# A Single Approach to Decide Chase Termination on Linear Existential Rules

Michel Leclère, Marie-Laure Mugnier, Michaël Thomazo, Federico Ulliana

# A Single Approach to Decide Chase Termination on Linear Existential Rules

Michel Leclère[1], Marie-Laure Mugnier[1], Michaël Thomazo[2], and Federico Ulliana[1]

[1]University of Montpellier, CNRS, Inria, LIRMM, France
[2]Inria, DI ENS, ENS, CNRS, PSL University, France

October 4, 2018

**Abstract**

Existential rules, long known as tuple-generating dependencies in database theory, have been intensively studied in the last decade as a powerful formalism to represent ontological knowledge in the context of ontology-based query answering. A knowledge base is then composed of an instance that contains incomplete data and a set of existential rules, and answers to queries are logically entailed from the knowledge base. This brought again to light the fundamental chase tool, and its different variants that have been proposed in the literature. It is well-known that the problem of determining, given a chase variant and a set of existential rules, whether the chase will halt on any instance, is undecidable. Hence, a crucial issue is whether it becomes decidable for known subclasses of existential rules. In this work, we consider linear existential rules, a simple yet important subclass of existential rules that generalizes inclusion dependencies. We show the decidability of the *all instance* chase termination problem on linear rules for three main chase variants, namely *semi-oblivious*, *restricted* and *core* chase. To obtain these results, we introduce a novel approach based on so-called derivation trees and a single notion of forbidden pattern. Besides the theoretical interest of a unified approach and new proofs, we provide the first positive decidability results concerning the termination of the restricted chase, proving that chase termination on linear existential rules is decidable for both versions of the problem: Does *every* fair chase sequence terminate? Does *some* fair chase sequence terminate?

## 1 Introduction

The chase procedure is a fundamental tool for solving many issues involving tuple-generating dependencies, such as data integration [Len02], data-exchange [FKMP05], query answering using views [Hal01] or query answering on probabilistic databases [OHK09]. In the last decade, tuple-generating dependencies raised a renewed interest under the name of *existential rules* for the problem known as ontology-based query

answering. In this context, the aim is to query a knowledge base $(I, \Sigma)$, where $I$ is an instance and $\Sigma$ is a set of existential rules (see e.g. the survey chapters [CGL09, MT14]). In more classical database terms, this problem can be recast as querying an instance $I$ under incomplete data assumption, provided with a set of constraints $\Sigma$, which are tuple-generating dependencies. The chase is a fundamental tool to solve dependency-related problems as it allows one to compute a (possibly infinite) *universal model* of $(I, \Sigma)$, *i.e.*, a model that can be homomorphically mapped to any other model of $(I, \Sigma)$. Hence, the answers to a conjunctive query (and more generally to any kind of query closed by homomorphism) over $(I, \Sigma)$ can be defined by considering solely this universal model.

Several variants of the chase have been introduced, and we focus in this paper on the main ones: semi-oblivious [Mar09] (aka skolem [Mar09]), restricted [BV81, FKMP05] (aka standard [One12]) and core [DNR08]. It is well known that all of these produce homomorphically equivalent results but terminate for increasingly larger subclasses of existential rules.

Any chase variant starts from an instance and exhaustively performs a sequence of rule applications according to a redundancy criterion which characterizes the variant itself. The question of whether a chase variant terminates on *all instances* for a given set of existential rules is known to be undecidable when there is no restriction on the kind of rules [BLMS11, GM14]. A number of *sufficient* syntactic conditions for termination have been proposed in the literature for the semi-oblivious chase (see e.g. [One12, GHK$^+$13, Roc16] for syntheses), as well as for the restricted chase [CDK17] (note that the latter paper also defines a sufficient condition for non-termination). However, only few positive results exist regarding the termination of the chase on specific classes of rules. Decidability was shown for the semi-oblivious chase on guarded-based rules (linear rules, and their extension to (weakly-)guarded rules) [CGP15]. Decidability of the core chase termination on guarded rules for a fixed instance was shown in [Her12].

In this work, we provide new insights on the chase termination problem for *linear* existential rules, a simple yet important subclass of guarded existential rules, which generalizes inclusion dependencies [Fag81] and practical ontological languages [CGL12]. Precisely, the question of whether a chase variant terminates on all instances for a set of linear existential rules is studied in two fashions:

- does *every* (fair) chase sequence terminate?
- does *some* (fair) chase sequence terminate?

It is well-known that these two questions have the same answer for the semi-oblivious and the core chase variants, but not for the restricted chase. Indeed, this last one may admit both terminating and non-terminating sequences over the same knowledge base. We show that the termination problem is decidable for linear existential rules, whether we consider any version of the problem and any chase variant.

We study chase termination by exploiting in a novel way a graph structure, namely the *derivation tree*, which was originally introduced to solve the ontology-based (conjunctive) query answering problem for the family of greedy-bounded treewidth sets of existential rules [BMRT11, Tho13], a class that generalizes guarded-based rules and in

particular linear rules. We first use derivation trees to show the decidability of the termination problem for the semi-oblivious and restricted chase variants, and then generalize them to *entailment trees* to show the decidability of termination for the core chase. For any chase variant we consider, we adopt the same high-level procedure: starting from a finite set of canonical instances (representative of all possible instances), we build a (set of) tree structures for each canonical instance, while forbidding the occurrence of a specific pattern, we call *unbounded-path witness*. The built structures are finite thanks to this forbidden pattern, and this allows us to decide if the chase terminates on the associated canonical instance. By doing so, we obtain a uniform approach to study the termination of several chase variants, that we believe to be of theoretical interest per se. The derivation tree is moreover a simple structure and the algorithms built on it are likely to lead to an effective implementation. Let us also point out that our approach is constructive: if the chase terminates on a given instance, the algorithm that decides termination actually computes the result of the chase (or a superset of it in the case of the core chase), otherwise it pinpoints a forbidden pattern responsible for non-termination.

Besides providing new theoretical tools to study chase termination, we obtain the following results for linear existential rules:

- a new proof of the decidability of the semi-oblivious chase termination, building on different objects than the previous proof provided in [CGP15]; we show that our algorithm provides the same complexity upper-bound;
- the decidability of the restricted chase termination, for both versions of the problem, i.e., termination of all (fair) chase sequences and termination of some (fair) chase sequence; to the best of our knowledge, these are the first positive results on the decidability of the restricted chase termination;
- a new proof of the decidability of the core chase termination, with different objects than previous work reported in [Her12]; although this latter paper solves the question of the core chase termination given a *single* instance, the results actually allow to infer the decidability of the *all* instance version of the problem, by noticing that only a finite number of instances need to be considered (see the next section).

The paper is organized as follows. After introducing some preliminary notions (Section 2), we define the main components of our framework, namely derivation trees and unbounded-path witnesses (Section 3). We build on these objects to prove the decidability of the semi-oblivious and restricted chase termination (Section 4). Finally, we generalize derivation-trees to entailment trees and use them to prove the decidability of the core chase termination (Section 5). Detailed proofs are provided in the appendix.

# 2 Preliminaries

We consider a logical *vocabulary* composed of a finite set of predicates and an infinite set of constants. An *atom* $\alpha$ has the form $r(t_1, \ldots, t_n)$ where $r$ is a predicate of arity $n$ and the $t_i$ are terms (i.e., variables or constants). We denote by $\mathit{terms}(\alpha)$ (resp. $\mathit{vars}(\alpha)$) the set of terms (resp. variables) in $\alpha$ and extend the notations to a set of

atoms. A *ground* atom does not contain any variable. It is convenient to identify the existential closure of a conjunction of atoms with the set of these atoms. An *instance* is a set of (non-necessarily ground) atoms, which is finite unless otherwise specified. Abusing terminology, we will often see an instance as its isomorphic model.

Given two sets of atoms $S$ and $S'$, a *homomorphism* from $S'$ to $S$ is a substitution $\pi$ of $\mathit{vars}(S')$ by $\mathit{terms}(S)$ such that $\pi(S') \subseteq S$. It holds that $S \models S'$ (where $\models$ denotes classical logical entailment) iff there is a homomorphism from $S'$ to $S$. An endomorphism of $S$ is a homomorphism from $S$ to itself. A set of atoms is a *core* if it admits only injective endomorphisms. Any finite set of atoms is logically equivalent to one of its subsets that is a core, and this core is unique up to isomomorphism (i.e., bijective variable renaming). Given sets of atoms $S$ and $S'$ such that $S \cap S' \neq \emptyset$, we say that $S$ *folds* onto $S'$ if there is a homomorphism $\pi$ from $S$ to $S'$ such that $\pi$ is the identity on $S \cap S'$. The homomorphism $\pi$ is called a *folding*. In particular, it is well-known that any set of atoms *folds* onto its core.

An existential rule (or simply *rule*) is of the form $\sigma = \forall \mathbf{x} \forall \mathbf{y}.[\mathrm{body}(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z}.\mathrm{head}(\mathbf{x}, \mathbf{z})]$ where $\mathrm{body}(\mathbf{x}, \mathbf{y})$ and $\mathrm{head}(\mathbf{x}, \mathbf{z})$ are non-empty conjunctions of atoms on variables, respectively called the *body* and the *head* of the rule, also denoted by $\mathrm{body}(\sigma)$ and $\mathrm{head}(\sigma)$, and $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ are pairwise disjoint tuples of variables. The variables of $\mathbf{z}$ are called *existential variables*. The variables of $\mathbf{x}$ form the *frontier* of $\sigma$, which is also denoted by $\mathrm{fr}(\sigma)$. For brevity, we will omit universal quantifiers in the examples. A *knowledge base* (KB) is of the form $\mathcal{K} = (I, \Sigma)$, where $I$ is an instance and $\Sigma$ is a finite set of existential rules.

A rule $\sigma = \mathrm{body}(\sigma) \rightarrow \mathrm{head}(\sigma)$ is *applicable* to an instance $I$ if there is a homomorphism $\pi$ from $\mathrm{body}(\sigma)$ to $I$. The pair $(\sigma, \pi)$ is called a *trigger* for $I$. The result of the application of $\sigma$ according to $\pi$ on $I$ is the instance $I' = I \cup \pi^s(\mathrm{head}(\sigma))$, where $\pi^s$ (here $s$ stands for *safe*) extends $\pi$ by assigning a distinct fresh variable (also called a *null*) to each existential variable. We also say that $I'$ is obtained by *firing* the trigger $(\sigma, \pi)$ on $I$. By $\pi_{|\mathrm{fr}(\sigma)}$ we denote the restriction of $\pi$ to the domain $\mathrm{fr}(\sigma)$.

**Definition 1 Derivation.** *A* $\Sigma$-derivation *(or simply* derivation *when* $\Sigma$ *is clear from the context) from an instance* $I = I_0$ *to an instance* $I_n$ *is a sequence* $I_0, (\sigma_1, \pi_1), I_1 \ldots, I_{n-1}, (\sigma_n, \pi_n), I_n$*, such that for all* $1 \leq i \leq n$*:* $\sigma_i \in \Sigma$*,* $(\sigma_i, \pi_i)$ *is a trigger for* $I_{i-1}$*,* $I_i$ *is obtained by firing* $(\sigma_i, \pi_i)$ *on* $I_{i-1}$*, and* $I_i \neq I_{i-1}$*. We may also denote this derivation by the associated sequence of instances* $(I_0, \ldots, I_n)$ *when the triggers are not needed. The notion of derivation can be naturally extended to an* infinite *sequence.*

We briefly introduce below the main chase variants and refer to [One12] for a detailed presentation.

The *semi-oblivious* chase prevents several applications of the same rule through the same mapping of its frontier. Given a derivation from $I_0$ to $I_i$, a trigger $(\sigma, \pi)$ for $I_i$ is said to be *active according to the semi-oblivious criterion*, if there is no trigger $(\sigma_j, \pi_j)$ in the derivation with $\sigma = \sigma_j$ and $\pi_{|\mathrm{fr}(\sigma)} = \pi_{j|\mathrm{fr}(\sigma_j)}$. The *restricted* chase performs a rule application only if the added set of atoms is not redundant with respect to the current instance. Given a derivation from $I_0$ to $I_i$, a trigger $(\sigma, \pi)$ for $I_i$ is said to be *active according to the restricted criterion* if $\pi$ cannot be extended to a homomorphism from $(\mathrm{body}(\sigma) \cup \mathrm{head}(\sigma))$ to $I_i$ (equivalently, $\pi^s(\mathrm{head}(\sigma))$ does not fold onto $I_i$). A

*semi-oblivious (resp. restricted) chase sequence* of $I$ with $\Sigma$ is a possibly infinite $\Sigma$-derivation from $I$ such that each trigger $(\sigma_i, \pi_i)$ in the derivation is active according to the semi-oblivious (resp. restricted) criterion.

Furthermore, a (possibly infinite) chase sequence is required to be *fair*, which means that a possible rule application is not indefinitely delayed. Formally, if some $I_i$ in the derivation admits an active trigger $(\sigma, \pi)$, then there is $j > i$ such that, either $I_j$ is obtained by firing $(\sigma, \pi)$ on $I_{j-1}$, or $(\sigma, \pi)$ is not an active trigger anymore on $I_j$. A *terminating* chase sequence is a finite fair sequence.

In its original definition [DNR08], the *core* chase proceeds in a breadth-first manner, and, at each step, first fires in parallel all active triggers according to the restricted chase criterion, then computes the core of the result. Alternatively, to bring the definition of the core chase closer to the above definitions of the semi-oblivious and restricted chases, one can define a *core chase sequence* as a possibly infinite sequence $I_0, (\sigma_1, \pi_1), I_1, \ldots$, alternating instances and triggers, such that each instance $I_i$ is obtained from $I_{i-1}$ by first firing the active trigger $(\sigma_i, \pi_i)$ according to the restricted criterion, then computing the core of the result. An instance admits a terminating core chase sequence in that sense if and only if the core chase as originally defined terminates on that instance.

For the three chase variants, fair chase sequences compute a (possibly infinite) *universal model* of the KB, but only the core chase stops if and only if the KB has a *finite* universal model.

It is well-known that, for the semi-oblivious and the core chase, if there is a terminating chase sequence from an instance $I$ then all fair sequences from $I$ are terminating. This is not the case for the restricted chase, since the order in which rules are applied has an impact on termination, as illustrated by Example 1.

**Example 1.** *Let $\Sigma = \{\sigma_1, \sigma_2\}$, with $\sigma_1 = p(x, y) \rightarrow \exists z\, p(y, z)$ and $\sigma_2 = p(x, y) \rightarrow p(y, y)$. Let $I = p(a, b)$. The KB $(I, \Sigma)$ has a finite universal model, for example, $I^* = \{p(a, b), p(b, b)\}$. The semi-oblivious chase does not terminate on $I$ as $\sigma_1$ is applied indefinitely, while the core chase terminates after one breadth-first step and returns $I^*$. The restricted chase has a terminating sequence, for example, $(\sigma_2, \{x \mapsto a, y \mapsto b\})$, which yields $I^*$ as well, but it also has infinite fair sequences, for example, the breadth-first sequence that applies $\sigma_1$ before $\sigma_2$ at each step.*

We study the following problems for the semi-oblivious, restricted and core chase variants:

- *(All instance) all sequence termination:* Given a set of rules $\Sigma$, is it true that, for any instance, all fair sequences are terminating?
- *(All instance) one sequence termination:* Given a set of rules $\Sigma$, is it true that, for any instance, there is a terminating sequence?

Note that, according to the terminology of [GO18], these problems can be recast as deciding whether, for a chase variant, a given set of rules belongs to the class $\text{CT}_{\forall\forall}$ or $\text{CT}_{\forall\exists}$, respectively.

An existential rule is called *linear* if its body and its head are both composed of a single atom (e.g., [CGL12]). Linear rules generalize *inclusion dependencies* [Fag81] by allowing several occurrences of the same variable in an atom. They also generalize positive inclusions in the description logic DL-Lite$_{\mathcal{R}}$ (the formal basis of the web ontological language OWL2 QL) [CDL$^+$07], which can be seen as inclusion dependencies restricted to unary and binary predicates.

Note that the restriction of existential rules to rules with a single head is often made in the literature, considering that any existential rule with a complex head can be decomposed into several rules with a single head, by introducing a fresh predicate for each rule. However, while this translation preserves the termination of the semi-oblivious chase, it is not the case for the restricted and the core chases. Hence, considering linear rules with a complex head would require to extend the techniques developed in this paper.

To simplify the presentation, we assume in the following that each rule frontier is of size at least one. This assumption is made without loss of generality. [1]

We first point out that the termination problem on linear rules can be recast by considering solely instances that contain a single atom (as already remarked in several contexts).

**Proposition 1.** *Let $\Sigma$ be a linear set of rules. The semi-oblivious (resp. restricted, core) chase terminates on all instances if and only if it terminates on all singleton instances.*

We will furthermore rely on the following notion of the type of an atom.

**Definition 2 Type of an atom.** *The* type of an atom $\alpha = r(t_1, \ldots, t_n)$*, denoted by* type$(\alpha)$*, is the pair $(r, \mathcal{P})$ where $\mathcal{P}$ is the partition of $\{1, \ldots, n\}$ induced by term equality (i.e., $i$ and $j$ are in the same class of $\mathcal{P}$ iff $t_i = t_j$).*

Note that there are finitely (more specifically, exponentially) many types for a given vocabulary.

If two atoms $\alpha$ and $\alpha'$ have the same type, then there is a *natural mapping* from $\alpha$ to $\alpha'$, denoted by $\varphi_{\alpha \to \alpha'}$, and defined as follows: it is a bijective mapping from terms$(\alpha)$

[1] For instance, it can always be ensured by adding a position to all predicates, which is filled by the same fresh constant in the initial instance, and by a new frontier variable in each rule.

to $\mathit{terms}(\alpha')$, that maps the $i$-th term of $\alpha$ to the $i$-th term of $\alpha'$. Note that $\varphi_{\alpha\to\alpha'}$ may not be an isomorphism, as constants from $\alpha$ may not be mapped to themselves. However, if $(\sigma, \pi)$ is a trigger for $\{\alpha\}$, then $(\sigma, \varphi_{\alpha\to\alpha'} \circ \pi)$ is a trigger for $\{\alpha'\}$, as there are no constants in the considered rules.

Together with Proposition 1, this implies that one can check all instance all sequence termination by checking all sequence termination on a finite set of instances, called *canonical instances*: for each type, there is exactly one canonical instance that has this type.

We will consider different kinds of tree structures, which have in common to be *trees of bags*: these are rooted trees, whose nodes, called *bags*, are labeled by an atom.[2] We define the following notations for any node $B$ of a tree of bags $\mathcal{T}$:

- $\mathit{atom}(B)$ is the label of $B$;
- $\mathit{terms}(B) = \mathit{terms}(\mathit{atom}(B))$ is the set of terms of $B$;
- $\mathit{terms}(B)$ is divided into two sets of terms, those *generated* in $B$, denoted by $\mathit{generated}(B)$, and those shared with its parent, denoted by $\mathit{shared}(B)$; precisely, $\mathit{terms}(B) = \mathit{shared}(B) \cup \mathit{generated}(B)$, $\mathit{shared}(B) \cap \mathit{generated}(B) = \emptyset$, and if $B$ is the root of $\mathcal{T}$, then $\mathit{generated}(B) = \mathit{terms}(B)$ (hence $\mathit{shared}(B) = \emptyset$), otherwise $B$ has a parent $B_p$ and $\mathit{generated}(B) = \mathit{terms}(B) \setminus \mathit{terms}(B_p)$ (hence, $\mathit{shared}(B) = \mathit{terms}(B_p) \cap \mathit{terms}(B)$).

We denote by $\mathit{atoms}(\mathcal{T})$ the set of atoms that label the bags in $\mathcal{T}$.

Finally, we recall some classical mathematical notions. A *subsequence* $S'$ of a sequence $S$ is a sequence that can be obtained from $S$ by deleting some (or no) elements without changing the order of the remaining elements. The *arity* of a tree is the maximal number of children for a node. A *prefix* $T'$ of a tree $T$ is a tree that can be obtained from $T$ by repeatedly deleting some (or no) leaves of $T$.

# 3 Derivation Trees

A classical tool to reason about the chase is the so-called *chase graph* (see e.g., [CGL12]), which is the directed graph consisting of all atoms that appear in the considered derivation, and with an arrow from a node $n_1$ to a node $n_2$ iff $n_2$ is created by a rule application on $n_1$ and possibly other atoms. [3] In the specific case of KBs of the form $(\{\alpha\}, \Sigma)$, where $\alpha$ is an atom and $\Sigma$ is a set of linear rules, the chase graph is a tree. We recall below its definition in this specific case, in order to emphasize its differences with another tree, called *derivation tree*, on which we will actually rely.

**Definition 3 Chase Graph for Linear Rules.** *Let $I$ be a singleton instance, $\Sigma$ be a set of linear rules and $I = I_0, (\sigma_1, \pi_1), I_1 \ldots, I_{n-1}, (\sigma_n, \pi_n), I_n$ be a semi-oblivious $\Sigma$-derivation from $I$. The* chase graph *(also called* chase tree*) assigned to $S$ is a tree of bags built as follows:*

[2] Furthermore the trees we will consider are decomposition trees of the associated set of atoms. That is why we use the classical term of *bag* to denote a node.

[3] Note that the chase graph in [DNR08] is a different notion.

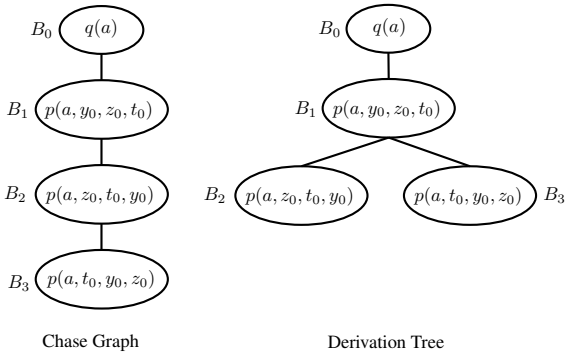


Figure 1: Chase Graph and Derivation Tree of Example 2

- *the set of bags is in bijection with $I_n$ via the labeling function* atom()*;*
- *the set of edges is in bijection with the set of triggers in $S$ and is built as follows: for each trigger $(\sigma_i, \pi_i)$ in $S$, there is an edge $(B, B')$ with* atom$(B) = \pi_i(\mathrm{body}(\sigma_i))$ *and* atom$(B') = \pi_i^s(\mathrm{head}(\sigma_i))$.

**Example 2.** *Let $I = q(a)$ and $\Sigma = \{\sigma_1, \sigma_2\}$ where $\sigma_1 = q(x) \to \exists y \exists z \exists t\ p(x, y, z, t)$ and $\sigma_2 = p(x, y, z, t) \to p(x, z, t, y)$. Let $S = I, (\sigma_1, \pi_1), I_1, (\sigma_2, \pi_2), I_3, (\sigma_2, \pi_3), I_3$ with $\pi_1 = \{x \mapsto a\}$, $\pi_1^s(\mathrm{head}(\sigma_1)) = p(a, y_0, z_0, t_0)$, $\pi_2 = \{x \mapsto a, y \mapsto y_0, z \mapsto z_0, t \mapsto t_0\}$ and $\pi_3 = \{x \mapsto a, y \mapsto z_0, z \mapsto t_0, t \mapsto y_0\}$. The chase graph associated with $S$ is a path of four nodes as represented in Figure 1.*

To check termination of a chase variant on a given KB $(\{\alpha\}, \Sigma)$, the general idea is to build a tree of bags associated with the chase on this KB in such a way that the occurrence of some forbidden pattern indicates that a path of unbounded length can be developed, hence the chase does not terminate. The forbidden pattern is composed of two distinct nodes such that one is an ancestor of the other and, intuitively speaking, these nodes "can be extended in similar ways", which leads to an arbitrarily long path that repeats the pattern.

Two atoms with the same type admit the same rule triggers, however, within a derivation, the same rule applications cannot necessarily be performed on both of them because of the presence of other atoms (this is true already for datalog rules, since the same atom is never produced twice). Hence, on the one hand we will specialize the notion of type, into that of a *sharing type*, and, on the other hand, adopt another tree structure, called a *derivation tree*, in which two nodes with the same sharing type have the required similar behavior.

**Definition 4 Sharing type and Twins.** *Given a tree of bags, the* sharing type *of a bag $B$ is a pair* (type(atom$(B)$), $P$) *where $P$ is the set of positions in* atom$(B)$ *in which a term of* shared$(B)$ *occurs. We denote the fact that two bags $B$ and $B'$ have the same sharing type by $B \equiv_{st} B'$. Furthermore, we say that two bags $B$ and $B'$ are* twins

*if they have the same sharing type, the same parent $B_p$ and if the natural mapping $\varphi_{\mathsf{atom}(B)\to\mathsf{atom}(B')}$ is the identity on the terms of* $\mathsf{atom}(B_p)$.

We can now specify the forbidden pattern that we will consider: it is a pair of two distinct nodes with the same sharing type, such that one is an ancestor of the other.

**Definition 5 Unbounded-Path Witness.** *An* unbounded-path witness *(UPW) in a derivation tree is a pair of distinct bags $(B, B')$ such that $B$ and $B'$ have the same sharing type and $B$ is an ancestor of $B'$.*

As explained below on Example 2, the chase graph is not the appropriate tool to use this forbidden pattern as a witness of chase non-termination.

*Example 2 (cont'd).* $B_1$, $B_2$ and $B_3$ have the same classical type, $t = (p, \{\{1\}, \{2\}, \{3\}, \{4\}\})$. The sharing type of $B_1$ is $(t, \{1\})$, while $B_2$ and $B_3$ have the same sharing type $(t, \{1, 2, 3, 4\})$. $B_2$ and $B_3$ fulfill the condition of the forbidden pattern, however it is easily checked that any derivation that extends this derivation is finite.

Derivation trees were introduced as a tool to define the *greedy bounded treewidth set (gbts)* family of existential rules [BMRT11, Tho13]. A derivation tree is associated with a derivation, however it does not have the same structure as the chase graph. The fundamental reason is that, when a rule $\sigma$ is applied to an atom $\alpha$ via a homomorphism $\pi$, the newly created bag is not necessarily attached in the tree as a child of the bag labeled by $\alpha$. Instead, it is attached as a child of the *highest* bag in the tree labeled by an atom that contains $\pi(\mathrm{fr}(\sigma))$, the image by $\pi$ of the frontier of $\sigma$ (note that $\pi(\mathrm{fr}(\sigma))$ remains the set of terms shared between the new bag and its parent).

In the following definition, a derivation tree is not associated with *any* derivation, but with a semi-oblivious derivation, which has the advantage of yielding trees with bounded arity (Proposition 12 in the Appendix). This is appropriate to study the termination of the semi-oblivious chase, and later the restricted chase, as a restricted chase sequence is a specific semi-oblivious chase sequence.

**Definition 6 Derivation Tree.** *Let $I = \{\alpha\}$ be a singleton instance, $\Sigma$ be a set of linear rules, and $S = I_0, (\sigma_1, \pi_1), I_1, \ldots, (\sigma_n, \pi_n), I_n$ be a semi-oblivious $\Sigma$-derivation. The* derivation tree *assigned to $S$ is a tree of bags $\mathcal{T}$ built as follows:*

- *the root of the tree, $B_0$, is such that $\mathsf{atom}(B_0) = \alpha$;*
- *for each trigger $(\sigma_i, \pi_i)$, $0 < i \leq n$, let $B_i$ be the bag such that $\mathsf{atom}(B_i) = \pi_i^s(\mathrm{head}(\sigma_i))$. Let $j$ be smallest integer such that $\pi_i(\mathrm{fr}(\sigma_i)) \subseteq \mathsf{terms}(B_j)$: $B_i$ is added as a child to $B_j$.*

*By extension, we say that a derivation tree $\mathcal{T}$ is* associated with $\alpha$ and $\Sigma$ *if there exists a semi-oblivious $\Sigma$-derivation $S$ from $\alpha$ such that $\mathcal{T}$ is assigned to $S$.*

*Example 2 (cont'd).* The derivation tree associated with $S$ is represented in Figure 1. Bags have the same sharing types in the chase tree and in the derivation tree. However, we can see here that they are not linked in the same way: $B_3$ was a child of $B_2$ in the

chase tree, it becomes a child of $B_1$ in the derivation tree. Hence, the forbidden pattern cannot be found anymore in the tree.

Note that every non-root bag $B$ shares a least one term with its parent (since the rule frontiers are not empty), furthermore this term is generated in its parent (otherwise $B$ would have been added at a higher level in the tree).

# 4 Semi-Oblivious and Restricted Chase Termination

We now use derivation trees and sharing types to characterize the termination of the semi-oblivous chase. The fundamental property of derivation trees that we exploit is that, when two nodes have the same sharing type, the considered (semi-oblivious) derivation can always be extended so that these nodes have the same number of children, and in turn these children have the same sharing type. We first specify the notion of *bag copy*.

**Definition 7 Bag Copy.** *Let $\mathcal{T}, \mathcal{T}'$ be two (possibly equal) trees of bags. Let $B$ be a bag of $\mathcal{T}$ and $B'$ be a bag of $\mathcal{T}'$ such that $B \equiv_{st} B'$. Let $B_c$ be a child of $B$. A* copy *of $B_c$* under *$B'$ is a bag $B'_c$ such that $\mathsf{atom}(B'_c) = \varphi^s(\mathsf{atom}(B_c))$, where $\varphi^s$ is a substitution of $\mathsf{terms}(B_c)$ defined as follows:*

- *if $t \in \mathsf{shared}(B_c)$, then $\varphi^s(t) = \varphi_{\mathsf{atom}(B)\to\mathsf{atom}(B')}(t)$, where $\varphi_{\mathsf{atom}(B)\to\mathsf{atom}(B')}$ is the natural mapping from $\mathsf{atom}(B)$ to $\mathsf{atom}(B')$;*
- *if $t \in \mathsf{generated}(B_c)$, then $\varphi^s(t)$ is a fresh new variable.*

Let $\mathcal{T}_e$ be obtained from a derivation tree $\mathcal{T}$ by adding a copy of a bag: strictly speaking, $\mathcal{T}_e$ may not be a derivation tree in the sense that there may be no derivation to which it can be assigned (intuitively, some rule applications that would allow to produce the copy may be missing). Rather, there is some derivation tree of which $\mathcal{T}_e$ is a *prefix* (intuitively, one can add bags to $\mathcal{T}_e$ to obtain a derivation tree). That is why the following proposition considers more generally prefixes of derivation trees.

**Proposition 2.** *Let $\mathcal{T}$ be a prefix of a derivation tree, $B$ and $B'$ be two bags of $\mathcal{T}$ such that $B \equiv_{st} B'$, and $B_c$ be a child of $B$. Then:* (a) *the tree obtained from $\mathcal{T}$ by adding the copy $B'_c$ of $B_c$ under $B'$ is a prefix of a derivation tree, and* (b) *it holds that $B_c \equiv_{st} B'_c$.*

The size of a derivation tree without UPW is bounded, since its arity is bounded (Proposition 12 in the Appendix) and its depth is bounded by the number of sharing types. It remains to show that a derivation tree that contains a UPW can be extended to an arbitrarily large derivation tree. We recall that similar property would not hold for the chase tree, as witnessed by Example 2.

**Proposition 3.** *There exists an arbitrary large derivation tree associated with $\alpha$ and $\Sigma$ if and only if there exists a derivation tree associated with $\alpha$ and $\Sigma$ that contains an unbounded-path witness.*

The previous proposition yields a characterization of the existence of an infinite semi-oblivious derivation. At this point, one may notice that an infinite semi-oblivious derivation is not necessarily fair. However, from this infinite derivation one can always build a fair derivation by inserting missing triggers. Obviously, this operation has no effect on the termination of the semi-oblivious chase. More precaution will be required for the restricted chase.

One obtains an algorithm to decide termination of the semi-oblivious chase for a given set of rules: for each canonical instance, build a semi-oblivious derivation and the associated derivation tree by applying rules until a UPW is created (in which case the answer is no) or all possible rule applications have been performed; if no instance has returned a negative answer, the answer is yes.

**Corollary 1.** *The all-sequence termination problem for the semi-oblivious chase on linear rules is decidable.*

This algorithm can be modified to run in polynomial space (which is optimal [CGP15]), by guessing a canonical instance and a UPW of its derivation tree.

**Proposition 4.** *The all-sequence termination problem for the semi-oblivious chase on linear rules is in* PSPACE.

We now consider the restricted chase. To this aim, we call *restricted derivation tree* associated with $\alpha$ and $\Sigma$ a derivation tree associated with a restricted $\Sigma$-derivation from $\alpha$. We first point out that Proposition 2 is not true anymore for a restricted derivation tree, as the order in which rules are applied matters.

**Example 3.** *Consider a restricted tree that contains bags $B$ and $B'$ with the same sharing type, labeled by atoms $q(t, u)$ and $q(v, w)$ respectively, where the second term is generated. Consider the following rules (the same as in Example 1):*
$\sigma_1 : q(x, y) \to \exists z\ q(y, z)$
$\sigma_2 : q(x, y) \to q(y, y)$
*Assume $B$ has a child $B_c$ labeled by $q(u, z_0)$ obtained by an application of $\sigma_1$, and $B'$ has a child $B'_1$ labeled by $q(w, w)$ obtained by an application of $\sigma_2$. It is not possible to extend this tree by copying $B_c$ under $B'$. Indeed, the corresponding application of $\sigma_1$ does not comply with the restricted chase criterion: it would produce an atom of the form $q(w, z_1)$ that folds into $q(w, w)$.*

We thus prove a weaker proposition by considering that $B'$ is a leaf in the restricted derivation tree.

**Proposition 5.** *Let $\mathcal{T}$ be a prefix of a restricted derivation tree, $B$ and $B'$ be two bags of $\mathcal{T}$ such that $B \equiv_{st} B'$ and $B'$* is a leaf. *Let $B_c$ be a child of $B$. Then:* (a) *the tree obtained from $\mathcal{T}$ by adding the copy $B'_c$ of $B_c$ under $B'$ is a prefix of a restricted derivation tree, and* (b) *it holds that $B_c \equiv_{st} B'_c$.*

*Proof:* Let $S$ be the restricted derivation associated with $\mathcal{T}$. Let $S_c$ be the subsequence of $S$ that starts from $B$ and produces the strict descendants of $B$. Obviously, any rule application in $S_c$ is performed on a descendant of $B$, hence we do not care about rule applications that produce bags that are not descendants of $B$. We prove the property by

induction on the length of $S_c$. If $S_c$ is empty, the property holds with $\mathcal{T}_e = \mathcal{T}$. Assume the property is true for $0 \leq |S_c| \leq k$. Let $|S_c| = k + 1$. By induction hypothesis, there is an extension $\mathcal{T}'$ of $\mathcal{T}$ such that the subtree of $B$ restricted to the first $k$ elements of $S_c$ is 'quasi-isomorphic' to the subtree rooted in $B'$ (via a bijective substitution defined by the natural mappings between sharing types, say $\phi$) . Let $(\sigma, \pi)$ be the last trigger of $S_c$, and assume it applies to a bag $B_d$. In $\mathcal{T}'$, there is a bag $B'_d = \phi(B_d)$. Hence, $\sigma$ can be applied to $B'_d$ with the homomorphism $\phi \circ \pi$. Any folding of the produced bag $B''$ to a bag in $\mathcal{T}'$ necessarily maps $B''$ to a bag in the subtree rooted in $B'_d$ (because $B'_d$ and $B''$ share a term generated in $B'_d$, that only occurs in the subtree rooted in $B'_d$ and remains invariant by the folding). Since $B_d$ and $B'_d$ have quasi-isomorphic subtrees, and $(\sigma, \pi)$ satisfies the restricted chase criterion, so does $(\sigma, \phi \circ \pi)$. Furthermore, the quasi-isomorphism $\phi$ preserves the sharing types. Hence, $B'_d$ is added exactly like the bag produced by $(\sigma, \pi)$. We conclude that the property holds true at rank $k + 1$. □

The previous proposition allows us to obtain a variant of Proposition 3 adapted to the restricted chase:

**Proposition 6.** *There exists an arbitrary large restricted derivation tree associated with $\alpha$ and $\Sigma$ if and only if there exists a restricted derivation tree associated with $\alpha$ and $\Sigma$ that contains an unbounded-path witness.*

It is less obvious than in the case of the semi-oblivious chase that the existence of an infinite derivation entails the existence of an infinite *fair* derivation. However, this property still holds:

**Proposition 7.** *For linear rules, every (infinite) non-terminating restricted derivation is a subsequence of a fair restricted derivation.*

Similarly to Proposition 3 for the semi-oblivious chase, Proposition 6 provides an algorithm to decide termination of the restricted chase. The difference is that it is not sufficient to build a single derivation for a given canonical instance; instead, all possible restricted derivations from this instance have to be built (note that the associated restricted derivation trees are finite for the same reasons as before, and there is obviously a finite number of them). Hence, we obtain:

**Corollary 2.** *The all-sequence termination problem for the restricted chase on linear rules is decidable.*

A rough analysis of the proposed algorithm provides a CO-N2EXPTIME upper-bound for the complexity of the problem, by guessing a derivation that is of length at most double exponential, and checking whether there is a UPW in the corresponding derivation tree.

Importantly, the previous algorithm is naturally able to consider solely some type of restricted derivations, i.e., build only derivation trees associated with such derivations, which is of theoretical but also of practical interest. Indeed, implementations of the restricted chase often proceed by building *breadth-first* sequences (which are intrinsically fair), or variants of these. As witnessed by the next example, the termination of all breadth-first sequences is a strictly weaker requirement than the termination of all fair sequences, in the sense that the restricted chase terminates on more sets of rules.

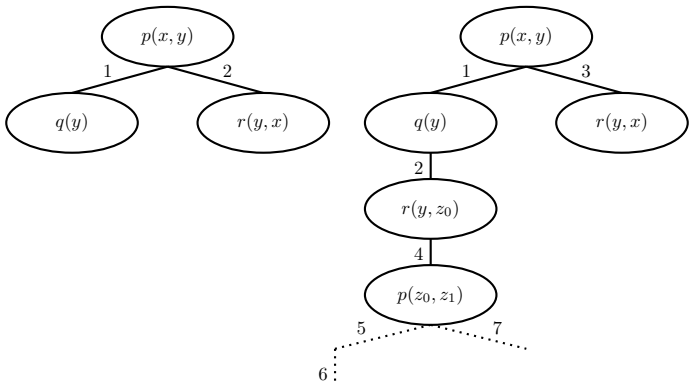


Figure 2: Finite versus Infinite Derivation Tree for Example 4

**Example 4.** *Consider the following set of rules:*
$\sigma_1 = p(x, y) \to q(y)$ $\quad$ $\sigma_2 = p(x, y) \to r(y, x)$
$\sigma_3 = q(y) \to \exists z\, r(y, z)$ $\quad$ $\sigma_4 = r(x, y) \to \exists z\, p(y, z)$
*All breadth-first restricted derivations terminate, whatever the initial instance is. Remark that every application of $\sigma_1$ is followed by an application of $\sigma_2$ in the same breadth-first step, which prevents the application of $\sigma_3$. However, there is a fair restricted derivation that does not terminate (and this is even true for any instance). Indeed, an application of $\sigma_2$ can always be delayed, so that it comes too late to prevent the application of $\sigma_3$. See Figure 2: on the left, a finite derivation tree associated with a breadth-first derivation from instance $p(x, y)$; on the right, an infinite derivation tree associated with a (non breadth-first) fair infinite derivation from the same instance. The numbers on edges give the order in which bags are created.*

We now prove the decidability of the one-sequence termination problem, building on the same objects as before, but in a different way. Indeed, a (restricted) derivation tree $\mathcal{T}$ that contains a UPW $(B, B')$ is a witness of the existence of an infinite (restricted fair) derivation, but does not prove that *every* (restricted fair) derivation that extends $\mathcal{T}$ is infinite. To decide, we will consider trees associated with a *sharing type* instead of a type. A derivation tree associated with a sharing type $T$ has a root bag whose sharing type is $T$, and is built as for usual root bags, except that shared terms are taken into account, i.e., triggers $(\sigma, \pi)$ such that $\pi(\mathrm{fr}(\sigma)) \subseteq \mathit{shared}(T)$ are simply ignored. The algorithm proceeds as follows:

1. For each sharing type $T$, generate all restricted derivations trees associated with $T$, stopping the construction of a tree when, for each leaf $B_L$, either there is no active trigger on $\mathit{atom}(B_L)$ or $B_L$ forms a UPW with one of its ancestors.

2. Mark all the sharing types that have at least one associated tree without UPW.

3. Propagate the marks until stability: if a sharing type $T$ has at least one tree for which all UPWs $(B, B')$ are such that the sharing type of $B$ is marked, then

mark $T$.

4. If all sharing types that correspond to instances (i.e., without shared terms) are marked, return *yes*, otherwise return *no*.

**Proposition 8.** *The previous algorithm terminates and returns yes if and only if there is a terminating restricted sequence.*

*Proof:*(Sketch) Termination follows from the finiteness of the set of sharing types and the bound on the size of a tree. Concerning the correctness of the algorithm, we show that a terminating restricted derivation cannot have a derivation tree that contains an unmarked UPW, i.e., whose associated sharing type is not marked. By contradiction: assume there is a terminating restricted derivation whose derivation tree contains an unmarked UPW; consider such an unmarked UPW $(B, B')$ such that $B'$ is of maximal depth in the tree. The subtree of $B'$ necessarily admits as prefix one of the restricted derivation trees associated with the sharing type of $B'$ built by the algorithm, otherwise the derivation would not be fair. Moreover, since the sharing type of $B'$ is not marked, this prefix contains an unmarked UPW. Hence, the tree contains an unmarked UPW $(B'', B''')$ with $B'''$ of depth strictly greater than the depth of $B'$, which contradicts the hypothesis. □

**Corollary 3.** *The one-sequence termination problem for the restricted chase on linear rules is decidable.*

By guessing a terminating restricted derivation, which must be of size at most double exponential, and checking that the obtained instance is indeed a universal model, we obtain a N2EXPTIME upper bound for the complexity of the one-sequence termination problem.

We conclude this section by noting that the previous Example 4 may give the (wrong) intuition that, given a set of rules, it is sufficient to consider breadth-first sequences to decide if there exists a terminating sequence. The following example shows that it is not the case: here, no breadth-first sequence is terminating, while there exists a terminating sequence for the given instance.

**Example 5.** *Let $\Sigma = \{\sigma_1, \sigma_2, \sigma_3\}$ with $\sigma_1 = p(x, y) \rightarrow \exists z\ p(y, z)$, $\sigma_2 = p(x, y) \rightarrow h(y)$, and $\sigma_3 = h(x) \rightarrow p(x, x)$. In this case, for every instance, there is a terminating restricted chase sequence, where the application of $\sigma_2$ and $\sigma_3$ prevents the indefinite application of $\sigma_1$. However, starting from $I = \{p(a, b)\}$, by applying rules in a breadth-first fashion one obtains a non-terminating restricted chase sequence, since $\sigma_1$ and $\sigma_2$ are always applied in parallel from the same atom, before applying $\sigma_3$.*

As for the all-sequence termination problem, the algorithm may restrict the derivations of interest to specific kinds.

# 5 Core Chase Termination

We now consider the termination of the core chase of linear rules. Keeping the same approach, we prove that the finiteness of the core chase is equivalent to the existence

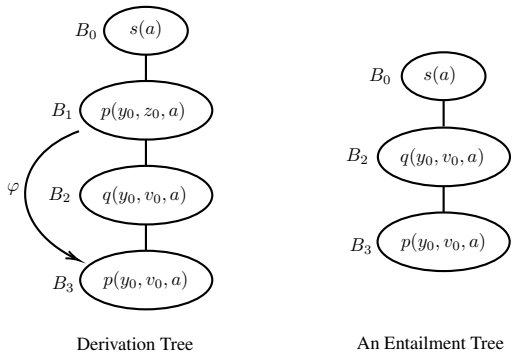


Figure 3: Derivation tree and entailment tree for Example 6

of a finite tree of bags whose set of atoms is a minimal universal model. We call this a *(finite) complete core*. To bound the size of a complete core, we show that it cannot contain an unbounded-path witness. Note that in the binary case, it would be possible to work again on derivation trees, but this is not true anymore for arbitrary arity. Indeed, as shown in Example 6, there are linear sets of rules for which no derivation tree form a complete core (while it holds for binary rules). We thus introduce a more general tree structure, namely *entailment trees*.

**Example 6.** *Let us consider the following rules:*

$s(x) \rightarrow \exists y \exists z\ p(y, z, x)$ $\qquad$ $p(y, z, x) \rightarrow \exists v\ q(y, v, x)$

$q(y, v, x) \rightarrow p(y, v, x)$

*Let $I = \{s(a)\}$. The first rule applications yield a derivation tree $\mathcal{T}$ which is a path of bags $B_0, B_1, B_2, B_3$ respectively labeled by the following atoms:* $s(a), p(y_0, z_0, a), q(y_0, v_0, a)$ *and* $p(y_0, v_0, a)$. *$\mathcal{T}$ is represented on the left of Figure 3. Let $A$ be this set of atoms. First, note that $A$ is not a core: indeed it is equivalent to its strict subset $A'$ defined by $\{B_0, B_2, B_3\}$ with a homomorphism $\pi$ that maps* atom$(B_1)$ *to* atom$(B_3)$. *Trivially, $A'$ is a core since it does not contain two atoms with the same predicate. Second, note that any further rule application on $\mathcal{T}$ is redundant, i.e., generates a set of atoms equivalent to $A$ (and $A'$). Hence, $A'$ is a complete core, however there is no derivation tree that corresponds to it. There is even no* prefix *of a derivation tree that corresponds to it (which ruins the alternative idea of building a prefix of a derivation tree that would be associated with a complete core). In particular, note that $\{B_0, B_1, B_2\}$ is indeed a core, but it is not complete.*

In the following definition of entailment tree, we use the notation $\alpha_1 \rightarrow \alpha_2$, where $\alpha_i$ is an atom, to denote the rule $\forall X(\alpha_1 \rightarrow \exists Y\ \alpha_2)$ with $X =$ *vars*$(\alpha_1)$ and $Y =$ *vars*$(\alpha_2) \setminus X$.

**Definition 8 Entailment Tree.** *An* entailment tree *associated with $\alpha$ and $\Sigma$ is a tree of bags $\mathcal{T}$ such that:*

1. *$B_r$, the root of $\mathcal{T}$, is such that $\Sigma \models \alpha \rightarrow$* atom$(B_r)$ *and $\Sigma \models$* atom$(B_r) \rightarrow \alpha$*;*

2. *For any bag $B_c$ child of a node $B$, the following holds:* (i) $\mathsf{terms}(B_c) \cap \mathsf{generated}(B) \neq \emptyset$ (ii) *The terms in* $\mathsf{generated}(B_c)$ *are variables that do not occur outside the subtree of $\mathcal{T}$ rooted in $B_c$* (iii) $\Sigma \models \mathsf{atom}(B) \rightarrow \mathsf{atom}(B_c)$.

3. *there is no pair of twins.*

Note that $\alpha$ is not necessarily the root of the entailment tree, as it may not belong to the result of the core chase on $\alpha$ (hence Point 1).

First note that an entailment tree is independent from any derivation. The main difference with a derivation tree is that it employs a more general parent-child relationship, that relies on entailment rather than on rule application, hence the name entailment tree. Intuitively, with respect to a derivation tree, one is allowed to move a bag $B$ higher in the tree, provided that it contains at least one term generated in its new parent $B_p$; then, the terms of $B$ that are not shared with $B_p$ are freshly renamed. Finally, since the problem of whether an atom is entailed by a linear existential rule knowledge base is decidable (precisely PSPACE-complete [CGL09]), one can actually generate all non-twin children of a bag and keep a tree with bounded arity.

Derivation trees are entailment trees, but not necessarily conversely. A crucial distinction between these two structures is the following statement, which does not hold for derivation trees, as illustrated by Example 6.

**Proposition 9.** *If the core chase associated with $\alpha$ and $\Sigma$ is finite, then there exists an entailment tree $\mathcal{T}$ such that the set of atoms associated with $\mathcal{T}$ is a complete core.*

*Example 6 (cont'd).* The tree defined by the path of bags $B_0$, $B_2$, $B_3$ is an entailment tree, represented on the right of Figure 3, which defines a complete core.

Differently from the semi-oblivious case, we cannot conclude that the chase does not terminate as soon as a UPW is built, because the associated atoms may later be mapped to other atoms, which would remove the UPW. Instead, starting from the initial bag, we recursively add bags that do not generate a UPW (for instance, we can recursively add all such non-twin children to a leaf). Once the process terminates (the non-twin condition and the absence of UPW ensure that it does), we check that the obtained set of atoms $C$ is complete (i.e., is a model of the KB): for that, it suffices to perform each possible rule application on $C$ and check if the resulting set of atoms is equivalent to $C$. See Algorithm 1. The set $C$ may not be a core, but it is complete iff it contains a complete core.

We now focus on the key properties of entailment trees associated with complete cores. We first introduce the notion of *redundant bags*, which captures some cases of bags that cannot appear in a finite core. As witnessed by Example 6, this is not a characterization: $B_1$ is not redundant (according to next Definition 9), but cannot belong to a complete core.

**Definition 9 Redundancy.** *Given an entailment tree, a bag $B_c$ child of $B$ is redundant if there exists an atom $\beta$ (that may not belong to the tree) with* (i) $\Sigma \models \mathsf{atom}(B) \rightarrow \beta$; (ii) *there is a homomorphism from* $\mathsf{atom}(B_c)$ *to $\beta$ that is the identity on* $\mathsf{shared}(B_c)$ (iii) $|\mathsf{terms}(\beta) \setminus \mathsf{terms}(B)| < |\mathsf{terms}(B_c) \setminus \mathsf{terms}(B)|$.

Note that $B_c$ may be redundant even if the "cause" for redundancy, i.e., $\beta$, is not in the tree yet. The role of this notion in the proofs is as follows: we show that if a complete entailment tree contains a UPW then it contains a redundant bag, and that a complete core cannot contain a redundant bag, hence a UPW. To prove this, we rely on next Proposition 10, which is the counterpart for entailment trees of Proposition 2: performing a bag copy from an entailment tree results in an entailment tree (the notion of prefix is not needed, since a prefix of an entailment tree is an entailment tree) and keeps the properties of the copied bag.

**Proposition 10.** *Let $B$ be a bag of an entailment tree $\mathcal{T}$, $B'$ be a bag of an entailment tree $\mathcal{T}'$ such that $B \equiv_{st} B'$. Let $B_c$ be a child of $B$ and $B'_c$ be a copy of $B_c$ under $B'$. Let $\mathcal{T}''$ be the extension of $\mathcal{T}'$ where $B'_c$ is added as a child of $B'$. Then* (i) *$\mathcal{T}''$ is an entailment tree;* (ii) *$B_c$ and $B'_c$ have the same sharing type;*(iii) *$B'_c$ is redundant if and only if $B_c$ is redundant.*

In light of this, the copy of a bag can be naturally extended to the copy of the whole subtree rooted in a bag, which is crucial element in the proof of next Proposition 11:

**Proposition 11.** *A complete core cannot contain* (i) *a redundant bag* (ii) *an unbounded-path witness.*

**Corollary 4.** *The all-sequence termination problem for the core chase on linear rules is decidable.*

**Algorithm 1:** Deciding core chase termination

**Input :** A set of linear rules
**Output:** `true` if and only if the core chase terminates on all instances

```
for each canonical atom α do
    Let T be the entailment tree restricted to α;
    while a bag B can be added to T respecting twin-free entailment tree
      condition and without creating a UPW do
        add B to T
    if there is a rule σ applicable to atoms(T) through π s.t.
      atoms(T) ⊭ atoms(T) ∪ π^s(head(σ)) then
        return false
return true
```

A rough complexity analysis of this algorithm yields a 2ExpTime upper bound for the termination problem. Indeed, the exponential number of (sharing) types yields a bound on the number of canonical instances to be checked, the arity of the tree, as well as the length of a path without UPW in the tree, and each edge can be generated with a call to a PSpace oracle.

# 6 Concluding remarks

We have shown the decidability of chase termination over linear rules for three main chase variants (semi-oblivious, restricted, core) following a novel approach based on derivation trees, and their generalization to entailment trees, and a single notion of forbidden pattern. As far as we know, these are the first decidability results for the restricted chase, on both versions of the termination problem (i.e., *all sequence* and *one sequence* termination). The simplicity of the structures and algorithms make them subject to implementation.

We leave for future work the study of the precise complexity of the termination problems. A straightforward analysis of the complexity of the algorithms that decide the termination of the restricted and core chases yields upper bounds, however we believe that a finer analysis of the properties of sharing types would provide tighter upper bounds. Future work also includes the extension of the results to more complex classes of existential rules: linear rules with a complex head, which is relevant for the termination of the restricted and core chases, and more expressive classes from the guarded family. Derivation trees were precisely defined to represent derivations with guarded rules and their extensions (i.e., greedy bounded treewidth sets), hence they seem to be a promising tool to study chase termination on that family.

# References


[BLMS11] 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.

[BMRT11] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Walking the complexity lines for generalized guarded existential rules. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 712–717. IJCAI/AAAI, 2011.

[BV81] Catriel Beeri and Moshe Y. Vardi. The implication problem for data dependencies. In Shimon Even and Oded Kariv, editors, *Automata, Languages and Programming, 8th Colloquium, Acre (Akko), Israel, July 13-17, 1981, Proceedings*, volume 115 of *Lecture Notes in Computer Science*, pages 73–85. Springer, 1981.

[CDK17] David Carral, Irina Dragoste, and Markus Krötzsch. Detecting chase (non)termination for existential rules with disjunctions. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 922–928. ijcai.org, 2017.

[CDL+07] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query

answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[CGL09] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. Datalog extensions for tractable query answering over ontologies. In Roberto De Virgilio, Fausto Giunchiglia, and Letizia Tanca, editors, *Semantic Web Information Management - A Model-Based Perspective*, pages 249–279. Springer, 2009.

[CGL12] 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.

[CGP15] Marco Calautti, Georg Gottlob, and Andreas Pieris. Chase termination for guarded existential rules. In Tova Milo and Diego Calvanese, editors, *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 91–103. ACM, 2015.

[DNR08] Alin Deutsch, Alan Nash, and Jeffrey B. Remmel. The chase revisited. In Maurizio Lenzerini and Domenico Lembo, editors, *Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada*, pages 149–158. ACM, 2008.

[Fag81] Ronald Fagin. A normal form for relational databases that is based on domians and keys. *ACM Trans. Database Syst.*, 6(3):387–415, 1981.

[FKMP05] 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.

[GHK+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.

[GM14] Tomasz Gogacz and Jerzy Marcinkowski. All-instances termination of chase is undecidable. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2014.

[GO18] Gösta Grahne and Adrian Onet. Anatomy of the chase. *Fundam. Inform.*, 157(3):221–270, 2018.

[Hal01] Alon Y. Halevy. Answering queries using views: A survey. *VLDB J.*, 10(4):270–294, 2001.

[Her12] André Hernich. Computing universal models under guarded tgds. In Alin Deutsch, editor, *15th International Conference on Database Theory, ICDT '12, Berlin, Germany, March 26-29, 2012*, pages 222–235. ACM, 2012.

[Len02] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*, pages 233–246, 2002.

[Mar09] Bruno Marnette. Generalized schema-mappings: from termination to tractability. In Jan Paredaens and Jianwen Su, editors, *Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA*, pages 13–22. ACM, 2009.

[MT14] Marie-Laure Mugnier and Michaël Thomazo. An introduction to ontology-based query answering with existential rules. In *Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*, pages 245–278, 2014.

[OHK09] Dan Olteanu, Jiewen Huang, and Christoph Koch. SPROUT: lazy vs. eager query plans for tuple-independent probabilistic databases. In *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*, pages 640–651, 2009.

[One12] Adrian Onet. *The chase procedure and its applications*. PhD thesis, Concordia University, Canada, 2012.

[Roc16] Swan Rocher. *Querying Existential Rule Knowledge Bases: Decidability and Complexity. (Interrogation de Bases de Connaissances avec Règles Existentielles : Décidabilité et Complexité)*. PhD thesis, University of Montpellier, France, 2016.

[Tho13] Michaël Thomazo. *Conjunctive Query Answering Under Existential Rules - Decidability, Complexity, and Algorithms*. PhD thesis, Montpellier 2 University, France, 2013.

# A Proofs for Section 2 (Preliminaries)

**Proposition 1.** *Let $\Sigma$ be a linear set of rules. The semi-oblivious (resp. restricted, core) chase terminates on all instances if and only if it terminates on all singleton instances.*

*Proof:* Obviously, the fact that a chase variant does not halt on an atomic instance implies the fact that it does not terminate on all instances. On the other direction, we can easily see that if the chase does not halt on an instance then it will not halt on one of its atoms. For a chase variant that does not terminate there exists an infinite derivation whose associated chase graph is also infinite. As the arity of the nodes in the chase graph is bounded by the size of the ruleset, the chase graph must contains an infinite path starting from a node of the initial instance. Because the chase graph for linear rules forms a tree it follows that this infinite path is created by a single atom of the initial instance.

□

# B Proofs for Section 3 (Derivation Trees)

**Proposition 12.** *The arity of a derivation tree is bounded.*

*Proof:* We first point out that a bag has a bounded number of twin children. Since we consider semi-oblivious derivations, a bag $B_p$ cannot have two twin children $B_{c_1}$ and $B_{c_2}$, created by applications of the same rule $\sigma$. Indeed, although these rule applications may map $\mathrm{body}(\sigma)$ to distinct atoms, the associated homomorphisms, say $\pi_1$ and $\pi_2$, would have the same restriction to the rule frontier, i.e., $\pi_{1|\mathrm{fr}(\sigma)} = \pi_{2|\mathrm{fr}(\sigma)}$. Hence, all twin children of a bag come from applications of distinct rules. It follows that the arity of a node is bounded by the number of atom types $\times$ the cardinal of the ruleset.
□

# C Proofs for Section 4 (Semi-Oblivious and Restricted Chase Termination)

**Proposition 2.** *Let $\mathcal{T}$ be a prefix of a derivation tree, $B$ and $B'$ be two bags of $\mathcal{T}$ such that $B \equiv_{st} B'$, and $B_c$ be a child of $B$. Then:* (a) *the tree obtained from $\mathcal{T}$ by adding the copy $B'_c$ of $B_c$ under $B'$ is a prefix of a derivation tree, and* (b) *it holds that $B_c \equiv_{st} B'_c$.*

*Proof:* Let $B$ and $B'$ be two atoms of $\mathcal{T}$ having the same sharing type. Let $B_c$ be a child of $B$ created by a trigger $(\sigma, \pi)$. By definition of derivation tree, $\pi$ maps the rule frontier $\mathrm{fr}(\sigma)$ to *terms*$(B)$, without this being possible for the parent of $B$. Furthermore, we know that $\pi$ maps $\mathrm{body}(\sigma)$ to a (possibly strict) descendant of $B$. We assume that $\mathcal{T}$ does not already contain the image of $\mathrm{head}(\sigma)$ via $\pi$, otherwise the thesis trivially holds. Let $S$ be the derivation associated with $\mathcal{T}$ and $\alpha_0, \ldots, \alpha_k$ be the path

of the *chase-graph* associated with $S$ such that $\alpha_0 = \mathit{atom}(B)$ and $\alpha_k = \mathit{atom}(B_c)$, whose sequence of associated rule applications is $(\sigma_1, \pi_1), \ldots, (\sigma_k, \pi_k) = (\sigma, \pi)$. We define $\hat{\pi}_i^{\text{safe}}(t) = \varphi_{\mathit{atom}(B) \to \mathit{atom}(B')} \circ \pi_i(t)$ whenever $\pi_i(t) \in \mathit{terms}(B)$ and otherwise $\hat{\pi}_i^{\text{safe}}(t)$ to be a fresh new variable consistently used over the rule applications, that is, such that $\pi_i^{\text{safe}}(t) = \pi_j^{\text{safe}}(t)$ if and only if $\hat{\pi}_i^{\text{safe}}(t) = \hat{\pi}_j^{\text{safe}}(t)$. Then, for all $1 \leq i \leq k$, we extend $S$ by adding a trigger $(\sigma_i, \hat{\pi}_i)$[4] whenever $\hat{\pi}_i^{\text{safe}}(\text{head}(\sigma_i))$ is not an atom already produced by $S$ thereby obtaining a new derivation $S'$. Let $\mathcal{T}'$ be an extension of $\mathcal{T}$ where a bag labeled with the atom $\hat{\pi}_i^{\text{safe}}(\text{head}(\sigma_i))$ is added for each new trigger in $S'$ and attached to the highest descendant of $B'$ whose set of terms contains $\hat{\pi}_i(\text{fr}(\sigma_i))$. Clearly, $\mathcal{T}'$ is a derivation tree associated with $S'$. We now show that $\mathcal{T}'$ contains a node $B'_c$ which is a copy of $B_c$ under $B'$. As $B$ is the parent of $B_c$, the image of $\text{fr}(\sigma)$ via $\pi$ contains at least one term which is generated in $B$ (and in general only terms generated by the ancestors of $B$). Therefore, because $B$ and $B'$ have the same sharing type, the image of $\text{fr}(\sigma)$ via $\varphi_{\mathit{atom}(B) \to \mathit{atom}(B')} \circ \pi$ contains at least one term generated in $B'$ (and in general only terms generated by the ancestors of $B'$). So, $B'$ is the only possible parent of $B'_c$ in $\mathcal{T}'$. Moreover, it is easy to see that $B_c \equiv_{st} B'_c$. Let $\mathcal{T}''$ be the extension of $\mathcal{T}$ with $B'_c$ under $B'$. It can be easily verified that $\mathcal{T}''$ is a prefix of the derivation tree $\mathcal{T}'$, in the sense that it is a tree of bags which can be obtained by recursively removing some of the leaves of $\mathcal{T}'$, i.e., those corresponding to the triggers in $S' \setminus S$ which are different from $(\sigma, \pi)$.

□

**Proposition 3.** *There exists an arbitrary large derivation tree associated with $\alpha$ and $\Sigma$ if and only if there exists a derivation tree associated with $\alpha$ and $\Sigma$ that contains an unbounded-path witness.*

*Proof:* If there is no derivation tree having an unbounded-path witness, then the depth of all derivation trees is upper bounded by the number of sharing types. As derivation trees are of bounded arity, all derivation trees must be of bounded size.

If there is a derivation tree $\mathcal{T}$ having an unbounded-path witness $(B, B')$, we show that there are arbitrary large derivation trees. We do so by contradiction. Let us assume that $(B, B')$ is a UPW be two such bags such that $B'$ is of maximal depth among all such pairs and among all trees, which by hypothesis are of bounded size. Let $B_c$ be the child of $B$ that is on the shortest path from $B$ to $B'$ (possibly $B_c = B'$). By Proposition 2, $B'$ has a child $B'_c$ that has the same sharing type as $B_c$. By Proposition 2, $B'$ has a child $B'_c$ that has the same sharing type as $B_c$, either in the same tree, or in an extension of this tree, which is in contradiction with the fact that $B'$ was of maximal depth. Hence there are arbitrary large derivation trees. □

**Proposition 4.** *The all-sequence termination problem for the semi-oblivious chase on linear rules is in* PSPACE.

*Proof:* Let $\mathcal{T}$ be a derivation tree of root the canonical instance $\{\alpha\}$ that contains a UPW $(B, B')$, where the sharing type of both bags is $ST$. We show that there exists a semi-oblivious derivation of length at most exponential whose derivation tree has root

[4] $\hat{\pi}_i$ is the restriction of $\hat{\pi}_i^{\text{safe}}$ to the variables of the body of $\sigma_i$.

$\{\alpha\}$ and that contains a UPW $(B_s, B'_s)$ where the sharing type of both bags is $ST$. First, by Proposition 2, we conclude that it is not necessary to have twice the same sharing type on the path from the root to $B'$ in the derivation tree. It is thus enough to show that to generate a child $B_c$ from its parent $B_p$, a derivation of length at most exponential is necessary. Let us consider the chase graph of the derivation generating $\mathit{atom}(B_c)$ from $\mathit{atom}(B_p)$. This chase graph can be assumed w.l.o.g. to be a path. It there are no pairs of atoms having the same sharing type on this path, then the derivation is of length at most exponential. Otherwise, we show that we can build a shorter semi-oblivious derivation that generates $\mathit{atom}(B_c)$. Let us thus assume that there is $B$ and $B'$ such that both have the same sharing type, and the terms of $B_p$ that appear in $B$ appear in the same position in $B'$, and that $B'$ is on the path from $B$ to $B_c$ in the chase graph. A derivation similar to that applicable after $B'$ is actually applicable to $B$, by Proposition 2. A copy of $B_c$ under $B_p$ is thus generated by this derivation, which proves our claim.

We now describe the algorithm. We guess the canonical instance and the sharing type $ST$ of the UPW. We then check that there is a descendant (not necessarily a child) of that canonical instance that has sharing type $ST$. This can be done by guessing the shortest derivation creating a bag of sharing type $ST$. It is only necessary to remember the sharing type of the "current" bag, as we know that any bag created during a derivation is added as a descendant of the root. We then want to prove that a bag of sharing type $ST$ can have a (strict) descendant of sharing type $ST$. In contrast with the case of the root, a trigger applied below a bag $B$ does not necessarily create a bag that is as well below $B$ – it could be added higher up in the tree. We thus have to remember the shared variables of $B$, and verify at each step that the shared variables of the currently considered bag are not a subset of them. This leads to a PSPACE procedure. □

**Proposition 6.** *There exists an arbitrary large restricted derivation tree associated with $\alpha$ and $\Sigma$ if and only if there exists a restricted derivation tree associated with $\alpha$ and $\Sigma$ that contains an unbounded-path witness.*

*Proof:* If there is no restricted derivation tree with a UPW, then the size of any restricted derivation tree is bounded since a restricted derivation tree is a derivation tree. We prove the other direction by contradiction. Assume that the size of restricted derivation trees is bounded whereas the forbidden pattern occurs in some of them. Consider a restricted chase sequence $S$ with associated restricted derivation tree $\mathcal{T}$ that contains a UPW $(B, B')$ of maximal depth among all such pairs and all trees, and such that $B'$ is a leaf (we can do the latter assumption since the prefix of any restricted derivation is a restricted derivation).

Let $B_c$ be the child of $B$ that is on the shortest path from $B$ to $B'$ (possibly $B_c = B'$). By Proposition 5, there is a restricted derivation tree that extends $\mathcal{T}$ such that $B'$ has a child $B'_c$ of the same sharing type as $B_c$, hence $(B_c, B'_c)$ is a UPW of depth strictly greater than $(B, B')$, which contradicts the hypothesis. □

**Proposition 7.** *For linear rules, every (infinite) non-terminating restricted derivation is a subsequence of a fair restricted derivation.*

*Proof:* Let $S$ be a non-terminating restricted derivation. In particular, there exists a least one infinite branch in the associated derivation tree. Let us consider the following

derivation: when the node $B_k$ of depth $k$ on this branch has been generated, complete the corresponding subsequence by trying to apply (i.e., while respecting the restricted criterion) all currently applicable triggers that add a bag a depth at most $k-1$. These additional rule applications cannot prevent the creation of any bag that is below $B_k$ in the derivation tree. Indeed, let $\alpha_c$ be an atom possibly created by a rule application, whose bag would be attached as a child of a bag $B$; since $\alpha_c$ shares a variable with $\mathit{atom}(B)$ that is generated in $B$, which thus only occurs in the subtree of $B$, the only possibility for $\alpha_c$ to fold into the current instance, is to be mapped to an atom in the subtree of $B$. By construction, any possible rule application will be performed or inhibited at some point, which implies that the derivation that we build in this fashion is fair. □

# D Proofs for Section 5 (Core Chase Termination)

**Proposition 9.** *If the core chase associated with $\alpha$ and $\Sigma$ is finite, then there exists an entailment tree $\mathcal{T}$ such that the set of atoms associated with $\mathcal{T}$ is a complete core.*

*Proof:* Let $\mathcal{T}$ be the derivation tree associated with a derivation containing a core $C$ of $\mathit{chase}(\alpha, \Sigma)$. Let $\varphi$ be an idempotent homomorphism from the atoms of $\mathcal{T}$ to $C$. We assign to each bag $B$ of $\mathcal{T}$ a set of trees $\{T_1, \ldots, T_{n_B}\}$ such that:

1. each tree contains only elements of $C$;
2. the forest assigned to $B$ contains exactly once the elements of $C$ appearing in the subtree rooted in $B$;
3. for each pair $(B_p, B_c)$ of bags in some $T_i$ such that $B_p$ is a parent of $B_c$, $\Sigma \models \mathit{atom}(B_p) \rightarrow \mathit{atom}(B_c)$;
4. each $T_i$ is a decomposition tree;
5. for each $T_i$, the root of $T_i$ contains all the terms that belong both to $T_i$ and to $C \setminus T_i$;
6. each term $t$ belonging to distinct $T_i$ and $T_j$ of the forest assigned with a bag $B$ also belongs to the parent of $B$.

Moreover, we will show that if $\varphi(B)$ is a descendant of $B$ (including $B$) in $\mathcal{T}$, then its associated forest is a tree.

- if $B$ is a leaf, we consider two cases:
  - $B$ belongs to the core: we assign it a single tree, containing only a root being itself. All conditions are trivial.
  - $B$ does not belong to the core: we assign it an empty forest, and all conditions are trivial.
- if $B$ is an internal node, let $\{T_1, \ldots, T_n\}$ be the union of the forests assigned to the children of $B$. We distinguish three cases:

– $B$ is in the core: we assign to $B$ the tree $T$ containing $B$ as root, and having as children the roots of $\{T_1, \ldots, T_n\}$.
  * 1. 2.: holds by induction assumption, the fact that different $T_i$'s cover disjoint subtrees of $\mathcal{T}$, and the fact that $B$ belongs to the core
  * 3.: it is enough to check this for the pairs (root of $T$, root of $T_i$). The root of $T$ is an ancestor of root of $T_i$ in $\mathcal{T}$, hence $\Sigma \models \mathit{atom}(\mathrm{root}(T)) \rightarrow \mathit{atom}(B_i)$, where $B_i$ is the root of $T_i$
  * 4. if $t$ appears in $T$ but in no $T_i$, it appears only in $B$ and the connectivity of the substructure containing $t$ holds. If it belongs to some $T_i$ and to $C \setminus T_i$, it must belong to the root of $T_i$ by assumption 6.. If $t$ belongs to $C \setminus T$, it belongs to $B$ by connectivity of $\mathcal{T}$. If $t$ belongs to another $T_j$, we distinguish two cases: $T_j$ is in the same forest as $T_i$, and then by induction assumption 7. on the child of $B$ to which this forest is associated, $t$ belongs to $B$. Or $T_j$ is in the forest of another child of $B$, and then by connectivity property for $t$, it belongs to $B$. Hence the connectivity property for $t$ in $T$ is fulfilled.
  * 5. By connectivity of $\mathcal{T}$, as $B$ is the root of $T$
  * 6. true as there is only one tree
– $\varphi(B) \neq B$ but is a descendant of $B$. By induction assumption 2., there exists exactly one tree among the trees associated with children of $B$ containing $\varphi(B)$. Let assume w.l.o.g that it is $T_1$, of root $B_1$. We build the following tree $T$: for all $T_i \neq T_1$, we add to $B_1$ a subtree by putting the root of $T_i$ under $B_1$.
  * 1. No added elements, hence by induction assumption 1.
  * 2. No added elements, hence by induction assumption 2.
  * 3. To check for pairs ($B_1$,$B_i$), where $B_i$ is the root of $T_i$. $\Sigma \models \mathit{atom}(B_1) \rightarrow \mathit{atom}(\varphi(B))$, as $\varphi(B)$ is a descendant of $B_1$ in $T_1$. Moreover, $\Sigma \models \mathit{atom}(\varphi(B)) \rightarrow \mathit{atom}(B_i)$, as $\varphi(B)$ is more specific than $B$, and $\varphi$ is the identity on shared terms.
  * 4. for all term $t$ appearing in a single tree, the connectivity property holds by induction assumption 4.. Let $t$ appearing in two trees. $t$ appears in the roots of both tree by 6., and must appear in $B$ by connectivity of $\mathcal{T}$, hence in $\varphi(B)$, and hence in $B_1$ (by 6.). As $B_1$ and the roots of both trees are neighbor, this proves the result.
  * 5. let $t$ belonging to $T$ and to $C \setminus T$. By connectivity of $\mathcal{T}$, $t$ belongs to $B$, hence to $\varphi(B)$ (because $\varphi(t) = t$). As $t$ belongs both to $T_1$ and to $C \setminus T_1$, $t$ belongs to $B_1$, and hence to the root of the assigned tree.
  * 6. true as there is only one tree.
– $\varphi(B)$ is not a descendant of $B$. We assign to $B$ the union of the forests associated to its children.
  * 1.-5 By induction assumption
  * 6. let $t$ belonging to two trees $T_1$ and $T_2$. If $T_1$ and $T_2$ come from forest associated to two different children, $t$ belongs to $B$ by connectivity

of $\mathcal{T}$. If $T_1$ and $T_2$ come from the same forest, $t$ belongs to $B$ by induction assumption 7. Then $t$ belongs to $B$. As $t$ is in $C$, $t$ belongs to $\varphi(B)$. By connectivity of $\mathcal{T}$, it belongs to the parent of $B$, because that parent is on the path from $B$ to $\varphi(B)$, which proves 6.

Finally, we check that the following property is satisfied: for any bag $B$, if $B$ is in the core, then a single tree with root $B$ is assigned to it. If $\alpha$ is in the core, we have built such a tree. It remains to obtain an entailment tree: for that, we have to bring up nodes at the highest level with respect to shared terms. We may also have to say something about 'generated' if it still appear in the definition of an entailment tree. □

**Proposition 10.** *Let $B$ be a bag of an entailment tree $\mathcal{T}$, $B'$ be a bag of an entailment tree $\mathcal{T}'$ such that $B \equiv_{st} B'$. Let $B_c$ be a child of $B$ and $B'_c$ be a copy of $B_c$ under $B'$. Let $\mathcal{T}''$ be the extension of $\mathcal{T}'$ where $B'_c$ is added as a child of $B'$. Then* (i) *$\mathcal{T}''$ is an entailment tree;* (ii) *$B_c$ and $B'_c$ have the same sharing type;*(iii) *$B'_c$ is redundant if and only if $B_c$ is redundant.*

Another important property of entailment trees (which is also satisfied by derivation trees) is that its structure provides information on where a bag may be mapped by $\varphi$ if its parent is left invariant by $\varphi$.

**Lemma 1.** *Let $\mathcal{T}$ be an entailment tree. Let $\varphi$ be a homomorphism from the atoms of $\mathcal{T}$ to themselves. Let $B_p$ such that $\varphi_{|\text{terms}(B_p)}$ is the identity. Let $B_c$ be a child of $B_p$. Then $\varphi(B_c)$ is in the subtree rooted in $\varphi(B_p) = B_p$.*

*Proof:* $B_c$ is a child of $B_p$ thus there exists at least one term generated in $B_p$ that is a term of $B_c$. As $\varphi$ is the identity on $B_p$, that term belongs as well to $\varphi(\mathit{atom}(B_c))$. Thus $\varphi(\mathit{atom}(B_c))$ should also be in a bag that is in the subtree rooted in $B_p$. □

**Proposition 13.** *A complete core cannot contain a redundant bag.*

*Proof:* Let $\mathcal{T}$ be a complete entailment tree, and let $\hat{B}$ be a bag that is redundant. We prove that there exists a non-injective endomorphism of $\mathcal{T}$, showing that $\mathcal{T}$ cannot be a core. For any entailment tree $\mathcal{T}_p$ that is a prefix of $\mathcal{T}$, we build $\mathcal{T}'_p$ and a mapping $\varphi$ from the terms of $\mathcal{T}_p$ to the terms of $\mathcal{T}'_p$ as follows:

- for any prefix of $\mathcal{T}_p$ that does not contain $\hat{B}$, we define $\mathcal{T}'_p = \mathcal{T}_p$ and $\varphi$ the identity
- for the prefix that contains all nodes of $\mathcal{T}$, including $\hat{B}$, except the descendants of $\hat{B}$, we define $\mathcal{T}'_p$ as $\mathcal{T}_p$ to which we add a leaf (if necessary) to the parent of $\hat{B}$ in $\mathcal{T}$, which is the witness of the redundancy of $\hat{B}$. We define $\varphi$ as the identity on any term that is not generated in $\hat{B}$, and as its image by the $\varphi$ witnessing the redundancy pattern on terms generated in $\hat{B}$.
- if we have defined $\mathcal{T}'_p$ for $\mathcal{T}_p$, and we want to define $\varphi$ for $\mathcal{T}'_n$ for $\mathcal{T}_n$ which is $\mathcal{T}_p$ to which a leaf $B_d$ has been added, we add where it belongs the bag $\varphi(B_d)$, where we extend $\varphi$ to term generated in $B_d$ by choosing fresh images.

By construction, $\mathcal{T}'$ is an entailment tree, and $\varphi$ is a homomorphism from $\mathcal{T}$ to $\mathcal{T}'$. Moreover, $\varphi$ is not injective: indeed, as $\hat{B}$ is redundant, $\varphi$ is not injective on the terms of $\hat{B}$.

As $\mathcal{T}$ is complete, there exists a homomorphism from $\mathcal{T}'$ to $\mathcal{T}$. Hence the composition of the two homomorphisms is a homomorphism from $\mathcal{T}$ to itself, which is not injective, as $\varphi$ is not. Hence $\mathcal{T}$ is not a core. □

**Proposition 14.** *A complete core cannot contain any unbounded-path witness.*

*Proof:* We prove the result by contradiction. Let us assume that $\mathcal{T}$ is a complete core containing an unbounded-path witness $(B, B')$. Let us choose $(B, B')$ such that $B'$ is of maximal depth with respect to its branch, that is, there is no unbounded-path witness $(B''', B'')$ with $B''$ a strict descendant of $B'$.

Let $B_c$ be the child of $B$ on the path from $B$ to $B'$. Let us denote by $\mathcal{T}_{B_c}$ the subtree of $\mathcal{T}$ which is rooted at $B_c$ and by $\mathcal{T}_{B'_c}$ a copy of $\mathcal{T}_{B_c}$ under $B'$ whose root is $B'_c$. Then, let $\mathcal{T}'$ be the extension of $\mathcal{T}$ where $\mathcal{T}_{B'_c}$ is added as a child of $B'$. We want to show that there exists a bag $B'_r$ child of $B'$ and a mapping from $\mathcal{T}_{B'_c}$ into $\mathcal{T}_{B'_r}$, which is the identity on the terms of $\mathcal{T}$. More precisely, we want to show that for each $B'_d$ descendant of $B'_c$ the following properties hold.

1. the image of $B'_d$ belongs to $\mathcal{T}_{B'_r}$
2. the image of a term generated in $B'_d$ is a term generated in a bag of $\mathcal{T}_{B'_r}$

We do so by induction on $k$ the distance between $B'_d$ and $B'_c$ in $\mathcal{T}$.

- If $k = 0$ then $B'_d = B'_c$. Because $\mathcal{T}$ is a complete core, there exists a homomorphism from the atoms of $\mathcal{T}'$ to those of $\mathcal{T}$ which is the identity on the terms of $\mathcal{T}$. We show that the image of $B'_c$ is a strict descendant of a child of $B'$. Note first that no child of $B'$ in $\mathcal{T}$ can be a safe renaming of $B'_c$. Indeed, by Proposition 10, $B_c$ and $B'_c$ have the same sharing type and therefore $B'_c$ (as well as any safe renaming of its generated terms) cannot be a child of $B'$ because the couple $(B_c, B'_c)$ would form an unbounded-path witness with $B'_c$ strictly deeper than $B'$. Then, if $B'_c$ maps to $B'$ then the couple $(B', B'_c)$ is redundant and therefore also $(B, B_c)$ is redundant, by Proposition 10, which in turn implies that $\mathcal{T}$ is not a core, because of Proposition 13. Finally, if $B'_c$ maps to any child of $B'$ then it does so by specializing the sharing type of $B'_c$ (as we showed that no safe renaming of $B'_c$ can be a child of $B'$), which means that $B'_c$ is redundant. Therefore, by Proposition 10, $B_c$ is also redundant and hence, by Proposition 13, $\mathcal{T}$ is not a core. This proves that the image of $B'_c$ is a strict descendant of some $B'_r$ child of $B'$.

  Now, to prove the second point, let $t$ be a term generated in $B'_c$ and $t'$ its image. It is easy to see that for entailment trees any term that belongs to two bags in ancestor-descendant relationship also belongs to the bags on the shortest path between them. Therefore, if $t'$ is generated by a strict ancestor of $B'_r$ then $t'$ belongs to the terms of $B'$. This means that starting from the sharing type of

$B'_c$ one can build a strictly more specific sharing type where the position corresponding to the generated term $t$ becomes shared with $B'$. From this one can find a node $B''_c$ which is strictly more specific than $B'_c$ and that can be added as a child of $B'$. This means that $B'_c$ is redundant and by Proposition 10 also $B_c$ is redundant, so $\mathcal{T}$ is not a core.

- Let us assume that both properties hold for all bags at distance $k$ from $B'_c$. We want to show that they still holds for the bags at distance $k+1$.

  Let $B'_d$ be a node at distance $k+1$ from $B'_c$ whose parent is $B'_\delta$. By definition, $B'_d$ contains a term generated by $B'_\delta$ and, by induction, we know that the image of this term is generated in a bag of $\mathcal{T}_{B'_r}$. Thus, it follows that the image of $B'_d$ belongs to $\mathcal{T}_{B'_r}$ as required by the first point.

  For the second point we reason by contradiction and show that when the property does not hold then $\mathcal{T}$ admits a non-injective endomorphism and thus it is not a core. We proceed with the following construction. Let $\mathcal{T}_{B_r}$ be a copy of $\mathcal{T}_{B'_r}$ under $B$ and $\mathcal{T}''$ the extension of $\mathcal{T}$ where $\mathcal{T}_{B_r}$ is added as a child of $B$. We know by induction that there exists a homomorphism from $\mathcal{T}'$ to $\mathcal{T}$ mapping all nodes at distance $k+1$ from $B'_c$ to the subtree rooted at $B'_r$. From this, we can conclude that there exists a homomorphism from $\mathcal{T}_{(k+1)}$ to $\mathcal{T}''$, where $\mathcal{T}_{(k+1)}$ is the prefix of $\mathcal{T}$ which includes all nodes of $\mathcal{T}$ except for the descendants of $B_c$ that are at distance strictly greater than $k+1$ from it. Now, we further extend $\mathcal{T}''$ by adding an image for all nodes which are at distance strictly greater than $k+1$ from $B_c$ thereby obtaining a new entailment tree $\mathcal{T}'''$. It follows that $\mathcal{T}$ can be mapped to $\mathcal{T}'''$. Beside, since $\mathcal{T}$ is complete there exists an homomorphism from $\mathcal{T}'''$ to $\mathcal{T}$. So, by composing these two homomorphisms we get a homomorphism from $\mathcal{T}$ to $\mathcal{T}$.

  We show that the homomorphism from $\mathcal{T}$ to $\mathcal{T}'''$ is non-injective. Recall that to construct $\mathcal{T}'$ the whole subtree rooted at $B'_c$ has been copied from the subtree rooted at $B_c$. Let us denote by $B_d$ the node at distance $k+1$ from $B_c$ from which $B'_d$ has been copied under $B'_\delta$. Let $t$ be a term generated at position $i$ in $B'_d$. If its image was generated by a strict ancestor of $B'_r$ then this would also belong to the terms of $B'$. By Proposition 10, $B_d$ and $B'_d$ have the same sharing types, hence the mapping from $\mathcal{T}_{(k+1)}$ to $\mathcal{T}''$ (and thus that from $\mathcal{T}$ to $\mathcal{T}'''$) maps the generated term at position $i$ of $B_d$, we call $s$, to a distinct term in $B$, we call $s'$. Moreover, the homomorphism is the identity on $s'$. Therefore, the homomorphism from $\mathcal{T}$ to $\mathcal{T}'''$ is non-injective as both $s'$ and $s$ have the same image.

To finish the proof, we proceed with the following construction. Let $\mathcal{T}^*$ be an entailment tree derived from $\mathcal{T}$ where $i)$ the whole subtree rooted at $B'_r$ has been copied under $B$ and $ii)$ the subtree rooted at $B_c$ has been removed. Note that $\mathcal{T}^*$ is of size strictly smaller than that of $\mathcal{T}$ because we added a bag for each descendant node of $B'_r$, which is a strict descendant of bag $B_c$, and that this last one has been removed. Now, because $\mathcal{T}_{B'_c}$ maps to $\mathcal{T}_{B'_r}$ it follows that $\mathcal{T}_{B_c}$ maps to $\mathcal{T}_{B_r}$ and by extending this homomorphism to the identity on all other terms we get that $\mathcal{T}$ can be mapped to $\mathcal{T}^*$. Hence, $\mathcal{T}$ is not a core. □