Preserving Constraints with the Stable Chase

David Carral
Center for Advancing Electronics Dresden (cfaed), TU Dresden, Germany
david.carral@tu-dresden.de

Markus Krötzsch
Center for Advancing Electronics Dresden (cfaed), TU Dresden, Germany
markus.kroetzsch@tu-dresden.de

Maximilian Marx
Center for Advancing Electronics Dresden (cfaed), TU Dresden, Germany
maximilian.marx@tu-dresden.de

Ana Ozaki
Center for Advancing Electronics Dresden (cfaed), TU Dresden, Germany
ana.ozaki@tu-dresden.de

Sebastian Rudolph
Computational Logic Group, TU Dresden, Germany
sebastian.rudolph@tu-dresden.de

Abstract

Conjunctive query answering over databases with constraints – also known as (tuple-generating) dependencies – is considered a central database task. To this end, several versions of a construction called *chase* have been described. Given a set Σ of dependencies, it is interesting to ask which constraints not contained in Σ that are initially satisfied in a given database instance are preserved when computing a chase over Σ. Such constraints are an example for the more general class of *incidental constraints*, which when added to Σ as new dependencies do not affect certain answers and might even speed up query answering. After formally introducing incidental constraints, we show that deciding incidentality is undecidable for tuple-generating dependencies, even in cases for which query entailment is decidable. For dependency sets with a finite universal model, the core chase can be used to decide incidentality. For the infinite case, we propose the *stable chase*, which generalises the core chase, and study its relation to incidental constraints.

2012 ACM Subject Classification Theory of computation → Database constraints theory, Theory of computation → Logic and databases

Keywords and phrases Incidental constraints, Tuple-generating dependencies, Infinite core chase, Universal Model, BCQ entailment

Digital Object Identifier 10.4230/LIPIcs.ICDT.2018.12

Acknowledgements This work is partly supported by the German Research Foundation (DFG) in CRC 912 (HAEC) and in Emmy Noether grant KR 4381/1-1.

1 Introduction

The *chase* [7, 14, 23] is an essential family of algorithms used to solve entailment questions in databases in the presence of constraints, such as computing certain answers to queries in data integration scenarios. Given a database instance $\mathcal{I}$ and a set of dependencies Σ, chase procedures compute an instance that extends $\mathcal{I}$ and satisfies all constraints in Σ, and that is *universal* in the sense that it admits a homomorphism into any other model of $\mathcal{I}$ and Σ. In

particular, such a universal model can be used for query answering, as it entails exactly the certain answers to conjunctive queries over $\mathcal{I}$ and Σ.

Now $\mathcal{I}$ might satisfy constraints that are not part of Σ, and it is a relevant question to ask whether or not they are *preserved* by the chase, i.e., whether they still hold in the computed universal model. This can be viewed as an extension of integrity checks to the virtual, possibly infinite views that are defined by a set of dependencies. Moreover, constraints that are preserved in this sense can safely be assumed to hold, and hence be used in algorithms. For instance, query rewriting algorithms can benefit from additional constraints [22, 25].

For the case of Datalog rules (full dependencies) Σ, constraint preservation is a known problem in databases [1, 28], which is typically further generalised by asking if some set of constraints Γ is implied by Σ given arbitrary input instances $\mathcal{I}$ that merely satisfy certain input constraints Γ'. Constraint preservation then is the special case where $\Gamma = \Gamma'$. Traditionally, one is asking which constraints Γ are satisfied in the (unique, finite) least model of Σ, but there have also been works that consider all (first-order) models [29].

Unfortunately, however, these simple notions of constraint preservation (or implication) are no longer meaningful if we consider more general theories Σ that may contain tuple-generating dependencies. Which constraints are preserved then becomes highly sensitive to the details of the chase, since a constraint might be preserved in some universal models of $\mathcal{I}$ and Σ but not in others. It is often possible to preserve a constraint even if it is not logically entailed by $\mathcal{I}$ and Σ. How can we find out if any universal model preserves a particular constraint, and how can we possibly compute such a model? The answer is not obvious, especially in the general case where universal models are necessarily infinite.

To tackle this problem, we propose the notion of *incidental* constraints to capture the intuitive idea of a constraint being "preservable" (possibly with some effort). Concretely, a constraint ρ is incidental for $\mathcal{I}$ and Σ if adding ρ to Σ does not lead to any additional answer to conjunctive queries over $\mathcal{I}$ (and thereby to many other positive queries). Constraints that do not hold in $\mathcal{I}$ may therefore be incidental, too.

We only require conjunctive query equivalence rather than semantic equivalence, since the main use of the chase is positive query answering. Incidentality therefore is not the same as logical entailment. For example, any constraint whose premise is not entailed (as a Boolean conjunctive query, BCQ) is incidental, and is satisfied by all universal models, yet may not be a entailed. Dependencies can be incidental even if violated in some universal models:

▶ **Example 1.** Consider the dependency $\rho = R(x, y) \to \exists z.R(y, z)$ and an instance $\mathcal{I}$ with a single relation $R(n_0, n_1)$ where n_0 and n_1 are nulls. Then $\mathcal{I}$ and ρ has a universal model that is an infinite R-chain, starting at $R(n_0, n_1)$. The dependency $\rho' = R(y, z) \to \exists x.R(x, y)$ is not satisfied in this model, but is incidental for $\mathcal{I}$ and ρ. Indeed, $\mathcal{I}$ and $\{\rho, \rho'\}$ has a universal model that is a two-way infinite R-chain, which entails the same queries as the one-way infinite chain, but is not a universal model of $\mathcal{I}$ and ρ.

We study incidentality and the related problem of recognising incidental constraints. This problems turns out to be hard: it is on the second level of the arithmetic hierarchy, and remains undecidable even in cases where conjunctive query answering is decidable. We give a complete (and computable) characterisation for theories that admit a finite model. Even for cases where a finitary computation procedure is impossible, we seek a deeper understanding of models that preserve incidental constraints. This leads us to develop a new notion of chase, which we use to establish the existence of core models that characterise both BCQ answers and the entailed incidental relationships. In summary, our main contributions are as follows:

- We formalise a new notion of constraint preservation based on incidental dependencies.
- We show that incidentality is not recursively enumerable in general, and remains undecidable even in restricted cases.
- We show that the *core chase* [14] can be used to decide incidentality for cases where a finite universal model exists.
- We develop the *stable chase* as a generalisation of the core chase to the infinite case.
- We show that the stable chase produces a core that can be used both for query answering and for characterising full incidental dependencies.

Finally, we combine our results to establish the existence of a model that entails the same queries as a universal model and that satisfies exactly the tuple-generating depenencies that are incidental. This model can no longer be universal, but it is a core.

2 Preliminaries

We consider countably infinite, disjoint sets of *constants* Δ_c and of *nulls* Δ_n. A *schema* $\mathcal{S}$ is a finite set of relation symbols, where $\mathsf{ar}(R)$ is the *arity* of $R \in \mathcal{S}$. An *instance* $\mathcal{I}$ over $\mathcal{S}$ assigns to each relation symbol $R \in \mathcal{S}$ a (possibly infinite) $\mathsf{ar}(R)$-ary relation $R^{\mathcal{I}}$ over $\Delta_c \cup \Delta_n$. We often refrain from mentioning the schema $\mathcal{S}$ of an instance $\mathcal{I}$ explicitly, assuming that such a signature has been fixed. The *active domain* of $\mathcal{I}$, denoted by $\Delta^{\mathcal{I}}$, is the set of all *domain elements* that occur in relations of $\mathcal{I}$. We write $\boldsymbol{a}$ for a tuple $\langle a_1, \dots, a_n \rangle$ of domain elements.

Morphisms. Let $\mathcal{I}$ and $\mathcal{J}$ be instances over a schema $\mathcal{S}$. A *homomorphism* $h : \mathcal{I} \to \mathcal{J}$ is a function from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{J}}$ such that (i) $h(c) = c$ for all $c \in \Delta^{\mathcal{I}} \cap \Delta_c$, and (ii) $\boldsymbol{a} \in R^{\mathcal{I}}$ implies $h(\boldsymbol{a}) \in R^{\mathcal{J}}$ for all $R \in \mathcal{S}$ and $\boldsymbol{a} = \langle a_1, \dots, a_n \rangle \in (\Delta^{\mathcal{I}})^{\mathsf{ar}(R)}$, where $h(\boldsymbol{a})$ is short for tuple $\langle h(a_1), \dots, h(a_n) \rangle$. It is *strong* if (ii) is strengthened to require $\boldsymbol{a} \in R^{\mathcal{I}}$ if and only if $h(\boldsymbol{a}) \in R^{\mathcal{J}}$.[1] An *embedding* is an injective strong homomorphism, and an *isomorphism* is a bijective strong homomorphism (i.e., a surjective embedding). An *endomorphism* of $\mathcal{I}$ is a homomorphism $h : \mathcal{I} \to \mathcal{I}$.

Dependencies and Queries. We use a countably infinite set Δ_v of *variables*, disjoint from $\Delta_c \cup \Delta_n$. A *term* is an element $t \in \Delta_v \cup \Delta_c$. We use letters x, y, z, u, v, w and expressions such as $\boldsymbol{x}$ for tuples $\langle x_1, \dots, x_\ell \rangle$ of the corresponding elements. We treat such tuples as sets when order is not relevant. An *atom* is a formula $R(\boldsymbol{t})$ with $R \in \mathcal{S}$ and $|\boldsymbol{t}| = \mathsf{ar}(R)$. First-order formulae are defined as usual. We write $\varphi[\boldsymbol{x}]$ to emphasise that the free variables in φ are a subset of $\boldsymbol{x}$. A *tuple generating dependency* (*TGD*) is a formula of the form

$$\forall \boldsymbol{x}, \boldsymbol{z}.(\varphi[\boldsymbol{x}, \boldsymbol{z}] \to \exists \boldsymbol{y}.\psi[\boldsymbol{x}, \boldsymbol{y}]) \tag{1}$$

where the *body* φ and the *head* ψ are conjunctions of atoms, and ψ contains at least one conjunct. TGDs never contain free variables, hence we usually omit the universal quantifiers. A TGD is *full* if it does not contain existentially quantified variables. A *Boolean conjunctive query* (*BCQ*), or simply a *query*, is a formula of the form $\exists \boldsymbol{y}.\varphi[\boldsymbol{y}]$ with φ a conjunction of atoms. We allow TGDs with empty bodies to assert facts (possibly with existentials), and we often omit $\to$ in this case. We generally assume that Σ denotes a *finite* set of TGDs.

A conjunction of atoms φ (resp. a BCQ $q = \exists \boldsymbol{y}.\varphi[\boldsymbol{y}]$) gives rise to a finite instance $\mathcal{I}_\varphi$ (resp. $\mathcal{I}_q$), obtained by treating φ as a set of relational tuples using a fresh null n_x in place

[1] Strong homomorphisms were called *full* by Deutsch et al. [14].

of each variable x. Conversely, every finite instance $\mathcal{I}$ induces a conjunction $\varphi_{\mathcal{I}}$ that has an atom for every relational tuple, using fresh variables x_n in place of nulls n. The BCQ $q_{\mathcal{I}}$ then is the existential closure of $\varphi_{\mathcal{I}}$. Note that TGDs can encode a given finite instance $\mathcal{I}$ using a dependency $\to q_{\mathcal{I}}$. This is why we will generally state our results for sets Σ of TGDs without mentioning an additional instance.

Universal Models and Cores. Instances naturally correspond to first-order interpretations. We let $\models$ denote first-order modelhood and entailment. Note that, for an instance $\mathcal{I}$ and a finite instance $\mathcal{J}$, we have $\mathcal{I} \models q_{\mathcal{J}}$ iff there exists a homomorphism $h : \mathcal{J} \to \mathcal{I}$. The set of all BCQs modelled (entailed) by an interpretation $\mathcal{I}$ or a set of TGDs Σ is denoted with $\mathsf{BCQ}(\mathcal{I})$ and $\mathsf{BCQ}(\Sigma)$, respectively. A model $\mathcal{J} \models \Sigma$ is *universal* if, for every model $\mathcal{I} \models \Sigma$, there is a homomorphism $h : \mathcal{J} \to \mathcal{I}$. In this case, $\mathsf{BCQ}(\mathcal{J}) = \mathsf{BCQ}(\Sigma)$, i.e., $\mathcal{J}$ and Σ are *BCQ-equivalent* [14]. Two instances $\mathcal{I}$ and $\mathcal{J}$ are *BCQ-equivalent* if $\mathsf{BCQ}(\mathcal{I}) = \mathsf{BCQ}(\mathcal{J})$.

▶ **Definition 2.** An instance $\mathcal{I}$ is a *core* if every endomorphism of $\mathcal{I}$ is an embedding. A core $\mathcal{I}$ is called *a core of* $\mathcal{J}$ if there is an endomorphism h of $\mathcal{J}$ such that $\mathcal{I}$ is the restriction of $\mathcal{J}$ to the image of h.

Definition 2 corresponds to Bauslaugh's property **IN** [5], and has also been used, e.g., in studies of constraint satisfaction [8]. Bauslaugh favours a stronger definition based on isomorphisms instead of endomorphisms (property **ISN**), but this forces cores to be unique up to isomorphism, which is too restrictive for our needs. There are several further definitions of cores, all of which differ only on infinite instances [5, 6]. For finite instances, Definition 2 agrees with the one in [14] and a unique core (up to isomorphism) always exists, whereas (for Definition 2) infinite instances may have no core or several cores (see examples in Section 4).

Applying Rules. A TGD ρ as in (1) is *applicable* to an instance $\mathcal{I}$ if there is a homomorphism $h : \mathcal{I}_\varphi \to \mathcal{I}$. We then extend h to $\mathcal{I}_\psi$ by defining, for all variables $y \in \boldsymbol{y}$ that are existentially quantified, $h(n_y) = n_{y,\rho,h}$ to be a null that is specific for y, ρ, and h, where we assume that all nulls of the form $n_{y,\rho,h}$ exist and are mutually distinct. Let $\rho(\mathcal{I})$ denote the union of $\mathcal{I}$ with all sets of the form $h(\mathcal{I}_\psi)$ for some extended homomorphism $h : \mathcal{I}_\varphi \to \mathcal{I}$. For a set Σ of TGDs, we set $\Sigma(\mathcal{I}) = \bigcup_{\rho \in \Sigma} \rho(\mathcal{I})$.

3 Incidental Dependencies

It is intuitive to ask whether a dependency ρ that holds for a finite instance $\mathcal{I}$ is "preserved" by a given set Σ of TGDs. We formalise this as follows, where we omit $\mathcal{I}$ since it can be captured by a TGD in Σ:

▶ **Definition 3.** A TGD ρ is *incidental* for a set Σ of TGDs if $\mathsf{BCQ}(\Sigma) = \mathsf{BCQ}(\Sigma \cup \{\rho\})$. The set of all incidental TGDs of Σ is denoted $\mathsf{ICDT}(\Sigma)$.

Clearly, $\Sigma \subseteq \mathsf{ICDT}(\Sigma)$. Indeed, every TGD that is logically entailed is also incidental. However, the converse is not true, as illustrated in Example 1 (where the instance $\mathcal{I}$ can be expressed by a TGD $\to \exists x, y.R(x, y)$). In particular, incidental TGDs are not automatically "preserved" in an arbitrary chase procedure, hence we avoid this terminology, though it was used previously, e.g., related to constraint preservation under non-recursive full TGDs [28].

Note that our notion of incidental TGDs is not specific to BCQs. Indeed, BCQ-equivalent sets of TGDs are also equivalent with respect to many other types of negation-free queries, such as Datalog queries and its numerous fragments, including (unions of) conjunctive regular

path queries [18, 11], monadic [12] and linear [19] Datalog queries, (nested) monadically defined queries [27, 9] and many more. Queries with negation, however, are not preserved: in Example 1, $\Sigma \models \exists x.\forall y.\neg R(y,x)$ whereas $\Sigma \cup \{\rho\} \not\models \exists x.\forall y.\neg R(y,x)$.

A noteworthy property is that a TGD is incidental exactly if it does not lead to a newly entailed BCQ in a single derivation step. To state this formally, we use $\rho(p)$ to abbreviate the BCQ $q_{\rho(\mathcal{I}_p)}$ for any BCQ p and TGD ρ.

▶ **Theorem 4.** *For a TGD ρ and a set Σ of TGDs, $\rho \in \mathsf{ICDT}(\Sigma)$ iff $\rho(q) \in \mathsf{BCQ}(\Sigma)$ for every $q \in \mathsf{BCQ}(\Sigma)$.*

Proof. ($\Rightarrow$) For the contrapositive, assume that $q \in \mathsf{BCQ}(\Sigma)$ and $\rho(q) \notin \mathsf{BCQ}(\Sigma)$. Then ρ is applicable to $\mathcal{I}_q$. Let $\mathcal{J}$ be any model of Σ. Since $\mathcal{J} \models q$, there is a homomorphism $\mathcal{I}_q \to \mathcal{J}$, hence ρ is applicable to $\mathcal{J}$. Therefore, any instance $\mathcal{J}' \supseteq \mathcal{J}$ that satisfies ρ entails $\rho(q)$. Since $\mathcal{J}$ was arbitrary, we find that $\Sigma \cup \{\rho\} \models \rho(q)$. Hence ρ is not incidental for Σ.

($\Leftarrow$) Assume that $\rho(q) \in \mathsf{BCQ}(\Sigma)$ for all $q \in \mathsf{BCQ}(\Sigma)$. Suppose for a contradiction that $\Sigma \cup \{\rho\} \models q$ for some $q \notin \mathsf{BCQ}(\Sigma)$. Then there is a finite derivation $\mathcal{I} = \rho_k(\rho_{k-1}(\dots\rho_0(\mathcal{I}_\emptyset)\dots))$ with $\mathcal{I}_\emptyset$ the empty instance and rules $\rho_i \in \Sigma \cup \{\rho\}$, such that $\mathcal{I} \models q$. Let q w.l.o.g. be such that k is minimal. Then $\Sigma \models q_\mathcal{J}$ for $\mathcal{J} = \rho_{k-1}(\dots\rho_0(\mathcal{I}_\emptyset)\dots)$, and we have $\rho_k = \rho$. By the assumption on ρ, also $\Sigma \models \rho(q_\mathcal{J})$. Therefore $\Sigma \models q_\mathcal{I}$, since $\Sigma \models q_\mathcal{I} = q_{\rho(\mathcal{J})}$ is isomorphic to $\rho(q_\mathcal{J})$. Hence, since $\mathcal{I} \models q$ implies $q_\mathcal{I} \models q$, we get the desired contradiction $\Sigma \models q$. ◀

An important insight from the preceding theorem is that incidentality for some set Σ can be established solely on $\mathsf{BCQ}(\Sigma)$.

▶ **Lemma 5.** *For every two BCQ-equivalent sets Σ, Σ' of TGDs, $\mathsf{ICDT}(\Sigma) = \mathsf{ICDT}(\Sigma')$.*

Proof. Let $\rho \in \mathsf{ICDT}(\Sigma)$ be an incidental TGD for Σ. Then, by Theorem 4, $\rho(q) \in \mathsf{BCQ}(\Sigma)$ for every $q \in \mathsf{BCQ}(\Sigma)$. Due to BCQ equivalence, this means $\rho(q) \in \mathsf{BCQ}(\Sigma')$ for every $q \in \mathsf{BCQ}(\Sigma')$, which, by the other direction of Theorem 4, implies that $\rho \in \mathsf{ICDT}(\Sigma')$. The converse follows by symmetry. ◀

One of the uses of this result is to show the following theorem, which asserts that individual incidental TGDs are also jointly incidental, i.e., do not entail any additional BCQs together.

▶ **Theorem 6.** *For every set Σ of TGDs, $\mathsf{BCQ}(\Sigma) = \mathsf{BCQ}(\mathsf{ICDT}(\Sigma))$.*

Proof. Let $q \in \mathsf{BCQ}(\mathsf{ICDT}(\Sigma))$ be a BCQ. Then, by compactness, there is a finite subset $\Gamma = \{\gamma_1, \dots, \gamma_k\} \subseteq \mathsf{ICDT}(\Sigma)$ such that $q \in \mathsf{BCQ}(\Sigma \cup \Gamma)$. But then $\mathsf{BCQ}(\Sigma) = \mathsf{BCQ}(\Sigma \cup \{\gamma_1\}) = \mathsf{BCQ}(\Sigma \cup \{\gamma_1, \gamma_2\}) = \dots = \mathsf{BCQ}(\Sigma \cup \Gamma)$: Since γ_1 is incidental for Σ, we have $\mathsf{BCQ}(\Sigma) = \mathsf{BCQ}(\Sigma \cup \{\gamma_1\})$. By Lemma 5, γ_2 is incidental for $\Sigma \cup \{\gamma_1\}$, i.e., $\mathsf{BCQ}(\Sigma \cup \{\gamma_1\}) = \mathsf{BCQ}(\Sigma \cup \{\gamma_1, \gamma_2\})$. Further applications of Lemma 5 show that γ_k is incidental for $\Sigma \cup \{\gamma_1, \dots, \gamma_{k-1}\}$, yielding the above equality. This shows $q \in \mathsf{BCQ}(\Sigma)$. Hence, $\mathsf{BCQ}(\mathsf{ICDT}(\Sigma)) \subseteq \mathsf{BCQ}(\Sigma)$, and by monotonicity, we also have $\mathsf{BCQ}(\Sigma) \subseteq \mathsf{BCQ}(\mathsf{ICDT}(\Sigma))$. ◀

▶ **Definition 7.** Incidental is the following decision problem. Given a set Σ of TGDs and a TGD ρ, is ρ incidental for Σ?

Since BCQ entailment checking over a set of TGDs is undecidable, it is not surprising that the same is true for Incidental. However, the problem is actually on the second level of the arithmetic hierarchy [26], i.e., strictly harder than query answering, and neither incidental dependencies nor non-incidental dependencies can be recursively enumerated (are in RE):

▶ **Theorem 8.** INCIDENTAL *is* Π^0_2*-complete, and in particular neither in* RE *nor in* coRE.

Proof. For membership note that we can characterise incidentality by quantifying over (finite) derivations (or proofs) in some theory. Indeed, a TGD ρ is incidental for Σ if: for all derivations that show $\Sigma \cup \{\rho\} \models q$ for some BCQ q, there is a derivation that shows $\Sigma \models q$. Using Gödel numbers for representing derivations, this condition can be expressed as a $\forall\exists$-sentence in first-order arithmetic.

We show hardness by many-one reduction from the *universal halting problem*, which is as follows: given a (deterministic) Turing machine $\mathcal{M}$, does $\mathcal{M}$ halt on all inputs? Universal halting is known to be complete for Π^0_2 (see [26, Theorem VIII], and apply Post's Theorem).

For the reduction, we construct for a given TM $\mathcal{M}$ a set $\Sigma_\mathcal{M}$ of TGDs and a full TGD ρ such that ρ is incidental for Σ iff $\mathcal{M}$ halts universally. The rules of $\Sigma_\mathcal{M}$ consist of three parts: Σ_1 ensures that each model contains representations of all possible inputs; Σ_2 simulates $\mathcal{M}$ on a particular input; Σ_3 marks elements of an accepting TM simulation with a specific unary relation halted. The rule ρ then asserts that initial elements in TM simulations are always marked by halted, which is incidental if all runs have indeed terminated. The detailed constructions in each case are given in the appendix. ◀

There are many known classes of TGD sets for which query answering becomes decidable, such as acyclic TGDs or guarded TGDs [15, 16, 2, 3, 21, 13], and INCIDENTAL is indeed in coRE in this case.

▶ **Theorem 9.** *Let $\mathcal{C}$ be a class of sets of TGDs over which BCQ entailment is decidable. There is an algorithm that, given $\Sigma \in \mathcal{C}$, enumerates all TGDs ρ such that $\rho \notin \mathsf{ICDT}(\Sigma)$.*

Proof. Let ρ be an arbitrary TGD. Then ρ is non-incidental iff there is some BCQ q such that either $\Sigma \models q$ but $\Sigma \cup \{\rho\} \not\models q$, or $\Sigma \not\models q$ but $\Sigma \cup \{\rho\} \models q$. Due to monotonicity of TGDs, only the second case can occur. Now, enumerating all q such that $\Sigma \not\models q$ and checking $\Sigma \cup \{\rho\} \models q$ yields a semi-decision procedure for non-incidentality. Using a suitable diagonalisation, we can enumerate all $\rho \notin \mathsf{ICDT}(\Sigma)$. ◀

By Theorem 9, establishing non-incidentality of a given rule ρ is in RE, even if $\Sigma \cup \{\rho\} \notin \mathcal{C}$. On the other hand, INCIDENTAL in general remains undecidable even if BCQ-entailment is decidable, and even when asking only for the incidentality of one fixed full dependency.

▶ **Theorem 10.** *There is a class $\mathcal{C}$ of sets of TGDs for which BCQ answering is decidable, and a full dependency ρ for which $\Sigma \cup \{\rho\} \in \mathcal{C}$ for all $\Sigma \in \mathcal{C}$, such that the following problem is undecidable: given some $\Sigma \in \mathcal{C}$, is ρ incidental for Σ?*

Proof. We show undecidability by reducing the halting problem of deterministic Turing machines when started on the empty tape. Consider a Turing machine $\mathcal{M} = \langle Q, \Gamma, \delta, q_s, q_e \rangle$ as in the proof of Theorem 8, which w.l.o.g. does not return to its initial state q_s in any run. We consider predicate symbols as used in the proof of Theorem 8, and define the set $\tau(\mathcal{M})$ of TGDs to contain the rules Σ_2 as in this proof, together with the additional rules (facts):

$$\to \exists v, w.\mathsf{head}_{q_s}(v) \wedge \mathsf{symbol}_\square(v) \wedge \mathsf{right}(v, w) \wedge \mathsf{symbol}_\square(w) \wedge \mathsf{end}(w) \tag{2}$$

$$\to \exists v.\mathsf{right}(v, v) \wedge \mathsf{right}^+(v, v) \wedge \mathsf{next}(v, v) \wedge \mathsf{end}(v) \wedge \bigwedge_{q \in Q \setminus \{q_s\}} \mathsf{head}_q(v) \wedge \bigwedge_{\sigma \in \Gamma} \mathsf{symbol}_\sigma(v) \tag{3}$$

Here, (2) encodes the initial configuration of $\mathcal{M}$ on the empty tape, which is the start of a Turing machine simulation as effected by Σ_2; and (3) creates an element that stands in

all possible relations not involving head_{q_s}. Let $\rho = \mathsf{head}_{q_e}(x) \to \mathsf{halted}(x)$, and let $\mathcal{C}$ be the class of all TGD sets of the form $\tau(\mathcal{M})$ or $\tau(\mathcal{M}) \cup \{\rho\}$.

BCQ answering over TGD sets of $\mathcal{C}$ is decidable. Indeed, any BCQ that does not contain head_{q_s} is trivially entailed by any $\Sigma \in \mathcal{C}$, due to (3). On the other hand, if a connected component in a BCQ contains head_{q_s}, then it describes a property of a finite initial segment of the simulation of a TM, which can be checked effectively.

For a Turing machine $\mathcal{M}$, the full TGD ρ is incidental for $\tau(\mathcal{M})$ iff $\mathcal{M}$ does *not* halt on the empty input. Indeed, if $\mathcal{M}$ does not halt, then the only occurrence of head_{q_e} in a universal model of $\tau(\mathcal{M})$ is in the element created due to (3), and ρ is already satisfied by this element. Conversely, if $\mathcal{M}$ halts, then head_{q_e} occurs for an element that is connected to the starting sequence a created due to (2). Hence, there is a BCQ of the form $q = \exists \boldsymbol{x}.\mathsf{head}_{q_s}(x_0) \wedge p_1(x_0, x_1) \wedge \ldots \wedge p_n(x_{n-1}, x_n) \wedge \mathsf{halting}(x_n)$ with $p_i \in \{\mathsf{right}, \mathsf{next}\}$, such that $\tau(\mathcal{M}) \not\models q$ and $\tau(\mathcal{M}) \cup \{\rho\} \models q$. ◀

The previous result is particularly interesting since it only considers situations where query answering is decidable, both for the TGDs with and without the candidate dependency ρ. In spite of this general result, concrete classes of TGD sets with decidable BCQ entailment may allow us to decide Incidental, as discussed in the next section.

4 Cores and Incidentals

In this section we relate incidental dependencies with the notion of a core of an instance. Theorem 11 shows that if a set of TGDs has a finite universal model $\mathcal{I}$ then all incidental dependencies follow from the core of $\mathcal{I}$. It then follows from Theorem 11 that if the core chase [14] (also, see Definition 19) terminates then Incidental is decidable. In the following, let $\mathsf{core}(\mathcal{I})$ denote the core of a finite instance $\mathcal{I}$.

▶ **Theorem 11.** *Let Σ be a set of TGDs with a finite universal model $\mathcal{I}$ and let ρ be a TGD. Then, $\rho \in \mathsf{ICDT}(\Sigma)$ iff $\mathsf{core}(\mathcal{I}) \models \rho$.*

Proof. ($\Rightarrow$) Consider $\rho = \varphi[\boldsymbol{x}, \boldsymbol{z}] \to \exists \boldsymbol{y}.\psi[\boldsymbol{x}, \boldsymbol{y}]$ with $\rho \in \mathsf{ICDT}(\Sigma)$. Let $h : \mathcal{I}_\varphi \to \mathsf{core}(\mathcal{I})$ be some homomorphism, and assume that it is extended to $\mathcal{I}_\psi$ using new nulls to map to as defined before. Then set $\mathcal{J} := \mathsf{core}(\mathcal{I}) \cup h(\mathcal{I}_\psi)$, i.e. the core with the consequence of ρ under h added (possibly by adding new elements). $\mathcal{J}$ is finite since $\mathsf{core}(\mathcal{I})$ is, and clearly $\rho(\mathsf{core}(\mathcal{I})) \models q_\mathcal{J}$. Therefore $\Sigma \cup \{\rho\} \models q_\mathcal{J}$, and hence $\Sigma \models q_\mathcal{J}$ by incidentality. So $\mathsf{core}(\mathcal{I}) \models q_\mathcal{J}$ since $\mathsf{core}(\mathcal{I})$ is a universal model, and we obtain a homomorphism $g : \mathcal{J} \to \mathsf{core}(\mathcal{I})$. But then the restriction of g to elements of $\Delta^{\mathsf{core}(\mathcal{I})}$ is an endomorphism, and therefore an embedding since $\mathsf{core}(\mathcal{I})$ is a core. Every embedding on a finite core is an isomorphism [20, 5], so g has an inverse $g^- : \mathsf{core}(\mathcal{I}) \to \mathsf{core}(\mathcal{I})$. For $\mathcal{K} = g(h(\mathcal{I}_\psi \cup \mathcal{I}_\psi))$ we have $\mathcal{K} \subseteq \mathsf{core}(\mathcal{I})$ and hence $g^-(\mathcal{K}) \subseteq \mathsf{core}(\mathcal{I})$. Since $g^-(g(h(\mathcal{I}_\psi))) = h(\mathcal{I}_\psi)$, we can find a homomorphism h' such that $h'(\mathcal{I}_\varphi) = h(\mathcal{I}_\varphi)$ and $h'(\mathcal{I}_\psi) \subseteq g^-(\mathcal{K}) \subseteq \mathsf{core}(\mathcal{I})$ (h' may differ from h in the choice of null values for existentially quantified variables). This shows that ρ is satisfied by $\mathsf{core}(\mathcal{I})$ for the particular match h. Since h was arbitrary, we obtain $\mathsf{core}(\mathcal{I}) \models \rho$.

($\Leftarrow$) This follows by direct application of the definitions. ◀

Given this connection between finite cores and incidental dependencies, one may ask whether it extends to cases where the set of TGDs does not admit a finite universal model. Unfortunately, this it not the case: Example 1 shows a case where an incidental dependency does not hold in a universal model that is in fact a core (the one-way infinite chain).

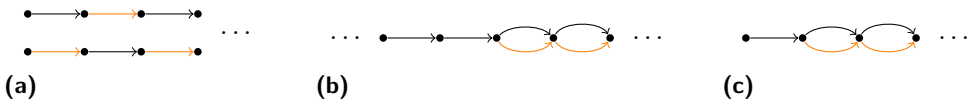

Figure 1 Universal models that have (a) two non-isomorphic cores, (b) no core, and (c) a core that is not a model, where R is black and S is orange (grey).

This discrepancy between incidentals and cores goes together with a general loss of good properties of the core on infinite models. Finite instances (i) always have a core, which is (ii) unique up to isomorphism [17, 20], and (iii) the core of a finite universal model of a set of TGDs is also a universal model [14]. Examples 12, 13, and 14 show that we no longer have any of these properties when dealing with infinite universal models.

▶ **Example 12.** Let Σ consist of the following TGDs:

$$\exists x, y.R(x, y) \qquad \exists x, y.S(z, w) \qquad R(x, y) \to \exists z.S(y, z) \qquad S(x, y) \to \exists z.R(y, z)$$

Figure 1a illustrates a universal model $\mathcal{I}$ of Σ. The upper and the lower chain of relations each by itself is a core of $\mathcal{I}$, but the chains are not isomorphic, so property (ii) does not hold.

▶ **Example 13.** Let Σ consist of the following three TGDs:

$$\exists x, y.R(x, y) \wedge S(x, y) \qquad R(x, y) \wedge S(x, y) \to \exists z.R(y, z) \wedge S(y, z) \qquad R(y, z) \to \exists x.R(x, y)$$

Figure 1b illustrates a universal model of Σ, which is not a core, since there are non-embedding endomorphisms that map parts of the single chain into the double chain. In fact, one can see that this instance does not have a core.

▶ **Example 14.** Let Σ consist of the following TGDs:

$$\exists x, y.R(x, y) \wedge S(x, y) \qquad R(x, y) \wedge S(x, y) \to \exists z.R(y, z) \wedge S(y, z) \qquad S(y, z) \to \exists x.R(x, y)$$

Figure 1c shows a universal model $\mathcal{I}$ of Σ. It is not a core, since there is a non-embedding endomorphism that maps each element to its right neighbour. This results in an instance that is isomorphic to $\mathcal{I}$ with the left-most node and its R-relation removed, which is a core of $\mathcal{I}$ but not a model for the third rule in Σ.

Nevertheless, cores can be relevant in finding instances that satisfy incidental dependencies. To this end, we consider a particularly well-behaved type of core that can be obtained as a limit of a growing sequence of finite cores.

▶ **Definition 15** (Core Cover)**.** An instance $\mathcal{J}$ has a *core cover* if there are finite subinstances $\mathcal{J}_0 \subseteq \mathcal{J}_1 \subseteq \mathcal{J}_2 \subseteq \ldots$ with $\mathcal{J} = \bigcup_{i \geq 0} \mathcal{J}_i$ such that, for all $\mathcal{J}_i$, every homomorphism $h : \mathcal{J}_i \to \mathcal{J}$ is an embedding.

▶ **Theorem 16.** *If an instance has a core cover then it is a core.*

Proof. Consider an instance $\mathcal{J}$ with core cover $(\mathcal{J}_i)_{i \geq 0}$, and an endomorphism $h : \mathcal{J} \to \mathcal{J}$. By Definition 15, the restriction $h_i : \mathcal{J}_i \to \mathcal{J}$ is an embedding for all $i \geq 0$. With $\mathcal{J} = \bigcup_{i \geq 0} \mathcal{J}_i$ and $\mathcal{J}_i \subseteq \mathcal{J}_{i+1}$ it follows that h is an embedding, otherwise, since injectivity and being strong both are finitary conditions, there would be a non-embedding $h_i : \mathcal{J}_i \to \mathcal{J}$. ◀

We remark that the condition that $\mathcal{J}_i \subseteq \mathcal{J}_{i+1}$ is needed for Theorem 16 to hold. Figure 2 illustrates an instance that satisfies the remaining conditions of Definition 15 for a set of disjoint instances $(\mathcal{J}_i)_{i \geq 0}$, but which is not a core.

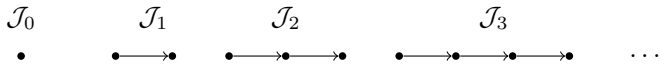

Figure 2 An instance that satisfies most conditions of Definition 15 but is not a core.

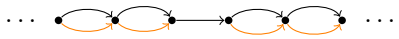

Figure 3 A core without a core cover, using two relations R (black) and S (orange/grey).

▶ **Example 17.** Having a core cover is a sufficient but not a necessary condition for an instance to be a core. Figure 3 illustrates an instance $\mathcal{I}$ that is a core. Indeed, any endomorphism must preserve the adjacency in this two-way infinite chain. But since one pair of elements is not S-related, only this very same pair can be mapped to this position in the chain, so the only endomorphism is the identity mapping.

However, $\mathcal{I}$ has no core cover, since any finite subset of $\mathcal{I}$ that contains the pair without the S connection can be mapped by a non-strong endomorphism into a sufficiently long fully R-S-connected segment of $\mathcal{I}$.

The next theorem shows that cores with core covers can characterise the set of full incidental dependencies for a set of TGDs.

▶ **Theorem 18.** *Let Σ be a set of TGDs and let $\mathcal{I}$ be an instance. Assume that $\mathsf{BCQ}(\mathcal{I}) = \mathsf{BCQ}(\Sigma)$ and $\mathcal{I}$ has a core cover. Then, $\rho \in \mathsf{ICDT}(\Sigma)$ iff $\mathcal{I} \models \rho$, for any full dependency ρ.*

Proof. ($\Rightarrow$) Let $(\mathcal{I}_i)_{i\geq 0}$ be a core cover for $\mathcal{I}$, and consider a full dependency $\rho : \varphi \to \psi$ that is incidental for Σ. If $\mathcal{I} \models \varphi$ for some homomorphism $h : \mathcal{I}_\varphi \to \mathcal{I}$, then there is $\mathcal{I}_i$ such that h can be considered as a homomorphism $\mathcal{I}_\varphi \to \mathcal{I}_i$. Let $\mathcal{J} := \mathcal{I}_i \cup h(\mathcal{I}_\psi)$, where we note that h does not introduce new nulls since ρ is full. Similar to the proof of Theorem 11, we find that $\Sigma \cup \{\rho\} \models q_\mathcal{J}$. Therefore $\mathcal{I} \models q_\mathcal{J}$ as ρ is incidental, and there is a corresponding homomorphism $g : \mathcal{J} \to \mathcal{I}$. Since $\Delta^\mathcal{J} = \Delta^{\mathcal{I}_i}$, g is a homomorphism $g : \mathcal{I}_i \to \mathcal{I}$, and therefore an embedding (Definition 15). This shows that $h(\mathcal{I}_\varphi \cup \mathcal{I}_\psi) \subseteq \mathcal{I}_i$ as required. Since h and $\mathcal{I}_i$ was arbitrary, we conclude that $\mathcal{I} \models \rho$.

($\Leftarrow$) This follows by direct application of the definitions. ◀

Given Theorem 18 and the observation that a core cover is closely related to a bottom-up construction of a core, one naturally wonders if a chase-like procedure could be used to obtain a suitable model. The prime candidate is the *core chase* of Deutsch et al. [14]:

▶ **Definition 19.** The *core chase sequence* for a set Σ of TGDs is a sequence $\mathcal{I}_0, \mathcal{I}_1, \ldots$ of instances, where $\mathcal{I}_0$ is the empty instance, and, for each $i > 0$, $\mathcal{I}_i$ is the core of $\Sigma(\mathcal{I}_{i-1})$. A finite core chase sequence $\mathcal{I}_0, \ldots, \mathcal{I}_\ell$ is *terminating* if $\mathcal{I}_\ell \models \Sigma$, and in this case, $\mathcal{I}_\ell$ is called the *core chase*.

Intuitively, the procedure defined by Deutsch et al. consists on applying the rules and taking the core of the resulting instance in each step. Deutsch et al. do not define the core chase for cases where Σ require infinite models, and indeed the limit of infinite core chase sequences is not defined here. While this issue can be repaired by using a more sophisticated definition, the deeper problem is that the result of applying the rules and then taking the core in each step may not be a core. This can be seen, e.g., from the TGDs in Example 12, on which an infinite core chase would simply produce the universal model shown in Figure 1b, which is not a core.

5 The Stable Chase

In the following section, we show that all sets of TGDs admit a BCQ-equivalent model that is a core and that characterises full incidental dependencies. To this end, we introduce the stable chase, a novel variant of the chase. Our approach can be viewed as a generalisation of the core chase where core computation is performed by looking for non-embedding homomorphisms of an instance into any future instance along a chase sequence. If such a homomorphism is found, all instances in the current chase sequence are rewritten as follows:

▶ **Definition 20.** Consider a homomorphism $h : \mathcal{I} \to \mathcal{J}$ on finite instances over $\mathcal{S}$, and let $\prec$ be a strict total order on $\Delta^{\mathcal{I}}$. The *h-rewriting* of an instance $\mathcal{K}$ is obtained as follows:
1. For all $R \in \mathcal{S}$ and $\boldsymbol{a} \in (\Delta^{\mathcal{K}} \cap \Delta^{\mathcal{I}})^{\mathsf{ar}(R)}$, with $h(\boldsymbol{a}) \in R^{\mathcal{J}}$, insert $\boldsymbol{a} \in R^{\mathcal{K}}$.
2. Replace all $a \in \Delta^{\mathcal{K}} \cap \Delta^{\mathcal{I}}$ by the $\prec$-least element $b \in \Delta^{\mathcal{K}} \cap \Delta^{\mathcal{I}}$ for which $h(a) = h(b)$.

The h-rewriting of a sequence of instances is the sequence of h-rewritings of its members.

▶ **Example 21.** Let $\mathcal{I}_{i,j}$ be the instance occurring in the i-th row, j-th column of Figure 4. Moreover, let $h : \mathcal{I}_{3,2} \to \mathcal{I}_{3,3}$ be the homomorphism that maps n_i to n_{i+1} for every $-1 \leq i \leq 2$. Then, $\mathcal{I}_{4,2}$ is the h-rewriting of $\mathcal{I}_{3,2}$, and (the sequence) $\mathcal{I}_{4,1}, \mathcal{I}_{4,2}, \mathcal{I}_{4,3}$ is the h-rewriting of (the sequence) $\mathcal{I}_{3,1}, \mathcal{I}_{3,2}, \mathcal{I}_{3,3}$.

We proceed with the definition of a *stabilising chase sequence* for a set of TGDs, which is a chase sequence that evolves in the sense that also previously derived instances may be modified at a later stage. The limit of this construction will yield a chase sequence from which we can obtain the potentially infinite core we are looking for.

▶ **Definition 22** (Stabilising Chase Sequence)**.** A *stabilising chase sequence* for a set Σ of TGDs is a series $\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{Q}}_0, \boldsymbol{\mathcal{Q}}_1, \ldots$ of chase sequences. Each $\boldsymbol{\mathcal{Q}}_k = \boldsymbol{\mathcal{Q}}_{k,0} \cdots \boldsymbol{\mathcal{Q}}_{k,\ell(k)}$ is a finite chase sequence of length $\ell(k) + 1$ consisting of instances $\boldsymbol{\mathcal{Q}}_{k,i}$, such that the following hold:
1. $\boldsymbol{\mathcal{Q}}_0$ is the singleton sequence containing the empty instance;
2. for all $k \geq 0$, either
   - **(2.a)** $\boldsymbol{\mathcal{Q}}_{k+1} = \boldsymbol{\mathcal{Q}}_{k,0}, \cdots, \boldsymbol{\mathcal{Q}}_{k,\ell(k)}, \Sigma(\boldsymbol{\mathcal{Q}}_{k,\ell(k)})$ is $\boldsymbol{\mathcal{Q}}_k$ extended by $\Sigma(\boldsymbol{\mathcal{Q}}_{k,\ell(k)})$, or
   - **(2.b)** $\boldsymbol{\mathcal{Q}}_{k+1}$ is the h-rewriting of $\boldsymbol{\mathcal{Q}}_k$ for some homomorphism $h : \boldsymbol{\mathcal{Q}}_{k,i} \to \boldsymbol{\mathcal{Q}}_{k,j}$ with $0 \leq i \leq j$ that is not an embedding,

where we require that the order $\prec$ from Definition 20 is an extension of the (partial) order in which new nulls are introduced, and that all possible rewritings will eventually be applied: if there is a homomorphism $h : \boldsymbol{\mathcal{Q}}_{k,i} \to \boldsymbol{\mathcal{Q}}_{k,j}$ as in (2.b), then there is $k' > k$ such that h is an embedding from the sub-structure of $\boldsymbol{\mathcal{Q}}_{k',i}$ on which h is defined to $\boldsymbol{\mathcal{Q}}_{k,j}$.

Our requirement on $\prec$ ensures that in cases where two elements are merged by a homomorphism in step (2.b), we will always pick one as a representative that has the longest history in the chase sequence. This ensures monotone growth of the domain within a sequence.

While we define the stabilising chase sequence $\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{Q}}_0, \boldsymbol{\mathcal{Q}}_1, \ldots$ to be infinite, it may happen that neither new derivations nor core constructions are possible at some stage. The process can still continue with step (2.a), appending copies of the last instance of the chase sequence, even if they contain no new derivations. Finite termination of the chase is therefore captured in the sequence becoming constant at some point.

▶ **Example 23.** Figure 4 illustrates a stabilising chase sequence $\boldsymbol{\mathcal{Q}}$ for the set of TGDs from Example 13. $\boldsymbol{\mathcal{Q}}_4$ is the h-rewriting of $\boldsymbol{\mathcal{Q}}_3$ for the non-strong homomorphism h denoted with dotted arrows in the figure.

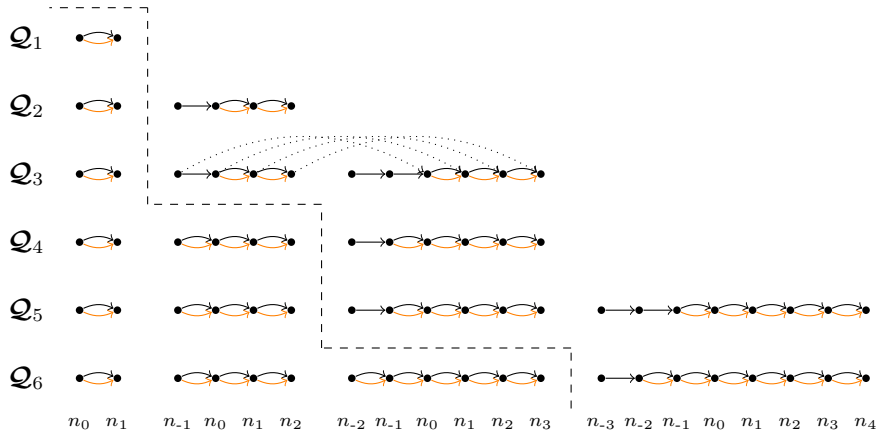

Figure 4 Stabilising chase sequence $\mathcal{Q}$ of Example 13 without the initial sequence $\mathcal{Q}_0$; relations R and S are denoted in black and orange (grey), respectively; domain elements are named below each column of instances.

▶ **Example 24.** A stabilising chase sequence might not be unique. For the set Σ of TGDs from Example 12, parallel chase steps as in (2.a) of Definition 22 yield instances that contain finite initial segments $R(a_0, a_1), S(a_1, a_2), \ldots$ and $S(b_0, b_1), R(b_1, b_2), \ldots$ of parallel chains as in Figure 1a. Non-embedding homomorphisms collapse the lower chain into (a longer future version of) the upper, or vice versa. In each case, the chase produces initial segments of a single infinite chain, which might begin with R or S depending on the chosen homomorphism.

For a particular stabilising chase sequence, however, the instances occurring in the i-th positions of the sequence will eventually stabilise to a unique structure.

▶ **Definition 25.** An instance $\mathcal{I}$ is *stable for position* i in a stabilising chase sequence $\mathcal{Q}$ if there is $k \geq 0$ such that $\mathcal{I} = \mathcal{Q}_{k',i}$ for all $k' \geq k$.

▶ **Lemma 26.** *There is a unique stable instance for every stabilising chase sequence* $\mathcal{Q}$ *and position* $i \geq 0$. *This stable instance is a core.*

Proof. There are three ways in which the finite structure $\mathcal{Q}_{\ell,i}$ may evolve for some $\ell \geq 0$: (1) $\mathcal{Q}_{\ell,i} = \mathcal{Q}_{\ell+1,i}$; (2) $\Delta^{\mathcal{Q}_{\ell,i}} \supset \Delta^{\mathcal{Q}_{\ell+1,i}}$; or (3) $R^{\mathcal{Q}_{\ell,i}} \subset R^{\mathcal{Q}_{\ell+1,i}}$ for some relational symbol R. The (not mutually exclusive) cases (2) and (3) can only occur for a finite number of times. For (2), it is clear that the finite domain cannot decrease in size infinitely often. Moreover, domain elements are only ever renamed if two of them are merged by a homomorphism during rewriting. The finite bound for (3) follows since there can only be at most finitely many relations over a finite domain. Therefore, there is some k for which $\mathcal{Q}_{k,i}$ is stable.

Stable instances are cores: otherwise they would admit a non-embedding endomorphism, which would eventually be used in step (2.b) of Definition 22, contradicting stability. ◀

We may therefore denote the stable instance for position i in $\mathcal{Q}$ by $\mathsf{st}(\mathcal{Q}, i)$, and use the sequence of stable instances to define an infinite structure:

▶ **Definition 27** (Stable Chase)**.** If $\mathcal{Q}$ is a stabilising chase sequence for some set Σ of TGDs, then $(\mathsf{st}(\mathcal{Q}, i))_{i\geq 0}$ is a *stable chase sequence* for Σ, and $\bigcup_{i\geq 0} \mathsf{st}(\mathcal{Q}, i)$ is a *stable chase* for Σ.

▶ **Example 28.** In Figure 4, all instances below the dashed line are stable in the stabilising chase sequence $\mathcal{Q}$ from Example 23 (using the TGDs Σ from Example 13). The corresponding

stable chase sequence is $(\boldsymbol{\mathcal{S}}_i)_{i\geq 0}$ where, for every $i \geq 0$, $\boldsymbol{\mathcal{S}}_i$ is a chain of length $2i$ of elements connected by R and S. This sequence $\boldsymbol{\mathcal{S}}$ is unique up to isomorphism in this case. The stable chase for Σ then is a two-way infinite chain of elements sequentially connected by R and S.

6 Properties of the Stable Chase

We start by showing that every set of TGDs admits a stable chase. We then show that the stable chase algorithm yields a model of Σ (Theorem 30) that is BCQ-equivalent to Σ (Theorem 31), and that is a core (Theorem 32). We show that it satisfies all full incidental dependencies (Theorem 33) and that it coincides with the result of the core chase in finite cases (Theorem 34). Nevertheless, we observe that the stable chase is neither unique nor a universal model. Finally, we show the existence of another BCQ-equivalent model that is a core and entails all incidental dependencies.

▶ **Theorem 29.** *Every set of TGDs has a stable chase sequence.*

Proof. We show that every set of TGDs admits a stabilising chase sequence $\boldsymbol{\mathcal{Q}}$. Indeed, let $\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{Q}}_0, \boldsymbol{\mathcal{Q}}_1, \ldots$ be a stabilising chase sequence constructed as follows:

1. Set $\boldsymbol{\mathcal{Q}}_0$ as the singleton sequence containing the empty instance.
2. For every $k \geq 0$: If every homomorphism $h : \boldsymbol{\mathcal{Q}}_{k,i} \to \boldsymbol{\mathcal{Q}}_{k,j}$ for every $0 \leq i \leq j$ is an embedding, then let $\boldsymbol{\mathcal{Q}}_{k+1} = \boldsymbol{\mathcal{Q}}_{k,0} \cdots \boldsymbol{\mathcal{Q}}_{k,\ell(k)} \Sigma(\boldsymbol{\mathcal{Q}}_{k,\ell(k)})$. Otherwise, $\boldsymbol{\mathcal{Q}}_{k+1}$ is the h-rewriting of $\boldsymbol{\mathcal{Q}}_k$ with h some (arbitrarily chosen) non-embedding homomorphism from some instance of $\boldsymbol{\mathcal{Q}}_k$ to another.

It is clear that the resulting series $\boldsymbol{\mathcal{Q}}$ satisfies 1 and 2 from Definition 22.

It remains to verify the fairness condition on the application of step (2.b). Consider some $k \geq 0$, some $0 \leq i \leq j$, and some non-embedding homomorphism $h : \boldsymbol{\mathcal{Q}}_{k,i} \to \boldsymbol{\mathcal{Q}}_{k,j}$. Then, let $\boldsymbol{\mathcal{Q}}_{k'}$ be the sequence in $\boldsymbol{\mathcal{Q}}$ with the same length as $\boldsymbol{\mathcal{Q}}_k$ such that k' is maximal (note that, $k' \geq k$). By item (2) and Definition 22, every homomorphism from $\boldsymbol{\mathcal{Q}}_{k'',i}$ to $\boldsymbol{\mathcal{Q}}_{k'',j}$ with $k'' \geq k'$ is an embedding. Moreover, we can show via induction that there is a homomorphism $h' : \boldsymbol{\mathcal{Q}}_{k,j} \to \boldsymbol{\mathcal{Q}}_{k'',j}$ for every $k'' \geq k'$. Note that, given some $k'' \geq k$, the existence of a non-embedding homomorphism $h'' : \boldsymbol{\mathcal{Q}}_{k'',i} \to \boldsymbol{\mathcal{Q}}_{k,j}$ would imply the existence of another homomorphism from $\boldsymbol{\mathcal{Q}}_{k'',i}$ to $\boldsymbol{\mathcal{Q}}_{k'',j}$ which is not an embedding either (namely, $h' \circ h''$). Hence, for every $k'' \geq k'$, every homomorphism $h'' : \boldsymbol{\mathcal{Q}}_{k'',i} \to \boldsymbol{\mathcal{Q}}_{k,j}$ is an embedding. ◀

▶ **Theorem 30.** *If $\mathcal{C}$ is a stable chase for Σ, then $\mathcal{C} \models \Sigma$.*

Proof. Let $\boldsymbol{\mathcal{Q}}$ be the stabilising chase sequence from which $\mathcal{C}$ was extracted. Consider any rule $\varphi \to \exists \boldsymbol{y}.\psi \in \Sigma$ that is applicable to $\mathcal{C}$ based on some homomorphism $h : \mathcal{I}_\varphi \to \mathcal{C}$. Since $\mathcal{I}_\varphi$ is finite, there is $i \geq 0$ such that h restricts to a homomorphism $\mathcal{I}_\varphi \to \mathsf{st}(\boldsymbol{\mathcal{Q}}, i)$. Let k be the least number such that $\boldsymbol{\mathcal{Q}}_{k,i} = \mathsf{st}(\boldsymbol{\mathcal{Q}}, i)$. By Definition 22, we find that $\boldsymbol{\mathcal{Q}}_{k,i} \subseteq \boldsymbol{\mathcal{Q}}_{k,j}$ for all $i \leq j \leq \ell(k)$. Moreover, there is $k' > k$ with $\ell(k') = \ell(k) + 1$ and $\boldsymbol{\mathcal{Q}}_{k',\ell(k)+1} = \Sigma(\boldsymbol{\mathcal{Q}}_{k'-1,\ell(k)})$ (step 2.a). Since $\mathsf{st}(\boldsymbol{\mathcal{Q}}, i) = \boldsymbol{\mathcal{Q}}_{k,i} = \boldsymbol{\mathcal{Q}}_{k'-1,i} \subseteq \boldsymbol{\mathcal{Q}}_{k'-1,\ell(k)}$, rule $\varphi \to \exists \boldsymbol{y}.\psi$ is applicable to $\boldsymbol{\mathcal{Q}}_{k'-1,\ell(k)}$ under h. Therefore, $\boldsymbol{\mathcal{Q}}_{k',\ell(k)+1}$ contains the result of this rule application, and by Definition 20 this remains true (possibly for some renaming of new nulls) in $\mathsf{st}(\boldsymbol{\mathcal{Q}}, \ell(k) + 1)$ and hence in $\mathcal{C}$. ◀

▶ **Theorem 31.** *If $\mathcal{C}$ is a stable chase for Σ, then $\mathcal{C}$ and Σ are BCQ-equivalent.*

Proof. By Theorem 30 and the definition of BCQ entailment, $\Sigma \models q$ implies $\mathcal{C} \models q$ for all BCQs q.

For the converse, let $\boldsymbol{\mathcal{Q}}$ be some stabilising chase sequence for $\mathcal{C}$. We show that $\mathcal{C} \models q$ implies $\Sigma \models q$ for every BCQ q. By compactness, it suffices to show that $\mathsf{st}(\boldsymbol{\mathcal{Q}}, i) \models q$ implies $\Sigma \models q$ for all for any $i \geq 0$. We show the stronger claim $\boldsymbol{\mathcal{Q}}_{k,i} \models q$ implies $\Sigma \models q$ for all $i \geq 0$ by induction on k.

For the base case, $\boldsymbol{\mathcal{Q}}_{0,0}$ is the empty instance, and the claim is immediate. Now assume the claim holds for all instances of $\boldsymbol{\mathcal{Q}}_k$. Definition 22 has two ways for constructing $\boldsymbol{\mathcal{Q}}_{k+1}$:

(2.a) Then the only new instance is $\Sigma(\boldsymbol{\mathcal{Q}}_{k,\ell(k)})$. Since the claim holds for $\mathcal{I} = \boldsymbol{\mathcal{Q}}_{k,\ell(k)}$, we find that $\Sigma \models q_{\mathcal{I}}$ for the corresponding BCQ $q_{\mathcal{I}}$. Therefore, any rule application that is possible on $\mathcal{I}$ is possible (up to isomorphism) in any model of Σ, and hence $\Sigma \models q_{\Sigma(\mathcal{I})}$, which entails the claim.

(2.b) Let $h : \boldsymbol{\mathcal{Q}}_{k,i} \to \boldsymbol{\mathcal{Q}}_{k,j}$ be the homomorphism used for the rewriting. Then h restricts to a homomorphism $\boldsymbol{\mathcal{Q}}_{k+1,i} \to \boldsymbol{\mathcal{Q}}_{k,j}$. By Definition 22, we find that $\boldsymbol{\mathcal{Q}}_{k+1,i'} \subseteq \Sigma(\boldsymbol{\mathcal{Q}}_{k+1,i'-1})$ for all $i' > i$. Therefore, $\boldsymbol{\mathcal{Q}}_{k+1,\ell(k)} \subseteq \Sigma^{\ell(k)-i}(\boldsymbol{\mathcal{Q}}_{k+1,i})$ (‡). It suffices to consider BCQs q that are entailed by $\boldsymbol{\mathcal{Q}}_{k+1,\ell(k)}$ (where $\ell(k) = \ell(k+1)$), since they subsume all BCQ entailment in any instance of $\boldsymbol{\mathcal{Q}}_{k+1}$. By (‡), $\boldsymbol{\mathcal{Q}}_{k+1,\ell(k)} \models q$ implies $\boldsymbol{\mathcal{Q}}_{k+1,i}, \Sigma \models q$. Using the homomorphism $h : \boldsymbol{\mathcal{Q}}_{k+1,i} \to \boldsymbol{\mathcal{Q}}_{k,j}$, we get $\boldsymbol{\mathcal{Q}}_{k,j}, \Sigma \models q$ and hence $\Sigma \models q$.

◀

▶ **Theorem 32.** *If $\mathcal{C}$ is a stable chase for Σ, then $\mathcal{C}$ is a core.*

Proof. By Definition 27, there is some rewritten chase sequence $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{S}}_0, \boldsymbol{\mathcal{S}}_1, \ldots$ with $\mathcal{C} = \bigcup_{i \geq 0} \boldsymbol{\mathcal{S}}_i$. Moreover, for every $i \geq j \geq 0$ and every homomorphism $h : \boldsymbol{\mathcal{S}}_i \to \boldsymbol{\mathcal{S}}_j$, h is an embedding. Since every element of $\boldsymbol{\mathcal{S}}$ is finite, every homomorphism h mapping such element to $\mathcal{C}$ is also an embedding. Since $\boldsymbol{\mathcal{S}}_{i-1} \subseteq \boldsymbol{\mathcal{S}}_i$ for every $i \geq 1$, we conclude that $\boldsymbol{\mathcal{S}}$ is a core cover for $\mathcal{C}$. Therefore, we can apply Theorem 16 to show that the theorem follows. ◀

The previous observation that the stable chase sequence yields a core cover, together with the BCQ-equivalence of stable chase and Σ (Theorem 31), lets us apply Theorem 18 to conclude that the stable chase does indeed characterise the full incidental dependencies:

▶ **Theorem 33.** *Every stable chase of Σ entails exactly those full dependencies that are incidental for Σ.*

As one would expect in the light of Theorem 8, the stable chase does not constitute a semi-decision procedure for incidentality or non-incidentality. On the one hand, the stable chase may not terminate, on the other hand we cannot even decide if a given finite instance in a stabilising chase sequence is already stable.

The core chase can be viewed as a special case of the stable chase procedure, since it can be obtained by prioritising step (2.b) in Definition 22, while applying it only to the last instance in a chase sequence $\boldsymbol{\mathcal{Q}}_k$ (this forces the homomorphism that is used to be an endomorphism). For finite models, this does not change the outcome, and indeed the stable chase coincides with the core chase whenever the latter is defined:

▶ **Theorem 34.** *If a set Σ of TGDs has a finite universal model, then the stable chase over Σ is equal to the result of the core chase, up to isomorphism.*

Proof. Deutsch et al. showed that the core chase yields a finite universal model in this case [14, Theorem 7]. Let $\mathcal{U}$ be this model, and let $\mathcal{C}$ be a stable chase of Σ. Since $\mathcal{U}$ is universal, there is a homomorphism $h : \mathcal{C} \to \mathcal{U}$, since $\mathcal{C}$ is a model (Theorem 30). Moreover, since $\mathcal{U}$ is finite, $\mathcal{U} \models q_{\mathcal{U}}$, and since $\mathcal{U}$ and $\mathcal{C}$ are BCQ-equivalent (Theorem 31), there is a homomorphism $h' : \mathcal{U} \to \mathcal{C}$. Therefore, the function $h \circ h'$ is an endomorphism over $\mathcal{C}$ with a

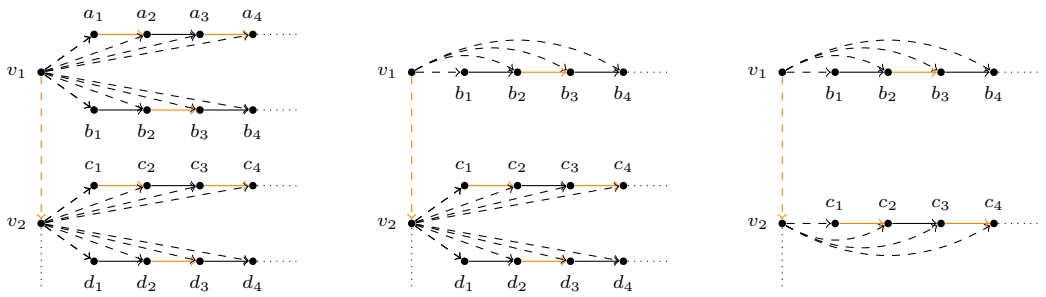

Figure 5 Instances $\mathcal{I}$ (left), $\mathcal{J}$ (middle), and $\mathcal{K}$ (right). Roles U, V, R, and S are represented with dashed and black, dashed and orange (possibly grey), black, and orange (possibly grey) arrows, respectively; dotted edges indicate the continuation of a sequence of elements up to some length.

finite range. Since $\mathcal{C}$ is a core (Theorem 32), every endomorphism (including $h \circ h'$) must be injective and hence, $\mathcal{C}$ must be finite. Since $\mathcal{C}$ is finite, BCQ-equivalent to $\mathcal{U}$, and a core, we conclude that $\mathcal{C}$ is equal to $\mathcal{U}$ up to isomorphism. ◀

We continue with some limitations of the stable chase: it may not yield a universal model, it may admit uncountably many non-isomorphic results, and it cannot be used to characterise non-full incidental TGDs. As already pointed out in Section 4, there are sets of TGDs that only admit universal models which are not cores (e.g., see the set of TGDs from Example 14). Hence, since the stable chase is guaranteed to yield a core (Theorem 32), it may not always produce a universal model. To illustrate the other limitations, consider the following example.

▶ **Example 35.** Consider a set Σ of TGDs containing the following dependencies.

$$\exists x, y.V(x, y) \qquad U(x, y) \wedge R(y, z) \to U(x, z)$$
$$V(x, y) \to \exists z.V(y, z) \qquad U(x, y) \wedge S(y, z) \to U(x, z)$$
$$V(x, y) \to \exists z, w.U(x, z) \wedge R(z, w) \qquad R(x, y) \to \exists z.S(y, z)$$
$$V(x, y) \to \exists z, w.U(x, z) \wedge S(z, w) \qquad S(x, y) \to \exists z.R(y, z)$$

Moreover, let $\mathcal{I}$, $\mathcal{J}$, and $\mathcal{K}$ be the instances depicted in Figure 5.

By iteratively applying the chase step (2.a) from Definition 22 during the computation of some stabilising chase sequence $\boldsymbol{\mathcal{Q}}$ of Σ, we can produce an instance such as $\mathcal{I}$ containing an arbitrarily long V chain, and two alternating R and S chains linked to every element v_i of such V-chain. Applying step (2.b) from Definition 22, we can, for each pair of chains in $\mathcal{I}$ linked to the same v_i, collapse the lower chain into the upper, or vice versa. In each case, the chase will produce initial segments of a single alternating infinite chain, which might begin with R or S. Applying such h-rewritings, we can produce instances such as $\mathcal{J}$ and $\mathcal{K}$.

The h-rewritings discussed above are somehow similar to the rewriting discussed in Example 24. However, in the current example, we have an infinite number of rewritings to consider–one for each element v_i in the infinite V chain. Taking into account all of these choices, we can generate uncountably many different stable chase sequences which can in turn be used to define uncountably many non-isomorphic stable chases.

Finally, note that there are (non-full) incidental TGDs for Σ, such as $V(x, y) \to \exists z.V(z, x)$, which are not entailed by any stable chase of Σ.

As highlighted by the previous example, instances resulting from the stable chase sequence may not be used to characterise non-full TGDs. Nevertheless, we can show that, for a set of

TGDs, there is an instance that satisfies all incidentals. While this result shows the existence of a suitable structure, it does not offer a constructive way of approximating it, since it relies on the (infinite) set of all incidentals to be given.

▶ **Theorem 36.** *Given a set Σ of TGDs, there is an instance $\mathcal{I}$ such that:*
1. *$\mathcal{I}$ is a core;*
2. *$\mathcal{I} \models \Sigma$;*
3. *$\mathsf{BCQ}(\mathcal{I}) = \mathsf{BCQ}(\Sigma)$; and*
4. *$\rho \in \mathsf{ICDT}(\Sigma)$ iff $\mathcal{I} \models \rho$, for any TGD ρ.*

Proof. To show this theorem we sketch how one can adapt the stable chase so that it can deal with infinite sets of TGDs. In this case infinite instances may occur in a stabilising chase of $\mathsf{ICDT}(\Sigma)$ and hence, the stable chase is not well-defined. To avoid this, we slightly modify Definition 22: In (2.a), instead of setting $\boldsymbol{\mathcal{Q}}_{k+1}$ as the extension of $\boldsymbol{\mathcal{Q}}_k$ with $\mathsf{ICDT}(\Sigma)(\boldsymbol{\mathcal{Q}}_{k,\ell(k)})$, we define this sequence as the extension of $\boldsymbol{\mathcal{Q}}_k$ with $\rho(\boldsymbol{\mathcal{Q}}_{k,\ell(k)})$ for some $\rho \in \mathsf{ICDT}(\Sigma)$. Moreover, we must also ensure fairness of the application of the rules in $\mathsf{ICDT}(\Sigma)$; i.e., each rule in $\mathsf{ICDT}(\Sigma)$ must be applied after the computation of a finite amount of sequences. With this modified version of the stable chase, one can show that it maintains its main properties. Then, by Theorem 29 there is some stable chase $\mathcal{I}$ of $\mathsf{ICDT}(\Sigma)$ which, by Theorem 32, $\mathcal{I}$ is a core; by Theorem 30, $\mathcal{I} \models \mathsf{ICDT}(\Sigma)$ and hence, $\mathcal{I}$ entails all subsets of $\mathsf{ICDT}(\Sigma)$, including Σ; and by Theorem 31, $\mathsf{BCQ}(\mathcal{I}) = \mathsf{BCQ}(\Sigma)$. Also, if $\rho \in \mathsf{ICDT}(\Sigma)$, then $\mathcal{I} \models \rho$ since $\mathcal{I} \models \mathsf{ICDT}(\Sigma)$. Conversely, if $\mathcal{I} \models \rho$ then $\rho(q) \in \mathsf{BCQ}(\Sigma)$ for every $q \in \mathsf{BCQ}(\Sigma)$. Therefore, by Theorem 4, $\rho \in \mathsf{ICDT}(\Sigma)$. ◀

7 Conclusion

To the best of our knowledge, this is the first study on constraint implication in the presence of arbitrary theories of tuple-generating dependencies. This idea is embodied in our new notion of incidental dependencies, which correspond to constraints that can be safely assumed to hold when checking BCQ entailment, despite not being a consequence of the given TGD set. Even for a single, fixed instance, finding incidental dependencies remains a challenging problem which is highly undecidable.

Our work reveals close connections between incidental dependencies and cores. If a finite universal model exists, its unique core perfectly characterises the incidentals. The correspondence breaks down if models become infinite, but we can still find cases where cores characterise at least all full incidental dependencies. However, one then has to be content with cores that are BCQ-equivalent to the universal models, but that are not universal themselves. To obtain such cores, we presented the stable chase as a generalisation of the core chase that can be used to build infinite models, and which is interesting in its own right.

On the theoretical level, several questions remain for future work: Is there a construction alike the stable chase which produces a BCQ-equivalent model which is indicative of *all* incidental TGDs (not just the full ones), without knowing all incidentals beforehand? What are the computational characteristics of Incidental for restricted classes of TGDs (such as guarded [4], sticky [10], etc.)? Obviously, all classes that warrant a finite universal model (such as diverse versions of acyclic TGDs [16, 24, 21, 13] and full TGDs) guarantee decidability, but the exact complexity of checking incidentality of individual TGDs would still be of interest. Further questions arise when considering equality-generating dependencies in addition to TGDs. Finally, it is of great importance to understand how known incidentals can be exploited toward more efficient practical query answering, as already suggested in some previous works [22, 25].

References

1 Serge Abiteboul and Richard Hull. Data functions, datalog and negation. In *Proc. SIGMOD Int. Conf. on Management of Data (SIGMOD'88)*, pages 143–153. ACM, 1988.

2 Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. Extending decidable cases for rules with existential variables. In Craig Boutilier, editor, *Proc. 21st Int. Joint Conf. on Artificial Intelligence (IJCAI'09)*, pages 677–682. IJCAI, 2009.

3 Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artificial Intelligence*, 175(9–10):1620–1654, 2011.

4 Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Walking the complexity lines for generalized guarded existential rules. In Toby Walsh, editor, *Proc. 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11)*, pages 712–717. AAAI Press/IJCAI, 2011.

5 Bruce L. Bauslaugh. Core-like properties of infinite graphs and structures. *Discrete Mathematics*, 138(1):101–111, 1995.

6 Bruce L. Bauslaugh. Cores and compactness of infinite directed graphs. *J. of Comb. Theory Ser. B*, 68(2), 1996.

7 Catriel Beeri and Moshe Y. Vardi. A proof procedure for data dependencies. *J. ACM*, 31(4):718–741, 1984.

8 Manuel Bodirsky. The core of a countably categorical structure. In Volker Diekert and Bruno Durand, editors, *Proc. 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS'05)*, volume 3404 of *Lecture Notes in Computer Science*, pages 110–120. Springer, 2005.

9 Pierre Bourhis, Markus Krötzsch, and Sebastian Rudolph. Reasonable highly expressive query languages. In Qiang Yang and Michael Wooldridge, editors, *Proc. 24th Int. Joint Conf. on Artificial Intelligence (IJCAI'15)*, pages 2826–2832. AAAI Press, 2015.

10 Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *J. of Artif. Intell.*, 193:87–128, 2012.

11 Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. Reasoning on regular path queries. *SIGMOD Record*, 32(4):83–92, 2003.

12 Stavros S. Cosmadakis, Haim Gaifman, Paris C. Kanellakis, and Moshe Y. Vardi. Decidable optimization problems for database logic programs (preliminary report). In Janos Simon, editor, *Proc. 20th Annual ACM Symposium on Theory of Computing (STOC'88)*, pages 477–490. ACM, 1988.

13 Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, and Zhe Wang. Acyclicity notions for existential rules and their application to query answering in ontologies. *J. of Artificial Intelligence Research*, 47:741–808, 2013.

14 Alin Deutsch, Alan Nash, and Jeffrey B. Remmel. The chase revisited. In Maurizio Lenzerini and Domenico Lembo, editors, *Proc. 27th Symposium on Principles of Database Systems (PODS'08)*, pages 149–158. ACM, 2008.

15 Alin Deutsch and Val Tannen. Reformulation of XML queries and constraints. In Diego Calvanese, Maurizio Lenzerini, and Rajeev Motwani, editors, *Proc. 9th Int. Conf. on Database Theory (ICDT'03)*, volume 2572 of *LNCS*, pages 225–241. Springer, 2003.

16 Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.

17 Ronald Fagin, Phokion G. Kolaitis, and Lucian Popa. Data exchange: Getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, 2005.

18 Daniela Florescu, Alon Levy, and Dan Suciu. Query containment for conjunctive queries with regular expressions. In Alberto O. Mendelzon and Jan Paredaens, editors, *Proc. 17th Symposium on Principles of Database Systems (PODS'98)*, pages 139–148. ACM, 1998.

19 Georg Gottlob and Christos H. Papadimitriou. On the complexity of single-rule datalog queries. *Inf. Comput.*, 183(1):104–122, 2003.

20 Pavol Hell and Jaroslav Nešetřil. The core of a graph. *Discrete Mathematics*, 109:117–126, 1992.

21 Markus Krötzsch and Sebastian Rudolph. Extending decidable existential rules by joining acyclicity and guardedness. In Toby Walsh, editor, *Proc. 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11)*, pages 963–968. AAAI Press/IJCAI, 2011.

22 Markus Krötzsch and Veronika Thost. Ontologies for knowledge graphs: Breaking the rules. In Yolanda Gil, Elena Simperl, Paul Groth, Freddy Lecue, Markus Krötzsch, Alasdair Gray, Marta Sabou, Fabian Flöck, and Hideaki Takeda, editors, *Proc. 15th Int. Semantic Web Conf. (ISWC'16)*, volume 9981 of *LNCS*, pages 376–392. Springer, 2016.

23 David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. Testing implications of data dependencies. *ACM Transactions on Database Systems*, 4:455–469, 1979.

24 Bruno Marnette. Generalized schema-mappings: from termination to tractability. In Jan Paredaens and Jianwen Su, editors, *Proc. 28th Symposium on Principles of Database Systems (PODS'09)*, pages 13–22. ACM, 2009.

25 Mariano Rodriguez-Muro, Roman Kontchakov, and Michael Zakharyaschev. Ontology-based data access: Ontop of databases. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul T. Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha F. Noy, Chris Welty, and Krzysztof Janowicz, editors, *Proc. 12th Int. Semantic Web Conf. (ISWC'13)*, volume 8218 of *Lecture Notes in Computer Science*, pages 558–573. Springer, 2013.

26 Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. MIT Press, paperback edition edition, 1987.

27 Sebastian Rudolph and Markus Krötzsch. Flag & check: Data access with monadically defined queries. In Richard Hull and Wenfei Fan, editors, *Proc. 32nd Symposium on Principles of Database Systems (PODS'13)*, pages 151–162. ACM, 2013.

28 Yehoshua Sagiv. Optimizing datalog programs. In Moshe Y. Vardi, editor, *Proc. Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'87)*, pages 349–362. ACM, 1987.

29 Ke Wang and Li-Yan Yuan. Preservation of integrity constraints in definite DATALOG programs. *Inf. Process. Lett.*, 44(4):185–193, 1992.

8 Appendix: Proof of Theorem 8

▶ **Theorem 8.** INCIDENTAL *is* Π^0_2*-complete, and in particular neither in* RE *nor in* coRE.

Proof. For membership note that we can characterise incidentiality by quantifying over (finite) derivations (or proofs) in some theory. Indeed, a TGD ρ is incidental for Σ if: for all derivations that show $\Sigma \cup \{\rho\} \models q$ for some BCQ q, there is a derivation that shows $\Sigma \models q$. Using Gödel numbers for representing derivations, this condition can be expressed as a $\forall\exists$-sentence in first-order arithmetic.

We show hardness by many-one reduction from the *universal halting problem*, which is as follows: given a (deterministic) Turing machine $\mathcal{M}$, does $\mathcal{M}$ halt on all inputs? Universal halting is known to be complete for Π^0_2 (see [26, Theorem VIII], and apply Post's Theorem).

For the reduction, we construct for a given TM $\mathcal{M}$ a set $\Sigma_\mathcal{M}$ of TGDs and a full TGD ρ such that ρ is incidental for Σ iff $\mathcal{M}$ halts universally. The rules of $\Sigma_\mathcal{M}$ consist of three parts: Σ_1 ensures that each model contains representations of all possible inputs; Σ_2 simulates $\mathcal{M}$ on a particular input; Σ_3 marks elements of an accepting TM simulation with a specific unary relation halted. The rule ρ then asserts that initial elements in TM simulations are always marked by halted, which is incidental if all runs have indeed terminated.

In detail, let Q be the set of states, Γ be the tape alphabet with blank symbol $\square$, and $\delta : Q \times \Gamma \to Q \times \Gamma \times \{r, l\}$ the (total, deterministic) transition function of $\mathcal{M}$. Let $q_s, q_e \in Q$ be the starting and halting state, respectively. We assume without loss of generality that $\mathcal{M}$ never returns to q_s during a run. Our encoding uses the following unary relation symbols:

- symbol_σ: marks tape positions with symbol $\sigma \in \Gamma$
- head_q: marks tape position of head with TM in state $q \in Q$
- end: marks last (explicitly represented) tape position
- halted: used to mark halting configurations

as well as binary relations g, f (used to generate inputs), right, right^+ (right tape neighbour and its transitive closure), and next (tape cell in next configuration).

Now Σ_1 contains the followig rules

$$\rightarrow \exists y.\mathsf{symbol}_\square(y) \tag{4}$$

$$\mathsf{symbol}_\sigma(x) \rightarrow \exists y.g(x,y) \wedge \mathsf{symbol}_{\sigma'}(y) \tag{5}$$

$$\mathsf{symbol}_\sigma(x) \rightarrow \exists y.f(x,y) \wedge \mathsf{symbol}_\sigma(y) \wedge \mathsf{head}_{q_s}(y) \tag{6}$$

$$\mathsf{symbol}_\sigma(x) \wedge g(x,y) \wedge f(y,z) \rightarrow \exists v.f(x,v) \wedge \mathsf{right}(z,v) \tag{7}$$

$$\mathsf{symbol}_\square(x) \wedge f(x,y) \rightarrow \mathsf{end}(y) \tag{8}$$

each instantiated for all $\sigma, \sigma' \in \Gamma$. Models of Σ_1 projectively contain an infinite g-tree with root labelled $\mathsf{symbol}_\square$ (4) and other nodes labeled by symbols symbol_σ (5). Each node f-relates to the start of an initial sequence (6), which continues as a parallel copy of the finite path up until the root of the tree (7). The last cell of each initial tape is marked with end (8).

The set Σ_2 that simulates $\mathcal{M}$ is defined as follows. The following rules generate an infinite grid of TM tapes, with each tape one cell longer than the previous:

$$\mathsf{end}(x) \rightarrow \exists v, w.\mathsf{next}(x,v) \wedge \mathsf{right}(v,w) \wedge \mathsf{end}(w) \wedge \mathsf{symbol}_\square(w) \tag{9}$$

$$\mathsf{right}(x,y) \wedge \mathsf{next}(y,z) \rightarrow \exists v.\mathsf{next}(x,v) \wedge \mathsf{right}(v,z) \tag{10}$$

For every transition $\delta(q,\sigma) = \langle q', \sigma', r\rangle$, Σ_2 contains a rule:

$$\begin{aligned}&\mathsf{head}_q(x) \wedge \mathsf{symbol}_\sigma(x) \wedge \mathsf{right}(x,y) \wedge \mathsf{next}(x,x') \wedge \mathsf{next}(y,y') \\ &\qquad \rightarrow \mathsf{symbol}_{\sigma'}(x') \wedge \mathsf{head}_{q'}(y')\end{aligned} \tag{11}$$

and for every transition $\delta(q,\sigma) = \langle q', \sigma', l\rangle$ the rule

$$\begin{aligned}&\mathsf{head}_q(x) \wedge \mathsf{symbol}_\sigma(x) \wedge \mathsf{right}(y,x) \wedge \mathsf{next}(x,x') \wedge \mathsf{next}(y,y') \\ &\qquad \rightarrow \mathsf{symbol}_{\sigma'}(x') \wedge \mathsf{head}_{q'}(y').\end{aligned} \tag{12}$$

Finally, Σ_2 also contains the following rules for all $\sigma \in \Gamma$ and $q \in Q$:

$$\mathsf{right}(x,y) \rightarrow \mathsf{right}^+(x,y) \tag{13}$$

$$\mathsf{right}(x,y) \wedge \mathsf{right}^+(y,z) \rightarrow \mathsf{right}^+(x,z) \tag{14}$$

$$\mathsf{symbol}_\sigma(x) \wedge \mathsf{right}^+(x,y) \wedge \mathsf{head}_q(y) \wedge \mathsf{next}(x,x') \rightarrow \mathsf{symbol}_\sigma(x') \tag{15}$$

$$\mathsf{symbol}_\sigma(x) \wedge \mathsf{right}^+(y,x) \wedge \mathsf{head}_q(y) \wedge \mathsf{next}(x,x') \rightarrow \mathsf{symbol}_\sigma(x') \tag{16}$$

Rules (13) and (14) define right^+ to be (a superset of) the transitive closure of right, used by rules (15) and (16) to preserve tape contents at positions different from the head position. It is not hard to see that rules (9)–(16) create a simulatation of $\mathcal{M}$ for each starting configuration.

Finally, Σ_3 contains the rules

$$\mathsf{head}_{q_e}(x) \to \mathsf{halting}(x) \tag{17}$$

$$\mathsf{right}(x, y) \wedge \mathsf{halting}(y) \to \mathsf{halting}(x) \tag{18}$$

$$\mathsf{next}(x, y) \wedge \mathsf{halting}(y) \to \mathsf{halting}(x) \tag{19}$$

which propagate halting back to the first position of the first tape if the TM ever reaches q_e.

We claim that ρ is incidental for $\Sigma_{\mathcal{M}} = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3$ iff $\mathcal{M}$ is universally halting. Indeed, if $\mathcal{M}$ is universally halting, then any universal model of $\Sigma_{\mathcal{M}}$ will have halting propagated back to the first cell of the first tape in each TM simulation, so that the rule is already satisfied in this model.

Conversely, if $\mathcal{M}$ is not universally halting, then there is an input $w = w_1 \cdots w_n$ on which it does not halt. Any universal model of $\Sigma_{\mathcal{M}}$ contains an initial tape for w, with the first position not marked by halting. The BCQ $\exists \boldsymbol{x}.\mathsf{halting}(x_1) \wedge \mathsf{head}_{q_s}(x_1) \wedge \mathsf{symbol}_{w_1}(x_1) \wedge \mathsf{right}(x_1, x_2) \wedge \ldots \wedge \mathsf{symbol}_{w_n}(x_n) \wedge \mathsf{right}(x_n, x_{n+1}) \wedge \mathsf{end}(x_n)$ is not entailed by $\Sigma_{\mathcal{M}}$ but by $\Sigma_{\mathcal{M}} \cup \{\rho\}$. ◀