Normalisations of Existential Rules: Not so Innocuous!

David Carral\(^1\), Lucas Larroque\(^2\)*, Marie-Laure Mugnier\(^3\), Michaël Thomazo\(^3\)

\(^1\)LIRMM, Inria, University of Montpellier, CNRS, Montpellier, France
\(^2\)DI ENS, ENS, CNRS, PSL University, Paris, France
\(^3\)Inria, DI ENS, ENS, CNRS, PSL University, Paris, France

\{david.carral, michael.thomazo\}@inria.fr, lucas.larroque@ens.psl.eu, mugnier@lirmm.fr

Abstract

Existential rules are an expressive knowledge representation language mainly developed to query data. In the literature, they are often supposed to be in some normal form that simplifies technical developments. For instance, a common assumption is that rule heads are atomic, i.e., restricted to a single atom. Such assumptions are considered to be made without loss of generality as long as all sets of rules can be normalised while preserving entailment. However, an important question is whether the properties that ensure the decidability of reasoning are preserved as well. We provide a systematic study of the impact of these procedures on the different chase variants with respect to chase (non-)termination and FO-rewritability. This also leads us to study open problems related to chase termination of independent interest.

1 Introduction

Existential rules are an expressive knowledge representation language mainly developed to query data (Baget et al. 2009; Calì, Gottlob, and Lukasiewicz 2009). Such rules are an extension of first-order function-free Horn rules (like those of Datalog) with existentially quantified variables in the rule heads, which allows to infer the existence of unknown individuals.

Querying a knowledge base (KB) \( K = (R, F) \), where \( R \) is a set of existential rules and \( F \) a set of facts, consists in computing all the answers to queries that are logically entailed from \( K \). Two main techniques have been developed, particularly in the context of the fundamental (Boolean) conjunctive queries. The chase is a bottom-up process that expands \( F \) by rule applications from \( R \) towards a fixpoint. It produces a universal model of \( K \), i.e., a model of \( K \) that homomorphically maps to all models of \( K \), which is therefore sufficient to decide query entailment. Query rewriting is a dual technique, which consists in rewriting a query \( q \) with the rules in \( R \) into a query \( q' \) such that \( q \) is entailed by \( K \) if and only if \( q' \) is entailed by \( F \) solely.

Conjunctive query answering being undecidable for existential rules (Beeri and Vardi 1981), both the chase and query rewriting may not terminate. There is however a wide range of rule subclasses defined by syntactic restrictions that ensure chase termination on any set of facts (see, e.g., various acyclicity notions in (Grau et al. 2013)) or the existence, for any conjunctive query, of a (finite) rewriting into a first-order query, a property referred to as FO-rewritability (Calvanese et al. 2007).

In the literature, existential rules are often supposed to be in some normal form that simplifies technical developments. For instance, a common assumption is that rule heads are atomic, i.e., restricted to a single atom. On the one hand, the use of single-head rules greatly simplifies the presentation of theoretical arguments (e.g., (Calì, Gottlob, and Pieris 2012)). On the other hand, this restriction may also simplify implementations; e.g., the optimisation procedure presented in (Tsamoura et al. 2021) exploits single-head rules to clearly establish the provenance of each fact computed during the chase. Moreover, after normalisation, we can apply existing methods to effectively determine if the chase terminates for an input single-head existential rule set if this set is linear (Leclère et al. 2019) or guarded (Gogacz, Marcinkowski, and Pieris 2020). Normal form assumptions are often made without loss of generality as long as all sets of rules can be normalised while preserving all interesting entailments. However, an important question is whether the properties that ensure the decidability of reasoning are preserved as well. In particular, what is the impact of common normalisation procedures on fundamental properties like chase termination or FO-rewritability?

In fact, the chase is a family of algorithms, which differ from each other in their termination properties. Here, we consider the four main chase variants, namely: the oblivious chase (Calì, Gottlob, and Kifer 2008a), the semi-oblivious (aka skolem) chase (Marnette 2009), the restricted (aka standard) chase (Fagin et al. 2003) and the core chase (Deutsch, Nash, and Remmel 2008). As the core chase has the inconvenience of being non-monotonic (i.e., the produced set of facts does not grow monotonically), we actually study a monotonic variant that behaves similarly regarding termination, namely the equivalent chase (Rocher 2016). The ability of a chase variant to halt on a given KB is directly related to its power of reducing logical redundancies introduced by rules. The oblivious chase blindly performs all possible rule applications, while the equivalent chase terminates exactly when the KB admits a finite universe model. The other variants lie between these two extremes. For practical efficiency reasons, the most implemented variant is the restricted chase. However, it is the only variant sensitive to

\*Our work started when Lucas was intern at LIRMM-Inria.
the order of rule applications: for a given KB, there may be sequences of rule applications that terminate, while others do not. We study a natural strategy, called Datalog-first restricted chase, which prioritises Datalog rules (whose head does not include existential quantifiers) thus achieving termination in many real-world cases (Carral, Dragoste, and Krötzsch 2017). Moreover, experiments have shown that it does not include existential quantifiers) thus achieving termination in many real-world cases (Carral, Dragoste, and Krötzsch 2017). Moreover, experiments have shown that it indeed is a very efficient strategy (Urbani et al. 2018).

On the other hand, we consider two well-known normalisation procedures of a set of rules: single-piece decomposition, which breaks rule heads into subsets called pieces and outputs a logically equivalent rule set (Baget et al. 2011; König et al. 2013); and atomic-decomposition, which requires to introduce fresh predicates and outputs a set of atomic-head rules that form a conservative extension of the original set, hence preserve entailment (Cali, Gottlob, and Pieris 2012).

**Contributions.** We provide a systematic study of the impact of these procedures on the different chase variants with respect to chase (non-)termination and FO-rewritability. This also leads us to solve some open problems related to chase termination, which are of independent interest.

Although the relationships between most Chase variants with respect to chase termination are well understood (Grahne and Onet 2018), the question remained open regarding the restricted chase and its Datalog-first version. Unexpectedly, we found that Datalog-first strategies are not always optimal: we exhibit a rule set \( R \) such that the restricted chase has a terminating sequence on any KB \( \langle R, F \rangle \) but there is a KB \( \langle R, F' \rangle \) on which no Datalog-first strategy terminates (Section 3).

While it appears that none of the considered decompositions influences FO-rewritability, the situation is very different concerning chase termination, as summarized in Table 1. Note that we distinguish between two behaviors for restricted chases: \( \exists \) means that at least one chase sequence terminates on any KB (“sometimes-termination”) and \( \forall \) that all sequences terminate on any KB (“termination”). Single-piece-decomposition (Section 4) has no impact on the oblivious and equivalent chases (noted \( + \)), a positive impact on the semi-oblivious chase (noted \( \neq \)), and an erratic impact on (Datalog-first)-restricted chase (noted \( \neq \)).

<table>
<thead>
<tr>
<th></th>
<th>( \emptyset )</th>
<th>( \text{SO} )</th>
<th>( \text{3-R} )</th>
<th>( \text{3DF-R} )</th>
<th>( \forall \text{-DF-R} )</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Piece</td>
<td>( + )</td>
<td>( \neq )</td>
<td>( \neq )</td>
<td>( \neq )</td>
<td>( \neq )</td>
<td>( = )</td>
</tr>
<tr>
<td>One-Way</td>
<td>( = )</td>
<td>( = )</td>
<td>( = )</td>
<td>( = )</td>
<td>( = )</td>
<td>( = )</td>
</tr>
<tr>
<td>Two-Way</td>
<td>( = )</td>
<td>( + )</td>
<td>( = )</td>
<td>( = )</td>
<td>( = )</td>
<td>( = )</td>
</tr>
</tbody>
</table>

Table 1: Impact on chase termination. Chase variants are denoted as follows: \( \emptyset \): Oblivious; \( \text{SO} \): semi-oblivious; \( \text{DF}\text{-}R \): (Datalog-first) restricted; \( E \): equivalent.

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

An *atom* is a FOL formula \( P(t) \) with \( P \) a \( |t| \)-ary predicate and \( t \in \text{Terms} \). For a formula \( U \), we write \( U[x] \) to indicate that \( x \) is the set of all free variables that occur in \( U \).

**Definition 1.** An (existential) rule \( R \) is a FOL formula

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

for some atoms \( B \) and \( H \). The set \( \vec{x} \) is the *frontier* of \( R \). If \( \vec{x} \) is empty, then \( R \) is a Datalog rule.

Next, we often denote a rule such as \( R \) above by \( B \rightarrow H \) or \( B \rightarrow \exists \vec{x}. H \), omitting all or some quantifiers.

A *factbase* \( F \) is an existentially closed (finite) conjunction of atoms. A *Boolean conjunctive query* (BCQ) has the same form as a factbase, and we often identify both notions.

A *knowledge base* (KB) \( K \) is a tuple \( \langle R, F \rangle \) with \( R \) a rule set and \( F \) a factbase. We often identify rule bodies, rule heads, and factbases with (finite) sets of atoms.

Given atom sets \( F \) and \( F' \), a *homomorphism* \( \pi \) from \( F \) to \( F' \) is a function with domain \( \text{Vars}(F) \) such that \( \pi(F) \subseteq F' \); \( \pi \) is an *isomorphism* from \( F \) to \( F' \) if additionally, \( \pi \) is injective and \( \pi^{-1} \) is a homomorphism from \( F' \) to \( F \). A homomorphism \( \pi \) from \( F \) to \( F' \) is a *retraction* if \( \pi \) is the identity over \( \text{Vars}(F) \cap \text{Vars}(F') \) (next, we often use this notion with \( F' \subseteq F \)).
We identify logical interpretations with atom sets. An atom set $F$ satisfies a rule $R = B \rightarrow H$ if, for every homomorphism $\pi$ from $B$ to $F$, there is an extension $\tilde{\pi}$ of $\pi$ with $\tilde{\pi}(H) \subseteq F$; equivalently, $F$ is a model of $R$. An atom set $M$ is a model of a factbase $F$ if there is a homomorphism from $F$ to $M$, and it is a model of a KB $(R, F)$ if it is a model of $F$ and satisfies all rules in $\mathcal{R}$. Given KBs or atom sets $A$ and $B$, $A$ entails $B$, written $A \models B$, if every model of $A$ is a model of $B$; $A$ and $B$ are equivalent if $A \models B$ and $B \models A$. Given atom sets $F$ and $F'$, it is known that $F \models F'$ iff there is a homomorphism from $F'$ to $F$.

**Definition 2.** A model $M$ of a KB $\mathcal{K}$ is universal if there is a homomorphism from $M$ to every model of $\mathcal{K}$.

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

Next, we will consider transformations of rule sets that introduce fresh predicates. To specify the relationships between a rule set and its decomposition, we will rely on the notion of conservative extension:

**Definition 3 (Conservative extension).** Let $\mathcal{R}$ and $\mathcal{R}'$ be two rule sets such that $\text{Preds}(\mathcal{R}) \subseteq \text{Preds}(\mathcal{R}')$. The set $\mathcal{R}'$ is a conservative extension of the set $\mathcal{R}$ if (1) the restriction of any model of $\mathcal{R}'$ to the predicates in $\text{Preds}(\mathcal{R})$ is a model of $\mathcal{R}$, and (2) any model $M$ of $\mathcal{R}$ can be extended to a model $M'$ of $\mathcal{R}'$ that has the same domain (i.e., $\text{Terms}(M) = \text{Terms}(M')$) and agrees with $M$ on the interpretation of the predicates in $\text{Preds}(\mathcal{R})$ (i.e., they have the same atoms with predicates in $\text{Preds}(\mathcal{R})$).

When $\mathcal{R}'$ is a conservative extension of $\mathcal{R}$, for any factbase $F$ the KBs $(\mathcal{R}, F)$ and $(\mathcal{R}', F)$ entail the same (closed) formulas on $\text{Preds}(\mathcal{R})$, in particular BCQs.

**The chase** The chase is a family of procedures that repeatedly apply rules to a factbase until a fixpoint is reached. We formally define such procedures before stating their correctness with respect to factbase entailment in Proposition 7.

**Definition 4 (Triggers and derivations).** Given a fact set $F$, a trigger $t$ on $F$ is a tuple $(R, \pi)$ with $R = B \rightarrow \exists z.H$ a rule and $\pi$ a homomorphism from $B$ to $F$. Let $\text{support}(t) = \pi(B)$ and $\text{output}(t) = \pi^R(H)$, where $\pi^R$ is the extension of $\pi$ that maps every variable $z \in \pi$ to the fresh variable $z_t$ that is unique for $z$ and $t$. A derivation from a KB $\mathcal{K} = (\mathcal{R}, F)$ is a sequence $D = (\emptyset, F_0), (t_1, F_1), \ldots$ such that:

1. Every $F_i$ in $D$ is a factbase; moreover, $F_0 = F$.
2. Every $t_i$ in $D$ is a trigger $(R, \pi)$ on $F_{i-1}$ such that $R \in \mathcal{R}$, output($t_i$) $\not\subseteq F_{i-1}$, and $F_i = F_{i-1} \cup \text{output}(t_i)$.

The result of $D$, written res($D$), is the union of all the factbases in $D$. Let triggers($D$) be the set of all triggers in $D$ and length($D$) = |triggers($D$)| be the length of $D$.

Different chase variants build specific derivations according to different criteria of trigger applicability. Below, the letters $\mathcal{O}$, $\mathcal{SO}$, $\mathcal{R}$, and $\mathcal{E}$ respectively refer to so-called oblivious, semi-oblivious, restricted, and equivalent variants.

**Definition 5 (Applicability).** A trigger $t = (R, \pi)$ on a factbase $F$ is (i) $\mathcal{O}$-applicable on $F$ if output($t$) $\not\subseteq F$, (ii) $\mathcal{SO}$-applicable on $F$ if output($t'$) $\not\subseteq F$ for every trigger $t' = (R, \pi')$ with $\pi(x) = \pi'(x)$ for all $x \in \text{fr}(R)$, (iii) $\mathcal{R}$-applicable on $F$ if there is no retraction from $F$ to output($t$) to $F$, and (iv) $\mathcal{E}$-applicable on $F$ if there is no homomorphism from $F$ to output($t$) to $F$.

**Example 1.** Consider the KB $\mathcal{K} = (\mathcal{R}, F)$ with $\mathcal{R} = \{R = P(x, y) \rightarrow \exists z. P(y, z) \land P(z, y)\}$ and $F = \{P(a, b)\}$ with $a$ and $b$ some constants. The trigger $t_1 = (R, \pi_1)$ with $\pi_1 = \{x \mapsto a, y \mapsto b\}$ is $\mathcal{X}$-applicable on $F_0 = F$ (for any $\mathcal{X}$), and output($t_1$) = $\{P(b, z_1), P(z_1, b)\}$. There are two new triggers on $F_1 = F \cup \text{output}(t_1)$, both $\mathcal{O}$- and $\mathcal{SO}$-applicable, but neither $\mathcal{R}$- nor $\mathcal{E}$-applicable. For instance, consider $t_2 = (R, \pi_2)$ with $\pi_2 = \{x \mapsto b, y \mapsto z_1\}$ and output($t_2$) = $\{P(z_1, z_2), P(z_2, z_1)\}$: there is a retraction from $F_1 \cup \text{output}(t_2)$ to $F_1$, which maps $z_2$ to $b$.

**Definition 6 ((DFX)-X-Chase).** For an X $\in \{\mathcal{O}, \mathcal{SO}, \mathcal{R}, \mathcal{E}\}$, an X-derivation from a KB $\mathcal{K} = (\mathcal{R}, F)$ is a derivation $D$ such that every trigger $t_i \in \text{triggers}(D)$ is X-applicable on $F_i$; $D$ is a DFX-X-derivation if it gives priority to Datalog rules: for any $t_i = (R, \pi) \in \text{triggers}(D)$, if $R$ is a Non-Datalog rule, then $F_{i-1}$ satisfies every Datalog rule in $\mathcal{R}$. A (DFX)$^+$-X-derivation is fair if for every $F_i$ occurring in $D$ and trigger $t$ X-applicable on $F_i$, there is some $j > i$ such that $t$ is not X-applicable on $F_j$. A (DFX)$^+$X-derivation is terminating if it is fair and finite.

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

**Proposition 7.** Consider a BCQ $Q$, a KB $\mathcal{K}$, and some fair X-derivation $D$ from $\mathcal{K}$ where $X \in \{\mathcal{Y}, DFX^+, DFX^{-}\}$ and $Y \in \{\mathcal{O}, \mathcal{SO}, \mathcal{R}, \mathcal{E}\}$. Then, $\mathcal{K} \models Q$ iff res($D$) $\models Q$.

**Decidable Classes of Rule Sets** We now define classes of rule sets that ensure the decidability of BCQ entailment, based either on chase termination or on query rewrritability:

**Definition 8 (Chase-Terminating Sets).** For a $\mathcal{Y} \in \{\mathcal{O}, \mathcal{SO}, \mathcal{R}, \mathcal{E}\}$ and an $X \in \{\mathcal{Y}, DFX^+, DFX^{-}\}$, let $CT_\mathcal{X}$ (resp. $CT_\mathcal{X}^+$) be the set of all rule sets $\mathcal{R}$ such that every (resp. some) fair X-derivation from every KB $(\mathcal{R}, F)$ is finite.

When $\mathcal{R} \in CT_\mathcal{X}^+$ (resp. $CT_\mathcal{X}^+$), $\mathcal{R}$ ensures the termination (resp. sometimes termination) of the X-chase.

**Example 2.** Consider the KB $\mathcal{K} = \{R, F\}$ from Example 1. All fair $\mathcal{O}$- or $\mathcal{SO}$-derivations from $\mathcal{K}$ are infinite. The only one fair $\mathcal{R}$-derivation (resp. $\mathcal{E}$-derivation) from $\mathcal{K}$ is $D = (\emptyset, F_0), (t_1, F_1)$. Any fair $\mathcal{R}$-derivation from a KB with $\mathcal{R}$ is finite and hence, $\mathcal{R} \in CT_\mathcal{E}$ (and $\mathcal{R} \in CT_\mathcal{E}^+$).

\footnote{The equivalent chase behaves as the better-known core chase regarding termination: it halts exactly when the KB has a finite universal model. The difference lies in the fact that the core chase computes a minimal universal model (i.e., a core). The equivalent chase has the advantage of being monotonic ($\forall_i F_i \subseteq F_{i+1}$).}
Definition 9 (FO-rewritability). A rule set \( R \) is \( \text{FO-rewritable} \) if for any BCQ \( Q \), there is a (finite) BCQ set \( \{Q_1, \ldots, Q_n\} \) such that, for every factbase \( F \), \( \langle R, F \rangle \models Q \) iff \( F \models Q_i \) for some \( 1 \leq i \leq n \).

In our proofs, we rely on a property equivalent to FO-rewritability: the bounded derivation depth property, which has the advantage of being based on (a breadth-first version of) the chase (Calì, Gottlob, and Lukasiewicz 2009). See (Gottlob et al. 2014) about the equivalence between both properties.

Definition 10 (BDDP). For a rule and a factbase \( F \), let \( R(F) \supseteq F \) be the minimal factbase that includes \( \text{output}(t) \) for every trigger \( t \) with \( R \). For a rule set \( R \), let \( R(F) = \bigcup_{R \subseteq R} R(F) \). For a KB \( K = \langle R, F \rangle \), let \( Ch_0(K) = F \) and \( Ch_i(K) = R(Ch_{i-1}(K)) \) for every \( i \geq 1 \).

A rule set \( R \) has the bounded derivation depth property (BDDP) if, for any BCQ \( Q \), there is some \( k \geq 0 \) such that, for every factbase \( F, \langle R, F \rangle \models Q \) iff \( Ch_k(\langle R, F \rangle) \models Q \).

Normalisation Procedures. Finally, we formally define normalisation procedures and their impact on the above properties. A normalisation procedure is a function that maps rule sets to rule sets (complying with a certain shape) such that for any rule set \( R \), \( \langle R, F \rangle \models Q \) iff \( f(\langle R, F \rangle) \models Q \) for any factbase \( F \) and BCQ \( Q \) on \( \text{Preds}(R) \).

Definition 11. Consider some \( X \in \{\emptyset, SO, R, DF-R, E\} \). Then, a normalisation procedure \( f \):

- Preserves termination of the \( X \)-chase if \( f(CT^{X}_{\emptyset}) \subseteq CT^{X}_{\emptyset} \); it preserves sometimes-termination of the \( X \)-chase if \( f(CT^{X}_{\emptyset}) \subseteq CT^{X}_{\emptyset} \).
- Preserves non-termination of the \( X \)-chase if \( f(CT^{X}_{\emptyset}) \subseteq CT^{X}_{\emptyset} \). Otherwise, \( f \) may gain termination.
- Preserves rewritability if it maps FO-rewritable rule sets to FO-rewritable rule sets.

3 Generality of Chase-Terminating Rule Sets

One of our goals is to study normalisation procedures that preserve membership over the sets of chase-terminating rule sets from Definition 8. To be systematic, we quantify the equality and strict-subset relations between these sets in Theorems 12 and 13, respectively. Grahne and Onet already proved most of the claims in these theorems (see Theorem 4.5, Propositions 4.6 and 4.7, and Corollary 4.8 in (Grahne and Onet 2018)); we reprove some of them again.

Theorem 12. For every \( X \in \{\emptyset, SO, E\} \), we have that

\[
CT^X_{\emptyset} \subseteq CT^X_{\emptyset} \subseteq CT^{DF-R}_{\emptyset} = CT^{DF-R}_{\emptyset}
\]

Sketch. To show that the theorem holds if \( X = \emptyset \) (resp. \( X = SO \)), it suffices to prove that all fair \( X \)-derivations from an input KB \( K \) produce the same result (resp. same result up to isomorphism); see forthcoming Lemma 16.

All fair \( E \)-derivations from an input KB \( K \) are finite iff \( K \) admits a finite universal model (Rocher 2016). Hence, the theorem holds if \( X = E \).

The equalities in Theorem 12 simplify our work: for instance, if a function preserves termination of the oblivious chase, then we know that it also preserves sometimes-termination of this variant. Alas, the remaining sets of chase-terminating rule sets are not equal:

Theorem 13. The following hold:

\[
CT^X_{\emptyset} \subseteq CT^{SO}_{\emptyset} \subseteq CT^X_{\emptyset} \subseteq CT^{DF-R}_{\emptyset} = CT^{DF-R}_{\emptyset} \subseteq CT^{DF-R}_{\emptyset} \subseteq CT^{E}_{\emptyset} \subseteq CT^{E}_{\emptyset}
\]

Sketch. The subset inclusions follow by definition; we present some rule sets to show that these are strict:

\[
\{P(x, y) \rightarrow \exists z.P(x, z)\} \in CT^{SO}_{\emptyset} \setminus CT^{X}_{\emptyset}
\]

\[
\{P(x, y) \rightarrow \exists z.P(x, z) \land P(y, z)\} \in CT^{SO}_{\emptyset} \setminus CT^{X}_{\emptyset}
\]

\[
\{P(x, y) \rightarrow \exists z.P(x, z), P(x, y) \rightarrow P(y, x)\} \in CT^{DF-R}_{\emptyset} \setminus CT^{SO}_{\emptyset}
\]

\[
\{P(x, y) \rightarrow \exists z.P(x, z) \land P(y, z), P(x, y) \rightarrow \exists z.P(x, z), P(x, y) \rightarrow P(y, x)\} \in CT^{DF-R}_{\emptyset} \setminus CT^{SO}_{\emptyset}
\]

Moreover, the rule set \( R = \{(2-6)\} \) is in \( CT^{DF-R}_{\emptyset} \setminus CT^{SO}_{\emptyset} \):

\[
A(x) \rightarrow R(x, x)
\]

\[
R(x, y) \land S(y, z) \rightarrow S(x, z)
\]

\[
A(x) \land S(x, y) \rightarrow A(y)
\]

\[
A(x) \rightarrow \exists z.R(x, z)
\]

\[
R(x, y) \rightarrow \exists z.S(y, z)
\]

To show that \( R \notin CT^{DF-R}_{\emptyset} \), we prove that the KB \( K = \langle R, \{A(a)\} \rangle \) does not admit terminating \( DF-R \)-derivations. Specifically, all fair \( DF-R \)-derivations from \( K \) yield the same result, which is depicted in Figure 1. Rule (2) is applied first, then the following pattern is repeated: apply rule (6) followed by Datalog rules (3), (4) and (2). Rule (5) is never applicable since priority is given to rule (2). To show that \( R \in CT^{R}_{\emptyset} \) we verify that every KB of the form \( \langle R, F \rangle \) admits a terminating \( R \)-derivation. We can produce such a derivation by exhaustively applying the rules in \( R \) in the following order: first, apply rules (4), (5) and (6); then, apply (3); finally, apply (2).

Our main achievement is showing that \( CT^{R}_{\emptyset} \setminus CT^{DF-R}_{\emptyset} \) is non-empty; thus proving that Datalog-first strategies are not necessarily the most terminating for the restricted chase.

To understand why this strategy results in a terminating \( R \)-derivation, we suggest to first try it on \( F = \{A(a)\} \), which yields \( \{A(a), R(a, z_1), S(z_1, z_2), S(a, a), R(a, a)\} \).

R, S
A
S
a : A
R, S
A
S
R, S
A
R, S
A
Figure 1: The only result of the \( DF-R \)-chase from the KB \( K = \langle R, \{A(a)\} \rangle \) introduced in the proof of Theorem 13.
4 Single-Piece Decomposition

The single-piece decomposition (piece-decomposition in short) is a procedure that splits a rule \( R = B \rightarrow \exists z.H \) into several rules \( R_1, \ldots, R_n \) that have the same body as \( R \), and whose head is a subset of \( H \) that (directly or indirectly) shares some existential variable in \( H \).

**Definition 14.** The piece graph of a rule \( R = B \rightarrow \exists z.H \) is the graph whose vertices are the atoms in \( H \), and whose head is a subset of \( H \) that (directly or indirectly) shares some existential variable in \( H \).

The piece-decomposition of a rule \( R = B \rightarrow \exists z.H \) is the rule set whose vertices are the atoms in \( H \), and whose head is a subset of \( H \) that (directly or indirectly) shares some existential variable in \( H \).

**Example 3.** Consider the rule (7) and its single-piece decomposition sp((7)) = \{\{8–10\}\}:

\[
\begin{align*}
R(x, y) &\rightarrow \exists z, u.P(x, z) \land A(z) \land A(u) \land P(x, y) \quad (7) \\
R(x, y) &\rightarrow \exists z.P(x, z) \land A(z) \quad (8) \\
R(x, y) &\rightarrow \exists u.A(u) \quad (9) \\
R(x, y) &\rightarrow P(x, y) \quad (10)
\end{align*}
\]

Piec-decomposition is indeed a normalisation procedure, since it preserves logical equivalence:

**Proposition 15.** A rule set \( R \) is equivalent to the set sp(\( R \)).

The following lemma is later applied to show that the piece-decomposition preserves termination of the oblivious and semi-oblivious chase in Theorem 18:

**Lemma 16.** Consider some fair \( X \)-derivations \( D \) and \( D' \) from a KB \( K \). If \( X = \emptyset \), then res(\( D \)) = res(\( D' \)). If \( X = \{\emptyset, \mathcal{O}\} \), then res(\( D \)) is isomorphic to res(\( D' \)).

**Definition 17.** Given some \( X \in \{\emptyset, \mathcal{O}, \mathcal{S}\} \) and a KB \( K \), let \( \text{Ch}_X(K) \) be some (arbitrarily chosen) atom set that is isomorphic to the result of all fair \( X \)-derivations from \( K \).

**Theorem 18.** The piece-decomposition preserves the termination of the \( \mathcal{O} \)-chase and \( \mathcal{S} \)-chase.

**Sketch.** Consider some \( X \in \{\emptyset, \mathcal{O}, \mathcal{S}\} \) and some \( X \)-

The piece-decomposition does not preserve the termination of any restricted chase variant. The reason is that it allows for intertwining the application of split rules that come from different original rules, resulting in new application strategies that may lead to non-termination.

**Theorem 19.** The piece-decomposition does not preserve termination of the \( R \)- or the \( \mathcal{D} \mathcal{F} \)-\( R \)-chase.

**Sketch.** Consider the rule set \( R = \{\{(11,12)\}\} \) and its piece-decomposition sp(\( R \)) = \{\{12–14\}\}:

\[
\begin{align*}
P(x, y) &\rightarrow P(y, y) \land A(y) \quad (11) \\
P(x, y) &\rightarrow P(y, y) \quad (13) \\
A(x) &\rightarrow \exists z.P(x, z) \quad (12) \\
P(x, y) &\rightarrow A(y) \quad (14)
\end{align*}
\]

The set \( R \) is in \( CT^R_{\mathcal{O}} \), because triggers with (12) are not \( R \)-applicable to the output of triggers with (11). The set sp(\( R \)) is not in \( CT^R_{\mathcal{O}} \) because the KB \( \langle \text{sp}(R), \{A(a)\} \rangle \) admits the following non-terminating \( R \)-derivation (\( \emptyset, F_0, (t_1, F_1), \ldots:\n\]

\[
\begin{align*}
F_0 &= \{A(a)\}, \quad F_4 = \{P(z_1, z_1)\} \cup F_3, \\
F_1 &= \{P(a, z_1)\} \cup F_0, \quad F_5 = \{A(z_2)\} \cup F_4, \\
F_2 &= \{A(z_1)\} \cup F_1, \quad F_6 = \{P(z_2, z_3)\} \cup F_5, \\
F_3 &= \{P(z_1, z_2)\} \cup F_2, \quad \ldots
\end{align*}
\]

This derivation is built by first applying rule (12) (leading to \( F_1 \)), then indefinitely repeating the sequence of rule applications (14), (12), and (13). In contrast, the only fair \( R \)-derivation with \( R \) would apply (12) then (11), leading to \( \langle A(a), P(a, z_1), P(z_1, z_1), A(z_1) \rangle \).

To get a similar behavior with the \( \mathcal{D} \mathcal{F} \)-\( R \)-chase, we introduce "dummy" existential variables in rules (11) and (12), so that their piece-decomposition has no Datalog rules:

\[
\begin{align*}
P(x, y, v) &\rightarrow \exists u, w.P(y, y, u) \land A(y, w) \quad (15) \\
A(x, v) &\rightarrow \exists z, u.P(x, z, u) \quad (16)
\end{align*}
\]

Applying analogous arguments we can show that \( R' = \{(15), (16)\} \) is in \( CT^R_{\mathcal{D} \mathcal{F}} \) and that \( \text{sp}(R') \) is not.

Initially, we believed that the piece-decomposition would preserve sometimes-termination of the \( R \)-chase. Our intuition was that, given a terminating \( R \)-derivation from a KB \( K = \langle R, F \rangle \), we could replicate this derivation from \( \langle \text{sp}(R), F \rangle \) by applying the split rules in \( \text{sp}(R) \) piece by piece. Surprisingly, this is not always possible:

**Theorem 20.** The piece-decomposition does not preserve the sometimes-termination of the \( R \)-chase.

**Sketch.** The following set \( R = \{\{(17–22)\}\} \) is in \( CT^R_{\mathcal{O}} \) and its piece-decomposition sp(\( R \)) = \{\{18–24\}\} is not. The set \( R \) is adapted from \{\{2–6\}\} (proof of Th. 13). Note that (17) is split into two equivalent rules (23) and (24). To show that sp(\( R \)) \( \not\in \) \( CT^R_{\mathcal{O}} \), we start again from \( \{A(a)\} \). Again, some \( R \)-atom is created and leads to apply other rules. With \( R \), applying (17) then (18) creates an atom of form \( R(y_1, z_1) \), while with sp(\( R \)), applying (23) then (18) creates an atom \( R(y_1, y_1) \). This loop leads to non-termination.

\[
\begin{align*}
A(x) &\rightarrow \exists y, z.U(x, y) \land H(y, z) \land U(x, z) \land H(z, x) \quad (17) \\
U(x, y) &\land U(x, z) \rightarrow R(y, z) \quad (18) \\
R(x, z) &\land R(y, z) \rightarrow \exists v.R(z, v) \quad (19) \\
R(x, y) &\land R(y, z) \rightarrow \exists u.S(z, v) \quad (20) \\
R(x, y) &\land S(z, y) \rightarrow S(x, z) \quad (21) \\
A(x) &\land U(x, y) \land S(y, z) \rightarrow \exists v.H(z, v) \land A(v) \quad (22) \\
A(x) &\rightarrow \exists y.U(x, y) \land H(y, x) \quad (23) \\
A(x) &\rightarrow \exists z.U(x, z) \land H(z, x) \quad (24)
\end{align*}
\]

**Theorem 21.** The piece-decomposition does not preserve the sometimes-termination of the \( \mathcal{D} \mathcal{F} \)-\( R \)-chase.
Sketch. The set $\mathcal{R} = \{(25–29)\}$ is in $CT_{\mathcal{V}_3}^{S_0}$, while $sp(\mathcal{R}) = \{(26–31)\}$ is not. Note that the only difference with $\{(2–6)\}$ is the atom $H(x, y)$ in the first rule, making it non-Datalog, which prevents its early application.

\[
A(x) \rightarrow \exists y. R(x, x) \land H(x, y) \quad (25)
\]
\[
R(x, y) \land S(y, z) \rightarrow S(x, y) \quad (26)
\]
\[
A(x) \land S(x, y) \rightarrow A(y) \quad (27)
\]
\[
A(x) \rightarrow \exists y. R(x, y) \quad (28)
\]
\[
A(x) \rightarrow R(x, x) \quad (30)
\]
\[
R(x, y) \rightarrow \exists z. S(y, z) \quad (29)
\]
\[
A(x) \rightarrow \exists y. H(x, y) \quad (31)
\]

Since piece-decomposition preserves logical equivalence (Proposition 15), one directly obtains that it preserves termination of the equivalent chase. Indeed, $(\mathcal{R}, F)$ admits a finite universal model iff $(sp(\mathcal{R}), F)$ admits one:

**Theorem 22.** The piece-decomposition preserves the termination of the $E$-chase.

The piece-decomposition may gain termination:

**Theorem 23.** The piece-decomposition may gain termination (and sometimes-termination) of the $\mathcal{SO}$-, the $\mathcal{R}$-, and the $DF\mathcal{R}$-chase but not of the $\mathcal{O}$- and $\mathcal{E}$-chase.

Sketch. Consider the set $\mathcal{R} = \{ P(x, y) \rightarrow \exists z. P(x, z) \land R(x, y) \}$, which is not in $CT_{\mathcal{V}_3}^{SO}$, $CT_{\mathcal{V}_3}^{SO}$, $CT_{\mathcal{V}_3}^{DR}$, or $CT_{\mathcal{V}_3}^{DFR}$. However, sp($\mathcal{R}$) is in all of these sets.

Concerning the $O$-chase, we show via induction that $Ch_0( (\mathcal{R}, F) )$ and $Ch_0( (sp(\mathcal{R}), F) )$ are isomorphic for any rule set $\mathcal{R}$ and factbase $F$ (where $Ch_0(\cdot)$ is the function from Definition 17, which maps a KB to its only $O$-chase result). Hence, all $O$-derivations from $(\mathcal{R}, F)$ are terminating iff all $O$-derivations from $(sp(\mathcal{R}), F)$ are terminating. Concerning the $E$-chase, we rely again on Proposition 15.

Example 4. Consider the rule $(32)$ and its one-way atomic decomposition $1ad((32)) = \{(33–35)\}$:

\[
R(x, y) \rightarrow \exists z. P(x, z) \land S(x, y, z) \quad (32)
\]
\[
R(x, y) \rightarrow \exists z. X_{(32)}(x, y, z) \quad (33)
\]
\[
X_{(32)}(x, y, z) \rightarrow P(x, z) \quad (34)
\]
\[
X_{(32)}(x, y, z) \rightarrow S(x, y, z) \quad (35)
\]

Note that piece-decomposition would not decompose rule $(32)$, i.e., sp($(32)$) = $\{(32)\}$

Strictly speaking, $\mathcal{R}$ and $1ad(\mathcal{R})$ cannot be logically equivalent because they are built on different sets of predicates; however, it is straightforward to check that $1ad(\mathcal{R})$ is a conservative extension of $\mathcal{R}$. Therefore, one-way atomic decomposition is indeed a normalisation procedure.

The following is a corollary of forthcoming Theorem 34:

**Theorem 26.** The one-way atomic decomposition preserves termination of the $O$- and $\mathcal{SO}$-chase.

An interesting phenomenon occurs with the one-way atomic decomposition: the notions of $SO$-applicability on $\mathcal{R}$ and $R$-applicability on $1ad(\mathcal{R})$ coincide:

**Lemma 27.** Consider a KB $(\mathcal{R}, F)$ and a finite derivation $D = (F_0, F_1, \ldots, F_n)$ from $1ad(\mathcal{R}), F_0$. Then, a trigger $t$ with a rule in $1ad(\mathcal{R})$ is $R$-applicable on $F_0$ iff it is $SO$-applicable on $F_n$.

**Proof.** $(\Rightarrow)$: From Definition 5. $(\Leftarrow)$: Let $t = (R, \pi)$ with $R \in 1ad(\mathcal{R})$. If $R$ is Datalog, the notions of $SO$- and $R$-applicability coincide for every factbase. Otherwise, $R$ is of the form $B[\vec{x}, \vec{y}] \rightarrow \exists \vec{z}. X_{R'}(\vec{x}, \vec{z})$ where $X_{R'} \notin \exists \vec{x} \in \mathcal{R}$. Then, for every trigger $t' = (R', \pi')$ with $\pi(\vec{x}) = \pi'(\vec{x})$, it holds that output$(t') \subseteq F_n$. By Definition 25, $R$ is the only rule in $1ad(\mathcal{R})$ with $X_{R'}$ in its head, hence $\pi''(X_{R'}(\vec{x}, \vec{z})) \notin F_n$ for every extension $\pi''$ of $\pi$. Hence, $t$ is $R$-applicable on $F_n$.

Intuitively, Lemma 27 implies that, after applying the one-way atomic decomposition, $R$-applicability becomes as loose and restrictive as $SO$-applicability. Therefore:

**Theorem 28.** The one-way atomic decomposition does not preserve termination nor sometimes-termination of the $R$- or the $DF\mathcal{R}$-chase.

**Proof.** By Theorem 13, there is a rule set $\mathcal{R} \notin CT_{\mathcal{V}_3}^{SO}$ that is in $CT_{\mathcal{V}_3}^{DR}$, $CT_{\mathcal{V}_3}^{DFR}$, and $CT_{\mathcal{V}_3}^{DFR}$. By Theorem 26, $1ad(\mathcal{R})$ is not in $CT_{\mathcal{V}_3}^{SO}$; that is, there is some KB $\mathcal{K}$ of the form $(1ad(\mathcal{R}), F)$ that does not admit any terminating $SO$-derivation. By Lemma 27, every terminating $R$-derivation from $\mathcal{K}$ is also a terminating $SO$-derivation from $\mathcal{K}$. Therefore, $\mathcal{K}$ does not admit any terminating $R$-derivation, hence $1ad(\mathcal{R})$ is not in $CT_{\mathcal{V}_3}^{DR}$, $CT_{\mathcal{V}_3}^{DFR}$, or $CT_{\mathcal{V}_3}^{DR}$.

**Theorem 29.** The one-way atomic decomposition does not preserve the termination of the $E$-chase.

---

Such rule set in given in the proof of Theorem 13; see also $\{(36)\}$ in the proof of Theorem 29.
Proof. Consider the rule set \( R = \{36\} \) (see also Example 1) and its decomposition \( lad(R) = \{37–39\} \):

\[
P(x, y) \Rightarrow \exists z. P(y, z) \land P(z, y) \quad (36)
\]

\[
P(x, y) \Rightarrow \exists z. X_{36}(y, z) \quad (37)
\]

\[
X_{36}(y, z) \Rightarrow P(y, z) \quad (38)
\]

\[
X_{36}(y, z) \Rightarrow P(y, z) \quad (39)
\]

The rule set \( R \) is in \( CT^R_{\forall} \) since every \( E \)-derivation from a KB \((R, F)\) yields a finite result, which is a subset of \( F \cup \{P(a, z_0), P(z_0, a) \mid a \in \text{Terms}(F), z_0 \notin \text{Terms}(F)\} \).

The rule set \( lad(R) \) is not in \( CT^R_{\forall} \) since the KB \( K = \langle lad(R), \{P(a, b)\} \rangle \) has no terminating \( E \)-derivation. In fact, all fair \( E \)-derivations from \( K \) yield the same result:

\[
\{P(a, b), X_{36}(b, z_1), P(b, z_1), P(z_1, b)\} \cup \\
\{X_{36}(z_i, z_{i+1}), P(z_i, z_{i+1}), P(z_{i+1}, z_i) \mid i \geq 1\}
\]

\( \square \)

Again, to show that the one-way decomposition preserves FO-rewritability, we show that it preserves BDDP.

Theorem 30. A rule set \( R \) is BDDP iff \( lad(R) \) is BDDP.

Sketch. (\( \Rightarrow \)) For factbases \( F \) restricted to the original vocabulary \( \Sigma \), we prove that \( Ch_1(F, R) = Ch_{2ad}(F, lad(R)) \mid \Sigma \).

Dealing with arbitrary factbases is tackled by a weakening of this correspondence. (\( \Leftarrow \)) If \( R \) is not BDDP, there are \( Q \) and \( \{F_i\}_{i \in \mathbb{N}} \) such that for all \( i \), \( F_i, R \models Q \) and \( Ch_1(F_i, R) \models \neg Q \). Since for all \( F_i \) on \( \Sigma \), \( Ch_{2ad}(F_i, R) \models \neg Q \), it holds that \( Ch_{2ad}(F_i, lad(R)) \) \models \neg Q \), hence \( lad(R) \) is not BDDP.

\( \square \)

6 Two-Way Atomic Decomposition

Despite the fact that it produces a conservative extension of the original rule set, the one-way atomic decomposition does not preserve the existence of a finite universal model; hence, it does not preserve equivalent chase termination.

Example 5. As in the proof of Theorem 29, consider \( R = \{36\} \), its decomposition \( lad(R) = \{37–39\} \), and the factbase \( F = \{P(a, b)\} \). Then, \( U = \{P(a, b), P(b, z_1), P(z_1, b)\} \) is a finite universal model for \((R, F)\) that cannot be extended (keeping the same domain) into a universal model of \( (lad(R), F) \). Indeed, the set

\[
\{P(a, b), P(b, z_1), P(z_1, b), X_{36}(b, z_1), X_{36}(z_1, b)\}
\]

is the smallest extension of \( U \) that is a model for \( (lad(R), F) \), but it is not universal.

Hence, we define a notion similar to that of conservative extension, but whose purpose is to guarantee the preservation of the equivalent chase termination.

Definition 31. Let \( R \) and \( R' \) be two rule sets such that \( \text{Preds}(R) \subseteq \text{Preds}(R') \). The set \( R' \) is a universal-conservative extension of the set \( R \) if, for any factbase \( F \) with \( \text{Preds}(F) \subseteq \text{Preds}(R) \),

1. The restriction of any universal model of \( (R', F) \) to the predicates in \( \text{Preds}(R) \) is a universal model of \( (R, F) \).
2. Any universal model \( M \) of \( (R, F) \) can be extended to a universal model of \( (R', F) \) that has the same domain and agrees with \( M \) on the interpretation of \( \text{Preds}(R) \).

We now introduce a normalisation procedure that produces universal-conservative extensions:

Definition 32. The two-way atomic decomposition of a rule \( R = B[x, y] \Rightarrow \exists z. H[x, z] \) is the rule set \( 2ad(R) = lad(R) \cup \{H[x, z] \Rightarrow X_R(x, z)\} \), with \( X_R \) the fresh predicate in \( lad(R) \). For a rule set \( R \), we let \( 2ad(R) = \bigcup_{R' \subseteq R} 2ad(R') \).

Example 6. Consider again the rule \( (32) \) from Example 4. Then its two-way atomic decomposition is \( 2ad((32)) = Iad((32)) \cup \{40\} \):

\[
P(x, z) \land S(x, y, z) \Rightarrow X_{32}(x, y, z)
\]

We establish that this new decomposition is indeed a normalisation procedure that has the desired property.

Proposition 33. The rule set \( 2ad(R) \) is a conservative extension and a universal-conservative extension of \( R \).

We can now focus our interest again on chase termination. Both atomic decompositions behave like the single-piece decomposition (Theorem 18) regarding the oblivious and the semi-oblivious chase:

Theorem 34. Both atomic decompositions preserve the termination of the \( O \)-chase and the \( SO \)-chase.

Sketch. Consider \( X \in \{O, SO\} \) and \( K = \langle R, F \rangle \) a KB. First note that for these \( X \)-chases, applying a rule cannot prevent the application of another rule. Hence, since \( lad(R) \subseteq 2ad(R) \), it is sufficient to prove the result for \( 2ad(R) \). The proof is similar to that of Theorem 18: we show by induction on an arbitrary derivation \( D \) from \( K \) that there is an injective homomorphism from \( Ch_X(2ad(R), F) \) restricted to the predicates in \( \text{Preds}(R) \) to \( Ch_X(K) \), which leads to a similar conclusion.

\( \square \)

The behavior of the restricted chase is again less easily characterized, as we will see in the next results.

Theorem 35. The two-way atomic decomposition preserves sometimes-termination of the \( R \)-chase; it may also gain termination of this chase variant.

Sketch. To prove preservation, consider a KB \( K = \langle R, F \rangle \) such that \( R \in CT^R_{\forall} \). Then, there is a terminating \( R \)-derivation \( D \) from \( K \). We can then show by induction that if a trigger \( t = \langle R, \pi \rangle \) with \( R = B \rightarrow H \) is applied at some step, the trigger \( t' = \langle B \rightarrow X_R, \pi \rangle \) is applicable at the same step, then the triggers \( t_i = \langle X_R \rightarrow H_i, \pi_R \rangle \) are also, and that applying \( t' \) and all the \( t_i \) successively yields the same result (when restricted to the predicates in \( R \)) as applying \( t \). This shows that we can replicate a terminating derivation, and thus that the sometimes-termination is preserved.

We now present an example where we gain termination. Consider the rule set \( R = \{41–45\} \):

\[
A(x) \Rightarrow \exists y, z. R(x, x, x) \land R(x, y, z)
\]

\[
R(x, y, z) \Rightarrow \exists t. R(x, x, t)
\]
\begin{align*}
R(x, x, y) \rightarrow \exists z.S(x, y, z) & \quad (43) \\
R(x, x, y) \land S(x, y, z) \rightarrow S(x, x, x) & \quad (44) \\
A(x) \land S(x, x, y) \rightarrow A(y) & \quad (45)
\end{align*}

There is no terminating \( \mathbb{R} \)-derivation on the KB \( \langle \mathcal{R}, \{A(a)\} \rangle \) but there is one on \( 2ad(\mathcal{R}), F \) for any \( F \).

**Theorem 36.** The two-way atomic decomposition does not preserve the termination of the \( \mathbb{R} \)-chase.

**Proof.** Consider the rule set \( \mathcal{R} = \{(36)\} \) introduced in the proof of Theorem 29 and \( 2ad(\mathcal{R}) = \{(37)–39, (46)\} \):

\begin{align*}
P(x, y) \rightarrow \exists z. P(y, z) \land P(z, y) & \quad (36) \\
P(x, y) \rightarrow \exists z. X_{36}(y, z) & \quad (37) \\
X_{36}(y, z) \rightarrow P(y, z) & \quad (38) \\
X_{36}(y, z) \rightarrow P(z, y) & \quad (39) \\
P(y, z) \land P(z, y) \rightarrow X_{36}(y, z) & \quad (46)
\end{align*}

The \( \mathbb{R} \)-chase yields the same result as the \( \mathbb{E} \)-chase on \( \mathcal{R} \), so \( \mathcal{R} \in C_{\mathbb{T}_{\mathbb{R}}}^{\mathbb{E}} \mathcal{R} \). We then construct an infinite derivation from \( 2ad(P(a, b)) \). First, apply (37), and (38). Then, repeat the following pattern: (37), (38) (on the new variable), then (39) and (46) (on the variables of the previous loop). Applying (37) again before applying (46) yields an infinite chain.

Again, the \( \mathbb{R} \)-chase is not well-behaved with respect to atomic decomposition. However, the \( \mathbb{D} \mathbb{F} \mathbb{R} \)-chase behaves exactly as desired regarding the two-way atomic decomposition. In fact, we can show an even stronger result: any \( \mathbb{D} \mathbb{F} \mathbb{R} \)-derivation from a KB \( \langle \mathcal{R}, F \rangle \) can be replicated by a \( \mathbb{D} \mathbb{F} \mathbb{R} \)-derivation from \( 2ad(\mathcal{R}), F \), and conversely.

**Theorem 37.** The 2-way atomic decomposition has no impact on the (sometimes-)termination of the \( \mathbb{D} \mathbb{F} \mathbb{R} \)-chase; i.e., \( C_{\mathbb{T}_{\mathbb{D} \mathbb{F} \mathbb{R}}}^{\mathbb{E}} = 2ad(C_{\mathbb{T}_{\mathbb{D} \mathbb{F} \mathbb{R}}}^{\mathbb{E}}) \) and \( C_{\mathbb{T}_{\mathbb{D} \mathbb{F} \mathbb{R}}}^{\mathbb{F} \mathbb{R}} = 2ad(C_{\mathbb{T}_{\mathbb{D} \mathbb{F} \mathbb{R}}}^{\mathbb{F} \mathbb{R}}) \).

**Sketch.** One can prove that any fair \( \mathbb{D} \mathbb{F} \mathbb{R} \)-derivation \( D \) from \( \langle \mathcal{R}, F \rangle \) with \( \mathcal{R} \in C_{\mathbb{T}_{\mathbb{D} \mathbb{F} \mathbb{R}}}^{\mathbb{E}} \) can be replicated to yield a fair \( \mathbb{D} \mathbb{F} \mathbb{R} \)-derivation \( D' \) from \( 2ad(\mathcal{R}), F \) such that \( D' \) is finite if and only if \( D \) is. The reciprocal is also true.

The following result follows from Proposition 33

**Theorem 38.** The two-way atomic decomposition preserves the termination of the \( \mathbb{E} \)-chase.

The single-piece decomposition may gain termination for some chase variants (Theorem 23); we are interested to know if the same can happen with atomic decompositions. Unfortunately, there is no way for a non-terminating rule set to gain termination, as stated next:

**Proposition 39.** If a chase variant does not terminate on a rule set \( \mathcal{R} \), it does not terminate on \( 1ad(\mathcal{R}) \) and \( 2ad(\mathcal{R}) \).

**Sketch.** For each trigger with a rule \( R \) in the original infinite fair derivation, one can consider the corresponding triggers with \( 1ad(\mathcal{R}) \) or \( 2ad(\mathcal{R}) \), and thus produce an infinite fair derivation.

Regarding FO-rewritability, the two-way atomic decomposition behaves similarly to the one-way atomic decomposition (which can be proven similarly, see Theorem 30).

**Theorem 40.** A rule set \( \mathcal{R} \) is \( \text{BDDP} \) if \( 2ad(\mathcal{R}) \) is \( \text{BDDP} \).

7 No Normalisation for the Restricted Chase

Normalisation procedures studied so far do not maintain the status of the termination of the \( \mathbb{R} \)-chase. This raises the question of the existence of such a procedure. We show here that no computable function can map rule sets to sets of rules having atomic head while preserving termination and non-termination of the \( \mathbb{R} \)-chase. To do that, we show that with atomic-head rules, the class of rule sets \( C_{\mathbb{T}_{\mathbb{R}}}^{\mathbb{F} \mathbb{R}} \) for which every fair \( \mathbb{R} \)-derivation from \( \langle \mathcal{R}, F \rangle \) is finite is a recursively enumerable set. With arbitrary rules, we show it is hard for \( \Pi^0_2 \), the second level of the arithmetic hierarchy (Rogers 1987). A complete problem for \( \Pi^0_2 \) is to decide whether a given Turing machine halts on every input word; it remains complete when inputs are restricted to words on a unary alphabet.

**Proposition 41.** For any factbase \( F \), the subset of \( C_{\mathbb{T}_{\mathbb{F} \mathbb{R}}}^{\mathbb{F} \mathbb{R}} \) composed of sets of atomic-head rules is recognizable.

**Sketch.** With atomic-head rules, it is known that the existence of an infinite fair restricted derivation is equivalent to the existence of an infinite restricted derivation (Gogacz, Marczinkowski, and Pieris 2020). Using König’s lemma, one can show that the chase terminates iff there exists a \( k \) such that any fair \( \mathbb{R} \)-derivation is of length at most \( k \).

**Proposition 42.** There exists a factbase \( F \) such that \( C_{\mathbb{T}_{\mathbb{F} \mathbb{R}}}^{\mathbb{F} \mathbb{R}} \) is \( \Pi^0_2 \)-hard.

**Sketch.** Given a Turing machine (TM) \( M \) whose input alphabet is unary, we build a KB \( \mathcal{K} = \langle \mathcal{R}_w \cup \mathcal{R}_M, F \rangle \) s.t. every fair \( \mathbb{R} \)-derivation from \( \mathcal{K} \) is finite iff \( M \) halts on every input. Regardless the chase variant, simulating a TM with a rule set such that the chase terminates whenever the TM halts is classical; we reuse the rule set \( \mathcal{R}_M \) provided in (Bourgaux et al. 2021), which we recall in Figure 4 for self-containment. We show that we can assume wlog that all the rules of \( \mathcal{R}_M \) are applied after all the rules of \( \mathcal{R}_w \) (listed in Figure 3). The set \( \mathcal{R}_w \) is used to generate from \( F \) arbitrarily large input tape representations in a terminating way. To ensure that any fair \( \mathbb{R} \)-derivation from \( \langle \mathcal{R}_w, F \rangle \) terminates, we reuse the emergency brake technique from (Krötzsch, Marx, and Rudolph 2019), which allows one to stop the derivation at any desired length. The representation of an input word of length \( j \) is a set of atoms of the shape \( \{\text{Nxt}(c^j_1, c^j_{i+1}), S_1(c^j_i) \mid 0 \leq i < j\} \cup \{S_i(c^j_i), \text{Frst}(c^j_i), \text{End}(c^j_i)\} \). As detailed below, the factbase \( F \) contains the representation of the input words of length 0 and 1 (Item 1), atoms used as seeds to build larger words (Item 2) and atoms that initialize the emergency brake (Items 3 and 4):

1. \( \text{Frst}(c^j_1), S_1(c^j_0), \text{Nxt}(c^j_0, c^j_1), \text{End}(c^j_1), S_i(c^j_i), \text{Frst}(c^j_i), \text{End}(c^j_i), S_i(c^j_i) \)
2. \( \text{Int}(a), \text{NF}(a, n_{f_1}), R(n_{f_1}), \text{NF}(n_{f_1}, b), D(n_{f_1}, b) \)

\footnote{Note that this contradicts the first item of Theorem 5.1 in (Grahne and Onet 2018). However, no proof is given for that statement, which is incorrectly attributed to (Deutsch, Nash, and Remmel 2008).}
The chase works as follows: after generating a non-final (NF) chain with Rule (47), the brake (B) is made real (R) by Rule (48), which prevents any extension of the non-final chain through restricted rule applications. A final (F) element is added after each non-final element by Rule (49), and from each final element a tape is created, by traversing the chain, marking as done (D) processed elements, thanks to Rules (50)-(53). Figure 2 depicts the result of any \( \mathcal{R} \)-chase derivation from \( \mathcal{R}_w, F \) in which \( \mathcal{R}(b) \) has been derived after exactly one application of Rule (47). Rule (54) sets the initial state on the first cell.

As it is known recursively enumerable sets are a strict subsets of \( \Pi^0_2 \) (Rogers 1987), the following theorem follows.

**Theorem 43.** No computable function \( f \) exists that maps rule sets to rule sets having atomic-head rules such that \( \mathcal{R} \in CT_{\mathcal{R}^{H}}^R \) if and only if \( f(\mathcal{R}) \in CT_{\mathcal{R}^{H}}^R \).

8 Conclusion

As shown in this paper, normalisation procedures do have an impact, sometimes unexpected, on chase termination. This is particularly true regarding the restricted chase, which is the most relevant in practice but also the most difficult to control. We extend the understanding of its behavior by three results. We show that the Datatalog-first strategy is in fact not always the most terminating, which goes against a common belief. We introduce a new atomic-decomposition (two-way), which behaves nicely, in particular regarding the Datalog-first restricted chase, but still has a negative impact on the restricted chase termination. This leads to us to show a more fundamental decidability result, which implies that no computable atomic-decomposition exists that exactly preserves the termination of the restricted chase (i.e., termination and non-termination). Note however that our result does not rule out the existence of a computable normalisation procedure into atomic-head rules that would improve the termination of the restricted chase, although this seems unlikely. Future work includes investigating normalisation procedures for first-order logical formulas, to translate these into the existential rule framework.
Acknowledgements

This work was partly supported by the ANR project CQFD (ANR-18-CE23-0003).

References


