Oblivious Chase Termination

Citation for published version:
Calautti, M & Pieris, A 2019, Oblivious Chase Termination: The Sticky Case. in P Barcelo & M Calautti (eds), 22nd International Conference on Database Theory (ICDT 2019)., 17, LIPICS, vol. 127, 22nd International Conference on Database Theory, Lisbon, Portugal, 26/03/19. https://doi.org/10.4230/LIPIcs.ICDT.2019.17

Digital Object Identifier (DOI):
10.4230/LIPIcs.ICDT.2019.17

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Early version, also known as pre-print

Published In:
22nd International Conference on Database Theory (ICDT 2019)

Oblivious Chase Termination: The Sticky Case

Marco Calautti
School of Informatics, University of Edinburgh
mcalautt@inf.ed.ac.uk

Andreas Pieris
School of Informatics, University of Edinburgh
apieris@inf.ed.ac.uk

Abstract

The chase procedure is one of the most fundamental algorithmic tools in database theory. A key algorithmic task is uniform chase termination, i.e., given a set of tuple-generating dependencies (tgds), is it the case that the chase under this set of tgds terminates, for every input database? In view of the fact that this problem is undecidable, no matter which version of the chase we consider, it is natural to ask whether well-behaved classes of tgds, introduced in different contexts such as ontological reasoning, make our problem decidable. In this work, we consider a prominent decidability paradigm for tgds, called stickiness. We show that for sticky sets of tgds, uniform chase termination is decidable if we focus on the (semi-)oblivious chase, and we pinpoint its exact complexity: PSPACE-complete in general, and NLOGSPACE-complete for predicates of bounded arity. These complexity results are obtained via graph-based syntactic characterizations of chase termination that are of independent interest.

2012 ACM Subject Classification Theory of Computation → Database query languages (principles), database constraints theory, logic and databases

Keywords and phrases Chase procedure, tuple-generating dependencies, stickiness, termination, computational complexity

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

The *chase procedure* (or simply chase) is a fundamental algorithmic tool that has been successfully applied to several database problems such as containment of queries under constraints [1], checking logical implication of constraints [3, 17], computing data exchange solutions [10], and query answering under constraints [5], to name a few. The chase procedure accepts as an input a database D and a set Σ of constraints and, if it terminates, its result is a finite instance D_Σ that is a *universal* model of D and Σ, i.e., is a model that can be homomorphically embedded into every other model of D and Σ. In other words, D_Σ acts as a representative of all the other models of D and Σ. This is the reason for the ubiquity of the chase in database theory, as discussed in [8]. Indeed, many key database problems can be solved by simply exhibiting a universal model.

A prominent class of constraints that can be naturally treated by the chase procedure is the class of *tuple-generating dependencies (tgds)*, i.e., sentences of the form $\forall\bar{x}\forall\bar{y}\,(\phi(\bar{x},\bar{y}) \rightarrow \exists\bar{z}\,\psi(\bar{x},\bar{z}))$, where ϕ and ψ are conjunctions of atoms. Given a database D and a set Σ of tgds, the chase adds new atoms to D (possibly involving null values that act as witnesses for the existentially quantified variables) until the final result satisfies Σ. For example, given the database $D = \{R(c)\}$, and the tgd $\forall x(R(x) \rightarrow \exists y\, P(x,y) \wedge R(y))$, the database atom *triggers* the tgd, and the chase will add in D the atoms $P(c,\perp_1)$ and $R(\perp_1)$ in order to satisfy it, where $\perp_1$ is a (labeled) null representing some unknown value. However, the new atom $R(\perp_1)$ triggers again the tgd, and the chase is forced to add the atoms $P(\perp_1,\perp_2), R(\perp_2)$, where $\perp_2$ is a new null. The result of the chase is the instance $\{R(c), P(c,\perp_1)\} \cup \bigcup_{i>0}\{R(\perp_i), P(\perp_i,\perp_{i+1})\}$, where $\perp_1,\perp_2,\ldots$ are nulls.

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The above example shows that the chase procedure may not terminate, even for very simple databases and sets of tgds. This fact motivated a long line of research on identifying subclasses of tgds that ensure the termination of the chase procedure, no matter how the input database looks like. A prime example is the class of *weakly-acyclic* tgds [10], which is the standard language for data exchange purposes, and guarantees the termination of the semi-oblivious and restricted (a.k.a. standard) chase. Inspired by weak-acyclicity, the notion of *rich-acyclicity* has been proposed in [16], which guarantees the termination of the oblivious chase. Many other sufficient conditions for chase termination can be found in the literature; see, e.g., [8, 9, 13, 15, 18, 19] — this list is by no means exhaustive, and we refer the reader to [14] for a comprehensive survey.

With so much effort spent on identifying sufficient conditions for the termination of the chase procedure, the question that immediately comes up is whether a sufficient condition that is also *necessary* exists. In other words, given a set Σ of tgds, is it possible to decide whether, for every database D, the chase on D and Σ terminates? This question has been addressed in [11], and has been shown that the answer is negative, no matter which version of the chase we consider, namely the oblivious, semi-oblivious and restricted chase. The problem remains undecidable even if the database is known; this has been established in [8] for the restricted chase, and it was observed in [18] that the same proof shows undecidability also for the (semi-)oblivious chase.

The undecidability proof given in [11] constructs a sophisticated set of tgds that goes beyond existing well-behaved classes of tgds that enjoy certain syntactic properties, which in turn ensure useful model-theoretic properties. This has been already observed in [4], where it is shown that the chase termination problem is decidable if we focus on the (semi-)oblivious version of the chase, and classes of tgds based on the notion of *guardedness*. Guardedness is one of the main decidability paradigms that gives rise to robust tgd-based languages [2, 5, 6] that capture important database constraints and lightweight description logics. The key model-theoretic property of guarded-based languages, which explains their robust behaviour, is the tree-likeness of the underlying universal models [5]. On the other hand, there are interesting statements that are inherently non-tree-like, and thus not expressible via guarded-based languages. Such a statement consists of the tgds $\forall x \forall y (R(x,y) \rightarrow \exists z\, R(y,z) \wedge P(z))$ and $\forall x \forall y (P(x) \wedge P(y) \rightarrow S(x,y))$, which compute the cartesian product of a unary relation that stores infinitely many elements.

The inability of guarded-based tgds to express non-tree-like statements like the one above, has motivated a long line of research on isolating well-behaved classes of tgds that go beyond tree-like models and guardedness. The main decidability paradigm obtained from this effort is known as *stickiness* [7]. The key idea underlying stickiness can be described as follows: variables that appear more than once in the left-hand side of a tgd, known as the body of the tgd, should be inductively propagated (or "stick") to every atom in the left-hand side of the tgd; more details are given in Section 2. It is easy to verify that the above non-tree-like statement is trivially sticky since none of the body variables occurs more than once. The crucial question that comes up is the following: *given a sticky set Σ of tgds, is it possible to decide whether the chase terminates for every input database?*

The main goal of this work is to study the chase termination problem for sticky sets of tgds, and give a definite answer to the above fundamental question. In fact, we focus on the (semi-)oblivious versions of the chase, and we show that deciding termination for sticky sets of tgds is decidable, and provide precise complexity results: PSPACE-complete in general, and NLOGSPACE-complete for predicates of bounded arity. Although the (semi-)oblivious versions of the chase are considered as non-standard ones, they have certain advantages that classify them as important algorithmic tools, and thus they deserve our attention. In particular, unlike the restricted chase, the application of a tgd does not require checking if the head of the tgd is already satisfied by the instance, and this guarantees technical clarity and efficiency; for a more thorough discussion on the advantages of the oblivious and semi-oblivious chase see [5, 18].

Summary of Contributions. Our plan of attack and results can be summarized as follows:

- In Section 4, we provide a semantic characterization of non-termination of the (semi-)oblivious chase under sticky sets of tgds via the existence of path-like infinite chase derivations, which forms the basis for our decision procedure.
- By exploiting the above semantic characterization, we then provide, in Section 5, a syntactic characterization of chase termination via graph-based conditions. To this end, we extend recent syntactic characterizations from [4] of the termination of the (semi-)oblivious chase under *constant-free* linear tgds (tgds with one body atom), to linear tgds with constants. The transition from constant-free tgds to tgds with constants turned out to be more challenging than expected.
- Finally, in Section 6, by exploiting the graph-based syntactic characterization from the previous section, we establish the precise complexity of our problem: PSPACE-complete in general, and NLOGSPACE-complete for predicates of bounded arity.

Full proofs are provided in a clearly marked appendix.

2 Preliminaries

We consider the disjoint countably infinite sets $\mathbf{C}$, $\mathbf{N}$, and $\mathbf{V}$ of *constants*, *(labeled) nulls*, and (regular) *variables* (used in dependencies), respectively. A fixed lexicographic order is assumed on $(\mathbf{C} \cup \mathbf{N})$ such that every null of $\mathbf{N}$ follows all constants of $\mathbf{C}$. We refer to constants, nulls and variables as *terms*. Let $[n] = \{1, \ldots, n\}$, for any integer $n \geq 1$.

Relational Databases. A *schema* $\mathbf{S}$ is a finite set of relation symbols (or predicates) with associated arity. We write R/n to denote that R has arity $n > 0$. A position $R[i]$ in $\mathbf{S}$, where $R/n \in \mathbf{S}$ and $i \in [n]$, identifies the i-th argument of R. An *atom* over $\mathbf{S}$ is an expression of the form $R(\bar{t})$, where $R/n \in \mathbf{S}$ and $\bar{t}$ is an n-tuple of terms. We write $\mathsf{var}(\alpha)$ for the set of variables occurring in an atom α; this notation naturally extends to sets of atoms. A *fact* is an atom whose arguments consist only of constants. An *instance* over $\mathbf{S}$ is a (possibly infinite) set of atoms over $\mathbf{S}$ that contain constants and nulls, while a *database* over $\mathbf{S}$ is a finite set of facts over $\mathbf{S}$. The *active domain* of an instance I, denoted $\mathsf{dom}(I)$, is the set of all terms, i.e., constants and nulls, occurring in I.

Substitutions and Homomorphisms. A *substitution* from a set of terms T to a set of terms T' is a function $h : T \to T'$ defined as follows: $\emptyset$ is a substitution (empty substitution), and if h is a substitution, then $h \cup \{t \mapsto t'\}$, where $t \in T$ and $t' \in T'$, is a substitution. The restriction of h to a subset S of T, denoted $h_{|S}$, is the substitution $\{t \mapsto h(t) \mid t \in S\}$. A *homomorphism* from a set of atoms A to a set of atoms B is a substitution h from the set of terms in A to the set of terms in B such that (i) $t \in \mathbf{C}$ implies $h(t) = t$, i.e., h is the identity on $\mathbf{C}$, and (ii) $R(t_1, \ldots, t_n) \in A$ implies $h(R(t_1, \ldots, t_n)) = R(h(t_1), \ldots, h(t_n)) \in B$.

Tuple-Generating Dependencies. A *tuple-generating dependency* σ is a first-order sentence

$$\forall \bar{x} \forall \bar{y} \, (\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \, \psi(\bar{x}, \bar{z})) \,,$$

where $\bar{x}, \bar{y}, \bar{z}$ are tuples of variables of $\mathbf{V}$, while $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ are conjunctions of atoms (possibly with constants). For brevity, we write σ as $\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \, \psi(\bar{x}, \bar{z})$, and use comma instead of $\wedge$ for joining atoms. We refer to $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ as the *body* and *head* of σ, denoted $\mathsf{body}(\sigma)$ and $\mathsf{head}(\sigma)$, respectively. The *frontier* of the tgd σ, denoted $\mathsf{fr}(\sigma)$, is the set of variables $\bar{x}$, i.e., the variables that appear both in the body and the head of σ. The *schema* of a set Σ of tgds, denoted $\mathsf{sch}(\Sigma)$, is the set of predicates in Σ. We also write $\mathsf{const}(\Sigma)$ for the set of constants occurring in Σ. An instance I satisfies a tgd σ as the one above, written $I \models \sigma$, if the following holds: whenever there exists a homomorphism h such that $h(\phi(\bar{x}, \bar{y})) \subseteq I$, then there exists $h' \supseteq h_{|\bar{x}}$ such that

$h'(\psi(\bar{x}, \bar{z})) \subseteq I$. Note that, by abuse of notation, we sometimes treat a conjunction of atoms as a set of atoms. The instance I satisfies a set Σ of tgds, written $I \models \Sigma$, if $I \models \sigma$ for each $\sigma \in \Sigma$.

One of the main syntactic paradigms for tgds is stickiness [7]. The key property underlying this condition is as follows: variables that appear more than once in the body of a tgd should be inductively propagated (or "stick") to every head atom. This is graphically illustrated as follows

$$R(x,y),\ P(y,z) \rightarrow \exists w\ T(x,y,w) \qquad\qquad R(x,y),\ P(y,z) \rightarrow \exists w\ T(x,y,w)$$

$$T(x,y,z)\ \rightarrow \exists w\ S(y,w) \qquad\qquad T(x,y,z)\ \rightarrow \exists w\ S(x,w)$$

where the first set of tgds is sticky, while the second is not. The formal definition is based on an inductive procedure that marks the variables that may violate the property described above. Roughly, during the base step of this procedure, a variable that appears in the body of a tgd but not in every head atom is marked. Then, the marking is inductively propagated from head to body. Stickiness requires every marked variable to appear only once in the body of a tgd. The formal definition follows. Let Σ be a set of tgds; w.l.o.g., we assume that the tgds in Σ do not share variables. Given an atom $R(\bar{t})$ and a variable x in $\bar{t}$, $\mathsf{pos}(R(\bar{t}), x)$ is the set of positions in $R(\bar{t})$ at which x occurs. Let $\sigma \in \Sigma$ and x a variable in the body of σ. We inductively define when x *is marked in* Σ:

- If x does not occur in every atom of $\mathsf{head}(\sigma)$, then x is marked in Σ.
- Assuming that $\mathsf{head}(\sigma)$ contains an atom of the form $R(\bar{t})$ and $x \in \bar{t}$, if there exists $\sigma' \in \Sigma$ that has in its body an atom of the form $R(\bar{t}')$, and each variable in $R(\bar{t}')$ at a position of $\mathsf{pos}(R(\bar{t}), x)$ is marked in Σ, then x is marked in Σ.

The set Σ is *sticky* if there is no tgd that contains two occurrences of a variable that is marked in Σ. We denote by $\mathbb{S}$ the class of sticky finite sets of tgds. Let us clarify that we work with finite sets of tgds only. Thus, in the rest of the paper, a set of tgds is always finite.

The Tgd Chase Procedure. The tgd chase procedure (or simply chase) takes as an input a database D and a set Σ of tgds, and constructs a (possibly infinite) instance I such that $I \supseteq D$ and $I \models \Sigma$. A crucial notion is that of trigger for a set of tgds on some instance. Consider a set Σ of tgds and an instance I. A *trigger* for Σ on I is a pair (σ, h), where $\sigma \in \Sigma$ and h is a homomorphism such that $h(\mathsf{body}(\sigma)) \subseteq I$. An application of (σ, h) to I returns the instance $J = I \cup h'(\mathsf{head}(\sigma))$, where $h' \supseteq h_{|\mathsf{fr}(\sigma)}$ is such that (i) for each existentially quantified variable z of σ, $h'(z) \in \mathbf{N}$ does not occur in I and follows lexicographically all nulls in I, and (ii) for each pair (z, w) of distinct existentially quantified variables of σ, $h'(z) \neq h'(w)$. Such a trigger application is denoted as $I\langle\sigma, h\rangle J$.

The main idea of the chase is, starting from a database D, to exhaustively apply triggers for the given set Σ of tgds on the instance constructed so far. However, the choice of the type of the next trigger to be applied is crucial since it gives rise to different variations of the chase procedure. In this work, we focus on the *oblivious* [5] and the *semi-oblivious* [12, 18] chase.

Oblivious. A finite sequence $I_0, I_1, \ldots, I_n$ of instances, where $n \geq 0$, is said to be a *terminating oblivious chase sequence* of I_0 w.r.t. a set Σ of tgds if: (i) for each $0 \leq i < n$, there exists a trigger (σ, h) for Σ on I_i such that $I_i\langle\sigma, h\rangle I_{i+1}$; (ii) for each $0 \leq i < j < n$, assuming that $I_i\langle\sigma_i, h_i\rangle I_{i+1}$ and $I_j\langle\sigma_j, h_j\rangle I_{j+1}$, $\sigma_i = \sigma_j$ implies $h_i \neq h_j$, i.e., h_i and h_j are different homomorphisms; and (iii) there is no trigger (σ, h) for Σ on I_n such that $(\sigma, h) \notin \{(\sigma_i, h_i)\}_{0 \leq i < n}$. In this case, the result of the chase is the (finite) instance I_n. An infinite sequence $I_0, I_1, \ldots$ of instances is said to be a *non-terminating oblivious chase sequence* of I_0 w.r.t. Σ if: (i) for each $i \geq 0$, there exists a trigger (σ, h) for Σ on I_i such that $I_i\langle\sigma, h\rangle I_{i+1}$; (ii) for each $i, j > 0$ such that $i \neq j$, assuming that $I_i\langle\sigma_i, h_i\rangle I_{i+1}$ and $I_j\langle\sigma_j, h_j\rangle I_{j+1}$, $\sigma_i = \sigma_j$ implies $h_i \neq h_j$; and (iii) for each $i \geq 0$, and for every trigger (σ, h)

for Σ on I_i, there exists $j \geq i$ such that $I_j\langle\sigma, h\rangle I_{j+1}$; this is known as the *fairness condition*, and guarantees that all the triggers eventually will be applied. The result of the chase is $\bigcup_{i \geq 0} I_i$.

Semi-oblivious. This is a refined version of the oblivious chase, which avoids the application of some superfluous triggers. Roughly speaking, given a tgd σ, for the semi-oblivious chase, two homomorphisms h and g that agree on the frontier of σ, i.e., $h_{|\mathsf{fr}(\sigma)} = g_{|\mathsf{fr}(\sigma)}$, are indistinguishable. To formalize this, we first define the binary relation $\sim_\sigma$ on the set of all possible substitutions from the terms in $\mathsf{body}(\sigma)$ to $(\mathbf{C} \cup \mathbf{N})$, denoted S_σ, as follows: $h \sim_\sigma g$ iff $h_{|\mathsf{fr}(\sigma)} = g_{|\mathsf{fr}(\sigma)}$. It is easy to verify that $\sim_\sigma$ is an equivalence relation on the elements of S_σ. A (terminating or non-terminating) oblivious chase sequence $I_0, I_1, \ldots$ is called *semi-oblivious* if the following holds: for every $i, j \geq 0$ such that $i \neq j$, assuming that $I_i\langle\sigma_i, h_i\rangle I_{i+1}$ and $I_j\langle\sigma_j, h_j\rangle I_{j+1}$, $\sigma_i = \sigma_j = \sigma$ implies $h_i \not\sim_\sigma h_j$, i.e., h_i and h_j belong to different equivalence classes.

Henceforth, we write o-chase and so-chase for oblivious and semi-oblivious chase, respectively. A useful notion that we are going to use in our proofs is the so-called chase relation [7], which essentially describes how the atoms generated during the chase depend on each other. Fix a non-terminating $\star$-chase sequence $s = (I_i)_{i \geq 0}$, where $\star \in \{\mathsf{o}, \mathsf{so}\}$, of a database D w.r.t. a set Σ of tgds, and assume that for each $i \geq 0$, $I_i\langle\sigma_i, h_i\rangle I_{i+1}$, i.e., I_{i+1} is obtained from I_i via the application of the trigger (σ_i, h_i) to I_i. The *chase relation* of s, denoted $\prec_s$, is a binary relation over $\bigcup_{i \geq 0} I_i$ such that $\alpha \prec_s \beta$ iff there exists $i \geq 0$ such that $\alpha \in h_i(\mathsf{body}(\sigma_i))$ and $\beta \in I_{i+1} \setminus I_i$.

3 Chase Termination Problem

It is well-known that due to the existentially quantified variables, a $\star$-chase sequence, where $\star \in \{\mathsf{o}, \mathsf{so}\}$, may be infinite. This is true even for very simple settings: it is easy to verify that the only $\star$-chase sequence of $D = \{R(a, b)\}$ w.r.t. the set Σ consisting of the single tgd $R(x, y) \rightarrow \exists z\, R(y, z)$ is non-terminating. The question that comes up is, given a set Σ of tgds, whether we can check that, for every database D, all or some (semi-)oblivious chase sequences of D w.r.t. Σ are terminating. Before formalizing the above problem, let us recall the following central classes of tgds:

$$\begin{aligned}
\mathbb{CT}^{\star}_{\forall\forall} &= \{\Sigma \mid \text{for } \textit{every} \text{ database } D,\ \textit{every}\ \star\text{-chase sequence of } D \text{ w.r.t. } \Sigma \text{ is terminating}\} \\
\mathbb{CT}^{\star}_{\forall\exists} &= \{\Sigma \mid \text{for } \textit{every} \text{ database } D,\ \text{there } \textit{exists} \text{ a terminating } \star\text{-chase sequence of } D \text{ w.r.t. } \Sigma\}.
\end{aligned}$$

The main problems tackled in this work are defined as follows, where $\mathbb{C}$ is a class of tgds:

PROBLEM :	$\mathsf{CT}^{\star}_{\forall\forall}(\mathbb{C})$
INPUT :	A set $\Sigma \in \mathbb{C}$ of tgds.
QUESTION :	Is $\Sigma \in \mathbb{CT}^{\star}_{\forall\forall}$?

PROBLEM :	$\mathsf{CT}^{\star}_{\forall\exists}(\mathbb{C})$
INPUT :	A set $\Sigma \in \mathbb{C}$ of tgds.
QUESTION :	Is $\Sigma \in \mathbb{CT}^{\star}_{\forall\exists}$?

It is well-known that $\mathbb{CT}^{\mathsf{o}}_{\forall\forall} = \mathbb{CT}^{\mathsf{o}}_{\forall\exists} \subset \mathbb{CT}^{\mathsf{so}}_{\forall\forall} = \mathbb{CT}^{\mathsf{so}}_{\forall\exists}$ [12]. This immediately implies that, after fixing the version of the chase in consideration, i.e., oblivious or semi-oblivious, the above decision problems are equivalent. Henceforth, for a class $\mathbb{C}$ of tgds, we simply refer to the problem $\mathsf{CT}^{\star}_{\forall}(\mathbb{C})$, and we write $\mathbb{CT}^{\star}_{\forall}$ for the classes $\mathbb{CT}^{\star}_{\forall\forall}$ and $\mathbb{CT}^{\star}_{\forall\exists}$, where $\star \in \{\mathsf{o}, \mathsf{so}\}$.

We know that our main problem is undecidable if we consider arbitrary tgds. In fact, assuming that $\mathbb{TGD}$ denotes the class of arbitrary tgds, we have that:

▶ **Theorem 1.** *For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\mathsf{CT}^{\star}_{\forall}(\mathbb{TGD})$ is undecidable, even for binary and ternary predicates.*

The above result has been shown in [11]. However, the employed set of tgds for showing this undecidability result is far from being sticky. This led us to ask whether $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ is decidable. This is a non-trivial problem, and pinpointing its complexity is the main goal of this work.

Some Useful Results. Before proceeding with the complexity analysis, let us recall a couple of technical results that would allows us to significantly simplify our later analysis.

It would be useful if a special database exists that gives rise to a non-terminating chase sequence in case there exists one. Interestingly, such a database exists, which is known as the critical database for a set of tgds [18]. Formally, given a set Σ of tgds, the *critical database* for Σ is defined as

$$\mathsf{cr}(\Sigma) \;=\; \begin{cases} \{R(c,\ldots,c) \mid R \in \mathsf{sch}(\Sigma)\}, \text{where } c \in \mathbf{C} \text{ is a fixed constant} & \text{if } \mathsf{const}(\Sigma) = \emptyset, \\ \\ \{R(c_1,\ldots,c_n) \mid R \in \mathsf{sch}(\Sigma) \text{ and } (c_1,\ldots,c_n) \in \mathsf{const}(\Sigma)^n\} & \text{if } \mathsf{const}(\Sigma) \neq \emptyset. \end{cases}$$

In other words, $\mathsf{cr}(\Sigma)$ consists of all the atoms that can be formed using the predicates and the constants in Σ; if Σ is constant-free, then we consider an arbitrary constant of $\mathbf{C}$. The following result from [18] shows that $\mathsf{cr}(\Sigma)$ is indeed the desired database:

▶ **Proposition 2.** *Consider a set Σ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\Sigma \notin \mathbb{CT}_{\forall}^{\star}$ iff there exists a non-terminating $\star$-chase sequence of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.*

Even though we can focus on the critical database and check whether it gives rise to a non-terminating chase sequence s, the main difficulty is to ensure that s enjoys the fairness condition. Interestingly, as it has been recently shown in [4], we can neglect the fairness condition, which significantly simplifies the required analysis. To formalize this result, we need to recall the notion of the infinite chase derivation, which is basically a non-terminating chase sequence without the fairness condition. Fix $\star \in \{\mathsf{o}, \mathsf{so}\}$. We define $\diamond_{\sigma}^{\star}$ as $\neq$, if $\star = \mathsf{o}$, and $\not\simeq_{\sigma}$, if $\star = \mathsf{so}$. An *infinite $\star$-chase derivation* of a database D w.r.t. a set Σ of tgds is an infinite sequence $(I_i)_{i \geq 0}$ of instances, where $I_0 = D$, such that: (i) for each $i \geq 0$, there exists a trigger (σ_i, h_i) for Σ in I_i with $I_i\langle\sigma_i, h_i\rangle I_{i+1}$, and (ii) for each $i \neq j$, $\sigma_i = \sigma_j = \sigma$ implies $h_i \diamond_{\sigma}^{\star} h_j$. The following holds:

▶ **Proposition 3.** *Consider a database D and a set Σ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, there is a non-terminating $\star$-chase sequence of D w.r.t. Σ iff there is an infinite $\star$-chase derivation of D w.r.t. Σ.*

By combining Propositions 2 and 3, we immediately get the following useful result:

▶ **Corollary 4.** *Consider a set Σ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\Sigma \notin \mathbb{CT}_{\forall}^{\star}$ iff there exists an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t Σ.*

4 Semantic Characterization of Chase Non-Termination

We proceed to characterize the non-termination of the (semi-)oblivious chase under sticky sets of tgds. In particular, we show that if a sticky set Σ of tgds does not belong to $\mathbb{CT}_{\forall}^{\star}$, for $\star \in \{\mathsf{o}, \mathsf{so}\}$, then we can always isolate a *linear* infinite $\star$-chase derivation δ_{ℓ} of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. Roughly, linearity means that there is an infinite simple path $\alpha_0, \alpha_1, \alpha_2 \ldots$ in the chase relation of δ_{ℓ} such that $\alpha_0 \in \mathsf{cr}(\Sigma)$ and α_i is constructed during the i-th trigger application, while all the atoms that are needed to construct this path, and are not already on the path, are atoms of $\mathsf{cr}(\Sigma)$. Notice that the chase relation of a $\star$-chase derivation is defined in the same way as the chase relation of a $\star$-chase sequence.

▶ **Definition 5.** Consider a set Σ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, an infinite $\star$-chase derivation $\delta = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, where $I_i\langle\sigma_i, h_i\rangle I_{i+1}$ for $i \geq 0$, is called *linear* if there exists an infinite sequence of distinct atoms $(\alpha_i)_{i \geq 0}$ such that the following hold:

- $\alpha_0 \in \mathsf{cr}(\Sigma)$.
- For each $i \geq 0$, $\alpha_{i+1} \in I_{i+1} \setminus I_i$, and there exists $\beta \in \mathsf{body}(\sigma_i)$ such that $h_i(\beta) = \alpha_i$ and $h_i(\mathsf{body}(\sigma_i) \setminus \{\beta\}) \subseteq \mathsf{cr}(\Sigma)$. ◼

A simple example that illustrates the notion of linear infinite o-chase derivation follows:

▶ **Example 6.** Let Σ be the sticky set consisting of the tgd

$$\sigma = P(x, y, z), R(y, w) \rightarrow \exists v\, P(z, y, v), R(y, v).$$

Consider the infinite o-chase derivation $\delta = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, where

$$\begin{array}{rl}
I_0 \ = \ \{P(c, c, c), R(c, c)\} & \langle \sigma, h_0 = \{x \mapsto c, y \mapsto c, z \mapsto c, w \mapsto c\}\rangle \\
I_1 = I_0 \cup \{P(c, c, \bot_1), R(c, \bot_1)\} & \langle \sigma, h_1 = \{x \mapsto c, y \mapsto c, z \mapsto \bot_1, w \mapsto c\}\rangle \\
I_2 = I_1 \cup \{P(\bot_1, c, \bot_2), R(c, \bot_2)\} & \langle \sigma, h_2 = \{x \mapsto \bot_1, y \mapsto c, z \mapsto \bot_2, w \mapsto c\}\rangle \\
\vdots & \\
I_{i+1} = I_i \cup \{P(\bot_i, c, \bot_{i+1}), R(c, \bot_{i+1})\} & \langle \sigma, h_{i+1} = \{x \mapsto \bot_i, y \mapsto c, z \mapsto \bot_{i+1}, w \mapsto c\}\rangle \\
\vdots &
\end{array}$$

Let $\alpha_0 = P(c, c, c)$, $\alpha_1 = P(c, c, \bot_1)$, and $\alpha_i = P(\bot_{i-1}, c, \bot_i)$ for $i > 1$. It is easy to very that δ is linear due to $(\alpha_i)_{i \geq 0}$. Indeed, $\alpha_0 \in \mathsf{cr}(\Sigma)$, and for every $i \geq 0$, α_i belongs to $I_{i+1} \setminus I_i$, while $h_i(P(x, y, z)) = \alpha_i$ and $h_i(R(y, w)) = R(c, c) \in \mathsf{cr}(\Sigma)$. ◼

We are now ready to present the main characterization of non-termination of the (semi-)oblivious chase under sticky sets of tgds via linear infinite $\star$-chase derivations.

▶ **Theorem 7.** *Consider a set $\Sigma \in \mathbb{S}$ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\Sigma \notin \mathbb{CT}_{\forall}^{\star}$ iff there exists a linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.*

By Corollary 4, it suffices to show the following: the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ implies the existence of a linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. This is a rather involved result, which is established in two main steps:

1. We show that the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ implies the existence of an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ such that the chase relation of δ contains a special path rooted at an atom of $\mathsf{cr}(\Sigma)$, called *continuous*. Intuitively, continuity ensures the continuous propagation of a new null on the path in question.
2. By exploiting the existence of a continuous path, we construct a linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. In fact, due to stickiness, we can convert an infinite suffix P of the continuous path in $\prec_\delta$, together with all the atoms that are needed to generate the atoms on P via a single trigger application, into a linear infinite $\star$-chase derivation δ_ℓ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. As we shall see, stickiness helps us to ensure that δ_ℓ is linear, while continuity allows us to show that δ_ℓ is infinite.

We proceed to give some more details for the above two steps. Although we keep the following discussion informal, we give enough evidence for the validity of Theorem 7.

4.1 Existence of a Continuous Path

Let us first make the notion of the path in the chase relation of a derivation more precise. Given an infinite $\star$-chase derivation $\delta = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, a *finite δ-path* is a finite sequence of atoms $(\alpha_i)_{0 \leq i \leq n}$ such that $\alpha_0 \in I_0$ and $\alpha_i \prec_\delta \alpha_{i+1}$. Analogously, we can define *infinite δ-paths*, which are infinite sequences of atoms rooted at an atom of I_0.

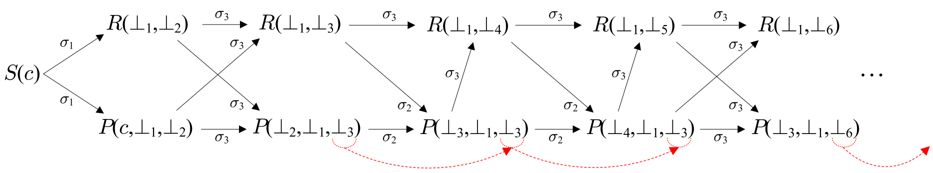
The intention underlying continuity is to ensure the continuous propagation of a new null on a path. Roughly, a δ-path $(\alpha_i)_{0 \leq i \leq n}$ is continuous if, assuming that $\alpha_{\ell_0}, \ldots, \alpha_{\ell_m}$ are the atoms on the path where nulls are invented, with $\ell_0 < \cdots < \ell_m$, then α_{ℓ_m} is the last atom of the path, i.e.,

$\ell_m = n$, and for every α_{ℓ_i}, there exists at least one null invented in α_{ℓ_i} that is *necessarily propagated* until the atom $\alpha_{\ell_{i+1}}$ in case $\star = \mathsf{so}$ (resp., the atom before $\alpha_{\ell_{i+1}}$ in case $\star = \mathsf{o}$). An infinite δ-path $(\alpha_i)_{i \geq 0}$, where $\alpha_{\ell_0}, \alpha_{\ell_1}, \ldots$ are the atoms on the path where a null is invented, with $\ell_0 < \ell_1 < \cdots$, is continuous if every finite δ-path $(\alpha_i)_{0 \leq i \leq \ell_j}$, for $j \geq 0$, is continuous. Here is an example.

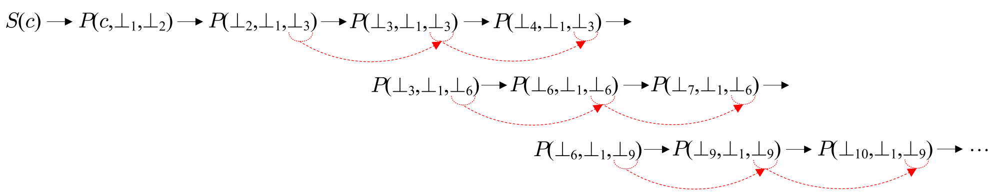
▶ **Example 8.** Consider the sticky set $\Sigma = \{\sigma_1, \sigma_2, \sigma_3\}$, where

$$\begin{array}{rcl} \sigma_1 & = & S(x) \ \rightarrow \ \exists y \exists z\, P(x,y,z), R(y,z) \\ \sigma_2 & = & P(x,y,z), R(y,w) \ \rightarrow \ P(w,y,z) \end{array}$$

and σ_3 is the tgd used in Example 6. It is easy to verify that there exists an infinite o-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ such that the following is part of $\prec_\delta$; a black edge from α to β labeled by σ means that (α, β) belongs to $\prec_\delta$ due to a trigger that involves the tgd σ:



It can be verified that the path with P-atoms in the figure is a continuous infinite δ-path. Let us explain the reason. The first atom in which a null is invented is $P(c, \bot_1, \bot_2)$, with $\bot_1, \bot_2$ being the new nulls, and continuity is satisfied since the next atom invents a null, that is, $\bot_3$. Now, since the null $\bot_3$ is propagated (this is indicated via the red dashed arrows) until the atom before the next null generator $P(\bot_3, \bot_1, \bot_6)$, continuity is satisfied. In the rest of the path the same pattern is repeated, and thus continuity is globally satisfied. In fact, the pattern that we can extract is the following



where the continuous propagation of a new null, shown via the red arrows, can be easily observed. ◀

We can show, via a graph-theoretic argument, that the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ implies the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ that admits a continuous infinite path. Let us briefly explain the key idea underlying this result. If we know that an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ exists, then we can construct an infinite $\star$-chase derivation $\delta' = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ (by essentially rearranging the triggers of δ in order to obtain a derivation of a convenient form) such that the following statement holds:

there exists an infinite directed acyclic rooted graph $G = (N \cup \{\bullet\}, E, \lambda)$ of finite degree, where $\bullet$ is the root, $N \subseteq \bigcup_{i \geq 0} I_i$, every node of N is reachable from $\bullet$, and λ labels the edges of E with finite sequences of atoms from $\bigcup_{i \geq 0} I_i$, such that, for every finite path $\bullet, v_1, \ldots, v_n$, for $n \geq 1$, $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{n-1}, v_n)$ is a continuous finite δ'-path.

By applying König's lemma[1] on G, we get that G contains an infinite simple path $P = \bullet, v_1, v_2, \ldots$.

---

[1] König's lemma is a well-known result from graph theory that states the following: for an infinite directed rooted graph, if every node is reachable from the root, and every node has finite out-degree, then there exists an infinite directed simple path from the root.

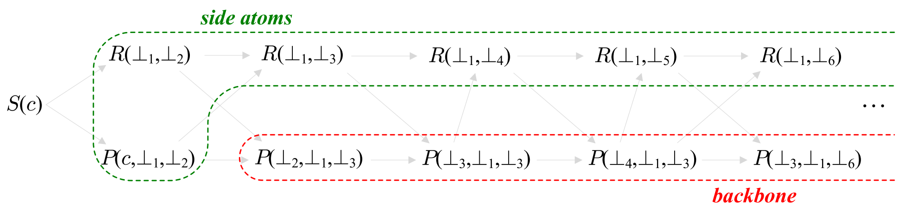
We claim that $P' = \lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots$ is a continuous δ'-path, which establishes the claim. By contradiction, assume that P' is not a continuous δ'-path. This implies that there exists a finite prefix $\bullet, v_1, \ldots, v_n$ of P such that $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{n-1}, v_n)$ is not a continuous δ'-path, which contradicts the fact that the label of every finite path starting from $\bullet$ is a continuous finite δ'-path.

4.2 From Continuous Paths to Linear Infinite Derivations

We now discuss that the existence of an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ such that a continuous infinite δ-path exists implies the existence of a linear infinite $\star$-chase derivation δ_ℓ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. Starting from δ, we are going to construct a sequence of instances, which eventually will lead to the desired derivation δ_ℓ. The construction proceeds in three main steps:

Useful part of δ. We first isolate a useful part of the $\star$-chase derivation $\delta = (I_i)_{i \geq 0}$. Recall that there exists a continuous infinite δ-path $P = (\alpha_i)_{i \geq 0}$. By stickiness, there exists $j \geq 0$ such that α_j is the last atom on P in which a term t becomes *sticky*. The latter means that the first time t participates in a join is during the trigger application that generates α_j, and thus t occurs in (or sticks to) every atom of $\{\alpha_i\}_{i \geq j}$. Let $k \geq j$ be the integer such that α_k is the first atom on P after α_j in which a new null is invented. The useful part of δ that we are going to focus on is the infinite sequence of atoms $(\alpha_i)_{i \geq k}$, which we call the *backbone*, and the atoms of $\bigcup_{i \geq 0} I_i$, which we call *side atoms*, that are needed to generate the atoms on the backbone via a single trigger application. In other words, for a backbone atom α, if α is obtained via the trigger (σ, h) for Σ on instance I_i, for some $i \geq 0$, then the atoms $h(\mathsf{body}(\sigma))$, excluding the backbone atoms, are side atoms.

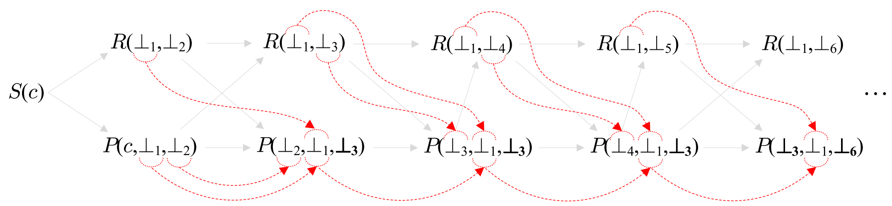
▶ **Example 9.** Consider again the set $\Sigma \in \mathbb{S}$ from Example 8. As discussed above, there exists an infinite o-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ such that a continuous infinite δ-path exists (see the figures above). The useful part of δ is as shown below



Observe that the last atom on the continuous path in which a term becomes sticky is $P(\perp_2, \perp_1, \perp_3)$; in fact, the sticky term is $\perp_1$, which is the only sticky term on the continuous path. It happened that $P(\perp_2, \perp_1, \perp_3)$ invents also a new null, that is, $\perp_3$, and therefore the suffix of the continuous path that starts at $P(\perp_2, \perp_1, \perp_3)$ is the backbone. It is now easy to verify that all the other atoms, apart from $S(c)$, indeed contribute in the generation of a backbone atom via a single trigger application. ■

Renaming step. We proceed to rename some of the nulls that occur in backbone atoms or side atoms. In particular, for every null $\perp$ occurring in a side atom α, we apply the following renaming steps; fix a constant $c \in \mathsf{dom}(\mathsf{cr}(\Sigma))$: (i) every occurrence of $\perp$ in α is replaced by c, and (ii) every occurrence of $\perp$ in a backbone atom β that is propagated from α to β is replaced by c. For a backbone or side atom α, let $\rho(\alpha)$ be the atom obtained from α after globally applying the above renaming steps. We now define the sequence of instances $\delta' = (J_i)_{i \geq 0}$ as follows: $J_0 = \{\rho(\alpha) \mid \alpha \text{ is a side atom }\} \subseteq \mathsf{cr}(\Sigma)$ and $J_i = J_{i-1} \cup \{\rho(\alpha_{k+i-1})\} \cup H$, where H is the set of atoms that are generated together with α_{k+i-1} (since we can have a conjunction of atoms in the head of a tgd) after renaming the nulls that do not occur in $\rho(\alpha_{k+i-1})$ to c. Notice that H is empty in case of single-head tgds. It is crucial to observe that a new null generated in a backbone atom never participates in a join. This is because the first backbone atom α_k comes after the atom α_j, which is the last atom on P in which a term becomes sticky. This fact allows us to modify triggers from δ in order to construct, for every $i \geq 0$, a trigger (σ_i, h_i) such that $J_i\langle\sigma_i, h_i\rangle J_{i+1}$.

▶ **Example 10.** We consider again our running example. Before renaming the nulls that appear in side atoms, we first need to understand how nulls are propagated from side atoms to backbone atoms during the chase. This is depicted in the following figure



Notice that the boldfaced occurrences of the nulls $\perp_3, \perp_6, \ldots$ are not propagated from side atoms, but generated on the backbone, and thus will not be renamed. Let us recall that the existence of such nulls is guaranteed by continuity. By applying the renaming step, i.e., by replacing every null in a side atom with the constant c, and then propagate it to the backbone as indicated above, we get the sequence of instances $J_0 = \{R(c,c), P(c,c,c)\} \subseteq \mathsf{cr}(\Sigma)$, $J_1 = J_0 \cup \{P(c,c,\perp_3), R(c,\perp_3)\}$, $J_2 = J_1 \cup \{P(c,c,\perp_3)\}$, $J_3 = J_2 \cup \{P(c,c,\perp_3)\}$, $J_4 = J_3 \cup \{P(\perp_3,c,\perp_6), R(c,\perp_6)\}, \ldots$. Observe that, due to stickiness, none of the nulls $\perp_3, \perp_6, \ldots$ generated on the backbone participates in a join. This means that the renaming step preserves all the joins, and thus, by adapting triggers from δ, we can devise a valid trigger for each pair (J_i, J_{i+1}) of instances. ∎

Pruning step. At this point, one may be tempted to think that $\delta' = (J_i)_{i \geq 0}$, with $J_i\langle\sigma_i, h_i\rangle J_{i+1}$ for $i \geq 0$, is the desired linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. It is easy to verify that we have the infinite sequence of atoms $(\rho(\alpha_i))_{i \geq k-1}$ such that $\rho(\alpha_{k-1}) \in \mathsf{cr}(\Sigma)$ since α_{k-1} is a side atom, and for each $i \geq k-1$, $J_{i-k+2} \supseteq J_{i-k+1} \cup \{\rho(\alpha_{i+1})\}$, and there exists $\beta \in \mathsf{body}(\sigma_i)$ such that $h_i(\beta_i) = \alpha_i$ and $h_i(\mathsf{body}(\sigma_i) \setminus \{\beta\}) \subseteq \mathsf{cr}(\Sigma)$. However, we cannot conclude yet that δ' is the desired derivation for the following two reasons: (i) triggers may repeat, i.e., we may have $i \neq j$ such that $\sigma_i = \sigma_j = \sigma$ and $h_i \diamond_\sigma^\star h_j$, where $\diamond_\sigma^\star$ is $=$ (resp., $\sim_\sigma$) if $\star = \mathsf{o}$ (resp., $\star = \mathsf{so}$), and (ii) we may have $i \neq j$ such that $\rho(\alpha_i) = \rho(\alpha_j)$, i.e., the sequence of atoms $(\rho(\alpha_i))_{i \geq k-1}$ does not consist of distinct atoms. This can be easily fixed by pruning the subderivation between the two repeated triggers or atoms. But since this pruning step may be applied infinitely many times, the question that comes up is whether the obtained $\star$-chase derivation δ'' is infinite. Interestingly, this is the case due to continuity. Since the backbone $(\alpha_i)_{i \geq k}$ is part of a continuous δ-path, we conclude that two repeated triggers or atoms are necessarily between two atoms α and β in which new nulls are invented, while no atom between α and β invents a new null. The fact that we have infinitely many pairs of such atoms on the backbone, we immediately conclude that after the pruning step the obtained $\star$-chase derivation is infinite. Thus, δ'' is a linear infinite $\star$-chase derivation of J_0 w.r.t. Σ. Since $J_0 \subseteq \mathsf{cr}(\Sigma)$, we can easily construct a linear infinite $\star$-chase derivation δ_ℓ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ by simply adding to J_0 the set of atoms $\mathsf{cr}(\Sigma) \setminus J_0$, and the claim follows.

▶ **Example 11.** Coming back to our running example, it can be seen that the sequence of instances devised in Example 10 is not the desired linear derivation due to repeated triggers and atoms. However, after applying the pruning step, we get the sequence of instances $J'_0 = J_0$, $J'_1 = J_1$, $J'_2 = J'_1 \cup \{P(\perp_3,c,\perp_6), R(c,\perp_6)\}$, $J'_3 = J'_2 \cup \{P(\perp_6,c,\perp_9), R(c,\perp_9)\}, \ldots$. Now, it is easy to verify that after adding the atom $S(c)$ in J'_0, we get (modulo null renaming) the linear infinite o-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ given in Example 6. ∎

5 Graph-Based Characterization of Chase Termination

In this section, we characterize the termination of the (semi-)oblivious chase for sticky sets of tgds via graph-based conditions. More precisely, we show that a set $\Sigma \in \mathbb{S}$ belongs to $\mathbb{CT}_{\forall}^{\star}$ iff a linearized version of it, i.e., a set of linear tgds obtained from Σ, enjoys a condition similar to rich-acyclicity [16], if $\star = \mathsf{o}$, and weak-acyclicity [10], if $\star = \mathsf{so}$. Recall that linear tgds are tgds with only one body atom [6]; we write $\mathbb{L}$ for the class of linear tgds. The proof of the above result proceeds in two steps:

1. We first show that the given sticky set Σ of tgds can be rewritten into a set of linear tgds, while this rewriting preserves chase termination. This heavily relies on Theorem 7, which establishes that non-termination of the (semi-)oblivious chase coincides with the existence of a linear infinite chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.
2. We then extend recent characterizations from [4], which are based on extensions of rich-acyclicity and weak-acyclicity, of the termination of the (semi-)oblivious chase under constant-free linear tgds in order to deal with constants in the tgds. Notice that, although in other contexts, e.g., query answering and containment under tgds, the transition from constant-free tgds to tgds with constants is relatively straightforward, in the context of chase termination the constants in the tgds cause additional complications that must be carefully treated.

We proceed to give more details for the above two steps.

5.1 Linearization

Before presenting the linearization procedure, we need to introduce some auxiliary notions. Given a tgd σ and an atom $\alpha \in \mathsf{body}(\sigma)$, let $V_{\alpha,\sigma} = \mathsf{var}(\mathsf{body}(\sigma) \setminus \{\alpha\})$, that is, the set of body variables of σ that do not occur only in α. Moreover, given a set Σ of tgds, a tgd $\sigma \in \Sigma$, and an atom $\alpha \in \mathsf{body}(\sigma)$, let $M_{\alpha,\sigma}^{\Sigma} = \{h \mid h : V_{\alpha,\sigma} \to \mathsf{dom}(\mathsf{cr}(\Sigma))\}$, i.e., the set of all possible mappings from the variables of $V_{\alpha,\sigma}$ to the constants occurring in the critical database for Σ.

▶ **Definition 12.** Consider a set Σ of tgds. The *linearization* of a tgd $\sigma \in \Sigma$ (w.r.t. Σ) of the form $\phi(\bar{x}, \bar{y}) \to \exists \bar{z}\, \psi(\bar{x}, \bar{z})$, denoted $\mathsf{Lin}(\sigma)$, is the set of linear tgds

$$\bigcup_{\alpha \in \phi(\bar{x},\bar{y})} \; \bigcup_{h \in M_{\alpha,\sigma}^{\Sigma}} \{h(\alpha) \to \exists \bar{z}\, h(\psi(\bar{x}, \bar{z}))\}.$$

The linearization of Σ is defined as $\mathsf{Lin}(\Sigma) = \bigcup_{\sigma \in \Sigma} \mathsf{Lin}(\sigma)$. ■

In simple words, the linearization procedure converts a tgd σ into a set of linear tgds by keeping only one atom α from $\mathsf{body}(\sigma)$, while the variables in $\mathsf{body}(\sigma) \setminus \{\alpha\}$ are instantiated with constants from $\mathsf{cr}(\Sigma)$ in all the possible ways. Theorem 7 allows us to show that the linearization procedure preserves the termination of the (semi-)oblivious chase whenever the input set of tgds is sticky.

▶ **Theorem 13.** *Consider a set $\Sigma \in \mathbb{S}$ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\Sigma \in \mathbb{CT}_{\forall}^{\star}$ iff $\mathsf{Lin}(\Sigma) \in \mathbb{CT}_{\forall}^{\star}$.*

5.2 Acyclicity Conditions

We proceed to extend the characterizations of the termination of the (semi-)oblivious chase under constant-free linear tgds established in [4]. The goal is to show that, given a set of tgds $\Sigma \in \mathbb{L}$, which may contain constants, $\Sigma \in \mathbb{CT}_{\forall}^{\mathsf{o}}$ iff Σ is critically-richly-acyclic, and $\Sigma \in \mathbb{CT}_{\forall}^{\mathsf{so}}$ iff Σ critically-weakly-acyclic, where critical-rich- and critical-weak-acyclicity are appropriate extensions of rich- and weak-acyclicity proposed in [4]. These notions rely on the dependency graph of a set of tgds, which we now recall. We assume a fixed order on the head-atoms of tgds. For a tgd σ with $\mathsf{head}(\sigma) = \alpha_1, \ldots, \alpha_k$, we write (σ, i) for the *single-head* tgd, i.e., the tgd with only one atom in its head, obtained from σ by keeping only the atom α_i, and the existentially quantified variables in α_i. Recall that $\mathsf{pos}(\alpha, x)$ is the set of positions in α at which x occurs.

▶ **Definition 14.** The *dependency graph* of a set Σ of tgds is a labeled directed multigraph $\mathsf{dg}(\Sigma) = (N, E, \lambda)$, where $N = \mathsf{pos}(\mathsf{sch}(\Sigma))$, $\lambda : E \to \Sigma \times \mathbb{N}$, and E contains only the following edges. For each $\sigma \in \Sigma$ with $\mathsf{head}(\sigma) = \alpha_1, \ldots, \alpha_k$, for each $x \in \mathsf{fr}(\sigma)$, and for each $\pi \in \mathsf{pos}(\mathsf{body}(\sigma), x)$:

- For each $i \in [k]$, and for each $\pi' \in \mathsf{pos}(\alpha_i, x)$, there is a *normal* edge $e = (\pi, \pi') \in E$ with $\lambda(e) = (\sigma, i)$.
- For each existentially quantified variable z in σ, for each $i \in [k]$, and for each $\pi' \in \mathsf{pos}(\alpha_i, z)$, there is a *special* edge $e = (\pi, \pi') \in E$ with $\lambda(e) = (\sigma, i)$. ■

A normal edge (π, π') keeps track of the fact that a term may propagate from π to π' during the chase. Moreover, a special edge (π, π'') keeps track of the fact that the propagation of a value from π to π' also creates a null at position π''. As we shall see, the dependency graph is appropriate when we consider the semi-oblivious chase. For the oblivious chase, we need an extended version of it. The *extended dependency graph* of Σ, denoted $\mathsf{edg}(\Sigma)$, is obtained from $\mathsf{dg}(\Sigma)$ by simply adding special labeled edges from the positions where non-frontier variables occur to the positions where existentially quantified variables occur.

Two well-known classes of tgds, introduced in the context of data exchange, that guarantee the termination of the oblivious and semi-oblivious chase are rich-acyclicity and weak-acyclicity, respectively. A set Σ is richly-acyclic (resp., weakly-acyclic) if there is no cycle in $\mathsf{edg}(\Sigma)$ (resp., $\mathsf{dg}(\Sigma)$) that contains a special edge. It would be very useful if, whenever we focus on linear tgds, rich- and weak-acyclicity are also necessary conditions for the termination of the oblivious and semi-oblivious chase, respectively. Unfortunately, this is not the case. A simple counterexample follows:

▶ **Example 15.** Consider the set Σ of linear tgds consisting of $R(x, x) \to \exists z\, R(z, x)$. It is easy to verify that in $\mathsf{dg}(\Sigma) = \mathsf{edg}(\Sigma)$ there is a cycle that contains a special edge. However, for $\star \in \{\mathsf{o}, \mathsf{so}\}$, there is only one $\star$-chase sequence of $\mathsf{cr}(\Sigma)$ w.r.t. Σ that is terminating; thus, $\Sigma \in \mathbb{CT}_{\forall}^{\star}$. ■

As it has been shown in [4], there is an extension of rich- and weak-acyclicity, called critical-rich- and critical-weak-acyclicity, that whenever we focus on linear tgds it provides a necessary and sufficient condition for the termination of the oblivious and semi-oblivious chase, respectively. However, the analysis performed in [4] considers only tgds without constants, while after the linearization of a sticky set Σ of tgds, even if Σ is constant-free, the obtained set $\mathsf{Lin}(\Sigma)$ contains at least one constant. Thus, in order to be able to apply critical-rich- and critical-weak-acyclicity on $\mathsf{Lin}(\Sigma)$, we first need to appropriately extend these notions to linear tgds with constants.

A crucial notion underlying critical-rich- and critical-weak-acyclicity is the notion of compatibility among two single-head linear tgds. Intuitively, if a single-head linear tgd σ_1 is compatible with a single-head linear tgd σ_2, then the atom obtained during the chase by applying σ_1 may trigger σ_2. It is clear that the presence of constants in the tgds affects the way that we define compatibility. We assume the reader is familiar with the notion of unification. Given two atoms α, β, we write $\mathsf{mgu}(\alpha, \beta)$ for their most general unifier. For brevity, we write Π_t^{σ} for the set of positions $\mathsf{pos}(\mathsf{body}(\sigma), t)$, i.e., the set of positions at which the term t occurs in the body of σ. We also write $\mathsf{term}(\alpha, \Pi)$, where α is an atom, and Π a set of positions, for the set of terms occurring in α at positions of Π.

▶ **Definition 16.** Consider two single-head linear tgds σ_1 and σ_2. We say that σ_1 is *compatible* with σ_2 if the following hold:

1. $\mathsf{head}(\sigma_1)$ and $\mathsf{body}(\sigma_2)$ unify.

2. For each $x \in \mathsf{var}(\mathsf{body}(\sigma_2))$, either $\mathsf{term}(\mathsf{head}(\sigma_1), \Pi_x^{\sigma_2}) = \{z\}$ for some existentially quantified variable z in σ_1, or $\mathsf{term}(\mathsf{head}(\sigma_1), \Pi_x^{\sigma_2}) \subseteq \mathsf{fr}(\sigma_1) \cup \{c\}$ for some constant c.
3. For each $c \in \mathsf{const}(\mathsf{body}(\sigma_2))$, $\mathsf{term}(\mathsf{head}(\sigma_1), \Pi_c^{\sigma_2}) \subseteq \mathsf{fr}(\sigma_1) \cup \{c\}$. ◼

Having the notion of compatibility among two single-head linear tgds in place, we can recall the resolvent of a sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds, which is in turn a single-head tgd. Roughly, such a resolvent mimics the behavior of the sequence $\sigma_1, \ldots, \sigma_n$ during the chase. Notice that the existence of such a resolvent is not guaranteed, but if it exists, this implies that we may have a sequence of trigger applications that involve the tgds $\sigma_1, \ldots, \sigma_n$ in this order. In such a case, we call the sequence $\sigma_1, \ldots, \sigma_n$ active. The formal definitions follow:

▶ **Definition 17.** The *resolvent* of a sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds, denoted $[\sigma_1, \ldots, \sigma_n]$, is inductively defined as follows; for brevity, we write ρ for $[\sigma_1, \ldots, \sigma_{n-1}]$:

1. $[\sigma_1] = \sigma_1$;
2. $[\sigma_1, \ldots, \sigma_n] = \gamma(\mathsf{body}(\rho)) \to \gamma(\mathsf{head}(\sigma_n))$, where $\gamma = \mathsf{mgu}(\mathsf{head}(\rho), \mathsf{body}(\sigma_n))$, if $\rho \neq \Diamond$ and ρ is compatible with σ_n; otherwise, $[\sigma_1, \ldots, \sigma_n] = \Diamond$.

The sequence $\sigma_1, \ldots, \sigma_n$ is called *active* if $[\sigma_1, \ldots, \sigma_n] \neq \Diamond$. ◼

At this point, one may think that the right extension of rich- and weak-acyclicity, which will provide a necessary condition for the termination of the oblivious and semi-oblivious chase under linear tgds, is to allow cycles with special edges in the underlying dependency graph as long as the corresponding sequence of single-head tgds, which can be extracted from the edge labels, is not active. Unfortunately, this is not enough. If a cycle with a special edge is labeled with an active sequence, then we can only conclude that it will be traversed at least once during the chase. However, it is not guaranteed that it will be traversed infinitely many times.

▶ **Example 18.** Consider the set Σ of linear tgds consisting of

$$\sigma_1 = R(x,y,z) \ \to\ P(x,y,z) \qquad \sigma_2 = P(x,y,x) \ \to\ \exists z\, R(y,z,x).$$

In $\mathsf{dg}(\Sigma) = \mathsf{edg}(\Sigma)$ there is an active cycle that contains a special edge; e.g., $C = R[2], P[2], R[2]$, which corresponds to the sequence of tgds σ_1, σ_2. It is easy to see that $[\sigma_1, \sigma_2] \neq \Diamond$, and thus C is active. Despite the existence of an active cycle that contains a special edge, we can show that $\Sigma \in \mathbb{CT}_{\forall}^{\star}$, where $\star \in \{\mathsf{o}, \mathsf{so}\}$. Indeed, every $\star$-chase sequence of $\mathsf{cr}(\Sigma)$ w.r.t. Σ is terminating. ◼

A cycle that is labeled with an active sequence $\sigma_1, \ldots, \sigma_n$, and contains a special edge, will be certainly traversed infinitely many times if the resolvent of the sequence $\rho, \ldots, \rho$ of length k, where $\rho = [\sigma_1, \ldots, \sigma_n]$, exists, for every $k > 0$. Interestingly, for ensuring the latter condition, it suffices to consider sequences of length at most $(\omega + 1)$, where ω is the arity of the predicate of $\mathsf{body}(\sigma_1)$. This brings us to critical sequences. For brevity, we write σ^k for the sequence $\sigma, \ldots, \sigma$ of length k.

▶ **Definition 19.** A sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds is *critical* if $\sigma_1, \ldots, \sigma_n$ is active, and $[\sigma_1, \ldots, \sigma_n]^{\omega+1}$ is active, where ω is the arity of the predicate of $\mathsf{body}(\sigma_1)$. ◼

We are now ready to recall critical-rich- and critical-weak-acyclicity. They are essentially rich- and weak-acyclicity, with the difference that a cycle in the underlying graph is considered as "dangerous", not only if it contains a special edge, but also if it is labeled with a critical sequence.

▶ **Definition 20.** Consider a set $\Sigma \in \mathbb{L}$ of tgds, and let $G = (N, E, \lambda)$ be either $\mathsf{edg}(\Sigma)$ or $\mathsf{dg}(\Sigma)$. A cycle $v_0, v_1, \ldots, v_n, v_0$ in G is *critical* if $\lambda(v_0, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_n, v_0)$ is critical. We say that Σ is *critically-richly-acyclic* (resp., *critically-weakly-acyclic*), if no critical cycle in $\mathsf{edg}(\Sigma)$ (resp., $\mathsf{dg}(\Sigma)$) contains a special edge. ◼

The desired result follows:

▶ **Theorem 21.** *Consider a set $\Sigma \in \mathbb{L}$ of tgds. The following hold:*

- $\Sigma \in \mathbb{CT}_{\forall}^{\mathsf{o}}$ *iff* Σ *is critically-richly-acyclic.*
- $\Sigma \in \mathbb{CT}_{\forall}^{\mathsf{so}}$ *iff* Σ *is critically-weakly-acyclic.*

The "if" directions of the above result are shown by giving proofs similar to the ones given in [16] and [10] for showing that rich-acyclicity and weak-acyclicity guarantees the termination of the oblivious and restricted chase, respectively. The interesting direction is the "only if" direction. By Corollary 4, it suffices to show that if Σ is not critically-richly-acyclic (resp., critically-weakly-acyclic), then there exists an infinite o-chase (resp., so-chase) derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. This is a non-trivial result that requires a couple of auxiliary lemmas.

The equality type of an atom is a set of equalities among positions, as well as among positions and constants, that faithfully describes its shape. Formally, for an atom $\alpha = R(t_1, \ldots, t_n)$, the *equality type* of α is $\mathsf{eqtype}(\alpha) = \{R[i] = R[j] \mid t_i = t_j\} \cup \{R[i] = c \mid c \in \mathbf{C} \text{ and } t_i = c\}$. For a linear tgd σ, we write $\mathsf{eqtype}(\sigma)$ for the equality type of the atom $\mathsf{body}(\sigma)$. The next result establishes a useful connection between active sequences and equality types:

▶ **Lemma 22.** *Consider a single-head linear tgd σ such that σ^i is active, for some $i > 1$, and* $\mathsf{eqtype}([\sigma^{i-1}]) = \mathsf{eqtype}([\sigma^i])$. *Then, σ^{i+1} is active, and* $\mathsf{eqtype}([\sigma^i]) = \mathsf{eqtype}([\sigma^{i+1}])$.

Despite the fact that the above lemma has been already shown in [4] for constant-free tgds, it turned out that the proof from [4] cannot be easily extended to tgds with constants. Thus, we had to devise a completely new proof that exploits further properties of the resolvent of a sequence $\sigma, \ldots, \sigma$. In fact, we show via an inductive argument that $[\sigma^i] = [[\sigma^{i-1}], \sigma]$ is the same (modulo variable renaming) with $[\sigma, [\sigma^{i-1}]]$, which in turn allows us to easily establish Lemma 22. Having the connection between active sequences and equality types provided by Lemma 22, we show the next lemma, which states that critical cycles can be traversed infinitely many times during the chase.

▶ **Lemma 23.** *Consider a critical sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds. For every $k > 0$, $[\sigma_1, \ldots, \sigma_n]^k$ is active.*

As for Lemma 22, even though the above result has been shown in [4] for constant-free tgds, we had to devise a completely new proof in order to deal with the constants that are present in the tgds. Let us briefly explain how Lemma 23 is shown. For brevity, let $\rho = [\sigma_1, \ldots, \sigma_n]$. Since $\sigma_1, \ldots, \sigma_n$ is critical, by definition we get that $\rho^{\omega+1}$ is active, where ω is the arity of the predicate of $\mathsf{body}(\sigma_1)$. The crucial step is to also show that $\mathsf{eqtype}([\rho^{\omega}]) = \mathsf{eqtype}([\rho^{\omega+1}])$. Then, by iteratively applying Lemma 22, we obtain that ρ^k is active for every $k > \omega + 1$. Since $\rho^{\omega+1}$ is active, we can conclude that ρ^k is active for every $1 \leq k \leq \omega + 1$, and Lemma 23 follows.

We can show via an inductive argument that an active sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds mimics the sequence of trigger applications that involve the tgds $\sigma_1, \ldots, \sigma_n$ (in this order), starting from an atom in the critical instance; in particular, the ground version of $\mathsf{body}([\rho^{\omega+1}])$. This fact, together with Lemma 23, allow us to show that a critical cycle of minimal length in the extended dependency graph (resp., dependency graph) that contains a special edge, gives rise to an infinite o-chase (resp., so-chase) derivation $\mathsf{cr}(\Sigma)$ w.r.t. Σ, and Theorem 21 follows.

It is now easy to see that Theorems 13 and 21 establish the main result of this section:

▶ **Corollary 24.** *Consider a set $\Sigma \in \mathbb{S}$ of tgds. The following hold:*

- $\Sigma \in \mathbb{CT}_{\forall}^{\mathsf{o}}$ *iff* $\mathsf{Lin}(\Sigma)$ *is critically-richly-acyclic.*
- $\Sigma \in \mathbb{CT}_{\forall}^{\mathsf{so}}$ *iff* $\mathsf{Lin}(\Sigma)$ *is critically-weakly-acyclic.*

6 Complexity of Chase Termination

In this final section, we pinpoint the complexity of the $\star$-chase termination problem under sticky sets of tgds. In particular, we establish the following complexity result:

▶ **Theorem 25.** *For* $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ *is* PSPACE*-complete, and* NLOGSPACE*-complete for predicates of bounded arity. The lower bounds hold even for tgds without constants.*

Upper Bounds. The problem $\mathsf{CT}^{\star}_{\forall}$ under constant-free linear tgds is PSPACE-complete, in general, and NLOGSPACE-complete for predicates of bounded arity [4]. However, despite the fact that, by Corollary 24, we can reduce $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ to $\mathsf{CT}^{\star}_{\forall}(\mathbb{L})$, we cannot directly exploit the complexity results from [4] for two reasons: (i) the linearized version of Σ contains at least one constant, while the results from [4] apply only to constant-free tgds, and (ii) the linearization procedure takes exponential time, in general, and polynomial time in the case of bounded-arity predicates; thus, we cannot afford to explicitly compute the set $\mathsf{Lin}(\Sigma)$, and then check for critical-rich- and critical-weak-acyclicity. Therefore, a more refined procedure is needed.

We focus on the complement of our problem, i.e., given a set $\Sigma \in \mathbb{S}$ of tgds, we want to check whether $\Sigma \notin \mathbb{CT}^{\star}_{\forall}$. By Corollary 24, it suffices to show that $\mathsf{Lin}(\Sigma)$ is not critically-richly-acyclic, if $\star = \mathsf{o}$, and not critically-weakly-acyclic, if $\star = \mathsf{so}$. The latter problems can be seen as a generalization of the standard graph reachability problem. Indeed, we need to check whether there exists a node v in the (extended) dependency graph of $\mathsf{Lin}(\Sigma)$ that is reachable from itself via a critical cycle that contains a special edge. However, as discussed above, we cannot explicitly construct $\mathsf{Lin}(\Sigma)$ and its (extended) dependency graph G. Instead, the above reachability check should be performed on a compact representation of G, which is the set Σ itself. We show that this check can be performed via a non-deterministic procedure that uses $O(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space, where ω is the maximum arity over all predicates in Σ, and m is the maximum number of atoms occurring in a tgd of Σ, and the desired upper bounds follow.

Lower Bounds. The PSPACE-hardness is shown by providing a polynomial time reduction from the acceptance problem of a deterministic polynomial space Turing machine M. Such a reduction can be easily devised if we are allowed to join a variable in the body of a tgd and then lose it, or if we can use constants in the body of a tgd. In this case, a configuration of M can be straightforwardly encoded in a single predicate *Config* of polynomial arity. However, if we want the set of tgds to be sticky and constant-free, then we need a more clever encoding for a configuration of M, which increases the arity of *Config*, but only polynomially.

The NLOGSPACE-hardness is immediately inherited from [4], where it is shown that $\mathsf{CT}^{\star}_{\forall}(\mathbb{L})$ is NLOGSPACE-hard, even for linear tgds that are constant-free, each body variable occurs only once (and thus, stickiness is trivially satisfied), and only unary and binary predicates are used.

7 Conclusions

We have shown that the uniform (semi-)oblivious chase termination problem for sticky sets of tgds is decidable, and obtained precise complexity results. This is done by first characterizing the termination of the (semi-)oblivious chase for sticky sets of tgds via graph-based conditions that are of independent interest. In particular, to check whether the oblivious (resp., semi-oblivious) chase terminates for a sticky set Σ of tgds, we simply need to linearize it, i.e., convert it, via an easy procedure, into a set $\mathsf{Lin}(\Sigma)$ of linear tgds, and then check whether $\mathsf{Lin}(\Sigma)$ enjoys an acyclicity condition in the spirit of rich-acyclicity (resp., weak-acyclicity). The next natural step is to concentrate on the restricted (a.k.a. standard) version of the chase, which makes the problem even more challenging due to its non-deterministic behaviour that cannot be captured via static graph-based conditions.

References

1 Alfred V. Aho, Yehoshua Sagiv, and Jeffrey D. Ullman. Efficient optimization of a class of relational expressions. *ACM Trans. Database Syst.*, 4(4):435–454, 1979.

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

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

4 Marco Calautti, Georg Gottlob, and Andreas Pieris. Chase termination for guarded existential rules. In *PODS*, pages 91–103, 2015.

5 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.

6 Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.

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

8 Alin Deutsch, Alan Nash, and Jeff B. Remmel. The chase revisisted. In *PODS*, pages 149–158, 2008.

9 Alin Deutsch and Val Tannen. Reformulation of XML queries and constraints. In *ICDT*, pages 225–241, 2003.

10 Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

11 Tomasz Gogacz and Jerzy Marcinkowski. All-instances termination of chase is undecidable. In *ICALP*, pages 293–304, 2014.

12 Gösta Grahne and Adrian Onet. Anatomy of the chase. *Fundam. Inform.*, 157(3):221–270, 2018.

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. Artif. Intell. Res.*, 47:741–808, 2013.

14 Sergio Greco, Cristian Molinaro, and Francesca Spezzano. *Incomplete Data and Data Dependencies in Relational Databases*. Morgan & Claypool Publishers, 2012.

15 Sergio Greco, Francesca Spezzano, and Irina Trubitsyna. Stratification criteria and rewriting techniques for checking chase termination. *PVLDB*, 4(11):1158–1168, 2011.

16 André Hernich and Nicole Schweikardt. Cwa-solutions for data exchange settings with target dependencies. In *PODS*, pages 113–122, 2007.

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

18 Bruno Marnette. Generalized schema-mappings: from termination to tractability. In *PODS*, pages 13–22, 2009.

19 Michael Meier, Michael Schmidt, and Georg Lausen. On chase termination beyond stratification. *PVLDB*, 2(1):970–981, 2009.

A Appendix

B Proof of Theorem 7

Before we proceed with the proof of the theorem, we point out that by Corollary 4, the ($\Leftarrow$) direction of the claim is easily shown. Furthermore, by exploiting once again Corollary 4, proving the ($\Rightarrow$) direction boils down to prove the following: if there exists an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, then there exists a linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.

We accomplish the above in two main steps. We first show that if an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ exists, then we can construct an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ whose chase relation $\prec_\delta$ admits a special infinite path, which we call *continuous*. Then, we show that the existence of such a continuous path implies the existence of a linear infinite $\star$-chase derivation, and the claim will follow. To this end, in order to simplify the proofs, in this section we assume w.l.o.g. that Σ is in *normal form* [7]. That is, for each $\sigma \in \Sigma$, $\mathsf{head}(\sigma)$ contains only one atom. We further assume that no two atoms in $\mathsf{body}(\sigma)$ share the same relation symbol. This last assumption will allow us to simplify the definition of a continuous path. It is not difficult to see that rewriting a set of tgds Σ' into a set Σ'' satisfying the above conditions preserves chase termination, i.e. $\Sigma' \in \mathbb{CT}^\star_\forall$ iff $\Sigma'' \in \mathbb{CT}^\star_\forall$, for $\star \in \{\mathsf{o}, \mathsf{so}\}$.

B.1 Finding an infinite continuous path

We proceed with the first part of the proof. We start by defining the notion of δ-path for some infinite $\star$-chase derivation δ, and then we introduce the notion of continuous path.

▶ **Definition 26.** Consider an infinite $\star$-chase derivation $\delta = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, with $I_i\langle\sigma_i, h_i\rangle I_{i+1}$, for $i \geq 0$. A *δ-path* is a (possibly infinite) sequence of atoms $(\alpha_i)_{i \geq 0}$ such that $\alpha_0 \in I_0$ and $\alpha_i \prec_\delta \alpha_{i+1}$, for $i \geq 0$. Furthermore, for an atom $\alpha \in \{\alpha_i\}_{i \geq 0}$, let $f_\delta(\alpha) = (I_i, \sigma_i, h_i)$, where $\alpha \in I_{i+1} \setminus I_i$, for some $i \geq 0$.

In what follows, let $\Sigma_\exists$ and $\Sigma_\forall$ be the sets of tgds in Σ with and without existential quantifiers, respectively. Furthermore, in order to provide uniform definitions for both the oblivious and the semi-oblivious chase, we refer to the graphs $\mathsf{edg}(\Sigma)$ and $\mathsf{dg}(\Sigma)$ with o-dependency graph and so-dependency graph respectively.

▶ **Definition 27.** Consider an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ and a finite δ-path $(\alpha_i)_{0 \leq i \leq n}$, with $f_\delta(\alpha_i) = (I_i, \sigma_i, h_i)$ for $i > 0$, and where $\{i \mid \sigma_i \in \Sigma_\exists\} = \{\ell_0, \ldots, \ell_m\}$ and $\ell_0 < \ldots < \ell_m$. The δ-path $(\alpha_i)_{i \geq 0}$ is *continuous* if $\ell_m = n$ and there exists a sequence of nulls $(\perp_i)_{0 \leq i \leq m}$ such that $\perp_i \in \mathsf{dom}(\alpha_{\ell_i}) \setminus \mathsf{dom}(I_{\ell_i})$ and for every $0 \leq i < m$, there is a path $\pi_0, \ldots, \pi_{n_i}$, where $n_i = \ell_{i+1} - \ell_i$, in the $\star$-dependency graph (N, E, λ) of Σ such that for every $0 \leq j < n_i$, $\lambda(\pi_j, \pi_{j+1}) = \sigma_{\ell_i+j+1}$ and $\pi_j \in \mathsf{pos}(\alpha_{\ell_i+j}, \perp_i)$.

Intuitively, a continuous δ-path $(\alpha_i)_{i \geq 0}$, is a path where all the atoms $\alpha_{\ell_0}, \alpha_{\ell_1}, \ldots, \alpha_{\ell_m}$ where a null is generated, are such that at least one null generated in α_{ℓ_i} must be propagated via the application of some triggers in δ, involving the atoms $\alpha_{\ell_i}, \alpha_{\ell_i+1}, \ldots, \alpha_{\ell_{i+1}}$. Here, the assumption that no two atoms share a relation symbol in the body of a tgd is crucial, as for every atom α in $(\alpha_i)_{i \geq 0}$, the body atom of some tgd in Σ to which α is mapped is uniquely determined.

We now extend the above definition to infinite paths as follows. An infinite δ-path $(\alpha_i)_{i \geq 0}$, with $f_\delta(\alpha_i) = (I_i, \sigma_i, h_i)$ for $i > 0$, where $\{i \mid \sigma_i \in \Sigma_\exists\} = \{\ell_i\}_{i \geq 0}$ and $\ell_0 < \ell_1 < \ldots$, is *continuous* if for $j \geq 0$, $(\alpha_j)_{0 \leq i \leq \ell_j}$ is continuous.

With the notion of continuous path in place, we are finally ready to prove our first main result.

▶ **Proposition 28.** *If there exists an infinite $\star$-chase derivation of* $\mathsf{cr}(\Sigma)$ *w.r.t.* Σ*, then there exists a continuous infinite δ-path, where δ is an infinite $\star$-chase derivation of* $\mathsf{cr}(\Sigma)$ *w.r.t.* Σ*.*

Proof. The proof of this proposition is further divided in two main parts. We first show that the existence of an infinite $\star$-chase derivation implies the existence of another infinite $\star$-chase derivation which we call *local*. Then, we prove that from a local infinite $\star$-chase derivation δ, it is possible to construct an infinite, rooted, directed acyclic graph (DAG) G. Such a graph enjoys some properties that allow us to conclude that G admits an infinite path which encodes a continuous infinite δ-path. The last claim is proved by exploiting the well-known König's lemma.

Local infinite $\star$-chase derivations. We start by defining *local* $\star$-chase derivations, that is derivations in which the application of a trigger (σ, h) is performed "as close as possible" to the first instance containing the atoms $h(\mathsf{body}(\sigma))$.

▶ **Definition 29.** Consider an infinite $\star$-chase derivation $\delta = (I_i)_{i \geq 0}$, where $I_i\langle\sigma_i, h_i\rangle I_{i+1}$, for $i \geq 0$. We say that δ is *local* if there exists a sequence of indices $0 = n_0 \leq n_1 \leq \ldots$ such that, for every $i \geq 0$, if i is even, then the tgds $\sigma_{n_i}, \ldots, \sigma_{n_{i+1}-1}$ are in $\Sigma_\forall$, otherwise they are in $\Sigma_\exists$. Moreover, for every $i \geq 0$ and $n_i \leq j \leq n_{i+1} - 1$, $h_j(\mathsf{body}(\sigma_j)) \subseteq I_{n_i}$.

We can easily show that whenever there exists an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, local infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ exists.

▶ **Lemma 30.** *If there exists an infinite $\star$-chase derivation of* $\mathsf{cr}(\Sigma)$ *w.r.t.* Σ*, then there exists an infinite local $\star$-chase derivation of* $\mathsf{cr}(\Sigma)$ *w.r.t.* Σ*.*

Proof. Let δ be an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. We can easily construct a local infinite $\star$-chase derivation $\eta = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, starting from δ, as follows. We inductively define a sequence of indices $n_0 \leq n_1 \leq \ldots$ and a sequence of instances $\eta = (I_i)_{i \geq 0}$ such that *(i)* $n_0 = 0$ and $I_{n_0} = \mathsf{cr}(\Sigma)$; *(ii)* for every $i > 0$, n_i is such that $(\sigma_0, h_0), \ldots, (\sigma_{m-1}, h_{m-1})$, with $m = n_i - n_{i-1}$, are all the triggers of δ such that $h_j(\mathsf{body}(\sigma_j)) \subseteq I_{n_{i-1}}$, for $0 \leq j \leq m - 1$. It is clear that we can apply all such triggers in any order, starting from $I_{n_{i-1}}$, obtaining a sequence of instances $I_{n_{i-1}}, \ldots, I_{n_i}$, where for every $0 \leq j \leq m-1$, $I_{n_{i-1}+j}\langle\sigma_j, h_j\rangle I_{n_{i-1}+j+1}$ and $h_j(\mathsf{body}(\sigma_j)) \subseteq I_{n_{i-1}}$. Clearly, η is a local infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. ◻

Infinite rooted DAG. We now proceed by proving that from a local infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, it is possible to extract an infinite, rooted, DAG $G = (N, E, \lambda)$, with labeling function λ. As we will see later, such a graph enjoys some properties that allow us to conclude that G admits an infinite (simple) path such that the concatenation of the labels of its edges coincides with a continuous infinite δ-path.

▶ **Lemma 31.** *If $\delta = (I_i)_{i \geq 0}$ is a local infinite $\star$-chase derivation of* $\mathsf{cr}(\Sigma)$ *w.r.t.* Σ*, then there exists an infinite, rooted DAG of finite degree $G = (N \cup \{\bullet\}, E, \lambda)$, where $\bullet$ is the root of G, $N \subseteq \bigcup_{i \geq 0} I_i$, every node in N is reachable from $\bullet$ and λ maps edges to sequences of atoms in $\bigcup_{i \geq 0} I_i$ such that, for every path $\bullet, v_1, \ldots, v_n$ in G, with $n \geq 1$, $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{n-1}, v_n)$ is a continuous δ-path.*

Proof. Let $\Sigma = \Sigma_\forall \cup \Sigma_\exists$. We represent the local $\star$-chase derivation δ as follows:

$$J_0 \overset{\Sigma_\forall^0}{\rightarrow} J_0^\forall \overset{\Sigma_\exists^0}{\rightarrow} J_1 \overset{\Sigma_\forall^1}{\rightarrow} J_1^\forall \overset{\Sigma_\exists^1}{\rightarrow} J_2, \ldots$$

where $J_0 = \mathsf{cr}(\Sigma)$ and for every $i \geq 0$, $\Sigma_\exists^i \subseteq \Sigma_\exists$, $\Sigma_\forall^i \subseteq \Sigma_\forall$, and $J_i \overset{\Sigma_\forall^i}{\rightarrow} J_i^\forall$ and $J_i^\forall \overset{\Sigma_\exists^i}{\rightarrow} J_{i+1}$ denote the $\star$-chase derivations in δ whose triggers use only tgds in $\Sigma_\forall^i$ and atoms in J_i and only tgds in $\Sigma_\exists^i$

and atoms in $J_i^\forall$, respectively. Please note that in general, it can be the case that $J_i \setminus J_{i-1}^\forall = \emptyset$, for $i > 0$. For the sake of clarity, we assume that for every $i > 0$, $J_i \setminus J_{i-1}^\forall \neq \emptyset$. The proof given below can be easily adapted to the more general case. We now inductively construct the infinite DAG $G = (N \cup \{\bullet\}, E, \lambda)$. In particular, we show that at each inductive step $i > 0$, for every atom $\beta \in J_i \setminus J_{i-1}^\forall$, there exists a path $\bullet, v_1, \ldots, v_i$ in G, such that $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{i-1}, v_i) = (\beta_i)_{0 \leq i \leq n}$, with $n \geq i$, is a continuous δ-path, with $\beta_n = \beta$.

Base Step. Let $i = 1$ and consider an atom $\beta \in J_i \setminus J_{i-1}^\forall$. Then, $\beta \in N$. By definition of $J_{i-1}^\forall \overset{\Sigma_\exists^{i-1}}{\to} J_i$, there exist two instances I' and J' in $J_{i-1}^\forall \overset{\Sigma_\exists^{i-1}}{\to} J_i$ such that $I'\langle \sigma, h\rangle J'$, for some $\sigma \in \Sigma_\exists^{i-1}$ and homomorphism h, and such that $J' \setminus I' = \{\beta\}$. Then, $\alpha \in N$, for every atom $\alpha \in h(\mathsf{body}(\sigma))$. Furthermore, we let $(\bullet, \alpha) \in E$ and $(\alpha, \beta) \in E$, whereas $\lambda(\bullet, \alpha) = \alpha$ and $\lambda(\alpha, \beta) = \beta$. Since $\sigma \in \Sigma_\exists$, there exists a null $\perp \in \mathsf{dom}(\beta) \setminus \mathsf{dom}(I')$ and since $h(\mathsf{body}(\sigma)) \subseteq \mathsf{cr}(\Sigma)$, we conclude that $\lambda(\bullet, \alpha), \lambda(\alpha, \beta) = \alpha, \beta$ is indeed a continuous δ-path.

Inductive Step. Let $i > 1$ and let $\gamma \in J_i \setminus J_{i-1}^\forall$. By definition of $J_{i-1}^\forall \overset{\Sigma_\exists^{i}}{\to} J_i$ there exist two instances I' and J' in $J_{i-1}^\forall \overset{\Sigma_\exists^{i}}{\to} J_i$ such that $I'\langle \sigma, h\rangle J'$, for some $\sigma \in \Sigma_\exists^i$ and homomorphism h and such that $J' \setminus I' = \{\gamma\}$. Since $\sigma \in \Sigma_\exists$, there exists a null $\perp \in \mathsf{dom}(\gamma) \setminus \mathsf{dom}(I')$. Since $i > 1$ and since δ is local, $h(\mathsf{body}(\sigma)) \subseteq J_{i-1}^\forall$ and there must be an atom $b \in \mathsf{body}(\sigma)$ such that, letting $\alpha = h(b)$, $\alpha \in J_{i-1}^\forall \setminus J_{i-2}^\forall$. In particular, there is a null $\perp' \in \mathsf{dom}(\alpha) \cap \mathsf{dom}(J_{i-1})$ generated at the previous step, by the derivation $J_{i-2}^\forall \overset{\Sigma_\exists^{i-2}}{\to} J_{i-1}$. This comes from the fact that every trigger in $J_{i-1} \overset{\Sigma_\forall^{i-1}}{\to} J_{i-1}^\forall$ does not introduce new nulls and from the fact that triggers using only atoms in $J_{i-2}^\forall$ have been already applied in $J_{i-2}^\forall \overset{\Sigma_\exists}{\to} J_{i-1}$. Furthermore, if $\star = \mathsf{so}$, the fact that triggers using only atoms in $J_{i-2}^\forall$ have been already applied, implies that such an atom α is also such that $\mathsf{pos}(\alpha, \perp') \cap \mathsf{frpos}(\sigma) \neq \emptyset$, where $\mathsf{frpos}(\sigma)$ denotes the set of positions in $\mathsf{head}(\sigma)$ in which a frontier variable occurs. Then, in the $\star$-dependency graph of Σ there is an edge $\pi_1 \to \pi_2$, labeled with σ, where $\pi_1 \in \mathsf{pos}(\alpha, \perp')$ and $\pi_2 \in \mathsf{pos}(\gamma, \perp)$. Now, let $\beta \in J_{i-1} \setminus J_{i-2}^\forall$ be the atom in which the null $\perp'$ has been generated in the derivation $J_{i-2}^\forall \overset{\Sigma_\exists^{i-2}}{\to} J_{i-1}$. That is, there are two instances I'' and J'' in $J_{i-2}^\forall \overset{\Sigma_\exists^{i-2}}{\to} J_{i-1}$, where $I''\langle \sigma', h'\rangle J''$ such that $J'' \setminus I'' = \{\beta\}$. Since $\perp' \in \mathsf{dom}(\alpha) \cap \mathsf{dom}(J_{i-1})$, there must be some atom $b' \in \mathsf{body}(\sigma')$ such that $\beta = h'(b')$, and there must also be an edge $\pi_0 \to \pi_1$, labeled with σ', in the $\star$-dependency graph of Σ, such that $\pi_0 \in \mathsf{pos}(\beta, \perp')$. By inductive hypothesis, there exists a path $\bullet, v_1, \ldots, v_{i-1}$ in G, such that $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{i-2}, v_{i-1}) = (\beta_i)_{0 \leq i \leq n}$, where $n \geq i - 1$, is a continuous δ-path and $\beta_n = \beta$. But this and discussion above implies that the sequence $\beta_0, \ldots, \beta_{n-1}, \beta, \alpha, \gamma$ is a η-continuous path. Thus, we let $\gamma \in N$ and $(\beta, \gamma) \in E$. Furthermore, we let $\lambda(\beta, \gamma) = \alpha, \gamma$.

As shown above, for every $i > 0$ and every atom $\beta \in J_i - J_{i-1}^\forall$, there exists a (finite) η-continuous path $(\beta_i)_{0 \leq i \leq n}$, where $\beta_n = \beta$.

Clearly, the constructed graph G is infinite, as at each step $i > 0$, at least a new node is added to N. The graph is of finite degree, as for every $i > 0$, and every atom $\beta \in J_i \setminus J_{i-1}^\forall$, all the edges from β to some other node in N are only introduced in step $i + 1$, and all such edges are finitely many. This ends our proof. ◻

With Lemma 30 and Lemma 31 in place, we are finally ready to prove our proposition. Indeed, combining the above two lemmas, the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ implies the existence of an infinite, rooted DAG of finite degree $G = (N \cup \{\bullet\}, E, \lambda)$ such that, for some infinite $\star$-chase derivation $\delta = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, $\bullet$ is the root,

$N \subseteq \bigcup_{i\geq 0} I_i$, every node in N is reachable from $\bullet$ and for every path $\bullet, v_1, \ldots, v_n$, for $n \geq 1$, $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{n-1}, v_n)$ is a continuous δ-path. We now exploit the well-known König's lemma for infinite, rooted, directed acyclic graphs with finite degree. We recall that König's lemma states that for every infinite, rooted DAG $G = (N \cup \{\bullet\}, E)$ of finite degree, with root $\bullet$, such that every node in N is reachable from $\bullet$, there exists an *infinite* (simple) path $\bullet, v_1, v_2, \ldots$ in G. Let $\bullet, v_1, v_2, \ldots$ be such an infinite simple path in G and let $(\alpha_i)_{i\geq 0} = \lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots$. Note that $(\alpha_i)_{i\geq 0}$ is a δ-path, as for every $i > 0$, $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots \lambda(v_{i-1}, v_i)$ is a δ-path. Thus, for every $i > 0$, let $f_\delta(\alpha_i) = (J_i, \sigma_i, h_i)$. Assume, towards a contradiction, that $(\alpha_i)_{i\geq 0}$ is not continuous. Then, there must exists $\ell \in \{i \mid \sigma_i \in \Sigma_\exists\}$ such that $(\alpha_i)_{0\leq i\leq \ell}$ is not continuous. By definition of G, there must also exist $j > \ell$ and $k > 0$, such that $\alpha_j = v_k$. Then, by definition of finite continuous δ-path, $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{k-1}, v_k) = (\alpha_i)_{0\leq i\leq j}$ is not continuous, which contradicts our hypothesis, thus a continuous infinite δ-path exists. This concludes our proof. ☐

B.2 Finding a linear infinite $\star$-chase derivation

With Proposition 28 in place, we can proceed with the second part of our proof. The main goal of this section is to show that if there exists an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ and a continuous infinite δ-path P, then a linear infinite $\star$-chase derivation δ_ℓ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ exists. As already described in the main body of the paper, the proof below proceeds in three main steps. We first isolate a suffix P' of P (the backbone) where all the terms participating in a join are coming from the side atoms of P'. Furthermore, by stickiness of Σ all such terms are finitely many. Then, we identify all the side atoms needed to generate the atoms in P', and we apply a careful renaming of nulls occurring in them, by modifying the homomorphisms used to generate the atoms in P'. Finally, we apply a pruning step in order to remove eventually repeated triggers and atoms.

▶ **Proposition 32.** *Consider an infinite $\star$-chase derivation $\delta = (J_i)_{i\geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, where $J_i\langle\rho_i, g_i\rangle J_{i+1}$, for each $i \geq 0$ and such that a continuous infinite δ-path exists. Then, there exists a linear infinite $\star$-chase derivation $\delta_\ell = (I_i)_{i\geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.*

Proof. Let $(\gamma_i)_{i\geq 0}$ be such an infinite continuous δ-path, where for each $i > 0$, let $f_\delta(\gamma_i) = (\Gamma_i, \kappa_i, f_i)$. In order to prove the claim, we need to introduce some useful notions. From stickiness of Σ, for every $i > 0$ and every variable x occurring more than once in $\mathsf{body}(\kappa_i)$, $f_i(x) \in \mathsf{dom}(\gamma_j)$, $\forall j \geq i$ [7]. That is, if a term participates in a join during the chase in some atom, this term will "stick" to all the next atoms in the path. In such a case, we say that the term $f_i(x)$ *becomes sticky* at position i. The above and the fact that the maximum arity of an atom is finite imply that there exists $i > 0$ and $\exists S \subseteq \mathsf{dom}(\gamma_i)$ such that for every term $t \in S$, there exists $j < i$ such that t becomes sticky at position j, and for every $j \geq i$, no other term, except for the ones in S, becomes sticky at position j. We say that such an i is a *saturated* position with sticky terms S.

Finding the backbone. From the definition of infinite continuous δ-path, if $\{i \mid \kappa_i \in \Sigma_\exists\} = \{\ell_i\}_{i\geq 0}$, with $\ell_0 < \ell_1 < \ldots$, then, for $j \geq 0$, $(\gamma_i)_{0\leq i\leq \ell_j}$ is continuous. We can restate the above definition as follows. $(\gamma_i)_{i\geq 0}$ is an infinite continuous δ-path if, given $\{i \mid \kappa_i \in \Sigma_\exists\} = \{\ell_i\}_{i\geq 0}$, with $\ell_0 < \ell_1 < \ldots$, there exists a sequence of nulls $(\perp_i)_{i\geq 0}$ such that $\perp_i \in \mathsf{dom}(\gamma_{\ell_i}) \setminus \mathsf{dom}(\Gamma_{\ell_i})$ and for every $i \geq 0$, there is a path $\pi_0, \ldots, \pi_{n_i}$, where $n_i = \ell_{i+1} - \ell_i$, in the $\star$-dependency graph (N, E, λ) of Σ such that for every $0 \leq j < n_i$, $\lambda(\pi_j, \pi_{j+1}) = \kappa_{\ell_i+j+1}$ and $\pi_j \in \mathsf{pos}(\gamma_{\ell_i+j}, \perp_i)$. Furthermore, recalling that we assume Σ is in normal form and thus no predicate symbol occurs more than once in the body of tgds in Σ, for every $i > 0$, let β_i be the *only* atom in $\mathsf{body}(\kappa_i)$ such that $f_i(\beta_i) = \gamma_{i-1}$.

Note that if for some $i > 0$, i is saturated with some sticky terms S, then for every $j > i$, also j is saturated with sticky terms S. Thus, there must exists $i \geq 0$ and an index ℓ_i such that ℓ_i (and every other index $\ell_{i+1}, \ell_{i+2}, \ldots$) is saturated with some sticky terms S.

In order to simplify the notation, assume that $\ell_1 - 1$ is saturated with some sticky terms S. We now want to construct a sequence of instances, based on the sequence of atoms $\gamma_{\ell_1-1}, \gamma_{\ell_1}, \gamma_{\ell_1+1} \ldots, \gamma_{\ell_2}, \ldots$, where γ_{ℓ_1-1} will become (after the proper renaming) the starting atom in the initial instance, whereas each atom in the sequence $\gamma_{\ell_1}, \gamma_{\ell_1+1} \ldots, \gamma_{\ell_2}, \ldots$, which we call *backbone*, will be part (after the renaming) of each of the next instances.

Side atoms of the backbone and renaming step. For the construction of the sequence of instances discussed above, we need first to identify the side atoms used to generate the atoms in the backbone. To this end, we inspect the homomorphisms used to generate the atoms in the backbone and identify the variables occurring in the side atoms that might inject null values in the backbone itself. We then modify these homomorphisms on such variables.

In what follows, fix an arbitrary constant $c \in \mathsf{dom}(\mathsf{cr}(\Sigma))$ and let $N = \mathsf{dom}(\gamma_{\ell_1-1}) \cap \mathbf{N}$ be the nulls occurring in γ_{ℓ_1-1}. For every $i > 0$ and every $0 \leq j \leq n_i - 1$, we define the following substitution:

$$\begin{array}{rcl} f^i_j & = & \{u \mapsto c \mid u \mapsto t \in f_{\ell_i+j} \text{ and } t \in N\} \cup \\ & & \{u \mapsto c \mid u \mapsto t \in f_{\ell_i+j} \text{ and } u \in \mathsf{var}(\mathsf{body}(\kappa_{\ell_i+j}) \setminus \{\beta_{\ell_i+j}\}) \text{ and } t \in \mathbf{N} \setminus N\} \cup \\ & & \{u \mapsto t \mid u \mapsto t \in f_{\ell_i+j} \text{ and either } t \in \mathbf{C} \text{ or} \\ & & \qquad\qquad u \notin \mathsf{var}(\mathsf{body}(\kappa_{\ell_i+j}) \setminus \{\beta_{\ell_i+j}\}) \text{ and } t \notin N \text{ hold}\} \end{array}$$

Roughly speaking, each substitution f^i_j is obtained from the original substitution f_{ℓ_i+j}, where *1)* variables that were mapped to nulls in N are now mapped to the constant c; *2)* variables in the "side atoms" of β_{ℓ_i+j} are now forcedly mapped to the constant c; *3)* all the other mappings satisfying none of the two properties above, are kept untouched. Thus, for every i and j, β_{ℓ_i+j} is the only atom in $\mathsf{body}(\kappa_{\ell_i+j})$ such that the application of f^i_j to it can give rise to an atom containing nulls. All other atoms $\beta \in \mathsf{body}(\kappa_{\ell_i+j}) \setminus \{\beta_{\ell_i+j}\}$ are such that $f^i_j(\beta) \in \mathsf{cr}(\Sigma)$. Since every mapping of the form $t \mapsto t$ in f_{ℓ_i+j}, where $t \in \mathbf{C}$, does not satisfy any of the two first properties, they are left untouched in f^i_j, and thus f^i_j is still a homomorphism. Finally, we consider the homomorphism $\hat{f}^i_j = f^i_j \cup (f'_{\ell_i+j} \setminus f_{\ell_i+j})$, where f'_{ℓ_i+j} is the extension of f_{ℓ_i+j} such that $f'_{\ell_i+j}(\mathsf{head}(\kappa_{\ell_i+j})) = \gamma_{\ell_i+j}$. We now construct the following sequence of instances:

$$I^1_{-1}, I^1_0, \ldots, I^1_{n_1-1} = I^2_{-1}, I^2_0, \ldots, I^2_{n_2-1} = I^3_{-1}, I^3_0, \ldots$$

where $I^1_{-1} = \mathsf{cr}(\Sigma)$ and for each $i > 0$ and each $0 \leq j \leq n_i - 1$, $I^i_j = I^i_{j-1} \cup \{\hat{f}^i_j(\mathsf{head}(\kappa_{\ell_i+j}))\}$. Note that $f^1_0(\beta_{\ell_1}) \in I^1_{-1}$ and that $f^1_0(\beta_{\ell_1})$ coincides with γ_{ℓ_1-1}, where all nulls occurring in it are replaced with the constant c.

In what follows, let $i > 0$ and $0 \leq j \leq n_i - 1$. We start by showing that $\hat{f}^i_j(\mathsf{head}(\kappa_{\ell_i+j})) = f^i_{j+1}(\beta_{\ell_i+j+1})$. Recall that δ is an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, thus $f'_{\ell_i+j}(\mathsf{head}(\kappa_{\ell_i+j})) = f_{\ell_i+j+1}(\beta_{\ell_i+j+1})$, where f'_{ℓ_i+j} is the extension of f_{ℓ_i+j} such that $f'_{\ell_i+j}(\mathsf{head}(\kappa_{\ell_i+j})) = \gamma_{l_i+j}$. Let $\mathsf{head}(\kappa_{\ell_i+j}) = R(u_1, \ldots, u_n)$ and let $\beta_{\ell_i+j+1} = R(t_1, \ldots, t_n)$. We show that for every $1 \leq k \leq n$, $\hat{f}^i_j(u_k) = f^i_{j+1}(t_k)$. This will immediately imply that $\hat{f}^i_j(\mathsf{head}(\kappa_{\ell_i+j})) = f^i_j(\beta_{\ell_i+j+1})$. Let $k \in \{1, \ldots, n\}$. We know that $f'_{\ell_i+j}(u_k) = f_{\ell_i+j+1}(t_k)$. We distinguish three only possible cases:

Case 1: $t = f'_{\ell_i+j}(u_k) = f_{\ell_i+j+1}(t_k)$ is a constant. Thus, from the definition of f^i_{j+1}, f^i_j and $\hat{f}^i_j$, $f'_{\ell_i+j}(u_k) = \hat{f}^i_j(u_k)$ and $f_{\ell_i+j+1}(t_k) = f^i_{j+1}(t_k)$. Thus, $\hat{f}^i_j(u_k) = f^i_{j+1}(t_k) = t$.

Case 2: $t = f'_{\ell_i+j}(u_k) = f_{\ell_i+j+1}(t_k)$ is a null in N, i.e. a null among the ones occurring in γ_{ℓ_1-1}. In this case, according to the definition of f^i_{j+1}, f^i_j and $\hat{f}^i_j$, we obtain that $\hat{f}^i_j(u_k) = f^i_{j+1}(t_k) = c$.

Case 3: $t = f'_{\ell_i+j}(u_k) = f_{\ell_i+j+1}(t_k)$ is a null in $\mathbf{N} \setminus N$. Note that then t_k is a variable and $t_k \notin \mathsf{var}(\mathsf{body}(\kappa_{\ell_i+j+1}) \setminus \{\beta_{\ell_i+j+1}\})$, because otherwise it means that t_k occurs more than once in $\mathsf{body}(\kappa_{\ell_i+j+1})$ (recall that t_k also occurs in β_{ℓ_i+j+1}) and it is mapped to a term not in S, which is not possible, since $\ell_i + j + 1$ is a saturated position with terms S. Then, from the definition of f^i_{j+1}, we obtain that $f^i_{j+1}(t_k) = f_{\ell_i+j+1}(t_k)$. It now remains to show that $\hat{f}^i_j(u_k) = f'_{\ell_i+j}(u_k)$. Assume towards a contradiction that $\hat{f}^i_j(u_k) \neq f'_{\ell_i+j}(u_k)$. Note that by definition of $\hat{f}^i_j$ and f'_{ℓ_i+j}, $\hat{f}^i_j = f^i_j \cup g$ and $f'_{\ell_i+j} = f_{\ell_i+j} \cup g$, where $g = f'_{\ell_i+j} \setminus f_{\ell_i+j}$. So, if $\hat{f}^i_j(u_k) \neq f'_{\ell_i+j}(u_k)$, it means that $f^i_j(u_k) \neq f_{\ell_i+j}(u_k)$. However, recall that also $\ell_i + j$ is a saturated position with sticky terms S, and thus, since $f_{\ell_i+j}(u_k) \in \mathbf{N} \setminus N$, $u_k \notin \mathsf{var}(\mathsf{body}(\kappa_{\ell_i+j}) \setminus \{\beta_{\ell_i+j}\})$. But then, by definition of f^i_j, $f^i_j(u_k) = f_{\ell_i+j}(u_k)$. This and the fact that $\hat{f}^i_j = f^i_j \cup g$ and $f'_{\ell_i+j} = f_{\ell_i+j} \cup g$ imply that indeed $\hat{f}^i_j(u_k) = f'_{\ell_i+j}(u_k)$. With this last case, we have finally shown that $\hat{f}^i_j(\mathsf{head}(\kappa_{\ell_i+j})) = f^i_{j+1}(\beta_{\ell_i+j+1})$.

We now proceed by recalling that, as discussed before, $f^i_j(\beta_{\ell_i+j})$ is the only atom in $f^i_j(\mathsf{body}(\kappa_{\ell_i+j}))$ that might contain nulls, all other atoms belong to $\mathsf{cr}(\Sigma)$, thus since $I^i_j = I^i_{j-1} \cup \{\hat{f}^i_j(\mathsf{head}(\kappa_{\ell_i+j}))\}$ and $I^1_{-1} = \mathsf{cr}(\Sigma)$ and since we have shown that $\hat{f}^i_j(\mathsf{head}(\kappa_{\ell_i+j})) = f^i_{j+1}(\beta_{\ell_i+j+1})$, we obtain that $f^i_j(\mathsf{body}(\kappa_{\ell_i+j})) \subseteq I^i_{j-1}$. Finally, since $\hat{f}^i_j$ is an extension of f^i_j such that $I^i_j = I^i_{j-1} \cup \{\hat{f}^i_j(\mathsf{head}(\kappa_{\ell_i+j}))\}$, it holds that $I^i_{j-1}\langle \kappa_{\ell_i+j}, f^i_j\rangle I^i_j$.

Pruning step. Note that even though we have shown that $I^i_{j-1}\langle\kappa_{\ell_i+j}, f^i_j\rangle I^i_j$, for every $i > 0$ and $0 \leq j \leq n_i - 1$, this does not imply that $I^1_{-1}, I^1_0, \ldots$ is a valid infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. We still need to argue about the triggers $(\kappa_{\ell_i+j}, f^i_j)$. It is possible that there exist two such triggers that are repeated (according to equality if $\star = \mathsf{o}$ or according to the relation $\sim$ if $\star = \mathsf{so}$). However, remember that $\gamma_{\ell_1-1}, \gamma_{\ell_1}, \ldots, \gamma_{\ell_1+n_1-1}, \gamma_{\ell_2}, \ldots$ is part of an infinite continuous δ-path. Thus, for every $i \geq 0$, a *fresh new null* generated in γ_{ℓ_i} is propagated to all atoms γ_{ℓ_i+k}, with $0 < k < n_i$, via the rule κ_{ℓ_i+k}, furthermore, if $\star = \mathsf{so}$, such a null is also propagated to the atom $\gamma_{\ell_{i+1}}$, via the rule $\kappa_{\ell_i+n_i}$. So, if two triggers are the same, they must be triggers used to construct some of the instances in $I^i_1, \ldots, I^i_{n_i-1}, I^{i+1}_0$. Then, there might exist two pairs of instances I^i_j, I^i_{j+1} and I^i_k, I^i_{k+1}, where $I^i_j\langle\kappa_{l_i+j+1}, f^i_{j+1}\rangle I^i_{j+1}$ and $I^i_k\langle\kappa_{l_i+k+1}, f^i_{k+1}\rangle I^i_{k+1}$, with $0 \leq j < k \leq n_i - 1$, such that $\kappa = \kappa_{\ell_i+j+1} = \kappa_{\ell_i+k+1}$ but $f^i_{j+1} = f^i_{k+1}$, if $\star = \mathsf{o}$ or $f^i_{j+1} \sim_\kappa f^i_{k+1}$, if $\star = \mathsf{so}$. It is easy to see that it is enough to discard all the instances I^i_l where $j < l \leq k$. Thus all triggers in this new sequence will be distinct (according to equality or $\sim$, depending on $\star$). Note however that the fact that all the remaining triggers are distinct does not necessarily imply that $I^i_{j+1} \setminus I^i_j \neq \emptyset$, for every $0 \leq j \leq n_i - 1$. Consider then, for every $0 \leq j \leq n_i - 1$, the instances I^i_{j+1} such that $I^i_{j+1} = I^i_j$. Then, there must be another instance I^i_{k+1}, with $0 < k < j$ such that $I^i_{k+1} \setminus I^i_k = \{\hat{f}^i_{k+1}(\mathsf{head}(\kappa_{\ell_i+k+1}))\}$, where $\hat{f}^i_{k+1}(\mathsf{head}(\kappa_{\ell_i+k+1}) = \hat{f}^i_{j+1}(\mathsf{head}(\kappa_{\ell_i+j+1}))$. In other words, the atom generated in I^i_{j+1} has already been generated in a previous step, in I^i_{k+1}. Thus, we can remove all the instances I^i_l, where $k + 1 < l \leq j + 1$. Note that the obtained sequence is still infinite, because, for every $i > 0$, a fresh new null is generated in I^i_0. Thus the obtained sequence, let it be $\delta_\ell = (I_i)_{i \geq 0}$, is an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, where $I_0 = \mathsf{cr}(\Sigma)$ and $I_i\langle\sigma_i, h_i\rangle I_{i+1}$ with $I_{i+1} = I_i \cup \{\alpha_{i+1}\}$, for $i \geq 0$. From the whole construction above, every atom $\alpha_0 \in h_0(\mathsf{body}(\sigma_0))$ is such that $\alpha_0 \prec_{\delta_\ell} \alpha_1$ and $\alpha_i \prec_{\delta_\ell} \alpha_{i+1}$. Finally, all such atoms are distinct and there exists an atom $\beta_i \in \mathsf{body}(\sigma_i)$ such that $\alpha_i = h_i(\beta_i)$ and $h_i(\mathsf{var}(\mathsf{body}(\sigma_i) \setminus \{\beta_i\})) \subseteq \mathsf{cr}(\Sigma)$. This ends the proof. $\square$

As already discussed at the beginning of the main section, the theorem immediately follows from Proposition 28, Proposition 32 and Corollary 4.

C Proof of Theorem 13

As already did for the previous section, we assume w.l.o.g. that our set of tgds Σ contains only one atom in the head.
($\Leftarrow$) Assume that $\mathsf{Lin}(\Sigma) \not\in \mathsf{CT}^{\star}_{\forall}$. From Corollary 4, there exists an infinite $\star$-chase derivation of $\mathsf{cr}(\mathsf{Lin}(\Sigma))$ w.r.t. $\mathsf{Lin}(\Sigma)$, $(I_i)_{i \geq 0}$, where $I_i\langle \sigma_i, h_i\rangle I_{i+1}$, for each $i \geq 0$. We now construct an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. For each $i \geq 0$, there is a tgd σ'_i in Σ such that $\sigma_i \in \mathsf{Lin}(\sigma'_i)$, that is, σ_i is obtained from the linearization of σ_i. Furthermore, let V_i be the set of variables in $\mathsf{body}(\sigma'_i)$ not occurring in h_i. That is, $V_i = \mathsf{var}(h_i(\mathsf{body}(\sigma'_i)))$. By definition of $\mathsf{Lin}(\sigma'_i)$, $V_i = V_{\alpha,\sigma'_i}$, for some atom $\alpha \in \mathsf{body}(\sigma'_i)$. Then, by definition of $\mathsf{Lin}(\sigma'_i)$, there must be a homomorphism $g_i \in M^{\Sigma}_{\alpha,\sigma'_i}$ such that $\mathsf{body}(\sigma_i) = g_i(\alpha)$ and $\mathsf{head}(\sigma_i) = g_i(\mathsf{head}(\sigma'_i))$. Let $h'_i = h_i \cup g_i$. Since $I_i\langle \sigma_i, h_i\rangle I_{i+1}$ and $\mathsf{body}(\sigma_i) = g_i(\alpha)$ hold and since g_i maps variables to constants in $\mathsf{cr}(\Sigma)$, h'_i is a homomorphism from $\mathsf{body}(\sigma'_i)$ to I_i, i.e. $h'_i(\mathsf{body}(\sigma'_i)) \subseteq I_i$. Thus, (σ'_i, h'_i) is a trigger for Σ on I_i. Furthermore, since $\mathsf{head}(\sigma_i) = g_i(\mathsf{head}(\sigma'_i))$, by construction of h'_i, $h_i(\mathsf{head}(\sigma_i)) = h_i(g_i(\mathsf{head}(\sigma'_i))) = h'_i(\mathsf{head}(\sigma'_i))$. This implies that $I_i\langle \sigma'_i, h'_i\rangle I_{i+1}$ holds. Finally, to show that $(I_i)_{i\geq 0}$ is indeed a valid $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, note that $I_0 = \mathsf{cr}(\Sigma) = \mathsf{cr}(\mathsf{Lin}(\Sigma))$ and also note that $h'_i \supseteq h_i$ and $\mathsf{var}(\mathsf{body}(\sigma'_i)) \supseteq \mathsf{var}(\mathsf{body}(\sigma_i))$. The last two expressions imply that whenever two triggers (σ, h_j) and (σ, h_k), for some $j, k \geq 0$, are such that $h_j \diamond^{\star}_{\sigma} h_k$, where $\diamond^{\mathsf{o}}_{\sigma}$ is $\neq$ and $\diamond^{\mathsf{so}}_{\sigma}$ is $\not\sim_{\sigma}$, also the two triggers (σ', h'_j) and (σ', h'_k) are such that $h'_j \diamond^{\star}_{\sigma'} h'_k$. Thus, $(I_i)_{i\geq 0}$ is also an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.
($\Rightarrow$) Assume that $\Sigma \not\in \mathsf{CT}^{\star}_{\forall}$. From Theorem 7, there exists an infinite $\star$-chase derivation $\delta = (I_i)_{i\geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, where $I_i\langle \sigma_i, h_i\rangle I_{i+1}$, for $i \geq 0$ and an infinite sequence of distinct atoms $(\alpha_i)_{i\geq 0}$, such that $\alpha_0 \in \mathsf{cr}(\Sigma)$, for each $i \geq 0$, $\alpha_i \prec_{\delta} \alpha_{i+1}$ and for each $i \geq 0$, $\alpha_{i+1} \in I_{i+1} \setminus I_i$ and there is an atom $\beta_i \in \mathsf{body}(\sigma_i)$ such that $\alpha_i = h_i(\beta_i)$ and $h_i(\mathsf{body}(\sigma_i) \setminus \{\beta_i\}) \subseteq \mathsf{cr}(\Sigma)$. Let us define, for every $i \geq 0$, $X_i = \mathsf{var}(\mathsf{body}(\sigma_i) \setminus \{\beta_i\})$. That is, the variables occurring in the atoms of $\mathsf{body}(\sigma_i) \setminus \{\beta_i\}$. Then, let $g_i = h_{i|X_i}$, we define the following set of linear tgds $\Sigma' = \{g_i(\beta_i) \rightarrow g_i(\mathsf{head}(\sigma_i))\}_{i\geq 0}$. Note that by construction of g_i and from the fact that $h_i(\mathsf{body}(\sigma_i) \setminus \{\beta_i\}) \subseteq \mathsf{cr}(\Sigma)$, $\Sigma' \subseteq \mathsf{Lin}(\Sigma)$. Since $\Sigma' \subseteq \mathsf{Lin}(\Sigma)$, $\Sigma' \not\in \mathbb{CT}^{\star}_{\forall}$ implies $\mathsf{Lin}(\Sigma) \not\in \mathbb{CT}^{\star}_{\forall}$. Hence, it is sufficient to show that there exists an infinite $\star$-chase derivation δ' of $\mathsf{cr}(\Sigma') \subseteq \mathsf{cr}(\Sigma)$ w.r.t. Σ'. To this end, for every $i \geq 0$, let $\sigma'_i = g_i(\beta_i) \rightarrow g_i(\mathsf{head}(\sigma_i))$, and let $f_i = h_i \setminus g_i$, i.e. the restriction of h_i on the variables in $\mathsf{var}(\beta_i) \setminus \mathsf{var}(\mathsf{body}(\sigma_i) \setminus \{\beta_i\})$. By construction of f_i, we immediately show that $f_i(\mathsf{body}(\sigma'_i)) = f_i(g_i(\beta_i)) = h_i(\beta_i)$. This, and the fact that $h_i(\beta_i) \in I_i$, implies that $f_i(\mathsf{body}(\sigma'_i) \in I_i$. Furthermore, let h'_i be the extension of h_i such that $h'_i(\mathsf{head}(\sigma_i)) = \alpha_{i+1} \in I_{i+1} \setminus I_i$. Then, since $f_i \cup g_i = h_i$, $f'_i = f_i \cup (h'_i \setminus h_i)$ is also an extension of f'_i, and by construction it is such that $f'_i(\mathsf{head}(\sigma'_i)) = f'_i(g_i(\mathsf{head}(\sigma_i))) = \alpha_{i+1}$. Thus, $I_i\langle \sigma'_i, f_i\rangle I_{i+1}$. What it remains to show is that for every $i \neq j$, $\sigma'_i = \sigma'_j = \sigma$ implies $f_i \diamond^{\star}_{\sigma} f_j$. Towards a contradiction, assume that there exist $i \neq j$ such that $\sigma'_i = \sigma'_j = \sigma$, but $f_i \diamond^{\star}_{\sigma} f_j$ does not hold. This implies that $\alpha_{i+1} = \alpha_{j+1}$. But this contradicts the fact that the atoms $(\alpha_i)_{i\geq 0}$ are distinct.

D Proof of Theorem 21

As already discussed in the main body of the paper, the proof of the ($\Leftarrow$) direction is along the lines of the proofs given in [16] and [10] for showing that rich-cyclicity and weak-acyclicity guarantee the termination of the oblivious and the restricted chase, respectively. Considering the ($\Rightarrow$) direction, we prove that if there is a critical cycle in $\mathsf{edg}(\Sigma)$, then $\Sigma \not\in \mathbb{CT}^{\mathsf{o}}_{\forall}$. The same proof can be used for showing the claim for the semi-oblivious chase, where the dependency graph is used instead.

Before we proceed further, let us first introduce some useful notions. In what follows, fix a constant $c \in \mathsf{cr}(\Sigma)$. Let $\sigma_0, \ldots, \sigma_{n-1}$ be a critical sequence of (single-head linear) tgds. Recall that from criticality of $\sigma_0, \ldots, \sigma_{n-1}$, the resolvent $\rho = [\sigma_0, \ldots, \sigma_{n-1}]$ exists and also the resolvent $[\rho^{\omega_{\sigma_0}+1}]$ exists, where ω_{σ_0} is the arity of the predicate of $\mathsf{body}(\sigma_0)$. We denote by $\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})$, the *canonical version* of the atom $\mathsf{body}([\rho^{\omega_{\sigma_0}+1}])$. That is, $\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})$ is the atom obtained from $\mathsf{body}([\rho^{\omega_{\sigma_0}+1}])$ by replacing every variable in it with c. Note that trivially, there exists an homomorphism from $\mathsf{body}([\rho^{\omega_{\sigma_0}+1}])$ to $\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})$. Furthermore, from Lemma 22 and from the definition of resolvent, every sequence $\rho_0, \ldots, \rho_{m-1}$, where for $k \in \mathbb{N}_0$, $k \cdot n \leq i < (k+1) \cdot n$ implies $\rho_i = \sigma_{i-k \cdot n}$ is such that $\mathsf{eqtype}(\mathsf{body}([\rho_0, \ldots, \rho_{m-1}])) \subseteq \mathsf{eqtype}(\mathsf{body}([\rho^{\omega_{\sigma_0}+1}]))$. Since there is also an homomorphism to $\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})$ from every atom α such that $\mathsf{eqtype}(\alpha) \subseteq \mathsf{eqtype}(\mathsf{body}([\rho^{\omega_{\sigma_0}+1}]))$, there exists an homomorphism from $\mathsf{body}([\rho_0, \ldots, \rho_{m-1}])$ to $\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})$ as well. We are now ready to show the following auxiliary lemma.

▶ **Lemma 33.** *Let $\sigma_0, \ldots, \sigma_{n-1}$ be a critical sequence of (single-head linear) tgds and let $I_0, I_1, \ldots, I_m$ be the sequence of instances such that $I_0 = \{\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})\}$ and $I_i\langle\rho_i, h_i\rangle I_{i+1}$, for $0 \leq i < m$, where for $k \in \mathbb{N}_0$, $k \cdot n \leq i < (k+1) \cdot n$ implies $\rho_i = \sigma_{i-k \cdot n}$. Moreover, assume that, for every $1 \leqslant i < m$, $h_i(\mathsf{body}(\rho_i)) \in (I_i \setminus I_{i-1})$*[2]*. Then, the atom obtained by applying (ρ_{m-1}, h_{m-1}) to I_{m-1} coincides (modulo null renaming) with the atom obtained by applying (ρ, μ) to I_0, where $\rho = [\rho_0, \ldots, \rho_{m-1}]$ and μ is the homomorphism mapping $\mathsf{body}(\rho)$ to $\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})$.*

Proof. The proof is by induction on $n > 0$.

Base Step. The claim holds trivially, since $\sigma_0 = [\sigma_0] = \rho$.

Inductive Step. By induction hypothesis, the atom obtained by applying (ρ_{m-2}, h_{m-2}) to I_{m-2} coincides, modulo null renaming, with the atom obtained by applying $(\hat{\rho}, g)$ to I_0, where $\hat{\rho} = [\rho_0, \ldots, \rho_{m-2}]$ and g is the homomorphism mapping $\mathsf{body}(\hat{\rho})$ to $\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})$. Therefore, $h_{m-1}(\mathsf{body}(\rho_{m-1})) = g'(\mathsf{head}(\hat{\rho}))$, where $g' \supseteq g$ maps each existential variable x of $\hat{\rho}$ to a "fresh" null. By construction, ρ is the tgd $\theta(\mathsf{body}(\hat{\rho})) \rightarrow \theta(\mathsf{head}(\rho_{m-1}))$, where $\theta = \mathsf{mgu}(\mathsf{head}(\hat{\rho}), \mathsf{body}(\rho_{m-1}))$. Assuming that $h'_{m-1}(\mathsf{head}(\rho_{m-1}))$, where $h'_{m-1} \supseteq h_{m-1}$, is the atom obtained by applying (ρ_{m-1}, h_{m-1}) to I_{m-1}, it is clear that $(g' \cup h'_{m-1})$ is a unifier for $\mathsf{head}(\hat{\rho})$ and $\mathsf{body}(\rho_{m-1})$. By definition of the most general unifier, there exists a substitution λ such that $(\lambda \circ \theta) = (g' \cup h'_{m-1})$. Observe that

$$\lambda(\mathsf{body}(\rho)) = \lambda(\theta(\mathsf{body}(\hat{\rho}))) = (g' \cup h'_{m-1})(\mathsf{body}(\hat{\rho})) = g(\mathsf{body}(\hat{\rho})) = \mathsf{can}(\sigma_0, \ldots, \sigma_{n-1}),$$

and

$$\lambda(\mathsf{head}(\rho)) = \lambda(\theta(\mathsf{head}(\rho_{m-1}))) = (g' \cup h'_{m-1})(\mathsf{head}(\rho_{m-1})) = h'_{m-1}(\mathsf{head}(\rho_{m-1})).$$

Since $\lambda_{|\mathsf{fr}(\rho)} = \mu$, the claim follows. ◻

We are now ready to prove that if Σ is not critically-richly-acyclic, than $\Sigma \notin \mathbb{CT}^{\circ}_{\forall}$. By hypothesis, there exists a critical cycle in $\mathsf{edg}(\Sigma)$ that contains a special edge; let $v_0, v_1, \ldots, v_n$ be such a cycle $(v_0 = v_n)$ with $\lambda(v_i, v_{i+1}) = (\sigma_i, k_i)$, for each $0 \leqslant i < n$. Assuming that the above cycle is one of the shortest cycles in $\mathsf{edg}(\Sigma)$ that contains a special edge, we can show that there exist sequences $I_0 = \{\mathsf{can}((\sigma_0, k_0), \ldots, (\sigma_{n-1}, k_{n-1}))\}, I_1, \ldots$ and $(\rho_0, h_0), (\rho_1, h_1), \ldots$, where, for $k \in \mathbb{N}_0$, $k \cdot n \leqslant i < (k+1) \cdot n$ implies $\rho_i = \sigma_{i-k \cdot n}$, such that:

[2] This additional assumption simply says that the tgd ρ_i is triggered by the atom obtained after applying ρ_{i-1}.

1. for each $i \geqslant 0$, $I_i\langle \rho_i, h_i\rangle I_{i+1}$; and
2. for each $i \neq j \geqslant 0$, $\rho_i = \rho_j$ implies $h_i \neq h_j$.

This immediately implies that Σ admits an infinite o-chase derivation, as needed. The proof for the existence of the above sequences is by induction on $i \geqslant 0$.

Base Step. Let $I_0 = \{\mathsf{can}((\sigma_0, k_0), \ldots, (\sigma_{n-1}, k_{n-1}))\}$. Clearly, there is an homomorphism h_0 such that $h_0(\mathsf{body}(\sigma_0)) \in I_0$, as discussed above. Since $\rho_0 = \sigma_0$, we conclude that (ρ_0, h_0) is a trigger for Σ on I_0, and claim (1) follows. Since $I_0\langle \rho_0, h_0\rangle I_1$ involves only one trigger, claim (2) holds trivially.

Inductive Step. By induction hypothesis, $I_0 = \{\mathsf{can}((\sigma_0, k_0), \ldots, (\sigma_{n-1}, k_{n-1}))\}, \ldots, I_{i+1}$ is a sequence such that for $0 \leq j \leq i$, $I_j\langle \rho_j, h_j\rangle I_{j+1}$ and for each $0 \leq j \neq k \leq i$, $\rho_j = \rho_k$ implies $h_j \neq h_k$. In order to show claim (1), by Lemma 33, it suffices to show that there exists a homomorphism that maps $\mathsf{body}(\rho_{i+1})$ to the atom obtained by applying (τ, g) to $\{\mathsf{can}((\sigma_0, k_0), \ldots, (\sigma_{n-1}, k_{n-1}))\}$, where $\tau = [(\rho_0, j_0), (\rho_1, j_1), \ldots, (\rho_i, j_i)]$ whereas g is the homomorphism mapping $\mathsf{body}(\tau)$ to $\mathsf{can}((\sigma_0, k_0), \ldots, (\sigma_{n-1}, k_{n-1}))$; the $j_i's$ refer to the head-atoms of $\rho_i's$ that appear on the critical cycle. Let θ be the most general unifier of $\mathsf{head}(\tau)$ and $\mathsf{body}(\rho_{i+1})$ (it exists from Lemma 23). Furthermore, let μ be the substitution that maps the variables of $\mathsf{var}(\theta(\mathsf{body}(\rho_{i+1})))$ occurring at positions in Π to the constant c, where Π are the frontier positions of τ such that $\mathsf{term}(\theta(\mathsf{head}(\tau)), \Pi)$ are variables, and μ also maps all the other variables of $\mathsf{var}(\theta(\mathsf{body}(\rho_{i+1})))$ to nulls, according to the atom obtained after the application of (τ, g) to $\{\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})\}$. From the fact that τ is compatible with ρ_{i+1} and from the definition of $\mathsf{can}((\sigma_0, k_0), \ldots, (\sigma_{n-1}, k_{n-1}))$, we obtain that μ is well-defined and $\mu \circ \theta$ is a homomorphism from $\mathsf{body}(\rho_{i+1})$ to the atom obtained by applying (τ, g) to $\{\mathsf{can}(\sigma_0, \ldots, \sigma_{n-1})\}$ and the claim follows.

We proceed to establish claim (2). By induction hypothesis, it suffices to show that, for each $0 \leqslant j \leqslant i$, $\rho_j = \rho_{i+1}$ implies $h_j \neq h_{i+1}$. Assuming that $\rho_{i+1} = \sigma_{(i+1)-k\cdot n}$, for some $k \in \mathbb{N}_0$, we consider the cases where $0 \leqslant j \leqslant k \cdot n$ and $k \cdot n < j \leqslant i$.

Case 1. Assume first that $0 \leqslant j \leqslant k \cdot n$. For each $0 \leqslant j \leqslant k \cdot n$ such that $\rho_j \neq \rho_{i+1}$, the claim follows immediately. Consider now an arbitrary $j \in \{0, \ldots, k \cdot n\}$ such that $\rho_j = \rho_{i+1}$. Observe that in π there exists an edge (u, w) such that $\lambda(u, w) = (\rho_{i+1}, k')$, for some $k' > 0$. Thus, due to the occurrence of a special edge in π — w.l.o.g., we assume that is the first edge of π — we can conclude that h_{i+1} maps the variables in $\mathsf{var}(\mathsf{body}(\rho_{i+1}))$ occurring in u to a null $\perp \in \mathbf{N}$ that was invented during or after the trigger application $I_{k\cdot n}\langle \rho_{k\cdot n}, h_{k\cdot n}\rangle I_{k\cdot n+1}$. Therefore, $\perp$ does not occur in I_j, which in turn implies that h_j maps the variables in $\mathsf{var}(\mathsf{body}(\rho_j))$ occurring in u to a term other than $\perp$. Thus, $h_j \neq h_{i+1}$, and the claim follows.

Case 2. Towards a contradiction, assume that there exists $j \in \{k \cdot n + 1, \ldots, i\}$ such that $\rho_j = \rho_{i+1}$ and $h_j = h_{i+1}$, i.e., $(\rho_j, h_j) = (\rho_{i+1}, h_{i+1})$. Thus, the application of the trigger (ρ_{i+1}, h_{i+1}) can be avoided, and obtain a shorter chase sequence. This implies that in $\mathsf{edg}(\Sigma)$ there exists a cycle that contains a special edge with length less that n. But this contradicts the fact that $v_0, v_1, \ldots, v_n$ is one of the shortest cycles that contains a special edge, and the claim follows.

E Proof of Lemma 22

In order to show the claim, we first need to show an auxiliary lemma. The lemma below essentially states that the resolvent of σ^k, with $k \geq 2$, can be computed as the resolvent of σ and $[\sigma^{k-1}]$.

▶ **Lemma 34.** *If σ is a single-head linear tgd, then for every $k \geq 2$, $[\sigma^k] = [\sigma, [\sigma^{k-1}]]$ (modulo variable renaming).*

Proof. We will prove the statement by induction on k. The proof below relies on Lemma 35, which we prove later on in this proof, stating that for every single-head linear tgd $\bar{\sigma}$, $[\sigma, \bar{\sigma}, \sigma] = [\sigma, [\bar{\sigma}, \sigma]]$, up to variable renaming.

Base Step. If $k = 2$, then $[\sigma, \sigma] = [\sigma, [\sigma]]$ holds trivially. If $k = 3$, then $[\sigma, \sigma, \sigma] = [\sigma, [\sigma, \sigma]]$ holds by Lemma 35.

Inductive Step. Assume that $k \geq 4$. From the definition of resolvent of σ^k, $[\sigma^k] = [[\sigma^{k-1}], \sigma]$. By inductive hypothesis, $[[\sigma^{k-1}], \sigma] = [[\sigma, [\sigma^{k-2}]], \sigma]$. Furthermore, if we let $\bar{\sigma} = [\sigma^{k-2}]$, from the definition of resolvent, $[[\sigma, \bar{\sigma}], \sigma] = [\sigma, \bar{\sigma}, \sigma]$. Finally, by inductive hypothesis again, $[\sigma, \bar{\sigma}, \sigma] = [\sigma, [\bar{\sigma}, \sigma]]$. Since $[\bar{\sigma}, \sigma] = [\sigma^{k-1}]$, by definition of resolvent, we conclude that $[\sigma^k] = [\sigma, [\sigma^{k-1}]]$ and the claim follows.

It now remains to show that the following lemma holds.

▶ **Lemma 35.** *Given a single-head linear tgd $\bar{\sigma}$, it holds that $[\sigma, \bar{\sigma}, \sigma] = [\sigma, [\bar{\sigma}, \sigma]]$ (modulo variable renaming).*

Proof. Before proceeding with the proof, we establish some simple, yet useful properties of equality types and compatible tgds. First of all, note that by definition of equality type, for every two atoms α and α', it holds that $\mathsf{eqtype}(\alpha) = \mathsf{eqtype}(\alpha')$ iff α and α' are the same, up to variable renaming. The above property allows us to establish the following. Let α, α', β and β' be atoms such that $\mathsf{eqtype}(\alpha) = \mathsf{eqtype}(\alpha')$ and $\mathsf{eqtype}(\beta) = \mathsf{eqtype}(\beta')$. Then, α and β unify iff α' and β' unify. Furthermore, $\theta = \mathsf{mgu}(\alpha, \beta)$ and $\theta' = \mathsf{mgu}(\alpha', \beta')$ are the same, up to variable renaming. The above property on equality types allows us to establish the following property on compatible tgds as well. Let σ_1, σ_2 and σ_2' be single-head linear tgds, where $\mathsf{eqtype}(\sigma_2) = \mathsf{eqtype}(\sigma_2')$. Then, σ_1 is compatible with σ_2' iff σ_1 is compatible with σ_2, i.e., $[\sigma_1, \sigma_2'] \neq \Diamond$ iff $[\sigma_1, \sigma_2] \neq \Diamond$. By inspecting the definition of compatibility, we can show that if we assume the weaker $\mathsf{eqtype}(\sigma_2) \subseteq \mathsf{eqtype}(\sigma_2')$, then $[\sigma_1, \sigma_2'] \neq \Diamond \Rightarrow [\sigma_1, \sigma_2] \neq \Diamond$. Finally, let σ_1 and σ_2 be two tgds such that σ_1 is compatible with σ_2. From the definition of resolvent of σ_1 and σ_2, $[\sigma_1, \sigma_2] = \theta(\mathsf{body}(\sigma_1)) \rightarrow \theta(\mathsf{head}(\sigma_2))$, where $\theta = \mathsf{mgu}(\mathsf{head}(\sigma_1), \mathsf{body}(\sigma_2))$. Since θ is a *function* mapping variables to other variables/constants, it follows that $\mathsf{eqtype}(\mathsf{body}(\sigma_1)) \subseteq \mathsf{eqtype}(\theta(\mathsf{body}(\sigma_1))) = \mathsf{eqtype}(\mathsf{body}([\sigma_1, \sigma_2]))$.

We now proceed with the proof. We first show that $[\sigma, \bar{\sigma}, \sigma] \neq \Diamond$ iff $[\sigma, [\bar{\sigma}, \sigma]] \neq \Diamond$.

($\Rightarrow$) Assume that $[\sigma, \bar{\sigma}, \sigma]$ exists. We start by showing that $[\bar{\sigma}, \sigma]$ exists. By definition of resolvent, $[[\sigma, \bar{\sigma}], \sigma]$ exists. This implies that $[\sigma, \bar{\sigma}]$ and σ are compatible. Furthermore, recall that $[\sigma, \bar{\sigma}] = \theta(\mathsf{body}(\sigma)) \rightarrow \theta(\mathsf{head}(\bar{\sigma}))$, where $\theta = \mathsf{mgu}(\mathsf{head}(\sigma), \mathsf{body}(\bar{\sigma}))$. Since θ is a *function* mapping variables to either other variables or constants, we obtain that $\mathsf{eqtype}(\mathsf{head}(\bar{\sigma})) \subseteq \mathsf{eqtype}(\theta(\mathsf{head}(\bar{\sigma})))$. Then, by definition of compatibility, the tgd $\bar{\sigma}$ is compatible with σ as well, which in turn implies that $[\bar{\sigma}, \sigma]$ exists.

We now show that $[\sigma, [\bar{\sigma}, \sigma]]$ exists. Note that the existence of $[\sigma, \bar{\sigma}, \sigma]$ implies that the body and the head of both σ and $\bar{\sigma}$ must have the same relation symbol R. Let $[\bar{\sigma}, \sigma] = \theta(\mathsf{body}(\bar{\sigma})) \rightarrow \theta(\mathsf{head}(\sigma))$, where $\theta = \mathsf{mgu}(\mathsf{head}(\bar{\sigma}), \mathsf{body}(\sigma))$. Towards a contradiction, assume that $[\sigma, [\bar{\sigma}, \sigma]]$ does not exist, i.e. σ is not compatible with $[\bar{\sigma}, \sigma]$. Thus, at least one of the three conditions of Definition 16 is not satisfied.

(*Unification*) Assume first that $\mathsf{head}(\sigma)$ and $\mathsf{body}([\bar{\sigma}, \sigma])$ do not unify, i.e. $\mathsf{head}(\sigma)$ and $\theta(\mathsf{body}(\bar{\sigma}))$ do not unify. Recall that if two atoms α, β unify, then every atom γ such that $\mathsf{eqtype}(\gamma) = \mathsf{eqtype}(\alpha)$ unifies with β as well. Recall also that $\mathsf{head}(\sigma)$ and $\mathsf{body}(\bar{\sigma})$ unify (indeed σ is compatible with $\bar{\sigma}$), so let $\lambda = \mathsf{mgu}(\mathsf{head}(\sigma), \mathsf{body}(\bar{\sigma}))$, and recall that $\mathsf{eqtype}(\mathsf{body}(\bar{\sigma})) \subseteq \mathsf{eqtype}(\theta(\mathsf{body}(\bar{\sigma})))$. Then, $\mathsf{eqtype}(\mathsf{body}(\bar{\sigma})) \subsetneq \mathsf{eqtype}(\theta(\mathsf{body}(\bar{\sigma})))$. This, roughly speaking, means that one element in $\mathsf{eqtype}(\theta(\mathsf{body}(\bar{\sigma}))) \setminus \mathsf{eqtype}(\mathsf{body}(\bar{\sigma}))$ is the "reason" why $\mathsf{head}(\sigma)$

and $\theta(\mathsf{body}(\bar{\sigma}))$ do not unify. We now formalize the above statement, by considering the two only possible reasons why such atoms do not unify.

The first reason of why $\mathsf{head}(\sigma)$ and $\theta(\mathsf{body}(\bar{\sigma}))$ do not unify is the existence of an equality of the form $R[i] = c \in \mathsf{eqtype}(\theta(\mathsf{body}(\bar{\sigma}))) \setminus \mathsf{eqtype}(\mathsf{body}(\bar{\sigma}))$, where c is a constant, satisfying the following: if x is the variable such that $\{R[i]\} = \mathsf{pos}(\mathsf{body}(\bar{\sigma}), x)$. i.e. the variable occuring in $R[i]$ in $\mathsf{body}(\bar{\sigma})$, then $x \mapsto d \in \lambda$, where d is a constant different than c. That is, the variable x in $R[i]$ must unify with d, when $\mathsf{head}(\sigma)$ and $\mathsf{body}(\bar{\sigma})$ unify, but the application of θ to $\mathsf{body}(\bar{\sigma})$ replaces x with the constant c. Recall however that $\theta = \mathsf{mgu}(\mathsf{head}(\bar{\sigma}), \mathsf{body}(\sigma))$, thus x must also occur in $\mathsf{head}(\bar{\sigma})$. But by hypothesis $[\sigma, \bar{\sigma}]$ and σ are compatible, implying that $\lambda(\mathsf{head}(\bar{\sigma}))$ and $\mathsf{body}(\sigma)$ unify. But since $x \mapsto d \in \lambda$, and x is forced to unify with c, we get a contradiction.

The second reason of why $\mathsf{head}(\sigma)$ and $\theta(\mathsf{body}(\bar{\sigma}))$ do not unify is the existence of an equality of the form $R[i] = R[j] \in \mathsf{eqtype}(\theta(\mathsf{body}(\bar{\sigma}))) \setminus \mathsf{eqtype}(\mathsf{body}(\bar{\sigma}))$ satisfying the following: if x and y are the two variables such that $\{R[i]\} = \mathsf{pos}(\mathsf{body}(\bar{\sigma}), x)$ and $\{R[j]\} = \mathsf{pos}(\mathsf{body}(\bar{\sigma}), y)$, i.e. the variables occurring in $R[i]$ and $R[j]$ in $\mathsf{body}(\bar{\sigma})$ respectively, then $x \mapsto c \in \lambda$ and $y \mapsto d \in \lambda$, where $c \neq d$ are constants. In other words, the two variables x and y must unify with c and d respectively, when $\mathsf{head}(\sigma)$ and $\mathsf{body}(\bar{\sigma})$ unify, but the application of θ to $\mathsf{body}(\bar{\sigma})$ replaces x and y with the same variable, say z. Recall however that $\theta = \mathsf{mgu}(\mathsf{head}(\bar{\sigma}), \mathsf{body}(\sigma))$, thus x and y must also occur in $\mathsf{head}(\bar{\sigma})$. But by hypothesis $[\sigma, \bar{\sigma}]$ and σ are compatible, implying that $\lambda(\mathsf{head}(\bar{\sigma}))$ and $\mathsf{body}(\sigma)$ unify. But since $x \mapsto c \in \lambda$, $y \mapsto d \in \lambda$ and x, y are both mapped to z we get a contradiction.

(*Variable compatibility*) Assume that there exists a variable $x \in \mathsf{var}(\theta(\mathsf{body}(\bar{\sigma})))$ such that there are at least two distinct terms $t, u \in \mathsf{term}(\mathsf{head}(\sigma), \Pi_x^{[\bar{\sigma},\sigma]})$, where either at least one of them is an existential variable in σ or both t and u are constants. Since σ is compatible with $\bar{\sigma}$ and since $\mathsf{body}([\bar{\sigma}, \sigma]) = \theta(\mathsf{body}(\bar{\sigma}))$, there must exist two positions $R[i], R[j] \in \Pi_x^{[\bar{\sigma},\sigma]}$ such that $R[i] = R[j] \in \mathsf{eqtype}(\theta(\mathsf{body}(\bar{\sigma}))) \setminus \mathsf{eqtype}(\mathsf{body}(\bar{\sigma}))$. That is, the variables occurring in $R[i]$ and $R[j]$ in $\mathsf{body}(\bar{\sigma})$ are unified by θ. Let x and y be the two variables in $R[i]$ and $R[j]$ respectively, in $\mathsf{body}(\bar{\sigma})$, that is x and y are such that $\{R[i]\} = \mathsf{pos}(\mathsf{body}(\bar{\sigma}), x)$ and $\{R[j]\} = \mathsf{pos}(\mathsf{body}(\bar{\sigma}), y)$. Since $\lambda = \mathsf{mgu}(\mathsf{head}(\sigma), \mathsf{body}(\bar{\sigma}))$, $x \mapsto t \in \lambda$ and $y \mapsto u \in \lambda$. That is, x and y are unified with t and u respectively. However, recall that $\theta = \mathsf{mgu}(\mathsf{head}(\bar{\sigma}), \mathsf{body}(\sigma))$, thus x and y must also occur in $\mathsf{head}(\bar{\sigma})$. By hypothesis $[\sigma, \bar{\sigma}]$ and σ are compatible, implying that $\lambda(\mathsf{head}(\bar{\sigma}))$ and $\mathsf{body}(\sigma)$ unify. But since $x \mapsto t \in \lambda$, $y \mapsto u \in \lambda$ and x, y are both mapped to the same variable by θ, $[\sigma, \bar{\sigma}]$ cannot be compatible with σ, and we obtain a contradiction.

(*Constant compatibility*) Assume there exists a constant $c \in \mathsf{const}(\theta(\mathsf{body}(\bar{\sigma})))$ such that it holds that $\mathsf{term}(\mathsf{head}(\sigma), \Pi_c^{[\bar{\sigma},\sigma]}) \not\subseteq \mathsf{fr}(\sigma) \cup \{c\}$. That is, there exists a term $t \in \mathsf{term}(\mathsf{head}(\sigma), \Pi_c^{[\bar{\sigma},\sigma]})$ which is either an existential variable of σ or a constant different than c. Since σ is compatible with $\bar{\sigma}$ and since $\mathsf{body}([\bar{\sigma}, \sigma]) = \theta(\mathsf{body}(\bar{\sigma}))$, there must exist a position $R[i] \in \Pi_c^{[\bar{\sigma},\sigma]}$ such that $R[i] = c \in \mathsf{eqtype}(\theta(\mathsf{body}(\bar{\sigma}))) \setminus \mathsf{eqtype}(\mathsf{body}(\bar{\sigma}))$. That is, the variable occurring in $R[i]$ in $\mathsf{body}(\bar{\sigma})$ is unified with c by θ. Let x be such a variable, then $x \mapsto c \in \theta$. Since $\theta = \mathsf{mgu}(\mathsf{head}(\bar{\sigma}), \mathsf{body}(\sigma))$, the variable x must also occur in $\mathsf{head}(\bar{\sigma})$. Furthermore, since σ is compatible with $\bar{\sigma}$, there is a most general unifier $\lambda = \mathsf{mgu}(\mathsf{head}(\sigma), \mathsf{body}(\bar{\sigma}))$ which unifies the term t with x. This implies that t will appear in $\mathsf{head}([\sigma, \bar{\sigma}]) = \lambda(\mathsf{head}(\bar{\sigma}))$ at the same position as x either as an existential variable of $[\bar{\sigma}, \sigma]$ (if t is an existential variable of σ) or as a constant different than c (if t is a constant in $\mathsf{head}(\sigma)$). However, x is mapped to c by θ, hence $[\sigma, \bar{\sigma}]$ is not compatible with σ, and we obtain a contradiction. This ends the proof of the existence of $[\sigma, [\bar{\sigma}, \sigma]]$.

($\Leftarrow$) The other direction of the implication requires a proof which is analogous to the one given above, and thus it is omitted.

It now remains to show that when $[\sigma, \bar{\sigma}, \sigma] \neq \Diamond$ and $[\sigma, [\bar{\sigma}, \sigma]] \neq \Diamond$, then $[\sigma, \bar{\sigma}, \sigma]$ and $[\sigma, [\bar{\sigma}, \sigma]]$ are the same, up to variable renaming. We start by proving that the following equalities hold:

1. $\mathsf{eqtype}(\mathsf{body}([\sigma, \bar{\sigma}, \sigma])) = \mathsf{eqtype}(\mathsf{body}([\sigma, [\bar{\sigma}, \sigma]]))$;
2. $\mathsf{eqtype}(\mathsf{head}([\sigma, \bar{\sigma}, \sigma])) = \mathsf{eqtype}(\mathsf{head}([\sigma, [\bar{\sigma}, \sigma]]))$.

In what follows, let

- $\lambda = \mathsf{mgu}(\mathsf{head}(\sigma), \mathsf{body}(\bar{\sigma}))$,
- $\lambda' = \mathsf{mgu}(\lambda(\mathsf{head}(\bar{\sigma})), \mathsf{body}(\sigma))$,
- $\theta = \mathsf{mgu}(\mathsf{head}(\bar{\sigma}), \mathsf{body}(\sigma))$,
- $\theta' = \mathsf{mgu}(\mathsf{head}(\sigma), \theta(\mathsf{body}(\bar{\sigma})))$.

Furthermore, recall that

- $\mathsf{eqtype}(\mathsf{body}([\sigma, \bar{\sigma}, \sigma])) = \mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma))))$,
- $\mathsf{eqtype}(\mathsf{body}([\sigma, [\bar{\sigma}, \sigma]])) = \mathsf{eqtype}(\theta'(\mathsf{body}(\sigma)))$,
- $\mathsf{eqtype}(\mathsf{head}([\sigma, \bar{\sigma}, \sigma])) = \mathsf{eqtype}(\lambda'(\mathsf{head}(\sigma)))$,
- $\mathsf{eqtype}(\mathsf{head}([\sigma, [\bar{\sigma}, \sigma]])) = \mathsf{eqtype}(\theta'(\theta(\mathsf{head}(\sigma))))$.

(*Item 1*) We first show that $\mathsf{eqtype}(\mathsf{body}([\sigma, \bar{\sigma}, \sigma])) \supseteq \mathsf{eqtype}(\mathsf{body}([\sigma, [\bar{\sigma}, \sigma]]))$, which is equivalent to show that $\mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma)))) \supseteq \mathsf{eqtype}(\theta'(\mathsf{body}(\sigma)))$. We show that for every equality e, $e \in \mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma))))$ implies $e \in \mathsf{eqtype}(\theta'(\mathsf{body}(\sigma)))$. Note that the previous implication is equivalent to say that for every equality e, $e \in \mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma))))$ and $e \notin \mathsf{eqtype}(\theta'(\mathsf{body}(\sigma)))$ implies $e \in \mathsf{eqtype}(\theta'(\mathsf{body}(\sigma)))$. Assume then that there exists an element in $\mathsf{eqtype}(\theta'(\mathsf{body}(\sigma)))$ which is not in $\mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma))))$. For the sake of presentation, assume that such an element is of the form $R[i] = R[j]$ and that $R[i] = R[j] \notin \mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma))))$ because two different variables appear in $R[i]$ and $R[j]$ in $\lambda'(\lambda(\mathsf{body}(\sigma)))$. We can provide a similar argument to the one given below for the case where only the term occurring in $R[i]$ is a variable and the case where such an element is of the form $R[i] = c$, where c is a constant. If $R[i] = R[j] \notin \mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma))))$, it means that also $R[i] = R[j] \notin \mathsf{eqtype}(\mathsf{body}(\sigma))$ (since $\mathsf{eqtype}(\mathsf{body}(\sigma)) \subseteq \mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma)))))$. Thus, let $R[k]$ and $R[l]$ be the positions in $\mathsf{head}(\sigma)$ in which occur the variables also occurring in $R[i]$ and $R[j]$ in $\mathsf{body}(\sigma)$. In other words, $\{R[k]\} = \mathsf{pos}(\mathsf{head}(\sigma), x)$, where x is the variable such that $\{R[i]\} = \mathsf{pos}(\mathsf{body}(\sigma), x)$, and $\{R[l]\} = \mathsf{pos}(\mathsf{head}(\sigma), y)$, where y is the variable such that $\{R[j]\} = \mathsf{pos}(\mathsf{body}(\sigma), y)$. Note that by construction, $R[k] = R[l] \notin \mathsf{eqtype}(\mathsf{head}(\sigma))$. Also note that since $R[i] = R[j] \notin \mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma))))$, it means that $R[k] = R[l] \notin \mathsf{eqtype}(\lambda(\mathsf{head}(\sigma)))$. Thus, from the fact that λ is the most general unifier of $\mathsf{head}(\sigma)$ and $\mathsf{body}(\bar{\sigma})$, the fact that $R[k] = R[l] \notin \mathsf{eqtype}(\lambda(\mathsf{head}(\sigma)))$ and the fact that $R[k] = R[l] \notin \mathsf{eqtype}(\mathsf{head}(\sigma))$, it must be the case that $R[k] = R[l] \notin \mathsf{eqtype}(\mathsf{body}(\bar{\sigma}))$. Note that $R[k] = R[l] \notin \mathsf{eqtype}(\mathsf{body}(\bar{\sigma}))$ implies that $R[m] = R[n] \notin \mathsf{eqtype}(\mathsf{head}(\bar{\sigma}))$, where $R[m]$ is the position in $\mathsf{head}(\bar{\sigma})$ containing the variable occurring in $R[k]$ in $\mathsf{body}(\bar{\sigma})$ and $R[n]$ is the position in $\mathsf{head}(\bar{\sigma})$ containing the variable occurring in $R[l]$ in $\mathsf{body}(\bar{\sigma})$. This and the fact that by assumption, $R[i] = R[j] \in \mathsf{eqtype}(\theta'(\mathsf{body}(\sigma)))$, the fact that $R[k] = R[l] \notin \mathsf{eqtype}(\mathsf{head}(\sigma))$, and the fact that $R[k] = R[l] \notin \mathsf{body}(\bar{\sigma})$ allow us to conclude that $R[m] = R[n] \in \mathsf{eqtype}(\theta(\mathsf{head}(\bar{\sigma})))$. Furthermore, since $R[m] = R[n] \in \mathsf{eqtype}(\theta(\mathsf{head}(\bar{\sigma})))$, where $\theta = \mathsf{mgu}(\mathsf{head}(\bar{\sigma}), \mathsf{body}(\sigma))$, since λ' is the most general unifier of $\lambda(\mathsf{head}(\bar{\sigma}))$ and $\mathsf{body}(\sigma)$, and from the fact that $\mathsf{eqtype}(\mathsf{head}(\bar{\sigma})) \subseteq \mathsf{eqtype}(\lambda(\mathsf{head}(\bar{\sigma})))$, we obtain that $R[m] = R[n] \in \mathsf{eqtype}(\lambda'(\lambda(\mathsf{head}(\bar{\sigma}))))$. This, by definition of $R[n]$ and $R[m]$, finally implies that $R[i] = R[j] \in \mathsf{eqtype}(\lambda'(\lambda(\mathsf{body}(\sigma))))$.

The proof for showing that $\mathsf{eqtype}(\mathsf{body}([\sigma, \bar{\sigma}, \sigma])) \subseteq \mathsf{eqtype}(\mathsf{body}([\sigma, [\bar{\sigma}, \sigma]]))$ proceeds in exactly the same way as the one given above, and for this reason it is omitted.

(*Item 2*) This item can be proved with a discussion which is symmetric to the one proposed for Item 1, where the equality types of the head atoms of $[\sigma, \bar{\sigma}, \sigma]$ and $[\sigma, [\bar{\sigma}, \sigma]]$ are considered instead.

We now show that indeed $[\sigma, \bar{\sigma}, \sigma] = [\sigma, [\bar{\sigma}, \sigma]]$, up to variable renaming. To this end, given a tgd σ', we define the following binary relation among positions of $\mathsf{body}(\sigma')$ and positions of $\mathsf{head}(\sigma')$. In particular, if $\sigma' = P(t_1, \ldots, t_n) \rightarrow \exists \bar{z} S(u_1, \ldots, u_m)$, for every $1 \leq i \leq n$ and $1 \leq j \leq m$, we say that $P[i] \rightarrow_{\sigma'} S[j]$ holds, if there is a variable x such that $\{P[i]\} = \mathsf{pos}(\mathsf{body}(\sigma'), x)$ and $\{S[j]\} = \mathsf{pos}(\mathsf{head}(\sigma'), x)$. In other words, x occurs in both $P[i]$ and $S[j]$, in $\mathsf{body}(\sigma')$ and $\mathsf{head}(\sigma')$, respectively. Since we have already shown that $\mathsf{eqtype}(\mathsf{body}([\sigma, \bar{\sigma}, \sigma])) = \mathsf{eqtype}(\mathsf{body}([\sigma, [\bar{\sigma}, \sigma]]))$ and $\mathsf{eqtype}(\mathsf{head}([\sigma, \bar{\sigma}, \sigma])) = \mathsf{eqtype}(\mathsf{head}([\sigma, [\bar{\sigma}, \sigma]]))$, if we are able to show also that for every two positions $R[i]$ and $R[j]$ in $\mathsf{body}(\sigma)$ and $\mathsf{head}(\sigma)$ respectively, $R[i] \rightarrow_{[\sigma, \bar{\sigma}, \sigma]} R[j]$ iff $R[i] \rightarrow_{[\sigma, [\bar{\sigma}, \sigma]]} R[j]$, the claim will immediately follow.

From the definition of resolvent of two tgds, it is not difficult to see that for every two linear, single-head tgds σ_1, σ_2, such that the resolvent $[\sigma_1, \sigma_2]$ exists, for every position $P[i]$ of $\mathsf{body}(\sigma_1)$, and position $T[k]$ of $\mathsf{head}(\sigma_2)$, $P[i] \rightarrow_{[\sigma_1, \sigma_2]} T[k]$ iff there exists a position $S[j]$ of $\mathsf{head}(\sigma_1)$ and $\mathsf{body}(\sigma_2)$ such that $P[i] \rightarrow_{\sigma_1} S[j]$ and $S[j] \rightarrow_{\sigma_2} T[k]$. In other words, the variable occurring in $P[i]$ in $\mathsf{body}(\sigma_1)$ is propagated to $T[k]$ during the resolving process. Now, we start by showing that $R[i] \rightarrow_{[\sigma, \bar{\sigma}, \sigma]} R[j]$ implies $R[i] \rightarrow_{[\sigma, [\bar{\sigma}, \sigma]]} R[j]$. Assume $R[i] \rightarrow_{[\sigma, \bar{\sigma}, \sigma]} R[j]$. From the definition of resolvent, $[\sigma, \bar{\sigma}, \sigma] = [[\sigma, \bar{\sigma}], \sigma]$. Furthermore, since $R[i] \rightarrow_{[\sigma, \bar{\sigma}, \sigma]} R[j]$, then there must exist a position $R[k]$ in $\mathsf{head}([\sigma, \bar{\sigma}])$ and $\mathsf{body}(\sigma)$ such that $R[i] \rightarrow_{[\sigma, \bar{\sigma}]} R[k]$ and $R[k] \rightarrow_{\sigma} R[j]$. Similarly, there must be a position $R[l]$ in $\mathsf{head}(\sigma)$ and $\mathsf{body}(\bar{\sigma})$ such that $R[i] \rightarrow_{\sigma} R[l]$ and $R[l] \rightarrow_{\bar{\sigma}} R[k]$. So, we have obtained that $R[i] \rightarrow_{\sigma} R[l]$, $R[l] \rightarrow_{\bar{\sigma}} R[k]$, $R[k] \rightarrow_{\sigma} R[j]$. But then, from the last two expressions, we conclude that $R[l] \rightarrow_{[\bar{\sigma}, \sigma]} R[j]$. Then, from $R[i] \rightarrow_{\sigma} R[l]$ and $R[l] \rightarrow_{[\bar{\sigma}, \sigma]} R[j]$, we finally conclude that $R[i] \rightarrow_{[\sigma, [\bar{\sigma}, \sigma]]} R[j]$. With a similar reasoning, we can prove that $R[i] \rightarrow_{[\sigma, [\bar{\sigma}, \sigma]]} R[j]$ implies $R[i] \rightarrow_{[\sigma, \bar{\sigma}, \sigma]} R[j]$, and the claim follows. ☐

This ends the proof of Lemma 34. ☐

We now proceed with the proof of Lemma 22. Recall that for every two atoms α and β, $\mathsf{eqtype}(\alpha) = \mathsf{eqtype}(\beta)$ iff α and β are the same, modulo variable renaming. Thus, for every two tgds σ_1 and σ_2, $[\sigma_1, \sigma_2] \neq \Diamond$ implies $[\sigma_1, \sigma_2'] \neq \Diamond$, for every tgd σ_2' with $\mathsf{eqtype}(\mathsf{body}(\sigma_2')) = \mathsf{eqtype}(\mathsf{body}(\sigma_2))$. Let σ be a single-head linear tgd such that σ^i is active, for some $i > 1$, and $\mathsf{eqtype}([\sigma^{i-1}]) = \mathsf{eqtype}([\sigma^i])$. Note that $[\sigma^i] = [\sigma, [\sigma^{i-1}]]$, from Lemma 34, which means that σ is compatible with $[\sigma^{i-1}]$. Furthermore, $\mathsf{eqtype}([\sigma^{i-1}]) = \mathsf{eqtype}([\sigma^i])$. Thus, σ is compatible with $[\sigma^i]$, i.e. $[\sigma, [\sigma^i]]$ exists. From Lemma 34, $[\sigma, [\sigma^i]] = [\sigma^{i+1}]$ and the claim follows. What is remaining to show is that indeed $\mathsf{eqtype}([\sigma^{i+1}]) = \mathsf{eqtype}([\sigma^i])$. Since $\mathsf{eqtype}([\sigma^{i-1}]) = \mathsf{eqtype}([\sigma^i])$, the two most general unifiers $\theta = \mathsf{mgu}(\mathsf{head}(\sigma), \mathsf{body}([\sigma^{i-1}]))$ and $\gamma = \mathsf{mgu}(\mathsf{head}(\sigma), \mathsf{body}([\sigma^i]))$ are the same, up to variable renaming. In particular, their restriction to the variables $X = \mathsf{var}(\mathsf{head}(\sigma))$, i.e., $\theta_{|X}$ and $\gamma_{|X}$ are the same, up to variable renaming. From the definition of resolvent and from Lemma 34, we know that $\mathsf{body}([\sigma^{i+1}]) = \gamma(\mathsf{body}(\sigma))$ and $\mathsf{body}([\sigma^i]) = \theta(\mathsf{body}(\sigma))$. Since $\theta_{|X}$ and $\gamma_{|X}$ are the same, up to variable renaming, we conclude that $\mathsf{eqtype}(\mathsf{body}([\sigma^{i+1}])) = \mathsf{eqtype}(\mathsf{body}([\sigma^i]))$. This ends our proof.

F Proof of Lemma 23

Let $\sigma_1, \ldots, \sigma_n$ be a critical sequence of linear, single-head tgds. That is, the resolvent $\rho = [\sigma_1, \ldots, \sigma_n]$ exists and the sequence $\rho^{\omega+1}$ is active, where ω is the arity of the predicate of $\mathsf{body}(\sigma_1)$. Note that since $\rho^{\omega+1}$ is active, for each $1 \leq i \leq \omega + 1$, ρ^i is active, i.e. the resolvent $[\rho^i]$ exists. We start by showing that for every $1 < i \leq \omega + 1$, $\mathsf{eqtype}(\mathsf{body}([\rho^{i-1}])) \subseteq \mathsf{eqtype}(\mathsf{body}([\rho^i]))$. By definition of resolvent, $[\rho^i] = [[\rho^{i-1}], \rho] = \gamma(\mathsf{body}([\rho^{i-1}])) \rightarrow \gamma(\mathsf{head}(\rho))$, where γ is the most general unifier $\gamma = \mathsf{mgu}(\mathsf{head}([\rho^{i-1}]), \mathsf{body}(\rho))$. Since γ is a *function*, mapping variables to either other variables or constants, two occurrences of the same variable in $\mathsf{body}([\rho^{i-1}])$ cannot be mapped to distinct terms, by γ. Thus, $\mathsf{eqtype}(\mathsf{body}([\rho^{i-1}])) \subseteq \mathsf{eqtype}(\gamma(\mathsf{body}([\rho^{i-1}]))) = \mathsf{eqtype}(\mathsf{body}([\rho^{i}]))$. We now show that there exists $1 < i \leq \omega+1$ such that $\mathsf{eqtype}(\mathsf{body}([\rho^{i-1}])) = \mathsf{eqtype}(\mathsf{body}([\rho^{i}]))$. We distinguish two cases: either exactly ω distinct variables occur in $\mathsf{body}(\rho)$, which also means that no constants appear in $\mathsf{body}(\rho)$, or the number of distinct variables occurring in $\mathsf{body}(\rho)$ is strictly less than ω. In the latter case, constants may or may not appear in $\mathsf{body}(\rho)$. Assume first that $|\mathsf{var}(\mathsf{body}(\rho))| = \omega$. Trivially, $\mathsf{eqtype}(\mathsf{body}([\rho^{i-1}])) = \mathsf{eqtype}(\mathsf{body}([\rho^{i}]))$, for $1 < i \leq \omega + 1$, as no constants appear in $\mathsf{body}(\rho)$. Assume instead that $|\mathsf{var}(\mathsf{body}(\rho))| < \omega$. Then, let us consider all the indices $1 < j \leq \omega+1$ such that $\mathsf{eqtype}(\mathsf{body}([\rho^{j-1}])) \neq \mathsf{eqtype}(\mathsf{body}([\rho^{j}]))$. As discussed before, $\mathsf{eqtype}(\mathsf{body}([\rho^{j-1}])) \subseteq \mathsf{eqtype}(\mathsf{body}([\rho^{j}]))$, thus $\mathsf{eqtype}(\mathsf{body}([\rho^{j-1}])) \subsetneq \mathsf{eqtype}(\mathsf{body}([\rho^{j}]))$. This means that at least one variable x, occurring in some position of $\mathsf{body}([\rho^{j-1}])$ changed. Furthermore, since the equality type changed, only one of following two cases must hold: either x becomes equal to some other variable in $\mathsf{body}([\rho^{i-1}])$, or x becomes a constant. Each variable can change in only one of the aforementioned ways. Furthermore, there are at most $\omega - 1$ distinct variables in $\mathsf{body}(\rho)$ thus the equality type can change at most $\omega - 1$ times. Thus, it holds that $\mathsf{eqtype}(\mathsf{body}([\rho^{i-1}])) \subsetneq \mathsf{eqtype}(\mathsf{body}([\rho^{i}]))$ only if $1 < i \leq \omega$. Since $\rho^{\omega+1}$ is active, we obtain that $\mathsf{eqtype}(\mathsf{body}([\rho^{\omega}])) = \mathsf{eqtype}(\mathsf{body}([\rho^{\omega+1}]))$. From Lemma 22, $\mathsf{eqtype}(\mathsf{body}([\rho^{\omega}])) = \mathsf{eqtype}(\mathsf{body}([\rho^{\omega+1}]))$ implies that $\rho^{\omega+2}$ is active and $\mathsf{eqtype}(\mathsf{body}([\rho^{\omega+1}])) = \mathsf{eqtype}(\mathsf{body}([\rho^{\omega+2}]))$. By recursively applying Lemma 22, we conclude that for every $k > 0$, $[\rho^{k}]$ exists, i.e. ρ^{k} is active, ending our proof.

G Proof of Theorem 25

We prove the upper-bounds first. In particular, we focus on the complement of our problem. That is, given a set of sticky tgds Σ, check whether $\Sigma \notin \mathbb{CT}^{\star}_{\forall}$, for $\star \in \{\mathsf{o}, \mathsf{so}\}$.

G.1 $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ is in PSpace and in NLogSpace for predicates of bounded arity

Consider a set of tgds $\Sigma \in \mathbb{S}$. We show that checking whether $\Sigma \notin \mathbb{CT}^{\star}_{\forall}$ is in NSPACE$(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$, where ω is the maximum arity of predicates in Σ and m is the maximum number of atoms occuring in a tgd in Σ. This allows us to uniformly prove the claim for both the general case and the bounded arity case. To this end, by Theorem 13 and by definition of critical-rich-acyclicity (resp., critical-weak-acyclicity), we need to show that the problem of deciding whether a critical cycle in $\mathsf{edg}(\mathsf{Lin}(\Sigma))$ (resp., $\mathsf{dg}(\mathsf{Lin}(\Sigma))$) that contains a special edge exists is in NSPACE$(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$. In what follows, let $G = (N, E, \lambda)$ be either $\mathsf{edg}(\mathsf{Lin}(\Sigma))$ or $\mathsf{dg}(\mathsf{Lin}(\Sigma))$.

The problem under consideration can be conceived as an extended version of graph reachability. More precisely, we need to decide whether there exists a node $v \in N$ that is reachable from itself via a cycle $\pi = v, v_1, v_2, \ldots, v_{n-1}, v$, and the following hold: (i) π is critical, or, equivalently, $\lambda(v, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{n-1}, v)$ is critical; and (ii) (v, v_1) is special, or (v_{n-1}, v) is special, or (v_i, v_{i+1}) is special, for some $i \in [n-2]$. This can be achieved by applying the following non-deterministic procedure:

1. Guess an edge $e_1 = (v_1, v_2) \in E$.
2. If e_1 is special, then $\mathit{flag} := 1$; otherwise, $\mathit{flag} := 0$.
3. $\sigma_1 := \lambda(e_1)$ and $\mathit{origin} := v_1$.
4. Repeat

a. If there is no edge $(u, w) \in E$ such that $u = v_2$, then $reject$; otherwise, guess an edge $e_2 = (v_2, v_3) \in E$.

b. If e_2 is special, then $flag := 1$.

c. $\sigma_2 := \lambda(e_2)$.

d. If σ_1 is not compatible with σ_2, then $reject$; otherwise, $e_1 = (v_1, v_2) := e_2 = (v_2, v_3)$ and $\sigma_1 := [\sigma_1, \sigma_2]$.

Until ($v_3 = origin$).

5. If $flag = 0$, then $reject$.

6. If $[\underbrace{\sigma_1, \ldots, \sigma_1}_{k}] \neq \bot$, for each $k \in [\omega + 1]$, then $accept$; otherwise, $reject$.

It is not difficult to verify that the above procedure is correct. In fact, the repeat-until statement seeks for an active cycle π in G, and if it exists, the resolvent of the tgds that label the edges of π is stored in σ_1. If such an active cycle does not exist, then the algorithm rejects. Finally, the algorithm returns $accept$ iff π contains a special edge (i.e., if $flag = 1$), and π is critical (i.e., the resolvent of the sequence $\sigma_1, \ldots, \sigma_1$ of length k, for each $k \in [\omega + 1]$, exists). The rest of the proof is devoted to show that the above nondeterministic procedure runs in space $O(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot |\Sigma|))$.

First, observe that encoding a position of $\mathsf{sch}(\Sigma)$ requires $\log(\omega \cdot |\mathsf{sch}(\Sigma)|)$ space, encoding a predicate of $\mathsf{sch}(\Sigma)$ requires $\log(|\mathsf{sch}(\Sigma)|)$ space, and encoding a variable/constant occurring in Σ requires $O(\log(\omega \cdot m \cdot |\Sigma|))$ space – we assume that the tgds of Σ do not share variables, and thus $O(\omega \cdot m \cdot |\Sigma|)$ variables may occur in Σ. Finally, encoding an atom, requires $O(\log(|\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space. We now proceed by discussing the space needed by each step of the procedure above. In particular, we provide the space needed by the initialization steps 1,2 and 3, and the space needed by a single iteration of the Repeat-Until loop, as the space used at one iteration can be reused for the next one.

Step 1.. We now show what is the space required to guess the edge $e_1 = (v_1, v_2) \in E$ (actually, any edge in E). In particular, we point out that constructing the set $\mathsf{Lin}(\Sigma)$ explicitly, in order to later construct G, might require exponential space. We then show how to guess an edge of E without explicitly constructing $\mathsf{Lin}(\Sigma)$ and G. First guess a tgd $\sigma \in \Sigma$, which means storing a pointer to σ in $O(\log(|\Sigma|))$ space. Then, guess an atom α and an atom β, from the body and head of σ, respectively. The latter requires to store two atoms in $O(\log(|\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space. We now need to guess an homomorphism $h \in M^{\Sigma}_{\alpha,\sigma}$. In general, each homomorphism in $M^{\Sigma}_{\alpha,\sigma}$ contains at most $\omega \cdot m$ pairs of the form $x \mapsto t$, where x is a variable in σ and t is a constant in $\mathsf{dom}(\mathsf{cr}(\Sigma))$. However, for the purpose of the algorithm, we need to focus only on the restriction of such h's to the variables in $X_{\alpha,\beta} = \mathsf{var}(\alpha) \cup \mathsf{var}(\beta)$, which are at most 2ω. Thus, we guess an homomorphism h' such that $\exists h \in M^{\Sigma}_{\alpha,\sigma}$, where $h' = h_{|X_{\alpha,\beta}}$. As discussed, there are at most 2ω pairs of the form $x \mapsto t$ in h'. Furthermore, each variable x and constant t can be stored in $O(\log(\omega \cdot m \cdot |\Sigma|))$ space. Thus, guessing h' requires $O(\omega \log(\omega \cdot m \cdot |\Sigma|))$ space. Then, we apply h' on α and β in place, without using any additional space. The above procedure implies that $h'(\alpha) \rightarrow h'(\beta)$ is the label of some edge in E and that if there exists an edge in E labeled with a single-head linear tgd, the above procedure is able to guess such a tgd, if needed. Now, the procedure is ready to guess one of the edges labeled by $h'(\alpha) \rightarrow h'(\beta)$, by guessing one position v_1 in $h'(\alpha)$ and one position v_2 in $h'(\beta)$. Both positions can be stored in $O(\log(\omega \cdot |\mathsf{sch}(\Sigma)|))$ space. Then, we need to check whether $(v_1, v_2) \in E$, which means checking whether v_1 contains a variable x in $h'(\alpha)$ and v_2 contains a variable y in $h'(\beta)$. In case $x \neq y$, check whether y is existential (if $G = \mathsf{dg}(\mathsf{Lin}(\Sigma))$, we also need to check that x occurs in $h'(\beta)$). Clearly, the above check is feasible in constant space. Therefore, we can guess and maintain an edge of G in $O(\log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space.

Step 2. Given an edge $(v_1, v_2) \in E$, from the construction above, checking whether an edge is special, requires checking whether the variable occurring in v_2 is existential, which is feasible in constant space.

Step 3. Constructing $\sigma_1 = \lambda(e_1)$ does not require any space, as the tgd $h'(\alpha) \rightarrow h'(\beta)$ constructed at step 1 is indeed $\lambda(e_1)$. Storing *origin* $:= v_1$ just requires copying v_1, which requires $O(\log(\omega \cdot |\mathsf{sch}(\Sigma)|))$ space.

Steps 4.a, 4.b, 4.c. Checking whether there is no edge $(u, v) \in E$ such that $u = v_2$ requires the guessing of (u, v) which, as already discussed, is feasible in $O(\log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space. The same applies for $e_2 = (v_2, v_3) \in E$, which will ovewrite (u, v). As discussed above, checking whether e_2 is special is feasible in constant space and constructing σ_2 is also constant space, as it's been already constructed at step 4.a.

Step 4.d. To check whether σ_1 is compatible with σ_2, we need to check first that $\mathsf{head}(\sigma_1)$ and $\mathsf{body}(\sigma_2)$ unify, and if this is the case, compute the most general unifier $\theta = \mathsf{mgu}(\mathsf{head}(\sigma_1), \mathsf{body}(\sigma_2))$. To this end, we provide a simplified version of Robinson's unification algorithm. The main difference with the original algorithm is in the fact that the algorithm below does not consider function symbols and nested terms. Assuming that $\mathsf{head}(\sigma_1) = R(t_1, \ldots, t_n)$ and $\mathsf{body}(\sigma_2) = R(u_1, \ldots, u_n)$, the unification algorithm constructs the most general unifier θ as follows:

1. $\theta := \big(\{t_i \rightarrow t_i\}_{i \in [n]} \cup \{u_i \rightarrow u_i\}_{i \in [n]}\big)$.
2. $\mathit{ctr} := 1$.
3. Repeat
   - a. $t := \theta(t_{\mathit{ctr}})$ and $u := \theta(u_{\mathit{ctr}})$.
   - b. If t is a variable, then $\theta := \{t \mapsto u\} \circ \theta$.
   - c. Else if u is a variable, then $\theta := \{u \mapsto t\} \circ \theta$.
   - d. Else If $t \neq u$, then **fail**.
   - e. $\mathit{ctr} := \mathit{ctr} + 1$.

   Until $(\mathit{ctr} = n + 1)$.
4. Return θ.

The above algorithm runs in $O(\omega \log(\omega \cdot m \cdot |\Sigma|))$ space, i.e., the space needed to maintain θ, t and u. Consequently, we can check whether $\mathsf{head}(\sigma_1)$ and $\mathsf{body}(\sigma_2)$ unify in $O(\omega \log(\omega \cdot m \cdot |\Sigma|))$ space, in which case the unifier is constructed, needing $O(\omega \log(\omega \cdot m \cdot |\Sigma|))$ space. Checking whether condition 2 of Definition 16 of compatibility holds, requires iterating over each variable $x \in \mathsf{var}(\mathsf{body}(\sigma_2))$, store $\Pi_x^{\sigma_2}$ in $O(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|))$ space and check whether either $\mathsf{term}(\mathsf{head}(\sigma_1), \Pi_x^{\sigma_2}) = \{z\}$, for some existential variable z of σ_1, or $\mathsf{term}(\mathsf{head}(\sigma_1), \Pi_x^{\sigma_2}) \subseteq \mathsf{fr}(\sigma_1) \cup \{c\}$, for some constant c. The aforementoned two checks do not require any additional space. Similarly, we can check condition 3 of Definition 16 in space $O(\omega \log(\omega \cdot \mathsf{sch}(|\Sigma|)))$. Storing the edges e_1 and e_2 does not require any additional space, as the space for e_1 and e_2 has been already allocated. Constructing $[\sigma_1, \sigma_2]$ simply requires the application of the most general unifier θ to $\mathsf{body}(\sigma_1)$ and $\mathsf{head}(\sigma_2)$. Overall, this step requires $O(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space.

Steps 5 and 6. Step 5 requires constant space. By providing a similar analysis to the one given for step 4.d, we can show that the criticality check can be done using $O(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space. Summing up, we get that the above nondeterministic procedure runs in $O(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space, as needed. This completes our proof.

G.2 $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ is PSPACE-hard

To show that $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ is PSPACE-hard, it suffices to show that the complement of our problem is PSPACE-hard. The proof is by reduction from the acceptance problem of a deterministic polynomial space Turing machine M on an input $I = a_1 \ldots a_m$. Let $M = (S, \Lambda, \delta, s_1)$, where S is a finite set of states, $\Lambda = \{0, 1, \sqcup\}$ is the tape alphabet with $\sqcup$ be the blank symbol, $\delta : S \times \Lambda \to (S \times \Lambda \times \{\leftarrow, -, \rightarrow\})$ is the transition function, and $s_1 \in S$ is the initial state. We assume that M is well-behaved and never tries to read beyond its tape boundaries, always halts, and uses exactly $n = m^k$ tape cells, where $k > 0$. We represent configurations using a subset of the strings in $S\ (\Lambda\ \{\uparrow, \natural\})^+$, i.e., the state of the configuration is placed at the beginning of the string and the tape of the machine is encoded by a sequence of the form $b_1, \mathsf{cur}_1, b_2, \mathsf{cur}_2, \ldots, b_n, \mathsf{cur}_n$, where each b_i is the value at the i-th cell and $\mathsf{cur}_i \in \{\uparrow, \natural\}$ denotes whether the cursor is on the i-th cell ($\uparrow$) or not ($\natural$). In this notation, the initial configuration is $s_1, a_1, \uparrow, a_2, \natural, \ldots a_m, \natural, (\sqcup, \natural)^{n-m}$. We assume that $S = \{s_1, \ldots, s_{|S|}\}$ and that the machine accepts its input if it reaches a configuration with state s_2.

Our goal is to construct a set of constant-free sticky tgds Σ such that the machine M accepts on input I iff $\Sigma \notin \mathbb{CT}^{\star}_{\forall}$, with $\star \in \{\mathsf{o}, \mathsf{so}\}$. We first show how we encode each state and tape symbol together with the cursor state in our set of tgds. In what follows, let y and z be two variables which intuitively represent a null and a constant respectively. The encoding of each symbol $0, 1, \sqcup$ in Λ is a tuple of 3 variables defined as follows: $\mathbf{0} = z, y, y$; $\mathbf{1} = y, z, y$ and $\boldsymbol{\sqcup} = y, y, z$. The encoding of a state $s_i \in S$ is defined as the tuple of $|S|$ variables $\boldsymbol{s_i} = y^{i-1}, z, y^{|S|-i}$. Finally, we define the encoding of the symbol $\uparrow$ as $\boldsymbol{\uparrow} = z, y$ and the encoding of the symbol $\natural$ as $\boldsymbol{\natural} = y, z$. Intuitively, each symbol will be encoded inside the chase as a tuple of terms, where all terms are nulls, except for the one in the position identifying the encoded symbol. During the construction of our set of sticky tgds, we will need to consider a variation of the encodings defined above, where for every symbol $a \in \Lambda \cup S \cup \{\uparrow, \natural\}$, each occurrence of the variable y in $\boldsymbol{a}$ in some position i is replaced with a fresh new variable y^i_a, We denote such an encoding with $\hat{\boldsymbol{a}}$.

In our construction we use the $(|S|+5\cdot n+2)$-ary predicate Config to represent the configurations of M. The first $|S|$ positions will contain the encoding of the state of the configuration. The next $5 \cdot n$ positions will contain the encoding of the tape where each cell is encoded with 5 positions: 3 positions for the cell value and 2 positions for the cursor state. The last two positions will carry our two variables y and z, respectively. We are now ready to present the set Σ of sticky tgds. We will use a ternary predicate R to generate our initial configuration as follows:

$$R(x, y, z) \to \mathit{Config}(\boldsymbol{s_1}, \boldsymbol{a_1}, \boldsymbol{\uparrow}, \boldsymbol{a_2}, \boldsymbol{\natural}, \ldots, \boldsymbol{a_m}, \boldsymbol{\natural}, \underbrace{\boldsymbol{\sqcup}, \boldsymbol{\natural}, \ldots, \boldsymbol{\sqcup}, \boldsymbol{\natural}}_{2\cdot(n-m)}, y, z)$$

We now simulate the transition function of M. We consider the three different cases where the cursor moves left, right, or stays at the same position. In what follows, we use $\bar{x}_i$ to denote the tuple of 3 variables $x^0_i, x^1_i, x^{\sqcup}_i$, denoting the i-th cell in the tape. We also use $\bar{x}^c_i$ to denote the tuple of 2 variables $x^{\uparrow}_i, x^{\natural}_i$ to denote the cursor state of the i-th cell.

Left: For each transition rule $\delta(s_i, a) = (s_j, b, \leftarrow)$, we introduce, for each $1 \le \ell \le n$, the tgd:

$$\mathit{Conf}(\hat{\boldsymbol{s_i}}, \bar{x}_1, \bar{x}^c_1, \ldots, \bar{x}_{\ell-1}, \bar{x}^c_{\ell-1}, \hat{\boldsymbol{a}}, \hat{\boldsymbol{\uparrow}}, \bar{x}_{\ell+1}, \bar{x}^c_{\ell+1}, \ldots, \bar{x}_n, \bar{x}^c_n, y, z) \to \mathit{Conf}(\boldsymbol{s_j}, \bar{x}_1, \bar{x}^c_1, \ldots, \bar{x}_{\ell-1}, \boldsymbol{\uparrow}, \boldsymbol{b}, \boldsymbol{\natural}, \bar{x}_{\ell+1}, \bar{x}^c_{\ell+1}, \ldots, \bar{x}_n, \bar{x}^c_n, y, z).$$

Right: For each transition rule $\delta(s_i, a) = (s_j, b, \rightarrow)$, we introduce, for each $1 \le \ell \le n$, the tgd:

$$\mathit{Conf}(\hat{\boldsymbol{s_i}}, \bar{x}_1, \bar{x}^c_1, \ldots, \bar{x}_{\ell-1}, \bar{x}^c_{\ell-1}, \hat{\boldsymbol{a}}, \hat{\boldsymbol{\uparrow}}, \bar{x}_{\ell+1}, \bar{x}^c_{\ell+1}, \ldots, \bar{x}_n, \bar{x}^c_n, y, z) \to \mathit{Conf}(\boldsymbol{s_j}, \bar{x}_1, \bar{x}^c_1, \ldots, \bar{x}_{\ell-1}, \bar{x}^c_{\ell-1}, \boldsymbol{b}, \boldsymbol{\natural}, \bar{x}_{\ell+1}, \boldsymbol{\uparrow}, \bar{x}_{\ell+2}, \bar{x}^c_{\ell+2}, \ldots, \bar{x}_n, \bar{x}^c_n, y, z).$$

Stay: For each transition rule $\delta(s_i, a) = (s_j, b, -)$, we introduce, for each $1 \leq \ell \leq n$, the tgd:

$$Conf(\hat{\boldsymbol{s_i}}, \bar{x}_1, \bar{x}_1^c, \ldots, \bar{x}_{\ell-1}, \bar{x}_{\ell-1}^c, \hat{\boldsymbol{a}}, \hat{\uparrow}, \bar{x}_{\ell+1}, \bar{x}_{\ell+1}^c, \ldots, \bar{x}_n, \bar{x}_n^c, y, z) \rightarrow$$
$$Conf(\boldsymbol{s_j}, \bar{x}_1, \bar{x}_1^c, \ldots, \bar{x}_{\ell-1}, \bar{x}_{\ell-1}^c, \boldsymbol{b}, \uparrow, \bar{x}_{\ell+1}, \bar{x}_{\ell+1}^c, \ldots, \bar{x}_n, \bar{x}_n^c, y, z).$$

Finally, once the accepting configuration is reached, an atom of relation symbol R is generated, and a fresh new null is generated in position $R[2]$.

$$Conf(\hat{\boldsymbol{s_2}}, \bar{x}_1, \bar{x}_1^c, \ldots, \bar{x}_n, \bar{x}_n^c, y, z) \rightarrow \exists w R(y, w, z).$$

Our construction is now complete. It is not difficult to show that M accepts on input I iff $\Sigma \notin \mathbb{CT}_\forall^\star$, with $\star \in \{\mathsf{o}, \mathsf{so}\}$. What is critical to show is that Σ is indeed a set of sticky tgds.

By definition of stickiness, Σ is sticky if after the marking process, there is no tgd that contains two occurrences of a variable which is marked in Σ. Note that the only variable occurring more than once in the body of each tgd of Σ is z. Thus, we need to show that z is not marked in each tgd. We show the claim by induction on the number of applications of the marking procedure. To this end, we say that a variable x is marked at step 0, if there is a tgd $\sigma \in \Sigma$ such that $x \in \mathsf{var}(\mathsf{body}(\sigma))$ but $x \notin \mathsf{var}(\mathsf{head}(\sigma))$. We also say that x is marked at step $i > 0$, if there is a tgd $\sigma \in \Sigma$ such that $x \in \mathsf{var}(\mathsf{body}(\sigma)) \cap \mathsf{var}(\mathsf{head}(\sigma))$, and there is a tgd $\sigma' \in \Sigma$ where $\mathsf{body}(\sigma')$ and $\mathsf{head}(\sigma)$ are atoms of the same relation symbol and all variables in $\mathsf{var}(\mathsf{body}(\sigma'))$ at a position of $\mathsf{pos}(\mathsf{head}(\sigma), x)$ are marked at step $i - 1$. Clearly, if one is able to show that for every $i \geq 0$, z is not marked at step i, the claim will follow immediately. We prove such a claim by induction on $i \geq 0$.

Base Step. Since z occurs in both the body and the head of every tgd of Σ, z is not marked at step 0.

Inductive Step. Assume $i > 0$. The variable z is marked at the current step i if there exists a tgd $\sigma \in \Sigma$ with $z \in \mathsf{var}(\mathsf{body}(\sigma)) \cap \mathsf{var}(\mathsf{head}(\sigma))$ and there is another tgd $\sigma' \in \Sigma$ such that each variable in $\mathsf{var}(\mathsf{body}(\sigma'))$ at a position $\mathsf{pos}(\mathsf{head}(\sigma), z)$ is marked at step $i - 1$. Note that for every tgd $\sigma \in \Sigma$, z occurs in the last position of both $\mathsf{body}(\sigma)$ and $\mathsf{head}(\sigma)$. Consider now a tgd $\sigma \in \Sigma$, where $z \in \mathsf{var}(\mathsf{head}(\sigma))$. By inductive hypothesis, z is not marked at any step $j < i$, which means that for every tgd $\sigma' \in \Sigma$, where $\mathsf{body}(\sigma')$ and $\mathsf{head}(\sigma)$ have the same relation symbol, the occurrence of z in the last position of $\mathsf{body}(\sigma')$ is not marked at step $i-1$. Thus, since z also occurs in $\mathsf{head}(\sigma)$ in the last position, z is not marked at step i. This ends our proof.

G.3 $\mathsf{CT}_\forall^\star(\mathbb{S})$ is NLogSpace-hard for predicates of bounded arity

Let $\mathbb{SL}$ be the class of all sets of *simple linear* tgds. A tgd σ is simple linear if it is linear and no variable in $\mathsf{body}(\sigma)$ occurs more than once in $\mathsf{body}(\sigma)$. It has been shown in [4] that $\mathsf{CT}_\forall^\star(\mathbb{SL})$ is NLogSpace-hard even for constant-free tgds with unary and binary predicates. This and the fact that $\mathbb{SL} \subset \mathbb{S}$ imply that $\mathsf{CT}_\forall^\star(\mathbb{S})$ is NLogSpace-hard for predicates of bounded arity.