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 fundamental reasoning problem although undecidable due to non-termination of the main reasoning algorithm used—the chase. Several acyclicity conditions have been formulated that ensure chase termination. In this paper, we show that acyclicity can also be practically relevant for description logic (DL) reasoning. Due to the high complexity of answering CQs over DL ontologies, applications often solve this problem using materialisation, in which ontology consequences are precomputed using variants of the chase. Due to the non-termination problem, the execution of the algorithm is restricted only to rules that fall within the OWL 2 RL profile, which results in incomplete reasoning. After presenting two novel acyclicity conditions (model-faithful acyclicity (MFA) and model-summarising acyclicity (MSA)), we investigate the practical applicability of these and other acyclicity conditions for DL query answering. Our experiments reveal that many existing ontologies are MSA and that materialisation is 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.

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; they have, e.g., been used as declarative rules for data transformation in data exchange—the problem of transforming a database structure according to a source schema into a database structured according to a target schema (Fagin et al. 2005). They are also the basis for KR formalisms, such as datalog²° (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), and it can be characterised using chase (Johnson and Klug 1984; Maier, Mendelzon, and Sagiv 1979): a technique, closely related to hypertableau (Motik, Shearer, and Horrocks 2009), that computes facts implied by the rules in a forward-chaining manner, thus producing a universal model in which the query is evaluated.

Rules with existentially quantified variables in the head—which we call generating rules—may cause chase to generate new individuals, and cycles involving generating rules may lead to non-termination; moreover, determining whether chase terminates on a set of rules is undecidable. Chase, however, has been used to identify decidable classes of existential rules, and this has been done in two ways. In the first approach, rules are restricted such that their (possibly infinite) universal models can be represented using finitary means; this includes rules with universal models of bounded treewidth (Baget et al. 2011) and guarded rules (Calì et al. 2010). In the second approach, rules are checked using an acyclicity condition that is sufficient (but not necessary) to prove 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 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, Munnig, and Thomazo 2011) joint acyclicity (JA) (Krötzsch and Rudolph 2011), and super-weak acyclicity (SWA) (Marnette 2009).

Ayclicity conditions are relevant for at least two reasons. First, unlike guarded rules, they do not restrict the shape of the structures that the rules can axiomatise; rather, they ensure that the rules can axiomatise only finite structures. Second, they ensure that the chase result can be stored and manipulated as if it were a database. This is important in data exchange, where the goal is to materialise the resulting database.

In this paper, we show that acyclicity is also relevant for description logics (DLs), the KR formalisms underpinning the OWL 2 ontology language (Cuenca Grau et al. 2008). CQ answering is a key reasoning
service in many DL applications, which has been studied for many DLs (Calvanese et al. 2007; Krötzsch, Rudolph, and Hitzler 2007b; 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).

Due to the high complexity of answering CQs over expressive DLs, however, applications often solve this problem using materialisation, in which ontology consequences are precomputed using forward-chaining and stored in a semantic data store; examples include Oracle’s Semantic Data Store, Sesame, Jena, OWLlim, and DLE-Jena (Wu et al. 2008; Medítkos and Bassiliades 2008; Kiryakov, Ognyanov, and Manov 2005). 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 using only inference rules corresponding to the part of the ontology that is in the OWL 2 RL profile; this excludes generating rules and so is terminating but incomplete in general. Even if generating rules are partially supported, as is the case in systems such as OWLlim and Jena (Bishop and Bojanov 2011), this is typically rather ad hoc, does not guarantee completeness, and may even result in non-termination. Acyclicity conditions can be used to address these issues: if a Horn DL ontology is acyclic, a complete materialisation can be computed without the risk of non-termination.

Motivated by the practical importance of chase termination, we explore the landscape of acyclicity conditions, and present two novel conditions: model-faithful acyclicity (MFA) and model-summarising acyclicity (MSA). We then go investigate the practical applicability of these and other acyclicity conditions for query answering over DL ontologies.

Roughly speaking, our 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 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 show that MSA is strictly more general than SWA—one of the most general acyclicity conditions currently is used.

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. Singularisation is orthogonal to acyclicity: after computing the transformed rules, one can use MFA, MSA, or any acyclicity notion to check whether the result is acyclic; if so, chase with 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 will 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 turn our attention to the theoretical and practical questions of applying acyclicity to the problem of CQ answering over DLs. On the theoretical side, we show that checking MFA and MSA of Horn-SHIQ ontologies is PSPACE- and PTIME-complete, respectively, and that CQs can be answered over acyclic Horn-SHIQ ontologies in PSPACE. The latter problem is EXPTime-hard for general (i.e., not acyclic) Horn-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 lead to undecidability.

On the practical side, we explore the limits of reasoning with acyclic OWL 2 ontologies via materialisation. We checked MFA, MSA, and JA of a library with 149 Horn ontologies; 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. Materialisation is typically not too large. Our results suggest that principled, materialisation-based reasoning for ontologies beyond the OWL 2 RL profile may be feasible.

Preliminaries

We use the standard notions of constants, function symbols, and predicate symbols, where the latter two are associated with integer arity: \( \approx \) is the equality predicate. 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 \( \varphi \), and \( \varphi \sigma \) is the result of applying a substitution \( \sigma \) to \( \varphi \). A term, atom, or formula is ground if it does not contain variables; a fact is a ground atom. The depth \( \text{dep}(t) \) of a term \( t \) is defined as 0 if \( t \) is a constant or a variable, and \( \text{dep}(\vec{t}) = 1 + \max_{i=1}^{n} \text{dep}(t_i) \) if \( t = f(\vec{t}) \). Satisfaction of a sentence \( \varphi \) in an interpretation \( I \) (written \( I \models \varphi \)), and entailment of a sentence \( \psi \) from a sentence \( \varphi \) (written \( \varphi = \psi \)), 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 $t' = f(t)$ and $t'$ is a subterm of some $t_i \in \bar{t}$; if additionally $t' \neq t$, then $t'$ is a proper subterm of $t$. An atom $P(t)$ contains a term $t$ if $t \in \bar{t}$, 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

$$\forall \bar{x} \forall \bar{z} [\varphi(\bar{x}, \bar{z}) \rightarrow \exists \bar{y} \psi(\bar{x}, \bar{y})]$$

where $\varphi(\bar{x}, \bar{z})$ and $\psi(\bar{x}, \bar{y})$ are conjunctions of atoms, and $\bar{x}$, $\bar{y}$, and $\bar{z}$ are pair-wise disjoint. Formula $\varphi$ is the body, formula $\psi$ is the head, and quantifiers $\forall \bar{x} \forall \bar{z}$ are often omitted. If $\bar{y}$ is empty, the rule is database. 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 $Q(\bar{x})$ of the form $\exists \bar{y} \varphi(\bar{x}, \bar{y})$, where $\varphi(\bar{x}, \bar{y})$ is a conjunction of atoms; the query is Boolean if $\bar{y}$ is empty. A substitution $\theta$ mapping $\bar{x}$ to constants is an answer to $Q(\bar{x})$ w.r.t. a set of rules $\Sigma$ and instance $I$ if $\Sigma \cup I \models Q(\bar{x})\theta$.

In first-order logic, $\approx$ is commonly assumed to have a predefined interpretation. The semantics of $\approx$, however, can be axiomatised explicitly. Let $\Sigma$ be a set of rules; w.l.o.g. we assume that no rule in $\Sigma$ contains $\approx$ in the body. Then, $\Sigma_{\approx} = \emptyset$ if $\approx$ does not occur in $\Sigma$; otherwise, $\Sigma_{\approx}$ contains the following rules:

$$a \approx a$$
$$x_1 \approx x_2 \rightarrow x_2 \approx x_1$$
$$x_1 \approx x_2 \wedge x_2 \approx x_3 \rightarrow x_1 \approx x_3$$

$$(\bar{P}(\bar{x}) \wedge x_i \approx x_i') \rightarrow \bar{P}(x_1, \ldots, x_i', \ldots, x_n)$$

The consequences of $\Sigma$ (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, the skolemisation of an existential rule $r$ of the form (1) is the rule

$$\varphi(\bar{x}, \bar{z}) \rightarrow \psi(\bar{x}, \bar{y})\theta$$

where for each $y_i \in \bar{y}$ we have $\theta(y_i) = f_r^i(x)$ with $f_r^i$, a function symbol globally unique for $r$ and $y_i$. The skolemisation $\text{sk}(\Sigma)$ of a set of rules $\Sigma$ is obtained by skolemising each rule in $\Sigma$. For each CQ $Q(\bar{x})$, instance $I$, and substitution $\sigma$, we have $\Sigma \cup I \models Q(\bar{x})\sigma$ if and only if $\text{sk}(\Sigma) \cup \Sigma_{\approx} \cup I \models Q(\bar{x})\sigma$.

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 $\psi$ for each substitution $\sigma$ from variables in $r$ to terms in $I$ such that $\varphi \sigma \subseteq I$; furthermore, for $\Omega$ a set of rules, $\Omega(I) = \bigcup_{r \in \Omega} r(I)$. Let $I$ be a finite set of ground atoms, and let $\Sigma$ be a set of rules. Let $\Sigma_f \Sigma_{\approx}$, and let $\Sigma_f$ and $\Sigma_{\approx}$ be the subsets of $\Sigma$ containing rules with and without function symbols, respectively. The chase sequence for $I$ and $\Sigma$ is a sequence of sets of facts $I_0^\infty, I_1^\infty, \ldots$ where $I_0^\infty = I$, and $I_k^\infty$ for $k > 0$ is as follows:

- if $\Sigma_{\approx}(I_{k-1}^\infty) \not\subseteq I_{k-1}^\infty$, then $I_k^\infty = I_{k-1}^\infty \cup \Sigma_{\approx}(I_{k-1}^\infty)$,
- otherwise $I_k^\infty = I_{k-1}^\infty \cup \Sigma_f(I_{k-1}^\infty)$.

The chase of $I$ and $\Sigma$ is defined as $I^\infty = \bigcup_k I_k^\infty$; note that $I^\infty$ can be infinite. Chase can be used as a ‘database’ for answering CQs: a substitution $\sigma$ is an answer to $Q$ over $\Sigma$ and $I$ iff $I^\infty \models Q\sigma$. Chase of $I$ and $\Sigma$ terminates if $i \geq 0$ exists such that $I_k^\infty = I_{k+1}^\infty$ for each $j > i$; chase of $\Sigma$ terminates universally if the chase of $I$ and $\Sigma$ terminates for each $I$. If the skolem chase of $I$ and $\Sigma$ terminates, then the nonoblivious chase (Fagin et al. 2005), and the core chase (Deutsch, Nash, and Remmel 2008) of $I$ and $\Sigma$ terminate as well; hence, our results are applicable to all these chase variants.

The critical instance $I^c$ for rules $\Sigma$ contains all facts constructed using all predicates in $\Sigma$, all constants in the body of a rule in $\Sigma$, and a fresh constant $\ast$. If the skolem chase for $I^c$ and $\Sigma$ terminates, then the skolem chase of $\Sigma$ terminates universally (Marnette 2009).

Universal chase termination is undecidable, and various sufficient acyclicity conditions have been proposed. In the following, let $\Sigma$ 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 $P_{y_1}(x)$ 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 $\text{Pos}_B(w)$ of body positions of $w$ consists of all positions $P_{y_1}(x)$ for which $P(t_1, \ldots, t_n) \in \varphi(\bar{x}, \bar{z})$ exists with $t_i = w$. The set $\text{Pos}_H(w)$ is defined analogously.

Weak acyclicity (WA) was proposed by Fagin et al. (2005), and it is applicable to rules with $\approx$. The WA dependency graph $D_{\Sigma}$ for $\Sigma$ is defined as follows. Vertices of $D_{\Sigma}$ are positions. Graph $D_{\Sigma}$ contains, for each $r \in \Sigma$ of the form (1), each variable $x$, and each $P_{y_1}(x)$, a regular edge from $P_{y_1}$ to each $Q_{y_1} \in \text{Pos}_H(x)$ such that $Q \neq \approx$ and, for each $y \in \bar{y}$ and each $Q_{y_1} \in \text{Pos}_H(y)$ such that $Q \neq \approx$, a special edge from $P_{y_1}$ to $Q_{y_1}$. Set $\Sigma$ is WA if $D_{\Sigma}$ does not contain a cycle going through a special edge. Atoms involving equality are effectively ignored by WA.

Joint acyclicity (JA) is a generalisation of WA (Krötzsch and Rudolph 2011) applicable to equality-free rules. For an existentially quantified variable $y$ in $\Sigma$, let $\text{Move}(y)$ be the smallest set of positions such that $\text{Pos}_B(y) \subseteq \text{Move}(y)$, and $\text{Pos}_H(x) \subseteq \text{Move}(y)$ for each universally quantified variable $x$ with $\text{Pos}_B(x) \subseteq \text{Move}(y)$. The JA dependency graph of $\Sigma$ is as follows. The vertices are the existentially quantified variables in $\Sigma$. The graph has an edge from each $y$ to each $y'$ such that the rule in which $y'$ occurs also contains a universally quantified variable $x$ such that
\(\text{Pos}_R(x) \subseteq \text{Move}(y)\) and \(\text{Pos}_H(x) \neq \emptyset\). Set \(\Sigma\) is JA if its JA dependency graph does not contain a cycle.

Super-weak acyclicity (SWA) (Marnette 2009) is more general than JA on rules in which a variable occurs more than once in a 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 \(\Sigma\) is rewritten into a set of rules \(\Sigma'\) and, subsequently, \(\Sigma'\) is checked for some existing notions of acyclicity (e.g., weak acyclicity). Since this rewriting technique is orthogonal to the aforementioned termination conditions and can be combined with any acyclicity condition we do not examine it in more detail.

**Model-Faithful Acyclicity**

Ensuring (universal) chase termination can often be beneficial, and even necessary if the chase result is to be physically stored and manipulated as a database. Conditions such as JA, however, do not guarantee chase termination on certain commonly occurring rules.

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

\[
\begin{align*}
    r_1 &= A(x) \rightarrow \exists y_1.R(x, y_1) \land B(y_1) \quad (7) \\
    r_2 &= R(x, z) \land B(z) \rightarrow A(x) \quad (8) \\
    r_3 &= B(x) \rightarrow \exists y_2.R(x, y_2) \land C(y_2) \quad (9)
\end{align*}
\]

Note that \(\text{Move}(y_1) = \{R_i, B, R_1, A_1\}\); hence, the JA dependency graph has a cyclic edge from \(y_1\) to itself. Chase of \(\Sigma\), 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.

Note that rules \(r_1\) and \(r_2\) encode the DL axiom \(A \equiv \exists R.B\), and rule \(r_3\) encodes \(B \subseteq \exists R.C\); 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 \(x\) 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 run 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 when running chase of \(\Sigma\) in 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 to 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 \(\Sigma\) are transformed into a new set of rules \(\Sigma'\), which tracks rule dependencies using fresh predicates; then, \(\Sigma\) is identified as being acyclic if \(\Sigma'\) entails 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 new acyclicity notion model-faithful acyclicity because it estimates rule application precisely, by examining the actual model of \(\Sigma\).

**Definition 2.** For each rule \(r = \varphi(\vec{x}, \vec{z}) \rightarrow \exists y.\psi(\vec{x}, \vec{y})\) and each variable \(y_i \in \vec{y}\), let \(F_i^r\) be a fresh unary predicate unique for \(r\) and \(y_i\); also, let \(S\) and \(D\) be fresh binary predicates and let \(C\) be a fresh nullary predicate.

Then, MFA(r) is the following rule:

\[
\varphi(\vec{x}, \vec{z}) \rightarrow \exists y.\psi(\vec{x}, \vec{y}) \land \bigwedge_{y_i \in \vec{y}} F_i^r(y_i) \land \bigwedge_{x_j \in \vec{x}} S(x_j, y_i)\]

For \(\Sigma\) a set of rules, MFA(\(\Sigma\)) is the smallest set that contains MFA(r) for each rule \(r \in \Sigma\), rules (10)–(11), and rule (12) instantiated for each predicate \(F_i^r\):

\[
\begin{align*}
    S(x_1, x_2) &\rightarrow D(x_1, x_2) \quad (10) \\
    D(x_1, x_2) \land S(x_2, x_3) &\rightarrow D(x_1, x_3) \quad (11) \\
    F_1^r(x_1) \land D(x_1, x_2) \land F_2^r(x_2) &\rightarrow C \quad (12)
\end{align*}
\]

Set \(\Sigma\) is model-faithful acyclic (MFA) w.r.t. an instance \(I\) if \(I \cup \text{MFA}(\Sigma) \not\subseteq C\); furthermore, \(\Sigma\) is universally MFA\(^1\) if \(\Sigma\) is MFA w.r.t. \(I^*_r\).

MFA is defined as a semantic, rather than a syntactic condition, and entailment \(I \cup \text{MFA}(\Sigma) \not\subseteq C\) 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 characterizes derivations of skolem chase in a particular way.

\(^1\)In the rest of this paper we typically omit ‘universally’.
Definition 3. A term $t$ is cyclic if some $f(s)$ is a subterm of $t$, and some $f(\bar{x})$ is a proper subterm of $f(s)$.

Proposition 4. A set of rules $\Sigma$ is not MFA w.r.t. an instance $I$ iff $I_{\text{MFA}(\Sigma)}$ contains a cyclic term.

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

1. For each term $t = f_i(t')$ occurring in $I_{\Sigma'}^k$, we have $F_i^t(t) \in I_{\Sigma'}^k$. Conversely, $F_i^t(t) \in I_{\Sigma'}^k$ implies $t = f_i(t')$.

2. For each term $t = f_i(t')$ occurring in $I_{\Sigma'}^k$, and each $t'' \in I$, we have $S(t', t) \in I_{\Sigma'}^k$. Conversely, $S(t', t) \in I_{\Sigma'}^k$ implies $t = f_i(t')$ and $t'' \in I$.

3. For all terms $t$ and $t''$ occurring in $I_{\Sigma'}^k$, such that $t'$ is a proper subterm of $t$, we have $D(t', t) \in I_{\Sigma'}^{k+2}$. Conversely, $D(t', t) \in I_{\Sigma'}^{k+2}$ implies $t'$ is a proper subterm of $t$.

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

(Claims 1 and 2, the second part) Predicate $S$ and each predicate $F_i^t$ occur in $\Sigma'$ only in the head of some rule 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 $I_{\Sigma'}^0$ does not contain functional terms. Assume now that the claim holds for some $k - 1$, and consider arbitrary term $t = f_i(t')$ occurring in $I_{\Sigma'}^{k-1}$ such that $t'$ is a subterm of some $t_i \in I$. By Claim 2, we have $S(t_i, t) \in I_{\Sigma'}^k$; furthermore, $t_i$ occurs in $I_{\Sigma'}^{k-1}$, so by the induction assumption we have $D(t', t_i) \in I_{\Sigma'}^{k+1}$. Finally, rules without functional terms are applied before rules with functional terms; hence, by rule (10) we have $D(t_i, t) \in I_{\Sigma'}^{k+1}$, and by rule (11) we have $D(t', t) \in I_{\Sigma'}^{k+2}$, 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 $I_{\Sigma'}^\infty$ contains a cyclic term $t$. Then, some $t_i = f_i^t(s)$ is a subterm of $t$ and $t_s = f_i^t(\bar{x})$ is a proper subterm of $t_i$. By Claims 1 and 3, then $\{F_i^t(t_1), D(t_1, t_1), F_i^t(t_1)\} \subseteq I_{\Sigma'}^\infty$. But then, since $\Sigma'$ contains rule (12), we have $c \subseteq I_{\Sigma'}^\infty$, so $\Sigma$ is not MFA. The proof of the converse claim is analogous.

This characterisation immediately implies termination of skolem chase of MFA rules in $2\text{ExpTime}$. The latter already holds if the rules are WA; hence, CQ answering for MFA rules is not harder than for WA.

Proposition 5. If a set of rules $\Sigma$ is MFA w.r.t. an instance $I$, then the skolem chase for $I$ and $\Sigma$ terminates in double exponential time.

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

Proposition 5 implies that answering a BCQ over MFA rules is in $2\text{ExpTime}$; furthermore, Cali, 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 $2\text{ExpTime}$, and that checking universal MFA is $2\text{ExpTime}$-hard. These results provide tight bounds for both problems.

Theorem 6. For $\Sigma$ a set of rules, deciding whether $\Sigma$ is MFA w.r.t. an instance $I$ is in $2\text{ExpTime}$, and deciding whether $\Sigma$ is universally MFA is $2\text{ExpTime}$-hard. Both results hold even if the arity of predicates in $\Sigma$ is bounded.

Proof. (Membership) Let $\Sigma' = \text{MFA}(\Sigma)$, let $I_{\Sigma'}^0, I_{\Sigma'}^1, \ldots$ be the chase sequence for $I$ and $\Sigma'$, and let $\varphi, p$, and $a$ be as stated in the proof of Proposition 5. The number of different atoms that can be constructed from $\varphi$ terms is bounded by $k = p \cdot \varphi^a$; note that this is doubly exponential even if $a$ is bounded. Let $k' = k + 3$ and consider $I_{\Sigma'}^k$. If $I_{\Sigma'}^k = I_{\Sigma'}^{k+2}$, then $I_{\Sigma'}^{k+3} = I_{\Sigma'}^{k+2}$, so $\Sigma$ is MFA if and only if $C \subseteq I_{\Sigma'}^k$. Otherwise, we have $I_{\Sigma'}^{k+1} \subseteq I_{\Sigma'}^k$; but then, $I_{\Sigma'}^{k+1}$ clearly contains at least one cyclic term $t = f_i^t(\bar{x})$ such that $t'$ is a subterm
Let \( f \). Since \( I^k_c \), satisfies Claims 1–3 from the proof of Proposition 4, we have \( S(t_i, t) \in I^{k+2}_\Sigma \); finally, by rule (12) and the fact that rules without functional terms are applied before rules with functional terms, we have \( C \in I^k_c \). But then, \( C \in I^k_c \), so by Proposition 4 set \( \Sigma \) is not MFA.

(Hardness) We prove the claim by a reduction from the problem of checking whether \( I \cup \Sigma \models Q \), where \( \Sigma \) is a \textit{weakly acyclic} set of rules without equality and with predicates of bounded arity, \( I \) is an instance, and \( Q = \exists y. \xi(y) \) is a Boolean conjunctive query. It is known that if \( \Sigma \) is weakly acyclic, then \( \Sigma \) is SWA (Marnette 2009); by Theorem 13 then \( \Sigma \) is MFA. Cali, Gottlob, and Pieris (2010) shows that, for such \( I \), \( \Sigma \), and \( Q \), deciding \( I \cup \Sigma \models Q \) is 2ExpTime-complete.

Without loss of generality we assume that the rules in \( \Sigma \) 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 \( y_r \), add an atom \( O_r(y_r) \) to the body of \( r \) for \( O_r \) a fresh predicate, and add a fact \( O_r(c) \) 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 \( Q \) does not contain constants.

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

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

\[
\hat{I} = \bigwedge_{P(v_{c_1}, \ldots, v_{c_k}, w') \in I} \hat{P}(v_{c_1}, \ldots, v_{c_k}, w') \tag{13}
\]

Let \( U \) be a fresh unary predicate, and let \( B \) be a fresh binary predicate. Then, rules \( r_Q \) and \( r_I \) are defined as shown in (14) and (15), respectively.

\[
r_Q = \hat{\xi}(\hat{y}, w) \rightarrow U(w) \tag{14}
\]

\[
r_I = U(w) \rightarrow \exists w', \hat{v}_c[B(w, w') \land \hat{I}] \tag{15}
\]

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

The following property (\( \odot \)) follows immediately from the definition of \( r_Q, \Sigma \), and \( \hat{I} \); for \( \theta \) be an arbitrary substitution that maps \( \hat{v}_c \) and \( w \) to distinct terms, \( \hat{I} \) is MFA, \( \Sigma \) is MFA as well, and so the chase for \( \hat{I} \) terminates.

We next show that \( \Omega \) is not universally MFA if and only if \( I \cup \Sigma \models Q \). Let \( I' \) be the chase of \( I^k_\hat{r} \) and \( \Omega \). Let \( f \) be the function symbol used to skolemise \( \exists w' \) in (15); let \( g_c \) be the function symbols used to skolemise \( \exists v_c \) in (15); and let \( \theta \) be a substitution mapping \( w' \) to \( f(*) \) and each variable \( v_{c_i} \) to \( g_c(*) \). Clearly, \( U(*) \in I' \), so by rule (15) we have \( \hat{I} \cup \{r_Q \} \models U(f(*)) \) and only if \( U(f(*)) \in I' \); by property (\( \odot \)) we have that \( U(f(*)) \in I' \) if and only if \( I \cup \Sigma \models Q \).

Assume now that \( U(f(*)) \not\in I' \). Then, the body of rule (15) is not satisfied, so \( I' \) does not contain term \( f(f(*)) \). Together with property (\( \odot \)), we conclude that \( I' \) does not contain a cyclic term, so by Proposition 4 we have that \( \Omega \) is MFA.

The results of Theorem 6 are somewhat discouraging: acyclicity according to existing criteria can 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. In the rest of this section, we show that the complexity is lower for rules of the following shape.

**Definition 7.** A rule \( r \) of the form (1) is an \( \exists \)-1 rule if \( \hat{y} \) is empty or \( \hat{x} \) contains at most one variable.

As discussed later on, \( \exists \)-1 rules capture (extensions of) Horn DLs. We next show that BCQ answering and MFA checking for \( \exists \)-1 rules is easier than for the general rules. The following theorem provides the upper bound; the lower bounds are given later on for a smaller class of rules that capture DLs.

**Theorem 8.** Let \( \Sigma \) be a set of \( \exists \)-1 rules, and \( \hat{I} \) be an instance. Checking whether \( \Sigma \) is MFA w.r.t. \( I \) is in ExpTime. Furthermore, if \( \Sigma \) is MFA, then answering a BCQ over \( \Sigma \) and \( \hat{I} \) is in ExpTime as well.

**Proof.** Let \( \Sigma' = MFA(\Sigma) \). Since \( \Sigma \) contains only \( \exists \)-1 rules, \( \Sigma' \) also contains only \( \exists \)-1 rules; consequently, all functional terms in \( sk(\Sigma') \) are of the form \( f(x) \). But then, the total number of different monicyclic terms is \( \varphi = c \cdot f^I \), 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 \( p \cdot \varphi^p \), where \( p \) is the number of predicates and \( a \) is the maximum arity of a predicate in \( \Sigma' \). Note that this is exponential even if \( a \) is bounded. As in the proof of Proposition 5, we can now show that the either the chase for \( \Sigma' \) and \( \hat{I} \) terminates or a cyclic term is derived in exponential time, which proves that the complexity of checking whether \( \Sigma \) is MFA w.r.t. \( I \) is in ExpTime.

Finally, since \( I^k_c \subseteq I^k_c \), if \( \Sigma \) is MFA, then \( I^k_c \) can be computed in exponential time, so a BCQ over \( \Sigma \) 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 sk(Σ) 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 define model-summarising acyclicity, which “summarises” the models of Σ by using the same constant to satisfy an existential quantifier, instead of introducing deeper terms.

Definition 9. Let S, D, and F_P be as in Definition 2; furthermore, for each rule r = ϕ(⃗x, ⃗z) → ∃⃗yψ(⃗x, ⃗y) and each variable y_i in ⃗y, let c^r_i be a fresh constant unique for r and y_i. Then, MSA(r) is the following rule, where θ is a substitution that maps each variable y_i in ⃗y to c^r_i:

ϕ(⃗x, ⃗z) → ψ(⃗x, y)θ ∧ ∩ y_i∈⃗y |F_P^r(y_i)θ ∧ ∩ x_j∈⃗x S(x_j, y_i)θ|

For Σ, a set of rules, MSA(Σ) is the smallest set that contains MSA(r) for each rule r ∈ Σ, rules (10)-(11), and rule (12) (12) instantiated for each predicate F_P. Set Σ is model-summarising acyclic (MSA) w.r.t. an instance I if I ∪ MSA(Σ) /∈ C; furthermore, Σ is universally MSA if Σ is MSA w.r.t. I^∞.

Note that MSA(Σ) is a set of datalog rules; hence, MSA can be checked using any datalog engine. This connection with datalog provides the complexity upper bound of 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 reasoning in datalog is O(n^v) where v is the max. number of variables in a rule and n is the size of Σ; hence, we expect MSA checking to be feasible if the rules in Σ are ‘short’.

Theorem 10. 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 coNP and coNP-hard, respectively, if the arity of the predicates in Σ is bounded.

Proof. (Membership) Let Σ' = MSA(Σ). Note that the total number of atoms occurring in a clause of I and Σ' is p · c^n, where p is the number of predicates, c is the number of constants, and n is the maximum arity of the predicates in Σ'; this number is clearly exponential if n is not bounded. The rest of the proof is the same as in Theorem 6. If n is bounded, then the number of atoms becomes polynomial; hence, to check whether I ∪ Σ' /∈ C, one can guess a proof for C, that is 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 ∪ Σ' /∈ C is in NP; since I ∪ Σ' /∈ C if and only if Σ = 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 ∪ Σ |= Q is ExpTime-complete in general (Dantsin et al. 2001). Furthermore, checking whether Σ ∪ I |= Q is already NP-hard even if the arity of predicates is bounded; so checking I ∪ Σ |= Q is NP-hard as well.

Let Ω be defined as in the proof of Theorem 6. The proof that Ω is not MSA if and only if I ∪ Σ |= Q is analogous to the proof of Theorem 6, and we omit it for the sake of brevity.

We finish this section by proving strict inclusion relationships between MFA, MSA, and SWA. In particular, Theorem 11 and Example 12 show that that MFA is strictly more general than MSA.

Theorem 11. 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 Σ_1 = MFA(Σ) and Σ_2 = MSA(Σ). Furthermore, let h be a mapping of ground terms to constants such that h(t) = c if t is of the form f^r_c(...), and h(t) = t if t is a constant; for an atom A = P(t_1,...,t_n), let h(A) = P(h(t_1),...,h(t_n)); and for an instance I, let h(I) = {h(A) | A ∈ I}. Finally, let P_{Σ_2} ⊆ P_{Σ_1} be the chase sequence for I and Σ_1, and let I_{Σ_2} ⊆ I_{Σ_1} be the chase sequence for I and Σ_2. Note that sk(Σ_2) = Σ_2 differs from sk(Σ_1) only in that the former contains the constant c^r_i in place of each functional term f^r_i(⃗x). Thus, by a straightforward induction on i, one can show that h(I_{Σ_2}) ⊆ P_{Σ_2}, for each i; this implies h(P_{Σ_2}) ⊆ P_{Σ_2}. Consequently, C ⊆ P_{Σ_2} clearly implies C ⊆ P_{Σ_2}; hence, if Σ is MSA, then Σ is MFA as well, as required.

Example 12. Let Σ be the set of rules (16)-(19).

\[ A(x) \rightarrow \exists y.R(x,y) \land B(y) \]
\[ B(x) \rightarrow \exists y.S(x,y) \land T(y,x) \]
\[ A(z) \land S(z,x) \rightarrow C(x) \]
\[ C(z) \land T(z,x) \rightarrow A(x) \]

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 MFA, but not SWA.

Theorem 13. If a set of equality-free rules Σ is SWA, then Σ is universally MFA as well.

Before proving Theorem 13, 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 ≤ i ≤ n. For sets of places P and P', we write P ⊆ P' if, for each place A_i ∈ P, a place A'_i ∈ P' and substitutions σ and σ' exist such that Aσ = A'σ'. Let Σ be a set of equality-free rules (SWA is defined only for rules that do not contain equality). Let r ∈ Σ be a rule of the form (1), and let w be a variable; then, ln(r, w) (resp. Out(r, w)) is the set of all places A_i such that A ∈ ϕ(⃗x, ⃗z) (resp. A ∈ ψ(⃗x, ⃗y)) and A contains w at position i; Out(r, w) = \{A_i | A_i ∈ Out(r, w)\} for θ the substitution used to skolemise r; finally, MoveΣ(r, w) is the smallest set of places that contains Out(r, w) such that, for each r' ∈ Σ and each
variable $w'$ occurring in $r'$, if $\text{ln}(r', w') \subseteq \text{Move}_2(r, w)$, then $\text{Out}(r', w') \subseteq \text{Move}_2(r, w)$. A rule $r$ triggers a rule $r'$ in $\Sigma$ 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 $\text{ln}(r', x') \subseteq \text{Move}_2(r, y)$. Set $\Sigma$ is SWA if its triggers relation is acyclic.

**Proof of Theorem 13.** SWA is applicable to $\Sigma$ only if $\Sigma$ contains the explicit axiomatisation of the equality predicate; thus, we assume the latter to be the case and we consider $\approx$ to be a regular predicate. Let $\Sigma' = \text{MSA}(\Sigma)$, let $\rho^0, \rho^1, \ldots$ be the chase sequence for $I^*_\Sigma$ and $\Sigma'$, and let $I^\infty$ be the chase of $I^*_\Sigma$ and $\Sigma'$. Furthermore, let $\rho$ be the function that maps constants to themselves and that is defined on ground functional terms as $\rho(f^i_r(t)) = c^i_r$. 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 $P(t) \in I^{\infty}$ where $P \notin \{S, D, C\}$, and each $t_j \in t^i$ such that $t_j = c^i_r$ is the constant used to replace $y_i$ in $r$, a substitution $\theta$ and a place $A_{j} \in \text{Move}_2(r, y_i)$ exist such that $P(t) = \rho(\theta A)$. The proof is by induction on the length of the chase. Since $\rho(\rho(t_1, \ldots, t_n)) = P(\rho(t_1), \ldots, \rho(t_n))$, we have $\rho^0 = I^*_\Sigma$ does not contain a constant of the form $c^i_r$, property (⊗) holds vacuously for $I^0$. Assume now that property (⊗) holds for some $I^k$, and consider arbitrary rule $r \in \Sigma$, variable $y_i$, fact $P(t) \in I^{k+1}$ with $P \notin \{S, D, C\}$, and $t_j \in t^i$ such that $t_j = c^i_r$. Fact $P(t)$ is derived in $I^{k+1}$ from the head atom $H$ of some rule $r^{k+1} \in \text{MSA}(\Sigma)$. Let $\sigma$ be the substitution used in the rule application; clearly, we have $H \sigma = P(t)$. Furthermore, let $r^{k+1} \in \Sigma$ be the rule that produces $r^{k+1} \in \text{MSA}(\Sigma)$, 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 $H^3 \sigma = P(t)$. Now if $H$ contains $c^i_r$ in position $j$, then $r = r^1$ since $r^1$ is the only rule that contains $c^i_r$: thus, $H^3|_j \in \text{Out}(r, y_i) \subseteq \text{Move}_2(r, y_i)$, so property (⊗) holds. Otherwise, $H$ contains at position $j$ a universally quantified variable $x$ such that $\sigma(x) = c^i_r$. Let $B_1, \ldots, B_3$ be the body atoms of $r^3$ that contain $x$: clearly, $\{B_1, B_2, B_3\} \subseteq I^{k-1}$. All these atoms satisfy the induction assumption, so for each $B_m \in \{B_1, \ldots, B_3\}$ and each $\ell$ such that $B_m$ contains variable $x$ at position $\ell$, a place $B'_m|_{|\ell} \in \text{Move}_2(r, y_i)$ and substitution $\theta_m$ exist such that $B_m \sigma = B'_m \theta_m$. Let $\sigma'$ be the modification of $\sigma$ obtained by setting $\sigma'(w) = \theta_m(w)$ for each variable $w$ for which $\theta_m(w)$ is a functional term; clearly, we have $B_m \sigma' = B'_m \theta_m$. But then, we have $\text{ln}(r^1, x) \subseteq \text{Move}_2(r, y_i)$; by the definition of $\text{Move}_2$, we then have $H^3|_j \in \text{Move}_2(r, y_i)$, and property (⊗) holds.

We additionally prove the following property (\wedge): if $S(c^i_r, c'^i_r) \in I^{\infty}$ 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 $c^i_r$, and let $k$ be the smallest integer such that $S(c^i_r, c'^i_r) \in I^k$. Clearly, $S(c^i_r, c'^i_r)$ is derived in $I^k$ from the head atom $S(x, c'^i_r)$ of rule $r'$. Let $\sigma$ be the substitution used in the rule application; thus, $\sigma(x) = c'^i_r$. Let $B_1, \ldots, B_n$ be the body atoms of $r$ that contain $x$; clearly, we have $\{B_1, \ldots, B_n\} \subseteq I^{k-1}$. All these atoms satisfy property (⊗) for each $B_m \in \{B_1, \ldots, B_n\}$ and each $\ell$ such that $B_m$ contains variable $x$ at position $\ell$, a place $B'_m|_{|\ell} \in \text{Move}_2(r, y_i)$ and substitution $\theta_m$ exist such that $B_m \sigma = B'_m \theta_m$. But then, as in the previous paragraph we have $\text{ln}(r^1, x) \subseteq \text{Move}_2(r, y_i)$; hence, $r$ triggers $r'$.

Assume now that $\Sigma$ is not MSA, so $C \in I^{\infty}$: due to rules (12), we have $\{F^i_r(t), D(t, t'), F^j_r(t')\} \subseteq I^{\infty}$ for some $F^i_r$. But then, since predicate $F^i_r$ occurs in $\Sigma'$ only in an atom $F^i_r(c^i_r)$, we have $t = t' = c^i_r$. Finally, since $D$ is axiomatised in $\Sigma'$ as the transitive closure of $\Sigma$, clearly $r$ triggers itself, and so $\Sigma$ 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 slightly more robust but still fail to capture many practically relevant rule sets.

**Example 14.** Consider the following set of rules.

\begin{align*}
A(x) \land B(x) &\rightarrow \exists y.[R(x, y) \land B(y)] & (20) \\
R(z, x_1) \land R(z, x_2) &\rightarrow x_1 \approx x_2 & (21)
\end{align*}

On these rules and the critical instance, the skolem chase derives the following infinite set of facts.

\begin{align*}
R(\ast, f(\ast)) \quad B(f(\ast)) &\quad \ast \approx f(\ast) \quad A(f(\ast)) \\
R(f(\ast), f(f(\ast))) \quad B(f(f(\ast))) &\quad \ldots
\end{align*}

Example 14 shows that equalities between terms tend to proliferate during chase, which can lead to non-termination. Interestingly, the rules in the example are WA. This is because WA is sufficient for termination of the 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 14 shows that this is not the case for rules with equality.

Marnette (2009) proposed a solution to this problem based on a technique called *singularisation*. 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 $\Sigma$ 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 $\Sigma$ 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 $w \in \bar{F} \cup \bar{\Sigma}$ to a single occurrence of $w$ in $\varphi$; all other
variable occurrences are unmarked and all constants are also unmarked. A marking $M$ of $\Sigma$ has exactly one marking $M_r$ for each $r \in \Sigma$. The singularity of $\Sigma$ under $M$ is the set $\text{Sing}(\Sigma, M)$ 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), (3), (4).

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

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

**Example 15.** Singularity of the marked rule (22) produces rule (23). Note that singularity is applied ‘globally’ to all rules, even to those without equality.

$$A(\hat{x}) \land B(x) \land R(x, \hat{z}) \rightarrow C(x) \quad (22)$$

$$A(x) \land B(x) \land R(x, \hat{z}) \land x \approx x_1 \land x \approx x_2 \rightarrow C(x) \quad (23)$$

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

**Example 16.** Rule (20) from Example 14 admits the following two markings:

$$A(\hat{x}) \land B(x) \rightarrow \exists y[R(x, y) \land B(y)] \quad (24)$$

$$A(x) \land B(\hat{x}) \rightarrow \exists y[R(x, y) \land B(y)] \quad (25)$$

The skolem chase does not universally terminate for the singularity obtained from (25); in contrast, the singularity obtained from (24) is JA.

We use $\text{MFA}^3$ and $\text{MFA}^v$ to denote the classes of rule sets that are in $\text{MFA}$ for some singularity and for all singularisations, respectively; notions $\text{MSA}^3$, $\text{MSA}^v$, $\text{JA}^3$, and $\text{JA}^v$ are defined analogously. Clearly, $X^v \subseteq X^3$ for each $X \in \{\text{MFA}, \text{MSA}, \text{JA}, \text{JA}^v\}$. Example 16 shows this inclusion to be proper.

**Theorem 17.** $\text{JA}^v = \text{WA}$.

**Proof.** ($\text{JA}^v \subseteq \text{WA}$) We prove the contrapositive, so assume that $\Sigma \notin \text{WA}$. 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 $\Sigma$. 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 $x_{p-q}$ denote one (arbitrary but fixed) such variable; we say that $x_{p-q}$ 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 a cycle in the dependency graph. Let $p_i \rightarrow p_{i+1}$ and $p_j \rightarrow p_{j+1}$ be two edges (regular or special) such that $x_{p_i-p_{i+1}} = x_{p_j-p_{j+1}}$. If $p_i \rightarrow p_{i+1}$ is special, then a special edge $p_j \rightarrow p_{j+1}$ exists, and we obtain a shorter cycle by replacing the path between $p_i$ and $p_j$ with $p_i \rightarrow p_{j+1}$. 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 $p_i \rightarrow p_{i+1}$ and $p_j \rightarrow p_{j+1}$ exist. If the path between $p_{j+1}$ and $p_j$ contains a special edge, we obtain a shorter cycle by replacing the path between $p_i$ and $p_{j+1}$ with $p_j \rightarrow p_{j+1}$. 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 through a special edge to which each variable contributes at most one edge such that $p_{n+1} = p_1$, and let $x_1 = x_{p_1-p_2}, \ldots, x_n = x_{p_n-p_1}$ be the corresponding contributing variables. Let $M$ be a marking for $\Sigma$ where each $x_i$ is marked in position $p_i$ in some body atom of its rule. We claim that $\text{Sing}(\Sigma, M)$ is not JA.

Consider a subpath $q_1 \rightarrow \ldots \rightarrow q_m$ $(m \geq 3)$ of $\Pi$ such that $q_1 \rightarrow q_2$ and $q_{m-1} \rightarrow q_m$ 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 $q_1 \rightarrow q_2$, and let $w$ be an existentially quantified variable at position $q_{m-1}$ that was used to justify the special edge $q_{m-1} \rightarrow q_m$. We claim that the JA dependency graph has an edge from $v$ to $w$.

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

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

\(\text{JA}^v \supseteq \text{WA}\) Assume that $\Sigma \notin \text{JA}^v$. Then a marking $M$ for $\Sigma$ exists such that $\text{Sing}(\Sigma, M)$ is not JA. Consider some existentially quantified variable $v$. For each position $p \in \text{Move}(v)$ (w.r.t. $\text{Sing}(\Sigma, M)$) where $p$ does not involve the $\approx$ predicate, there is a path of regular edges $p_1 \rightarrow \ldots \rightarrow p_m = p$ in the WA dependency graph of $\Sigma$ such that $v$ occurs on position $p_1$; this property $(*)$ can be easily shown by induction over the construction of $\text{Move}(v)$, and we omit the details for the sake of brevity. Thus, by $(*)$ and the definition of the JA dependency graph, for each edge $v \rightarrow w$ in the JA dependency graph, a path $p_1 \rightarrow \ldots \rightarrow p_m \rightarrow p_{m+1}$ exists in the WA dependency graph of $\Sigma$ where $p_m \rightarrow p_{m+1}$ is a special edge and all other edges are regular. Clearly, we have $\Sigma \notin \text{WA}$. \qed
Checking all possible markings may not be feasible (there are exponentially many in the total number of variables occurring more than once in a rule body). Theorem 17 shows that $\text{JA}^\Sigma$ can be decided using WA. For the other cases, the following lemma shows how to reduce the number of markings.

**Lemma 18.** Let $M$ and $M'$ be markings for $\Sigma$ 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 $\text{Sing}(\Sigma, M)$ is $\text{JA}/\text{MSA}/\text{MFA}$ if and only if $\text{Sing}(\Sigma, M')$ is $\text{JA}/\text{MSA}/\text{MFA}$.

Despite this optimisation, the number of markings to check is still exponential; hence, we next describe a useful approximation. Let $\text{Sing}_k(\Sigma) = \bigcup_{M \in \mathcal{M}} \text{Sing}(\Sigma, M)$, where $\mathcal{M}$ is a set of all markings for $\Sigma$ that agree on all variables occurring only in the body of a rule in $\Sigma$. By Lemma 18, it is irrelevant how the markings of body variables are defined in $\mathcal{M}$. Let $\text{MFA}^\Sigma$ be the class containing rule sets $\Sigma$ for which $\text{Sing}_k(\Sigma)$ is in $\text{MFA}$; $\text{MSA}^\Sigma$ and $\text{JA}^\Sigma$ are defined analogously. As the following theorem shows, $\text{Sing}_k(\Sigma)$ provides a ‘lower bound’ on the result attainable via singularisation.

**Theorem 19.** For each $X \in \{\text{MFA}, \text{MSA}, \text{JA}\}$, we have that $X^\Sigma \subseteq X^\Sigma_k$. The size of $\bigcup_{M \in \mathcal{M}} \text{Sing}(\Sigma, M)$ is exponential in the maximal number of variables that occur more than once in the body of any one rule in $\Sigma$, and it is linear in the number of rules in $\Sigma$.

**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 $\Sigma$ occurs $k$ times in $\bigcup_{M \in \mathcal{M}} \text{Sing}(\Sigma, M)$, 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 $\text{Sing}_k(\Sigma)$ is linear in the size of $\Sigma$. For the general case, we can obtain the same complexity bounds despite the exponential increase in the number of rules:

**Theorem 20.** Deciding whether $\Sigma$ is in $\text{MFA}^\Sigma$ (MFA$^\Sigma$) is $2\text{ExpTime}$-complete. Deciding whether $\Sigma$ is in $\text{MSA}^\Sigma$ (MSA$^\Sigma$) is $\text{ExpTime}$-complete.

**Proof.** If $\Sigma$ contains no equality, it is easy to see that $\Sigma$ is in $\text{MFA}^\Sigma$ (MFA$^\Sigma$) iff it is in MFA. The same can be observed for MSA. Hardness thus follows from Theorem 6 and Theorem 10.

For membership, we first consider the case of MFA$^\Sigma$, MFA$^\Sigma$ and MSA$^\Sigma$, MSA$^\Sigma$. Each of these properties can be decided by considering all of the at most exponentially many markings. Since $\text{Sing}(\Sigma, M)$ is linear in the size of $\Sigma$, the property can be checked for each marking in $2\text{ExpTime}$ for MFA; Theorem 6 and ExpTime (for MSA; Theorem 10). This yields the required bound since an exponential factor is not significant for the considered complexity classes.

For the case of $\text{MFA}^\Sigma$ and $\text{MSA}^\Sigma$, membership follows by observing that the membership of $\text{MFA}$ and MSA in $2\text{ExpTime}$ 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 $\text{Sing}_k(\Sigma)$ is exponentially larger than $\Sigma$, 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 $2\text{ExpTime}/\text{ExpTime}$.

**Acyclicity of DL Ontologies**

We now turn our attention to 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: $R^-(t_1, t_2)$ 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, in which concepts are not nested. This is w.l.o.g., as each ontology can be normalised using a linear algorithm, 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-SRIQ TBox $T$ consists of axioms shown in 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, $T$ 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 as a nondeterministic finite automaton. Apart from Horn-SRIQ, we also consider Horn-SRI $T$Boxes, which do not contain rules of Type 2, as well as Horn-SHIQ $T$Boxes, where $R = S = T$ in all rules of Type 6; all Horn-SHIQ $T$Boxes are regular.

Each Horn-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 $T$Box $T$ unsatisfiable w.r.t. an instance $I$. This
Due to this close correspondence between DL axioms and existential rules, this automaton can be exponential, which may require one to examine all nodes in the tree-like structures; although regularity ensures that matching lower bound for WA Horn-Rules Σ is acyclic; and regularity ensures that need expressivity beyond what is available in OWL. For arbitrary input $S_I$ be an instance, and let $F$ be a fact. Then, checking whether $I \cup T \models F$ is EXPTime-hard.

**Theorem 21.** Let $T$ be a WA Horn-SRI TBox, let $I$ be an instance, and let $F$ be a fact. Then, checking whether $I \cup T \models F$ is EXPTime-hard.

**Proof.** Let $M = (S, Q, \delta, Q_0)$ be a deterministic Turing machine, where $S$ is a finite set of states, $Q$ is a finite set of states that contains the accepting state $Q_0$, $\delta : Q \times S \to Q \times S \times \{\leftarrow, \rightarrow\}$ is a transition function, and $Q_0 \in Q$ is an initial state. Furthermore, assume that an integer $k$ exists such that $M$ halts on each input of length $n$ in time $2^n k$. For arbitrary input $S_i$, ... $S_n$, we construct an MFA set of Horn-SRI rules $T$ and an instance $I$ such that $I \cup T \models Q_0(a)$ if and only if $M$ accepts the input. To simplify the presentation, we will use a slightly more general syntax for the rules in $T$ 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, $\ell$ 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 $T$. Set $T$ will contain only Horn rules without empty heads, so it will be satisfiable in a minimal Herbrand model. For readability, we will break $T$ into parts and prove for each part various facts about this minimal Herbrand model.

The first part of $T$ contains rules (26)–(28) for each $i > 0$, and rule (29) for each $i > 1$.

- $A_{i-1}(x) \rightarrow \exists y, [L_i(x, y) \land A_i(y)]$ (26)
- $A_1(x) \rightarrow \exists y, [y \land A_1(y)]$ (27)
- $L_i(x, z) \land R_i(z, x') \rightarrow T_i(x, x')$ (28)
- $R_i(z, x) \land T_i(z, z') \land R_i(z', x') \rightarrow T_i(x, x')$ (29)

On $I$, these rules axiomatise existence of a triangular structure in the top part of Figure 1 containing $T_i$ links.

The second part of $T$ contains rule (30), rules (31)–(33) for each $i > 0$, and rule (34) for each $i > 1$.

- $A_i(x) \rightarrow B_0(x)$ (30)
- $B_{i-1}(x) \rightarrow \exists y, [U_i(x, y) \land B_i(y)]$ (31)
- $B_1(x) \rightarrow \exists y, [D_1(x, y) \land B_1(y)]$ (32)
- $U_i(x, z) \land D_i(z, x') \rightarrow V_i(x, x')$ (33)
- $D_i(x, z) \land V_{i-1}(z, z') \land U_i(z', x') \rightarrow V_i(x, x')$ (34)

These rules axiomatise existence of triangular structures in the bottom part of Figure 1 containing $V_i$ links.

The third part of $T$ contains rule (35), and rules (36) and (37) for each $i > 0$.

- $T_i(x, x') \rightarrow H_0(x, x')$ (35)
- $U_i(x, z) \land H_{i-1}(z, z') \land U_i(z', x') \rightarrow H_i(x, x')$ (36)
- $D_i(x, z) \land H_{i-1}(z, z') \land D_i(z', x') \rightarrow H_i(x, x')$ (37)

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 1. In the rest of this proof we abbreviate conjunctions of the form $R_1(x_0, x_1) \land \ldots \land R_{\ell}(x_{\ell-1}, x_{\ell})$ as $R(x_0, x_{\ell})$, and we use an analogous abbreviation for $U_i, \ldots, U_{\ell}$.

The fourth part of $T$ contains rule (38), rules (39) and (40) for each $1 \leq j \leq n$, and rules (41)–(42), where
\( S_0 \) is the empty tape symbol. Remember that the input to \( \mathcal{M} \) is given as \( S_1, \ldots, S_{n_i} \).

\[
\begin{align*}
A_0(z) \land R^\ell(z, z') \land U^\ell(z', x) &\rightarrow O_1(x) \land Q_0(x) \quad (38) \\
O_j(z) \land V_\ell(z, x) &\rightarrow O_{j+1}(x) \quad (39) \\
O_j(x) &\rightarrow S_{i_j} \quad (40) \\
O_{n+1}(z) \land V_\ell(z, x) &\rightarrow O_{n+1}(x) \quad (41) \\
O_{n+1}(x) &\rightarrow S_{i_{n+1}}(x) \quad (42)
\end{align*}
\]

Rule (38) labels the grid origin and sets the initial state as shown in Figure 1. Rules (39) ensure that the \( n \) subsequent nodes are labelled with \( O_2, \ldots, O_{n+1} \), and rule (41) propagates \( n+1 \) to the rest of the \( V_\ell \)-chain. Finally, rules (40) and (42) ensure that nodes labelled with \( O_j \) are also labelled with \( S_{i_j} \), and that nodes labelled with \( O_{n+1} \) are labelled with \( S_{i_{n+1}} \). Thus, this part of \( \mathcal{T} \) ensures that the right-most \( V_\ell \)-chain in the grid contains the initial state of the tape of \( \mathcal{M} \).

The fifth part of \( \mathcal{T} \) contains rules (43)–(44) for each \( Q_k \in \mathcal{Q} \) and rules (45)–(46). These rules essentially ensure that all nodes before and after a node labelled with some \( Q_k \in \mathcal{Q} \) are labelled with \( N_1 \) and \( N_2 \), respectively, thus indicating that the head is not above the node.

\[
\begin{align*}
Q_k(z) \land V_\ell(x, z) &\rightarrow N_1(x) \quad (43) \\
Q_k(z) \land V_\ell(z, x) &\rightarrow N_2(x) \quad (44) \\
N_1(z) \land V_\ell(x, z) &\rightarrow N_1(x) \quad (45) \\
N_2(z) \land V_\ell(x, z) &\rightarrow N_2(x) \quad (46)
\end{align*}
\]

The sixth part of \( \mathcal{T} \) contains rules (47)–(48) instantiated for each \( S_k \in \mathcal{S} \); these rules ensure that the contents of the tape is copied for all nodes that do not contain the head.

\[
\begin{align*}
N_1(z) \land S_k(z) \land H_\ell(z, x) &\rightarrow S_k(x) \quad (47) \\
N_2(z) \land S_k(z) \land H_\ell(z, x) &\rightarrow S_k(x) \quad (48)
\end{align*}
\]

The seventh part of \( \mathcal{T} \) contains rules (49)–(50) instantiated for each \( S_k \in \mathcal{S} \) and each \( Q_k \in \mathcal{Q} \) such that \( \delta(Q_k, S_k) = (Q_{k'}, S_{k'}, \rightarrow) \). These rules encode moves of \( \mathcal{M} \) where the head moves up.

\[
\begin{align*}
Q_k(z) \land S_k(z) \land H_\ell(z, x) &\rightarrow S_{k'}(x) \quad (49) \\
Q_k(z) \land S_k(z) \land H_\ell(z, z') \land V_\ell(x, z') &\rightarrow Q_{k'}(x) \quad (50)
\end{align*}
\]

The eighth part of \( \mathcal{T} \) contains rules (51)–(52) instantiated for each \( S_k \in \mathcal{S} \) and each \( Q_k \in \mathcal{Q} \) such that \( \delta(Q_k, S_k) = (Q_{k'}, S_{k'}, \rightarrow) \). These rules encode moves of \( \mathcal{M} \) where the head moves down.

\[
\begin{align*}
Q_k(z) \land S_k(z) \land H_\ell(z, x) &\rightarrow S_{k'}(x) \quad (51) \\
Q_k(z) \land S_k(z) \land H_\ell(z, z') \land V_\ell(z', x) &\rightarrow Q_{k'}(x) \quad (52)
\end{align*}
\]

The ninth part of \( \mathcal{T} \) contains rules (53)–(56) for each \( 1 \leq i \leq \ell \); these rules simply ensure that acceptance is propagated back to the root of the upper tree.

\[
\begin{align*}
Q_a(z) \land U_i(z, x) &\rightarrow Q_a(x) \quad (53) \\
Q_a(z) \land D_i(x, z) &\rightarrow Q_a(x) \quad (54) \\
Q_a(z) \land L_i(x, z) &\rightarrow Q_a(x) \quad (55) \\
Q_a(z) \land R_i(x, z) &\rightarrow Q_a(x) \quad (56)
\end{align*}
\]

The above discussion shows that labelling of the nodes in the grid shown in Figure 1 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 \( V_f \)-chain, and \( H_f \)-links connect tape cells at successive time instants. Thus, \( I \cup T \models Q_\alpha(a) \) if and only if \( M \) accepts \( S_1, \ldots, S_n \) in time \( 2^k \). It is straightforward to see that \( T \) is WA, so the claim of this theorem holds. \( \square \)

Note that Theorem 21 applies to Horn-SRI and thus does not rely on a particular treatment of equality.

The proof of Theorem 21 can be adapted to obtain the lower bound for checking MFA of Horn-SRI rules.

**Proposition 22. Checking whether a Horn-SRI TBox is universally MFA is ExpTime-hard.**

**Proof.** Let \( M \) be an arbitrary deterministic Turing machine and let \( S_1, \ldots, S_n \) be an input string on which \( M \) terminates in time \( 2^n k \). For each \( M \) and \( S_1, \ldots, S_n \), let \( T \) be as in the proof of Theorem 21. Furthermore, let \( T' \) be the extension of \( T \) with the following rule:

\[
Q_\alpha(x) \rightarrow \exists y. [B(x, y) \land A_\alpha(y)] 
\]

(57)

By an argument analogous to the one used in the proof of Theorem 6, one can see that \( T' \) is not universally MFA if and only if \( M \) accepts \( S_1, \ldots, S_n \). \( \square \)

As we show next, however, the complexity or query answering drops to PSPACE for MFA Horn-SHITQ ontologies. In contrast, checking entailment of a single answer drops to \( \text{PSpace} \) for Horn ontologies since it uses concepts with disjunctions. Checking by Baader et al. (2007) is not applicable to Horn-SRI since it uses concepts with disjunctions.

**Theorem 23. Let \( T \) be Horn-SHITQ TBox, let \( I \) be an instance such that \( T \) is MFA w.r.t. \( I \), and let \( Q \) be a BCQ. Then, deciding \( I \cup T \models Q \) is PSPACE-complete.**

**Proof (Membership).** Assume that \( Q \) is of the form \( Q = \exists y_1 B_1 \land \ldots \land B_n \), let \( T = \text{sk}(T) \), let \( f \) be the number of function symbols in \( T \), and let \( c \) be the number of constants in \( I \). Since \( \bot \) is just a regular atomic concept, \( I \cup T \) is always satisfiable in the chase \( I^\infty \) of \( I \) and \( T \). Furthermore, \( I \cup T \models Q \) if and only if a substitution from the variables in \( y \) to the terms in \( I^\infty \) exists such that \( B_i \theta \in I^\infty_T \) for each \( 1 \leq i \leq n \); the latter clearly holds if and only if \( I \cup \bar{T} \models B_i \theta \). As shown in the proof of Theorem 8, each term in \( I^\infty_T \) is of the form \( g_1(\ldots g_n(a) \ldots) \), where \( n \leq f \). Thus, the first step in deciding \( I \cup T \models Q \) is to examine all possible \( \theta \) and then check \( I \cup \bar{T} \models B_i \theta \); this can clearly be done using a deterministic Turing machine that uses polynomial space to store \( \theta \).

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

\[
E(t') 
\]

(58)

\[
E(z) \land R(z, x) \rightarrow D(x) 
\]

(59)

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

\[
R(x_1, z) \land R(x, z_2) \rightarrow R(x_1, x_2) 
\]

(60)

and then replacing each rule of the form

\[
A(z) \land R(z, x) \rightarrow B(x) 
\]

(61)

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

\[
A(z) \land R(z, x) \rightarrow Q_{A,R,B}(x) 
\]

(62)

\[
Q_{A,R,B}(z) \land R(z, x) \rightarrow Q_{A,R,B}(x) 
\]

(63)

\[
Q_{A,R,B}(x) \rightarrow B(x) 
\]

(64)

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

Let \( \Xi \) be \( T'' \) extended with the equality axioms (3) and (5). Since \( \Xi \) does not occur in the body of the rules in \( T'' \), we have that \( I \cup T'' \not\models D(t) \) if and only if \( I \cup \Xi \not\models D(t) \). Let \( I^\infty \) be the chase for \( I \) and \( \Xi \); then \( I \cup \Xi \not\models D(t) \) if and only if \( D(t) \not\in I^\infty \). Note that \( \Xi \) 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 \( Q_2(t_2) \rightarrow \text{false} \). Thus, can be used in the same way as in (Motik, Shearer, and Horrocks 2009) to show that each assertion in \( I^\infty \) 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 \( \Xi \), and the claim follows straightforwardly from the \( I^\infty \) form of rules of Types 1–5.

Let \( f_1, \ldots, f_n \) be all function symbols occurring in \( \Xi \). 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, $D(t) \not\in I^n_\Sigma$ if and only if a Herbrand interpretation $J$ exists in which all assertions are of the form mentioned above, such that $I \subseteq J$, $I^n_\Sigma \subseteq J$, $J \models \Xi$, and $D(t) \not\in J$. We next show how to check existence of such $J$ using a nondeterministic Turing machine that uses polynomial space. By the algorithm can be implemented using a nondeterministic Turing machine that uses polynomial space.

We first guess an interpretation $J_0 \subseteq I$ for the constants in $I$, and we check whether all rules in $\Xi$ 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 $f_1(s), \ldots, f_n(s)$.
3. If $D(t) \in J^i$, return false.
4. Check whether $J^j$ coincides with each $J^i$, $j < i$ on the common terms; if not, return false.
5. Check whether the equality symmetry rule (5) is satisfied in $J^j$; if not, return false.
6. Check whether $J^j \cup J^j \mathbin{\cup} \ldots \cup J^j$ satisfies each rule in $\Xi$ if the central variable of the rule is mapped to $s$; if this is not the case for any 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 $D(t) \not\in J$. Furthermore, recursion depth of our algorithm is $n$ and at each recursion level we need to keep a polynomially sized interpretation $J$, 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).** Let $\varphi$ be a QBF of the form $\varphi = Q_0 x_1 \ldots Q_n x_n . C_1 \wedge \ldots \wedge C_k$, where $x_1, \ldots, x_n$ are the variables of $\varphi$, each $Q_i \in \{ \exists, \forall \}$ for $1 \leq i \leq n$ is a quantifier, and each $C_j$ for $1 \leq j \leq k$ is a clause of the form $C_j = L_{j,1} \vee L_{j,2} \vee L_{j,3}$. Checking validity of $\varphi$ is the canonical PSPACE-hard problem.

In the proof of this result, for a binary predicate $P$, we abbreviate $P(x_0, x_1) \wedge \ldots \wedge P(x_{m-1}, x_m)$ with $P^m(x_0, x_m)$. Let $T$ be the Horn-SHIQ TBox containing rules (65)–(68) for each $1 \leq i \leq n$, rule (69) for each literal $L_{j,m} = x_k$ in clause $C_j$, rule (70) for each literal $L_{j,m} = \neg x_k$ in clause $C_j$, rule (71), rule (72) for each $1 \leq i \leq n$ such that $Q_i = \exists$, and rule (73) for each $1 \leq i \leq n$ such that $Q_i = \forall$.

- $A_{i-1}(x) \rightarrow \exists y . [X^+_{i}(x, y) \wedge A_{i}(x)]$ (65)
- $A_{i-1}(x) \rightarrow \exists y . [X^-_{i}(x, y) \wedge A_{i}(x)]$ (66)
- $X^+_{i}(x, x') \rightarrow P(x, x')$ (67)
- $X^-_{i}(x, x') \rightarrow P(x, x')$ (68)
- $X^+_{i}(z, z') \wedge P^{\neg \ell}(z, x) \wedge A_{n}(x) \rightarrow C_{j}(x)$ (69)
- $X^-_{i}(z, z') \wedge P^{\neg \ell}(z, x) \wedge A_{n}(x) \rightarrow C_{j}(x)$ (70)
- $C_{j}(x) \wedge \ldots \wedge C_{k}(x) \rightarrow F_{n}(x)$ (71)
- $P(x, z) \wedge F_{n}(z) \rightarrow F_{i-1}(x)$ (72)
- $X^+_{i}(x, z) \wedge F_{j}(z) \wedge X^-_{i}(x, z') \wedge F_{i}(z') \rightarrow F_{i-1}(x)$ (73)

Strictly speaking, rules (69), (70), and (73) are not Horn-SHIQ rules, but they can be transformed into Horn-SHIQ rules by replacing parts of their bodies with fresh concepts. Furthermore, it is straightforward to see what $T$ is WA and, thus, MFA.

Let $I = \{ A_0(a) \}$, and let $I^n_\Sigma$ be the chase of $I$ and $T$. Due to rules (65)–(66), $I^n_\Sigma$ 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 $t$ and $f$, respectively, then each leaf node corresponds to one possible assignment of $x_1, \ldots, x_n$. Rules (69) and (70) then clearly label each leaf node with the clauses that are true in the node, and rule (71) labels each leaf node with $F_{n}$ for which all clauses are true. Finally, rules (72) and (73) label each interior node of the tree with $F_{i-1}$ according to the semantics of the appropriate quantifier of $\varphi$. Clearly, $\varphi$ is valid if and only if $I \cup T \models F_{0}(a)$, which implies our claim.

Although the proof of Theorem 23 takes into account ontology rules with equality (i.e., rules of Type 2), it assumes that equality is axiomatised by $\Sigma_{\Sigma}$ and hence it does not directly apply to singularised Horn-SHIQ rules. We conjecture, however, that the result in the theorem holds regardless of singularisation.

The restriction to Horn-SHIQ rules also makes checking MFA easier. Roughly speaking, checking MFA w.r.t. an instance can be done by a minor variation of the query answering algorithm.

**Theorem 24.** Let $T$ be Horn-SHIQ $TBox$, and let $I$ be an instance. Then, deciding whether $T$ is MFA w.r.t. $I$ is in PSPACE, and deciding whether $T$ is universally MFA is PSPACE-hard.

**Proof.** (Membership) Rules in MFA($T$) are ‘almost’ Horn-SHIQ rules: rule (11) can be made a Horn-SHIQ rule by replacing $5$ 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 23. 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 $\varphi$ be an arbitrary quantified Boolean formula, and let $T$ be as in the hardness proof of Theorem 21. Furthermore, let $T'$ be the extension of $T$ with the following rule:

$$F_0(x) \rightarrow \exists y. [B(x, y) \land A_0(y)] \quad (74)$$

By an argument analogous to the one used in the proof of Theorem 6, one can see that $T'$ is not universally MFA if and only if $\varphi$ is valid.

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

**Theorem 25.** Let $T$ be Horn-$\text{SHIQ}$ TBox, and let $I$ be an instance. Then, deciding whether $T$ is MSA w.r.t. $I$ is in PTime, and deciding whether $T$ is universally MSA is PTime-hard.

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

(Hardness) A monotone circuit $C$ is a finite directed acyclic graph with input vertices $v_1, \ldots, v_n$ and one output vertex $v_o$. Each non-input vertex $v$ in $C$ is assigned a label $\lambda(v) \in \{\land, \lor\}$. A valuation $\mu$ is an assignment of input vertices to $t$ and $f$. The value of $C$ on $\mu$ is an assignment $\mu_C$ of all vertices in $C$ to $t$ and $f$ that coincides with $\mu$ 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 $\lambda(v) = \lor$, then $\mu_C(v) = \mu_C(v_1) \lor \cdots \lor \mu_C(v_n)$.
- If $\lambda(v) = \land$, then $\mu_C(v) = \mu_C(v_1) \land \cdots \land \mu_C(v_n)$.

The problem of deciding for a given $C$ and $\mu$ whether $\mu_C(v_o) = t$ is PTime-hard.

Let $C$ be a monotone circuit. Then, $T$ is the Horn-$\text{SHIQ}$ 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 (75) if $\lambda(v) = \land$ and rule (76) for each $1 \leq i \leq n$ if $\lambda(v) = \lor$.

$$V_i(x) \land \ldots \land V_n(x) \rightarrow V(x) \quad (75)$$

$$V_i(x) \rightarrow V(x) \quad (76)$$

Given an assignment $\mu$, let $I_\mu$ be an instance that, for each input vertex $v_i$ of $C$, contains $V_i(a)$ if and only if $\mu(v_i) = t$. Clearly, $I_\mu \cup T \models V_o(a)$ if and only if $\mu_C(v_o) = t$.

Now let $T'$ be the extension of $T$ with the following rule:

$$V_o(x) \rightarrow \exists y. [B(x, y) \land \bigwedge_{i \in I} V_i(y)] \quad (77)$$

By an argument analogous to the one used in the proof of Theorem 6, one can see that $T'$ is not universally MSA if and only if $\mu_C(v_o) = t$.

This result also holds for singularised rules.

**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. All test ontologies are available online.\(^2\)

**Acylicity 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 $X^\bot$ instead of $X$ for each $X \in \{\text{MFA, MSA, JA}\}$ as a ‘lower-bound’ for acyclicity. To obtain an ‘upper bound’ for acyclicity, we checked whether the ontology was already cyclic when ignoring the rules containing equality. These steps produced a set of existential rules, which were further modified as required to encode the desired acyclicity 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

\(^2\)http://www.hermit-reasoner.com/2011/acyclicity/TestCorpus.zip
the MFA test exceeded it 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 11). 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 found to be MSA, JA, and WA.

Note that 7 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).

### 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. This suggests that the materialisation of these ontologies might not be too large.

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 made available as general vocabularies, whereas ABoxes are application-specific and are thus usually not made publicly available. Because of that, we ran 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.\(^3\) The TBox of LUBM contains 8 generating rules and has depth 1; the ABox before materialisation contains 100, 543 facts. Materialisation required only 1 second, and it produced 150, 530 new facts (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 required only 0.01 seconds, and it added 975 new facts (with 151 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.

Thus, materialisation seems practically feasible for many ontologies: for the 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

\(^3\)http://kmi.open.ac.uk/semanticweb/ontologies/owl/kmi-basic-portal-ontology.owl
Table 4: Materialisation times (in seconds) and sizes

<table>
<thead>
<tr>
<th>Depth</th>
<th>#</th>
<th>time</th>
<th>gen. size</th>
<th>mat. size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>max</td>
<td>avg</td>
<td>max</td>
<td>avg</td>
</tr>
<tr>
<td>&lt; 5</td>
<td>82</td>
<td>69</td>
<td>0.9</td>
<td>27  2</td>
</tr>
<tr>
<td>5–9</td>
<td>13</td>
<td>68</td>
<td>11</td>
<td>37  11</td>
</tr>
<tr>
<td>10–80</td>
<td>14</td>
<td>549</td>
<td>101</td>
<td>281 51</td>
</tr>
</tbody>
</table>

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 which generalise all existing acyclicity conditions that we know of.

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 becomes computationally easier; second, under acyclicity conditions it is possible to extend Horn ontologies with arbitrary SWRL rules without affecting neither decidability nor worst-case complexity; finally, acyclicity enables principled extensions of ontology materialisation-based reasoners; furthermore, since many existing ontologies turn out to be acyclic, our results open the door for practical CQ answering beyond the OWL 2 RL profile.

References


