**Title**
Data Exchange, Data Integration, and Chase



**Authors**
Nash, Alan
Deutsch, Alin
Remmel, Jeff

**Publication Date**
2006-04-26

Peer reviewed

# Data Exchange, Data Integration, and Chase

Alan Nash
anash@cs.ucsd.edu

Alin Deutsch
deutsch@cs.ucsd.edu

Jeff Remmel
remmel@math.ucsd.edu

University of California, San Diego

## ABSTRACT

We study the problem of computing certain answers to a query over a target schema for a source instance under constraints which relate the source and target schemas. Prior work has shown that, for restricted constraints (source-to-target and target-to-target embedded dependencies) and unions of conjunctive queries, there is a special instance, known as a *universal solution*, such that running the query on it essentially yields the certain answers. Such a universal solution does not always exist for even slight extensions of these classes of constraints and queries.

We show that there may be a finite set of instances, which we call a *universal set solution*, which suffices to compute the certain answers. We also introduce the notion of a *$k$-universal set* solution, which is sufficient to compute the certain answers to queries with at most $k$ variables, even when no universal set solution exists. We show how to compute such universal and $k$-universal set solutions for universal-existential constraints and existential queries.

The main algorithm for computing the universal set solution is an extended chase. We provide a completeness result for this chase and sufficient conditions for termination, which strictly extend the best previously known conditions (such as weak acyclicity). We also introduce a new kind of chase to compute $k$-universal set solutions.

## 1. INTRODUCTION

Recent years have witnessed the development of two related research fields: *data integration* and *data exchange*. In both cases, the setup consists of a source schema $\sigma$ and a target schema $\tau$ which are related by a set of constraints $\Sigma$.

In data exchange we are interested in transfering a source instance $S$ of $\sigma$ to a target instance $T$ of $\tau$ such that $S$ and $T$ satisfy the constraints in $\Sigma$. Such $T$ is called a *solution* to the data exchange problem in [8]. This task is motivated by such activities as data migration, data sharing, and even schema evolution. *The data exchange problem* consists of finding such a solution $T$, given $S$ and $\Sigma$. Depending on the constraints in $\Sigma$, there are usually infinitely many solutions. [9] suggests that the preferred solution should be the *core of a universal solution*, which is unique up to isomorphism and is the smallest among the most general solutions.

In data integration, the target schema is the schema of a mediator, against which client applications can pose their query $Q$. The target instance is virtual, i.e. not materialized, and the task of the mediator is to compute the answer to $Q$ using the source instance $S$. The answer to $Q$ is defined as the intersection of the results of $Q$ on all solutions: $\bigcap_{\text{solution } T} Q(T)$. This intersection is called the *certain answers* to $Q$. We refer to the task of computing the certain answers as *the data integration problem*. As in data exchange, there may be infinitely many solutions $T$ corresponding to $S$.

[8] reveals a beautiful connection between the data exchange and the data integration problems: if the client query $Q$ is a union of conjunctive queries and the constraints in $\Sigma$ are source-to-target and target-to-target embedded dependencies, then any universal solution $U$ of the data exchange problem provides a solution to the data integration problem: the certain answers to $Q$ are the tuples in $Q(U)$ over the active domain of the source. Moreover, [8] shows that a universal solution can be computed using the chase procedure. The importance of these results is that they eliminate the challenge posed by the non-algorithmic definition of certain answers, which is based on infinitely many possible solutions. All one needs to do to find the certain answers is to compute a universal solution using the chase, and run the client query over it.

Unfortunately, the connection between the two problems breaks down as soon as more expressive queries or constraints are considered. As already observed in [8], if $Q$ contains even one inequality, then its result on a universal solution may *strictly contain* the certain answers. Furthermore, it is shown in [10] that, when the constraints in $\Sigma$ are not source-to-target, there simply may not be any universal solution, yet the set of certain answers may be non-empty. Looking at constraints beyond the source-to-target class is motivated in [10, 12] in the setting of peer data management. This case is also relevant to incorporating materialized warehouses and cached queries into the mediator for data integration [5].

**Contributions.** This paper makes the following contributions.

**1. Universal set solutions.** We restore and strengthen the connection between data exchange and data integration by introducing the notion of a *universal set solution*. This is a finite set $U$ of solutions, sufficient to compute the certain answers to a query $Q$ of arity $r$ as follows:

$$\text{certain answers of } Q = \text{dom}(S)^r \cap \bigcap_{T \in U} Q(T).$$

**2. Wider classes of queries and constraints.** We show that this type of connection holds for wider classes of queries than just unions of conjunctive queries and for wider classes of constraints than just source-to-target and target-to-target embedded dependencies. In particular, we show how to compute certain answers for

- arbitrary monotonic queries,
- unions of conjunctive queries with inequality and negation ($\text{UCQ}^{\neg,\neq}$or $\exists$-queries), and
- embedded dependencies extended with disjunction, inequality, and negation ($\forall\exists$-constraints), arbitrarily expressed over the combined schemas $\sigma$ and $\tau$ (i.e. not restricted to be source-to-target and target-to-target) which satisfy the terminating conditions in item 8 below.

**3. Data exchange.** Our work provides solutions to the data exchange problem beyond the setting of source-to-target embedded

dependencies studied in the literature [8, 10]. In our more general setting, the single universal solution may not exist, or may not be useful for correctly computing certain answers to queries beyond conjunctive queries. Previous research does not provide any guidelines on what solution to materialize in that case. A natural choice is to materialize one solution from the universal set solution, if it exists. We show how to compute such solutions using the chase.

**4. Relaxing universality.** We address the case when there is no universal set solution, showing that certain answers can sometimes be computed for unions of conjunctive queries with at most $k$ variables by finding a "not quite universal" set solution which we call $k$-universal. We introduce an interesting variant of the chase to compute $k$-universal set solutions.

**5. Query containment, etc.** Our results, concepts, and techniques are widely applicable beyond the data exchange and integration settings. In particular, they provide a unified solution for deciding query containment under constraints for expressive classes of queries and constraints which allow disjunction, inequalities and negated literals.

The machinery we develop to carry out the research program detailed above involves the following technical contributions.

**6. Templates.** We introduce the concept of a *template* of a class of instances, which generalizes the notion of a universal set solution and is relevant to the data exchange, data integration and containment problems. Given constraints $\Sigma$, we distinguish between a template for the set of all finite models of $\Sigma$ which we call *weak* templates and a template for all (both finite and infinite) models of $\Sigma$ which we call a *strong* template. We show that some embedded dependencies $\Sigma$ have a weak template, but no strong template. It turns out that a universal set solution for $S$ under $\Sigma$ is precisely a weak template for constraints $\Sigma_S$ satisfying $T \models \Sigma_S$ iff $(S, T) \models \Sigma$.

**7. New chase.** We show that strong templates are precisely what any "chase-like" procedure computes if it terminates. We start by formalizing the well-known ordered chase procedure to provide a unifying treatment of all chase extensions from recent prior work to cover arbitrary $\forall\exists$ constraints. We show that this ordered chase is incomplete, that is, it may fail to find a strong template even when one exists.

We then introduce a novel chase procedure, which we call the *unordered minimizing chase*, that is order-independent. We show that the unordered minimizing chase is complete for finding strong templates. That is, the unordered minimizing chase terminates if and only if there is a strong template. From the point of view of completeness, this is the best any chase-like procedure can achieve. In particular, the unordered minimizing chase terminates whenever any chase-like procedure terminates and if there is a universal set solution that any chase-like procedure can find, the unordered minimizing chase will.

**8. New termination conditions.** We provide a widely-applicable, sufficient condition for termination of the ordered chase, which can be checked effectively on the constraints $\Sigma$. If this condition holds, we say that $\Sigma$ is *stratified*. This new stratification condition is strictly more general than the best previously known such condition: weak acyclicity [8, 6].

**Related Work.** We have already discussed above the pioneering work in [8]. The chase was introduced in [14] where its connection to logical implication was established (an early ancestor was introduced in [2]). Related formulations of the chase for various kinds of dependencies were introduced in [13, **?**]. The chase was extended to embedded dependencies in [3], to include disjunction and inequality in [6], and to arbitrary $\forall\exists$-sentences in [4].

**Paper Outline.** In Section 2 we introduce some notation and basic concepts. In Section 3 we show how to compute certain answers using universal set solutions. Universal set solutions are special kinds of templates; we introduce the latter in Section 4. In Section 5 we show how to compute templates using the chase. In Section 6 we extend our results to constraints beyond embedded dependencies and in Section 7 to mappings other than homomorphisms and queries beyond UCQ. In Section 8 we show how templates apply to checking containment under constraints and implication. Then in Section 9 we show how to compute certain answers to queries with $k$-variables, even when universal set solutions do not exist. and finally we conclude in Section 10.

## 2. PRELIMINARIES

**Basics.** A schema $\sigma$ is a list of constants and relation symbols and their arities. An instance $A$ *over* $\sigma$ has one relation for every relation symbol in $\sigma$, of the same arity. Unless we say otherwise, we refer to finite instances. For an instance $A$, we write $\mathrm{dom}(A)$ for the domain of $A$, $|A|$ for the size of $\mathrm{dom}(A)$, and $R^A$ for the value of the relation $R$ in $A$. We need to consider instances which have two types of values: constants and variables. The latter are also known as *labelled nulls* [8]. If $A, B$ are both over $\sigma$, we write $A \subseteq B$ if for every relation symbol $R \in \sigma$, $R^A \subseteq R^B$ and $A \sqsubseteq B$ if for every relation symbol $R \in \sigma$, $R^A = R^B|\mathrm{dom}(A)$.

**Queries.** We consider the class CQ of conjunctive queries and the class UCQ of unions of conjunctive queries and their extensions to include inequality ($\mathrm{CQ}^{\neq}$,$\mathrm{UCQ}^{\neq}$), negation ($\mathrm{CQ}^{\neg}$,$\mathrm{UCQ}^{\neg}$), or both ($\mathrm{CQ}^{\neg,\neq}$,$\mathrm{UCQ}^{\neg,\neq}$). Notice that $\mathrm{UCQ}^{\neg,\neq}$ is the same as the class of existential queries $\exists$Q. A query $Q$ is *monotonic* if $A \subseteq B$ implies $Q(A) \subseteq Q(B)$. We write MonQ for the class of monotonic queries.

**Constraints.** We consider constraints $\xi$ of the form

$$\phi(\bar{u}, \bar{w}) \rightarrow \exists \bar{v}\, \psi(\bar{u}, \bar{v})$$

where $\phi$ and $\psi$ are conjunctions of atoms, which may include equations. Such constraints are known as *embedded dependencies*. We call $\phi$ the *premise* and $\psi$ the *conclusion*. For a given constraint $\xi$, we write $P_\xi$ for the former and $C_\xi$ for the latter and we write $\exists P_\xi$ for $\exists \bar{w} P_\xi$ and $\exists C_\xi$ for $\exists \bar{v} C_\xi$. If $\bar{v}$ is empty, then $\xi$ is a *full dependency*. If $\psi$ consists only of equations, then $\xi$ is an *equality-generating dependency (EGD)*. If $\psi$ consists only of relational atoms, then $\xi$ is a *tuple-generating dependency (TGD)*. Every set $\Sigma$ of embedded dependencies is equivalent to a set of EGDs and TGDs. We write $A \models \Sigma$ if the instance $A$ satisfies all the constraints in $\Sigma$. We will extend our treatment to more expressive constraints in Section 6. All sets of constraints we refer to are finite.

**Homomorphisms.** A function $h : \mathrm{dom}(A) \rightarrow \mathrm{dom}(B)$ is a *homomorphism* if whenever $R(\bar{a})$ holds in $A$, $R(h(\bar{a}))$ holds in $B$ and if $h(c) = c$ for every constant in $A$. A homomorphism $h : A \rightarrow A$ is an *endomorphism* of $A$. We write $A \rightarrow B$ in case there is a homomorphism between $A$ and $B$. A homomorphism $r : A \rightarrow B \subseteq A$ is a *retraction* if $r$ is the identity on $\mathrm{dom}(B)$. A retraction is *proper* if it is not surjective. If $A \rightarrow B$ and $B \rightarrow A$, we say that $A$ and $B$ are homomorphically equivalent and we write $A \leftrightarrow B$. If $K$ is a set of instances, we extend $\rightarrow$ to sets of structures $K, L$ as follows:

$$K \rightarrow L \text{ iff } (\forall B \in L)(\exists A \in K)(A \rightarrow B).$$

It is easy to verify that this extension of $\rightarrow$ is reflexive and transitive. Notice that $K \subseteq L$ implies $L \rightarrow K$. We say that a structure or set of structures $T$ is *universal* for $K$ if $T \rightarrow K$.

**Data exchange.** We consider the setting where we have two schemas $\sigma$ and $\tau$ which do not share any relation symbols. Given an instance $S$ over $\sigma$ and instance $T$ over $\tau$, the instance $(S,T)$ over $\sigma \cup \tau$ is the instance which has all the relations in $S$ and all those in $T$. Given a set of constraints $\Sigma$ over $\sigma \cup \tau$, we say that $T$ is a *solution* for $S$ under $\Sigma$ if $(S,T) \models \Sigma$. When $\Sigma$ is clear from context, we simply say that $T$ is a solution for $S$. We say that $U$ is a *universal solution* for $S$ if it is a solution for $S$ and if it is universal for the set of all solutions for $S$. Furthermore, we require the homomorphisms witnessing this universality to be the identity on $\mathrm{dom}(S)$.

A constraint $\xi$ over $\sigma \cup \tau$ is *source-to-target* if the premise of $\xi$ is over $\sigma$ and the conclusion of $\xi$ is over $\tau$. We will consider the special case of settings where $\Sigma = \Sigma_{st} \cup \Sigma_t$ with $\Sigma_{st}$ a set of source-to-target TGDs and $\Sigma_t$ a set of EGDs and TGDs. With these restrictions, $(\sigma, \tau, \Sigma_{st}, \Sigma_t)$ is known in the literature [8] as a *data exchange setting*.

**Data integration.** We consider the same setting as for data exchange, adding an $r$-ary client query $Q$ over the target schema $\tau$. Given source instance $S$, we are interested in finding the *certain answers* to $Q$ for $S$ under $\Sigma$, denoted $\mathrm{cert}_Q^\Sigma(S)$ and defined as

$$\mathrm{cert}_Q^\Sigma(S) = \bigcap_{(S,T)\models\Sigma} Q(T).$$

# 3. CERTAIN ANSWERS

In this section, we show that, even when a single universal solution does not exist or when a universal solution exists but is insufficient for computing certain answers, there may exist an adequate *universal set solution* which does allows us to compute certain answers.

DEFINITION 1. Given a data exchange setting, we say that $U$ is a *universal set solution* for source instance $S$ under constraints $\Sigma$ iff $U$ is finite, contains only solutions, and is universal for the set of all solutions: $U \to \{T \,:\, (S,T) \models \Sigma\}$.

The following example shows that, if the queries are unions of conjunctive queries and the constraints are standard embedded dependencies beyond the source-to-target class, then it can be the case that there is no universal solution, so the certain answers cannot be computed in this way. However, there may be a universal *set* solution $U$, which suffices to compute certain answers correctly to any union of conjunctive queries $Q$, by computing $\bigcap_{T\in U} Q(T)$.

EXAMPLE 1. Let the source schema and target schema consist of the binary relation symbol $E$, respectively the quaternary $F$, and consider a source instance $S$:

$$S = \{E(a,b_1), E(b_1,c), E(a,b_2), E(b_2,c)\}.$$

The constraint set $\Sigma = \{\xi_{st}, \xi_{ts}\}$ connects the two schemas as follows:

$$\xi_{st}: \quad E(x,y), E(y,z) \to \exists u \exists w\, F(x,u,z,w)$$
$$\xi_{ts}: \quad F(x,u,z,w) \to E(x,u), E(u,z)$$

Consider the target query $Q(x,z) :- F(x,b_1,z,w) \vee F(x,b_2,z,w)$. It is easy to check that the set of solutions for $S$ contains, among others, $T_1 = \{F(a,b_1,c,w_1)\}$, $T_2 = \{F(a,b_2,c,w_2)\}$, where $w_1, w_2$ are distinct. Indeed, $(S,T_1)$ and $(S,T_2)$ satisfy $\Sigma$. Note that there are infinitely many solutions, since $w_i$ can take infinitely many values. However, it can be shown that each solution must include either $T_1$ or $T_2$, for some value of $w_1$, respectively $w_2$. Therefore, $Q$ has the certain answer $(a,c)$.

Note that there is no single universal solution $C$ yielding the certain answers to $Q$. This is because by universality, $C$ would have to map homomorphically into both $T_1$ and $T_2$ and therefore cannot contain the values $b_1$ or $b_2$. The answer to $Q$ on $C$ would therefore be empty and thus not coincide with the certain answers.

However, the certain answers can be computed from a universal *set* solution. It turns out (as will be detailed later) that a universal set solution $U$ in this setting contains precisely two elements, $U = \{\{F(a,b_1,c,w_1)\}, \{F(a,b_2,c,w_2)\}\}$, where $w_1, w_2$ are variables (labelled nulls) which can in principle take any value. It is easy to check that the certain answers to $Q$ can be obtained by computing $\bigcap_{T\in U} Q(T)$. Indeed,

$$Q(\{F(a,b_1,c,w_1)\}) \cap Q(\{F(a,b_2,c,w_2)\}) = \{(a,c)\} \cap \{(a,c)\} = \{(a,c)\}.$$

Example 1 is not a fortunate accident: Theorem 1 below shows that universal set solutions always yield the certain answers.

THEOREM 1. *If $W$ is a universal set solution for $S$ under $\Sigma$ and $Q \in \mathrm{UCQ}$ has arity $r$, then*

$$\mathrm{cert}_Q^\Sigma(S) = \mathrm{dom}(S)^r \cap \bigcap_{T\in W} Q(T).$$

PROOF. The inclusion $\subseteq$ is clear, since $W$ consists only of solutions for $S$ under $\Sigma$ and since $\mathrm{cert}_Q^\Sigma(S) \subseteq \mathrm{dom}(S)^r$ by genericity of $Q$. For the opposite inclusion it is enough to show that for every solution $T'$, there is $T \in W$ such that $Q(T) \cap \mathrm{dom}(S)^r \subseteq Q(T')$. Accordingly, pick a solution $T'$. Then there must be $T \in W$ and a homomorphism $h$ such that $h : T \to T'$ and $h$ is the identity on $\mathrm{dom}(S)$. Since UCQ is closed under homomorphisms by Theorem 12 below, we have $\bar{a} \in Q(T)$ implies $h(\bar{a}) \in Q(T')$. Furthermore, since $h$ is the identity on $\mathrm{dom}(S)$, we have $\bar{a} \in \mathrm{dom}(S)^r$ implies $h(\bar{a}) = \bar{a}$ and therefore

$$\mathrm{dom}(S)^r \cap Q(T) \subseteq \mathrm{dom}(S)^r \cap Q(T') \subseteq Q(T')$$

as desired. □

**Other kinds of universality.** In Definition 1, universality of a set solution is defined with respect to homomorphisms. We mention here that it is very useful to consider universality with respect to other kinds of mappings, since the resulting universal set solutions yield the certain answers for more expressive queries. We consider such mappings in Section 7.

THEOREM 2. *If $W$ is a universal set solution for $S$ under $\Sigma$ for*

1. *injective homomorphisms and $Q \in \mathrm{MonQ}$,*
2. *full homomorphisms and $Q \in \mathrm{UCQ}^\neg$, or*
3. *embeddings and $Q \in \mathrm{UCQ}^{\neg,\neq}$,*

*and $Q$ has arity $r$, then*

$$\mathrm{cert}_Q^\Sigma(S) = \mathrm{dom}(S)^r \cap \bigcap_{T\in W} Q(T).$$

PROOF. Essentially the same as that of Theorem 1, also using Theorem 12 below. □

The following example (adapted from [8]) pertains to part 1 in Theorem 2. It shows that, even if the constraints are source-to-target embedded dependencies, if the query contains even one non-equality, the universal solution is insufficient for correctly computing the certain answers. However, there is a universal *set* solution $U$ for injective homomorphisms which, according to Theorem 2, suffices for correct computation of certain answers.

EXAMPLE 2. Let the source schema consist of the binary relation symbol $E$, and the target schema contain binary relations $F, G$. Consider a source instance

$$S = \{E(a_1, b_1), E(a_2, b_2)\}.$$

The two schemas are connected by $\Sigma = \{\xi_{st}\}$, where

$$\xi_{st} : E(x, z) \rightarrow \exists y\, F(x, y), G(y, z)$$

Consider the query $Q(x, z) :- F(x, y), G(y', z), (y \neq y')$. $Q$ has no certain answers, since its result on the solution

$$T_1 = \{F(a_1, y), G(y, b_1), F(a_2, y), G(y, b_2)\}$$

is already empty. However, a universal solution according to [8] is

$$T_0 = \{F(a_1, y_1), G(y_1, b_1), F(a_2, y_2), G(y_2, b_2)\}$$

and, as observed there, the result of $Q$ on $T_0$ is non-empty: $Q(T_0) = \{(a_1, b_2), (a_2, b_1)\}$. As shown in Section 5, there exists a universal *set* solution $U$ for injective homomorphisms, which correctly captures the certain answers to $Q$. Indeed, $U$ contains two instances, $U = \{T_0, T_1\}$. Then $Q(T_0) \cap Q(T_1)$ correctly yields the empty set of certain answers. Notice that there is a homomorphism, but no injective homomorphism, from $T_0$ into $T_1$.

Universal set solutions do not always exist:

EXAMPLE 3. Consider the constraints $\Sigma$:

$$\begin{aligned} S(x, y) &\rightarrow T(x, y) \\ T(x, y) &\rightarrow \exists z\, T(y, z) \\ T(x, y), T(y, z) &\rightarrow T(x, z) \end{aligned}$$

and a source $S$ containing a single edge. Then any solution, since it must be finite, must have a cycle. But for any finite set of solutions $U$, there must be a solution $C_n$ which is a cycle larger than any cycle in $U$. Then $U \not\rightarrow C_n$. Therefore, there is no universal set solution for $S$ under $\Sigma$.

# 4. TEMPLATES

Universal set solutions depend not only on the constraints $\Sigma$, but also on the source $S$. In this section, we generalize universal set solutions by introducing the concept of *templates*. Given a set of constraints $\Sigma$, we define a set of constraints $\Sigma_S$ such that the universal set solution for $S$ under $\Sigma$ is precisely a template for $\Sigma_S$. Templates have other applications outside of data exchange and data integration, including checking query containment under constraints. We will see in Section 5 that the chase is a general algorithm for producing templates, not just universal set solutions, so the extra generality gives a better understanding of the power and limitations of the chase.

Furthermore, we will gain some insight on the nature of the universal set solution for $S$ under $\Sigma$ by looking at the form of the constraints $\Sigma_S$. In particular, it will turn out that for embedded dependencies $\Sigma$, $\Sigma_S$ is not always equivalent to a set of embedded dependencies, but when it is, then there is a universal solution if and only if there is a universal set solution.

DEFINITION 2. We define $\Sigma_S$ so that

$$T \models \Sigma_S \text{ iff } (S, T) \models \Sigma$$

by replacing every occurrence in $\Sigma$ of $R\bar{x}$ where $R$ is a relational symbol in $\sigma$ with $\bigvee_{\bar{c} \in R^S} \bar{x} = \bar{c}$. This may give constraints which have disjunction in the premise, but we "normalized away" such disjunctions. On the other hand, when there are any constraints with relation symbols from $\sigma$ in the conclusion, we may get disjunction in the conclusion which can not be normalized away.[1]

EXAMPLE 4. We revisit Example 2, where $\Sigma = \{\xi_{st}\}$,

$$\xi_{st} : E(x, z) \rightarrow \exists y\, F(x, y), G(y, z)$$

and the source $S$ is $\{E(a_1, b_1), E(a_2, b_2)\}$. The set $\Sigma_S$ contains the single constraint

$$x = a_1, z = b_1 \vee x = a_2, z = b_2 \quad \rightarrow \quad \exists y\, F(x, y), G(y, z)$$

which is equivalent to the set of TGDs

$$\begin{aligned} x = a_1, z = b_1 &\rightarrow \exists y\, F(x, y), G(y, z) \\ x = a_2, z = b_2 &\rightarrow \exists y\, F(x, y), G(y, z) \end{aligned}$$

It is easy to see that $T$ is a solution for $S$ under $\Sigma$ iff $T \models \Sigma_S$.

DEFINITION 3. A set of finite structures $T$ is a *template* for a set of structures $K$ if it satisfies the following conditions:

1. (universality) $T \rightarrow K$,
2. (conformance) $T \subseteq K$,
3. (finiteness) $T$ is finite, and
4. (minimality) there is no $T' \subset T$ such that $T' \rightarrow T$.

These conditions imply that there is no $T'$ satisfying 1, 2, and 3 such that $|T'| \leq |T|$. We write $[K]$ for the template of $K$, if it exists and we write $[\Sigma]$ for $[\mathrm{mod}(\Sigma)]$, where $\mathrm{mod}(\Sigma)$ is the class of all finite models of $\Sigma$.

The connection between universal set solutions and templates is as follows: a template for $\Sigma_S$ is a universal set solution for $S$ under $\Sigma$. (We give the precise statement in Proposition 3 below.) Therefore, templates can also be used to compute certain answers.

THEOREM 3. *If $[\Sigma_S]$ exists and $Q$ is a conjunctive query of arity $r$, then*

$$\begin{aligned} \mathrm{cert}_Q^{\Sigma}(S) &= \mathrm{dom}(S)^r \cap \bigcap_{T \in [\Sigma_S]} Q(T) \\ &= \{\bar{c} : \Sigma_S \models Q(\bar{c})\}. \end{aligned}$$

PROOF. The proof of the first equation is very similar to that of Theorem 1, except that we use the fact that if $T'$ is a solution for $S$ under $\Sigma$, then $T' \models \Sigma_S$ and therefore there is $T \in [\Sigma_S]$ such that $T \rightarrow T'$. The second equation follows immediately from the definition of certain answers and $\Sigma_S$. □

**Why standard data exchange admits a universal solution.** We have defined templates for arbitrary constraints, which are not necessarily source-to-target, and may contain disjunction. This case goes beyond the data exchange setting in the literature [8]. We now prove from the basic properties of templates the fact that in the particular case of embedded source-to-target dependencies, the template is a singleton (or, equivalently, there exists a universal solution).

PROPOSITION 1. *If $\Sigma$ is a set of embedded dependencies where all conclusions are over $\tau$, then $\Sigma_S$ is a set of embedded dependencies. In particular, this holds in the case where $\Sigma$ is a set of source-to-target TGDs and target-to-target TGDs and EGDs, as in [8].*

[1] Since here we start chasing with an empty instance, we will need to allow for chase steps (see Section 6) of the form $A_n \stackrel{\xi,\bar{a}}{\rightarrow} A_{n+1}$ where $\bar{a}$ is not necessarily in $A_n$ for these constraints to fire once we chase.

Embedded dependencies have some nice closure properties.

THEOREM 4. *If $\Sigma$ is a set of embedded dependencies, then $\Sigma$ is closed under retractions and products. That is:*

1. *If $A \models \Sigma$ and $A \hookrightarrow B$, then $B \models \Sigma$.*
2. *If $A, B \models \Sigma$, then $A \times B \models \Sigma$.*

In particular, closure under products is enough to guarantee that universal set solutions are in fact simply universal solutions.

PROPOSITION 2. *If $[K]$ exists and $K$ is closed under products, then $[K]$ is a singleton. Furthermore, in this case* $\mathrm{core}(\{\prod_{A\in K} A\})$ *is a template for $K$.*

COROLLARY 1. *If there is a universal set solution for $S$ under $\Sigma$ and $\Sigma$ is a set of source-to-target and target-to-target TGDs and EGDs, $\Sigma_S$ is a set of embedded dependencies, or $\Sigma_S$ is closed under products, then there is a universal solution for $S$ under $\Sigma$.*

**Templates for finite and infinite solutions.** We say that $T$ is a *strong template* for $\Sigma$ if $T$ is a template for $\mathrm{imod}(\Sigma)$, the class of all models (finite and infinite) of $\Sigma$. We say that $T$ is a *weak template* for $\Sigma$ if $T$ is a template for $\mathrm{mod}(\Sigma)$, the class of all finite models of $\Sigma$.

PROPOSITION 3. *A weak template for $\Sigma_S$ is precisely a universal set solution for $S$ under $\Sigma$.*

Clearly, any strong template is also a weak template. However, the converse is not true, as shown by the following separation result. We discuss in Section 5 how we compute both template flavors.

THEOREM 5.

1. *There is a set $\Sigma$ of TGDs which has a weak template, but no strong template.*
2. *There is a set $\Sigma$ of TGDs which has no weak template.*

PROOF. (1) Consider the set of axioms $\Sigma_1$:

$$\begin{array}{lll} \xi_1: & & \exists x, y\, E(x,y) \\ \xi_2: & E(x,y) & \rightarrow \exists z\, E(y,z) \\ \xi_3: & E(x,y), E(y,z) & \rightarrow E(x,z) \end{array}$$

Any model of axioms $\xi_1$ and $\xi_2$ must have an infinite walk. Therefore, if the model is finite, it must have a cycle. If it has a cycle, then by axiom $\xi_3$ it must have a self-loop. Since the structure with a single self-loop satisfies these axioms, it is a weak template for $\Sigma_1$. On the other hand, the transitive closure of an infinite path also satisfies axioms $\xi_1, \xi_2$, and $\xi_3$, but no finite structure with a cycle has a homomorphism into it. Therefore $\Sigma_1$ has no strong template.

(2) Now consider $\Sigma := \{\xi_1, \xi_2\}$. As we have seen above, any finite model of $\Sigma$ must have a cycle. But for any finite set $U$ of such models, there is another model $C_n$, a cycle larger than any cycle in $U$ and therefore $U \not\rightarrow C_n$. □

# 5. COMPUTING TEMPLATES

Given a set of constraints $\Sigma$, we are interested in computing the template $[\Sigma]$. In this section we concentrate on computing templates for embedded dependencies. To this end, we start from the well-known chase procedure [2, 14, 13, 7, 3, 1], extending it to a novel procedure called the *unordered minimizing chase* which turns out to be strictly better at computing templates than the standard chase. We will show in Section 6 how to extend the chase to compute templates for larger classes of constraints and under more general universality assumptions, as required by queries which are more expressive than unions of conjunctive queries.

By Theorem 4 and Proposition 2, if they exist, templates for embedded dependencies consist of a single structure. To simplify the presentation we refer to this single structure also as a template.

DEFINITION 4. **(Chase Step)** If $\xi$ is a TGD or EGDs, we write $A \overset{\xi,\bar{a}}{\rightarrow} B$ if

1. $A \models \exists P_\xi(\bar{a})$,
2. $A \not\models \exists C_\xi(\bar{a})$, and
3. $B = \begin{cases} A\bar{a} \oplus C_\xi & \text{if } \xi \text{ is a TGD} \\ h(A) & \text{if } \xi \text{ is an EGD} \end{cases}$

where $A\bar{a} \oplus C_\xi$ is the result of attaching to $A$ a copy of $C_\xi$ by identifying $\bar{a}$ with the free variables of $C_\xi$ and where $h(a_i) = h(a_j) = a_i$ and $h$ is the identity elsewhere in case $\exists P_\xi$ has as free variables $\bar{u}$ and $C_\xi$ is $u_i = u_j$. If 1 and 2 hold, we say that $\xi$ *applies* to $A$ on $\bar{a}$. We do not require $\bar{a} \in \mathrm{dom}(A)$, which is important in case the premise has constants.

DEFINITION 5. (**Chase Sequence**) Assume $\Sigma$ is a set of EGDs and TGDs.

1. A $\Sigma$-*chase sequence* $S$ (or just *chase sequence* if $\Sigma$ is clear from context) is a sequence of structures $A_0, A_1, \ldots$ such that every structure $A_{s+1}$ in it is obtained from the previous one $A_s$ by a chase step. That is, there are $\xi \in \Sigma$ and $\bar{a}$ such that $A_s \overset{\xi,\bar{a}}{\rightarrow} A_{s+1}$. We say that $S$ starts with $A$ if $A_0 = A$.
2. A chase sequence $A = A_0, \ldots, A_n$ is *terminating* if $A_n \models \Sigma$. In this case we say that $A^\Sigma = A_n$ is the *result* of the chase.
3. We say that the chase *terminates* if there is a terminating chase sequence. $A^\Sigma$ is defined whenever there is some terminating chase sequence starting with $A$, but its value may depend on the chase sequence. We will see later that all chase results are homomorphically equivalent, so we can often speak about $A^\Sigma$ without referring to a particular chase sequence.
4. We say that an infinite chase sequence is *fair* if whenever $\xi$ applies to $A_s$ on $\bar{a}$ there is some $r > s$ such that $A_r \models \exists C_\xi(\bar{a})$.
5. If $\Sigma$ consists of TGDs only and $A = A_0, A_1, \ldots$ is an infinite chase sequence, we set $A^\Sigma_\omega = \bigcup_i A_i$. If no sequence is specified, we take $A^\Sigma_\omega$ to be obtained as above from any fair infinite chase sequence. Then $A^\Sigma_\omega$ is only defined up to homomorphic equivalence as in the case of $A^\Sigma$.

The following theorem lists some essential properties of the chase, which we will generalize for the extended chase. There results are considered folklore or appear implicitly in proofs related to the chase [2, 14, 13, 7, 3, 1].

THEOREM 6. *If $\Sigma$ is a set of TGDs and EGDs, $A = A_0, A_1, \ldots$ is a finite or infinite chase sequence, and*

$$K = \{B \in \mathrm{imod}(\Sigma) : A \rightarrow B\},$$

*then:*

1. $A_0 \rightarrow A_1 \rightarrow A_2 \rightarrow \ldots$
2. $A_0, A_1, A_2, \ldots \rightarrow K$

3. *If $A^\Sigma$ is defined, then $A^\Sigma \models \Sigma$.*
4. *If $\Sigma$ consist of TGDs only, then $A^\Sigma_\omega$ is universal for $K$ and $A^\Sigma_\omega \models \Sigma$.*
5. *If $B \models \Sigma$ and there is a homomorphism $h : A_n \to B$, then there is a homomorphism $h' : A_{n+1} \to B$ extending $h$ (if $\xi$ is a TGD) or identifying some values in the domain of $h$ (if $\xi$ is an EGD).*
6. *If $B \models \Sigma$, $A^\Sigma$ is defined, and there is a homomorphism $h : A \to B$, then there is a homomorphism $h' : A^\Sigma \to B$.*

PROOF. See the appendix. □

COROLLARY 2.

1. *If $A^\Sigma$ is defined, then $A^\Sigma$ is a template for*
$$K = \{B \in \mathrm{imod}(\Sigma) : A \to B\}.$$
2. *If $\emptyset^\Sigma$ is defined, then $\emptyset^\Sigma$ is a strong template for $\Sigma$.*

PROOF. Immediate from parts 2 and 3 in Theorem 6. □

We say that any sequence is *chase-like* if it satisfies the conditions of Theorem 6. We say that an algorithm is *chase-like* if it produces chase-like sequences. It would be nice to have an algorithm that satisfied the conditions of Theorem 6 only for finite instances, but it is not clear at all how such algorithm can be obtained. Certainly, it can not be obtained by simply adding a copy of the conclusion of some constraints to the next instance in the sequence. Therefore, any chase-like algorithm, if it terminates, produces a strong template. A natural question, then, is the following:

Is the chase complete for strong templates?

That is, for any $\Sigma$ with a strong template, will the chase always find it? More precisely, will any (long enough) $\Sigma$-chase sequence terminate? For the chase outlined above, the answer is no, as the following example shows.

EXAMPLE 5. Consider the set $\Sigma$ consisting of following TGDs:

$$\begin{array}{lrll} \xi_1\colon & & \to & \exists u, v\, E(u,v), E(v,u) \\ \xi_2\colon & E(x,y), E(y,x) & \to & \exists u\, E(u,u) \\ \xi_3\colon & E(x,y) & \to & \exists u\, E(x,u), E(u,y) \end{array}$$

The singleton template consisting of the self-loop is a template for $\Sigma$, yet any $\Sigma$-chase sequence starting with $\emptyset$ must be infinite. This is because $\xi_1$ must fire first to give a cycle of length 2. Assume $\xi_2$ fires next to give a disjoint self-loop. From now on, ignore this loop. Set $A_0 := C_2$, the cycle of length 2. Now it is easy to show that if $A_s \overset{\xi_3, a, b}{\to} A_{s+1}$ where $a \neq b$, then two new edges $ac$ and $cb$ are added to $A_{s+1}$ and that

$$A_{s+1} \not\models \exists C_{\xi_3}(ac) \text{ and } A_{s+1} \not\models \exists C_{\xi_3}(cb).$$

Therefore, if $A_{s+1} \not\models \xi_3$, which results in an infinite chase sequence.

**A Complete Chase.** We now define a novel chase procedure, the *unordered minimized chase* or *um-chase*, which is order-independent and which, more importantly, is complete for strong templates, and therefore superior to the chase introduced above (which we also call the *ordered chase*) for the task of finding templates, which is what chase-like algorithms are all about.

Intuitively, the unordered chase step proceeds by first firing all applicable standard chase steps simultaneously, then minimizing the resulting structure by computign its core. We formalize the procedure below.

DEFINITION 6. (**Unordered-chase step**) If $\Sigma$ is a set of TGDs, we write $A \overset{\Sigma}{\to} B$ if

1. $A \not\models \Sigma$ and
2. $B = \bigcup_{\xi \in \Sigma, \bar{a} \in \mathrm{dom}(A), A \overset{\xi,\bar{a}}{\to} D} D$.

That is, $B$ is the structure obtained from $A$ by simultaneously firing all applying chase steps. If $\Sigma$ also contains EGDs, then we also identify all elements which have been identified by every constraint $\xi$ and any tuple $\bar{a}$ such that $\xi$ applies to $A$ on $\bar{a}$.

For the minimization part of the next step, we need to define cores. An instance is a *core* if it has no proper retractions. A core $C$ of an instance $A$ is a retract of $A$ which is a core. Cores of an instance $A$ are unique up to isomorphism and therefore we can talk about *the core* of $A$. It follows that $A$ and $B$ are homomorphically equivalent iff their cores are isomorphic.

DEFINITION 7. (**Unordered-minimizing-chase step**) We write $A \overset{\Sigma\downarrow}{\to} B$ if $A \overset{\Sigma}{\to} B'$ and $B = \mathrm{core}(B')$.

We extend the definition of chase sequence to um-chase sequence in the obvious way. Notice that um-chase sequences are determined up to isomorphism, since cores are determined up to isomorphism. We use the notation $A^\Sigma$ to refer also to the result of a terminating um-chase sequence. Such result is unique up to isomorphism. Similarly, we use $A^\Sigma_\omega$ also for the um-chase. Which chase we have in mind should be clear from context. Theorem 6 also holds for the um-chase.

THEOREM 7. *If $\Sigma$ is a set of EGDs and TGDs, then*

1. *The unordered minimizing chase terminates iff there is a strong template for $\Sigma$.*
2. *The result of the unordered minimizing chase is a strong template for $\Sigma$.*

PROOF. See the appendix. □

COROLLARY 3. *If $\Sigma$ is a set of embedded dependencies and any chase-like algorithm terminates on input $A$ under $\Sigma$ to give $U$, then the unordered minimizing chase terminates on input $A$ and gives the strong template $A^\Sigma$ which is homomorphically equivalent to $U$.*

PROOF. If such chase-like algorithm terminates on input $A$ under $\Sigma$ giving $U$, then $U$ is a strong template by Corollary 2. Therefore, by Theorem 7 part 1, the unorderd minimizing chase terminates and by part 2 gives a strong template $A^\Sigma$. Since both $U$ and $A^\Sigma$ satisfy $\Sigma$ by Theorem 6, we have $U \to A^\Sigma$ and $A^\Sigma \to U$ by their universality. □

## 5.1 Conditions for Termination

A widely-applicable, sufficient condition on $\Sigma$ for the termination of the chase, weak acyclicity was identified by Alin Deutsch and Lucian Popa and first published in [6] and [8]. We give this definition below, then we introduce a strictly more general condition, *stratification*, which is also sufficient for termination of the chase.

DEFINITION 8. (**Weakly Acyclic**)[6, 8] A *position* is a pair $(R, i)$ (which we write $R^i$) where $R$ is a relation symbol of arity $r$ and $i$ satisfies $1 \leq i \leq r$. The *dependency graph* of a set $\Sigma$ of TGDs is a directed graph where the vertices are the positions of the relation symbols in $\Sigma$ and, for every TGD $\xi$ of the form

$$\phi(\bar{u}, \bar{w}) \to \exists \bar{v}\, \psi(\bar{u}, \bar{v})$$

there is an edge between $R^i$ and $S^j$ whenever (1) some $u \in \{\bar{u}\}$ occurs in $R^i$ in $\phi$ and in $S^j$ in $\psi$ or (2) some $u \in \{\bar{u}\}$ appears in $R^i$ in $\phi$ and some $v \in \{\bar{v}\}$ occurs in $S^j$ in $\psi$. Furthermore, these latter edges are labelled with $\exists$ and we call them *existential edges*. $\Sigma$ is *weakly acyclic* if its dependency graph has no cycles with an existential edge.

Notice that any set of source-to-target TGDs is weakly acyclic. We say that a set $\Sigma$ of TGDs and EGDs is *weakly acyclic* if the $\Sigma' \subseteq \Sigma$ consisting of the TGDs in $\Sigma$ is weakly acyclic.

THEOREM 8. *[8, 6] For every weakly-acyclic set $\Sigma$ of EGDs and TGDs, there are $b$ and $c$ such that, for any $A$, regardless of the order of the chase,*

1. *$A^\Sigma$ is defined, and*
2. *$A^\Sigma$ can be computed in $O(|A|^b)$ steps and in time $O(|A|^c)$.*

Now consider the following example.

EXAMPLE 6. Consider the set $\Sigma = \{\xi\}$ where $\xi$ is the following TGD:

$$\exists y\, R(x,y), R(y,x) \rightarrow \exists u, v\, R(x,u), R(u,v), R(v,x).$$

It is easy to check that $\Sigma$ is not weakly acyclic, yet it is clear that $A^\Sigma$ is defined for the ordered chase for any chase order since introducing 3-cycles will never create any new 2-cycles.

We introduce a strictly wider, effectively checkable condition which is also sufficient for termination of the chase, which is motivated by the example above.

DEFINITION 9. (**Stratified**) Given EGDs or TGDs $\alpha$ and $\beta$ we write $\alpha \prec \beta$ if there exists $A$, $B$, $\bar{a} \in \mathrm{dom}(A)$, and $\bar{b} \in \mathrm{dom}(B)$ such that

1. $\beta$ does not apply to $A$ on $\bar{b}$, possibly because $\{\bar{b}\} \not\subseteq \mathrm{dom}(A)$,
2. $A \overset{\alpha,\bar{a}}{\rightarrow} B$ for some $B$, and
3. $\beta$ applies to $B$ on $\bar{b}$

The *chase graph* $G$ of a set of TGDs $\Sigma$ has as vertices the constraints in $\Sigma$ and there is an edge between two constraints $\alpha, \beta \in \Sigma$ if $\alpha \prec \beta$. We say that $C$ is a *cycle-component* of $G$ if $C$ is a connected component of the graph $G'$ consisting only of those edges which participate in cycles. A set of EGDs and TGDs $\Sigma$ is *stratified* if every cycle component of the chase graph of $\Sigma$ is weakly-acyclic.

THEOREM 9. *If $\Sigma$ is a stratified set of EGDs and TGDs, then there exists $c$ such that for every $A$, the length of every chase sequence $A_0^\Sigma, A_1^\Sigma, \ldots$ is bounded by $|A|^c$.*

PROOF. (sketch) Consider the chase graph $G$ of $\Sigma$ and its associated graph $G'$ where every cycle component has been replaced by a single vertex. $G'$ is acyclic. Each vertex in $G'$ is a set of weakly-acyclic TGDs or a single TGD. Now chase as follows. Pick a vertex $v$ of indegree 0 in $G'$, chase with the associated TGD or weakly-acyclic set of TGDs until this subchase terminates, remove $v$ from $G'$ and repeat. Each subchase must terminate because $\beta$ fires after $\alpha$ only if $\alpha \prec \beta$ and because a subchase with a weakly-acyclic set of TGDs terminates by Theorem 8. □

THEOREM 10. *Given TGDs $\alpha$ and $\beta$, checking whether $\alpha \prec \beta$ holds is decidable.*

PROOF. See the appendix. □

THEOREM 11. *Weakly-acyclic sets of EGDs and TGDs are stratified, but not conversely.*

PROOF. If a set of EGDs and TGDs is weakly acyclic, then it is stratified by the definition. The set $\Sigma = \{\xi\}$ from Example 6 satisfies $\xi \not\prec \xi$ and therefore is stratified, yet not acyclic. To see this, notice if $A \overset{\alpha,\bar{a}}{\rightarrow} B$ then $B$ has no new 2-cycles or 1-cycles, that is, no such cycles which are not already in $A$. □

# 6. BEYOND EMBEDDED DEPENDENCIES

In this section we extend our consideration to constraints of the form

$$\bigvee_{1 \le i \le p} \phi_i(\bar{u}, \bar{w}) \rightarrow \exists \bar{v} \bigvee_{1 \le i \le c} \psi_i(\bar{u}, \bar{v})$$

where each $\phi_i$ and $\psi_i$ is a conjunction of relational atoms, negated relational atoms, equations, or inequalities. We call such constraints *negation disjunctive embedded dependencies* or *NDEDs* to be consistent with the name *disjunctive embedded dependencies* or *DEDs* for the same class of constraints without negation introduced in [6]. It is easy to check that every $\forall\exists$ constraint is equivalent to an NDED. We extend the results of the previous sections to these kinds of constraints. To do this, we first show how to handle disjunction and then we show how to handle negation.

**Adding disjunction.** First of all, notice DEDs that are not closed under products as the following example shows. This is in constrast to embedded dependencies (cf. Theorem 4).

EXAMPLE 7. The following disjunctive TGD $\xi$

$$\exists u, v Euv, Evu \text{ or } \exists u, v, w Euv, Evw, Ewu$$

is not closed under products. We have $C_2, C_3 \models \xi$ where $C_k$ is the directed cycle of length $k$, yet $C_2 \times C_3 = C_6$ and $C_6 \not\models \xi$.

As a consequence, a template for a set $\Sigma$ of such constraints is not necessarily a singleton set. We now explain an extended chase for DEDs introduced in [6]. The comments above imply that such chase must have at each step not a single instance, but instead a set of instances. Therefore, we aim to define $K \overset{\xi}{\rightarrow} L$ where $K$ and $L$ are finite sets of instances and $\xi$ is an DED to parallel Definition 4. Then we extend the results from Section 5 and we show how to extend them to NDEDs.

DEFINITION 10. (**Extended Chase Step**) First assume that $\xi$ is a DED of the form shown above. Set

$$\xi_i := \bigvee_{1 \le i \le p} \phi_i(\bar{u}, \bar{w}) \rightarrow \exists \bar{v}, \psi_i(\bar{u}, \bar{v})$$

so that $P_\xi = P_{\xi_1} = \ldots = P_{\xi_p}$ and $C_\xi = \bigvee_{1 \le i \le c} C_{\xi_i}$. We write $A \overset{\xi,\bar{a}}{\rightarrow} \{B_1, \ldots, B_c\}$ if

1. $A \models \exists P_\xi(\bar{a})$,
2. $A \not\models \exists C_\xi(\bar{a})$, and
3. for each $i$, $A \overset{\xi_i,\bar{a}}{\rightarrow} B_i$.

If 1 and 2 hold, we say that $\xi$ *applies* to $A$ on $\bar{a}$. Notice that this is consistent since if 2 holds, then also $A \not\models \exists C_{\xi_i}(\bar{a})$ for every $i$. That is, we create one new instance for every disjunct in the conclusion.

We write $K \overset{\xi,\bar{a}}{\rightarrow} L$ where $K$ and $L$ are finite sets of instance if

$$L = K_1 \cup \bigcup_{A \in K_2, A \overset{\xi,\bar{a}}{\rightarrow} M} M$$

where

$$K_1 := \{A \in K : \{\bar{a}\} \not\subseteq A \text{ or } A \not\models \exists P_\xi(\bar{a}) \text{ or } A \models \exists C_\xi(\bar{a})\}$$

and $K_2 := K - K_1$. That is, $K_1$ is the set of instances in $K$ to which $\xi$ does not apply on $\bar{a}$ and $K_2$ is the set of instances in $K$ to which $\xi$ does apply on $\bar{a}$. The instances in $L$ are those obtained by a chase step with $\xi$ and $\bar{a}$ from an instance in $K$ or those instances in $K$ to which $\xi$ does not apply on $\bar{a}$.

Given this definition of a chase step, the definitions of chase sequence, chase result, etc. from Section 5 extend naturally to the situation where at each step we have a finite set of instances instead of a single instance. This extension was made in [6]. We keep the notation of Section 5 since this makes the presentation simpler and more intuitive. Notice that $A^{\Sigma}_{\omega}$ may now be an infinite set of instances. Then Theorems 6 and Theorems 7 go through with DEDs instead of EGDs and TGDs. Only minor and straightforward changes are needed in their proofs.

**Adding negation.** We now explain a further extension to handle negation, introduced in [4]. We first extend DEDs with constraints that may have $\bot$ as their conclusion and we extend the definition of a chase step as follows. If $\xi$ has $\bot$ as its conclusion, then

$$K \overset{\xi,\bar{a}}{\to} L \text{ iff } L = \{A \in K, A \not\models \exists P_{\xi}(\bar{a})\}.$$

That is, if $\xi$ applies to $A$ on $\bar{a}$, it "kills" $A$. We call such constraint *DEDFs* for *DEDs with falsehood*. Implicitly, such constraints are already needed for a much smaller kinds of constraints in order to handle contradictions which arise, for example, from equating two constants.

Using DEDFs, we can simulate NDEDs. The details of this simulation are given in the next section; here we only provide a rough sketch. Given a set $\Sigma$ of NDEDs over signature $\sigma$, we compute a set $\hat{\Sigma}$ of DEDFs over the signature

$$\hat{\sigma} := \sigma \cup \{\hat{R} : R \in \Sigma \cup \{N\}$$

where $\hat{R}\bar{x}$ "stands for" $\neg R\bar{x}$ and $Nxy$ "stands for" $x \neq y$. If the chase terminates, the result will be a template under embeddings, rather than homomorphisms. From such result, it is straightforward to extract a template under homomorphisms (cf. Theorem 14).

# 7. BEYOND UCQ AND HOMOMORPHISMS

In order to be able to compute certain answers to queries which are not unions of conjunctive queries, we need set solutions which are universal not under homomorphisms, but under other kinds of mappings. Universality under other kinds of mappings is useful also for other applications, including checking containment under constraints.

We are interested in the following kinds of homomorphisms. We say that a homomorphism $h : A \to B$ is *full* if $A \models R(\bar{x})$ iff $B \models R(h\bar{x})$ for all relations $R$ in $A$ and $B$. An *embedding* is a full injective homomorphism. We write $\mathrm{hom}$ for the set of homomorphisms on finite instances, $\mathrm{ihom}$ for injective homomorphisms, $\mathrm{fhom}$ for full homomorphisms, and $\mathrm{emb}$ for embeddings.

THEOREM 12.

1. $\mathrm{UCQ}$ *is closed under homomorphisms.*
2. $\mathrm{MonQ}$ *is closed under injective homomorphisms.*
3. $\mathrm{UCQ}^{\neg}$ *is closed under full homomorphisms.*
4. $\mathrm{UCQ}^{\neg,\neq}$ *is closed under embeddings.*

*That is, for any of these classes of queries and the corresponding class of mappings,*

$$h : A \dashrightarrow B \text{ and } \bar{a} \in Q(A) \text{ implies } h(\bar{a}) \in Q(B).$$

PROOF. See the appendix. □

We fix some family $F$ of mappings on finite instances such as injective homomorphisms and we write $A \dashrightarrow B$ if there is $h : A \to B$ in $F$. We extend $\dashrightarrow$ to sets of structures as we did for $\to$. We say that a structure or set of structures $T$ is $F$*-universal* for $K$ if $T \dashrightarrow K$.

DEFINITION 11. A set of finite structures $T$ is an $F$*-template* for a set of structures $K$ if it satisfies the following conditions:

1. ($F$-universality) $T \dashrightarrow K$,
2. (conformance) $T \subseteq K$,
3. (finiteness) $T$ is finite, and
4. (minimality) there is no $T' \subset T$ such that $T' \to T$.

Therefore, a plain template is a $\mathrm{hom}$-template.

It turns out that we can compute strong templates under $F$ in case $F \in \{\mathrm{ihom}, \mathrm{fhom}, \mathrm{emb}\}$ by using a reduction to the case of homomorphisms. Other families of mappings may be handled by similar reductions.

THEOREM 13. *Given $F \in \{\mathrm{ihom}, \mathrm{fhom}, \mathrm{emb}\}$ and a signature $\sigma$, there is a signature $\hat{\sigma} \supset \sigma$ and constraints $\Lambda$ such that for every $A, B$ over $\sigma$, there are unique expansions $\hat{A}, \hat{B}$ over $\hat{\sigma}$ of $A, B$ satisfying $\hat{A}, \hat{B} \models \Lambda$. Furthermore,*

$$h : A \dashrightarrow B \text{ iff } h : \hat{A} \to \hat{B}$$

PROOF. (a) If $F$ consists of injective homomorphisms, set $\hat{\sigma} := \sigma \cup \{N\}$ where $N$ is a new binary relation symbol and set $\Lambda$ to contain the constraints

$$x = y \vee N(x, y) \qquad x = y, N(x, y) \to \bot$$

where $N$ stands for $\neq$. Now assume 1, 2, and 3 hold. If $h : \hat{A} \to \hat{B}$ is a homomorphism, then also $h$ is a homomorphism $A \to B$. Now if $x \neq y$, we must have $A \models Nxy$ by the first constraint in $\Lambda$ and therefore $B \models Nh(x)h(y)$. Then the second constraint in $\Lambda$ implies $h(x) \neq h(y)$. That is, $h$ is injective. The converse is obvious.

(b) If $F$ consists of full homomorphisms, set $\hat{\sigma} := \sigma \cup \{\hat{R} : R \in \sigma\}$ where each relation symbol $\hat{R}$ is new and of the same arity as $R$. Set $\Lambda$ to contain all constraints of the form

$$R(\bar{x}) \vee \hat{R}(\bar{x}) \qquad R(\bar{x}), \hat{R}(\bar{x}) \to \bot$$

for every relation symbol $R \in \sigma$. The rest of the proof is similar to case (a).

(c) Embeddings are precisely full injective homomorphisms, so this case is handled by combining (a) and (b).

The one-to-one correspondence between $A$ and $\hat{A} \models \Lambda$ is clear. □

THEOREM 14. *If $\Sigma$ is a set of NDEDs over signature $\sigma$ and $F \in \{\mathrm{ihom}, \mathrm{fhom}, \mathrm{emb}\}$, then there is a signature $\hat{\sigma}$ and a set of DEDFs $\hat{\Sigma}$ such that the following are equivalent*

1. *There is a strong template for $\Sigma$.*
2. *There is a strong $F$-template for $\Sigma$.*
3. *There is a strong template for $\hat{\Sigma}$.*
4. *The unordered chase with minimization terminates for $\hat{\Sigma}$, producing a strong template for $\hat{\Sigma}$.*

*Furthermore, $\hat{\Sigma}$ can be obtained efficiently from $\Sigma$ and each of these strong templates can be efficiently obtained from the other.*

PROOF. See the appendix. □

Theorem 14 allows us to compute templates and universal set solutions for NDED constraints.

# 8. CONTAINMENT

Templates are useful for checking containment, containment under constraints, and implication as we show next.

DEFINITION 12. We write $P \sqsubseteq Q$ in case $P(A) \subseteq Q(A)$ for all finite structures $A$. and $P \sqsubseteq_\Sigma Q$ in case $P(A) \subseteq Q(A)$ for all finite structures $A \models \Sigma$. The *containment problem* consist of, given $P$ and $Q$, deciding whether $P \sqsubseteq Q$. The *containment under constraints problem* consist of, given $P, Q$, and $\Sigma$, deciding whether $P \sqsubseteq_\Sigma Q$.

In order to state general results in a simple manner we consider some fixed class of mappings $F$ closed under composition, such as hom, fhom, ihom, or emb and we consider the corresponding $F$-templates. We write $A \dashrightarrow B$ if there is a mapping $h : A \to B$ such that $h \in F$. We say that a class of structures $K$ is *closed* under $\dashrightarrow$ if $A \in K, A \dashrightarrow B$ imply $B \in K$. Similarly, we say that $\Sigma$ is *closed* under $\dashrightarrow$ if $\mathrm{mod}\,(\Sigma)$ is.

In a sense, an $F$-template is an incomplete, but finite description of a class $K$, unless that class $K$ is closed under $F$. The importance of the result below, is that reduces an infinite problem (part 1) to checking finitely many instances for the existences of a mapping (part 3).

THEOREM 15 (CONTAINMENT). *If $K, L$ are sets of structures, $L$ is closed under $\dashrightarrow$, and $[K]$ and $[L]$ exists, then the following are equivalent.*

1. $K \subseteq L$.
2. $[K] \subseteq L$.
3. $[L] \dashrightarrow [K]$.

PROOF. If (1) holds, then $[K] \subseteq K \subseteq L$ so (2) holds. If (2) holds, then $[L] \dashrightarrow L$ and therefore $[L] \dashrightarrow [K]$ so (3) holds. Now assume (3) holds and pick $A \in K$. Since $[K] \dashrightarrow K$, there is $B \in [K]$ such that $B \dashrightarrow A$. Since $[L] \dashrightarrow [K]$, there is $C \in [L]$ such that $C \dashrightarrow B \dashrightarrow A$. Since $[L] \subseteq L$, $C \in L$. Since $F$ is closed under composition, $C \dashrightarrow A$ and since $L$ is closed under $\dashrightarrow$, $A \in L$. This shows that (1) holds. □

We want to be able to speak of a set of instances associated with a query $Q$. Therefore, given a query $Q$, we define a sentence $\hat{Q}$ obtained from $Q$ by replacing the free variables $\bar{x}$ of $Q$ with new constants $\bar{c}$. Notice that with this definition, we have by Theorem 12:

1. If $Q \in \mathrm{UCQ}$, then $\hat{Q}$ is closed under homomorphisms.
2. If $Q \in \mathrm{MonQ}$, then $\hat{Q}$ is closed under inj. homomorphisms.
3. If $Q \in \mathrm{UCQ}^\neg$, then $\hat{Q}$ is closed under full homomorphisms.
4. If $Q \in \mathrm{UCQ}^{\neg,\neq}$, then $\hat{Q}$ is closed under embeddings.

To simplify the notation, we write $[Q]$ instead of $[\hat{Q}]$. If $Q \in \mathrm{CQ}$, then $[Q]$ is precisely what is known as the *frozen instance* of $Q$. Therefore, for the case of queries, templates generalize the notion of frozen instances. Furthermore, we have the following natural generalization of what is known as *the homomorphism theorem* for conjunctive queries [1]:

COROLLARY 4 (QUERY CONTAINMENT). *If $P, Q \in \mathcal{L}$ and $\mathcal{L}$ is closed under the mappings $F$, then*

$$P \sqsubseteq Q \text{ iff } [Q] \to [P].$$

The following important result follows from Theorems 15 and 6.

THEOREM 16 (IMPLICATION). *If $P, Q \in \mathcal{L}$, $\mathcal{L}$ is closed under the mappings $F$, and $\Sigma$ is a set of sentences, then the following are equivalent whenever all templates mentioned exist:*

1. $P \sqsubseteq_\Sigma Q$.
2. $\Sigma \models \forall\bar{x}(P \to Q)$.
3. $\Sigma, \hat{P} \models \hat{Q}$.
4. $\mathrm{mod}(\Sigma \cup \{\hat{P}\}) \subseteq \mathrm{mod}(\hat{Q})$.
5. $[Q] \to [\Sigma \cup \{\hat{P}\}]$

*Furthermore, if some $\Sigma$-chase terminates on $P$, then 1-5 are also equivalent to:*

6. $P^\Sigma \sqsubseteq Q$

Partial results similar to Theorems 15 and 16, except for the mention of templates, are known for several special cases including conjunctive queries and embedded dependencies. Our contribution is identifying the crucial roles of templates in them and therefore providing a uniform generalization to any kinds of constraints, mappings, and queries closed under such mappings for which we can find templates.

Notice also that in some cases, we do not need to chase to obtain a template, but can simply asserts its existence to obtain the desired result. In particular, we can then 'use' a weak template, which we know can not be found by chasing. An example of a result of this kind is the following generalization of Theorem 5 in [4]:

THEOREM 17. *For any set $\Sigma$ of universal-existential constraints, if there is some $c$ such that for any $P, Q \in \mathrm{UCQ}^{\neg,\neq}$, a weak template $T = [\Sigma \cup \{\hat{P}\}]$ exists and every instance $A$ in it satisfies $|A| \leq |P|^c$, then we can check whether $P \sqsubseteq_\Sigma Q$ holds in $\mathbf{\Pi_2^P}$.*

PROOF. Assume the hypotheses. Now for every $A$ satisfying $|A| \leq |P|^c$, $A \models \Sigma$, and $A \models \hat{P}$, check whether $A \models \hat{Q}$. If this holds, then we must have $T \subseteq \mathrm{mod}(\hat{Q})$, and therefore $P \sqsubseteq_\Sigma Q$ by Theorem 16. Otherwise, we have a counterexample. □

# 9. QUERIES WITH $K$-VARIABLES

As we have seen in Theorem 5, there are sets of TGDs with no templates and, more specifically, there are data integration settings which have no universal set solutions as Example 3 shows. Can we still compute certain answers in such situations? We show that in many situations we can, by relaxing the notion of universal set solution as follows. We write $A \xrightarrow{k} B$ if

$$(\forall A' \sqsubseteq A)(|A'| \leq k \to A' \to B).$$

That is, every restriction of $A$ to at most $k$ elements has a homomorphism into $B$. This notion can easily be generalized to other kinds of mappings and it also applies to checking containment.

If $A \xrightarrow{k} B$, we say that there is a $k$-homomorphism from $A$ to $B$, even though this fact is not usually witnessed by a single mapping. Nevertheless, the binary relation given by $\xrightarrow{k}$ is reflexive and transitive, and therefore $\xrightarrow{k}$ gives a preordering. It turns out this is all we need to define templates in the most general way: we simply replace universality in the definition of template with universality under some preordering. Here we define $k$-templates for homomorphisms; a similar definition applies to other mappings.

DEFINITION 13. A set of finite structures $T$ is a *$k$-template* for a set of structures $K$ if it satisfies the following conditions:

1. ($k$-universality) $T \xrightarrow{k} K$,
2. (conformance) $T \subseteq K$,
3. (finiteness) $T$ is finite, and
4. (minimality) there is no $T' \subset T$ such that $T' \to T$.

We define $k$-universal solutions similarly, using $k$-universality instead of plain universality. It turns out that $k$-universality is all we need to answer existential queries with $k$ variables (we assume variables are not reused). More precisely, $\mathrm{CQ}_k$ is the set of conjunctive queries with at most $k$ variables (the existential quantification has been pushed out). and $\mathrm{UCQ}_k$ are unions of $\mathrm{CQ}_k$ queries. We define the other families of existential queries in a similar way.

THEOREM 18.

1. *$UCQ_k$ is closed under $k$-homomorphisms.*
2. *$\mathrm{MonQ}_k$ is closed under injective $k$-homomorphisms.*
3. *$\mathrm{UCQ}_k^{\neg}$ is closed under full $k$-homomorphisms.*
4. *$\mathrm{UCQ}_k^{\neg,\neq}$ is closed under $k$-embeddings.*

*That is, for any of these classes of queries and the corresponding class of mappings,*

$$h : A \overset{k}{\dashrightarrow} B \text{ and } \bar{a} \in Q(A) \text{ implies } h(\bar{a}) \in Q(B).$$

PROOF. See the appendix. □

THEOREM 19. *If $W$ is a $k$-universal set solution for $S$ under $\Sigma$ for*

1. *homomorphisms and $Q \in \mathrm{CQ}_k$,*
2. *injective homomorphisms and $Q \in \mathrm{MonQ}_k$,*
3. *full homomorphisms and $Q \in \mathrm{UCQ}_k^{\neg}$, or*
4. *embeddings and $Q \in \mathrm{UCQ}_k^{\neg,\neq}$,*

*and $Q$ has arity $r$, then*

$$\mathrm{cert}_Q^{\Sigma}(S) = \mathrm{dom}(S)^r \cap \bigcap_{T \in W} Q(T).$$

PROOF. (sketch) Similar to the proof of Theorems 1 and 2 using Theorem 18 instead of Theorem 12. □

THEOREM 20. *For every $k$, there is a set $\Sigma_{k+1}$ of TGDs such that $\Sigma$ has a weak template and a strong $k$-template, but no strong $(k+1)$-template*

PROOF. (sketch) Consider the set of axioms $\Sigma_{k+1}$:

$$\begin{array}{lllll} \xi_1\text{:} & & & & \exists x, y\, E(x,y) \\ \xi_2\text{:} & E(x,y) & & \rightarrow & \exists z\, E(y,z) \\ \xi_3^{k+2}\text{:} & E(u_0,u_1), \ldots, E(u_{k+1},u_{k+2}) & & \rightarrow & E(u_0,u_{k+2}) \end{array}$$

$\{C^{k+1}\}$ is a weak template for $\Sigma_{k+1}$ and also a strong $k$-template. If $G \models \Sigma_{k+1}$, then $G$ must contain a cycle of length at most $k+1$, so it can not be a strong $(k+1)$-template. □

To compute $k$-templates for $\Sigma$ when $\Sigma$ is closed under products, we can use the following variant of the chase. Compute a chase sequence $\emptyset = A_0, A_1, \ldots$ using some chase-like algorithm, for example the unordered minimizing chase. Simulatenously, compute a chain of models of $\Sigma$: $B_0 \leftarrow B_1 \leftarrow \ldots$. Such a chain can be obtained, for example, by picking some enumeration of the models of $\Sigma$ and setting $B_n$ to be the product of the first $n+1$ models. Alternatively, given $B_n$, we can set $B_{n+1} := M_{n+1}$ were $M_{n+1}$ is the $(n+1)$th model in this enumeration in case $M_{n+1} \rightarrow B_{n+1}$ and $B_{n+1} := B_n$ otherwise. If we find some $n, m$ such that $B_m \overset{k}{\rightarrow} A_n$ we stop. The result is $B_m$, which has the desired universality property since $B_m \overset{k}{\rightarrow} A_n \rightarrow \mathrm{mod}(\Sigma)$ by Theorem 6. This can be generalized to universal-existential constraints by considering the situation where each $A_i$ and $B_j$ is a finite set of instances instead of a single instance. We can show the following.

THEOREM 21. *If there is a strong $k$-template for $\Sigma$, then the procedure above using the unordered minimizing chase will terminate and find it.*

# 10. CONCLUSION

We have introduced universal set solutions and their more general counterparts, templates, and we have shown how to use them to compute certain answers. We have shown that any chase-like algorithm produces strong templates and we have presented a new chase, the unordered minimizing chase, which will always find a strong template if there is one. We did not explore the complexity of this new chase. Finding the core of a general structure is NP-complete. However, we hope that the techniques for computing cores of chase results in [9, 11] can be applied to give improvements in efficiency.

We have also introduced new conditions for chase termination, wider than those previously known. We have shown how to use templates to check containment and containment under constraints. Finally, we have shown that we can relax the notions of universal set solutions and templates to $k$-universal set solutions and $k$-templates and have shown that these more relaxed notions are still sufficient to compute certain answers to queries with at most $k$ variables and for testing containment of queries with at most $k$ variables.

We have also shown that the finite and unrestricted versions of templates differ. We have an interesting situation: since we only care about finite instances, we really need weak templates, but all chase-like algorithms produce only strong templates. It is natural to ask whether there are algorithms that can compute weak templates. This remains a hard open problem, connected with the fundamental differences between finite model theory and unrestricted model theory. We would like to close with an intriguing connection.

Given a set of instances $K$, set $\bar{K} := \{B : \exists A \in K, A \rightarrow B\}$. Whether we consider only finite instances or unrestricted instances should be clear from context. The following result follows from the infinite and finite versions of the "preservation under homomorphisms" theorems, the latter recently proved by Benjamin Rossman [15].

THEOREM 22.

1. *$\Sigma$ has a strong template iff $\bar{K}$ is axiomatizable by a first-order sentence, where $K$ is the set of unrestricted models of $\Sigma$.*
2. *$\Sigma$ has a weak template iff $\bar{K}$ is axiomatizable by a first-order sentence where $K$ is the set of finite models of $\Sigma$.*

# 11. REFERENCES


[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison Wesley, 1995.

[2] A. V. Aho, C. Beeri, and J. D. Ullman. The theory of joins in relational databases. *ACM Trans. Database Syst.*, 4(3):297–314, 1979.

[3] C. Beeri and M. Y. Vardi. A proof procedure for data dependencies. *J. ACM*, 31(4):718–741, 1984.

[4] A. Deutsch, B. Ludaescher, and A. Nash. Rewriting queries using views with access patterns under integrity constraints. In *ICDT*, 2005.

[5] A. Deutsch and V. Tannen. Mars: A system for publishing xml from mixed and redundant storage. In *VLDB*, pages 201–212, 2003.

[6] A. Deutsch and V. Tannen. Reformulation of XML Queries and Constraints. In *ICDT*, 2003.

[7] R. Fagin. Horn clauses and database dependencies. *J. ACM*, 29(4):952–985, 1982.

[8] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data Exchange: Semantics and Query Answering. ICDT 2003, full version in: Theor. Comput. Sci. 336(1): 89-124 (2005).

[9] R. Fagin, P. G. Kolaitis, and L. Popa. Data Exchange: Getting to the Core. In *ACM PODS*, pages 90–101, 2003. Full version in ACM TODS, 30(1):147-210(2005).

[10] A. Fuxman, P. G. Kolaitis, R. J. Miller, and W. C. Tan. Peer data exchange. In *PODS*, 2005.
[11] G. Gottlob and A. Nash. Data exchange: Computing cores in polynomial time. Manuscript, Submitted for publication, 2005.
[12] A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov. Schema Mediation in Peer Data Management Systems. ICDE 2003.
[13] Maier, Sagiv, and Yannakakis. On the complexity of testing implication of functional and join dependencies. *J. ACM*, 1981.
[14] D. Maier, A. O. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Trans. Database Syst.*, 4(4):455–469, 1979.
[15] B. Rossman. Existential positive types and preservation under homomorphisisms. In *LICS*, pages 467–476, 2005.

# APPENDIX

# A. ADDITIONAL PROOFS

PROOF. (**Theorem 6**)

1. Follows from the definition of $\stackrel{\xi,\bar{a}}{\rightarrow}$.
2. Follows from 5 below.
3. Follows immediately from the definition of $A^{\Sigma}$.
4. If $A^{\Sigma}_{\omega} \not\models \Sigma$, then for some $\bar{a}$ and $\xi \in \Sigma$, we must have $A^{\Sigma}_{\omega} \models \exists P_{\xi}(\bar{a})$ and $A^{\Sigma}_{\omega} \not\models \exists C_{\xi}(\bar{a})$. But then this must also hold for $A_s$ for some $s$, the former because the range of any homomorphism $h : \exists P_{\xi} \rightarrow A^{\Sigma}_{\omega}\bar{a}$ is finite and the latter by monotonicity of $\exists C_{\xi}$. But then, by definition of fairness, there must be $A_r$ for some $r > s$ such that $A_r \models \exists C_{\xi}(\bar{a})$, contradicting $A^{\Sigma}_{\omega} \not\models \exists C_{\xi}(\bar{a})$.
5. Assume $A_n \stackrel{\xi,\bar{a}}{\rightarrow} A_{n+1}$. Then since $A_n \models \exists P_{\xi}(\bar{a})$, $B \models \exists P_{\xi}(h(\bar{a}))$. Therefore, there is $\bar{b}$ (possibly empty) such that $B \models C_{\xi}(h(\bar{a}), \bar{b})$. If $\xi$ is a TGD, then we can map $C_{\xi}$ to $C_{\xi}(h(\bar{a}), \bar{b})$ to get the desired extension $h'$. If $\xi$ is an EGD, then $h$ must map the equated values to the same value in $B$, so the restriction $h|\mathrm{dom}(A_{n+1})$ is also a homomorphism.
6. Follows from 5 above.

□

PROOF. (**Theorem 7**, sketch) Part 2 follows from Theorem 6. Therefore, if the unordered minimizing chase terminates, then there is a strong template for $\Sigma$, namely the result of the chase.

For the converse of part 1, assume first that $\Sigma$ consists only of TGDs and assume there is a strong template $\{T\}$ for $\Sigma$ and there is no finite unordered chase with minimization sequence starting with $\emptyset$. Then there must be an infinite unordered chase sequence starting with $\emptyset$: $\emptyset = A_0, A_1, A_2, \ldots$. Set $A^{\Sigma}_{\omega} = \bigcup_i A_i$, which is well defined because for all $i$, $A_i \subseteq A_{i+1}$. Since $\{T\}$ is a strong template for $\Sigma$ and $A^{\Sigma}_{\omega} \rightarrow T$ by Theorem 6, $T \rightarrow A^{\Sigma}_{\omega}$. Since $T$ is finite, $T \rightarrow A_n$ for some $n$ and, by Theorem 6, $A_n \rightarrow T$. But then $\mathrm{core}(T)$ and $\mathrm{core}(A_n)$ are isomorphic and therefore both satisfy $\Sigma$. Now consider the unordered chase with minimization sequence starting with $\emptyset$: $A_0, A_1, \ldots$. It is easy to verify by induction that for every $s$, $A_s = \mathrm{core}(A_s)$. In particular, $A^{\Sigma}_n = \mathrm{core}(A_n)$ and therefore $A^{\Sigma}_n \models \Sigma$ and this sequence is finite.

If $\Sigma$ consists of EGDs and TGDs, we can simulate the EGDs with TGDs to obtain $\bar{\Sigma}$ as explained in [11]. Then also $\mathrm{core}(T)$ and $\mathrm{core}(A_n)$ are isomorphic for some $n$ as above and the rest of the argument goes through unchanged. □

PROOF. (**Theorem 10**) Assume $\alpha \prec \beta$. Then, by the definition, there are $A, B, \bar{a}, \bar{b}$ satisfying conditions 1, 2, and 3 of the definition. In particular, there is a homomorphism $h : \exists P_{\beta} \rightarrow B\bar{b}$. Set $B'$ to be a minimal instance such that $h(B) \subseteq B'$ and such that there is $A' \subseteq B'$ satisfying $A' \stackrel{\alpha,\bar{a}}{\rightarrow} B'$. Then $A', B', \bar{a}, \bar{b}$ also satisfies condition 2 of the definition by construction, and conditions 1 by monotonicity of $P_{\beta}$ and condition 3 by monotonicity of $C_{\beta}$ together with $h(B) \subseteq B'$. Furthermore, such $B'$ must satisfy $|B'| \leq |C_{\alpha}| + |P_{\beta}|$ so we only need to examine a finite set of candidates $A', B'$. In fact, it is enough to consider unions of homomorphic images of $P_{\beta}$ with $C_{\alpha}$ as candidates for $B$ and remove from them an induced substructure isomorphic to $C_{\alpha}$ to get candidates for $A$. □

PROOF. (**Theorem 12**, sketch)

1. Assume that $h : A \rightarrow B$ is a homomorphism, $\bar{a} \in Q(A)$, $Q \in \mathrm{UCQ}$, and $Q := \bigvee_{1 \leq i \leq k} Q_i$ where each $Q_i \in \mathrm{CQ}$. Then $\bar{a} \in Q_i(A)$ for some $i$ and this happens iff there is a homomorphism $g : Q_i \rightarrow A\bar{a}$, that is a homomorphism from $Q_i$ to $A$ which maps the free variables $\bar{x}$ of $Q_i$ to $\bar{a}$. But then $h \circ g : Q_i \rightarrow Bh(\bar{a})$ and therefore $h(\bar{a}) \in Q(B)$.
2. Assume that $h : A \rightarrow B$ is an injective homomorphism, $\bar{a} \in Q(A)$, and $Q \in \mathrm{MonQ}$. Set $A' = h(A)$. That is, for every relation in $A$, set $R^{A'} = h(R^A)$. Then $A' \subseteq B$ and $h$ is an isomorphism between $A$ and $A'$. Therefore, by genericity, $h(\bar{a}) \in Q(A')$ and by monotonicity, $h(\bar{a}) \in Q(A)$.
3. Simlar to part 1. If $\bar{a} \in Q_i(A)$ then there is a homomorphism $g : Q_i \rightarrow A\bar{a}$ which also preserves the absence of some tuples. Composing $h$ with $g$ gives a homomorphism which preserves the absence of those tuples.
4. Similar to part 3, but with embeddings.

□

PROOF. (**Theorem 14**) Set $\hat{\sigma}$ and $\Lambda$ as in the proof of Theorem 13 and set $\hat{\Sigma} := \Sigma \cup \Lambda$.

1 implies 3: Assume there is a strong template $T$ for $\Sigma$. Set $\hat{T}$ to consist of all expansions of all homomorphic images of instances in $T$ to $\hat{\sigma}$ that satisfy $\Lambda$. Then $\hat{T}$ satisfies all conditions for a template except minimality as follows. Conformance and finiteness are obvious. If $\hat{B} \models \hat{\Sigma}$, then its reduction $B$ to $\sigma$ satisfies $\Sigma$ and therefore there is $A \in T$ and a homomorphism $h : A \rightarrow B$. Then there is $A' = h(A) \in T$ such that $g : A' \rightarrow B$ is injective.

3 implies 2: Assume there is a strong template $\hat{T}$ for $\hat{\Sigma}$. Then the set $T$ of structures in $\hat{T}$ reduced to the signature $\sigma$ all conditions for an $F$-template except minimality as follows. Conformance and finitiness are obvious. Universality is satisfied by Theorem 13, since if $B \models \Sigma$, there is an expansion $\hat{B}$ of $B$ such that $\hat{B} \models \hat{\Sigma}$. Then there is $\hat{A} \in \hat{T}$ such that $\hat{A} \rightarrow \hat{B}$ and therefore the reduction $A$ of $\hat{A}$ to $\sigma$ satisfies $A \dashrightarrow B$. It is easy to obtain $T' \subseteq T$ which satisfies minimality as well.

2 implies 1: Assume there is a strong $F$-template $T$ for $\Sigma$. Since every $F$-mapping is a homomorphism, $T$ satisfies all conditions for a template, except for minimality. It is easy to obtain $T' \subseteq T$ which satisfies minimality as well.

3 iff 4: This follows from Theorem 7.

The proof above shows how to obtain each of the strong templates from the others. □

PROOF. (**Theorem 18**) (1) Assume that $A \stackrel{k}{\rightarrow} B$, $\bar{a} \in Q(A)$, $Q \in \mathrm{UCQ}$, and $Q := \bigvee_{1 \leq i \leq k} Q_i$ where each $Q_i \in \mathrm{CQ}$. Then $\bar{a} \in Q_i(A)$ for some $i$ and this happens iff there is a homomorphism $g : Q_i \rightarrow A\bar{a}$, that is a homomorphism from $Q_i$ to $A$ which maps the free variables $\bar{x}$ of $Q_i$ to $\bar{a}$. Since $|g(Q_i)| \leq k$, we have $A' \sqsubseteq A$ and homomorphism $h : A' \rightarrow B$. But then $h \circ g : Q_i \rightarrow Bh(\bar{a})$ and therefore $h(\bar{a}) \in Q(B)$. The proof of 2, 3, and 4 is similar to the corresponding parts in Theorem 18. □