Don't Repeat Yourself: Termination of the Skolem Chase on Disjunctive Existential Rules

Lukas Gerlach

Knowledge-Based Systems Group
Technische Universität Dresden, Germany

Abstract. *Disjunctive Existential Rules* are a fragment of first order logic that is already expressive enough to capture many description logics. *Conjunctive Query answering* is an important reasoning task over such rules. Although this problem is undecidable, different variants of the *chase* provide sound and complete algorithms that can be used as a reasoning basis. Since it is undecidable if these algorithms terminate for a specific rule set, sufficient conditions for termination, called *acyclicity notions*, are introduced. We develop *Disjunctive Model Faithful Acyclicity (DMFA)* as a novel acyclicity notion for the *disjunctive skolem chase* variant by using ideas from *Model Faithful Acyclicity (MFA)* for the non-disjunctive skolem chase and *Restricted Model Faithful Acyclicity (RMFA)* for the restricted chase. Our research shows that our notion captures MFA and is able give a better approximation for termination on disjunctive existential rules. Acyclicity notions for the restricted chase like RMFA still capture DMFA but these notions are not sound for the disjunctive skolem chase. Our results encourage the use of the disjunctive skolem chase in practical applications, which is implementable using well optimized ASP solvers.

1 Introduction

In the field of knowledge representation and database theory, reasoning over knowledge bases is a fundamental task. Knowledge bases consist of facts that are given explicitly and rules the allow to infer further knowledge from existing facts. In our setting, we formulate facts and rules using fragments of first order logic. We consider knowledge bases featuring *existential rules* [4] or *tuple generating dependencies* as they are called in database theory [1]. More precisely, we consider *disjunctive existential rules*, which are more expressive than existential rules and are already powerful enough to capture many description logics [19].

Example 1. The following rules encode information about frozen pizza and pizza delivery.

$$\begin{aligned}
\mathit{Pizza}(x) &\to \mathit{InFridge}(x) \ \vee \exists y.(\mathit{DeliveryService}(y) \wedge \mathit{Delivers}(y,x)) \\
\mathit{DeliveryService}(x) &\to \exists z.(\mathit{Pizza}(z) \wedge \mathit{Delivers}(x,z)) \\
\mathit{InFridge}(x) &\to \mathit{Cold}(x) \\
\mathit{Delivers}(x,y) \wedge \mathit{Slow}(x) &\to \mathit{Cold}(y)
\end{aligned}$$

Intuitively, the first rule states that each pizza is in a fridge or it is delivered by some delivery service. The second rule states that for each delivery service there exists a pizza that was delivered by it. By the third rule everything that is in a fridge must be cold and by the fourth rule something is also cold if it was delivered by something slow, e.g. a slow delivery service. ▲

An important reasoning task for knowledge bases is query answering. Formally, we consider *Conjunctive Queries (CQs)* [1], where the answers of the query are the possible assignments of the free variables.

Example 2. We may ask, which delivery services deliver cold pizza. This can be formulated using the following CQ:

$$\exists y.(\mathit{DeliveryService}(x) \wedge \mathit{Delivers}(x,y) \wedge \mathit{Pizza}(y) \wedge \mathit{Cold}(y))$$

We consider the rules from Example 1 and two different sets of facts A and B.

$$\begin{aligned}
A &\coloneqq \{\, \mathit{DeliveryService}(d), \mathit{Slow}(d) \,\} \\
B &\coloneqq \{\, \mathit{Pizza}(c), \mathit{InFridge}(c) \,\}
\end{aligned}$$

By using the second and the fourth rule starting on A, we can derive that there exists a delivered pizza that is cold. Hence, delivery service d delivers cold pizza and is therefore an answer to the query. Starting on B, there is not necessarily a delivery service at all since the first rule is already fulfilled by $\mathit{Pizza}(c)$ and $\mathit{InFridge}(c)$. We can still derive $\mathit{Cold}(c)$ using the third rule but the query still has no answer since c was arguably not delivered. Hence, there are no delivery services that deliver cold pizza in case B. In fact, there are no delivery services at all. ▲

CQ answering can also be formulated as a decision problem by checking if a possible answer to a CQ over a certain knowledge base is indeed valid. However, CQ answering is undecidable for knowledge bases that feature (disjunctive) existential rules [5].

There are various approaches towards CQ answering. We consider the *(disjunctive) chase* [6,17], which is a sound and complete fixed point algorithm that can be used as a basis for reasoning over (disjunctive) existential rules. There are different variants of the concrete chase procedure like the *restricted chase* [8] and the *skolem chase* [9,16,18]. However, the computation of any of these variants may not terminate, otherwise we could decide CQ answering. It is even undecidable to check if this computation terminates [5,15]. This is also the case for non-disjunctive existential rules.

Example 3. Using the rules from Example 1 and the fact *Pizza*(c), the chase produces the following infinite chain of facts (Fig. 1):

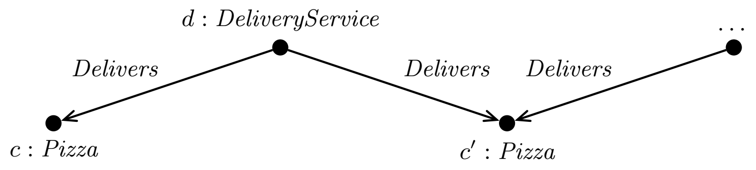


Fig. 1. Non-terminating Chase

Intuitively, we can derive that there exists a *DeliveryService*(d) that delivered *Pizza*(c) using the first rule. By using the second rule on d we can derive that there exists a *Pizza*(c') that d delivers and so on. Note that Fig. 1 only shows this intuition. For a formal definition of the chase sequence, see Definition 24. ▲

There are differences in termination between the different chase variants. For instance, the restricted chase terminates in more cases than the skolem chase. Still, the latter can be implemented using optimized *answer set programming (ASP)* solvers [2,3,13,12,14]. For this reason, we focus on the skolem chase in our work. We discuss this in more detail in Section 2.5.

To resolve the issue of chase termination, sufficient conditions, called *acyclicity notions*, are introduced. Because of the undecidability of chase termination [5,15], this is the best we can do. For the termination of the skolem chase using non-disjunctive existential rules, acyclicity notions like *model faithful acyclicity (MFA)* [9] already provide good results. In the evaluation of [8], MFA marks 72.5% of the rule sets without disjunctions as terminating. However, acyclicity notions for chase variants on disjunctive existential rules barely exist or often do not provide good results, at least for the skolem chase. For the restricted chase, *restricted model faithful acyclicity (RMFA)* marks 34.6% of the rule sets with disjunctions as terminating [8], which can be considered the benchmark for our work. Though, RMFA cannot be used as an acyclicity notion for the skolem chase, since the skolem chase terminates in less cases than the restricted chase. Therefore, if RMFA markes a rule set as terminating, it may still not terminate w.r.t. the skolem chase. To obtain an acyclicity notion for the skolem chase on rule sets with disjunctions, we can modify MFA such that disjunctions are treated as conjunctions. However, this naive extension of MFA marks only 25.3% of the rule sets with disjunctions as terminating in [8]. We think that we can improve upon this result and achieve a result closer to RMFA because the modified version of MFA does not take disjunctions into account.

Our goal is to construct a new acyclicity notion, called *Disjunctive Model Faithful Acyclicity (DMFA)* that guarantees termination for a disjunctive version of the skolem chase. The concrete definition of this chase variant is based on ideas of the disjunctive chase [6], the skolem chase [9,16,18], and the restricted chase

[8]. Also, our chase variant can be implemented using ASP solvers. Using this chase variant as a foundation, our contributions are the following:

- We define DMFA based on ideas of MFA and RMFA.
- We establish relations of DMFA towards MFA and RMFA.
- We investigate complexity bounds for the DMFA check as well as for reasoning with rule sets that are DMFA.

We expect DMFA to improve upon the results of MFA for rule set termination w.r.t. the disjunctive skolem chase. We also expect our notion to come close to the results of RMFA. However, by using the disjunctive skolem chase, DMFA necessarily marks less rule sets as terminating, since it must not mark rule sets as terminating that only terminate w.r.t. to the restricted chase but not w.r.t. the disjunctive skolem chase. To the best of our knowledge, this is the first approach of defining an acyclicity notion tailored towards the disjunctive skolem chase, which promises significant advances, especially for practical concerns.

Note that this work is purely theoretical. The practical evaluation of DMFA against MFA and RMFA is addressed in upcoming work. We also plan to develop cyclicity notions, i.e. sufficient conditions for non-termination, for the disjunctive skolem chase later on.

2 Preliminaries

In this section, we introduce the basic definitions and notions used in the rest of the thesis. We assume familiarity with standard first order logic syntax and semantics.

2.1 Disjunctive Existential Rules

We use standard first order logic notions to describe the disjunctive existential rule fragment. We define $\mathsf{Var}, \mathsf{Const}, \mathsf{Pred}$, and Func to be the pairwise disjoint, countably infinite sets of *variables*, *constants*, *predicate symbols*, and *function symbols*, respectively. Each predicate symbol and function symbol is associated with its *arity* by the function $ar\colon \mathsf{Pred} \cup \mathsf{Func} \to \mathbb{N}$. We assume that $\mathbb{N}$ includes 0. The set Term of *terms* is the smallest set such that $\mathsf{Var} \cup \mathsf{Const} \subseteq \mathsf{Term}$ and $f(t_1, \ldots, t_n) \in \mathsf{Term}$ for all $t_1, \ldots, t_n \in \mathsf{Term}$ and $f \in \mathsf{Func}$ with $ar(f) = n$. If $n = 0$ we write f instead of $f()$. We abbreviate lists of terms $t_1, \ldots, t_n$ as $\vec{t}$. We treat $\vec{t}$ to be the set $\{ t_1, \ldots, t_n \} \subseteq \mathsf{Term}$ when suitable. A term t' is called a *subterm* of another term t if either $t = t'$ or t is of the form $f(\vec{s})$ and t' is a subterm of some term in $\vec{s}$. The term t' is a *proper subterm* of t if t' is a subterm of t and $t \neq t'$.

Definition 4. *A term $t \in \mathsf{Term}$ is* cyclic *if, for some $f \in \mathsf{Func}$ and some lists of terms $\vec{s}$ and $\vec{u}$ of length $ar(f)$, the term $f(\vec{s})$ is a subterm of t and $f(\vec{u})$ is a proper subterm of $f(\vec{s})$.*

Example 5. The term $h(g(f(a), h(b)))$ is cyclic, whereas $h(g(f(a), f(b)))$ is not. ▲

An *atom* is an expression of the form $P(\vec{t})$ for some $P \in \mathsf{Pred}$ and a list of terms $\vec{t}$ of length $ar(P)$. As with terms, we also write P instead of $P()$ if $ar(P) = 0$. The set Atom is the set of all atoms. An atom or term is *ground* if it does not feature syntactic occurrences of variables. We denote the sets of ground atoms and ground terms by GAtom and GTerm, respectively. *Facts* are ground atoms. For comprehensibility, we sometimes represent fact sets featuring predicate symbols of arity at most 2 as graphs in the following way.

Example 6. The fact set $\{ P(a), Q(a), R(a, b) \}$ can be represented as the following graph (Fig. 2):

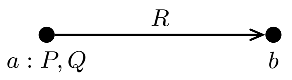


Fig. 2. Fact set as graph

We treat a conjunction of atoms $a_1 \wedge \cdots \wedge a_n$ as the set that contains all of its conjuncts $\{ a_1, \ldots, a_n \}$ when suitable. For a conjunction of atoms φ and a list of variables $\vec{v}$, we write $\varphi(\vec{v})$ to indicate that φ features occurrences of exactly the variables in $\vec{v}$. A *substitution* $\theta : \mathsf{Var} \to \mathsf{GTerm}$ is a partial mapping from variables to ground terms. The application of a substitution θ on an expression φ, denoted as $\varphi\theta$, is the expression that results from φ by replacing every occurrence of a variable x in φ with $\theta(x)$ if $\theta(x)$ is defined.

Example 7. Given a substitution θ that maps every variable in Var to the ground term $f(a)$ and the expression $\varphi := P(g(x), y)$, the application of θ on φ is $\varphi\theta = P(g(f(a)), f(a))$. ▲

In a similar way to substitutions, we allow the remapping of constants in terms and facts. We extend every (total) mapping $\mu : \mathsf{Const} \to \mathsf{GTerm}$ such that for a term or atom t, $\mu(t)$ is the term or atom that results from t by replacing every occurrence of a constant c in t with $\mu(c)$, respectively. The mapping μ can also be applied to term/fact sets and sets of term/fact sets in the obvious way.

Definition 8. *A* (disjunctive existental) rule *is an expression of the form*

$$\forall \vec{x} \forall \vec{y}. [\varphi(\vec{x}, \vec{y}) \to \bigvee_{i=1}^{n} \exists \vec{z}_i . \psi_i(\vec{x}_i, \vec{z}_i)]$$

where $n \geq 1$*;* $\vec{x}, \vec{y}$*, and* $\vec{z}_i$ *are pairwise disjoint lists of variables;* $\bigcup_{i=1}^{n} \vec{x}_i = \vec{x}$*; and* $\varphi(\vec{x}, \vec{y})$ *and* $\psi_i(\vec{x}_i, \vec{z}_i)$ *are non-empty conjunctions of atoms that do not contain occurrences of function symbols or constants.*

The universal quantifiers in rules are usually omitted. The conjunctions of atoms $\varphi(\vec{x}, \vec{y})$ and the expression $\bigvee_{i=1}^{n} \exists \vec{z}_i.\psi_i(\vec{x}_i, \vec{z}_i)$ are called the *body* and the *head* of the rule, respectively. We denote the conjunctions of atoms φ and ψ_i by B_ρ and H_ρ^i, respectively. The variables $\vec{x}$ that occur in the body and the head are called the *frontier* of the rule. We define $\mathit{branch}(\rho) \coloneqq n$. A rule ρ is called *Datalog* if $\mathit{branch}(\rho) = 1$ and H_ρ^1 does not contain existentially quantified variables. We denote the subset of all Datalog rules in a rule set R by R_{dlog}.

An *instance* is a set of function free facts. A *knowledge base* is a pair $\langle R, I\rangle$ where R is a rule set and I is an instance. We assume w.l.o.g. that existentially quantified variables do not reoccur across rules in a given rule set. Combined with Definition 8 we obtain that, for each existentially quantified variable v in a rule set R, there is a unique conjunction of atoms in the head of some rule in R that contains v (†).

Definition 9. *A* model *of a knowledge base $\mathcal{K} \coloneqq \langle R, I\rangle$ is a fact set M such that $I \subseteq M$ and $M \models R$ under first order logic semantics.*

Example 10. Consider a knowledge base $\mathcal{K}$ consisting of an instance $\{\, \mathit{Pizza}(c) \,\}$ and a singleton rule set containing the rule

$$\mathit{Pizza}(x) \to \mathit{InFridge}(x) \vee \exists y.(\mathit{Delivers}(y, x) \wedge \mathit{DeliveryService}(y))$$

The set of fact sets $\{\, \mathit{Pizza}(c), \mathit{InFridge}(c) \,\}$ is a model of $\mathcal{K}$. ▲

2.2 Query Answering

Definition 11. *A* conjunctive query (CQ) *is an expression of the form $\sigma \coloneqq \exists \vec{z}.\varphi(\vec{x}, \vec{z})$ where $\vec{x}$ and $\vec{z}$ are lists of variables and $\varphi(\vec{x}, \vec{z})$ is a conjunction of function free atoms. If the list of variables $\vec{x}$ is empty, σ is called a* boolean conjunctive query (BCQ)*.*

A list of terms $\vec{t}$ is an *answer* to a CQ $\exists \vec{z}.\varphi(\vec{x}, \vec{z})$ w.r.t. a knowledge base $\mathcal{K}$ if, for each model M of $\mathcal{K}$, there exists a substitution θ with $\vec{x}\theta = \vec{t}$ such that $\varphi\theta \subseteq M$. If σ is a BCQ, then σ is *entailed* by $\mathcal{K}$ if the empty list of terms is an answer to σ. The decision problem of deciding if a certain list of terms $\vec{t}$ is an answer to a CQ $\exists \vec{z}.\varphi(\vec{x}, \vec{z})$ can be reduced to BCQ entailment by replacing $\vec{x}$ with $\vec{t}$ in φ.

Example 12. Consider the CQ $\exists x.(\mathit{Pizza}(x) \wedge \mathit{InFridge}(y, x))$. To decide if the term *myFridge* is an answer to the CQ w.r.t. a knowledge base $\mathcal{K}$, we can check if the BCQ $\exists x.(\mathit{Pizza}(x) \wedge \mathit{InFridge}(\mathit{myFridge}, x))$ is entailed by $\mathcal{K}$. ▲

For brevity and since CQ answering can be reduced to BCQ entailment, we only consider BCQ entailment in the following.

2.3 Universal Model Sets

Formally, to check BCQ entailment for a knowledge base, we would need to check entailment of the BCQ for every of the infinitely many, possibly infinitely large models of the knowledge base. It suffices to consider only a universal model set $\mathcal{U}$ [6] such that for every model M there exists a model in $\mathcal{U}$ that can be homomorphically embedded into M. For two fact sets F and F', a mapping τ over GTerm is called a *homomorphism* from F to F' if $\tau(t) = t$ for all $t \in \mathsf{Const}$ and $\tau(F) \subseteq F'$ where $\tau(F) \coloneqq \{ P(\tau(t_1), \ldots, \tau(t_n)) \mid P(t_1, \ldots, t_n) \in F \}$.

Definition 13. *A set of models $\mathcal{U}$ of a knowledge base $\mathcal{K}$ is called a* universal model set *if, for every model M of $\mathcal{K}$, there exists a model $U \in \mathcal{U}$ such that there exists a homomorphism from U to M.*

Proposition 14. *A BCQ $\sigma \coloneqq \exists \vec{z}.\varphi(\vec{z})$ is* entailed *by a knowledge base $\mathcal{K}$ iff it is entailed by each model in some universal model set $\mathcal{U}$ of $\mathcal{K}$, that is, for each $U \in \mathcal{U}$, there exists a substitution θ with $\varphi\theta \subseteq U$.*

Proof. We show both directions of the claim separately. Consider the BCQ $\sigma \coloneqq \exists \vec{z}.\varphi(\vec{z})$ and a knowledge base $\mathcal{K}$. If σ is entailed by $\mathcal{K}$, then by definition there exists a substitution θ with $\varphi\theta \subseteq M$ for every model M of $\mathcal{K}$. In particular, this holds for every model U in some universal model set $\mathcal{U}$ of $\mathcal{K}$. For the other direction, we show that σ is entailed by $\mathcal{K}$ if it is entailed by each model in some universal model set $\mathcal{U}$ of $\mathcal{K}$.

1. Let M be a model of $\mathcal{K}$ and let $\mathcal{U}$ be a universal model set for a knowledge base $\mathcal{K}$ such that, for each model $U \in \mathcal{U}$, there exists a substitution θ with $\varphi\theta \subseteq U$.
2. By (1): There exists a model $U \in \mathcal{U}$ such that there exists a homomorphism τ from U to M.
3. By (1) and (2): $\tau \circ \theta$ is a substitution with $\varphi(\tau \circ \theta) \subseteq M$.
4. By (3): σ is entailed by $\mathcal{K}$.

□

Example 15. Consider the BCQ $\sigma \coloneqq \exists x.(\mathit{Pizza}(x) \wedge \mathit{InFridge}(x))$ and the knowledge base $\mathcal{K}$ consisting of the instance $\{ \mathit{Pizza}(c) \}$ and a singleton rule set containing the rule:

$$\mathit{Pizza}(x) \to \mathit{InFridge}(x) \vee \exists y.(\mathit{Delivers}(y, x) \wedge \mathit{DeliveryService}(y))$$

The following set of fact sets is a universal model set of $\mathcal{K}$:

$$\{ \{ \mathit{Pizza}(c), \mathit{InFridge}(c) \}, \{ \mathit{Pizza}(c), \mathit{Delivers}(f(c), c), \mathit{DeliveryService}(f(c)) \} \}$$

By Proposition 14, σ is not entailed by $\mathcal{K}$ since there is no substitution θ with:

$$(\mathit{Pizza}(x) \wedge \mathit{InFridge}(x))\theta \subseteq \{ \mathit{Pizza}(c), \mathit{Delivers}(f(c), c), \mathit{DeliveryService}(f(c)) \}$$

▲

2.4 Disjunctive Skolem Chase

In general, the chase is a sound and complete procedure for computing a universal model set of a knowledge base. For our concrete definition, the chase result is only defined if the computation terminates. Hence, technically the procedure is only complete for terminating cases.

For the definition of the disjunctive skolem chase, we introduce a formal way of applying rules to facts. We use ideas from existing definitions of the non-disjunctive skolem chase [9,16,18], the disjunctive chase [6] and the restricted chase [8]. One issue is, how to generate new facts for rules with existentially quantified variables. We tackle this issue by skolemizing the rules first. That means, we introduce function symbols for each existentially quantified variable. Those function symbols are used as a naming convention, to keep track of facts that have already been introduced by a specific rule.

Definition 16. *The* skolemization $sk(\rho)$ *of a rule* ρ *is defined as*

$$B_\rho(\vec{x},\vec{y}) \to \bigvee\nolimits_{i=1}^{n} \psi_i^{sk}(\vec{x})$$

where $n = branch(\rho)$, $\vec{x}$ *is the frontier of* ρ, *and for each* $1 \le i \le n$, $\psi_i^{sk}(\vec{x})$ *is obtained from* $H_\rho^i(\vec{x}_i,\vec{z}_i)$ *by replacing each of the existentially quantified variables* $z \in \vec{z}_i$ *with* $f^z(\vec{x})$ *where* f^z *is a fresh function symbol unique for* z *with arity* $|\vec{x}|$.

Note that the mapping from z to $f^z(\vec{x})$ is well defined within a rule set because there is a unique conjunction of atoms that contains z according to (†) and by that, there is a unique term $f^z(\vec{x})$ for each z. For readability, we also denote the conjunctions of atoms in the head of $sk(\rho)$ by $sk(H_\rho^i) := \psi_i^{sk}$. To see why the mapping from z to $f^z(\vec{x})$ may not be well defined if (†) does not hold, regard the following example.

Example 17. Consider the following rule set for which (†) does not hold.

$$\begin{aligned} \rho_1 &= P(x) \to \exists z.Q(x,z) & sk(\rho_1) &= P(x) \to Q(x, f^z(x)) \\ \rho_2 &= Q(x,y) \to \exists z.R(x,y,z) & sk(\rho_2) &= Q(x,y) \to R(x,y,f^z(x,y)) \end{aligned}$$

The variable z is mapped to two different terms in this rule set, so the image of z is not well defined. In particular, the function symbol f^z itself is also not well defined because it occurs with two different arities. ▲

Using skolemized rules, we define how a rule is applied to a set of facts. For rule sets without disjunctions, these applications are usually defined for a set of facts and they yield another set of facts [9,18]. Regarding disjunctive rules, we have to look at the different head disjuncts individually. Thus, we use a set of fact sets rather than just a single set of facts to store the results of the rule application for each individual disjunct [6].

Definition 18. *A pair* $\lambda := \langle \rho, \theta \rangle$ *of a rule* ρ *and a substitution* θ *that is defined exactly on the universally quantified variables in* ρ *is called* trigger. *In the context of a fact set* F*:*

– *The trigger λ is* active *if $B_\rho\theta \subseteq F$.*
– *The trigger λ is* obsolete *if $sk(H^i_\rho)\theta \subseteq F$ for some $1 \le i \le branch(\rho)$.*
– *The trigger λ is* applicable *to F if it is active and it is not obsolete.*

If λ is applicable to F, then the application *of λ on F is defined as the set of fact sets $\lambda(F) := \{ F \cup sk(H^i_\rho)\theta \mid 1 \le i \le branch(\rho) \}$.*

Example 19. Consider $F := \{ \mathit{Pizza}(c) \}$, a substitution θ with $\theta(x) = c$ and $\rho := \mathit{Pizza}(x) \to \mathit{InFridge}(x) \lor \exists y.(\mathit{Delivers}(y, x) \land \mathit{DeliveryService}(y))$.

$$\begin{aligned}\langle \rho, \theta\rangle(F) = \{ &\{ \mathit{Pizza}(c), \mathit{Delivers}(f^y(c), c), \mathit{DeliveryService}(f^y(c)) \},\\ &\{ \mathit{Pizza}(c), \mathit{InFridge}(c) \}\}\end{aligned}$$

▲

Definition 20. *Let R be a rule set, F a fact set and $\mathcal{F}$ a set of fact sets.*

– *The* application *of R on F is defined as the set of fact sets*

$$R(F) := \begin{cases} \{ F \}, & \text{if } \Lambda^F_R = \emptyset \\ \bigcup_{\lambda \in \Lambda^F_R} \lambda(F), & \text{otherwise} \end{cases}$$

where Λ^F_R is the set of all triggers using rules in R that are applicable to F.
– *The* application *of R on $\mathcal{F}$ is the set of fact sets $R(\mathcal{F}) := \bigcup_{F \in \mathcal{F}} R(F)$.*
– *The* saturating application *of R_{dlog} on F, written $R^*_{dlog}(F)$, is defined as the smallest superset of the fact set F such that $R_{dlog}(R^*_{dlog}(F)) = \{ R^*_{dlog}(F) \}$.*
– *The* saturating application *of R_{dlog} on $\mathcal{F}$ is defined as the set of fact sets $R^*_{dlog}(\mathcal{F}) := \{ R^*_{dlog}(F) \mid F \in \mathcal{F} \}$.*

The saturating application of Datalog rules is used in Definition 24 as this can lead to termination of the disjunctive skolem chase in more cases. We discuss this in detail in Section 2.5.

Remark 21. The saturating application $R^*_{\text{dlog}}(F)$ of R_{dlog} on a fact set F can be regarded as a sequence of applications of R_{dlog} starting on F. Even though the individual applications of R_{dlog} may yield multiple fact sets, these fact sets collapse into a single fact set, namely $R^*_{\text{dlog}}(F)$, after a finite number of applications. We obtain that there is a number $n \in \mathbb{N}$ such that

$$R^n_{\text{dlog}}(F) := \underbrace{R_{\text{dlog}}(\dots R_{\text{dlog}}(}_{n \text{ times}} F) \dots) = \{ R^*_{\text{dlog}}(F) \}$$

We define $R^0_{\text{dlog}}(F) := \{ F \}$. ▲

A valuable insight for later considerations is that constant remappings may allow more triggers featuring Datalog rules to be applied.

Example 22. Consider the fact set $F \coloneqq \{ P(a,b) \}$ and the Datalog rule:

$$\rho \coloneqq P(x,x) \to Q(x)$$

There exists no trigger featuring ρ that is applicable to F. However, if we remap constants in F, we can map both a and b to a. Then, the trigger $\langle \rho, \theta \rangle$ with θ mapping x to a is applicable. ▲

In general, we obtain the following result.

Lemma 23. *For each mapping* $\mu : \mathsf{Const} \to \mathsf{GTerm}$, *each rule set* R*, and each fact set* F*, we have* $\mu(R^*_{dlog}(F)) \subseteq R^*_{dlog}(\mu(F))$.

Proof. According to Remark 21, we show that $\mu(F_i)$ is a subset of some fact set in $R^i_{\text{dlog}}(\mu(F))$ for each fact set $F_i \in R^i_{\text{dlog}}(F)$ via induction over $i \in \mathbb{N}$. For the base case $i = 0$, we have that $F^0 \in R^0_{\text{dlog}}(F) = \{ F \}$ iff $\mu(F^0) \in R^0_{\text{dlog}}(\mu(F)) = \{ \mu(F) \}$. For the induction hypothesis, we assume that $\mu(F_k)$ is a subset of some fact set in $R^k_{\text{dlog}}(\mu(F))$ for each fact set $F_k \in R^k_{\text{dlog}}(F)$ where $k \in \mathbb{N}$. For the induction step, we show that $\mu(F_{k+1})$ is a subset of some fact set in $R^{k+1}_{\text{dlog}}(\mu(F))$ for each fact set $F_{k+1} \in R^{k+1}_{\text{dlog}}(F)$.

1. Let $F_{k+1} \in R^{k+1}_{\text{dlog}}(F)$ and $F_{k+1} \notin R^k_{\text{dlog}}(F)$. Otherwise the result follows from the induction hypothesis.
2. By (1): There is a fact set $F_k \in R^k_{\text{dlog}}(F)$ and a trigger $\langle \rho, \theta \rangle$ with $\rho \in R_{\text{dlog}}$ and $F_{k+1} \in \langle \rho, \theta \rangle (F_k)$.
3. By (2): $\mu(F_{k+1}) \in \langle \rho, \mu \circ \theta \rangle (\mu(F_k))$
4. By (3) and the induction hypothesis: $\mu(F_{k+1})$ is a subset of some fact set in $\langle \rho, \mu \circ \theta \rangle (F')$ for some $F' \in R^k_{\text{dlog}}(\mu(F))$.
5. By (4): $\mu(F_{k+1})$ is a subset of some fact set in $R_{\text{dlog}}(R^k_{\text{dlog}}(\mu(F)))$.
6. By (5): $\mu(F_{k+1})$ is a subset of some fact set in $R^{k+1}_{\text{dlog}}(\mu(F))$.

□

The formal application of rules to facts enables us to compute new facts. To obtain the definition of the disjunctive skolem chase, we apply rules exhaustively, until no new facts are obtained.

Definition 24. *The* chase sequence *of a knowledge base* $\mathcal{K} \coloneqq \langle R, I \rangle$ *is the sequence of sets of fact sets* $\mathcal{F}^0_{\mathcal{K}}, \mathcal{F}^1_{\mathcal{K}}, \dots$ *defined inductively via* $\mathcal{F}^0_{\mathcal{K}} \coloneqq \{ I \}$ *and* $\mathcal{F}^i_{\mathcal{K}} \coloneqq R(R^*_{dlog}(\mathcal{F}^{i-1}_{\mathcal{K}}))$ *for all* $i > 0$.

Recall that we use the saturating application of R_{dlog} in each step of the chase sequence since this leads to termination more often (see Section 2.5). A knowledge base $\mathcal{K}$ is *terminating* if $\mathcal{F}^k_{\mathcal{K}} = \mathcal{F}^{k+1}_{\mathcal{K}}$ for some $k \in \mathbb{N}$. A rule set R is *terminating* if $\langle R, I \rangle$ is terminating for every instance I. The *chase* of a terminating knowledge base $\mathcal{K}$ is defined as $Ch(\mathcal{K}) \coloneqq \mathcal{F}^k_{\mathcal{K}}$, where k is the smallest number such that $\mathcal{F}^k_{\mathcal{K}} = \mathcal{F}^{k+1}_{\mathcal{K}}$. If such a k does not exist, i.e. if $\mathcal{K}$ is not terminating, then $Ch(\mathcal{K})$ is undefined. Note that both problems of checking if a rule set is terminating [15] and checking if a knowledge base is terminating [5,10] are undecidable, even if the rules do not contain disjunctions.

Proposition 25. *The chase $Ch(\mathcal{K})$ of a terminating knowledge base $\mathcal{K}$ is a universal model set of $\mathcal{K}$.*

Proof. We prove the claim in two parts. First, we show that each $U \in Ch(\mathcal{K})$ is a model of $\mathcal{K} = \langle R, I \rangle$.

1. Since $\mathcal{F}^0_{\mathcal{K}} = \{ I \}$: $I \subseteq U$.
2. Suppose for a contradiction that $U \not\models R$.
3. By (2): There exists a rule $\rho \in R$ that is not satisfied by U.
4. By (3): There exists a trigger $\langle \rho, \theta \rangle$ that is applicable to U.
5. By (4): $U \notin Ch(\mathcal{K})$. ↯
6. By (2) and (5): $U \models R$.
7. By (1) and (6): U is a model of $\mathcal{K}$.

Second, we show the universality of $Ch(\mathcal{K})$, that is, we show that for every model M of $\mathcal{K}$ there exists a model $U \in Ch(\mathcal{K})$ such that there exists a homomorphism τ from U to M. More precisely, we show $\tau(F) \subseteq M$ for some $F \in \mathcal{F}^i_{\mathcal{K}}$ for all $i \in \mathbb{N}$ by induction over i. For the base case $i = 0$, we have that $\mathcal{F}^0_{\mathcal{K}} = \{ I \}$. Hence, we show that $\tau(I) \subseteq M$.

1. Since M is a model of $\mathcal{K}$, $I \subseteq M$.
2. By (1) and since $\tau(t) = t$ for all $t \in \mathsf{Const}$: $\tau(I) \subseteq M$.

For the induction hypothesis, we assume that $\tau(F') \subseteq M$ for some $F' \in \mathcal{F}^{k-1}_{\mathcal{K}}$. For the induction step, we show that $\tau(F) \subseteq M$ for some $F \in \mathcal{F}^k_{\mathcal{K}}$ where $k \geq 1$. We do this by constructing a suitable fact set F from F'. We either have that $F' \in \mathcal{F}^k_{\mathcal{K}}$ and the claim for the induction step follows trivially if we set $F := F'$ or there exists a trigger $\lambda := \langle \rho, \theta \rangle$ with $\lambda(F') \subseteq \mathcal{F}^k_{\mathcal{K}}$. We assume the latter in the following and show that $\tau(F' \cup sk(H^i_\rho)\theta) \subseteq M$ for some $1 \leq i \leq branch(\rho)$. The claim then follows if we set $F := F' \cup sk(H^i_\rho)\theta$, because then we obtain $F' \cup sk(H^i_\rho)\theta \in \lambda(F') \subseteq \mathcal{F}^k_{\mathcal{K}}$.

1. By induction hypothesis, $\tau(F') \subseteq M$.
2. By (1) and since $B_\rho\theta \subseteq F'$: $B_\rho(\tau \circ \theta) \subseteq M$.
3. By (2) and since $M \models R$: There exists a substitution θ' with $B_\rho(\tau \circ \theta) = B_\rho\theta'$ and $H^i_\rho\theta' \subseteq M$ for some $1 \leq i \leq branch(\rho)$.
4. By (3): The substitutions $\tau \circ \theta$ and θ' may only differ for variables that only occur in H^i_ρ, hence variables that are replaced by the skolemization. We define the mapping of the newly introduced skolem terms in τ in the fashion that $\tau(sk(H^i_\rho)\theta) = H^i_\rho\theta'$. This is possible w.l.o.g. since $sk(H^i_\rho)\theta$ is only introduced for the single application of λ. Otherwise λ is already obsolete.
5. By (4): $\tau(sk(H^i_\rho)\theta) \subseteq M$.
6. By (1) and (5): $\tau(F' \cup sk(H^i_\rho)\theta) \subseteq M$.

□

Since the chase on a terminating knowledge base $\mathcal{K}$ yields a universal model, we can use its result to check BCQ entailment for $\mathcal{K}$.

Corollary 26. *A BCQ* $\exists \vec{z}.\varphi(\vec{z})$ *is* entailed *by a terminating knowledge base* $\mathcal{K}$ *iff it is entailed for each* $F \in Ch(\mathcal{K})$*, that is for each* $F \in Ch(\mathcal{K})$ *there exists a substitution* θ *with* $\varphi\theta \subseteq F$*.*

The claim follows directly from Propositions 14 and 25.

Example 27. Based on Example 15, we present the universal model set that the disjunctive skolem chase yields for the knowledge base consisting of the instance $I \coloneqq \{\, Pizza(c) \,\}$ and the rule set containing the following two rules:

$$\begin{aligned} Pizza(x) &\to InFridge(x) \vee \exists y.(DeliveryService(y) \wedge Delivers(y, x)) \\ InFridge(x) &\to Cold(x) \end{aligned}$$

We obtain the following chase sequence:

$$\begin{aligned} \mathcal{F}^0_{\mathcal{K}} &= \{\, I \,\} \\ \mathcal{F}^1_{\mathcal{K}} &= \{\, F_1 = I \cup \{\, InFridge(c) \,\} \\ &\qquad F_2 = I \cup \{\, Delivers(f^y(c), c), DeliveryService(f^y(c)) \,\} \,\} \\ \mathcal{F}^2_{\mathcal{K}} &= \{\, F_1 \cup \{\, Cold(c) \,\}, F_2 \,\} = Ch(\mathcal{K}) \end{aligned}$$

The set of fact sets $Ch(\mathcal{K})$ is a universal model set of $\mathcal{K}$ by Proposition 25. ▲

2.5 Discussion on some Properties of the Disjunctive Skolem Chase

We discuss some design decisions for the definition of the disjunctive skolem chase. We look into the skolemization, the notion of applicability and the prioritized application of Datalog rules in detail. After that, we discuss theoretical and practical advantages of our chase definition over the restricted chase.

First, we consider the skolemization. Recall from Definition 16 that skolemized rules are of the form

$$B_\rho(\vec{x}, \vec{y}) \to \bigvee\nolimits_{i=1}^{n} \psi_i^{sk}(\vec{x})$$

where $n = branch(\rho)$, $\vec{x}$ is the frontier of ρ, and for each $1 \leq i \leq n$, $\psi_i^{sk}(\vec{x})$ is obtained from $H^i_\rho(\vec{x}_i, \vec{z}_i)$ by replacing each of the existentially quantified variables $z \in \vec{z}_i$ with $f^z(\vec{x})$ where f^z is a fresh function symbol unique for z with arity $|\vec{x}|$. It is possible to define a correct chase procedure using $f^z(\vec{x}, \vec{y})$ rather than $f^z(\vec{x})$. But when computing the chase, less facts are introduced if we use terms of the form $f^z(\vec{x})$. Taking a hint from [16], we say that the use of $f^z(\vec{x}, \vec{y})$ yields the *oblivious* version our chase definition, whereas the use of $f^z(\vec{x})$ gives us a *semi-oblivious* version. The semi-oblivious chase version leads to a better performance in general and even to termination in more cases. In our case, the skolem chase refers to the semi-oblivious version.

Example 28. We show that the semi-oblivious chase terminates in more cases than the oblivious chase. Consider the instance $I \coloneqq \{\, P(a, b) \,\}$ and the singleton rule set R with the rule:

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

Using the semi-oblivious skolemization we obtain the skolemized rule:

$$P(x,y) \to P(x, f^z(x))$$

Whereas with the oblivious skolemization, we obtain:

$$P(x,y) \to P(x, f^z(x,y))$$

The semi-oblivious chase of $\langle R, I \rangle$ is $\{ \{ P(a,b), P(a, f^z(a)) \} \}$ (left in Fig. 3). The oblivious chase of $\langle R, I \rangle$ is not defined in this case, since $\langle R, I \rangle$ is not terminating w.r.t. the oblivious chase. The following infinite chain of facts is derived in the chase sequence of $\langle R, I \rangle$ (right in Fig. 3).

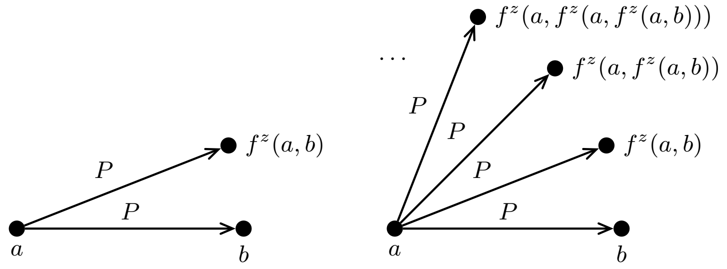


Fig. 3. Semi-Oblivious vs Oblivious Chase

▲

Next, we justify our design decision for the notion of applicability. Recall Definition 18. A trigger λ is applicable to some fact set F if λ is active and not obsolete w.r.t. F. It is possible to define a correct variant of the disjunctive skolem chase requiring only that a trigger is active to be applicable, though other adjustments in the definition are required as well. We demand that a trigger is not obsolete to prevent application if the trigger has been applied before. In some cases this even prevents some triggers from being applied at all. This not only enhances computational performance but also leads to termination more often.

Example 29. Assume that a trigger is applicable if it is active (and possibly obsolete). Consider the rule set R consisting of the following rules:

$$\begin{aligned} \mathit{Pizza}(x) &\to \mathit{Cold}(x) \lor \exists y.(\mathit{NextOrder}(x,y) \land \mathit{Pizza}(y)) \\ \mathit{SlowDeliveryService}(x) &\to \exists z.(\mathit{Pizza}(z) \land \mathit{Cold}(z)) \end{aligned}$$

Additionally, consider a slow pizza delivery service. We represent this by the instance $I \coloneqq \{ \mathit{SlowDeliveryService}(s) \}$. Intuitively, we expect that no new pizza is ordered from the slow delivery service. Although, assuming that a trigger is applicable if it is active, the chase sequence for $\mathcal{K} \coloneqq \langle R, I \rangle$ is:

$$
\begin{aligned}
&\mathcal{F}^0_{\mathcal{K}} = \{ I \} \\
&\mathcal{F}^1_{\mathcal{K}} = \{ F_0 = (I \cup \{ \mathit{Pizza}(f^z), \mathit{Cold}(f^z) \}) \} \\
&\mathcal{F}^2_{\mathcal{K}} = \{ F_0, F_1 = (F_0 \cup \{ \mathit{NextOrder}(f^z, f^y(f^z)), \mathit{Pizza}(f^y(f^z)) \}) \} \\
&\mathcal{F}^3_{\mathcal{K}} = \{ F_0, F_1, F_1 \cup \{ \mathit{Cold}(f^y(f^z)) \}, \\
&\qquad\quad F_1 \cup \{ \mathit{NextOrder}(f^y(f^z), f^y(f^y(f^z))), \mathit{Pizza}(f^y(f^y(f^z))) \} \} \\
&\qquad\quad \dots
\end{aligned}
$$

Note that besides the problem with non-termination, there is another issue, namely that intermediate fact sets like F_1 are not removed. This is a side effect from the definition of rule set application, which relies on the fact that obsolete triggers are not applied. Opposingly, with the original definition for applicability that requires triggers to be active and not obsolete, we get:

$$
\begin{aligned}
&\mathcal{F}^0_{\mathcal{K}} = \{ I \} \\
&\mathcal{F}^1_{\mathcal{K}} = \{ F_0 = (I \cup \{ \mathit{Pizza}(f^z), \mathit{Cold}(f^z) \}) \} = Ch(\mathcal{K})
\end{aligned}
$$

▲

A closely related property of our chase definition is the prioritized application of Datalog rules, which is not part of some existing definitions of the skolem chase as well [18]. In general, this can lead to a more efficient computation of the chase sequence and because of our notion of applicability, this can even lead to termination more often.

Example 30. Assume that we do not prioritize the application of Datalog rules. Consider the rule set R similar to Example 29 consisting of the following rules:

$$
\begin{aligned}
\mathit{Pizza}(x) &\to \mathit{Cold}(x) \lor \exists y.(\mathit{NextOrder}(x, y) \land \mathit{Pizza}(y)) \\
\mathit{Pizza}(x) \land \mathit{SlowDelivery}(x) &\to \mathit{Cold}(x)
\end{aligned}
$$

Additionally, consider that a pizza was ordered but the delivery was very slow. We represent this by the instance $I \coloneqq \{ \mathit{Pizza}(c), \mathit{SlowDelivery}(c) \}$. We expect, that no new pizza is ordered given the condition. Although, if we do not prioritize the application of Datalog rules, the chase sequence for $\mathcal{K} \coloneqq \langle R, I \rangle$ is:

$$
\begin{aligned}
&\mathcal{F}^0_{\mathcal{K}} = \{ I \} \\
&\mathcal{F}^1_{\mathcal{K}} = \{ F_0 = (I \cup \{ \mathit{Cold}(c) \}), F_1 = (I \cup \{ \mathit{NextOrder}(o, f^y(c)), \mathit{Pizza}(f^y(c)) \}) \} \\
&\mathcal{F}^2_{\mathcal{K}} = \{ F_0, F_1 \cup \{ \mathit{Cold}(c) \}, F_1 \cup \{ \mathit{Cold}(f^y(c)) \}, \\
&\qquad\quad F_1 \cup \{ \mathit{NextOrder}(f^y(c), f^y(f^y(c))), \mathit{Pizza}(f^y(f^y(c))) \} \} \\
&\qquad\quad \dots
\end{aligned}
$$

Whereas with the prioritized application of Datalog rules we get:

$$\mathcal{F}^0_{\mathcal{K}} = \{ \{ \mathit{Pizza}(c), \mathit{SlowDelivery}(c) \} \}$$
$$\mathcal{F}^1_{\mathcal{K}} = \{ \{ \mathit{Pizza}(c), \mathit{SlowDelivery}(c), \mathit{Cold}(c) \} \} = R^*_{\text{dlog}}(\mathcal{F}^0_{\mathcal{K}}) = Ch(\mathcal{K})$$

▲

Despite the more complicated definition of applicability, the complexity of checking applicability for a single rule is still in NP, which is the same as for skolem chase definitions for rule sets without disjunctions [16].

Proposition 31. *Let F be a fact set and ρ a rule. Deciding if there exists a substitution θ, such that $\langle \rho, \theta \rangle$ is applicable to F, is* NP*-complete.*

Proof (Sketch). First, we show membership in NP. We guess a substitution θ nondeterministically. The size of θ is polynomial w.r.t. the number of variables in ρ and functions symbols in F and $sk(\rho)$. To verify that $\langle \rho, \theta \rangle$ is applicable to F, we check that $B_\rho\theta \subseteq F$ and $sk(H^i_\rho)\theta \nsubseteq F$ for all $1 \leq i \leq \mathit{branch}(\rho)$. This check is possible in polynomial time w.r.t. the size of ρ and F.

For hardness, we provide a reduction from the decision problem of checking if there exists a substitution θ such that $\langle \rho, \theta \rangle$ is active w.r.t. F, where ρ does not contain disjunctions. This problem is NP-hard according to [16]. We construct the rule ρ' by replacing predicate symbols in the head of ρ by fresh predicate symbols of the same arity (that do not occur in ρ or F). This ensures, that no trigger featuring ρ' is obsolete. The construction can be done in polynomial time. There is a substitution θ such that $\langle \rho, \theta \rangle$ is active w.r.t. F iff there is a substitution θ' such that $\langle \rho', \theta' \rangle$ is applicable to F. Hence, NP-hardness follows. □

Finally, we argue why we use the disjunctive skolem chase and not the restricted chase, even though the latter terminates in more cases. This property results from the applicability condition of the restricted chase. We briefly introduce this condition for comparison.

Definition 32. *A trigger $\lambda \coloneqq \langle \rho, \theta \rangle$ is* r-obsolete *w.r.t. a fact set F if $F \models \exists \vec{z}_i.H^i_\rho(\vec{x}_i, \vec{z}_i)\theta$ for some $1 \leq i \leq \mathit{branch}(\rho)$. A trigger λ is* r-applicable *to F if it is active and not r-obsolete.*

The r-obsolete triggers for a rule ρ form a superset of the obsolete triggers for ρ. Thus, r-applicability is a stronger condition than our applicability condition for the disjunctive skolem chase in Definition 18.

Example 33. Consider the fact set $\{\mathit{Pizza}(c), \mathit{DeliveryService}(d), \mathit{Delivers}(d, c)\}$ and the rule $\rho \coloneqq \mathit{DeliveryService}(x) \to \exists z.(\mathit{Pizza}(z) \wedge \mathit{Delivers}(x, z))$. The trigger $\langle \rho, \theta \rangle$ with $\theta(x) = d$ is r-obsolete but not obsolete.

However, the biggest advantage of using the disjunctive skolem chase over the restricted chase is that it can be implemented using ASP solvers [13], which are well optimized [2,3,12,14]. It is not surprising that rule sets can be represented in ASP, since ASP is based on rules of a similar form. Although ASP does not allow for existential quantifiers, function symbols are allowed so that skolemized rule sets can be encoded. What is more crucial is the applicability condition for a rule. ASP applies active triggers by design. Also, ASP computes $\subseteq$-minimal models [13], which is essentially what we achieve by requiring non-obsolete triggers in our applicability condition. In general, we cannot compute the restricted chase using ASP as r-applicability, more precisely r-obsoleteness, requires an entailment check. Hence, using the ASP based implementation, the disjunctive skolem chase promises to be more performant in practice.

From a more theoretical point of view, r-applicability indeed leads to a higher computational complexity. Consider the problem of checking if for a given fact set F and a rule ρ, there exists a substitution θ such that $\langle \rho, \theta \rangle$ is applicable to F. As seen in Proposition 31, this check is NP-complete for the disjunctive skolem chase. For the restricted chase or *standard chase*, as it is also referred to in other work, we check for r-applicability instead of applicability. This check is Σ_2^{P}-complete [16].

3 Disjunctive Model Faithful Acyclicity

In this section we define *Disjunctive Model Faithful Acyclicity (DMFA)* as a sufficient condition for rule set termination of the disjunctive skolem chase. If we can verify that a rule set R is terminating, we can safely compute a universal model set for every knowledge base featuring R using the disjunctive skolem chase. By Corollary 26, we are therefore able to decide BCQ entailment for each such knowledge base.

3.1 The idea behind DMFA

For rule sets without disjunctions, rule set termination can be reduced to knowledge base termination. The reduction constructs a generalized knowledge base for the rule set using a special instance.

Definition 34. *The* critical instance $I_R^\star$ *of a rule set R is the instance that contains all facts that can be constructed using predicate symbols in R and a fresh constant $\star$.*

According to Theorem 2 in [18], we get the anticipated result.

Theorem 35. *A rule set R without disjunctions is terminating if and only if $\langle R, I_R^\star \rangle$ is terminating.*

Note that [18] uses a slightly different definition of the skolem chase but this does not affect the result for rule sets without disjunctions in our case. The following observations help to see why Theorem 35 holds. At first, we observe that for a rule set R without disjunctions and a fact set F, $R(F)$ subsumes $R(F')$

for every fact set $F' \subseteq F$, i.e. each fact set in $R(F')$ is a subset of some fact set in $R(F)$. Secondly, consider the mapping $\sigma : C \to C$ with $\sigma(c) \coloneqq \star$ for all $c \in \mathsf{Const}$. For every instance I that uses only predicate symbols in R, we observe $\sigma(I) \subseteq I^\star_R$. As anticipated by the first observation, this also holds for every chase step, i.e. $\mathcal{F}^i_{\langle R, I^\star_R \rangle}$ subsumes $\sigma(\mathcal{F}^i_{\langle R, I \rangle})$ for every $i \in \mathbb{N}$. In this sense, the chase on $\langle R, I^\star_R \rangle$ iteratively computes a set of fact sets that subsumes all chase sequences of all other knowledge bases featuring R (‡). Thus, R is terminating if $\langle R, I^\star_R \rangle$ is terminating. The other direction follows trivially.

Unfortunately, the first observation is not true anymore for rule sets that contain disjunctions. Indeed, Theorem 35 does not hold for disjunctive rule sets in general, as shown by the following counterexample.

Example 36. Consider the rule set R that contains only the rule:

$$\rho \coloneqq \mathit{Pizza}(x) \to \mathit{Cold}(x) \vee \exists y.(\mathit{NextOrder}(x, y) \wedge \mathit{Pizza}(y))$$

The critical instance of R is $I^\star_R = \{\, \mathit{Pizza}(\star), \mathit{Cold}(\star), \mathit{NextOrder}(\star, \star) \,\}$. Given a substitution θ that maps x to $\star$, the trigger $\langle \rho, \theta \rangle$ is not applicable to $I^\star_R$ because $\mathit{Cold}(\star)$ is already part of the instance. Hence, $\langle R, I^\star_R \rangle$ is terminating. In contrast, the rule set R is not terminating. Consider the instance $I \coloneqq \{\, \mathit{Pizza}(c) \,\}$ and a substitution θ' that maps x to c. The application of $\langle \rho, \theta' \rangle$ is not prevented and thus, $\mathcal{K} \coloneqq \langle R, I \rangle$ is not terminating.

$$\begin{aligned}
\mathcal{F}^0_{\mathcal{K}} &= \{\, I \,\} \\
\mathcal{F}^1_{\mathcal{K}} &= \{\, F_1 = (I \cup \{\, \mathit{Cold}(c) \,\}), \\
&\qquad F_2 = (I \cup \{\, \mathit{NextOrder}(c, f^y(c)), \mathit{Pizza}(f^y(c)) \,\}) \,\} \\
\mathcal{F}^2_{\mathcal{K}} &= \{\, F_1, F_2 \cup \{\, \mathit{Cold}(f^y(c)) \,\}, \\
&\qquad F_2 \cup \{\, \mathit{NextOrder}(f^y(c), f^y(f^y(c))), \mathit{Pizza}(f^y(f^y(c))) \,\} \,\} \\
&\dots
\end{aligned}$$

▲

We elaborate on why this example fails. The knowledge base $\langle R, I^\star_R \rangle$ does not fulfill the property (‡). This is because some of the disjunctive rules may already be satisfied by the critical instance itself, so for some rules there are no applicable triggers. Recalling Definition 18, a trigger is applicable to some fact set F if it is active and not obsolete w.r.t. F. The main goal of obsoleteness is to prevent application if a trigger was applied before. The introduction of new facts may make a trigger obsolete even if this trigger has not been applied yet. In Example 36, this leads to the initial prevention of trigger applications for the critical instance.

To address this issue, we can compute the chase for the critical instance using a modified rule set where all disjunctions are replaced by conjunctions.

Proposition 37. *Consider a rule set R and a rule set R' that results from R by replacing disjunctions with conjunctions. If R' is terminating, then R is terminating.*

Proof. For every instance I, $i \in \mathbb{N}$ and fact set $F \in \mathcal{F}^i_{\langle R,I \rangle}$, we show that there exists a fact set $F' \in \mathcal{F}^i_{\langle R',I \rangle}$ with $F \subseteq F'$. The proof is via induction over i. For the base case $i = 0$, $\mathcal{F}^0_{\langle R,I \rangle} = \{ I \} = \mathcal{F}^0_{\langle R',I \rangle}$, so the claim holds. For the induction hypothesis, we assume that the claim holds for $i = k$. For the induction step, we show the claim for $k+1$. Consider the fact set $F \in \mathcal{F}^{k+1}_{\langle R,I \rangle}$.

1. For R', we can allow w.l.o.g. that triggers featuring rules in R' are applicable if they are active, since the application of an obsolete trigger does not yield any new facts.
2. By definition: There exists a trigger $\langle \rho, \theta \rangle$ and a fact set $F_{\text{before}} \in \mathcal{F}^k_{\langle R,I \rangle}$, such that $F \in \langle \rho, \theta \rangle (R^*_{\text{dlog}}(F_{\text{before}}))$.
3. By (2) and the induction hypothesis: There exists a fact set $F'_{\text{before}} \in \mathcal{F}^k_{\langle R',I \rangle}$ with $F_{\text{before}} \subseteq F'_{\text{before}}$.
4. We have $R_{\text{dlog}} \subseteq R'_{\text{dlog}}$.
5. By (1), (3), and (4): $R^*_{\text{dlog}}(F_{\text{before}}) \subseteq {R'}^*_{\text{dlog}}(F'_{\text{before}})$.
6. Let $\rho' \in R'$ be the rule that results from ρ by replacing disjunctions with conjunctions and let F' be the single fact set in $\langle \rho', \theta \rangle ({R'}^*_{\text{dlog}}(F'_{\text{before}}))$.
7. By (1), (5), and (6): $F \subseteq F'$.

□

Although this approach is sufficient, it marks many terminating rule sets as non-terminating.

Example 38. Consider the following rule set:

$$\begin{aligned} \mathit{Pizza}(x) &\to \mathit{LastForToday}(x) \vee \exists z.(\mathit{NextOrder}(x,z) \wedge \mathit{Pizza}(z)) \\ \mathit{NextOrder}(y,x) &\to \mathit{LastForToday}(x) \end{aligned}$$

The intuition of the first rule is that new pizzas are ordered until one is declared the last one. The second rule states that at most one follow-up order is allowed. The rule set is terminating, since the application of Datalog rules is prioritized and thus, the first rule is applied at most once. If the disjunction is replaced by a conjunction, the first rule can be applied infinitely often, which leads to a non-terminating chase sequence. ▲

A second possibility for fixing the issue in Example 36 is to create a set of fact sets that acts as a critical instance, but it is unclear if such set of fact sets even exists. For example, it is known that this approach is not possible for the restricted chase [16].

We examine a third possibility. Instead of altering the critical instance, we construct a fact set from it, without using the disjunctive skolem chase. By that, we aim to obtain a fact set that subsumes all other chase sequences according to (‡). Similar to the disjunctive skolem chase, this construction applies rules to fact sets in a chase-like procedure. The key difference lies in the condition for a trigger to be applied to a fact set. For this purpose, we introduce a new property for triggers called blocking.

3.2 Blocked Triggers

The notion of blocking for a trigger λ allows us to make statements about the applicability of λ independent of a concrete chase sequence. The idea of blocked triggers was originally introduced for RMFA [8], since the definition of the restricted chase leads to an issue which is closely related to Example 36. The main goal for blocked triggers and also the main result of this section is given in Theorem 49. Although the concrete result is quite involved, the intuition is rather simple. If a trigger λ is blocked in the context of a rule set R, then λ is never applicable to any fact set in the chase sequence of any knowledge base that features R ($\diamond$). Once the notion of blocking is established, we are able to construct a fact set according to ($\ddagger$) by exhaustively applying all active triggers that are not blocked starting on the critical instance. By doing so, we eliminate the problem that too many triggers are not applied, as seen in Example 36, while keeping the benefit of introducing only as many facts as necessary.

To obtain the desired result for blocked triggers ($\diamond$), we have to introduce a formal way of identifying facts that are involved in the derivation of new facts. In contrast to applicability, we do not consider fact sets that may be derived during the chase. Instead, solely based on the rule set, we compute a set of facts that are required to derive a certain term. We do this by partially backtracking rule applications using the function symbols that are introduced during skolemization.

For each term t, we aim to define F_t as a set of facts that are necessarily involved in the derivation of t. By ($\dagger$), we know that, for each existentially quantified variable z in a rule set R, there is a unique conjunction of atoms in the head of some rule $\mathit{rule}(z, R)$ in R. Furthermore, $\mathit{rule}(z, R)$ is the only rule that may introduce ground terms of the form $t = f^z(\vec{s})$ in the chase sequence of a knowledge base featuring R as a rule set. For brevity, we also write $\mathit{rule}(z)$ instead of $\mathit{rule}(z, R)$ since R is clear from the context.

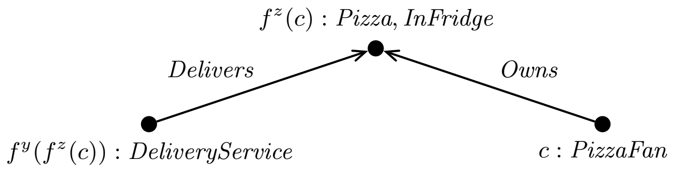
Definition 39. *Let t be a ground term defined using constants in Const and functions symbols that occur in the skolemized rules of a rule set R. We define the fact set F_t inductively. If $t \in \mathsf{Const}$, then $F_t := \emptyset$. Otherwise, t is of the form $f^z(\vec{s})$. We define the substitution θ^t on the variables in $B_{\mathit{rule}(z)}(\vec{x}, \vec{y})$ that maps the frontier $\vec{x}$ to $\vec{s}$ and $\vec{y}$ to fresh constants $\vec{c^t_y}$ that are unique for t and the variables in $\vec{y}$. We define F_t to be the smallest fact set such that*

- $B_{\mathit{rule}(z)}\theta^t \subseteq F_t$,
- $\mathit{sk}(H^k_{\mathit{rule}(z)}(\vec{x}_k, \vec{z}_k))\theta^t \subseteq F_t$ *for the* $1 \leq k \leq \mathit{branch}(\mathit{rule}(z))$ *with* $z \in \vec{z}_k$, *and*
- $F_s \subseteq F_t$ *for every term* $s \in \vec{s}$.

Using the substitution θ^t, we acquire a template trigger $\langle \mathit{rule}(z), \theta^t \rangle$ that potentially yields facts containing the term $f^z(\vec{s})$ when applied. The variables $\vec{y}$ can be mapped to arbitrary ground terms in actual trigger applications. The fresh constants $\vec{c^t_y}$ can be considered placeholders for such ground terms.

Example 40. Consider the following rules and the term $f^y(f^z(c))$. The fact set $F_{f^y(f^z(c))}$ is represented as a graph (Fig. 4):

$$\begin{aligned} \mathit{Pizza}(x) &\rightarrow \mathit{InFridge}(x) \vee \exists y.(\mathit{DeliveryService}(y) \wedge \mathit{Delivers}(y,x)) \\ \mathit{PizzaFan}(x) &\rightarrow \exists z.(\mathit{Pizza}(z) \wedge \mathit{InFridge}(z) \wedge \mathit{Owns}(x,z)) \end{aligned}$$



$f^z(c)$: *Pizza, InFridge*
Delivers
Owns
$f^y(f^z(c))$: *DeliveryService*
c : *PizzaFan*

Fig. 4. The fact set $F_{f^y(f^z(c))}$

▲

We show a result for F_t that is important for the notion of blocking to work as expected (◇). We prove our intuition of F_t, namely that the facts in F_t are indeed necessarily involved in the derivation of t. However, for the result to hold, we have to allow the remapping of the freshly introduced constants to arbitrary ground terms.

Proposition 41. *Consider a knowledge base $\mathcal{K}$ and a fact set F in the chase sequence of $\mathcal{K}$. Then, there is a mapping $\mu : \mathsf{Const} \rightarrow \mathsf{GTerm}$ that maps every constant in F to itself such that $\mu(F_t) \subseteq F$ for every term t that occurs in F.*

Proof. We show the claim by induction over the structure of t. Let μ be the identity mapping over Const. In the induction, we show how μ can be modified such that the claim holds. For the base case, we have that $t \in \mathsf{Const}$. Then, $F_t = \emptyset$ and $\emptyset \subseteq F$, so the claim holds. For the induction step, t is of the form $f^z(s_1, \ldots, s_n)$ and for the induction hypothesis, we assume that the claim holds for each term $s_1, \ldots, s_n$.

1. By definition, $F_t = B_{rule(z)}\theta^t \cup sk(H^k_{rule(z)}(\vec{x}_k, \vec{z}_k))\theta^t \cup F_{s_1} \cup \cdots \cup F_{s_n}$, for the $1 \leq k \leq branch(rule(z))$ with $z \in \vec{z}_k$.
2. Since t occurs in F, some trigger $\langle \rho, \theta \rangle$ has been applied in the chase sequence of $\mathcal{K}$ where θ is a substitution with $\vec{x}\theta^t = \vec{x}\theta$ and thus, $B_{rule(z)}\theta \cup sk(H^k_{rule(z)})\theta \subseteq F$.
3. By (2) and since θ^t maps $\vec{y}$ to fresh constants $c^{\vec{t}}_{y}$ unique for t and the variables in $\vec{y}$: We can adjust μ such that $\mu(c^{\vec{t}}_{y}) = \vec{y}\theta$. Moreover, we obtain $\theta = \mu \circ \theta^t$.
4. By (3): $\mu(B_{rule(z)}\theta^t) \cup \mu(sk(H^k_{rule_z})\theta^t) \subseteq F$.
5. By the induction hypothesis: $\mu(F_{s_1}) \cup \cdots \cup \mu(F_{s_n}) \subseteq F$.

6. By (1), (4), and (5):$\mu(F_t) \subseteq F$.

$\square$

To see how we define blocking by using F_t, consider a trigger $\langle \rho, \theta \rangle$. If some disjunct in the head of ρ only derives facts that are already in the body of ρ or in F_t for some term t in $B_\rho\theta$, then we know that λ is always obsolete if it is active. In this case we want λ to be blocked according to ($\diamond$).

Definition 42. *A trigger $\langle \rho, \theta \rangle$ is* weakly blocked *in the context of a rule set R if $\rho \notin R_{dlog}$ and for some $1 \leq k \leq branch(\rho)$, $sk(H_\rho^k)\theta \subseteq F_{\rho,\theta}$ where $F_{\rho,\theta} \coloneqq R^*_{dlog}(B_\rho\theta \cup \bigcup\{ F_t \mid t \text{ is a term in } B_\rho\theta \})$.*

Example 43. Given the rules

$$\rho_1 \coloneqq \mathit{Pizza}(x) \to \mathit{InFridge}(x) \vee \exists y.(\mathit{DeliveryService}(y) \wedge \mathit{Delivers}(y, x))$$
$$\rho_2 \coloneqq \mathit{PizzaFan}(x) \to \exists z.(\mathit{Pizza}(z) \wedge \mathit{InFridge}(z) \wedge \mathit{Owns}(x, z))$$

the trigger $\langle \rho_1, \theta \rangle$ is weakly blocked if θ is a substitution that maps x to $f^z(c)$, because $\mathit{InFridge}(f^z(c))$ is in $F_{f^z(c)}$. ▲

To construct a fact set according to (‡) that generalizes over the chase sequences of all knowledge bases that feature R, weak blocking is not sufficient. Since we are using the critical instance of R as a starting point, we have to make sure that blocking of a trigger $\langle \rho, \theta \rangle$ is independent of the concrete constants that are used in the image of θ, especially if the same constant occurs multiple times in the image of θ. Otherwise we may block too many triggers as in Example 36.

Example 44. Given the singleton rule set R containing the rule:

$$\rho \coloneqq \mathit{Pizza}(x) \wedge \mathit{IsIn}(x, y) \wedge \mathit{Oven}(y) \wedge \mathit{Hot}(y) \to \mathit{Hot}(x) \vee \mathit{NotFinished}(x)$$

Then, the trigger $\langle \rho, \theta \rangle$ is weakly blocked if θ is a substitution that maps x and y to the same ground term, e.g. $\star$, since $\mathit{Hot}(\star)$ occurs in the body of the rule. Given a substitution θ' which maps x to $\mathit{myPizza}$ and y to myOven, $\langle \rho, \theta' \rangle$ is not weakly blocked. ▲

To resolve this issue, we define the substitution θ_u for a substitution θ that names constants in the image of θ apart, such that every constant occurs at most once in the image of θ_u. Note that the following definition and the associated lemma are quite technical. For our further considerations, the intuition that constants occur at most once in the image of θ_u suffices.

Definition 45. *For a substitution θ, we define the substitution θ_u for each variable v via $\theta_u(v) \coloneqq \tau_v(\theta(v), \langle\rangle)$ if $\theta(v)$ is defined, where $\tau_v : (\mathsf{GTerm} \times \bigcup_{i \in \mathbb{N}} \mathbb{N}^i) \to \mathsf{GTerm}$ is defined via*

$$\tau_v(t, l) \coloneqq \begin{cases} c_v^l, & \text{if } t \in \mathsf{Const} \\ f(\tau_v(s_1, l \cdot \langle 1 \rangle), \ldots, \tau_v(s_n, l \cdot \langle n \rangle)), & \text{if } t \text{ is of the form } f(s_1, \ldots, s_n) \end{cases}$$

where $\cdot$ denotes tuple concatenation and $c_v^l \in \mathsf{Const}$ are fresh constants unique for l and v.

We show that θ_u is independent of the concrete constants in θ as anticipated.

Lemma 46. *Consider a substitution θ and a mapping μ over* Const*. Then, $\theta_u = \theta'_u$ for the substitution $\theta' \coloneqq \mu \circ \theta$.*

Proof. According to Definition 45, we prove the claim by showing that $\tau_v(t, l) = \tau_v(\mu(t), l)$ for every $v \in \mathsf{Vars}$ and every $l \in \bigcup_{i \in \mathbb{N}} \mathbb{N}^i$ by induction over the structure of t.

– For the base case of the induction, we show the claim for the ground term $t \in \mathsf{Const}$. We have that $\tau_v(t, l) = c_v^l = \tau_v(\mu(t), l)$.
– For the induction step we show the claim for the ground term t of the form $f(s_1, \dots, s_n)$. For the induction hypothesis we assume that $\tau_v(s, l') = \tau_v(\mu(s), l')$ holds for every $s \in \vec{s}$ and every $l' \in \bigcup_{i \in \mathbb{N}} \mathbb{N}^i$. We have that

$$\begin{aligned}\tau_v(t, l) &= f(\tau_v(s_1, l \cdot \langle 1 \rangle), \dots, \tau_v(s_n, l \cdot \langle n \rangle)) \\ &= f(\tau_v(\mu(s_1), l \cdot \langle 1 \rangle), \dots, \tau_v(\mu(s_n), l \cdot \langle n \rangle)) \\ &= \tau_v(f(\mu(s_1), \dots, \mu(s_n)), l) \\ &= \tau_v(\mu(t), l)\end{aligned}$$

□

Definition 47. *A trigger $\langle \rho, \theta \rangle$ is* blocked *in the context of a rule set R if $\langle \rho, \theta_u \rangle$ is weakly blocked in the context of R.*

An anticipated consequence of this definition is that a trigger that is blocked is also weakly blocked. Although the results seems obvious from the notions, the proof is a little more involved.

Proposition 48. *If a trigger $\lambda \coloneqq \langle \rho, \theta \rangle$ is blocked, then λ is weakly blocked.*

Proof.

1. Assume that λ is blocked.
2. By (1) and the definition of blocking: $sk(H_\rho^k)\theta_u \subseteq F_{\rho, \theta_u}$ for some $1 \le k \le$ *branch*(ρ) and there exists a mapping $\mu : \mathsf{Const} \to \mathsf{Const}$ with $\theta = \mu \circ \theta_u$, which follows from Definition 45.
3. By (2): For each $\phi \in sk(H_\rho^k)\theta$, there exists $\phi' \in sk(H_\rho^k)\theta_u$ with $\phi = \mu(\phi')$.
4. By (2) and (3):
$$\phi \in \mu(F_{\rho, \theta_u}) = \mu(R^*_{\text{dlog}}(B_\rho \theta_u \cup \bigcup \{ F_t \mid t \text{ is a term in } B_\rho \theta_u \}))$$
5. By (4) and Lemma 23: $\phi \in R^*_{\text{dlog}}(\mu(B_\rho \theta_u \cup \bigcup \{ F_t \mid t \text{ is a term in } B_\rho \theta_u \}))$.
6. By (5): $\phi \in R^*_{\text{dlog}}(B_\rho \theta \cup \mu(\bigcup \{ F_t \mid t \text{ is a term in } B_\rho \theta_u \}))$.
7. By (6): $\phi \in R^*_{\text{dlog}}(B_\rho \theta \cup \bigcup \{ F_{\mu(t)} \mid t \text{ is a term in } B_\rho \theta_u \})$ follows by induction over the structure of terms as the mapping of freshly introduced constants in F_t can be set accordingly for μ.
8. By (7): $\phi \in F_{\rho, \theta} = R^*_{\text{dlog}}(B_\rho \theta \cup \bigcup \{ F_t \mid t \text{ is a term in } B_\rho \theta \})$
9. By (3) and (8): $sk(H_\rho^k)\theta \subseteq F_{\rho, \theta}$.

10. By (9): λ is weakly blocked.

$\square$

We formally show the main result for blocking and prove that the intuition ($\diamond$) indeed holds for the presented definition.

Theorem 49. *Consider a knowledge base $\mathcal{K} \coloneqq \langle R, I \rangle$, a fact set F that occurs in the chase sequence of $\mathcal{K}$, and some trigger $\lambda \coloneqq \langle \rho, \theta \rangle$ with $\rho \in R$. If λ is blocked, then λ is not applicable to $R^*_{dlog}(F)$.*

Proof. We assume that λ is blocked and hence, $\rho \notin R_{\text{dlog}}$. Suppose for a contradiction that λ is applicable to $R^*_{\text{dlog}}(F)$. By Proposition 48, we obtain that λ is weakly blocked and hence, $sk(H^k_\rho)\theta \subseteq F_{\rho,\theta}$ for some $1 \leq k \leq branch(\rho)$. We show that $sk(H^k_\rho)\theta \subseteq R^*_{\text{dlog}}(F)$.

1. By definition: $F_{\rho,\theta} = R^*_{\text{dlog}}(B_\rho\theta \cup \bigcup\{\, F_t \mid t \text{ is a term in } B_\rho\theta \,\})$.
2. Since λ is applicable to $R^*_{\text{dlog}}(F)$: $B_\rho\theta \subseteq R^*_{\text{dlog}}(F)$.
3. By (2) and Proposition 41: There exists a mapping $\mu : \mathsf{Const} \to \mathsf{GTerm}$ that maps every constant in $B_\rho\theta$ to itself, such that $\mu(F_t) \subseteq R^*_{\text{dlog}}(F)$ for each term t that occurs in $B_\rho\theta$.
4. By (2) and (3): $\mu(B_\rho\theta \cup \bigcup\{\, F_t \mid t \text{ is a term in } B_\rho\theta \,\}) \subseteq R^*_{\text{dlog}}(F)$.
5. By (4): $R^*_{\text{dlog}}(\mu(B_\rho\theta \cup \bigcup\{\, F_t \mid t \text{ is a term in } B_\rho\theta \,\})) \subseteq R^*_{\text{dlog}}(R^*_{\text{dlog}}(F))$.
6. By (5): $\mu(R^*_{\text{dlog}}(B_\rho\theta \cup \bigcup\{\, F_t \mid t \text{ is a term in } B_\rho\theta \,\})) \subseteq R^*_{\text{dlog}}(F)$ follows from Lemma 23.
7. By (1) and (6): $\mu(F_{\rho,\theta}) \subseteq R^*_{\text{dlog}}(F)$.
8. Since λ is weakly blocked: $\mu(sk(H^k_\rho)\theta) \subseteq \mu(F_{\rho,\theta})$.
9. By (3): $\mu(B_\rho\theta) = B_\rho\theta$ and thus, $\mu(sk(H^k_\rho)\theta) = sk(H^k_\rho)\theta$.
10. By (7), (8), and (9): $sk(H^k_\rho)\theta \subseteq R^*_{\text{dlog}}(F)$.

We find that λ is not applicable to $R^*_{\text{dlog}}(F)$. ↯ $\square$

3.3 Disjunctive Model Faithful Acyclicity (DMFA)

We define the construction of a fact set according to (‡) using the newly introduced notion of blocked triggers.

Definition 50. *For a rule set R, we define $\mathsf{DMFA}(R)$ to be the smallest fact set such that $I^\star_R \subseteq \mathsf{DMFA}(R)$ and, for every trigger $\langle \rho, \theta \rangle$ with $\rho \in R$ that is active w.r.t. $\mathsf{DMFA}(R)$ and not blocked, we have $sk(H^i_\rho)\theta \subseteq \mathsf{DMFA}(R)$ for all $1 \leq i \leq branch(\rho)$.*

Note that this corresponds to a non-disjunctive version of the skolem chase where disjunctions are treated as conjunctions and the conditions for the application of triggers are adjusted. We prove that $\mathsf{DMFA}(R)$ subsumes the chase sequences of all other knowledge bases featuring R, according to (‡). Formally, this is described in the following lemma.

Lemma 51. *Let $\sigma : \mathsf{Const} \to \mathsf{Const}$ with $\sigma(c) := \star$ for all $c \in \mathsf{Const}$ and let $\mathcal{K} := \langle R, I \rangle$ be a knowledge base. For the chase sequence of $\mathcal{K}$, we obtain $\sigma(\bigcup \mathcal{F}^i_{\mathcal{K}}) \subseteq \mathsf{DMFA}(R)$ for every $i \in \mathbb{N}$.*

To simplify the proof, we extend the chase sequence from Definition 24 to include individual steps for the applications of Datalog rules. We define the *extended chase sequence* for $\mathcal{K}$ as the sequence of sets of fact sets $\mathcal{E}^0_{\mathcal{K}}, \mathcal{E}^1_{\mathcal{K}}, \ldots$ inductively via $\mathcal{E}^0_{\mathcal{K}} := \{ I \}$ and for all $i > 0$:

$$\mathcal{E}^i_{\mathcal{K}} := \begin{cases} R(\mathcal{E}^{i-1}_{\mathcal{K}}), & \text{if } R_{\text{dlog}}(\mathcal{E}^{i-1}_{\mathcal{K}}) = \mathcal{E}^{i-1}_{\mathcal{K}} \\ R_{\text{dlog}}(\mathcal{E}^{i-1}_{\mathcal{K}}), & \text{otherwise} \end{cases}$$

According to Remark 21, the extended chase sequence of $\mathcal{K}$ subsumes the chase sequence of $\mathcal{K}$.

Proof (of Lemma 51). We show that $\sigma(\bigcup \mathcal{E}^i_{\mathcal{K}}) \subseteq \mathsf{DMFA}(R)$ for every $i \in \mathbb{N}$ by induction over i. For $i = 0$, we show that $\sigma(\phi) \in \mathsf{DMFA}(R)$ for all $\phi \in \bigcup \mathcal{E}^0_{\mathcal{K}}$.

1. Since $\mathcal{E}^0_{\mathcal{K}} = \{ I \}$, every $\phi \in I$ is of the form $P(c_1, \ldots, c_n)$ where $c_1, \ldots, c_n$ are constants.
2. By (1) and the definition of σ:
$$\sigma(\phi) = \sigma(P(c_1, \ldots, c_n)) = P(\sigma(c_1), \ldots, \sigma(c_n)) = P(\underbrace{\star, \ldots, \star}_{n \text{ times}}) \in I^\star_R$$
3. By (2) and the definition of $\mathsf{DMFA}(R)$: $\sigma(\phi) \in \mathsf{DMFA}(R)$

For the induction hypothesis, we assume that $\sigma(\phi') \in \mathsf{DMFA}(R)$ for all $\phi' \in \bigcup \mathcal{E}^{k-1}_{\mathcal{K}}$ where $k \geq 1$. For the induction step, we show that $\sigma(\phi) \in \mathsf{DMFA}(R)$ for all $\phi \in \bigcup \mathcal{E}^k_{\mathcal{K}}$.

1. Let $\phi \in \bigcup \mathcal{E}^k_{\mathcal{K}}$ with $\phi \notin \bigcup \mathcal{E}^{k-1}_{\mathcal{K}}$ as otherwise the claim follows directly from the induction hypothesis.
2. By (1): There exists a fact set F in $\mathcal{E}^{k-1}_{\mathcal{K}}$, with $\phi \notin F$. Moreover, there exists a trigger $\lambda := \langle \rho, \theta \rangle$ with $\rho \in R$ that is applicable to F with $\phi \in \lambda(F)$. In particular, if $R_{\text{dlog}}(F) \neq \{ F \}$, then $\rho \in R_{\text{dlog}}$.
3. By (2): $B_\rho\theta \subseteq F$ and ϕ is in $sk(H^i_\rho)\theta$ for some $1 \leq i \leq \mathit{branch}(\rho)$.
4. By (3) and the induction hypothesis: $B_\rho\theta' \subseteq \mathsf{DMFA}(R)$ for $\theta' := (\sigma \circ \theta)$.
5. By (2): $\lambda = \langle \rho, \theta \rangle$ is not blocked since either $\rho \in R_{\text{dlog}}$ or λ is applicable to $R^*_{\text{dlog}}(F) = F$. In the latter case, the contrapositive of Theorem 49 implies that λ is not blocked.
6. By (4), (5), and Lemma 46: $\langle \rho, \theta' \rangle$ is not blocked.
7. By (4) and (6): $\langle \rho, \theta' \rangle$ is active w.r.t. $\mathsf{DMFA}(R)$ and not blocked.
8. By (7) and the definition of $\mathsf{DMFA}(R)$: $sk(H^i_\rho)\theta' \subseteq \mathsf{DMFA}(R)$.
9. By (3) and (8): $\sigma(\phi) \in \mathsf{DMFA}(R)$.

□

We use $\mathsf{DMFA}(R)$ to define a sufficient condition for the termination of R. For this sake, regard the structure of facts in the chase sequence of non-terminating knowledge bases. We find that some of these facts contain cyclic terms, because there is only a finite amount of acyclic terms.

Lemma 52. *Let* $\mathsf{p}, \mathsf{c}, \mathsf{f}$ *be the numbers of predicate symbols, constants, and function symbols and* l, a *the maximum arities of function symbols and predicate symbols that occur in a given skolemized rule set* R*, respectively. The number of facts that do not contain a cyclic term and can be constructed from predicate symbols, constants, and function symbols from* R *is bound by* $\mathsf{p} \cdot ((\mathsf{f} \cdot \mathsf{c})^{\mathsf{f} \cdot \mathsf{l}^{\mathsf{f}}+1})^{\mathsf{a}}$.

Proof. 1. The maximum nesting depth of a non cyclic term is f.
2. The number of terms with a depth of 0 is c.
3. The number of terms with a depth of $i \in [1, \mathsf{f}]$ is bound by

$$\underbrace{\mathsf{f} \cdot (\ldots \mathsf{f} \cdot (\mathsf{f}}_{i \text{ times}} \cdot \mathsf{c} \underbrace{{}^{\mathsf{l}})^{\mathsf{l}} \ldots)^{\mathsf{l}}}_{i \text{ times}} \leq \underbrace{\mathsf{f}^{\mathsf{l}^i} \cdot (\ldots \mathsf{f}^{\mathsf{l}^2} \cdot (\mathsf{f}^{\mathsf{l}}}_{i \text{ times}} \cdot \mathsf{c} \underbrace{{}^{\mathsf{l}})^{\mathsf{l}} \ldots)^{\mathsf{l}}}_{i \text{ times}} = (\mathsf{f}^i \cdot \mathsf{c})^{\mathsf{l}^i} \leq (\mathsf{f} \cdot \mathsf{c})^{i \cdot \mathsf{l}^i}$$

4. By (1), (2), and (3): The total number of terms is bound by

$$\mathsf{c} + \sum_{i=1}^{\mathsf{f}} (\mathsf{f} \cdot \mathsf{c})^{i \cdot \mathsf{l}^i} \leq \mathsf{f} \cdot (\mathsf{f} \cdot \mathsf{c})^{\mathsf{f} \cdot \mathsf{l}^{\mathsf{f}}} \leq (\mathsf{f} \cdot \mathsf{c})^{\mathsf{f} \cdot \mathsf{l}^{\mathsf{f}}+1}$$

5. By (4): The total number of facts is bound by $\mathsf{p} \cdot ((\mathsf{f} \cdot \mathsf{c})^{\mathsf{f} \cdot \mathsf{l}^{\mathsf{f}}+1})^{\mathsf{a}}$.

□

Note that the bound is finite, because $\mathsf{p}, \mathsf{c}, \mathsf{f}, \mathsf{l}$, and a are finite. Hence, the absence of cyclic terms in the chase sequence of a knowledge base $\mathcal{K}$ is sufficient for $\mathcal{K}$ to be terminating. We use this property on the constructed fact set $\mathsf{DMFA}(R)$ to define DMFA. The absence of cyclic terms is a good measure because cyclic terms indicate that a single rule is applied multiple times. This is a strong hint towards non-termination.

Definition 53. *A rule set* R *is* DMFA *if* $\mathsf{DMFA}(R)$ *does not contain a cyclic term.*

We prove that DMFA is indeed a sufficient condition for rule set termination.

Theorem 54. *If a rule set* R *is DMFA, then* R *is terminating.*

Proof. We show the contrapositive; that is, we show that R is not DMFA if R is not terminating.

1. Assume that R is not terminating.
2. By (1): There exists an instance I such that $\langle R, I \rangle$ is not terminating.
3. By (2): There are infinitely many facts in the chase sequence of $\langle R, I \rangle$.
4. By (3) and Lemma 52: The chase sequence of $\langle R, I \rangle$ contains at least one cyclic term t.

5. By (4) and Lemma 51: The cyclic term $\sigma(t)$ is in $\mathsf{DMFA}(R)$.
6. By (5): R is not DMFA.

$\square$

Note that there are rule sets that are not DMFA but still terminating.

Example 55. Consider the singleton rule set R:

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

The chase of $\langle R, I_R^\star = \{ P(\star,\star), Q(\star) \} \rangle$ is (Fig. 5):

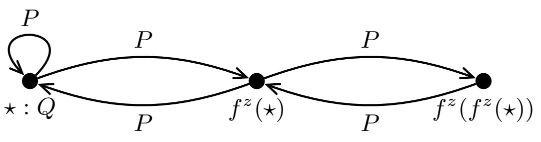


Fig. 5. Terminating rule set that is not DMFA

In particular we obtain $Ch(\langle R, I_R^\star \rangle) = \mathsf{DMFA}(R)$. As $Ch(\langle R, I_R^\star \rangle)$ is finite, we find that R is terminating but not DMFA, since $\mathsf{DMFA}(R)$ contains the cyclic term $f^z(f^z(\star))$. ▲

3.4 Relation of DMFA to MFA and RMFA

We show the relation of DMFA to the existing acyclicity notions MFA [9] and RMFA [8]. Although MFA is originally only defined for rule sets without disjunctions [9], we can also apply it to disjunctive rule sets by treating disjunctions as conjunctions and using Proposition 37.

We briefly introduce definitions of MFA and RMFA mostly based on our previously defined notions and compare them to DMFA. We do not use the original definitions for MFA and RMFA. Instead, we define both notions in a similar way to DMFA for easier comparison. Still, the definitions are mostly equivalent. The only difference is that for MFA we provide a trivial extension that implicitly treats disjunctions as conjunctions. This extension of MFA can therefore also be applied to disjunctive rule sets, which is not the case for the original MFA definition.

Definition 56. *For a rule set R, we define $\mathsf{MFA}(R)$ to be the smallest fact set such that $I_R^\star \subseteq \mathsf{MFA}(R)$ and, for every trigger $\langle \rho, \theta \rangle$ with $\rho \in R$ that is active w.r.t. $\mathsf{MFA}(R)$, we have $sk(H_\rho^i)\theta \subseteq \mathsf{MFA}(R)$ for all $1 \le i \le branch(\rho)$. The rule set R is MFA if $\mathsf{MFA}(R)$ does not contain a cyclic term.*

For rule sets without disjunctions, MFA and DMFA exactly coincide.

Proposition 57. *A rule set R without disjunctions is MFA iff it is DMFA.*

Proof (Sketch). According to Definitions 50 and 56, $\mathsf{MFA}(R)$ and $\mathsf{DMFA}(R)$ only differ in the sense that $\mathsf{DMFA}(R)$ disallows blocked triggers whereas $\mathsf{MFA}(R)$ allows all active triggers including blocked ones. We observe that triggers featuring rules without disjunctions can never be blocked because triggers featuring Datalog rules are not blocked by definition and for triggers featuring non-disjunctive rules with existential quantifiers, the newly introduced skolem terms cannot occur in the rule body. Hence, we obtain $\mathsf{MFA}(R) = \mathsf{DMFA}(R)$. Thus, the disjunction-free rule set R is MFA iff it is DMFA. □

In general, DMFA captures more rule sets than MFA.

Theorem 58. *The rule sets that are* MFA *form a strict subset of the rule sets that are* DMFA.

Proof (Sketch). First, we show that a rule set R is DMFA if R is MFA. Since the computation of $\mathsf{DMFA}(R)$ does not allow blocked triggers, we obtain $\mathsf{DMFA}(R) \subseteq \mathsf{MFA}(R)$. Thus, if $\mathsf{MFA}(R)$ does not contain a cyclic term, then $\mathsf{DMFA}(R)$ does not contain a cyclic term.

Second, we show that there exists a rule set that is DMFA but not MFA. Consider the rule set R' from Example 38:

$$\begin{aligned} \mathit{Pizza}(x) &\to \mathit{LastForToday}(x) \vee \exists z.(\mathit{NextOrder}(x,z) \wedge \mathit{Pizza}(z)) \\ \mathit{NextOrder}(y,x) &\to \mathit{LastForToday}(x) \end{aligned}$$

As triggers featuring the first rule are blocked for the substitution that maps x to $f^z(\star)$, we obtain

$$\begin{aligned} \mathsf{DMFA}(R') = \{\, & \mathit{NextOrder}(\star,\star), \mathit{Pizza}(\star), \mathit{LastForToday}(\star), \\ & \mathit{NextOrder}(\star, f^z(\star)), \mathit{Pizza}(f^z(\star)), \mathit{LastForToday}(f^z(\star)) \,\} \end{aligned}$$

and thus, R' is DMFA. However, R' is not MFA, since $\mathsf{MFA}(R')$ contains infinitely many facts with at least one cyclic term, which is of the form $f^z(\ldots f^z(\star)\ldots)$. □

Te define RMFA, we have to use another notion of blocking.

Definition 59. *A trigger* $\langle \rho, \theta \rangle$ *is* r-blocked *in the context of a rule set* R *if* $\rho \notin R_{dlog}$ *and we have that* $F_{\rho,\theta_u} \models \exists \vec{z}_k . H^k_\rho(\vec{x}_k, \vec{z}_k)\theta_u$ *for some* $1 \leq k \leq \mathit{branch}(\rho)$.

The key difference to the notion of blocked triggers is that the r-blocked check considers an entailment relation whereas the blocked check considers a subset relation. Indeed, blocking is subsumed by r-blocking.

Lemma 60. *If a trigger* $\langle \rho, \theta \rangle$ *is blocked, then* $\langle \rho, \theta \rangle$ *is r-blocked.*

Proof. Since $\langle \rho, \theta \rangle$ is blocked, we have that $sk(H^k_\rho)\theta_u \subseteq F_{\rho,\theta_u}$ for some $1 \leq k \leq \mathit{branch}(\rho)$. Therefore, $F_{\rho,\theta_u} \models \exists \vec{z}_k . H^k_\rho \theta_u$ by assigning each variable $z \in \vec{z}_k$ the value $f^z(\vec{x})\theta_u$ where $\vec{x}$ is the frontier of ρ. Hence, $\langle \rho, \theta \rangle$ is r-blocked. □

Definition 61. *For a rule set R, we define* $\mathsf{RMFA}(R)$ *to be the smallest fact set such that* $I_R^\star \subseteq \mathsf{RMFA}(R)$ *and, for every trigger* $\langle \rho, \theta \rangle$ *with* $\rho \in R$ *that is active w.r.t.* $\mathsf{RMFA}(R)$ *and not r-blocked, we have* $sk(H_\rho^i)\theta \subseteq \mathsf{RMFA}(R)$ *for all* $1 \leq i \leq branch(\rho)$*. The rule set R is RMFA if* $\mathsf{RMFA}(R)$ *does not contain a cyclic term.*

RMFA allows the characterization of rule sets that terminate w.r.t. the restricted chase but not necessarily w.r.t. the disjunctive skolem chase. DMFA must not mark such rule sets as terminating as DMFA is a sufficient condition for rule set termination w.r.t. the disjunctive skolem chase. This is a strong argument why DMFA cannot capture RMFA. In fact, no acyclicity notion for the disjunctive skolem chase is able to capture RMFA for this reason. In the following, we also provide an example for a rule set that is RMFA and even terminating w.r.t. the disjunctive skolem chase but not DMFA.

Theorem 62. *The rule sets that are DMFA form a strict subset of the rule sets that are RMFA. This even holds if we only consider rule sets that are terminating w.r.t. the disjunctive skolem chase.*

Proof (Sketch). First, we show that a rule set R is RMFA if R is DMFA by showing the contrapositive; that is, we show that R is not DMFA if R is not RMFA. According to Definitions 50 and 61, $\mathsf{DMFA}(R)$ and $\mathsf{RMFA}(R)$ only differ in the sense that $\mathsf{DMFA}(R)$ disallows blocked triggers whereas $\mathsf{RMFA}(R)$ disallows r-blocked triggers. Since every trigger that is blocked is also r-blocked by Lemma 60, we obtain $\mathsf{RMFA}(R) \subseteq \mathsf{DMFA}(R)$. Thus, if $\mathsf{RMFA}(R)$ contains a cyclic term, then $\mathsf{DMFA}(R)$ also contains a cyclic term. Therefore, if R is not RMFA, then R is not DMFA.

Second, we show that there exists a rule set that is RMFA and terminating w.r.t. the disjunctive skolem chase but not DMFA. Consider the singleton rule set R':

$$P(y, x) \wedge Q(y) \rightarrow \exists z.(P(x, z) \wedge P(z, x))$$

We know from Example 55 that R' is terminating w.r.t. the disjunctive skolem chase but not DMFA. Still, R' is RMFA because ρ can only applied once using the trigger $\langle \rho, \theta \rangle$ where θ maps x and y to $f^z(\star)$. We obtain the new facts $P(\star, f^z(\star))$ and $P(f^z(\star), \star)$. However, the triggers with the substitutions that map y to $\star$ and x to $f^z(\star)$ or vice versa are both r-blocked. $\square$

Regarding MFA, DMFA, and RMFA, we obtain the following picture. Note that every represented subset is in fact non-empty.

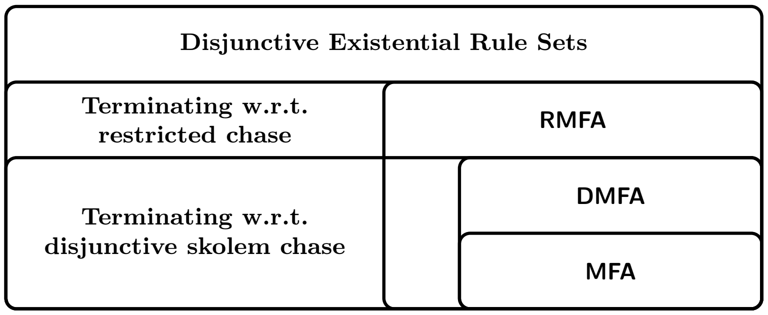

Disjunctive Existential Rule Sets
Terminating w.r.t. restricted chase
RMFA
Terminating w.r.t. disjunctive skolem chase
DMFA
MFA

Fig. 6. Hierarchy of acyclicity notions

3.5 Complexity Results for DMFA Rulesets

From the comparison to MFA and RMFA, we see that DMFA exactly lies in-between. Checking if a rule set is MFA or RMFA is each 2ExpTime-complete. Therefore, we expect that checking if a rule set is DMFA is also 2ExpTime-complete. At first, we prove that this is indeed the case. Beyond that, we study complexity bounds of the disjunctive skolem chase and BCQ entailment for rule sets that are DMFA.

Theorem 63. *Checking if a rule set R is DMFA is in* 2ExpTime.

Proof. We sketch an algorithm for computing DMFA(R) in 2ExpTime. We compute DMFA(R) iteratively in a chase-like procedure and check for a cyclic term in each step. If we encounter a cyclic term, we reject. If we do not encounter a cyclic term and no new facts are derived, we accept. The number of steps in the chase-like procedure is doubly exponentially bounded, since there are only doubly exponentially many facts without a cyclic term, according to Lemma 52. We show that the computation of a single step including the cyclicity check for the newly introduced terms is in 2ExpTime.

1. The number of rules is linear.
2. The number of variables in a single rule is linear.
3. The number of acyclic terms is doubly exponentially bounded by the proof of Lemma 52.
4. By (2) and (3): For a single rule ρ, the number of substitutions that are defined exactly on the universally quantified variables in ρ and do not map variables to cyclic terms is doubly exponentially bounded.
5. By (1) and (4): The number of triggers that do not map variables to cyclic terms is doubly exponentially bounded.
6. The check if a single trigger is active is in 2ExpTime, since the number of facts that have been derived before the current step is doubly exponentially bounded.

7. The check if a single trigger $\langle \rho, \theta \rangle$ is blocked is in ExpTime, since the number of terms in the image of θ is linear and the number of subterms for each term in the image of θ is exponentially bounded. Hence, for each term t in the image of θ, the size of F_t is exponentially bounded. The application of R^*_{dlog} within the blocked check is also in ExpTime.
8. By (6) and (7): The check if a single trigger is active and not blocked is in 2ExpTime and thus, the application of the trigger is computable in 2ExpTime.
9. The number of new terms introduced by the application of a trigger is linear and the cyclicity check for each of those terms is in ExpTime since the number of subterms of a term is exponentially bounded.
10. By (5), (8), and (9): A single step of the chase-like procedure is computable in 2ExpTime.

Since each of the doubly exponentially many steps can be computed in 2ExpTime, the overall chase-like procedure can be computed in 2ExpTime. □

Theorem 64. *Checking if a rule set R is DMFA is 2ExpTime-hard.*

Note that 2ExpTime-hardness follows from Proposition 57 and the fact that checking MFA for rule sets without disjunctions is 2ExpTime-complete (see Lemma 7 and Theorem 8 in [9]). To recall the argument, we show hardness in a similar way to the proof that is used for MFA. For this purpose, we have to introduce another acyclicity notion called *weak acyclicity* [11].

Definition 65. *For a rule set R, construct the* dependency graph $\mathfrak{G}$ *of R as follows. The nodes of $\mathfrak{G}$ are defined as the set of the* predicate positions *of R: $\{ \langle P, i \rangle \mid P \text{ is a predicate symbol in } R, 1 \leq i \leq ar(P) \}$. For every rule ρ in R and every position $\langle P, i \rangle$ in the head of ρ that contains some variable x:*

- *If x is universally quantified, add an edge $\langle Q, j \rangle \rightarrow \langle P, i \rangle$ to $\mathfrak{G}$ for all positions $\langle Q, j \rangle$ in the body of ρ that feature x.*
- *If x is existentially quantified, add a* special edge $\langle Q, j \rangle \xrightarrow{*} \langle P, i \rangle$ *to $\mathfrak{G}$ for all positions $\langle Q, j \rangle$ in the body of ρ.*

The rule set R is weakly acyclic *if $\mathfrak{G}$ does not contain a cycle that involves a special edge.*

The main result we need for weak acyclicity is that it is subsumed by MFA and hence also by DMFA (Theorem 58).

Proposition 66. *The rule sets that are weakly acyclic form a subset of the rule sets that are MFA.*

Note that the subset relation is in fact strict but we only require the subset relation for our matter.

Proof.

1. Suppose for a contradiction, that there exists a weakly acyclic rule set R that is not MFA.
2. By (1): $\mathsf{MFA}(R)$ contains a cyclic term featuring nested occurrences of some function symbol f^z.
3. By (†): There is a unique rule featuring z in its head at some position $\langle P, i\rangle$.
4. By (2) and (3): There is a cycle in the dependency graph of R through $\langle P, i\rangle$ that contains a special edge.
5. By (4): R is not weakly acyclic. $\lightning$

$\square$

We can now prove 2ExpTime-hardness for checking DMFA.

Proof (of Theorem 64). We reduce the 2ExpTime-complete problem of BCQ-entailment for a BCQ $\sigma := \exists \vec{z}.\varphi(\vec{z})$ and a knowledge base $\langle R, I\rangle$ featuring a weakly acyclic rule set without disjunctions [7] to the check if R_{DMFA} is DMFA where R_{DMFA} is constructed from $\langle R, I\rangle$.

1. Let $R' := R \cup \{\, \varphi(\vec{z}) \to B) \,\}$ where B is a fresh nullary predicate symbol.
2. By (1): σ is entailed by $\langle R, I\rangle$ iff B is entailed by $\langle R', I\rangle$.
3. For each constant c that occurs in I, we introduce a fresh variable v_c. Let $\vec{v}_c$ be a list of those variables and let A be a fresh nullary predicate symbol. We set:
$$R'' := R' \cup \{\, A \to \exists \vec{v}_c. \bigwedge\nolimits_{P(c_1,\ldots,c_m)\in I} P(v_{c_1}, \ldots, v_{c_m}) \,\}$$
4. By (3): B is entailed by $\langle R', I\rangle$ iff B is entailed by $\langle R'', \{\, A \,\}\rangle$.
5. For every n-ary predicate symbol P, we introduce a fresh $n+1$-ary predicate symbol $\hat{P}$. We define
$$R''' := \bigcup \{\, \hat{\varphi}(\vec{x}, \vec{y}, w) \to \exists \vec{z}.\hat{\psi}(\vec{x}, \vec{z}, w) \mid \varphi(\vec{x}, \vec{y}) \to \exists \vec{z}.\psi(\vec{x}, \vec{z}) \in R'' \,\}$$
where w is a fresh variable and for every conjunction of atoms φ, we set $\hat{\varphi} := \bigwedge_{P(\vec{x})\in\varphi} \hat{P}(\vec{x}, w)$.
6. By (5): B is entailed by $\langle R'', \{\, A \,\}\rangle$ iff $\hat{B}(a)$ is entailed by $\langle R''', \{\, \hat{A}(a) \,\}\rangle$ where a is a fresh constant. This follows by induction over both chase sequences.
7. By (1), (3), (5), and since R is weakly acyclic: R''' is weakly acyclic.
8. Let Q be a fresh binary predicate symbol and let $R_{\mathsf{DMFA}} := R''' \cup \{\, \rho_{\text{cycle}} \,\}$, where
$$\rho_{\text{cycle}} := \hat{B}(x) \wedge Q(y, x) \to \exists z.(Q(x, z) \wedge \hat{A}(z))$$
9. By (7) and (8): We claim that R_{DMFA} is not DMFA iff $\hat{B}(a)$ is entailed by $\langle R''', \{\, \hat{A}(a) \,\}\rangle$. Note that $\{\, \hat{B}(\star), Q(\star, \star) \,\} \subseteq I^{\star}_{R_{\mathsf{DMFA}}}$ and thus, we have that $\{\, \hat{B}(\star), Q(\star, \star), Q(\star, f^z(\star)), \hat{A}(f^z(\star)) \,\} \subseteq \mathsf{DMFA}(R_{\mathsf{DMFA}})$. We now sketch a proof for both directions of the claim separately.
   - If $\hat{B}(a)$ is not entailed by $\langle R''', \{\, \hat{A}(a) \,\}\rangle$, then no cyclic term is introduced by ρ_{cycle} since each trigger of the form $\langle \rho_{\text{cycle}}, \theta\rangle$ is only active (and not blocked) if θ maps x and y to $\star$. Hence, R_{DMFA} is DMFA if R''' is DMFA. Since R''' is weakly acylic, R''' is indeed DMFA by Theorem 58 and Proposition 66.

– Assume that $\hat{B}(a)$ is entailed by $\langle R''', \{ \hat{A}(a) \} \rangle$. If for some substitution θ, $\langle \rho_{\text{cycle}}, \theta \rangle$ is active and not blocked, then $\langle \rho_{\text{cycle}}, \theta' \rangle$ is active and not blocked at some point, where θ' is defined via $\theta'(x) := f^z(\theta(x))$ and $\theta'(y) := \theta(x)$. Since the trigger $\langle \rho_{\text{cycle}}, \theta^\star \rangle$, where $\theta^\star$ maps x and y to $\star$, is active and not blocked, $\mathsf{DMFA}(R_{\mathsf{DMFA}})$ contains a cyclic term of the form $f^z(\dots f^z(\star)\dots)$ and thus, R_{DMFA} is not DMFA.

10. By (2), (4), (6), and (9): σ is entailed by $\langle R, I \rangle$ iff R_{DMFA} is not DMFA.

Since the construction of R_{DMFA} is possible in polynomial time, 2ExpTime-hardness follows. □

From Theorems 63 and 64, we immediately obtain the following result.

Corollary 67. *Checking if a rule set R is DMFA is* 2ExpTime*-complete.*

The complexity for the DMFA check drops if we restrict rule sets to only contain at most one frontier variable per rule.

Theorem 68. *Checking DMFA for rule set R where every non-datalog rule in R has at most one frontier variable is* ExpTime*-complete.*

Proof (Sketch). We show membership as follows: If all non-datalog rules in R feature at most one frontier variable, then every function symbol used in the skolemization of R has arity at most 1. In this case, according to the proof of Lemma 52, the number of facts that do not contain a cyclic term and can be constructed from predicate symbols, constants and function symbols from R is (singly) exponentially bounded. Hence, every 2ExpTime bound in the proof of Theorem 63 becomes an ExpTime bound and thus, checking DMFA for R is in ExpTime in this case.

Hardness follows from the fact that the MFA check for a rule set without disjunctions where every non-datalog rule has at most one frontier variable is ExpTime-hard (see Theorem 10 in [9]). Proposition 57 yields the claim. □

Next, we consider the computation of the disjunctive skolem chase for rule sets that are DMFA.

Theorem 69. *Let R be a rule set that is DMFA and let $\mathcal{K}$ be a knowledge base featuring R. The chase of $\mathcal{K}$ can be computed in* 3ExpTime.

Proof. By Lemma 51, the chase sequence of $\mathcal{K}$ does not contain a cyclic term. The number of steps in the chase sequence of $\mathcal{K}$ is doubly exponentially bounded, since there are only doubly exponentially many facts without a cyclic term, according to Lemma 52 and every step introduces at least one new fact unless a fixed point is reached. We show that the computation of a single step is in 3ExpTime.

1. The number of rules is linear.
2. The number of variables in a single rule is linear.

3. The number of acyclic terms is doubly exponentially bounded by the proof of Lemma 52.
4. By (2) and (3): For a single rule ρ, the number of substitutions that are defined exactly on the universally quantified variables in ρ and do not map variables to cyclic terms is doubly exponentially bounded.
5. By (1) and (4): The number of triggers that do not map variables to cyclic terms is doubly exponentially bounded.
6. The check if a single trigger is active and not obsolete w.r.t. a single fact set is in 2ExpTime, since the number of facts that have been derived before the current step is doubly exponentially bounded.
7. The number of different fact sets is the size of the power set of all facts which is triply exponentially bounded.
8. By (5), (6), and (7): A single step can be computed in 3ExpTime.

Since each of the doubly exponentially many steps can be computed in 3ExpTime, the overall chase-like procedure can be computed in 3ExpTime. □

In the following we consider BCQ entailment. It is possible to compute the chase of $\mathcal{K}$ in 3ExpTime and check the entailment of a BCQ for each of the fact sets, which is in 3ExpTime in total, but we can get a tighter bound.

Theorem 70. *Let R be a rule set that is DMFA and let $\mathcal{K}$ be a knowledge base featuring R. BCQ entailment for $\mathcal{K}$ is in coN2ExpTime.*

Proof. We modify the application of triggers in a way, that we guess one of the resulting fact sets nondeterministically. The upper bound of 3ExpTime in Theorem 69 results only from the fact that there are possibly triply exponentially many fact sets to consider. By picking only one fact set non-deterministically in each step, we obtain an upper bound of N2ExpTime.

1. A single fact set in $Ch(\mathcal{K})$ can be computed in N2ExpTime.
2. A BCQ $\exists\vec{z}.\varphi(\vec{z})$ is entailed by $\mathcal{K}$ according to Corollary 26 if for each $F \in Ch(\mathcal{K})$ there exists a substitution θ with $\varphi\theta \subseteq F$.
3. By (2): a BCQ $\exists\vec{z}.\varphi(\vec{z})$ is not entailed by $\mathcal{K}$ if there exists a fact set F in $Ch(\mathcal{K})$ such that $\varphi\theta \nsubseteq F$ for all substitutions θ.
4. By (1) and (3): we can check if a BCQ is not entailed by $\mathcal{K}$ in N2ExpTime.
5. By (4): BCQ entailment for $\mathcal{K}$ is in coN2ExpTime.

□

Theorem 71. *Let R be a rule set that is DMFA and let $\mathcal{K}$ be a knowledge base featuring R. BCQ entailment for $\mathcal{K}$ is coN2ExpTime-hard.*

Proof (Sketch). As for RMFA (see Theorem 7 in [8]) we can modify the reduction from the word problem of 2ExpTime-bounded turing machines to BCQ entailment for weakly acyclic rule sets without disjunctions [7], such that non-determinism is represented by allowing disjunctions in rules. Since every rule set that is weakly acyclic is also DMFA by Theorem 58 and Proposition 66, the claim follows. □

From Theorems 70 and 71 we immediately obtain the following result.

Corollary 72. *Let R be a rule set that is* DMFA *and let $\mathcal{K}$ be a knowledge base featuring R. BCQ entailment for $\mathcal{K}$ is* coN2ExpTime*-complete.*

4 Conclusion

We have shown the construction of DMFA as a novel acyclicity notion tailored towards the disjunctive skolem chase. Based on the idea of r-blocked triggers in [8], our main contribution is the notion of blocked triggers that generalizes trigger obsoleteness for the disjunctive skolem chase by considering solely rule sets and not concrete fact sets. We have proven that DMFA is able to mark more rule sets as terminating than a trivial disjunctive version of MFA. We have also seen that RMFA subsumes DMFA but we expect DMFA not to be able to capture RMFA because RMFA guarantees termination for the restricted chase, which generally terminates in more cases than the disjunctive skolem chase. However, Theorem 62 provides an example for a rule set that is RMFA but not DMFA and does in fact terminate with respect to the disjunctive skolem chase. We have shown that the complexity results for checking DMFA and reasoning with rule sets that are DMFA are essentially the same as for MFA and RMFA.

Our work enables and encourages further research towards acyclicity and cyclicity notions for the disjunctive skolem chase. Up to this point our contributions are purely theoretical. In upcoming work, we will provide a reference implementation of DMFA and evaluate how DMFA performs in practice compared to MFA and RMFA. As for [8], this evaluation will be based on real world description logic ontologies that are transformed into disjunctive existential rules. We also plan to develop and evaluate cyclicity notions, i.e. sufficient conditions for non-termination, for the disjunctive skokem chase. By that, we aim to mark the majority of rule sets either as terminating or non-terminating w.r.t. the disjunctive skolem chase. Depending on the results, we may tweak the used (a)cyclicity notions. For instance, based on DMFA(R), other conditions beside the occurrence of a cyclic term can be evaluated. A simple adjustment would be to require a cyclic term of order at least k, meaning that there are at least k nesting levels with the same function symbol.

Furthermore, our results motivate the usage of ASP solvers for reasoning with disjunctive existential rules. Since ASP solvers are well optimized and many description logics are captured by disjunctive existential rules [19], an ASP based implementation of the disjunctive skolem chase promises to provide the basis for more efficient reasoning for such logics. We can use the same transformation from description logics to disjunctive existential rules as we will use for the evaluation of DMFA. We believe that this allows for significant improvements in the practical applications of description logics and reasoning w.r.t. knowledge bases in general.

Acknowledgements

Many thanks to Prof. Dr. Markus Krötzsch for the possibility of conducting this thesis in the Knowledge-Based Systems Research Group at the TU Dresden. Special thanks to David Carral, Ph.D. for supervising this thesis, providing a lot of helpful ideas and remarks, and very fruitful discussions.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Adrian, W.T., Alviano, M., Calimeri, F., Cuteri, B., Dodaro, C., Faber, W., Fuscà, D., Leone, N., Manna, M., Perri, S., Ricca, F., Veltri, P., Zangari, J.: The ASP system DLV: advancements and applications. Künstliche Intell. 32(2-3), 177–179 (2018)
3. Alviano, M., Amendola, G., Dodaro, C., Leone, N., Maratea, M., Ricca, F.: Evaluation of disjunctive programs in WASP. In: Balduccini, M., Lierler, Y., Woltran, S. (eds.) Logic Programming and Nonmonotonic Reasoning - 15th International Conference, LPNMR 2019. Lecture Notes in Computer Science, vol. 11481, pp. 241–255. Springer (2019)
4. Baget, J., Leclère, M., Mugnier, M., Salvat, E.: On rules with existential variables: Walking the decidability line. Artif. Intell. 175(9-10), 1620–1654 (2011)
5. Beeri, C., Vardi, M.Y.: The implication problem for data dependencies. In: Even, S., Kariv, O. (eds.) Automata, Languages and Programming, 8th Colloquium. Lecture Notes in Computer Science, vol. 115, pp. 73–85. Springer (1981)
6. Bourhis, P., Manna, M., Morak, M., Pieris, A.: Guarded-based disjunctive tuple-generating dependencies. ACM Trans. Database Syst. 41(4), 27:1–27:45 (2016)
7. Calì, A., Gottlob, G., Pieris, A.: Query answering under non-guarded rules in datalog+/-. In: Hitzler, P., Lukasiewicz, T. (eds.) Web Reasoning and Rule Systems - Fourth International Conference, RR 2010. Lecture Notes in Computer Science, vol. 6333, pp. 1–17. Springer (2010)
8. Carral, D., Dragoste, I., Krötzsch, M.: Restricted chase (non)termination for existential rules with disjunctions. In: Sierra, C. (ed.) Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017. pp. 922–928. ijcai.org (2017)
9. Cuenca Grau, B., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., Wang, Z.: Acyclicity notions for existential rules and their application to query answering in ontologies. J. Artif. Intell. Res. 47, 741–808 (2013)
10. Deutsch, A., Nash, A., Remmel, J.B.: The chase revisited. In: Lenzerini, M., Lembo, D. (eds.) Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008. pp. 149–158. ACM (2008)
11. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. 336(1), 89–124 (2005)
12. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P.: Theory solving made easy with clingo 5. In: Carro, M., King, A., Saeedloei, N., Vos, M.D. (eds.) Technical Communications of the 32nd International Conference on Logic Programming, ICLP 2016 TCs. OASICS, vol. 52, pp. 2:1–2:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
13. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer Set Solving in Practice. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2012)
14. Gebser, M., Kaufmann, B., Schaub, T.: Multi-threaded ASP solving with clasp. Theory Pract. Log. Program. 12(4-5), 525–545 (2012)
15. Gogacz, T., Marcinkowski, J.: All-instances termination of chase is undecidable. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014. Lecture Notes in Computer Science, vol. 8573, pp. 293–304. Springer (2014)

16. Grahne, G., Onet, A.: Anatomy of the chase. Fundam. Inform. 157(3), 221–270 (2018)
17. Maier, D., Mendelzon, A.O., Sagiv, Y.: Testing implications of data dependencies. ACM Trans. Database Syst. 4(4), 455–469 (1979)
18. Marnette, B.: Generalized schema-mappings: from termination to tractability. In: Paredaens, J., Su, J. (eds.) Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009. pp. 13–22. ACM (2009)
19. Rudolph, S., Krötzsch, M., Hitzler, P.: Type-elimination-based reasoning for the description logic shiqbs using decision diagrams and disjunctive datalog. Logical Methods in Computer Science 8(1) (2012)