Computing Universal Models Under Guarded TGDs

André Hernich
Humboldt-Universität zu Berlin, Germany
hernich@informatik.hu-berlin.de

ABSTRACT

A universal model of a database D and a set Σ of integrity constraints is a database that extends D, satisfies Σ, and is most general in the sense that it contains sound and complete information. Universal models have a number of applications including answering conjunctive queries, and deciding containment of conjunctive queries, with respect to databases with integrity constraints. Furthermore, they are used in slightly modified form as solutions in data exchange. In general, it is undecidable whether a database possesses a universal model, but in the past few years researchers identified various settings where this problem is decidable, and even efficiently solvable.

This paper focuses on computing universal models under finite sets of guarded TGDs, non-conflicting keys, and negative constraints. Such constraints generalize inclusion dependencies, and were recently shown to be expressive enough to capture certain members of the DL-Lite family of description logics. The main result is an algorithm that, given a database without null values and a finite set Σ of such constraints, decides whether there is a universal model, and if so, outputs such a model. If Σ is fixed, the algorithm runs in polynomial time. The algorithm can be extended to cope with databases containing nulls; however, in this case, polynomial running time can be guaranteed only for databases with bounded block size.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computation on discrete structures*; H.2.4 [**Database Management**]: Systems—*Relational databases, rule-based databases, query processing*

General Terms

Algorithms, Languages, Theory

Keywords

data exchange, chase, core, guarded Datalog +/-

ICDT 2012 March 26–30, 2012, Berlin, Germany

1. INTRODUCTION

Since its introduction in the late seventies, the chase procedure [1, 33, 4] has become an indispensable tool with numerous applications in database theory. Initially, it was developed to decide the implication problem for various types of data dependencies (see, e.g., [1, 33, 4]). Then researchers realized that the chase is not only useful to solve the implication problem, but also a number of other problems including containment of conjunctive queries in the presence of data dependencies [28]; answering conjunctive queries on databases with data dependencies [28], in data integration [31], and over ontologies [7]; and computing universal solutions in data exchange [17].

The chase is a procedure that "repairs" a given database instance I so that the resulting database satisfies a set Σ of constraints. Typical constraints the chase is able to deal with are tuple-generating dependencies (TGDs) and equality-generating dependencies (EGDs) [4] (which together capture most of the data dependencies in the literature, including inclusion dependencies, functional dependencies, join dependencies, and multivalued dependencies). TGDs enforce the presence of certain tuples based on certain other tuples in the database, while EGDs assert that certain values that occur in certain tuples are equal. Given I and Σ as input, the chase adds tuples to I and identifies values as enforced by the TGDs and EGDs in Σ in order to arrive at a *model of I and Σ*, that is, a database $J \supseteq I$ such that $J \models \Sigma$ (if I contains nulls, then $J \supseteq I$ must be replaced by the existence of a homomorphism from I to J). If this procedure terminates, it yields not only a model J of I and Σ. The distinguishing feature of the chase is that this model J is *universal* in the sense that it admits homomorphisms into all models of I and Σ [4, 17, 15].

So, a *universal model* of an database I and a set Σ of constraints is a finite model J of I and Σ that admits homomorphisms into all models K of I and Σ [15]. It is important here that J itself is finite, while the models K might well be infinite. If we require only homomorphisms into finite models, as is usually done in data exchange [17], J is called *weak universal model*. Universal models (strong or weak) may exist, even if the chase does not terminate.

Having access to an arbitrary universal model J of I and Σ is often enough in situations where, traditionally, the chase procedure was used (see, e.g., [15]).

Example 1.1. We consider the problem of answering a Boolean conjunctive query q on I with respect to Σ. This problem asks whether q is true in all models of I and Σ, which is denoted by $I \cup \Sigma \models q$. If J is a universal model of

I and Σ, then $I \cup \Sigma \models q$ is equivalent to $J \models q$. Indeed, if $J \models q$, then for all models K of I and Σ we have $K \models q$, since there is a homomorphism from J to any such model K, and conjunctive queries are preserved under homomorphisms. Therefore, $I \cup \Sigma \models q$. On the other hand, if we have $I \cup \Sigma \models q$, then, clearly, $J \models q$.

It should be mentioned, however, that the chase procedure cannot always be replaced by a universal model. For example, there are efficient algorithms for evaluating conjunctive queries on the possibly infinite result of the chase procedure if the set of constraints has certain structural properties [28, 11, 5, 6, 7, 9, 8] (see also Section 3). The main idea underlying these algorithms is that, in order to evaluate a Boolean conjunctive query on the possibly infinite chase result, it suffices to run the chase procedure for a finite number of steps (which can be determined from the set of constraints), and to evaluate the query on the resulting database (which might turn out to be not a model). A universal model might still be of interest in such a case. If it is possible to compute a universal model, then one could evaluate conjunctive queries directly on that model, without recomputing the finite initial portion of the chase result each time; if this is not possible, then one could use the techniques proposed in the above-cited papers.

A slight variation of the concept of a universal model is even an essential part of the theory of data exchange [17]. Data exchange is the problem of translating databases from a source schema into a target schema, whereby providing access to the source database through a materialized database over the target schema. Formally, we are given a source schema σ, a target schema τ, a source database I over σ, and a set Σ of constraints over the union of the two schemas σ and τ that describes the relationship between source and target. Typically, the constraints in Σ are particular TGDs and EGDs—source-to-target TGDs which enforce certain tuples in the target if certain other tuples in the source are present, and target TGDs and target EGDs which are TGDs and EGDs expressed over the target schema. The goal is to compute a *solution* for I under Σ, which is a finite database J over τ such that the union $I \cup J$ of the two databases I and J is a model of I and Σ. A *universal solution* for I under Σ is a solution that admits homomorphisms into all solutions for I under Σ. In [17], the case was made that universal solutions have many properties that make them the preferable solutions in data exchange. For the settings typically considered in data exchange, there is a tight relationship between universal solutions and *weak* universal models: a solution J for I under Σ is a universal solution if and only if $I \cup J$ is a weak universal model of I and Σ.

In this light, an important problem is to decide whether a database and a set of constraints admits a strong (resp., weak) universal model, and if so, to compute one. Unfortunately, it is undecidable whether a strong (resp., weak) universal model of a database and a set of TGDs and EGDs exists. This is even true if we restrict attention to some fixed finite set of TGDs [27] (see also [26, Section 2.3] for a proof tailored directly for universal models). On the other hand, there is a long line of research on finding more and more general structural properties of sets Σ of TGDs and EGDs such that for all databases I, the chase terminates on input I and Σ [4, 14, 17, 15, 35, 29, 30, 22, 19]. For all these properties, it is possible to decide whether there is a universal model of I and Σ, and if so, to compute one, in time $O(n^k)$, where n is the size of I and k depends only on Σ. Thus, if Σ is fixed, these properties guarantee polynomial-time algorithms for computing universal models.

There are still important classes $\mathcal{C}$ of constraints such that given $\Sigma \subseteq \mathcal{C}$, the chase may not terminate for all databases I and Σ, hence Σ does not exhibit any of the above structural properties. For example, this is true for the class of inclusion dependencies (see, e.g., Section 2.4). According to [24], together with functional dependencies, inclusion dependencies are the most widely used integrity constraints in practice. Recently, it was shown that sets of linear TGDs (which includes sets of inclusion dependencies), together with certain other constraints, called negative constraints and non-conflicting keys (a definition will be given in Section 6.1), are expressive enough to capture two members of the DL-Lite family of description logics [7]. Furthermore, [7] shows that it is possible to evaluate conjunctive queries on the possibly infinite chase result in time $O(n^k)$, where n is the size of I and k depends only on Σ and the query. This result holds for other, more expressive sets of constraints like sets of guarded TGDs or sticky TGDs together with negative constraints and non-conflicting keys [9, 8].

Results. In this paper, we study the complexity of the following problem: Given a database I and a finite set Σ of guarded TGDs (and possibly other constraints like negative constraints and non-conflicting keys), decide whether there is a universal model of I and Σ, and if so, compute such a model. We focus mainly on the *data complexity*, which measures the complexity as a function of the size of I.

The main result (Theorem 5.1) is that the following problem can be solved in time $O(n^k)$, where n is the size of I, and k depends only on Σ:

Input: a database instance I without null values, and a finite set Σ of guarded TGDs

Task: Decide whether there a universal model of I and Σ. If so, compute a core model of I and Σ.

Here, a core model is the *core of the universal models* introduced in [18], which, informally, is the smallest universal model of I and Σ.

We generalize the main result to more general sets of constraints, and to databases with nulls. On the one hand, we show that it is not problematic if Σ additionally contains negative constraints and non-conflicting keys (Theorem 6.5). On the other hand, we show that databases with nulls can be handled, at the price of an increased complexity: the problem becomes NP-hard. However, if we restrict our attention to databases with bounded block size, which typically arise in data exchange as the result of "applying" the source-to-target TGDs, the problem can still be solved in time $O(n^k)$, where k now depends both on Σ and on the maximum number of nulls in a block of the input database.

As an additional result, we show that for the sets Σ of constraints considered in the above results, strong universal models and weak universal models coincide, so that the algorithms can be used both to decide the existence of strong universal models as well as weak universal models, and to compute such a model if it exists (Proposition 7.2).

Organization. The paper is structured as follows. Section 2 presents basic notation and results. Section 3 gives a brief overview of related work on computing universal models. Section 4 introduces an important technical tool: guarded chase forests. Using guarded chase forests, we then prove in Section 5 that for databases I without null values, and finite sets Σ of guarded TGDs, universal models can be computed in time $O(n^k)$, where n is the size of I and k depends only on Σ. This result is extended in Section 6 to more general sets of constraints, and to databases with nulls. Furthermore, in Section 7 we show that strong and weak universal models coincide under finite sets of guarded TGDs.

2. BASICS

This section gives basic notation and results needed throughout the paper. We let $[m, n]$ be the set of all integers p with $m \leq p \leq n$, and we define $[n] := [1, n]$. Mappings $f\colon A \to B$ are extended to tuples $\bar{a} = (a_1, \dots, a_k)$ over A via $f(\bar{a}) := (f(a_1), \dots, f(a_k))$.

2.1 Databases

A *schema* is a finite set σ of relation symbols R, where each $R \in \sigma$ has an arity $\mathrm{ar}(R) \geq 1$. A *σ-instance* I maps each $R \in \sigma$ to a relation R^I of arity $\mathrm{ar}(R)$. Instances are finite if not indicated otherwise. The *active domain* of I (i.e., the set of all values that occur in I) is denoted by $\mathrm{dom}(I)$. We assume that $\mathrm{dom}(I) \subseteq \mathit{Dom}$, where Dom is the union of two fixed disjoint infinite sets—the set *Const* of all *constants*, and the set *Null* of all *(labeled) nulls*. Constants are denoted by letters c, d, e and variants like c', c_1. Nulls serve as placeholders, or variables, for unknown constants, and we will denote them by $\perp$ and variants like $\perp', \perp_1$. Let $\mathrm{const}(I) := \mathrm{dom}(I) \cap \mathit{Const}$ and $\mathrm{nulls}(I) := \mathrm{dom}(I) \cap \mathit{Null}$. A *ground* instance is an instance without nulls. The *size* of an instance I is $\|I\| := \sum_{R \in \sigma} \mathrm{ar}(R) \cdot |R^I|$.

An *atom* is an expression of the form $R(\bar{a})$, where R is a relation symbol, and $\bar{a} \in \mathit{Dom}^{\mathrm{ar}(R)}$. We often view an instance I as the set of all atoms $R(\bar{a})$ with $\bar{a} \in R^I$. This enables us to apply set theoretic notation to instances. For example, we write $I \cup J$, $I \cap J$, and $I \setminus J$ for the union, intersection, and difference of two instances I and J, and $I \subseteq J$ if I is a subinstance of J. Given a mapping $f\colon \mathit{Dom} \to \mathit{Dom}$, we let $f(I)$ be the instance $\{R(f(\bar{a})) \mid R(\bar{a}) \in I\}$.

Let I and J be σ-instances. A *homomorphism* from I to J is a mapping $h\colon \mathrm{dom}(I) \to \mathrm{dom}(J)$ such that $h(I) \subseteq J$, and $h(c) = c$ for all $c \in \mathrm{const}(I)$. We write $I \to J$ if there is a homomorphism from I to J. If $I \to J$ and $J \to I$, we call I and J *homomorphically equivalent.* An *isomorphism* from I to J is a bijective homomorphism h from I to J such that h^{-1} is a homomorphism from J to I. If there is an isomorphism from I to J, we say that I and J are *isomorphic*, and denote this by $I \cong J$. An instance $J \subseteq I$ is a *core* of I if $I \to J$ and $I \not\to K$ for every $K \subsetneq J$. Each finite instance has a core, and cores of homomorphically equivalent instances are isomorphic [25]. In particular, every two cores of an instance are isomorphic.

For a σ-instance I, let q_I be the *canonical query* of I. That is, fix an enumeration $\perp_1, \dots, \perp_k$ of all the nulls in I, and an enumeration $R_1(\bar{u}_1), \dots, R_n(\bar{u}_n)$ of all the atoms in I. Then q_I is the Boolean conjunctive query $\exists x_1 \cdots \exists x_k \bigwedge_{i=1}^{n} R_i(\bar{v}_i)$, where each $\bar{v}_i$ is obtained from $\bar{u}_i$ by replacing each occurrence of a null $\perp_j$ in $\bar{u}_i$ with x_j. There is a tight connection between canonical queries and homomorphisms, first observed by Chandra and Merlin [13]: $J \models q_I$ iff $I \to J$.

2.2 Constraints

We write $\varphi(x_1, \dots, x_k)$ to denote a formula φ with free variables $\{x_1, \dots, x_k\}$. Given such a formula $\varphi(x_1, \dots, x_k)$, values $a_1, \dots, a_k \in \mathit{Dom}$, and an instance I, we write $I \models \varphi(a_1, \dots, a_k)$ if φ is satisfied in I under the assignment mapping x_i to a_i for every $i \in [k]$. By referring to the *atoms of* $\varphi(a_1, \dots, a_k)$, where φ is a conjunction of relational atomic formulas, we mean the set of all atoms obtained from an atomic formula in φ by replacing each x_i with a_i. Let $\mathrm{dom}(\varphi)$ be the set of all constants that occur in φ.

As constraints we consider *tuple-generating dependencies (TGDs)* and *equality-generating dependencies (EGDs)* [4]. A TGD over σ is an FO-sentence $\theta = \forall\bar{x}\forall\bar{y}\,(\varphi(\bar{x}, \bar{y}) \to \exists\bar{z}\,\psi(\bar{x}, \bar{z}))$ over σ, where φ and ψ are conjunctions of relational atomic formulas. We call $\mathrm{body}(\theta) = \varphi$ the *body* of θ, and $\mathrm{head}(\theta) = \psi$ its *head.* A tgd θ is *guarded* if there is an atom in $\mathrm{body}(\theta)$, called *guard*, that contains all variables that occur in $\mathrm{body}(\theta)$ [6]. A *linear TGD* (also known as *local-as-view TGD*) is a TGD that contains precisely one atom in its body [6]. An *inclusion dependency (ID)* is a TGD of the form $\forall\bar{x}\forall\bar{y}\,(R(\bar{u}) \to \exists\bar{z}\,S(\bar{v}))$ such that no variable occurs twice in $\bar{u}$ or $\bar{v}$. Note that IDs are linear, and that linear TGDs are guarded. An *equality-generating dependency (EGD)* over σ is an FO-sentence over σ of the form $\forall\bar{x}\,(\varphi(\bar{x}) \to y = z)$, where φ is a conjunction of relational atomic formulas, and y, z are variables in $\bar{x}$.

To simplify the presentation, we will assume that TGDs do not contain any constants. It is not hard to adapt this paper's results to TGDs with constants.

2.3 Universal Models

Given a σ-instance I and a set Σ of constraints over σ, a *model of I and Σ* is a possibly infinite σ-instance J such that $I \to J$ and $J \models \Sigma$. Note that $I \to J$ boils down to $I \subseteq J$ if I is ground.

A *universal model* of I and Σ is a finite model J of I and Σ such that for all models K of I and Σ we have $J \to K$ [15]. It is important here that J is *finite*, but it must admit homomorphisms into all models, including infinite ones. If J is not required to be finite, we call J *quasi-universal* model of I and Σ. Universal models are sometimes called *strong universal models.* Certain applications (e.g., data exchange) require *weak universal models*, which are only required to admit homomorphisms into all finite models. More precisely, a weak universal model of I and Σ is a finite model J of I and Σ such that for all finite models K of I and Σ we have $J \to K$ [15]. Every (strong) universal model of I and Σ is a weak universal model of I and Σ, but not necessarily vice versa [15]. We shall see in Section 7 that the converse is true if Σ is a finite set of guarded TGDs.

In general, there may be no universal model (strong or weak) for I and Σ, even when Σ is a finite set of IDs.

Example 2.1. Consider $\theta := \forall x\forall y(E(x, y) \to \exists z\,E(y, z))$ and the instance $I := \{E(c, d)\}$, where c and d are distinct constants. It is not hard to see that there is no weak universal model (and hence no strong one) of I and $\Sigma := \{\theta\}$.

Suppose, to the contrary, that J is a weak universal model of I and Σ. Since J is finite and $J \models \theta$, there is a largest integer $n \geq 1$ such that there are distinct values $a_0, a_1, \dots, a_n \in \mathrm{dom}(J)$ with $a_0 = c$, $a_1 = d$, $(a_{i-1}, a_i) \in E^J$ for every $i \in [n]$, and $(a_n, a_i) \in E^J$ for some $i \leq n$. Let K be a cycle on $n + 2$ nodes. That is, pick a sequence

$e_0, e_1, \dots, e_{n+1}$ of distinct constants with $e_0 = c$ and $e_1 = d$, and let $K = \{E(e_{i-1}, e_i) \mid i \in [n+1]\} \cup \{E(e_{n+1}, e_0)\}$. Then it is clear that $J \not\to K$, which, since K is a finite model of I and Σ, means that J is not weakly universal.

Furthermore, there is a schema σ, and a finite set Σ of TGDs over σ such that it is undecidable whether there is a (strong or weak) universal model of a given ground σ-instance I and Σ [27] (see [26] for a proof tailored directly for universal models). Nevertheless, universal models can often be computed via the chase, which we introduce below.

Note that every two universal models of I and Σ are homomorphically equivalent. Consequently, the cores of universal models of I and Σ are isomorphic. Hence, if there is at least one universal model of I and Σ, then there is an instance that is isomorphic to the cores of all universal models of I and Σ. If Σ is a set of TGDs and EGDs, this instance is a universal model of I and Σ [18] (this may not be true if Σ contains constraints other than TGDs and EGDs). This suggests the following definition:

A *core model* of I and Σ is a model of I and Σ that is isomorphic to every core of every universal model of I and Σ. Up to isomorphism there is a unique core model of I and Σ, and core models of I and Σ exist if and only if universal models of I and Σ exist.

2.4 The Chase

The *chase* [4] is a procedure which, given an instance I and a finite set Σ of TGDs and EGDs, adds tuples to I and identifies values in order to obtain a universal model of I and Σ. There are several flavors of the chase (see, e.g., [28, 4, 6]), of which we use mainly the *oblivious chase* [6].

We first introduce the oblivious chase for TGDs. Let Σ be a finite set of TGDs. The oblivious chase starts with the input instance, and applies the following *TGD chase rule* for TGDs in Σ in a breadth-first fashion:

TGD CHASE RULE: A TGD $\theta = \forall \bar{x} \forall \bar{y} \, (\varphi(\bar{x}, \bar{y}) \to \exists \bar{z} \, \psi(\bar{x}, \bar{z}))$ *applies* to an instance I with an assignment α for the variables in φ if $I \models \varphi(\alpha)$. The result of applying θ to I with α is the instance J obtained from I as follows: Let β be an assignment for the variables in ψ such that β coincides with α on all variables in φ, and $\beta(\bar{z})$ is a tuple of distinct nulls that do not occur in I. Then J is the union of I and the atoms of $\psi(\beta)$.[1] We write $I \vdash_{\theta,\alpha} J$ if θ applies to I with α, and J is the result of applying θ to I with α.

More precisely, the *oblivious chase for I and Σ* starts with $I_0^\Sigma := I$, and proceeds in steps $i = 1, 2, \dots$. In step $i \geq 1$, let $(\theta_1, \alpha_1), \dots, (\theta_k, \alpha_k)$ be an enumeration of all pairs (θ, α) consisting of a TGD $\theta \in \Sigma$, and an assignment α such that θ applies to I_{i-1}^Σ with α, and θ was not applied with α before. If no such pair exists, we define $I_i^\Sigma := I_{i-1}^\Sigma$. Otherwise, I_i^Σ is obtained by applying each θ_i with α_i. That is, we define I_i^Σ such that $I_{i-1}^\Sigma \vdash_{\theta_1,\alpha_1} J_1 \vdash_{\theta_2,\alpha_2} \cdots \vdash_{\theta_k,\alpha_k} I_i^\Sigma$.[2] The *result* of the oblivious chase for I and Σ is $I^\Sigma := \bigcup_{i \geq 0} I_i^\Sigma$, which is unique up to isomorphism. We say that the oblivious chase *terminates* if there is some $i \geq 0$ with $I_i^\Sigma = I_{i+1}^\Sigma$.

EXAMPLE 2.2. Let Σ consist of the guarded TGDs

$$\theta_1 := \forall x, y \, \big(R(x,y) \wedge Q(y) \to \exists z \, \big(S(x,z,y) \wedge S(y,x,z)\big)\big),$$
$$\theta_2 := \forall x, y, z \, \big(S(x,y,z) \to \big(R(x,z) \wedge P(z)\big)\big),$$
$$\theta_3 := \forall x \, \big(P(x) \to Q(x)\big),$$

and let $I = \{R(c,d), P(d)\}$. Then we have $I_1^\Sigma = I \cup \{Q(d)\}$ and $I_2^\Sigma = I_1^\Sigma \cup \{S(c, \bot_1, d), S(d, c, \bot_1)\}$. Now, θ_2 is applied to I_2^Σ with the assignment mapping x, y, z to $c, \bot_1, d$, respectively. This generates the atoms $R(c,d)$ and $P(d)$. Note that these atoms belong to I. Moreover, θ_2 is applied to I_2^Σ with the assignment sending x, y, z to $d, c, \bot_1$. This generates the new atoms $R(d, \bot_1)$ and $P(\bot_1)$. Hence, $I_3^\Sigma = I_2^\Sigma \cup \{R(d, \bot_1), P(\bot_1)\}$. Altogether, I^Σ will be infinite.

The oblivious chase can be extended to finite sets Σ of TGDs and EGDs. To obtain I_{i+1}^Σ from I_i^Σ, we first apply all TGDs as before to I_i, resulting in an instance $\tilde{I}_i^\Sigma$. Then we apply all possible EGDs in Σ to $\tilde{I}_i^\Sigma$ until the resulting instance I_{i+1}^Σ satisfies the EGDs in Σ. Here, an EGD $\theta = \forall \bar{x} \, (\varphi(\bar{x}) \to y = z)$ *applies* to an instance K with α if $K \models \varphi(\alpha)$ and $\alpha(y) \neq \alpha(z)$. The application of θ to K with α *fails* if both $\alpha(y)$ and $\alpha(z)$ are constants. If the application does not fail, then the result of applying θ to K with α is the instance obtained from K by identifying $\alpha(y)$ and $\alpha(z)$ in K, that is, by replacing one of the nulls in $\{\alpha(y), \alpha(z)\}$ with the other value in $\{\alpha(y), \alpha(z)\}$. In particular, if exactly one of $\alpha(y)$ or $\alpha(z)$ is a null, say $\alpha(y)$, then every occurrence of $\alpha(y)$ in K is replaced by the constant $\alpha(z)$. If for some $i \geq 0$, an application of an EGD to $\tilde{I}_i^\Sigma$ fails, then we say that the oblivious chase for I and Σ *fails*, and we let I^Σ be undefined. Otherwise the *result* of the oblivious chase for I and Σ is $I^\Sigma := \{A \mid \text{there is an } i \geq 0 \text{ such that } A \in I_j^\Sigma \text{ for all } j \geq i\}$.

THEOREM 2.3 ([4, 17]). *Let I be a σ-instance, and let Σ be a finite set of TGDs and EGDs over σ. If I^Σ is defined, then I^Σ is a quasi-universal model of I and Σ. In particular, if I^Σ is finite, it is a universal model of I and Σ.*

The restricted chase is defined like the oblivious chase with the exception that a TGD is applied only if its head is not satisfied. Theorem 2.3 remains true if we replace I^Σ by the result of the restricted chase.

3. RELATED WORK

There is a long line of research on finding more and more general structural properties of a set Σ of TGDs and EGDs such that for all instances I, the restricted chase for I and Σ terminates [4, 14, 17, 15, 35, 29, 30, 22, 19]. For example, [14] studies *acyclic sets of IDs* for which the restricted chase is guaranteed to terminate. A much more general property is *weak acyclicity* [17]. If Σ is the union of a weakly acyclic set of TGDs, and a set of EGDs, then a universal model of I and Σ exists if and only if the restricted chase for I and Σ terminates and does not fail (and in this case, its result is a universal model for I and Σ). Furthermore, the number of steps of the restricted chase until a fixed point is reached is bounded by a

[1] Although the choice of nulls is not important here, we can make J well-defined as follows: We assume a linear order on *Null*. With respect to this linear order, we pick the first $k := |\bar{z}|$ nulls $\bot_1, \dots, \bot_k$ that are not in $\mathrm{dom}(I)$. Then β could map the ith variable in $\bar{z}$ to $\bot_i$.

[2] As with applications of TGDs, the order in which the TGDs are applied is not important here. We obtain a deterministic chase as follows. First, we fix a linear order on Σ and a linear order on *Dom*. The two orderings induce an ordering $\leq$ on pairs (θ, α) consisting of a TGD $\theta \in \Sigma$, and an assignment α for the variables in θ's body. When defining I_i^Σ we then order the pairs $(\theta_1, \alpha_1), \dots, (\theta_k, \alpha_k)$ according to $\leq$.

polynomial in the size of I (where the polynomial depends on Σ). Hence, if Σ is fixed, there is a polynomial time algorithm that, given an instance I, decides whether there is a universal model of I and Σ, and if so, outputs such a model. What has been said about weak acyclicity above is true for most of the other chase termination conditions mentioned above. Exceptions are *stratification* [15] and the initial proposal of *inductive restriction* in [29]. These properties ensure chase termination for at least one order of applying the TGDs in the chase, which can be determined from the set of TGDs. In [30], alternative definitions are proposed which resolve this issue. An excellent overview of the above chase termination conditions is given in [22].

All structural properties of TGDs and EGDs mentioned in the preceding paragraph ensure that the restricted chase terminates for *all* instances. On the other hand, if Σ is a finite set of IDs, the restricted chase may not terminate on some instances (e.g., recall Example 2.1). So, while some sets of IDs (and, more generally, linear TGDs or guarded TGDs) exhibit some of these structural properties, there is a large number of such sets which do not possess any of them. Moreover, there are finite sets of linear TGDs, where the restricted chase does not terminate, even though a universal model exists:

EXAMPLE 3.1. Consider the set Σ consisting of the following TGDs: $\forall x \forall y \forall z \forall u\, (R(x,y,z,u) \to \exists v\, R(x,y,u,v))$ and $\forall x \forall y \forall z\, (R(x,y,y,z) \to R(x,y,y,x))$. Let $I = \{R(c,d,c,d)\}$, where c,d are distinct constants. Then $J := I \cup \{R(c,d,d,c)\}$ is a universal model of I and Σ, since $I \subseteq J$, $J \models \Sigma$, and all models of I and Σ must contain the atoms of J (since these are contained in I^Σ). However, the restricted chase for I and Σ does not terminate. Intuitively, this is true since the atom $R(c,d,d,c)$ that could prevent the chase from generating an atom $R(c,d,d,\perp)$ with $\perp \in \mathit{Null}$ is produced *only after* $R(c,d,d,\perp)$ is generated. It is not hard to see that the presence of $R(c,d,d,\perp)$ enforces a nonterminating restricted chase for I and Σ.

Interestingly, it is often possible to deal with infinite chase results. For example, there is a wealth of research on evaluating conjunctive queries on the possibly infinite result of the oblivious chase [28, 11, 5, 6, 7, 9, 8]. Several researchers identified structural properties of sets Σ of TGDs and EGDs such that given an instance I and a Boolean conjunctive query q, we can decide whether $I^\Sigma \models q$, and that this can be done in polynomial time if Σ and q are fixed. Note that if there is at least one universal model of I and Σ, then this problem is equivalent to evaluating q on some universal model of I and Σ. Johnson and Klug [28] dealt with certain finite sets of IDs and functional dependencies. Their result was improved in [5] to a more general class of sets of IDs and functional dependencies, in [6, 7] to finite sets of guarded TGDs, EGDs, and negative constraints such that the EGDs "do not interfere" with the TGDs (see Section 6.1 for a precise definition), and in [9] to so-called *sticky* sets of TGDs, which are incomparable to sets of linear TGDs (EGDs that "do not interfere" with the TGDs, and negative constraints may also be added).

The basic idea behind all these results goes back to Johnson and Klug [28]. One shows that $I^\Sigma \models q$ implies $I_i^\Sigma \models q$, where i is bounded by a number s depending only on Σ and q (s can be computed from Σ and q). To decide whether $I^\Sigma \models q$, it then suffices to evaluate q on I_s^Σ, which can be computed in polynomial time if Σ and q are fixed. In Section 5 we use roughly the same basic idea to prove that universal models can be computed for finite sets of guarded TGDs: We show that if there is a universal model of a ground instance I and a finite set Σ of guarded TGDs, then such a model can be obtained from I_i^Σ, where i is bounded by a number depending only on Σ.

4. GUARDED CHASE FORESTS

An important basic technical tool used in [6, 7] are *guarded chase forests*. In this section, we review guarded chase forest and collect a few basic results.

4.1 Definition and Basic Properties

Basically, the guarded chase forest for an instance I and a set Σ of guarded TGDs is obtained by taking the atoms of I^Σ as nodes, and introducing an edge from an atom A to an atom B if, in the oblivious chase for I and Σ, B is the result of applying a TGD $\theta \in \Sigma$ with an assignment α such that $A = R(\alpha(\bar{u}))$, where $R(\bar{u})$ is the guard of θ. Note, however, that this may not yield a forest, since different applications of TGDs may introduce the same atom (recall Example 2.2). Therefore, we modify the construction of guarded chase forests as follows.

DEFINITION 4.1. Let I be an instance, and let Σ be a finite set of guarded TGDs. The *guarded chase forest* $\mathcal{G}_{I,\Sigma}$ *for* I *and* Σ is inductively constructed as follows.

We start with the forest $\mathcal{F}_0$ that contains, for each atom $A \in I$, a unique node v with label A, and no edges.

Let $i \geq 0$. The forest $\mathcal{F}_{i+1}$ is obtained from $\mathcal{F}_i$ by adding new nodes and edges as follows. Let atoms$(\mathcal{F}_i)$ be the set of all labels of nodes of $\mathcal{F}_i$. For each TGD $\theta \in \Sigma$ with guard $R(\bar{u})$, each assignment α for the variables in body(θ), and each node v in $\mathcal{F}_i$ we do the following: We say that θ *applies to* v *with* α if θ applies to atoms$(\mathcal{F}_i)$ with α, and $R(\alpha(\bar{u}))$ is the label of v. If θ applies to v with α, then we *apply* θ *to* v *with* α unless this has been done before. That is, we pick an assignment β for the variables in head(θ) as in the TGD chase rule, and add, for each atom $S(\bar{v})$ in head(θ), a new child with label $S(\beta(\bar{v}))$ to v.

$\mathcal{G}_{I,\Sigma}$ is the union of all the forests $\mathcal{F}_i$, over all $i \geq 0$. This is well-defined, since each $\mathcal{F}_{i+1}$ is an extension of $\mathcal{F}_i$.

For example, if I and Σ are as in Example 2.2, then, up to renaming of nulls, the first four levels of $\mathcal{G}_{I,\Sigma}$ are as shown in Figure 1.

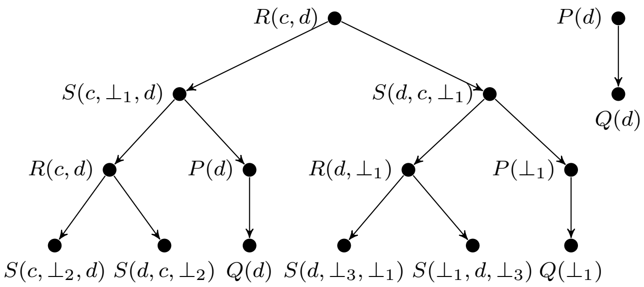

Figure 1: The first four levels of the guarded chase forest for I and Σ from Example 2.2.

For each node v of $\mathcal{G}_{I,\Sigma}$, let $\lambda(v)$ be the label of v. Similarly, for a subforest $\mathcal{F}$ of $\mathcal{G}_{I,\Sigma}$, let $\lambda(\mathcal{F})$ be the set of all labels of nodes in $\mathcal{F}$. The *depth of a node* v in $\mathcal{G}_{I,\Sigma}$, denoted by $\mathrm{depth}(v)$, is the length of the unique path from a root of $\mathcal{G}_{I,\Sigma}$ to v. The *depth of an atom* A in $\mathcal{G}_{I,\Sigma}$ is the minimum depth of a node in $\mathcal{G}_{I,\Sigma}$ with label A. In particular, each atom in I has depth 0 in $\mathcal{G}_{I,\Sigma}$. Let $\mathcal{G}^d_{I,\Sigma}$ be the subforest of $\mathcal{G}_{I,\Sigma}$ induced by all nodes of depth at most d.

Notice that, in general, $\lambda(\mathcal{G}_{I,\Sigma})$ contains more atoms than I^Σ. For example, in Figure 1, the atoms $S(c,\bot_2,d)$ and $S(d,c,\bot_2)$ would not be present in I^Σ (since θ_1 was applied before with the assignment mapping x, y to c, d). However, it is easy to see that:

PROPOSITION 4.2. *Let I be an instance, and let Σ be a finite set of guarded TGDs. Then $\lambda(\mathcal{G}_{I,\Sigma})$ is homomorphically equivalent to I^Σ.*

We now recall a few key results from [7]. First, we need to give a few definitions. The *cloud* of an atom A in $\lambda(\mathcal{G}_{I,\Sigma})$, denoted $\mathrm{cloud}(A)$, is the set of all atoms B in $\lambda(\mathcal{G}_{I,\Sigma})$ such that $\mathrm{dom}(\{B\}) \subseteq \mathrm{dom}(\{A\})$.[3] Two atoms $A, B \in \lambda(\mathcal{G}_{I,\Sigma})$ are *X-equivalent* for some $X \subseteq \mathit{Dom}$ if there is a bijective mapping $f\colon \mathrm{dom}(\mathrm{cloud}(A)) \to \mathrm{dom}(\mathrm{cloud}(B))$ such that $f(\{A\}) = \{B\}$, $f(\mathrm{cloud}(A)) = \mathrm{cloud}(B)$, $f(x) = x$ for all $x \in X \cap \mathrm{dom}(\{A\})$, and $f^{-1}(x) = x$ for all $x \in X \cap \mathrm{dom}(\{B\})$.

LEMMA 4.3 ([7]). *Let I be an instance, let Σ be a finite set of guarded TGDs, let v, w be nodes in $\mathcal{G}_{I,\Sigma}$, and let T_v and T_w be the subtrees of $\mathcal{G}_{I,\Sigma}$ rooted at v and w, respectively. If $\lambda(v)$ and $\lambda(w)$ are $\emptyset$-equivalent, then there is a bijection $f\colon \mathrm{dom}(\lambda(T_v)) \to \mathrm{dom}(\lambda(T_w))$ such that $f(\{\lambda(v)\}) = \{\lambda(w)\}$ and $f(\lambda(T_v)) = \lambda(T_w)$.*

LEMMA 4.4 ([7]). *Let σ be a schema, let I be a σ-instance, let Σ be a finite set of guarded TGDs over σ, and let $A \in \lambda(\mathcal{G}_{I,\Sigma})$. If $P \subseteq \lambda(\mathcal{G}_{I,\Sigma})$ contains more than*

$$\delta := (2w)^w \cdot 2^{(2w)^{w\cdot|\sigma|}} \quad (w := \max\{\mathrm{ar}(R) \mid R \in \sigma\})$$

atoms, then P contains two $\mathrm{dom}(\{A\})$*-equivalent atoms.*

From Lemma 4.4, the authors of [7] infer the following:

LEMMA 4.5 ([7]). *Let σ be a schema, let I and J be σ-instances, and let Σ be a finite set of guarded TGDs over σ. Suppose that $J \to \lambda(\mathcal{G}_{I,\Sigma})$. Then, $J \to \lambda(\mathcal{G}^{|J|\cdot\delta}_{I,\Sigma})$, where δ is as in Lemma 4.4.*

REMARK 4.6. In [7], the authors assume that all TGDs have single-atom heads. In fact, in [6, Lemma 10], they show that, as far as their results are concerned, this assumption can be made without loss of generality. Specifically, they show that a TGD $\theta = \forall\bar{x}\forall\bar{y}\,(\varphi(\bar{x},\bar{y}) \to \exists\bar{z}\,\psi(\bar{x},\bar{z}))$ can be safely replaced by the following TGDs: $\forall\bar{x}\forall\bar{y}\,(\varphi(\bar{x},\bar{y}) \to \exists\bar{z}\,R_\theta(\bar{x}\bar{z}))$, where R_θ is a fresh relation symbol, and for each atom A in ψ a TGD $\forall\bar{x}\forall\bar{z}\,(R_\theta(\bar{x}\bar{z}) \to A)$. Therefore, the above-mentioned three lemmas from [7] also hold for sets of TGDs with multiple-atom heads.

Note that the above-mentioned reduction to sets of TGDs with single-atom heads does not preserve existence of universal solutions. For example, let $I = \{E(c,c)\}$, and let Σ consist of the TGD $\forall x\forall y\,\big(E(x,y) \to \exists z\exists u\,(E(y,z) \wedge E(z,u))\big)$. Then I is a universal model of I and Σ, but it is not hard to see that there is no universal model of I and Σ', where Σ' consists of the three TGDs $\forall x\forall y\,(E(x,y) \to \exists z\exists u\,R(y,z,u))$, $\forall x\forall y\forall u\,(R(y,z,u) \to E(y,z))$, and $\forall x\forall y\forall u\,(R(y,z,u) \to E(z,u))$ resulting from the reduction. Therefore, we cannot make this simplifying assumption in Section 5, where we prove this paper's main result.

It is not obvious how to obtain the first d levels of the guarded chase forest for I and Σ. The naive approach of computing consecutive levels of $\mathcal{G}_{I,\Sigma}$ in a breadth-first fashion until level d is reached may fail (although it succeeds for sets of linear TGDs), because the application of a guarded TGD generating an atom at depth $\leq d$ may require an atom at depth $> d$. However, [7] shows that there is a constant Δ that is computable from Σ such that all atoms needed to generate an atom at depth $\leq d$ can be found in the first $d + \Delta$ levels of $\mathcal{G}_{I,\Sigma}$. This leads to:

LEMMA 4.7 ([7], SEE THE REVISED VERSION). *There is an algorithm that, given a schema σ, a ground σ-instance I, a finite set Σ of guarded TGDs over σ, and a number d as input, computes $\mathcal{G}^d_{I,\Sigma}$ in time $O(\|I\|^k)$, where k depends only on Σ and d.*

4.2 Core Computation

We will later need to compute the core of $\lambda(\mathcal{G}^\ell_{I,\Sigma})$ for some constant ℓ. To show that for fixed Σ this is possible in time polynomial in the size of $\mathcal{G}^\ell_{I,\Sigma}$, we use a result of [21]. To apply this result, we need to show that guarded chase forests have hypertree decompositions of small width.

DEFINITION 4.8 ([20]). Let I be an instance. A *hypertree decomposition* of I is a triple (T,χ,λ), where $T = (V,E)$ is a rooted tree, χ is a mapping from V to subsets of $\mathrm{nulls}(I)$, and λ is a mapping from V to subsets of I such that:

1. For each atom $A \in I$ there is a node $v \in V$ such that all nulls in A occur in $\chi(v)$.
2. For each $\bot \in \mathrm{nulls}(I)$ the subgraph of T induced by the nodes $v \in V$ with $\bot \in \chi(v)$ is a rooted tree.
3. For each $v \in V$ we have $\chi(v) \subseteq \mathrm{nulls}(\lambda(v))$.
4. For each $v \in V$ we have $\mathrm{nulls}(\lambda(v)) \cap \chi(T_v) \subseteq \chi(v)$, where $\chi(T_v)$ is the union of the $\chi(w)$ over all nodes w in the subtree of T rooted at v.

The *width of* (T,χ,λ) is defined as $\min\{|\lambda(v)| \mid v \in V\}$.

It is easy to turn $\mathcal{G}_{I,\Sigma}$ into a hypertree decomposition of $\lambda(\mathcal{G}_{I,\Sigma})$ whose width is at most the maximum number of atoms in the head of a TGD in Σ. Specifically, construct the following tree T^* and labeling λ^*. We start with an empty tree and add all nodes of $\mathcal{G}_{I,\Sigma}$. Then, for each node v of $\mathcal{G}_{I,\Sigma}$, we do the following: We set $\lambda^*(v) := \lambda(v)$. Furthermore, if during the construction of $\mathcal{G}_{I,\Sigma}$, a TGD θ is applied to v with α (using an assignment β for the variables in $\mathrm{head}(\theta)$), we add a new node v^*, label it with $\lambda^*(v^*) := \{S(\beta(\bar{v})) \mid S(\bar{v}) \text{ is an atom in } \mathrm{head}(\theta)\}$, add an edge from v to v^*, and add edges from v^* to the children w of v in $\mathcal{G}_{I,\Sigma}$ that are the result of applying θ to v with α. We let $\chi^*(v) := \mathrm{nulls}(\lambda^*(v))$ for all nodes v of T^*. It is now easy to verify that (T^*,χ^*,λ^*) is a hypertree decomposition of $\mathcal{G}_{I,\Sigma}$ whose width is at most the maximum number of atoms in the head of a TGD in Σ.

[3] In [7], clouds are called *types*. We use "cloud" here in order to avoid confusion with FO-types introduced later. It should also be noted that clouds are called *restricted clouds* in [6].

Together with the following result, it is possible to compute the core of $\lambda(\mathcal{G}^{\ell}_{I,\Sigma})$ in time polynomial in the size of $\mathcal{G}^{\ell}_{I,\Sigma}$ (and exponential in the maximum number of atoms in the head of a TGD in Σ).

Theorem 4.9 ([21]). *Let I be an instance, and (T, χ, λ) a hypertree decomposition of I of width k. Then a core of I can be computed in time $O(t \cdot n^{k+1})$, where t is the number of nodes of T, and n is the size of I.*

Corollary 4.10. *Let I be an instance, let Σ be a finite set of guarded TGDs, and let $\ell \geq 0$. Then a core of $\lambda(\mathcal{G}^{\ell}_{I,\Sigma})$ can be computed in time $O(t \cdot n^{k+1})$, where t is the number of nodes of $\mathcal{G}^{\ell}_{I,\Sigma}$, n is the size of $\lambda(\mathcal{G}^{\ell}_{I,\Sigma})$, and k is the maximum number of atoms in the head of a TGD in Σ.*

5. UNIVERSAL MODELS OF GROUND INSTANCES AND GUARDED TGDS

In this section we prove the key result of this paper:

Theorem 5.1. *There is an algorithm that, given a schema σ, a ground σ-instance I, and a finite set Σ of guarded TGDs over σ as input, decides whether there is a universal model of I and Σ, and if so, computes a core model of I and Σ. The running time of the algorithm is $O(\|I\|^k)$, where k depends only on Σ.*

The remaining part of this section is devoted to a proof of Theorem 5.1. Let us start with an overview.

5.1 Proof Overview and Basic Results

Let σ be a schema, let I be a ground σ-instance, and let Σ be a finite set of guarded TGDs over σ. How can we decide whether a universal model of I and Σ exists?

Recall from Section 2.3 that a universal model of I and Σ exists if and only if a core model of I and Σ exists, and that a core model of I and Σ is a particular universal model of I and Σ. Therefore, rather than deciding whether a universal model of I and Σ exists, we decide whether a core model of I and Σ exists, and if so, we compute one. It is easy to see:

Proposition 5.2.

1. *If there is a core model of I and Σ, then $\lambda(\mathcal{G}_{I,\Sigma})$ has a finite core (equivalently, all cores of $\lambda(\mathcal{G}_{I,\Sigma})$ are finite).*
2. *If J is a finite core of $\lambda(\mathcal{G}_{I,\Sigma})$, then J is a core model of I and Σ.*

Hence, to decide whether a core model of I and Σ exists, it suffices to check whether there is a finite core of $J^* := \lambda(\mathcal{G}_{I,\Sigma})$. To check whether such a core exists, we check for a homomorphism h from J^* to J^* such that $K := h(J^*)$ is finite. Then K has a finite core, and this core is a core of J^*. We show that if a homomorphism h as above exists, then there is a homomorphism from J^* to $\lambda(\mathcal{G}^{d}_{I,\Sigma})$, where d is bounded by a number depending only on Σ (Lemma 5.6). This is the key part of the whole proof. Hence, all that remains is to check whether there is a homomorphism from J^* to $\lambda(\mathcal{G}^{d}_{I,\Sigma})$, and if so, to compute a core of $\lambda(\mathcal{G}^{d}_{I,\Sigma})$. This is easy: All we need to do is to compute a core J of $\lambda(\mathcal{G}^{d}_{I,\Sigma})$, and to check whether $J \models \Sigma$. If so, J is a core model of I and Σ. Otherwise, no such core model exists.

Let us summarize what we have established so far:

Proposition 5.3. *Let $d \geq 0$.*

1. *If $\lambda(\mathcal{G}_{I,\Sigma}) \to \lambda(\mathcal{G}^{d}_{I,\Sigma})$, then there is a core of $\lambda(\mathcal{G}^{d}_{I,\Sigma})$ that satisfies Σ (or, equivalently, all cores of $\lambda(\mathcal{G}^{d}_{I,\Sigma})$ satisfy Σ).*
2. *If J is a core of $\lambda(\mathcal{G}^{d}_{I,\Sigma})$ with $J \models \Sigma$, then J is a core model of I and Σ.*

5.2 FO-Types

We recall the notion of FO-types, which is used in the proof of the main lemma, Lemma 5.6, below.

The *quantifier rank* of a FO-formula φ, denoted by $\mathrm{qr}(\varphi)$, is the maximum nesting depth of quantifiers in φ. It is defined by induction on the structure of φ as follows: If φ is atomic, then $\mathrm{qr}(\varphi) = 0$; otherwise we have $\mathrm{qr}(\neg\varphi) = \mathrm{qr}(\varphi)$, $\mathrm{qr}(\varphi \star \psi) = \max\{\mathrm{qr}(\varphi), \mathrm{qr}(\psi)\}$ for $\star \in \{\wedge, \vee, \to, \leftrightarrow\}$, and $\mathrm{qr}(\exists x\, \varphi) = \mathrm{qr}(\forall x\, \varphi) = 1 + \mathrm{qr}(\varphi)$ (see, e.g., [16, 32]).

Let $k \geq 0$. If $k \geq 1$, let us also fix pairwise distinct variables $x_1, \dots, x_k$. Let I be a σ-instance, and let $\bar{a} = (a_1, \dots, a_k) \in \mathit{Dom}^k$. Construct a logical structure $\mathcal{A}_{I,\bar{a}}$ with universe $\mathrm{dom}(I) \cup \{a_1, \dots, a_k\}$ and relations R^I for each $R \in \sigma$. The $\mathrm{FO}_{q,k}$*-type of $\bar{a}$ in I*, denoted $\mathrm{tp}_q(I, \bar{a})$, is the set of all constant-free FO-formulae $\varphi(x_1, \dots, x_k)$ over σ such that $\mathrm{qr}(\varphi) \leq q$ and $\mathcal{A}_{I,\bar{a}} \models \varphi(\bar{a})$.

Up to logical equivalence, there are only finitely many $\mathrm{FO}_{q,k}$-types (see, e.g., [16, 32]). To give a more precise bound, let us define $\mathrm{tow}\colon \mathbb{N}^2 \to \mathbb{N}$ such that for all $x, y \in \mathbb{N}$, $\mathrm{tow}(0, y) := y$, and $\mathrm{tow}(x + 1, y) := 2^{\mathrm{tow}(x,y)}$. Then:

Lemma 5.4. *Let σ be a schema, let $q, k \in \mathbb{N}$, and let $T_{\sigma,q,k}$ be the set containing precisely one representative of each $\mathrm{FO}_{q,k}$-type. Then $|T_{\sigma,q,k}| \leq \mathrm{tow}(q + 1, t)$, where $t = 2(|\sigma|+1)(k+q)^w$ and w is the maximum of 2 and the maximal arity of a relation symbol in σ.*

We need the following composition lemma. A proof of this lemma can be found, for example, in [34, 23].

Lemma 5.5 (see, e.g., [34, 23]). *Let σ be a schema, let I, J be σ-instances, and let $\bar{a} = (a_1, \dots, a_k) \in \mathit{Dom}^k$ such that $\mathrm{dom}(I) \cap \mathrm{dom}(J) \subseteq \{a_1, \dots, a_k\}$. Then for all $q \geq 0$, $\mathrm{tp}_q(I \cup J, \bar{a})$ is determined by $\mathrm{tp}_q(I, \bar{a})$ and $\mathrm{tp}_q(J, \bar{a})$.*

5.3 Main Lemma

We are now ready to prove the main technical lemma of this section.

Lemma 5.6. *Let σ be a schema, let I be a ground σ-instance, and let Σ be a finite set of guarded TGDs over σ. If $\lambda(\mathcal{G}_{I,\Sigma})$ has a finite core, then there is a homomorphism from $\lambda(\mathcal{G}_{I,\Sigma})$ to $\lambda(\mathcal{G}^{d}_{I,\Sigma})$, where d depends only on Σ (and can be computed from Σ).*

The remaining part of the present section is devoted to a proof of Lemma 5.6.

Consider a ground σ-instance I and a finite set Σ of guarded TGDs over σ. Suppose $\lambda(\mathcal{G}_{I,\Sigma})$ has a finite core. Then there is a homomorphism h from $\lambda(\mathcal{G}_{I,\Sigma})$ to $\lambda(\mathcal{G}_{I,\Sigma})$ such that $h(\lambda(\mathcal{G}_{I,\Sigma}))$ is such a finite core. In particular, there is a finite subforest $\mathcal{F}$ of $\mathcal{G}_{I,\Sigma}$ such that:

(F1) $\mathcal{F}$ is closed under ancestors, that is, for each node v in $\mathcal{F}$, all ancestors of v in $\mathcal{G}_{I,\Sigma}$ are nodes of $\mathcal{F}$, and

(F2) $\lambda(\mathcal{F})$ contains a core of $\lambda(\mathcal{G}_{I,\Sigma})$ (since $\lambda(\mathcal{F}) \subseteq \lambda(\mathcal{G}_{I,\Sigma})$, this means that $\lambda(\mathcal{F})$ and $\lambda(\mathcal{G}_{I,\Sigma})$ have isomorphic cores)

(e.g., take $\mathcal{F}$ to be $\mathcal{G}^d_{I,\Sigma}$, where d is the maximum depth of an atom in a finite core of $\lambda(\mathcal{G}_{I,\Sigma})$).

Let us pick a finite subforest $\mathcal{F}$ of $\mathcal{G}_{I,\Sigma}$ with properties (F1) and (F2) such that $\sum_{v \in V(\mathcal{F})} \text{depth}(v)$ is minimal among all such forests. Here, $V(\mathcal{F})$ denotes the set of nodes of $\mathcal{F}$. More generally, given any graph G, we let $V(G)$ be the set of nodes of G.

Let K be a core of $\lambda(\mathcal{F})$. Note that K is, in particular, a core of $\lambda(\mathcal{G}_{I,\Sigma})$.

Lemma 5.7. *For each atom $A \in K$ there is a unique node v in $\mathcal{F}$ with $\lambda(v) = A$.*

Proof. For a contradiction, suppose that there exists an atom $A \in K$ and distinct nodes v_1, v_2 in $\mathcal{F}$ such that $\lambda(v_1) = \lambda(v_2) = A$. Without loss of generality, we assume that $\text{depth}(v_1) \geq \text{depth}(v_2)$. Let T_1 and T_2 be the subtrees of $\mathcal{G}_{I,\Sigma}$ rooted at v_1 and v_2, respectively.

By the construction of $\mathcal{G}_{I,\Sigma}$, there is an isomorphism f from T_1 to T_2 with

$$f(v_1) = v_2, \tag{5.1}$$

and an isomorphism g from $\lambda(T_1)$ to $\lambda(T_2)$ such that for each node v of T_1 we have

$$g(\lambda(v)) = \lambda(f(v)). \tag{5.2}$$

Let T'_1 be the subtree of $\mathcal{F}$ induced by the nodes in T_1. Furthermore, let $\mathcal{F}'$ be the forest obtained from $\mathcal{F}$ by removing from $\mathcal{F}$ all nodes (and edges) of T'_1, and adding all nodes and edges of $f(T'_1)$, where $f(T'_1)$ is the tree obtained from T'_1 by renaming each node v in T'_1 to $f(v)$. See Figure 2 for an illustration.

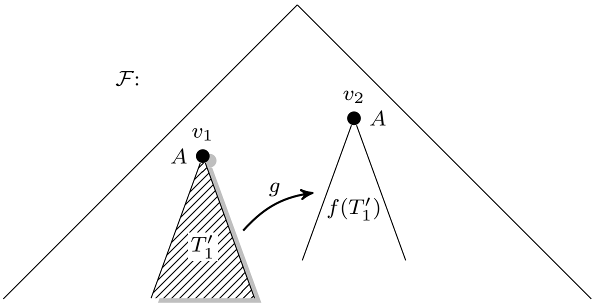

Figure 2: Construction of the forest $\mathcal{F}'$ from $\mathcal{F}$. The subtree T'_1 below v_1 is removed from $\mathcal{F}$, and an isomorphic copy $f(T'_1)$ of T'_1 is glued to $\mathcal{F}$ below v_2.

Clearly, $\mathcal{F}'$ is a finite subforest of $\mathcal{G}_{I,\Sigma}$. Furthermore, it is easy to see that $\mathcal{F}'$ satisfies property (F1).

We claim that $\mathcal{F}'$ also satisfies property (F2). First we extend g to a homomorphism g' from $\lambda(\mathcal{F})$ to $\lambda(\mathcal{F}')$. Consider the mapping $g' \colon \text{dom}(\lambda(\mathcal{F})) \to \text{dom}(\lambda(\mathcal{F}'))$ defined as

$$g'(a) := \begin{cases} g(a), & \text{if } a \in \text{dom}(\lambda(T'_1)), \\ a, & \text{if } a \in \text{dom}(\lambda(\mathcal{F}) \setminus \lambda(T'_1)). \end{cases}$$

This mapping is well-defined. First, (5.1) and (5.2) imply that $g(\lambda(v_1)) = \lambda(v_2) = \lambda(v_1)$. Therefore, we have $g(a) = a$ for all values that occur in $\lambda(v_1)$, and these are all values that could occur both in an atom of $\lambda(T'_1)$ and in an atom of $\lambda(\mathcal{F}) \setminus \lambda(T'_1)$. Note that $g'(\lambda(T'_1)) = \lambda(f(T'_1)) \subseteq \lambda(\mathcal{F}')$ and $g'(\lambda(\mathcal{F}) \setminus \lambda(T'_1)) = \lambda(\mathcal{F}) \setminus \lambda(T'_1) \subseteq \mathcal{F}'$. Hence, g' is a homomorphism from $\lambda(\mathcal{F})$ to $\lambda(\mathcal{F}')$.

Since $\lambda(\mathcal{F}') \subseteq \lambda(\mathcal{G}_{I,\Sigma})$ and $\lambda(\mathcal{G}_{I,\Sigma}) \to \lambda(\mathcal{F})$, there is a homomorphism from $\lambda(\mathcal{F}')$ to $\lambda(\mathcal{F})$. It follows that $\lambda(\mathcal{F})$ and $\lambda(\mathcal{F}')$ are homomorphically equivalent, so they have isomorphic cores. Therefore, since $\mathcal{F}$ satisfies property (F2), $\mathcal{F}'$ satisfies property (F2), too.

Altogether, $\mathcal{F}'$ is a finite subforest of $\mathcal{G}_{I,\Sigma}$ with properties (F1) and (F2). We now show that

$$\sum_{v \in V(\mathcal{F}')} \text{depth}(v) < \sum_{v \in V(\mathcal{F})} \text{depth}(v), \tag{5.3}$$

which is impossible by the choice of $\mathcal{F}$, and thus leads to the desired contradiction. Since $\text{depth}(v_1) \geq \text{depth}(v_2)$, and v_1, v_2 are the roots of T'_1 and $f(T'_1)$, respectively, we have

$$\sum_{v \in V(f(T'_1))} \text{depth}(v) \leq \sum_{v \in V(T'_1)} \text{depth}(v). \tag{5.4}$$

Note that $V(\mathcal{F}') \setminus V(f(T'_1)) \subseteq V(\mathcal{F}) \setminus V(T'_1)$, and since $v_2 \in V(f(T'_1))$ and $v_2 \notin V(T'_1)$, the inclusion is strict. Therefore,

$$\sum_{v \in V(\mathcal{F}') \setminus V(f(T'_1))} \text{depth}(v) < \sum_{v \in V(\mathcal{F}) \setminus V(T'_1)} \text{depth}(v). \tag{5.5}$$

From (5.4) and (5.5), it is not hard to obtain (5.3). □

By Lemma 5.7, we can pick for each atom $A \in K$ a unique node v_A of $\mathcal{F}$ such that $\lambda(v_A) = A$. Let $V_K := \{v_A \mid A \in K\}$. Let B be the set of all nodes of $\mathcal{F}$ that belong to V_K or have at least two children. Note that by the choice of $\mathcal{F}$, we have:

(P1) All roots of $\mathcal{F}$ (i.e., those nodes that represent atoms of I) belong to V_K, since $I \subseteq K$.

(P2) All leaves of $\mathcal{F}$ belong to V_K.

Nodes with exactly one child in $\mathcal{F}$ either belong to $V_K \subseteq B$, or do not belong to B at all. On the other hand, nodes with at least two children in $\mathcal{F}$ either belong to V_K or to $B \setminus V_K$. Figure 3 illustrates this situation.

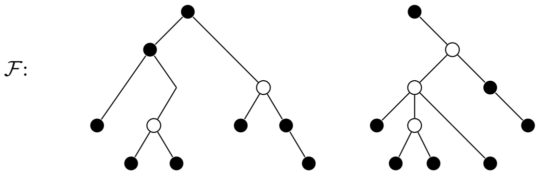

Figure 3: A possible configuration of the nodes in B. Black nodes represent nodes in V_K, white nodes are nodes in $B \setminus V_K$.

In what follows we bound, for every path P in $\mathcal{F}$, the number of nodes in B that occur on P, and the number of nodes between any two consecutive B-nodes on P. This enables us to bound the overall depth of $\mathcal{F}$.

Recall the definition of X-equivalent from Section 4. For atoms $A, B \in \mathcal{G}_{I,\Sigma}$, we write $A \sim B$ if they are $\emptyset$-equivalent. From Lemma 4.4 it follows that:

Proposition 5.8. *For any set $P \subseteq \lambda(\mathcal{G}_{I,\Sigma})$ with $|P| > \delta$, where δ is as in Lemma 4.4, there are distinct atoms $A, B \in P$ with $A \sim B$.*

Let w be the maximum arity of a relation symbol in σ. Define $q := \max\{\mathrm{qr}(\theta) \mid \theta \in \Sigma\}$, and denote by t the number of $\mathrm{FO}_{q,w}$-types. Also define

$$s := \delta \cdot t.$$

Note that s depends only on Σ, and that an upper bound on s can be computed from Σ (cf. Lemma 4.4 for δ, and Lemma 5.4 for an upper bound on t).

Lemma 5.9. *Every path in $\mathcal{F}$ contains $\leq s$ nodes from B.*

Proof. For a contradiction, suppose that P is a path in $\mathcal{F}$ with more than s nodes from B. Let $v_1, v_2, \ldots, v_{s+1}$ be the first $s+1$ nodes from B on P.

Let $i \in [s+1]$. Denote by T_i the subtree of $\mathcal{G}_{I,\Sigma}$ rooted at v_i, and by T'_i the subtree of $\mathcal{F}$ rooted at v_i. Define

$$K_i := K \cap \lambda(T'_i).$$

Furthermore, let $X_i := \mathrm{dom}(\{\lambda(v_i)\})$, and let $\bar{a}_i$ be the tuple of values in $\lambda(v_i)$, so that $\lambda(v_i) = R(\bar{a}_i)$ for some $R \in \sigma$.

By the choice of s (and Proposition 5.8), there are $i, j \in [s+1]$ with $i < j$ such that

$$\lambda(v_i) \sim \lambda(v_j), \tag{5.6}$$

$$\mathrm{tp}_q\big(K_i, \bar{a}_i\big) = \mathrm{tp}_q\big(K_j, \bar{a}_j\big). \tag{5.7}$$

In particular, (5.6) and (5.7) imply:

> There is a unique bijective mapping $f\colon X_j \to X_i$ with $f(\bar{a}_j) = \bar{a}_i$. For each $a \in X_j$, we have $a \in \mathrm{dom}(K_j)$ if and only if $f(a) \in \mathrm{dom}(K_i)$. (5.8)

The latter statement in (5.8) is an easy consequence of the definition of $\mathrm{tp}_q(\cdot,\cdot)$.

We now prune K by replacing K_i with an isomorphic copy of K_j. Then we show that the pruned instance is a universal model of I and Σ, but has at least one atom less than K. This is the desired contradiction, since K is the smallest universal model of I and Σ.

Step 1: Construction of the pruned instance J.
Let $h\colon \mathrm{dom}(\lambda(T_j)) \to \mathit{Dom}$ be such that $h(\bar{a}_j) = \bar{a}_i$, and $h(a) = a$ for all values $a \in \mathrm{dom}(\lambda(T_j))$ that do not occur in $\bar{a}_j$ (i.e., for all nulls a created in T_j). By (5.8), such a mapping h exists. We now remove K_i from K, and add $h(K_j)$ instead:

$$J := (K \setminus K_i) \cup h(K_j).$$

See Figure 4 for an illustration.

Step 2: J is a universal model of I and Σ.
Recall that h is bijective, and that $h(\bar{a}_j) = \bar{a}_i$. Furthermore, by (5.8), for each $a \in X_j$, we have $a \in \mathrm{dom}(K_j)$ if and only if $h(a) \in \mathrm{dom}(K_i)$. Therefore,

$$\mathrm{tp}_q\big(h(K_j), \bar{a}_i\big) = \mathrm{tp}_q\big(h(K_j), h(\bar{a}_j)\big) = \mathrm{tp}_q\big(K_j, \bar{a}_j\big),$$

which, by (5.7), implies $\mathrm{tp}_q(K_i, \bar{a}_i) = \mathrm{tp}_q(h(K_j), \bar{a}_i)$. Note that $\bar{a}_i$ contains all values that might occur both in K_i (resp., $h(K_j)$) and $K \setminus K_i$. Thus, by Lemma 5.5, we have $\mathrm{tp}_q(K, \bar{a}_i) = \mathrm{tp}_q(J, \bar{a}_i)$. Since $K \models \Sigma$, and q is the maximum quantifier rank of a TGD in Σ, this implies $J \models \Sigma$. It is also clear that $I \subseteq J$. Therefore, J is a model of I and Σ.

To show that J is a universal model of I and Σ, it suffices to show that $J \to \lambda(\mathcal{G}_{I,\Sigma})$, since $\lambda(\mathcal{G}_{I,\Sigma})$ is quasi-universal by Proposition 4.2.

From (5.6) and Lemma 4.3, we know that there is a bijection $f\colon \mathrm{dom}(\lambda(T_j)) \to \mathrm{dom}(\lambda(T_i))$ such that $f(\{\lambda(v_j)\}) = \{\lambda(v_i)\}$ and $f(\lambda(T_j)) = \lambda(T_i)$. In particular, $f(\bar{a}_j) = \bar{a}_i$. Let $g := f \circ h^{-1}$. Then,

> g is an isomorphism from $h(\lambda(T_j))$ to $\lambda(T_i)$ with $g(\bar{a}_i) = \bar{a}_i$.

Now,

$$g(h(K_j)) \subseteq g\big(h\big(\lambda(T_j)\big)\big) \subseteq \lambda(T_i) \subseteq \lambda(\mathcal{G}_{I,\Sigma}). \tag{5.9}$$

To obtain a homomorphism from J to $\lambda(\mathcal{G}_{I,\Sigma})$, we extend g so that $g(c) = c$ for all $c \in \mathrm{dom}(K \setminus K_i)$. This is possible since $\bar{a}_i$ contains all values in $\mathrm{dom}(K \setminus K_i) \cap \mathrm{dom}(h(K_j))$, and $g(\bar{a}_i) = \bar{a}_i$. Then

$$g(K \setminus K_i) = K \setminus K_i \subseteq \lambda(\mathcal{G}_{I,\Sigma}). \tag{5.10}$$

Altogether, (5.9) and (5.10) imply that g is a homomorphism from J to $\lambda(\mathcal{G}_{I,\Sigma})$, as desired.

Step 3: J has less atoms than K.
It remains to show that $|J| < |K|$. Recall that $v_i \in B$. This means that $v_i \in V_K$, or v_i has at least two children in $\mathcal{F}$.

If $v_i \in V_K$, then $K_j \subsetneq K_i$, since $K_j \subseteq K_i$, $\lambda(v_i) \in K_i$, and $\lambda(v_i) \notin K_j$ (the latter follows from $v_i \neq v_j$ and Lemma 5.7). Consequently, $|h(K_j)| = |K_j| < |K_i|$, and hence,

$$|J| \leq |K \setminus K_i| + |h(K_j)| < |K \setminus K_i| + |K_i| = |K|.$$

Now assume that v_i has at least two children in $\mathcal{F}$. Pick distinct children w_1, w_2 of v_i such that there is a path from w_1 to v_j. By property (P2), there is an atom $A \in K$ such that v_A is reachable from w_2 (we only have to pick a leaf in the subtree of $\mathcal{F}$ rooted at w_2). Using Lemma 5.7, we obtain $A \in K_i \setminus K_j$. This yields $K_j \subsetneq K_i$, and it follows as above that $|J| < |K|$.

Altogether, we have constructed a universal model J of I and Σ with $|J| < |K|$. But since K is a universal model of I and Σ with a minimal number of atoms, this is impossible, and we have the desired contradiction. □

Lemma 5.10. *If P is a path in $\mathcal{F}$ without nodes from B, then P contains at most δ nodes, where δ is as in Lemma 4.4.*

Proof. Follows from Lemma 4.4 in the same way as in the proof of Lemma 4 in [7]. □

From the preceding two lemmas it follows that all nodes in $\mathcal{F}$ have depth at most $d := (\delta+1)\cdot s$, where d depends only on Σ. By property (F2), $\lambda(\mathcal{F})$ contains a core of $\lambda(\mathcal{G}_{I,\Sigma})$. Hence, there is a homomorphism from $\lambda(\mathcal{G}_{I,\Sigma})$ to $\lambda(\mathcal{F}) \subseteq \lambda(\mathcal{G}^d_{I,\Sigma})$. Since δ and s can be computed from Σ, d can be computed from Σ. This completes the proof of Lemma 5.6.

5.4 The Algorithm

Given a schema σ, a ground σ-instance I, and a finite set Σ of guarded TGDs over σ, the following algorithm decides whether there is a universal model of I and Σ, and if so, computes a core model of I and Σ:

1. Compute the number d from Lemma 5.6 (which depends only on Σ).
2. Compute $\mathcal{F} := \mathcal{G}^d_{I,\Sigma}$.
3. Compute a core K of $\lambda(\mathcal{F})$.
4. If $K \models \Sigma$, then output K; otherwise output "There is no universal model of I and Σ".

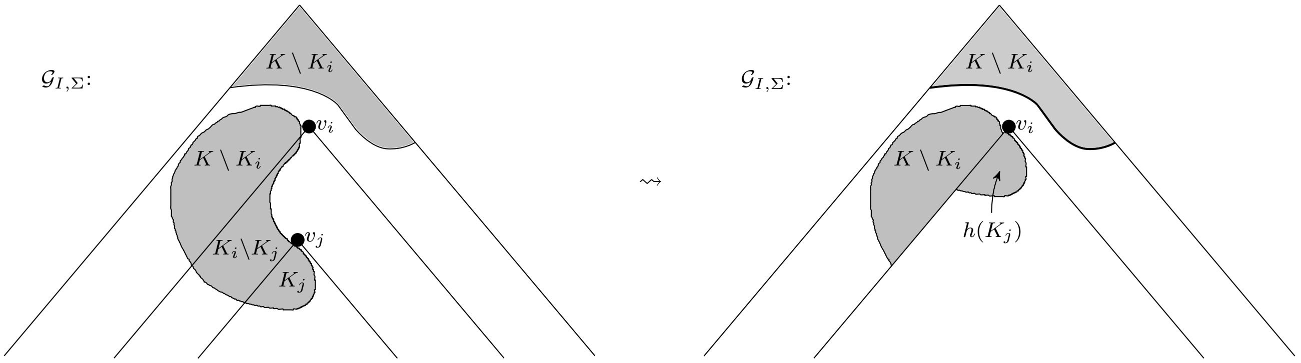

Figure 4: Construction of J from K. The subinstance K_i is removed from K, and an isomorphic copy $h(K_j)$ of K_j is glued to K.

By Lemma 5.6 and Proposition 5.3, the algorithm is correct and complete. It remains to show that the algorithm can be implemented so that it runs in time $O(\|I\|^k)$ for a number k that depends only on Σ. To this end, we show that each of the four steps needs at most time $O(\|I\|^{k'})$, for a number k' that depends only on Σ. This is certainly true for the first and the last step, and for steps 2 and 3 it follows from Lemma 4.7 and Corollary 4.10, respectively. Altogether, this proves Theorem 5.1.

6. EXTENSIONS

In this section, we generalize Theorem 5.1 to more general sets of constraints, and to instances with nulls.

6.1 Adding Negative Constraints and EGDs

We begin by extending Theorem 5.1 to the case where the set Σ contains not only guarded TGDs, but also negative constraints and certain EGDs, including non-conflicting keys considered in [7]. Such sets of constraints were recently shown to be expressive enough to capture several members of the DL-Lite family of description logics [7].

6.1.1 Negative Constraints

A *negative constraint* as defined in [7] is a FO-sentence of the form $\forall \bar{x}\,(\varphi \rightarrow \bot)$, where φ is a conjunction of relational atomic formulae. Here, $\bot$ is interpreted as "false". So, a negative constraint $\forall \bar{x}\,(\varphi \rightarrow \bot)$ is satisfied in an instance I if for all tuples $\bar{a} \in (\mathrm{dom}(I) \cup \mathrm{dom}(\varphi))^{|\bar{x}|}$ we have $I \not\models \varphi(\bar{a})$.

Extending Theorem 5.1 to finite sets Σ of guarded TGDs and negative constraints is very easy. It boils down to deciding whether there is a universal model J with respect to the set of TGDs in Σ such that J satisfies all negative constraints in Σ. If so, J is a universal model of I and Σ. This is similar to (and in fact follows from) the corresponding result in [7] which shows that to answer a Boolean conjunctive query with respect to Σ, it suffices to check that the query is true with respect to the TGDs in Σ, and that none of the negative constraints in Σ holds with respect to those TGDs.

PROPOSITION 6.1. *Let I be a ground σ-instance, let Σ be a finite set of TGDs and EGDs over σ, and let Σ_N be a finite set of negative constraints over σ. Then for all σ-instances J the following are equivalent:*

1. *J is a universal model of I and $\Sigma \cup \Sigma_N$.*
2. *J is a universal model of I and Σ, and $J \models \Sigma_N$.*

6.1.2 EGDs

Next we incorporate EGDs. The interaction of EGDs and TGDs often leads to undecidability. For example, answering conjunctive queries with respect to finite sets of keys and IDs is undecidable [10]. To this end, Calì, Lembo, and Rosati [11] studied sets of keys and IDs that are *non-key-conflicting*, that is, they "do not conflict" with the keys. An example of such sets are sets of foreign key constraints. The notion of non-key-conflicting IDs has been generalized by Calì, Gottlob, and Lukasiewicz [7] to TGDs. Before we present their generalization, we introduce the following stronger property introduced in [7].

DEFINITION 6.2 ([7]). Let σ be a schema, let Σ_T be a set of TGDs over σ, and let Σ_E be a set of EGDs over σ. We call Σ_E *separable* from Σ_T if for all σ-instances I,

1. If $I \models \Sigma_E$, then $I^{\Sigma_T \cup \Sigma_E}$ is defined.
2. If $I^{\Sigma_T \cup \Sigma_E}$ is defined, then for all Boolean conjunctive queries q we have $I^{\Sigma_T \cup \Sigma_E} \models q$ if and only if $I^{\Sigma_T} \models q$.

PROPOSITION 6.3. *Let σ be a schema, let Σ_T be a set of TGDs over σ, and let Σ_E be a set of EGDs over σ that is separable from Σ_T. Then for all σ-instances I and J such that I is ground we have: J is a universal model of I and $\Sigma_T \cup \Sigma_E$ iff J is a universal model of I and Σ_T, and $J \models \Sigma_E$.*

PROOF. *"Only if":* Suppose J is a universal model of I and $\Sigma_T \cup \Sigma_E$. Then J is a model of I and Σ_T, and $J \models \Sigma_E$. It remains to show that $J \rightarrow K$ for each model K of I and Σ_T. To this end, let q_J be the canonical query of J. Note that $I^{\Sigma_T \cup \Sigma_E}$ is defined. This is an immediate consequence of $J \models \Sigma_E$ and $I \subseteq J$, which imply $I \models \Sigma_E$, and Definition 6.2(1). Since J is a universal model of I and $\Sigma_T \cup \Sigma_E$, we have $J \rightarrow I^{\Sigma_T \cup \Sigma_E}$, and therefore $I^{\Sigma_T \cup \Sigma_E} \models q_J$. Now, property 2 in Definition 6.2 tells us that $I^{\Sigma_T} \models q_J$, that is, $J \rightarrow I^{\Sigma_T}$. Since I^{Σ_T} is a quasi-universal model of I and Σ_T, we conclude that $J \rightarrow K$ for each model K of I and Σ_T, as desired.

"If": Suppose J is a universal model of I and Σ_T, and $J \models \Sigma_E$. Then J is a model of I and $\Sigma_T \cup \Sigma_E$. Since all models of I and $\Sigma_T \cup \Sigma_E$ are models of Σ_T, we also have that $J \rightarrow K$ for all models K of I and $\Sigma_T \cup \Sigma_E$. Hence, J is a universal model of I and $\Sigma_T \cup \Sigma_E$. □

As an immediate consequence of Proposition 6.1 and 6.3, we obtain:

Corollary 6.4. *Let σ be a schema, and let Σ be a finite set of TGDs, EGDs, and negative constraints over σ. Suppose the set of all EGDs in Σ is separable from the set Σ_T of all TGDs in Σ. Then for all ground σ-instances I we have:*

1. *If there is a universal model of I and Σ, then the core model of I and Σ_T satisfies Σ.*
2. *If the core model of I and Σ_T satisfies Σ, then it is a universal model of I and Σ.*

Proof. *Ad 1:* Suppose J is a universal model of I and Σ. Let J^* be the core of J. Then, J^* is a universal model of I and Σ. By Proposition 6.1, J^* is a universal model of I and $\Sigma \setminus \Sigma_N$, and furthermore, $J^* \models \Sigma_N$, where Σ_N is the set of all negative constraints in Σ. Let Σ_E be the set of all EGDs in Σ. Then, by Proposition 6.3, J^* is a universal model of I and $\Sigma_T = (\Sigma \setminus \Sigma_N) \setminus \Sigma_E$, and $J^* \models \Sigma_E$. Altogether, J^* is the core model of I and Σ_T, and $J^* \models \Sigma$, as desired.

Ad 2: Immediately from Propositions 6.1 and 6.3. □

Corollary 6.4 enables us to lift Theorem 5.1 from finite sets of guarded TGDs to finite sets Σ of guarded TGDs, EGDs, and negative constraints such that the set of EGDs in Σ is separable from the set of TGDs in Σ. It implies that to decide whether a given ground instance I has a universal model under such a set Σ, it suffices to do the following:

1. Check whether there is a universal model of I and the set Σ_T of TGDs in Σ (using an algorithm as guaranteed by Theorem 5.1). If so, let J be the core model of I and Σ_T; otherwise reject.
2. If $J \models \Sigma$, then output J; otherwise reject.

Thus, we have:

Theorem 6.5. *There is an algorithm that solves*

Input: *a ground instance I; a finite set Σ of guarded TGDs, EGDs, and negative constraints such that the set of EGDs in Σ is separable from the set of TGDs in Σ*
Task: *Decide whether there is a universal model of I and Σ. If so, compute a core model of I and Σ.*

in time $O(\|I\|^k)$, where k depends only on Σ.

As a sufficient *syntactic* condition for sets of TGDs and *keys* that implies separability, [7] introduces *non-key-conflicting TGDs* which are a generalization of non-key-conflicting IDs from [11]. Recall that a key of a relation R is a set $K \subseteq [\text{ar}(R)]$. A σ-instance I satisfies a key K of R if for every two tuples $\bar{a} = (a_1, \dots, a_{\text{ar}(R)})$ and $\bar{b} = (b_1, \dots, b_{\text{ar}(R)})$ in R^I, where $a_i = b_i$ for all $i \in K$, we have $\bar{a} = \bar{b}$. A key K of R, $r := \text{ar}(R)$, can be written as a set of EGDs, e.g., as

$$\left\{\forall \bar{x}\, (\forall y_i)_{i \in [r] \setminus K}\, \big(R(\bar{x}) \wedge R(\bar{z}) \to x_j = y_j\big) \mid j \in [r] \setminus K\right\}$$

where we let $\bar{x} = (x_1, \dots, x_{\text{ar}(R)})$, and $\bar{z} = (z_1, \dots, z_{\text{ar}(R)})$ with $z_i := x_i$ if $i \in K$, and $z_i := y_i$ otherwise. In the following, we view sets of keys as sets of EGDs.

Definition 6.6 ([7], see revised version). Let K be a key of R, and let θ be a TGD of the form

$$\forall \bar{x} \forall \bar{y}\, \big(\varphi(\bar{x}, \bar{y}) \to \exists \bar{z}\, \big(R_1(\bar{u}_1) \wedge \dots \wedge R_k(\bar{u}_k)\big)\big).$$

K is said to be *non-conflicting* with θ if for all $i \in [k]$ with $R_i = R$ we have:

- the set X_i of positions in $\bar{u}_i$, where a variable from $\bar{x}$ occurs, is not a proper superset of K,
- if $X_i = K$, then each variable from $\bar{z}$ that occurs in $\bar{u}_i$ occurs exactly once in head(θ).

A set Σ_K of keys is non-conflicting with a set Σ_T of TGDs if every key in Σ_K is non-conflicting with every TGD in Σ_T.

In [7] it is shown that if Σ_K is a set of keys that is non-conflicting with a set Σ_T of TGDs, then Σ_K is separable from Σ_T. Thus, Theorem 6.5 leads to a polynomial time algorithm for computing universal models with respect to finite sets of guarded TGDs, keys, and negative constraints, where the keys are non-conflicting with the TGDs.

6.2 Instances with Nulls

Theorem 5.1 can also be extended to certain instances with nulls. It is impossible, however, to obtain a polynomial time algorithm, unless PTIME = NP. Such an algorithm could be easily turned into a polynomial time algorithm to decide whether an undirected graph is 3-colorable:

Proposition 6.7. *There is a schema σ and a set Σ of IDs such that it is* NP*-hard to decide, for a given σ-instance I, whether there is a universal model of I and Σ.*

However, it is possible to obtain a polynomial time algorithm if all *blocks* of the input instance are of constant size.

Definition 6.8 ([18]). The *Gaifman graph of the nulls* of an instance I is the undirected graph G_I whose nodes are all nulls of I, and which has an edge between two nulls $\bot, \bot'$ if $\bot \neq \bot'$ and there is an atom $A \in I$ such that both $\bot$ and $\bot'$ occur in A. A *block* of I is the set of nulls in a connected component of G_I.

For example, the size of blocks of instances that arise in data exchange typically is bounded by a constant [18]. We can now extend the main technical lemma of Section 5, Lemma 5.6, to instances with nulls.

Lemma 6.9. *Let σ be a schema, and let I be a σ-instance whose largest block has size b. let Σ be a finite set of guarded TGDs over σ. If $\lambda(\mathcal{G}_{I,\Sigma})$ has a finite core, then there is a homomorphism from $\lambda(\mathcal{G}_{I,\Sigma})$ to $\lambda(\mathcal{G}^d_{I,\Sigma})$, where d depends only on Σ and b (and can be computed from Σ and b).*

Proof. Let I be a σ-instance with blocks $B_1, \dots, B_n$ such that $|B_i| \leq b$ for all $i \in [n]$. Suppose that $\lambda(\mathcal{G}_{I,\Sigma})$ has a finite core. Let h be a homomorphism from $\lambda(\mathcal{G}_{I,\Sigma})$ to $\lambda(\mathcal{G}_{I,\Sigma})$ such that $J := h(\lambda(\mathcal{G}_{I,\Sigma}))$ is a finite core.

Let $i \in [n]$. We write $I[B_i]$ for the subinstance of I consisting of all atoms of I that contain a null from B_i. Since $h(I[B_i]) \subseteq J \subseteq \lambda(\mathcal{G}_{I,\Sigma})$, we have $h(I[B_i]) \to \lambda(\mathcal{G}_{I,\Sigma})$. Lemma 4.5 implies $h(I[B_i]) \to \lambda(\mathcal{G}^{b \cdot \delta}_{I,\Sigma})$, where δ is as in Lemma 4.4.

We can now use Lemma 5.6 to show that there is a homomorphism h' from $\lambda(\mathcal{G}_{I,\Sigma})$ to $\lambda(\mathcal{G}_{I,\Sigma})$ such that $h'(\lambda(\mathcal{G}_{I,\Sigma}))$ is a core, and the maximum depth of an atom in $h'(\lambda(\mathcal{G}_{I,\Sigma}))$ is at most $b \cdot \delta + d$, where d is as in Lemma 5.6. □

Using Lemma 6.9 and the results from Section 6.1, we obtain that Theorem 6.5 is still true if whenever all blocks in I have size bounded by b, then the constant k depends not only on Σ, but also on b.

7. WEAK UNIVERSAL MODELS

Much like strong universal models play an important role for reasoning over all models, including infinite ones, weak universal models are useful for reasoning over finite models. In this section, we show that for finite sets of guarded TGDs, strong universal models and weak universal models are one and the same concept, whereby proving the analog of Theorem 5.1 for weak universal models.

Equivalence of strong and weak universality under finite sets of guarded TGDs can be inferred rather easily from a recent result by Bárány, Gottlob, and Otto [3] on the finite controllability of query answering over finite sets of guarded FO-sentences. Let me briefly recall this result. *Guarded first logic (guarded* FO*)*, introduced in [2], is a restriction of first-order logic FO. It requires existential and universal quantification to be of the form $\exists\bar{x}\,(\varphi \wedge \psi)$ and $\forall\bar{x}\,(\varphi \rightarrow \psi)$, respectively, where φ is an atomic FO-formula (this includes equality atoms) containing all the variables in $\bar{x}$, and ψ is a guarded FO-formula whose free variables occur in φ. A *guarded* FO*-sentence* is a guarded FO-formula without free variables. For a set Σ of FO-sentences over a schema σ, and a Boolean query q over σ, we write $\Sigma \models q$ if q is true in every possibly infinite σ-instance satisfying Σ, and we write $\Sigma \models_{\text{fin}} q$ if q is true in every *finite* σ-instance satisfying Σ. Now, extending earlier work by Rosati [36], Bárány, Gottlob, and Otto were able to show:

THEOREM 7.1 ([3]). *If Σ is a finite set of guarded* FO*-sentences over a schema σ, and q is a union of conjunctive queries over σ, then $\Sigma \models q$ if and only if $\Sigma \models_{\text{fin}} q$.*

Thanks to Theorem 7.1, it is not hard to prove:

PROPOSITION 7.2. *Let Σ be a finite set of guarded TGDs, and let I, J be instances. Then, J is a weak universal model of I and Σ iff J is a strong universal model of I and Σ.*

PROOF. Since strong universal models are weak, it suffices to show that if J is a weak universal model of I and Σ, then J is strongly universal.

Suppose J is a weak universal model of I and Σ. Then for every finite model K of I and Σ we have $J \to K$. Notice that the finite models K of I and Σ are precisely the finite σ-instances K satisfying $\{q_I\} \cup \Sigma$ (recall from Section 2.1 that q_I denotes the canonical query of I). Hence,

$$\{q_I\} \cup \Sigma \models_{\text{fin}} q_J. \tag{7.1}$$

We would now like to apply Theorem 7.1 to deduce $\{q_I\} \cup \Sigma \models q_J$, proving that J is a strong universal model of I and Σ. However, q_I and the sentences in Σ are not necessarily guarded FO-sentences (q_I is if I is ground).

To this end, we transform q_I into a guarded FO-sentence q'_I as follows. We pick an injective mapping $h\colon \operatorname{dom}(I) \to \mathit{Const}$ that is the identity on $\operatorname{const}(I)$, and maps no null in I to a constant that occurs in J or in Σ. Then we define $q'_I := q_{h(I)}$. Note that q'_I is a guarded FO-sentence, since the instance $h(I)$ is ground, so that $q_{h(I)}$ is quantifier-free. Furthermore, for every possibly infinite σ-instance K with $K \models q'_I$ we have $h(I) \subseteq K$. Therefore, for every such K, there is a homomorphism from I to K, namely h, and since $I \to K$ implies $K \models q_I$, we have

$$q'_I \models q_I. \tag{7.2}$$

Here, we write $q'_I \models q_I$ as abbreviation for $\{q'_I\} \models q_I$.

The next step is to transform Σ into a finite set Σ' of guarded FO-sentences as described in [6, Lemma 10]; see also the description of this transformation in Remark 4.6. By construction,

$$\Sigma' \models \Sigma, \tag{7.3}$$

where we write $\Gamma \models \Gamma'$, for finite sets Γ, Γ' of logical sentences, to express that every possibly infinite instance satisfying Γ also satisfies Γ'.

It now follows from (7.1)–(7.3) that $\{q'_I\} \cup \Sigma' \models_{\text{fin}} q_J$. Hence, Theorem 7.1 yields $\{q'_I\} \cup \Sigma' \models q_J$. But this implies $\{q_I\} \cup \Sigma \models q_J$. To see this, observe that $\{q'_I\} \cup \Sigma' \models q_J$ implies that q_J is true in $h(I)^{\Sigma'}$. Indeed, $h(I)^{\Sigma'}$ is a model of $h(I)$ and Σ'. Therefore, $h(I)^{\Sigma'} \models \{q'_I\} \cup \Sigma'$, and thus, $h(I)^{\Sigma'} \models q_J$. By the choice of h, we have $I^{\Sigma'} \models q_J$, and it is easy to see that this implies $I^{\Sigma} \models q_J$. Therefore, by Theorem 2.3, $\{q_I\} \cup \Sigma \models q_J$. Altogether, this proves that J is a strong universal model of I and Σ. □

Proposition 7.2 still holds in the presence of negative constraints, as introduced in Section 6.1. It is unclear whether it holds for sets of guarded TGDs, negative constraints, and non-conflicting keys. The fact that such constraints can express knowledge bases in the description logic *DL-Lite$_F$* [12], and that there are *DL-Lite$_F$* knowledge bases that only have infinite models [37] implies that, in general, over sets of guarded TGDs, negative constraints, and non-conflicting keys, query answering over finite models differs from query answering in the unrestricted case. However, this does not rule out the possibility that weak universal models and strong universal ones coincide for such sets of constraints. I leave this as an open question.

8. CONCLUDING REMARKS

This paper's main result is an algorithm for computing universal models under finite sets Σ of guarded TGDs, negative constraints, and non-conflicting keys. The algorithm's running time is polynomial if Σ is fixed and the input database has bounded block size:

THEOREM (SUMMARY OF THE MAIN RESULT). There is an algorithm which solves the following problem in time $O(\|I\|^k)$, where k depends only on Σ and the maximum size of a block of I.

Input: an instance I, and a finite set Σ of guarded TGDs, EGDs, and negative constraints such that the set of EGDs in Σ is separable from the set of TGDs in Σ

Task: Decide whether there is a universal model of I and Σ. If so, compute a core model of I and Σ.

The algorithm should be seen as a proof of concept. Very likely, more efficient algorithms exist.

For one thing, the constant d provided by the proof of Lemma 5.6 cannot be bounded by an elementary function, say in the maximum number of universally or existentially quantified variables in a TGD in Σ, since the number of logically non-equivalent $\mathrm{FO}_{q,k}$-types grows non-elementary with q. This leads to a running time which is non-elementary in the size of I and Σ. One can do better here, by replacing $\mathrm{FO}_{q,k}$-types with other notions of type, yielding considerably smaller complexity bounds (a lower bound of 2-EXPTIME follows from results in [7]). Precise bounds for the combined complexity (which considers the set Σ as part of the input) are subject to work in progress.

Let me emphasize that, while the algorithm presented in this paper also computes weak universal models under finite sets of guarded TGDs and negative constraints, the question of how to compute weak universal models in the presence of guarded TGDs, negative constraints, and non-conflicting keys (or just guarded TGDs and non-conflicting keys) is open. Another interesting open question is whether it is possible to compute universal models under sticky sets of TGDs. Sticky sets of TGDs were introduced in a recent paper by Calì, Gottlob, and Pieris [8], where they showed that—just like sets of guarded TGDs—together with non-conflicting keys and negative constraints, they capture certain members of the DL-Lite family of description logics. Sticky sets of TGDs seem to require a completely different machinery than the one used in this paper.

9. REFERENCES

[1] 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.

[2] H. Andréka, I. Németi, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[3] V. Bárány, G. Gottlob, and M. Otto. Querying the guarded fragment. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, 2010.

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

[5] A. Calì, D. Calvanese, G. D. Giacomo, and M. Lenzerini. Data integration under integrity constraints. *Information Systems*, 29(2):147–163, 2004.

[6] A. Calì, G. Gottlob, and M. Kifer. Taming the infinite chase: Query answering under expressive relational constraints. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 70–80, 2008.

[7] A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *Proceedings of the 28th ACM Symposium on Principles of Database Systems (PODS)*, pages 77–86, 2009.

[8] A. Calì, G. Gottlob, and A. Pieris. Advanced processing for ontological queries. *PVLDB*, 3(1):554–565, 2010.

[9] A. Calì, G. Gottlob, and A. Pieris. Query rewriting under non-guarded rules. In *Proceedings of the 4th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW)*, 2010.

[10] A. Calì, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proceedings of the 22th ACM Symposium on Principles of Database Systems (PODS)*, pages 260–271, 2003.

[11] A. Calì, D. Lembo, and R. Rosati. Query rewriting and answering under constraints in data integration systems. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 16–21, 2003.

[12] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[13] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the 9th ACM Symposium on Theory of Computing (STOC)*, pages 77–90, 1977.

[14] S. S. Cosmadakis and P. C. Kanellakis. Functional and inclusion dependencies: A graph theoretic approach. *Advances in Computing Research*, 3:163–184, 1986.

[15] A. Deutsch, A. Nash, and J. Remmel. The chase revisited. In *Proceedings of the 27th ACM Symposium on Principles of Database Systems (PODS)*, pages 149–158, 2008.

[16] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer, 1999.

[17] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[18] R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: Getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, 2005.

[19] F. Geerts and B. Marnette. Static analysis of schema-mappings ensuring oblivious termination. In *Proceedings of the 13th International Conference on Database Theory (ICDT)*, pages 183–195, 2010.

[20] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *J. Comput. Syst. Sci.*, 64(3):579–627, 2002.

[21] G. Gottlob and A. Nash. Efficient core computation in data exchange. *J. ACM*, 55(2):Article 9, 2008.

[22] S. Greco and F. Spezzano. Chase termination: A constraints rewriting approach. *PVLDB*, 3(1):93–104, 2010.

[23] M. Grohe. Logic, graphs, and algorithms. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, volume 2 of *Texts in Logic and Games*, pages 357–422. Amsterdam University Press, 2008.

[24] M. Gyssens. Database dependencies. In *Encyclopedia of Database Systems*, pages 704–708. Springer, 2009.

[25] P. Hell and J. Nešetřil. The core of a graph. *Discrete Mathematics*, 109(1–3):117–126, 1992.

[26] A. Hernich. *Foundations of Query Answering in Relational Data Exchange*. PhD thesis, Goethe-Universität Frankfurt am Main, 2010. Published at Logos Verlag Berlin, ISBN 978-3-8325-2735-8, 2010.

[27] A. Hernich, L. Libkin, and N. Schweikardt. Closed world data exchange. *ACM Trans. Database Syst.*, 36(2):Article 14, 2011.

[28] D. S. Johnson and A. C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. In *Proceedings of the 1st ACM Symposium on Principles of Database Systems*, pages 164–169, 1982.

[29] G. Lausen, M. Meier, and M. Schmidt. On chase termination beyond stratification. *PVLDB*, 2(1):970–981, 2009.

[30] G. Lausen, M. Meier, and M. Schmidt. On chase

termination beyond stratification. *CoRR*, abs/0906.4228, 2009.

[31] M. Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the 21th ACM Symposium on Principles of Database Systems (PODS)*, pages 229–240, 2002.

[32] L. Libkin. *Elements of Finite Model Theory.* Springer, 2004.

[33] D. Maier, A. O. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Trans. Database Syst.*, 4(4):455–469, 1979.

[34] J. A. Makowsky. Algorithmic uses of the Feferman-Vaught Theorem. *Annals of Pure and Applied Logic*, 126(1–3):159–213, 2004.

[35] B. Marnette. Generalized schema mappings: From termination to tractability. In *Proceedings of the 28th ACM Symposium on Principles of Database Systems (PODS)*, pages 13–22, 2009.

[36] R. Rosati. On the decidability and finite controllability of query processing in databases with incomplete information. In *Proceedings of the 25th ACM Symposium on Principles of Database Systems (PODS)*, pages 356–365, 2006.

[37] R. Rosati. Finite model reasoning in *DL-Lite.* In *Proceedings of the 5th European Semantic Web Conference (ESWC)*, pages 215–229, 2008.