Will My Favorite Chases Terminate if Evaluating Conjunctive Queries Does? One Does Not Simply Decide This

Lucas Larroque[1] , **Quentin Manière**[2]
[1]Inria, DI ENS, ENS, CNRS, PSL University, Paris, France
[2]LIRMM, Inria, University of Montpellier, CNRS, Montpellier, France
{lucas.larroque, quentin.maniere}@inria.fr

Abstract

Existential rules are a prominent formalism to enrich a database with knowledge from the domain of interest, but make even basic reasoning tasks on the resulting knowledge base undecidable. To circumvent this, several classes of rules offering various useful properties have been identified. One such class, for instance, contains all sets of rules on which the chase algorithm always terminates, which guarantees the existence of a finite universal model. However, these classes are often abstract rather than concrete: it may be undecidable to check whether a given set of rules belongs to them. Given that the most studied classes of existential rules are designed for reasoning on databases, thus ensuring decidable conjunctive query entailment, we ask: Within a class that supports decidable query entailment, do the usual abstract classes become concrete? We answer in the negative for classes based upon the termination of all classical chase variants and for the *bts* class.

1 Introduction

One of the main tasks in knowledge representation and reasoning is ontology-based data access (OBDA). In OBDA, an ontology is used to enrich a database with domain knowledge, enabling the inference of new information that is not explicitly stored in the database. Existential rules (also known as Datalog$^{\pm}$, or tuple-generating dependencies) are a prominent formalism to represent ontologies in OBDA [Fagin, 2018]. A fundamental reasoning task in OBDA is the Boolean conjunctive query (BCQ) entailment problem, which consists in deciding whether a BCQ is entailed by a database and an ontology of interest. Unfortunately, with ontologies expressed as sets of existential rules, BCQ entailment is undecidable in general [Beeri and Vardi, 1981; Calì *et al.*, 2012]. To ensure decidable BCQ entailment, three main properties have been proposed.

First, the chase procedure [Maier *et al.*, 1979] is a materialization-based algorithm that, given a database and a set of existential rules, iteratively applies the rules to the database to produce a (possibly infinite) universal model of the input. If finite, this universal model can then be queried using regular BCQ evaluation techniques to decide entailment. Many variants of the chase procedure exist, differing in the way rules are applied and redundancies are handled. In this paper, we consider the oblivious [Calì *et al.*, 2013], semi-oblivious (*a.k.a.* Skolem) [Marnette, 2009], restricted (*a.k.a.* standard) [Fagin *et al.*, 2005a], and core chase variants [Deutsch *et al.*, 2008]. Guaranteed termination of the chase thus implies decidable BCQ entailment, but this property is also desirable in data exchange settings, where one is interested in translating data from a source schema to a target schema while preserving certain properties [Fagin *et al.*, 2005b]. It also enables computing aggregates, which otherwise make little sense on infinite models [Afrati and Kolaitis, 2008], and ensures finite controllability, *i.e.* only considering finite models does not impact query answers.

Second, query rewriting techniques aim at rewriting the input BCQ into a first-order query that can be directly evaluated over the input database. Rule sets for which such a rewriting exists (and is computable) are called *fus*, for finite unification set [Baget *et al.*, 2009]. Again, *fus* rule sets have benefits beyond decidability of BCQ entailment, as they allow answering the rewritten query over any database. In particular, this allows answering queries even without edit access, or answering queries efficiently in streaming settings [Ronca *et al.*, 2018].

Third, bounded treewidth sets (*bts*) [Calì *et al.*, 2013] are classes of existential rules that ensure that, for any database, the universal model produced by the chase has bounded treewidth. This property renders BCQ entailment decidable by using Courcelle's theorem [Courcelle, 1990], which states that any property definable in monadic second-order logic can be decided in linear time over structures of bounded treewidth. In particular, this helps with efficient answer counting [Feier *et al.*, 2023].

Deciding whether a given set of existential rules has any of these properties (*i.e.* whether it guarantees termination of a certain chase variant, is *fus*, or is *bts*) is, here again, undecidable in general. The classes of such sets of rules are thus called *abstract*, as opposed to *concrete* classes for which membership is decidable. Efforts have been devoted to proving that, when restricting to some concrete classes of existential rules, membership in the abstract classes presented above becomes decidable. This is especially the case for chase termination: oblivious and semi-oblivious chase termination have been proven decidable for guarded and for sticky rules [Calautti *et al.*, 2015;

Calautti and Pieris, 2019], while core chase termination is decidable for guarded rules [Hernich, 2012] but remains open for sticky rules. Restricted chase termination appears more difficult: its decidability has been proven for single-head guarded and sticky rules [Gogacz *et al.*, 2020], but remains open in the multi-head case for both classes. The multi-head case for linear rules, a subclass of guarded rules, has been solved only recently [Gerlach *et al.*, 2025].

These approaches vary greatly and often require sophisticated constructions, but they all share two properties: the considered concrete class always enjoys decidable BCQ entailment, and termination of the considered variant of the chase is always proved decidable. This raises the following question: do we get decidable chase termination (or *fus* membership? or *bts* membership?) for free when considering concrete classes of rules with decidable BCQ entailment? For *fus*, the answer is known to be negative, as shown by [Gaifman *et al.*, 1993], who proved that boundedness of Datalog (which coincides with *fus* membership here) is undecidable. Surprisingly, for chase termination and *bts* membership, the question remains open. We answer it negatively in this paper:

Theorem 1. *There exists a concrete class $\mathcal{C}$ of sets of existential rules such that:*

1. *Chase termination is undecidable in $\mathcal{C}$ for the oblivious, semi-oblivious, restricted, and core chases,*
2. *bts membership is undecidable in $\mathcal{C}$, and*
3. *BCQ entailment is decidable in $\mathcal{C}$.*

This paper is dedicated to proving Theorem 1 by explicitly constructing such a class $\mathcal{C}$, in Section 3, made of sets of rules that simulate Minsky machines. We then establish the negative results regarding chase termination and *bts* membership in Section 4, before proving decidability of BCQ entailment in Section 5. Our simulation of Minsky machines is inspired from [Gogacz and Marcinkowski, 2014], a paper interested in termination of the oblivious, semi-oblivious and restricted chases. We modify their construction in non-trivial ways to also support the core chase and to introduce a flooding mechanism that is instrumental in achieving the properties regarding *bts* membership and BCQ entailment. Detailed proofs can be found in the appendix of the extended version [Larroque and Manière, 2026].

2 Preliminaries

We mostly follow [Carral *et al.*, 2022] for the preliminaries and assume some basic knowledge of first-order logic.

First-Order Logic (FOL) We define Preds, Consts, and Vars to be mutually disjoint, countably infinite sets of *predicates*, *constants*, and *variables*, respectively. Every $\mathtt{P} \in \mathsf{Preds}$ has an *arity* $\mathsf{ar}(\mathtt{P}) \geq 0$. Let $\mathsf{Terms} = \mathsf{Consts} \cup \mathsf{Vars}$ be the set of *terms*. We write lists $t_1, \ldots, t_n$ of terms as $\bar{t}$ and often treat them as sets. For a formula or set thereof φ, let $\mathsf{Preds}(\varphi)$, $\mathsf{Consts}(\varphi)$, $\mathsf{Vars}(\varphi)$, and $\mathsf{Terms}(\varphi)$ be the sets of all predicates, constants, variables, and terms that occur in φ, respectively.

A ($\mathtt{P}$-)*atom* is a FOL formula $\mathtt{P}(\bar{t})$ with $\mathtt{P}$ a $|\bar{t}|$-ary predicate and $\bar{t} \in \mathsf{Terms}$. For a formula φ, $\varphi[\bar{x}]$ indicates that $\bar{x}$ contains all free variables that occur in φ.

Definition 2. An *(existential) rule* R is a FOL formula

$$\forall \bar{x} \forall \bar{y} \left(B[\bar{x}, \bar{y}] \rightarrow \exists \bar{z}\, H[\bar{y}, \bar{z}] \right) \tag{1}$$

where $\bar{x}$, $\bar{y}$, and $\bar{z}$ are pairwise disjoint lists of variables; and B and H are (finite) non-empty conjunctions of constant-free atoms, called the *body* and the *head* of R, respectively. The set $\bar{y}$ is the *frontier* of R, denoted with $\mathsf{fr}(R)$. If $\bar{z}$ is empty, then R is a *Datalog* rule.

We usually omit universal quantifiers in rules.

An *instance* $\mathcal{I}$ is an existentially closed conjunction of atoms. Its *active domain* $\mathsf{adom}(\mathcal{I})$ is the set of all terms occurring in $\mathcal{I}$. A *Boolean conjunctive query* (BCQ) has the same form as an instance, and we often identify both notions. A *knowledge base* (KB) $\mathcal{K}$ is a tuple $\langle \mathcal{R}, \mathcal{I} \rangle$ with $\mathcal{R}$ a rule set and $\mathcal{I}$ an instance. We often identify rule bodies, rule heads, and instances with sets of atoms.

Given atom sets $\mathcal{I}$ and $\mathcal{I}'$, a *homomorphism* h from $\mathcal{I}$ to $\mathcal{I}'$ is a function with domain $\mathsf{Vars}(\mathcal{I})$ such that $h(\mathcal{I}) \subseteq \mathcal{I}'$; h is an *isomorphism* from $\mathcal{I}$ to $\mathcal{I}'$ if additionally, h is injective and h^{-1} is a homomorphism from $\mathcal{I}$ to $\mathcal{I}'$. A homomorphism h from $\mathcal{I}$ to $\mathcal{I}'$ is a *retraction* if h is the identity over $\mathsf{Vars}(\mathcal{I}) \cap \mathsf{Vars}(\mathcal{I}')$.

We identify logical interpretations with atom sets. An atom set $\mathcal{I}$ *satisfies* a rule $R = B \rightarrow \exists \bar{z}\, H$ if, for every homomorphism h from B to $\mathcal{I}$, there is an extension $\hat{h}$ of h with $\hat{h}(H) \subseteq \mathcal{I}$; equivalently, $\mathcal{I}$ is a *model* of R. An atom set $\mathcal{M}$ is a *model of an instance* $\mathcal{I}$ if there is a homomorphism from $\mathcal{I}$ to $\mathcal{M}$, and it is a *model of a KB* $\langle \mathcal{R}, \mathcal{I} \rangle$ if it is a model of $\mathcal{I}$ and satisfies all rules in $\mathcal{R}$. Given KBs or atom sets A and B, A *entails* B, written $A \models B$, if every model of A is a model of B; A and B are *equivalent* if $A \models B$ and $B \models A$. Given atom sets $\mathcal{I}$ and $\mathcal{I}'$, it is known that $\mathcal{I} \models \mathcal{I}'$ iff there is a homomorphism from $\mathcal{I}'$ to $\mathcal{I}$. A model $\mathcal{M}$ of a KB $\mathcal{K}$ is *universal* if there is a homomorphism from $\mathcal{M}$ to every model of $\mathcal{K}$.

Every KB $\mathcal{K}$ admits some (possibly infinite) universal model [Deutsch *et al.*, 2008]. Hence, $\mathcal{K} \models q$ for any BCQ q iff there is a homomorphism from a universal model of $\mathcal{K}$ to q. The *BCQ entailment* problem takes as input a KB $\mathcal{K}$ and a BCQ q and asks if $\mathcal{K} \models q$; it is undecidable [Beeri and Vardi, 1981].

The chase The chase is a family of procedures that repeatedly apply rules until a fixpoint is reached.

Definition 3 (Triggers and derivations)**.** Given an instance $\mathcal{I}$, a *trigger* t on $\mathcal{I}$ is a tuple $\langle R, \pi \rangle$ with $R = B \rightarrow \exists \bar{z}\, H$ a rule and π a homomorphism from B to $\mathcal{I}$. Let $\mathsf{supp}(t) = \pi(B)$ and $\mathsf{out}(t) = \pi^R(H)$, where π^R extends π by mapping all $z \in \bar{z}$ to the fresh variable z_t that is unique for z and t. A *derivation* from a KB $\langle \mathcal{R}, \mathcal{I} \rangle$ is a sequence $\mathcal{D} = (\emptyset, \mathcal{I}_0), (t_1, \mathcal{I}_1), \ldots$ such that:

1. Every $\mathcal{I}_i$ in $\mathcal{D}$ is an instance; moreover, $\mathcal{I}_0 = \mathcal{I}$.
2. Every t_i in $\mathcal{D}$ is a trigger $\langle R, \pi \rangle$ on $\mathcal{I}_{i-1}$ such that $R \in \mathcal{R}$, $\mathsf{out}(t_i) \not\subseteq \mathcal{I}_{i-1}$, and $\mathcal{I}_i = \mathcal{I}_{i-1} \cup \mathsf{out}(t_i)$.

The *result* $\mathsf{res}(\mathcal{D})$ of $\mathcal{D}$ is the union of all instances in $\mathcal{D}$, and $\mathsf{trigs}(\mathcal{D})$ is the set of all triggers in $\mathcal{D}$.

Different chase variants build specific derivations according to different criteria of trigger applicability. Below, the letters

$\mathbb{O}$, $\mathbb{SO}$, $\mathbb{R}$, and $\mathbb{E}$ respectively refer to the oblivious, semi-oblivious, restricted, and equivalent[1] variants.

Definition 4 (Applicability)**.** A trigger $t = \langle R, h\rangle$ on an instance $\mathcal{I}$ is $\mathbb{O}$*-applicable* on $\mathcal{I}$ if $\mathsf{out}(t) \not\subseteq \mathcal{I}$; $\mathbb{SO}$*-applicable* on $\mathcal{I}$ if $\mathsf{out}(t') \not\subseteq \mathcal{I}$ for every trigger $t' = (R, \pi')$ with $\pi(\mathsf{fr}(R)) = \pi'(\mathsf{fr}(R))$; $\mathbb{R}$*-applicable* on $\mathcal{I}$ if there is no retraction from $\mathcal{I} \cup \mathsf{out}(t)$ to $\mathcal{I}$; $\mathbb{E}$*-applicable* on $\mathcal{I}$ if there is no homomorphism from $\mathcal{I} \cup \mathsf{out}(t)$ to $\mathcal{I}$.

Example 1. *Consider the KB $\mathcal{K} = \langle \mathcal{R}, \mathcal{I}\rangle$ with $\mathcal{R} = \{R = \mathtt{P}(x,y) \to \exists z\, \mathtt{P}(y,z), \mathtt{P}(z,z)\}$ and $\mathcal{I} = \{\mathtt{P}(a,a)\}$ for some a. The trigger $t_1 = (R, \{x \mapsto a, y \mapsto a\})$, whose output is $\mathsf{out}(t_1) = \{\mathtt{P}(a, z_{t_1}), \mathtt{P}(z_{t_1}, z_{t_1})\}$, is $\mathbb{O}$ and $\mathbb{SO}$-applicable on $\mathcal{I}$ but not $\mathbb{R}$ or $\mathbb{E}$-applicable, since there is a retraction from $\mathcal{I} \cup \mathsf{out}(t_1)$ to $\mathcal{I}$ that maps z_{t_1} to a.*

Definition 5 ($\mathbb{X}$-Chase)**.** For an $\mathbb{X} \in \{\mathbb{O}, \mathbb{SO}, \mathbb{R}, \mathbb{E}\}$, an $\mathbb{X}$*-derivation from a KB* $\mathcal{K} = \langle \mathcal{R}, \mathcal{I}\rangle$ is a derivation $\mathcal{D}$ such that every trigger $t_i \in \mathsf{trigs}(\mathcal{D})$ is $\mathbb{X}$-applicable on $\mathcal{I}_i$. An $\mathbb{X}$-derivation $\mathcal{D}$ is *fair* if for every $\mathcal{I}_i$ occurring in $\mathcal{D}$ and $\mathbb{X}$-applicable trigger t on $\mathcal{I}_i$, there is some $j > i$ such that t is not $\mathbb{X}$-applicable on $\mathcal{I}_j$. An $\mathbb{X}$-derivation is *terminating* if it is fair and finite.

The result of any fair $\mathbb{X}$-derivation is a universal model of the KB, for $\mathbb{X} \in \{\mathbb{O}, \mathbb{SO}, \mathbb{R}\}$, and has a retraction to a universal model for $\mathbb{X} = \mathbb{E}$. Therefore, we obtain:

Proposition 6. *Consider a BCQ q, a KB $\mathcal{K}$, and some fair $\mathbb{X}$-derivation $\mathcal{D}$ from $\mathcal{K}$ where $\mathbb{X} \in \{\mathbb{O}, \mathbb{SO}, \mathbb{R}, \mathbb{E}\}$. Then, $\mathcal{K} \models q$ iff $\mathsf{res}(\mathcal{D}) \models q$.*

These chase variants induce abstract classes, whose relationship is summarized in [Carral *et al.*, 2022].

Definition 7 (Chase-Terminating Sets)**.** For an $\mathbb{X} \in \{\mathbb{O}, \mathbb{SO}, \mathbb{R}, \mathbb{E}\}$, let $\mathrm{CT}^{\mathbb{X}}_{\forall}$ (resp. $\mathrm{CT}^{\mathbb{X}}_{\exists}$) be the set of all rule sets $\mathcal{R}$ such that every (resp. some) fair $\mathbb{X}$-derivation from every KB $\langle \mathcal{R}, \mathcal{I}\rangle$ is finite.

Proposition 8. *For all $\mathbb{X} \in \{\mathbb{O}, \mathbb{SO}, \mathbb{E}\}$, $\mathrm{CT}^{\mathbb{X}}_{\forall} = \mathrm{CT}^{\mathbb{X}}_{\exists}$, and $\mathrm{CT}^{\mathbb{O}}_{\forall} \subset \mathrm{CT}^{\mathbb{SO}}_{\forall} \subset \mathrm{CT}^{\mathbb{R}}_{\forall} \subset \mathrm{CT}^{\mathbb{R}}_{\exists} \subset \mathrm{CT}^{\mathbb{E}}_{\forall}$.*

Example 2. *Consider the KB $\mathcal{K} = \langle \mathcal{R}, \mathcal{I}\rangle$ from Example 1. All fair $\mathbb{O}$- or $\mathbb{SO}$-derivations from $\mathcal{K}$ are infinite. The only fair $\mathbb{R}$ and $\mathbb{E}$-derivation from $\mathcal{K}$ is $\mathcal{D} = (\emptyset, \mathcal{I})$. Any fair $\mathbb{R}$-derivation from a KB with $\mathcal{R}$ is finite and hence, $\mathcal{R} \in \mathrm{CT}^{\mathbb{R}}_{\forall}$ (and thus $\mathcal{R} \in \mathrm{CT}^{\mathbb{E}}_{\forall}$).*

bts The class *bts* is defined from the notion of treewidth and contains $\mathrm{CT}^{\mathbb{X}}_{Q}$ for all $\mathbb{X} \in \{\mathbb{O}, \mathbb{SO}, \mathbb{R}, \mathbb{E}\}$ and $Q \in \{\exists, \forall\}$ [Calì *et al.*, 2013]. We only introduce a sufficient condition for a rule set not to be *bts*. Given a binary predicate $\mathtt{P}$, the $\mathtt{P}$*-graph* of an instance $\mathcal{I}$ is the directed graph $(\mathsf{adom}(\mathcal{I}), E)$ where $(a, b) \in E$ if and only if $\mathtt{P}(a, b) \in \mathcal{I}$.

Proposition 9. *If there is an infinite clique in the $\mathtt{P}$-graph of $\mathsf{res}(\mathcal{D})$ for some binary predicate $\mathtt{P}$ and derivation $\mathcal{D}$ from $\mathcal{K} = \langle \mathcal{I}, \mathcal{R}\rangle$, then $\mathcal{R}$ is not bts.*

[1]The equivalent chase, like the better-known core chase, halts exactly when the KB has a finite universal model [Delivorias *et al.*, 2021], while being monotonic ($\forall i, \mathcal{I}_i \subseteq \mathcal{I}_{i+1}$).

2.1 Three-Counter Machines

A *three-counter machine* (3CM) $\mathcal{M}$ is essentially a two-counter automaton augmented with a strictly increasing time counter. Formally, it is a tuple $\langle Q, q_1, \delta\rangle$ where Q is a finite set of states including the *initial state* q_1, and $\delta : Q \times \{0,1\}^2 \to Q \times \{-1, 0, 1\}^2$ is a transition function. A *configuration* of $\mathcal{M}$ is a tuple $\langle q, v_1, v_2, t\rangle$ where $q \in Q$ is the current state and $v_1, v_2, t \in \mathbb{N}$ are the values of the three counters. The *initial configuration* of $\mathcal{M}$ is $\langle q_1, 0, 0, 0\rangle$. Given a configuration $C = \langle q, v_1, v_2, t\rangle$, we define $\mathsf{next}(C) = \langle q', v_1', v_2', t+1\rangle$ where $\delta(q, b_1, b_2) = \langle q', d_1, d_2\rangle$ where $b_i = 0$ if $v_i = 0$ and $b_i = 1$ otherwise, and $v_i' = \max(v_i + d_i, 0)$ for $i \in \{1, 2\}$. Note that next is a partial function since δ may be undefined for some inputs. A 3CM $\mathcal{M}$ *halts* if there is a finite sequence of configurations $C_0, \ldots, C_n$ such that C_0 is the initial configuration, $C_{i+1} = \mathsf{next}(C_i)$ for every $0 \leq i < n$, and $\mathsf{next}(C_n)$ is undefined. The halting problem for 3CMs is undecidable [Minsky, 1967].

3 The Rule Set

We define the class $\mathcal{C} = \{\, \mathcal{R}_{\mathcal{M}} \mid \mathcal{M} \text{ a 3CM} \,\}$, using the rule set $\mathcal{R}_{\mathcal{M}}$ given in Figure 1. As it is recognizable in linear time, it is indeed a concrete class. The rule set $\mathcal{R}_{\mathcal{M}}$ is a adapted from the one used to prove Theorem 1 in [Gogacz and Marcinkowski, 2014], stating that all-instance chase termination is undecidable for the oblivious chase. It provides a reduction from the halting problem for 3CMs to chase termination.

Roughly speaking, the chase on KB $\langle\{\mathtt{End}(w)\}, \mathcal{R}_{\mathcal{M}}\rangle$ looks like a sequence of elements connected through a binary predicate $\mathtt{S}$, ending in the "well of positivity" w. Each element in the chase runs its own private computation of $\mathcal{M}$, and is only allowed to generate a predecessor if the simulation of $\mathcal{M}$ goes on for long enough. Consequently, this chain of elements is infinite if and only if $\mathcal{M}$ does not halt. To elaborate, we first need to explain how $\mathcal{M}$ can be simulated by a sequence of natural numbers.

Following Section 3.3 in [Gogacz and Marcinkowski, 2014], there is a way to encode configurations of a 3CM $\mathcal{M}$ as natural numbers using prime factorization. Let $p_1, p_2, \ldots$ be the sequence of prime numbers, and $Q = \{q_1, \ldots, q_m\}$. Then, a configuration $\langle q_i, v_1, v_2, t\rangle$ of $\mathcal{M}$ is encoded as the natural number

$$\mathsf{enc}(q_i, v_1, v_2, t) = p_i \cdot p_{m+1}^{v_1} \cdot p_{m+2}^{v_2} \cdot p_{m+3}^{t}.$$

In particular, the initial configuration $\langle q_1, 0, 0, 0\rangle$ is encoded as $2 = p_1$. Let $p = p_1 \ldots p_{m+3}$.

Proposition 10. *There exist natural numbers q_i, r_i for all $1 \leq i \leq p$ such that $\frac{q_i}{r_i}$ is irreducible, and for any configuration C of $\mathcal{M}$ such that $i = \mathsf{enc}(C) \bmod p$, if $\mathsf{next}(C)$ is defined, then $\mathsf{enc}(\mathsf{next}(C)) = \frac{q_i}{r_i} \cdot \mathsf{enc}(C)$, and $\frac{q_i}{r_i} \cdot \mathsf{enc}(C) = \mathsf{enc}(C)$ otherwise.*

For further details, we refer the reader to Theorem 5 in [Gogacz and Marcinkowski, 2014]. Let g be the function defined as $g(n) = \frac{q_{n \bmod p}}{r_{n \bmod p}} \cdot n$. Then, $\mathsf{next}(C)$ is defined if and only if $g(\mathsf{enc}(C)) \neq \mathsf{enc}(C)$, and in that case $\mathsf{enc}(\mathsf{next}(C)) = g(\mathsf{enc}(C))$. Let $\mathcal{G} = \{\, g^i(2) \mid i \in \mathbb{N} \,\}$.

$$\mathtt{S}_0(x,y) \to \mathtt{S}(x,y) \qquad (R_{\mathtt{S}}) \qquad\qquad \mathtt{End}(x) \to \mathtt{S}(x,x) \qquad (R_{\mathtt{S}}^{\mathtt{End}})$$

$$\mathtt{R}_i(x,y), \mathtt{S}(y,z) \to \mathtt{R}_{i+1}(x,z) \qquad (R_{\mathtt{R},i}) \qquad\qquad \mathtt{S}^2(x,y) \to \mathtt{G}(x,y) \qquad (R_{\mathtt{G}}^{\mathsf{init}})$$

$$\mathtt{T}_i(x,y,z), \mathtt{S}^{r_i}(y,y'), \mathtt{S}^{q_i}(z,z') \to \mathtt{T}_i(x,y',z') \qquad (R_{\mathtt{T},i}) \qquad\qquad \mathtt{G}(x,y), \mathtt{R}_i(x,y), \mathtt{T}_i(x,y,z) \to \mathtt{G}(x,z) \qquad (R_{\mathtt{G},i}^{\mathsf{step}})$$

$$\mathtt{S}(x,y) \to \mathtt{Flood}(y) \qquad (R_{\mathsf{flood}}^{\mathsf{prop}})$$

$$\mathtt{Flood}(x), \mathtt{Flood}(y), \mathtt{Flood}(z) \to \mathtt{G}(x,y), \bigwedge_{j=0}^{p-1} \mathtt{R}_j(x,y), \mathtt{T}_j(x,y,z) \qquad (R_{\mathsf{flood}}^{\mathsf{gen}})$$

$$\mathtt{G}(y,z), \mathtt{End}(z) \to \exists x\, \mathtt{S}_0(x,y), \mathtt{R}_0(x,x), \bigwedge_{j=0}^{p-1} \mathtt{T}_j(x,x,x) \qquad (R_{\exists})$$

Figure 1: Rules in $\mathcal{R}_{\mathcal{M}}$ for all $i < p$. In Rule $R_{\mathtt{R},i}$, $\mathtt{R}_p$ is interpreted as $\mathtt{R}_0$. $\mathtt{S}^n(x,y)$ denotes an $\mathtt{S}$-path of length n from x to y.

Since the third counter always increases when a subsequent configuration exists, we get the following.

Proposition 11. *$\mathcal{M}$ halts* iff *$\mathcal{G}$ is bounded.*

Then, as mentioned earlier, the chase is shaped as a sequence of elements that form a chain connected by predicates $\mathtt{S}$, where $\mathtt{S}(x,y)$ means that y thinks it is the successor of x. The use of "thinks" is intentional, as we run a simulation of $\mathcal{M}$ from the point of view of each element. All predicates used for the simulation ($\mathtt{G}$, $\mathtt{R}_i$ and $\mathtt{T}_i$) feature this element in first position, which thinks itself as 0. The atom $\mathtt{G}(x,y)$ signifies that from the point of view of x, the number represented by y belongs to $\mathcal{G}$. This relation is computed according to Proposition 10, using Rules $R_{\mathtt{G}}^{\mathsf{init}}$ and $R_{\mathtt{G},i}^{\mathsf{step}}$. Specifically, the initial configuration is encoded by 2, represented by the second successor of each element (Rule $R_{\mathtt{G}}^{\mathsf{init}}$). Rule $R_{\mathtt{G},i}^{\mathsf{step}}$ then implements the transition to the next configuration using predicates $\mathtt{R}_i$ and $\mathtt{T}_i$. The atom $\mathtt{R}_i(x,y)$ indicates that, from x's perspective, the number represented by y has a remainder of i modulo p, as generated by Rule $R_{\mathtt{R},i}$. Similarly, $\mathtt{T}_i(x,y,z)$ implies that $\frac{y}{z} = \frac{r_i}{q_i}$ from x's perspective. This is ensured by Rule $R_{\mathtt{T},i}$, which implements multiplication by successive additions. These predicates are initialized for each element x by Rule $R_{\exists}$. This rule allows an element y to create a predecessor x (specifically $\mathtt{S}_0(x,y)$, which implies $\mathtt{S}(x,y)$ via Rule $R_{\mathtt{S}}$) if the simulation of $\mathcal{M}$ from y reaches the well of positivity w (i.e., when $\mathtt{G}(y,z)$ and $\mathtt{End}(z)$ hold for some z). This process is illustrated in Figure 2 for predicate $\mathtt{T}_i$.

The reduction described so far mirrors the one in [Gogacz and Marcinkowski, 2014]. However, the same reduction does not work here. First, it applies only to the oblivious chase; the restricted and equivalent chases always terminate, even if $\mathcal{M}$ does not halt. Second, arguing that BCQ entailment is decidable is difficult (and likely false) because the predicates $\mathtt{R}_i$, $\mathtt{T}_i$, and $\mathtt{G}$ lack structure. We introduce two modifications to address these issues.

First, to accommodate all chase variants, we introduce predicates $\mathtt{S}_0$ and $\mathtt{End}$, which derive $\mathtt{S}$ via Rules $R_{\mathtt{S}}$ and $R_{\mathtt{S}}^{\mathtt{End}}$. In particular, $\mathtt{S}_0$ has no self-loop over the well of positivity w (where $\mathtt{End}(w)$ holds, at the end of the chain), preventing any attempt at homomorphically embedding the whole chain into w. Thus, even the equivalent chase is forced to create an infinite chain if $\mathcal{M}$ does not halt.

Second, to make BCQ entailment decidable, we employ a flooding mechanism using the predicate $\mathtt{Flood}$, with Rules $R_{\mathsf{flood}}^{\mathsf{gen}}$ and $R_{\mathsf{flood}}^{\mathsf{prop}}$. This ensures that once the simulation from x reaches w and generates a predecessor, x connects to all its successors via $\mathtt{R}_i$, $\mathtt{T}_i$, and $\mathtt{G}$. As a result, all elements whose simulation reaches the well of positivity show the same behavior regarding BCQ answering, except for predicates $\mathtt{S}_0$, $\mathtt{S}$ and $\mathtt{End}$ which are easier to handle.

4 Undecidability of Chase Termination and *bts*-Recognition

This section is devoted to Points 1 and 2 of Theorem 1, which are the negative properties of our class: chase termination for all chase variants and *bts* are both undecidable in $\mathcal{C}$. Regarding Oblivious and Semi-Oblivious chase termination, this is mostly guaranteed *by construction*, as we adapted the class proposed in [Gogacz and Marcinkowski, 2014] that already had these properties. Remarkably, our set of rules also exhibits this behavior for the equivalent chase. To prove this, we make use of Proposition 8 as much as possible. More precisely, we show that if $\mathcal{M}$ does not halt, then the equivalent chase does not terminate on $\mathcal{R}_{\mathcal{M}}$ and some specific instance $\mathcal{I}_{\mathtt{E}}$ (Theorem 14), and that if $\mathcal{M}$ does halt, then the oblivious chase terminates on the so-called critical instance, and thus on all instances (Theorem 16). These two results, along with Proposition 8 and the undecidability of the halting problem for 3CMs entail that chase termination is undecidable for $\mathcal{C}$. The undecidable *bts* recognition is then a consequence of the above, joint with our freshly added flooding mechanism.

In this section, $\mathbb{X}$ always denotes an element of $\{\mathbb{O}, \mathbb{SO}, \mathbb{R}, \mathbb{E}\}$. We denote the result of some fair $\mathbb{X}$-derivation from $\mathcal{I}, \mathcal{R}$ with $\mathsf{Ch}^{\mathbb{X}}(\mathcal{I}, \mathcal{R})$. Note that this object is not unique in general, as different derivations may produce different results. This is however not an issue for our purpose, since we do not really consider the restricted chase in proofs, and it is the only variant for which the order of application of triggers matters. It will also be convenient to consider the chase as an alternation of closing under Datalog rules and then under non-Datalog rules. We define $\mathsf{Ch}_0^{\mathbb{X}}(\mathcal{I}) = \mathsf{Ch}^{\mathbb{X}}(\mathcal{I}, \mathcal{R}_{\mathcal{M}}^{\forall})$ and for all $i \geq 0$, $\mathsf{Ch}_{i+1}^{\mathbb{X}}(\mathcal{I}) = \mathsf{Ch}^{\mathbb{X}}(\mathsf{Ch}^{\mathbb{X}}(\mathsf{Ch}_i^{\mathbb{X}}(\mathcal{I}), \mathcal{R}_{\mathcal{M}}^{\exists}), \mathcal{R}_{\mathcal{M}}^{\forall})$, where $\mathcal{R}^{\forall}$ and $\mathcal{R}^{\exists}$ denote the sets of Datalog and non-Datalog rules in $\mathcal{R}$, respectively. Note that this process still induces a fair derivation.

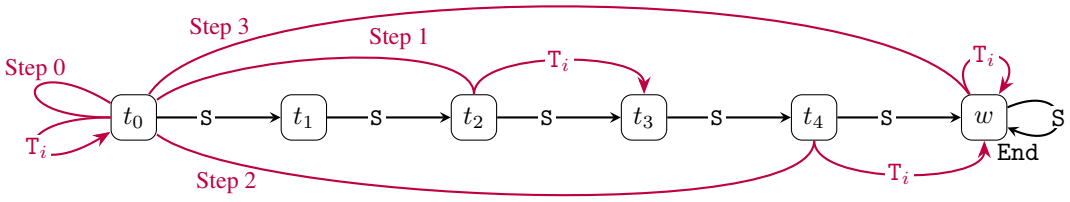

Figure 2: Representation of predicates $\mathtt{S}_0$, $\mathtt{End}$ and atoms with shape $\mathtt{T}_i(t_0, \_, \_)$ in $\mathsf{Ch}_5^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ for some i such that $\frac{r_i}{q_i} = \frac{2}{3}$. The $\mathtt{S}_0$-atoms, as described in Lemma 12, form a chain ending in w, the only term with $\mathtt{End}(w)$. Then, $\mathtt{T}_i$-atoms are as described by Lemma 13: First, $\mathtt{T}_i(t_0, t_0, t_0)$ is created by the application of Rule $(R_{\exists})$ that creates t_0 (step 0). Then, $\mathtt{T}_i(t_0, t_2, t_3)$, $\mathtt{T}_i(t_0, t_4, t_6)$, and $\mathtt{T}_i(t_0, t_6, t_9)$ (note that $w = t_6 = t_9$) are created by successive applications of Rule $(R_{\mathtt{T},i})$ (steps 1, 2 and 3, respectively).

4.1 Non-Termination of the Chase

We first prove that if $\mathcal{M}$ does not halt, then $\mathcal{R}_{\mathcal{M}} \notin \mathrm{CT}_Q^{\mathbb{X}}$ for $Q \in \{\forall, \exists\}$, by considering only the instance $\mathcal{I}_{\mathtt{E}} = \{\mathtt{End}(w)\}$. Before proving this, we need two lemmas about the structure of the chase.

Lemma 12. *For all $n > 0$, if $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}}) \neq \mathsf{Ch}_{n-1}^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$, then $\mathsf{adom}(\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})) = \{t_0, \ldots, t_{n-1}, w\}$ for some terms $t_0, \ldots, t_{n-1}$, and the $\mathtt{S}_0$-atoms of $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ are exactly $\{\, \mathtt{S}_0(t_i, t_{i+1}) \mid 0 \leq i < n \,\}$ where $t_n = w$.*

Proof sketch. By induction on n. Both the base case and inductive step follow from the fact that there is a single existential rule, and thus the only way for $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ to be different from $\mathsf{Ch}_{n-1}^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ is for Rule $R_{\exists}$ to be applied. This rule features a single frontier variable, which has to be mapped to the term created at the previous step. □

In the following lemma, proven by a simple induction, we refer to terms of $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ as $t_0, \ldots, t_{n-1}$ as in Lemma 12. We also use t_i for $i \geq n$ to refer to w. Note that even for such i's, $\mathtt{S}(t_i, t_{i+1}) \in \mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$.

Lemma 13. *For all $n > 0$, if $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}}) \neq \mathsf{Ch}_{n-1}^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$,*

(i) $\mathtt{R}_i(t_0, t_k) \in \mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ *if and only if* $k = i \mod p$;

(ii) $\mathtt{T}_i(t_0, t_k, t_l) \in \mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ *if and only if there is m such that $k = mr_i$ and $l = mq_i$;*

(iii) For all $k \in \mathbb{N}$, $\mathtt{G}(t_0, t_{g^k(2)}) \in \mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$;

(iv) If $\mathcal{G}$ is bounded by $n-1$, then for all l such that $\mathtt{G}(t_0, t_l) \in \mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$, there is k such that $l = g^k(2)$.

Theorem 14. *If $\mathcal{M}$ does not halt, there is no terminating $\mathbb{X}$-chase sequence from $\langle \{\mathtt{End}(w)\}, \mathcal{R}_{\mathcal{M}} \rangle$.*

Proof sketch. We simply prove by induction on n that if $\mathcal{M}$ does not halt and $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}}) \neq \mathsf{Ch}_{n-1}^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$, then Rule $R_{\exists}$ is $\mathbb{E}$-applicable on $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$. In this case, $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ is as described in Lemma 12. Then, by Proposition 11, $\mathcal{G}$ is unbounded, so $\mathtt{G}(t_0, w) \in \mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ by Item (iv) of Lemma 13. As the $\mathtt{S}_0$-atoms of $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$ are exactly $\{\, \mathtt{S}_0(t_i, t_{i+1}) \mid 0 \leq i < n \,\}$, there is no way to homomorphically map the result of applying $R_{\exists}$ into $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$, so this rule is indeed $\mathbb{E}$-applicable. □

4.2 Termination of the Chase

To prove that the $\mathbb{X}$-chase terminates if $\mathcal{M}$ halts (Theorem 16), it suffices to prove that in this case, the oblivious chase terminates on the so-called critical instance. The critical instance is the instance containing a single constant w, and in which all possible facts over the schema of the rule set are present. Indeed, if the oblivious chase terminates on the critical instance, it terminates on all instances [Marnette, 2009], and thus all chase variants terminate too via Proposition 8. Let $\mathcal{J}$ be the critical instance for the schema of $\mathcal{R}_{\mathcal{M}}$, and $\mathcal{J}_n = \mathsf{Ch}_n^{\mathbb{X}}(\mathcal{J})$.

Lemma 15. *For all $n \in \mathbb{N}$, $\mathcal{J}_n \simeq \mathsf{Ch}_n^{\mathbb{O}}(\mathcal{I}_{\mathtt{E}}) \cup \{\mathtt{S}_0(w, w)\}$.*

Proof sketch. By induction on n. The base case is obtained using Rules $R_{\mathtt{S}}^{\mathsf{End}}$, $R_{\mathsf{flood}}^{\mathsf{prop}}$ and $R_{\mathsf{flood}}^{\mathsf{gen}}$. For the inductive step, the additional $\mathtt{S}_0$-atom of $\mathcal{J}_n$ can only produce $\mathtt{S}(w, w)$, which already belongs to $\mathcal{J}$. Thus, both instances feature the same applicable triggers. □

Theorem 16. *If $\mathcal{M}$ halts, then the $\mathbb{X}$-chase terminates on $\langle \mathcal{I}, \mathcal{R}_{\mathcal{M}} \rangle$ for all instances $\mathcal{I}$.*

Proof sketch. If $\mathcal{M}$ halts, then $\mathcal{G}$ is bounded by n and by Items (iii) and (iv) of Lemma 13, $\mathtt{G}(t_0, w) \notin \mathcal{J}_{n+1}$. Thus, Rule $R_{\exists}$ is not applicable on $\mathcal{J}_{n+1}$, which entails the result as discussed before Lemma 15. □

4.3 Treewidth

We then prove that *bts*-membership in $\mathcal{C}$ is undecidable.

Theorem 17. *$\mathcal{R}_{\mathcal{M}}$ is bts if and only if $\mathcal{M}$ halts.*

If $\mathcal{M}$ halts, then by Theorem 16, the $\mathbb{X}$-chase terminates, so the rule set is clearly *bts*. The converse direction is a direct consequence of the flooding mechanism.

Lemma 18. *If $\mathcal{M}$ does not halt, then there is an infinite clique in the $\mathtt{R}_0$-graph of $\mathsf{Ch}^{\mathbb{X}}(\{\mathtt{End}(w)\}, \mathcal{R}_{\mathcal{M}})$.*

Proof. Assume $\mathcal{M}$ does not halt. By Theorem 14, for all n, $\mathsf{Ch}_n^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}}) \neq \mathsf{Ch}_{n-1}^{\mathbb{X}}(\mathcal{I}_{\mathtt{E}})$, so by Lemma 12, there is an infinite sequence of terms $u_0, u_1, \ldots$ such that $u_0 = w$, and for all $i \geq 0$, $\mathtt{S}(u_{i+1}, u_i) \in \mathsf{Ch}^{\mathbb{X}}(\{\mathtt{End}(w)\}, \mathcal{R}_{\mathcal{M}})$. By Rule $R_{\mathsf{flood}}^{\mathsf{prop}}$, for all $i, j \geq 0$, $\mathtt{Flood}(u_i) \in \mathsf{Ch}^{\mathbb{X}}(\{\mathtt{End}(w)\}, \mathcal{R}_{\mathcal{M}})$, so by Rule $R_{\mathsf{flood}}^{\mathsf{gen}}$, $\mathtt{R}_0(u_i, u_j)$ is also in the chase result, so the $\mathtt{R}_0$-graph contains an infinite clique. □

Lemma 18 and Proposition 9 conclude the proof.

5 Decidability of BCQ Entailment

We now turn to decidability of BCQ entailment, that is Point 3 of Theorem 1. Consider an instance $\mathcal{I}$, a set of rules $\mathcal{R}_\mathcal{M}$ from our class $\mathcal{C}$ constructed in Section 3, and a BCQ q. To decide whether $\langle \mathcal{I}, \mathcal{R}_\mathcal{M} \rangle \models q$, our algorithm computes the result of the semi-oblivious chase up to depth $M := 2\,|q| + 1$, that is $\mathsf{Ch}^{\mathbb{SO}}_M(\mathcal{I})$, and then checks whether $\mathsf{Ch}^{\mathbb{SO}}_M(\mathcal{I}) \models q$. This clearly terminates as we only require to apply finitely many rules in the first step, and then evaluate a usual BCQ on a finite instance in the second step. It remains to prove the correctness:

Lemma 19. $\langle \mathcal{I}, \mathcal{R}_\mathcal{M} \rangle \models q$ *iff* $\mathsf{Ch}^{\mathbb{SO}}_M(\mathcal{I}) \models q$.

The rest of this section is devoted to proving the above. We first define $\mathcal{I}_n := \mathsf{Ch}^{\mathbb{SO}}_n(\mathcal{I})$ for all n, and $\mathcal{I}_\infty := \bigcup_{n \geq 0} \mathcal{I}_n$. The successive $\mathcal{I}_n$ again induce a fair derivation, therefore $\mathcal{I}_\infty$ is a universal model of $\langle \mathcal{I}, \mathcal{R}_\mathcal{M} \rangle$. Thus, Lemma 19 can be rephrased as $\mathcal{I}_\infty \models q$ iff $\mathcal{I}_M \models q$ using Proposition 6. With this formulation, the 'if' direction is the easiest one: indeed, if $\mathcal{I}_M \models q$, we get $\mathcal{I}_\infty \models q$ from the fact that $\mathcal{I}_\infty$ contains $\mathcal{I}_M$.

We now turn to proving the 'only-if' direction of Lemma 19. Assume $\mathcal{I}_\infty \models q$. Compactness guarantees that there exists N such that $\mathcal{I}_N \models q$. If $N \leq M$, we are done since $\mathcal{I}_N \subseteq \mathcal{I}_M$. Otherwise, if $N > M$, consider a homomorphism h from q to $\mathcal{I}_N$. We want to construct a homomorphism $h' : q \to \mathcal{I}_M$ based on h. Our main issue is atoms that h maps to $\mathcal{I}_N \setminus \mathcal{I}_M$, for which we need to find replacements within $\mathcal{I}_M$.

Intuitively, we distinguish between the $\mathtt{End}$-atoms (which cannot be introduced during the chase), the $\mathtt{S}_0$-atoms, the $\mathtt{S}$-atoms, and the others. Indeed, apart from within the original instance $\mathcal{I}$, the $\mathtt{S}_0$- and $\mathtt{S}$-atoms form chains ending on some element from $\mathsf{adom}(\mathcal{I})$. Therefore, if h involves such $\mathtt{S}_0$- and $\mathtt{S}$-atoms from $\mathcal{I}_N \setminus \mathcal{I}_M$, we know they lie along a chain, say at distance ℓ to $\mathcal{I}$. We can then prove that a sufficient part of this chain already belongs to $\mathcal{I}_M$, due to $M \geq 2\,|q|$. To build h', we can thus 'shift' h by $\ell - M$ atoms to use instead this part of the chain. Furthermore, we can establish that *every* term from $\mathcal{I}_N$ lies on such a chain, therefore, when we shift our homomorphism along the chain, due to the combination of Rules $R^{\mathsf{prop}}_{\mathsf{flood}}$ and $R^{\mathsf{gen}}_{\mathsf{flood}}$, all the $\mathtt{P}$-atoms are present for $\mathtt{P} \notin \{\mathtt{End}, \mathtt{S}_0, \mathtt{S}\}$. The only exception is when such a $\mathtt{P}$-atom features a term t that is the starting point of a chain (thus Rule $R^{\mathsf{prop}}_{\mathsf{flood}}$ may not trigger on t) and that $\ell \leq M$ (so that we don't shift). But $\ell \leq M$ guarantees the atom already belongs to $\mathcal{I}_M$.

We summarize considerations on $\mathtt{S}_0$- and $\mathtt{S}$-atoms in the next lemma, whose proof is a simple induction:

Lemma 20. *For all $n \geq 0$ and $a \in \mathsf{adom}(\mathcal{I})$, there exist integers $0 \leq m_a \leq n$ s.t.: (i) the domain of $\mathcal{I}_n$ consists of distinct terms a_i with $a \in \mathsf{adom}(\mathcal{I})$ and $0 \leq i \leq m_a$, and where $a_0 = a$; and (ii) $\mathtt{S}_0$ atoms of $\mathcal{I}_n$ are exactly:*

$$\{\mathtt{S}_0(a,b) \mid \mathtt{S}_0(a,b) \in \mathcal{I}\}$$
$$\cup\, \{\mathtt{S}_0(a_{i+1}, a_i) \mid 0 \leq i < m_a, a \in \mathsf{adom}(\mathcal{I})\},$$

and, as a corollary, the $\mathtt{S}$ atoms of $\mathcal{I}_n$ are exactly:

$$\{\mathtt{S}(a,b) \mid \mathtt{S}(a,b) \in \mathcal{I} \text{ or } \mathtt{S}_0(a,b) \in \mathcal{I}\}$$
$$\cup\, \{\mathtt{S}(a,a) \mid \mathtt{End}(a) \in \mathcal{I}\}$$
$$\cup\, \{\mathtt{S}(a_{i+1}, a_i) \mid 0 \leq i < m_a, a \in \mathsf{adom}(\mathcal{I})\}.$$

Note that the above echoes Lemma 12, except that the end point can now be any element from the original instance $\mathcal{I}$. Notice that using the semi-oblivious chase also prevents branching: each element from $\mathsf{adom}(\mathcal{I})$ can only be the end point of one such chain of fresh elements. The result of the chase is depicted in Figure 3.

We now clarify that, as the chase progresses, new facts either stem from the flooding mechanism or involve one of the freshly introduced starting points of a chain.

Lemma 21. *Let $n \geq 0$ and $\mathtt{P}(t_1, \ldots, t_m)$ an atom. If $\mathtt{P}(t_1, \ldots, t_m) \in \mathcal{I}_{n+1} \setminus \mathcal{I}_n$ then one of the following holds:*

1. *$t_1 \in \mathsf{adom}(\mathcal{I}_{n+1}) \setminus \mathsf{adom}(\mathcal{I}_n)$ and there are $k_2, \ldots, k_m \geq 0$ s.t. $\mathtt{S}^{k_2}(t_1, t_2), \ldots, \mathtt{S}^{k_m}(t_1, t_m) \in \mathcal{I}_{n+1}$, where $\mathtt{S}^k(x,y)$ denotes an $\mathtt{S}$-path of size k from x to y.*
2. *for every $1 \leq i \leq m$, we have $\mathtt{Flood}(t_i) \in \mathcal{I}_{n+1}$.*

Proof sketch. Apart from Rules $R^{\mathsf{prop}}_{\mathsf{flood}}$ and $R^{\mathsf{gen}}_{\mathsf{flood}}$ that relate to the flooding, notice that the first term occurring in an atom of some head also appears as the first term in an atom from the body. The rest follows by induction. □

The above guarantees that any element a_i, as described in Lemma 20, appears *exactly* at depth i in the chase:

Lemma 22. *Let a_i be a term from $\mathsf{adom}(\mathcal{I}_n)$ as described in Lemma 20, thus with $i \leq n$. We have $a_i \in \mathsf{adom}(\mathcal{I}_i)$.*

Proof sketch. We proceed by induction on i. For a_0, the claim is trivial as $a_0 \in \mathsf{adom}(\mathcal{I})$. For a_{i+1}, we know it is introduced by applying Rule $R_\exists$ on a_i (mapping y to a_i), thus relying on an atom $\mathtt{G}(a_i, e)$ to be satisfied for some $\mathtt{End}(e) \in \mathcal{I}$. Using Lemma 21, one can establish that $\mathtt{G}(a_i, e) \in \mathcal{I}_i$, thus a_{i+1} is introduced at step $i+1$. □

To proceed with the construction of homomorphism $h' : q \to \mathcal{I}_M$ as previously sketched. We thus decompose the homomorphism $h : q \to \mathcal{I}_N$ to identify which parts of q are mapped along $\mathtt{S}_0$-$\mathtt{S}$-chains. To do so, we define the $\{\mathtt{S}_0, \mathtt{S}\}$-graph of an instance $\mathcal{J}$: its nodes are elements of $\mathsf{adom}(\mathcal{J})$ and there is an edge from a to b if and only if $\mathtt{S}_0(a,b)$ or $\mathtt{S}(a,b)$ belongs to $\mathcal{J}$. We denote this graph by $G^{\mathcal{J}}_{\mathtt{S},\mathtt{S}_0}$. We then consider weakly connected components (WCC) of this graph. Two terms a and b are in the same WCC if there is an undirected path from a to b in $G^{\mathcal{J}}_{\mathtt{S},\mathtt{S}_0}$.

We now come back to defining h'. Let $V_1, \ldots, V_\ell$ be the WCCs of $G^q_{\mathtt{S},\mathtt{S}_0}$. We apply Lemma 20 to $\mathcal{I}_N$ and obtain the terms as described in its statement. For each $1 \leq l \leq \ell$, let d_l be the minimum d such that $h(v) = a_d$ for some $v \in V_l$ and $a \in \mathsf{adom}(\mathcal{I})$. We define the mapping $h' : \mathsf{Vars}(q) \to \mathsf{adom}(\mathcal{I}_M)$ by 'shifting' each WCC forward along the chain it lies on, unless it is already close to $\mathcal{I}$:

$$h'(v) := \begin{cases} h(v) & \text{if } d_l \leq |q| \text{ where } v \in V_l \\ a_{i-d_l} & \text{otherwise, if } h(v) = a_i \text{ and } v \in V_l \end{cases}$$

We first verify that h' indeed maps elements within $\mathsf{adom}(\mathcal{I}_M)$. Consider a variable v of q and let V_l be its WCC in $G^q_{\mathtt{S},\mathtt{S}_0}$. Let v_{min} be the variable of V_l that realizes d_l, for which $h(v_{min}) = a_{d_l}$ for some $a \in \mathsf{adom}(\mathcal{I})$. By definition of V_l, there exists an undirected (simple) path in $G^q_{\mathtt{S},\mathtt{S}_0}$ from v to

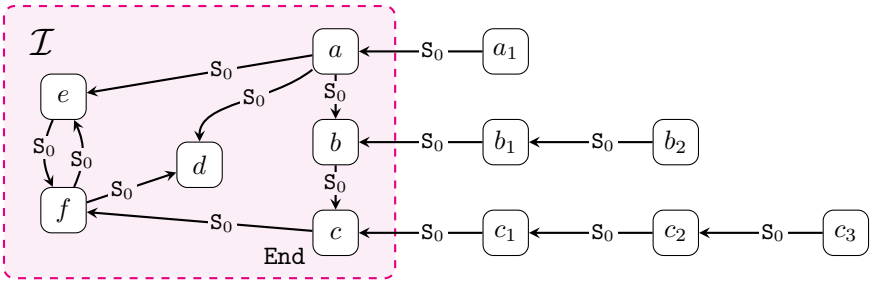

Figure 3: Representation of predicates $\mathtt{S}_0$ and $\mathtt{End}$ in $\mathcal{I}_\infty$ under the hypothesis that $\mathcal{G}$ is bounded by 2. The magenta boxed part of the figure is the instance $\mathcal{I}$. The $\mathtt{S}_0$-atoms, as described in Lemma 20, form several chains leading to elements of $\mathcal{I}$. Each of the $\mathtt{S}_0$-chains only grows until the distance to the $\mathtt{End}$-predicate is bigger than the bound on $\mathcal{G}$ (here 2).

v_{min}, whose length is at most $|q|$. From h being a homomorphism, the atoms that form this path are mapped via h on atoms described by Lemma 20. It is thus clear that $h(v)$ is some a_i for $a \in \mathsf{adom}(\mathcal{I})$ and $0 \leq i \leq d_l + |q|$. Now, if $d_l \leq |q|$, then $h'(v) = h(v) = a_i$ and we have $i \leq 2\,|q|$. Otherwise $h'(v) = a_{i-d_l}$ and $i - d_l \leq |q| \leq 2\,|q|$. In both cases, Lemma 22 warrants $h'(v) \in \mathsf{adom}(\mathcal{I}_M)$.

We now claim that h' is a homomorphism from q to $\mathcal{I}_M$. Let us denote the restriction of $\mathcal{I}_N$ to $\mathsf{adom}(\mathcal{I}_n)$ for some $n \leq N$ by $\mathcal{I}_N|_n$. We actually prove that h' is a homomorphism from q to $\mathcal{I}_N|_{M-1}$. This is sufficient as all atoms from $\mathcal{I}_N$ that only involve terms from $\mathsf{adom}(\mathcal{I}_{M-1})$ are already in $\mathcal{I}_M$. This is stated in the following lemma, whose proof is by induction and heavily relies on the previous lemmas:

Lemma 23. *For every $n \geq M$, $\mathcal{I}_n|_{M-1} = \mathcal{I}_M|_{M-1}$. In particular, for $n = N$ we have $\mathcal{I}_N|_{M-1} = \mathcal{I}_M|_{M-1}$.*

We finally prove that h' is a homomorphism from q to $\mathcal{I}_N|_{M-1}$. Since $\mathtt{End}$-atoms cannot be introduced during the chase, any $\mathtt{End}$-atom $\mathtt{End}(t)$ is mapped through h to an atom of $\mathcal{I}$, and thus $h'(t) = h(t)$, so $\mathtt{End}(h'(t)) \in \mathcal{I}_N|_{M-1}$. The case of $\mathtt{S}_0$- and $\mathtt{S}$-atoms is also easy: consider an atom $\mathtt{P}(t, t') \in q$ for $\mathtt{P} \in \{\mathtt{S}_0, \mathtt{S}\}$. By definition, t and t' belong to a same WCC V_l. If h and h' coincide on V_l (that is the case if $d_l \leq |q|$), then $\mathtt{P}(h'(t), h'(t'))$ immediately belongs to $\mathcal{I}_M$ via Lemma 23. Otherwise, $h(t) = a_i$ and $h(t') = a_j$ for some $a \in \mathsf{adom}(\mathcal{I})$ and i, j such that $i = j + 1$ or $i = j - 1$ by Lemma 20. Thus, $h'(t) = a_{i-d_l}$ and $h'(t') = a_{j-d_l}$, and $i - d_l$ and $j - d_l$ satisfy the same relation and are smaller than M, so $\mathtt{P}(h'(t), h'(t'))$ belongs to $\mathcal{I}_M$.

It remains to treat the case of atoms based upon other predicates. Consider an atom $\mathtt{P}(t_1, \ldots, t_m) \in q$ for some $\mathtt{P} \notin \{\mathtt{S}_0, \mathtt{S}\}$ of arity m. Since h is a homomorphism, we have $\mathtt{P}(h(t_1), \ldots, h(t_m)) \in \mathcal{I}_N$. If all $h(t_i) \in \mathsf{adom}(\mathcal{I})$, then all $h'(t_i) = h(t_i)$ and we are done. Otherwise, we can apply Lemma 21. In the case where $h(t_i)$ are all flooded, then so are the $h'(t_i)$ and Rule $R^{\mathsf{gen}}_{\mathsf{flood}}$ concludes. Otherwise, all $h(t_i)$ belong to the same WCC V_l. Again, if $d_l \leq |q|$, then h' and h coincide on V_l and we are done. Otherwise, all t_i's map to some a_{k_i} for $a \in \mathsf{adom}(\mathcal{I})$ and $k_i < m_a$, since $a_{k_i + d_l} \in \mathsf{adom}(\mathcal{I}_N)$. Thus, by Rule $R^{\mathsf{prop}}_{\mathsf{flood}}$, $h'(t_i)$ is flooded, and thus again Rule $R^{\mathsf{gen}}_{\mathsf{flood}}$ concludes.

6 Conclusion

We have shown that decidable BCQ entailment does not imply the decidability of *bts* membership or chase termination for concrete classes of existential rules. This explains the necessity of diverse, class-specific techniques for proving termination, as BCQ entailment offers no systematic tool for this purpose. Future work includes proving that entailment of recursive C2RPQs is also decidable for the class presented here. Conversely, it remains an open problem to identify a homomorphism-closed query language whose decidable entailment is sufficient to guarantee the decidability of chase termination.

Acknowledgments

This work has been partially supported by ANR grant EXPAND (ANR-25-CE23-1215). The authors would like to thank David Carral who hinted this topic and provided useful insights. He still has not played *Celeste*, though.

References

[Afrati and Kolaitis, 2008] Foto Afrati and Phokion G. Kolaitis. Answering aggregate queries in data exchange. In *Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'08*, page 129–138. Association for Computing Machinery, 2008.

[Baget *et al.*, 2009] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Êric Salvat. Extending decidable cases for rules with existential variables. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09*, page 677–682. Morgan Kaufmann Publishers Inc., 2009.

[Beeri and Vardi, 1981] Catriel Beeri and Moshe Y. Vardi. The implication problem for data dependencies. In *Proceedings of the 8th Colloquium on Automata, Languages and Programming, ICALP'81*, page 73–85. Springer-Verlag, 1981.

[Calautti and Pieris, 2019] Marco Calautti and Andreas Pieris. Oblivious chase termination: The sticky case. In *Proceedings of the 22nd International Conference on Database Theory, ICDT'19*, pages 17:1–17:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

[Calautti *et al.*, 2015] Marco Calautti, Georg Gottlob, and Andreas Pieris. Chase termination for guarded existential rules. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS'15*, page 91–103. Association for Computing Machinery, 2015.

[Calì *et al.*, 2013] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013.

[Calì *et al.*, 2012] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *Journal of Web Semantics*, 14:57–83, 2012.

[Carral *et al.*, 2022] David Carral, Lucas Larroque, Marie-Laure Mugnier, and Michaël Thomazo. Normalisations of existential rules: Not so innocuous! In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR'22*, pages 102–111, 2022.

[Courcelle, 1990] Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

[Delivorias *et al.*, 2021] Stathis Delivorias, Michel Leclère, Marie-Laure Mugnier, and Federico Ulliana. Characterizing boundedness in chase variants. *Theory Pract. Log. Program.*, 21(1):51–79, 2021.

[Deutsch *et al.*, 2008] Alin Deutsch, Alan Nash, and Jeff Remmel. The chase revisited. In *Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'08*, page 149–158. Association for Computing Machinery, 2008.

[Fagin *et al.*, 2005a] 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. Database Theory.

[Fagin *et al.*, 2005b] Ronald Fagin, Phokion G. Kolaitis, and Lucian Popa. Data exchange: getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, 2005.

[Fagin, 2018] Ronald Fagin. Tuple-generating dependencies. In *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[Feier *et al.*, 2023] Cristina Feier, Carsten Lutz, and Marcin Przybylko. Answer counting under guarded tgds. *Log. Methods Comput. Sci.*, 19(3), 2023.

[Gaifman *et al.*, 1993] Haim Gaifman, Harry G. Mairson, Yehoshua Sagiv, and Moshe Y. Vardi. Undecidable optimization problems for database logic programs. *J. ACM*, 40(3):683–713, 1993.

[Gerlach *et al.*, 2025] Lukas Gerlach, Lucas Larroque, Jerzy Marcinkowski, and Piotr Ostropolski-Nalewaja. About the multi-head linear restricted chase termination. In *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning, KR'25*, pages 346–355, 2025.

[Gogacz and Marcinkowski, 2014] Tomasz Gogacz and Jerzy Marcinkowski. All-instances termination of chase is undecidable. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming, ICALP'14*, pages 293–304. Springer, 2014.

[Gogacz *et al.*, 2020] Tomasz Gogacz, Jerzy Marcinkowski, and Andreas Pieris. All-instances restricted chase termination. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS'20*, page 245–258. Association for Computing Machinery, 2020.

[Hernich, 2012] André Hernich. Computing universal models under guarded tgds. In *Proceedings of the 15th International Conference on Database Theory, ICDT'12*, page 222–235. Association for Computing Machinery, 2012.

[Larroque and Manière, 2026] Lucas Larroque and Quentin Manière. Will my favorite chases terminate if evaluating conjunctive queries does? One does not simply decide this, 2026. arXiv:2605.12349 [cs.DB].

[Maier *et al.*, 1979] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. Testing implications of data dependencies. *ACM Trans. Database Syst.*, 4(4):455–469, 1979.

[Marnette, 2009] Bruno Marnette. Generalized schema-mappings: from termination to tractability. In *Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'09*, page 13–22. Association for Computing Machinery, 2009.

[Minsky, 1967] Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.

[Ronca *et al.*, 2018] Alessandro Ronca, Mark Kaminski, Bernardo Cuenca Grau, Boris Motik, and Ian Horrocks. Stream reasoning in temporal datalog. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI'18*, pages 1941–1948. AAAI Press, 2018.