General Acyclicity and Cyclicity Notions for the Disjunctive Skolem Chase

Lukas Gerlach[1], **David Carral**[2]

[1] Knowledge-Based Systems Group, TU Dresden, Dresden, Germany
[2] LIRMM, Inria, University of Montpellier, CNRS, Montpellier, France
lukas.gerlach@tu-dresden.de, david.carral@inria.fr

Abstract

The disjunctive skolem chase is a sound, complete, and potentially non-terminating procedure for solving boolean conjunctive query entailment over knowledge bases of disjunctive existential rules. We develop novel acyclicity and cyclicity notions for this procedure; that is, we develop sufficient conditions to determine chase termination and non-termination. Our empirical evaluation shows that our novel notions are significantly more general than existing criteria.

1 Introduction

Solving query entailment over knowledge bases (KBs) of disjunctive existential rules is a relevant decision problem, which is readily defined as follows:

- Input: a set $\mathcal{R}$ of disjunctive existential rules, a set $\mathcal{F}$ of facts, and a boolean conjunctive query (BCQ) γ.
- Output: yes iff γ is a logical consequence of the KB $\langle \mathcal{R}, \mathcal{F} \rangle$ under standard first-order semantics.[1]

One approach to solve BCQ entailment in practice is to apply the *disjunctive skolem chase* (Bourhis et al. 2016), which is a materialization procedure that aims to compute a finite universal model set for an input KB. If fully computed, this model set can then be used to solve query entailment: a BCQ γ is a logical consequence of a KB $\mathcal{K}$ iff γ is satisfied by every model in a universal model set of $\mathcal{K}$ iff γ is satisfied by every model in the output of the chase on input $\mathcal{K}$.

Because the chase is sound and complete for BCQ entailment, and this problem is undecidable (Beeri and Vardi 1981); the chase does not terminate on all inputs. Even worse, we cannot decide if this procedure terminates on a given input (Gogacz and Marcinkowski 2014; Grahne and Onet 2018). Hence, the best one can do is to study *acyclicity notions*; that is, sufficient conditions that confirm chase termination. In our context, acyclicity notions are sufficient conditions that characterize terminating rule sets: A rule set $\mathcal{R}$ is terminating if the chase terminates on every KB of the form $\langle \mathcal{R}, \mathcal{F} \rangle$. To know if our acyclicity notions are as general as they can be, we also study *cyclicity notions*; that is, sufficient conditions for non-termination. In this paper, we focus on the skolem chase variant (Marnette 2009), which makes use of skolem terms that are used to satisfy existential restrictions when computing a universal model set.

While acyclicity notions for rule sets without disjunctions have been around for a while (Fagin et al. 2005; Marnette 2009; Krötzsch and Rudolph 2011; Cuenca Grau et al. 2013; Baget et al. 2014; Karimi, Zhang, and You 2021), the first acyclicity notions for disjunctive rule sets were proposed fairly recently (Carral, Dragoste, and Krötzsch 2017). In their work, Carral, Dragoste, and Krötzsch extended *model-faithful acyclicity* (MFA)[2] for the disjunctive setting and developed the first cyclicity notion for the (disjunctive) skolem chase, named *model-faithful cyclicity* (MFC). To the best of our knowledge, these are the only existing (a)cyclicity notions for non-deterministic rule sets.

We have empirically verified that MFA and MFC are quite effective at determining (non-)termination of rule sets *without disjunctions*: Using both notions, we are able to establish (non-)termination of around 99% of the deterministic rule sets in our evaluation; we present these results in an extended technical report (Gerlach and Carral 2023). However, in the presence of disjunctions, we could only establish the termination status of around 67% of the considered rule sets using MFA and MFC; see Section 5.

Our main goal is thus clear: We aim to develop general (a)cyclicity notions that can be used to determine chase (non-)termination of most real-world rule sets *with disjunctions*. More precisely, our contributions are as follows:

- In Sections 3 and 4, we present our novel (a)cyclicity notions, respectively. Moreover, we study the complexity of checking if a rule set is (a)cyclic and the complexity of solving BCQ entailment over KBs with acyclic rule sets.
- In Section 5, we empirically show that MFA and MFC are significantly less general than our novel conditions, which allow us to establish (non-)termination of many of the (non-deterministic) rule sets in our test suite.
- In Sections 6 and 7, we discuss related research and elaborate on possible follow-up work, respectively.

There is a technical report (Gerlach and Carral 2023) online with full proofs and further empirical results.

[1]Rules, facts, and BCQs are first-order logic formulas, which we formally define in the following section.

[2]MFA was originally introduced in (Cuenca Grau et al. 2013) as a very general skolem acyclicity notion for deterministic rule sets.

2 Preliminaries

We define Cons, Vars, Funs, and Preds to be mutually disjoint, finite (albeit large enough) sets of *constants*, *variables*, *function symbols*, and *predicates*, respectively, such that every $s \in \texttt{Funs} \cup \texttt{Preds}$ has an arity $\mathsf{ar}(s) \geq 1$. For every $i \geq 1$, let $\texttt{Funs}_i = \{f \mid \mathsf{ar}(f) = i\}$ and $\texttt{Preds}_i = \{P \mid \mathsf{ar}(P) = i\}$. The set Terms of *terms* includes $\texttt{Cons} \cup \texttt{Vars}$ and contains $f(t_1, \ldots, t_n)$ for every $n \geq 1$, $f \in \texttt{Funs}_n$, and $t_1, \ldots, t_n \in \texttt{Terms}$. A term t is *functional* if $t \notin \texttt{Cons} \cup \texttt{Vars}$. Given a first-order formula or a term υ, and a set $\texttt{X} \in \{\texttt{Cons}, \texttt{Vars}, \texttt{Funs}_{(i)}, \texttt{Terms}, \texttt{Preds}_{(i)} \mid i \geq 1\}$; let $\texttt{X}(\upsilon)$ be the set of all elements in X that occur in υ.

We write lists $t_1, \ldots, t_n$ of terms as $\vec{t}$, which we often treat as sets. For a term t, let $\mathsf{depth}(t) = 1$ if t is not functional and $\mathsf{depth}(t) = 1 + \max(\mathsf{depth}(s_1), \ldots, \mathsf{depth}(s_n))$ if t is of the form $f(s_1, \ldots, s_n)$. A term s is a *subterm* of another term t if $t = s$, or t is of the form $f(\vec{s})$ and s is a subterm of some term in $\vec{s}$. For a term t, let $\mathsf{subterms}(t)$ be the set of all subterms of t. A term is *cyclic* if it has a subterm of the form $f(\vec{s})$ such that $f \in \texttt{Funs}(\vec{s})$.

An *atom* is a first-order formula of the form $P(\vec{t})$ where P is a $|\vec{t}|$-ary predicate and $\vec{t}$ is a term list. A *fact* is a variable-free atom. For a first-order formula υ, we write $\upsilon[\vec{x}]$ to indicate that $\vec{x}$ is the set of all free variables that occur in υ; that is, those variables that are not explicitly quantified in υ.

Definition 1. *A* (disjunctive existential) rule *is a constant- and function-free first-order formula of the form*

$$\forall \vec{w}, \vec{x}. \big(\beta[\vec{w}, \vec{x}] \to \bigvee\nolimits_{i=1}^{n} \exists \vec{y}_i . \eta_i[\vec{x}_i, \vec{y}_i]\big) \quad (1)$$

where $n \geq 1$; $\vec{w}$, $\vec{x}$, $\vec{y}_1, \ldots, \vec{y}_n$ are pairwise disjoint lists of variables; $\bigcup_{i=1}^{n} \vec{x}_i = \vec{x}$; $\vec{x}_1, \ldots, \vec{x}_n$ are non-empty; and β, $\eta_1, \ldots, \eta_n$ are non-empty conjunctions of atoms.

A rule ρ as in (1) is *deterministic* if $n = 1$, *generating* if it features at least one existential variable, and *datalog* if it is deterministic and not generating. We call $\vec{x}$ the *frontier* of ρ and denote it as $\mathsf{frontier}(\rho)$. Moreover, let $\mathsf{body}(\rho) = \beta$ and $\mathsf{head}_i(\rho) = \eta_i$ for every $1 \leq i \leq n$. Often, we omit universal quantifiers when writing rules and treat conjunctions of atoms, such as $\mathsf{body}(\rho)$, as sets.

A *(boolean conjunctive) query* γ is a first-order formula of the form $\exists \vec{y}.\beta[\vec{y}]$ with β a non-empty atom conjunction. A *knowledge base* (KB) $\mathcal{K}$ is a pair $\langle \mathcal{R}, \mathcal{I} \rangle$ with $\mathcal{R}$ a rule set and $\mathcal{I}$ an *instance*; that is, a function-free fact set. We write $\mathcal{K} \models \gamma$ to denote that (the first-order formula) $\bigwedge_{\rho \in \mathcal{R}} \rho \wedge \bigwedge_{\varphi \in \mathcal{I}} \varphi$ entails γ under standard first-order semantics. In the following, we provide a procedural definition of query entailment via the chase algorithm; see Proposition 1. Without loss of generality, we assume that (†) $y \notin \texttt{Vars}(\mathcal{R} \setminus \{\rho\})$ for every rule set $\mathcal{R}$, every rule $\rho = \beta \to \bigvee_{i=1}^{n} \exists \vec{y}_i . \eta_i$ in $\mathcal{R}$, and every $y \in \bigcup_{i=1}^{n} \vec{y}_i$; that is, existentially quantified variables do not reoccur across different rules in the same rule set.

A *(ground) substitution* σ is a partial function that maps variables to terms without occurrences of variables. We use $[x_1/t_1, \ldots, x_n/t_n]$ to denote the substitution that maps the variable x_i to the term t_i for every $1 \leq i \leq n$. For a first-order formula υ, let $\upsilon\sigma$ be the formula that results from replacing every occurrence of every variable x in the domain of σ in υ with $\sigma(x)$.

Consider a rule ρ as in (1) in a rule set $\mathcal{R}$. For every $y \in \bigcup_{i=1}^{n} \vec{y}_i$, let $f_y \in \texttt{Funs}$ be a fresh $|\vec{x}|$-ary function symbol, which is unique for ρ within $\mathcal{R}$ due to (†). For every $1 \leq i \leq n$, let $\mathsf{sk}(\eta_i)$ be the conjunction obtained by replacing every occurrence of every variable $y \in \vec{y}_i$ in η_i by $f_y(\vec{x})$. Let $\mathsf{sk}(\rho) = \beta \to \bigvee_{i=1}^{n} \mathsf{sk}(\eta_i)$ and $\mathsf{sk}(\mathcal{R}) = \{\mathsf{sk}(\rho) \mid \rho \in \mathcal{R}\}$.

A *trigger* λ is a pair $\langle \rho, \sigma \rangle$ with ρ a rule as in (1) and σ a substitution with domain $\vec{w} \cup \vec{x}$. The trigger λ is *loaded* for a fact set $\mathcal{F}$ if $\beta\sigma \subseteq \mathcal{F}$; it is *active* for $\mathcal{F}$ if $\mathsf{sk}(\eta_i)\sigma \nsubseteq \mathcal{F}$ for all $1 \leq i \leq n$. Let $\mathsf{out}_i(\lambda) = \mathsf{sk}(\eta_i)\sigma$ for $1 \leq i \leq n$; $\mathsf{out}(\lambda) = \{\mathsf{out}_i(\lambda) \mid 1 \leq i \leq n\}$. A fact set $\mathcal{F}$ is *closed under* a rule ρ if no trigger with ρ is loaded and active for $\mathcal{F}$.

Consider a rule set $\mathcal{R}$. An *$\mathcal{R}$-term* is a term defined using the function symbols that occur in $\mathsf{sk}(\mathcal{R})$, some constants, and some variables. A substitution is an *$\mathcal{R}$-substitution* if its range is a set of $\mathcal{R}$-terms. An *$\mathcal{R}$-trigger* is a trigger with a rule from $\mathcal{R}$ and an $\mathcal{R}$-substitution.

Definition 2. *A* (skolem) chase tree *of a KB $\langle \mathcal{R}, \mathcal{I} \rangle$ is a directed tree $T = \langle V, E, \mathsf{fct}, \mathsf{trg} \rangle$ such that:*

1. *Let V be a set of vertices, E a set of edges,* fct *a labeling function that maps the vertices in V to fact sets (*fact labels*), and* trg *a labeling function that maps the vertices in V to $\mathcal{R}$-triggers (*trigger labels*) or ϵ.*
2. *For the root r of T, we have $\mathsf{fct}(r) = \mathcal{I}$ and $\mathsf{trg}(r) = \epsilon$.*
3. *Consider some non-leaf vertex $v \in V$ with children $U = \{u \mid \langle v, u \rangle \in E\}$. There is an $\mathcal{R}$-trigger λ that is loaded and active for $\mathsf{fct}(v)$, $\{\mathsf{fct}(u) \mid u \in U\} = \{\mathcal{F} \cup \mathsf{fct}(v) \mid \mathcal{F} \in \mathsf{out}(\lambda)\}$, $|U| = |\mathsf{out}(\lambda)|$, and $\mathsf{trg}(u) = \lambda$ for each $u \in U$. Moreover, if ρ is not datalog, then $\mathsf{fct}(v)$ is closed under every datalog rule in $\mathcal{R}$ (that is, datalog-first).*
4. *Every leaf fact label is closed under all rules in $\mathcal{R}$. Moreover, for every $\mathcal{R}$-trigger λ, there is some $k \geq 1$ such that λ is not loaded or not active for $\mathsf{fct}(v)$ for every $v \in V$ of depth at least k (that is, fairness).*

Consider a chase tree $T = \langle V, E, \mathsf{fct}, \mathsf{trg} \rangle$ for a KB $\langle \mathcal{R}, \mathcal{I} \rangle$. A *branch* B of a chase tree T is a sequence $v_1, v_2, \ldots \in V$ such that v_1 is the root of T, $\langle v_i, v_{i+1} \rangle \in E$ for every $1 \leq i < |B|$, and if B is finite then its last element is a leaf in T. That is, a branch is a maximal path in T.

A KB *terminates* if it only admits finite chase trees. A rule set $\mathcal{R}$ *terminates* if every KB of the form $\langle \mathcal{R}, \mathcal{I} \rangle$ terminates. It is undecidable to determine if $\mathcal{R}$ terminates already for deterministic rule sets (Gogacz and Marcinkowski 2014).

The *result* of a chase tree T is the set of all fact sets that can be constructed by taking the union of all fact labels in a branch of T. Hence, the result of a finite chase tree T is the set of fact labels of its leaves. In the presence of disjunctions, chase trees for the same KB may yield different results:

Example 1. *Consider the KB $\langle \{P(x, y) \to \exists z.H(y) \wedge S(y, z), P(x, y) \to H(y) \vee \exists w.P(y, w)\}, \{P(a, b)\} \rangle$. We can produce a finite chase tree by prioritizing the application of the first rule and an infinite one by delaying it. The former results in: $\{\{P(a, b), H(b), S(b, f_z(b))\}\}$*

Finite chase results can be used to solve query entailment:

Proposition 1. *Consider the result $\mathfrak{R}$ of some (arbitrarily chosen) chase tree of a $\mathcal{K}$. Then, $\mathcal{K}$ entails a query $\gamma = \exists \vec{y}.\beta$ iff $\mathcal{F} \models \gamma$ for every $\mathcal{F} \in \mathfrak{R}$ iff for every $\mathcal{F} \in \mathfrak{R}$ there is a substitution σ with $\beta\sigma \subseteq \mathcal{F}$.*

3 Acyclicity Notions

In Section 3.1, we recall MFA (Cuenca Grau et al. 2013). In Section 3.2, we present disjunctive model-faithful acyclity, based on ideas from (Carral, Dragoste, and Krötzsch 2017).

3.1 Model-Faithful Acyclicity (MFA)

To determine if a deterministic rule set $\mathcal{R}$ is MFA we check the fact set $\mathsf{MFA}(\mathcal{R})$, which contains all facts that may occur in a chase tree of a KB with $\mathcal{R}$ modulo replacement of constants with $\star$; we formalize this intuition in Lemma 1.

Definition 3. *The* critical instance $\mathcal{I}_\star$ *is the set of all facts with any predicate in (the finite set)* Preds *and the special constant* $\star$*; that is,* $\mathcal{I}_\star = \{P(\star, \dots, \star) \mid P \in \texttt{Preds}\}$.

For a deterministic rule set $\mathcal{R}$*, let* $\mathsf{MFA}(\mathcal{R}) \supseteq \mathcal{I}_\star$ *be the minimal fact set that includes* $\mathsf{out}_1(\lambda)$ *for every (deterministic)* $\mathcal{R}$*-trigger* λ *that is loaded for* $\mathsf{MFA}(\mathcal{R})$.

Definition 4. *A* constant mapping g *is a partial function from* Cons *to* Terms*. For a term* t*, let* $g(t)$ *be the term that results from replacing every occurrence of every* c *in the domain of* g *in* t *with* $g(c)$.

We can prove the following via induction on a chase tree:

Lemma 1. *For a fact label* $\mathcal{F}$ *in a chase tree of a KB with a deterministic rule set* $\mathcal{R}$*, we have* $g^\star(\mathcal{F}) \subseteq \mathsf{MFA}(\mathcal{R})$ *where* $g^\star$ *is the constant mapping that maps every constant to* $\star$.

Consider a chase tree T for a KB $\langle \mathcal{R}, \mathcal{I} \rangle$. By Lemma 1, the depth of the terms that occur in T is bounded by the depth of the terms in $\mathsf{MFA}(\mathcal{R})$ since $\mathsf{depth}(t) = \mathsf{depth}(g^\star(t))$ for every term t. Since only a finite number of terms of bounded depth can be defined with the constants that occur in $\mathcal{I}$, finiteness of $\mathsf{MFA}(\mathcal{R})$ implies finiteness of T. Therefore:

Lemma 2. *If* $\mathsf{MFA}(\mathcal{R})$ *is finite for some deterministic rule set* $\mathcal{R}$*, then* $\mathcal{R}$ *terminates.*

Consider a deterministic rule set $\mathcal{R}$. Then, $\mathsf{MFA}(\mathcal{R})$ is finite iff $\langle \mathcal{R}, \mathcal{I}_\star \rangle$ terminates. Gogacz and Marcinkowski have shown that we cannot decide the latter; hence, we cannot decide if $\mathsf{MFA}(\mathcal{R})$ is finite either. However, we can compute this set up to the occurrence of a *cyclic* term.

Definition 5. *A deterministic rule set* $\mathcal{R}$ *is* MFA *if no cyclic term occurs in* $\mathsf{MFA}(\mathcal{R})$.

The occurrence of a cyclic term indicates that a rule ρ is applied in a chase tree to produce a descendant of a term introduced to satisfy ρ. In many real-world cases, this implies that infinitely many applications of ρ may follow.

The following theorem is a corollary of Lemma 2 and the fact that, for a deterministic rule set $\mathcal{R}$, the fact set $\mathsf{MFA}(\mathcal{R})$ is finite if it does not feature cyclic terms:

Theorem 3. *Deterministic MFA rule sets terminate.*

MFA was originally defined for rule sets without disjunctions (Cuenca Grau et al. 2013). Carral, Dragoste, and Krötzsch came up with a straightforward way to extend this acyclicity notion for the disjunctive setting; see Theorem 4.

Definition 6. *For a rule* ρ *as in* (1) *and a rule set* $\mathcal{R}$*, let* $\rho^\wedge = \beta \to \exists \vec{y}_1, \dots, \vec{y}_n . \bigwedge_{i=1}^n \eta_i$ *and* $\mathcal{R}^\wedge = \{\rho^\wedge \mid \rho \in \mathcal{R}\}$.

Theorem 4. *A rule set* $\mathcal{R}$ *terminates if* $\mathcal{R}^\wedge$ *terminates.*

Applying Theorems 3 and 4, we can extend MFA (and any other deterministic skolem acyclicity notion) so it can be applied rule sets with disjunctions:

Definition 7. *A rule set* $\mathcal{R}$ *is MFA if* $\mathcal{R}^\wedge$ *is MFA.*

Corollary 5. *MFA rule sets terminate.*

3.2 Disjunctive MFA (DMFA)

To determine if a (possibly non-deterministic) rule set $\mathcal{R}$ is DMFA, we look for cyclic terms in the fact set $\mathsf{DMFA}(\mathcal{R})$, which has the same property as $\mathsf{MFA}(\mathcal{R}^\wedge)$. Namely, this fact set contains all facts that may occur in a chase tree of a KB with $\mathcal{R}$ modulo replacement of all constants with $\star$; see Lemma 9. However, $\mathsf{DMFA}(\mathcal{R})$ is a tighter over-approximation than $\mathsf{MFA}(\mathcal{R}^\wedge)$; in fact, later on we show that $\mathsf{DMFA}(\mathcal{R})$ is a subset of $\mathsf{MFA}(\mathcal{R}^\wedge)$ for every rule set $\mathcal{R}$; see the proof of Theorem 11.

In order to minimize $\mathsf{DMFA}(\mathcal{R})$, we adjust the notion of blockedness,[3] which we use to characterize harmless triggers that are never applied in any chase tree:

Example 2. *Consider* $\mathcal{R} = \{(2\text{–}5)\}$*, which is a slightly simplified subset of rule set* `00007.owl` *in the Oxford Ontology Repository (see Section 5):*

$$\mathit{evidence}(x) \to \exists w.\mathit{Confidence}(x, w) \quad (2)$$

$$\mathit{Confidence}(x, y) \to \mathit{confidence}(y) \quad (3)$$

$$\mathit{Confidence}(x, y) \to \exists z.\mathit{XRef}(y, z) \quad (4)$$

$$\mathit{XRef}(x, y) \to \mathit{evidence}(x) \vee \mathit{confidence}(x) \quad (5)$$

Consider a chase tree $T = \langle V, E, \mathsf{fct}, \mathsf{trg} \rangle$ *for a KB of the form* $\langle \mathcal{R}, \mathcal{I} \rangle$ *and suppose for a contradiction that* $\mathsf{trg}(v) = \langle (5), [x/f_w(t), y/f_z(f_w(t))] \rangle$ *for some* $v \in V$ *and a term* t*. Then,* $\mathit{Confidence}(t, f_w(t)) \in \mathsf{fct}(p)$ *with* p *the parent of* v *since* $f_w(t)$ *may only be introduced in* T *via the application of* (2)*. Moreover,* $\mathit{confidence}(f_w(t)) \in \mathsf{fct}(p)$ *since* $\mathsf{fct}(p)$ *is closed under* (3)*; see Item 3 in Definition 2. But then, the trigger* $\mathsf{trg}(v)$ *is not active for* $\mathsf{fct}(p)$*! In fact, we can use blockedness to show that triggers such as* $\mathsf{trg}(v)$ *may never occur as a trigger labels in a chase tree of a KB with* $\mathcal{R}$.

To define blockedness, we introduce the fact set $\mathcal{U}(\mathcal{R}, \lambda)$ for a given rule set $\mathcal{R}$ and a trigger λ. Intuitively, this fact set can be "homomorphically embedded" into the fact label of a vertex v in a chase tree T of a KB with $\mathcal{R}$ if λ is applied to v in T; see Lemma 6.

Definition 8. *Let* $\mathcal{R}$ *be a rule set and* t *an* $\mathcal{R}$*-term.*

- *If* t *is not functional, then* $\mathcal{U}(\mathcal{R}, t) = \emptyset$.
- *Otherwise,* t *is of the form* $f_y(\vec{s})$ *and there is exactly one rule* $\rho = \beta[\vec{w}, \vec{x}] \to \bigvee_{i=1}^n \exists \vec{y}_i . \eta_i[\vec{x}_i, \vec{y}_i] \in \mathcal{R}$ *and exactly one* $1 \leq \ell \leq n$ *with* $y \in \vec{y}_\ell$*. Then,* $\mathcal{U}(\mathcal{R}, t) = \beta\sigma \cup \mathsf{out}_\ell(\langle \rho, \sigma \rangle) \cup \bigcup_{s \in \vec{s}} \mathcal{U}(\mathcal{R}, s)$ *where* σ *is a substitution with* $\vec{x}\sigma = \vec{s}$ *and* $\vec{w}\sigma = \vec{c}$ *for fresh constants* $\vec{c}$.

Consider an $\mathcal{R}$*-trigger* $\lambda = \langle \rho, \sigma \rangle$*. Then, let* $\mathcal{U}(\mathcal{R}, \lambda)$ *be the minimal fact set that includes* $\mathsf{body}(\rho)\sigma$ *and* $\mathcal{U}(\mathcal{R}, t)$ *for every* t *in the range of* σ*, and that is closed under every datalog rule in* $\mathcal{R}$ *if* ρ *is not datalog.*

[3](Carral, Dragoste, and Krötzsch 2017) have introduced a very similar notion for the restricted chase.

An $\mathcal{R}$-trigger λ is blocked *for a rule set $\mathcal{R}$ if its rule is not datalog and λ is not active for $\mathcal{U}(\mathcal{R}, \lambda)$.*

Lemma 6. *Consider a chase tree $T = \langle V, E, \mathsf{fct}, \mathsf{trg}\rangle$ of a KB $\langle \mathcal{R}, \mathcal{I}\rangle$. Then, for every $v \in V$, there is a constant mapping g that is the identity on $\mathtt{Cons}(\mathsf{fct}(v))$ such that*

A. $g(\mathcal{U}(\mathcal{R}, t)) \subseteq \mathsf{fct}(v)$ *for every* $t \in \mathtt{Terms}(\mathsf{fct}(v))$ *and*
B. $g(\mathcal{U}(\mathcal{R}, \mathsf{trg}(u))) \subseteq \mathsf{fct}(v)$ *for every* $\langle v, u\rangle \in E$.

Consider some trigger λ (with a non-datalog rule) that is blocked for $\mathcal{R}$ and suppose for a contradiction that λ is the trigger label of a vertex u in a chase tree T of a KB with $\mathcal{R}$. Then, λ is not active for the fact label of the parent v of u in T by Lemma 6, which contradicts Definition 2. Therefore:

Lemma 7. *If a trigger λ is blocked for $\mathcal{R}$, then λ does not occur as a trigger label in any chase tree of a KB with $\mathcal{R}$.*

Relying on blockedness, we can safely ignore many facts when we define the over-approximation $\mathsf{DMFA}(\mathcal{R})$:

Definition 9. *For a rule set $\mathcal{R}$, let $\mathsf{DMFA}(\mathcal{R}) \supseteq \mathcal{I}_\star$ be the fact set that includes all sets in $\mathsf{out}(\langle \rho, \sigma\rangle)$ for every $\mathcal{R}$-trigger $\langle \rho, \sigma\rangle$ such that (i) $\langle \rho, \sigma\rangle$ is loaded for $\mathsf{DMFA}(\mathcal{R})$ and (ii) $\langle \rho, \sigma_r\rangle$ is not blocked for $\mathcal{R}$.*

In the above, let σ_r be a substitution such that, for every $x \in \mathsf{domain}(\sigma)$, the term $\sigma_r(x)$ is obtained by replacing every occurrence of a constant in $\sigma(x)$ with a fresh constant.[4]

We need σ_r to generalize over all possible KBs with $\mathcal{R}$. All "less general" triggers will also be blocked:

Lemma 8. *For a trigger $\langle \rho, \sigma\rangle$, a rule set $\mathcal{R}$, and a constant mapping g; if $\langle \rho, \sigma_r\rangle$ is blocked for $\mathcal{R}$, then so is $\langle \rho, g \circ \sigma_r\rangle$.*

Armed with Lemmas 7 and 8, we can readily show the following result via induction on the structure of a chase tree:

Lemma 9. *For a fact label $\mathcal{F}$ in a chase tree of a KB $\langle \mathcal{R}, \mathcal{I}\rangle$, we have that $g^\star(\mathcal{F}) \subseteq \mathsf{DMFA}(\mathcal{R})$ where $g^\star$ is the constant mapping that maps every constant to $\star$.*

As for MFA, we simply compute $\mathsf{DMFA}(\mathcal{R})$ up to the occurrence of a cyclic term to check if a rule set $\mathcal{R}$ is DMFA:

Definition 10. *A rule set $\mathcal{R}$ is* DMFA *if no cyclic term occurs in* $\mathsf{DMFA}(\mathcal{R})$.

Theorem 10. *DMFA rule sets terminate.*

A rule set $\mathcal{R}$ is DMFA if it is MFA since $\mathsf{DMFA}(\mathcal{R})$ is a subset of $\mathsf{MFA}(\mathcal{R})$ by Definitions 3 and 9. Furthermore, the rule set in Example 2 is DMFA but not MFA. Therefore:

Theorem 11. *If a rule set $\mathcal{R}$ is MFA, then it is DMFA. Moreover, the converse of this implication does not hold.*

The number of acyclic terms that one can define with the functions in $\mathtt{Funs}(\mathsf{sk}(\mathcal{R}))$ and $\star$ is double-exponential in $\mathcal{R}$; hence, so is $|\mathsf{DMFA}(\mathcal{R})|$. Moreover, for an instance $\mathcal{I}$, we have that $|\mathtt{Terms}(\mathsf{DMFA}(\mathcal{R}))| \cdot |\mathtt{Cons}(\mathcal{I})|$ is an upper bound for the number of terms in any chase tree of $\langle \mathcal{R}, \mathcal{I}\rangle$. Once we realise these claims, we can readily show that:

Theorem 12. *DMFA-membership is* 2ExpTime*-complete.*

Theorem 13. *Deciding query entailment for a KB with an DMFA rule set is co*N2ExpTime*-complete.*

[4]For example, $\sigma_r = [x/f(b,c), y/d]$ if $\sigma = [x/f(a,a), y/a]$.

4 Cyclicity Notions

Cyclicity notions are sufficient conditions that characterize non-terminating rule sets. In fact, the conditions we consider in this section imply a stronger form of non-termination:

Definition 11. *A rule set $\mathcal{R}$* never terminates *if there is a KB $\langle \mathcal{R}, \mathcal{I}\rangle$ that does not admit any finite chase tree.*

In Section 4.1, we recall MFC (Carral, Dragoste, and Krötzsch 2017). In Section 4.2, we present disjunctive model-faithful cyclicity (DMFC), which is based on ideas from the same authors.

4.1 Model Faithful Cyclicity (MFC)

Intuitively speaking, the idea behind MFC (Carral, Dragoste, and Krötzsch 2017) is to check if a generating rule is reapplied when starting on a minimal instance that mimics a fact label where the rule has just been applied. If the rule is indeed reapplied and yields a cyclic term, then it can be applied infinitely many times; see Theorem 14.

Definition 12. *For a rule ρ as in* (1) *and some $1 \leq k \leq n$, let $\mathcal{I}_{\rho,k} = \mathsf{body}(\rho)\sigma_{uc} \cup \mathsf{out}_k(\langle \rho, \sigma_{uc}\rangle)$ where σ_{uc} is a substitution that maps every variable x to a fresh constant c_x. If ρ is deterministic, we define $\mathcal{I}_\rho = \mathcal{I}_{\rho,1}$.*

Given a rule set $\mathcal{R}$ and a deterministic rule $\rho \in \mathcal{R}$, we first define the fact set $\mathsf{MFC}(\mathcal{R}, \rho)$, which consists of facts that appear on all branches of all chase trees of $\langle \mathcal{R}, \mathsf{body}(\rho)\sigma_{uc}\rangle$. Note that we use $\mathsf{body}(\rho)\sigma_{uc}$ instead $\mathcal{I}_\rho$ in the previous KB because the latter may feature function symbols and hence, it may not be an instance.

Definition 13. *For a rule set $\mathcal{R}$ and a deterministic rule $\rho \in \mathcal{R}$, let $\mathsf{MFC}(\mathcal{R}, \rho) \supseteq \mathcal{I}_\rho$ be the minimal fact set that includes $\mathsf{out}_1(\lambda)$ for every $\mathcal{R}$-trigger λ such that (i) λ is loaded for $\mathsf{MFC}(\mathcal{R}, \rho)$, (ii) the rule in λ is deterministic, and (iii) the substitution in λ does not feature cyclic terms in its range.*

Condition *(iii)* ensures that $\mathsf{MFC}(\mathcal{R}, \rho)$ is always finite.

Definition 14. *A rule set $\mathcal{R}$ is* MFC *if a ρ-cyclic term occurs in $\mathsf{MFC}(\mathcal{R}, \rho)$ for a deterministic rule $\rho \in \mathcal{R}$. That is, a term of the form $f(\vec{s})$ with $f \in \mathtt{Funs}(\mathsf{sk}(\rho))$ and $f \in \mathtt{Funs}(\vec{s})$.*

If $\mathcal{R}$ is MFC, then $\langle \mathcal{R}, \mathsf{body}(\rho)\sigma_{uc}\rangle$ does not terminate:

Theorem 14. *MFC rule sets are never terminating.*

Sketch. If a rule set $\mathcal{R}$ is MFC, then $\mathsf{MFC}(\mathcal{R}, \rho)$ features a ρ-cyclic term t for a deterministic rule $\rho \in \mathcal{R}$. Hence, there is a list of $\mathcal{R}$-triggers applied during the construction of $\mathsf{MFC}(\mathcal{R}, \rho)$ that leads to t. More precisely, there is a list $\lambda_1, \ldots, \lambda_n$ such that, for every $1 \leq i \leq n$: *(i)* $\lambda_i = \langle \rho_i, \sigma_i\rangle$, *(ii)* $\mathsf{out}_1(\lambda_i) \subseteq \mathsf{MFC}(\mathcal{R}, \rho)$, *(iii)* ρ_i is deterministic, *(iv)* λ_i is loaded for $\mathcal{I}_\rho \cup \bigcup_{j=1}^{i-1} \mathsf{out}_1(\lambda_j)$, *(v)* $\mathsf{out}_1(\lambda_n)$ features a ρ-cyclic term, and *(vi)* $\bigcup_{j=1}^{n-1} \mathsf{out}_1(\lambda_j)$ does not. This list can be extended into an infinite sequence: For every $1 \leq i \leq n$ and $j \geq 1$, let λ_i^j be the $\mathcal{R}$-trigger $\langle \rho_i, g^{\circ j-1} \circ \sigma_i\rangle$ where g is the constant mapping with $\sigma_n = g \circ \sigma_{uc}$.[5]

Consider a chase tree $T = \langle V, E, \mathsf{fct}, \mathsf{trg}\rangle$ of the KB $\mathcal{K} = \langle \mathcal{R}, \mathsf{body}(\rho)\sigma_{uc}\rangle$. Then, for every branch $v_1, v_2, \ldots \in V$ of

[5]Note that $g^{\circ 0} = \mathsf{id}_{\mathtt{Terms}}$, $g^{\circ 1} = g$, $g^{\circ 2} = g \circ g$, and so on.

T, we can show via structural induction that the following holds: For every $1 \leq i \leq n$ and $j \geq 1$, the trigger λ_i^j is loaded for $\mathsf{fct}(v_k)$ for some $k \geq 1$ and $\mathsf{out}_1(\lambda_i^j) \subseteq \mathsf{fct}(v_\ell)$ for some $\ell \geq k$. Hence, every branch of T is infinite by *(v)* and *(vi)* and hence, $\mathcal{K}$ does not admit finite chase trees. $\square$

The induction step at the end of the previous sketch is easy to show once one realizes that:

Lemma 15. *Consider a vertex v in a branch B of a chase tree $T = \langle V, E, \mathsf{fct}, \mathsf{trg}\rangle$ of a KB $\langle \mathcal{R}, \mathcal{I}\rangle$, and an $\mathcal{R}$-trigger λ. If λ features a deterministic rule and is loaded for $\mathsf{fct}(v)$, then $\mathsf{fct}(v) \cup \mathsf{out}_1(\lambda) \subseteq \mathsf{fct}(u)$ for some $u \in B$.*

Intuitively, this means that, once a deterministic trigger is loaded for a vertex v in a chase tree, every branch with v includes the output of this trigger. Note that such a result does not hold for non-deterministic triggers; see Example 2.

4.2 Disjunctive Model-Faithful Cyclicity (DMFC)

We ignore non-deterministic rules when deciding MFC membership (see Definition 13). Hence, this notion fails to characterise non-terminating rule sets such as:

Example 3. *Consider the rule set $\mathcal{R} = \{R(x,y) \to A(y) \vee B(y), A(x) \to \exists y.R(x,y)\}$, which never terminates since every chase tree for $\langle \mathcal{R}, \{A(c)\}\rangle$ features (exactly) one infinite branch. However, $\mathcal{R}$ is not MFC; to establish never termination we need to take the disjunctive rule into account.*

We consider head-choices to deal with disjunctive rules:

Definition 15. *A* head-choice *is a function* hc *that maps every rule $\beta \to \bigvee_{i=1}^n \exists \vec{y}_i.\eta_i$ to some $1 \leq j \leq n$. For a trigger $\lambda = \langle \rho, \sigma\rangle$, let $\mathsf{out}_{\mathsf{hc}}(\lambda) = \mathsf{out}_{\mathsf{hc}(\rho)}(\lambda)$.*

Later on, we show that some rule sets are not terminating by focusing on the branch in a tree induced by a head-choice:

Definition 16. *For a chase tree $T = \langle V, E, \mathsf{fct}, \mathsf{trg}\rangle$ and a head-choice hc, let $\mathsf{branch}(T, \mathsf{hc}) = v_1, v_2, \ldots$ be the branch of T such that $\mathsf{fct}(v_{i+1}) = \mathsf{out}_{\mathsf{hc}}(\mathsf{trg}(v_{i+1})) \cup \mathsf{fct}(v_i)$ for every $1 \leq i < |\mathsf{branch}(T, \mathsf{hc})|$.*

To use disjunctive rules to witness non-termination, we identify triggers that need to be applied once they are loaded. To do so, we define *unblockable triggers*[6] $\lambda = \langle \rho, \sigma\rangle$ for a rule set $\mathcal{R}$ and a head-choice hc, which satisfy the following:

I. Consider a chase tree T of a KB with $\mathcal{R}$. If λ becomes loaded in $\mathsf{branch}(T, \mathsf{hc})$, then $\mathsf{out}_{\mathsf{hc}}(\lambda)$ is eventually included in $\mathsf{branch}(T, \mathsf{hc})$; that is, Lemma 16.

II. Unblockability propagates across an infinite family of triggers. Namely, if a constant mapping g is *reversible* (see Definition 18), then the trigger $\langle \rho, g \circ \sigma\rangle$ is also unblockable; that is Lemma 17.

Definition 17. *Let $\mathcal{R}$ be a rule set and t an $\mathcal{R}$-term.*

- *If t is not functional, then $\mathcal{H}(\mathcal{R}, t) = \emptyset$.*
- *Otherwise, t is of the form $f_y(\vec{s})$ and there is exactly one rule $\rho = \beta[\vec{w}, \vec{x}] \to \bigvee_{i=1}^n \exists \vec{y}_i.\eta_i[\vec{x}_i, \vec{y}_i] \in \mathcal{R}$ and exactly one $1 \leq \ell \leq n$ with $y \in \vec{y}_\ell$. Then, $\mathcal{H}(\mathcal{R}, t) = \mathsf{out}_\ell(\langle \rho, \sigma\rangle) \cup \bigcup_{s \in \vec{s}} \mathcal{H}(\mathcal{R}, s)$ where σ is a substitution with $\vec{x}\sigma = \vec{s}$.*

[6] Again, (Carral, Dragoste, and Krötzsch 2017) introduced a very similar notion for the restricted chase.

Consider an $\mathcal{R}$-trigger $\lambda = \langle \rho, \sigma\rangle$. Then, let $\mathcal{H}(\mathcal{R}, \lambda)$ be the minimal fact set that includes $\mathcal{H}(\mathcal{R}, t)$ for every t in the range of σ restricted to variables in $\mathsf{frontier}(\rho)$. Additionally, let the term-skeleton *of λ be $\mathsf{skeleton}_{\mathcal{R}}(\lambda) = \mathtt{Terms}(\mathcal{H}(\mathcal{R}, \lambda)) \cup \mathtt{Cons}(\{\sigma(x) \mid x \in \mathsf{frontier}(\rho)\})$.*

For a rule $\rho = \beta \to \bigvee_{i=1}^n \exists \vec{y}_i.\eta_i$, let $\mathsf{star}(\rho) = \beta \to \bigvee_{i=1}^n \eta_i'$ be the (non-generating) rule where η_i' is the conjunction that results from replacing every occurrence of every $y \in \vec{y}_i$ in η_i with $\star$.

For a rule set $\mathcal{R}$, a head-choice hc, and an $\mathcal{R}$-trigger λ, let $\mathcal{O}(\mathcal{R}, \mathsf{hc}, \lambda)$ be the minimal fact set that includes:

- *The set $\mathcal{H}(\mathcal{R}, \lambda)$.*
- *The set of all facts that can be defined using any predicate and constants in $\mathtt{Cons}(\mathsf{skeleton}_{\mathcal{R}}(\lambda)) \cup \{\star\}$.*
- *The set $\mathsf{out}_{\mathsf{hc}}(\langle \mathsf{star}(\rho), \sigma\rangle)$ for every $\mathcal{R}$-trigger $\langle \rho, \sigma\rangle$ loaded for $\mathcal{O}(\mathcal{R}, \mathsf{hc}, \lambda)$ with $\mathsf{out}_{\mathsf{hc}}(\lambda) \neq \mathsf{out}_{\mathsf{hc}}(\langle \rho, \sigma\rangle)$.*

The trigger λ is unblockable *for $\mathcal{R}$ and hc if it features a deterministic rule or if it is active for $\mathcal{O}(\mathcal{R}, \mathsf{hc}, \lambda)$.*

Lemma 16. *Consider a chase tree $T = \langle V, E, \mathsf{fct}, \mathsf{trg}\rangle$ of a KB $\langle \mathcal{R}, \mathcal{I}\rangle$, a head-choice hc, some $v \in \mathsf{branch}(T, \mathsf{hc})$, and an $\mathcal{R}$-trigger λ. If λ is loaded for $\mathsf{fct}(v)$, and it is unblockable for $\mathcal{R}$ and hc; then $\mathsf{fct}(v) \cup \mathsf{out}_{\mathsf{hc}}(\lambda) \subseteq \mathsf{fct}(u)$ for some $u \in \mathsf{branch}(T, \mathsf{hc})$.*

Sketch. For a term t, let $h_\lambda(t) = t$ if $t \in \mathsf{skeleton}_{\mathcal{R}}(\lambda)$ and $h_\lambda(t) = \star$ otherwise. We have that, if $\mathsf{out}_{\mathsf{hc}}(\lambda) \not\subseteq \mathsf{fct}(w)$ for some $w \in \mathsf{branch}(T, \mathsf{hc})$, then $h_\lambda(\mathsf{fct}(w)) \subseteq \mathcal{O}(\mathcal{R}, \mathsf{hc}, \lambda)$. That is, $\mathcal{O}(\mathcal{R}, \mathsf{hc}, \lambda)$ "over-approximates" fact labels in $\mathsf{branch}(T, \mathsf{hc})$ that do not include $\mathsf{out}_{\mathsf{hc}}(\lambda)$.

Assume that the premise of the lemma holds. If ρ is deterministic, the claim holds by Lemma 15. Otherwise, λ is active for the "over-approximation" $\mathcal{O}(\mathcal{R}, \mathsf{hc}, \lambda)$. Hence, λ remains active for the fact labels in $\mathsf{branch}(T, \mathsf{hc})$ up until its corresponding output is included in the branch. $\square$

Definition 18. *Consider a set T of terms that includes $\mathsf{subterms}(t)$ for every $t \in T$. A constant mapping g is* reversible *for T if (i) the domain of g includes $\mathtt{Cons}(T)$, (ii) $t \neq s$ implies $g(t) \neq g(s)$ for every $t, s \in T$, and (iii) for every $c \in \mathtt{Cons}(T)$ and every $s \in \mathsf{subterms}(g(c))$, there is no functional term $u \in T$ with $g(u) = s$.*

Lemma 17. *Consider a rule set $\mathcal{R}$, a head-choice hc, an $\mathcal{R}$-trigger $\langle \rho, \sigma\rangle$, and a constant mapping g that is reversible for $\mathsf{skeleton}_{\mathcal{R}}(\langle \rho, \sigma\rangle)$. If $\langle \rho, g \circ \sigma\rangle$ is an $\mathcal{R}$-trigger and $\langle \rho, \sigma\rangle$ is unblockable for $\mathcal{R}$ and hc, then so is $\langle \rho, g \circ \sigma\rangle$.*

Sketch. Assume that the premise of the lemma holds. If ρ is deterministic, the claim holds by Definition 17. Otherwise, $\langle \rho, \sigma\rangle$ is active for $\mathcal{O}(\mathcal{R}, \mathsf{hc}, \langle \rho, \sigma\rangle)$. Hence, $\langle \rho, g \circ \sigma\rangle$ is also active for $\mathcal{O}(\mathcal{R}, \mathsf{hc}, \langle \rho, g \circ \sigma\rangle)$ since $h(\mathcal{O}(\mathcal{R}, \mathsf{hc}, \langle \rho, g \circ \sigma\rangle)) \subseteq \mathcal{O}(\mathcal{R}, \mathsf{hc}, \langle \rho, \sigma\rangle)$ with h the function defined as follows: For a term t, let $h(t) = s$ if there is a term s that occurs in $\mathsf{skeleton}_{\mathcal{R}}(\langle \rho, \sigma\rangle)$ with $g(s) = t$, and $h(t) = \star$ otherwise. Note that h is well-defined because g is reversible (see *(ii)* in Definition 18), and the inclusion holds because $\mathcal{O}(\mathcal{R}, \mathsf{hc}, \langle \rho, \sigma\rangle)$ contains all facts that can be defined with constants in $\mathsf{skeleton}_{\mathcal{R}}(\langle \rho, \sigma\rangle)$ and $\star$. $\square$

We are ready to define DMFC and prove it sound:

Definition 19. *Consider a rule set $\mathcal{R}$, a head-choice* hc*, and a rule $\rho \in \mathcal{R}$. Then, let $\mathsf{DMFC}(\mathcal{R}, \mathsf{hc}, \rho) \supseteq \mathcal{I}_{\rho,\mathsf{hc}(\rho)}$ be the fact set that includes $\mathsf{out}_{\mathsf{hc}}(\lambda)$ for every $\mathcal{R}$-trigger $\lambda = \langle \psi, \sigma \rangle$ such that (i) λ is loaded for* $\mathsf{DMFC}(\mathcal{R}, \mathsf{hc}, \rho)$*, (ii) λ is unblockable for $\mathcal{R}$ and* hc*, (iii) there are no cyclic terms in the range of σ, (iv) there is a frontier variable $x \in \mathsf{frontier}(\psi)$ with $\sigma(x)$ being functional if ψ is non-datalog, and (v) σ is injective if $\psi = \rho$.*

Definition 20. *A rule set $\mathcal{R}$ is* DMFC *if* $\mathsf{DMFC}(\mathcal{R}, \mathsf{hc}, \rho)$ *features a ρ-cyclic term for a $\rho \in \mathcal{R}$ and a head-choice* hc.

Theorem 18. *DMFC rule sets are never terminating.*

Sketch. If a rule set $\mathcal{R}$ is DMFC, then $\mathsf{DMFC}(\mathcal{R}, \mathsf{hc}, \rho)$ features a ρ-cyclic term t for some head-choice hc and some $\rho \in \mathcal{R}$. Then, there is a list $\lambda_1, \ldots, \lambda_n$ of unblockable $\mathcal{R}$-triggers applied during the construction of $\mathsf{DMFC}(\mathcal{R}, \mathsf{hc}, \rho)$ that yields t. More precisely; for every $1 \leq i \leq n$; let $\lambda_i = \langle \rho_i, \sigma_i \rangle$; $\mathsf{out}_{\mathsf{hc}}(\lambda_i) \subseteq \mathsf{DMFC}(\mathcal{R}, \mathsf{hc}, \rho)$; the trigger λ_i is unblockable for $\mathcal{R}$ and hc, and is loaded for $\mathcal{I}_\rho \cup \bigcup_{j=1}^{i-1} \mathsf{out}_{\mathsf{hc}}(\lambda_j)$; the function σ_n is injective; and $\mathsf{out}_{\mathsf{hc}}(\lambda_n)$ features a ρ-cyclic term and $\bigcup_{j=1}^{n-1} \mathsf{out}_{\mathsf{hc}}(\lambda_j)$ does not. As in Theorem 14, we extend this list into an infinite sequence: For every $1 \leq i \leq n$ and every $j \geq 1$, let $\lambda_i^j = \langle \rho_i, g^{\circ j-1} \circ \sigma_i \rangle$ where g is the constant mapping with $\sigma_n = g \circ \sigma_{uc}$.

Let $\mathcal{F} = \mathcal{I}_{\rho,\mathsf{hc}(\rho)} \cup \bigcup_{j \geq 1} \bigcup_{i=1}^n \mathsf{out}_{\mathsf{hc}}(\lambda_i^j)$ and assume (for now) that g is reversible for $\mathtt{Terms}(\mathcal{F})$. We show that $\mathsf{branch}(T, \mathsf{hc})$ is infinite for every tree T of $\langle \mathcal{R}, \mathsf{body}(\rho)\sigma_{uc} \rangle$. First, $\mathcal{I}_{\rho,\mathsf{hc}(\rho)}$ occurs in some fact label in $\mathsf{branch}(T, \mathsf{hc})$; otherwise, λ_n would not be unblockable. Then, by induction, for every $1 \leq i \leq n$ and $j \geq 1$, the trigger λ_i^j is loaded for some fact label in $\mathsf{branch}(T, \mathsf{hc})$ and hence, some fact label in the branch includes $\mathsf{out}_{\mathsf{hc}}(\lambda_i^j)$ by Lemma 16. We can apply this lemma here because g is reversible and hence, λ_i^j is unblockable by Lemma 17.

It remains to show that g is reversible for $\mathtt{Terms}(\mathcal{F})$ to complete our proof. First, we show the claims below:

a. There are no ρ-cyclic terms in $\bigcup_{j=1}^{n-1} \mathsf{out}_{\mathsf{hc}}(\lambda_j)$. Therefore, for every constant c in $\mathcal{F}$, the term $g(c)$ does not feature nested function symbols from $\mathsf{sk}(\rho)$.[7]
b. By *(iv)* in Definition 19: For every functional term t occurring in $\mathsf{DMFC}(\mathcal{R}, \mathsf{hc}, \rho)$, there is some subterm s of t that is also functional and that occurs in $\mathcal{I}_{\rho,\mathsf{hc}(\rho)}$; that is, s is of the form $f(\vec{c})$ with $f \in \mathtt{Funs}(\mathsf{sk}(\rho))$ and $\vec{c}$ a list containing every constant in $\sigma_{uc}(\mathsf{frontier}(\rho))$. We can extend this claim to all functional terms in $\mathcal{F}$ via induction.
c. There is some constant $c \in \sigma_{uc}(\mathsf{frontier}(\rho))$ such that $g(c)$ features a function symbol from $\mathsf{sk}(\rho)$. Otherwise $\mathsf{out}_{\mathsf{hc}}(\lambda_n)$ would not feature a ρ-cyclic term.
d. By (b) and (c): For every functional term t in $\mathcal{F}$, the term $g(t)$ features nested function symbols from $\mathsf{sk}(\rho)$.

To verify that g is reversible for the terms in $\mathcal{F}$ we separately prove *(i)*, *(ii)*, and *(iii)* from Definition 18. The first one holds since the domain of g is $\mathtt{Cons}(\mathcal{I}_{\rho,\mathsf{hc}(\rho)})$.

To show *(ii)*, we check that $g(t) \neq g(s)$ for every $t, s \in \mathtt{Terms}(\mathcal{F})$ with $t \neq s$ via structural induction on t. Regarding the base case, we consider two cases: If t and s are constants, then $g(t) \neq g(s)$ since (σ_n and) g are injections. If t is a constant and s is functional, then $g(s)$ features nested function symbols from $\mathsf{sk}(\rho)$ by (d) and $g(t)$ does not by (a). Regarding the induction step, we again consider two cases: If t and s are functional terms of the form $f(\vec{t})$ and $h(\vec{s})$, respectively, with $f \neq h$; then $g(t) \neq g(s)$ since $g(t) = f(g(\vec{t}))$ and $g(s) = h(g(\vec{s}))$. If t and s are functional terms of the form $f(t_1, \ldots, t_n)$ and $f(s_1, \ldots, s_n)$, respectively; then $t_i \neq s_i$ for some $1 \leq i \leq n$ since $t \neq s$, $g(t_i) \neq g(s_i)$ by induction hypothesis, and $g(t) \neq g(s)$.

Finally, we show that *(iii)* holds by contradiction. Consider a functional term $t \in \mathtt{Terms}(\mathcal{F})$ with $g(c) = t$ for some constant c and assume that there is a functional term u and a subterm s of t such that $g(u) = s$. By (a), the term t does not feature nested function symbols from $\mathsf{sk}(\rho)$; hence, s does not feature them either. However, s features nested functional symbols from $\mathsf{sk}(\rho)$ by (d)! $\square$

Because of *(iv)* and *(v)* in Definition 19, DMFC is not more general than MFC. However, in our experiments, we did not find a single rule set that is MFC but not DMFC.

Regarding complexity, checking MFC and DMFC is dominated by the number of acyclic terms, which is double-exponential in the size of the given rule set (Cuenca Grau et al. 2013; Carral, Dragoste, and Krötzsch 2017).

Theorem 19. *(D)MFC-membership is* 2EXPTIME*-comp.*

5 Evaluation

We present experiments to show the generality of our notions in practice. We describe our implementation, the rule sets we use, and the results of our experiments. The tools, rule sets, and results of the evaluation are available online.[8] Further information on the concrete steps to reproduce the evaluation steps is also provided there.

To avoid an exponential number of checks, we consider a simplified version of DMFC in our implementation:

Definition 21. *For a rule $\rho = \beta \to \bigvee_{j=1}^n \exists \vec{z}_j.\eta_j$ and some $i \geq 1$, let $\mathsf{hc}_i(\rho) = n$ if $i > n$ and $\mathsf{hc}_i(\rho) = i$ if $i \leq n$. A rule set $\mathcal{R}$ is DMFC^s if, for some $\rho \in \mathcal{R}$ and some $i \geq 1$, the fact set $\mathsf{DMFC}(\mathcal{R}, \rho, \mathsf{hc}_i)$ features a ρ-cyclic term.*

By definition, DMFC^s implies DMFC so it ensures never termination. We consider an improvement of DMFA in our implementation, which guarantees termination by Lemma 9:

Definition 22. *A rule set $\mathcal{R}$ is DMFA^k for some $k \geq 1$ if* DMFA *($\mathcal{R}$) does not feature any k-cyclic term; that is, a term with $k + 1$ nested occurrences of the same function symbol.*

We obtain the rule sets in the evaluation from OWL ontologies via normalization and translation into rules; see Section 6 in (Cuenca Grau et al. 2013). We drop OWL axioms with "at-most restrictions" and "nominals" because

[7]The term $f_y(f_z(c))$ features nested function symbols from $\mathsf{sk}(A(x) \to \exists y, z.R(x, y, z))$ while $f_w(f_y(c), f_z(d))$ does not.

[8]https://doi.org/10.5281/zenodo.7375461 Gerlach and Carral

	#∃	# tot.	# fin.	MFA	DMFA	DMFA^2	MFC	DMFC^s
OXFD	1–19	37	36	21	28	28	4	8
	20–99	18	17	3	3	3	10	14
	100+	82	26	4	6	6	14	19
	1+	**137**	**79**	**28 (35%)**	**37 (46%)**	**37 (46%)**	**28 (35%)**	**41 (51%)**
ORE15	1–19	103	98	51	66	66	18	31
	20–99	119	105	32	33	35	54	69
	100–999	278	219	5	6	119	89	100
	1–999	**500**	**422**	**88 (20%)**	**105 (24%)**	**220 (52%)**	**161 (38%)**	**200 (47%)**
MOWL	1–19	1361	1283	676	725	732	173	515
	20–99	894	740	104	114	121	301	610
	100–299	448	254	25	25	111	103	143
	1–299	**2703**	**2277**	**805 (35%)**	**864 (37%)**	**964 (42%)**	**577 (25%)**	**1268 (55%)**

Table 1: Skolem Chase Termination: Non-Deterministic Rule Sets

their translation requires the use of equality; one can incorporate this feature via axiomatisation (Carral and Urbani 2020). The ontologies come from the Oxford Ontology Repository (OXFD),[9] the dataset of the OWL Reasoner Evaluation 2015 (ORE15),[10] and the Manchester OWL Corpus (MOWL).[11] Here, we only consider rule sets with at least one disjunctive and one generating rule. Deterministic rule sets are covered largely by MFA and MFC already; we include results for these rule sets in our technical report (Gerlach and Carral 2023).

We count the number of rule sets that are MFA, DMFA$^{(2)}$, MFC, and DMFC^s and present our results in Table 1. We set a timeout of 30 minutes for each check and only consider rule sets for which all checks finished; we indicate the number of attempted vs finished rule sets by # tot. and # fin., respectively. We group results by the number of generating rules, indicated by $\#\exists$. For instance, in the second row in Table 1 we indicate: There are 18 rule sets in the OXFD corpus with at least 20 but at most 99 generating rules; all checks finished for 17 of these; 3 of these are MFA; etc.

If we use MFA and MFC, the percentage of finished rule sets that are fully classified (i.e., sets that are MFA or MFC) for OXFD, ORE15, and MOWL are 70%, 58%, and 60%, respectively. Our improved notions are significantly more general; if we apply them, we can now classify 97%, 99%, and 97% of the finished rule sets in these repositories. Moreover, the use of DMFA^2 allows us to detect that many (hitherto unclassified) rule sets terminate for the skolem chase!

6 Related Work

Leclère et al. and Calautti, Gottlob, and Pieris showed that checking chase termination for linear and guarded deterministic rule sets, respectively, is decidable.

Definition 23. *A rule ρ is* linear *if it features a single atom in its body; it is* guarded *if it features an atom in its body that contains all of the universally quantified variables in ρ.*

[9]https://www.cs.ox.ac.uk/isg/ontologies/

[10]https://doi.org/10.5281/zenodo.18578 Parsia et al.

[11]https://doi.org/10.5281/zenodo.16708 Matentzoglu et al.

Note that all linear rules are guarded, and that over half of the rule sets in Table 1 are not guarded since they contain rules of the form $\bigwedge_{i=1}^{n} R_i(x_{i-1}, x_i) \to R(x_0, x_n)$ with $n \geq 2$. In the extended technical report (Gerlach and Carral 2023), we present separate results for non-guarded rules, which are quite similar percentage-wise to those in Table 1.

Theorem 4 allows us to extend any deterministic skolem acyclicity notions for non-deterministic rule sets. Instead of MFA, we could consider the following:

Definition 24. *For a computable function δ over the naturals, a rule set $\mathcal{R}$ is δ-*bounded *if the depth of terms in* $\mathsf{MFA}(\mathcal{R})$ *is bounded by* $\delta(|\mathcal{R}|)$.

For a computable function δ over the naturals, we can decide δ-bounded membership and this property implies termination (Zhang, Zhang, and You 2015). Alternatively, instead of considering δ-boundedness, one can simply increase the number k in Definition 22 to achieve a similar effect. In fact, we ran some tests and only found 2 rule sets that are DMFA^5 but not DMFA^2. Hence, we have decided to not publish results for $k > 2$ and believe that using δ-boundedness would not result in a big increase in performance in practice.

7 Conclusions and Future Work

We present novel (a)cyclicity notions that allow us to establish the termination status of most rule sets in our test suite.

As for immediate future work, we plan to extend our notions to the restricted chase and investigate why some rule sets are not classified as (non)-terminating. Potentially, we fail to capture these because they are "sometimes" non-terminating; that is, they may occur in KBs that admit finite and infinite chase trees. We would also like to develop a normalisation procedure that preserves both query entailment and chase termination.

As a long term goal, we would like to adapt our notions so they can be applied in other areas of knowledge representation and reasoning. For instance, we believe that we can use our ideas to *(i)* show if an ASP program with function symbols does or does not admit a finite solution or *(ii)* determine if DPLL(T) algorithms used in automated theorem proving will terminate or not for many real-world inputs.

Acknowledgments

Lukas is funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in project 389792660 (TRR 248, Center for Perspicuous Systems), by the Bundesministerium für Bildung und Forschung (BMBF, Federal Ministry of Education and Research) under European ITEA project 01IS21084 (InnoSale, Innovating Sales and Planning of Complex Industrial Products Exploiting Artificial Intelligence), by BMBF and DAAD (German Academic Exchange Service) in project 57616814 (SECAI, School of Embedded and Composite AI), and by the Center for Advancing Electronics Dresden (cfaed).

David is funded by the ANR project CQFD (ANR-18-CE23-0003).

References

Baget, J.; Garreau, F.; Mugnier, M.; and Rocher, S. 2014. Extending Acyclicity Notions for Existential Rules. In Schaub, T.; Friedrich, G.; and O'Sullivan, B., eds., *ECAI 2014 - 21st European Conference on Artificial Intelligence, 2014, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, 39–44. IOS Press.

Beeri, C.; and Vardi, M. Y. 1981. The Implication Problem for Data Dependencies. In Even, S.; and Kariv, O., eds., *Automata, Languages and Programming, 8th Colloquium, Israel, 1981, Proceedings*, volume 115 of *Lecture Notes in Computer Science*, 73–85. Springer.

Bourhis, P.; Manna, M.; Morak, M.; and Pieris, A. 2016. Guarded-Based Disjunctive Tuple-Generating Dependencies. *ACM Trans. Database Syst.*, 41(4): 27:1–27:45.

Calautti, M.; Gottlob, G.; and Pieris, A. 2015. Chase Termination for Guarded Existential Rules. In Milo, T.; and Calvanese, D., eds., *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Australia, 2015*, 91–103. ACM. ISBN 978-1-4503-2757-2.

Carral, D.; Dragoste, I.; and Krötzsch, M. 2017. Restricted Chase (Non)Termination for Existential Rules with Disjunctions. In Sierra, C., ed., *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Australia, 2017*, 922–928. ijcai.org.

Carral, D.; and Urbani, J. 2020. Checking Chase Termination over Ontologies of Existential Rules with Equality. In *The 34th AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, USA, 2020*, 2758–2765. AAAI Press.

Cuenca Grau, B.; Horrocks, I.; Krötzsch, M.; Kupke, C.; Magka, D.; Motik, B.; and Wang, Z. 2013. Acyclicity Notions for Existential Rules and Their Application to Query Answering in Ontologies. *J. Artif. Intell. Res.*, 47: 741–808.

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

Gerlach, L.; and Carral, D. 2022. General Acyclicity and Cyclicity Notions for the Disjunctive Skolem Chase - Evaluation Material. https://doi.org/10.5281/zenodo.7375461. Accessed: 2022-12-01.

Gerlach, L.; and Carral, D. 2023. General Acyclicity and Cyclicity Notions for the Disjunctive Skolem Chase (Extended Technical Report). https://iccl.inf.tu-dresden.de/web/Inproceedings3348. Accessed: 2023-03-01.

Gogacz, T.; and Marcinkowski, J. 2014. All-Instances Termination of Chase is Undecidable. In Esparza, J.; Fraigniaud, P.; Husfeldt, T.; and Koutsoupias, E., eds., *Automata, Languages, and Programming - 41st Int. Colloquium, ICALP 2014, Denmark, 2014, Proc., Part II*, volume 8573 of *Lecture Notes in Computer Science*, 293–304. Springer.

Grahne, G.; and Onet, A. 2018. Anatomy of the Chase. *Fundam. Informaticae*, 157(3): 221–270.

Karimi, A.; Zhang, H.; and You, J. 2021. Restricted Chase Termination for Existential Rules: A Hierarchical Approach and Experimentation. *Th. Pract. Log. Program.*, 21(1): 4–50.

Krötzsch, M.; and Rudolph, S. 2011. Extending Decidable Existential Rules by Joining Acyclicity and Guardedness. In Walsh, T., ed., *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Spain, 2011*, 963–968. IJCAI/AAAI.

Leclère, M.; Mugnier, M.; Thomazo, M.; and Ulliana, F. 2019. A Single Approach to Decide Chase Termination on Linear Existential Rules. In Barceló, P.; and Calautti, M., eds., *22nd International Conference on Database Theory, ICDT 2019, 2019, Portugal*, volume 127 of *LIPIcs*, 18:1–18:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Marnette, B. 2009. Generalized schema-mappings: from termination to tractability. In Paredaens, J.; and Su, J., eds., *Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, 2009, USA*, 13–22. ACM.

Matentzoglu, N.; Tang, D.; Parsia, B.; and Sattler, U. 2014. The Manchester OWL Repository: System Description. In Horridge, M.; Rospocher, M.; and van Ossenbruggen, J., eds., *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Italy, 2014*, volume 1272 of *CEUR Workshop Proceedings*, 285–288. CEUR-WS.org.

Parsia, B.; Matentzoglu, N.; Gonçalves, R. S.; Glimm, B.; and Steigmiller, A. 2016. The OWL Reasoner Evaluation (ORE) 2015 Resources. In Groth, P.; Simperl, E.; Gray, A. J. G.; Sabou, M.; Krötzsch, M.; Lécué, F.; Flöck, F.; and Gil, Y., eds., *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Japan, 2016, Proc, Part II*, volume 9982 of *Lecture Notes in Computer Science*, 159–167.

Zhang, H.; Zhang, Y.; and You, J. 2015. Existential Rule Languages with Finite Chase: Complexity and Expressiveness. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, USA*, 1678–1685. AAAI Press.