The Chase Revisited

Alin Deutsch*
UC San Diego
deutsch@cs.ucsd.edu

Alan Nash†
IBM Almaden
anash3@gmail.com

Jeff Remmel
UC San Diego
remmel@math.ucsd.edu

ABSTRACT

We revisit the classical chase procedure, studying its properties as well as its applicability to standard database problems. We settle (in the negative) the open problem of decidability of termination of the standard chase, and we provide sufficient termination conditions which are strictly less over-conservative than the best previously known. We investigate the adequacy of the standard chase for checking query containment under constraints, constraint implication and computing certain answers in data exchange. We find room for improvement after gaining a deeper understanding of the chase by separating the algorithm from its result. We identify the properties of the chase result that are essential to the above applications, and we introduce the more general notion of an *F-universal model set*, which supports query and constraint languages that are closed under a class F of mappings. By choosing F appropriately, we extend prior results all the way to existential first-order queries and $\forall\exists$-first-order constraints (and various standard sublanguages). We show that the standard chase is incomplete for finding universal model sets, and we introduce the *extended core chase* which is complete, i.e. finds an F-universal model set when it exists. A key advantage of the new chase is that *the same algorithm* can be applied for the mapping classes F of interest, by simply modifying appropriately the set of constraints given as input. Even when restricted to the typical input in prior work (unions of conjunctive queries and embedded dependencies), the new chase supports certain answer computation and containment/implication tests in strictly more cases than the incomplete standard chase.

1. INTRODUCTION

The chase [3, 22, 21, 12, 4, 2] is a fundamental algorithm that has been widely used in databases. Examples of its uses include: (i) checking containment of queries under constraints (which in turn is used in such query rewriting tasks as minimization, rewriting using views, and semantic optimization), (ii) checking implication of constraints, (iii) computing solutions to data exchange problems, and (iv) computing certain answers in data integration settings. The applicability of the same tool to these seemingly different problems is not accidental, and it is due to a deeper, tool-independent reason: to solve these problems, it suffices to exhibit a representative instance U with two key properties, and the chase is an algorithm for finding such an instance. It takes as input an initial instance I and a set of constraints Σ and, if it terminates (which is not guaranteed), its result is a finite instance U satisfying:

(a) U is a model of Σ and I

(we say that U is a model[1] of Σ and I if U satisfies the constraints Σ and there is a homomorphism from I to U), and

(b) U is *universal* for Σ and I: that is, it has a homomorphism into every model of Σ and I.

We call a finite instance with these properties a *universal model* for Σ and I. We can show that the applications above use both properties of a universal model *essentially*, and nothing else (as detailed in Section 3). The common applicability of the chase can therefore be explained simply as follows:

The chase is an algorithm for computing universal models.

A universal model can be thought of as a generalization of the notion of a universal solution used in data exchange, in a sense formalized in Section 3. We do not claim the introduction of the universal model concept as a contribution; rather it is the starting point for our study of the chase. By separating the algorithm from its result, we not only learn more about the power and limitations of the chase, but we also identify extensions (of both algorithm and result) which allow us to handle more expressive classes of constraints and queries. What follows is a study of applications of universal models and the adequacy of the chase as an algorithm to compute them. In particular, we investigate the following questions:

1. When does the chase terminate?
2. Could there be universal models even when the chase does not terminate?
3. Could there be other algorithms that will find them?

The chase has been primarily applied to embedded dependencies and to unions of conjunctive queries. It is hence natural to ask:

4. Can we handle larger classes of queries and constraints?

*Partially funded by NSF grants IIS-0705589, IIS-0415257 and IIS-0347968 and an Alfred P. Sloan fellowship.

†Current affiliation: Tradeworx, 54 Broad St. Suite 200, Red Bank, NJ 07701.

[1] We use the word “model” in this somewhat non-standard way because we do not know of any compact way to denote this relationship between U, Σ, and I. Notice that in the special case where I is empty, this is the standard notion of “model” and notice also that given Σ and I, we can define a set of embedded dependencies Σ_I such that $U \models \Sigma_I$ iff U is a model of Σ and I in our sense.

PODS'08, June 9–12, 2008, Vancouver, BC, Canada.
Copyright 2008 ACM 978-1-60558-108-8/08/06 ...$5.00.

The insights we obtain through our study lead to the natural generalization of universal models to universal model *sets*, and the generalization of universality to mappings other than homomorphisms. These generalizations allow us to apply the same techniques to handle larger classes of queries and constraints, in particular involving disjunction, negation, and inequalities.

Contributions. We show that, given an instance I and a set Σ of embedded dependencies, it is undecidable whether the chase of I with Σ terminates, thus settling a fundamental open problem. In prior work, effectively-checkable conditions on a set of embedded dependencies Σ have been identified which are sufficient for the chase of any instance I with Σ to terminate. Our undecidability result implies that any such conditions will not detect termination in some cases. We provide new sufficient termination conditions which are effectively-checkable and are better than the best previously known ones, "weak acyclicity" [13, 11], in that they fail to detect termination in strictly fewer cases.

We show that universal models formalize in an algorithm-independent fashion the notion of chase result, by proving that all terminating chase sequences produce universal models. This implies that our termination conditions are sufficient for the existence of a universal model. It is natural to ask whether, quite apart from the chase, there are conditions on a set of constraints that are effectively-checkable, sufficient, and necessary for the existence of universal models. We show that no such conditions exists: given set Σ of embedded dependencies and instance I, it is undecidable whether a universal model for Σ and I exists.

This result raises the next-best hope (after decidability): is there a proof procedure for universal models? That is, a procedure which is *complete*, i.e. guaranteed to terminate and find a universal model whenever one exists? A natural first candidate is the standard chase. We show that the standard chase is not complete for finding universal models. We introduce a new kind of chase, which we call the *core chase*, whose step consist of two phases: (a) "firing" all applicable standard chase steps simultaneously, then (b) computing the core of the result. We show that the core chase is complete for finding universal models. In particular, this implies that there are instances of the problems of containment, implication, data exchange and certain answer computation which cannot be solved using the standard chase because it does not terminate, but are solved by the core chase (we exhibit concrete examples).

So far, we have discussed the case of embedded dependencies and conjunctive queries. We consider richer constraint and query languages next. These include disjunction, negation, and inequality, expressing queries by existential first-order formulas and constraints by $\forall\exists$ first-order sentences. Our results extend significantly the prior work on certain answer computation, which tackled primarily the case of unions of conjunctive queries and a restricted kind of embedded dependencies [13, 15]. Our extensions apply also to query containment and constraint implication.

We show that, for more expressive constraints, there are cases when there is no universal model for Σ and I, yet there is a finite set of models which together is universal for Σ and I. We therefore introduce the more general concept of *universal model set*. We next consider notions of universality under mappings other than homomorphisms. Given a class of mappings F, we obtain the notion of *F-universal model set*. By choosing the appropriate class F, we show how to handle general $\forall\exists$ constraints and general existential queries, and their subclasses (such as unions of conjunctive queries with negation or inequality) for the problems mentioned above.

We then turn to the problem of computing F-universal model sets. We show how to extend any homomorphism-based chase that can handle disjunction, to obtain a chase that achieves universality for other mappings while supporting constraints with disjunction, negation and inequality. A key advantage of our extension is that *the same algorithm* yields universality with respect to the various mappings involved. We achieve this by simply modifying the set of constraints. We show that, in particular, the core chase can be extended in this way, and we prove that the extended core chase is complete for finding F-universal model sets. Prior work [8] showed how to chase (incompletely) with larger classes of constraints, but not how to obtain universality for mappings other than homomorphisms. Our results enable the application of the extended core chase instead, which supports certain answers for queries beyond unions of conjunctive queries, and successful containment/implication checks in strictly more cases than handled by the incomplete chase extensions in prior work.

One implication of our contributions so far is that they enable solutions to containment/implication checks and certain answer computation when the standard chase does not terminate. We investigate the potential for solving these problems when not even the extended core chase terminates (or, equivalently by completeness, when no F-universal model set exists). F-universal models and model sets are finite structures, but they have mappings into all models of Σ and I, whether finite or infinite. As explained above, this definition is motivated by the fact that it is precisely what a terminating chase produces. However, we observe that, in many applications (including those described above), it suffices to consider a relaxed notion of F-universality, which we call *weak F-universality*, defined as the existence of F-mappings only into all *finite* models of Σ and I. By definition, every F-universal model is also a weak F-universal model, but we show that the converse is false, even when Σ is a set of embedded dependencies and F is the class of homomorphisms. In the following, we refer to plain universality as *strong*, to emphasize the distinction. We show that whenever the standard or core chase terminate, weak and strong universality coincide. Prior work focuses on the case where the standard chase terminates, which is perhaps why it did not distinguish between weak and strong universality. As we did for strong universal models, we show that it is undecidable whether a weak universal model for Σ and I exists even when Σ is a set of embedded dependencies and the universality is with respect to homomorphisms. This result raises the completeness question. In contrast to the case of strong universality, we prove that there exists no complete procedure for finding weak universal models. Since the core chase finds all weak universal models that are also strong, and since no procedure can find all weak universal models, this prompts the question whether there is at least some incomplete procedure for finding strictly weak (i.e. weak but not strong) universal models. We show that an essential aspect of both the standard and core chase is that strong universality is preserved at every step. The same holds for any variation of the chase which introduces new witnesses in the freest possible way; we call any such algorithm *chase-like*. Therefore, if procedures for finding strictly weak universal models do exist, we need to seek them beyond chase-like algorithms.

Paper Outline. After introducing basic definitions and notation in Section 2, we define universal models and describe their applications in Section 3. We study the termination of the standard chase in Section 4, its adequacy for computing universal models in Section 5, and introduce the core chase in Section 6. Section 7 generalizes universal models to F-universal model sets and shows their applications to richer query and constraint languages. In Section 8, we extend the core chase to compute F-universal model sets. We discuss weak universality in Section 9, related work in Section 10, and conclude in Section 11. Proofs are given in the full version [9].

2. PRELIMINARIES

Basics. A schema σ is a list of constants and relation symbols and their arities. An instance A *over* σ has one relation for every relation symbol in σ, of the same arity. For an instance A, we write $\mathrm{dom}(A)$ for the active 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 *labeled nulls*. If A, B are both over σ, we write $A \subseteq B$ if for every relation symbol $R \in \sigma$, $R^A \subseteq R^B$.

Homomorphisms and Other Mappings. A function $h : \mathrm{dom}(A) \to \mathrm{dom}(B)$ 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. We write $A \to B$ in case there is a homomorphism between A and B. 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.

If $A \to B$ and $B \to A$, we say that A and B are homomorphically equivalent and we write $A \leftrightarrow B$. We extend $\to$ to sets of instances K, L: $K \to L$ iff $(\forall B \in L)(\exists A \in K)(A \to B)$. We call an instance or set of instances T *universal* for K if $T \to K$.

A homomorphism $r : A \to B \subseteq A$ is a *retraction* if r is the identity on $\mathrm{dom}(B)$. In this case we say that A *retracts* to B and that B is a *retract* of A. A retraction is *proper* if it is not surjective. 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 [18] and therefore we can talk about “the” core of A, which we denote $\mathrm{core}(A)$.

Queries. We consider the class CQ of conjunctive queries (with equality) and the class UCQ of unions of conjunctive queries (with equality) 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.

Query Q is *contained in* query P (denoted $Q \sqsubseteq P$) if for every finite instance A, $Q(A) \subseteq P(A)$.

Every conjunctive query Q can be regarded as a symbolic database instance $\mathrm{db}(Q)$, the so-called “frozen instance” or *canonical instance* [7]. $\mathrm{db}(Q)$ is obtained by regarding each atom in Q as a tuple over the domain consisting of Q’s variables and constants. Q’s free variables are treated as *constants* $\bar{c}$.

Constraints. We consider constraints ξ of the form

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

where ϕ and ψ are conjunctions of atoms, which may include equations. Such constraints are known as *embedded dependencies* and are sufficiently expressive to specify all usual integrity constraints, such as keys, foreign keys, inclusion, join, multivalued dependencies, etc. [12, 2]. We call ϕ the *premise* and ψ the *conclusion*. For a given constraint ξ, we write P_ξ for the former and C_ξ for the latter and we write P'_ξ for $\exists \bar{w} P_\xi$ and C'_ξ for $\exists \bar{v} C_\xi$. If ψ consists only of equations, then ξ is an *equality-generating dependency (egd)*. If ψ consists only of relational atoms, then ξ is a *tuple-generating dependency (tgd)*. Every set Σ of embedded dependencies is equivalent to a set of tgds and egds [2]. We write $A \models \Sigma$ if the instance A satisfies all the constraints in Σ. We will extend our treatment to more expressive constraints in Section 7. All sets of constraints we refer to are finite.

We say that query Q is *contained in* query P *under set of constraints* Σ (denoted $Q \sqsubseteq_\Sigma P$) if for every finite instance A such that $A \models \Sigma$, we have $Q(A) \subseteq P(A)$.

Standard Chase. In this section we concentrate on the standard chase with embedded dependencies. For the reader’s convenience, we present a brief overview here (the formal definitions are given in [9]).

A *chase step* $A \xrightarrow{\xi,\bar{a}} B$ takes an instance A on which a tgd or egd ξ fails on $\bar{a}$ (we say that ξ applies to A on $\bar{a}$) and adds some tuples (for tgds) or collapses some elements (for egds) so that $B \models \xi(\bar{a})$. A Σ-*chase sequence* S (or just *chase sequence* if Σ is clear from context) is a sequence of instances $A_0, A_1, \ldots$ such that every instance A_{s+1} in it is obtained from the previous one A_s by a chase step. 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. We will see later that all chase results are homomorphically equivalent, so we can speak about A^Σ (unique up to homomorphic equivalence) without referring to a particular chase sequence.

Data exchange. We consider the setting where we have two schemas σ and τ which do not share any relation symbols. Given an instance S over σ and instance T over τ, 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 Σ over $\sigma \cup \tau$, we say that T is a *solution* for S under Σ if $(S, T) \models \Sigma$. When Σ 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 $\mathrm{dom}(S)$. Equivalently, the values in $\mathrm{dom}(S)$ are seen as constants. A constraint ξ over $\sigma \cup \tau$ is *source-to-target* if the premise of ξ is over σ and the conclusion of ξ is over τ and *target* if the premise and conclusion are both over τ. Among others, we consider the special case $\Sigma = \Sigma_{st} \cup \Sigma_t$ with Σ_{st} a set of source-to-target tgds and Σ_t a set of target tgds and egds. With these restrictions, $(\sigma, \tau, \Sigma_{st}, \Sigma_t)$ is known in the literature [20, 13] as a *data exchange setting*.

Certain Answers. We consider the same setting as for data exchange, adding an r-ary client query Q over the target schema τ. Given source instance S, we are interested in finding the *certain answers* to Q for S under Σ, $\mathrm{cert}^\Sigma_Q(S) = \bigcap_{(S,T) \models \Sigma} Q(T)$.

3. UNIVERSAL MODELS

In this section we define universal models and show how they come into play in several classical database problems.

DEFINITION 1. A *universal model* for a set of embedded dependencies Σ and an instance I is a finite instance U such that:

1. $I \to U$ (we say that U is a model of I)
2. $U \models \Sigma$ (U is a model of Σ), and
3. For every instance A, if $A \models \Sigma$ and $I \to A$ then $U \to A$ (U is universal for Σ and I).

Note that the instances A in point (3) are unrestricted: they can be finite or infinite. We justify this choice in Section 5, where we show that it faithfully captures chase results. In Section 9, we discuss an alternative definition focusing only on finite models.

It is immediate from Definition 1 that any two universal models for Σ are homomorphically equivalent. We write $[I^\Sigma]$ for the universal model for Σ and I (unique up to homomorphism), if one exists. If I is the empty instance $\emptyset$, we write $[\Sigma]$ short for $[\emptyset^\Sigma]$. In our applications, homomorphic equivalence is sufficient; that is, our results hold for any choice of universal model.

Universal models are relevant to a list of database problems, including the following.

Query Containment Under Constraints. A key technique in solving conjunctive query containment is to view a query Q as its canonical instance $\mathrm{db}(Q)$.

We first observe that canonical instances are universal models. 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 $\bar{c}$ (the same fresh constants used for Q s free variables when constructing the canonical instance $\mathrm{db}(Q)$).

LEMMA 1. *If Q is a conjunctive query, then $\mathrm{db}(Q)$ is a universal model for $\Sigma = \{\hat{Q}\}$ and $I = \emptyset$.*

It follows that $[\{\hat{Q}\}]$ always exists, and is homomorphically equivalent to $\mathrm{db}(Q)$. We can therefore restate the homomorphism theorem [7] in terms of universal models:

PROPOSITION 1. *Given $P, Q \in \mathrm{CQ}$, we have*

$$P \sqsubseteq Q \textit{ iff } \mathrm{db}(Q) \to \mathrm{db}(P) \textit{ iff } [\{\hat{Q}\}] \to [\{\hat{P}\}].$$

Containment under constraints similarly reduces to reasoning about universal models.

PROPOSITION 2. *For conjunctive queries P and Q and a set Σ of embedded dependencies, if $[\{\hat{P}\} \cup \Sigma]$ exists, then*

$$P \sqsubseteq_{\Sigma} Q \textit{ iff } [\{\hat{Q}\}] \to [\{\hat{P}\} \cup \Sigma].$$

Constraint Implication. Implication can be reduced to the problem of query containment under constraints and therefore can be solved using universal models. This is due to the well-known observation that every embedded dependency ξ is equivalent (i.e. satisfied by the same instances) to an expression of form $\forall \bar{x}\ P(\bar{x}) \to Q(\bar{x})$ where P and Q are conjunctive queries with equalities [2].

PROPOSITION 3. *Let Σ be a set of embedded dependencies and ξ an embedded dependency written equivalently as $\forall \bar{x}\ P(\bar{x}) \to Q(\bar{x})$ with $P, Q \in \mathrm{CQ}^{=}$. If $[\{\hat{P}\} \cup \Sigma]$ exists, then*

$$\Sigma \models \xi \textit{ iff } P \sqsubseteq_{\Sigma} Q \textit{ iff } [\{\hat{Q}\}] \to [\{\hat{P}\} \cup \Sigma].$$

Universal Solutions. Universal models are slight generalizations of universal solutions in data exchange settings [13], and can be used to compute them. Essentially, a universal solution is the target half of a universal model for the involved source and constraints. More specifically, let $(\sigma, \tau, \Sigma_{st}, \Sigma_t)$ be a data exchange setting where the source-target and target constraints in Σ_{st}, respectively Σ_t are embedded dependencies. If universal model $U = [(S, \emptyset)^{\Sigma_{st} \cup \Sigma_t}]$ exists, (where $(S, \emptyset)$ is a $\sigma \cup \tau$-instance), the restriction $U \mid_{\tau}$ of U to schema τ is a universal solution for S.

Certain Answers. [13] reveals a beautiful connection between the data exchange and the certain answer computation problems: if the client query Q is a union of conjunctive queries and the constraints in Σ are source-to-target and target embedded dependencies, then any universal solution U of the data exchange problem provides a means to compute certain answers: the certain answers to Q are the tuples in $Q(U)$ over the active domain of the source. It follows from the above discussion that certain answers can also be computed from universal models.

PROPOSITION 4. *Consider a data exchange setting $(\sigma, \tau, \Sigma_{st}, \Sigma_t)$ where constraints in $\Sigma_{st} \cup \Sigma_t$ are embedded dependencies. Let S be a source instance and Q a conjunctive query of arity r expressed against schema τ. If $U = [(S, \emptyset)^{\Sigma_{st} \cup \Sigma_t}]$ exists, then*

$$\mathrm{cert}_Q^{\Sigma}(S) = \mathrm{dom}(S)^r \cap Q(U).$$

4. CHASE TERMINATION

In this section, we revisit the common tool for attacking the applications mentioned in Section 3, namely the classical chase procedure (reviewed in Section 2). The chase is successfully applied *provided it terminates*. More specifically, we retrieve classical results on the applications of the chase by substituting in all propositions from Section 3 every occurrence of a universal model $[I^{\Sigma}]$ with the result I^{Σ} of chasing I with Σ. Every test "if $[I^{\Sigma}]$ exists" is replaced by "if the chase of I with Σ terminates, yielding I^{Σ}".

Therefore a fundamental question, of interest independent of universal models, is:

> When does the chase terminate?

Given its importance, it is surprising that this question has not been settled previously. We do so next.

We recall first that when several chase steps apply, the standard chase picks one nondeterministically (as long as fairness is preserved) [2]. Consequently, there are instances and sets of constraints for which certain choices lead to a terminating chase sequence, while others to non-termination [2]. When asking whether the chase terminates, one must therefore refine the question to pertain to the termination of either *some* or *all* chase sequences. It turns out that both refinements are undecidable:

THEOREM 1. *Consider an instance I and a set Σ of tgds.*

1. *It is undecidable whether some chase sequence of I with Σ terminates;*
2. *It is undecidable whether all chase sequences of I with Σ terminate.*

The undecidability holds even over a fixed schema, and even if I is the empty instance.

By Theorem 1, we can not hope for effectively-checkable, sufficient, and necessary conditions for chase termination. A sufficient condition on a set of tgds Σ for the termination of the chase, *weak acyclicity*, was given in [11, 13]. We reproduce the definition in the full version [9], but encourage the reader to abstract from it, focusing on its effect: the chase with weakly acyclic sets of tgds and egds is guaranteed to terminate [11, 13] in PTIME in the size of I (the formal result is recalled for the reader's convenience in Theorem 18 in [9]). We say that a set Σ of tgds and egds is *weakly acyclic* if the set $\Sigma' \subseteq \Sigma$ consisting of the tgds in Σ is weakly acyclic.

We introduce the condition of *stratification* on a set of dependencies, which is also sufficient for termination of the chase and is less restrictive than weak acyclicity (stratification is implied by, yet not equivalent to, weak acyclicity). This new condition is motivated by the following example.

EXAMPLE 1. Consider $\Sigma = \{\xi\}$ where ξ is the following tgd:

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

It is easy to check that Σ is not weakly acyclic, yet it is clear that for any A, A^{Σ} is defined for any chase order since introducing 3-cycles will never create any new 2-cycles. That is, firing ξ will never cause ξ to fire yet again. □

DEFINITION 2. (**Stratified**) Given tgds or egds α and β 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. β does not apply to A on $\bar{b}$, possibly because $\{\bar{b}\} \not\subseteq \mathrm{dom}(A)$,
2. $A \overset{\alpha, \bar{a}}{\to} B$, and

3. β applies to B on $\bar{b}$. That is, $B \not\models \beta(\bar{b})$.

Intuitively, $\alpha \prec \beta$ if firing α may cause β to fire.

The *chase graph* $G(\Sigma)$ of a set of tgds Σ has as vertices the constraints in Σ and there is an edge between two constraints $\alpha, \beta \in \Sigma$ iff $\alpha \prec \beta$. A set of tgds and egds Σ is *stratified* if the set of constraints in every cycle of $G(\Sigma)$ is weakly-acyclic.

THEOREM 2. *For every stratified set Σ of tgds and egds, there are integers b and c upper bounded by the size of Σ such that for every instance A,*

1. *every chase sequence of A with Σ terminates, and*
2. *A^{Σ} can be computed in $O(|A|^b)$ steps and in time $O(|A|^c)$.*

THEOREM 3. *Given tgds α and β, we can check whether $\alpha \prec \beta$ in $\mathbf{NP}$. Therefore, we can check whether Σ is stratified in $\mathbf{coNP}$.*

Note that the stratification check is performed once and for all off-line, when the constraints are declared, so its complexity is immaterial for the run-time response (when the new constraints arrives for testing implication, or the queries arrive for testing containment, or for computing certain answers).

The following result formalizes the sense in which stratification is strictly better than weak acyclicity at avoiding false negatives, i.e. failures on sets of dependencies with which the chase actually terminates.

THEOREM 4. *All weakly-acyclic sets of tgds and egds are stratified, but not conversely.*

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

5. CHASING FOR UNIVERSAL MODELS

We first show that the universal model is the right notion to formalize the chase result in an algorithm-independent fashion, by proving that every terminating chase sequence produces a universal model (Theorem 5).

THEOREM 5. *Let Σ be a set of embedded dependencies and I an instance. If the standard chase of I with Σ terminates, it yields a universal model for Σ and I.*

Theorem 5 and Theorem 2 immediately imply that, quite apart from the chase, stratification is a sufficient existence condition for universal models:

COROLLARY 1. *There exists a universal model $[I^{\Sigma}]$ for every instance I and every stratified set Σ of embedded dependencies.*

Next we investigate whether there are universal model existence conditions that are sufficient, necessary, and efficiently checkable. We answer this question negatively, proving that it is undecidable whether a universal model exists (Theorem 6).

THEOREM 6. *It is undecidable, given an instance I and a set Σ of tgds and egds, whether a universal model for Σ and I exists (even over a fixed schema σ and if $I = \emptyset$).*

Incompleteness of the Standard Chase. Given the undecidability of checking existence of a universal model, we turn to the next-best hope: is there a procedure which is *complete* for finding universal models? That is, whenever a universal model exists, the procedure terminates and finds such a model (possibly diverging otherwise).

A natural first candidate is the standard chase itself. More precisely, we ask: if a universal model for Σ and I exists, will *any* chase sequence of I with Σ terminate, yielding one?

The answer is no, as the following example shows.

EXAMPLE 2. Consider the set Σ consisting of following tgds:

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

The universal model consisting of the self-loop is a universal model for Σ, yet any Σ-chase sequence starting with $\emptyset$ must be infinite. This is because ξ_1 must fire first to give a cycle of length 2. Assume ξ_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 $A_s \overset{\xi_3,a,b}{\rightarrow} A_{s+1}$ where $a \neq b$, then two new edges ac and cb are added to A_{s+1} and that $A_{s+1} \not\models C'_{\xi_3}(ac)$ and $A_{s+1} \not\models C'_{\xi_3}(cb)$. Therefore, $A_{s+1} \not\models \xi_3$, and this leads to an infinite chase sequence. □

6. THE CORE CHASE

We introduce a new chase procedure, the *core chase*, which is complete for finding universal models. To distinguish among the two chase flavors, we refer to the classical chase mentioned so far as *standard chase*. Intuitively, the core chase step proceeds by first firing all applicable standard chase steps simultaneously, then minimizing the resulting instance by computing its core. We formalize the procedure below.

DEFINITION 3. (**Parallel chase step**)
If Σ 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 instance obtained from A by simultaneously firing all applicable standard chase steps. If Σ also contains egds, then we also identify all elements which have been identified by every egd ξ and any tuple $\bar{a}$ such that ξ applies to A on $\bar{a}$.

A core chase step is a parallel chase step, followed by a core computation:

DEFINITION 4. (**Core 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 core chase sequence in the obvious way. Notice that core chase sequences are determined up to isomorphism, since cores are determined up to isomorphism [18] and since at every step all applicable standard chase steps are fired (instead of picking one non-deterministically as in the standard chase). We use the notation A^{Σ} to refer also to the result of a terminating core chase sequence. Such result is unique up to isomorphism.

EXAMPLE 3. Consider the set Σ of tgds from Example 2, for which the standard chase does not terminate. In the first step, constraint ξ_1 fires to give a cycle of length 2 and the minimization step does nothing, since the core of a cycle of length 2 is itself. In the second step, both constraints ξ_2 and ξ_3 fire, the latter twice on the 2-cycle. Now the minimization step reduces everything to the self-loop introduced by ξ_2. At this point, Σ is satisfied and therefore the core chase terminates. □

The core chase is complete for finding universal models. In the result below, termination of the core chase refers to the unique (up to isomorphism) core chase sequence determined by Σ and I.

THEOREM 7. *If I is an instance and Σ is a set of tgds and egds, then there exists a universal model for Σ and I iff the core chase of I with Σ terminates and yields such a model.*

In particular, Theorem 7 implies that there are instances of the problems of containment, implication, data exchange and certain answer computation which cannot be solved using the standard chase because it does not terminate, but are solved by the core chase. We illustrate for the case of containment.

EXAMPLE 4. Consider $\Sigma = \{\xi_2, \xi_3\}$, with ξ_2, ξ_3 from Example 2, and conjunctive queries $Q_1() :- E(x, y), E(y, x)$ and $Q_2() :- E(x, x)$. Similar reasoning as in Example 2 shows that the standard chase of Q_1 with Σ does not terminate. However, $Q_1 \sqsubseteq_\Sigma Q_2$ holds: the core chase of Q_1 with Σ terminates and yields exactly Q_2. □

7. RICHER LANGUAGES

Our results so far pertain to conjunctive queries and embedded dependencies. In this section, we exploit the insights on the chase (obtained by separating the algorithm from its result) to extend the notion of universal model to apply to more expressive query and constraint languages, which include disjunction, negation, and inequalities. It turns out that two extensions are required.

First, we show that there are cases when no single model is universal for a class K of models (and therefore none of our motivating applications can be solved as described in Section 3), yet there is a set of models U such that every model in K has a mapping from some model in U. We call such U a *universal model set*, and show that its existence enables solutions to the problems in the introduction even for constraint sets with no universal model.

The second extension is motivated by the observation that queries and constraints expressed in languages beyond conjunctive queries and embedded dependencies can distinguish among two homomorphically equivalent universal model sets: queries yield distinct answers on them, and so do constraints when viewed as boolean queries. We say that these languages are not closed under homomorphism. It turns out that, to extend the applicability of universal model sets to languages that are not closed under homomorphisms, we need to extend the notion of universality to other kinds of mappings, namely precisely those under which the corresponding languages are closed. We call the model sets that are universal w.r.t. a class F of mappings *F-universal*.

In the remainder of the section, we present the definition and applications of F-universal model sets. We discuss their computation in Section 8.

We start by defining closure under a class of mappings F.

DEFINITION 5. A query Q is closed under a class of mappings F if for every $h \in F$, every instances A, B and every tuple $\bar{a}$,

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

We say that a query language $\mathcal{L}$ is closed under F if every $Q \in \mathcal{L}$ is closed under F.

The following result is part of the folklore. It lists some well-known examples of query languages and mapping classes under which they are closed. We denote with hom,ihom,fhom,emb the class of homomorphisms, injective homomorphisms, full homomorphisms, respectively embeddings.

THEOREM 8.

1. UCQ *and* $\mathrm{Datalog}$ *are closed under* hom.
2. MonQ *is closed under* ihom.
3. $\mathrm{UCQ}^{\neg}$ *is closed under* fhom.
4. $\mathrm{UCQ}^{\neg,\neq}$ *is closed under* emb.

In order to state general results in a simple manner we consider some fixed class of mappings F and we write $A \dashrightarrow_F B$ if there is a mapping $h : A \to B$ such that $h \in F$. We extend $\dashrightarrow_F$ to sets of instances as we did for $\to$. We say that an instance or set of instances T is *F-universal* for K if $T \dashrightarrow_F K$. We say that a class of instances K is *closed* under $\dashrightarrow_F$ if $A \in K, A \dashrightarrow_F B$ imply $B \in K$. Similarly, we say that a set Σ of first-order sentences is *closed* under $\dashrightarrow_F$ if so is $\mathrm{mod}(\Sigma)$, the class of all models of Σ.

DEFINITION 6. A set U of finite instances is an *F-universal model set* for a set of instances K if it satisfies the following conditions:

1. (F-universality) $(\forall M \in K)(\exists T \in U)\ T \dashrightarrow_F M$,
2. (conformance) $U \subseteq K$,
3. (finiteness) U is finite, and
4. (minimality) there is no $U' \subset U$ such that $U' \dashrightarrow_F U$.

Given a set Σ of first-order sentences and an instance I, we say that U is an F-universal model set for Σ and I if it is an F-universal model set for the class of models of both Σ and I.

For conciseness, we refer to model sets that are hom-universal as plain universal.

The F-universal model set for a set of sentences Σ and an instance I is unique up to equivalence under F-mappings (in the sense formalized in Proposition 5 below). We can therefore refer to "the" model set, denoting it as $[I^\Sigma]_F$.

PROPOSITION 5. *If T and T' are both F-universal model sets for a set of instances K, then $|T| = |T'|$ and there is an ordering $T_1, \ldots, T_n$ of the elements of T and an ordering $T'_1, \ldots, T'_n$ of the elements of T' such that $T_i \dashrightarrow_F T'_i$ and $T'_i \dashrightarrow_F T_i$ for $1 \le i \le n$.*

Certain Answers. As reviewed in Section 3, the standard way to compute certain answers to a query Q is to run Q over the universal solution. This was applied in prior work to UCQ queries and constraints given by source-target and target tgds [13]. This approach breaks down as soon as more expressive queries or constraints are considered. As already observed in [13], if the query contains even one inequality, then its result on a universal solution may *strictly contain* the certain answers. Furthermore, it is shown in [15] that, when the constraints in Σ are not only source-to-target, it may be that there exists no universal solution, yet the set of certain answers is non-empty. Looking at constraints beyond the source-to-target class is motivated in [15, 17] 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 [10].

We show here that, while in these settings (and beyond) one cannot use universal solutions to compute certain answers, they are computable from F-universal model sets. Essentially, given query Q closed under class F of mappings, and F-universal model set U, the computation involves taking the intersection of Q's results on each member of U: $\bigcap_{M \in U} Q(M)$. Our results significantly extend prior work on certain answer computation: from unions of conjunctive queries, source-target tgds and target tgds and egds [13], all the way to existential first-order queries and constraints given by universal-existential first-order sentences.

The following example (adapted from [13]) shows that, even in a classical data exchange setting in which the constraints are source-to-target embedded dependencies, if the query contains even one inequality, then even if a universal solution exists, it is inadequate for computing the certain answers. However, it also shows that there is an ihom-universal model set U that suffices for correct computation of certain answers.

EXAMPLE 5. Let the source schema consist of the binary relation symbol E, and the target schema contain binary relations F and 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. Yet according to [13], a universal solution is

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

and the result of Q on T_0 is non-empty (when interpreting labeled nulls as distinct constants as in [13]): $Q(T_0) = \{(a_1, b_2), (a_2, b_1)\}$.

However, using the techniques described in Section 8, we can show that there exists an ihom-universal model set U (universal for injective homomorphisms), which correctly captures the certain answers to Q under the same interpretation of labeled nulls. Indeed, $U = \{U_0 = (S, T_0), U_1 = (S, T_1)\}$. Then $\mathrm{cert}^{\Sigma}_{Q}(S) = Q(U_0) \cap Q(U_1) = \emptyset$ correctly yields the empty set of certain answers.

Notice that there is a homomorphism, but no injective homomorphism, from (S, T_0) into (S, T_1). U_0 is therefore a hom-universal model, but not an ihom-universal model for Σ and S. According to Theorem 9 below, since Q is closed under ihom but not hom, using U_0 alone is inappropriate. □

The following example shows that there are cases when certain answers cannot be computed based on any universal solution because there is none, and yet there is a universal model set enabling the computation. This is illustrated in a standard peer data exchange setting [15], with UCQ queries and constraints expressed as source-to-target and target-to-source tgds.

EXAMPLE 6. Let the source schema and target schema consist of the binary relation symbol E, respectively the quaternary relation symbol 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:

$$\begin{aligned} \xi_{st} : \quad & E(x, y), E(y, z) \rightarrow \exists u \exists w\; F(x, u, z, w) \\ \xi_{ts} : \quad & F(x, u, z, w) \rightarrow E(x, u), E(u, z) \end{aligned}$$

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 Σ. Note that there are infinitely many solutions, since w_1 and w_2 can be replaced by any other 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 hom-universal model set. It turns out (as will be detailed later) that a universal model set U in this setting contains precisely two elements, $U = \{U_1 = (S, T_1), U_2 = (S, T_2)\}$, where w_1, w_2 are interpreted as labeled nulls. It is easy to check that the certain answers to Q can be obtained by computing $Q(U_1) \cap Q(U_2)$. Indeed, $\mathrm{cert}^{\Sigma}_{Q}(S) = Q(U_1) \cap Q(U_2) = \{(a, c)\} \cap \{(a, c)\} = \{(a, c)\}$. □

The next result summarizes the application of F-universal model sets to certain answer computation. It applies to *generalized peer data exchange settings*. Standard peer data exchange [15] strictly generalizes (plain) data exchange [13] by allowing target-source tgds in addition to the source-target tgds and target tgds and egds of data exchange. We generalize peer data exchange settings by allowing constraints expressed by arbitrary $\forall\exists$-sentences over the combined source and target schemas. $\forall\exists$-sentences have the form $\forall\bar{x}\exists\bar{y}\; \phi(\bar{x}, \bar{y})$, with ϕ a quantifier-free first-order formula. Prior work [15] shows how to compute certain answers in peer data exchange for unions of conjunctive queries. This is a particular case of Theorem 9(1) below.

THEOREM 9. *Consider a generalized peer data exchange setting (σ, τ, Σ), with Σ a set of $\forall\exists$-sentences over $\sigma \cup \tau$. Let S be a σ-instance, and Q be a query of arity r over τ. Let U be an F-universal model set for Σ and S, $U = [S^{\Sigma}]_F$. If*

1. $F = \mathsf{hom}$ *and* $Q \in \mathrm{UCQ} \cup \mathrm{Datalog}$*, or*
2. $F = \mathsf{ihom}$ *and* $Q \in \mathrm{MonQ}$*, or*
3. $F = \mathsf{fhom}$ *and* $Q \in \mathrm{UCQ}^{\neg}$*, or*
4. $F = \mathsf{emb}$ *and* $Q \in \mathrm{UCQ}^{\neg,\neq}$*,*

then

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

Query Containment Under Constraints. We generalize Proposition 2, showing how to use F-universal model sets to check containment under constraints for expressive query and constraint languages.

THEOREM 10. *Given a set Σ of $\forall\exists$-sentences, a class F of mappings, and queries P, Q, if*

1. $P, Q \in \mathrm{UCQ}$ *and* $F = \mathsf{hom}$*, or*
2. $P, Q \in \mathrm{MonQ}$ *and* $F = \mathsf{ihom}$*, or*
3. $P, Q \in \mathrm{UCQ}^{\neg}$ *and* $F = \mathsf{fhom}$*, or*
4. $P, Q \in \mathrm{UCQ}^{\neg,\neq}$ *and* $F = \mathsf{emb}$*,*

then $[\{\hat{Q}\}]_F$ exists, and if $[\{\hat{P}\} \cup \Sigma]_F$ exists as well, the following are equivalent:

1. $P \sqsubseteq_{\Sigma} Q$.
2. $(\forall B \in [\{\hat{P}\} \cup \Sigma]_F)\; (\exists A \in [\{\hat{Q}\}]_F)\; A \dashrightarrow_F B$.

Constraint Implication. We extend our consideration of the implication problem to constraints of the form

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

where each ϕ_i and ψ_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 [11]. It is easy to check that every set of $\forall\exists$-sentences is equivalent to a set of NDEDs of form (1), in which $P, Q \in \mathrm{UCQ}^{\neg,\neq}$.

The following corollary of Theorem 10 and Theorem 8 generalizes Proposition 3 from embedded dependencies to NDEDs:

COROLLARY 2. *Given NDED $d := P(\bar{u}) \to Q(\bar{u})$ and set Σ of NDEDs, we have that $[\{\hat{Q}\}]_{\mathsf{emb}}$ exists, and if $[\{\hat{P}\} \cup \Sigma]_{\mathsf{emb}}$ exists as well, the following are equivalent:*

1. *$\Sigma \models d$.*
2. *$(\forall B \in [\{\hat{P}\} \cup \Sigma]_{\mathsf{emb}})\ (\exists A \in [\{\hat{Q}\}]_{\mathsf{emb}})\ A \dashrightarrow_{\mathsf{emb}} B$.*

Partial results on containment and implication are known for several query and constraint languages, based on the standard chase and its extensions [8]. Our contribution in Theorem 10 and Corollary 2 is to provide a general and uniform treatment of these problems by reducing them to finding F-universal model sets for appropriate F. Moreover, the separation from the traditionally used tool (the chase, incomplete already for embedded dependencies), creates the opportunity to improve the prior results by using a complete algorithm instead. We do so in Section 8.

Why standard data exchange admits a universal solution. We have shown that universal model sets apply to generalized peer data exchange settings, in which constraints are more expressive than embedded dependencies and are not necessarily source-to-target. In these settings, it may be that no universal model exists, but there is a universal model set (which is not a singleton). In contrast, in standard data exchange [13], one searches for a universal model (which yields a universal solution as shown in Section 3). It is natural to ask whether, when no universal model exists, one could find a universal model set instead. It turns out that this is not the case: we show next that in standard data exchange settings, the associated hom-universal model set always degenerates to a singleton. This is a basic property of embedded dependencies, independent of the algorithm used to find their universal model set.

PROPOSITION 6. *For any instance I and set Σ of embedded dependencies, if $[I^{\Sigma}]_{\mathsf{hom}}$ exists, it is a singleton.*

COROLLARY 3. *Given standard data exchange setting $(\sigma, \tau, \Sigma_{st}, \Sigma_t)$ and source instance S, if hom-universal model set $U = [S^{\Sigma_{st} \cup \Sigma_t}]_{\mathsf{hom}}$ exists, then U must be a singleton, $U = \{T\}$. Moreover, $T\mid_{\tau}$ is a universal solution.*

8. FINDING F-UNIVERSAL MODEL SETS

In this section, we present a complete algorithm for finding F-universal model sets. A key advantage of our solution is that *the same algorithm* yields universality for various classes of mappings F, by simply taking as input appropriate constraints.

Our result is enabled by two orthogonal contributions. First, we show how to extend any homomorphism-based chase that can handle disjunction to obtain a chase that achieves universality for other mapping classes (though not yet completeness). As a bonus, this same extension allows us to support chasing with constraints that are significantly more expressive than embedded dependencies: NDEDs. Then we show that, in particular, the core chase can be extended in this way, and we prove that the extended core chase is complete for finding F-universal model sets for NDEDs.

We start recalling techniques from our prior work for extending the standard chase to constraints with disjunction and inequality [11] and negation [8]. Our contribution consists in relating these extensions to F-universality, and in combining them with the core chase to obtain completeness.

Chase with DEDs. We recall an extended chase for DEDs introduced in [11]. The definition of the extended chase step parallels Definition 7 in [9], working with *sets* of instances instead. The *extended chase step* $K \overset{\xi,\bar{a}}{\to} L$ (where K and L are finite sets of instances and ξ an NDED of form (1)), yields the set of instances L when ξ applies on $\bar{a}$ to at least one instance in $I \in K$. Every such I is replaced by a set of instances, each extending I to satisfy one disjunct in the conclusion of ξ. The full definition is given in Appendix A of [9].

Chase with DEDFs. We explain next a further extension introduced in [8] to support negation. It extends DEDs with constraints that may have $\bot$ (for "false") as their conclusion. We call such constraints *DEDFs* for *DEDs with falsehood*. A chase step with a DEDF is defined as follows. If ξ has $\bot$ as its conclusion, then

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

That is, if ξ applies to A on $\bar{a}$, it "kills" A, removing it from the result. The definitions of chase sequence, chase result, etc. extend naturally to DEDFs. Note that the chase with DEDFs remains homomorphism-based.

F-Universality. Using appropriate DEDFs, we "trick" the chase with DEDFs into yielding F-universal model sets for NDEDs as follows. Given set Σ of NDEDs over schema σ, we extend σ to $\hat{\sigma} := \sigma \cup \{\hat{R} : R \in \sigma\} \cup \{N\}$. We construct a set of DEDs $\hat{\Sigma}$ by replacing each negated literal $\neg R(\bar{x})$ and inequality $x \neq y$ in Σ with $\hat{R}(\bar{x})$, respectively $N(x, y)$. Notice that the constraints in $\hat{\Sigma}$ are DEDs. Next, we construct a set Λ of DEDFs over $\hat{\sigma}$ such that if the chase of I with the DEDFs from $\hat{\Sigma} \cup \Lambda$ terminates, it yields an F-universal model set for Σ and I. Λ is defined as follows.

(a) If $F \in \mathsf{ihom}$ or N appears in $\hat{\Sigma}$, set Λ to contain the DEDFs

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

(b) If $F \in \mathsf{fhom}$, set Λ to contain all DEDFs 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 $R \in \sigma$ with $\hat{R}$ appearing in $\hat{\Sigma}$.

(c) If $F \in \mathsf{emb}$, since embeddings are precisely full injective homomorphisms, set Λ to the union of the DEDF sets in (a) and (b).

The fact that, when it terminates, the chase of I with $\hat{\Sigma} \cup \Lambda$ yields $[I^{\Sigma}]_F$, follows from the following basic (and chase-independent) property of the involved DEDFs.

THEOREM 11. *Given schema σ, if $F \in \{\mathsf{ihom}, \mathsf{fhom}, \mathsf{emb}\}$, then every instance I over σ has a unique expansion $\hat{I}$ over $\hat{\sigma}$ (i.e. $\hat{I}\mid_{\sigma} = I$) such that $\hat{I} \models \Lambda$ and $\bot \notin \hat{I}$. Moreover, for every set Σ of NDEDs, $I \models \Sigma$ iff $\hat{I} \models \hat{\Sigma}$. Further, for every mapping h and instances A, B, $h : A \dashrightarrow_F B$ iff $h : \hat{A} \to \hat{B}$ is a homomorphism.*

Extended Core Chase. We obtain the *extended core chase* by following each step of the chase with DEDFs by a core computation (see details in Definition 11 in Appendix A of [9]). Notice that the resulting chase remains homomorphism-based. We prove that this chase is complete for F-universal model sets for NDEDs, when using appropriate DEDFs:

THEOREM 12. *If Σ is a set of NDEDs over schema σ and $F \in \{\mathsf{hom}, \mathsf{ihom}, \mathsf{fhom}, \mathsf{emb}\}$, then for $\hat{\sigma}, \hat{\Sigma}$ and Λ constructed as described above, the following are equivalent for every instance I:*

1. *There is an F-universal model set for Σ and I over σ.*
2. *There is a* hom*-universal model set for the class* $K = \{M \mid M$ *over* $\hat{\sigma}, M \models \hat{\Sigma} \cup \Lambda, I \rightarrow M, \bot \notin M\}$.
3. *The extended core chase of I with $\hat{\Sigma} \cup \Lambda$ terminates, and the restriction of its result to σ is $[I^{\Sigma}]_F$.*

9. STRONG AND WEAK UNIVERSALITY

We now investigate the potential of solving such applications as containment/implication tests and certain answer computation when no F-universal model sets exist (or, equivalently by completeness, when the extended core chase does not terminate). For simplicity, we discuss only universal models, but the results carry over in a straightforward fashion to F-universal model sets.

Recall from Definition 1 that a universal model U has homomorphisms into every model A of Σ and I, be it finite or infinite. This definition is motivated by the fact that it captures precisely the chase result (Theorem 5), and it is clearly adequate for resolving problems that involve only finite instances—this is why the chase is useful in databases. However, we note that, for resolving problems that involve only finite instances (as is the case in all database applications), it suffices to produce U which has homomorphisms only into every *finite* model of Σ and I (in the sense that the results in Sections 3 and 7 hold even under this interpretation of universality). We say that such U is *weakly universal* and, to emphasize the distinction, we refer from now on to the plain universality notion of Definition 1 as *strong*. By definition, every strong universal model is also a weak universal model, but (by the following separation theorem) not conversely:

THEOREM 13.
1. *There is a set Σ of tgds which has a strong universal model.*
2. *There is a set Σ of tgds which has a strictly weak universal model (i.e. one that is weak but not strong).*
3. *There is a set Σ of tgds which has no weak universal model.*

This raises the hope of resolving applications with no strong universal model by exhibiting a weak universal model instead, which leads us to study the computation of weak universal models. As in Theorem 6 for strong universal models, we find that checking the existence of a weak universal model is undecidable:

THEOREM 14. *It is undecidable, given a set Σ of tgds and egds and an instance I, whether a weak universal model for Σ and I exists (even over a fixed signature σ and if $I = \emptyset$).*

However, in contrast to strong universal models and the core chase (Theorem 7), we show that there is no complete procedure for weak universal models (Corollary 4 below). This follows from the following stronger result which establishes the exact complexity of checking a given structure for weak/strong universality.

THEOREM 15. *Given a set Σ of tgds and an instance U:*
1. *it is undecidable (in fact, RE-complete) to check whether U is a strong universal model for Σ and*
2. *it is undecidable (in fact, coRE-complete) to check whether U is a weak universal model for Σ.*

Since a complete procedure for finding weak universal models would imply that checking whether an instance U is a weak universal model is in RE (run the procedure and if it terminates, check for homomorphic equivalence with U) and thus decidable by Theorem 15(2), we have the following important corollary.

COROLLARY 4. *There is no complete procedure for finding weak universal models.*

It is natural to wonder whether one can improve on the core chase in the search for weak universal models. Since by Theorem 7 the core chase finds precisely the weak universal models that are also strong, we ask whether there are algorithms for finding strictly weak universal models. Of course, such algorithms are unavoidably incomplete by Corollary 4.

Given that the core chase does not qualify, and neither does the standard chase by Theorem 5, it is natural to wonder about the power of the chase (any chase) towards finding strictly weak universal models. There are several flavors of the chase procedure, even for the case of embedded dependencies. We have already discussed the standard chase and the core chase. There is also the parallel chase in which all constraints which apply are fired simultaneously (but no core is computed). We identify what makes an algorithm "chase-like" and it follows easily that no chase-like algorithm can find strictly-weak universal models. We summarize below what is common to all chase procedures we are aware of. All known chase procedures starting with A produce a sequence of instances $A_0, A_1, \ldots$ with the following properties which are known to be the essential properties of the standard chase. These results are considered folklore or appear implicitly in proofs related to the chase [3, 22, 21, 12, 4, 2].

THEOREM 16. *If Σ is a set of tgds and egds, and $A = A_0, A_1, \ldots$ is a finite or infinite standard chase sequence, then:*

1. $A_0 \rightarrow A_1 \rightarrow A_2 \rightarrow \ldots$
2. *If B is an instance (possibly infinite) and $B \models \Sigma$ and $A \rightarrow B$, then $A_0, A_1, A_2, \ldots \rightarrow B$*
3. *If A^{Σ} is defined, then $A^{\Sigma} \models \Sigma$.*
4. *If $B \models \Sigma$ and there is a homomorphism $h : A_n \rightarrow B$, then there is a homomorphism $h' : A_{n+1} \rightarrow B$ extending h (if ξ is a tgd) or identifying some values in the domain of h (if ξ is an egd).*
5. *If $B \models \Sigma$, A^{Σ} is defined, and there is a homomorphism $h : A \rightarrow B$, then there is a homomorphism $h' : A^{\Sigma} \rightarrow B$ extending h.*

Calling *chase-like* every procedure that produces a sequence of instances satisfying the properties in Theorem 16, we obtain as a corollary of our results so far that (i) the core chase is itself chase-like; (ii) whenever any chase-like procedure terminates on I and Σ, so does the core chase; and (iii) every chase-like procedure produces strong universal models. To find procedures that yield strictly weak universal models, one must therefore consider completely different breeds of algorithms.

One loose end we have left is the following. It is clear from the definition that a strong universal model is also a weak universal model. The same holds for F-universal models sets, but it is less obvious. This is because if T is a strong universal model set, then it clearly satisfies conditions 1, 2, and 3 for K, the set of finite models of Σ. T also satisfies minimality for K, as follows. Assume, to get a contradiction, that $T' \subset T$ is a universal model set for K. Since T consists of finite instances, we must have $T' \rightarrow T$ so T' is also a strong universal model set for Σ, contradicting minimality of T.

10. RELATED WORK

We have already discussed above the line of data exchange work in [13, 15]. In the restricted context of standard peer data exchange, [15] independently introduced the concept of a *universal basis*, which is a particular case of a hom-universal model set restricted to the target schema. The concept was used only to compute certain answers to unions of conjunctive queries when the constraints between the source and the target include a restricted form of disjunctive egds. A result similar to Theorem 14 appeared in [19], which shows undecidability of existence of a solution in data exchange.

Instead of using a universal solution, for restricted classes of queries and constraints, the certain answers of a query Q can also be computed by finding a rewriting of Q over the source [1, 26] (the constraints in [1] are expressed by view definitions). Such rewritings are based explicitly or implicitly on universal solutions. Notable exceptions are [5, 6], which present rewriting algorithms based on the properties of the chase (in general infinite). Rewriting-based results do not readily apply to our more expressive query and constraint classes, for which no rewriting algorithm is known.

The chase was introduced in [22] where its connection to logical implication was established (an early ancestor appeared in [3]). Related formulations of the chase for various kinds of dependencies were introduced in [21, 25]. The chase was extended to embedded dependencies in [4], to include disjunction and inequality in [11], and to arbitrary $\forall\exists$-sentences in [8]. However, all prior chase flavors are incomplete, and they do not preserve universality beyond homomorphisms.

11. CONCLUSION

The common applicability of the standard chase to a list of classical database problems (typically involving unions of conjunctive queries and embedded dependencies) is due to a deeper connection between them: they all can be solved by exhibiting a universal model, and the chase happens to be one (imperfect) algorithm for computing universal models. By separating the algorithm from its result, we gain a deeper understanding that enables us (i) to improve the algorithm (the complete core chase supports certain answer computation and containment/implication checks in strictly more cases than the incomplete standard chase); (ii) to generalize universal models to F-universal model sets which apply to much richer query and constraint languages; and (iii) to extend the core chase to a complete procedure for F-universal model sets.

We also show that the finite (weak) and unrestricted (strong) versions of universality differ in general, but coincide whenever the chase terminates. We show that there is no complete procedure for weak universal models, and that all chase-like algorithms preserve strong universality. Therefore, to find even incomplete procedures for strictly weak universal models, one needs to look beyond chase-like algorithms.

We did not explore the complexity of the core chase. Finding the core of a general structure is NP-complete [7]. However, we hope that the techniques for computing cores of chase results in [14, 16] can be applied to improve efficiency. In addition, we observe that one way to think about the completeness of the chase is as follows. It is easy to check that as soon as a chase sequence produces an instance that is homomorphically equivalent to a model U (which must be universal) of Σ and I, every subsequent instance in that sequence will also be homomorphically equivalent to U. It may be that none of them are themselves models of Σ and I and in this case the chase sequence is non-terminating. However, if the constraints are closed under computing cores, as embedded dependencies are, then as soon as we compute the core of any of these instances, we have a model of Σ and I and the chase sequence terminates. An interesting corollary of this is that we can compute the core arbitrarily infrequently, as long as we keep computing it (i.e. we alternate increasingly long but finite sequences of parallel chase steps with single core chase steps). If we ever enter the homomorphism class of a model, eventually we will detect this once we compute the core. The cost of the core computation can thus be amortized, being spread arbitrarily thin across the chase sequence.

We close with an intriguing connection that offers a characterization of existence of hom-universal model sets. Given a set of instances K, set $\bar{K} := \{B \mid \exists A \in K, A \to B\}$. That is, $\bar{K}$ is the homomorphic closure of K. The following result follows from the infinite and finite versions of the "preservation under homomorphisms" theorems, the latter recently proven by B. Rossman [24].

THEOREM 17.

1. *Σ and I have a strong* hom*-universal model set iff $\bar{K}$ is axiomatizable by a first-order sentence, where K is the set of unrestricted models of Σ and I.*
2. *Σ and I have a weak* hom*-universal model set iff $\bar{K}$ is axiomatizable by a first-order sentence, where K is the set of finite models of Σ and I.*

12. REFERENCES

[1] S. Abiteboul and O. M. Duschka. Complexity of answering queries using materialized views. In *PODS*, 1998.

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

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

[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. *Inf. Syst.*, 29(2), 2004.

[6] A. Calì, D. Lembo, and R. Rosati. Query rewriting and answering under constraints in data integration systems. In *IJCAI*, 2003.

[7] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *STOC*, 1977.

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

[9] A. Deutsch, A. Nash, and J. Remmel. The Chase Revisited (full version). UCSD Tech. Report 2008, http://db.ucsd.edu.

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

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

[12] R. Fagin. Horn clauses and database dependencies. *JACM*, 29(4),'82.

[13] 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).

[14] R. Fagin, P. G. Kolaitis, and L. Popa. Data Exchange: Getting to the Core. In *PODS*, 2003. Full version in TODS, 30(1), 2005.

[15] A. Fuxman, P. G. Kolaitis, R. J. Miller, and W. C. Tan. Peer data exchange. In *PODS*, 2005. Full version in TODS, 31(4), 2006.

[16] G. Gottlob and A. Nash. Data exchange: Computing cores in polynomial time. In *PODS*, 2006.

[17] A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov. Schema Mediation in Peer Data Management Systems. ICDE 2003.

[18] P. Hell and J. Nešetřil. The core of a graph. *Discr. Math.*, 109(1-3):117–126, 1992.

[19] P. G. Kolaitis, J. Panttaja, and W. C. Tan. The complexity of data exchange. In *PODS*, pages 30–39, 2006.

[20] M. Lenzerini. Data Integration: A Theoretical Perspective. In *ACM PODS*, pages 233–246, 2002.

[21] Maier, Sagiv, and Yannakakis. On the complexity of testing implication of functional and join dependencies. *J. ACM*, 1981.

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

[23] A. Nash, A. Deutsch, and J. Remmel. Data exchange, data integration, and the chase. UCSD Tech. Report CS2006-0859, 2006.

[24] B. Rossman. Existential positive types and preservation under homomorphisms. In *LICS*, pages 467–476, 2005.

[25] M. Vardi. Inferring multivalued dependencies from functional and join dependencies. *Acta Informatica*, 1983.

[26] C. Yu and L. Popa. Constraint-Based XML Query Rewriting For Data Integration. In *SIGMOD*, pages 371–382, 2004.