Pushing the Boundaries of Tractable Ontology Reasoning

David Carral[1], Cristina Feier[2], Bernardo Cuenca Grau[2], Pascal Hitzler[1], and Ian Horrocks[2]

[1] Department of Computer Science, Wright State University, Dayton US
[2] Department of Computer Science, University of Oxford, Oxford UK

Abstract. We identify a class of Horn ontologies for which standard reasoning tasks such as instance checking and classification are tractable. The class is general enough to include the OWL 2 EL, QL, and RL profiles. Verifying whether a Horn ontology belongs to the class can be done in polynomial time. We show empirically that the class includes many real-world ontologies that are not included in any OWL 2 profile, and thus that polynomial time reasoning is possible for these ontologies.

1 Introduction

In recent years there has been growing interest in so-called *lightweight* ontology languages, which are based on logics with favourable computational properties. The most prominent examples of lightweight ontology languages are the EL, QL and RL profiles of OWL 2 [23]. Standard reasoning tasks, such as classification and fact entailment, are feasible in polynomial time for all profiles, and many highly scalable profile-specific reasoners have been developed [3,6,8,16,24,26,28].

All the OWL 2 profiles are *Horn* languages: any ontology in a profile can be translated into a set of first-order Horn clauses. However, many Horn OWL 2 ontologies fall outside the profiles, and when reasoning with such ontologies we are forced to resort to a fully-fledged OWL 2 reasoner if a completeness guarantee is required. Indeed, in contrast to the lightweight logics underpinning the profiles, the logics required to capture Horn OWL 2 ontologies are intractable: standard reasoning is ExpTime-complete for the description logic Horn-$\mathcal{SHOIQ}$ and 2-ExpTime-complete for the more expressive Horn-$\mathcal{SROIQ}$ [25].

Our aim is to push the tractability boundaries of lightweight ontology languages, and devise efficiently implementable reasoning algorithms that can be applied to most existing Horn ontologies. In our recent work, we took a first step towards achieving this goal by defining a new class of tractable ontologies based on a role (aka property) safety condition, the idea behind which is to preclude the interactions between language constructs that are ultimately responsible for intractability [9]. We showed that Horn-$\mathcal{SHOIQ}$ ontologies in the QL, RL and EL profiles contain only safe roles,[3] and that for ontologies containing only safe

[3] The intersection of the normative profiles and Horn-$\mathcal{SHOIQ}$ excludes certain features such as property chain axioms.

roles, standard reasoning tasks are still tractable even if the ontology is not captured by any of the profiles. However, our evaluation revealed that, although this usefully extends the range of ontologies for which tractable reasoning is known to be possible, many real-world Horn ontologies contain (a relatively small number of) unsafe roles, and for these ontologies tractability remains unclear.

In this paper we go a step farther and define a new class of Horn-$\mathcal{SHOIQ}$ ontologies in which unsafe roles are allowed to occur, but only under certain restrictions. Membership in this class can be efficiently checked by first generating a graph from the materialisation of a Datalog program, and then checking whether the generated graph is an oriented forest. We call the ontologies satisfying this condition *role safety acyclic* (RSA), and show that standard reasoning tasks remain tractable for RSA ontologies. To this end, we employ a reasoning algorithm based on a translation from a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$ into a set $\mathcal{N}_{\mathcal{O}}$ of first-order Horn rules with function symbols. We show that this transformation preserves standard reasoning outcomes and hence one can reason over $\mathcal{N}_{\mathcal{O}}$ instead of $\mathcal{O}$. Furthermore, if $\mathcal{O}$ is RSA, then the *Skolem chase* [10,22] terminates in polynomially many steps when applied to $\mathcal{N}_{\mathcal{O}}$, and yields a Herbrand model of polynomial size from which the relevant reasoning outcomes can be directly retrieved. Finally, we propose a relaxation of the acyclicity condition for which tractability of reasoning is no longer guaranteed, but that still ensures termination of the Skolem chase over $\mathcal{N}_{\mathcal{O}}$ with a Herbrand model of exponential size. We refer to ontologies satisfying this relaxed condition as weakly RSA (WRSA).

We have tested our acyclicity conditions over two large ontologies repositories. Our results show that a large proportion of out-of-profile ontologies are RSA. Our conditions can thus have immediate practical implications: on the one hand, RSA identifies a large class of ontologies for which reasoning is known to be tractable, and on the other hand, we show that reasoning for both RSA and WRSA ontologies can be implemented using existing Logic Programming engines with support for function symbols, such as DLV [21] and IRIS [5].

Finally, we note that our notion of acyclicity is related to (yet, incomparable with) existing acyclicity notions applicable to existential rules and ontologies [4,10,11,18,22]. Unlike existing notions, our main goal is to ensure tractability of reasoning rather than chase termination. Indeed, even if $\mathcal{O}$ is RSA, the Skolem chase applied to (the clausification of) $\mathcal{O}$ may not terminate.[4]

This paper comes with an extended version with all proofs of our results.[5]

2 Preliminaries

The Logic Horn-$\mathcal{SHOIQ}$ We assume basic familiarity with the logics underpinning standard ontology languages, and refer the reader to the literature for further details [1,13,14]. We next define Horn-$\mathcal{SHOIQ}$ [20,25] and specify its semantics via translation into first-order logic with built-in equality. W.l.o.g. we restrict our attention to ontologies in a normal form close to those in [19,25].

[4] We defer a detailed discussion to the Related Work section.

[5] http://www.cs.ox.ac.uk/isg/TR/RSAcheck.pdf

	Horn-$\mathcal{SHOIQ}$ axioms α	First-order sentences $\pi(\alpha)$
(R1)	$R_1 \sqsubseteq R_2$	$R_1(x,y) \rightarrow R_2(x,y)$
(R2)	$R_1 \sqsubseteq R_2^-$	$R_1(x,y) \rightarrow R_2(y,x)$
(R3)	$\mathsf{Tra}(R)$	$R(x,y) \wedge R(y,z) \rightarrow R(x,z)$
(T1)	$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B$	$A_1(x) \wedge \ldots \wedge A_n(x) \rightarrow B(x)$
(T2)	$A \sqsubseteq \{a\}$	$A(x) \rightarrow x \approx a$
(T3)	$\exists R.A \sqsubseteq B$	$R(x,y) \wedge A(y) \rightarrow B(x)$
(T4)	$A \sqsubseteq \leq 1 S.B$	$A(x) \wedge S(x,y) \wedge B(y) \wedge S(x,z) \wedge B(z) \rightarrow y \approx z$
(T5)	$A \sqsubseteq \exists R.B$	$A(x) \rightarrow \exists y.(R(x,y) \wedge B(y))$
(T6)	$\mathsf{Ran}(R) = A$	$R(x,y) \rightarrow A(y)$
(T7)	$A \sqsubseteq \exists R.\{a\}$	$A(x) \rightarrow R(x,a)$
(A1)	$A(a)$	$A(a)$
(A2)	$R(a,b)$	$R(a,b)$

Fig. 1. Horn-$\mathcal{SHOIQ}$ syntax and semantics, where $A_{(i)} \in N_\mathsf{C}$, $B \in N_\mathsf{C}$, $R_{(i)}, S \in N_\mathsf{R}$ with S simple, and $a, b \in N_\mathsf{I}$. Universal quantifiers are omitted. Axioms (T6) and (T7) are redundant, but are useful for defining (resp.) the EL and the RL profiles.

A (DL) signature Σ consists of disjoint countable sets of *concept names* N_C, *role names* N_R and *individuals* N_I, where we additionally assume that $\{\top, \bot\} \subseteq N_\mathsf{C}$. A *role* is an element of $N_\mathsf{R} \cup \{R^- | R \in N_\mathsf{R}\}$. The function $\mathsf{Inv}(\cdot)$ is defined over roles as follows, where $R \in N_\mathsf{R}$: $\mathsf{Inv}(R) = R^-$ and $\mathsf{Inv}(R^-) = R$.

An *RBox* $\mathcal{R}$ is a finite set of axioms (R1)-(R3) in Fig. 1. We denote with $\sqsubseteq_\mathcal{R}$ the minimal relation over roles in $\mathcal{R}$ s.t. $R \sqsubseteq_\mathcal{R} S$ and $\mathsf{Inv}(R) \sqsubseteq_\mathcal{R} \mathsf{Inv}(S)$ hold if $R \sqsubseteq S \in \mathcal{R}$. We define $\sqsubseteq_\mathcal{R}^*$ as the reflexive-transitive closure of $\sqsubseteq_\mathcal{R}$. A role R is *transitive* in $\mathcal{R}$ if there exists S s.t. $S \sqsubseteq_\mathcal{R}^* R$, $R \sqsubseteq_\mathcal{R}^* S$ and either $\mathsf{Tra}(S) \in \mathcal{R}$ or $\mathsf{Tra}(\mathsf{Inv}(S)) \in \mathcal{R}$. A role R is *simple* in $\mathcal{R}$ if no transitive role S exists s.t. $S \sqsubseteq_\mathcal{R}^* R$. A TBox $\mathcal{T}$ is a finite set of axioms (T1)-(T5) in Fig. 1.[6] An ABox $\mathcal{A}$ is a finite, non-empty set of assertions (A1) and (A2) in Fig. 1. An ontology $\mathcal{O} = \mathcal{R} \cup \mathcal{T} \cup \mathcal{A}$ consists of an RBox $\mathcal{R}$, TBox $\mathcal{T}$, and ABox $\mathcal{A}$. The signature of $\mathcal{O}$ is the set of concept names, role names, and individuals occurring in $\mathcal{O}$.

We define the semantics of a Horn-$\mathcal{SHOIQ}$ ontology by means of a mapping π from Horn-$\mathcal{SHOIQ}$ axioms into first-order sentences with equality as specified in Fig. 1. This mapping is extended to map ontologies to first-order knowledge bases in the obvious way. Ontology satisfiability and entailment in first-order logic with built-in equality (written $\models$) are defined as usual.

We sometimes treat $\top$ and $\bot$ as ordinary unary predicates, the meaning of which is axiomatised. For a finite signature Σ, we denote with $\mathcal{F}_\Sigma^{\top\bot}$ the smallest set with a sentence $A(x) \rightarrow \top(x)$ for each $A \in N_\mathsf{C}$ and $R(x,y) \rightarrow \top(x) \wedge \top(y)$ for each $R \in N_\mathsf{R}$. This is w.l.o.g. for Horn theories: a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$ with signature Σ is satisfiable iff $\pi(\mathcal{O}) \cup \mathcal{F}_\Sigma^{\top\bot} \not\models \exists y.\bot(y)$. Furthermore, $\mathcal{O} \models \alpha$ with $\mathcal{O}$ satisfiable and α an axiom over Σ iff $\pi(\mathcal{O}) \cup \mathcal{F}_\Sigma^{\top\bot} \models \pi(\alpha)$.

Similarly, we may treat the equality predicate $\approx$ as ordinary and denote with $\mathcal{F}_\Sigma^{\approx}$ its axiomatisation as a congruence relation over Σ, and we denote with $\models_\approx$

[6] For presentational convenience, we omit axioms $A \sqsubseteq \geq n\, R.B$. These can be simulated using axioms $A \sqsubseteq \exists R.B_i$ and $B_i \sqcap B_j \sqsubseteq \bot$ for $1 \leq i < j \leq n$.

the entailment relationship where equality is treated as an ordinary predicate. Axiomatisation of equality preserves entailment: for each set $\mathcal{F}$ of sentences with signature Σ and each sentence α over Σ, we have $\mathcal{F} \models \alpha$ iff $\mathcal{F} \cup \mathcal{F}^{\approx}_{\Sigma} \models_{\approx} \alpha$.

OWL 2 Profiles The OWL 2 specification defines three normative profiles, EL, QL, and RL, all of which are captured by Horn-$\mathcal{SROIQ}$. In this paper we restrict our attention to the intersection of these profiles with Horn-$\mathcal{SHOIQ}$ (which excludes features such as property chain axioms), as this greatly simplifies the algorithms and proofs. A Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$ is: *(i) EL* if it does not contain axioms of the form (R2) or (T4); *(ii) RL* if it does not contain axioms of the form (T5); and *(iii) QL* if it does not contain axioms of the form (R3), (T2) or (T4), each axiom (T1) satisfies $n = 1$, and each axiom (T3) satisfies $A = \top$.

Horn rules and Datalog A Horn rule is a first-order sentence of the form

$$\forall \boldsymbol{x} \forall \boldsymbol{z}.[\varphi(\boldsymbol{x}, \boldsymbol{z}) \rightarrow \psi(\boldsymbol{x})]$$

where tuples of variables $\boldsymbol{x}$, $\boldsymbol{z}$ are disjoint, $\varphi(\boldsymbol{x}, \boldsymbol{z})$ is a conjunction of function-free atoms, and $\psi(\boldsymbol{x})$ is a conjunction of atoms (possibly with function symbols). A fact is a ground, function-free atom. A Horn program $\mathcal{P}$ consists of a finite set of Horn rules and facts. A rule (program) is *Datalog* if it is function-free.[7] Forward-chaining reasoning over Horn programs can be realised by means of the *Skolem chase* [10,22]. We adopt the treatment of the Skolem chase from [10].

A set of ground atoms S' is a *consequence* of a Horn rule r on a set of ground atoms S if a substitution σ exists mapping the variables in r to the terms in S such that $\varphi\sigma \subseteq S$ and $S' \subseteq \psi\sigma$. The result of *applying* r to S, written $r(S)$, is the union of all consequences of r on S. For $\mathcal{H}$ a set of Horn rules, $\mathcal{H}(S) = \bigcup_{r \in \mathcal{H}} r(S)$. Let S be a finite set of ground atoms, let $\mathcal{H}$ be a set of rules, and let Σ be the signature of $\mathcal{H} \cup S$. Let $\mathcal{H}' = \mathcal{H} \cup \mathcal{F}^{\approx}_{\Sigma} \cup \mathcal{F}^{\top\bot}_{\Sigma}$.The *chase sequence* for S and $\mathcal{H}$ is a sequence of sets of ground atoms $S^0_{\mathcal{H}}, S^1_{\mathcal{H}}, \ldots$ where $S^0_{\mathcal{H}} = S$ and, for each $i > 0 : S^i_{\mathcal{H}} = S^{i-1}_{\mathcal{H}} \cup \mathcal{H}(S^{i-1}_{\mathcal{H}})$.

The *Skolem chase* of the program $\mathcal{P} = \mathcal{H} \cup S$ is defined as the (possibly infinite) Herbrand interpretation $I^{\infty}_{\mathcal{P}} = \bigcup_i S^i_{\mathcal{H}}$. The Skolem chase can be used to determine fact entailment: for each fact α it holds that $\mathcal{P} \models \alpha$ iff $\alpha \in I^{\infty}_{\mathcal{P}}$. The Skolem chase of $\mathcal{P}$ *terminates* if $i \geq 0$ exists such that $S^i_{\mathcal{H}} = S^j_{\mathcal{H}}$ for each $j > i$.

If $\mathcal{P}$ is a Datalog program, then $I^{\infty}_{\mathcal{P}}$ is the finite least Herbrand model of $\mathcal{P}$, which we refer to as the *materialisation* of $\mathcal{P}$. Furthermore, by slight abuse of notation, we sometimes refer to the Skolem chase of a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$ as the chase for the program obtained from $\pi(\mathcal{O})$ by standard Skolemisation of existentially quantified variables into functional terms.

3 The Notion of Role Safety

In contrast to the logics underpinning the OWL 2 profiles, the logics required to capture existing Horn ontologies are intractable. In particular, satisfiability is

[7] We adopt a more liberal definition of Datalog that allows conjunction in rule heads.

ExpTime-hard already for Horn-$\mathcal{ALCI}$ (the fragment of Horn-$\mathcal{SHOIQ}$ without nominals [15,19] or cardinality restrictions).

A closer look at existing complexity results reveals that the main source of intractability is the phenomenon typically known as *and-branching*: due to the interaction between existential quantifiers over a role R (i.e., axioms of type (T5)) and universal quantifiers over R (encoded by axioms of type (T3) and (R2)), an ontology may only be satisfied in models of exponential size. The same effect can be achieved via the interaction between existential quantifiers and cardinality restrictions (axioms of type (T4)): reasoning in the extension of the EL profile with counting is also known to be ExpTime-hard [2].

And-branching can be tamed by precluding the harmful interactions between existential quantifiers and universal quantifiers, on the one hand and existential quantifiers and cardinality restrictions, on the other hand. If we disallow existential quantifiers altogether (axioms (T5)), then we obtain the RL profile, and ontologies become equivalent to Datalog programs with equality. Similarly, if we disallow the use of inverse roles and cardinality restrictions, thus precluding both universal quantification over roles and counting, then we obtain the EL profile.

The main idea behind our notion of role safety is to identify a subset of the roles in an ontology over which these potentially harmful interactions between language constructs cannot occur. On the one hand, if a role does not occur existentially quantified in axioms of type (T5), then its "behaviour" is similar to that of a role in an RL ontology, and hence it is *safe*. On the other hand, if a role occurs existentially quantified, but no axioms involving inverse roles or counting apply to any of its super-roles, then the role behaves like a role in an EL ontology, and hence it is also *safe*.

Definition 1. *Let $\mathcal{O} = \mathcal{R} \cup \mathcal{T} \cup \mathcal{A}$ be an ontology. A role R in $\mathcal{O}$ is* safe *if either it does not occur in axioms of type $A \sqsubseteq \exists R.B$, or the following properties hold for each role S:*

1. *$R \not\sqsubseteq^*_{\mathcal{R}} S$ and $R \not\sqsubseteq^*_{\mathcal{R}} \mathsf{Inv}(S)$ if S occurs in a concept $\leq 1\, S.B$;*
2. *$R \not\sqsubseteq^*_{\mathcal{R}} \mathsf{Inv}(S)$ if S occurs in an axiom of type $\exists R.A \sqsubseteq B$ with $A \neq \top$.*

Example 1. Consider the example ontology $\mathcal{O}_{\mathsf{Ex}}$ in Figure 2, which is not captured by any of the normative profiles. The role Attends is safe: although it occurs existentially quantified in axiom (2), its inverse AttendedBy does not occur in an axiom of type (T3), and the ontology does not contain cardinality restrictions. In contrast, the role AttendedBy is unsafe since it occurs existentially quantified in (5) and its inverse role Attends occurs negatively in (3). ◇

Note that Definition 1 explains why (Horn-$\mathcal{SHOIQ}$) ontologies captured by any of the normative profiles contain only safe roles: in the case of EL, roles can be existentially quantified, but there are no inverse roles or cardinality restrictions, and hence conditions 1 and 2 in Definition 1 hold trivially; in the case of RL, roles do not occur existentially quantified in axioms of type (T5); and in the case of QL, there are no cardinality restrictions, all axioms of type (T3) satisfy $A = \top$, and hence conditions 1 and 2 also hold.

$$\mathsf{LazySt} \sqsubseteq \mathsf{Student} \tag{1}$$
$$\mathsf{Student} \sqsubseteq \exists \mathsf{Attends}.\mathsf{Course} \tag{2}$$
$$\exists \mathsf{Attends}.\mathsf{MorningCourse} \sqsubseteq \mathsf{DiligentSt} \tag{3}$$
$$\mathsf{LazySt} \sqcap \mathsf{DiligentSt} \sqsubseteq \bot \tag{4}$$
$$\mathsf{Course} \sqsubseteq \exists \mathsf{AttendedBy}.\mathsf{Student} \tag{5}$$
$$\mathsf{Attends}^{-} \sqsubseteq \mathsf{AttendedBy} \tag{6}$$
$$\mathsf{AttendedBy}^{-} \sqsubseteq \mathsf{Attends} \tag{7}$$
$$\mathsf{LazySt}(\mathsf{David}) \tag{8}$$

Fig. 2. Example ontology $\mathcal{O}_{\mathsf{Ex}}$

4 Role Safety Acyclicity

In this section, we propose a novel *role safety acyclicity* (RSA) condition that is applicable to Horn-$\mathcal{SHOIQ}$ ontologies and that does not completely preclude unsafe roles. Instead, our condition restricts the way in which unsafe roles are used so that they cannot lead to the interactions between language constructs that are at the root of ExpTime-hardness proofs; in particular, *and-branching.*

To check whether an ontology $\mathcal{O}$ is RSA we first generate a directed graph $G_{\mathcal{O}}$ by means of a Datalog program $\mathcal{P}_{\mathcal{O}}$. The edges in $G_{\mathcal{O}}$ are generated from the extension of a fresh "edge" predicate E in the materialisation of $\mathcal{P}_{\mathcal{O}}$. Intuitively, the relevant facts over E in the materialisation stem from the presence in $\mathcal{O}$ of existential restrictions over unsafe roles. Once the directed graph $G_{\mathcal{O}}$ has been generated, we check that it is a directed acyclic graph (DAG) and that it does not not contain "diamond-shaped" subgraphs; the former requirement will ensure termination of our reasoning algorithm in Section 5, while the latter is critical for tractability. Furthermore, we define a weaker version of RSA (WRSA) where $G_{\mathcal{O}}$ is only required to be a DAG. Although this relaxed notion does not ensure tractability of reasoning, it does guarantee termination of our reasoning algorithm, and hence is still of relevance in practice.

Definition 2. *Let $\mathcal{O}$ be an ontology, let Σ be the signature of $\mathcal{O}$, and let π be the mapping defined in Figure 1. Let PE and E be fresh binary predicates, and let U be a fresh unary predicate. Furthermore, for each pair of concepts A, B and each role R from Σ, let $v^A_{R,B}$ be a fresh constant. Let Ξ be the function mapping each axiom α in $\mathcal{O}$ to a datalog rule as given next, and let $\Xi(\mathcal{O}) = \{\Xi(\alpha) \mid \alpha \text{ in } \mathcal{O}\}$:*

$$\Xi(\alpha) = \begin{cases} A(x) \rightarrow R(x, v^A_{R,B}) \wedge B(v^A_{R,B}) \wedge \mathsf{PE}(x, v^A_{R,B}) & \textit{if } \alpha = A \sqsubseteq \exists R.B \\ \pi(\alpha) & \textit{Otherwise.} \end{cases}$$

Then, $\mathcal{P}_{\mathcal{O}}$ is the following datalog program:

$$\mathcal{P}_{\mathcal{O}} = \Xi(\mathcal{O}) \cup \{\mathsf{U}(x) \wedge \mathsf{PE}(x, y) \wedge \mathsf{U}(y) \rightarrow \mathsf{E}(x, y)\} \cup \{\mathsf{U}(v^A_{R,B}) \mid R \textit{ is unsafe}\}$$

$$\begin{aligned}
\mathsf{LazySt(x)} &\rightarrow \mathsf{Student(x)} \\
\mathsf{Student(x)} &\rightarrow \mathsf{Attends}(\mathsf{x}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}) \wedge \mathsf{Course}(\mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}) \wedge \mathsf{PE}(\mathsf{x}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}) \\
\mathsf{Attends(x,y)} \wedge \mathsf{MorningCourse(y)} &\rightarrow \mathsf{DiligentSt(y)} \\
\mathsf{LazySt(x)} \wedge \mathsf{DiligentSt(x)} &\rightarrow \bot\mathsf{(x)} \\
\mathsf{Course(x)} &\rightarrow \mathsf{AttendedBy}(\mathsf{x}, \mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}) \wedge \mathsf{Student}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}) \wedge \mathsf{PE}(\mathsf{x}, \mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}) \\
\mathsf{Attends(y,x)} &\rightarrow \mathsf{AttendedBy(x,y)} \\
\mathsf{AttendedBy(x,y)} &\rightarrow \mathsf{Attends(y,x)} \\
\mathsf{U(x)} \wedge \mathsf{PE(x,y)} \wedge \mathsf{U(y)} &\rightarrow \mathsf{E(x,y)} \\
\mathsf{LazySt(David)} & \\
\mathsf{U}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}) &
\end{aligned}$$

Fig. 3. Checking acyclicity of our example ontology $\mathcal{O}_{\mathsf{Ex}}$.

Let $G_{\mathcal{O}}$ be the smallest directed graph having an edge (c,d) for each fact $\mathsf{E}(c,d)$ s.t. $\mathsf{E}(c,d) \in I^{\infty}_{\mathcal{P}_{\mathcal{O}}}$. Then, $\mathcal{O}$ is Role Safety Acyclic (RSA) if $G_{\mathcal{O}}$ is an oriented forest.[8] *Finally, $\mathcal{O}$ is* weakly RSA *(WRSA) if $G_{\mathcal{O}}$ is a DAG.*

The core of the program $\mathcal{P}_{\mathcal{O}}$ is obtained from $\mathcal{O}$ by translating its axioms into first-order logic in the usual way with the single exception of existentially quantified axioms α, which are translated into Datalog by Skolemising the (unique) existential variable in $\pi(\alpha)$ into a constant. The fresh predicate PE is used to track all facts over roles R generated by the application of Skolemised rules, regardless of whether the relevant role R is safe or not. In this way, PE records "possible edges" in the graph. The safety distinction is realised by the unary predicate U, which is populated with all fresh constants introduced by the Skolemisation of existential restrictions over the unsafe roles. Finally, the rule $\mathsf{U}(x) \wedge \mathsf{PE}(x,y) \wedge \mathsf{U}(y) \rightarrow \mathsf{E}(x,y)$ ensures that only possible edges between Skolem constants in the extension of U eventually become edges in the graph.

Example 2. Figure 3 depicts the rules in the program $\mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}}$ for our example ontology $\mathcal{O}_{\mathsf{Ex}}$. The constant $\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}$ is the only fresh constant introduced by the Skolemisation of an existential restriction ($\exists\mathsf{AttendedBy.Student}$) over an unsafe role ($\mathsf{AttendedBy}$), and hence the predicate U is populated with just $\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}$.

Next consider the application of the Skolem chase on $\mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}}$, which applies to the initial facts $S = \{\mathsf{LazySt(David)}, \mathsf{U}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}})\}$ and rules $\mathcal{H} = \mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}} \setminus S$. The chase terminates after the following iterations:

$$\begin{aligned}
S^1_{\mathcal{H}} &= S \cup \{\mathsf{Student(David)}\} \\
S^2_{\mathcal{H}} &= S^1_{\mathcal{H}} \cup \{\mathsf{Attends}(\mathsf{David}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}), \mathsf{Course}(\mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}), \mathsf{PE}(\mathsf{David}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}})\} \\
S^3_{\mathcal{H}} &= S^2_{\mathcal{H}} \cup \{\mathsf{AttendedBy}(\mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}, \mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}), \mathsf{Student}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}), \mathsf{PE}(\mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}, \mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}})\} \\
S^4_{\mathcal{H}} &= S^3_{\mathcal{H}} \cup \{\mathsf{Attends}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}), \mathsf{PE}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}})\}
\end{aligned}$$

[8] An oriented forest is a disjoint union of oriented trees; that is, a DAG whose underlying undirected graph is a forest.

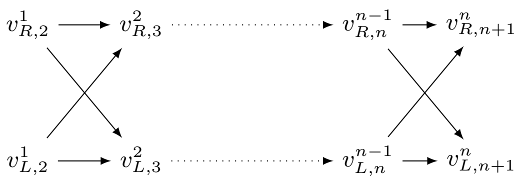

Fig. 4. An acyclic graph which is not an oriented forest

No more atoms are derived in subsequent steps and hence $I^{\infty}_{P_{\mathcal{O}_{\mathsf{Ex}}}} = S^4_{\mathcal{H}}$. Note that the graph induced by the auxiliary PE predicate is cyclic; in contrast, the extension of E is empty and $G_{\mathcal{O}_{\mathsf{Ex}}}$ has no edges. Clearly, $\mathcal{O}_{\mathsf{Ex}}$ is thus RSA. ◇

The following example illustrates the difference between RSA and WRSA.

Example 3. Consider the (family of) ontologies $\mathcal{O}_n$ consisting of the fact $A_1(a)$ and the following axioms for each $n \geq 1$ and each $1 \leq i \leq n$:

$$\begin{aligned} A_i &\sqsubseteq \exists L.A_{i+1}, & A_i &\sqsubseteq \exists R.A_{i+1} \\ \top &\sqsubseteq \leq 1L.\top, & \top &\sqsubseteq \leq 1R.\top. \end{aligned}$$

Clearly, both R and L are unsafe roles since they are defined as functional. The program $\mathcal{P}_{\mathcal{O}_n}$ then contains facts $A_1(a)$, $\mathsf{U}(v^i_{L,i+1})$, and $\mathsf{U}(v^i_{R,i+1})$ for each $1 \leq i \leq n$, as well as the following rules for each $1 \leq i \leq n$:

$$\begin{aligned} A_i(x) &\rightarrow A_{i+1}(v^i_{L,i+1}) \wedge L(x, v^i_{L,i+1}) \wedge \mathsf{PE}(x, v^i_{L,i+1}) \\ A_i(x) &\rightarrow A_{i+1}(v^i_{R,i+1}) \wedge L(x, v^i_{R,i+1}) \wedge \mathsf{PE}(x, v^i_{R,i+1}) \\ \mathsf{U}(x) \wedge \mathsf{PE}(x, y) \wedge \mathsf{U}(y) &\rightarrow \mathsf{E}(x, y) \end{aligned}$$

The chase terminates in $n+1$ steps. The graph $G_{\mathcal{O}_n}$ induced by the edge predicate E is given in Figure 4. Note that the graph is always a DAG, but it is a tree only if $n < 3$; hence all ontologies $\mathcal{O}_n$ are WRSA, but they are RSA only for $n < 3$. ◇

The following theorem establishes that checking RSA and WRSA is tractable. Intuitively, the program $\mathcal{P}_{\mathcal{O}}$ is linear in the size of $\mathcal{O}$ and each of its rules contains at most three variables regardless of $\mathcal{O}$; as a result, the materialisation (and hence also the resulting graph) is polynomially bounded.

Theorem 1. *Checking whether an ontology $\mathcal{O}$ is RSA (resp. WRSA) is feasible in polynomial time in the size of $\mathcal{O}$.*

5 Reasoning Over Acyclic Ontologies

In this section, we show that standard reasoning tasks are tractable for RSA ontologies. To this purpose, we propose a translation from a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$ into a set $\mathcal{N}_{\mathcal{O}}$ of first-order Horn rules, which may contain function

$$\begin{aligned}
\mathsf{LazySt(x)} &\rightarrow \mathsf{Student(x)}\\
\mathsf{Student(x)} &\rightarrow \mathsf{Attends(x, v^{St}_{At,Co})} \wedge \mathsf{Course(v^{St}_{At,Co})}\\
\mathsf{Attends(x,y)} \wedge \mathsf{MorningCourse(y)} &\rightarrow \mathsf{DiligentSt(y)}\\
\mathsf{LazySt(x)} \wedge \mathsf{DiligentSt(x)} &\rightarrow \bot\mathsf{(x)}\\
\mathsf{Course(x)} &\rightarrow \mathsf{AttendedBy(x, f^{Co}_{Ia,St}(x))} \wedge \mathsf{Student(f^{Co}_{Ia,St}(x))}\\
\mathsf{Attends(y,x)} &\rightarrow \mathsf{AttendedBy(x,y)}\\
\mathsf{AttendedBy(x,y)} &\rightarrow \mathsf{Attends(y,x)}\\
\mathsf{LazySt(David)} &
\end{aligned}$$

Fig. 5. Running Example: Reasoning

symbols in the head. Axioms in $\mathcal{O}$ are translated directly into first-order rules as specified in Fig. 1. As can be seen, axioms of type (T5) are translated into rules with existentially quantified variables in the head; such variables are eliminated via Skolemisation into a constant (if the corresponding role is safe) or into a function term (if the corresponding role is unsafe).

Definition 3. *Let $\mathcal{O}$ be an ontology, let Σ be the signature of $\mathcal{O}$, and let π be the mapping defined in Fig. 1. Furthermore, for each pair of concepts A, B and each safe role R from Σ, let $v^A_{R,B}$ be a fresh constant, and for each pair of concepts A, B and each unsafe role R from Σ, let $f^A_{R,B}$ be a fresh unary function symbol. Let Λ be the function mapping each axiom α in $\mathcal{O}$ to a Datalog rule as given next:*

$$\Lambda(\alpha) = \begin{cases} A(x) \rightarrow R(x, v^A_{R,B}) \wedge B(v^A_{R,B}) & \textit{if } \alpha = A \sqsubseteq \exists R.B \textit{ with } R \textit{ safe}\\ A(x) \rightarrow R(x, f^A_{R,B}(x)) \wedge B(f^A_{R,B}(x)) & \textit{if } \alpha = A \sqsubseteq \exists R.B \textit{ with } R \textit{ unsafe}\\ \pi(\alpha) & \textit{Otherwise.} \end{cases}$$

Finally, we define the Horn program $\mathcal{N}_{\mathcal{O}}$ as the set $\{\Lambda(\alpha) \mid \alpha \text{ in } \mathcal{O}\}$.

Example 4. Figure 5 depicts the rules of the Horn program $\mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}}$ for our running example $\mathcal{O}_{\mathsf{Ex}}$. Let us compare $\mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}}$ with the Datalog program $\mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}}$ in Fig. 3, which we used for acyclicity checking. In contrast to $\mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}}$, the program $\mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}}$ contains function terms involving unsafe roles; furthermore, $\mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}}$ does not include the auxiliary graph generation predicates from $\mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}}$. Next, consider the application of the Skolem chase on $\mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}}$, i.e., to the initial fact $S = \{\mathsf{LazySt(David)}\}$ and rules $\mathcal{H} = \mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}} \setminus S$. We can check that the chase terminates after four iterations and generates function terms of depth at most one. Furthermore, the only fact that is derived over the individuals from $\mathcal{O}_{\mathsf{Ex}}$ is $\mathsf{Student(David)}$. ◇

We next show that this translation preserves satisfiability, subsumption, and instance retrieval reasoning outcomes, regardless of whether the ontology $\mathcal{O}$ is acyclic or not. Thus, we can reason over $\mathcal{N}_{\mathcal{O}}$ instead of $\mathcal{O}$ without sacrificing correctness. Since $\mathcal{N}_{\mathcal{O}}$ is a strengthening of $\mathcal{O}$, due to the Skolemisation of some existential quantifiers into constants, completeness is trivial. To show soundness, we propose an embedding of the Skolem chase of $\mathcal{N}_{\mathcal{O}}$ into the chase of $\mathcal{O}$. This embedding is not a homomorphism, as it does not homomorphically preserve binary facts; however, we can show that unary facts are indeed preserved.

Theorem 2. *The following properties hold for each ontology $\mathcal{O}$, concept names A, B and constants a and b, where Σ is the signature of $\mathcal{O}$ and c is a fresh constant not in Σ:*

1. *$\mathcal{O}$ is satisfiable iff $\mathcal{N}_{\mathcal{O}}$ is satisfiable iff $I^{\infty}_{\mathcal{N}_{\mathcal{O}}}$ contains no fact over $\bot$.*
2. *$\mathcal{O} \models A(a)$ iff $\mathcal{N}_{\mathcal{O}} \models A(a)$ iff $A(a) \in I^{\infty}_{\mathcal{N}_{\mathcal{O}}}$;*
3. *$\mathcal{O} \models A \sqsubseteq B$ iff $\mathcal{N}_{\mathcal{O}} \cup \{A(c)\} \models B(c)$ iff $B(c) \in I^{\infty}_{\mathcal{N}_{\mathcal{O}} \cup \{A(c)\}}$.*

A closer inspection of the proof of the theorem (see our online technical report) reveals that preservation of binary facts can also be ensured if the relevant role satisfies certain properties. The following example illustrates the only situation for which binary facts may not be preserved.

Example 5. Consider the ontology $\mathcal{O}$ consisting of ABox assertions $A(a)$, $A(b)$, TBox axiom $A \sqsubseteq \exists R.B$ and RBox axioms $R \sqsubseteq S$, $R \sqsubseteq S^{-}$, and $\mathsf{Tra}(S)$. Clearly, R is a safe role, and the fresh individual $v^{A}_{R,B}$ is introduced by Skolemisation. We can check that $\mathcal{N}_{\mathcal{O}} \models \{S(a, v^{A}_{R,B}), S(v^{A}_{R,B}, b)\}$ and hence $\mathcal{N}_{\mathcal{O}} \models S(a, b)$ since role S is transitive. Note, however that $\mathcal{O} \not\models S(a, b)$ since $\mathcal{O}$ has a canonical tree model in which a and b are not S-related. ◇

Proposition 1. *Let $\mathcal{O}$ be an ontology with signature Σ. Furthermore, let $R \in \Sigma$ be a role name satisfying at least one of the following properties:* (i) *R is simple,* (ii) *for every axiom of type $A \sqsubseteq \exists S.B$ in $\mathcal{O}$, with S being a safe role $S \not\sqsubseteq^{*}_{\mathcal{R}} R$, or* (iii) *for every axiom of type $A \sqsubseteq \exists S.B$ in $\mathcal{O}$, with S being a safe role $S \not\sqsubseteq^{*}_{\mathcal{R}} R^{-}$. Then, $\mathcal{O} \models R(a, b)$ iff $\mathcal{N}_{\mathcal{O}} \models R(a, b)$ iff $R(a, b) \in I^{\infty}_{\mathcal{N}_{\mathcal{O}}}$.*

Example 6. Coming back to our running example, recall that the only relevant facts contained in the chase of $\mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}}$ are $\mathsf{LazySt}(\mathsf{David})$ and $\mathsf{Student}(\mathsf{David})$. Thus, we can conclude that $\mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}}$ is satisfiable and does not entail unary facts other than these ones. Furthermore, all roles in $\mathcal{O}_{\mathsf{Ex}}$ are simple and hence we can also conclude that $\mathcal{O}_{\mathsf{Ex}}$ entails no relevant binary facts. ◇

So far, we have established that we can dispense with the input ontology $\mathcal{O}$ and reason over the Horn program $\mathcal{N}_{\mathcal{O}}$ instead. The Skolem chase of $\mathcal{N}_{\mathcal{O}}$, however, may still be infinite. We next show that acyclicity of $\mathcal{O}$ provides a polynomial bound on the size of the Skolem chase of $\mathcal{N}_{\mathcal{O}}$. Intuitively, every functional term occurring in an atom of the chase of $\mathcal{N}_{\mathcal{O}}$ corresponds to a single path in $G_{\mathcal{O}}$, and the size of the graph is polynomial in $\mathcal{O}$. In an oriented forest there is at most one path between any two nodes, which bounds polynomially the number of possible functional terms. In contrast, the latter condition does not hold for DAGs, where only a bound in the length of paths can be guaranteed.

Theorem 3. *Let $\mathcal{O}$ be an RSA ontology with signature Σ. Then, the Skolem chase of $\mathcal{N}_{\mathcal{O}}$ terminates with a Herbrand model of polynomial size. Furthermore, if $\mathcal{O}$ is WRSA, then the Skolem chase of $\mathcal{N}_{\mathcal{O}}$ terminates with a Herbrand model of size at most exponential.*

Example 7. As already mentioned, the chase for $\mathcal{N}_{\mathcal{O}_{\mathsf{Ex}}}$ terminates and computes only ground atoms of functional depth at most one. Consider, however, the chase for the programs $\mathcal{N}_{\mathcal{O}_n}$ corresponding to the family of ontologies $\mathcal{O}_n$ in Example 3. Program $\mathcal{N}_{\mathcal{O}_n}$ contains the following rules for every $1 \leq i \leq n$:

$$A_i(x) \to A_{i+1}(f^i_{L,i+1}(x)) \wedge L(x, f^i_{L,i+1}(x))$$
$$A_i(x) \to A_{i+1}(f^i_{R,i+1}(x)) \wedge R(x, f^i_{R,i+1}(x))$$

When initialised with the fact $A_1(a)$, the Skolem chase will generate in each step i the following atoms:

$$A_i(f^{i+1}_{L,i}(t_i)), A_i(f^{i+1}_{R,i}(t_i)), L(t_i, f^{i+1}_{L,i}(t_i)), R(t_i, f^{i+1}_{R,i}(t_i)),$$

where $t_i \in \{g_i(\ldots(g_2(a))\ldots) \mid g_j = f^j_{L,j-1} \text{ or } g_j = f^j_{R,j-1}, 2 \leq j \leq i\}$. Note that for every i, the number of terms t_i is exponential in i. ◇

Theorems 2 and 3 suggest a reasoning algorithm for acyclic ontologies $\mathcal{O}$. First, compute the program $\mathcal{N}_{\mathcal{O}}$ as in Definition 3. Then, run the Skolem chase for $\mathcal{N}_{\mathcal{O}}$ and read out the reasoning outcomes from the computed Herbrand model. If $G_{\mathcal{O}}$ is an oriented forest (i.e., $\mathcal{O}$ is RSA) we can implement our algorithm efficiently, which yields the following result as a corollary of the previous theorems.

Theorem 4. *Satisfiability and unary fact entailment is feasible in polynomial time for the class of RSA ontologies.*

In contrast to RSA, our algorithm runs in exponential time for WRSA ontologies. We next show that, indeed, reasoning with WRSA ontologies is intractable under standard complexity-theoretic assumptions.

Theorem 5. *Unary fact entailment is* PSPACE*-hard for WRSA ontologies.*

Finally, note that our reasoning technique can be implemented by reusing existing Logic Programming engines with support for function symbols [21,5].

6 Stronger Notions of Acyclicity

Note that Theorem 4 does not make any claims about the tractability of concept subsumption for RSA ontologies. To check whether $\mathcal{O} \models A \sqsubseteq B$ we need to extend $\mathcal{N}_{\mathcal{O}}$ with an assertion $A(c)$ over a fresh individual c, run the Skolem chase, and check whether $B(c)$ is derived (see Theorem 2). However, as illustrated by the following example, RSA is not robust under addition of ABox assertions.

Example 8. Let $\mathcal{O}$ consist of a fact $B(c)$ and the following axioms:

$$A \sqsubseteq B \quad B \sqsubseteq C \quad A \sqsubseteq \exists R.A \quad \top \sqsubseteq \leq 1.R.\top$$

Ontology $\mathcal{O}$ is RSA because the rule corresponding to the "dangerous" axiom $A \sqsubseteq \exists R.A$ involving the unsafe role R does not fire during materialisation; as a result, the graph generated by $\mathcal{P}_{\mathcal{O}}$ is empty. Indeed, the chase terminates on $\mathcal{N}_{\mathcal{O}}$ and determines satisfiability as well as all the facts entailed by $\mathcal{O}$. In contrast, if we add the fact $A(c)$ to $\mathcal{N}_{\mathcal{O}}$ to determine the subsumers of A, the chase will no longer terminate because the ontology $\mathcal{O}$ extended with $A(c)$ is now cyclic. $\diamond$

To ensure tractability of subsumption and classification, we therefore propose the following stronger notion of acyclicity.

Definition 4. *Let $\mathcal{O}$ be an ontology with signature Σ. For each concept name $A \in \Sigma$, let c_A be a fresh constant and let $\mathcal{A}_{\mathsf{Cl}} = \{A(c_A) \mid A \in \Sigma\}$. We say that $\mathcal{O}$ is RSA for classification if $\mathcal{O}$ extended with $\mathcal{A}_{\mathsf{Cl}}$ is RSA.*[9]

Tractability of subsumption immediately follows from our results in Section 5.

Proposition 2. *Checking whether $\mathcal{O} \models A \sqsubseteq B$ is feasible in polynomial time for ontologies $\mathcal{O}$ that are acyclic for classification.*

Although this notion is well-suited for TBox reasoning, data-intensive applications where the ABox changes frequently require a further strengthening.

Definition 5. *An ontology $\mathcal{O}$ is* universally RSA *if $\mathcal{O} \cup \mathcal{A}'$ is RSA for every ABox $\mathcal{A}'$.*

Checking whether $\mathcal{O} = \mathcal{R} \cup \mathcal{T} \cup \mathcal{A}$ is universally RSA can be reduced to checking whether the ontology $\mathcal{O}$ extended with a special *critical* ABox $\mathcal{A}_*^{\mathcal{O}}$ is RSA, where $\mathcal{A}_*^{\mathcal{O}}$ consists of all facts that can be constructed using concept and role names from $\mathcal{O}$, all individuals occurring in $\mathcal{T}$, and a fresh individual $*$.

Proposition 3. *An ontology $\mathcal{O}$ is* universally RSA *iff $\mathcal{O} \cup \mathcal{A}_*^{\mathcal{O}}$ is RSA.*

Example 9. The critical ABox for our example ontology $\mathcal{O}_{\mathsf{Ex}}$ consists of all facts $A(*)$ and $R(*,*)$ for A a concept name and R a role name from $\mathcal{O}_{\mathsf{Ex}}$. It can be checked that $\mathcal{O}_{\mathsf{Ex}}$ is universally RSA, and hence also RSA for classification. $\diamond$

Universal RSA is, however, a rather strict condition, especially in the presence of equality. The following example illustrates that, e.g., every ontology with a functional role used in an existential restriction is not universally RSA.

Example 10. Consider $\mathcal{O}$ consisting of axioms $A \sqsubseteq \exists R.B$ and $\top \sqsubseteq \leq 1R.\top$. The critical ABox contains facts $A(*)$, $B(*)$, and $R(*,*)$. The corresponding Datalog program entails a fact $R(*, v_{R,B}^A)$ due to axiom $A \sqsubseteq \exists R.B$. Due to the functionality of R, the individuals $*$ and $v_{R,B}^A$ become equal, and hence we have $A(v_{R,B}^A)$ and eventually also $R(v_{R,B}^A, v_{R,B}^A)$. Since R is unsafe, the graph contains a cyclic edge $\mathsf{E}(v_{R,B}^A, v_{R,B}^A)$. Indeed, the chase of both $\mathcal{O}$ and $\mathcal{N}_{\mathcal{O}}$ is infinite. $\diamond$

[9] Note that ontologies that are RSA for classification are also RSA.

It is well-known that the Skolem chase often does not terminate in the presence of equality [10,22]. The standard approach to circumvent this issue is to exploit the so-called *singularisation technique* [22]. Roughly speaking, singularisation replaces equality $\approx$ in $\mathcal{O}$ with a fresh predicate Eq. The Eq predicate is axiomatised in a similar way to equality, but without the usual replacement rules (i.e., rules of the form $A(x) \wedge \mathsf{Eq}(x, y) \rightarrow A(y)$, for each concept name A, are not included in the axiomatisation); instead, the premises of rules in the ontology are modified to compensate for the lack of replacement rules. After application of the singularisation transformation, the ontology is thus equality-free. Singularisation preserves reasoning outcomes in a well-understood way, and it is effective in addressing non-termination problems.

We have exploited this technique by checking acyclicity over a singularisation $\mathcal{O}_s$ of the input ontology $\mathcal{O}$, instead of checking acyclicity over $\mathcal{O}$ itself (see our online TR for further details). If the singularised ontology $\mathcal{O}_s$ is acyclic, then our results in Section 5 ensure that the chase $I^{\infty}_{\mathcal{N}_{\mathcal{O}_s}}$ of $\mathcal{N}_{\mathcal{O}_s}$ is finite and captures reasoning outcomes over $\mathcal{O}_s$. The properties of singularisation then ensure that reasoning outcomes over the original $\mathcal{O}$ are also preserved, and they can be retrieved from $I^{\infty}_{\mathcal{N}_{\mathcal{O}_s}}$. The use of singularisation significantly increased the number of universally acyclic ontologies in our evaluation (see Section 8).

7 Related Work

In recent years the computational properties of Horn Description Logics have been extensively investigated. The logical underpinnings for the EL and QL profiles of OWL 2 are provided by, respectively, the Horn logics $\mathcal{EL}^{++}$ [2] and DL-Lite$_R$ [7], while the RL profile is based on Datalog and its intersection with DLs [12]. Hustadt et al. proposed the expressive logic Horn-$\mathcal{SHIQ}$, and establised its complexity [15]. Krötzsch et al. studied the complexity of a wide range of Horn DLs with complexities in-between the tractable logics underpinning the profiles and Horn-$\mathcal{SROIQ}$ [20,19]. Finally, the exact complexity of Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ was determined by Ortiz et al. [25].

Our techniques in Section 5 extend the so-called combined approach to reasoning in EL [17,27], where ontologies are transformed into Datalog programs by means of Skolemisation of all existentially quantified variables into constants. Skolemisation into constants was also exploited by Zhou et al. [29] to compute upper bounds to query answers.

Finally, in the literature we can find a wide range of acyclicity conditions that are sufficient to ensure chase termination. Weak acyclicity [11] was one of the first such notions, and was subsequently extended to joint acyclicity [18], acyclicity of a graph of rule dependencies [4], and super-weak acyclicity [22], amongst others. The notion of acyclicity closest to ours is model summarising acyclicity (MSA) [10], where acyclicity can also be determined by the materialisation of a Datalog program. Unlike existing acyclicity notions, ours was designed to ensure tractability of reasoning rather than chase termination. In particular, the Skolem chase of our example RSA ontology $\mathcal{O}_{\mathsf{Ex}}$ is infinite and hence $\mathcal{O}_{\mathsf{Ex}}$ cannot

Repository	Reasoning Task	Total	Safe	RSA		Cyclic		Time-out	
				no Sing.	Sing.	no Sing.	Sing.	no Sing.	Sing.
Oxford Ontology Repository	Satisfiability	126	37	37+43	37+44	46	39	0	6
	Classification	126	37	37+35	37+35	52	49	2	5
	Universality	126	37	37+2	37+31	87	57	0	1
Ontology Design Patterns	Satisfiability	23	14	14+9	14+9	0	0	0	0
	Classification	23	14	14+8	14+8	1	1	0	0
	Universality	23	14	14+4	14+8	5	1	0	0

Table 1. Acyclicity evaluation results for ontologies outside the OWL 2 profiles.

be captured by any acyclicity condition designed for chase termination. Instead, our notion ensures termination of the Skolem chase over a particular *transformed* Horn program $\mathcal{N}_\mathcal{O}$, which we can use for reasoning over $\mathcal{O}$. Another important difference is that, in contrast to the chase of $\mathcal{O}$, the chase of the transformed program $\mathcal{N}_\mathcal{O}$ is not a universal model of $\mathcal{O}$, and hence it does not preserve answers to general conjunctive queries (but only for satisfiability and fact entailment). Finally, although existing acyclicity conditions guarantee termination of the chase, none of them ensures polynomiality of the computed Herbrand model. Indeed, checking fact entailment over Horn-$\mathcal{SHI}$ ontologies that are weakly acyclic [11] (the most basic acyclicity notion for chase termination) is Pspace-hard [10].

8 Proof of Concept

We have implemented RSA and WRSA checkers using RDFox [24] as a Datalog reasoner. For testing, we used the ontologies in the Oxford Repository and the Design Patterns repository. The former is a large repository currently containing 761 real-world ontologies; the latter contains a wide range of smaller ontologies that capture design patterns commonly used in ontology modeling (these ontologies are particularly interesting as they highlight common interactions between language constructs). Experiments were performed on a laptop with 16 GB RAM and an Intel Core 2.9 GHz processor running Java v.1.7.0_21, with a timeout of 30 min. The software and data used for testing are available online.[10]

Our results are summarised in Table 1. For each repository, we first selected those ontologies that are Horn-$\mathcal{SHOIQ}$ and are not captured by any of the OWL 2 profiles. We found 126 such ontologies in the Oxford Repository and 23 in the Design Patterns repository. We then tested our acyclicity conditions for satisfiability (Def. 2), classification (Def. 4) and universality (Def. 5) on all these ontologies.[11] We performed tests both with and without singularisation. Interestingly, in both repositories we could not find any ontology that is WRSA but not RSA, and hence the two notions coincided for all our tests.

[10] https://www.dropbox.com/sh/w1kh3vuhnvindv1/AAD59BK3s5LlD7xCblIsrlSHa

[11] For classification and universality, we disregarded the ABox part of the ontologies.

As we can observe, 37 ontologies in the Oxford Repository contained only safe roles, and hence are RSA. Without singularisation, we found 43 additional ontologies with unsafe roles that are RSA, 35 of which were also RSA for classification and only 2 universally acyclic. When using singularisation the number of additional RSA ontologies increased significantly, and we obtained 29 additional universally RSA ontologies, but unfortunately our tests timed-out for several ontologies. This can be explained by the fact that the use of singularisation leads to more complicated Datalog rules for which RDFox is not optimised.

In the case of the Design Patterns repository, all ontologies are RSA. We only found one ontology that was not universally RSA when using singularisation. Ontologies in this repository are smaller, and we encountered no time-outs.

9 Conclusions and Future Work

We have proposed the new tractable class of RSA ontologies, which is based on the notion of safe roles, and a novel acyclicity condition. Our experiments suggest that a significant proportion of out-of-profile ontologies are RSA; as a result, we can exploit a worst-case optimal algorithm that runs in polynomial time to solve standard reasoning tasks over such ontologies, where only worst-case exponential algorithms were applicable before. This result thus opens the door to further optimisation of ontology reasoning.

So far, our experiments have established that many ontologies satisfy our RSA condition. Our next goal is to develop and optimise our reasoning algorithm as well as our acyclicity checker. We also plan to extend our techniques to apply to Horn-$\mathcal{SROIQ}$ and hence to all Horn OWL 2 ontologies.

Acknowledgements. Work supported by the Royal Society, the EPSRC project Score!, the NSF under award 1017255 "III: Small: TROn: Tractable Reasoning with Ontologies" and "La Caixa" Foundation.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, second edn. (2007)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI. pp. 364–369 (2005)
3. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL - a polynomial-time reasoner for life science ontologies. In: IJCAR. pp. 287–291 (2006)
4. Baget, J.F., Mugnier, M.L., Thomazo, M.: Towards farsighted dependencies for existential rules. In: RR. pp. 30–45 (2011)
5. Bishop, B., Fischer, F.: IRIS - integrated rule inference system. In: ARea (2008)
6. Bishop, B., Kiryakov, A., Ognyanoff, D., Peikov, I., Tashev, Z., Velkov, R.: OWLim: A family of scalable semantic repositories. Semantic Web J. 2(1), 33–42 (2011)
7. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. Automated Reasoning (JAR) 39(3), 385–429 (2007)

8. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F.: The MASTRO system for ontology-based data access. Semantic Web J. 2(1), 43–53 (2011)
9. Carral, D., Feier, C., Cuenca Grau, B., Hitzler, P., Horrocks, I.: $\mathcal{EL}$-ifying ontologies. In: IJCAR (2014)
10. Cuenca Grau, B., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., Wang, Z.: Acyclicity notions for existential rules and their application to query answering in ontologies. JAIR 47, 741–808 (2013)
11. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. 336(1), 89–124 (2005)
12. Grosof, B., Horrocks, I., Volz, R., Decker, S.: Description logic programs: combining logic programs with description logic. In: WWW. pp. 48–57 (2003)
13. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Doherty, P., Mylopoulos, J., Welty, C. (eds.) Proc. 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06). pp. 57–67. AAAI Press (2006)
14. Horrocks, I., Sattler, U.: A tableaux decision procedure for $\mathcal{SHOIQ}$. In: IJCAI. pp. 448–453 (2005)
15. Hustadt, U., Motik, B., Sattler, U.: Data complexity of reasoning in very expressive description logics. In: IJCAI. pp. 466–471 (2005)
16. Kazakov, Y., Krötzsch, M., Simančík, F.: The incredible ELK: From polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. J. Autom. Reas. (JAR) (2013)
17. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to ontology-based data access. In: IJCAI. pp. 2656–2661 (2011)
18. Krötzsch, M., Rudolph, S.: Extending decidable existential rules by joining acyclicity and guardedness. In: IJCAI. pp. 963–968 (2011)
19. Krötzsch, M., Rudolph, S., Hitzler, P.: Complexity boundaries for Horn description logics. In: AAAI. pp. 452–457 (2007)
20. Krötzsch, M., Rudolph, S., Hitzler, P.: Complexities of Horn description logics. ACM Trans. Comp. Log. 14(1), 2:1–2:36 (2013)
21. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. ACM Trans. Comput. Log. 7(3), 499–562 (2006)
22. Marnette, B.: Generalized schema-mappings: from termination to tractability. In: PODS. pp. 13–22 (2009)
23. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.): OWL 2 Web Ontology Language: Profiles. W3C Recommendation (27 October 2009), available at `http://www.w3.org/TR/owl2-profiles/`
24. Motik, B., Nenov, Y., Piro, R., Horrocks, I., Olteanu, D.: Parallel materialisation of Datalog programs in centralised, main-memory RDF systems. In: AAAI (2014)
25. Ortiz, M., Rudolph, S., Simkus, M.: Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In: KR (2010)
26. Rodriguez-Muro, M., Calvanese, D.: High performance query answering over DL-Lite ontologies. In: KR (2012)
27. Stefanoni, G., Motik, B., Horrocks, I.: Introducing nominals to the combined query answering approaches for $\mathcal{EL}$. In: AAAI (2013)
28. Wu, Z., Eadon, G., Das, S., Chong, E.I., Kolovski, V., Annamalai, M., Srinivasan, J.: Implementing an inference engine for RDFS/OWL constructs and user-defined rules in Oracle. In: ICDE. pp. 1239–1248 (2008)
29. Zhou, Y., Grau, B.C., Horrocks, I., Wu, Z., Banerjee, J.: Making the most of your triple store: Query answering in OWL 2 using an RL reasoner. In: WWW (2013)