lemma. The truncated Tucker lemma on $\mathcal{B}^n_{\leq k}$ follows from the octahedral Tucker lemma:

Proof of Theorem 3.20 from Theorem 3.15. We argue by contradiction. Suppose $\lambda : \mathcal{B}^n_{\leq k} \to \{1, \pm 2, \ldots, \pm n\}$ is antipodal. We define $\lambda' : \mathcal{B}^n \to \{1, \pm 2, \ldots, \pm n\}$. For $(A, B) \in \mathcal{B}^n$, define $\lambda'(A, B) = \lambda(A_{\leq k}, B_{\leq k})$. The map λ' is clearly antipodal, so by Theorem 3.15, there are $(A, B), (C, D)$ in $\mathcal{B}^n$ that are complementary with respect to λ'. We claim that $(A_{\leq k}, B_{\leq k})$ and $(C_{\leq k}, D_{\leq k})$ are k-complementary with respect to λ. By definition of λ', $\lambda(A_{\leq k}, B_{\leq k}) = -\lambda(C_{\leq k}, D_{\leq k})$, so it remains to show that $(A_{\leq k}, B_{\leq k}) \preceq (C_{\leq k}, D_{\leq k})$. Since $C \cap D = \emptyset$ and $A \subseteq C$ and $B \subseteq D$, it follows that

$$C_{\leq k} \cap D_{\leq k} = A_{\leq k} \cap D_{\leq k} = A_{\leq k} \cap B_{\leq k} = B_{\leq k} \cap C_{\leq k} = \emptyset.$$

Moreover, $A \subseteq C$ implies that $A_{\leq k} \preceq C_{\leq k}$. This is because

$$(A_{\leq k} \cup C_{\leq k})_{\leq k} = (A \cup C)_{\leq k} = C_{\leq k}.$$

The same argument shows that $B_{\leq k} \preceq D_{\leq k}$. □

Definition 3.21. Let $1 < 2k \leq n$. The *truncated octahedral ball* $\mathcal{B}^n_k$ is:

$$\mathcal{B}^n_k := \left\{(A, B) : A, B \in \binom{n}{k} \cup \{\emptyset\},\ A \cap B = \emptyset,\ \text{and } (A, B) \neq (\emptyset, \emptyset)\right\}.$$

The fact that $(\emptyset, \emptyset)$ is excluded from $\mathcal{B}^n_k$ is only a technical convenience. Corresponding to this, the value "1" will now be omitted from the range of λ. We say that $\lambda : \mathcal{B}^n_k \to \{\pm 2k \ldots, \pm n\}$ is *antipodal* provided that $\lambda(A, B) = -\lambda(B, A)$ for all $(A, B) \in \mathcal{B}^n_k$.

Theorem 3.22 (Truncated Tucker lemma on $\mathcal{B}^n_k$)**.** *Let $n \geq 2k > 1$. If $\lambda : \mathcal{B}^n_k \to \{\pm 2k \ldots, \pm n\}$ is antipodal, then there are two elements in $\mathcal{B}^n_k$ that are k-complementary.*

Proof of Theorem 3.22 from Theorem 3.20. Suppose that $\lambda : \mathcal{B}^n_k \to \{\pm 2k, \ldots, \pm n\}$ is antipodal; we must show it has k-complementary pairs. We extend λ to an anitipodal $\lambda' : \mathcal{B}^n_{\leq k} \to \{1, \pm 2, \cdots \pm n\}$. Let "$\leq$" be any total order on $\binom{n}{\leq k}$ that extends $\preceq$. Let $(A, B) \in \mathcal{B}^n_{\leq k}$. The value of $\lambda'(A, B)$ is defined by cases:

Case 1: If $|A| < k$ and $|B| < k$, then define

$$\lambda'(A, B) = \begin{cases} 1 + |A| + |B| & \text{if } A \leq B \\ -(1 + |A| + |B|) & \text{if } B < A. \end{cases}$$

Case 2: If $\max\{|A|,|B|\} = k$ and $\min\{|A|,|B|\} < k$, then define

$$\lambda'(A,B) = \begin{cases} \lambda(A,\emptyset) & \text{if } |B| < k \\ \lambda(\emptyset,B) & \text{if } |A| < k. \end{cases}$$

Case 3: If $|A| = |B| = k$, then define $\lambda'(A,B) = \lambda(A,B)$.

The map λ' is clearly antipodal; hence by Theorem 3.20 there are $(A_1,B_1) \preceq (A_2,B_2)$ that are k-complementary with respect to λ', so $\lambda'(A_1,B_1) = -\lambda'(A_2,B_2)$. We prove this gives rise to k-complementary pairs for λ. The argument splits into cases depending on how λ' assigns values to (A_1,B_1) and (A_2,B_2).

Suppose that $\lambda'(A_1,B_1)$ is assigned by case 1, then $\lambda'(A_2,B_2)$ must also be assigned by case 1, since case 1 only assigns values to $\{1,\pm 2,\ldots,\pm(2k-1)\}$, and cases 2 and 3 only assign values to $\{\pm 2k,\ldots,\pm n\}$. Also, $A_1 \preceq A_2$ and $B_1 \preceq B_2$ where at least one of these precedences is proper; this implies that $|A_1| \leq |A_2|$ and $|B_1| \leq |B_2|$ where at least one of these inequalities must be proper. Thus $1+|A_1|+|B_1| < 1+|A_2|+|B_2|$, so $\lambda'(A_1,B_1)$ and $\lambda'(A_2,B_2)$ differ in absolute value. This contradicts the fact that (A_1,B_1) and (A_2,B_2) are k-complementary w.r.t. λ'. Thus it is impossible that both $\lambda'(A_1,B_1)$ and $\lambda'(A_2,B_2)$ are assigned by case 1.

Suppose $\lambda'(A_1,B_1)$ and $\lambda'(A_2,B_2)$ are both assigned by case 2. Without loss of generality $|B_1| < k$, which implies $|A_1| = |A_2| = k$ and $|B_2| < k$. This implies that $\lambda(A_1,\emptyset) = -\lambda(A_2,\emptyset)$. But $(A_1,\emptyset) \preceq (A_2,\emptyset)$, so these form a k-complementary pair for λ.

Suppose $\lambda'(A_1,B_1)$ is assigned by case 2 and $\lambda'(A_2,B_2)$ is assigned by case 3. Without loss of generality $|B_1| < k$. This implies that $\lambda(A_1,\emptyset) = -\lambda(A_2,B_2)$. But $(A_1,\emptyset) \preceq (A_2,B_2)$, so these form a k-complementary pair for λ.

Suppose $\lambda'(A_1,B_1)$ and $\lambda'(A_2,B_2)$ are both assigned by case 3. Thus $\lambda(A_1,B_1) = -\lambda(A_2,B_2)$, so these form a k-complementary pair for λ.

Suppose $\lambda'(A_1,B_1)$ is assigned by case 3 and $\lambda'(A_2,B_2)$ is assigned by case 2. This is impossible because $|A_1| = |B_1| = k$, and $A_1 \preceq A_2$, $B_1 \preceq B_2$, so $|B_1| = |B_2| = k$. □

For fixed parameter k, the two truncated Tucker lemmas have polynomial size propositional translations. We will only describe the translation of the truncated Tucker lemma on $\mathcal{B}_k^n$. A similar translation works for the truncated Tucker lemma on $\mathcal{B}_{\leq k}^n$. For each $(A,B) \in \mathcal{B}_k^n$, and for each $i \in \{\pm 2k,\ldots,\pm n\}$, let $p_{A,B,i}$ be a propositional variable

with the intended meaning that $p_{A,B,i}$ is true when $\lambda(A,B) = i$. The following formula $\mathrm{Ant}(\vec{p})$ states that the map is total and antipodal:

$$\bigwedge_{(A,B)\in\mathcal{B}_k^n} \bigvee_{i\in\{\pm 2k,\ldots,\pm n\}} (p_{A,B,i} \wedge p_{B,A,-i}).$$

The following formula $\mathrm{Comp}(\vec{p})$ states that there exists two elements in $\mathcal{B}_k^n$ that are k-complementary:

$$\bigvee_{\substack{(A_1,B_1),(A_2,B_2)\in\mathcal{B}_k^n,\\ (A_1,B_1)\preceq(A_2,B_2)\\ i\in\{\pm 2k,\ldots,\pm n\}}} (p_{A_1,B_1,i} \wedge p_{A_2,B_2,-i}).$$

The truncated Tucker tautology Tucker_k^n is defined to be $\mathrm{Ant}(\vec{p}) \to \mathrm{Comp}(\vec{p})$. (We could add an additional hypothesis, that for each A, B there is at most one i such that $p_{A,B,i}$, but this is not needed for the Tucker tautologies to be valid.) There are $< n^{2k}$ members (A,B) in $\mathcal{B}_k^n$. Hence, for fixed k, there are only polynomially many variables $p_{A,B,i}$, and the truncated Tucker tautologies have size polynomially bounded by n. On the other hand, the propositional translation of the octahedral Tucker lemma requires an exponential number of propositional variables in n, since the cardinality of $\mathcal{B}^n$ is exponential in n.

The proof of Theorem 3.22 from Theorem 3.20 can be readily translated into polynomial size Frege proofs. That is, if propositional translations of the truncated Tucker lemma on $\mathcal{B}_{\leq k}^n$ are given as hypotheses, there are polynomial size Frege proofs of the polynomial translations of the truncated Tucker lemma on $\mathcal{B}_k^n$. Section 3.5.1 will prove a converse: the truncated Tucker lemma on $\mathcal{B}_{\leq k}^n$ follows from the truncated Tucker lemma on $\mathcal{B}_k^{n+2k-1}$ by polynomial size Frege proofs.

We next show that the Kneser-Lovász theorem (Theorem 3.1) follows from the truncated Tucker lemma on $\mathcal{B}_k^n$.

Proof of Theorem 3.1 from Theorem 3.22. Suppose for sake of contradiction that $c : \binom{n}{k} \to \{2k,\ldots,n\}$ is an $(n-2k+1)$-coloring of $\binom{n}{k}$. Let $\leq$ be a total order on $\binom{n}{k} \cup \{\emptyset\}$ that refines the partial order $\preceq$. Let $(A,B) \in \mathcal{B}_k^n$. Define $\lambda(A,B)$ as follows:

$$\lambda(A,B) = \begin{cases} c(A) & \text{if } A > B \\ -c(B) & \text{if } B > A \end{cases}$$

The map λ is clearly antipodal, so by Theorem 3.22, there is a pair $(A_1,B_1) \preceq (A_2,B_2) \in \mathcal{B}_k^n$ that is k-complementary. Since λ must assign (A_1,B_1) and (A_2,B_2) opposite signs,

it must be that either $A_1 < B_1 \leq B_2 < A_2$ or $B_1 < A_1 \leq A_2 < B_2$. In the former case, $c(B_1) = c(A_2)$ and in the latter case $c(A_1) = c(B_2)$. Since $B_1 \cap A_2 = A_1 \cap B_2 = \emptyset$, in either case we have a contradiction. □

The above proof of the Kneser-Lovász theorem from the truncated Tucker lemma can be readily translated into polynomial size constant depth Frege proofs.

Question 3.23. *Do the propositional translations of the truncated Tucker lemma have short (extended) Frege proofs?*

3.5.1 Equivalence Between the Truncated Tucker Lemmas

Theorem 3.24. *The truncated Tucker lemma on $\mathcal{B}_k^n$ implies the truncated Tucker lemma on $\mathcal{B}_{\leq k}^{n-2k+1}$.*

Proof. Let $1 < 2k \leq n$. Suppose that $\lambda : \mathcal{B}_{\leq k}^{n-2k+1} \to \{1, \pm 2, \ldots, \pm(n-2k+1)\}$ is an antipodal map. By renaming the range elements, we can instead write $\lambda : \mathcal{B}_{\leq k}^{n-2k+1} \to \{1, \pm 2k, \ldots, \pm(n-1)\}$. We will define $\lambda' : \mathcal{B}_k^n \to \{\pm 2k, \ldots, \pm n\}$ as follows: For $(A, B) \in \mathcal{B}_k^n$,

$$\lambda'(A,B) \;=\; \begin{cases} \lambda(A^*, B^*) & \text{if } A \neq \emptyset \text{ and } B \neq \emptyset \\ n & \text{if } A = \emptyset \\ -n & \text{if } B = \emptyset \end{cases}$$

where $A^* = \{a \in A : a \leq n - 2k + 1\}$. For $(A, B) \in \mathcal{B}_k^n$, we clearly have $(A^*, B^*) \in \mathcal{B}_{\leq k}^{n-2k+1}$. We also claim that $\lambda'(A, B)$ is never equal to 1. To prove this, suppose $\lambda'(A, B) = 1$. By the definition of λ', both A and B are nonempty. Thus $\lambda(A^*, B^*) = 1$ and consequently $A^* = B^* = \emptyset$. This means that A and B are both subsets of $\{n-2k+2, \ldots, n\}$, a set of cardinality $2k - 1$. But this contradicts $A \cap B = \emptyset$ and $|A| = |B| = k$.

The map λ' is clearly antipodal by definition. By the truncated Tucker lemma on $\mathcal{B}_k^n$, there are pairs $(A_1, B_1) \preceq (A_2, B_2) \in \mathcal{B}_k^n$ such that $\lambda'(A_1, B_1) = -\lambda'(A_2, B_2)$. We claim that $\lambda(A_1, B_1) \neq n$. Otherwise, $\lambda(A_2, B_2) = -n$, so $A_1 = \emptyset$ and $B_2 = \emptyset$, and this contradicts $(A_1, B_1) \preceq (A_2, B_2)$. Similarly, $\lambda(A_1, B_1) \neq -n$. It follows that all four sets A_1, B_1, A_2, B_2 are nonempty. Therefore, by the choice of (A_1, B_1) and (A_2, B_2),

$$\lambda(A_1^*, B_1^*) = -\lambda(A_2^*, B_2^*).$$

We now claim that $(A_1^*, B_1^*) \preceq (A_2^*, B_2^*)$. Since $A_1 \cap B_2 = \emptyset$ and $A_2 \cap B_1 = \emptyset$, we have $A_1^* \cap B_2^* = \emptyset$ and $A_2^* \cap B_1^* = \emptyset$. Also, since $A_1 \preceq A_2$,

$$(A_1 \cup A_2)_{\leq k} = A_2$$

From this we obtain

$$(A_1^* \cup A_2^*)_{\leq k} \;=\; ((A_1 \cup A_2)_{\leq k})^* \;=\; A_2^*.$$

Thus $A_1^* \preceq A_2^*$. The same argument shows $B_1^* \preceq B_2^*$. This establishes that (A_1^*, B_1^*) and (A_2^*, B_2^*) are k-complementary with respect to λ. □

Since the proofs of the equivalence of the two truncated Tucker lemmas can be translated into polynomial size Frege proofs, we have established:

Corollary 3.25. *The propositional translations of the truncated Tucker lemma on $\mathcal{B}_k^n$ have (quasi-)polynomial size Frege proofs if and only if the same holds for the truncated Tucker lemma on $\mathcal{B}_{\leq k}^n$.*

3.6 Short $e\mathcal{F}$ Proofs of the Truncated Tucker Lemma, $k = 1$ Case

In this section we prove the $k = 1$ case of the truncated Tucker lemma. The argument is readily formalizable as polynomial size extended Frege proofs. Note that when $k = 1$ the two versions of the truncated Tucker lemma are equivalent.

Recall the partial order $\preceq$ of Definition 3.17. When $k = 1$, this partial order is a total order where $\{i\} \preceq \{j\}$ iff $i \geq j$. Thus,

$$\emptyset \preceq \{n\} \preceq \{n-1\} \preceq \cdots \preceq \{2\} \preceq \{1\}$$

is a complete description of $\preceq$ on $\binom{n}{1}$.

Theorem 3.26. *The $k = 1$ case of the truncated Tucker lemma,* Tucker_1^n*, has polynomial size extended Frege proofs.*

The polynomial size extended Frege proofs of the $k = 1$ case of the truncated Tucker lemma are formed by formalizing the argument of Lemma 3.27 below.

Lemma 3.27. *Let $\lambda : \mathcal{B}_1^n \to \{\pm 2, \ldots, \pm n\}$ be an antipodal map with no 1-complementary pairs. Then there is an antipodal map $\lambda' : \mathcal{B}_1^{n-1} \to \{\pm 2, \ldots, \pm(n-1)\}$ with no 1-complementary pairs.*

Proof. Let $\lambda : \mathcal{B}_1^n \to \{\pm 2, \ldots, \pm n\}$ be an antipodal map with no 1-complementary pairs, and let $\ell = \lambda(\{n\}, \emptyset)$. We will define an antipodal map $\lambda' : \mathcal{B}_1^{n-1} \to \{\pm 2, \ldots, \pm n\} \setminus \{\pm \ell\}$. Let $(A, B) \in \mathcal{B}_1^{n-1}$. The value $\lambda'(A, B)$ will be defined by cases.

Case 1: If $(A, B) \in \mathcal{B}_1^{n-1}$ with $|A| = |B| = 1$, then $\lambda'(A, B) = \lambda(A, B)$.

Case 2: If $(A, \emptyset) \in \mathcal{B}_1^{n-1}$, then $\lambda'(A, \emptyset)$ is defined by cases:

Case 2a: If $\ell \notin \{\lambda(X, \emptyset) : \{n-1\} \preceq X \preceq A\}$, then define $\lambda'(A, \emptyset)$ to be $\lambda(A, \emptyset)$.

Case 2b: If case 2a does not apply, then define $\lambda'(A, \emptyset)$ to be $\lambda(A, \{n\})$.

Case 3: If $(\emptyset, B) \in \mathcal{B}_1^{n-1}$, then $\lambda'(\emptyset, B)$ is defined to be $-\lambda'(B, \emptyset)$, where $\lambda'(B, \emptyset)$ has already been defined by case 2.

The map λ' is antipodal because λ is.

Claim 3.28. *The map λ' never maps to ℓ or $-\ell$.*

The argument splits into cases.

- Suppose $(A, B) \in \mathcal{B}_1^{n-1}$, with $|A| = |B| = 1$. Then $\lambda'(A, B) = \lambda(A, B)$. Since $|A| = 1$, $\{n\} \preceq A$, and since $B \in \binom{n-1}{1}$, it follows that $\{n\} \cap B = \emptyset$. Additionally $\emptyset \preceq B$, and $\emptyset \cap A = \emptyset$. Therefore $(\{n\}, \emptyset) \preceq (A, B)$. Since λ has no 1-complementary pairs, and $\lambda(\{n\}, \emptyset) = \ell$, it follows that $\lambda(A, B) \neq -\ell$. Therefore $\lambda'(A, B) \neq -\ell$. Because λ' is antipodal, this also proves $\lambda'(A, B) \neq \ell$.

- Suppose $(A, \emptyset) \in \mathcal{B}_1^{n-1}$, and $\lambda'(A, \emptyset)$ is assigned by case 2a. For case 2a to apply, it must be that $\lambda(A, \emptyset) \neq \ell$. Furthermore, $\lambda(\{n\}, \emptyset) = \ell$, $(\{n\}, \emptyset) \preceq (A, \emptyset)$, and the fact that λ has no 1-complementary pairs imply that $\lambda(A, \emptyset) \neq -\ell$. Therefore, $\lambda'(A, \emptyset) = \lambda(A, \emptyset) \neq \pm \ell$.

- Suppose $(A, \emptyset) \in \mathcal{B}_1^{n-1}$, and $\lambda'(A, \emptyset)$ is assigned by case 2b. This implies that there is some $X \in \binom{n-1}{1}$ with $\{n-1\} \preceq X \preceq A$ such that $\lambda(X, \emptyset) = \ell$. Note that $\{n-1\} \preceq X$ implies that $\{n\} \cap X = \emptyset$. Since $(X, \emptyset) \preceq (A, \{n\})$, it follows that $\lambda(A, \{n\}) \neq -\ell$. Since $\lambda(\emptyset, \{n\}) = -\ell$ and $(\emptyset, \{n\}) \preceq (A, \{n\})$ it follows that $\lambda(A, \{n\}) \neq \ell$. Thus $\lambda'(A, \emptyset) = \lambda(A, \{n\}) \neq \pm \ell$.

- Suppose $(\emptyset, B) \in \mathcal{B}_1^{n-1}$. Then $\lambda'(\emptyset, B) = -\lambda'(B, \emptyset)$, and we have shown above that $\lambda'(B, \emptyset) \neq \pm\ell$.

This completes the proof of Claim 3.28.

Claim 3.29. *The map λ' has no 1-complementary pairs.*

We show the contrapositive. The argument splits into cases.

- Suppose $(A_1, B_1) \preceq (A_2, B_2) \in \mathcal{B}_1^{n-1}$ with $|A_1| = |B_1| = |A_2| = |B_2| = 1$. Then $\lambda'(A_1, B_1)$ and $\lambda'(A_2, B_2)$ both are assigned by case 1. Thus,

$$\lambda(A_1, B_1) = \lambda'(A_1, B_1) = -\lambda'(A_2, B_2) = -\lambda(A_2, B_2)$$

  Therefore λ has a 1-complementary pair.

- Suppose $(A_1, \emptyset) \preceq (A_2, B_2) \in \mathcal{B}_1^{n-1}$, with $\lambda'(A_1, \emptyset)$ assigned by case 2a and $\lambda'(A_2, B_2)$ assigned by case 1. So $\lambda(A_1, \emptyset) = -\lambda(A_2, B_2)$. Thus λ has a 1-complementary pair.

- Suppose $(A_1, \emptyset) \preceq (A_2, B_2) \in \mathcal{B}_1^{n-1}$, with $\lambda'(A_1, \emptyset)$ assigned by case 2b and $\lambda'(A_2, B_2)$ assigned by case 1. So $\lambda(A_1, \{n\}) = -\lambda(A_2, B_2)$. Since $(A_1, \{n\}) \preceq (A_2, B_2)$, it follows that λ has a 1-complementary pair.

- Suppose $(A_1, \emptyset) \preceq (A_2, \emptyset) \in \mathcal{B}_1^{n-1}$, with $\lambda'(A_1, \emptyset)$ and $\lambda'(A_2, \emptyset)$ both assigned by case 2a. So then $\lambda(A_1, \emptyset) = -\lambda(A_2, \emptyset)$, hence λ has a 1-complementary pair.

- Suppose $(A_1, \emptyset) \preceq (A_2, \emptyset) \in \mathcal{B}_1^{n-1}$, with $\lambda'(A_1, \emptyset)$ and $\lambda'(A_2, \emptyset)$ both assigned by case 2b. So then $\lambda(A_1, \{n\}) = -\lambda(A_2, \{n\})$, hence λ has a 1-complementary pair.

- Suppose $(A_1, \emptyset) \preceq (A_2, \emptyset) \in \mathcal{B}_1^{n-1}$, with $\lambda'(A_1, \emptyset)$ assigned by case 2a and $\lambda'(A_2, \emptyset)$ assigned by case 2b. Thus,

$$\lambda(A_1, \emptyset) = \lambda'(A_1, B_1) = -\lambda'(A_2, B_2) = -\lambda(A_2, \{n\})$$

  and since $(A_1, \emptyset) \preceq (A_2, \{n\})$, it follows that λ has a 1-complementary pair.

- Suppose $(A_1, B_1) \preceq (A_2, \emptyset) \in \mathcal{B}_1^{n-1}$ where $|A_1| = |B_1| = 1$. This impossible, because $B_1 \preceq \emptyset$, and no set of cardinality 1 precedes the emptyset under the partial order $\preceq$.

- Suppose $(A_1, \emptyset) \preceq (A_2, \emptyset) \in \mathcal{B}_1^{n-1}$, and $\lambda'(A_1, \emptyset)$ is assigned by case 2b and $\lambda'(A_2, \emptyset)$ is assigned by case 2a. Then there exists an $X \in \binom{n-1}{1}$ such that $\{n-1\} \preceq X \preceq A_1$ and $\lambda(X, \emptyset) = \ell$. Since $A_1 \preceq A_2$, it follows that $\{n-1\} \preceq X \preceq A_2$. This implies that $\lambda'(A_2, \emptyset)$ is not assigned by case 2a, so this case is impossible.

- Suppose $(A_1, \emptyset) \preceq (\emptyset, B_2) \in \mathcal{B}_1^{n-1}$. This is impossible, because $A_1 \preceq \emptyset$ implies that $A_1 = \emptyset$, but $(\emptyset, \emptyset) \notin \mathcal{B}_1^{n-1}$.

- The remaining cases involving case 3 of the definition of λ' follow from above, using the fact that if $(A_1, B_1) \preceq (A_2, B_2)$ form a 1-complementary pair, then $(B_1, A_1) \preceq (B_2, A_2)$ also form a 1-complementary pair.

This completes the proof of Claim 3.29. Claims 3.28 and 3.29 suffice to prove Lemma 3.27. □

We are now ready to sketch the proof of polynomial size extended Frege proofs of Tucker_1^n.

Proof of Theorem 3.26. To prove $\text{Tucker}_1^n(\vec{p})$, where $\vec{p}$ is a set of propositional variables encoding a map λ, we introduce by extension new variables $\vec{p}'$ to encode λ' as in Lemma 3.27. It is straightforward to see that the definition of λ' from λ can be carried out by polynomial size formulas. Furthermore, it is straightforward to argue that there are polynomial size proofs of $\neg\text{Tucker}_1^n(\vec{p}) \to \neg\text{Tucker}_1^{n-1}(\vec{p}')$ by formalizing the argument of Lemma 3.27. This process is repeated, introducing new propositional variables each round, until the proof reaches $\neg\text{Tucker}_1^2(\vec{p}'')$. From here, the proof concludes with a constant size proof of $\text{Tucker}_1^2(\vec{p}'')$. □

Chapter 3, in full, is a reprint of material that has been submitted for publication. Aisenberg, James; Bonet, Maria L.; Buss, Sam; Cračiun, Adrian; Istrate, Gabriel. The dissertation author was the primary investigator and author of this paper.

Chapter 4

2-D Tucker is PPA complete

4.1 Introduction

PPA and PPAD are classes of total NP search problems introduced by Papadimitriou [53]. The class PPA consists of the search problems reducible to the parity principle for undirected graphs, whereas the class PPAD consists of those reducible to the parity principle for directed graphs. The class PPAD has many complete problems from diverse areas of mathematics: Brouwer's theorem and Sperner's lemma in topology [53], Nash equilibria in game theory [26, 20, 21], and others. As discussed by [53, 27], several natural problems are known to be in PPA but not known to be in PPAD. One example is the Smith theorem about Hamiltonian cycles in cubic graphs [61]. Another is the integer factoring problem [12, 42]. However, few natural problems have been shown to be PPA-complete. By definition, the canonical problem LEAF is PPA-complete. For natural topological problems, it has been shown that Sperner's lemma and Tucker's lemma on two-dimensional non-orientable manifolds can be PPA-complete [35, 33, 27]. In addition, Deng et al. [27] show they are PPA-complete in the Möbius band, in two-dimensional projective space, and in the Klein bottle.

In this paper we show that the 2-D TUCKER search problem is PPA-complete. This is the usual

Tucker search problem in Euclidean space as defined by Papadimitriou [53]. This was erroneously claimed to be in PPAD by [53]. That paper used an argument by Freund and Todd [32] (a similar argument is given by [50]) to show that TUCKER is in PPA; it was then claimed that directionality techniques of Freund [30, 31] can put

TUCKER into PPAD. This last part is incorrect, as is discussed more in Section 4.3. However, the argument in [53] that TUCKER is in PPA is correct; likewise, the proofs that SPERNER and BROUWER are PPAD-complete are also correct.

The 3-D TUCKER search problem was shown in [53] to be hard for PPAD. Subsequently, it was shown that 2-D TUCKER is PPAD-hard [52]. This was extended by [28] to show that k-D TUCKER is PPAD-hard for all fixed $k \geq 2$. We improve these constructions to establish the following:

Theorem 4.1. 2-D TUCKER *is* PPA*-complete under many-one reductions. The same holds for* k-D TUCKER *for all* $k \geq 2$.

It follows that 2-D TUCKER is in PPAD if and only if PPAD = PPA. In the Type II (oracle) setting, it is known that PPAD $\neq$ PPA [7]. However, it is open whether these classes are equal in the non-relativized setting.

We write BORSUK–ULAM for the search problem associated with the Bursuk–Ulam theorem. Since BORSUK–ULAM and TUCKER are many-one reducible to each other [50, 53], another consequence of Theorem 4.1 is:

Corollary 4.2. BORSUK–ULAM *is* PPA*-complete.*

The search problems NECKLACE SPLITTING and DISCRETE HAM SANDWICH are known to be many-one reducible to TUCKER [50, 53]. From this, we know they are in PPA; it is now open whether they are in PPAD:

Question 4.3. *Is* NECKLACE SPLITTING *in* PPAD*, or* PPA*-complete? Is* DISCRETE HAM SANDWICH *in* PPAD*, or* PPA*-complete? Are they* PPAD*-hard?*

The octahedral Tucker lemma is a special case of the Tucker lemma in which the dimension k varies and the triangulation is the first barycentric subdivision of the k-dimensional hypercube. Thus, the size of the triangulation cannot be increased without also increasing the dimension (and the number of available labels). For the precise statement of the octahedral Tucker lemma, see [49, 63] or [3]. As a special case of TUCKER, the OCTAHEDRAL TUCKER search problem is known from [53] to be in PPA. This leaves open the following (also asked by [52]):

Question 4.4. *Is* OCTAHEDRAL TUCKER PPA*-complete? Is it in* PPAD*? Is it* PPAD*-hard?*

As already mentioned, it is open whether problems such as integer factoring, or Smith's theorem on cubic graphs give PPA-complete TFNP search problems. Papadimitriou [53] and Grigni [35] mention the Smith problem as a candidate for a PPA-complete problem that does not have a Turing machine explicitly encoded in its input.

4.1.1 Definitions

We now briefly review the search problems discussed in this paper. We first state the general form of Tucker's lemma, and then give the "rectangular" 2-D version that we will actually work with. For more information about Tucker's lemma and triangulations, see [50]. Let $B^k \subset \mathbb{R}^k$ be the closed k-dimensional ball, and S^{k-1} be its boundary. A triangulation T of B^k is antipodally symmetric if it is antipodally symmetric on the boundary — that is, if each simplex $\sigma \in T \cap S^{k-1}$ has the property that $-\sigma \in T$, where the negation of a simplex is the negation of each of its vertices. The set $V(T)$ of vertices of T is the set of 0-simplices in T.

Theorem 4.5 (Tucker's lemma)**.** *Let T be an antipodally symmetric triangulation of B^k, and let $\lambda : V(T) \to \{\pm 1, \ldots, \pm k\}$ be a function with the property that $\lambda(-v) = -\lambda(v)$ for all $v \in S^{k-1}$. Then there exists a 1-simplex $\{v_1, v_2\}$ in T with $\lambda(v_1) = -\lambda(v_2)$.*

To simplify our constructions, we will work with a rectangular 2-D version of Tucker's lemma, following Pálvölgyi [52]. For m a natural number, define $[m] = \{1, \ldots, m\}$.

Definition 4.6. Let $m \geq 2$. An *instance of the* 2-D TUCKER *search problem* is a function $\lambda : [m] \times [m] \to \{\pm 1, \pm 2\}$ with the property that for $1 \leq i, j \leq m$, $\lambda(i, 1) = -\lambda(m{-}i{+}1, m)$ and $\lambda(1, j) = -\lambda(m, m{-}j{+}1)$. A *solution* to such an instance of 2-D TUCKER is a pair of vertices (x_1, y_1), (x_2, y_2) with $|x_1 - x_2| \leq 1$ and $|y_1 - y_2| \leq 1$ such that $\lambda(x_1, y_1) = -\lambda(x_2, y_2)$. A solution (x_1, y_1), (x_2, y_2) is called a *complementary pair*.

Two points $(i, 1)$ and $(m{-}i{+}1, m)$ are called *antipodal*. Likewise, $(1, j)$ and $(m, m{-}j{+}1)$ are *antipodal*.

The $m \times m$ rectangular grid can be triangulated by the addition of diagonals, so it is clear that the existence of a solution to the 2-D TUCKER search problem is guaranteed by Tucker's lemma.

Definition 4.7. An *instance of the* LEAF *search problem* is an undirected graph G where

each node has degree at most 2, and there is a given ("standard") leaf ℓ with degree 1. A *solution* to LEAF is any other node of G with degree 1.

The class PPA is the set of total NP search problems reducible to LEAF under polynomial time many-one reductions [53]. As usual, we envision 2-D TUCKER and LEAF as Type II search problems in the sense of [7]. This means that instances of the search problems are exponentially big and are given by oracles: For 2-D TUCKER, the oracle specifies the values of the function λ. For LEAF, the oracle specifies the neighbors of any given node. In the Type II setting, it is known that PPAD is a proper subset of PPA.

4.2 Reduction from Leaf

We now show that 2-D TUCKER is PPA-hard. Since 2-D TUCKER is in PPA, this suffices to establish Theorem 4.1.

Theorem 4.8. 2-D TUCKER *is* PPA*-hard under many-one reductions.*

Proof. We give a reduction from LEAF. Let G be an instance of LEAF. We will describe λ, a labelling of the $m \times m$ grid with labels $\{\pm 1, \pm 2\}$. We will take $m = 4 \cdot 13 \cdot |G|$, where $|G|$ is the number of nodes in G. Our task is to define the values of $\lambda(i, j)$ for (i, j) a point on the $m \times m$ rectangular grid. The domain of λ will be referred to as *the grid*, and points (i, j) on the grid will be called *grid nodes*.

The reduction is similar to constructions of Papadimitriou [53] and especially Pálvölgyi [52]. The vast majority of the grid will be labelled with 1's (this is called the "environment"). The remainder of the grid will be filled with "wires": a wire consists of a strip of grid nodes of width three; the central "conductor" has label -1 and "insulators" on either side have labels ± 2. Wires are always directional. When travelling in the forward direction, the insulator on the left always has label 2, and the insulator on the right always has label -2.

We generally avoid exposing the conductor to the environment, as this would create complementary pairs between the conductor (-1) and the environment (1). We will route the wire in such a way that regions corresponding to solutions of G are the only wires exposed to the environment.

The grid is partitioned into 13×13 squares called *tiles*. A tile on the boundary is called a *boundary tile*. Two boundary tiles are *antipodal* if one of them contains some

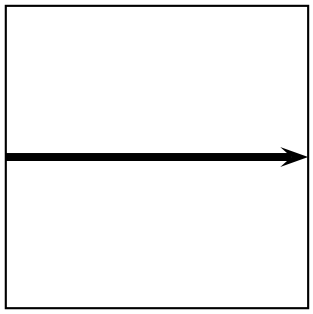
grid nodes antipodal to some grid nodes in the other. Specifically, this happens when the right column (resp., top row) of nodes in one tile are antipodal to the left column (resp., bottom row) of nodes in the other tile. In this case, since λ must be antipodal, the λ values of the nodes in the right column (resp., top row) of the first tile are the negations of the λ values of the nodes in the left column (resp., bottow row) in reverse order.

The schematic representation and its realization on the grid of a horizontal wire are shown in Figure 4.1. In figures representing the grid, 1's are indicated with blank space. The tile for the horizontal wire in the opposite direction can be obtained from the tile in Figure 4.1 by rotating 180°, or alternatively by reflecting about the horizontal axis. The tiles for the vertical wires can be obtained by rotating the horizontal ones 90°. Our tiles will typically have the conductor meet the edge of the tile at row 7 or column 7.

Notice that two wires can be in adjacent tiles without creating a complementary pair as long as they either are parallel or are joined head to tail. However, wires joined head to head or tail to tail do create complementary pairs, because the insulator labelled 2 is adjacent to the insulator labelled -2.

Recall that one node of G is given as the *standard* leaf ℓ, a degree 1 node. All other nodes $x, y, \ldots$ of G have degree ≤ 2; those of degree 1 are solutions to G as an instance of Leaf. Each node of G other than ℓ is assigned a region in the grid with two exposed edges: the *inbound edge* and the *outbound edge*, as pictured in Figure 4.2(a). The idea for our construction is that, when x has degree 2, the two exposed edges of x are wired to the edges of the two neighbors of x. If x has degree 0, its inbound and outbound edges are connected to each other. If x has only one neighbor, then one edge of x is exposed to the environment, creating a complementary pair. This is the only way that a complementary pair is formed; thus any complementary pair for λ corresponds to a solution to the instance G of Leaf.

Sometimes we are able to attach an outbound edge of a node x to an inbound edge of a neighboring node y. This is pictured schematically in Figure 4.2(a). However, since G is undirected, we will sometimes need to connect an outbound edge of x to an outbound edge of y. As shown in Figure 4.3(a), this creates unwanted complementary pairs. We thus use instead the construction shown in Figure 4.3(b). The outbound edge of x is routed "across the boundary", where it reverses direction (we shall see in

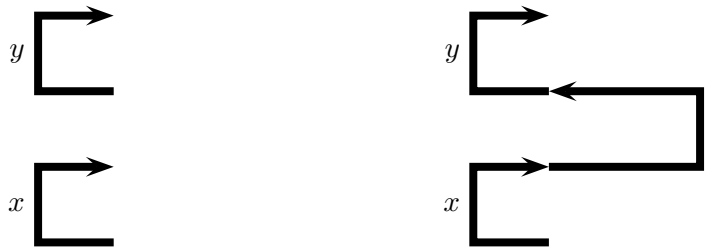


(a) Schematic representation

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1														1
2														2
3														3
4														4
5														5
6	2	2	2	2	2	2	2	2	2	2	2	2	2	6
7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	7
8	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	8
9														9
10														10
11														11
12														12
13														13
	1	2	3	4	5	6	7	8	9	10	11	12	13	

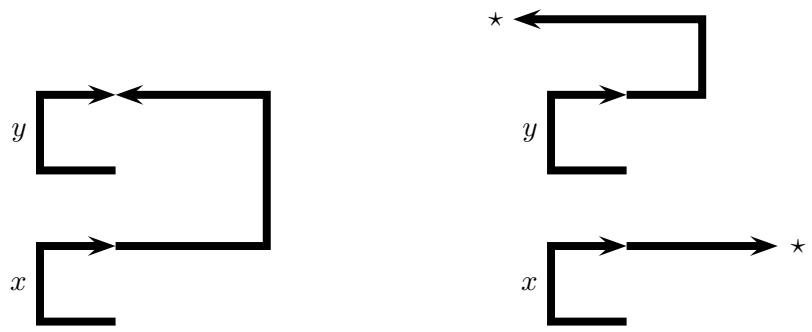
(b) Realization

Figure 4.1: A horizontal wire. (a) shows the schematic representation. (b) shows its realization with values of the labelling λ. The center of the wire has labels -1; the insulator labels 2 are on the left-hand side of the wire as it is traversed in its forward direction. The blank space represents grid nodes with label values of 1.



(a) Two nodes x and y (b) Connection of outbound to inbound

Figure 4.2: Two nodes and their connection. (a) Each node of G is assigned a region in the grid with an inbound edge and an outbound edge. (b) The schematic representation of connecting the outbound edge of x to the inbound edge of y.



(a) Incorrect connection (b) Correct connection

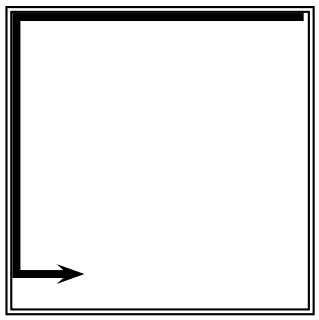
Figure 4.3: The outbound edge of x is connected to the outbound edge of y. When the boundary is crossed, the wire direction is reversed. The two locations in (b) marked with $\star$ are antipodal on the boundary.

Figure 4.5 how the reversal works), and then continues on to meet the outbound edge of y. A similar construction works to join an inbound edge of x to an inbound edge of y.

The rest of the proof shows how to apply the ideas behind the schematic representations shown in Figures 4.2(b) and 4.3(b) to define the labelling λ. For this, we must describe how the boundary is labelled, how a wire can cross the boundary and reverse direction, how two wires can cross each other in the grid, and the global strategy for routing wires.

First, we consider how to label the boundary of the grid, while preserving the antipodal property of λ. The underlying construction is shown in Figure 4.4; however it will need modification for wires that cross the boundary (as in Figures 4.3(b), 4.5 and 4.7). The boundary is represented by a double line in the figures. As shown in Figure 4.4, the outbound edge for the standard leaf ℓ emerges out the lower-left corner of the grid. The standard leaf, being of degree 1 in G, has only an outbound edge and no inbound edge. For simplicity, Figure 4.4(b) is shown scaled down to be 10×10 instead of its actual size of $m \times m$.

Let's describe the details of how a wire crosses the boundary and reverses direction. For this, refer first to Figures 4.3(b) and 4.5. There is a wire pointing to the right exiting the right boundary, and a wire pointing to the left exiting the left



(a) Schematic representation

	1	2	3	4	5	6	7	8	9	10	
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
2	-1	2	2	2	2	2	2	2	2	2	2
3	-1	2								1	3
4	-1	2								1	4
5	-1	2								1	5
6	-1	2								1	6
7	-1	2	2							1	7
8	-1	-1	-1							1	8
9	-2	-2	-2							1	9
10	1	1	1	1	1	1	1	1	1	1	10
	1	2	3	4	5	6	7	8	9	10	

(b) Realization (not to scale)

Figure 4.4: The boundary with no crossings. The figure shows a 10×10 grid, but in actuality it is an $m \times m$ grid. For this reason, we say that it is "not to scale".

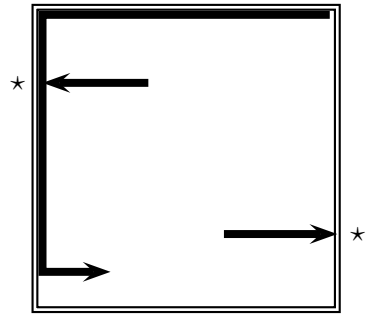
boundary. Recall that blank space indicates label values 1; thus, by examination, the antipodal property of λ holds on the boundary.

Figure 4.5 is "not to scale", and shows only label values needed for the wire crossing the boundary. The wire exiting to the left in Figure 4.5 is shown again inside its 13×13 tile in Figure 4.6. Note that it jogs downward two rows. This is to maintain the convention that the conductor of a wire, which is labelled -1, is in the middle row of its tile. The $\star$'s in Figure 4.5(b) mark the middle rows of antipodal tiles, thus antipodal boundary points of the grid. The left exiting wire, exiting from the antipodal tile, has label value 1 (not -1) on the middle row in the leftmost column. Figure 4.6 shows how this is implemented inside a 13×13 tile. The right column of Figure 4.6 has -1 in its middle position, so as to correctly match up with the continuation of the wire into the adjacent tile.

A similar construction allows wires to cross the boundary in the opposite direction. This is shown in Figures 4.7 and 4.8.

Since we are routing wires in a two-dimensional grid, wires will need to "cross each other". For this, following [52], we use the "avoided crossing" construction shown in Figure 4.9. We also need to let wires turn at right angles; this is very simple and shown in Figure 4.10.

We will now describe the global layout of the grid. Fix a total order $<$ on the nodes of G, with the standard leaf ℓ as the least element. The nodes are arranged

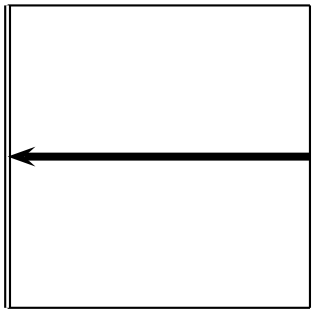


(a) Schematic representation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
2	-1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	-1	2																		1	3
4	-1	2																		1	4
5	-1	2																		1	5
6	-1	2																		1	6
7	2	2			-2	-2	-2	-2	-2											1	7
⋆ 8					-2	-1	-1	-1	-1											1	8
9	-2	-2	-2	-2	-2	-1	2	2	2											1	9
10	-1	-1	-1	-1	-1	-1	2													1	10
11	-1	2	2	2	2	2	2													1	11
12	-1	2												2	2	2	2	2	2	2	12
13	-1	2												-1	-1	-1	-1	-1	-1	-1	13 ⋆
14	-1	2												-2	-2	-2	-2	-2	-2	-2	14
15	-1	2																		1	15
16	-1	2																		1	16
17	-1	2	2	2	2															1	17
18	-1	-1	-1	-1	1															1	18
19	-2	-2	-2	-2	-2															1	19
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

(b) Realization (not to scale)

Figure 4.5: A wire crossing the boundary for joining two outbound edges. The realization (b) is "not to scale", and shown as 20×20. In actuality it is $m \times m$, and row 8 on the left is a row number i, and row 13 on the left is the antipodal point on row $m+1-i$. The wire jogs down two rows as it reaches the left boundary so as to make the antipodal property hold.



(a) Schematic representation

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	-1	2												1
2	-1	2												2
3	-1	2												3
4	-1	2												4
5	-1	2												5
6	2	2				-2	-2	-2	-2	-2	-2	-2	-2	6
7						-2	-1	-1	-1	-1	-1	-1	-1	7
8	-2	-2	-2	-2	-2	-2	-1	2	2	2	2	2	2	8
9	-1	-1	-1	-1	-1	-1	-1	2						9
10	-1	2	2	2	2	2	2	2						10
11	-1	2												11
12	-1	2												12
13	-1	2												13
	1	2	3	4	5	6	7	8	9	10	11	12	13	

(b) Realization

Figure 4.6: A boundary crossing tile.

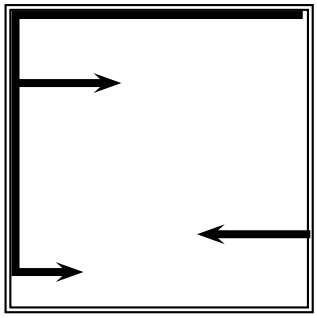
vertically in the lower-left quadrant of the grid according to the total order. Each inbound and outbound edge of each node has a horizontal lane that extends to the right boundary. At the tile antipodal to where the lane reaches the boundary, a new lane continues now in the upper half of the grid. Each inbound and outbound edge also has a vertical lane that extends from the top boundary to the bottom boundary. Each node of G has four horizontal lanes and two vertical lanes; since the lanes have width 13 and since $m = 4 \cdot 13 \cdot |G|$, the grid has sufficient space to hold the construction. The layout of the grid for a graph with three nodes is shown in Figure 4.11.

We will now describe how nodes are connected together. When x and y are neighbors in G, we will connect one edge of x in the grid with one edge of y in the grid. For this, we select either the outbound or inbound edge of x and either the outbound or inbound edge of y. This works even though G is undirected.

1. If x is a node in G with two neighbors y and z, with $y < z$, then the outbound edge of x connects to y and the inbound edge of x connects to z.

2. If x is a node with no neighbors, then the outbound edge of x connects to the inbound edge of x.

3. If x is the standard leaf ℓ, then the outbound edge of x connects to its one neighbor y. In this case, x has no inbound edge.

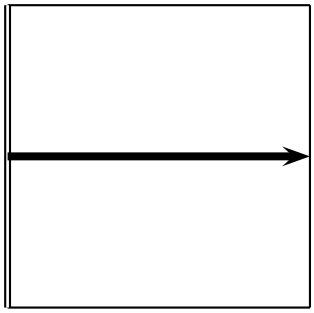


(a) Schematic representation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
2	-1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	-1	2																		1	3
4	-1	2																		1	4
5	-1	2	2	2	2	2	2	2	2											1	5
6	-1	-1	-1	-1	-1	-1	-1	-1	2											1	6
7	-2	-2	-2	-2	-2	-2	-2	-1	2	2	2									1	7
⋆ 8							-2	-1	-1	-1	-1									1	8
9	2	2					-2	-2	-2	-2	-2									1	9
10	-1	2																		1	10
11	-1	2																		1	11
12	-1	2												-2	-2	-2	-2	-2	-2	-2	12
13	-1	2												-1	-1	-1	-1	-1	-1	-1	13 ⋆
14	-1	2												2	2	2	2	2	2	2	14
15	-1	2																		1	15
16	-1	2																		1	16
17	-1	2	2	2	2															1	17
18	-1	-1	-1	-1	1															1	18
19	-2	-2	-2	-2	-2															1	19
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

(b) Realization (not to scale)

Figure 4.7: A wire crossing the boundary for joining two inbound edges.

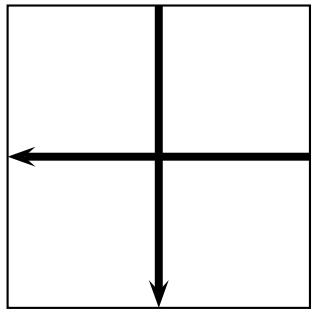


(a) Schematic representation

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	-1	2												1
2	-1	2												2
3	-1	2												3
4	-1	2	2	2	2	2	2	2						4
5	-1	-1	-1	-1	-1	-1	-1	2						5
6	-2	-2	-2	-2	-2	-2	-1	2	2	2	2	2	2	6
7						-2	-1	-1	-1	-1	-1	-1	-1	7
8	2	2				-2	-2	-2	-2	-2	-2	-2	-2	8
9	-1	2												9
10	-1	2												10
11	-1	2												11
12	-1	2												12
13	-1	2												13
	1	2	3	4	5	6	7	8	9	10	11	12	13	

(b) Realization

Figure 4.8: A boundary crossing tile.

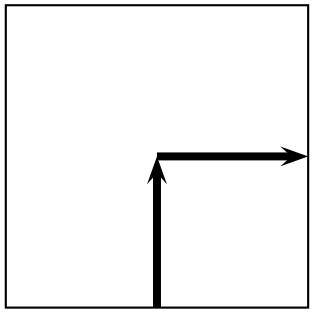


(a) Schematic representation

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1						-2	-1	2						1
2		-2	-2	-2	-2	-2	-1	2						2
3		-2	-1	-1	-1	-1	-1	2						3
4		-2	-1	2	2	2	2	2						4
5		-2	-1	2										5
6	-2	-2	-1	2						-2	-2	-2	-2	6
7	-1	-1	-1	2						-2	-1	-1	-1	7
8	2	2	2	2						-2	-1	2	2	8
9										-2	-1	2		9
10						-2	-2	-2	-2	-2	-1	2		10
11						-2	-1	-1	-1	-1	-1	2		11
12						-2	-1	2	2	2	2	2		12
13						-2	-1	2						13
	1	2	3	4	5	6	7	8	9	10	11	12	13	

(b) Realization

Figure 4.9: An avoided crossing. This effectively allows wires to cross each other.



(a) Schematic representation

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1														1
2														2
3														3
4														4
5														5
6						2	2	2	2	2	2	2	2	6
7						2	-1	-1	-1	-1	-1	-1	-1	7
8						2	-1	-2	-2	-2	-2	-2	-2	8
9						2	-1	-2						9
10						2	-1	-2						10
11						2	-1	-2						11
12						2	-1	-2						12
13						2	-1	-2						13
	1	2	3	4	5	6	7	8	9	10	11	12	13	

(b) Realization

Figure 4.10: A right angle.

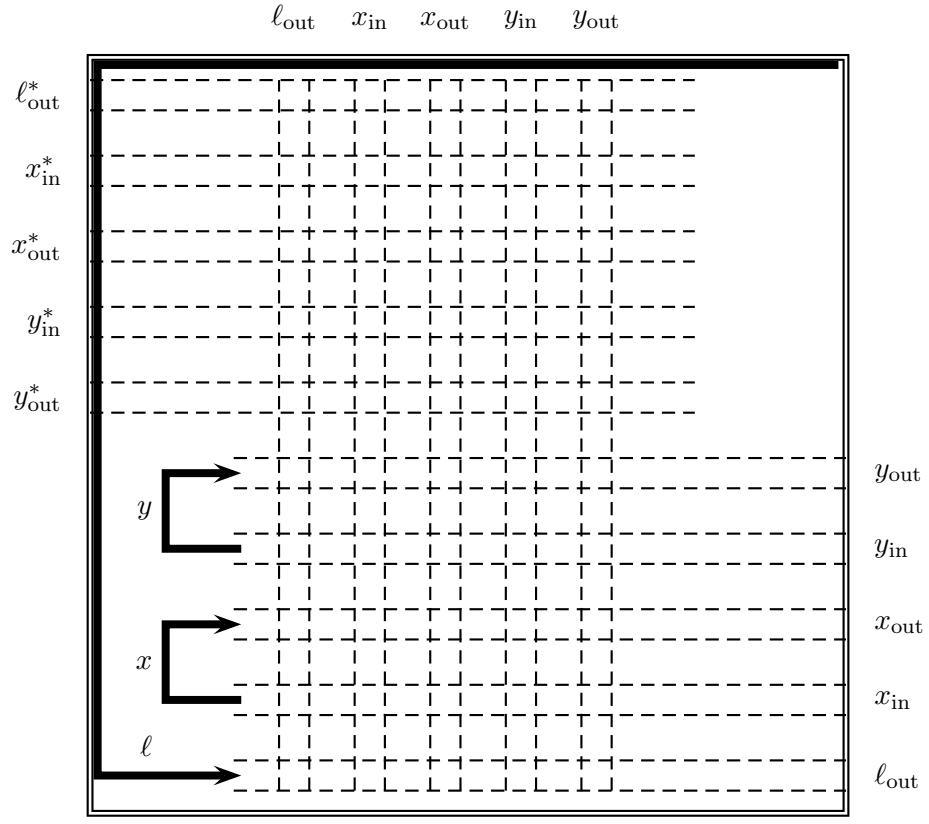


Figure 4.11: Global layout of the grid.

4. If x is a node that is not the standard leaf with only one neighbor y, then the outbound edge of x connects to y, and the inbound edge of x is exposed to the environment. This will create a complementary pair at x's inbound edge as desired.

Suppose that x and y are neighbors in G, with $x < y$. We will describe how x and y are connected together:

1. If x's outbound edge connects to y's inbound edge, then we add a wire that takes the following route: x's outbound edge, x's horizontal outbound lane, x's vertical outbound lane, y's horizontal inbound lane, and finally y's inbound edge.

2. If y's outbound edge connects to x's inbound edge, then we add a wire that takes the following route: y's outbound edge, y's horizontal outbound lane, x's vertical inbound lane, x's horizontal inbound lane, and finally x's inbound edge.

3. If x's and y's outbound edges connect together, then half of the route is as follows: start at x's outbound edge, continue along x's horizontal outbound edge to the boundary. The other half of the route is as follows: start at y's outbound edge, continue along y's horizontal outbound lane to x's vertical outbound lane. Follow x's vertical outbound lane up to x's reflected outbound horizontal lane. Continue along x's reflected outbound horizontal lane to the boundary.

4. If x's and y's inbound edges are connected together, then one path originates from the boundary at x's horizontal inbound lane into x's inbound edge. The other path originates at the antipodal boundary point, travels along x's reflected horizontal inbound path to x's vertical inbound lane, down to y's horizontal inbound lane, and into y's inbound edge.

If x is a node of G with no neighbors, then we must connect the outbound edge of x to the inbound edge of x. This is done by the following route: x's outbound edge to x's horizontal outbound lane, to x's vertical outbound lane, to x's horizontal inbound lane, to x's inbound edge.

The paths formed by the above procedure can cross each: if so, we use the avoided crossing construction. By inspection, at most two paths can intersect a given tile, and if so, they meet at right angles.

Claim 4.9. *The only complementary pairs in the grid that are formed by the above construction are at the inbound edge of a node $x \neq \ell$ of degree 1 in G.*

Claim 4.9 is obvious by inspection of the construction. It follows that there is a polynomial time method to find a degree 1 node $x \neq \ell$ in G, given the location of a complementary pair for λ in the grid.

Claim 4.10. *It is possible to decide in polynomial time which tile to place at a given position in the grid using only constantly many oracle queries to G.*

Claim 4.10 follows from the fact that a given tile can lie in at most two "lanes". To illustrate this, consider the following example. Consider a tile that is at the intersection of x's horizontal inbound lane, and y's vertical outbound lane. We query G about x's neighbors which, say, are $u_1 < u_2$. Thus the inbound edge of x connects to u_2. We then query G about u_2's neighbors in order to decide if x connects to u_2 at u_2's inbound or outbound edge. With this information, we can decide if the route taken on x's horizontal inbound lane passes through the tile, does not pass through this tile, or turns at a right angle at the tile. We will similarly query G about y's two neighbors, say $v_1 < v_2$, and then query G about v_1's neighbors. This is enough to determine what happens in the vertical lane. With all this information, we can decide how to assign λ values for this tile, namely as a blank tile, a horizontal wire, a vertical wire, a right angle, or an avoided crossing. This is accomplished with only queries to only four nodes G.

This completes the proof of Theorem 4.8 and hence Theorem 4.1. □

4.3 Tucker, Leaf, and LeafD

This section gives a quick sketch of the reduction from TUCKER to LEAF. The constructions are due to Freund [30, 31], Freund and Todd [32], Matoušek [50], and Papadimitriou [53]. However, it seems useful to repeat the arguments here to the illustrate the reduction from TUCKER to LEAF, as well as to point out why it does not give a reduction to the directed analogue LEAFD of LEAF. This illustrates the failure in the earlier argument that TUCKER is PPAD-complete. As we shall see, the reduction gives a graph G in which many of the edges can be coherently directed, but edges which connect antipodal simplices cannot be coherently directed.

Let a triangulation T of the unit ball in the L^1-norm and a labelling λ satisfy the hypotheses of the Tucker lemma in dimension 2. Further suppose (for sake of a contradiction) that there are no complementary 1-simplices in T. A 1-simplex in T is just an edge in T. Without loss of generality, refining T if necessary, we may assume that

the triangulation contains the origin, and no 1-simplex in T has endpoints in distinct quadrants. A simplex is defined to be *happy* if it is a 1-simplex and certain labels are present on its vertices, according to what region the simplex lies in, as given by the following table:

Midpoint of 1-simplex lies in	Required labels (λ values)
Positive x-axis	1
Negative x-axis	-1
Positive y-axis	2
Negative y-axis	-2
First quadrant (interior)	1, 2
Second quadrant (interior)	-1, 2
Third quadrant (interior)	-1, -2
Fourth quadrant (interior)	1, -2

By our assumptions on T, each 1-simplex has its interior lying in exactly one region. Without loss of generality, the origin has label 1, so the 1-simplex that lies in the positive x-axis with one endpoint at the origin is happy. This 1-simplex is called the *initial* 1-simplex.

A graph G is defined on the happy 1-simplices of T. The initial 1-simplex is a node of degree 1. All other happy 1-simplices will have degree 2 in G. The graph G is undirected; nonetheless, many (but not all) of its edges can be coherently directed. The different types of directed edges between happy 1-simplices are as shown in Figures 4.12 and 4.13: the curved arrows connect happy 1-simplicies; the arrows indicate the directions. For example, an edge in G connecting two happy 1-simplices in the first quadrant is directed so that the vertex with label 1 is on the left, and the vertex with label 2 is on the right. Two adjacent happy 1-simplices that lie on an axis have their edge directed away from the origin (e.g., rightward on the positive x-axis, leftward on the negative x-axis, etc.).

There are additional undirected edges between antipodal happy 1-simplices which lie on the boundary of the ball. These are as follows:

1. If σ is a happy 1-simplex, and both vertices of σ are on the boundary, then σ has $-\sigma$ as a neighbor in G. Note $-\sigma$ is happy, since σ is. An example is illustrated in Figure 4.13 with a dashed curve.

2. If σ is happy, σ lies in a 1-dimensional region (the x- or y-axis), one of σ's vertices v is on the boundary of the ball, and v has the required label to make σ happy, then σ has neighbor τ, the unique (and happy) 1-simplex that has $-v$ as a vertex.

Under the assumption that there are no complementary 1-simplices, a straightforward case analysis shows that the initial 1-simplex has degree 1 in G, and all other nodes have degree 2.

Since the construction of G from T is constructive, and the presence of edges in G only depends locally on T, the above gives a many-one polynomial time reduction from from TUCKER to LEAF in the case of two dimensions. Higher dimensions work analogously, but require considering k-simplices also for $k > 1$.

The edges in G that are not on the boundary of the ball can be coherently oriented as illustrated in Figure 4.12. This can be easily checked in the two dimensional case, and the general case is carried out by Freund [30, 31]. However, the undirected edges connecting antipodal simplices cannot be directed coherently. For example, the dashed curve of Figure 4.12 cannot be directed without creating a 1-simplex with two incoming edges in G. It was exactly this ability to "reverse directions" by connecting antipodal simplices that was exploited in the proof of Theorem 4.1.

Acknowledgements. We thank Christos Papadimitriou, Dömötör Pálvölgyi, and Xiaotie Deng for comments on early drafts of this paper.

Chapter 4, in full, is a reprint of material that has been submitted for publication. Aisenberg, James; Bonet, Maria L.; Buss, Sam. The dissertation author was the primary investigator and author of this paper.

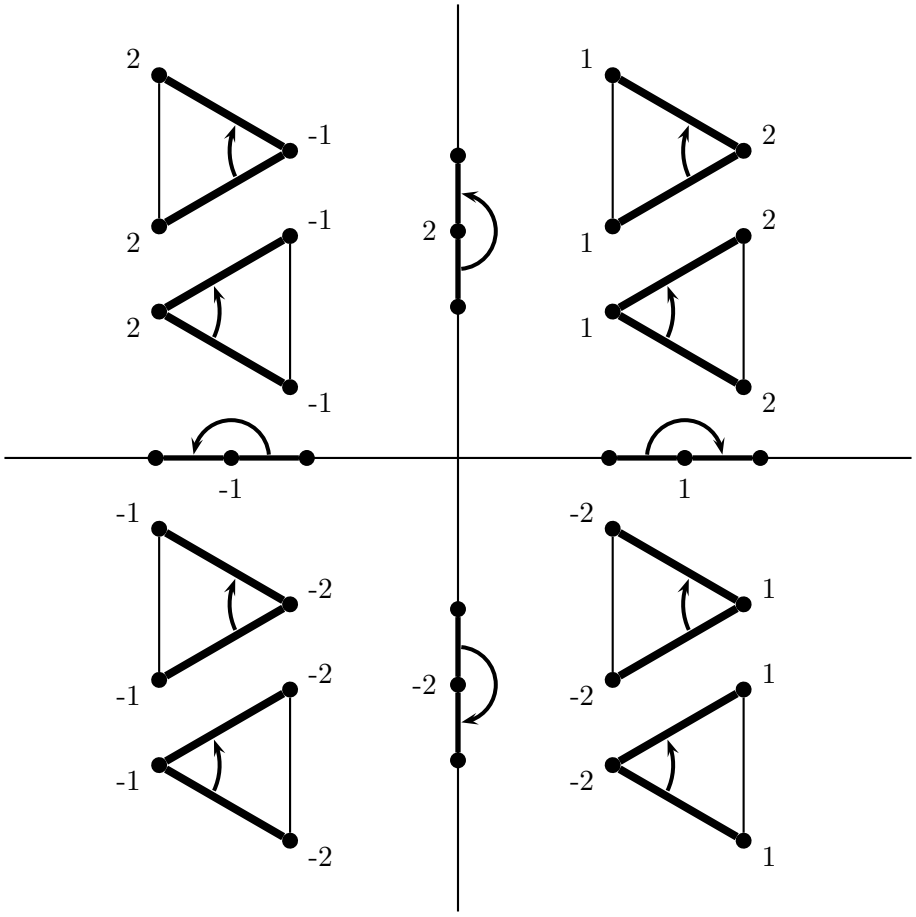


Figure 4.12: Happy 1-simplices, and their directed neighbors. 1-simplices which are happy are drawn with thick lines. Happy 1-simplices in quadrant I, II, III or IV (respectively) have their vertices labelled with a 1 and 2, with a -1 and 2, with a -1 and -2, or with a 1 and -2 (respectively). A 1-simplex in the positive x-axis, the positive y-axis, the negative x-axis, or the negative y-axis (respectively) have at least one vertex labelled 1, 2, -1, or -2 (respectively). The directed edges between happy vertices are shown by the curved arrows.

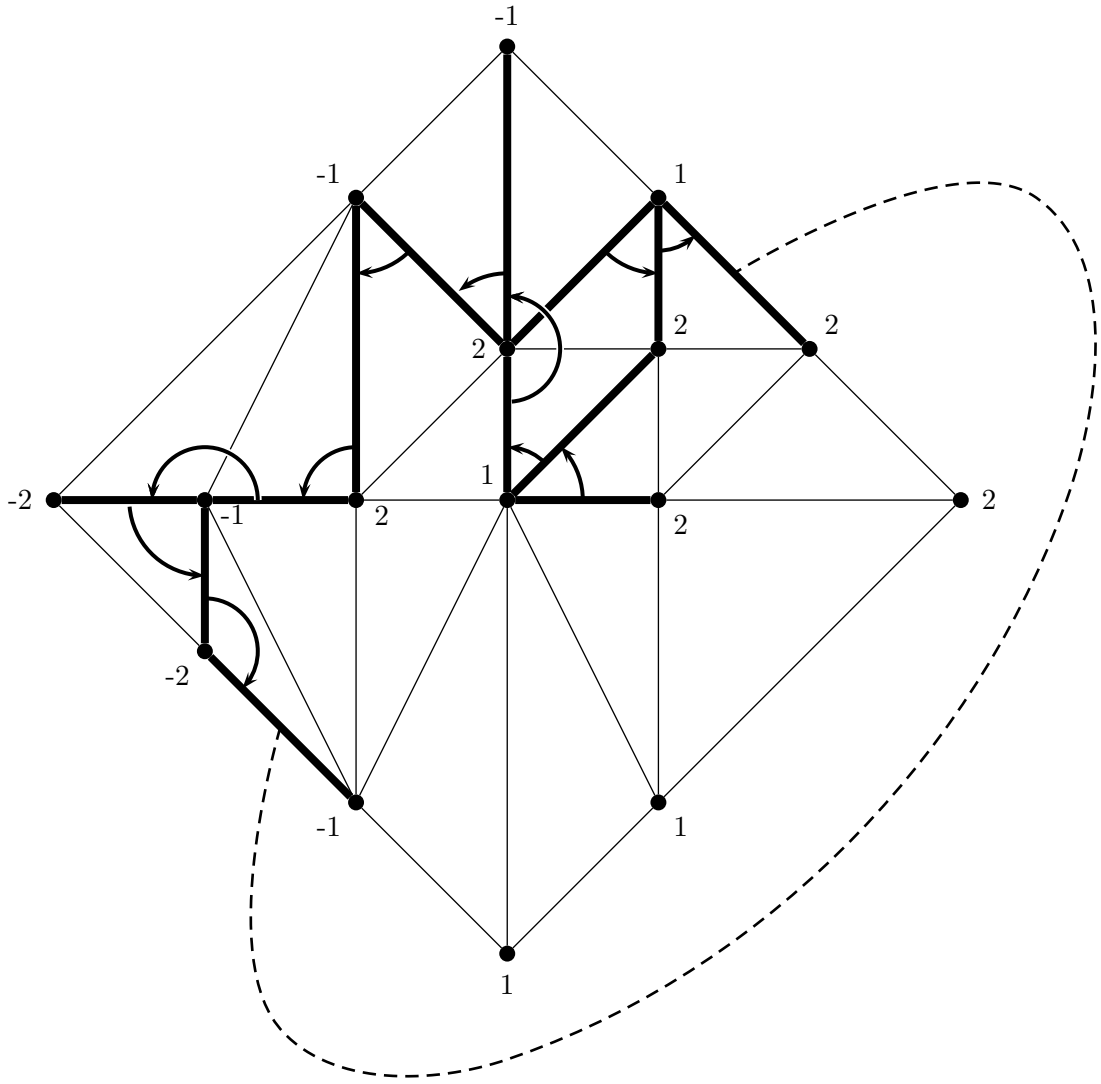


Figure 4.13: An example of an instance of TUCKER, and the graph G. Happy 1-simplices are indicated with thick lines. Arrows indicate the edges in G that can be directed. The dashed curve indicates an edge in G that is not given a direction; it connects a pair of antipodal happy 1-simplices.

Chapter 5

Total NP search problems

This chapter defines a number of total NP search problems based on the combinatorial principles discussed in the earlier chapters, establishes basic relationships between them, and poses a number of open questions. The new problems, based on Frankl's theorem, the truncated Tucker lemma, and the Kneser-Lovász theorem are shown to be PPP-hard, which is of interest because there are few known examples of PPP-hard problems.

5.1 Frankl's theorem

This section defines a total NP search problem based on Frankl's theorem (Theorem 2.1). One condition in the hypothesis of Frankl's theorem is that the matrix in question has distinct rows. For a total NP search problem, we must be able to recognize invalid inputs using only local information. Our search problem will take matrices with exponentially long rows as inputs. So one difficulty that we encounter with phrasing Frankl's theorem as a search problem is how to recognize that two exponentially long rows are identical in a way that is verifiable in polynomial time. One approach is to supply additional information about the matrix as part of the input to the problem. For example, a function $\text{Diff}(i, i')$ that inputs two indices for rows i and i' (with $i \neq i'$), and outputs an index of a column where the rows i and i' are different. A matrix has distinct rows iff such a $\text{Diff}(i, i')$ function exists. Moreover, the invalidity of an input is verifiable in polynomial time. To show that the input is invalid, it is only necessary to give two rows i and i' that the Diff says are supposed to disagree in some column j, but

do not. This approach could give a search problem based on Frankl's theorem, but the resulting search problem seems to be somewhat weak, as will be discussed below. To give a stronger search problem, we relax the requirement in the hypothesis of Frankl's theorem that the matrix in question has distinct rows. Instead, we will only require that certain rows of the matrix are distinct. The search problem we will define is based on the following corollary to Frankl's theorem. To state the corollary, recall from Section 2.1.1 that if r_1 and r_2 are rows in a matrix, then r_1 is equivalent to r_2 modulo column j if r_1 and r_2 differ only in column j.

Corollary 5.1. *Let t be a positive integer, and $m \leq \frac{2^t-1}{t}n$. It is not the case that there exists a 0/1 matrix A (not necessarily with distinct rows) such that for each column j, there is a set Q_j of distinct rows of A with $|Q_j| \geq 2^t$ and for each $r_1 \in Q_j$, there exists an r_2 in Q_j with r_1 equivalent to r_2 modulo column j.*

Proof. Suppose A is an $m \times n$ 0/1 matrix in violation of the statement of the theorem. Let $\widetilde{A}$ be the matrix obtained by deleting duplicate copies of rows (leaving the original copy). The resulting matrix is an $m' \times n$ 0/1 matrix with distinct rows where $m' \leq m$. For j a column, let P_j denote the set of rows r_1 of $\widetilde{A}$ such that there exists a row r_2 of $\widetilde{A}$ with r_1 equivalent to r_2 modulo column j. Observe that for each $r \in Q_j$, there is some $\widetilde{r} \in P_j$ where r and $\widetilde{r}$ are equal as rows (they may have different indices, however). This mapping is injective. Thus $|P_j| \geq 2^t$. Thus $\widetilde{A}$ violates Frankl's theorem. □

The search problem based on Frankl's theorem is as follows:

Definition 5.2. Let $t > 0$, and let $m \leq \frac{2^t-1}{t}n$. An instance of the *Frankl's theorem search problem*, FRANKL is a tuple of functions $(A, Q, \text{Diff}, \text{Pair})$ as follows:

$$A : [m] \times [n] \to \{0, 1\}$$
$$Q : [n] \times [2^t] \to [m]$$
$$\text{Diff} : [n] \times [2^t] \times [2^t] \to [n]$$
$$\text{Pair} : [n] \times [2^t] \to [2^t]$$

A solution to the search problem is one of the following:

1. a j, x, x' with $x \neq x'$, $\text{Diff}(j, x, x') = j'$, and $A(Q(j, x), j') = A(Q(j, x'), j')$,
2. a j, x where $\text{Pair}(j, x) = x'$ and $A(Q(j, x), j) = A(Q(j, x'), j)$, or

3. a j, x, j' where $j \neq j'$, $\text{Pair}(j, x) = x'$, and $A(Q(j, x), j') \neq A(Q(j, x'), j')$.

For an instance of FRANKL, we think of n as being exponentially large, and the functions A, P, Diff and Pair as given by function oracles (a type 2 search problem). The function A thus defines an exponentially large 0/1 matrix. The $Q(j, x)$ function defines the set Q_j in the corollary to Frankl's theorem above, a set of 2^t rows. These rows are supposed to be distinct, and have the property that if r_1 is a row in the set, then there is some $r_2 \neq r_1$ also in the set where r_1 is equivalent to r_2 modulo column j. To establish that the rows of $Q(j, x)$ are distinct, we supply a function $\text{Diff}(j, x, x')$, which gives a column j' where the rows $Q(j, x)$ and $Q(j, x')$ are supposed to be different. The $\text{Pair}(j, x)$ function is supposed to say that the row $Q(j, x)$ and the row $Q(j, \text{Pair}(j, x))$ are equivalent modulo column j.

The $t = 1$ case of Frankl's theorem is called Bondy's theorem. Bondy's theorem states that if A is a square matrix with distinct rows, there is a column that can be deleted so that the resulting matrix also has distinct rows. Or, in our relaxed form, Bondy's theorem states that if A is a square matrix, then there is a column that can be deleted such that the only identical rows of the resulting matrix were identical in A. Let BONDY be the special case of FRANKL where $t = 1$.

Theorem 5.3. BONDY *is* PPP*-hard.*

This is the same construction used to show that $I\Delta_0 + \Delta_0$-BONDY proves the pigeonhole principle [9].

Proof. We reduce PIGEON to BONDY. Let $f : [n] \to [n-1]$ be an instance of PIGEON. We construct an instance of BONDY as follows: let $A(i, j) = 1$ iff $f(j) = i$. For each j, if column j is deleted, say that rows $f(j)$ and n are identified. In other words, $Q(j, 0) = f(j)$, $Q(j, 1) = n$. Also $\text{Diff}(j, 0, 1) = \text{Diff}(j, 1, 0) = j$, and $\text{Pair}(j, 0) = 1$ and $\text{Pair}(j, 1) = 0$. It's clear that a solution to $(A, Q, \text{Diff}, \text{Pair})$ as an instance of BONDY immediately gives a solution to f as an instance of PIGEON. □

Question 5.4. *Is* BONDY *in* PPP*? Is* FRANKL *in* PPP*?*

Observe that the matrix in the above proof does not necessarily have distinct rows. Indeed, if the function f is not onto, then the row where every entry is 0 will occur multiple times. This is why we defined FRANKL without the requirement that all rows be distinct. If instead we were working with the distinct rows version of the

Frankl's theorem search problem, we could show that it was PPAD-hard, since the onto pigeonhole principle is PPAD-complete [13].

5.2 The octahedral Tucker lemma

This section discusses the octahedral Tucker lemma as a total NP search problem and the geometry of the octahedral ball. Pálvölgyi [52] noted that the octahedral Tucker lemma gives rise to a total NP search problem. We discuss it here only to provide some geometric intuition, which will be helpful when discussing the truncated Tucker lemma. Recall the octahedral ball $\mathcal{B}^n$:

$$\mathcal{B}^n = \{(A, B) : \ A, B \subseteq [n] \text{ and } A \cap B = \emptyset\}.$$

and that for two points $(A_1, B_1), (A_2, B_2) \in \mathcal{B}^n$, $(A_1, B_1) \subseteq (A_2, B_2)$ iff $A_1 \subseteq A_2$ and $B_1 \subseteq B_2$. Also recall the octahedral Tucker lemma (Theorem 3.15):

Theorem 5.5 (Octahedral Tucker lemma)**.** *If $\lambda : \mathcal{B}^n \to \{1, \pm 2, \ldots, \pm n\}$ is antipodal, then there are $(A_1, B_1) \subseteq (A_2, B_2) \in \mathcal{B}^n$ with $\lambda(A_1, B_1) = -\lambda(A_2, B_2)$.*

Finally, recall from Section 4.1 the Tucker lemma (Theorem 4.5) states:

Theorem 5.6 (Tucker's lemma)**.** *Let T be an antipodally symmetric triangulation of B^k, and let $\lambda : V(T) \to \{\pm 1, \ldots, \pm k\}$ be a function with the property that $\lambda(-v) = -\lambda(v)$ for all $v \in S^{k-1}$. Then there exists a 1-simplex $\{v_1, v_2\}$ in T with $\lambda(v_1) = -\lambda(v_2)$.*

The octahedral Tucker lemma is stated without making reference to a triangulation. To help understand the intuition behind the octahedral Tucker lemma, we describe this triangulation, which we denote T^n. In brief, T^n is the first barycentric subdivision of the standard triangulation of the n-ball in the ℓ_1-norm.

Let's briefly introduce the terminology needed to discuss triangulations. A set $S \subseteq \mathbb{R}^n$ is *convex* if for any two points $x, y \in S$, the line segment with endpoints x and y is a subset of S. For a set of points V, the *convex hull* of V, denoted $\text{conv}(V)$, is the minimum convex set containing V with respect to set inclusion. A set $V = \{v_0, v_1, \ldots, v_k\}$ is *affinely dependent* if there are real numbers $\alpha_0, \ldots, \alpha_k$, not all α_i's equal to 0 such that $\sum \alpha_i v_i = 0$ and $\sum \alpha_i = 0$. A set if *affinely independent* if it is not affinely dependent. The set $\text{conv}(V)$ is a *k-simplex* if $|V| = k + 1$ and V is affinely independent. For such a k-simplex, $\text{conv}(V)$, the elements of V are the *vertices* of

$\text{conv}(V)$. We will sometimes refer to a k-simplex by its vertex set. The simplex V' is a *face* of the simplex V if $V' \subseteq V$. If V' is a face of V and $V' \neq V$, then V' is a *proper face* of V.

A *simplicial complex* is a set of simplices X with the following properties: (1) if $\sigma \in X$ then any face of σ is also in X, and (2) if $\sigma, \tau \in X$ with $\sigma \cap \tau \neq \emptyset$ then $\sigma \cap \tau$ is a face of both σ and τ. A simplicial complex X is a *triangulation* of a set $S \subseteq \mathbb{R}^n$ if $\cup X = S$.

If $\sigma = \{v_0, \ldots, v_k\}$ is a k-simplex, define the *barycenter of* σ to be the point $\frac{1}{k+1}(v_0 + \cdots + v_k)$. In other words, the barycenter of a simplex is the average of its vertices. The barycenter of σ is denoted $\hat{\sigma}$. If X is a simplicial complex, the *barycentric subdivision of* X is the set of simplices of the form $\{\hat{\sigma}_0, \ldots, \hat{\sigma}_k\}$ where the σ_i's are simplices in X and σ_i is a proper face of σ_{i+1} for all i. Let X' denote the barycentric subdivision of X. The following theorem is well-known:

Theorem 5.7. *If X is a simplicial complex, then X' is a simplicial complex. Moreover, X and X' triangulate the same space.*

Proof. We prove the claim by induction on the number of simplices of X. For the base case, if X contains only one simplex, then that simplex is a point, and the claim is obvious. For the induction step, let σ be a simplex in X that is not the face of any other simplex in X. Then $Y = X \setminus \{\sigma\}$ is a simplicial complex, and the induction hypothesis applies. There are three kinds of simplices in X': simplices in Y', the 0-simplex $\{\hat{\sigma}\}$ and simplices whose vertex set is of the form $\rho \cup \{\hat{\sigma}\}$ where ρ is the vertex set of a simplex in Y'. It is straightforward to see that if τ_1 and τ_2 are two simplices in X' with $\tau_1 \cap \tau_2 \neq \emptyset$, then $\tau_1 \cap \tau_2$ is a face of both τ_1 and τ_2.

To prove the moreover, take any point x in the space triangulated by X. If x is in the space triangulated by Y, then x is in the space triangulated by Y', and so x is in the space triangulated by X'. Otherwise, x is in the interior of σ. If $x = \hat{\sigma}$, then it is clear that x is in the space triangulated by X', so suppose $x \neq \hat{\sigma}$. Consider the line passing through x and $\hat{\sigma}$. It intersects the proper faces of σ in two points. Let y be the unique point on a face of σ such that x is in the interior of the line segment connecting $\hat{\sigma}$ to y. Since y is in the space triangulated by Y, it is in the space triangulated by Y', so y is in the interior of some simplex $\rho \in Y'$. Furthermore, $\rho \cup \{\hat{\sigma}\}$ is in X', and x is in the interior of $\rho \cup \{\hat{\sigma}\}$. This shows that the space triangulated by X is a subset of the space triangulated by X'. The other direction is clear because every simplex in X'

is contained in some simplex in X. $\square$

We now describe a triangulation of the surface of the n-ball in the ℓ_1-norm. Let $x_1, \ldots, x_n$ be the standard basis in $\mathbb{R}^n$. The vertex sets of the simplices of the triangulation are non-empty subsets S of $\{\pm x_1, \ldots, \pm x_n\}$ with the property that $x \in S$ implies $-x \notin S$. It is clear that this is a triangulation of the surface of the n-ball. Call this triangulation X. Note that the simplices of this triangulation can be identified with elements in $\mathcal{B}^n \setminus \{(\emptyset, \emptyset)\}$. Moreover, the simplex identified with (A_1, B_1) is a face of the simplex identified with (A_2, B_2) iff $(A_1, B_1) \subseteq (A_2, B_2)$. Next, we take the barycentric subdivision of this simplicial complex, X'. The simplices of the resulting complex correspond to chains in $\mathcal{B}^n \setminus \{(\emptyset, \emptyset)\}$ in the $\subseteq$ partial order. In other words, if $(A_0, B_0) \subsetneq \cdots \subsetneq (A_k, B_k)$ then there is a simplex in the barycentric subdivision whose vertex set consisting of the barycenters of the simplices $(A_0, B_0), \ldots, (A_k, B_k)$. The triangulation T^n is obtained from X' by "filling in" the n-ball:

$$T^n := \{\{0\}\} \cup X' \cup \{\rho \cup \{0\} : \ \rho \in X'\}.$$

The vertices in T^n correspond to the points in $\mathcal{B}^n$, with $(\emptyset, \emptyset)$ corresponding to the 0-simplex at the origin. The octahedral ball in three dimensions is shown in Figure 5.1. One hemisphere of the three-dimensional octahedral ball is shown in Figure 5.2.

The Tucker lemma applies to triangulations that are antipodally symmetric on the boundary. In T^n, the origin is the only vertex that is not on the boundary, and if $(A, B) \neq (\emptyset, \emptyset)$, the vertex (A, B) is antipodal to the vertex (B, A). Thus, the requirement that for $(A, B) \neq (\emptyset, \emptyset)$, $\lambda(A, B) = -\lambda(B, A)$ enforces the boundary condition for the Tucker lemma.

The allowed labels are slightly different between the octahedral Tucker lemma and the Tucker lemma. We explain why this is without loss of generality. For an antipodal labelling of T^n with no complementary 1-simplices, it must be that the label of $(\emptyset, \emptyset)$ does not appear as a label of another vertex in T^n. Suppose it did, say the label of $(\emptyset, \emptyset)$ is a, and the label of (A, B) is also a for some $(A, B) \neq (\emptyset, \emptyset)$. So then the label of (B, A) is $-a$. But there is a 1-simplex joining $(\emptyset, \emptyset)$ and (B, A), forming a complementary 1-simplex. So then we can say without loss of generality that $(\emptyset, \emptyset)$ is labelled 1, and ± 1 does not appear as a label on the remaining vertices.

The octahedral Tucker lemma has an associated total NP search problem [52].

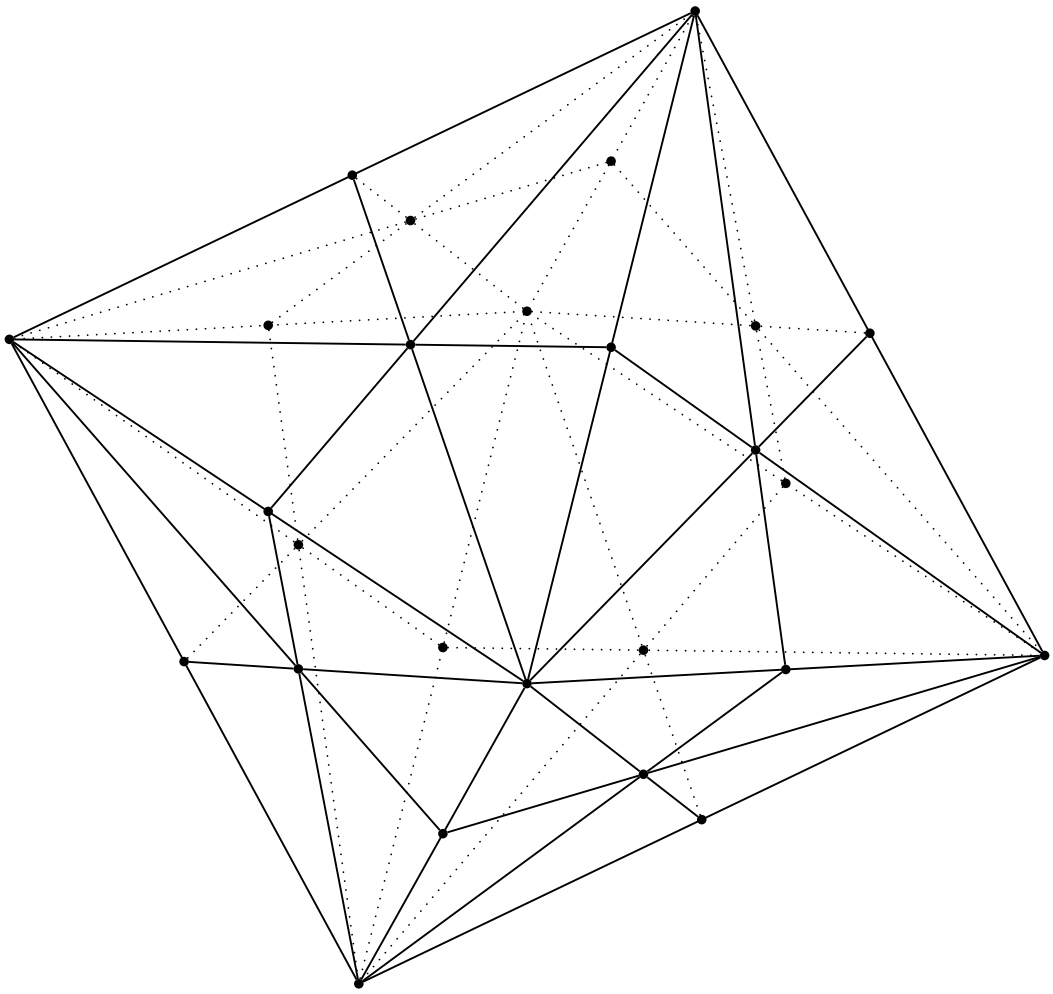


Figure 5.1: The octahedral ball for $n = 3$

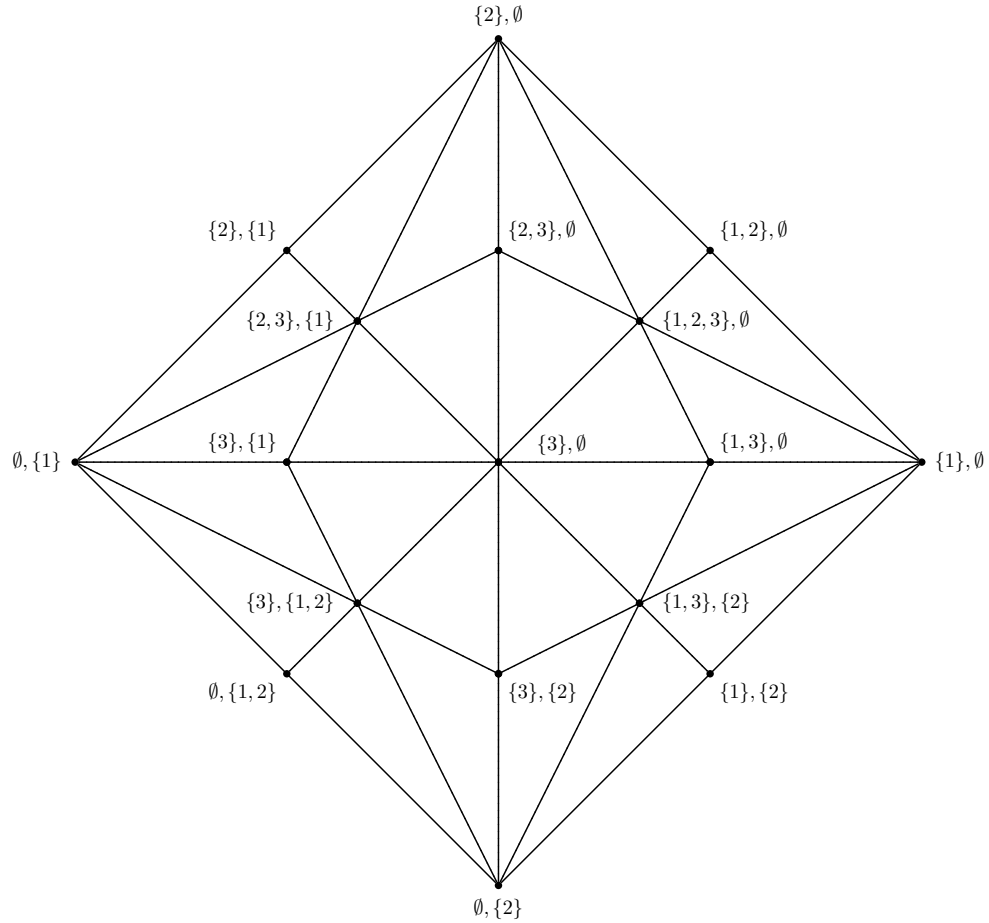


Figure 5.2: One hemisphere of the triangulation T^3 of the octahedral ball.

Definition 5.8. An instance of the *octahedral Tucker search problem* OCTAHEDRAL-TUCKER is an antipodal map $\lambda : \mathcal{B}^n \to \{1, \pm 2, \ldots, \pm n\}$. The solution to the search problem is one of the following:

1. a pair $(A_1, B_1), (A_2, B_2) \in \mathcal{B}^n$ with $(A_1, B_1) \subseteq (A_2, B_2)$, and $\lambda(A_1, B_1) = -\lambda(A_2, B_2)$,
2. an $(A, B) \in \mathcal{B}^n$ with $(A, B) \neq (\emptyset, \emptyset)$ and $\lambda(A, B) \neq -\lambda(B, A)$, or
3. an $(A, B) \in \mathcal{B}^n$ with $(A, B) \neq (\emptyset, \emptyset)$ and $\lambda(A, B) \notin \{\pm 2, \ldots, \pm n\}$.

Observe that $|\mathcal{B}^n| = 3^n$. We think of n as growing linearly, which gives an exponential size search space. The map λ is given by a function oracle (a type 2 search problem [7]). A solution to the search problem is guaranteed to exist by the Tucker lemma.

The size of an instance of octahedral Tucker lemma depends on the size of $\mathcal{B}^n$ (i.e., the 0-simplices of T^n). The known proofs of the octahedral Tucker lemma (via the Tucker lemma) [30, 31, 32, 50, 49, 53] reduce to the parity principle on higher dimensional simplices of T^n. To show that OCTAHEDRAL-TUCKER is in PPA, we prove an upper bound on $|T^n|$, the number of simplices in the triangulation T^n.

Proposition 5.9. $n!2^n \leq |T^n| \leq 2(n!)4^n$.

Proof. We prove that the number of n-simplices in T^n is exactly $n!2^n$. Pick a permutation of $\{1, \ldots, n\}$, call it $q_1, \ldots, q_n$. Also pick $\alpha_i \in \{1, -1\}$ for $i = 1, \ldots, n$. Define $(A_0, B_0) = (\emptyset, \emptyset)$, and

$$(A_i, B_i) = \begin{cases} (A_{i-1} \cup \{q_i\}, B_{i-1}) & \text{if } \alpha_i = 1 \\ (A_{i-1}, B_{i-1} \cup \{q_i\}) & \text{if } \alpha_i = -1. \end{cases}$$

The n-simplex $\{(A_0, B_0), \ldots, (A_n, B_n)\}$ corresponds uniquely to this permutation and choices of α_i's. Thus there are exactly $n!2^n$ many n-simplices, and $|T^n| \geq n!2^n$. For the upper bound, observe that each k-simplex is a face of at least one n-simplex. An n-simplex has 2^{n+1} many faces. Thus $|T^n| \leq (n!2^n)2^{n+1}$. □

Proposition 5.9 together with the argument that TUCKER is in PPA shows that OCTAHEDRAL-TUCKER is in PPA. Its exact complexity within PPA is unknown:

Question 5.10. *Is* OCTAHEDRAL-TUCKER PPAD-*hard? Is* OCTAHEDRAL-TUCKER PPA-*hard?*

5.3 The truncated Tucker lemma

The next total NP search problem we consider is based on the truncated Tucker lemma. With the octahedral Tucker lemma, there was not a substantial difference between the growth rate of the size of the search space (the vertices, $|\mathcal{B}^n| = 3^n$) and the size of the graph that gives an instance of the parity principle (the triangulation, $|T^n| \in O(n!4^n)$). With the truncated Tucker lemma, we will see a substantial difference in growth rates between vertices and the size of the triangulation. Recall from Section 3.5 the definition of the truncated octahedral ball, $\mathcal{B}^n_{\leq k}$:

$$\mathcal{B}^n_{\leq k} = \{(A, B) \in \mathcal{B}^n : \ |A| \leq k, \ |B| \leq k\}$$

For $A \subseteq [n]$, let $A_{\leq k}$ denote the k-least elements of A. For $A_1, A_2 \subseteq [n]$, write $A_1 \preceq A_2$ iff $(A_1 \cup A_2)_{\leq k} = A_2$. Write $(A_1, B_1) \preceq (A_2, B_2)$ iff $A_1 \preceq A_2$, $B_1 \preceq B_2$ and $A_i \cap B_j = \emptyset$ for $i, j \in \{1, 2\}$. Recall that the truncated Tucker lemma (Theorem 3.20) states:

Theorem 5.11 (Truncated Tucker lemma on $\mathcal{B}^n_{\leq k}$)**.** *Let $n \geq k \geq 1$. If $\lambda : \mathcal{B}^n_{\leq k} \to \{1, \pm 2 \ldots, \pm n\}$ is antipodal, then there are $(A_1, B_1) \preceq (A_2, B_2) \in \mathcal{B}^n_{\leq k}$ with $\lambda(A_1, B_1) = -\lambda(A_2, B_2)$.*

We describe $T^n_{\leq k}$, the triangulation of the truncated octahedral ball. Recall that we have previously associated the members of $\mathcal{B}^n$ with points in the n-ball. The vertices of $T^n_{\leq k}$ are the points in the n-ball associated with $\mathcal{B}^n_{\leq k} \subseteq \mathcal{B}^n$. The higher dimensional simplices of $T^n_{\leq k}$ correspond to chains in $\mathcal{B}^n_{\leq k}$ according to the partial order $\preceq$. The triangulation of the truncated octahedral ball for $k = 1$ and $n = 3$ is shown in Figures 5.3 and 5.4.

One way to understand the triangulation $T^n_{\leq k}$ is as follows. Start with the triangulation T^n. There are vertices in T^n that are not in $\mathcal{B}^n_{\leq k}$. Let (A, B) be such a vertex. Move the point corresponding to (A, B) from its current location to coincide with the point $(A_{\leq k}, B_{\leq k})$, adjusting higher-dimensional simplices that have (A, B) as a vertex appropriately. This process will collapse some simplices. For example, in T^n, there was a 1-simplex $\{(A, B), (A_{\leq k}, B_{\leq k})\}$. This procedure transforms this 1-simplex into the 0-simplex $\{(A_{\leq k}, B_{\leq k})\}$. It is clear that the result of this procedure is a simplicial complex if the input to the procedure was a simplicial complex, and both simplicial complexes triangulate the same space. Applying this procedure iteratively for every point in $\mathcal{B}^n \backslash \mathcal{B}^n_{\leq k}$ yields the triangulation $T^n_{\leq k}$. An example of this process is shown in Figure 5.5.

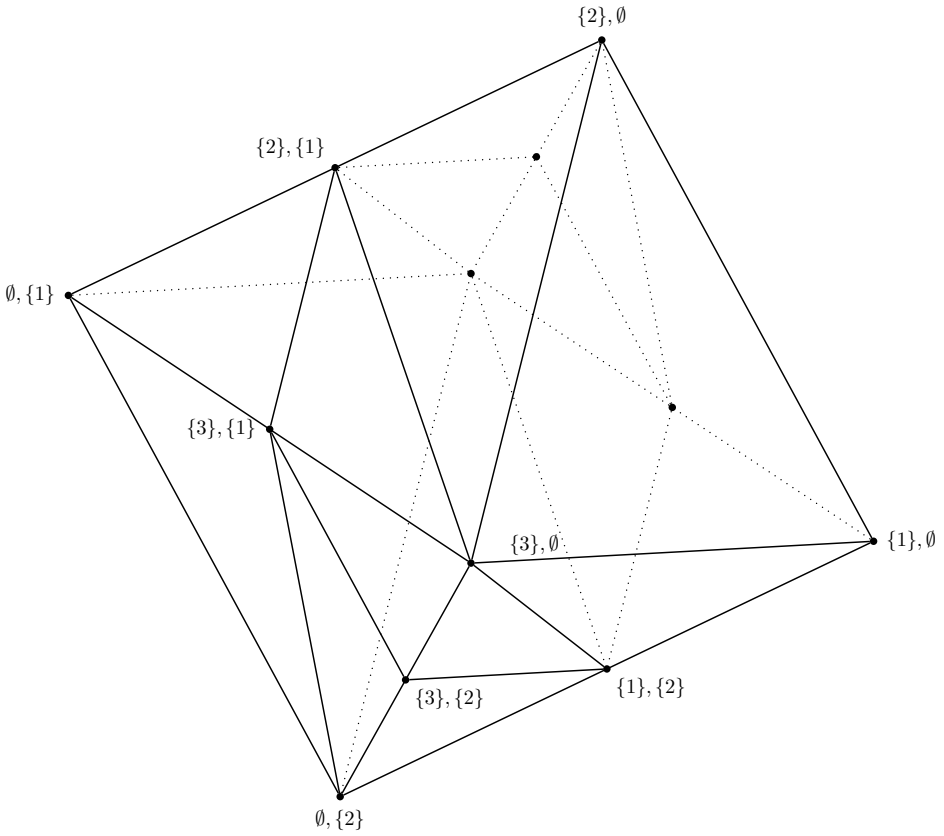


Figure 5.3: The triangulation $T^3_{\leq 1}$ of the truncated octahedral ball.

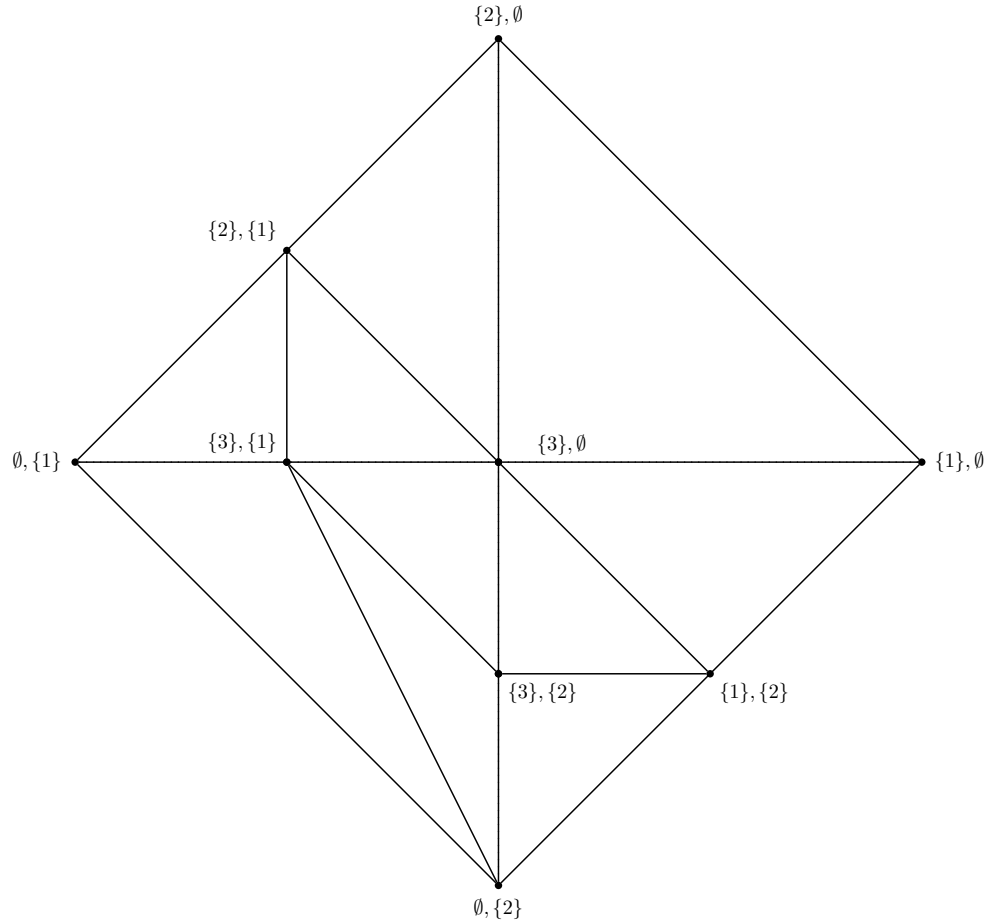


Figure 5.4: One hemisphere of the triangulation $T^3_{\leq 1}$ of the truncated octahedral ball.

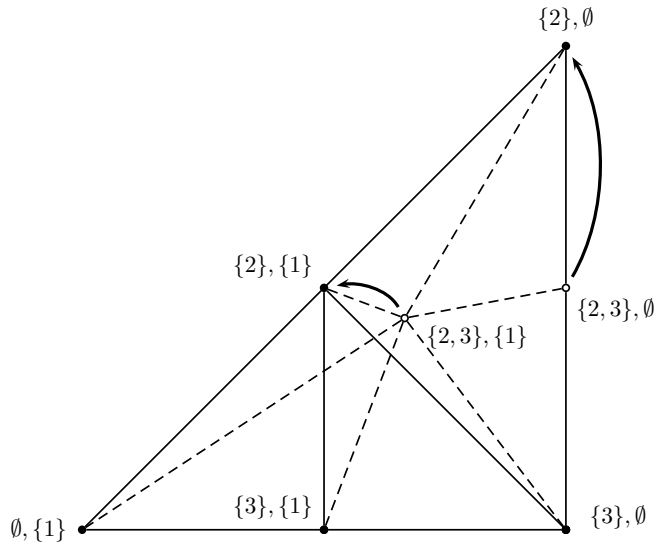


Figure 5.5: One face in the triangulation $T^3_{\leq 1}$ of the truncated octahedral ball. The point $(\{2,3\},\{1\})$ is moved to coincide with the point $(\{2\},\{1\})$ and the point $(\{2,3\},\emptyset)$ is moved to coincide with the point $(\{2\},\emptyset)$.

This way of obtaining $T^n_{\leq k}$ from T^n is reminiscent of the proof of the truncated Tucker lemma from the octahedral Tucker lemma. For that proof, we assumed we had violation of the truncated Tucker lemma on $\mathcal{B}^n_{\leq k}$, and we extended it to a violation of the octahedral Tucker lemma on $\mathcal{B}^n$ by assigning $(A,B) \in \mathcal{B}^n \setminus \mathcal{B}^n_{\leq k}$ the same label as $(A_{\leq k}, B_{\leq k})$.

Definition 5.12. Fix $k > 0$. An instance of the *k-truncated Tucker search problem* k-Truncated-Tucker (on $\mathcal{B}^n_{\leq k}$) is an antipodal map $\lambda : \mathcal{B}^n_{\leq k} \to \{1, \pm 2, \ldots, \pm n\}$. A solution to such an instance is one of the following:

1. a pair $(A_1,B_1),(A_2,B_2) \in \mathcal{B}^n_{\leq k}$ with $(A_1,B_1) \preceq (A_2,B_2)$ and $\lambda(A_1,B_1) = -\lambda(A_2,B_2)$,
2. an $(A,B) \in \mathcal{B}^n_{\leq k}$ such that $\lambda(A,B) \notin \{1, \pm 2, \ldots, \pm n\}$, or
3. an $(A,B) \in \mathcal{B}^n_{\leq k}$ with $(A,B) \neq (\emptyset,\emptyset)$ and $\lambda(A,B) \neq -\lambda(B,A)$.

Observe that $|\mathcal{B}^n_{\leq k}| \in O(n^{2k})$. For an instance of k-Truncated-Tucker (on $\mathcal{B}^n_{\leq k}$), we think of n as being exponentially big, and λ being given by a function oracle (a type 2 search problem [7]). A solution is guaranteed to exist by the Tucker lemma. Let's prove a bound on the size of the triangulation.

Proposition 5.13. $|T^n_{\leq k}| \in \Omega(2^n/\sqrt{n})$

Proof. Assume n is even. Consider subsets of $[n]$ of size $n/2$. Take such a subset S in decreasing sorted order: $a_1, \ldots, a_{n/2}$. Let $A_0 = \emptyset$, and $A_i = \{a_1, \ldots, a_i\}_{\leq k}$. Because the a_i's are in decreasing sorted order, the A_i's are all unique, and $(A_0, \emptyset) \preceq \cdots \preceq (A_{n/2}, \emptyset)$. The $n/2$-simplex $\{(A_0, \emptyset), \ldots, (A_{n/2}, \emptyset)\}$ corresponds uniquely to this subset S of $[n]$ of size $n/2$. There are $\Omega(2^n/\sqrt{n})$ such subsets. □

Although this bound is slightly worse than the one for the triangulation T^n, it is more interesting because k-Truncated-Tucker takes n to be exponentially big. Thus we have the strange property that k-Truncated-Tucker is proved total by considering the undirected parity principle on a double-exponentially big graph!

We define another total NP search problem based on the truncated Tucker lemma. Recall that we had previously defined two closely related truncated Tucker lemmas. Recall that

$$\mathcal{B}^n_k = \{(A, B) \in \mathcal{B}^n : \ |A| = k \text{ or } 0, \ |B| = k \text{ or } 0, A \cup B \neq \emptyset\}.$$

Definition 5.14. Fix $k > 0$. An instance of the *k-truncated Tucker search problem* k-Truncated-Tucker (on $\mathcal{B}^n_k$) is an antipodal map $\lambda : \mathcal{B}^n_k \to \{\pm 2k, \ldots, \pm n\}$. A solution to such an instance is one of the following:

1. a pair $(A_1, B_1), (A_2, B_2) \in \mathcal{B}^n_k$ with $(A_1, B_1) \preceq (A_2, B_2)$ and $\lambda(A_1, B_1) = -\lambda(A_2, B_2)$,
2. an $(A, B) \in \mathcal{B}^n_k$ such that $\lambda(A, B) \notin \{\pm 2k, \ldots, \pm n\}$, or
3. an $(A, B) \in \mathcal{B}^n_k$ such that $\lambda(A, B) \neq -\lambda(B, A)$.

As with $\mathcal{B}^n_{\leq k}$, we have that $|\mathcal{B}^n_k| \in O(n^{2k})$. Similarly, for an instance of k-Truncated-Tucker (on $\mathcal{B}^n_k$), we think of n as being exponentially big, and λ being given by a function oracle (a type 2 search problem). A solution is guaranteed to exist by the Tucker lemma. Previously we have seen that the two truncated Tucker lemmas were equivalent to each other. The same carries over for total NP search problems.

Theorem 5.15. *Fix $k > 0$. The search problems k-Truncated-Tucker on $\mathcal{B}^n_{\leq k}$ and k-Truncated-Tucker on $\mathcal{B}^n_k$ are many-one reducible to one another.*

Proof. One direction follows from the proof of Theorem 3.22 from Theorem 3.20. The other direction follows by the proof of Theorem 3.24. □

Theorem 5.16. *The search problem* 1-Truncated-Tucker *is* PPP*-hard under many-one reductions.*

Note: this also follows by Theorems 5.23 and 5.25.

Proof. We reduce Pigeon to 1-Truncated-Tucker. Let f be an instance of Pigeon. Define $\lambda : \mathcal{B}_1^n \to \{\pm 2, \ldots, \pm n\}$ as follows:

$$\lambda(A,B) = \begin{cases} f(i)+1 & A \succeq B, \quad A = \{i\} \\ -(f(j)+1) & B \succeq A, \quad B = \{j\} \end{cases}$$

The map λ is an instance of 1-Truncated-Tucker. Consider any solution to λ. By construction, λ is antipodal, so a solution is either a pair $(A_1, B_1) \preceq (A_2, B_2)$ with $\lambda(A_1, B_1) = -\lambda(A_2, B_2)$, or a $\lambda(A, B) \notin \{\pm 2, \ldots, \pm n\}$. For the latter case, by construction of λ, this clearly gives an $x \in [n]$ such that $f(x) \notin [n-1]$. For the former case, without loss of generality, $A_1 \succeq B_1$. Thus $B_2 \succeq A_2$. Say that $A_1 = \{i\}$ and $B_2 = \{j\}$ By definition of $\preceq$, $i \neq j$. Furthermore it must be that $f(i) = f(j)$, therefore i and j form a solution to f as an instance of Pigeon. □

The truncated Tucker search problems form a hierarchy:

Theorem 5.17. *Fix $k > 0$. There is a many-one reduction from k-*Truncated-Tucker *to $(k+1)$-*Truncated-Tucker*.*

Proof. Let $\lambda : \mathcal{B}_{\leq k}^n \to \{1, \pm 2, \ldots, \pm n\}$ be an instance of k-Truncated-Tucker. We will use it to define $\lambda' : \mathcal{B}_{\leq (k+1)}^n \to \{1, \pm 2, \cdots \pm n\}$, an instance of $(k+1)$-Truncated-Tucker. For $(A, B) \in \mathcal{B}_{\leq (k+1)}^n$, define $\lambda'(A, B) = \lambda(A_{\leq k}, B_{\leq k})$ A solution to λ' directly gives a solution to λ. The only interesting case is when $(A_1, B_1) \preceq (A_2, B_2) \in \mathcal{B}_{\leq (k+1)}^n$ and $\lambda'(A_1, B_1) = -\lambda'(A_2, B_2)$. Therefore $\lambda(A_{1,\leq k}, B_{1,\leq k}) = -\lambda(A_{2,\leq k}, B_{2,\leq k})$. This is a solution for λ because $(A_{1,\leq k}, B_{1,\leq k}) \preceq (A_{2,\leq k}, B_{2,\leq k})$, which follows because

$$(X \cup Y)_{\leq (k+1)} = Y \implies (X_{\leq k} \cup Y_{\leq k})_{\leq k} = Y_{\leq k}.$$

□

Question 5.18. *Is the (relativized) k-*Truncated-Tucker *hierarchy proper?*

Question 5.19. *Is k-*Truncated-Tucker *in* PPP *for any value of k?*

We remark that we have defined these truncated Tucker search problems so that k is constant and n grows exponentially in order to obtain an exponential size search space ($\mathcal{B}^n_{\leq k}$ or $\mathcal{B}^n_k$). We could modify this to let k be a function of n, and modify the growth rate of n to ensure that the growth rate of the size of the search space is exponential. The only additional requirement is that $k < n$ to match the statement of the truncated Tucker lemma on $\mathcal{B}^n_{\leq k}$, and $k < n/2$ to match the statement of the truncated Tucker lemma on $\mathcal{B}^n_k$. For example, if $k(n) = n/3$ and n grows linearly, then $\mathcal{B}^n_k$ has an exponential growth rate, giving a total NP search problem. In fact, this problem is reducible to Octahedral-Tucker, and so it is in PPA. It is interesting to note that the constant k truncated Tucker search problems are PPP-hard and that the linear k truncated Tucker search problems are in PPA. This raises the question of what can be said about growth rates of k between constant and linear.

We conclude this section by defining a large semantic class in TFNP based on the Tucker lemma that includes PPA and PPP. Let $(X, \preceq)$ be a partial order, Boundary(x) be a relation on X, and Antipode(x) be a map from X to X. Say that the tuple $((X, \preceq), \text{Boundary}(x), \text{Antipode}(x))$ is *antipodally symmetric on the n-ball* if there exists an antipodally symmetric triangulation T of B^n so that (1) the points in X are in one-to-one correspondence with the vertices of T, (2) if $\{v_1, v_2\}$ is a 1-simplex in T then the members of X corresponding to v_1 and v_2 are comparable by $\preceq$, and (3) for all $x \in X$, Boundary(x) holds if and only if the vertex corresponding to x is on the boundary and it is antipodal to the vertex corresponding to Antipode(x). The following is an immediate corollary to the Tucker lemma:

Corollary 5.20. *Let $((X, \preceq), \text{Boundary}(x), \text{Antipode}(x))$ be antipodally symmetric on the n-ball. If $\lambda : X \to \{\pm 1, \ldots, \pm n\}$ has the property that for all $x \in X$ where* Boundary(x) *holds, $\lambda(x) = -\lambda(\text{Antipode}(x))$, then there exists $x, y \in X$ with $x \preceq y$ and $\lambda(x) = -\lambda(y)$.*

If the $\preceq$ relation, Boundary(x) and Antipode(x) are computable in polynomial time, then the corollary above can be naturally translated into total NP search problems. These problems fix $\preceq$, Boundary(x) and Antipode(x) and take λ to be the input. There is a different search problem for each appropriate choice of $\preceq$, Boundary(x) and Antipode(x). Define the class Poset-Tucker by taking the set of all such search problems, then taking its closure under many-one reductions. Observe that for all k, k-Truncated-Tucker is in Poset-Tucker, thus PPP $\subseteq$ Poset-Tucker. Furthermore 2-

D Tucker is in Poset-Tucker, hence PPA $\subseteq$ Poset-Tucker. So Poset-Tucker is a very powerful class. Unfortunately it seems to be a semantic class.

Question 5.21. *Does* Poset-Tucker *have complete problems in it? Does it have other interesting syntactic subclasses?*

A natural approach to finding a complete problem for Poset-Tucker is taking the search problem based on Corollary 5.20 and making $\preceq$, Boundary(x) and Antipode(x) be part of the input (as descriptions of Turing machines, say). The difficulty with this approach is that in order to ensure that the resulting search problem is total, it must be able to handle invalid inputs.

5.4 The Kneser-Lovász theorem

In this section, we define the total NP search problems associated with the Kneser-Lovász theorem, and relate them to the truncated Tucker search problems and to PPP.

Definition 5.22. Fix $k > 0$. An instance of the *k-Kneser-Lovász search problem* k-Kneser-Lovász is a function $c : \binom{n}{k} \to \{2k, \ldots, n\}$. The solution to the search problem is one of the following:

1. two elements $A, B \in \binom{n}{k}$ with $A \cap B = \emptyset$ and $c(A) = c(B)$, or
2. an $A \in \binom{n}{k}$ such that $c(A) \notin \{2k, \ldots, n\}$.

We think of n as being exponentially big, and c being computed by a function oracle (a type 2 problem). A solution is guaranteed to exist by the pigeonhole principle. Observe that 1-Kneser-Lovász is equivalent to Pigeon, and so we immediately have:

Theorem 5.23. 1-Kneser-Lovász *is* PPP*-complete.*

Question 5.24. *Is* k-Kneser-Lovász *in* PPP *for any* k *other than* $k = 1$*?*

Theorem 5.25. *There is a many-one reduction from* k-Kneser-Lovász *to* k-Truncated-Tucker*.*

Proof. This follows from the proof of the Kneser-Lovász theorem from the truncated Tucker lemma. □

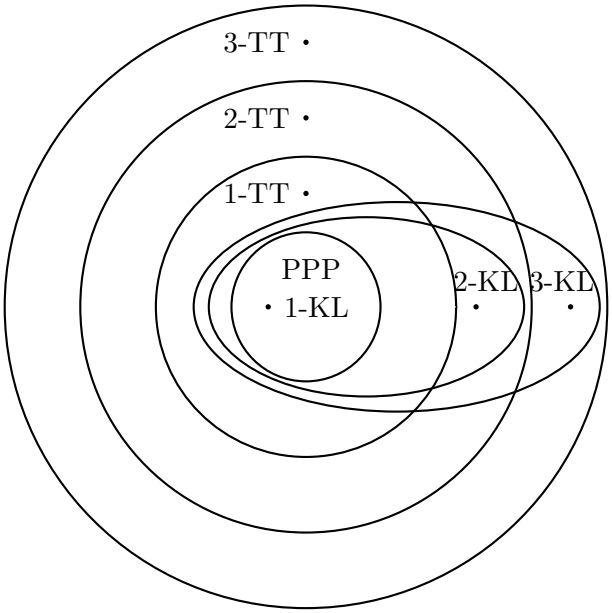


Figure 5.6: PPP, Kneser-Lovász hierarchy, and truncated Tucker hierarchy for $k = 1, 2$, and 3, assuming all possible separations.

Theorem 5.26 (Istrate-Crăciun, private communication)**.** *For fixed $k > 0$, k-*Kneser-Lovász *is many-one reducible to $(k+1)$-*Kneser-Lovász*.*

Proof. Let $c : \binom{n}{k} \to \{2k, \ldots n\}$ be an instance of k-Kneser-Lovász. Define $c' : \binom{n+2}{k+1} \to \{2(k+1), \ldots, n+2\}$ by $c'(A) = c(A_{\leq k}) + 2$. It is clear that a solution to c' as an instance of $(k+1)$-Kneser-Lovász immediately gives a solution to c as an instance of k-Kneser-Lovász. □

Question 5.27. *Is the (relativized) k-*Kneser-Lovász *hierarchy proper?*

The results described above are shown in Figure 5.6, assuming each level of the Kneser-Lovász search problem and truncated Tucker search problem is distinct.

Bibliography

[1] James Aisenberg, Maria Luisa Bonet, and Sam Buss. Quasi-polynomial size Frege proofs of Frankl's theorem on the trace of finite sets. To appear in *Journal of Symbolic Logic*.

[2] James Aisenberg, Maria Luisa Bonet, and Sam Buss. Tucker's Lemma is PPA complete. Preliminary manuscript, http://eccc.hpi-web.de/report/2015/163/, 2015.

[3] James Aisenberg, Maria Luisa Bonet, Sam Buss, Adrian Crăciun, and Gabriel Istrate. Short proofs of the Kneser-Lovász coloring principle. Submitted for publication, journal version of [4], 2015.

[4] James Aisenberg, Maria Luisa Bonet, Sam Buss, Adrian Crăciun, and Gabriel Istrate. Short proofs of the Kneser-Lovász coloring principle. In *Proc. 42th International Colloquium on Automata, Languages, and Programming (ICALP'15)*, Lecture Notes in Computer Science 9135, pages 44–55, 2015.

[5] M. Ajtai. The complexity of the pigeonhole principle. In *Proceedings of the 29-th Annual IEEE Symposium on Foundations of Computer Science*, pages 346–355, 1988.

[6] Jeremy Avigad. Plausibly hard combinatorial tautologies. In Paul Beame and Samuel R. Buss, editors, *Proof Complexity and Feasible Arithmetics*, pages 1–12. American Mathematical Society, 1997.

[7] Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *Journal of Computer and System Sciences*, 57(1):3–19, 1998.

[8] Arnold Beckmann and Samuel R. Buss. Improved witnessing and local improvement principles for second-order bounded arithmetic. *ACM Transactions on Computational Logic*, 15(1), 2014. Article 2, 35 pages.

[9] Maria Luisa Bonet, Samuel R. Buss, and Toniann Pitassi. Are there hard examples for Frege systems? In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 30–56, Boston, 1995. Birkhäuser.

[10] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62:708–728, 1997. An

earlier version appeared in *Proc. Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, 1995, pp. 575-584.

[11] Karol Borsuk. Drei Sätze über die n-dimensionale Euklidische Sphäre. *Fundamenta Mathematicae*, 1(20):177–190, 1933.

[12] Josh Buresh-Oppenheim. On the TFNP complexity of factoring. Unpublished manuscript, http://www.cs.toronto.edu/∼bureshop/factor.pdf, 2006.

[13] Josh Buresh-Oppenheim and Tsuyoshi Morioka. Relativized NP search problems and propositional proof systems. In *Proc. 19th IEEE Conference on Computational Complexity (CCC)*, pages 54–67, 2004.

[14] Sam Buss. Quasipolynomial size proofs of the propositional pigeonhole principle. To appear in *Theoretical Computer Science*, 2015.

[15] Samuel R. Buss. *Bounded Arithmetic*. Bibliopolis, 1986. Revision of 1985 Princeton University Ph.D. thesis.

[16] Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

[17] Samuel R. Buss. Propositional consistency proofs. *Annals of Pure and Applied Logic*, 52:3–29, 1991.

[18] Samuel R. Buss. Propositional proof complexity: An introduction. In U. Berger and H. Schwichtenberg, editors, *Computational Logic*, pages 127–178. Springer-Verlag, Berlin, 1999.

[19] Samuel R. Buss. Towards NP-P via proof complexity and proof search. *Annals of Pure and Applied Logic*, 163(9):1163–1182, 2012.

[20] Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272, 2006.

[21] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing the two-player Nash equilibrium. *Journal of the ACM*, 56(3):Article 14, 2009.

[22] Stephen A. Cook. Feasibly constructive proofs and the propositional calculus. In *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, pages 83–97, 1975.

[23] Stephen A. Cook and Phuong Nguyen. *Foundations of Proof Complexity: Bounded Arithmetic and Propositional Translations*. ASL and Cambridge University Press, 2010. 496 pages.

[24] Stephen A. Cook and Robert A. Reckhow. On the lengths of proofs in the propositional calculus, preliminary version. In *Proceedings of the Sixth Annual ACM Symposium on the Theory of Computing*, pages 135–148, 1974.

[25] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979.

[26] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing (STOC'06)*, pages 71–78, 2006.

[27] Xiaotie Deng, Jack Edmonds, Zhe Feng, Zhengyang Liu, Qi Qi, and Zeying Xu. Understanding PPA-completeness. Technical Report ECCC-TR15-120, Electronic Colloquium on Computational Complexity, August 2015.

[28] Xiaotie Deng, Qi Qi, and Jie Zhang. Direction preserving zero point computing and applications (extended abstract). In *Internet and Network Economics, 5th International Workshop (WINE)*, Lecture Notes in Computer Science 5929. Springer, 2009.

[29] Peter Frankl. On the trace of finite sets. *Journal of Combinatorial Theory, Series A*, 34:41–45, 1983.

[30] Robert M. Freund. Variable dimension complexes, part I: Basic theory. *Mathematics of Operations Research*, 9(4):479–497, 1984.

[31] Robert M. Freund. Variable dimension complexes, part II: A unified approach to some combinatorial lemmas in topology. *Mathematics of Operations Research*, 9(4):498–509, 1984.

[32] Robert M. Freund and Michael J. Todd. A constructive proof of Tucker's combinatorial lemma. *Journal of Combinotorial Theory, Series A*, 30:321–325, 1981.

[33] Katalin Friedl, Gábor Ivanyos, Miklos Santha, and Yves F. Verhoeven. Locally 2-dimensional Sperner problems complete for the Polynomial Parity Argument classes. In *Algorithms and Complexity, 6th Italian Conference (CIAC)*, Lecture Notes in Computer Science 3998, pages 380–391. Springer, 2006.

[34] Ira Gessel and Gian-Carlo Rota, editors. *Classic Papers in Combinatorics*. Birkhäuser, 1987.

[35] Michelangelo Grigni. A Sperner lemma complete for PPA. *Information Processing Letters*, 77(5-6):255–259, 2001.

[36] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[37] G.H. Hardy and E.M. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 6 edition, 2008.

[38] Pavel Hrubeš and Iddo Tzameret. The proof complexity of polynomial identities. In *Proc. 24th IEEE Conf. on Computational Complexity (CCC)*, pages 41–51, 2009.

[39] Pavel Hrubeš and Iddo Tzameret. Short proofs for determinant identities. *SIAM J. Computing*, 44(2):340–383, 2015.

[40] Gabriel Istrate and Adrian Crăciun. Proof complexity and the Kneser-Lovász theorem. In *Theory and Applications of Satisfiability Testing (SAT)*, Lecture Notes in Computer Science 8561, pages 138–153. Springer Verlag, 2014.

[41] Emil Jeřábek. Dual weak pigeonhole principle, Boolean complexity, and derandomization. *Annals of Pure and Applied Logic*, 124:1–37, 2004.

[42] Emil Jeřábek. Integer factoring and modular square roots. To appear in *Journal of Computer and System Sciences*, 201?

[43] Gyula O.H. Katona. A theorem of finite sets. In *Theory of Graphs: Proc. Coll. Tihany, Hungary, Sept. 1966*, pages 187–207. Akadémiai Kiadó and Academic Press, 1966. Reprinted in [34], pp. 361-380.

[44] Leszek Aleksander Kołodziejczyk, Phuong Nguyen, and Neil Thapen. The provably total NP search problems of weak second-order bounded arithmetic. *Annals of Pure and Applied Logic*, 162(2):419–446, 2011.

[45] Jan Krajíček. *Bounded Arithmetic, Propositional Calculus and Complexity Theory.* Cambridge University Press, Heidelberg, 1995.

[46] Jan Krajíček and Pavel Pudlák. The number of proof lines and the size of proofs in first-order logic. *Archive for Mathematical Logic*, 27:69–84, 1988.

[47] Joseph B. Kruskal. The number of simplices in a complex. In R. Bellman, editor, *Mathematical Optimization Techniques*, pages 251–278. University of California Press, 1963.

[48] László Lovász. Kneser's conjecture, chromatic number, and homotopy. *Journal of Combinatorial Theory, Series A*, 25(3):319 – 324, 1978.

[49] Jiří Matoušek. A combinatorial proof of Kneser's conjecture. *Combinatorica*, 24(1):163–170, 2004.

[50] Jiří Matoušek. *Using the Borsuk-Ulam Theorem: Lectures on Topological Methods in Combinatorics and Geometry.* Springer, second edition, 2008.

[51] Akihiro Nozaki, Toshiyasi Arai, and Noriko H. Arai. Polynomial-size Frege proofs of Bollobás' theorem on the trace of sets. *Proceedings of the Japan Academy, Series A. Math. Sci.*, 84(8):159–161, 2008.

[52] Dömötör Pálvölgyi. 2D-Tucker is PPAD-complete. In *Internet and Network Economics, 5th International Workshop (WINE)*, Lecture Notes in Computer Science 5929, pages 569–574. Springer, 2009.

[53] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[54] Pavel Pudlák. On the lengths of proofs of finitistic consistency statements in first order theories. In *Logic Colloquium '84*, pages 165–196. North-Holland, 1986.

[55] Pavel Pudlák. Improved bounds to the lengths of proofs of finitistic consistency statements. In *Logic and Combinatorics*, volume 65 of *Contemporary Mathematics*, pages 309–331. American Mathematical Society, 1987.

[56] Pavel Pudlák. Lower bounds for resolution and cutting planes proofs and monotone computations. *Journal of Symbolic Logic*, 62:981–998, 1997.

[57] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7:291–324, 1998.

[58] Robert A. Reckhow. *On the Lengths of Proofs in the Propositional Calculus*. PhD thesis, Department of Computer Science, University of Toronto, 1976. Technical Report #87.

[59] Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007.

[60] Richard Statman. Complexity of derivations from quantifier-free Horn formulae, mechanical introduction of explicit definitions, and refinement of completeness theorems. In R. Gandy and M. Hyland, editors, *Logic Colloquium '76*, pages 505–517, Amsterdam, 1977. North-Holland.

[61] Andrew G. Thomason. Hamiltonian cycles and uniquely edge colorable graphs. *Annals of Discrete Mathematics*, 3:259–268, 1978.

[62] Albert W. Tucker. Some topological properties of disk and sphere. In *Proceedings of the First Canadian Mathematical Congress*, pages 285–309. University of Toronto Press, 1946.

[63] Günter M. Ziegler. Generalized Kneser coloring theorems with combinatorial proofs. *Inventiones Mathematicae*, 147(3):671–691, 2002.