Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

# About the Multi-Head Linear Restricted Chase Termination

**Lukas Gerlach[1], Lucas Larroque[2], Jerzy Marcinkowski[3], Piotr Ostropolski-Nalewaja[3]**
[1]Knowledge-Based Systems Group, TU Dresden, Dresden, Germany
[2]Inria, DI ENS, ENS, CNRS, PSL University, Paris, France
[3]University of Wrocław, Poland
lukas.gerlach@tu-dresden.de, lucas.larroque@inria.fr, {jma, postropolski}@cs.uni.wroc.pl

**Abstract**

The chase is a ubiquitous algorithm in database theory. However, for existential rules (aka tuple-generating dependencies), its termination is not guaranteed, and even undecidable in general. The problem of termination becomes particularly difficult for the restricted (or standard) chase, for which the order of rule application matters. Thus, decidability of restricted chase termination is still open for many well-behaved classes such as linear or guarded multi-headed rules. We make a step forward by showing that all-instances restricted chase termination is decidable in the linear multi-headed case.

## 1 Introduction

The *chase* procedure was defined in late 1970s, and has been considered one of the most fundamental database theory algorithms since then. It has been applied to a wide spectrum of problems, including query answering and containment under constraints ((Aho, Sagiv, and Ullman 1979) and (Calì, Gottlob, and Kifer 2013) among other examples) computing solutions to data exchange problems (Fagin et al. 2005) and many others (see (Deutsch, Nash, and Remmel 2008) for details). And this is not just about theory: in several contexts the chase procedure has recently been efficiently implemented, see for example (Benedikt et al. 2017), (Urbani et al. 2018), (Ivliev et al. 2024).

The reason for this ubiquity, as elegantly explained in (Deutsch, Nash, and Remmel 2008), is that, for an input consisting of a database instance $\mathbb{I}$ and a set $\mathcal{T}$ of existential rules (also known as tuple-generating dependencies, or TGDs), *chase* returns a finite instance $Chase(\mathbb{I}, \mathcal{T})$ being a universal model of $\mathbb{I}$ and $\mathcal{T}$, i.e., a model that can be homomorphically embedded into every other model of $\mathbb{I}$ and $\mathcal{T}$. Intuitively, $Chase(\mathbb{I}, \mathcal{T})$ is a model of $\mathbb{I}$ and $\mathcal{T}$ which only contains information already implicitly present in $\mathbb{I}$ and $\mathcal{T}$. But it only returns such a finite instance when it **terminates**.

### 1.1 Chase in Five or Six Paragraphs

As we said, the input of *chase* comprises a database instance $\mathbb{I}$ and a set $\mathcal{T}$ of rules, which are formulas of the form:

$$\forall \bar{x}\bar{y}.\ \Phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}.\ \Psi(\bar{y}, \bar{z}) \tag{1}$$

where $\Phi$ and $\Psi$ are conjunctions of atoms, known as the *body* and the *head* of this rule.

At each execution step, the *chase* extends the instance $\mathbb{I}$ by adding new atoms. Specifically, *whenever* there is a rule $\rho$ in $\mathcal{T}$, as in (1), and tuples of terms $\bar{a}$ and $\bar{b}$ such that $\Phi(\bar{a}, \bar{b})$ holds in the current instance, a tuple $\bar{c}$ of fresh terms (called "nulls") is created, and new relational atoms are added to ensure that $\Psi(\bar{a}, \bar{c})$ also holds.[1] This process continues until a fixpoint is reached, yielding a model of both $\mathbb{I}$ and $\mathcal{T}$.

There are many different ways to formalize this idea. In particular, there are two ways the above *whenever* can be interpreted, leading to two fundamental variants of the *chase*.

One variant is the *restricted chase*, which is lazy - it only adds new atoms if $\Phi(\bar{a}, \bar{b})$ is true in the current structure, but $\exists \bar{z}\ \Psi(\bar{a}, \bar{z})$ is not. *Oblivious chase* is an eager version: it always adds a new witness $\Psi(\bar{a}, \bar{c})$ - where $\bar{c}$ is a tuple of fresh terms - for each tuple $\bar{a}, \bar{b}$ such that $\Phi(\bar{a}, \bar{b})$ is true.

The order of execution does not matter for *oblivious chase*. No matter the order in which the witnesses are added, we eventually get the same structure. But *restricted chase* is non-deterministic – different execution orderings in which tuples and rules are picked can lead to different fixpoints.

Clearly, *restricted chase*, in general, builds smaller instances than the oblivious one. It is very easy to devise an example where, according to the rules of *restricted chase*, no new atoms would be ever created, while *oblivious chase* builds an infinite instance. Following (Gogacz, Marcinkowski, and Pieris 2023) consider for example $\mathbb{I} = \{R(a, b)\}$ with $\forall x, y(R(x, y) \rightarrow \exists z R(x, z))$ as the only rule in $\mathcal{T}$. *Restricted chase* will detect that $\mathbb{I}$ already satisfies the rule, while *oblivious chase* will not terminate and will build an instance $\{R(a, b), R(a, v_1), R(a, v_2), \ldots\}$.

### 1.2 The Issue of Termination

As we said above, for *chase* to be useful, one needs to be sure that it terminates. The termination normally considered in this context is all-instance termination[2]: we want to know, for given $\mathcal{T}$, if *chase* terminates on all instances $\mathbb{I}$. This question again comes in two versions: all-instance termination of *oblivious chase* or all-instance termination of *restricted chase*, which means that *restricted chase terminates* regardless of the $\mathbb{I}$ and regardless of the execution order.

[1]The tuple $\bar{z}$ can be empty if $\rho$ is a Datalog rule.
[2]Which may be seen as a "static analysis" problem, since we analyze $\mathcal{T}$ before we know the data.

Surprisingly, only as late as in 2014 (Gogacz and Marcinkowski 2014) it was proven that, in general, all-instance termination - both in the oblivious and restricted variants - is undecidable. A stronger version of this result is presented in (Bednarczyk, Ferens, and Ostropolski-Nalewaja 2020), where undecidability is shown even for relational schemas only containing binary relations.

On the other hand, really a lot of work was done to identify sufficient conditions on the rules which guarantee all-instance termination. The most well-known example of such condition is *weak-acyclicity* (Fagin et al. 2005). But there are numerous other examples too (Deutsch and Tannen 2003; Fagin et al. 2005; Deutsch, Nash, and Remmel 2008; Grau et al. 2013; Greco, Spezzano, and Trubitsyna 2011; Marnette 2009; Meier, Schmidt, and Lausen 2009; Gerlach and Carral 2023).

The undecidability proofs given in (Gogacz and Marcinkowski 2014; Bednarczyk, Ferens, and Ostropolski-Nalewaja 2020; Carral et al. 2025) construct sets of rather complicated TGDs which go very far beyond the most popular well-behaved classes of rules, like linear, guarded or sticky. This leads to a natural question, whether while undecidable in general, all-instance termination could be decidable for sets of rules from such classes[3].

This question turned out to be relatively easy to answer for the case of *oblivious chase*. For guarded TGDs, the problem is 2EXPTIME-complete, and becomes PSPACE-complete for linear TGDs (Calautti, Gottlob, and Pieris 2015). The sticky case was addressed in (Calautti and Pieris 2019), where it is shown that the problem is PSPACE-complete. Things are much more subtle in the case of *restricted chase*, which is also more interesting in this context, because, as we explained, *restricted chase* is more likely to terminate than *oblivious chase*. Things are much more subtle because, as we explained before, in this case we need to deal with all possible execution orderings, each of them leading to different result. But, despite of the difficulties, some progress was made. First it was shown that the problem is decidable for single-head linear[4] TGDs (Leclère et al. 2019). Then decidability was proven, in (Gogacz, Marcinkowski, and Pieris 2023), for guarded and sticky classes of TGDs, but also only in the single-head case.

### 1.3 Fairness. Why Is the Single-Head Case Easier?

One important nuance that we skipped in our brief presentation of *chase* in Section 1.1 is fairness. For *restricted chase* to really create a model of $\mathbb{I}$ and $\mathcal{T}$, it must at some point apply each possible rule to every eligible $\Phi(\bar{a}, \bar{b})$ (unless, of course, $\exists \bar{z}\ \Psi(\bar{a}, \bar{z})$ gets satisfied earlier). Such execution orderings, when no rule application is starved, are called *fair*. Let us illustrate it with two examples:

[3]Notice that, unlike the "sufficient conditions" mentioned above, the well-behaved classes do not imply all-instance termination. They just, hopefuly, make all-instance termination decidable.

[4]A rule like in (1) is linear if $\Phi$ consists of a single atom and it is single-head if $\Psi$ only comprises one atom.

**Example 1.** Let $\mathcal{T}$ consist of two rules:

$$\forall x, y\ R(x, y) \ \rightarrow\ \exists z\ R(y, z) \qquad (\rho_1)$$
$$\forall x, y\ R(x, y) \ \rightarrow\ R(y, y) \qquad (\rho_2)$$

and let $\mathbb{I}$ consist of a single atom $R(a_0, a_1)$.

Then we can execute $\rho_1$ creating a new atom $R(a_1, a_2)$, then apply $\rho_1$ to this atom, creating $R(a_2, a_3)$ and so on, building an infinite instance. Such execution ordering would not be fair: application of $\rho_2$ to $R(a_0, a_1)$, and also to each of the created atoms, would starve.

On the other hand, we could start from applying $\rho_2$ to $R(a_0, a_1)$, creating $R(a_1, a_1)$. In an instance comprising these two atoms both rules would be already satisfied, and *restricted chase* would terminate at this point.

But there is also another execution ordering here, which is fair and which leads to an infinite chase: each time, after you use $\rho_1$ to create $R(a_n, a_{n+1})$, apply $\rho_2$ to $R(a_{n-1}, a_n)$ and produce $R(a_n, a_n)$. □

If the above example gives you the impression that analyzing fairness is complicated, it is the right impression. And, in decidability proofs, like in (Leclère et al. 2019; Gogacz, Marcinkowski, and Pieris 2023) at some point you need to show that, if your termination criterion fails, then there exists an instance leading to an infinite fair derivation.

For this reason, a very important tool in both (Leclère et al. 2019) and (Gogacz, Marcinkowski, and Pieris 2023) are Fairness Lemmas. In (Leclère et al. 2019) it is shown that, for $\mathcal{T}$ being a set of linear single-head rules, if there exists an infinite *restricted chase* then there exists also a fair one. In (Gogacz, Marcinkowski, and Pieris 2023) the authors show that it is actually true for **any** set of **single-head rules**. This allows them to just forget about the fairness condition.

But, as also noticed in both cited papers, this lemma does not hold for rules, even linear, which are multi-head:

**Example 2.** Let $\mathcal{T}$ consist of two rules:

$$\forall x, z, u\ R(x, u, u) \ \rightarrow\ \exists y\ R(x, y, u) \wedge R(y, u, u) \qquad (\gamma)$$
$$\forall y, z, u\ R(y, u, u) \ \rightarrow\ R(u, u, u) \qquad (\delta)$$

and let $\mathbb{I}$ consist of a single atom $R(a, b, b)$. It is easy to see that there exists an infinite restricted derivation: rule $\gamma$ can be applied indefinitely to atoms of the form $R(t_n, b, b)$, where $t_0 = a$. However, such a derivation is not fair with respect to rule $\delta$, and it suffices to apply $\delta$ once to prevent $\gamma$ from ever being applied again. □

For this reason, both the cited papers state decidability of restricted chase termination for multi-head rules as an open problem, whose solution would require new techniques.

### 1.4 Our Contribution

Our contribution in this paper is Theorem 15, saying that all-instances restricted chase termination is decidable for rule sets consisting of linear multi-head rules.

To keep the notations as light as possible, our proof of Theorem 15 is presented using:

**Assumption 3.** *We will assume that all rule heads we consider contain exactly two atoms.*

While this assumption is not free, as we do not have a procedure that normalizes rule sets into rules with exactly two atoms without impacting termination, the proof we present remains correct without this assumption, and no new arguments are needed.

**Organization** In Sections 2-4 we introduce the basic notions and notations, which are in principle standard, but tailored to be useful in Sections 5-8. In Section 5 we formally state our main result, Theorem 15, and present an overview of the proof of this theorem. The proof can be in a natural way split into three parts, and the three parts are covered in Sections 6-8. Some proofs are deferred to the appendix.

## 2 Preliminaries

Let $\mathfrak{P}$ be some finite set of *predicates*, and let $\mathfrak{C}$, $\mathfrak{V}$, and $\mathfrak{N}$ be mutually disjoint, countably infinite sets of *constants*, *variables*, and *labeled nulls* respectively. The function $ar : \mathfrak{P} \to \mathbb{N}$ maps each predicate to its *arity*. The *arity* of $\mathfrak{P}$, denoted with $ar(\mathfrak{P})$, is the maximal arity of its predicates. Elements from $\mathfrak{C} \cup \mathfrak{V} \cup \mathfrak{N}$ are called *terms*. A term $t$ is *ground* if $t \in \mathfrak{C} \cup \mathfrak{N}$. An *atom* is an expression of the form $P(\bar{t})$, where $P \in \mathfrak{P}$, $\bar{t}$ is a list of terms and $|\bar{t}| = ar(P)$. Given an atom $\alpha$ we denote its $i$-th term with $\alpha_i$. A *fact* is an atom with only ground terms. An *instance* is a set of facts. A singleton instance is called *atomic*. The *active domain* of a set (or a conjunction) $A$ of atoms is the set (denoted as $adom(A)$) of terms appearing in atoms of $A$.

**Existential Rules** An FO formula of the form

$$\forall \bar{x}\bar{y}.\, \Phi(\bar{x}, \bar{y}) \to \exists \bar{z}.\, \Psi(\bar{y}, \bar{z})$$

is called an *existential rule*, where $\bar{x}, \bar{y}$ and $\bar{z}$ are pairwise disjoint tuples of variables, and $\Phi$ and $\Psi$ are conjunctions of atoms. For a rule $\sigma$ as above, we denote $\Phi(\bar{x}, \bar{y})$ as $body(\sigma)$ and $\Psi(\bar{y}, \bar{z})$ as $head(\sigma)$. In this paper we only consider *linear* rule sets, which means that the body of each rule contains at most one atom. The position in one of the atoms of a rule (both in the body or the head) is called a *frontier position* if it contains one of the variables from the tuple $\bar{y}$.

Almost **all the objects we define in Sections 2 - 4 depend on parameters: an instance $\mathbb{I}$ and a rule set $\mathcal{T}$. We always assume that $\mathcal{T}$ is a set of linear rules adhering to Assumption 3.** This dependence is not reflected in the notation, as we do not believe this leads to misunderstandings.

**Homomorphisms.** A homomorphism $h$ from a set of atoms $A$ to a set of atoms $B$ is a function from $adom(A)$ to $adom(B)$ such that for every atom $\alpha(\bar{x}) \in A$ we have $\alpha(h(\bar{x})) \in B$. The notion of homomorphism naturally extends to queries, bodies, and heads of existential rules.

**Adresses and Forests.** Now we introduce the language we will later use to address atoms in the (real) oblivious chase. Recall that our rules are linear, so (intuitively) every atom $\alpha$ produced by *chase* has a unique "parent atom", from whom $\alpha$ was created by some rule of $\mathcal{T}$, and the genealogy of such $\alpha$ can always be traced back to $\mathbb{I}$.

**Definition 4** (Addresses)**.**
- Define the set $\mathbb{A}$ of *address symbols* as $\{\rho_\iota : \rho \in \mathcal{T} \text{ and } \iota \in \{1, 2\}\}$. We always reserve the letter $\iota$ for an index ranging over the set $\{1, 2\}$, and this is related to Assumption 3.
- *Address* is a word $\omega u \in \mathbb{I}\mathbb{A}^*$, i.e. a word whose first element is an atom from $\mathbb{I}$, followed by a sequence of address symbols. We will usually use letters $u, w, v$, and sometimes also $s, r$, to denote words from $\mathbb{I}\mathbb{A}^*$ or $\mathbb{A}^*$.
- For $w \in \mathbb{I}\mathbb{A}^*$ and $a, b \in \mathbb{A}$ we say that $wa$ is an *a-child* of $w$ (or just *child of* $w$), and that $wa$ and $wb$ are *siblings*.
- By an *infinite path* we mean an infinite sequence of adresses $w_0, w_1, w_2, \ldots$ such that $w_0 \in \mathbb{I}$ and $w_{n+1}$ is a child of $w_n$, for each $n$.

**Definition 5** (Forests)**.** A *forest* is a set $\mathbb{G}$ of addresses such that $\mathbb{I} \subseteq \mathbb{G} \subseteq \mathbb{I}\mathbb{A}^*$, and which is *prefix-closed*, meaning that if $\omega uv \in \mathbb{G}$ for some $\omega \in \mathbb{I}$ and $uv \in \mathbb{A}^*$ then also $\omega u \in \mathbb{G}$.

If $\mathbb{I}$ is a singleton we use the term *tree* instead of *forest*. For a forest $\mathbb{G}$ and $u \in \mathbb{G}$ we use the notation $\mathbb{G}_u$ for the set $\{uv \mid uv \in \mathbb{G}\}$ of all the descendants of $u$ in $\mathbb{G}$.

## 3 The Real Oblivious Chase

In this section, following (Gogacz, Marcinkowski, and Pieris 2023), we define the (real) *oblivious chase*, a static object, that serves as an arena where restricted chase takes place.

**Definition 6** (Rule application)**.** Let $\alpha$ be an atom, let $\rho \in \mathcal{T}$ be a rule, such that there exists a homomorphism $h$ from $body(\rho)$ to $\alpha$, and let $u$ be an address. Then $ruleApply(\rho, \alpha, u)$ is a pair of atoms obtained from $head(\rho)$ by replacing each universally quantified variable $x$ of $\rho$ with $h(x)$ and each existentially quantified variable $z$ with a labeled null of the form $z^\rho_u$. We denote the $\iota$-th atom of $ruleApply(\rho, \alpha, u)$ with $ruleApply_\iota(\rho, \alpha, u)$.

The above definition allows us to precisely identify when each labeled null was created. It will be useful in Section 6.

The following definition is one of the most important:

**Definition 7.** The *representation of the oblivious chase* (or simply *representation*) is a pair consisting of a forest $\mathbb{F}$ and a labeling function $\langle\_\rangle$ from $\mathbb{F}$ to ground atoms. We define $\mathbb{F}$ and $\langle\_\rangle$ by a simultaneous induction:

- if $\omega \in \mathbb{I}$ then $\omega \in \mathbb{F}$, and $\langle\omega\rangle = \omega$;
- If $u \in \mathbb{F}$, $\rho$ is a rule of $\mathcal{T}$, and there is homomorphism $h$ from $body(\rho)$ to $\langle u\rangle$ then for each $\iota$ it holds that $u\rho_\iota \in \mathbb{F}$ and $\langle u\rho_\iota\rangle = ruleApply_\iota(\rho, \langle u\rangle, u)$.

Note that this definition deliberately avoids discussing the notion of triggers and their application order. This is for a good reason: as we said before, the oblivious chase is defined statically, and serves as a scene for the restricted chase.

We will often need to talk about the set of atoms represented by a forest. Therefore, by slightly abusing notation:

**Definition 8** (Materialization)**.** Given a forest $\mathbb{G} \subseteq \mathbb{F}$ we define $\langle\mathbb{G}\rangle$ as $\{\langle u\rangle : u \in \mathbb{G}\}$.

This allows us to cleanly define the oblivious chase, and later, the restricted chases.

**Definition 9.** The *oblivious chase* is defined as $\langle\mathbb{F}\rangle$.

Chase, and restricted chase in particular, is all about atoms. But each atom of interest is somewhere in $\langle\mathbb{F}\rangle$ and **the only way we will ever refer to atoms will be by their addresses** in $\mathbb{F}$.

Notice that the labeling function $\langle \_ \rangle$ does not need to be one-to-one. To visualize this, consider Example 2 and addresses of the form $(R(a,b,b))(\gamma_2)^*\delta_1$. Each such address is labeled with $R(b,b,b)$. For this reason (Gogacz, Marcinkowski, and Pieris 2023) think of $\langle \mathbb{F} \rangle$ as a multiset and call it *the real oblivious chase*.

## 4 Triggers and Chase Derivations

A trigger is an address where an atom lives, and a rule which can be applied to this atom:

**Definition 10** (Triggers)**.** • A trigger is a pair $(\rho, u)$ consisting of a rule $\rho \in \mathcal{T}$ and of a $u \in \mathbb{F}$ such that there exists a (unique) homomorphism from $body(\rho)$ to $\langle u \rangle$.

- For a trigger $\pi$, we denote as $apply(\pi)$ the set of addresses $\{u\rho_1, u\rho_2\}$ and with $apply_\iota(\pi)$ the address $u\rho_\iota$.
- For a forest $\mathbb{G} \subseteq \mathbb{F}$, and a trigger $\pi = (\rho, u)$, we say that $\pi$:
  - *appears* (or *is*) *in* $\mathbb{G}$ *iff* $u \in \mathbb{G}$. We also write $\pi \in \mathbb{G}$.
  - is *active* in $\mathbb{G}$ *iff* $\pi \in \mathbb{G}$ and $apply(\pi) \not\subseteq \mathbb{G}$.

Note that every trigger (by definition) appears in $\mathbb{F}$. However, not every pair $(\rho, u)$ qualifies as a trigger. For such a pair to be a trigger, there must exist a homomorphism from the body of $\rho$ to $\langle u \rangle$. Since the bodies of rules consist of single atoms, at most one such homomorphism can exist.

The very idea of restricted chase is that blocked triggers are never applied:

**Definition 11** (Trigger blocking)**.** The pair $(v_1, v_2) \in \mathbb{G}^2$ is a *blocking team* for a trigger $(\rho, u)$ in $\mathbb{G} \subseteq \mathbb{F}$ *iff* there is a homomorphism $h$ such that $h(\langle u \rangle) = \langle u \rangle$ and $h(\langle u\rho_\iota \rangle) = \langle v_\iota \rangle$ for all $\iota$. A trigger that has a blocking team is *blocked*.

In other words, $(v_1, v_2)$ is a blocking team for $\pi$ if atoms $\langle v_1 \rangle, \langle v_2 \rangle$ can witness that the existential rule $\rho$ is satisfied, when applied to the atom $\langle u \rangle$. The following may be trivial to see, but is important to realize:

**Observation 12.** *Let $\rho$ be a rule of the form*

$$\forall \bar{x}\bar{y}.\, \phi(\bar{x}, \bar{y}) \to \exists \bar{z}.\, \psi_1(\bar{y}, \bar{z}) \wedge \psi_2(\bar{y}, \bar{z})$$

*for some atoms $\phi, \psi_1, \psi_2$. Then a pair $(v_1, v_2)$ is a blocking team for trigger $\pi = (\rho, u)$ if and only if, for each $\iota, \iota'$:*

*(i) the predicate symbols of $\langle v_\iota \rangle$ and of $\langle u\rho_\iota \rangle$ are equal;*

*(ii) if $j$ is a frontier position in $\psi_\iota$ then $\langle u\rho_\iota \rangle_j = \langle v_\iota \rangle_j$;*

*(iii) if $j$ is any position in $\psi_\iota$ and $j'$ is any position in $\psi_{\iota'}$ and $\langle u\rho_\iota \rangle_j = \langle u\rho_{\iota'} \rangle_{j'}$ then also $\langle v_\iota \rangle_j = \langle v_{\iota'} \rangle_{j'}$.*

Intuitively, the observation says that atoms in the blocking team of $(\rho, u)$ must contain at least the same positive information as the atoms $\langle u\rho_\iota \rangle$ do. But they may contain more: they may have more equalities between terms, and they can have something non-anonymous (for example constants from $\mathbb{I}$) on positions where $\langle u\rho_\iota \rangle$ have new anonymous terms. This makes them potentially more "capable" of producing new atoms, and also of blocking, than $\langle u\rho_\iota \rangle$ are.

Now, we introduce the notion of derivation, which intuitively corresponds to a growing sequence of forests that results from a sequence of trigger applications.

**Definition 13** (Chase derivations)**.** A *chase derivation $\mathcal{M}$ (of the rule set $\mathcal{T}$ over the instance $\mathbb{I}$)* is a (finite[5] or infinite) sequence of triggers $\{\pi_n^{\mathcal{M}}\}_{n \in \mathbb{N}}$ such that there exists a sequence of forests $\{\mathbb{M}_n\}_{n \in \mathbb{N}}$ called *derivation steps* where:

- $\mathbb{M}_0 = \mathbb{I}$;
- trigger $\pi_n^{\mathcal{M}}$ is active in $\mathbb{M}_n$;
- $\mathbb{M}_{n+1} = apply(\pi_n^{\mathcal{M}}) \cup \mathbb{M}_n$.

For a derivation $\mathcal{M}$, we denote the forest $\bigcup_{n \in \mathbb{N}} \mathbb{M}_n$ by $\mathbb{M}$.

Clearly, for each derivation $\mathcal{M}$ there is $\mathbb{M} \subseteq \mathbb{F}$.

**Definition 14** (Derivation Types)**.** Derivation $\mathcal{M}$ is:

- *oblivious* when $\mathbb{M} = \mathbb{F}$;
- *restricted* when no trigger $\pi_n^{\mathcal{M}}$ is blocked in $\mathbb{M}_n$;
- *fair* when each active trigger in $\mathbb{M}$ is blocked in $\mathbb{M}$.

Informally speaking, a fair restricted derivation is a sequence in which all active triggers eventually either get applied or become blocked. Notice also (still being informal) that all oblivious derivations are permutations of the same set of all triggers, and that such a permutation is an oblivious derivation whenever it satisfies the obvious constraint that trigger $(\rho, u)$ may not precede the creation of $u$.

## 5 Stating the Main Theorem

$\mathrm{CT}_\forall(\mathbb{I})$ is the class of rule sets $\mathcal{T}$ such that all fair restricted derivations of $\mathcal{T}$ over instance $\mathbb{I}$ are finite. $\mathrm{CT}_{\forall\forall}$ is the class of rule sets that belong to $\mathrm{CT}_\forall(\mathbb{I})$ for all finite instances $\mathbb{I}$.

**Theorem 15** (Main Theorem)**.** *Membership in $\mathrm{CT}_{\forall\forall}$ for sets of multi-headed linear existential rules is decidable.*

We dedicate the rest of this paper to the proof of Theorem 15. Consider a rule set $\mathcal{T}$, which will now remain fixed.

Let us now begin our proof with a high-level overview.

### 5.1 Overview of the Proof of Theorem 15

Our proof of Theorem 15 relies on three main lemmas.

First, in Section 6 we define (for a fixed $\mathbb{I}$) an object called mixed derivation and we show our First Main Lemma: for each $\mathbb{I}$, existence of a mixed derivation is equivalent to the existence of an infinite, fair, restricted derivation.

Then, in Section 7 we define, for an atom $\omega$, another object, called $\omega$-path-sensitive-derivation. This definition will not depend on $\mathbb{I}$. As our Second Main Lemma, in this section, we prove that existence of a mixed derivation for *any* $\mathbb{I}$ is equivalent to existence of an $\omega$-path-sensitive derivation.

Finally, in Section 8 we show how to write a Monadic Second Order Logic formula $\Omega_\omega$ (which of course depends on $\mathcal{T}$), which is true on the tree $\omega\mathbb{A}^*$ (with the relations $\rho_\iota$-child for $\rho_\iota \in \mathbb{A}$) if and only if there exists an $\omega$-path-sensitive derivation (this equivalence is stated as our Third Main Lemma). Since there are only finitely many pairwise non-isomorphic atoms $\alpha$, this, together with the well-known fact that Monadic Second Order Logic is decidable over

[5] Since we mainly consider infinite derivations, our notation is tailored for such derivations. In order to talk about finite derivations, $\mathbb{N}$ would need to be replaced with its initial segment.

such infinite trees, gives the decision procedure, and completes the proof of Theorem 15. Notice that our entire decision procedure is confined to Section 8. The only purpose of Sections 6 and 7 is to prove correctness of this procedure.

# 6 Mixed Derivations

A finite instance $\mathbb{I}$ remains fixed throughout this section.

**Definition 16.** A *mixed derivation* is a pair $(\mathcal{M}, \mathcal{R})$ where $\mathcal{M} = \{\pi_n^{\mathcal{M}}\}_{n\in\mathbb{N}}$ is an infinite oblivious derivation, and $\mathcal{R}$ is an infinite fair and restricted derivation, such that:

- $\mathcal{R}$ is a subsequence of $\mathcal{M}$;
- if $\pi_n^{\mathcal{M}} \in \mathcal{R}$ for some $n$ then $\pi_n^{\mathcal{M}}$ is not blocked in $\mathbb{M}_n$.

**Lemma 17** (First Main Lemma)**.** *The following statements are equivalent:*

1. *There exists an infinite restricted and fair derivation.*
2. *There exists a mixed derivation.*

We devote the rest of this section to the proof of the above theorem. The $(2 \Rightarrow 1)$ direction is trivial, so it is just the $(1 \Rightarrow 2)$ implication that we need to show. The proof, we believe, is not completely straightforward. Thus, we kindly ask the Reader to brace for complications, and nuances.

Suppose $\mathcal{R}$ is a fair infinite restricted derivation and let $\mathbb{R}$ be its associated forest of addresses (as per Definition 13).

In what follows, we aim to construct an oblivious derivation $\mathcal{M}$ such that $(\mathcal{M}, \mathcal{R})$ is indeed a mixed derivation.

What does it actually mean to construct this $\mathcal{M}$?

$\mathcal{M}$, as we know, is a permutation of all triggers. For $\mathcal{R}$ to be a subsequence of $\mathcal{M}$ we need to somehow start from $\mathcal{R}$ and then slot the triggers from $\mathcal{M} \setminus \mathcal{R}$ between[6] the triggers in the sequence $\mathcal{R}$. Clearly, we need to ensure that all triggers $(\rho, u)$ are slotted after $u$ is created, as stated at the end of Section 4. What else can possibly go wrong here?

Think about the last item in Definition 16, and imagine we have a trigger $\pi$ somewhere in the sequence $\mathcal{R}$. We may need to slot some triggers from $\mathcal{M} \setminus \mathcal{R}$ ahead of $\pi$. However, slotting triggers introduces new blocking teams from $\mathbb{F}$. We thus need to be very careful to make sure that no blocking team for $\pi$ is created by the triggers slotted ahead of $\pi$. One may think — and, thanks to the assumption that $\mathcal{R}$ is restricted, this is actually true, albeit not obvious — that each blocking team can only block finitely many triggers from $\mathcal{R}$. Thus, maybe we could simply, for each such blocking team, add its members after all its blockees from $\mathcal{R}$? It is not so simple, because while each blocking team does only block finitely many triggers, there may potentially be addresses participating in infinitely many blocking teams. To visualize this, consider again Example 2, and addresses of the form $(R(a,b,b))(\gamma_2)^*\delta_1$. Each such address participates in infinitely many blocking teams of triggers of the form $(\gamma,\ (R(a,b,b))(\gamma_2)^*)$.

[6]By $\mathcal{M} \setminus \mathcal{R}$ (which does not type, as $\mathcal{M}$ and $\mathcal{R}$ are sequences, not sets) we mean the set of triggers that do not appear in $\mathcal{R}$.

## 6.1 The Border of $\mathbb{R}$ and Beyond

Since $\mathcal{R}$ is restricted, it might happen for some address $u$ of $\mathbb{F}$ that while $u \in \mathbb{R}$ there is a trigger $\pi = (\rho, u)$ such that $u\rho_1, u\rho_2 \notin \mathbb{R}$ as $\pi$ is blocked by some blocking team of addresses of $\mathbb{R}$. Intuitively, we can imagine, that the border between $\mathbb{R}$ and $\mathbb{F} \setminus \mathbb{R}$ runs between such $u$ and its $\rho$-children

**Definition 18.** A trigger $(\rho, u) \notin \mathcal{R}$ is called *border* trigger if $u \in \mathbb{R}$, and is called *problematic* trigger if $u \notin \mathbb{R}$.

While at this point it may not be totally obvious for the Reader how problematic triggers earned their name, we will clarify this after Definition 23.

Next we introduce some terminology for addresses in $\mathbb{F} \setminus \mathbb{R}$. Notice that for each such address $v$ there is a (unique) border trigger $(\rho, u)$ such that $v = u\rho_\iota w$ for some $\iota$ and $w$.

**Definition 19** (Virtual addresses and ranks)**.** Addresses in $\mathbb{F} \setminus \mathbb{R}$ are called *virtual*. When we say that $u\rho_\iota w$ is a virtual address, we think that $(\rho, u)$ is a border trigger; in such cases, we sometimes say $u\rho_\iota w$ *belongs to* $u$.

We define the *rank* of an address, by $rk(u) = 0$ if $u \in \mathbb{R}$ and $rk(u\rho_\iota w) = |\rho_\iota w|$ if $u\rho_\iota w$ is virtual.

So, function $rk$ (rank) says how far an address is from $\mathbb{R}$.

**Definition 20.** Any term that appears in some atom of $\langle\mathbb{R}\rangle$ is called *global*. Any other term from $\langle\mathbb{F}\rangle$ is *local*.

We end this section with an obvious observation, which will be very useful soon. Claim (iii) says that each local term is localized to the pair of subtrees of $\mathbb{F}$ rooted in $apply(\pi)$ for exactly one border trigger $\pi$. Notice, however, that global terms can appear in multiple such subtrees.

**Observation 21.** *(i) Suppose $s, sw \in \mathbb{F}$ and a term $t$ occurs in both $\langle s\rangle$ and $\langle sw\rangle$. Then for each prefix $v$ of $w$ term $t$ occurs in $\langle sv\rangle$.*

*(ii) Suppose $s, w \in \mathbb{F}$ are s.t. neither of them is a prefix of the other, and some term $t$ occurs in both $\langle s\rangle$ and $\langle w\rangle$. Then there are prefixes $w'$ and $s'$ of $w$ and $s$ that are siblings, and such that $t$ occurs in both $\langle w'\rangle$ and $\langle s'\rangle$.*

*(iii) For each local term $t$ there exists exactly one border trigger $(\delta, v)$ such that if $t$ appears in some atom $\langle w\rangle$ then $w \in \mathbb{F}_{v\delta_1} \cup \mathbb{F}_{v\delta_2}$.*

## 6.2 Finding Better Versions of Elements of $\mathbb{F}$

We assumed that $\mathcal{R}$ is a restricted and **fair** derivation. So the only reason for a trigger $\pi$ to be a border trigger is that it was blocked by some blocking team $(v, v')$ of $\mathbb{R}$. Note that $(v, v')$ may not be the only blocking team for $\pi$ in $\mathbb{R}$, but for our reasoning we need to consider any such team:

**Definition 22.** We define $blockers$ as a function assigning to each border trigger $\pi$ a blocking team $(v_1, v_2)$ for $\pi$, such that $v_1, v_2 \in \mathbb{R}$.

Recall the remark we made after Definition 11, that members of blocking teams are "more capable" than the blockees. We will now formalize this intuition. We begin with:

**Definition 23.** Suppose $\pi = (\rho, u)$ is a border trigger and $blockers(\pi) = (v_1, v_2)$. Then we define $btr(u\rho_\iota) = v_\iota$.

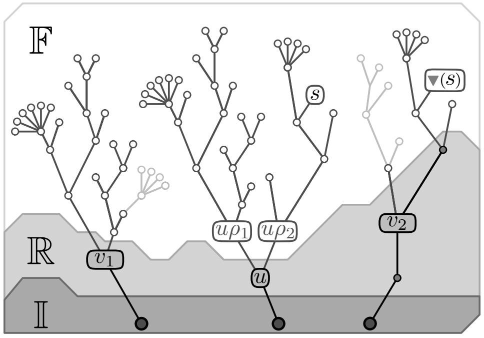


Figure 1: Section 6 in one picture. You can see a border trigger $\pi = (\rho, u)$: address $u$ is still in $\mathbb{R}$ while both $u\rho_\iota$ are right behind the border. The pair $(v_1, v_2)$ of addresses in $\mathbb{R}$ is a blocking team for $\pi$, and $v_\iota = btr(u\rho_\iota)$. One can also see that the tree rooted in $v_\iota$ contains (possibly among other addresses) a copy of the tree rooted in $u\rho_\iota$ (as Lemma 25(B) postulates).

In the above definition $btr$ is short of *better*. Now we might be ready to explain why the non-border triggers of $\mathcal{M} \setminus \mathcal{R}$ are more problematic than border triggers. In both cases we need to slot them between the triggers of $\mathcal{R}$. However, if we slot a border trigger $(\rho, u)$ after addresses $btr(u\rho_1)$ and $btr(u\rho_2)$ were already created in $\mathcal{R}$, then $u\rho_1$ and $u\rho_2$ will not block any trigger which comes later in $\mathcal{R}$. Why? Because $\langle btr(u\rho_1)\rangle$ and $\langle btr(u\rho_2)\rangle$ are better at blocking than $\langle u\rho_1\rangle$ and $\langle u\rho_2\rangle$, and they were already in $\langle\mathbb{R}\rangle$ at this point without blocking anyone, since $\mathcal{R}$ is restricted.

On the other hand, problematic triggers do not necessarily have blocking teams in $\mathbb{R}$. Thus, for a problematic trigger $\pi = (\rho, u)$, it is not obvious how to find *better* versions of $u\rho_\iota$ in $\mathbb{R}$. In consequence, there is no natural moment in the derivation $\mathcal{R}$ when we could already be sure that addresses produced by $\pi$ can no longer block any future triggers from $\mathcal{R}$, so $\pi$ can safely be slotted at that moment.

Now, we are going to extend the function *better* to addresses produced by problematic triggers. As we said in Section 4, members of a blocking team are also more capable of producing new addresses than their blockees. This will be formalized in Lemma 25.

Recall that given an atom $\langle u\rangle$, with $\langle u\rangle_i$ we denote the term at position $i$ in $\langle u\rangle$.

**Definition 24.** For $u, w \in \mathbb{F}$ we say that $w$ is *prettier* than $u$, denoted as $u \preccurlyeq w$, if there exists a homomorphism from $\langle u\rangle$ to $\langle w\rangle$ which is an identity on global terms.

In other words, $u \preccurlyeq w$ if and only if:

(i) $\langle u\rangle$ and $\langle w\rangle$ are atoms of the same relation;

(ii) if $\langle u\rangle_i$ is a global term then $\langle u\rangle_i = \langle w\rangle_i$;

(iii) if $\langle u\rangle_i = \langle u\rangle_j$ then $\langle w\rangle_i = \langle w\rangle_j$.

Now recall Observation 12. We are going to use it a lot.

**Lemma 25.** *Let $(\rho, u)$ be a border trigger. If $u\rho_\iota w \in \mathbb{F}$, then $(btr(u\rho_\iota))w \in \mathbb{F}$ and $u\rho_\iota w \preccurlyeq (btr(u\rho_\iota))w$.*

*Proof.* Let us start from a proof that $u\rho_\iota \preccurlyeq btr(u\rho_\iota)$. From Definition 23, there is a homomorphism from $u\rho_\iota$ to $btr(u\rho_\iota)$, so we just need to check whether it is an identity on global terms. First, notice that (since $(\rho, u)$ is a border trigger), if $\langle u\rho_\iota\rangle_i$ is a global term, then position $i$ is a frontier position in the body-atom of $\rho$. By Observation 12 (ii) and definition of $btr$, we get $\langle u\rho_\iota\rangle_i = \langle btr(u\rho_\iota)\rangle_i$ as required.

Since prettier parents produce prettier children, the lemma then follows by easy induction on the length of $w$. ☐

It follows from Lemma 25 that $\blacktriangledown$ is a function on $\mathbb{F}$:

**Definition 26.**
$$\blacktriangledown(u) = u, \qquad \text{(For } u \in \mathbb{R}\text{)}$$
$$\blacktriangledown(u\rho_\iota v) = (btr(u\rho_\iota))v \qquad \text{(For virtual } u\rho_\iota v\text{)}$$

The Reader may now want to find the virtual address $s$ in Figure 1, and its image $\blacktriangledown(s)$. Clearly, $rk(\blacktriangledown(s)) < rk(s)$, and it is no coincidence. Indeed, function $\blacktriangledown$, when applied to virtual addresses, is strictly monotonic with respect to rank: it moves addresses closer and closer to $\mathbb{R}$. In consequence, whatever address we start from, after a finite number of applications, $\blacktriangledown$ will map this address to $\mathbb{R}$:

**Lemma 27.** *If $w \in \mathbb{F}$ is a virtual address, then $rk(\blacktriangledown(w)) < rk(w)$. For each $w \in \mathbb{F}$ there exists a natural number $m_w$ such that $\blacktriangledown^{m_w}(w) \in \mathbb{R}$.*

*Proof.* For the proof of the first claim let $w = u\rho_\iota v$ where $(\rho, u)$ is a border trigger, and recall that, by Definition 19, $rk(u\rho_\iota v) = |\rho_\iota v|$. On the other hand, $btr(u\rho_\iota)$ is an address in $\mathbb{R}$, so $rk((btr(u\rho_\iota))v) \leq |v|$. Second claim follows immediately from the first. ☐

We now can extend our definition of $btr$ to the entire $\mathbb{F}$. Your better address in $\mathbb{R}$ is the one where function $\blacktriangledown$ ultimately maps you to:

**Definition 28.** For a problematic trigger $(\rho, u)$, define $btr(u\rho_\iota) = \blacktriangledown^{m_{u\rho_\iota}}(u\rho_\iota)$, and for any $v \in \mathbb{R}$ let $btr(v) = v$.

We are ready for our crucial Lemma and its Corollary.

**Lemma 29.** *Suppose $\pi$ is a trigger from $\mathcal{R}$ and $(w_1, w_2)$ is a blocking team of $\pi$ in $\mathbb{F}$. Then $(\blacktriangledown(w_1), \blacktriangledown(w_2))$ is also a blocking team for $\pi$.*

**Corollary 30.** *Suppose $\pi$ is a trigger from $\mathcal{R}$ and $(w_1, w_2)$ is a blocking team of $\pi$ in $\mathbb{F}$. Then $(btr(w_1), btr(w_2))$ is also a blocking team for $\pi$.*

Now, in order to end the proof of our First Main Lemma, we need to do two things: show how Corollary 30 implies First Main Lemma (which is now easy, see Section 6.5) and prove Lemma 29 (which we do not think looks obvious at this point) which is done in the next two subsections.

### 6.3 Proof of Lemma 29. The Easy Part

Throughout this subsection, we consider a border trigger $(\delta, v)$. Recall, from Lemma 25, that for each $w$, if the address $v\delta_\iota w$ is in $\mathbb{F}$ then the address $btr(v\delta_\iota)w$ also is in $\mathbb{F}$, and is prettier than $v\delta_\iota w$. So, one may think of $\mathbb{F}_{btr(v\delta_\iota)}$ as of a prettier copy of the set $\mathbb{F}_{v\delta_\iota}$.

We will begin this section with three very easy observations. The general message is that if two atoms in $\langle\mathbb{F}_{v\delta_\iota}\rangle$ share terms then the respective atoms in $\langle\mathbb{F}_{btr(v\delta_\iota)}\rangle$ also share terms at the respective positions.

The observations will be followed by one lemma, which is not difficult either, but there is certain depth there.

First, notice that that the prettier you are the more terms you share with your children:

**Observation 31.** *Let $w \in \mathbb{A}^*$ and $a \in \mathbb{A}$ and suppose that $\langle v\delta_\iota w\rangle_i = \langle v\delta_\iota wa\rangle_{i'}$ Then also $\langle btr(v\delta_\iota)w\rangle_i = \langle btr(v\delta_\iota)wa\rangle_{i'}$.*

*Proof.* Follows from Lemma 25 and Definition 6. □

And the prettier your parent is the more terms you share with your sibling:

**Observation 32.** *Let $w \in \mathbb{A}^*$ and $a, a' \in \mathbb{A}$. If $\langle v\delta_\iota wa\rangle_i = \langle v\delta_\iota wa'\rangle_{i'}$ then also $\langle btr(v\delta_\iota)wa\rangle_i = \langle btr(v\delta_\iota)wa'\rangle_{i'}$.*

*Proof.* Follows from Lemma 25 and Definition 6. □

By inductive application of Observation 31 we get:

**Observation 33.** *Let $w, w' \in \mathbb{A}^*$. If $\langle v\delta_\iota w\rangle_i = \langle v\delta_\iota ww'\rangle_{i'}$ then $\langle btr(v\delta_\iota)w\rangle_i = \langle btr(v\delta_\iota)ww'\rangle_{i'}$.*

Observations 31-33, with the Lemma 25, imply that there is a homomorphism from $\langle \mathbb{F}_{v\delta_\iota}\rangle$ to $\langle \mathbb{F}_{btr(v\delta_\iota)}\rangle$. Now we want something more, we want to prove that there is a homomorphism from $\langle \mathbb{F}_{v\delta_1} \cup \mathbb{F}_{v\delta_2}\rangle$ to $\langle \mathbb{F}_{btr(v\delta_1)} \cup \mathbb{F}_{btr(v\delta_2)}\rangle$.

**Lemma 34.** *Let $w, w' \in \mathbb{A}^*$. If $\langle v\delta_1 w\rangle_i = t = \langle v\delta_2 w'\rangle_{i'}$ for some term t. Then also $\langle btr(v\delta_1)w\rangle_i = \langle btr(v\delta_2)w'\rangle_{i'}$.*

*Proof.* From Observation 21 we know that $t = \langle v\delta_1\rangle_j = \langle v\delta_2\rangle_{j'}$ for some positions $j, j'$. Now recall that the pair $(btr(v\delta_1), btr(v\delta_2))$ is a blocking team for $(\delta, v)$. By condition (iii) of Observation 12 this gives us that $\langle btr(v\delta_1)\rangle_j = \langle btr(v\delta_2)\rangle_{j'}$. Now use Observation 33. □

### 6.4 Proof of Lemma 29. The Harder Part

For ease of reading, we restate Observation 12 here:

**Observation 12** (shortened)**.** *Let $\rho$ be a rule of the form*
$$\forall\bar{x}\bar{y}.\, \phi(\bar{x},\bar{y}) \to \exists\bar{z}.\, \psi_1(\bar{y},\bar{z}) \wedge \psi_2(\bar{y},\bar{z}).$$
*Then $(v_1, v_2)$ is a blocking team for trigger $(\rho, u)$ iff*

*(i) the predicate symbols of $\langle v_\iota\rangle$ and of $\langle u\rho_\iota\rangle$ are equal;*

*(ii) if j is a frontier position in $\psi_\iota$ then $\langle u\rho_\iota\rangle_j = \langle v_\iota\rangle_j$;*

*(iii) if j is any position in $\psi_\iota$ and $j'$ is any position in $\psi_{\iota'}$ and $\langle u\rho_\iota\rangle_j = \langle u\rho_{\iota'}\rangle_{j'}$ then also $\langle v_\iota\rangle_j = \langle v_{\iota'}\rangle_{j'}$.*

Now we finish the proof of Lemma 29. Let $\rho$ be as in Observation 12. Since $\pi$ is a trigger from $\mathcal{R}$, all terms which occur in atoms $\langle u\rangle$, $\langle u\rho_1\rangle$ and $\langle u\rho_2\rangle$ are global. Since $(w_1, w_2)$ is a blocking team for $\pi$, we know that it satisfies the conditions (i)–(iii) from Observation 12. Our goal is to prove that $(\blacktriangledown(w_1), \blacktriangledown(w_2))$ also satisfies these three conditions.

**Condition (i).** We know, from Lemma 25, that each of $\blacktriangledown(w_\iota)$ is prettier than $w_\iota$. This means, in particular, that $\blacktriangledown(w_\iota)$ and $w_\iota$ have the same predicate symbol, and thus condition (i) from Observation 12 is satisfied for $\blacktriangledown(w_\iota)$.

**Condition (ii).** Consider some frontier position $j$ in $\psi_\iota$. We know, from Observation 12 (ii) that $\langle u\rho_\iota\rangle_j = \langle w_\iota\rangle_j$. Since $\langle u\rho_\iota\rangle_j$ is a global term, and since $w_\iota \preccurlyeq \blacktriangledown(w_\iota)$, we get that condition (ii) from this Observation also holds for $\blacktriangledown(w_\iota)$.

**Condition (iii).** Here things are more complicated. Directly from the fact that $w_\iota \preccurlyeq \blacktriangledown(w_\iota)$ we get that condition (iii) is satisfied for $(\blacktriangledown(w_1), \blacktriangledown(w_2))$, but only if $\iota = \iota'$.

Recall that $(w_1, w_2)$ is a blocking team, so it satisfies condition (iii) for $\rho$. So, in order to prove that (iii) is also true for $(\blacktriangledown(w_1), \blacktriangledown(w_2))$ it suffices to show that if $\langle w_1\rangle_j = t = \langle w_2\rangle_{j'}$ for some term $t$ then also $\langle \blacktriangledown(w_1)\rangle_j = \langle \blacktriangledown(w_2)\rangle_{j'}$.

Now, if $t$ is global, then Lemma 25 and Def. 26 tell us that $t = \langle w_1\rangle_j = \langle \blacktriangledown(w_1)\rangle_j$ and $t = \langle w_2\rangle_{j'} = \langle \blacktriangledown(w_2)\rangle_{j'}$.

If $t$ is a local term, then from Observation 21 (iii) we learn that there exists trigger $(\delta, v)$ such that $w_1, w_2 \in \mathbb{F}_{v\delta_1} \cup \mathbb{F}_{v\delta_2}$. Now use Observation 33 and Lemma 34. □

### 6.5 From Corollary 30 to the First Main Lemma

In this subsection we construct $\mathcal{M}$ such that $(\mathcal{M}, \mathcal{R})$ is a mixed derivation. Define $\mathcal{M}$ as any oblivious derivation such that:

- $\mathcal{R}$ is a sub-sequence of $\mathcal{M}$, that is $\mathcal{M}$ respects the order of triggers in $\mathcal{R}$.
- If $\pi = (\rho, u)$ is a trigger such that $\pi \notin \mathcal{R}$, and if $\mu_1$ and $\mu_2$ are triggers in $\mathcal{R}$, which produce $btr(u\rho_1)$ and $btr(u\rho_1)$, then $\pi$ appears in $\mathcal{M}$ after both $\mu_1$ and $\mu_2$.

Note that each trigger that does not appear in $\mathcal{R}$ only needs to wait for at most two triggers from $\mathcal{M}$ to get its slot in $\mathcal{M}$. Therefore such a derivation $\mathcal{M}$ exists. It also follows easily from the construction, and from Corollary 30, that $(\mathcal{M}, \mathcal{R})$ is indeed a mixed derivation.

## 7 $\omega$-Sensitive-Path-Derivations

We now introduce yet another kind of derivation:

**Definition 35.** An *$\omega$-sensitive-path-derivation* is a pair $(\mathcal{M}, \mathcal{P})$ where $\mathcal{M} = \{\pi_n^{\mathcal{M}}\}_{n\in\mathbb{N}}$ is an infinite oblivious derivation over $\{\omega\}$, and $\mathcal{P}$ is an infinite restricted derivation over $\{\omega\}$, such that:

(i) $\mathcal{P}$ is a subsequence of $\mathcal{M}$;

(ii) if $\pi_n^{\mathcal{M}} \in \mathcal{P}$ for some $n$ then $\pi_n^{\mathcal{M}}$ is not blocked in $\mathbb{M}_n$;

(iii) the set $\{u : \exists\rho \in \mathcal{T}\ (\rho, u) \in \mathcal{P}\}$ is an infinite path.

Since we adopted the convention that the symbol $\mathbb{P}$ is kept for the set of all addresses created by $\mathcal{P}$, we will use $\mathbb{P}^0$ to denote the infinite path from (iii). It is easy to see that $\mathbb{P}^0 \subset \mathbb{P}$, and that $w\rho_\iota \in \mathbb{P}$ if and only if $w\rho_{\iota'} \in \mathbb{P}^0$ for some $\iota'$.

The above definition is similar to Definition 16 of Mixed Derivation, but with three differences: unlike the $\mathcal{R}$ in Definition 16, our $\mathcal{P}$ **does not** need to be fair. On the other hand, now we require that the set of the addresses of triggers in $\mathcal{R}$ (which we see as "sensitive", as opposed to the oblivious triggers in $\mathcal{M} \setminus \mathcal{R}$ ) is not some arbitrary forest, like in Definition 16, but something very specific, namely a path. The third difference is that $\omega$-sensitive-path-derivation does not start from some arbitrary instance $\mathbb{I}$ but from a single atom.

The following is our Second Main Lemma.

**Lemma 36** (Second Main Lemma)**.** *The following two statements are equivalent:*

1. *There exists a mixed derivation for some $\mathbb{I}$.*
2. *There exists an $\omega$-sensitive-path-derivation for some $\omega$.*

*Proof.* The proof can be found in the appendix. □

Notice that the expression *"$\omega$-sensitive-path-derivation"* in the lemma could not be replaced with *"an infinite restricted (but possibly not fair) chase derivation creating an infinite path."* That is, we cannot skip the conditions relating to $\mathcal{M}$. This is because, as Example 2 shows, such an infinite unfair derivation can easily exist even if $\mathcal{T} \in \mathrm{CT}_{\forall\forall}$.

Lemma 36 looks inconspicuous, but it constitutes a major step in our reduction from the problem $\mathrm{CT}_{\forall\forall}$ to Monadic Second Order Logic. And let us try to explain why.

$\mathbb{F}$ is (almost) a tree and the steps of a restricted derivation are its subtrees. So, one could think, nothing is more natural than the idea of employing MSOL, which is decidable on such trees, as a decision procedure for termination.

It is not that simple. Restricted chase is about the ordering of trigger applications, and it is not at all clear how this ordering could be expressed in MSOL. One could naively imagine the problem could be solved by a formula like *"For each $u \in \mathbb{F}$ there exists a set $\mathbb{G}$ of all addresses that were created before $u$"*. But this would not work, as there is no way to enforce consistency between such sets $\mathbb{G}$, that is to make sure that the sets produced by such universal quantifiers faithfully represent steps of the same derivation.

Lemma 36 (and Lemma 17) says that one can reduce $\mathrm{CT}_{\forall\forall}$ to the existence of a sort of infinite restricted chase, in which (due to its linear nature) the order of applications is undisputed and does not need to be guessed by our formula.

Of course, $\omega$-sensitive-path-derivations feature oblivious triggers which somehow need to interact with the sensitive ones but, as we will see in Lemma 39, they are manageable.

## 8 Employing Monadic Second Order Logic

In this last section of this paper we prove:

**Lemma 37** (Third Main Lemma)**.** *One can construct, for a given atom $\omega$, a sentence $\Omega_\omega$ of MSOL such that $\omega\mathbb{A}^* \models \Omega_\omega$ iff there exists an $\omega$-sensitive-path-derivation.*

Notice that, in view of Lemmas 36 and 17, and due to decidability of Monadic Second Order Logic on infinite trees, this will end the proof of Theorem 15.

For the rest of this paper we set $\mathbb{I} = \{\omega\}$, for a fixed $\omega$, so $\mathbb{F} \subseteq \omega\mathbb{A}^*$. Let us now characterize existence of $\omega$-sensitive-path-derivations in a language better suited for MSOL.

Let $\mathbb{P}^0$ be a path in $\mathbb{F}$, and (using notations from Section 7) let $\mathcal{P}$ be the (unique!) infinite derivation comprising all the triggers needed to create $\mathbb{P}^0$. Recall that the set $\mathbb{P}$ of addresses created by $\mathcal{P}$ contains, together with each address $u\rho_\iota$ in $\mathbb{P}^0$ also the address $u\rho_{\iota'}$ for $\iota' \neq \iota$. Notice also, that for each $n \in \mathbb{N}$ there is exactly one $u \in \mathbb{P}^0$ such that $|u| = n$.

**Definition 38.** Given an address $u \in \mathbb{P}^0$ we define:

- $\mathsf{bfr}(u)$ as the set of addresses $v \in \mathbb{P}$ such that $|v| \leq |u|$;
- $\mathsf{any}(u)$ as the set of addresses $v \notin \mathbb{P}$ without $u$ as prefix.

The idea is that $\mathsf{bfr}(u)$ are the addresses that must be created *before* $u$ in every oblivious derivation, and $\mathsf{any}(u)$ are the addresses that can be created at *any* time, before or after $u$.

For $u \in \mathbb{P}^0$ by $\pi_u$ denote the (unique) trigger of the form $(\rho, u)$ in $\mathcal{P}$. Also, $\mathsf{team}(v, v', \pi)$ is short for "$(v, v')$ is a blocking team for $\pi$." For $u \in \mathbb{P}^0$ define $\mathsf{fragile}(u)$ as the set:

$$\{w \in \mathbb{P}^0 \colon u \in \mathsf{bfr}(w) \wedge \exists v, s \in \mathsf{any}(u) \cup \mathsf{bfr}(w)\ \ \mathsf{team}(v, s, \pi_w)\}$$

**Lemma 39.** *The following statements are equivalent:*

1. *There exists an $\omega$-sensitive-path derivation*
2. *There exists an infinite path $\mathbb{P}^0 \subseteq \mathbb{F}$ s.t. for each $u \in \mathbb{P}^0$:*
   *(i) $\pi_u$ is not blocked by any team $(v,v')$ for $v,v' \in \mathsf{bfr}(u)$.*
   *(ii) The set $\mathsf{fragile}(u)$ is finite.*

Before we prove Lemma 39, let us explain how it implies Lemma 37. And this is actually by fairly routine reasoning.

**From Lemma 39 to Lemma 37.** Recall, that all the MSOL formula $\Omega_\omega$ will be able to see is $\omega\mathbb{A}^*$, and the relations $\rho_\iota$-$\mathsf{child}$, for each rule $\rho$ and $\iota$. Using this language we want to write a formula equivalent to Statement 2 of Lemma 39.

So, first of all, we need the formula to be able to guess, and verify (using monadic second order resources), which addresses $w$ are really in $\mathbb{F}$ and, if $w \in \mathbb{F}$, what is $\langle w \rangle$. There is a little problem here, since there are infinitely many atoms in $\langle \mathbb{F} \rangle$ and we can only afford to have finitely many unary predicates in $\Omega_\omega$. This is solved by guessing a function $[\_]$, which is a "compressed" version of $\langle \_ \rangle$. This guess can be made - using standard techniques - as the treewidth of $\langle \mathbb{F} \rangle$ is less than the arity of $\mathfrak{P}$.

Then, having defined this $[\_]$, we are able to write a formula[7] $\simeq_{i.j}$, with two free first order variables, such that $\omega\mathbb{A}^* \models u \simeq_{i.j} v$ if and only if $\langle u \rangle_i = \langle v \rangle_j$. Observation 21(i) and (ii) is helpful here.

With $\simeq_{i.j}$ at hand, writing $\Omega_\omega$ becomes a classroom exercise in MSOL programming. Indeed, the definitions of the notions of trigger and blocking team, as presented in Section 4, rely solely on equalities between terms - which we are now able to express. What then remains to be expressed in Statement 2 of Lemma 39, such as infinite paths or infinite sets, are part of the standard MSOL toolkit. The full argument can be found in the appendix.

Now, in order to finish the proof of Lemma 37, and hence also Theorem 15, we only need to prove Lemma 39.

### 8.1 Proof of Lemma 39

Let $\mathbb{P}^0$ be an arbitrary infinite path in $\mathbb{F}$, and let $\mathcal{P}$ relate to $\mathbb{P}^0$ in the usual way. In order to show Lemma 39 it is enough to prove that the following two conditions are equivalent:

A. There exists an oblivious derivation $\mathcal{M}$ such that $(\mathcal{M}, \mathcal{P})$ is an $\omega$-sensitive-path-derivation.

B. $\mathbb{P}^0$ satisfies conditions (i) and (ii) from Lemma 39 (2).

**B $\Rightarrow$ A.** Can be found in the appendix.

**A $\Rightarrow$ B.** Now suppose $(\mathcal{M}, \mathcal{P})$ is an $\omega$-sensitive-path-derivation, for some oblivious derivation $\mathcal{M}$.

First of all, notice that condition (i) is trivially satisfied for each $u$, since for a pair $(\mathcal{M}, \mathcal{P})$ to be an $\omega$-sensitive-path-derivation, $\mathcal{P}$ itself must be a restricted derivation.

[7] Strictly speaking, we have one such formula for each pair $i, j$ for $i$ and $j$ not greater than the arity of $\mathfrak{P}$.

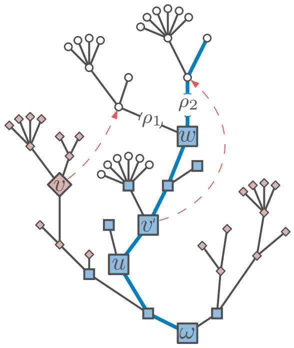


Figure 2: A visual guide to condition (ii) of Statement 2 from Lemma 39. Node $w$ appears in $\mathsf{fragile}(u)$ and trigger $\pi_w = (\rho, w)$ is blocked by a team of nodes $(v, v')$ represented with red dashed edges. Bold-blue path represents $\mathbb{P}^0$, blue square nodes are $\mathsf{bfr}(w)$, red diamond nodes are $\mathsf{any}(u)$.

We now focus on condition (ii). Let us fix some $u \in \mathbb{P}^0$. We need to show that the set $\mathsf{fragile}(u)$ is finite. The idea is that $u$ "controls" the communication between elements of $\mathsf{any}(u) \cup \mathsf{bfr}(u)$ and elements $w$ in $\mathbb{P}^0$ for which $u \in \mathsf{bfr}(w)$.

Let us now try to formalize this idea. Suppose $\pi_u = (\rho, u)$, which means that rule $\rho$ is the one that, applied to $\langle u \rangle$, produced the the element following $u$ on $\mathbb{P}^0$. Suppose also that $\rho$ is: $\forall \bar{x}\bar{y}.\, (\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}.\, \psi_1(\bar{y}, \bar{z}) \wedge \psi_2(\bar{y}, \bar{z}))$.

Then the following is of course true:

**Observation 40.** *If a term $t$ occurs in an atom of $\langle \mathsf{any}(u) \cup \mathsf{bfr}(u) \rangle$ and in some $\langle w \rangle$ such that $w \in \mathbb{P}^0$ and $u \in \mathsf{bfr}(w)$, then there is a frontier position $i$ in $\phi$ such that $t = \langle u \rangle_i$.*

Now, imagine you are the atom $\langle u \rangle$ and you want to somehow classify the atoms and the pairs in $\langle \mathsf{any}(u) \cup \mathsf{bfr}(u) \rangle$. This is what you would do:

**Definition 41.** For addresses $v, v' \in \mathsf{any}(u) \cup \mathsf{bfr}(u)$ we define $v \sim v'$ if and only if there is an isomorphism, from $\langle v \rangle$ to $\langle v' \rangle$, which is an identity on terms which occur in $\langle u \rangle$.

**Definition 42.** For addresses $v, v', s, s' \in \mathsf{any}(u) \cup \mathsf{bfr}(u)$ we define $(v, s) \approx (v', s')$ *iff* there is an isomorphism, from $(\langle v \rangle, \langle s \rangle)$ to $(\langle v' \rangle, \langle s' \rangle)$, which is an identity on terms in $\langle u \rangle$.

Both $\sim$ and $\approx$ are equivalence relations with finite index. Let now $w \in \mathbb{P}^0$ be such that $u \in \mathsf{bfr}(w)$. The next two observations follow easily from Observations 40 and 12:

**Observation 43.** *Suppose $v, v', s, s' \in \mathsf{any}(u) \cup \mathsf{bfr}(u)$ and $(v, s) \approx (v', s')$. Suppose also that $(v, s)$ is a blocking team for $\pi_w$. Then $(v', s')$ is also a blocking team for $\pi_w$.*

**Observation 44.** *Let $v, v' \in \mathsf{any}(u) \cup \mathsf{bfr}(u)$ and $v \sim v'$. Let also $s \in \mathbb{P}$ be such that $|u| \leq |s| \leq |w|$ and $(v, s)$ is a blocking team for $\pi_w$.Then $(v', s)$ is a blocking team for $\pi_w$.*

The above holds because whenever $\langle v \rangle_i = \langle s \rangle_j$ for some positions $i$ and $j$, we have that $\langle v \rangle_i$ is a term of $\langle u \rangle$ and hence (by Definition 41) we get $\langle v \rangle_i = \langle v' \rangle_i$.

We are going to present the set $\mathsf{fragile}(u)$ as a certain union of sets. But first we need:

**Definition 45.** Suppose $v, s \in \mathsf{any}(u) \cup \mathsf{bfr}(u)$. Then:

$\mathsf{harm}_2(v, s) = \{w \in \mathbb{P}^0 : u \in \mathsf{bfr}(w) \ \wedge \ \mathsf{team}(v, s, \pi_w)\}$

$\mathsf{harm}_1(v) =$
$\{w \in \mathbb{P}^0 : u \in \mathsf{bfr}(w) \ \wedge \ \exists_{s \in \mathsf{bfr}(w) \setminus \mathsf{bfr}(u)} \ \mathsf{team}(v, s, \pi_w)\}$

Our next observation (still assuming that $(\mathcal{M}, \mathcal{P})$ is an $\omega$-sensitive-path-derivation) is that:

**Lemma 46.** *The sets $\mathsf{harm}_1(v)$ and $\mathsf{harm}_2(v, s)$ are finite for all $v, s \in \mathsf{any}(u) \cup \mathsf{bfr}(u)$.*

*Proof.* Proof can be found in the appendix. □

It follows directly from the definition of $\mathsf{fragile}(u)$ that:

**Observation 47.** *The set $\mathsf{fragile}(u)$ is equal to:*

$\bigcup_{v,s \in \mathsf{any}(u) \cup \mathsf{bfr}(u)} \mathsf{harm}_2(v, s) \cup \bigcup_{v \in \mathsf{any}(u) \cup \mathsf{bfr}(u)} \mathsf{harm}_1(v)$

Using Observations 43 and 44 it is also easy to see that:

**Observation 48.** *If $v \sim v'$ then $\mathsf{harm}_1(v) = \mathsf{harm}_1(v')$. If $(v, s) \approx (v', s')$ then $\mathsf{harm}_2(v, s) = \mathsf{harm}_2(v', s')$.*

Now, since $\sim$ and $\approx$ are equivalence relations of finite index, it follows from Observation 48 that the union of sets in Observation 47 is in fact a finite union. Together with Lemma 46, this means, that $\mathsf{fragile}(u)$ is a finite union of finite sets and hence a finite set.

# 9 Conclusions and Future Work

The next natural step is the decidability of the same problem for multi-headed guarded rules.

**Definition 49.** An existential rule is *guarded* if its body contains an atom, the *guard*, featuring all variables of the body.

**Conjecture 50.** *Membership in $\mathrm{CT}_{\forall\forall}$ for sets of multi-headed guarded existential rules is decidable.*

However, it is still unclear how the techniques used for the linear case extend to the guarded case, since they rely heavily on the forest structure of the chase.

One could still define a forest structure in the guarded case, by setting $\langle w\rho_\iota \rangle$ as the $\iota$-th atom resulting from the application of rule $\rho$ with guard $\langle w \rangle$, but this induces other problems. Indeed, the creation of $\langle w\rho_\iota \rangle$ requires some side-atoms beside the guard, which may have addresses far from $w$. Consider for instance the guarded rule

$$E(x, y), A(y) \rightarrow \exists z.\, E(y, z)$$

and suppose that some vertices $v$ and $w$ are labelled with $A(b)$ and $E(a, b)$, respectively. Then, $w$ has a child $wa$ labelled $E(b, c)$, but any chase sequence must create $v$ before $wa$. We thus need to enforce a total order on distant elements of the forest, which cannot be done by MSOL means in any obvious way.

MSOL techniques have already been successful at dealing with the single-headed guarded case (Gogacz, Marcinkowski, and Pieris 2023), but in this case the tree considered is a join tree, which always exists in the single-headed case but not in the multi-headed case. A natural generalization of join trees would be tree decompositions, which could be used since the chase with guarded rules admit finite treewidth. However, it is even less clear how the techniques developed for the linear case could apply on a tree decomposition, especially taking into account the fact that nodes are now labelled with sets of atoms and not just single atoms, and that this does not solve the issue of having to enforce an order on distant elements of the forest.

## Acknowledgments

Jerzy Marcinkowski and Piotr Ostropolski-Nalewaja were supported by Polish National Science Center (NCN) grant 2022/45/B/ST6/00457

On TU Dresden side, this work was partly supported by DFG (German Research Foundation) in project 389792660 (TRR 248, Center for Perspicuous Systems) and in the CeTI Cluster of Excellence; by the Bundesministerium für Bildung und Forschung (BMBF) under European ITEA project 01IS21084 (InnoSale, Innovating Sales and Planning of Complex Industrial Products Exploiting Artifical Intelligence, in the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), and in project 13GW0552B (KIMEDS); by BMBF and DAAD (German Academic Exchange Service) in project 57616814 (SECAI, School of Embedded and Composite AI); and by the Center for Advancing Electronics Dresden (cfaed).

## References

Aho, A. V.; Sagiv, Y.; and Ullman, J. D. 1979. Efficient optimization of a class of relational expressions. *ACM Trans. Database Syst.* 4(4):435–454.

Bednarczyk, B.; Ferens, R.; and Ostropolski-Nalewaja, P. 2020. All-instances oblivious chase termination is undecidable for single-head binary tgds. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 1719–1725. ijcai.org.

Benedikt, M.; Konstantinidis, G.; Mecca, G.; Motik, B.; Papotti, P.; Santoro, D.; and Tsamoura, E. 2017. Benchmarking the chase. In Sallinger, E.; den Bussche, J. V.; and Geerts, F., eds., *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, 37–52. ACM.

Calautti, M., and Pieris, A. 2019. Oblivious chase termination: The sticky case. In *ICDT*, To appear.

Calautti, M.; Gottlob, G.; and Pieris, A. 2015. Chase termination for guarded existential rules. In *PODS*, 91–103.

Calì, A.; Gottlob, G.; and Kifer, M. 2013. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.* 48:115–174.

Carral, D.; Gerlach, L.; Larroque, L.; and Thomazo, M. 2025. Restricted chase termination: You want more than fairness. *Proc. ACM Manag. Data* 3(2 (PODS)).

Deutsch, A., and Tannen, V. 2003. Reformulation of XML queries and constraints. In *ICDT*, 225–241.

Deutsch, A.; Nash, A.; and Remmel, J. B. 2008. The chase revisisted. In *PODS*, 149–158.

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

Gerlach, L., and Carral, D. 2023. Do repeat yourself: Understanding sufficient conditions for restricted chase non-termination (technical report). In Marquis, P.; Son, T. C.; and Kern-Isberner, G., eds., *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning (KR 2023)*, volume 20 of *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 301–310. International Joint Conferences on Artificial Intelligence Organization.

Gogacz, T., and Marcinkowski, J. 2014. All-instances termination of chase is undecidable. In *ICALP*, 293–304.

Gogacz, T.; Marcinkowski, J.; and Pieris, A. 2023. Uniform restricted chase termination. *SIAM J. Comput.* 52(3):641–683.

Grau, B. C.; Horrocks, I.; Krötzsch, M.; Kupke, C.; Magka, D.; Motik, B.; and Wang, Z. 2013. Acyclicity notions for existential rules and their application to query answering in ontologies. *J. Artif. Intell. Res.* 47:741–808.

Greco, S.; Spezzano, F.; and Trubitsyna, I. 2011. Stratification criteria and rewriting techniques for checking chase termination. *PVLDB* 4(11):1158–1168.

Ivliev, A.; Gerlach, L.; Meusel, S.; Steinberg, J.; and Krötzsch, M. 2024. Nemo: Your friendly and versatile rule reasoning toolkit. In Marquis, P.; Ortiz, M.; and Pagnucco, M., eds., *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR 2024)*, 743–754. IJCAI Organization.

Leclère, M.; Mugnier, M.-L.; Thomazo, M.; and Ulliana, F. 2019. On chase termination for linear existential rules. In *ICDT*, To appear.

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

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

Urbani, J.; Krötzsch, M.; Jacobs, C. J. H.; Dragoste, I.; and Carral, D. 2018. Efficient model construction for horn logic with vlog - system description. In Galmiche, D.; Schulz, S.; and Sebastiani, R., eds., *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings*, volume 10900 of *Lecture Notes in Computer Science*, 680–688. Springer.