Normalisation Procedures for Existential Rules that Preserve Chase Termination

Lucas Larroque

Contents

1 Introduction

Existential rules are a logical framework used to model constraints and knowledge on data. An existential rule is a first-order formula of the form $B \rightarrow H$, with B and H conjunctions of atoms called the body and the head of the rule, respectively. In addition, the variables appearing in the body are universally quantified, and the ones in the head but not in the body are existentially quantified, making an existential rule a closed formula. For instance, the formula $\forall x.\ \mathit{Manager}(x) \rightarrow \exists y.\ \exists z.\ \mathit{ReportsTo}(x, y) \wedge \mathit{ReportsTo}(z, x)$ is a rule, in which $\mathit{Manager}(x)$ is the body and $\mathit{ReportsTo}(x, y) \wedge \mathit{ReportsTo}(z, x)$ the head. A fundamental problem in this framework is query entailment, which consists of deciding if a logical formula (representing a query) is entailed by a database added with existential rules. In this paper, a database is also called a factbase, and a pair composed of a factbase and a set of rules is called a knowledge base (also known as an ontology).

The problem of query entailment from a knowledge base is an undecidable problem. Several methods have been developed to solve it, which terminate on some specific classes of rules. The approach we will focus on is known as forward chaining [12], also called the chase in this framework. The chase is an algorithm that, by applying the rules from the rule set to the database, generates new elements to produce a universal model of the knowledge base. Then, if the chase halts, it suffices to query the universal model to produce the answers to a conjunctive query. One point that makes the chase difficult is that applying blindly the rules may produce redundancies, and then the chase may fail to terminate because of those, even if a finite universal model of the knowledge base exists. As such, a few variants of the chase have been developed, each one having a specific way of avoiding redundancies. The oblivious chase [4] naively applies every rule in all possible ways, and the equivalent chase [11] applies a rule only if it produces some non-redundant information, while the semi-oblivious [10] and the restricted (or standard) chase [6] have conditions in between.

Some articles that consider reasoning based on forward or backward chaining use certain properties on the rule sets they work with. For instance, [2] and [9] decompose rules into so-called pieces (or use single-piece rules directly). Pieces are subsets of a rule head that contain the minimal amount of information brought by the application of a rule: any rule can be decomposed according to its pieces, and we cannot decompose pieces further while conserving logical equivalence without introducing new predicates. In other works, such as [4], [7] and [13], single-head existential rules, i.e., with heads composed of a single atom, are used. As such, studying procedures that from any rule set produce a rule set with certain syntactic properties is an interesting concern. We call them normalization procedures.

The main question we try to answer in this report is the characterization of the impact of those normalization procedures on chase termination. Our first goal is to determine if these normalization procedures preserve the termination of the chase. We say that a normalization preserves the termination of the chase if, for a given chase variant, when the chase terminates on a rule set (for any factbase), the chase terminates as well on the rule set output by the normalization procedure. This is not obvious because deciding chase termination is undecidable. Another interesting property would be to make a rule set gain termination by going through a normalization procedure.

We are interested in three normalization procedures. First, the single-piece transformation transforms a rule set into a set of single-piece rules equivalent to the original set. For instance, applying this transformation on the rule R yields the rule set:

$$\{\mathit{Manager}(x) \to \exists y.\ \mathit{ReportsTo}(x,y),\ \mathit{Manager}(x) \to \exists z.\ \mathit{ReportsTo}(z,x)\}$$

We will show that this translation preserves the termination of every chase variant except the restricted chase. Then, we will see two decompositions that output a rule set with single-head rules. The first one is named the one-way atomic decomposition. Applied on R, it outputs the following rule set:

$$\{\mathit{Manager}(x) \to \exists y \exists z.\ X_R(x,y,z),\ X_R(x,y,z) \to \mathit{ReportsTo}(x,y),\\ X_R(x,y,z) \to \mathit{ReportsTo}(z,x)\}$$

This first decomposition will only preserve the termination of the oblivious and the semi-oblivious chase, which will motivate the definition of our last transformation, the two-way atomic decomposition, which has the same goal but also preserves the termination of the equivalent chase. Applying it on R yields the rule set obtained by the one-way atomic decomposition, with the addition of the rule $\mathit{ReportsTo}(x,y) \wedge \mathit{ReportsTo}(z,x) \to X_R(x,y,z)$, which we call the backwards rule. In addition, since both atomic decompositions introduce new rules without existential variables, called Datalog rules, it will motivate the study of the Datalog-first restricted chase, which a variant of the restricted chase in which Datalog rules are applied with higher priority. Whereas the termination of this chase variant is not preserved by the one-way atomic decomposition, it is preserved by the two-way atomic decomposition.

The results obtained are compiled in Table 1. As four cases are possible, we use four different symbols. The symbol $=$ means the termination is strictly preserved by the transformation (i.e., the chase terminates for a rule set if and only if it terminates on its translation), meanwhile a $+$ means if the chase terminates for a rule set then it terminates on its translation, but the converse is false, and a $-$ it is not preserved and never recovered. Finally, a $\neq$ means it can be gained or lost depending on the rule sets.

Table 1: Results

	O	**SO**	**R**	Df-**R**	**E**
Single-piece translation	$=$	$+$	$\neq$	$\neq$	$=$
One-way atomic decomposition	$=$	$=$	$-$	$-$	$-$
Two-way atomic decomposition	$=$	$=$	$-$	$=$	$=$

2 Preliminaries

2.1 Syntax

We will work in a first-order setting limited to *predicates*, *constants*, and *variables*. A *term* is a variable or a constant. Each predicate is associated with a natural integer, called the *arity* of the predicate.

Definition 2.1. An *atom* is of the form $p(e_1, \ldots, e_n)$ with p a predicate of arity n and $e_1, \ldots, e_n$ some terms. A *factbase* is an existentially closed conjunction of atoms.

It is often convenient to see a factbase as a set of atoms. Henceforth, we identify a factbase with the corresponding set of atoms. For instance, the factbase $\exists x, y, z.\ A(x, y) \wedge B(x, z, y)$ is seen as the set $\{A(x, y), B(x, z, y)\}$.

Let F be a logical formula or a set of logical formulas. The set of variables, constants, and terms appearing in F are denoted by $\mathit{var}(F)$, $\mathit{cnst}(F)$, and $\mathit{term}(F)$, respectively.

In the following, we may also group variables into tuples (which are seen as sets when convenient): let $\vec{x_1}, \ldots, \vec{x_n}$ be pairwise disjoint tuples of variables. Then, $X[\vec{x_1}, \ldots, \vec{x_n}]$ denotes a conjunction of atoms whose set of variables is exactly $\vec{x_1} \cup \cdots \cup \vec{x_n}$.

Definition 2.2. Let $\vec{x}$, $\vec{y}$ and $\vec{z}$ be pairwise disjoint tuples of variables, and $B[\vec{x}, \vec{y}]$ and $H[\vec{x}, \vec{z}]$ be conjunctions of atoms. An *(existential) rule* R is a first-order formula of the following form:

$$\forall \vec{x} \forall \vec{y}.\ B[\vec{x}, \vec{y}] \rightarrow \exists \vec{z}.\ H[\vec{x}, \vec{z}]$$

where B and H are called the *body* and the *head* of the rule, respectively. The set $\vec{x}$, which is shared between the body and the head, is called the *frontier* of the rule and is denoted by $\mathit{fr}(R)$. The set $\vec{z}$ is called the *existential variables* of R.

For the sake of readability, existential rules will be denoted by $B \rightarrow H$, with all variables appearing in B implicitly universally quantified, and variables in H but not in B existentially quantified. In addition, we assume atoms in the head to be pairwise distinct.

An existential rule without existential variables is called a *Datalog rule*.

Definition 2.3. A *knowledge base* is a tuple $\mathcal{K} = \langle \mathcal{R}, F \rangle$ where $\mathcal{R}$ is a rule set and F is a factbase.

Definition 2.4. Given sets of atoms F and F', a *homomorphism* from F to F' is a mapping $h : \mathit{var}(F) \rightarrow \mathit{term}(F')$ such that $h(F) \subseteq F'$.

A homomorphism is often denoted as a set of individual variable mappings. For example, the homomorphism from $\{x, y, z\}$ to $\{a, b\}$ such that $\sigma(x) = a$, $\sigma(y) = y$ and $\sigma(z) = b$ can be denoted by $\{x \rightarrow a, z \rightarrow b\}$.

Proposition 2.1. *A factbase F logically entails a factbase F' if and only if there is a homomorphism from F' to F.*

The proof of this result, known as the homomorphism theorem, can be found in [1].

Definition 2.5. Let F and F' be two sets of atoms such that $F' \subseteq F$. A homomorphism σ from F to F' is a *retraction* (from F to F') if $\sigma_{|F'} = id_{F'}$. If a retraction exists from F to F', F' is called a *retract* of F.

Note that, as per this definition, each factbase is a retract of itself.

2.2 Semantics

We rely on logical entailment as defined by classical first-order logic semantics. Given a formula f and an interpretation $\mathcal{M}$, the notation $\mathcal{M} \vDash f$ means that $\mathcal{M}$ is a model of f. Given two formulas f_1 and f_2, the notation $f_1 \vDash f_2$ means that f_1 entails f_2, i.e., if every model of f_1 is a model of f_2. A knowledge base $\langle \mathcal{R}, F \rangle$ is implicitly seen as the logical theory $\mathcal{R} \cup \{F\}$. We define below a few additional notions. In particular, it is convenient to see an interpretation as a (possibly infinite) set of atoms.

Definition 2.6. We say that a factbase F *satisfies* a rule $R = B \rightarrow H$ if for every homomorphism π from B to F, there is an extension $\hat{\pi}$ of π to the domain $\mathit{var}(B \cup H)$ such that $\hat{\pi}(H) \subseteq F$.

Definition 2.7. A *model* $\mathcal{M}$ of a knowledge base $\langle \mathcal{R}, F \rangle$ is a set of atoms such that $F \subseteq \mathcal{M}$ and every rule in $\mathcal{R}$ is satisfied by $\mathcal{M}$. A model $\mathcal{M}$ of a knowledge base $\mathcal{K}$ is said to be *universal* if for every model $\mathcal{M}'$ of $\mathcal{K}$ there is a homomorphism from $\mathcal{M}$ to $\mathcal{M}'$.

Universality is a useful notion, especially for the problem of Boolean conjunctive query entailment, where a Boolean conjunctive query is an existentially closed conjunction of atoms (hence has the same form as a factbase). This problem can be formulated the following way: given a Boolean conjunctive query q and a knowledge base $\mathcal{K}$, do we have $\mathcal{K} \vDash q$, i.e., $\mathcal{M}' \vDash q$ for every model $\mathcal{M}'$ of $\mathcal{K}$? On the one hand, if we have $\mathcal{M}$ a universal model of $\mathcal{K}$, and we establish that $\mathcal{M} \vDash q$, since for every model $\mathcal{M}'$ of $\mathcal{K}$, $\mathcal{M}' \vDash \mathcal{M}$, then we have $\mathcal{M}' \vDash q$ for every $\mathcal{M}'$ model of $\mathcal{K}$, hence $\mathcal{K} \vDash q$. On the other hand, if $\mathcal{M} \nvDash q$, this shows that $\mathcal{K} \nvDash q$. As such, we can reduce the problem of query entailment on every model of $\mathcal{K}$ to query entailment on only one model of $\mathcal{K}$ which is substantially easier.

2.3 The chase

The chase is a family of algorithms mainly used to solve factbase (or Boolean conjunctive query) entailment, that is to say to decide if an existentially closed formula is true for a given knowledge base. It relies on forward chaining, which is a process that infers new knowledge from already known facts and rules, until it finds what it looks for or it cannot infer any new knowledge. In fact, if it halts, it produces a universal model of the knowledge base (except in the case of the so-called equivalent chase, which produces a superset of a universal model, however this is not an issue for our work). Since the problem of factbase entailment is semi-decidable [3], a sound and complete procedure that does not terminate is the best we can hope for, as there is no sound and complete procedure that terminates. To define it properly, we first introduce the notion of a trigger.

Definition 2.8. Let F be a factbase and $R = B \rightarrow H$ be an existential rule. A *trigger* t (for F) is a pair (R, π) such that π is a homomorphism from B to F.

To define the applicability of a trigger, we first need to define a safe extension of its homomorphism that maps every existential variable of the rule to a fresh one, i.e. that does not occur in the current factbase nor in the rules.

Definition 2.9. Let $t = (R, \pi)$ be a trigger. We define π^R as the extension of π such that, for all z existential variable in R, $\pi^R(z) = z_t$, with z_t being a fresh variable, unique for z and t.

Note that $\pi^R = \pi$ exactly when R is a Datalog rule.

Definition 2.10. Let $t = (R, \pi)$ be a trigger with $R = B \rightarrow H$, and F a factbase. We say that t is *applicable* on F if $\pi(B) \in F$ and $\pi^R(H) \nsubseteq F$. We denote $\pi(B)$ and $\pi^R(H)$ by *support*(t) and *output*(t), and refer to them as the *support* and the *output* of t, respectively.

Now, that we have a basic notion of applicability, we can define derivations, before refining the notion of applicability to exhibit different chase variants.

Definition 2.11. Let $\mathcal{K} = \langle F, \mathcal{R} \rangle$ be a knowledge base. A *derivation* from $\mathcal{K}$ is a (possibly infinite) sequence $\mathcal{D} = (\emptyset, F_0), (t_1, F_1), (t_2, F_2), \ldots$ where:

- $F_0 = F$.
- for each $i > 0$,
  - t_i is a trigger for the factbase F_{i-1}.
  - $F_i = F_{i-1} \cup \mathit{output}(t_i)$.

The set of all triggers appearing in $\mathcal{D}$ is denoted by *triggers*$(\mathcal{D})$. The resulting factbase is *res*$(\mathcal{D}) = \bigcup_i F_i$.

Note that *res*$(\mathcal{D})$ is well defined for every derivation $\mathcal{D} = (\emptyset, F_0), (t_1, F_1), \ldots$ (finite or not) because $\{F_i\}_{i>0}$ is an increasing sequence for $\subset$ for every derivation.

The *k-prefix* of a derivation $\mathcal{D}$ is the subsequence of $\mathcal{D}$ that contains every element up to the k^{th} element, and is denoted by $\mathcal{D}_{|k}$ (if $\mathcal{D} = (\emptyset, F), (t_1, F_1), ...$ then $\mathcal{D}_{|k} = (\emptyset, F), (t_1, F_1), ..., (t_k, F_k)$. We consider the derivation $(\emptyset, F)$ to be of length 0). We can also say that $\mathcal{D}$ is an *extension* of $\mathcal{D}_{|k}$.

The chase variants we will study are the oblivious chase (**O**), the semi-oblivious chase (**SO**), the restricted chase (**R**) and the equivalent chase (**E**). They differ from each other in how they constrain trigger applicability, as specified in the next definition.

Definition 2.12. Let $\mathcal{K} = \langle F, \mathcal{R} \rangle$ be a knowledge base, $\mathcal{D}$ be a derivation from $\mathcal{K}$, and $t = (R, \pi)$ a trigger. Let us assume t is applicable on $res(\mathcal{D})$. Then t is:

- ***O**-applicable* on $\mathcal{D}$ if $t \notin triggers(\mathcal{D})$.
- ***SO**-applicable* on $\mathcal{D}$ if there is no trigger $(R, \pi') \in triggers(\mathcal{D})$ such that $\pi_{|fr(R)} = \pi'_{|fr(R)}$.
- ***R**-applicable* on $\mathcal{D}$ if there is no retraction from $res(\mathcal{D}) \cup output(t)$ to $res(\mathcal{D})$.
- ***E**-applicable* on $\mathcal{D}$ if there is no homomorphism from $res(\mathcal{D}) \cup output(t)$ to $res(\mathcal{D})$.

Definition 2.13. Let $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}, \mathbf{R}, \mathbf{E}\}$. A derivation $\mathcal{D}$ for which every trigger $t_i \in triggers(\mathcal{D})$ is **X**-applicable on $\mathcal{D}_{|i-1}$ is called an ***X**-derivation*.

An **X**-derivation $\mathcal{D} = (\emptyset, F), (t_1, F_1), \ldots$ is called a *Datalog-first **X**-derivation* (in short Df-**X**-derivation) if for each $t_i = (R_i, \pi_i)$ such that R_i is not Datalog, there is no Datalog rule **X**-applicable on F_{i-1}.

Definition 2.14. Let $\mathbf{Y} \in \{\mathbf{O}, \mathbf{SO}, \mathbf{R}, \mathbf{E}\}$ and $\mathbf{X} \in \{\mathbf{Y}, \text{Df-}\mathbf{Y}\}$. The class of all **X**-derivations is called the ***X**-chase*.

Note that we may sometimes omit the **X** before derivation or chase if the context makes it clear, and we may also replace **X** by the full name of the variant. For instance, **R**-derivation may be referred to as restricted derivation.

2.4 Chase termination classes

The notion of termination is reliant on the notion of fairness, which intuitively means that no trigger application is indefinitely postponed.

Definition 2.15. An **X**-derivation $\mathcal{D}$ is *fair* if for each trigger t applicable on $\mathcal{D}_{|k}$, there exists $n > k$ such that t is not **X**-applicable on $\mathcal{D}_{|n}$. An **X**-derivation is *terminating* if it is both fair and finite.

There are a few cases to consider regarding the termination of the different chase variants.

Definition 2.16. Let $\mathcal{K} = \langle \mathcal{R}, F \rangle$ be a a knowledge base. We say that the **X**-chase terminates on $\mathcal{K}$ if every fair **X**-derivation from $\mathcal{K}$ is finite. We say that the **X**-chase is *sometimes terminating* on $\mathcal{K}$ if there exists a terminating **X**-derivation from $\mathcal{K}$.

Definition 2.17. Let $\mathcal{R}$ be a rule set. We say that the **X**-chase *terminates* (resp. is *sometimes terminating*) on $\mathcal{R}$ if for every factbase F, the **X**-chase terminates (resp. sometimes terminating) on $\langle F, \mathcal{R} \rangle$. We denote the class of all those rule sets by $\mathsf{CT}^{\mathbf{X}}_{\forall}$ (resp. $\mathsf{CT}^{\mathbf{X}}_{\exists}$).

Some of those classes of rule sets are overlapping for certain chase variants, as we will see in the following proposition, which establishes most of the relations we consider.

Proposition 2.2. *The following equalities and inclusions hold:*

1. $CT_{\exists}^{\boldsymbol{O}} = CT_{\forall}^{\boldsymbol{O}}$
2. $CT_{\exists}^{\boldsymbol{SO}} = CT_{\forall}^{\boldsymbol{SO}}$
3. $CT_{\exists}^{\boldsymbol{E}} = CT_{\forall}^{\boldsymbol{E}}$
4. $CT_{\forall}^{\boldsymbol{O}} \subset CT_{\forall}^{\boldsymbol{SO}} \subset CT_{\forall}^{\boldsymbol{R}} \subset CT_{\forall}^{Df\text{-}\boldsymbol{R}} \subset CT_{\exists}^{Df\text{-}\boldsymbol{R}} \subseteq CT_{\exists}^{\boldsymbol{R}} \subset CT_{\forall}^{\boldsymbol{E}}$

Proof. Most of those results can be found in [8]. For the specific references and the rest of the proof, see section A.1. □

Thus, the notion of sometimes termination is only relevant in the case of the (Datalog-first) restricted chase. Points 1, 2 and 3 also show that the Datalog-first restricted chase is the only Datalog-first variant we need to consider, since the others behave as in the general case regarding termination.

Note that we only prove $\mathsf{CT}_{\exists}^{\mathrm{Df}\text{-}\mathbf{R}} \subseteq \mathsf{CT}_{\exists}^{\mathbf{R}}$, because we did not find any example showing the properness of the inclusion. We even think that the two sets may be equal, and thus set the following conjecture:

Conjecture 2.1. $CT_{\exists}^{Df\text{-}\boldsymbol{R}} = CT_{\exists}^{\boldsymbol{R}}$

3 Chase termination and normalization

In many applications of existential rules, one may want to ensure certain syntactic properties on the rule set they are working with. For instance, one may want to restrict the head of rules to the minimal amount of knowledge possible while keeping equivalence, as seen in [2] where the notion of piece is introduced to formalize this notion, or in [9] where the same notion is used in backward chaining mechanisms based on rewriting a query with the rules, instead of expanding the factbase as in the chase. Another interesting property would be to have rules with only one atom in the head, as it simplifies relevant notions and techniques. Calì et al. introduce a decomposition that guaranties this property in [4] (Theorem 8.1). In the following, we will refer to those decompositions as normalization procedures, that is to say functions that take a rule set as input and output another rule set with the desired properties.

The termination of the chase is an undecidable problem [8]. Therefore, we are interested in knowing whether the above normalization procedures affect the termination of the different chase variants, and if they do, whether the sets of terminating rule sets w.r.t a chase variant are comparable before and after the procedures.

Definition 3.1. Consider $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}, \mathbf{R}, \mathrm{Df}\text{-}\mathbf{R}, \mathbf{E}\}$ and f a function that takes a rule set as an input and returns another rule set. We say that f *preserves the termination of the* $\boldsymbol{X}$*-chase* if, for every rule set $\mathcal{R}$,

$$\mathcal{R} \in \mathsf{CT}_{\forall}^{\mathbf{X}} \implies f(\mathcal{R}) \in \mathsf{CT}_{\forall}^{\mathbf{X}}$$

We say that f *may lead the* $\boldsymbol{X}$*-chase to gain termination* if there is a rule set $\mathcal{R}$ such that

$$\mathcal{R} \notin \mathsf{CT}_{\forall}^{\mathbf{X}} \wedge f(\mathcal{R}) \in \mathsf{CT}_{\forall}^{\mathbf{X}}$$

We will first study a translation that yields a logically equivalent rule set.

3.1 Single-piece translation

The single-piece translation is a normalization procedure that, by grouping atoms that (directly or indirectly) share an existential variable, splits a rule into rules containing the minimal "unit" of knowledge brought by an application of the original rule in the chase, while still producing an equivalent rule set (Proposition 3.1). It permits a simplification of the rules involved without introducing fresh predicates. We will show that it preserves the termination of all chase variants, except the (Datalog-first) restricted chase (Theorems 3.1, 3.2, 3.3 and 3.4), and it makes the semi-oblivious chase gain termination (Theorem 3.5).

Definition 3.2. The *piece graph* of a rule $R = B \to H$ is the graph whose vertices are the atoms appearing in H, and with an edge between h_1 and h_2 if there is an existential variable in $\mathit{var}(h_1) \cap \mathit{var}(h_2)$. A *(rule) piece* of R is the conjunction of atoms corresponding to a connected component of its piece graph.

Definition 3.3. The *single-piece translation* of a rule $R = B \to H$ is the set $\mathit{sp}(R) = \{B \to H_i \mid H_i \text{ is a rule piece of } R\}$. The *single-piece translation* $\mathit{sp}(\mathcal{R})$ of a rule set $\mathcal{R}$ is the set $\bigcup_{R \in \mathcal{R}} \mathit{sp}(R)$.

The single-piece transformation produces an equivalent rule set:

Proposition 3.1. *A rule set $\mathcal{R}$ is equivalent to the set $\mathit{sp}(\mathcal{R})$.*

Proof. See section A.2. □

Let us now see for each chase variant how the single-piece translation impacts its termination.

3.1.1 Oblivious and semi-oblivious chase

Let us first note that the oblivious and the semi-oblivious chase produce the same atoms regardless of the derivation, that is to say all derivations from the same factbase produce the same set of atoms. More specifically, for both chase variants, a trigger that is applicable at some step can only become unapplicable at a later step due to an applied trigger that has produced the same set of atoms, or an isomorphic set of atoms (i.e., which is the same up to the name of fresh variables). As such, it leads to the following definition:

Definition 3.4. For $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}\}$, and $\mathcal{K}$ a knowledge base, consider $\mathcal{D}$ an arbitrary fair derivation from $\mathcal{K}$. We define $\mathit{Ch}_{\mathbf{X}}(\mathcal{K}) = \mathit{res}(\mathcal{D})$.

Theorem 3.1. *The single-piece translation preserves the termination of the oblivious and the semi-oblivious chases.*

Proof idea. For $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}\}$, F a factbase, and $\mathcal{R}$ a rule set, we will first show by induction on some arbitrarily chosen chase sequence that there is an injective homomorphism from $\mathit{Ch}_{\mathbf{X}}(\langle \mathit{sp}(\mathcal{R}), F\rangle)$ to $\mathit{Ch}_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$. Knowing this, if the $\mathbf{X}$-chase terminates on $\langle \mathcal{R}, F\rangle$, the set of atoms produced is finite, which means the same is true for $\langle \mathit{sp}(\mathcal{R}), F\rangle$, and implies the result.

For the full proof, see section A.3.

3.1.2 Restricted chase

Theorem 3.2. *The single-piece translation does not preserve the termination of the restricted chase.*

Proof. The rule set $\mathcal{R} = \{A(x) \to \exists y.\ P(x, y), P(x, y) \to P(y, y) \wedge A(y)\}$ provides a counter example, as the restricted chase terminates on $\mathcal{R}$ but not on $sp(\mathcal{R})$. A full explanation of the example can be found in section A.4. □

3.1.3 Datalog-first restricted chase

Theorem 3.3. *The single-piece translation does not preserve the termination of the Datalog-first restricted chase.*

Proof. We can add dummy existential variables to make the previous example work here. Consider the following rule set $\mathcal{R}$:

$$A(x, d_1) \to \exists y \exists d_2.\ P(x, y, d_2) \qquad P(x, y, d_1) \to \exists d_2 \exists d_3.\ P(y, y, d_2) \wedge A(y, d_3)$$

Then, the single-piece transformation will produce the following rule set $sp(\mathcal{R})$:

$$\begin{array}{ll} A(x, d_1) \to \exists y \exists d_2.\ P(x, y, d_2) & P(x, y, d_1) \to \exists d_2.\ P(y, y, d_2) \\ P(x, y, d_1) \to \exists d_3.\ A(y, d_3) & \end{array}$$

This rule set will behave exactly like the on in the proof of Theorem 3.2, even with the Datalog-first restricted chase, because it features no Datalog rule. □

3.1.4 Equivalent chase

Theorem 3.4. *The single-piece translation preserves the termination of the equivalent chase.*

Proof. According to [5] (Theorem 7), the equivalent chase terminates exactly when the knowledge base admits a finite universal model (as the core chase, discussed in [5], terminates exactly when the equivalent chase does). As stated in Proposition 3.1, the single-piece transformation yields an equivalent rule set, thus the knowledge base resulting from the single-piece transformation admits a finite universal model if and only if the original one does, as if a theory admits a final universal model M, then M is also a universal model for every equivalent theory. From this, we deduce that the single-piece transformation preserves the termination of the equivalent chase. □

3.1.5 Termination gain

Theorem 3.5. *The single-piece translation may make the semi-oblivious, the restricted, and the Datalog-first restricted chases gain termination.*

Proof. The rule set $\{P(x, y) \to \exists z.\ P(x, z) \wedge Q(x, y)\}$ provides an example of such a behavior for the three chase variants considered. See section A.5 for a full explanation. □

This phenomenon cannot be witnessed with the oblivious and the equivalent chase.

Theorem 3.6. *For a rule set $\mathcal{R}$ and a factbase F, if the oblivious (resp. equivalent) chase does not terminate on $\mathcal{K} = \langle F, \mathcal{R} \rangle$, it does not on $\langle F, sp(\mathcal{R}) \rangle$ either.*

Proof. First, let us note that the argument used to prove Proposition 3.4 in the case of the equivalent chase applies to prove this result too: we showed the equivalence.

For the oblivious chase, we only provide a sketch. One can adapt the proof of Theorem 3.1 to show that there is an injective homomorphism from $Ch_{\mathbf{X}}(\langle \mathcal{R}, F \rangle)$ to $Ch_{\mathbf{X}}(\langle sp(\mathcal{R}), F \rangle)$. It is due to the fact that, contrarily to the semi-oblivious case, a trigger is applicable if and only if its output is not in the database, and this condition is easy to preserve during the induction, so the proof does not change much. □

3.2 One-way atomic decomposition

The single-piece transformation works very well, but one may need to produce rules with a single atom in the head. To do this, we consider a normalization that produces a conservative extension of the initial rule set.

Definition 3.5. Let $R = B \to (H_1 \wedge \ldots \wedge H_n)[\vec{y}]$ be a rule, where, for every i, H_i is an atom. The *one-way atomic decomposition* applied to R outputs the following rules, denoted by *1ad*(R):

- $B \to X_R[\vec{y}]$
- for each $i \leq n$, $X_R[\vec{y}] \to H_i$

where X_R is a fresh predicate of arity $|\vec{y}|$.

The decomposition extends naturally to rule sets. For a rule set $\mathcal{R}$, we denote by *1ad*($\mathcal{R}$) the rule set produced by the one-way atomic decomposition.

The one-way atomic decomposition produces what is called a conservative extension.

Definition 3.6. Let $\mathcal{R}$ and $\mathcal{R}'$ be two rule sets. We say that $\mathcal{R}'$ is a *conservative extension* of $\mathcal{R}$ if, for any factbase F that only contains predicates appearing in $\mathcal{R}$,

1. The restriction of any model of $\langle \mathcal{R}', F \rangle$ to the predicates appearing in $\mathcal{R}$ is a model of $\langle \mathcal{R}, F \rangle$.
2. Any model $\mathcal{M}$ of $\langle \mathcal{R}, F \rangle$ can be extended into a model of $\langle \mathcal{R}', F \rangle$ that shares the same domain and agrees with $\mathcal{M}$ on the predicates in $\mathcal{R}$.

Hence, a conservative extension preserve query entailment (for queries expressed on the initial set of predicates). We will prove later, in Proposition 3.2, that *1ad*($\mathcal{R}$) is a conservative extension of the rule set $\mathcal{R}$.

3.2.1 Oblivious and semi-oblivious chase

Theorem 3.7. *The one-way atomic decomposition preserves the termination of the oblivious and the semi-oblivious chase.*

Proof. A stronger result will be proven in Theorem 3.10. □

3.2.2 Restricted and Datalog-first restricted chase

Theorem 3.8. *The one-way atomic decomposition does not preserve the termination of the restricted chase and the Datalog-first restricted chase.*

Proof. The rule set $\{P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)\}$ provides a counter example. For a full explanation, see section A.6. □

3.2.3 Equivalent chase

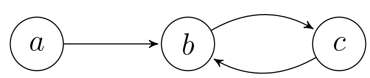
Theorem 3.9. *The one-way atomic decomposition does not preserve the termination of the equivalent chase.*

Proof. The previous counter-example, in Theorem 3.8, also works for the equivalent chase. The infinite fair derivation considered there is an **E**-derivation, as every trigger applied was **E**-applicable. □

3.3 Two-way atomic decomposition

As we just saw, even if the one-way atomic decomposition produces a conservative extension, it does not preserve the termination of the equivalent chase. It is a consequence of the fact that if the initial rule set admits a universal model, its extension to the fresh predicates introduced by the one-way atomic decomposition is a model, but not necessarily a universal model. The following example shows that.

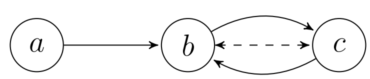
Consider the rule set $\{P(x,y) \to \exists z.\ P(y,z) \wedge P(z,y)\}$ and the factbase $\{P(a,b)\}$. This knowledge base admits a universal model, on which the equivalent chase halts, which is the factbase $\{P(a,b), P(b,c), P(c,b)\}$, which can be illustrated like this:



But the one-way decomposition yields the following rule set:

$$P(x,y) \to \exists z.\ X_R(y,z) \qquad X_R(x,y) \to P(x,y) \qquad X_R(x,y) \to P(y,x)$$

If we try to transform our universal model for the initial rule set into a model for the decomposed one, we get the following (with a dashed line for predicate X_R):



Which is not a universal model, and as such the equivalent chase will not find it. In fact, every universal model for this new rule set is infinite, and hence, the equivalent chase does not halt on it.

Thus, it motivates new definitions, both for a new extension and for a new normalization procedure.

Definition 3.7. Let $\mathcal{R}$ and $\mathcal{R}'$ be two rule sets and F a factbase. We say that $\mathcal{R}'$ is a *universal-conservative extension* of $\mathcal{R}$ if, for any factbase F that only contains predicates appearing in $\mathcal{R}$,

1. The restriction of any universal model of $\langle \mathcal{R}', F \rangle$ to the predicates appearing in $\mathcal{R}$ is a model of $\langle \mathcal{R}, F \rangle$.
2. Any universal model $\mathcal{M}$ of $\langle \mathcal{R}, F \rangle$ can be extended into a universal model of $\langle \mathcal{R}', F \rangle$ that shares the same domain and that agrees with $\mathcal{M}$ on the predicates in $\mathcal{R}$.

From this notion, that will ensure the termination of the equivalent chase, we define a normalization procedure that will produce a universal-conservative extension.

Definition 3.8. Let $R = B \to (H_1 \wedge \ldots \wedge H_n)[\vec{y}]$ be a rule, with H_i atoms for every i and j.
The output of the two-way atomic decomposition applied to R, denoted by *2ad*(R), is:

- $B \to X_R[\vec{y}]$
- for each $i \leq n$, $X_R[\vec{y}] \to H_i$
- $H_1 \wedge \ldots \wedge H_n \to X_R[\vec{y}]$, called the backwards rule

. where X_R is a fresh predicate of arity p.

The decomposition extends naturally to rule sets. For a rule set $\mathcal{R}$ we denote by *2ad*($\mathcal{R}$)the rule set produced by the one-way atomic decomposition.

Let us now show that two-way atomic decomposition indeed produces a universal-conservative extension.

Proposition 3.2. *The rule sets 1ad($\mathcal{R}$) and 2ad($\mathcal{R}$) are conservative extensions of the rule set $\mathcal{R}$.*

Proof. See section A.7. □

Proposition 3.3. *The rule set 2ad($\mathcal{R}$) is a universal-conservative extension of the rule set $\mathcal{R}$.*

Proof. For the full proof, see A.8. An interesting intuition to note is that the first point is true for any conservative extension, because extending a model (universal or not) then restricting it yields the same model as the original one, so (1) is quite easy to show. The issue for the second point is that restricting a model then extending it does not necessarily yield the original model. It is the case for the two-way atomic decomposition, but not for the one-way one, which explains the difference between the two procedures. □

3.3.1 Oblivious and semi-oblivious chase

Theorem 3.10. *Both atomic decompositions preserve the termination of the oblivious and the semi-oblivious chase.*

Proof idea. This proof, that you can read in section A.9, will work exactly like the proof of Theorem 3.1. The only variation is that for this transformation we only consider original predicates, and remove fresh predicates from $Ch_{\mathbf{X}}(\langle 1ad(\mathcal{R}), F\rangle)$.

3.3.2 Restricted chase

Theorem 3.11. *The two-way atomic decomposition does not preserve the termination of the restricted chase.*

Proof. The counter-example we used in the proof of Proposition 3.8, that is to say the rule set $\{P(x,y) \to \exists z.\ P(y,z) \wedge P(z,y)\}$, will work again here. See section A.10 for a full explanation. □

3.3.3 Datalog-first restricted chase

Theorem 3.12. *The two-way atomic decomposition preserves the termination of the Datalog-first restricted chase.*

Proof idea. For this proof, we will write Datalog-first derivations as an alternation of a trigger that introduces an existential variable, and a derivation closed under Datalog rules. This way, we will start with a derivation from $2ad(\mathcal{R})$, and construct a derivation from $\mathcal{R}$ that has the same number of existential triggers. We will then show that if the first one is fair, the second one is too.

Let $\mathcal{R}$ be a rule set and F be a factbase that does not contain any predicate introduced by the decomposition. We will discriminate the rules in $\mathcal{R}$ into the ones that introduce at least one existential variable, $\mathcal{R}^\exists$, and the Datalog ones, $\mathcal{R}^D$. In the following we will use the fact that any Datalog-first derivation can be decomposed following this schema: $(\emptyset, F), \mathcal{D}_0, (t_1, F_1), \mathcal{D}_1, \ldots, (t_n, F_n), \mathcal{D}_n, \ldots$ where forall i, t is a trigger that introduces an existential variable, and $\mathcal{D}_i^{ad}$ is a derivation closed under Datalog (i.e. that applies only Datalog rules) starting from the factbase F_i.

Lemma 3.1. *Let $\mathcal{D}^{ad} = (\emptyset, F^{ad}), \mathcal{D}_0^{ad}, (t_1^{ad}, F_1^{ad}), \mathcal{D}_1^{ad}, \ldots, (t_n^{ad}, F_n^{ad}), \mathcal{D}_n^{ad}, \ldots$ be a Datalog-first restricted derivation from $\langle 2ad(\mathcal{R}), F\rangle$. There is a Df-**R**-derivation $\mathcal{D} = (\emptyset, F), \mathcal{D}_0, (t_1, F_1), \mathcal{D}_1, \ldots, (t_n, F_n), \mathcal{D}_n, \ldots$ from $\langle \mathcal{R}, F\rangle$ such that:*

(1) There is an isomorphism h from $\gamma_\mathcal{R}(res(\mathcal{D}^{ad}))$ to $res(\mathcal{D})$.

(2) Forall i, if $t_i^{ad} = (B \to X_R, \pi)$, then $t_i = (B \to H, h \circ \pi)$, with $(B \to X_R) \in 2ad(B \to H)$.

(3) Forall i, $\mathcal{D}_i$ is $h(\mathcal{D}_i^{ad})$ in which we remove every trigger using a rule in $2ad(\mathcal{R}^\exists)$.

In addition, if $\mathcal{D}^{ad}$ is fair, then $\mathcal{D}$ is too.

Proof idea. The first result will be proven by an induction over n, constructing the isomorphism along the induction. Having done that, we will prove that if $\mathcal{D}^{ad}$ is fair, then $\mathcal{D}$ is by using the fact that the chase produces a universal model, and since we have a isomorphism between the results of the derivations, a trigger being applicable on one side means it is also applicable on the other, concluding the proof by contradiction. The full proof (probably the most interesting of this report) can be found in section A.11

Proof of the theorem. We can now conclude: If the Df-**R**-chase is not terminating on the knowledge base $\langle 2ad(\mathcal{R}), F\rangle$, let $\mathcal{D}^{ad}$ be a fair infinite derivation from $\langle 2ad(\mathcal{R}), F\rangle$. Applying Lemma 3.1, we can construct an infinite yet fair derivation $\mathcal{D}$ from $\langle \mathcal{R}, F\rangle$, thus the Df-**R**-chase is not terminating on $\langle \mathcal{R}, F\rangle$, which concludes the proof. □

3.3.4 Equivalent chase

Theorem 3.13. *The two-way atomic decomposition preserves the termination of the equivalent chase.*

Proof. As, according to Proposition 3.3, the two-way atomic decomposition is a universal-conservative extension, it will ensure the preservation of the termination of the equivalent chase. Indeed, if the equivalent chase terminates on the initial rule set, it admits a finite universal model. Thus, the decomposed rule set will admit a universal model of the same cardinality, and the equivalent chase will find it. □

3.4 Termination gain through atomic decompositions

As we studied termination gain through single-piece transformation, we are also interested in termination gain through both atomic decompositions. Unfortunately, there is no way for a non-terminating rule set to gain termination, as one can always replicate an infinite unfair derivation from the initial rule set with the normalized rule set. Indeed, applying every rule in $1ad(R)$ (or $2ad(R)$) at once for each rule R applied in the original derivation yields an infinite fair derivation. As such, neither the one-way nor the two-way atomic decomposition can extend the termination of a chase variant.

4 Conclusion and future outlook

In this paper, we have shown that one can take rule sets with a single piece in the head or even only one atom in the head while preserving the termination of every chase variant except the restricted chase. One can even consider single-atomic headed rules in the case of the restricted chase provided they only work with Datalog-first derivations. To do so, we introduced the notion of universal-conservative extension, that is to say a conservative extension which guaranties that if a finite universal model exists for the initial theory, one also exists for the extended one. The main caveat in our work is that we did not find any normalization procedure that would preserve the termination of the restricted chase. We thus set the following conjecture:

Conjecture 4.1. *There is no normalization procedure that preserves the termination of the restricted chase.*

Our intuition is that among rule sets on which the restricted chase terminates, those restricted to atomic-head rules seem to be strictly less expressive, which remains to be shown.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[2] J.-F. Baget, M. Leclère, M.-L. Mugnier, and E. Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.

[3] C. Beeri and M. Vardi. The implication problem for data dependencies. In *Proceedings of Automata, Languages and Programming, Eigth Colloquium (ICALP 1981)*, volume 115 of *Lecture Notes in Computer Science*, pages 73–85, 1981.

[4] A. Calì, G. Gottlob, and M. Kifer. Taming the infinite chase: Query answering under expressive relational constraints. In G. Brewka and J. Lang, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 70–80. AAAI Press, 2008.

[5] A. Deutsch, A. Nash, and J. B. Remmel. The chase revisited. In M. Lenzerini and D. Lembo, editors, *Proc. of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008*, pages 149–158. ACM, 2008.

[6] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. In D. Calvanese, M. Lenzerini, and R. Motwani, editors, *Database Theory - ICDT 2003, 9th International Conference, Siena, Italy, January 8-10, 2003, Proceedings*, volume 2572 of *Lecture Notes in Computer Science*, pages 207–224. Springer, 2003.

[7] T. Gogacz, J. Marcinkowski, and A. Pieris. All-instances restricted chase termination for linear tgds. *Künstliche Intell.*, 34(4):465–473, 2020.

[8] G. Grahne and A. Onet. Anatomy of the chase. *Fundam. Informaticae*, 157(3):221–270, 2018.

[9] M. König, M. Leclère, M.-L. Mugnier, and M. Thomazo. Sound, complete and minimal ucq-rewriting for existential rules. *CoRR*, abs/1311.3198, 2013.

[10] B. Marnette. Generalized schema-mappings: from termination to tractability. In J. Paredaens and J. 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.

[11] S. Rocher. *Querying Existential Rule Knowledge Bases: Decidability and Complexity.* PhD thesis, University of Montpellier, France, 2016.

[12] E. Salvat and M.-L. Mugnier. Sound and complete forward and backward chaining of graph rules. In P. W. Eklund, G. Ellis, and G. Mann, editors, *Conceptual Structures: Knowledge Representation as Interlingua, 4th International Conference on Conceptual Structures, ICCS '96, Sydney, Australia, August 19-22, 1996, Proceedings*, volume 1115 of *Lecture Notes in Computer Science*, pages 248–262. Springer, 1996.

[13] E. Tsamoura, D. Carral, E. Malizia, and J. Urbani. Materializing knowledge bases via trigger graphs. *Proc. VLDB Endow.*, 14(6):943–956, 2021.

Appendices

A Proofs

A.1 Proposition 2.2

Proof. Most of those results can be found in [8]. In the following we will refer to equalities by their number, and to the 6 strict inclusions numbered from left to right. The proofs of equalities 1 and 2, and inclusions 1 and 2 can be found in [8] as Proposition 4.4. and Theorem 4.5 (where the restricted chase is referred to as the standard chase). Equality 3 is Proposition 4.6 and inclusion 6 is Proposition 4.7 (with the knowledge that the equivalent and the core chase terminate on exactly the same inputs).

The results that are left to prove are inclusions 3, 4 and 5.

Inclusion 3 $\mathsf{CT}_{\forall}^{\mathbf{R}}$ is indeed a subset of $\mathsf{CT}_{\forall}^{\text{Df-}\mathbf{R}}$ because on a rule set on which any **R**-derivation is terminating, the class of Df-**R**-derivations being a subclass of the **R**-derivations, every Df-**R**-derivation will be terminating. The rule set $\{P(x,y) \to P(x,x),\ P(x,y) \to \exists z.\ P(y,z)\}$ proves that the inclusion is proper, as always applying the second rule before the first (with the same homomorphism) will yield an infinite yet fair derivation, but any Datalog-first restricted derivation will apply the first rule for each atom in the initial factbase then stop (the second rule will never be applied).

Inclusion 4 $\mathsf{CT}_{\forall}^{\text{Df-}\mathbf{R}}$ is a subset of $\mathsf{CT}_{\exists}^{\text{Df-}\mathbf{R}}$ because if every Df-**R**-derivation is terminating, there is a terminating Df-**R**-derivation. Regarding the properness of the inclusion, the rule set $\{P(x,y) \to \exists z.\ P(y,z),\ P(x,y) \to \exists w.\ P(y,w) \land P(w,y)\}$ will work, because there are no Datalog rules, hence the Datalog-first restricted and the restricted chases are the same on this input, and applying the second rule first for every atom in the factbase makes the chase terminate, but by applying successively the first then the second rule (with the same homomorphism) does not.

Inclusion 5 $\mathsf{CT}_{\exists}^{\text{Df-}\mathbf{R}}$ is a subset of $\mathsf{CT}_{\exists}^{\mathbf{R}}$ because every Datalog-first restricted derivation is a restricted derivation, so if there is a terminating Df-**R**-derivation, it is also a terminating **R**-derivation. □

A.2 Proposition 3.1

Proof. Let $R = B \to H$ be a rule in $\mathcal{R}$, $H_1, \ldots, H_m$ its pieces, and $R'_1, \ldots, R'_m$ the rules in $sp(R)$. Existential variables appearing in a piece are disjoint from those appearing in another piece (otherwise they would not be in different connected components in the piece graph of R). Thus, we can rewrite R in the following equivalent form:

$$R = \forall \vec{x} \forall \vec{y}.\ B[\vec{x}, \vec{y}] \to (\exists \vec{z}_1.\ H_1[\vec{x}, \vec{z}_1]) \land \ldots \land (\exists \vec{z}_m.\ H_m[\vec{x}, \vec{z}_m])$$

For all formulas A, B and C, $(A \to B) \land (A \to C)$ is equivalent to $A \to B \land C$, hence $R'_1 \land \ldots \land R'_m$ is equivalent to R. Every rule in $sp(\mathcal{R})$ being extracted from a rule in $\mathcal{R}$, we have the equivalence. □

A.3 Theorem 3.1

Proof. Let $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}\}$, $\mathcal{R}$ be a rule set, and F be a factbase. Let us show by induction that for any derivation $\mathcal{D} = (\emptyset, F), (t_1, F_1), \dots$ from $\langle sp(\mathcal{R}), F\rangle$, there is an injective homomorphism h from $res(\mathcal{D})$ to $Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$.

Step 0**:** We have $F \subseteq Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$.

Step n**:** Assume that the result is true up to step $n-1$. By induction hypothesis, there is an injective homomorphism h' from F_{n-1} to $Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$. The trigger t_n is $\mathbf{X}$-applicable on F_{n-1}, so $h'(support(t_n)) \subseteq Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$. Consider a fair derivation $\mathcal{D}'$ from $\langle \mathcal{R}, F\rangle$. By definition of $Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$, $res(\mathcal{D}') = Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$. Assume $t_n = (R, \pi)$, and consider the trigger $t = (R, \varphi)$, with $\varphi = h' \circ \pi$. We want to show that there is an injective homomorphism h that is an extension of h' such that $h(output(t_n)) \subseteq Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$. The argument will depend on the chase variant:

If $\mathbf{X} = \mathbf{O}$, as $h'(support(t_n)) \subseteq Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$ and $\mathcal{D}'$ is fair, t has its output in $res(\mathcal{D}')$ (otherwise t would be $\mathbf{O}$-applicable, which would contradict the fairness of $\mathcal{D}'$). As such, we define h as the extension of h' such that, for every existential variable z in R, $h(\pi^R(z)) = \varphi^R(z)$. h is indeed injective because h' was, and we injectively extend it. With this definition, $h(output(t_n)) = output(t)$, so we have the result we were looking for.

If $\mathbf{X} = \mathbf{SO}$, either $t \in triggers(\mathcal{D}')$, and we can apply the same argument as in the previous case, or it is not. In this second case, since t is not $\mathbf{SO}$-applicable on $\mathcal{D}'$ (because it is fair), there is a trigger $t' = (R, \varphi')$ in $triggers(\mathcal{D}')$ such that $\varphi'_{|fr(R)} = \varphi_{|fr(R)}$. As such, we define h as the extension of h' such that, for every existential variable z in R, $h(\pi^R(z)) = \varphi'^R(z)$. h is injective as in the previous point.

Since h is an extension of h', we still have $h(F_{n-1}) \subseteq Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$, and we just proved that $h(output(t_n)) \subseteq Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$. As such, $h(res(\mathcal{D})) \subseteq Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$.

Since the result of any derivation from $\langle sp(\mathcal{R}), F\rangle$ is injectively embedded in $Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$, we have that $|Ch_{\mathbf{X}}(\langle sp(\mathcal{R}), F\rangle)| \leq |Ch_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)|$. As such, if the $\mathbf{X}$-chase terminates on $\langle \mathcal{R}, F\rangle$, $Ch_{\mathbf{X}}(\langle sp(\mathcal{R}), F\rangle)$ is finite, which means that every derivation from $\langle sp(\mathcal{R}), F\rangle$ is finite, and the chase variant we consider terminates on $\langle sp(\mathcal{R}), F\rangle$. □

A.4 Theorem 3.2

Proof. Consider a rule set $\mathcal{R} = \{A(x) \to \exists y.\ P(x,y), P(x,y) \to P(y,y) \wedge A(y)\}$. The restricted chase terminates on this rule set: the first rule will never fire on the output of the second because it would be redundant, and none of those two rules can be applied on its own output, so the maximum length of any derivation from a factbase is twice the number of atoms in the factbase, which is finite.

The rule set $sp(\mathcal{R})$ contains the following rules:

$$R_1 = A(x) \to \exists y.\ P(x,y) \qquad R_2 = P(x,y) \to P(y,y)$$
$$R_3 = P(x,y) \to A(y)$$

Consider the factbase $A(a)$ and the following derivation (we denote the fresh variable created by R_1 at step i by z_i):

0. $(\emptyset, \{A(a)\})$

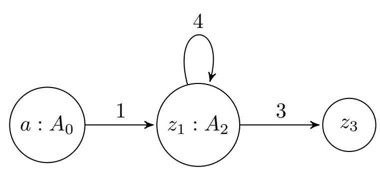
1. $((R_1, \{x \to a\}), \{A(a), P(a, z_1)\})$

2. $((R_3, \{x \to a, y \to z_1\}), \{A(a), P(a, z_1), A(z_1)\})$

3. $((R_1, \{x \to z_1\}), \{A(a), P(a, z_1), A(z_1), P(z_1, z_3)\})$

4. $((R_2, \{x \to a, y \to z_1\}), \{A(a), P(a, z_1), P(z_1, z_1), A(z_1), P(z_1, z_3)\})$

and so on. The next illustration is the state of the factbase after step 4. An arrow numbered i between x and y means the atom $P(x, y)$ is in the factbase since step i, and $x : A_i$ means $A(x)$ is in the factbase since step i.



From this we can repeat steps 2-3-4 indefinitely to produce an infinite yet fair derivation. Thus, the restricted chase is not terminating on this rule set, which concludes the proof. □

A.5 Theorem 3.5

Proof. Let us consider the rule set $\{P(x, y) \to \exists z.\ P(x, z) \land Q(x, y)\}$. Neither the (Datalog-first) restricted nor the semi-oblivious chase terminates on this knowledge base, but they both do after a single-piece transformation.

The single-piece transformation yields:

$$P(x, y) \to \exists z.\ P(x, z) \qquad\qquad P(x, y) \to Q(x, y)$$

With regard to the restricted chase, the first rule is not applicable, because the homomorphism that maps the new variable to y is a retraction. The second rule is Datalog, so it is only applicable a finite number of times. One can note that the Datalog-first restricted chase behaves exactly the same way.

With regard to the semi-oblivious chase, the first rule is only applicable once per atom using the P predicate, because its frontier is mapped is the same way when its body is mapped to its support and to its output. The Datalog rule does not matter for termination. □

A.6 Theorem 3.8

Proof. Consider the rule set $\{P(x, y) \to \exists z.\ P(y, z) \land P(z, y)\}$. The restricted chase is always terminating on this rule set, because if the rule fires, it cannot fire on its output. Thus, any derivation has a length that is at most the number of atoms in the initial factbase, so the chase always terminates. On this rule set, the one-way atomic decomposition yields the following rule set:

$$R_1 = P(x, y) \to \exists z.\ X_R(y, z) \qquad\qquad R_2 = X_R(x, y) \to P(x, y)$$
$$R_3 = X_R(x, y) \to P(y, x)$$

Consider the following derivation (we denote by z_i the fresh variable introduced by R_1 at step i):

0. $(\emptyset, \{P(a,b)\})$

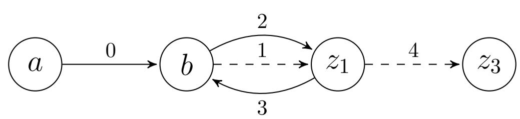
1. $((R_1, \{x \to a, y \to b\}), \{P(a,b), X_R(b,z_1)\})$

2. $((R_2, \{x \to b, y \to z_1\}), \{P(a,b), X_R(b,z_1), P(b,z_1)\})$

3. $((R_3, \{x \to b, y \to z_1\}), \{P(a,b), X_R(b,z_1), P(b,z_1), P(z_1,b)\})$

4. $((R_1, \{x \to b, y \to z_1\}), \{P(a,b), X_R(b,z_1), P(b,z_1), P(z_1,b), X_R(z_1,z_3)\})$

and so on. The next illustration is the state of the factbase after step 4. An arrow numbered i between x and y means the atom $P(x,y)$ is in the factbase since step i. A dashed arrow means the same but for the predicate X_R.



By repeating steps 2-3-4 one can construct a factbase that, at step $3i+1$, is a chain of $i+3$ terms, and the $i+1$ in the middle are pairwise in the same relationship as b and z_1 are at step 4. The last term in this construction is, like z_3 here, only linked to the others through X_R, which lets us repeat steps 2-3-4 on it to extend the chain.

Iterating the pattern infinitely, the last term disappears to let a fair and infinite derivation: although it is sometimes terminating, the restricted chase is not terminating on this input in some cases. One can also note that this derivation is Datalog-first, so the same result can be deduced for the Datalog-first chase. $\square$

A.7 Proposition 3.2

Proof. Let us first notice that for a knowledge base $\langle \mathcal{R}, F \rangle$, every model of $\langle \mathit{2ad}(\mathcal{R}), F \rangle$ is a model of $\langle \mathit{1ad}(\mathcal{R}), F \rangle$, because $\mathit{1ad}(\mathcal{R}) \subseteq \mathit{2ad}(\mathcal{R})$. As such, we only need to show that $\mathit{2ad}(\mathcal{R})$ is a conservative extension of $\mathcal{R}$ to have the result.

Let $\langle \mathcal{R}, F \rangle$ be a knowledge base. For (1), let $\mathcal{N}$ be a model of $\langle \mathit{2ad}(\mathcal{R}), F \rangle$. We denote its restriction to the predicates appearing in $\mathcal{R}$ by $\mathcal{M}$. We want to show that $\mathcal{M}$ is a model of $\langle \mathcal{R}, F \rangle$. First, $F \subseteq \mathcal{M}$, because F does not contain any fresh predicate. Let $R = B \to \bigwedge_i H_i$ be a rule and π a homomorphism from B to $\mathcal{M}$. Since $\mathcal{M}$ is a restriction of $\mathcal{N}$, π is a homomorphism from B to $\mathcal{N}$. Thus, since every rule in $\mathit{2ad}(\mathcal{R})$ is satisfied in $\mathcal{N}$, the rule $B \to X_R(\vec{x})$ is too, so the atom $\pi^R(X_R(\vec{x}))$ is in $\mathcal{N}$. Thus, since for every i, the rule $X_R \to H_i$ is satisfied, $\pi^R(H_i)$ is also in the database. As such, the homomorphism π^R is an extension of π such that for all i, $\hat{\pi}(H_i) \in \mathcal{M}$. Since no H_i features a fresh predicate, they are all in $\mathcal{M}$ too. Thus, every rule in $\mathcal{R}$ is satisfied in $\mathcal{M}$. As such, $\mathcal{M}$ is a model of $\langle \mathcal{R}, F \rangle$.

For (2), let $\mathcal{M}$ be a model of $\langle \mathcal{R}, F \rangle$. We extend $\mathcal{M}$ to $\mathcal{N}$ using the following method: for every rule $R = B \to H \in \mathcal{R}$, for every homomorphism π from H to $\mathcal{M}$, we add the atom $\pi(X_R(\vec{x}))$ to $\mathcal{M}$. With this definition, $\mathcal{M}$ and $\mathcal{N}$ share the same domain and agree on the predicates in $\mathcal{R}$. In addition, $\mathcal{N}$ contains F. Then, let R be a rule. Let us show that R is satisfied by case analysis on the form of R:

If $R = B \to X_R(\vec{x})$: If there is a homomorphism π from B to $\mathcal{N}$, then it is a homomorphism from B to $\mathcal{M}$. As such, there is an extension $\hat{\pi}$ of π such that $\hat{\pi}(H) \in \mathcal{M}$ because $\mathcal{M}$ is a model. Thus, $\hat{\pi}(X_R(\vec{x})) \in \mathcal{M}$, so R is satisfied.

If $R = X_R(\vec{x}) \to H_i$: If there is a homomorphism π from $X_R(\vec{x})$ to $\mathcal{N}$, then π is a homomorphism from H to $\mathcal{M}$ (else we would not have added $\pi(X_R(\vec{x}))$ to construct $\mathcal{N}$), so every atom in $\pi(H)$ is in $\mathcal{N}$, which means in particular $\pi(H_i) \in \mathcal{N}$, so R is satisfied.

If $R = H \to X_R(\vec{x})$: If there is a homomorphism π from H to $\mathcal{M}$, we added the atom $\pi(X_R(\vec{x}))$ to construct $\mathcal{N}$, so R is satisfied.

Thus, $\mathcal{N}$ is a model of $\langle \mathit{2ad}(\mathcal{R}), F\rangle$. □

A.8 Proposition 3.3

Proof. Let $\langle \mathcal{R}, F\rangle$ be a knowledge base. For (1), let $\mathcal{V}$ be a universal model of $\langle \mathit{2ad}(\mathcal{R}), F\rangle$. We want to show that $\mathcal{U}$, the restriction of $\mathcal{V}$ to the predicates appearing in $\mathcal{R}$, is a universal model of $\langle \mathcal{R}, F\rangle$. First, since $\langle \mathit{2ad}(\mathcal{R}), F\rangle$ is a conservative extension of $\langle \mathcal{R}, F\rangle$, $\mathcal{U}$ is a model. To show its universality, we will show that it can be homomorphically embedded in any other model of $\langle \mathcal{R}, F\rangle$. Let $\mathcal{M}$ be another model of $\langle \mathcal{R}, F\rangle$. We can extend $\mathcal{M}$ into a model $\mathcal{N}$ of $\langle \mathit{2ad}(\mathcal{R}), F\rangle$. Then, since $\mathcal{V}$ is a universal model, there is a homomorphism h from $\mathcal{V}$ to $\mathcal{N}$. Since $\mathcal{M}$ and $\mathcal{N}$ (resp. $\mathcal{U}$ and $\mathcal{V}$) share the same domain, h is a mapping from $\mathcal{U}$ to $\mathcal{M}$. Let $P(\vec{x})$ be an atom in $\mathcal{U}$. As such, P is a predicate in $\mathcal{R}$, and is also in $\mathcal{V}$. Since h is a homomorphism, $h(P(\vec{x})) \in \mathcal{N}$. Since our restriction only removes atoms featuring predicates not in $\mathcal{R}$, $h(P(\vec{x})) \in \mathcal{M}$, proving $\mathcal{U}$ is a universal model.

For (2), let $\mathcal{U}$ be a universal model of $\langle \mathcal{R}, F\rangle$. Consider the extension $\mathcal{V}$ of $\mathcal{U}$ defined in the context of $\mathit{2ad}(\mathcal{R})$ being a conservative extension of $\mathcal{R}$. It is a model of $\langle \mathit{2ad}(\mathcal{R}), F\rangle$. We want to show that it is a universal model. Let $\mathcal{N}$ be a model of $\langle \mathit{2ad}(\mathcal{R}), F\rangle$ and $\mathcal{M}$ the restriction of $\mathcal{N}$ that is a model of $\langle \mathcal{R}, F\rangle$. Since $\mathcal{U}$ is a universal model, there is a homomorphism h from $\mathcal{U}$ to $\mathcal{M}$. We show that h is also a homomorphism from $\mathcal{V}$ to $\mathcal{N}$. It is a mapping from $\mathcal{V}$ to $\mathcal{N}$, and for any atom $P(\vec{x})$ in $\mathcal{V}$ that features no fresh predicate, $h(P(\vec{x})) \in \mathcal{N}$. Let $X_R(\vec{y})$ be an atom in $\mathcal{V}$ that features a fresh predicate, with $R = B \to \bigwedge_i H_i$ the rule such that $(B \to X_R) \in \mathit{2ad}(R)$. Since $\mathcal{V}$ is a model, it features every $H_i(\vec{y_i})$ with $\vec{y_i}$ the restriction of $\vec{y}$ to the variables of H_i (because the rules $X_R \to H_i$ are all satisfied). As such, $\mathcal{N}$ also features those atoms. Since it is a model of $\langle \mathit{2ad}(\mathcal{R}), F\rangle$, it satisfies the rule $\bigwedge_i H_i \to X_R$, so $X_R(\vec{y}) \in \mathcal{N}$. Thus, $\mathcal{V}$ is a universal model. □

A.9 Theorem 3.10

For a set of rules $\mathcal{R}$ and a set of atoms A, let $\gamma_{\mathcal{R}}(A)$ be the maximal subset of A such that $\mathit{Preds}(A) \subseteq \mathit{Preds}(\mathcal{R})$. For instance, if $\mathcal{R} = \{P(a) \to Q(a)\}$ and $A = \{P(a), Q(b), Q(c), R(b, c, d), S(a, b)\}$, then $\gamma_{\mathcal{R}}(A) = \{P(a), Q(b), Q(c)\}$.

Proof. First note that in the oblivious and the semi-oblivious, firing a rule cannot prevent another one from firing. As such, since $\mathit{1ad}(\mathcal{R}) \subseteq \mathit{2ad}(\mathcal{R})$, if the oblivious (resp. semi-oblivious) chase terminates on $\mathit{2ad}(\mathcal{R})$, it will also terminate on $\mathit{1ad}(\mathcal{R})$. We can thus prove the result only for the two-way atomic decomposition.

Let $\mathbf{X} \in \{\mathbf{O}, \mathbf{SO}\}$, $\mathcal{R}$ be a rule set and F a factbase. Let us show by induction that for a derivations $\mathcal{D} = (\emptyset, F), (t_1, F_1), \ldots$ from $\langle \mathit{1ad}(\mathcal{R}), F\rangle$, there is an injective homomorphism h such that $h(\gamma_{\mathcal{R}}(\mathit{res}(\mathcal{D}))) \subseteq \mathit{Ch}_{\mathbf{X}}(\langle \mathcal{R}, F\rangle)$.

Step 0**:** $F \subseteq \mathit{Ch}_{\mathbf{X}}(\langle \mathcal{R}, F \rangle)$.

Step n**:** Assume the result up to step $n-1$. Thus, there is a homomorphism h' such that $h'(\gamma_{\mathcal{R}}(F_{n-1})) \subseteq \mathit{Ch}_{\mathbf{X}}(\langle \mathcal{R}, F \rangle)$. Depending on the trigger $t_n = (R^{ad}, \pi)$, with $R = B \to \bigwedge_i H_i$, we distinguish three cases:

If $R^{ad} = B \to X_R(\vec{x})$**,** $\gamma_{\mathcal{R}}(F_{n-1}) = \gamma_{\mathcal{R}}(F_n)$ so we have the result.

If $R^{ad} = X_R(\vec{x}) \to H_i$**,** since t_n is **X**-applicable on F_{n-1}, its support is in F_{n-1}. In addition, since F does not contain any fresh predicate, there is a $k < n$ such that $t_k = (B \to X_R(\vec{x}), \varphi)$ and $\pi = (\varphi^{R^{ad}})_{|var(R)}$ (if t_n's support was introduced by the backwards rule, it would not be applicable). Therefore, $h'(\gamma_{\mathcal{R}}(\mathit{support}(t_k))) \subseteq \mathit{Ch}_{\mathbf{X}}(\langle \mathcal{R}, F \rangle)$. Since the support of t_k is the body of the initial rule, $\gamma_{\mathcal{R}}(\mathit{support}(t_k)) = \mathit{support}(t_k)$, which implies that, if we set $t = (R, h \circ \varphi)$, $\mathit{support}(t) \subseteq \mathit{Ch}_{\mathbf{X}}(\langle \mathcal{R}, F \rangle)$. Similarly to the proof of Theorem 3.1, we can show that we can extend h' to a h such that $h(\mathit{output}(t_n)) \subseteq \mathit{Ch}_{\mathbf{X}}(\langle \mathcal{R}, F \rangle)$ (because in the **O**-chase, t was applied, and in the **SO**-chase we can find a trigger that shared t's frontier that was applied). We thus have the result.

If $R^{ad} = \bigwedge_i H_i \to X_R(\vec{x})$**,** as in the case of a rule of the form $B \to X_R(\vec{x})$, $\gamma_{\mathcal{R}}(F_{n-1}) = \gamma_{\mathcal{R}}(F_n)$ so we have the result by induction hypothesis.

We conclude with the same argument of cardinality as in Theorem 3.1. $\square$

A.10 Theorem 3.11

Proof. Let us consider the counter-example in the proof of Proposition 3.8, the rule set $\{P(x, y) \to \exists z.\ P(y, z) \wedge P(z, y)\}$. The restricted chase is again terminating on this knowledge base.

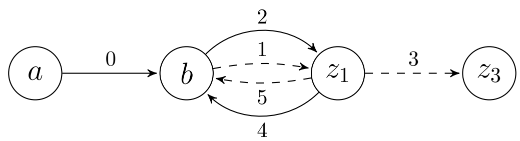
On this rule set, the two-way atomic decomposition will yield the following rule set:

$$R_1 = P(x,y) \to \exists z.\ X_R(y,z) \qquad R_3 = X_R(x,y) \to P(y,x)$$
$$R_2 = X_R(x,y) \to P(x,y) \qquad R_4 = P(x,y) \wedge P(y,x) \to X_R(x,y)$$

Consider the following derivation (we denote the variable x created at step i by x_i):

0. $(\emptyset, \{P(a,b)\})$
1. $((R_1, \{x \to a, y \to b\}), \{P(a,b), X_R(b,z_1)\})$
2. $((R_2, \{x \to b, y \to z_1\}), \{P(a,b), X_R(b,z_1), P(b,z_1)\})$
3. $((R_1, \{x \to b, y \to z_1\}), \{P(a,b), X_R(b,z_1), P(b,z_1), X_R(z_1,z_3)\})$
4. $((R_3, \{x \to b, y \to z_1\}), \{P(a,b), X_R(b,z_1), P(b,z_1), P(z_1,b), X_R(z_1,z_3)\})$
5. $((R_4, \{x \to z_1, y \to b\}),$
$\{P(a,b), X_R(b,z_1), X_R(z_1,b), P(b,z_1), P(z_1,b), X_R(z_1,z_3)\})$

and so on. The next illustration is the state of the factbase after step 5. An arrow numbered i between x and y means the atom $P(x, y)$ was in the factbase since step i. A dashed arrow means the same but for the predicate X_R.



In this case, we will repeat steps 2-3-4-5 to construct a factbase that, at step $4i+1$, is the same kind of chain of $i+3$ terms as in Proposition 3.8, with a last term enabling the pattern to continue.

Following the same line of reasoning as previously, we can construct an infinite yet fair **R**-derivation. $\square$

A.11 Lemma 3.1

Proof. We will prove the first result by induction over n.

$n = 0$ If $\mathcal{D}^{ad} = (\emptyset, F)$, then we can take $\mathcal{D} = \mathcal{D}^{ad}$ to have the result.

$n > 0$ If the result is true for $n-1$, and $\mathcal{D}^{ad} = (\emptyset, F), \mathcal{D}_0^{ad}, (t_1^{ad}, F_1^{ad}), \mathcal{D}_1^{ad}, \ldots, (t_n^{ad}, F_n^{ad}), \mathcal{D}_n^{ad}$, by induction hypothesis, there is a derivation $\mathcal{D}' = (\emptyset, F), \mathcal{D}_0, (t_1, F_1), \mathcal{D}_1, \ldots, (t_{n-1}, F_{n-1}), \mathcal{D}_{n-1}$ and an isomorphism h' such that (1), (2) and (3) hold for $\mathcal{D}^{ad}_{|n-1}$ with $\mathcal{D}'$.

Assume $t_n^{ad} = (B \to X_R, \pi)$ (note that this is the only possible form for any t_i^{ad}, as they are the only existential triggers possible). Let us define $t_n = (B \to H, \phi)$, with $\phi = h' \circ \pi$, as in (2) and $\mathcal{D}_n$ as in (3), and denote by $\mathcal{D}$ the derivation composed of the combination of $\mathcal{D}'$, (t_n, F_n) and $\mathcal{D}_n$.

We need to check that the trigger t_n is **R**-applicable on $\mathcal{D}_{n-1}$. First, $\mathit{support}(t_n) = h'(\mathit{support}(t_n^{ad}))$, and since t_n^{ad} is applicable on $\mathcal{D}^{ad}_{n-1}$, it follows that $\mathit{support}(t_n^{ad}) \in \mathit{res}(\mathcal{D}^{ad}_{n-1})$. As such, $\mathit{support}(t_n) \in \mathit{res}(\mathcal{D}_{n-1})$. We then need to check the condition specific to the restricted chase, but to do so we need to first extend h'.

We define h as the extension of h' defined by, for every existential variable z appearing in R, $h(\pi^R(z)) = \phi^R(z)$. It is thus still bijective, but we still need to prove that it is a homomorphism.

- From $\mathcal{D}$ to $\mathcal{D}^{ad}$: Each new atom A created by t_n was not in $\mathit{res}(\mathcal{D}_{n-1})$, and since we have an isomorphism, there is no atom A^{ad} in $\mathit{res}(\mathcal{D}^{ad}_{n-1})$ such that $h(A^{ad}) = A$. Thus, the rule $X_R \to A^{ad}$ is applicable after the application of t_n^{ad}, and since the derivation is Datalog-first this application will happen in $\mathcal{D}_n^{ad}$.

  If $\mathcal{D}_n$ creates an atom, it means it is from a Datalog rule R, and the rules in $\mathit{2ad}(R)$ will fire in $\mathcal{D}_n^{ad}$ since they are Datalog (and we have an isomorphism, so any applicable rule in one factbase is in the other).
- From $\mathcal{D}^{ad}$ to $\mathcal{D}$: Since t_n^{ad} will only create a X_R predicate, it is not important for our isomorphism.

  If $\mathcal{D}_n^{ad}$ creates an atom A^{ad}, we distinguish two cases:
  - Either the rule introducing A^{ad} is in $\mathit{2ad}(\mathcal{R}^{\exists})$, and in this case the rule introducing A^{ad} is in $\mathit{2ad}(B \to H)$, because every Datalog rule in $\mathit{2ad}(R')$ (with $R' \in \mathcal{R}^{\exists}$) is applicable only if $B' \to X_{R'}$ was applied first (with $R' = B' \to H'$). One could argue that $H' \to X_{R'}$ also produces an $X_{R'}$ atom which could lead to new atoms, but none of the $X_{R'} \to H'_i$ would be

applicable since they are in the premises of $H' \to X_{R'}$. As such, $h(A^{ad})$ is created by t_n in $\mathcal{D}$.

- Or the rule introducing A^{ad} is in *2ad*$(\mathcal{R}^D)$, and in this case the original rule is in $\mathcal{D}_n$, and it will produce $h(A^{ad})$.

As such, (1) is preserved. We can now settle the **R**-applicability of t_n. If there was a retraction σ from $res(\mathcal{D}_{n-1}) \cup output(t_n)$ to $res(\mathcal{D}_{n-1})$, then $\sigma^{ad} = h^{-1} \circ \sigma \circ h$ would be a retraction from $res(\mathcal{D}^{ad}_{n-1}) \cup output(t^{ad}_n)$ to $res(\mathcal{D}^{ad}_{n-1})$. Indeed, forall $u \in term(res(\mathcal{D}^{ad}_{n-1}))$, $h(u) \in term(res(\mathcal{D}_{n-1}))$ so $\sigma(h(u)) = h(u)$, and $\sigma^{ad}(u) = u$.

We now want to prove that $\sigma^{ad}(\pi^R(X_R[\vec{x}])) \in res(\mathcal{D}^{ad}_{n-1})$, as it is the only new atom introduced by t^{ad}_n. Assume the contrary. Then, since $\mathcal{D}^{ad}_{n-1}$ is closed under Datalog and the rule $\bigwedge_i H_i \to X_R$ is in the rule set, then there is a i such that $\sigma^{ad}(\pi^R(H_i[\vec{y}])) \notin \mathcal{D}^{ad}_{n-1}$. Thus, $\sigma \circ h \circ \pi(H_i[\vec{y}]) = \sigma(\phi(H_i[\vec{y}])) \notin \mathcal{D}_{n-1}$. Since $\phi(H_i[\vec{y}])$ is an atom produced by t_n, it contradicts the fact that σ is a retraction. As such, t_n is **R**-applicable on $\mathcal{D}_{n-1}$.

Since we defined everything to preserve (2) and (3), they are also true, and $\mathcal{D}$ is indeed a derivation.

We still have to prove that if $\mathcal{D}^{ad}$ is fair then $\mathcal{D}$ is too. Assume that $\mathcal{D}^{ad}$ is fair. As such, $res(\mathcal{D}^{ad})$ is a universal model of *2ad*$(\mathcal{R})$. Let $\mathcal{D}$ be the result of the previous lemma. According to (1), there is a isomorphism between $res(\mathcal{D}^{ad})$ in which we remove the new predicates and $res(\mathcal{D})$. Thus, according to Proposition 3.3, $res(\mathcal{D})$ is a universal model of $\mathcal{R}$.

If $res(\mathcal{D})$ is a universal model of $\mathcal{R}$, then no rule is **R**-applicable on $\mathcal{D}$, because if there was an applicable trigger (R, π), then $\pi(R)$ would not be satisfied in the model, which would contradict that it is a model. Thus, $\mathcal{D}$ is fair. $\square$