Acyclicity Conditions and their Application to Query Answering in Description Logics

Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, and **Zhe Wang**
Department of Computer Science, Oxford University
Oxford, United Kingdom

Abstract

Answering conjunctive queries (CQs) over a set of facts extended with existential rules is a key problem in knowledge representation and databases. This problem can be solved using the *chase* (aka *materialisation*) algorithm; however, CQ answering is undecidable for general existential rules, so the chase is not guaranteed to terminate. Several *acyclicity conditions* provide sufficient conditions for chase termination. In this paper, we present two novel such conditions—*model-faithful acyclicity* (MFA) and *model-summarising acyclicity* (MSA)—that generalise many of the acyclicity conditions known so far in the literature.

Materialisation provides the basis for several widely-used OWL 2 DL reasoners. In order to avoid termination problems, many of these systems handle only the OWL 2 RL profile of OWL 2 DL; furthermore, some systems go beyond OWL 2 RL, but they provide no termination guarantees. In this paper we investigate whether various acyclicity conditions can provide a principled and practical solution to these problems. On the theoretical side, we show that query answering for acyclic ontologies is of lower complexity than for general ontologies. On the practical side, we show that many of the commonly used OWL 2 DL ontologies are MSA, and that the facts obtained via materialisation are not too large. Thus, our results suggest that principled extensions to materialisation-based OWL 2 DL reasoners may be practically feasible.

Introduction

Existential rules are positive, function-free first-order implications that may contain existentially quantified variables in the head. In databases, they are known as *dependencies* (Abiteboul, Hull, and Vianu 1995) and are used to capture a wide range of schema constraints; in particular, they are used as declarative data transformation rules in *data exchange*—the process of transforming a database structured according to a source schema into a database structured according to a target schema (Fagin et al. 2005). They also provide the foundation for several prominent knowledge representation formalisms, such as Datalog$^{\pm}$ (Calì, Gottlob, and Pieris 2010; Calì et al. 2010).

Answering *conjunctive queries* (CQs) over a set of facts extended with existential rules is a fundamental reasoning problem in both database and KR settings. This problem is undecidable (Beeri and Vardi 1981) in general, and it can be characterised using *chase* (Johnson and Klug 1984; Maier, Mendelzon, and Sagiv 1979), a technique closely related to the hypertableau calculus (Motik, Shearer, and Horrocks 2009). The chase extends in a forward-chaining manner the original set of facts by introducing facts implied by the rules. The result of the chase is called the *universal* model, and an arbitrary conjunctive query can be answered over the original set of facts and the rules by simply evaluating the query in the universal model.

Rules with existentially quantified variables in the head—so-called *generating rules*—require the introduction of fresh individuals, and cyclic applications of generating rules may lead to non-termination; moreover, determining whether chase terminates on a set of rules and facts is undecidable. However, several decidable classes of existential rules have been identified, and the existing proposals can be classified into two main groups. In the first group, rules are *restricted* such that their (possibly infinite) universal models can be represented using finitary means. This group includes rules with universal models of bounded treewidth (Baget et al. 2011), guarded rules (Calì et al. 2010), and 'sticky' rules (Calì, Gottlob, and Pieris 2011). In the second group, one uses a sufficient (but not necessary) *acyclicity* condition that ensures chase termination. Roughly speaking, acyclicity conditions analyse information flow between the rules to ensure that no cyclic applications of generating rules are possible. *Weak acyclicity* (WA) (Fagin et al. 2005) was one of the first such notions, and it was extended to *safety* (SF) (Meier, Schmidt, and Lausen 2009), *stratification* (ST) (Deutsch, Nash, and Remmel 2008), *acyclicity of a graph of rule dependencies* (aGRD) (Baget, Mugnier, and Thomazo 2011), *joint acyclicity* (JA) (Krötzsch and Rudolph 2011), and *super-weak acyclicity* (SWA) (Marnette 2009).

Acyclicity conditions are relevant for at least two reasons. First, unlike guarded rules, acyclic rules can axiomatise structures of arbitrary shapes, as long as these structures are bounded in size. Second, the chase result for acyclic rules can be stored and manipulated as if it were a database. This is important in data exchange, where the goal is to materialise the transformed database.

In this paper, we argue that acyclicity is also relevant for *description logics* (DLs), the KR formalisms underpinning the OWL 2 DL ontology language (Cuenca Grau et al. 2008). CQ answering over DL ontologies is a key reasoning service in many DL applications, and the problem was studied for numerous different DLs (Calvanese et al. 2007; Krötzsch, Rudolph, and Hitzler 2007; Glimm et al. 2008; Ortiz, Calvanese, and Eiter 2008; Lutz, Toman, and Wolter 2009; Pérez-Urbina, Motik, and Horrocks 2010; Rudolph and Glimm 2010; Kontchakov et al. 2011).

Answering CQs over expressive DLs, however, is quite technical and of high computational complexity. Therefore, practical applications often solve this problem using *materialisation*, in which ontology consequences are precomputed using forward-chaining and stored in a semantic data store; examples of such systems include Oracle's Semantic Data Store (Wu et al. 2008), Sesame (Broekstra, Kampman, and van Harmelen 2002), Jena (Carroll et al. 2004), OWLim (Kiryakov, Ognyanov, and Manov 2005), and DLE-Jena (Meditskos and Bassiliades 2008). This approach is possible if (*i*) the ontology is *Horn* (Hustadt, Motik, and Sattler 2005), and (*ii*) forward-chaining is guaranteed to terminate. In practice, condition (*ii*) is achieved by computing the materialisation only w.r.t. the inference rules that correspond to the part of the ontology in the OWL 2 RL profile; this systematically excludes generating rules and is thus terminating, but incomplete in general. Generating rules are partially supported in systems such as OWLim (Bishop and Bojanov 2011) and Jena; however, such support is typically ad hoc and provides no completeness and/or termination guarantees. Acyclicity conditions can be used to address these issues: if a Horn ontology is acyclic, a complete materialisation can be computed without the risk of non-termination.

Motivated by the practical importance of chase termination, in this paper we present two new acyclicity conditions: *model-faithful acyclicity* (MFA) and *model-summarising acyclicity* (MSA). Roughly speaking, these acyclicity conditions use a particular model of the rules to analyse the implications between existential quantifiers, which is why we call them *model based*. In particular, MFA uses the actual 'canonical' model induced by the rules, which makes it a very general condition. We prove that checking whether a set of existential rules is MFA is 2EXPTIME-complete, and it becomes EXPTIME-complete if the predicates in the rules are of bounded arity. Due to the high complexity of MFA checking, MFA may be unsuitable for practical application, so we introduce MSA. Intuitively, MSA can be understood as MFA in which the analysis is performed over models that 'summarise' (or overestimate) the actual models. Checking MSA of existential rules can be realised via checking entailment of ground atoms in datalog programs; we use this close connection between MSA and datalog to prove that checking MSA is EXPTIME-complete for general existential rules, and that it becomes coNP-complete if the arity of rule predicates is bounded. Finally, we prove that MSA is strictly more general than SWA—one of the most general acyclicity conditions currently is use.

Both of these conditions can be applied to general existential rules *without* equality. Equality can be incorporated via *singularisation* (Marnette 2009)—a technique that transforms the rules to encode the effects of equality, but does not prevent the evaluation of a conjunctive query in the universal model. Singularisation is orthogonal to acyclicity: after computing the transformed rules, one can use MFA, MSA, or in fact an arbitrary acyclicty notion to check whether the result is acyclic; if so, the chase of the original set of rules will terminate. Unfortunately, singularisation is nondeterministic: some ways of transforming the rules may produce acyclic rule sets, but not all ways are guaranteed to do so. Thus, we refine singularisation to obtain an upper and a lower bound for acyclicity. We also show that, when used with JA, the lower bound coincides with WA.

Finally, we consider various theoretical and practical issues surrounding the use of acyclicity for CQ answering over DL ontologies. On the theoretical side, we show that checking MFA and MSA of Horn-$\mathcal{SHIQ}$ ontologies is PSPACE- and PTIME-complete, respectively, and that answering CQs over acyclic Horn-$\mathcal{SHIQ}$ ontologies is in PSPACE. The latter problem is EXPTIME-hard for general (i.e., not acyclic) Horn-$\mathcal{SHIQ}$ ontologies (Ortiz, Rudolph, and Simkus 2011), so acyclicity makes the problem easier. Furthermore, Horn ontologies can be extended with arbitrary SWRL rules (Horrocks and Patel-Schneider 2004) without affecting neither decidability nor worst-case complexity, provided that the union of the ontology and SWRL rules is acyclic; this is in contrast to the general case, where SWRL extensions of DLs lead to the undecidability.

On the practical side, we explore the limits of reasoning with acyclic OWL 2 DL ontologies via materialisation. We checked MFA, MSA, and JA for 149 Horn ontologies; furthermore, to estimate the impact of materialisation, we compared the size of the materialisation with the number of facts in the original ontologies. Our experiments revealed that many ontologies are MSA, and that some complex ones are MSA but not JA; furthermore, the universal models obtained via materialisation are typically not too large. Thus, our results suggest that principled, materialisation-based reasoning for ontologies beyond the OWL 2 RL profile may be practically feasible.

Preliminaries

We use the standard notions of constants, function symbols, and predicate symbols, where $\approx$ is the equality predicate. Each function or predicate symbol is associated with a nonnegative integer arity. Variables, terms, substitutions, atoms, and first-order formulae, sentences, interpretations (i.e., structures), and models are defined as usual. We abbreviate with $\vec{t}$ a vector of terms $t_1, \ldots, t_n$ and define $|\vec{t}| = n$. With $\varphi(\vec{x})$ we stress that $\vec{x} = x_1, \ldots, x_n$ are the free variables of a formula φ, and with $\varphi\sigma$ we denote the result of applying a substitution σ to φ. A term, atom, or formula is *ground* if it does not contain variables; a *fact* is a ground atom. The *depth* $\mathsf{dep}(t)$ of a term t is defined as 0 if t is a constant or a variable, and $\mathsf{dep}(t) = 1 + \max_{i=1}^{n} \mathsf{dep}(t_i)$ if $t = f(\vec{t})$. Satisfaction of a sentence φ in an interpretation I (written $I \models \varphi$), and entailment of a sentence ψ from a sentence φ (written $\varphi \models \psi$), are defined as usual. By a slight abuse of notation, we identify a conjunction of atoms with a set of atoms. A term t' is a *subterm* of a term t if $t' = t$ or

$t = f(\vec{t})$ and t' is a subterm of some $t_i \in \vec{t}$; if additionally $t' \neq t$, then t' is a *proper* subterm of t. An atom $P(\vec{t})$ *contains* a term t if $t \in \vec{t}$, and a set of atoms I *contains* t if some atom in I contains t.

An *instance* is a finite set of function-free facts. An *existential rule* (or just *rule*) is a sentence of the form

$$\forall \vec{x} \forall \vec{z}.[\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}.\psi(\vec{x}, \vec{y})] \tag{1}$$

where $\varphi(\vec{x}, \vec{z})$ and $\psi(\vec{x}, \vec{y})$ are conjunctions of atoms, and the tuples of variables $\vec{x}$, $\vec{y}$, and $\vec{z}$ are pairwise disjoint. Formula φ is the *body* and formula ψ is the *head* of the rule. For brevity, quantifiers $\forall \vec{x} \forall \vec{z}$ are often omitted. If $\vec{y}$ is empty, the rule is called a *datalog* rule. In database theory, satisfaction and entailment are often considered only w.r.t. *finite* interpretations under the *unique name assumption* (UNA), where distinct constants are interpreted as distinct elements; in contrast, such assumptions are not customary in KR. Since we study rules that can be satisfied in finite models, the restriction to finite satisfiability is immaterial; also, we do not assume UNA, which can be axiomatised if needed.

A *conjunctive query* (CQ) is a formula $Q(\vec{x})$ of the form $\exists \vec{y}.\varphi(\vec{x}, \vec{y})$, where $\varphi(\vec{x}, \vec{y})$ is a conjunction of atoms; the query is *Boolean* if $\vec{x}$ is empty. A substitution θ mapping $\vec{x}$ to constants is an *answer* to $Q(\vec{x})$ w.r.t. a set of rules Σ and instance I if $\Sigma \cup I \models Q(\vec{x})\theta$. Answering CQs is a fundamental reasoning problem in many applications of existential rules.

In first-order logic, the equality predicate $\approx$ is commonly assumed to have a predefined interpretation. The semantics of $\approx$, however, can be axiomatised explicitly. Let Σ be an arbitrary set of rules; w.l.o.g. we assume that no rule in Σ contains $\approx$ in the body. Then, $\Sigma_{\approx} = \emptyset$ if $\approx$ does not occur in Σ; otherwise, $\Sigma_{\approx}$ contains rules (2)–(4) and an instance of rule (5) for each predicate P occurring in Σ.

$$\rightarrow x \approx x \tag{2}$$

$$x_1 \approx x_2 \rightarrow x_2 \approx x_1 \tag{3}$$

$$x_1 \approx x_2 \wedge x_2 \approx x_3 \rightarrow x_1 \approx x_3 \tag{4}$$

$$P(\vec{x}) \wedge x_i \approx x'_i \rightarrow P(x_1, \ldots, x'_i, \ldots, x_n) \tag{5}$$

The consequences of Σ (where $\approx$ is treated as having a well-defined interpretation) and $\Sigma \cup \Sigma_{\approx}$ (where $\approx$ is treated as an ordinary predicate) coincide.

Sometimes we use *skolemisation* to interpret rules in *Herbrand* interpretations—possibly infinite sets of ground atoms. In particular, for each rule r of the form (1) and each variable $y_i \in \vec{y}$, let f_r^i be a function symbol that is globally unique for r and y_i; furthermore, let θ be the substitution such that $\theta(y_i) = f_r^i(\vec{x})$ for each $y_i \in \vec{y}$. Then, the skolemisation $\mathsf{sk}(r)$ of r is the rule

$$\varphi(\vec{x}, \vec{z}) \rightarrow \psi(\vec{x}, \vec{y})\theta \tag{6}$$

The skolemisation $\mathsf{sk}(\Sigma)$ of a set of rules Σ is obtained by skolemising each rule in Σ. For each CQ $Q(\vec{x})$, instance I, and substitution σ, we have $\Sigma \cup I \models Q(\vec{x})\sigma$ if and only if $\mathsf{sk}(\Sigma) \cup \Sigma_{\approx} \cup I \models Q(\vec{x})\sigma$.

Answering CQs can be characterised using *chase*, and we use the *skolem chase* variant (Marnette 2009). The result of *applying* a skolemised rule $r = \varphi \rightarrow \psi$ to a set of ground atoms I is the smallest set $r(I)$ that contains $\psi\sigma$ for each substitution σ from the variables in r to the terms occurring in I such that $\varphi\sigma \subseteq I$; furthermore, for Ω a set of skolemised rules, $\Omega(I) = \bigcup_{r \in \Omega} r(I)$. Let I be a finite set of ground atoms, let Σ be a set of rules, let $\Sigma' = \mathsf{sk}(\Sigma) \cup \Sigma_{\approx}$, and let Σ'_f and Σ'_n be the subsets of Σ' containing rules with and without function symbols, respectively. The *chase sequence* for I and Σ is a sequence of sets of facts $I_\Sigma^0, I_\Sigma^1, \ldots$ where $I_\Sigma^0 = I$, and I_Σ^i for $i > 0$ is defined as follows:

- if $\Sigma'_n(I_\Sigma^{i-1}) \not\subseteq I_\Sigma^{i-1}$, then $I_\Sigma^i = I_\Sigma^{i-1} \cup \Sigma'_n(I_\Sigma^{i-1})$,
- otherwise $I_\Sigma^i = I_\Sigma^{i-1} \cup \Sigma'_f(I_\Sigma^{i-1})$.

The *chase* of I and Σ is defined as $I_\Sigma^\infty = \bigcup_i I_\Sigma^i$; note that I_Σ^∞ can be infinite. Chase can be used as a 'database' for answering CQs: a substitution σ is an answer to Q over Σ and I iff $I_\Sigma^\infty \models Q\sigma$. Chase of I and Σ *terminates* if $i \geq 0$ exists such that $I_\Sigma^i = I_\Sigma^j$ for each $j \geq i$; chase of Σ terminates *universally* if the chase of I and Σ terminates for each I. If the skolem chase of I and Σ terminates, then the nonoblivious chase (Fagin et al. 2005), and the core chase (Deutsch, Nash, and Remmel 2008) of I and Σ terminate as well.

The *critical instance* I_Σ^* for a set of rules Σ contains all facts that can be constructed using all predicates occurring in Σ, all constants occurring in the body of a rule in Σ, and one special fresh constant $*$. If the skolem chase for I_Σ^* and Σ terminates, then the skolem chase of Σ terminates universally (Marnette 2009).

Universal chase termination is undecidable, and various sufficient *acyclicity* conditions have been proposed. In the following, let Σ be a set of rules where w.l.o.g. no variable occurs in more than one rule. A *position* is an expression of the form $P|_i$ where P is an n-ary predicate and $1 \leq i \leq n$. Given a rule r of the form (1) and a variable w, the set $\mathsf{Pos}_B(w)$ of *body positions* of w consists of all positions $P|_i$ for which $P(t_1, \ldots, t_n) \in \varphi(\vec{x}, \vec{z})$ exists with $t_i = w$. The set $\mathsf{Pos}_H(w)$ is defined analogously.

Weak acyclicity (WA) (Fagin et al. 2005) can be used with rules containing the equality predicate. The *WA dependency graph* D_Σ for Σ contains positions as vertices; furthermore, for each rule $r \in \Sigma$ of the form (1), each $x \in \vec{x}$, each $P|_i \in \mathsf{Pos}_B(x)$, and each $y \in \vec{y}$, graph D_Σ contains

- a *regular edge* from $P|_i$ to each $Q|_j \in \mathsf{Pos}_H(x)$ such that $Q \neq \approx$ and,
- a *special edge* from $P|_i$ to each $Q|_j \in \mathsf{Pos}_H(y)$ such that $Q \neq \approx$.

Set Σ is WA if D_Σ does not contain a cycle going through a special edge. Equality atoms are effectively ignored by WA.

Joint acyclicity (JA) (Krötzsch and Rudolph 2011) generalises WA, but it is applicable only to equality-free rules. For an existentially quantified variable y in Σ, let $\mathsf{Move}(y)$ be the smallest set of positions such that

- $\mathsf{Pos}_H(y) \subseteq \mathsf{Move}(y)$, and
- for each existential rule $r \in \Sigma$ and each universally quantified variable x occurring in r, if $\mathsf{Pos}_B(x) \subseteq \mathsf{Move}(y)$ then $\mathsf{Pos}_H(x) \subseteq \mathsf{Move}(y)$.

The *JA dependency graph* of Σ is as follows. The vertices are the existentially quantified variables occurring in Σ. Given arbitrary two such variables y_1 and y_2, the JA dependency graph contains an edge from y_1 to y_2 whenever the rule that contains y_2 also contains a universally quantified variable x such that $\mathsf{Pos}_H(x) \neq \emptyset$ and $\mathsf{Pos}_B(x) \subseteq \mathsf{Move}(y_1)$. Set Σ is JA if its JA dependency graph does not contain a cycle.

Super-weak acyclicity (SWA) (Marnette 2009) is a further generalisation of JA, and it can differ from JA only on rules in which a variable occurs more than once in some body atom. Since such rules are not obtained from DL knowledge bases, we omit the somewhat technical definition of SWA.

Spezzano and Greco (2010) suggest a rule rewriting framework for chase termination. A set of rules Σ is rewritten into a set of rules Σ', and then Σ' is checked using an arbitrary acyclicity notion (e.g., weak acyclicity). This technique is thus independent from the actual acyclicity condition used, and so we do not discuss it any further.

Model-Faithful Acyclicity

As the following example shows, known acyclicity conditions, such as JA, do not guarantee chase termination on certain commonly occurring rules.

Example 1. *Let Σ be the set of rules* (7)–(9)*.*

$$r_1 = \quad A(x_1) \rightarrow \exists y_1.R(x_1, y_1) \wedge B(y_1) \quad (7)$$

$$r_2 = \quad R(x_2, z) \wedge B(z) \rightarrow A(x_2) \quad (8)$$

$$r_3 = \quad B(x_3) \rightarrow \exists y_2.R(x_3, y_2) \wedge C(y_2) \quad (9)$$

Note that $\mathsf{Move}(y_1) = \{R|_2,\ B|_1,\ R|_1,\ A|_1\}$; hence, the JA dependency graph has a cyclic edge from y_1 to itself. The chase of Σ, however, terminates universally. Assume that f and g are used to skolemize r_1 and r_3. Given a fact $A(a)$, rule r_1 derives $R(a, f(a))$ and $B(f(a))$, and rule r_3 derives $R(f(a), g(f(a)))$ and $C(g(f(a)))$; after this, rule r_2 is not applicable to $R(f(a), g(f(a)))$ since variable z in r_2 cannot be matched to $B(g(f(a)))$, and so the chase terminates. $\Diamond$

Note that rules r_1 and r_2 in Example 1 encode the DL axiom $A \equiv \exists R.B$, and rule r_3 encodes $B \sqsubseteq \exists R.C$; such axioms abound in OWL ontologies. To enable applications of chase termination outlined in the introduction, we next propose a less restrictive acyclicity condition.

Acyclicity conditions try to estimate whether applying chase to a rule can produce facts that can (possibly by applying chase to other rules) repeatedly trigger the same rule. The key difference between various conditions is how rule applicability is determined. For example, JA and SWA consider each variable in a rule in isolation and do not check satisfaction of all body atoms at once; hence, they overestimate rule applicability. For example, rule (8) is not applicable to the facts generated by rule (9), but this can be determined only by considering variables x_2 and z in rule (8) simultaneously. More precise chase termination guarantees can be obtained by tracking rule applicability more 'faithfully'.

A simple solution is to be completely precise about rule applicability: one can run the skolem chase and then use sufficient checks to identify cyclic computations. Clearly, no sufficient, necessary, and computable condition for the latter can be given, so we must adopt a practical approach; for example, we can 'raise the alarm' and stop the process if the chase derives a term $f(\vec{t})$ where f occurs in $\vec{t}$. This condition can be further refined; for example, one could stop only if f occurs nested in a term some fixed number of times. The choice of the appropriate condition is thus application dependent; however, as our experiments show, checking only for one level of nesting suffices in many cases. In particular, no term $f(\vec{t})$ with f occurring in $\vec{t}$ is generated in the chase of Σ from Example 1.

Meier, Schmidt, and Lausen (2009) proposed a related idea, where the chase is extended to keep track of a 'monitor graph', which is used to track rule dependencies and then stop the chase if certain conditions are satisfied. This approach uses a variant of the chase that is don't-know nondeterministic: while all possible chase applications produce a model, not all applications will terminate, which can make acyclicity checking difficult.

In contrast, our notion of acyclicity is independent from any concrete notion of chase. The given rules Σ are transformed into a new set of rules Σ', which tracks rule dependencies using fresh predicates; then, Σ is identified as being acyclic if Σ' does not entail a special nullary predicate C. Since acyclicity is defined via entailment, it can be decided using any sound and complete theorem proving procedure for existential rules. Acyclicity guarantees termination of skolem chase, which then guarantees termination of nonoblivious and core chase as well. We call our notion *model-faithful acyclicity* because it estimates rule application precisely, by examining the actual model of Σ.

Definition 2. *For each rule $r = \varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}.\psi(\vec{x}, \vec{y})$ and each variable $y_i \in \vec{y}$, let F^i_r be a fresh unary predicate unique for r and y_i; furthermore, let S and D be fresh binary predicates, and let C be a fresh nullary predicate. Then, $\mathsf{MFA}(r)$ is the following rule:*

$$\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}.[\psi(\vec{x}, \vec{y}) \wedge \bigwedge_{y_i \in \vec{y}} [\mathsf{F}^i_r(y_i) \wedge \bigwedge_{x_j \in \vec{x}} \mathsf{S}(x_j, y_i)]]$$

For Σ a set of rules, $\mathsf{MFA}(\Sigma)$ is the smallest set that contains $\mathsf{MFA}(r)$ for each rule $r \in \Sigma$, rules (10)–(11)*, and rule* (12) *instantiated for each predicate F^i_r:*

$$\mathsf{S}(x_1, x_2) \rightarrow \mathsf{D}(x_1, x_2) \quad (10)$$

$$\mathsf{D}(x_1, x_2) \wedge \mathsf{S}(x_2, x_3) \rightarrow \mathsf{D}(x_1, x_3) \quad (11)$$

$$\mathsf{F}^i_r(x_1) \wedge \mathsf{D}(x_1, x_2) \wedge \mathsf{F}^i_r(x_2) \rightarrow \mathsf{C} \quad (12)$$

Set Σ is model-faithful acyclic *(MFA) w.r.t. an instance I if $I \cup \mathsf{MFA}(\Sigma) \not\models \mathsf{C}$; furthermore, Σ is* universally MFA[1] *if Σ is MFA w.r.t. I^*_Σ.*

Example 3. *Let Σ be the set of rules from Example 1. Then, $\mathsf{MFA}(r_1)$ and $\mathsf{MFA}(r_3)$ are given by* (13) *and* (14)*, resp.*

$$A(x_1) \rightarrow \exists y_1.R(x_1, y_1) \wedge B(y_1) \wedge \mathsf{F}^1_{r_1}(y_1) \wedge \mathsf{S}(x_1, y_1) \quad (13)$$

$$B(x_3) \rightarrow \exists y_2.R(x_3, y_2) \wedge C(y_2) \wedge \mathsf{F}^1_{r_3}(y_2) \wedge \mathsf{S}(x_3, y_2) \quad (14)$$

Then, $\mathsf{MFA}(\Sigma)$ consists of rules (8)*, rules* (13)–(14)*, rules* (10)–(11)*, and rule* (12) *instantiated for $\mathsf{F}^1_{r_1}$ and $\mathsf{F}^1_{r_3}$.*

[1] In the rest of this paper we typically omit 'universally'.

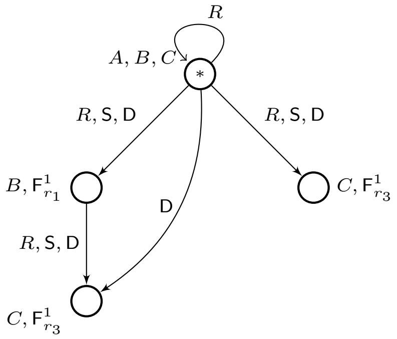

Figure 1: Example MFA

Set Σ is universally MFA: the interpretation shown in Figure 1 is a model of $I^*_\Sigma \cup \mathsf{MFA}(\Sigma)$ that does not satisfy C. $\Diamond$

MFA is defined as a semantic, rather than a syntactic condition, and entailment $I \cup \mathsf{MFA}(\Sigma) \not\models \mathsf{C}$ can be checked using sound and complete first-order calculus. In the following section we show that MFA is strictly more general than SWA. We next show that MFA characterises derivations of the skolem chase in a particular way.

Definition 4. *A term t is* cyclic *if some $f(\vec{s})$ is a subterm of t, and some $f(\vec{u})$ is a proper subterm of $f(\vec{s})$.*

Proposition 5. *A set of rules Σ is not MFA w.r.t. an instance I if and only if $I^\infty_{\mathsf{MFA}(\Sigma)}$ contains a cyclic term.*

Proof. Let $\Sigma' = \mathsf{MFA}(\Sigma)$, and let $I^0_{\Sigma'}, I^1_{\Sigma'}, \ldots$ be the chase sequence for I and Σ'. We next prove that the following claims hold for each integer k, as well as $k = \infty$.

1. $\mathsf{F}^i_r(t) \in I^k_{\Sigma'}$ for each term $t = f^i_r(\vec{t})$ occurring in $I^k_{\Sigma'}$; conversely, $\mathsf{F}^i_r(t) \in I^\infty_{\Sigma'}$ implies $t = f^i_r(\vec{t})$.
2. $\mathsf{S}(t', t) \in I^k_{\Sigma'}$ for each term $t = f^i_r(\vec{t})$ occurring in $I^k_{\Sigma'}$ and each term $t' \in \vec{t}$; conversely, $\mathsf{S}(t', t) \in I^\infty_{\Sigma'}$ implies $t = f^i_r(\vec{t})$ and $t' \in \vec{t}$.
3. $\mathsf{D}(t', t) \in I^{k+2}_{\Sigma'}$ for all terms t and t' occurring in $I^k_{\Sigma'}$ such that t' is a proper subterm of t; conversely, $\mathsf{D}(t', t) \in I^\infty_{\Sigma'}$ implies that t' is a proper subterm of t.

(Claims 1 and 2, the first part) The proof is by the induction on k. Set $I^0_{\Sigma'}$ does not contain functional terms, and so it clearly satisfies both claims. For the induction step, assume that both claims hold for $I^{k-1}_{\Sigma'}$ and consider $I^k_{\Sigma'}$. Since $I^{k-1}_{\Sigma'} \subseteq I^k_{\Sigma'}$, both claims clearly hold for each t that occurs in $I^{k-1}_{\Sigma'}$. Consider an arbitrary term $t = f^i_r(\vec{t})$ that does not occur in $I^{k-1}_{\Sigma'}$, and an arbitrary term $t' \in \vec{t}$. Clearly, t is introduced into $I^k_{\Sigma'}$ by an application of the skolemisation of $\mathsf{MFA}(r)$ for some rule $r \in \Sigma$. Since the head of $\mathsf{MFA}(r)$ contains atoms $\mathsf{F}^i_r(y_i)$ and $\mathsf{S}(x_j, y_i)$ for each $x_j \in \vec{x}$, we have $\mathsf{F}^i_r(t) \in I^k_{\Sigma'}$ and $\mathsf{S}(t', t) \in I^k_{\Sigma'}$ for each $t' \in \vec{t}$, and so we have $\mathsf{F}^i_r(t) \in I^\infty_{\Sigma'}$ and $\mathsf{S}(t', t) \in I^\infty_{\Sigma'}$ for each $t' \in \vec{t}$ as well. That these claims hold for $k = \infty$ is a straightforward consequence of the fact that $I^\infty_{\Sigma'} = \bigcup_k I^k_{\Sigma'}$.

(Claims 1 and 2, the second part) Predicate S and each predicate F^i_r occur in Σ' only in the head of some rule $\mathsf{MFA}(r)$, which clearly implies the claim.

(Claim 3, the first part for $k \neq \infty$) The proof is by induction on k. The base case holds vacuously since $I^0_{\Sigma'}$ does not contain functional terms. Assume now that the claim holds for some $k-1$, and consider an arbitrary term $t = f^i_r(\vec{t})$ occurring in $I^k_{\Sigma'}$ such that t' is a subterm of some $t_i \in \vec{t}$. By Claim 2, we have $\mathsf{S}(t_i, t) \in I^k_{\Sigma'}$; furthermore, t_i occurs in $I^{k-1}_{\Sigma'}$, so by the induction assumption we have $\mathsf{D}(t', t_i) \in I^{k+1}_{\Sigma'}$. Finally, the rules without functional terms are applied before the rules with functional terms; hence, by rule (10) we have $\mathsf{D}(t_i, t) \in I^{k+1}_{\Sigma'}$, and by rule (11) we have $\mathsf{D}(t', t) \in I^{k+2}_{\Sigma'}$, as required.

(Claim 3, the first part for $k = \infty$ and the second part) The 'proper subterm' relation is transitive, and rules (10) and (11) effectively define D as the transitive closure of S, which clearly implies this claim.

Assume now that $I^\infty_{\Sigma'}$ contains a cyclic term t. Then, some term $t_1 = f^i_r(\vec{s})$ is a subterm of t and some term $t_2 = f^i_r(\vec{u})$ is a proper subterm of t_1. By Claims 1 and 3, then we have $\{\mathsf{F}^i_r(t_2), \mathsf{D}(t_2, t_1), \mathsf{F}^i_r(t_1)\} \subseteq I^\infty_{\Sigma'}$. But then, since Σ' contains rule (12), we have $\mathsf{C} \in I^\infty_{\Sigma'}$, so Σ is not MFA. The proof of the converse claim is analogous. $\square$

This characterisation implies termination of skolem chase of MFA rules Σ in 2ExpTime. In particular, a term t derived by the skolem chase of $\Sigma' = \mathsf{MFA}(\Sigma)$ cannot be cyclic by Proposition 5; such t can then be seen as a tree with branching factor bounded by the maximum arity of a function symbol in $\mathsf{sk}(\Sigma')$ and with depth bounded by the number of function symbols in $\mathsf{sk}(\Sigma')$. The chase can thus generate at most doubly exponentially many different terms and atoms. The 2ExpTime bound already holds if the rules are WA; hence, CQ answering for MFA rules is not harder than for WA.

Proposition 6. *If a set of rules Σ is MFA w.r.t. an instance I, then the skolem chase for I and Σ terminates in double exponential time.*

Proof. Let $\Sigma' = \mathsf{MFA}(\Sigma)$, let c, f, and p be the number of constants, function symbols, and predicate symbols, respectively, occurring in $\mathsf{sk}(\Sigma')$, let ℓ be the maximum arity of a function symbol in $\mathsf{sk}(\Sigma')$, and let a be the maximum arity of a predicate symbol in $\mathsf{sk}(\Sigma')$. Consider now an arbitrary term t occurring in $I^\infty_{\Sigma'}$; clearly, t can be seen as a tree with branching factor ℓ containing constants in the leaf nodes and function symbols in the internal nodes; furthermore, since t is not cyclic, $\mathsf{dep}(t) \leq f$, so the tree has at most ℓ^f leaves and ℓ^f inner nodes. Thus, the number of different terms occurring in $I^\infty_{\Sigma'}$ is bounded by $\wp = (f)^{\ell^f} \cdot c^{\ell^f}$, and the number of different atoms in $I^\infty_{\Sigma'}$ is bounded by $p \cdot \wp^a$, which is clearly doubly exponential in Σ and I. Consequently, the size of $I^\infty_{\Sigma'}$ is at most doubly exponential in Σ and I. Furthermore, for an arbitrary set of facts I' and rule r, set $r(I')$ can be computed by examining all mappings of the variables in r to the terms occurring in I', which requires exponential time in the size of r and polynomial time in the size of I'. Consequently, $I^\infty_{\Sigma'}$ can be computed in time that is double exponential in I and Σ. Finally, it is straightforward to see that $I_{\Sigma}^{\infty} \subseteq I_{\Sigma'}^{\infty}$, so I_{Σ}^{∞} can be computed in double exponential time as well. □

Proposition 6 implies that answering a BCQ over MFA rules is in 2ExpTime; furthermore, Calì, Gottlob, and Pieris (2010) provide the matching lower bound for WA rules. We next prove that checking MFA w.r.t. a specific instance I is in 2ExpTime, and that checking universal MFA is 2ExpTime-hard. These results provide tight complexity bounds for both problems.

Theorem 7. *For Σ a set of rules, deciding whether Σ is MFA w.r.t. an instance I is in* 2ExpTime*, and deciding whether Σ is universally MFA is* 2ExpTime*-hard. Both results hold even if the arity of predicates in Σ is bounded.*

Proof. (Membership) Let $\Sigma' = \mathsf{MFA}(\Sigma)$, let $I_{\Sigma'}^{0}, I_{\Sigma'}^{1}, \ldots$ be the chase sequence for I and Σ', and let $\wp$, p, and a be as stated in the proof of Proposition 6. The number of different atoms that can be constructed from $\wp$ terms is bounded by $k = p \cdot \wp^{a}$; note that this is doubly exponential even if a is bounded. Let $k' = k + 3$ and consider $I_{\Sigma'}^{k}$. If $I_{\Sigma'}^{k} = I_{\Sigma'}^{k'}$, then $I_{\Sigma'}^{\infty} = I_{\Sigma'}^{k}$, so Σ is MFA if and only if $\mathsf{C} \in I_{\Sigma'}^{k}$. Otherwise, we have $I_{\Sigma'}^{k} \subsetneq I_{\Sigma'}^{k'}$; but then, $I_{\Sigma'}^{k+1}$ clearly contains at least one cyclic term $t = f_r^i(\vec{t})$ such that $t' = f_r^i(\vec{s})$ is a subterm of some $t_i \in \vec{t}$. Since $I_{\Sigma'}^{k}$ satisfies Claims 1–3 from the proof of Proposition 5, we have $\mathsf{S}(t_i, t) \in I_{\Sigma'}^{k+2}$; finally, by rule (12) and the fact that rules without functional terms are applied before rules with functional terms, we have $\mathsf{C} \in I_{\Sigma'}^{k'}$. But then, $\mathsf{C} \in I_{\Sigma'}^{\infty}$, so by Proposition 5 set Σ is not MFA.

(Hardness) We prove the claim by a reduction from the problem of checking $I \cup \Sigma \models Q$, where Σ is a *weakly acyclic* set of rules without equality and with predicates of bounded arity, I is an instance, and $Q = \exists \vec{y}.\xi(\vec{y})$ is a Boolean conjunctive query. Since Σ is weakly acyclic, Σ is SWA (Marnette 2009); by Theorem 15 it is MFA as well. Calì, Gottlob, and Pieris (2010) show that, for such I, Σ, and Q, deciding $I \cup \Sigma \models Q$ is 2ExpTime-complete.

Without loss of generality we assume that the rules in Σ do not contain constants: if a rule $r \in \Sigma$ contains a constant c, we can replace all occurrences of c in r with a fresh variable y_c, add an atom $O_c(y_c)$ to the body of r for O_c a fresh predicate, and add a fact $O_c(c)$ to I. It is straightforward to see that this transformation does not affect the answers of Q. We analogously assume w.l.o.g. that Q is free of constants.

To prove the claim of this theorem, we next construct a set of rules Ω such that $I \cup \Sigma \models Q$ if and only if Ω is not universally MFA. Before proceeding, we define some notation. For each n-ary predicate P, let $\hat{P}$ be a fresh $n+1$-ary predicate unique for P. For $A = P(\vec{t})$ an atom, let $\hat{A} = \hat{P}(\vec{t}, w)$ where w is a fresh variable not occurring in Σ or Q. For a conjunction of atoms φ, let $\hat{\varphi} = \bigwedge_{A \in \varphi} \hat{A}$. Finally, let $\hat{\Sigma}$ be the smallest set of rules containing $\hat{\varphi}(\vec{x}, \vec{z}, w) \rightarrow \exists \vec{y}.\hat{\psi}(\vec{x}, \vec{y}, w)$ for each rule $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}.\psi(\vec{x}, \vec{z})$ in Σ.

For c a constant, let v_c be a fresh variable unique for c, let $\vec{v}_c$ be the vector of all variables corresponding to constants in I, and let $\tilde{I}$ be defined as follows:

$$\tilde{I} = \bigwedge_{P(c_1, \ldots, c_k) \in I} \hat{P}(v_{c_1}, \ldots, v_{c_k}, w') \tag{15}$$

Let U be a fresh unary predicate, and let B be a fresh binary predicate. Rules r_Q and r_I are defined as shown in (16) and (17), respectively.

$$r_Q = \qquad \hat{\xi}(\vec{y}, w) \rightarrow U(w) \tag{16}$$

$$r_I = \qquad U(w) \rightarrow \exists w', \vec{v}_c.[B(w, w') \wedge \tilde{I}] \tag{17}$$

Finally, let $\Omega = \hat{\Sigma} \cup \{r_Q, r_I\}$.

The following property ($\diamondsuit$) follows immediately from the definition of r_Q, $\hat{\Sigma}$, and $\tilde{I}$: for an arbitrary substitution θ that maps $\vec{v}_c$ and w to distinct terms, it is the case that $\tilde{I}\theta \cup \hat{\Sigma} \cup \{r_Q\} \models U(w)\theta$ if and only if $I \cup \Sigma \models Q$. Furthermore, since Σ is MFA, $\hat{\Sigma}$ is clearly MFA as well, and so the chase for $\tilde{I}\theta \cup \hat{\Sigma}$ does not contain a cyclic term.

We next show that Ω is not universally MFA if and only if $I \cup \Sigma \models Q$. Let I' be the chase of I_{Ω}^{*} and Ω. Let f be the function symbol used to skolemise $\exists w'$ in (17); for each $v_c \in \vec{v}_c$ in (17), let g_c be the function symbols used to skolemise $\exists v_c$; and let θ be a substitution mapping w' to $f(*)$ and each variable $v_c \in \vec{v}_c$ to $g_c(*)$. Clearly, $U(*) \in I'$, so by rule (17) we have $\tilde{I}\theta \subseteq I'$; furthermore, $I' \setminus \tilde{I}\theta$ does not contain $g_c(*)$, and it can contain $f(*)$ only in an assertion involving predicate B which does not occur in Q. So, $\tilde{I}\theta \cup \hat{\Sigma} \cup \{r_Q\} \models U(f(*))$ if and only if $U(f(*)) \in I'$; by property ($\diamondsuit$), then $U(f(*)) \in I'$ if and only if $I \cup \Sigma \models Q$.

Assume now that $U(f(*)) \notin I'$. Then, the body of rule (17) is not satisfied, so I' does not contain term $f(f(*))$. Together with property ($\diamondsuit$), we conclude that I' does not contain a cyclic term, so Ω is MFA by Proposition 5.

Assume now that $U(f(*)) \in I'$. Then, the body of rule (17) is satisfied, so $B(f(*), f(f(*))) \in I'$; thus, I' contains a cyclic term, so Ω is not MFA by Proposition 5. □

The results of Theorem 7 are somewhat discouraging: using the existing conditions, acycilicity can typically be checked in PTime or in NP. We consider MFA to be an 'upper bound' of practically useful acyclicity conditions. We see two possibilities for improving these results. In the following section, we introduce an approximation of MFA that is easier to check; our evaluation shows that this condition often coincides with MFA in practice. Next, we show that the complexity is lower for rules of the following shape.

Definition 8. *A rule r of the form* (1) *is an* $\exists$-1 rule *if $\vec{y}$ is empty or $\vec{x}$ contains at most one variable.*

As we discuss in the following sections, $\exists$-1 rules capture (extensions of) Horn DLs. We next show that BCQ answering and MFA checking for $\exists$-1 rules is easier than for general rules. Intuitively, if Σ is MFA and contains only $\exists$-1 rules, then all functional terms in $\mathsf{sk}(\mathsf{MFA}(\Sigma))$ are unary and hence the number of different terms and atoms derivable by chase becomes exponentially bounded. The following theorem provides the upper bound; the lower bounds are given later on for a smaller class of rules that capture DLs.

Theorem 9. *Let Σ be a set of $\exists$-1 rules, and let I be an instance. Checking whether Σ is MFA w.r.t. I is in* ExpTime*. Furthermore, if Σ is MFA, then answering a BCQ over Σ and I is in* ExpTime *as well.*

Proof. Let $\Sigma' = \mathsf{MFA}(\Sigma)$. Since Σ contains only $\exists$-1 rules, Σ' also contains only $\exists$-1 rules; consequently, all functional terms in $\mathsf{sk}(\Sigma')$ are of arity 1. But then, the total number of different noncyclic terms is $\wp = c \cdot f^f$, where c is the number of constants in an instance and f is the number of function symbols in the rules. The total number of atoms is $p \cdot \wp^a$, where p is the number of predicates and a is the maximum arity of a predicate in Σ'. Note that this is exponential even if a is bounded. As in the proof of Proposition 6, we can now show that the either the chase for Σ' and I terminates or a cyclic term is derived in exponential time, which proves that the complexity of checking whether Σ is MFA w.r.t. I is in ExpTime.

Finally, since $I_\Sigma^\infty \subseteq I_{\Sigma'}^\infty$, if Σ is MFA, then I_Σ^∞ can be computed in exponential time, so a BCQ over Σ and I can be answered in ExpTime. □

Model-Summarising Acyclicity

The high cost of checking MFA of Σ is due to the fact that the arity of function symbols in $\mathsf{sk}(\Sigma)$ is unbounded, and that the depth of cyclic terms can be linear in Σ. To obtain an acyclicity condition that is easier to check, we must coarsen the structure used for the analysis of cycles. Thus, we introduce *model-summarising acyclicity*, which 'summarises' the models of Σ by reusing the same constant to satisfy an existential quantifier, instead of introducing deeper terms.

Definition 10. *Let S, D, and F_r^i be as in Definition 2; furthermore, for each rule $r = \varphi(\vec{x}, \vec{z}) \to \exists \vec{y}.\psi(\vec{x}, \vec{y})$ and each variable $y_i \in \vec{y}$, let c_r^i be a fresh constant unique for r and y_i. Then, $\mathsf{MSA}(r)$ is the following rule, where θ is the substitution that maps each variable $y_i \in \vec{y}$ to c_r^i:*

$$\varphi(\vec{x}, \vec{z}) \to \psi(\vec{x}, \vec{y})\theta \wedge \bigwedge_{y_i \in \vec{y}} \left[\mathsf{F}_r^i(y_i)\theta \wedge \bigwedge_{x_j \in \vec{x}} \mathsf{S}(x_j, y_i)\theta \right]$$

For Σ a set of rules, $\mathsf{MSA}(\Sigma)$ is the smallest set that contains $\mathsf{MSA}(r)$ for each rule $r \in \Sigma$, rules (10)–(11)*, and rule* (12) *instantiated for each predicate F_r^i. Set Σ is* model-summarising acyclic *(MSA) w.r.t. an instance I if $I \cup \mathsf{MSA}(\Sigma) \not\models \mathsf{C}$; furthermore, Σ is* universally MSA *if Σ is MSA w.r.t. I_Σ^*.*

Example 11. *Consider again the set of rules Σ in Example 1. $\mathsf{MSA}(r_1)$ and $\mathsf{MSA}(r_3)$ are given by the following rules* (18) *and* (19)*, respectively:*

$$A(x_1) \to R(x_1, c_{r_1}^1) \wedge B(c_{r_1}^1) \wedge \mathsf{F}_{r_1}^1(c_{r_1}^1) \wedge \mathsf{S}(x_1, c_{r_1}^1) \quad (18)$$

$$B(x_3) \to R(x_3, c_{r_3}^1) \wedge C(c_{r_3}^1) \wedge \mathsf{F}_{r_3}^1(c_{r_3}^1) \wedge \mathsf{S}(x_3, c_{r_3}^1) \quad (19)$$

Then, $\mathsf{MSA}(\Sigma)$ consists of rules (8)*, rules* (18)–(19)*, rules* (10)–(11)*, and rule* (12) *instantiated for $\mathsf{F}_{r_1}^1$ and $\mathsf{F}_{r_3}^1$.*

Set Σ is universally MSA: the interpretation shown in Figure 2 is a model of $I_\Sigma^* \cup \mathsf{MSA}(\Sigma)$ that does not satisfy C. ◊

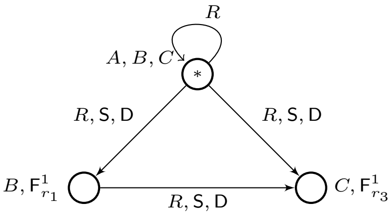

Figure 2: Example MSA

Note that $\mathsf{MSA}(\Sigma)$ is a set of datalog rules, so MSA can be checked using a datalog reasoner. This connection with datalog provides the upper complexity bound for checking MSA: the following theorem follows from the well known complexity results of checking entailment of a ground atom in a datalog program (Dantsin et al. 2001). The complexity of datalog reasoning is $O(n^v)$ where v is the maximum number of variables in a rule and n is the size of the set of facts that the rules are applied to; thus, checking MSA should be feasible if the rules in Σ are 'short' and so v is 'small'.

Theorem 12. *For Σ a set of rules, deciding whether Σ is MSA w.r.t. an instance I is in* ExpTime*, and deciding whether Σ is universally MSA is* ExpTime*-hard. The two problems are in co*NP *and co*NP*-hard, respectively, if the arity of the predicates in Σ is bounded.*

Proof. (Membership) Let $\Sigma' = \mathsf{MSA}(\Sigma)$. Note that the total number of atoms occurring in a chase of I and Σ' is $p \cdot c^a$, where p is the number of predicates, c is the number of constants, and a is the maximum arity of the predicates in Σ'; this number is clearly exponential if a is not bounded. The rest of the proof is the same as in Theorem 7. If a is bounded, then the number of atoms becomes polynomial; hence, to check whether $I \cup \Sigma' \models \mathsf{C}$, one can guess a proof for C as a polynomially long sequence of derived facts, and then check in polynomial time whether the proof is valid for I and Σ'. Thus, checking whether $I \cup \Sigma' \models \mathsf{C}$ is in NP; since $I \cup \Sigma' \not\models \mathsf{C}$ if and only if Σ is MSA w.r.t. I, we have that checking whether Σ is MSA w.r.t. I is in coNP.

(Hardness) Let Σ be a set of datalog rules, let I be an instance, and let Q be ground atom. Checking whether $I \cup \Sigma \models Q$ is ExpTime-complete in general (Dantsin et al. 2001). Furthermore, checking whether $I \models Q$ is already NP-hard even if the arity of predicates is bounded, so checking $I \cup \Sigma \models Q$ is NP-hard as well.

Let us define Ω as in the proof of Theorem 7. That Ω is not MSA if and only if $I \cup \Sigma \models Q$ can be shown as in Theorem 7, and we omit the proof for the sake of brevity. □

We finish this section by proving strict inclusion relationships between MFA, MSA, and SWA. In particular, Theorem 13 and Example 14 show that that MFA is strictly more general than MSA.

Theorem 13. *If a set of rules Σ is MSA (w.r.t. an instance I), then Σ is MFA (w.r.t. I) as well.*

Proof. Let $\Sigma_1 = \mathsf{MFA}(\Sigma)$ and let $\Sigma_2 = \mathsf{MSA}(\Sigma)$. Furthermore, let h be the mapping of ground terms to constants defined such that $h(t) = \mathsf{c}_r^i$ if t is of the form $f_r^i(\ldots)$, and $h(t) = t$ if t is a constant; for an atom $A = P(t_1, \ldots, t_n)$, let $h(A) = P(h(t_1), \ldots, h(t_n))$; and for an instance I, let $h(I) = \{h(A) \mid A \in I\}$. Finally, let $I^0_{\Sigma_1}, I^1_{\Sigma_1}, \ldots$ be the chase sequence for I and Σ_1, and let $I^0_{\Sigma_2}, I^1_{\Sigma_2}, \ldots$ be the chase sequence for I and Σ_2. Note that $\mathsf{sk}(\Sigma_2) = \Sigma_2$ differs from $\mathsf{sk}(\Sigma_1)$ only in that the former contains the constant c_r^i in place of each functional term $f_r^i(\vec{x})$. Thus, by a straightforward induction on i, one can show that $h(I^i_{\Sigma_1}) \subseteq I^i_{\Sigma_2}$ for each i; this implies $h(I^\infty_{\Sigma_1}) \subseteq I^\infty_{\Sigma_2}$. Consequently, $\mathsf{C} \notin I^\infty_{\Sigma_2}$ clearly implies $\mathsf{C} \notin I^\infty_{\Sigma_1}$; hence, if Σ is MSA, then Σ is MFA as well, as required. □

Example 14. *Let Σ be the set of rules* (20)–(23).

$$A(x) \to \exists y.R(x,y) \land B(y) \tag{20}$$

$$B(x) \to \exists y.S(x,y) \land T(y,x) \tag{21}$$

$$A(z) \land S(z,x) \to C(x) \tag{22}$$

$$C(z) \land T(z,x) \to A(x) \tag{23}$$

Σ is universally MFA, but not universally MSA. ◇

We now show that MSA is more general than SWA, and thus also more general than JA. The converse does not hold: the set Σ in Example 1 is MSA, but not SWA.

Theorem 15. *If a set of equality-free rules Σ is SWA, then Σ is universally MSA as well.*

Before proving the claim, we first recapitulate the definition of SWA (Marnette 2009). The definition uses the notion of a *place*—a pair $A|_i$ where A is an n-ary atom and $1 \leq i \leq n$. For arbitrary sets of places P and P', we write $P \sqsubseteq P'$ if, for each place $A|_i \in P$, a place $A'|_i \in P'$ and substitutions σ and σ' exist such that $A\sigma = A'\sigma'$. Let Σ be a set of equality-free rules (SWA is defined only for rules that do not contain equality). Let $r \in \Sigma$ be a rule of the form (1), and let w be a variable; then, $\mathsf{In}(r, w)$ (resp. $\mathsf{Out}'(r, w)$) is the set that contains each place $A|_i$ such that $A \in \varphi(\vec{x}, \vec{z})$ (resp. $A \in \psi(\vec{x}, \vec{y})$) and atom A contains variable w at position i; $\mathsf{Out}(r, w) = \{A\theta|_i \mid A|_i \in \mathsf{Out}'(r, w)\}$ where θ is the substitution used in the skolemisation of r; finally, $\mathsf{Move}_\Sigma(r, w)$ is the smallest set of places that contains $\mathsf{Out}(r, w)$ such that, for each $r' \in \Sigma$ and each variable w' occurring in r', if $\mathsf{In}(r', w') \sqsubseteq \mathsf{Move}_\Sigma(r, w)$, then $\mathsf{Out}(r', w') \subseteq \mathsf{Move}_\Sigma(r, w)$. A rule r *triggers* a rule r' in Σ if a variable x' occurring in both the body and the head of r' and an existentially quantified variable y occurring in the head of r exist such that $\mathsf{In}(r', x') \sqsubseteq \mathsf{Move}_\Sigma(r, y)$. Set Σ is SWA if its triggers relation is acyclic.

Proof of Theorem 15. SWA is applicable to Σ only if Σ contains the explicit axiomatisation of the equality predicate, so we assume this to be the case and consider $\approx$ to be a regular predicate. Let $\Sigma' = \mathsf{MSA}(\Sigma)$, let $I^0, I^1, \ldots$ be the chase sequence for I^*_Σ and Σ', and let I^∞ be the chase of I^*_Σ and Σ'. Furthermore, let ρ be the function that maps constants to themselves and that is defined on ground functional terms as $\rho(f_r^i(\vec{t})) = \mathsf{c}_r^i$. Finally, let $\rho(P(t_1, \ldots, t_n)) = P(\rho(t_1), \ldots, \rho(t_n))$.

We next prove the following property (♦): for each rule $r \in \Sigma$, each existentially quantified variable y_i occurring in r, each $P(\vec{t}) \in I^\infty$ where $P \notin \{\mathsf{S}, \mathsf{D}, \mathsf{C}\}$, and each $t_j \in \vec{t}$ such that $t_j = \mathsf{c}_r^i$ is the constant used to replace y_i in r, a substitution θ and a place $A|_j \in \mathsf{Move}_\Sigma(r, y_i)$ exist such that $P(\vec{t}) = \rho(A\theta)$. The proof is by induction on the length of the chase. Since $I^0 = I^*_\Sigma$ does not contain a constant of the form c_r^i, property (♦) holds vacuously for I^0. Assume now that property (♦) holds for some I^{k-1}, and consider an arbitrary rule $r \in \Sigma$, variable y_i, fact $P(\vec{t}) \in I^k \setminus I^{k-1}$ with $P \notin \{\mathsf{S}, \mathsf{D}, \mathsf{C}\}$, and $t_j \in \vec{t}$ such that $t_j = \mathsf{c}_r^i$. Fact $P(\vec{t})$ is derived in I^k from the head atom H of some rule $r^1 \in \mathsf{MSA}(\Sigma)$. Let σ be the substitution used in the rule application; clearly, we have $H\sigma = P(\vec{t})$. Furthermore, let $r^2 \in \Sigma$ be the rule that produces $r^1 \in \mathsf{MSA}(\Sigma)$, let r^3 be the skolemisation of r^2, and let H^3 be the head atom of r^3 that corresponds to H; clearly, we have $\rho(H^3\sigma) = P(\vec{t})$. Now if H contains c_r^i in position j, then $r = r^1$ since r^1 is the only rule that contains c_r^i; thus, $H^3|_j \in \mathsf{Out}(r, y_i) \subseteq \mathsf{Move}_\Sigma(r, y_i)$, so property (♦) holds. Otherwise, H contains at position j a universally quantified variable x such that $\sigma(x) = \mathsf{c}_r^i$. Let $B_1, \ldots, B_n$ be the body atoms of r^1 that contain x; clearly, $\{B_1\sigma, \ldots, B_n\sigma\} \subseteq I^{k-1}$. All these atoms satisfy the induction assumption, so for each $B_m \in \{B_1, \ldots, B_n\}$ and each ℓ such that B_m contains variable x at position ℓ, a place $B'_m|_\ell \in \mathsf{Move}_\Sigma(r, y_i)$ and substitution θ_m exist such that $B_m\sigma = \rho(B'_m\theta_m)$. Let σ' be the substitution obtained from σ by setting $\sigma'(w) = \theta_m(w)$ for each variable w for which $\theta_m(w)$ is a functional term; clearly, $B_m\sigma' = B'_m\theta_m$. But then, we have $\mathsf{In}(r^1, x) \sqsubseteq \mathsf{Move}_\Sigma(r, y_i)$; by the definition of Move_Σ we have $H^3|_j \in \mathsf{Move}_\Sigma(r, y_i)$, so property (♦) holds.

We additionally prove the following property (♡): if $\mathsf{S}(\mathsf{c}_r^i, \mathsf{c}_{r'}^{i'}) \in I^\infty$ for some i and i', then rule r triggers r'. Consider an arbitrary such fact, let y_i be the existentially quantified variable of r corresponding to c_r^i, and let k be the smallest integer such that $\mathsf{S}(\mathsf{c}_r^i, \mathsf{c}_{r'}^{i'}) \in I^k$. Clearly, $\mathsf{S}(\mathsf{c}_r^i, \mathsf{c}_{r'}^{i'})$ is derived in I^k from the head atom $\mathsf{S}(x, \mathsf{c}_{r'}^{i'})$ of rule r'. Let σ be the substitution used in the rule application; thus, $\sigma(x) = \mathsf{c}_r^i$. Let $B_1, \ldots, B_n$ be the body atoms of r that contain x; clearly, we have $\{B_1\sigma, \ldots, B_n\sigma\} \subseteq I^{k-1}$. All these atoms satisfy property (♦), so for each $B_m \in \{B_1, \ldots, B_n\}$ and each ℓ such that B_m contains variable x at position ℓ, a place $B'_m|_\ell \in \mathsf{Move}_\Sigma(r, y_i)$ and substitution θ_m exist such that $B_m\sigma = \rho(B'_m\theta_m)$. But then, as in the previous paragraph we have $\mathsf{In}(r', x) \sqsubseteq \mathsf{Move}_\Sigma(r, y_i)$, so r triggers r'.

Assume now that Σ is not MSA, so $\mathsf{C} \in I^\infty$; due to rules (12), we have $\{\mathsf{F}_r^i(t), \mathsf{D}(t, t'), \mathsf{F}_r^i(t')\} \subseteq I^\infty$ for some F_r^i. But then, since predicate F_r^i occurs in Σ' only in an atom $\mathsf{F}_r^i(\mathsf{c}_r^i)$, we have $t = t' = \mathsf{c}_r^i$. Finally, since D is axiomatised in Σ' as the transitive closure of S, clearly r triggers itself, and so Σ is not SWA. □

Handing Equality via Singularisation

JA and SWA can be applied to rules with equality provided that the rule set contains rules (2)–(5). In both cases, however, rules (2) and (5) lead to a cycle as soon as the rule set contains an existential quantifier. MFA and MSA are defined as entailment checks in first-order logic with equality, which effectively incorporates the rules of equality into these checks even if rules (2)–(5) are not explicitly stated. On rules with equality, MFA and MSA are slightly more general than JA and SWA, but they still fail to capture certain relevant cases, as the following example demonstrates.

Example 16. *Consider the following set of rules.*

$$A(x) \wedge B(x) \rightarrow \exists y.[R(x,y) \wedge B(y)] \tag{24}$$

$$R(z,x_1) \wedge R(z,x_2) \rightarrow x_1 \approx x_2 \tag{25}$$

On (24)–(25)*, skolem chase derives the following infinite set of facts; but then, by Proposition 5 this set of rules is not MFA; by Theorem 13, it is not MSA either.*

$$\begin{array}{cccc} R(*,f(*)) & B(f(*)) & * \approx f(*) & A(f(*)) \\ R(f(*),f(f(*))) & B(f(f(*))) & \dots & \end{array} \quad \Diamond$$

Example 16 shows that equalities between terms tend to proliferate during chase, which can lead to non-termination. Interestingly, the rules in Example 16 are WA. This is because WA is sufficient for termination of nonoblivious chase—a version of chase that expands an existential quantifier only if necessary. Already JA is more general than WA on rules without equality, so nonoblivious chase does not seem to provide an advantage over skolem chase w.r.t. termination on such rules; however, Example 16 shows that this is not the case for rules with equality.

Marnette (2009) proposed *singularisation* as a possible solution to this problem. The idea is to only partially axiomatise $\approx$ as being reflexive, symmetric, and transitive, but without the replacement property cf. rule (5). A set of rules Σ is modified in a way to take into account the lack of the replacement rules. This latter step is nondeterministic: there are many ways to modify Σ and, while some modifications will lead to chase termination, not all will do so.

We recapitulate the definition of singularisation. A *marking* M_r of a rule r of the form (1) is a mapping from each $w \in \vec{x} \cup \vec{z}$ to a single occurrence of w in φ; all other variable occurrences are *unmarked* and all constants are also unmarked. A *marking* M of a set of rules Σ contains exactly one marking M_r for each $r \in \Sigma$. The *singularisation* of Σ under M is the set $\mathsf{Sing}(\Sigma, M)$ containing

- for each $r \in \Sigma$, a rule obtained by replacing each unmarked occurrence of a body term t in r with a fresh variable z' and adding $t \approx z'$ to the body, and
- rules (2)–(4).

Note that $\mathsf{Sing}(\Sigma, M)$ is unique up to the renaming of the fresh variables. We identify the marked occurrence of a variable x in a rule as $\hat{x}$. The properties of singularisation can be summarised as follows: for an arbitrary set of rules Σ, a marking M for Σ, an instance I, and a fact $P(\vec{c})$, we have $I \cup \Sigma \cup \Sigma_{\approx} \models P(\vec{c})$ if and only if

$$I \cup \mathsf{Sing}(\Sigma, M) \models \exists \vec{y}.[P(\vec{y}) \wedge \bigwedge_{y_i \in \vec{y}} y_i \approx c_i].$$

Example 17. *Singularisation of the marked rule* (26) *produces rule* (27)*.*

$$A(\hat{x}) \wedge B(x) \wedge R(x,\hat{z}) \rightarrow C(x) \tag{26}$$

$$\begin{aligned} A(x) \wedge B(x_1) \wedge R(x_2,z) \wedge \\ x \approx x_1 \wedge x \approx x_2 \rightarrow C(x) \end{aligned} \tag{27}$$

Note that singularisation should be applied 'globally' to all rules, even to those without equality. $\Diamond$

The absence of rules (5) often allows the skolem chase to terminate on $\mathsf{Sing}(\Sigma, M)$; however, this may depend on the selected marking.

Example 18. *Rule* (24) *from Example 16 admits the following two markings:*

$$A(\hat{x}) \wedge B(x) \rightarrow \exists y.[R(x,y) \wedge B(y)] \tag{28}$$

$$A(x) \wedge B(\hat{x}) \rightarrow \exists y.[R(x,y) \wedge B(y)] \tag{29}$$

The skolem chase does not universally terminate for the singularisation obtained from (29)*; in contrast, the singularisation obtained from* (28) *is JA.* $\Diamond$

We use MFA$^\exists$ and MFA$^\forall$ to denote the classes of rule sets that are in MFA for *some* singularisation and for *all* singularisations, respectively; notions MSA$^\exists$, MSA$^\forall$, JA$^\exists$, and JA$^\forall$ are defined analogously. Clearly, $X^\forall \subseteq X^\exists$ for each $X \in \{\text{MFA}, \text{MSA}, \text{JA}\}$, and Example 18 shows this inclusion to be proper.

Theorem 19. *JA$^\forall$ = WA.*

Proof. (JA$^\forall \subseteq$ WA) We prove the contrapositive, so consider an arbitrary set of rules $\Sigma \notin$ WA. Then, the WA dependency graph contains a cycle through special edges. Assume w.l.o.g. that each variable occurs in at most one rule of Σ. By the definition of WA, each edge $p \rightarrow q$ in the dependency graph is justified by a rule r that has a universally quantified variable x at position p in some body atom of r. For each edge $p \rightarrow q$, let $x_{p \rightarrow q}$ denote one (arbitrary but fixed) such variable; we say that $x_{p \rightarrow q}$ *contributes to* $p \rightarrow q$.

We next show that a cycle through a special edge exists to which each variable contributes at most one edge. To this end, let $p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n \rightarrow p_{n+1} = p_1$ be an arbitrary cycle in the WA dependency graph. Let $p_i \rightarrow p_{i+1}$ and $p_j \rightarrow p_{j+1}$ be two edges (regular or special) such that $x_{p_i \rightarrow p_{i+1}} = x_{p_j \rightarrow p_{j+1}}$. If $p_i \rightarrow p_{i+1}$ is special, then a special edge $p_j \rightarrow p_{i+1}$ exists, and we obtain a shorter cycle by replacing the path between p_j and p_i with $p_j \rightarrow p_{i+1}$. If $p_j \rightarrow p_{j+1}$ is special, the situation is analogous. If neither $p_i \rightarrow p_{i+1}$ nor $p_j \rightarrow p_{j+1}$ is special, then regular edges $p_i \rightarrow p_{j+1}$ and $p_j \rightarrow p_{i+1}$ exist. If the path between p_{j+1} and p_i contains a special edge, we obtain a shorter cycle by replacing the path between p_i and p_{j+1} by $p_i \rightarrow p_{j+1}$. Otherwise, the path between p_{i+1} and p_j contains a special edge, and we obtain a shorter cycle by replacing the path between p_j and p_{i+1} with $p_j \rightarrow p_{i+1}$. This reduction of cycles can be applied recursively until we find a cycle of the required form.

Let $\Pi = p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n \rightarrow p_{n+1}$ be a cycle going through a special edge such that $p_{n+1} = p_1$ and each variable contributes at most one edge in the cycle, and let

$x_1 = x_{p_1 \to p_2}, \ldots, x_n = x_{p_n \to p_1}$ be the corresponding contributing variables. Let M be a marking for Σ where each x_i is marked in position p_i in some body atom of its rule. We claim that $\mathsf{Sing}(\Sigma, M)$ is not JA.

Consider a subpath $q_1 \to \ldots \to q_m$ ($m \geq 3$) of Π such that $q_1 \to q_2$ and $q_{m-1} \to q_m$ are (not necessarily distinct) special edges, and all other edges are regular. Let v be an existentially quantified variable at position q_2 that was used to justify the special edge $q_1 \to q_2$, and let w be an existentially quantified variable at position q_m that was used to justify the special edge $q_{m-1} \to q_m$. We claim that the JA dependency graph has an edge from v to w.

We show that $q_k \in \mathsf{Move}(v)$ for $2 \leq k \leq m-1$ by induction over k. For $k = 2$, $q_2 \in \mathsf{Pos}_H(v) \subseteq \mathsf{Move}(v)$ is immediate. For $k > 2$, note that $x_{q_{k-1} \to q_k}$ occurs only in the positions q_{k-1} and $\approx|_1$ in the body of the rule that justifies $q_{k-1} \to q_k$. By the induction hypothesis, $q_{k-1} \in \mathsf{Move}(v)$. By rule (2) of the equality theory, $\approx|_1 \in \mathsf{Move}(v)$. Thus, we have $q_k \in \mathsf{Move}(v)$.

Consequently, we have $q_{m-1} \in \mathsf{Move}(v)$. Since we have $\approx|_1 \in \mathsf{Move}(v)$, an edge $v \to w$ exists in the JA dependency graph. For all subpaths of the form $q_1 \to \ldots \to q_m$ one can find analogous edges, so the JA dependency graph is cyclic.

(JA$^\forall \supseteq$ WA) Assume that $\Sigma \notin$ JA$^\forall$. Then a marking M for Σ exists such that $\mathsf{Sing}(\Sigma, M)$ is not JA. Consider some existentially quantified variable v. For each position $p \in \mathsf{Move}(v)$ (w.r.t. $\mathsf{Sing}(\Sigma, M)$) where p does not involve the $\approx$ predicate, there is a path of regular edges $p_1 \to \ldots \to p_n = p$ in the WA dependency graph of Σ such that v occurs on position p_1; this property ($*$) can be easily shown by induction over the construction of $\mathsf{Move}(v)$, and we omit the details for the sake of brevity. Thus, by ($*$) and the definition of the JA dependency graph, for each edge $v \to w$ in the JA dependency graph, a path $p_1 \to \ldots \to p_n \to p_{n+1}$ exists in the WA dependency graph of Σ where $p_n \to p_{n+1}$ is a special edge and all other edges are regular. Clearly, we have $\Sigma \notin$ WA. □

Checking all possible markings may be infeasible: the number of candidates is exponential in the total number of variables that occur more than once in a rule body. Theorem 19 shows that JA$^\forall$ can be decided using WA. For the other cases, the following simple observation shows how to reduce the number of markings.

Proposition 20. *Let M and M' be markings for Σ that agree on all variables that occur in both body and head, but not necessarily on the variables that occur only in the body of a rule. Then* $\mathsf{Sing}(\Sigma, M)$ *is JA/MSA/MFA if and only if* $\mathsf{Sing}(\Sigma, M')$ *is JA/MSA/MFA.*

Despite this optimisation, the number of markings to check can still be exponential; hence, we next describe a useful approximation. Let $\mathsf{Sing}_\cup(\Sigma) = \bigcup_{M \in \mathcal{M}} \mathsf{Sing}(\Sigma, M)$, where $\mathcal{M}$ is a set of all markings for Σ that agree on all variables occurring only in the body of a rule in Σ. By Proposition 20, it is irrelevant how the markings of body variables are defined in $\mathcal{M}$. Let MFA$^\cup$ be the class containing rule sets Σ for which $\mathsf{Sing}_\cup(\Sigma)$ is in MFA; MSA$^\cup$ and JA$^\cup$ are defined analogously. As the following theorem shows, $\mathsf{Sing}_\cup(\Sigma)$ provides a 'lower bound' on the result attainable via singularisation.

Theorem 21. *For each $X \in \{MFA, MSA, JA\}$, we have that $X^\cup \subseteq X^\forall$. The size of $\bigcup_{M \in \mathcal{M}}$* $\mathsf{Sing}(\Sigma, M)$ *is exponential in the maximal number of variables that occur more than once in the body of any one rule in Σ, and it is linear in the number of rules in Σ.*

Proof. The first claim follows from the fact that all considered notions of acyclicity are monotone in the sense that every subset of an acyclic rule set is also acyclic. The second claim follows from the fact that each rule r in Σ occurs k times in $\bigcup_{M \in \mathcal{M}} \mathsf{Sing}(\Sigma, M)$, where k is the number of distinct markings of r. □

This result is of particular interest when dealing with rules that are obtained from DLs, where each rule has at most one variable that occurs in the head as well as multiple times in the body. On such rule sets, the size of $\mathsf{Sing}_\cup(\Sigma)$ is linear in the size of Σ. For the general case, we can obtain the same complexity bounds despite the exponential increase in the number of rules.

Theorem 22. *Deciding whether Σ is in MFA$^\cup$ (MFA$^\exists$, MFA$^\forall$) is* 2ExpTime*-complete. Deciding whether Σ is in MSA$^\cup$ (MSA$^\exists$, MSA$^\forall$) is* ExpTime*-complete.*

Proof. If Σ contains no equality, it is easy to see that Σ is in MFA$^\cup$ (MFA$^\exists$, MFA$^\forall$) iff it is in MFA. The same can be observed for MSA. Hardness thus follows from Theorem 7 and Theorem 12.

For membership, we first consider the cases of MFA$^\exists$, MFA$^\forall$, MSA$^\exists$, and MSA$^\forall$. Each of these properties can be decided by considering all of the at most exponentially many markings. Since $\mathsf{Sing}(\Sigma, M)$ is linear in the size of Σ, the property can be checked for each marking in 2ExpTime (for MFA; Theorem 7) and ExpTime (for MSA; Theorem 12). This yields the required bound since an exponential *factor* is not significant for the considered complexity classes.

For the case of MFA$^\cup$ and MSA$^\cup$, membership follows by observing that the membership of MFA and MSA in 2ExpTime and ExpTime, respectively, is obtained from the according bound of doubly/singly exponentially many ground facts that can potentially be derived before the property can be decided. While $\mathsf{Sing}_\cup(\Sigma)$ is exponentially larger than Σ, the maximal number of relevant ground facts is still the same since no new predicates or constant symbols are introduced. The increased number of rules leads to an exponential increase of the time to check applicability of all rules in each of the doubly/singly exponentially many steps. As above, this exponential factor does not affect membership in 2ExpTime/ExpTime. □

Acyclicity of DL Ontologies

We now consider applying acyclicity conditions to DL ontologies. DLs are KR formalisms that underpin the Web Ontology Language (OWL). DL ontologies are constructed from *atomic concepts* (i.e., unary predicates), *atomic roles* (i.e., binary predicates), and *individuals* (i.e., constants). Special atomic concepts $\top$ and $\bot$ denote universal truth and falsehood, respectively. For R an atomic role, R^- is an *inverse role*; inverse roles can be used in atoms, and $R^-(t_1, t_2)$ is an abbreviation for $R(t_2, t_1)$. A *role* is an atomic or an inverse role. DLs provide a rich set of *constructors* for building *concepts* (first-order formulae with one free variable) from atomic concepts and roles. DL ontologies consist of *axioms* about concepts and roles; these correspond to first-order sentences. For simplicity, we consider only normalised ontologies, where concepts are not nested. This is w.l.o.g. as each ontology can be normalised in linear time, and the normalised ontology is a conservative extension of the original one. In this paper, we consider only *Horn* DLs; ontologies in such DLs have at most one minimal Herbrand model, which is a prerequisite for materialisation-based reasoning—the main motivation for applying acyclicity to DLs.

A normalised Horn-$\mathcal{SRIQ}$ *TBox* $\mathcal{T}$ consists of axioms shown on the left-hand side of Table 1; in the table, A, B, and C are atomic concepts (including possibly $\top$ and $\bot$), R, S, T are (not necessarily atomic) roles, and n is a positive integer. To guarantee decidability of reasoning, $\mathcal{T}$ must satisfy certain *global* conditions (Kutz, Horrocks, and Sattler 2006), which we omit for brevity. Roughly speaking, only so-called *simple* roles are allowed to occur in axioms of Type 2, and axioms of Type 6 must be *regular* according to a particular condition; the latter condition ensures that axioms of Type 6 can be represented using a nondeterministic finite automaton. Apart from Horn-$\mathcal{SRIQ}$, we also consider Horn-$\mathcal{SRI}$ TBoxes, which do not contain rules of Type 2, as well as Horn-$\mathcal{SHIQ}$ TBoxes, where $R = S = T$ in all rules of Type 6; all Horn-$\mathcal{SHIQ}$ TBoxes are regular.

Each Horn-$\mathcal{SRIQ}$ axiom corresponds to an existential rule as shown in Table 1. A minor difference is that axioms in Table 1 can contain $\bot$ in the head, which can make a TBox $\mathcal{T}$ unsatisfiable w.r.t. an instance I. This can be handled by considering $\bot$ to be just another atomic concept, without special meaning. Technically, this ensures that $I \cup \mathcal{T}$ is satisfiable in the model constructed by the skolem chase; however, we consider $I \cup \mathcal{T}$ to be unsatisfiable if $I \cup \mathcal{T} \models \exists y.\bot(y)$. We consider a substitution θ to be an answer to a CQ $Q(\vec{x})$ w.r.t. a $\mathcal{T}$ and I if $I \cup \mathcal{T} \models \exists y.\bot(y)$ or $\mathcal{T} \cup I \models Q(\vec{x})\theta$. Due to this close correspondence between DL axioms and existential rules, in the rest of this paper we identify a TBox $\mathcal{T}$ with the corresponding set of rules.

We next investigate the complexity of BCQ answering over acyclic DL TBoxes. For the membership, note that all rules in Table 1 are $\exists$-1 rules; thus, Theorem 9 gives us an ExpTime upper bound. We next prove a matching lower bound for WA Horn-$\mathcal{SRI}$ rules. Intuitively, axioms of Type 6 allow us to axiomatise non-tree-like structures; although regularity ensures that axioms of Type 6 can be represented by a nondeterministic finite automaton, this automaton can be exponential, which may require one to examine all nodes in an exponential model of a Horn-$\mathcal{SRI}$ TBox. Furthermore, by Theorem 9, if we extend acyclic Horn-$\mathcal{SRIQ}$ rules Σ_1 with arbitrary SWRL rules Σ_2, reasoning stays ExpTime-complete, provided that $\Sigma_1 \cup \Sigma_2$ is acyclic; this is in contrast to general TBoxes for which SWRL extensions lead to undecidability. Thus, applications that need expressivity beyond what is available in OWL can benefit from the required expressivity without running into undecidability as long as the resulting ontology is acyclic.

Theorem 23. *Let $\mathcal{T}$ be a WA Horn-$\mathcal{SRI}$ TBox, let I be an instance, and let F be a fact. Then, checking whether $I \cup \mathcal{T} \models F$ is* ExpTime*-hard.*

Proof. Let $\mathcal{M} = (\mathcal{S}, \mathcal{Q}, \delta, Q_0, Q_a)$ be a deterministic Turing machine, where $\mathcal{S}$ is a finite set of symbols, $\mathcal{Q}$ is a finite set of states, $\delta : \mathcal{Q} \times \mathcal{S} \to \mathcal{Q} \times \mathcal{S} \times \{\leftarrow, \rightarrow\}$ is a transition function, $Q_0 \in \mathcal{Q}$ is the initial state, and Q_a the accepting state. Furthermore, assume that an integer k exists such that $\mathcal{M}$ halts on each input of length n in time 2^{n^k}. For arbitrary input $S_{i_1}, \ldots, S_{i_n}$, we construct an MFA set of Horn-$\mathcal{SRI}$ rules $\mathcal{T}$ and an instance I such that $I \cup \mathcal{T} \models Q_a(a)$ if and only if $\mathcal{M}$ accepts the input. To simplify the presentation, we will use a slightly more general syntax for the rules in $\mathcal{T}$ than what is allowed in Table 1; however, all of our rules can be brought into the required form by renaming parts of the rules with fresh predicates.

Let $\ell = n^k$; since k is a constant, ℓ is polynomial in n. Our construction uses a unary predicate for each symbol and state; for simplicity, we do not distinguish between the predicate and the symbol/state. In addition, the construction also uses binary predicates L_i, R_i, T_i, U_i, D_i, H_i, and V_i for $1 \leq i \leq \ell$, unary predicates A_i and B_i for $0 \leq i \leq \ell$, and unary predicates $O_1, \ldots, O_{n+1}$, N_1, and N_2. Instance I contains only the fact $A_0(a)$. We next present the rules of $\mathcal{T}$. Set $\mathcal{T}$ will contain only Horn rules without empty heads, so it will be satisfiable in a minimal Herbrand model. For readability, we will break $\mathcal{T}$ into parts and prove for each part various facts about this minimal Herbrand model.

The first part of $\mathcal{T}$ contains rules (30)–(32) for each $i > 0$, and rule (33) for each $i > 1$.

$$A_{i-1}(x) \to \exists y.[L_i(x, y) \wedge A_i(y)] \quad (30)$$

$$A_{i-1}(x) \to \exists y.[R_i(x, y) \wedge A_i(y)] \quad (31)$$

$$R_i(z, x) \wedge L_i(z, x') \to T_i(x, x') \quad (32)$$

$$L_i(z, x) \wedge T_{i-1}(z, z') \wedge R_i(z', x') \to T_i(x, x') \quad (33)$$

On I, these rules axiomatise existence of a triangular structure in the top part of Figure 3 containing T_i links.

The second part of $\mathcal{T}$ contains rule (34), rules (35)–(37) for each $i > 0$, and rule (38) for each $i > 1$.

$$A_\ell(x) \to B_0(x) \quad (34)$$

$$B_{i-1}(x) \to \exists y.[U_i(x, y) \wedge B_i(y)] \quad (35)$$

$$B_{i-1}(x) \to \exists y.[D_i(x, y) \wedge B_i(y)] \quad (36)$$

$$U_i(z, x) \wedge D_i(z, x') \to V_i(x, x') \quad (37)$$

$$D_i(z, x) \wedge V_{i-1}(z, z') \wedge U_i(z', x') \to V_i(x, x') \quad (38)$$

These rules axiomatise existence of triangular structures in the bottom part of Figure 3 containing V_i links.

The third part of $\mathcal{T}$ contains rule (39), and rules (40) and (41) for each $i > 0$.

$$T_\ell(x, x') \to H_0(x, x') \quad (39)$$

$$U_i(z, x) \wedge H_{i-1}(z, z') \wedge U_i(z', x') \to H_i(x, x') \quad (40)$$

$$D_i(z, x) \wedge H_{i-1}(z, z') \wedge D_i(z', x') \to H_i(x, x') \quad (41)$$

1.	$A \sqsubseteq \exists R.B$	$A(x) \to \exists y.[R(x,y) \wedge B(y)]$
2.	$A \sqsubseteq \leq 1\, R.B$	$A(z) \wedge R(z,x_1) \wedge B(x_1) \wedge R(z,x_2) \wedge B(x_2) \to x_1 \approx x_2$
3.	$A \sqcap B \sqsubseteq C$	$A(x) \wedge B(x) \to C(x)$
4.	$A \sqsubseteq \forall R.B$	$A(z) \wedge R(z,x) \to B(x)$
5.	$R \sqsubseteq S$	$R(x_1,x_2) \to S(x_1,x_2)$
6.	$R \circ S \sqsubseteq T$	$R(x_1,z) \wedge S(z,x_2) \to T(x_1,x_2)$

Table 1: Axioms of normalised Horn-$\mathcal{SRIQ}$ ontologies and corresponding rules

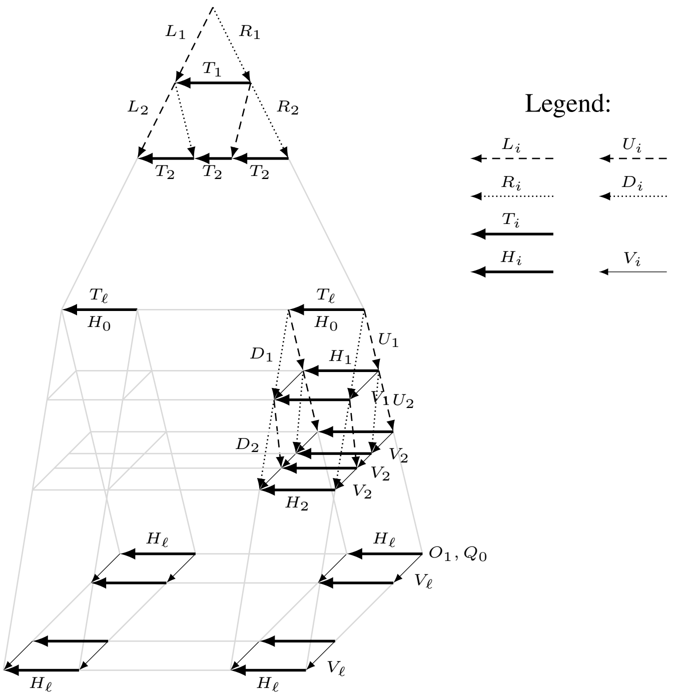

Figure 3: Grid Model of $\mathcal{T}$

These rules axiomatise existence of H_i links, which with V_i links form a grid of size $2^i \times 2^i$ shown in Figure 3.

In the rest of this proof we abbreviate conjunctions of the form $R_1(x_0, x_1) \wedge \ldots \wedge R_\ell(x_{\ell-1}, x_\ell)$ as $R^\ell(x_0, x_\ell)$, and $U_1(x_0, x_1) \wedge \ldots \wedge U_\ell(x_{\ell-1}, x_\ell)$ as $U^\ell(x_0, x_\ell)$

The fourth part of $\mathcal{T}$ contains rule (42), rules (43) and (44) for each $1 \leq j \leq n$, and rules (45)–(46), where $S_\sqcup$ is the empty tape symbol. Remember that the input to $\mathcal{M}$ is given as $S_{i_1}, \ldots, S_{i_n}$.

$$A_0(z) \wedge R^\ell(z, z') \wedge U^\ell(z', x) \to O_1(x) \wedge Q_0(x) \quad (42)$$

$$O_j(z) \wedge V_\ell(z, x) \to O_{j+1}(x) \quad (43)$$

$$O_j(x) \to S_{i_j} \quad (44)$$

$$O_{n+1}(z) \wedge V_\ell(z, x) \to O_{n+1}(x) \quad (45)$$

$$O_{n+1}(x) \to S_\sqcup(x) \quad (46)$$

Rule (42) labels the grid origin and sets the initial state as shown in Figure 3. Rules (43) ensure that the n subsequent nodes are labelled with $O_2, \ldots, O_{n+1}$, and rule (45) propagates O_{n+1} to the rest of the V_ℓ-chain. Finally, rules (44) and (46) ensure that nodes labelled with O_j are also labelled with S_{i_j}, and that nodes labeled with O_{n+1} are labeled with $S_\sqcup$. Thus, this part of $\mathcal{T}$ ensures that the right-most V_ℓ-chain in the grid contains the initial state of the tape of $\mathcal{M}$.

The fifth part of $\mathcal{T}$ contains rules (47)–(48) for each state $Q_k \in \mathcal{Q}$, and rules (49)–(50). These rules essentially ensure that all nodes before and after a node labelled with some state $Q_k \in \mathcal{Q}$ are labeled with N_1 and N_2, respectively, thus indicating that the head is not above the node.

$$Q_k(z) \wedge V_\ell(x, z) \to N_1(x) \quad (47)$$

$$Q_k(z) \wedge V_\ell(z, x) \to N_2(x) \quad (48)$$

$$N_1(z) \wedge V_\ell(x, z) \to N_1(x) \quad (49)$$

$$N_2(z) \wedge V_\ell(z, x) \to N_2(x) \quad (50)$$

The sixth part of $\mathcal{T}$ contains rules (51)–(52) instantiated for each symbol $S_k \in \mathcal{S}$; these rules ensure that the contents of the tape is copied for all nodes not containing the head.

$$N_1(z) \wedge S_k(z) \wedge H_\ell(z, x) \to S_k(x) \quad (51)$$

$$N_2(z) \wedge S_k(z) \wedge H_\ell(z, x) \to S_k(x) \quad (52)$$

The seventh part of $\mathcal{T}$ contains rules (53)–(54) instantiated for each symbol $S_k \in \mathcal{S}$ and each state $Q_k \in \mathcal{Q}$ such that $\delta(Q_k, S_k) = (Q_{k'}, S_{k'}, \leftarrow)$. These rules encode moves of $\mathcal{M}$ where the head moves up.

$$Q_k(z) \wedge S_k(z) \wedge H_\ell(z,x) \to S_{k'}(x) \quad (53)$$

$$Q_k(z) \wedge S_k(z) \wedge H_\ell(z,z') \wedge V_\ell(x,z') \to Q_{k'}(x) \quad (54)$$

The eighth part of $\mathcal{T}$ contains rules (55)–(56) instantiated for each symbol $S_k \in \mathcal{S}$ and each state $Q_k \in \mathcal{Q}$ such that $\delta(Q_k, S_k) = (Q_{k'}, S_{k'}, \rightarrow)$. These rules encode moves of $\mathcal{M}$ where the head moves down.

$$Q_k(z) \wedge S_k(z) \wedge H_\ell(z,x) \to S_{k'}(x) \quad (55)$$

$$Q_k(z) \wedge S_k(z) \wedge H_\ell(z,z') \wedge V_\ell(z',x) \to Q_{k'}(x) \quad (56)$$

The ninth part of $\mathcal{T}$ contains rules (57)–(60) for each $1 \leq i \leq \ell$; these rules simply ensure that acceptance is propagated back to the root of the upper tree.

$$Q_a(z) \wedge U_i(x,z) \to Q_a(x) \quad (57)$$

$$Q_a(z) \wedge D_i(x,z) \to Q_a(x) \quad (58)$$

$$Q_a(z) \wedge L_i(x,z) \to Q_a(x) \quad (59)$$

$$Q_a(z) \wedge R_i(x,z) \to Q_a(x) \quad (60)$$

The above discussion shows that labelling of the nodes in the grid shown in Figure 3 simulates the execution of $\mathcal{M}$ on input $S_{i_1}, \ldots, S_{i_n}$, where the contents of the tape at some time instant is represented by a V_ℓ-chain, and H_ℓ-links connect tape cells at successive time instants. Thus, $I \cup \mathcal{T} \models Q_a(a)$ if and only if $\mathcal{M}$ accepts $S_{i_1}, \ldots, S_{i_n}$ in time 2^ℓ. It is straightforward to see that $\mathcal{T}$ is WA, so the claim of this theorem holds. □

Note that Theorem 23 applies to Horn-$\mathcal{SRI}$ and thus does not rely on a particular treatment of equality.

The proof of Theorem 23 can be adapted to obtain the lower bound for checking MFA of Horn-$\mathcal{SRI}$ rules.

Proposition 24. *Checking whether a Horn-$\mathcal{SRI}$ TBox is universally MFA is* EXPTIME*-hard.*

Proof. Let $\mathcal{M}$ be an arbitrary deterministic Turing machine and let $S_{i_1}, \ldots, S_{i_n}$ be an input string on which $\mathcal{M}$ terminates in time 2^{n^k}. For such $\mathcal{M}$ and $S_{i_1}, \ldots, S_{i_n}$, let $\mathcal{T}$ be as in the proof of Theorem 23. Furthermore, let $\mathcal{T}'$ be the extension of $\mathcal{T}$ with the following rule:

$$Q_a(x) \to \exists y.[B(x,y) \wedge A_0(y)] \quad (61)$$

By an argument analogous to the one used in the proof of Theorem 7, one can see that $\mathcal{T}'$ is not universally MFA if and only if $\mathcal{M}$ accepts $S_{i_1}, \ldots, S_{i_n}$. □

As we show next, however, the complexity of query answering drops to PSPACE for MFA Horn-$\mathcal{SHIQ}$ ontologies. In contrast, checking entailment of a single fact is EXPTIME-hard in the general (i.e., not acyclic) case (Krötzsch, Rudolph, and Hitzler 2012). This drop in complexity is due to the fact that, if $R = S = T$ in all rules of Type 6, the automaton describing roles is of polynomial size, and the elimination of role inclusions by Demri and de Nivelle (2005) becomes polynomial. Thus, although acyclic TBoxes can axiomatise existence of polynomially deep and exponentially large structures, these structures are tree-like, which allows us to explore the structures one path at a time using the well-known tracing technique (Baader et al. 2007). The main difficulty in the membership proof of the following theorem is due to the fact that queries can contain transitive roles, so one cannot roll a query up into a concept. Since the TBox is Horn, however, one can guess places in the model that the query maps to. Given one such guess, one can ground the query and check entailment of each ground query atom individually, while taking transitive roles into account. Furthermore, note that PSPACE-hardness proof of concept satisfiability checking by Baader et al. (2007) is not applicable to Horn ontologies since it uses disjunctive concepts. Nonetheless, PSPACE-hardness can be proved by a reduction from checking QBF validity.

Theorem 25. *Let $\mathcal{T}$ be Horn-$\mathcal{SHIQ}$ TBox, let I be an instance such that $\mathcal{T}$ is MFA w.r.t. I, and let Q be a BCQ. Then, deciding $I \cup \mathcal{T} \models Q$ is* PSPACE*-complete.*

Proof (Membership). Let us assume that BCQ Q is of the form $Q = \exists \vec{y}.B_1 \wedge \ldots \wedge B_n$. Furthermore, let $\Upsilon = \mathsf{sk}(\mathcal{T})$, let f be the number of function symbols in Υ, and let c be the number of constants in I. Since $\bot$ is just a regular atomic concept, $I \cup \mathcal{T}$ is always satisfiable in the chase $I_\mathcal{T}^\infty$ of I and $\mathcal{T}$. Furthermore, $I \cup \mathcal{T} \models Q$ if and only if a substitution from the variables in $\vec{y}$ to the terms in $I_\mathcal{T}^\infty$ exists such that $B_i\theta \in I_\mathcal{T}^\infty$ for each $1 \leq i \leq n$; the latter clearly holds if and only if $I \cup \Upsilon \models B_i\theta$. As shown in the proof of Theorem (9), each term in $I_\mathcal{T}^\infty$ is of the form $g_1(...g_n(a)\ldots)$, where $n \leq f$. Thus, the first step in deciding $I \cup \mathcal{T} \models Q$ is to examine all possible θ and then check $I \cup \Upsilon \models B_i\theta$; this can clearly be done using a deterministic Turing machine that uses polynomial space to store θ.

If $B_i\theta$ is of the form $C(t)$, then let $\Upsilon' = \Upsilon$, and let $D = C$. Alternatively, if $B_i\theta$ is of the form $R(t', t)$, then let Υ' be Υ extended with the following rules, where D and E are fresh concepts not occurring in Υ and I:

$$\to E(t') \quad (62)$$

$$E(z) \wedge R(z,x) \to D(x) \quad (63)$$

It is straightforward to see that $I \cup \Upsilon \models D(t)$ if and only if $I \cup \Upsilon' \models D(t)$. Let Υ'' be obtained from Υ' by removing each rule of the form

$$R(x_1,z) \wedge R(z,x_2) \to R(x_1,x_2) \quad (64)$$

and then replacing each rule of the form

$$A(z) \wedge R(z,x) \to B(x) \quad (65)$$

with the following rules, where $Q_{A,R,B}$ is a fresh concept unique for A, R, and B:

$$A(z) \wedge R(z,x) \to Q_{A,R,B}(x) \quad (66)$$

$$Q_{A,R,B}(z) \wedge R(z,x) \to Q_{A,R,B}(x) \quad (67)$$

$$Q_{A,R,B}(x) \to B(x) \quad (68)$$

This corresponds to the well-known elimination of transitivity (Demri and de Nivelle 2005), so for the sake of brevity we omit the proof that $I \cup \Upsilon' \models D(t)$ if and only if $I \cup \Upsilon'' \models D(t)$.

Let Ξ be Υ'' extended with the equality axioms (3) and (5). Since $\approx$ does not occur in the body of the rules in Υ'', we have that $I \cup \Upsilon'' \not\models D(t)$ if and only if $I \cup \Xi \not\models D(t)$. Let I_Ξ^∞ be the chase for I and Ξ; then $I \cup \Xi \not\models D(t)$ if and only if $D(t) \notin I_\Xi^\infty$. Note that Ξ contains rules of Types 1–5 from Table 1, rules (3) and (5), and possibly rules of the form $\rightarrow Q_1(t_1)$ and $Q_2(t_2) \rightarrow \mathsf{false}$. This can be used in the same way as by Motik, Shearer, and Horrocks (2009) to show that each assertion in I_Ξ^∞ is of one of the following forms, where a and b are constants, and t is a term consisting of possibly zero unary function symbols:

- $C(t)$,
- $R(a,b)$, $R(a,f(b))$, $R(f(b),a)$, $R(t,f(t))$, $R(f(t),t)$, or
- $t \approx f(g(t))$, $f(t) \approx g(t)$, $a \approx b$, $a \approx f(b)$, or an equality symmetric to these ones.

The proof is by induction on the length of the chase sequence for I and Ξ, and the claim follows straightforwardly from the I_Ξ^∞ form of rules of Types 1–5.

Let $f_1, \dots, f_n$ be all function symbols occurring in Ξ. Furthermore, we say that x is the *central variable* in a rule of Type 1 or 3, and that z *central variable* in a rule of Type 2 or 4. W.l.o.g. we assume that the antecedent of a rule of Type 5 does not contain inverse roles; then, x_1 is the *central variable* of a rule of Type 5. Finally, in the equality replacement rules (5), the *central variable* is the variable being replaced.

Clearly, $D(t) \notin I_\Xi^\infty$ if and only if a Herbrand interpretation J exists in which all assertions are of the form mentioned above, such that $I \subseteq J$, $I_\Xi^\infty \subseteq J$, $J \models \Xi$, and $D(t) \notin J$. We next show how to check the existence of such J using a nondeterministic Turing machine that runs in polynomial space.

We first guess an interpretation $J_0 \subseteq I$ for the constants in I, and we check whether all rules in Ξ not of Type 1 are satisfied in J_0. If that is the case, we consider each constant c in J_0 and call the following procedure for $s = c$ and $i = 1$:

1. If $i = n+1$ return *true*.
2. Guess an interpretation J^i consisting of assertions of type mentioned above and that involves terms occurring in J^j with $j < i$, and $f_1(s), \dots, f_n(s)$.
3. If $D(t) \in J^i$, return *false*.
4. Check whether J^i coincides with each J^j, $j < i$ on the common terms; if not, return *false*.
5. Check whether the equality symmetry rule (5) is satisfied in J^i; if not, return *false*.
6. Check whether $J^i \cup J^{i-1} \cup \dots \cup J^0$ satisfies each rule in Ξ if the central variable of the rule is mapped to s; if this is not the case for each rule, return *false*.
7. For each $1 \leq k \leq n$, recursively call this procedure for $f_k(s)$ and $i+1$; if one of this call returns *false*, return *false* as well.
8. Return *true*.

Assume that this procedure returns *true* for each constant c, and let J be the union of all J^i considered in the process. It is straightforward to see that $I \subseteq J$, $J \models \Xi$, and $D(t) \notin J$. Furthermore, recursion depth of our algorithm is n and at each recursion level we need to keep a polynomially sized interpretation J^i, so our algorithm can be implemented using a nondeterministic Turing machine that uses polynomial space. By the Savitch's theorem, the algorithm can be implemented using a deterministic Turing machine that uses polynomial space, which proves our claim. □

Proof (Hardness). Consider an arbitrary quantified Boolean formula of the form $\varphi = Q_1 x_1 \dots Q_n x_n . C_1 \wedge \dots \wedge C_k$ defined over variables $x_1, \dots, x_n$, where each $Q_i \in \{\exists, \forall\}$, $1 \leq i \leq n$ is a quantifier, and each C_j, $1 \leq j \leq k$ is a clause of the form $C_j = L_{j,1} \vee L_{j,2} \vee L_{j,3}$. Checking validity of φ is the canonical PSPACE-hard problem.

In the rest of this proof, for a binary predicate P, we abbreviate $P(x_0,x_1) \wedge \dots \wedge P(x_{m-1},x_m)$ as $P^m(x_0,x_m)$. Let $\mathcal{T}$ be the Horn-$\mathcal{SHIQ}$ TBox containing rules (69)–(72) for each $1 \leq i \leq n$, rule (73) for each literal $L_{j,m} = x_\ell$ in clause C_j, rule (74) for each literal $L_{j,m} = \neg x_\ell$ in clause C_j, rule (75), rule (76) for each $1 \leq i \leq n$ such that $Q_i = \exists$, and rule (77) for each $1 \leq i \leq n$ such that $Q_i = \forall$.

$$A_{i-1}(x) \rightarrow \exists y.[X_i^+(x,y) \wedge A_i(x)] \quad (69)$$

$$A_{i-1}(x) \rightarrow \exists y.[X_i^-(x,y) \wedge A_i(x)] \quad (70)$$

$$X_i^+(x,x') \rightarrow P(x,x') \quad (71)$$

$$X_i^-(x,x') \rightarrow P(x,x') \quad (72)$$

$$X_\ell^+(z',z) \wedge P^{n-\ell}(z,x) \wedge A_n(x) \rightarrow C_j(x) \quad (73)$$

$$X_\ell^-(z',z) \wedge P^{n-\ell}(z,x) \wedge A_n(x) \rightarrow C_j(x) \quad (74)$$

$$C_1(x) \wedge \dots \wedge C_k(x) \rightarrow F_n(x) \quad (75)$$

$$P(x,z) \wedge F_i(z) \rightarrow F_{i-1}(x) \quad (76)$$

$$X_i^+(x,z) \wedge F_i(z) \wedge X_i^-(x,z') \wedge F_i(z') \rightarrow F_{i-1}(x) \quad (77)$$

Strictly speaking, rules (73), (74), and (77) are not Horn-$\mathcal{SHIQ}$ rules, but they can be transformed into Horn-$\mathcal{SHIQ}$ rules by replacing parts of their bodies with fresh concepts. It is straightforward to see what $\mathcal{T}$ is WA and, thus, MFA.

Let $I = \{A_0(a)\}$, and let $I_\mathcal{T}^\infty$ be the chase of I and $\mathcal{T}$. Due to rules (69)–(70), $I_\mathcal{T}^\infty$ contains a binary tree of depth n in which each leaf node is reachable from a via a path that, for each $1 \leq i \leq n$, contains either X_i^+ or X_i^-. If we interpret the presence of X_i^+ and X_i^- as assigning variable x_i to $\mathtt{t}$ and $\mathtt{f}$, respectively, then each leaf node corresponds to one possible assignment of $x_1, \dots, x_n$. Rules (73) and (74) then clearly label each leaf node with the clauses that are true in the node, and rule (75) labels each leaf node with F_n for which all clauses are true. Finally, rules (76) and (77) label each interior node of the tree with F_{i-1} according to the semantics of the appropriate quantifier of φ. Clearly, φ is valid iff $I \cup \mathcal{T} \models F_0(a)$, which implies our claim. □

Although the proof of Theorem 25 takes into account ontology rules with equality (i.e., rules of Type 2), it assumes that equality is axiomatised by $\Sigma_\approx$ and hence it does not directly apply to *singularised* Horn-$\mathcal{SHIQ}$ rules. We conjuecture, however, that the result in the theorem holds even if singularisation is applied.

The restriction to Horn-$\mathcal{SHIQ}$ rules also makes checking MFA w.r.t. an instance easier: this task can be accomplished by a minor variation of the query answering algorithm.

Theorem 26. *Let $\mathcal{T}$ be Horn-$\mathcal{SHIQ}$ TBox, and let I be an instance. Then, deciding whether $\mathcal{T}$ is MFA w.r.t. I is in* PSPACE, *and deciding whether $\mathcal{T}$ is universally MFA is* PSPACE-*hard.*

Proof. (Membership) Rules in $\mathsf{MFA}(\mathcal{T})$ are 'almost' Horn-$\mathcal{SHIQ}$ rules: rule (11) can be made a Horn-$\mathcal{SHIQ}$ rule by replacing S in the body with D (which clearly does not affect the consequences of the rule), and the fact that rule (12) contains a nullary atom in the head is immaterial. Thus, the claim can be proved by a straightforward adaptation of the membership proof of Theorem 25. The main difference in the algorithm is that, with n function symbols, we need to examine the models to depth $n + 1$; however, such an algorithm still uses polynomial space.

(Hardness) Let φ be an arbitrary QBF, and let $\mathcal{T}$ be as in the hardness proof of Theorem 23. Furthermore, let $\mathcal{T}'$ be the extension of $\mathcal{T}$ with the following rule:

$$F_0(x) \to \exists y.[B(x,y) \wedge A_0(y)] \qquad (78)$$

By an argument analogous to the one used in the proof of Theorem 7, one can see that $\mathcal{T}'$ is not universally MFA if and only if φ is valid. □

Finally, MSA provides us with a tractable condition for Horn-$\mathcal{SHIQ}$ rules. Intuitively, all rules in $\mathsf{MSA}(\mathcal{T})$ have a bounded number of variables and all predicates in $\mathsf{MSA}(\mathcal{T})$ are of bounded arity, which eliminates all sources of intractability in datalog reasoning. This result also holds for singularised rules.

Theorem 27. *Let $\mathcal{T}$ be Horn-$\mathcal{SHIQ}$ TBox, and let I be an instance. Then, deciding whether $\mathcal{T}$ is MSA w.r.t. I is in* PTIME, *and deciding whether $\mathcal{T}$ is universally MSA is* PTIME-*hard.*

Proof. (Membership) The datalog program $\mathsf{MSA}(\mathcal{T})$ contains predicates of bounded arity, so its chase w.r.t. I is polynomial in size. Furthermore, the number of variables in each rule in $\mathsf{MSA}(\mathcal{T})$ is bounded, so each rule can be applied in polynomial time. Thus, the chase of I and $\mathsf{MSA}(\mathcal{T})$ can be computed in polynomial time, which implies our claim.

(Hardness) A *monotone circuit* C is a finite directed acyclic graph with *input* vertices $v_{i_1}, \ldots, v_{i_n}$ and one *output* vertex v_o. Each non-input vertex v in C is assigned a label $\lambda(v) \in \{\wedge, \vee\}$. A *valuation* μ is an assignment of input vertices to t and f. The *value* of C on μ is an assignment μ_C of all vertices in C to t and f that coincides with μ on the input vertices, and that is defined inductively on each other vertex v with incoming edges from vertices $v_1, \ldots, v_n$ as follows.

- If $\lambda(v) = \vee$, then $\mu_C(v) = \mu_C(v_1) \vee \ldots \vee \mu_C(v_n)$.
- If $\lambda(v) = \wedge$, then $\mu_C(v) = \mu_C(v_1) \wedge \ldots \wedge \mu_C(v_n)$.

The problem of deciding for a given C and μ whether $\mu_C(v_o) = \mathsf{t}$ is PTIME-hard.

Let C be a monotone circuit. Then, $\mathcal{T}$ is the Horn-$\mathcal{SHIQ}$ TBox that uses a concept V_i for each vertex v_i of C and that, for each vertex v with incoming edges from vertices $v_1, \ldots, v_n$ contains rule (79) if $\lambda(v) = \wedge$ and rule (80) for each $1 \leq i \leq n$ if $\lambda(v) = \vee$.

$$V_1(x) \wedge \ldots \wedge V_n(x) \to V(x) \qquad (79)$$

$$V_i(x) \to V(x) \qquad (80)$$

Given an arbitrary assignment μ, let I_μ be the instance that, for each input vertex v_{i_j} of C, contains $V_{i_j}(a)$ if and only if $\mu(v_{i_j}) = \mathsf{t}$. Clearly, $I_\mu \cup \mathcal{T} \models V_o(a)$ if and only if $\mu_C(v_o) = \mathsf{t}$.

Now let $\mathcal{T}'$ be the extension of $\mathcal{T}$ with the following rule:

$$V_o(x) \to \exists y.[B(x,y) \wedge \bigwedge_{\mu(v_{i_j})=\mathsf{t}} V_{i_j}(y)] \qquad (81)$$

By an argument analogous to the one used in the proof of Theorem 7, one can see that $\mathcal{T}'$ is not universally MSA if and only if $\mu_C(v_o) = \mathsf{t}$. □

Experiments

We have evaluated the applicability of various acyclicity conditions in practice. First, we implemented MFA, MSA, JA, and WA checkers, and used them to check acyclicity of a large corpus of Horn ontologies. Our goal was to determine whether a substantial portion of these ontologies are acyclic and could thus be used with (suitably extended) materialisation-based reasoners. Second, we computed the materialisation of the acyclic Horn ontologies and compared the size of the materialisation with the size of the original ABox. The goal of these tests was to see whether materialisation-based reasoning is practically feasible.

Tests were performed on the Oxford Super Computer HAL system with 8 2.8GHz processors and 16GB RAM. We used a repository of 149 OWL ontologies whose TBox axioms can be transformed into existential rules. These ontologies include many of those in the Gardiner corpus (Gardiner, Tsarkov, and Horrocks 2006), the LUBM ontology, and a number of ontologies from the Open Biomedical Ontology (OBO) corpus. None of our test ontologies, however, has been obtained from typical conceptual models (i.e., ER or UML diagrams); due to the specific modelling patterns used in conceptual modelling, such ontologies are more likely to be cyclic. All test ontologies are available online.[2]

Acyclicity Tests

We implemented all acyclicity checks by adapting the HermiT reasoner. HermiT was used to transform an ontology into DL-clauses—formulae quite close to existential rules. DL-clauses were then preprocessed: at-least number restrictions in rule heads were replaced with existential quantification, atoms involving datatypes were eliminated, and DL-clauses with empty head were removed; datatypes and empty heads merely cause inconsistencies, and do not contribute to chase non-termination. If the DL-clauses contained equality, we check $X^\cup$ instead of X for each $X \in \{\text{MFA}, \text{MSA}, \text{JA}\}$ as a 'lower-bound' for acyclicity. To obtain an 'upper bound' for acyclicity, we checked whether the ontology was already cyclic when ignoring the

[2] http://hermit-reasoner.com/2011/acyclicity/TestCorpus.zip

G-rules	Total	MSA	JA	WA
ontologies without equality				
< 100	21	19	19	19
100–1K	33	30	30	23
1K–5K	18	14	14	12
5K–12K	9	8	6	6
12K–160K	7	5	3	3
ontologies with equality				
< 100	49	45	45	45
100–1K	0	0	0	0
1K–5K	2	0	0	0
5K–12K	5	3	0	0
12K–160K	5	0	0	0

Table 2: Results of acyclicity tests

rules containing equality. These steps produced a set of existential rules, which were further modified as required to encode the desired ayclicity check. Finally, HermiT was used to test universal acyclicity of the ontology by checking logical entailment w.r.t. the critical instance.

Each acyclicity test was given a $500s$ timeout. The MSA test exceeded this limit on 2 ontologies, whereas the MFA test exceeded the limit on 26 ontologies. Of the 149 ontologies tested, 124 (83%) were MSA. Moreover, MFA and MSA are indistinguishable w.r.t. the test ontologies—that is, all MFA ontologies were found to be MSA as well (the converse holds per Theorem 13). Results are shown in Table 2. Given the large number of test ontologies, we cannot show results for each ontology. Instead, ontologies are grouped by number of generating rules (G-rules); for each group, the table shows the number of ontologies (Total) and the number of ontologies found to be MSA, JA, and WA.

Note that seven large OBO ontologies were MSA but not JA; thus, MSA may be especially useful on large and complex ontologies. Table 3 shows for each of these ontologies the number of generating rules (G-rules), if it uses equality (Eq), expressivity (DL), and the number of classes (C), properties (P), and axioms (A).

Of the 25 non-MFA ontologies, only two ontologies are in $\mathcal{ELH}^r$ or certain versions of DL-Lite; this is interesting since combined approaches (Lutz, Toman, and Wolter 2009; Kontchakov et al. 2011) can be used to support CQ answering over these ontologies.

Materialisation Tests

To estimate the practicability of materialisation in acyclic ontologies, we measured the maximal depth of function symbol nesting in terms generated by skolem chase. This measure, which we call *ontology depth*, is of interest as it can be used to establish a bound on the size of the chase. Our tests revealed that most ontologies have small depths: out of the 124 MSA ontologies, 83 (66.9%) have depths less than 5; 13 (10.5%) have depths from 5 to 9; 24 (19.4%) have depths from 10 to 19; 2 (1.6%) have depths from 20 to 49; and 2 (1.6%) have depths from 50 to 80.

G-rules	Eq	DL	C	P	A
biological_process_xp_self.imports.owl					
10980	yes	$\mathcal{SRIF}$	22375	183	47454
go_xp_regulation.owl					
11187	no	$\mathcal{SH}$	27883	5	50941
biological_process_xp_cell.imports.owl					
11274	yes	$\mathcal{SRIF}$	24309	293	50386
cellular_component_xp_go.imports.owl					
11473	no	$\mathcal{SR}$	35236	8	64026
biological_. . . . _cellular_component.imports.owl					
11798	yes	$\mathcal{SRIF}$	25337	187	52759
go_xp_regulation.imports.owl					
23844	no	$\mathcal{SR}$	34293	8	104473
biological_. . . . _multi_organism_process.imports.owl					
24678	no	$\mathcal{SR}$	34410	21	104873

Table 3: MSA but not JA ontologies

We also computed the materialisation of several acyclic ontologies. Since our implementation is only prototypical, our primary goal was not to evaluate the performance of materialisation, but rather to estimate the increase in ABox size. Although this increase may not be perfectly linear, we believe that it can be estimated by examining moderately-sized ABoxes. Most of our test ontologies, however, do not have substantial ABoxes; ontologies are often publicly available as general vocabularies, whereas ABoxes are application-specific and are thus usually not publicly available. Because of this problem, we conducted two kinds of experiments.

First, we computed the materialisation of two ontologies with nontrivial ABoxes: LUBM with one university and the 'kmi-basic-portal' ontology. The TBox of LUBM contains 8 generating rules and has depth 1; the ABox before materialisation contains $100,543$ facts. Materialisation took only 1 second, and it produced $150,530$ new facts, of which $47,798$ were added by generating rules. The 'kmi-basic-portal' ontology has 10 generating rules and has depth 2; the ABox contains 179 facts. Materialisation took only 0.01 seconds, and it added 975 new facts, of which 151 were added by generating rules.

Second, for each of the 124 ontologies identified as MSA, we computed an ABox by instantiating each class and property with fresh individuals. We then computed the materialisation and measured the *generated size* (number of facts introduced by generating rules, divided by the facts in the initial ABox), the *materialisation size* (facts in the materialisation, divided by facts in the initial ABox), and the materialisation time. Since most generating rules in these ontologies had singleton body atoms (i.e., they are of the form $A(x) \rightarrow \exists R.C(x)$), these measures should provide a reasonable estimate of the increase in ABox size caused by materialisation. Of the 124 ontologies tested, 15 exceeded the $1,000s$ time limit for materialisation. The results for the other 109 ontologies are shown in Table 4. Ontologies are grouped by depth; each group shows the number of ontologies (#), and materialisation times, generated sizes, and materialisation sizes.

Depth	#	time		gen. size		mat. size	
		max	avg	max	avg	max	avg
< 5	82	69	0.9	27	2	35	5
5–9	13	68	11	37	11	41	13
10–80	14	549	101	281	51	283	53

Table 4: Materialisation times (in seconds) and sizes

Thus, materialisation seems practically feasible for many ontologies: for 82 ontologies with depth less than 5, materialisation increases the ontology size by a factor of 5. This suggests that principled, materialisation-based reasoning for ontologies beyond the OWL 2 RL profile may be feasible, especially for ontologies with relatively small depths.

Conclusion

In this paper, we have studied the problem of CQ answering over acyclic existential rules. We have proposed two novel acyclicity conditions that are sufficient to ensure chase termination and that generalise many of the existing acyclicity conditions known thus far.

We have then studied the problem of CQ answering over acyclic DL ontologies. Acyclicity provides several compelling benefits for DL query answering. First, the CQ answering problem over Horn ontologies is computationally easier than for general ontologies. Second, under acyclicity conditions it is possible to extend Horn ontologies with arbitrary SWRL rules while preserving both the decidability and the worst-case complexity of the formalism. Third, acyclicity enables principled extensions of ontology materialisation-based reasoners. Fourth, ontologies used in practice are often acyclic, so our results open the door to practical CQ answering beyond the OWL 2 RL profile.

Acknowledgments

This work was supported by the EU FP7 project SEALS and by the EPSRC projects ConDOR, ExODA, and LogMap. B. Cuenca Grau is supported by a Royal Society University Research Fellowship.

References

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

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition.

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

Baget, J.-F.; Mugnier, M.-L.; and Thomazo, M. 2011. Towards farsighted dependencies for existential rules. In *Proc. RR*, 30–45.

Beeri, C., and Vardi, M. Y. 1981. The implication problem for data dependencies. In *Proc. ICALP*, 73–85.

Bishop, B., and Bojanov, S. 2011. Implementing OWL 2 RL and OWL 2 QL rule-sets for OWLIM. In *Proc. OWLED*, volume 796 of *CEUR WS Proceedings*.

Broekstra, J.; Kampman, A.; and van Harmelen, F. 2002. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *Proc. ISWC*, 54–68.

Calì, A.; Gottlob, G.; Lukasiewicz, T.; Marnette, B.; and Pieris, A. 2010. Datalog+/-: A family of logical knowledge representation and query languages for new applications. In *Proc. LICS*, 228–242.

Calì, A.; Gottlob, G.; and Pieris, A. 2010. Query answering under non-guarded rules in Datalog+/-. In *Proc. RR*, 1–17.

Calì, A.; Gottlob, G.; and Pieris, A. 2011. New expressive languages for ontological query answering. In *Proc. AAAI*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Automated Reasoning* 39(3):385–429.

Carroll, J. J.; Dickinson, I.; Dollin, C.; Reynolds, D.; Seaborne, A.; and Wilkinson, K. 2004. Jena: Implementing the Semantic Web Recommendations. In *Proc. WWW—Alternate Track*, 74–83.

Cuenca Grau, B.; Horrocks, I.; Motik, B.; Parsia, B.; Patel-Schneider, P.; and Sattler, U. 2008. OWL 2: The next step for OWL. *J. Web Semantics* 6(4):309–322.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Comput. Surv.* 33(3):374–425.

Demri, S., and de Nivelle, H. 2005. Deciding Regular Grammar Logics with Converse Through First-Order Logic. *J. Logic, Language and Information* 14(3):289–329.

Deutsch, A.; Nash, A.; and Remmel, J. B. 2008. The chase revisited. In *Proc. PODS*, 149–158.

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

Gardiner, T.; Tsarkov, D.; and Horrocks, I. 2006. Framework for an automated comparison of description logic reasoners. In *Proc. ISWC*, 654–667.

Glimm, B.; Lutz, C.; Horrocks, I.; and Sattler, U. 2008. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *J. Artif. Intell. Res.* 31:157–204.

Horrocks, I., and Patel-Schneider, P. F. 2004. A proposal for an OWL rules language. In *Proc. WWW*, 723–731.

Hustadt, U.; Motik, B.; and Sattler, U. 2005. Data complexity of reasoning in very expressive description logics. In *Proc. IJCAI*, 466–471.

Johnson, D. S., and Klug, A. C. 1984. Testing containment of conjunctive queries under functional and inclusion dependencies. *J. Comput. Syst. Sci.* 28(1):167–189.

Kiryakov, A.; Ognyanov, D.; and Manov, D. 2005. OWLIM: A pragmatic semantic repository for OWL. In *WISE Workshops*, 182–192.

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zakharyaschev, M. 2011. The combined approach to ontology-based data access. In *Proc. IJCAI*, 2656–2661.

Krötzsch, M., and Rudolph, S. 2011. Extending decidable existential rules by joining acyclicity and guardedness. In *Proc. IJCAI*, 963–968.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2007. Conjunctive queries for a tractable fragment of OWL 1.1. In *Proc. ISWC*, 310–323.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2012. Complexities of Horn description logics. *ACM Trans. Comp. Log.* To appear.

Kutz, O.; Horrocks, I.; and Sattler, U. 2006. The Even More Irresistible $\mathcal{SROIQ}$. In *Proc. KR*, 68–78.

Lutz, C.; Toman, D.; and Wolter, F. 2009. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proc. IJCAI*.

Maier, D.; Mendelzon, A. O.; and Sagiv, Y. 1979. Testing implications of data dependencies. *ACM Trans. Database Syst.* 4(4):455–469.

Marnette, B. 2009. Generalized schema-mappings: from termination to tractability. In *Proc. PODS*, 13–22.

Meditskos, G., and Bassiliades, N. 2008. Combining a DL reasoner and a rule engine for improving entailment-based OWL reasoning. In *Proc. ISWC*, 277–292.

Meier, M.; Schmidt, M.; and Lausen, G. 2009. On chase termination beyond stratification. *Proc. VLDB* 2(1):970–981.

Motik, B.; Shearer, R.; and Horrocks, I. 2009. Hypertableau reasoning for description logics. *J. Artif. Intell. Res.* 36:165–228.

Ortiz, M.; Calvanese, D.; and Eiter, T. 2008. Data complexity of query answering in expressive description logics via tableaux. *J. Automated Reasoning* 41(1):61–98.

Ortiz, M.; Rudolph, S.; and Simkus, M. 2011. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In *Proc. IJCAI*, 1039–1044.

Pérez-Urbina, H.; Motik, B.; and Horrocks, I. 2010. Tractable query answering and rewriting under description logic constraints. *J. Applied Logic* 8(2):186–209.

Rudolph, S., and Glimm, B. 2010. Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! *J. Artif. Intell. Res.* 39:429–481.

Spezzano, F., and Greco, S. 2010. Chase termination: A constraints rewriting approach. *Proc. VLDB* 3(1):93–104.

Wu, Z.; Eadon, G.; Das, S.; Chong, E. I.; Kolovski, V.; Annamalai, M.; and Srinivasan, J. 2008. Implementing an inference engine for RDFS/OWL constructs and user-defined rules in Oracle. In *Proc. ICDE*, 1239–1248.