Complexities of Horn Description Logics

MARKUS KRÖTZSCH
Department of Computer Science, University of Oxford, United Kingdom
SEBASTIAN RUDOLPH
Institute AIFB, Karlsruhe Institute of Technology, Germany
and
PASCAL HITZLER
Kno.e.sis Center, Wright State University, Dayton OH, USA

Description Logics (DLs) have become a prominent paradigm for representing knowledge in a variety of application areas, partly due to their ability to achieve a favourable balance between expressivity of the logic and performance of reasoning. Horn description logics are obtained, roughly speaking, by disallowing all forms of disjunctions. They have attracted attention since their (worst-case) *data* complexities are in general lower than for their non-Horn counterparts, which makes them attractive for reasoning with large sets of instance data (ABoxes). It is therefore natural to ask whether Horn DLs also provide advantages for schema (TBox) reasoning, i.e., whether they also feature lower *combined* complexities. This paper settles this question for a variety of Horn DLs. An example of a tractable Horn logic is the DL underlying the ontology language OWL RL, which we characterise as the Horn fragment of the description logic $\mathcal{SROIQ}$ without existential quantifiers. If existential quantifiers are allowed, however, many Horn DLs become intractable. We find that Horn-$\mathcal{ALC}$ already has the same worst-case complexity as $\mathcal{ALC}$, i.e., ExpTime, but we also identify various DLs for which reasoning is PSpace-complete. As a side effect, we derive simplified syntactic definitions of Horn DLs, for which we exploit suitable normal form transformations.

Categories and Subject Descriptors: I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages; F.2.2. [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problem—*Complexity of Proof Procedures*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Computational Logic*

General Terms: Languages, Theory

Additional Key Words and Phrases: description logics, Horn logic, computational complexity

1. INTRODUCTION

One of the driving motivations behind description logic (DL) research is to design languages which maximise the expressive language features that are available for

Author's address: M. Krötzsch, Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, OX1 3QD Oxford, UK, markus.kroetzsch@cs.ox.ac.uk.
This work was supported by EPSRC grant *HermiT: Reasoning with Large Ontologies*, by the Deutsche Forschungsgemeinschaft (DFG) under the ExpresST project, and by the National Science Foundation under award 1017225 *III: Small: TROn—Tractable Reasoning with Ontologies.*

knowledge modelling, while at the same time striving for the most inexpensive languages in terms of computational complexity. A particularly prominent case in point is the DL-based Web Ontology Language OWL [OWL Working Group 2009], which is a W3C recommended standard since 2004. OWL (more precisely, OWL DL) is indeed among the most expressive knowledge representation languages which are also decidable.

Of particular interest for practical investigations are *tractable* DLs, i.e., DLs which are of polynomial worst-case time complexity [Grosof et al. 2003; Baader et al. 2005; Calvanese et al. 2007; Krötzsch et al. 2008; Krötzsch 2011]. While not being Boolean closed, and thus relatively inexpressive, they receive increasing attention as they promise to provide a good trade-off between expressivity and scalability. This is also reflected by the fact that the 2009 revision of the OWL standard adopted several of them as designated important fragments of OWL [Motik et al. 2009].

Many tractable DLs also turn out to be *Horn description logics*, although this term has originally been used for Horn-$\mathcal{SHIQ}$ only [Hustadt et al. 2005]. These logical languages are based on the idea of defining Horn logic fragments of DLs. In first-order logic, *Horn clauses* are disjunctions of atomic formulae and negated atomic formulae that contain at most one non-negated atom. Many kinds of rules in logic programming, and especially Datalog rules, are Horn clauses in this sense. The relationship with DLs has first been established by the reasoner *KAON2*, which translates DL axioms to first-order logic rules, possibly with disjunctions [Motik and Sattler 2006]. Horn-$\mathcal{SHIQ}$ has been obtained as a syntactic characterisation of $\mathcal{SHIQ}$ fragment for which this transformation yields (disjunction-free) Datalog.

Hornness often leads to computational advantages. Reasoning in Datalog, e.g., is ExpTime-complete w.r.t. the size of the rule set (*combined complexity*), and P-complete w.r.t. the number of ground facts (*data complexity*). In contrast, adding disjunctions to Datalog increases combined complexity to NExpTime and data complexity to (co-)NP. Similar advantages have been observed for Horn DLs. For example, the data complexity is P for Horn-$\mathcal{SHIQ}$, but (co-)NP for $\mathcal{SHIQ}$. Moreover, all common tractable DLs disallow any "non-Horn" use of disjunctions. Horn DLs were also shown to yield practical advantages for reasoning, even for algorithms that do not rely on reductions to Datalog [Motik et al. 2009; Kazakov 2009].

In spite of these encouraging results, Horn DLs are far from being understood properly. Even a general definition is missing, since the original definition of Horn-$\mathcal{SHIQ}$ is closely related to the reasoning algorithm of KAON2, and does not cover all features used in modern DLs. Furthermore, even in the cases covered by Horn-$\mathcal{SHIQ}$, it is not known how Hornness affects the combined complexity of reasoning. Indeed, reasoning for Horn-$\mathcal{SHIQ}$ is known to be in ExpTime (like for $\mathcal{SHIQ}$), but lower bounds for the combined complexity of reasoning have not been established yet. Complexities for Horn DLs that are smaller or larger than Horn-$\mathcal{SHIQ}$ are also unknown.

This paper closes these gaps. Its main contributions are:

—In Section 3, we propose Horn-$\mathcal{SROIQ}^{\text{free}}$ as a basis for defining Horn DLs that use arbitrary features of $\mathcal{SROIQ}$. We show that our direct syntactic definition generalises the more complicated conditions used to define Horn-$\mathcal{SHIQ}$.

—In Section 4, we study $\mathcal{RL}$, the description logic underlying the ontology language

OWL RL. We characterise this logic as the fragment of Horn-$\mathcal{SROIQ}^{\text{free}}$ without existential quantifiers, and we show that reasoning is P-complete for this logic.

—In Section 5, we consider Horn DLs that allow only limited forms of existential quantification of the form $\exists R.\top$, and we show that reasoning in these logics is PSPACE-complete.

—In Section 6, we add full existential quantification to obtain Horn-$\mathcal{ALC}$, and we show that this makes reasoning EXPTIME-hard, in spite of the restrictions that Horn DLs impose on the use of disjunctions. This shows that the combined complexity of all DLs between Horn-$\mathcal{ALC}$ and Horn-$\mathcal{SHIQ}$ is EXPTIME-complete.

An overview of related work is provided in Section 7, and the results are discussed in Section 8, where Fig. 16 gives an overview of our main results.

This article is a significantly rewritten and extended compilation of [Krötzsch et al. 2006a; Krötzsch et al. 2006b; Krötzsch et al. 2007]. Based on our definition of Horn DLs, further complexity results for Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ have meanwhile been established [Ortiz et al. 2010].

2. PRELIMINARIES AND NOTATION

We generally assume that the reader is familiar with basic description logics, but in order to make the paper relatively self-contained, we introduce them briefly here. A gentle first introduction to description logics with pointers to further reading can be found in [Krötzsch et al. 2012]; a textbook introduction to DLs in the context of Semantic Web technologies is provided in [Hitzler et al. 2009].

We first define a rather general description logic, called $\mathcal{SROIQ}^{\text{free}}$, and then specialise this definition, throughout the paper, as needed for introducing other DLs. In essence, $\mathcal{SROIQ}^{\text{free}}$ is the well-known DL $\mathcal{SROIQ}$ without any structural restrictions regarding simplicity or regularity; readers who are familiar with $\mathcal{SROIQ}$ may thus want to skip Sections 2.1 and 2.2, and concentrate on the syntactic simplifications discussed in Section 2.3.

2.1 Syntax

$\mathcal{SROIQ}^{\text{free}}$ and all other DLs considered herein are based on three disjoint sets of *individual names* $\mathbf{I}$, *concept names* $\mathbf{A}$, and *role names* $\mathbf{N}$. We call such a triple $\langle \mathbf{I}, \mathbf{A}, \mathbf{N} \rangle$ a *DL signature*. Throughout this work, we assume that these basic sets are finite, and consider them to be part of the given knowledge base when speaking about the "size of a knowledge base." We further assume $\mathbf{N}$ to be the union of two disjoint sets of *simple roles* $\mathbf{N}_{\text{s}}$ and *non-simple roles* $\mathbf{N}_{\text{n}}$. Later on, the use of simple roles in conclusions of logical axioms will be restricted to ensure, intuitively speaking, that relationships of these roles are not implied by *chains* of other role relationships. The reason for this is that, in some cases, simple roles can be used in axioms where non-simple roles might lead to undecidability.

The approach we take here assumes an *a priori* declaration of simple and non-simple role names. A common alternative approach is to derive a maximal set of simple roles from the structure of a given DL knowledge base. This *a posteriori* approach of determining the sets $\mathbf{N}_{\text{n}}$ or $\mathbf{N}_{\text{s}}$ is more adequate in practical applications, where it is often not viable to declare simplicity of roles in advance. Especially if ontologies are dynamic, simplicity of roles may need to be changed over time to suit the overall structure of axioms. For the investigation of theoretical properties, however, pre-supposing complete knowledge about the names of simple and non-simple roles can simplify definitions.

Definition 2.1. Consider a DL signature $\mathscr{S} = \langle \mathbf{I}, \mathbf{A}, \mathbf{N} \rangle$ with $\mathbf{N} = \mathbf{N}_\mathrm{s} \cup \mathbf{N}_\mathrm{n}$. The set $\mathbf{R}$ of $\mathcal{SROIQ}^\mathrm{free}$ *role expressions* (or simply *roles*) for $\mathscr{S}$ is defined by the following grammar:

$$\mathbf{R} ::= U \mid \mathbf{N} \mid \mathbf{N}^-$$

where U is called the *universal role*. The set $\mathbf{R}_\mathrm{s} \subseteq \mathbf{R}$ of all *simple role expressions* is defined to contain all role expressions that contain no non-simple role names. The set $\mathbf{R}_\mathrm{n}$ of *non-simple role expressions* is $\mathbf{R}_\mathrm{n} := \mathbf{R} \setminus \mathbf{R}_\mathrm{s}$. A bijective function $\mathrm{Inv} : \mathbf{R} \to \mathbf{R}$ is defined by setting $\mathrm{Inv}(R) := R^-$, $\mathrm{Inv}(R^-) := R$, and $\mathrm{Inv}(U) := U$ for all $R \in \mathbf{N}$.

The set $\mathbf{C}$ of $\mathcal{SROIQ}^\mathrm{free}$ *concept expressions* (or simply *concepts*) for $\mathscr{S}$ is defined by the grammar

$$\mathbf{C} ::= \top \mid \bot \mid \mathbf{A} \mid \{\mathbf{I}\} \mid \exists \mathbf{R}.\mathsf{Self} \mid \neg \mathbf{C} \mid (\mathbf{C} \sqcap \mathbf{C}) \mid (\mathbf{C} \sqcup \mathbf{C}) \mid \forall \mathbf{R}.\mathbf{C} \mid \exists \mathbf{R}.\mathbf{C} \mid \geqslant n\, \mathbf{R}.\mathbf{C} \mid \leqslant n\, \mathbf{R}.\mathbf{C}$$

where n is a non-negative integer.

Concepts are used to model classes while roles represent binary relationships. In some application areas of description logics, especially in relation to the Web Ontology Language OWL, "class" is used as a synonym for "concept." Similarly, it is also common to use the term "property" as a synonym for "role" in some contexts, but we will not make use of this terminology here.

Note that, in our formulation, the universal role U is introduced as a constant (or nullary operator) on roles, and not as a "special" role name. In particular $U \in \mathbf{R}_\mathrm{s}$. Treating U as a simple role deviates from earlier works on $\mathcal{SROIQ}$, but it can be shown that U can typically be allowed in axioms that are often restricted to simple roles (see Definition 2.4) without leading to undecidability or increased worst-case complexity of reasoning [Rudolph et al. 2008b].

Parentheses are typically omitted if the exact structure of a given concept expression is clear or irrelevant. Also, we usually assume a signature and corresponding sets of concept and role expressions to be given using the notation of Definition 2.1, mentioning it explicitly only to distinguish multiple signatures if necessary. Using these conventions, role and concept expressions can be combined into axioms:

Definition 2.2. A $\mathcal{SROIQ}^\mathrm{free}$ *RBox axiom* is an expression of one of the following forms:

—$R_1 \circ \ldots \circ R_k \sqsubseteq R$ where $R_1, \ldots, R_k, R \in \mathbf{R}$ and where $R \notin \mathbf{R}_\mathrm{n}$ only if $k = 1$ and $R_1 \in \mathbf{R}_\mathrm{s}$,

—$\mathsf{Ref}(R)$ (reflexivity), $\mathsf{Tra}(R)$ (transitivity), $\mathsf{Irr}(R)$ (irreflexivity), $\mathsf{Dis}(R, R')$ (role disjointness), $\mathsf{Sym}(R)$ (symmetry), $\mathsf{Asy}(R)$ (asymmetry), where $R, R' \in \mathbf{R}$.

A $\mathcal{SROIQ}^\mathrm{free}$ *TBox axiom* is an expression of the form $C \sqsubseteq D$ or $C \equiv D$ with $C, D \in \mathbf{C}$. A $\mathcal{SROIQ}^\mathrm{free}$ *ABox axiom* is an expression of the form $C(a)$, $R(a, b)$, or $a \approx b$ where $C \in \mathbf{C}$, $R \in \mathbf{R}$, and $a, b \in \mathbf{I}$.

RBox axioms of the form $R_1 \circ \ldots \circ R_k \sqsubseteq R$ are also known as *role inclusion axioms* (RIAs), and a RIA is said to be *complex* if $k > 1$. Expressions such as $\mathsf{Ref}(R)$ are called *role characteristics*. TBox axioms are also known as *terminological axioms* or *schema axioms*, and expressions of the form $C \sqsubseteq D$ are known as *generalised concept inclusions* (GCIs). ABox axioms are also called *assertional axioms*, where axioms $C(a)$ are *concept assertions*, axioms $R(a, b)$ are *role assertions*, and axioms $a \approx b$ are *equality assertions*.

Many of the above types of axioms can be expressed in terms of other axioms, so that substantial syntactic simplifications are possible in many DLs. Relevant abbreviations are discussed in Section 2.3 below. Logical theories in description logic are called *knowledge bases*:

Definition 2.3. A $\mathcal{SROIQ}^{\text{free}}$ *RBox* (*TBox*, *ABox*) is a set of $\mathcal{SROIQ}^{\text{free}}$ RBox axioms (TBox axioms, ABox axioms). A $\mathcal{SROIQ}^{\text{free}}$ knowledge base is the union of a (possibly empty) $\mathcal{SROIQ}^{\text{free}}$ RBox, TBox, and ABox.

The above definitions still disregard some additional restrictions that are relevant for ensuring decidability of common reasoning tasks. The next definition therefore introduces $\mathcal{SROIQ}$ as a decidable sublanguage of $\mathcal{SROIQ}^{\text{free}}$.

Definition 2.4. A $\mathcal{SROIQ}$ role expression is the same as a $\mathcal{SROIQ}^{\text{free}}$ role expression. A $\mathcal{SROIQ}$ concept expression C is a $\mathcal{SROIQ}^{\text{free}}$ concept expression such that all subconcepts D of C that are of the form $\exists S.\mathsf{Self}$, $\geqslant n\, S.E$, or $\leqslant n\, S.E$ are such that $S \in \mathbf{R}_{\text{s}}$ is simple.

A $\mathcal{SROIQ}^{\text{free}}$ RBox is *regular* if there is a strict (irreflexive) total order $\prec$ on $\mathbf{R}$ such that

—for $R \notin \{S, \text{Inv}(S)\}$, we find $S \prec R$ iff $\text{Inv}(S) \prec R$, and

—every RIA is of one of the forms:

$$R \circ R \sqsubseteq R, \qquad \text{Inv}(R) \sqsubseteq R,$$
$$R_1 \circ \ldots \circ R_k \sqsubseteq R, \quad R \circ R_1 \circ \ldots \circ R_k \sqsubseteq R, \quad R_1 \circ \ldots \circ R_k \circ R \sqsubseteq R$$

such that $R, R_1, \ldots, R_k \in \mathbf{R}$, and $R_i \prec R$ for $i = 1, \ldots, k$.

A $\mathcal{SROIQ}$ RBox is a regular $\mathcal{SROIQ}^{\text{free}}$ RBox that contains role characteristics of the forms $\mathsf{Irr}(S)$, $\mathsf{Dis}(S, T)$, and $\mathsf{Asy}(S)$ only for simple role names $S, T \in \mathbf{N}_{\text{s}}$. A $\mathcal{SROIQ}$ TBox (ABox) is a $\mathcal{SROIQ}^{\text{free}}$ TBox (ABox) that contains only $\mathcal{SROIQ}$ concept expressions. A $\mathcal{SROIQ}$ knowledge base is the union of a $\mathcal{SROIQ}$ RBox, TBox, and ABox. A $\mathcal{SROIQ}$ (RBox, TBox, or ABox) axiom is an axiom that occurs within some $\mathcal{SROIQ}$ knowledge base (in the RBox, TBox, or ABox).

A variety of different DLs has been studied, most of which can be described as sublanguages of $\mathcal{SROIQ}$. Names such as $\mathcal{SROIQ}$ are typically (partly) descriptive in that they encode some of the language constructors available in the language. The most common letters used in these acronyms are listed in Fig. 1. The name $\mathcal{ALC}$ refers to the simplest DL that is closed under Boolean constructors: it allows TBoxes and ABoxes that use $\top$, $\bot$, $\neg$, $\sqcap$, $\sqcup$, $\exists$, and $\forall$. The letter $\mathcal{S}$ denotes the extension of $\mathcal{ALC}$ with transitive roles.

For example, $\mathcal{SHIQ}$ is the fragment of $\mathcal{SROIQ}$ that does not allow nominals, and that restricts to RBox axioms of the form $\mathsf{Tra}(R)$, $S \sqsubseteq R$, and $\mathsf{Sym}(R)$ (which

Symbol	Expressive Feature	Example
$\mathcal{I}$	inverse roles	R^-
$\mathcal{O}$	nominals	$\{a\}$
$\mathcal{Q}$	qualified number restrictions	$\leqslant 3\, R.C$, $\geqslant 2\, S.D$
$\mathcal{H}$	role hierarchies	$R \sqsubseteq T$
$\mathcal{R}$	role inclusion axioms	$R \circ S \sqsubseteq T$

Fig. 1. Nomenclature for important DL features

Name	Syntax	Semantics
inverse role	R^-	$\{\langle x, y\rangle \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid \langle y, x\rangle \in R^\mathcal{I}\}$
universal role	U	$\Delta^\mathcal{I} \times \Delta^\mathcal{I}$
top	$\top$	$\Delta^\mathcal{I}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
disjunction	$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
nominals	$\{a\}$	$\{a^\mathcal{I}\}$
universal restriction	$\forall R.C$	$\{x \in \Delta^\mathcal{I} \mid \langle x, y\rangle \in R^\mathcal{I} \text{ implies } y \in C^\mathcal{I}\}$
existential restriction	$\exists R.C$	$\{x \in \Delta^\mathcal{I} \mid \text{for some } y \in \Delta^\mathcal{I}, \langle x, y\rangle \in R^\mathcal{I} \text{ and } y \in C^\mathcal{I}\}$
local reflexivity	$\exists S.\mathsf{Self}$	$\{x \in \Delta^\mathcal{I} \mid \langle x, x\rangle \in S^\mathcal{I}\}$
qualified number	$\leqslant n\, S.C$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid \langle x, y\rangle \in S^\mathcal{I} \text{ and } y \in C^\mathcal{I}\} \leq n\}$
restrictions	$\geqslant n\, S.C$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid \langle x, y\rangle \in S^\mathcal{I} \text{ and } y \in C^\mathcal{I}\} \geq n\}$

Fig. 2. Semantics of role and concept expressions in $\mathcal{SROIQ}^{\text{free}}$ for an interpretation $\mathcal{I}$ with domain $\Delta^\mathcal{I}$

is just syntactic sugar for $R^- \sqsubseteq R$ and thus covered by $\mathcal{HI}$). We will introduce a number of further $\mathcal{SROIQ}$ fragments later on. Some historic names do not follow a clear naming scheme, but we still adhere to Fig. 1 when extending such DLs.

2.2 Semantics and Inferencing

The semantics of description logics is typically specified by providing a model theory, from which notions like logical consistency and entailment can be derived in the usual way. We specify these notions for the most general case of $\mathcal{SROIQ}^{\text{free}}$ but they can readily be applied to DLs contained in $\mathcal{SROIQ}^{\text{free}}$. The basis for this approach is the definition of a DL interpretation:

Definition 2.5. An *interpretation* $\mathcal{I}$ for a $\mathcal{SROIQ}^{\text{free}}$ signature $\mathscr{S} = \langle \mathbf{I}, \mathbf{A}, \mathbf{N}\rangle$ is a pair $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I}\rangle$, where $\Delta^\mathcal{I}$ is a non-empty set and $\cdot^\mathcal{I}$ is a mapping with the following properties:

—if $a \in \mathbf{I}$ then $a^\mathcal{I} \in \Delta^\mathcal{I}$,

—if $A \in \mathbf{A}$ then $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$,

—if $R \in \mathbf{N}$ then $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

The mapping $\cdot^\mathcal{I}$ is extended to arbitrary role and concept expressions as specified in Fig. 2, where $\#S$ denotes the cardinality on the set S.

The set $\Delta^\mathcal{I}$ is called the *domain* of $\mathcal{I}$. We often do not mention an interpretation's signature $\mathscr{S}$ explicitly if it is irrelevant or clear from the context. We can now define when an interpretation is a model for some DL axiom.

Axiom α	Condition for $\mathcal{I} \models \alpha$
$R_1 \circ \ldots \circ R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
$\mathsf{Tra}(R)$	if $R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
$\mathsf{Ref}(R)$	$\langle x, x\rangle \in R^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$
$\mathsf{Irr}(S)$	$\langle x, x\rangle \notin S^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$
$\mathsf{Dis}(S, T)$	if $\langle x, y\rangle \in S^{\mathcal{I}}$ then $\langle x, y\rangle \notin T^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
$\mathsf{Sym}(R)$	if $\langle x, y\rangle \in R^{\mathcal{I}}$ then $\langle y, x\rangle \in R^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
$\mathsf{Asy}(S)$	if $\langle x, y\rangle \in S^{\mathcal{I}}$ then $\langle y, x\rangle \notin S^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$R(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$
$a \approx b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$

$\circ$ in the right column denotes standard composition of binary relations: $R^{\mathcal{I}} \circ T^{\mathcal{I}} := \{\langle x, z\rangle \mid \langle x, y\rangle \in R^{\mathcal{I}}, \langle y, z\rangle \in T^{\mathcal{I}}\}$

Fig. 3. Semantics of $\mathcal{SROIQ}^{\text{free}}$ axioms for an interpretation $\mathcal{I}$ with domain $\Delta^{\mathcal{I}}$

Definition 2.6. Given an interpretation $\mathcal{I}$ and a $\mathcal{SROIQ}^{\text{free}}$ (RBox, TBox, or ABox) axiom α, we say that $\mathcal{I}$ *satisfies* (or *models*) α, written $\mathcal{I} \models \alpha$, if the respective conditions of Fig. 3 are satisfied. $\mathcal{I}$ *satisfies* (or *models*) a $\mathcal{SROIQ}^{\text{free}}$ knowledge base KB, denoted as $\mathcal{I} \models$ KB, if it satisfies all of its axioms. In these situations, we also say that $\mathcal{I}$ is a *model* of the given axiom or knowledge base.

This allows us to derive standard model-theoretic notions as follows:

Definition 2.7. Consider $\mathcal{SROIQ}^{\text{free}}$ knowledge bases KB and KB′.

—KB is *consistent* (satisfiable) if it has a model and *inconsistent* (unsatisfiable) otherwise,

—KB *entails* KB′, written KB $\models$ KB′, if all models of KB are also models of KB′.

This terminology is extended to axioms by treating them as singleton knowledge bases. A knowledge base or axiom that is entailed is also called a *logical consequence*.

When description logics are applied as an ontology modelling language, it is important to discover logical consequences. The (typically automatic) process of deriving logical consequences is called *reasoning* or *inferencing*, and a number of standard reasoning tasks play a central rôle in DLs:

—*Inconsistency checking:* Is KB inconsistent?

—*Concept unsatisfiability:* Given a concept C, is there no model $\mathcal{I} \models$ KB such that $C^{\mathcal{I}} \neq \emptyset$?

—*Concept subsumption:* Given concepts C, D, does KB $\models C \sqsubseteq D$ hold?

—*Instance checking:* Given a concept C and individual name a, does KB $\models C(a)$ hold?

Further reasoning tasks are considered as "standard" in some works. Common problems include instance retrieval (finding *all* instances of a concept) and classification (computing *all* subsumptions between concept names). We restrict our selection here to ensure that all standard reasoning tasks can be viewed as decision problems that have a common worst-case complexity for all logics studied within this paper.

PROPOSITION 2.8. *The standard reasoning tasks in $\mathcal{SROIQ}^{\text{free}}$ can be reduced to each other in linear time, and this is possible in any fragment of $\mathcal{SROIQ}^{\text{free}}$ that includes axioms of the form $A(a)$ and $A \sqcap C \sqsubseteq \bot$.*

PROOF. We find that KB is inconsistent if the concept $\top$ is unsatisfiable. C is unsatisfiable in KB if KB $\models C \sqsubseteq \bot$. Given a fresh individual name a, we obtain KB $\models C \sqsubseteq D$ if KB $\cup \{C(a)\} \models D(a)$. For a fresh concept name A, KB $\models C(a)$ if KB $\cup \{A(a), A \sqcap C \sqsubseteq \bot\}$ is inconsistent. This cyclic reduction shows that all reasoning problems can be reduced to one another. □

2.3 Simplifications and Normal Forms

Description logics have a very rich syntax that often provides many different ways of expressing equivalent statements.

Every $\mathcal{SROIQ}^{\text{free}}$ GCI $C \sqsubseteq D$ can be expressed as $\top \sqsubseteq \neg C \sqcup D$, i.e., by stating that the concept $\neg C \sqcup D$ is universally valid. In the following, we will often tacitly assume that GCIs are expressed as universally valid concepts, and we will use concept expressions C to express axioms $\top \sqsubseteq C$. Nonetheless, we still use $\sqsubseteq$ whenever this notation appears to be more natural for a given purpose. Likewise, we consider $C \equiv D$ as an abbreviation for $\{C \sqsubseteq D, D \sqsubseteq C\}$, and omit $\equiv$ as an atomic constructor for axioms.

Many DL constructs can be considered as "syntactic sugar" in the sense that they can readily be expressed in terms of other operators. Examples are found by applying basic propositional equivalences such as $A \sqcup B \equiv \neg(\neg A \sqcap \neg B)$ or $\top \equiv A \sqcup \neg A$. These simplifications are applicable when dealing with DLs that are characterised by a set of operators which can freely be combined to form concept expressions. In this paper, however, we derive more complex syntactic restrictions to arrive at DLs that are not closed under typical propositional equivalences. We thus do not exclude any operators from our considerations.

There still are some general simplifications that we can endorse in the sequel, and which often reduce the number of cases that we need to consider:

—Whenever a DL features counting quantifiers, we use $\geqslant 1\, R.C$ instead of $\exists R.C$, and $\leqslant 0\, R.\neg C$ instead of $\forall R.C$.

—We exploit commutativity and associativity of $\sqcap$, as given by the equivalences $A \sqcap B \equiv B \sqcap A$ and $A \sqcap (B \sqcap C) \equiv (A \sqcap B) \sqcap C$, to generally disregard nesting and ordering of conjuncts. For example, "a concept of the form $\exists R.A \sqcap C$ with C arbitrary" is used to refer to concept expressions $B \sqcap \exists R.A$ ($C = B$) or $B \sqcap (B' \sqcap \exists R.A)$ ($C = B \sqcap B'$). This convention introduces some non-determinism, e.g., if $B' = \exists R.A$ in the previous example, but the choice will never be essential in our arguments.

—We exploit commutativity and associativity of $\sqcup$ as in the case of $\sqcap$.

Example 2.9. The GCIs

$$A_1 \sqsubseteq \exists R_1.B_1, \tag{1}$$

$$\exists R_2.A_2 \sqsubseteq B_2, \tag{2}$$

$$A_3 \sqsubseteq \forall R_3.B_3, \tag{3}$$

$$\forall R_4.A_4 \sqsubseteq B_4 \tag{4}$$

C	$\mathsf{pNNF}(C)$
$A, \{a\}, \exists R.\mathsf{Self}, \top, \bot$	C
$D_1 \sqcap D_2$	$\mathsf{pNNF}(D_1) \sqcap \mathsf{pNNF}(D_2)$
$D_1 \sqcup D_2$	$\mathsf{pNNF}(D_1) \sqcup \mathsf{pNNF}(D_2)$
$\leqslant n\, R.D$	$\leqslant n\, R.\neg\mathsf{pNNF}(\neg D)$
$\geqslant n\, R.D$	$\geqslant n\, R.\mathsf{pNNF}(D)$
$\neg A, \neg\{a\}, \neg\exists R.\mathsf{Self}$	C
$\neg\top$	$\bot$
$\neg\bot$	$\top$
$\neg(D_1 \sqcap D_2)$	$\mathsf{pNNF}(\neg D_1) \sqcup \mathsf{pNNF}(\neg D_2)$
$\neg(D_1 \sqcup D_2)$	$\mathsf{pNNF}(\neg D_1) \sqcap \mathsf{pNNF}(\neg D_2)$
$\neg\leqslant n\, R.D$	$\mathsf{pNNF}(\geqslant(n+1)\, R.D)$
$\neg\geqslant n\, R.D$	$\begin{cases} \bot & \text{if } n = 0 \\ \mathsf{pNNF}(\leqslant(n-1)\, R.D) & \text{if } n \geq 1 \end{cases}$
$\neg\neg D$	$\mathsf{pNNF}(D)$

Fig. 4. Positive negation normal form transformations for DL concept expressions ($A \in \mathbf{A}$ a concept name, $a \in \mathbf{I}$ an individual name, $R \in \mathbf{N}$ a role name, $D_{(i)} \in \mathbf{C}$ concept expressions)

can be expressed as universally valid concepts that use cardinality restrictions rather than universal and existential quantifiers as follows:

$$\neg A_1 \sqcup \geqslant 1\, R_1.B_1, \tag{5}$$

$$\neg\geqslant 1\, R_2.A_2 \sqcup B_2, \tag{6}$$

$$\neg A_3 \sqcup \leqslant 0\, R_3.\neg B_3, \tag{7}$$

$$\neg\leqslant 0\, R_4.\neg A_4 \sqcup B_4. \tag{8}$$

We will make use of a negation normal form transformation in the sequel. While the standard negation normal form transformation (see, e.g., [Hitzler et al. 2009, Chapter 5]) normalises the uses of negation in concept expressions, it does often not contribute significantly to a simplified presentation. The reason is that concepts D in expressions $\leqslant n\, R.D$ also occur under a negative *polarity*, i.e., they behave like negated subexpressions; see also Section 3. Therefore a modified version, called *positive negation normal form*, is more effective for our purposes.

Definition 2.10. A $\mathcal{SROIQ}^{\text{free}}$ concept expression C is in *positive negation normal form* (pNNF) if

—if $\leqslant n\, R.D$ is a subconcept of C, then D has the form $\neg D'$, and

—every other occurrence of $\neg$ in C is part of a subconcept $\neg D$ where D is of the form $\neg A$ (A a concept name), $\neg\{a\}$, or $\neg\exists R.\mathsf{Self}$.

Every concept expression C can be transformed into a semantically equivalent concept expression $\mathsf{pNNF}(C)$ that is in positive negation normal form. It is easy to see that this can be achieved in linear time using the recursive definitions of Fig. 4.

Example 2.11. The positive negation normal forms of the universal concepts of

$$\mathbf{C_1} ::= \mathbf{C_0} \mid \mathbf{A} \mid \{\mathbf{I}\} \mid \exists\mathbf{R}.\mathsf{Self} \mid \leqslant 0\,\mathbf{R}.\neg\mathbf{C_1} \mid \leqslant 1\,\mathbf{R}.\neg\mathbf{C_0} \mid \geqslant n\,\mathbf{R}.\mathbf{C_1} \mid \mathbf{C_1} \sqcap \mathbf{C_1} \mid \mathbf{C_1} \sqcup \mathbf{C_0} \mid \mathbf{C_0} \sqcup \mathbf{C_1}$$

$$\mathbf{C_0} ::= \top \mid \bot \mid \neg\mathbf{A} \mid \neg\{\mathbf{I}\} \mid \neg\exists\mathbf{R}.\mathsf{Self} \mid \leqslant 0\,\mathbf{R}.\neg\mathbf{C_0} \mid \mathbf{C_0} \sqcap \mathbf{C_0} \mid \mathbf{C_0} \sqcup \mathbf{C_0}$$

Fig. 5. Horn-$\mathcal{SROIQ}^{\text{free}}$ concept expressions in positive negation normal form ($n \geq 1$)

Example 2.9 are

$$\neg A_1 \sqcup \geqslant 1\, R_1.B_1, \tag{9}$$

$$\leqslant 0\, R_2.\neg\neg A_2 \sqcup B_2, \tag{10}$$

$$\neg A_3 \sqcup \leqslant 0\, R_3.\neg B_3, \tag{11}$$

$$\geqslant 1\, R_4.\neg A_4 \sqcup B_4. \tag{12}$$

Role expressions and RBox axioms also allow for a number of simplifications. $\mathsf{Sym}(R)$ and $\mathsf{Tra}(R)$ are equivalent to $R^- \sqsubseteq R$ and $R \circ R \sqsubseteq R$, respectively. $\mathsf{Ref}(R)$ is equivalent to $\top \sqsubseteq \exists R.\mathsf{Self}$ but the latter is not admissible in $\mathcal{SROIQ}$ if R is not simple. As an alternative, $\mathsf{Ref}(R)$ can be expressed by $\{\top \sqsubseteq \exists S.\mathsf{Self}, S \sqsubseteq R\}$ where S is a fresh simple role name. Irreflexivity $\mathsf{Irr}(S)$ and asymmetry $\mathsf{Asy}(S)$ are again equivalently expressed by $\exists S.\mathsf{Self} \sqsubseteq \bot$ and $\mathsf{Dis}(S, \mathrm{Inv}(S))$, respectively. In summary, $\mathsf{Dis}(S, T)$ is the only role characteristic that is not expressible in terms of other constructs in most DLs.

Finally, a number of simplifications can be applied to ABox axioms as well. Most importantly, DLs that support nominals can typically express ABox assertions as TBox axioms by transforming axioms $C(a)$, $R(a, b)$, and $a \approx b$ into $\{a\} \sqsubseteq C$, $\{a\} \sqsubseteq \exists R.\{b\}$, and $\{a\} \sqsubseteq \{b\}$, respectively.

3. A HORN FRAGMENT OF $\mathcal{SROIQ}$

We first provide a direct definition of a Horn fragment of $\mathcal{SROIQ}^{\text{free}}$, which will be the basis for the various Horn DLs studied herein. Our definition is motivated by the DL Horn-$\mathcal{SHIQ}$ [Hustadt et al. 2005], and we will show below that it is indeed a generalisation of the original definition of this logic.

Definition 3.1. A Horn-$\mathcal{SROIQ}^{\text{free}}$ knowledge base over a DL signature $\mathscr{S}$ is a set of $\mathcal{SROIQ}^{\text{free}}$ axioms which are

—RBox axioms over $\mathscr{S}$, or

—TBox axioms $C \sqsubseteq D$ over $\mathscr{S}$ such that $\mathsf{pNNF}(\neg C \sqcup D)$ is a $\mathbf{C_1}$ concept as defined in Fig. 5, or

—ABox axioms $C(a)$, $R(a, b)$, or $a \approx b$ over $\mathscr{S}$ such that $\mathsf{pNNF}(C)$ is a $\mathbf{C_1}$ concept as defined in Fig. 5, $R \in \mathbf{R}$, and $a, b \in \mathbf{I}$.

Note that Fig. 5 exploits some syntactic simplifications as discussed in Section 2, and in particular that existential and universal restrictions are not mentioned explicitly. When convenient, we will still use this notation when considering fragments of Horn-$\mathcal{SROIQ}^{\text{free}}$ below.

Example 3.2. Of the concept expressions in Example 2.11, (9), (10), and (11) are of the form $\mathbf{C_0} \sqcup \mathbf{C_1}$ and thus in $\mathbf{C_1}$ and in Horn-$\mathcal{SROIQ}^{\text{free}}$. In contrast, (12) has the form $\mathbf{C_1} \sqcup \mathbf{C_1}$ and is not in Horn-$\mathcal{SROIQ}^{\text{free}}$. Referring back to the

$C\|_\epsilon$	$= C$	$\mathsf{pol}(C, \epsilon)$	$= 1$	
$(\neg C)\|_{1p}$	$= C\|_p$	$\mathsf{pol}(\neg C, 1p)$	$= -\mathsf{pol}(C, p)$	
$(C_1 \square C_2)\|_{ip}$	$= C_i\|_p$	$\mathsf{pol}(C_1 \square C_2, ip)$	$= \mathsf{pol}(C_i, p)$	for $\square \in \{\sqcap, \sqcup\}$, $i \in \{1, 2\}$
$\leqslant n\, R.C\|_{3p}$	$= C\|_p$	$\mathsf{pol}(\leqslant n\, R.C, 3p)$	$= -\mathsf{pol}(C, p)$	
$\geqslant n\, R.C\|_{3p}$	$= C\|_p$	$\mathsf{pol}(\geqslant n\, R.C, 3p)$	$= \mathsf{pol}(C, p)$	

Fig. 6. Positions in a concept (left) and their polarity (right)

D	$\mathsf{pl}^+(D)$	$\mathsf{pl}^-(D)$
$\bot$	0	0
$\top$	0	0
A	1	0
$\neg C$	$\mathsf{pl}^-(C)$	$\mathsf{pl}^+(C)$
$\bigsqcap C_i$	$\max_i \mathsf{sgn}(\mathsf{pl}^+(C_i))$	$\sum_i \mathsf{sgn}(\mathsf{pl}^-(C_i))$
$\bigsqcup C_i$	$\sum_i \mathsf{sgn}(\mathsf{pl}^+(C_i))$	$\max_i \mathsf{sgn}(\mathsf{pl}^-(C_i))$
$\geqslant n\, R.C$	1	$\frac{n(n-1)}{2} + n\, \mathsf{sgn}(\mathsf{pl}^-(C))$
$\leqslant n\, R.C$	$\frac{n(n+1)}{2} + (n+1)\, \mathsf{sgn}(\mathsf{pl}^-(C))$	1

Fig. 7. Definition of $\mathsf{pl}^+(D)$ and $\mathsf{pl}^-(D)$

original GCIs in Example 2.9, one could therefore say that Horn DLs restrict the use of universal but not that of existential role restrictions.

The original definition of Horn-$\mathcal{SHIQ}$ in [Hustadt et al. 2005] is rather more complex than the above characterisation, using a recursive function that counts the positive literals that would be needed when decomposing an axiom into equisatisfiable formulae in disjunctive normal form. In the remainder of this section, we show that our definition leads to the same results. We first recall the definition from [Hustadt et al. 2005], which requires us to introduce some auxiliary concepts.

Subconcepts of some description logic concept are denoted by specifying their *position*. Formally, a position p is a finite sequence of natural numbers, where ϵ denotes the empty position. Given a concept C, $C|_p$ denotes the *subconcept* of C at position p, defined recursively as in Fig. 6 (left). In this paper, we consider only positions that are defined in this figure, and the set of *all positions in a concept* C is understood accordingly. Given a concept C and a position p in C, the *polarity* $\mathsf{pol}(C, p)$ of C at position p is defined as in Fig. 6 (right). Using this notation, we can state the following definition of Horn knowledge bases.

Definition 3.3. Let pl^+ and pl^- denote mutually recursive functions that map a $\mathcal{SHIQ}$ concept D to a non-negative integer as specified in Fig. 7 where $\mathsf{sgn}(0) = 0$ and $\mathsf{sgn}(n) = 1$ for $n > 0$. We define a function pl that assigns to each $\mathcal{SHIQ}$ concept C and position p in C a non-negative integer by setting:

$$\mathsf{pl}(C, p) = \begin{cases} \mathsf{pl}^+(C|_p) & \text{if } \mathsf{pol}(D, p) = 1, \\ \mathsf{pl}^-(C|_p) & \text{if } \mathsf{pol}(D, p) = -1, \end{cases}$$

A concept C is *Horn* if $\mathsf{pl}(C, p) \leq 1$ for every position p in C, including the empty position ϵ. A $\mathcal{SHIQ}$ knowledge base KB is *Horn* if $\neg C \sqcup D$ is Horn for each GCI $C \sqsubseteq D$ of KB, and C is Horn for each assertion $C(a)$ of KB.

Example 3.4. Let $E_{(5)}$, $E_{(6)}$, $E_{(7)}$, and $E_{(8)}$ denote the concepts in Example 2.9. Then we find $\mathsf{pl}(E_{(5)}, \epsilon) = \mathsf{pl}(E_{(6)}, \epsilon) = \mathsf{pl}(E_{(7)}, \epsilon) = 1$ whereas $\mathsf{pl}(E_{(8)}, \epsilon) = 2$.

Definition 3.3 corresponds to Definition 1 in [Hustadt et al. 2005], but the latter refers to $\mathcal{ALCHIQ}$[1] instead of $\mathcal{SHIQ}$. The reason is that the elimination procedure for transitive roles that is considered in [Hustadt et al. 2005] may introduce axioms that are not Horn in the above sense. However, it turns out that transitive roles – and $\mathcal{SROIQ}$ role chains in general – can also be eliminated without endangering the Hornness of a knowledge base (see, e.g., [Kazakov 2008]). Hence we can safely extend the definition to $\mathcal{SHIQ}$.

While suitable as a criterion for *checking* Hornness of single axioms or knowledge bases, Definition 3.3 is not particularly suggestive as a description of the class of Horn knowledge bases as a whole. Indeed, it is not readily clear for which formulae pl yields values smaller or equal to 1 for all possible positions in the formula. Moreover, Definition 3.3 is still overly detailed as pl calculates the *exact* number of positive literals being introduced when transforming some (sub)formula.

To show that Definition 3.1 is a suitable generalisation of Definition 3.3, we first observe that Hornness is not affected by transformation to positive negation normal form.

Lemma 3.5. *A $\mathcal{SHIQ}$ concept C is Horn according to Definition 3.3 iff its positive negation normal form $\mathsf{pNNF}(C)$ is Horn according to Definition 3.3.*

Proof. The result is shown by establishing that the steps of the normal form transformation in Fig. 4 do not affect the value of pl^+. Claim: for every concept C, we have $\mathsf{pl}^+(C) = \mathsf{pl}^+(\mathsf{pNNF}(C))$. This is shown by induction over the structure of C. The claim clearly holds if C is a concept name, $\top$, or $\bot$.

Consider the case that $C = \neg(D_1 \sqcap D_2)$. Then $\mathsf{pl}^+(C) = \mathsf{sgn}(\mathsf{pl}^-(D_1)) + \mathsf{sgn}(\mathsf{pl}^-(D_2)) = \mathsf{sgn}(\mathsf{pl}^+(\neg D_1)) + \mathsf{sgn}(\mathsf{pl}^+(\neg D_2))$. By the induction hypothesis this equals $\mathsf{sgn}(\mathsf{pl}^+(\mathsf{pNNF}(\neg D_1))) + \mathsf{sgn}(\mathsf{pl}^+(\mathsf{pNNF}(\neg D_2))) = \mathsf{pl}^+(\mathsf{pNNF}(\neg(D_1 \sqcap D_2)))$, as required. The other cases of the induction are similar.

The same could be shown for pl^- but this part can be omitted by noting that the concepts that are transformed in the recursive definition of pNNF are always in positive positions. □

Proposition 3.6. *A $\mathcal{SHIQ}$ concept C is Horn according to Definition 3.3 iff it is Horn according to Definition 3.1.*

Proof. "$\Leftarrow$" We need to show that $\mathsf{pNNF}(D) \in \mathbf{C_1}$ ($\mathsf{pNNF}(D) \in \mathbf{C_0}$) implies $\mathsf{pl}^+(D) \leq 1$ ($\mathsf{pl}^+(D) = 0$). Focussing on pl^+ suffices since subconcepts that occur with negative polarity within a concept in positive negation normal form are either atomic or of the form $\neg D'$ with $D' \in \mathbf{C_1}$. By Lemma 3.5, it suffices to show that $D \in \mathbf{C_1}$ ($D \in \mathbf{C_0}$) implies $\mathsf{pl}^+(D) \leq 1$ ($\mathsf{pl}^+(D) = 0$). This can be established with some easy inductions over the structure of $\mathbf{C_0}$ and $\mathbf{C_1}$.

We first establish the claim for $\mathbf{C_0}$. The base cases for $\mathcal{SHIQ}$ are concepts D of the form $\top$, $\bot$, and $\neg A$ with $A \in \mathbf{A}$. In each case, we have $\mathsf{pl}^+(D) = 0$. For the induction step, assume that the claim holds for concepts D' and D''. Case $D = \leqslant 0\, R.\neg D'$: $\mathsf{pl}^+(D) = 0 + \mathsf{sgn}(\mathsf{pl}^-(\neg D')) = \mathsf{sgn}(\mathsf{pl}^+(D')) = 0$. Case $D = D' \sqcap D''$: $\mathsf{pl}^+(D) = \max(\mathsf{sgn}(\mathsf{pl}^+(D')), \mathsf{sgn}(\mathsf{pl}^+(D'')) = 0$. Case $D = D' \sqcup D''$: $\mathsf{pl}^+(D) = \mathsf{sgn}(\mathsf{pl}^+(D')) + \mathsf{sgn}(\mathsf{pl}^+(D'')) = 0$.

[1] $\mathcal{ALCHIQ}$ is $\mathcal{SHIQ}$ without transitivity declarations for roles.

The induction for $\mathbf{C_1}$ is similar. We have $\mathsf{pl}^+(D) = 1$ for $D \in \mathbf{A}$, and $\mathsf{pl}^+(D) = 0$ for $D \in \mathbf{C_0}$ by the above induction. Now consider $D' \in \mathbf{C_1}$ with $\mathsf{pl}^+(D') \leq 1$. Case $D = \leqslant 0\, R.\neg D'$: $\mathsf{pl}^+(D) = 0 + \mathsf{sgn}(\mathsf{pl}^-(\neg D')) = \mathsf{sgn}(\mathsf{pl}^+(D')) \leq 1$. Case $D = \leqslant 1\, R.\neg E$ with $E \in \mathbf{C_0}$: $\mathsf{pl}^+(D) = 1 + 2\,\mathsf{sgn}(\mathsf{pl}^-(\neg E)) = 1 + 2\,\mathsf{sgn}(\mathsf{pl}^+(E)) = 1 + 0 = 1$. Case $D = \geqslant n\, R.D'$: $\mathsf{pl}^+(D) = 1$. The cases for $\sqcap$ and $\sqcup$ are similar to the case of $\mathbf{C_0}$.

"$\Rightarrow$" By Lemma 3.5, we can again restrict our attention to concepts in positive negation normal form. We first show that, whenever D in pNNF is such that $\mathsf{pl}^+(D) = 0$, we find that $D \in \mathbf{C_0}$. The contrapositive – if $D \notin \mathbf{C_0}$ then $\mathsf{pl}^+(D) \neq 0$ – can be shown by induction over the structure of D. To this end, we first describe the sets $\bar{\mathbf{C}}_\mathbf{0}$ and $\bar{\mathbf{C}}_\mathbf{1}$ of $\mathcal{SHIQ}$ concepts that are not in $\mathbf{C_0}$ and $\mathbf{C_1}$, respectively:

$$\bar{\mathbf{C}}_\mathbf{0} ::= \mathbf{A} \mid \leqslant 0\, \mathbf{R}.\neg\bar{\mathbf{C}}_\mathbf{0} \mid \leqslant n\, \mathbf{R}.\neg\mathbf{C} \mid \geqslant n\, \mathbf{R}.\mathbf{C} \mid \bar{\mathbf{C}}_\mathbf{0} \sqcap \mathbf{C} \mid \mathbf{C} \sqcap \bar{\mathbf{C}}_\mathbf{0} \mid \bar{\mathbf{C}}_\mathbf{0} \sqcup \mathbf{C} \mid \mathbf{C} \sqcup \bar{\mathbf{C}}_\mathbf{0},$$

$$\begin{aligned}\bar{\mathbf{C}}_\mathbf{1} ::= {} & \leqslant 0\, \mathbf{R}.\neg\bar{\mathbf{C}}_\mathbf{1} \mid \leqslant 1\, \mathbf{R}.\neg\bar{\mathbf{C}}_\mathbf{0} \mid \leqslant (n+1)\, \mathbf{R}.\neg\mathbf{C} \mid \geqslant n\, R.\bar{\mathbf{C}}_\mathbf{1} \mid \bar{\mathbf{C}}_\mathbf{1} \sqcap \mathbf{C} \mid \mathbf{C} \sqcap \bar{\mathbf{C}}_\mathbf{1} \mid \\ & \bar{\mathbf{C}}_\mathbf{0} \sqcup \bar{\mathbf{C}}_\mathbf{0} \mid \bar{\mathbf{C}}_\mathbf{1} \sqcup \mathbf{C} \mid \mathbf{C} \sqcup \bar{\mathbf{C}}_\mathbf{1},\end{aligned}$$

where $n \geq 1$. We begin with the induction for $D \in \bar{\mathbf{C}}_\mathbf{0}$. The claim $\mathsf{pl}^+(D) \neq 0$ is immediate for $D \in \mathbf{A}$. Now assume $D' \in \bar{\mathbf{C}}_\mathbf{0}$ with $\mathsf{pl}^+(D') \neq 0$. Case $D = \leqslant 0\, R.\neg D'$: $\mathsf{pl}^+(D) = \mathsf{sgn}(\mathsf{pl}^-(\neg D')) = \mathsf{sgn}(\mathsf{pl}^+(D')) = 1$. Case $D = \leqslant n\, R.\neg E$: $\mathsf{pl}^+(D) = n(n+1)/2 + 2\,\mathsf{sgn}(\mathsf{pl}^-(\neg E)) \geq 1$. Case $D = \geqslant n\, R.E$: $\mathsf{pl}^+(D) = 1$. Case $D = D' \sqcap E$: $\mathsf{pl}^+(D) = \max(\mathsf{pl}^+(D'), \mathsf{pl}^+(E)) \geq 1$. Case $D = D' \sqcup E$: $\mathsf{pl}^+(D) = \mathsf{pl}^+(D') + \mathsf{pl}^+(E) \geq 1$. The cases $D = E \sqcap D'$ and $D = E \sqcup D'$ are similar.

To complete the proof, we show that, whenever D in pNNF is such that $\mathsf{pl}(p, D) \leq 1$ for all positions p of D, we find that $D \in \mathbf{C_1}$. We use induction to show that $D \in \bar{\mathbf{C}}_\mathbf{1}$ implies that $\mathsf{pl}(D, p) > 1$ for some position p of D. If D contains a subconcept $E \in \bar{\mathbf{C}}_\mathbf{1}$ at some positive position p, the claim is immediate from the induction hypothesis. For the remaining cases, we show the claim for position $p = 0$. Assume that $D', D'' \in \bar{\mathbf{C}}_\mathbf{0}$. Case $D = \leqslant 1\, R.\neg D'$: $\mathsf{pl}^+(D) = 1 + \mathsf{sgn}(\mathsf{pl}^-(\neg D')) = 1 + \mathsf{sgn}(\mathsf{pl}^+(D')) = 2$ since $\mathsf{pl}^+(D') > 0$ has been shown above. Case $D = \leqslant (n+1)\, R.\neg F$: $\mathsf{pl}^+(D) = (n+1)(n+2)/2 + (n+2)\,\mathsf{sgn}(\mathsf{pl}^-(\neg F)) \geq 2$. Case $D = D' \sqcup D''$: $\mathsf{pl}^+(D) = \mathsf{pl}^+(D') + \mathsf{pl}^+(D'') \geq 2$. This finishes the proof. □

The previous result shows that Definition 3.1 is indeed a generalisation of the original definition of Horn-$\mathcal{SHIQ}$. The extension with nominals and Self expressions may appear natural, but it remains to be shown that it actually leads to appropriate results. We will not study Horn-$\mathcal{SROIQ}^{\text{free}}$ as such in the sequel, but we will rather consider various fragments of this logic.

4. THE TRACTABLE HORN DESCRIPTION LOGIC $\mathcal{RL}$

In this section, we study the fragment of Horn-$\mathcal{SROIQ}^{\text{free}}$ that is obtained by disallowing existential quantification (and $\geqslant n$ restrictions in general) in the positive negation normal form used to define Horn DLs. We call this description logic $\mathcal{RL}$ due to its close relation to the OWL RL profile of the Web Ontology Language [Motik et al. 2009]. It turns out that reasoning in $\mathcal{RL}$ is possible in polynomial time, which is in strong contrast to the ExpTime worst-case complexity that we establish for slightly more expressive Horn DLs later on.

Disallowing existentials as such does not usually lead to such a reduction of reasoning complexity in DLs, even if disjunctions and negations are excluded as well. Indeed, the standard reasoning tasks for the description logic $\mathcal{FL}_0$, that only allows the constructors $\top$, $\bot$, $\sqcap$, and $\forall$, are already ExpTime-complete [Baader et al. 2007]. Our below results thus show that Hornness can significantly reduce reasoning complexity.

Definition 4.1. A concept C in pNNF is an $\mathcal{RL}$ *concept* if

—C is in $\mathbf{C_1}$ of Fig. 5, and

—C contains only concept constructors $\top$, $\bot$, $\sqcap$, $\sqcup$, $\neg$, $\leqslant n$, $\exists R.\mathsf{Self}$, and $\{a\}$.

In other words, $\mathcal{RL}$ concepts are pNNF concepts of Horn-$\mathcal{SROIQ}^{\text{free}}$ that do not contain $\geqslant n$. The description logic $\mathcal{RL}$ supports the following axioms:

—$\mathcal{SROIQ}^{\text{free}}$ RBox axioms,

—TBox axioms $C \sqsubseteq D$ such that $\mathsf{pNNF}(\neg C \sqcup D)$ is an $\mathcal{RL}$ concept,

—ABox axioms $C(a)$, $R(a, b)$, and $a \approx b$ such that $\mathsf{pNNF}(C)$ is an $\mathcal{RL}$ concept and R is a $\mathcal{SROIQ}^{\text{free}}$ role.

Example 4.2. Consider again the GCIs in Example 2.9 and their positive negation normal forms given in Example 2.11. As noted in Example 3.2, (1), (2), and (3) are in Horn-$\mathcal{SROIQ}^{\text{free}}$. However, the positive negation normal forms of both (1) and (4) contain $\geqslant 1$ constructors and are thus not in $\mathcal{RL}$. The other two GCIs (2) and (3) are in $\mathcal{RL}$.

In order to show that reasoning for $\mathcal{RL}$ is possible in polynomial time, it is useful to transform axioms into a simpler normal form:

Lemma 4.3. *Every $\mathcal{RL}$ knowledge base* KB *can be transformed into an equisatisfiable knowledge base* KB$'$ *that only contains axioms of the following forms:*

—RBox axioms of $\mathcal{SROIQ}^{\text{free}}$ where all role inclusions are of the form $R \sqsubseteq S$ or $R \circ R' \sqsubseteq S$ for $R^{(\prime)}, S \in \mathbf{R}$,

—TBox axioms of one of the following forms:

$\top \sqsubseteq C$ *with* C *of form* A_1 *or* $\neg A_1$,

$A \sqsubseteq C$ *with* C *of form* $A_1, A_1 \sqcup \neg A_2, \{a\}, \exists R.\mathsf{Self}, \leqslant 0\, R.\neg A_1,$ *or* $\leqslant 1\, R.\neg\neg A_1$,

$B \sqsubseteq C$ *with* C *of form* $\bot, \neg A_1, \neg A_1 \sqcup \neg A_2, \neg\{a\}, \neg\exists R.\mathsf{Self},$ *or* $\leqslant 0\, R.\neg\neg A_1$,

where A, A_1, A_2 are atomic concepts, and B is an atomic concept or a negated atomic concept,

—ABox axioms $R(a, b)$, $A(a)$, or $a \approx b$ with $R \in \mathbf{R}$ and $A \in \mathbf{A}$.

Moreover, the size of KB$'$ *is polynomial in the size of* KB.

Proof. Let KB be the given $\mathcal{RL}$ knowledge base. Without loss of generality, we assume that ABox axioms in KB are already of the required form. Indeed, any axiom $C(a)$ where C is a complex concept can be replaced by axioms $X(a)$ and $X \sqsubseteq C$ for a fresh concept name X. Provided that $\mathsf{pNNF}(C)$ is an $\mathcal{RL}$ concept as required, $\mathsf{pNNF}(\neg X \sqcup C) = \neg X \sqcup \mathsf{pNNF}(C)$ is also an $\mathcal{RL}$ concept.

We now construct a set KB$'$ of axioms in the above forms, such that KB$'$ and KB are equisatisfiable. Initially, let KB$'$ contain the following axioms:

(1)	$B \sqsubseteq \top$	$\mapsto$	$\{\}$
(2)	$B \sqsubseteq C \sqcup \hat{C}_0$	$\mapsto$	$\{B \sqsubseteq C \sqcup \neg X, \neg X \sqsubseteq \hat{C}_0\}$
(3)	$B \sqsubseteq C \sqcup \hat{C}_1$	$\mapsto$	$\{B \sqsubseteq C \sqcup X, X \sqsubseteq \hat{C}_1\}$
(4)	$B \sqsubseteq C \sqcap C'$	$\mapsto$	$\{B \sqsubseteq C, B \sqsubseteq C'\}$
(5)	$B \sqsubseteq \leqslant 0\, R.\neg\hat{C}_1$	$\mapsto$	$\{B \sqsubseteq \leqslant 0\, R.\neg X, X \sqsubseteq \hat{C}_1\}$
(6)	$B \sqsubseteq \leqslant 1\, R.\neg\hat{C}_0$	$\mapsto$	$\{B \sqsubseteq \leqslant 1\, R.\neg\neg X, \neg X \sqsubseteq \hat{C}_0\}$
(7)	$B \sqsubseteq \leqslant 0\, R.\neg\hat{C}_0$	$\mapsto$	$\{B \sqsubseteq \leqslant 0\, R.\neg\neg X, \neg X \sqsubseteq \hat{C}_0\}$
(8)	$R_1 \circ \ldots \circ R_k \sqsubseteq S$	$\mapsto$	$\{R_1 \circ R_2 \sqsubseteq T_3, T_3 \circ R_3 \sqsubseteq T_4, \ldots, T_k \circ R_k \sqsubseteq S\}$

$C, C', \hat{C} \in \mathbf{C}$; $\hat{C}_0 \in \mathbf{C_0}$; $\hat{C}_1 \in \mathbf{C_1} \setminus \mathbf{C_0}$; with $\hat{C}, \hat{C}_0, \hat{C}_1$ not of the form A or $\neg A$ for $A \in \mathbf{A}$;
B of the form A or $\neg A$ for $A \in \mathbf{A}$; X a fresh concept name; $R_{(i)}, S \in \mathbf{R}$; T_i fresh role names

Fig. 8. Normal form transformation for $\mathcal{RL}$ concepts

—all ABox axioms and RBox axioms of KB,

—for every TBox axiom $C \sqsubseteq D \in$ KB, the two axioms

$$X \sqsubseteq \mathsf{pNNF}(\neg C \sqcup D) \text{ and } \top \sqsubseteq X$$

where $X \in \mathbf{A}$ is a fresh concept name.

It is clear that this initial set KB$'$ is equisatisfiable to KB. The purpose of introducing auxiliary concepts X in TBox axioms is to simplify the normalisation by reducing the number of distinct cases. TBox and RBox axioms in KB$'$ are now normalised by exhaustively applying the transformation rules in Fig. 8, that replace one axiom with a set of new axioms. The correctness of this transformation follows by observing that the following remain true throughout the transformation:

(1) KB$'$ and KB are equisatisfiable.

(2) Every GCI $C \sqsubseteq D \in$ KB$'$ is such that C is of the form A or $\neg A$ for some $A \in \mathbf{A}$ and D is an $\mathcal{RL}$ concept. Moreover, $C = \neg A$ only if $D \in \mathbf{C_0}$.

Both properties hold initially and are preserved by every rule application, so the claims follow by induction. Termination in a linear number of steps is immediate since each transformation rule decomposes a subconcept of the initial set KB$'$, and no such concept is ever duplicated.

Finally, it is easy to see that a GCI to which none of the rules is applicable, is in the required normal form. Complex concepts can only occur on the right-hand side of GCIs in KB$'$, and only for constructors $\sqcup$, $\sqcap$, and $\leqslant n$. Rules (6) to (8) are exhaustive for $\leqslant n$ since only $\mathcal{RL}$ concepts can occur. For rules (2) and (3) we have exploited commutativity of $\sqcup$ to reduce cases as discussed in Section 2.3. □

THEOREM 4.4. *The standard reasoning problems for $\mathcal{RL}$ are* P*-complete.*

PROOF. Hardness is immediate from the fact that checking entailment in propositional Horn logic is hard for P [Dantsin et al. 2001]. A propositional Horn logic clause $q_1 \wedge \ldots \wedge q_k \to p$ can be expressed in $\mathcal{RL}$ using the GCI $C_{q_1} \wedge \ldots \wedge C_{q_k} \to C_p$ where C_{q_i} and C_p are concept names, and the left-hand side is considered to be $\top$ if $k = 0$. With this encoding, the original propositional theory entails a proposition q if and only if its DL encoding entails $\top \sqsubseteq C_q$.

To show membership, we apply Lemma 4.3 to obtain an equisatisfiable $\mathcal{RL}$ knowledge base of polynomial size that only contains axioms in normal form. Every such

$\mathcal{RL}$ knowledge base in normal form can be translated to a semantically equivalent Datalog program. Here, *Datalog* refers to function-free and $\exists$-free Horn logic under first-order logic semantics (we have no need of considering non-monotonic Datalog semantics). We allow rules with empty head (interpreted as false) or with empty body (interpreted as true). ABox axioms $R(a, b)$ and $A(a)$ can be interpreted as Datalog facts and do not require transformation. Statements $a \approx b$ can be treated like ground facts if $\approx$ is used like a normal predicate, the special properties of which are axiomatised using a standard equality theory:

$$
\begin{aligned}
&\to x \approx x \\
x \approx y &\to y \approx x \\
x \approx y \wedge y \approx z &\to x \approx z \\
A(x) \wedge x \approx y &\to A(y) && \text{for all } A \in \mathbf{A} \\
R(x, y) \wedge x \approx z &\to R(z, y) && \text{for all } R \in \mathbf{N} \\
R(x, y) \wedge y \approx z &\to R(x, z) && \text{for all } R \in \mathbf{N}
\end{aligned}
$$

As usual, we omit universal quantifiers when writing Datalog formulae.

GCIs in normal form are also easy to translate. For example, $A \sqsubseteq A_1 \sqcup \neg A_2$ is expressed by the rule $A(x) \wedge A_2(x) \to A_1(x)$, and $\neg A \sqsubseteq \neg\exists R.\mathsf{Self}$ is expressed by $R(x, x) \to A(x)$. Axioms of the form $A \sqsubseteq \leqslant 1\, R.\neg\neg A'$ are expressed using $\approx$:

$$A(x) \wedge R(x, y_1) \wedge A'(y_1) \wedge R(x, y_2) \wedge A'(y_2) \to y_1 \approx y_2$$

Similarly, $\approx$ is used to model nominals, e.g., the GCI $A \sqsubseteq \{a\}$ can be expressed by $A(x) \to x \approx a$. All remaining (TBox and RBox) axioms are straightforward to express in Datalog along these lines. In addition, we need to include all axioms

$$R(x, y) \to R^-(y, x) \quad \text{and} \quad R^-(x, y) \to R(y, x) \quad \text{for all } R \in \mathbf{N}$$

to capture the semantics of inverse roles. Overall, it is easy to see that all required Datalog rules are obtained as (simple syntactic transformations of) the standard first-order translations of DL axioms (see, e.g., [Hitzler et al. 2009]). Soundness and completeness of the transformation follow from this observation.

The number of auxiliary axioms for equality and inverse roles are linear in the size of the knowledge base (which is always an upper bound for the size of the signature), hence the constructed Datalog program is linear in size. All of the above types of rules have at most three variables, hence P-completeness of satisfiability checking follows from the respective result for Datalog programs with a bound on the number of variables [Dantsin et al. 2001]. The proof is completed by noting that the reduction of standard reasoning problems to satisfiability checking is possible in $\mathcal{RL}$ according to Proposition 2.8. □

Thus, most of the OWL RL ontology language can be captured in our framework of Horn DLs, with three limitations:

(1) We did not consider datatypes [Motik et al. 2009]. Adding datatypes to DLs is no major difficulty but requires extended preliminary discussions that are beyond the scope of this work.

(2) OWL RL provides a special constructor "`hasValue`" for concept expressions $\exists R.\{a\}$, and allows them on the left-hand side of GCIs. This special case could be allowed above but was omitted for simplicity.

(3) OWL RL supports so-called *keys*, a form of Datalog rules that can imply the equality of the elements that are denoted by constant symbols.

The limitations (1) and (2) would largely be overcome when considering additional concept constructors. This is what is done in the definition of the OWL 2 ontology language. Regarding (3), keys are no DL axioms, and thus do not fit into the framework of Horn DLs either, but our above Datalog translation would make it easy to incorporate them as additional (Horn) Datalog rules. One can therefore say that OWL RL is, in essence, the *Horn fragment of OWL 2 without existential quantifiers*.

Description logics that can faithfully be expressed in Datalog have been called *Description Logic Programs* (DLP) [Grosof et al. 2003]. It is not hard to further extend DLP-like fragments with additional features, provided that they can be encoded by appropriate Datalog rules. For example, role conjunctions and concept products as discussed in [Rudolph et al. 2008a; 2008b] could easily be integrated into this setting as well. Moreover, restrictions regarding regularity of RBoxes or simplicity of roles are not necessary in DLP.

While it is easy to define and extend DLP-like logics, the property of being expressible in Datalog as such is not a suitable principle for defining description logics. Indeed, even under additional restrictions, one can find much larger DLs that have this property, but that require rather unwieldy syntactic definitions [Krötzsch et al. 2010]. Hornness, in contrast, appears to be a more natural way of defining DLP-like logics.

Another insight that we can take from the above is that existential quantification is, in a sense, the main reason for the computational complexity of Horn DLs. Indeed, the following sections will confirm that even limited uses of existential quantifiers lead to higher worst-case complexities of reasoning.

5. PSPACE-COMPLETE HORN DLS: FROM HORN-$\mathcal{FL}^-$ TO HORN-$\mathcal{FLOH}^-$

The description logic $\mathcal{FL}^-$ is the fragment of $\mathcal{ALC}$ that allows $\top$, $\bot$, $\sqcap$, $\forall$, and unqualified $\exists$, i.e., concept expressions of the form $\exists R.\top$ [Baader et al. 2007]. In this section, we study a corresponding fragment of Horn-$\mathcal{SROIQ}^{\text{free}}$, which we call Horn-$\mathcal{FL}^-$. It turns out that reasoning in this DL is PSPACE-complete, and that this remains true even when further extending the DL with nominals and role hierarchies.

Some care is needed when imposing the syntactic restrictions of $\mathcal{FL}^-$ on Horn DLs. The latter are defined with respect to the positive negation normal form of universal concepts, which may not be expressible in $\mathcal{FL}^-$.

Example 5.1. The GCI $A \sqcap B \sqsubseteq C$ is in $\mathcal{FL}^-$ but the corresponding universally valid concept expression $\neg(A \sqcap B) \sqcup C$ and its pNNF $\neg A \sqcup \neg B \sqcup C$ are not.

Disjunction could be included to overcome this issue – the Hornness conditions restrict its expressive power as done in $\mathcal{RL}$ in Section 4 – but then concepts such as $\forall R.\neg A \sqcup B$ would be expressible, whereas the corresponding GCI $\exists R.A \sqsubseteq B$

$$\mathbf{F} \ ::= \top \mid \bot \mid \mathbf{A} \mid \{\mathbf{I}\} \mid \forall \mathbf{R}.\mathbf{F} \mid \exists \mathbf{R}.\top \mid \ \mathbf{F} \sqcap \mathbf{F}$$
$$\mathbf{F_0} ::= \top \mid \bot \mid \mathbf{A} \mid \{\mathbf{I}\} \mid \qquad \exists \mathbf{R}.\top \mid \mathbf{F_0} \sqcap \mathbf{F_0}$$

Fig. 9. Grammars for specifying the syntax of Horn-$\mathcal{FLOH}^-$ axioms

cannot be expressed in $\mathcal{FL}^-$. Indeed, including axioms of this form would increase the complexity of reasoning to ExpTime (see Theorem 6.9). Therefore, we provide a direct syntactic definition for Horn-$\mathcal{FL}^-$:

Definition 5.2. $\mathcal{FL}^-$ is the fragment of $\mathcal{SROIQ}^{\text{free}}$ that supports ABox and TBox axioms using the concept constructors $\top$, $\bot$, $\sqcap$, $\forall$, and $\exists R.\top$. $\mathcal{FLOH}^-$ is the extension of $\mathcal{FL}^-$ with nominals $\{a\}$ ($a \in \mathbf{I}$) and role hierarchies. Since there are no inverse roles we have $\mathbf{R} = \mathbf{N}$.

The description logic Horn-$\mathcal{FLOH}^-$ allows for the following axioms, where $\mathbf{F}$ and $\mathbf{F_0}$ are defined as in Fig. 9:

—role inclusions $R \sqsubseteq S$ with $R, S \in \mathbf{R}$,
—concept inclusions $C \sqsubseteq D$ such that the concepts $C \in \mathbf{F_0}$ and $D \in \mathbf{F}$,
—concept assertions $C(a)$ such that $C \in \mathbf{F}$,
—role assertions $R(a, b)$ with $R \in \mathbf{R}$,
—equality assertions $a \approx b$.

Horn-$\mathcal{FL}^-$ is the fragment of Horn-$\mathcal{FLOH}^-$ that does not contain nominals or role inclusions.

Lemma 5.3. *Horn-$\mathcal{FLOH}^-$ is a fragment of Horn-$\mathcal{SROIQ}^{free}$.*

Proof. We must check whether $\mathsf{pNNF}(\neg C \sqcup D) \in \mathbf{C_1}$ for every Horn-$\mathcal{FLOH}^-$ GCI $C \sqsubseteq D$. Indeed, we find that $\mathsf{pNNF}(\neg C) \in \mathbf{C_0}$ and $D \in \mathbf{C_1}$. The latter can be checked easily by comparing the grammars. To see $\mathsf{pNNF}(\neg C) \in \mathbf{C_0}$, note that the positive negation normal form of negated $\mathbf{F_0}$ concepts is given by the following grammar:

$$\bar{\mathbf{F}}_\mathbf{0} ::= \bot \mid \top \mid \neg \mathbf{A} \mid \neg\{\mathbf{I}\} \mid \forall \mathbf{R}.\bot \mid \bar{\mathbf{F}}_\mathbf{0} \sqcup \bar{\mathbf{F}}_\mathbf{0}.$$

This is obtained by computing the positive negation normal form of each part of the grammar $\mathbf{F_0}$, where we use $\mathsf{pNNF}(\neg \mathbf{F_0}) := \mathbf{F_0}$ for the recursive case. Again, it is easy to see that this is a sublanguage of $\mathbf{C_0}$ as required.

For concept assertion, the result follows again from $\mathbf{F} \subseteq \mathbf{C_1}$. □

Note that, in spite of the lack of general forms of existential restrictions, it is possible to indirectly express arbitrary positive existentials in Horn-$\mathcal{FLOH}^-$.

Example 5.4. The GCI $A \sqsubseteq \exists R.B$ can be expressed by the following Horn-$\mathcal{FLOH}^-$ axioms using a fresh role name R':

$$A \sqsubseteq \exists R'.\top \sqcap \forall R'.B, \tag{13}$$
$$R' \sqsubseteq R. \tag{14}$$

In the following sections, we show that all logics between Horn-$\mathcal{FL}^-$ and Horn-$\mathcal{FLOH}^-$ are PSpace-complete. Adding further $\mathcal{SROIQ}$ features to Horn-$\mathcal{FLOH}^-$ typically leads to ExpTime-hard logics (see Theorem 6.9).

5.1 Hardness

We directly show that Horn-$\mathcal{FL}^-$ is PSPACE-hard by reducing the halting problem for polynomially space-bounded Turing machines to checking unsatisfiability in Horn-$\mathcal{FL}^-$.

Definition 5.5. A *deterministic Turing machine* (TM) $\mathcal{M}$ is defined as a tuple $(Q, \Sigma, \Delta, q_0, Q_A)$ where

—Q is a finite set of states,
—Σ is a finite *alphabet* that includes a *blank* symbol $\Box$,
—$\Delta \subseteq (Q \times \Sigma) \times (Q \times \Sigma \times \{l, r\})$ is a *transition relation* that is deterministic, i.e., $(q, \sigma, q_1, \sigma_1, d_1)$, $(q, \sigma, q_2, \sigma_2, d_2) \in \Delta$ implies $q_1 = q_2$, $\sigma_1 = \sigma_2$, and $d_1 = d_2$.
—$q_0 \in Q$ is the *initial state*, and
—$Q_A \subseteq Q$ is a set of *accepting states*.

A *configuration* of $\mathcal{M}$ is a word $\alpha \in \Sigma^* Q \Sigma^*$. A configuration α' is a *successor* of a configuration α if one of the following holds:

(1) $\alpha = w_l q \sigma \sigma_r w_r$, $\alpha' = w_l \sigma' q' \sigma_r w_r$, and $(q, \sigma, q', \sigma', r) \in \Delta$,
(2) $\alpha = w_l q \sigma$, $\alpha' = w_l \sigma' q' \Box$, and $(q, \sigma, q', \sigma', r) \in \Delta$,
(3) $\alpha = w_l \sigma_l q \sigma w_r$, $\alpha' = w_l q' \sigma_l \sigma' w_r$, and $(q, \sigma, q', \sigma', l) \in \Delta$,

where $q \in Q$ and $\sigma, \sigma', \sigma_l, \sigma_r \in \Sigma$ as well as $w_l, w_r \in \Sigma^*$. Given some natural number s, the possible *transitions in space* s are defined by additionally requiring that $|\alpha'| \leq s + 1$.

The set of *accepting configurations* is the least set which satisfies the following conditions. A configuration α is accepting iff

—$\alpha = w_l q w_r$ and $q \in Q_A$, or
—at least one of the successor configurations of α is accepting.

$\mathcal{M}$ *accepts* a given word $w \in \Sigma^*$ (in space s) iff the configuration $q_0 w$ is accepting (when restricting to transitions in space s).

The complexity class PSPACE is defined as follows.

Definition 5.6. A language L is accepted by a polynomially space-bounded TM iff there is a polynomial p such that, for every word $w \in \Sigma^*$, $w \in L$ iff w is accepted in space $p(|w|)$.

In this section, we exclusively deal with polynomially space-bounded TMs, and so we omit additions such as "in space s" when clear from the context.

In the following, we consider a fixed TM $\mathcal{M}$ denoted as in Definition 5.5, and a polynomial p that defines a bound for the required space. For any word $w \in \Sigma^*$, we construct a Horn-$\mathcal{FL}^-$ knowledge base $\mathrm{KB}_{\mathcal{M},w}$ and show that w is accepted by $\mathcal{M}$ iff $\mathrm{KB}_{\mathcal{M},w}$ is unsatisfiable. Intuitively speaking, the elements of an interpretation domain of $\mathrm{KB}_{\mathcal{M},w}$ represent possible configurations of $\mathcal{M}$, encoded by the following concept names:

—A_q for $q \in Q$: the TM is in state q

(1) Left and right transition rules:

$A_q \sqcap H_i \sqcap C_{\sigma,i} \sqsubseteq \exists S.\top \sqcap \forall S.(A_{q'} \sqcap H_{i+1} \sqcap C_{\sigma',i})$ with $\delta = (q,\sigma,q',\sigma',r)$, $i < p(|w|)-1$

$A_q \sqcap H_i \sqcap C_{\sigma,i} \sqsubseteq \exists S.\top \sqcap \forall S.(A_{q'} \sqcap H_{i-1} \sqcap C_{\sigma',i})$ with $\delta = (q,\sigma,q',\sigma',l)$, $i > 0$

(2) Memory:

$H_j \sqcap C_{\sigma,i} \sqsubseteq \forall S.C_{\sigma,i}$ $\quad i \neq j$

(3) Failure:

$A_q \sqsubseteq \bot$ $\quad q \in Q_A$

The axioms are instantiated for all $q, q' \in Q$, $\sigma, \sigma' \in \Sigma$, $i, j \in \{0, \ldots, p(|w|) - 1\}$, and $\delta \in \Delta$.

Fig. 10. Knowledge base $\mathrm{KB}_{\mathcal{M},w}$ simulating a polynomially space-bounded TM

—H_i for $i = 0, \ldots, p(|w|) - 1$: the TM is at position i on the storage tape

—$C_{\sigma,i}$ with $\sigma \in \Sigma$ and $i = 0, \ldots, p(|w|) - 1$: position i on the storage tape contains symbol σ

Based on these concepts, elements in each interpretation of a knowledge base encode certain states of the Turing machine. A role S will be used to encode the successor relationship between states. The initial configuration for word w is described by the concept expression I_w:

$$I_w := A_{q_0} \sqcap H_0 \sqcap C_{\sigma_0,0} \sqcap \ldots \sqcap C_{\sigma_{|w|-1},|w|-1} \sqcap C_{\Box,|w|} \sqcap \ldots \sqcap C_{\Box,p(|w|)-1},$$

where σ_i denotes the symbol at the ith position of w.

It is not hard to describe runs of the TM with Horn-$\mathcal{FL}^-$ axioms, but formulating the acceptance condition is slightly more difficult. The reason is that in absence of statements like $\exists S.A$ and $\forall S.A$ in the condition part of Horn-axioms, one cannot propagate acceptance from the final accepting configuration back to initial configuration. The solution is to provoke an inconsistency as soon as an accepting configuration is reached. Thus we arrive at the knowledge base $\mathrm{KB}_{\mathcal{M},w}$ given in Fig. 10. The following is obvious from the characterisation given in Definition 3.1.

LEMMA 5.7. $\mathrm{KB}_{\mathcal{M},w}$ *is in Horn-*$\mathcal{FL}^-$.

Next we need to investigate the relationship between elements of an interpretation that satisfies $\mathrm{KB}_{\mathcal{M},w}$ and configurations of $\mathcal{M}$. Given an interpretation $\mathcal{I}$ of $\mathrm{KB}_{\mathcal{M},w}$, we say that an element e of the domain of $\mathcal{I}$ *represents* a configuration $\sigma_1 \ldots \sigma_{i-1} q \sigma_i \ldots \sigma_m$ if $e \in A_q^{\mathcal{I}}$, $e \in H_i^{\mathcal{I}}$, and, for every $j \in \{0, \ldots, p(|w|) - 1\}$, $e \in C_{\sigma,j}^{\mathcal{I}}$ whenever

$$j \leq m \text{ and } \sigma = \sigma_j \qquad \text{or} \qquad j > m \text{ and } \sigma = \Box.$$

Note that we do not require uniqueness of the above, so that a single element might in fact represent more than one configuration. As we will see below, this does not affect our results. If e represents a configuration as above, we will also say that e has state q, position i, symbol σ_j at position j etc.

LEMMA 5.8. *Consider some interpretation* $\mathcal{I}$ *that satisfies* $\mathrm{KB}_{\mathcal{M},w}$. *If some element* e *of* $\mathcal{I}$ *represents a configuration* α *and some transition* δ *is applicable to* α, *then* e *has an* $S^{\mathcal{I}}$*-successor that represents the (unique) result of applying* δ *to* α.

Proof. Consider an element e, state α, and transition δ as in the claim. Then one of the axioms (1) applies, and e must also have an $S^{\mathcal{I}}$-successor. This successor represents the correct state, position, and symbol at position i of e, again by the axioms (1). By axiom (2), symbols at all other positions are also represented by all $S^{\mathcal{I}}$-successors of e. □

Lemma 5.9. *A word w is accepted by $\mathcal{M}$ iff $\{I_w(c)\} \cup \mathrm{KB}_{\mathcal{M},w}$ is unsatisfiable, where c some constant symbol.*

Proof. Let $\mathcal{I}$ be a model of $\{I_w(c)\} \cup \mathrm{KB}_{\mathcal{M},w}$. $\mathcal{I}$ being a model for $I_w(c)$, $c^{\mathcal{I}}$ clearly represents the initial configuration of $\mathcal{M}$ with input w. By Lemma 5.8, for any further configuration reached by $\mathcal{M}$ during computation, $c^{\mathcal{I}}$ has a (not necessarily direct) $S^{\mathcal{I}}$ successor representing that configuration.

But then $\mathcal{I}$ satisfies axiom (3) only if none of the configurations that are reached have an accepting state. Since $\mathcal{I}$ was arbitrary, $\{I_w(c)\} \cup \mathrm{KB}_{\mathcal{M},w}$ can only have a satisfying interpretation if $\mathcal{M}$ does not reach an accepting state.

It remains to show the converse: if $\mathcal{M}$ does not accept w, there is some interpretation $\mathcal{I}$ satisfying $\{I_w(c)\} \cup \mathrm{KB}_{\mathcal{M},w}$. We define a canonical interpretation M as follows. The domain of M is the set of all configurations of $\mathcal{M}$ that have size $p(|w|) + 1$ (i.e., that encode a tape of length $p(|w|)$, possibly with trailing blanks). The interpretations for the concepts A_q, H_i, and $C_{\sigma,i}$ are defined as expected so that every configuration is representing itself but no other configuration. Especially, I_w^M is the singleton set that contains only the initial configuration. Given two configurations α and α', and a transition δ, we define $(\alpha, \alpha') \in S^M$ iff there is a transition δ from α to α'.

It is easy to see that M satisfies the axioms (1) and (2) of Fig. 10. Axiom (3) is satisfied since, by our initial assumption, none of the configurations reached by $\mathcal{M}$ is in an accepting state. □

Theorem 5.10. *The standard reasoning problems for Horn-$\mathcal{FL}^-$ are hard for* PSpace.

Proof. By Lemma 5.9, the word problem for polynomially space-bounded TMs can be reduced to checking satisfiability of $\{I_w(c)\} \cup \mathrm{KB}_{\mathcal{M},w}$. The other standard reasoning problems can be reduced to satisfiability checking by Proposition 2.8. By Lemma 5.7, $\mathrm{KB}_{\mathcal{M},w}$ is in Horn-$\mathcal{FL}^-$, and the same is clear for $I_w(c)$. The reduction is polynomially bounded due to the restricted number of axioms: there are at most $p(|w|) \times |\Delta|$ axioms of type (1), $p(|w|)^2 \times |\Sigma|$ axioms of type (2), and $|Q|$ axioms of type (3). □

5.2 Membership

To show that inferencing for Horn-$\mathcal{FLOH}^-$ is in PSpace, we develop a tableau algorithm for deciding the satisfiability of a Horn-$\mathcal{FLOH}^-$ knowledge base. To this end, we first present a normal form transformation that allows us to restrict our attention to simple forms of axioms. We then present the tableau construction and show its correctness, and demonstrate that it can be executed in polynomial space.

Definition 5.11. A $\mathcal{FLOH}^-$ concept expression C is *basic* if it is of the form $A \in \mathbf{A}$, $\{a\}$, or $\exists R.\top$. The set of all basic concepts is denoted by $\mathbf{B}$, assuming that

$\hat{C} \sqsubseteq \hat{D} \mapsto \{\hat{C} \sqsubseteq X, X \sqsubseteq \hat{D}\}$	$\hat{C} \sqcap A \sqsubseteq B \mapsto \{\hat{C} \sqsubseteq X, X \sqcap A \sqsubseteq B\}$
$\bot \sqsubseteq C \mapsto \emptyset$	$A \sqsubseteq C \sqcap D \mapsto \{A \sqsubseteq C, A \sqsubseteq D\}$
$C \sqsubseteq \top \mapsto \emptyset$	$A \sqsubseteq \forall R.\hat{C} \mapsto \{A \sqsubseteq \forall R.X, X \sqsubseteq \hat{C}\}$

A, B basic concept expressions, $\top$, or $\bot$; X a fresh concept name;
C, D concept expressions; $\hat{C}$, $\hat{D}$ concept expressions that are not basic

Fig. 11. Normal form transformation for Horn-$\mathcal{FLOH}^-$

the underlying signature is irrelevant or clear from the context. A Horn-$\mathcal{FLOH}^-$ knowledge base is in *normal form* if it contains only axioms of the following forms

$$A \sqsubseteq C \qquad \top \sqsubseteq C \qquad A \sqsubseteq \forall R.C$$
$$A \sqcap B \sqsubseteq C \qquad A \sqsubseteq \bot \qquad R \sqsubseteq S$$

where $A, B, C \in \mathbf{B}$ basic concepts (including existential restrictions), R, S role names, and c, d individual names.

LEMMA 5.12. *Every Horn-$\mathcal{FLOH}^-$ knowledge base* KB *is equisatisfiable to a Horn-$\mathcal{FLOH}^-$ knowledge base that contains only axioms in the normal form of Definition 5.11, and that can be computed in linear time w.r.t. the size of* KB.

PROOF. ABox axioms $C(a)$ can be expressed as GCIs $\{a\} \sqsubseteq C$, yielding a GCI of form $\mathbf{F_0} \sqsubseteq \mathbf{F}$ as in Fig. 9. ABox axioms $R(a, b)$ can be expressed as $\{a\} \sqsubseteq \exists R.\{b\}$, which can in turn be expressed as in Example 5.4 using $\{a\} \sqsubseteq \exists R'.\top \sqcap \forall R'.\{b\}$ and $R' \sqsubseteq R$ for a new role R'. Assertions $a \approx b$ can be written as $\{a\} \sqsubseteq \{b\}$.

To express arbitrary GCIs, we exhaustively apply the transformation rules in Fig. 11, where each rule application consists in replacing the axiom on the left-hand side with the axioms on the right-hand side. It is easy to see that the resulting axioms are equisatisfiable to the original axioms for each rule, so the result follows by induction. It is also easy to see that only a linear number of steps are required, where it must be noted that the rule for $A \sqsubseteq C \sqcap D$ is only applicable if A is not a compound term, so that duplicating A leads to a constant increase in size only. □

Next, we are going to present a procedure for checking satisfiability of Horn-$\mathcal{FLOH}^-$ knowledge bases. In the following we assume without loss of generality that the DL signature under consideration has at least one individual name.

Definition 5.13. Consider a Horn-$\mathcal{FLOH}^-$ knowledge base KB in normal form, with $\mathbf{B}$ the set of basic concepts, $\mathbf{R}$ the set of roles, and $\mathbf{I}$ the set of individual names. A set of relevant concept expressions is defined by setting

$$\mathsf{cl}(\mathrm{KB}) := \mathbf{B} \cup \big\{\forall R.C \mid R \in \mathbf{R}, C \in \mathbf{B}\big\} \cup \big\{\top, \bot\big\}.$$

Given a possibly infinite set I of individual names, the set $\mathcal{T}_I$ of ABox expressions over $\mathsf{cl}(\mathrm{KB})$ and I is defined as follows:

$$\mathcal{T}_I := \big\{C(e) \mid C \in \mathsf{cl}(\mathrm{KB}),\, e \in I\big\} \cup \big\{R(e, f) \mid R \in \mathbf{R},\, e, f \in I\big\}.$$

A *tableau* for KB is given by a (possibly infinite) set I of individual names and a set $T \subseteq \mathcal{T}_I$, such that $\mathbf{I} \subseteq I$ and the conditions in Fig. 12 hold. A *tableau* is said to contain a *clash* if it contains a statement of the form $\bot(e)$.

(C1) If $e \in I$, then $\top(e) \in T$.
(C2) If $e \in \mathbf{I}$, then $\{e\}(e) \in T$.
(C3) If $A \sqsubseteq C \in$ KB and $A(e) \in T$, then $C(e) \in T$.
(C4) If $A \sqcap B \sqsubseteq C \in$ KB, $A(e) \in T$, and $B(e) \in T$, then $C(e) \in T$.
(C5) If $R \sqsubseteq S \in$ KB and $R(e,f) \in T$, then $S(e,f) \in T$.
(C6) If $\{f\}(e) \in T$, then
—$C(e) \in T$ iff $C(f) \in T$,
—$R(e,g) \in T$ iff $R(f,g) \in T$, and
—$R(g,e) \in T$ iff $R(g,f) \in T$,
for all $C \in \mathsf{cl}(\text{KB})$, $R \in \mathbf{R}$, and $g \in I$.
(C7) $\exists R.\top(e) \in T$ iff $R(e,f) \in T$ for some $f \in I$.
(C8) If $\forall R.C(e) \in T$, then $C(f) \in T$ for all $f \in I$ with $R(e,f) \in T$.

Fig. 12. Conditions for a tableau $\langle I, T\rangle$ for KB, with individuals I and ABox expressions $T \subseteq \mathcal{T}_I$

Example 5.14. Consider the following knowledge base KB in normal form:

(1)	$\{c\} \sqsubseteq A$	(2)	$A \sqsubseteq \exists R.\top$	(3)	$A \sqsubseteq \forall R.B$
(4)	$B \sqsubseteq \exists S.\top$	(5)	$B \sqsubseteq \forall S.C$	(6)	$B \sqsubseteq \forall S.D$
(7)	$C \sqcap D \sqsubseteq E$	(8)	$E \sqsubseteq \bot$		

Then a tableau $\langle I, T\rangle$ for KB with sets of individuals $I = \{c, d, e\}$ is given by:

$$T = \{\top(c), \top(d), \top(e), \{c\}(c), A(c), \exists R.\top(c), \forall R.B(c), R(c,d), \\ B(d), \exists S.\top(d), \forall S.C(d), \forall S.D(d), S(d,e), C(e), D(e), E(e), \bot(e)\}.$$

Another example tableau could be obtained, e.g., by replacing all occurrences of d in T by c. Both tableaux contain a clash, and indeed there is no clash-free tableau for KB.

PROPOSITION 5.15. *A Horn-$\mathcal{FLOH}^-$ knowledge base* KB *is satisfiable iff it has a clash-free tableau.*

PROOF. Assume that KB has a clash-free tableau $\langle I, T\rangle$. An interpretation $\mathcal{I}$ is defined as follows. Due to condition (C6) in Fig. 12, we can define an equivalence relation $\sim$ on I by setting $e \sim f$ iff there is some $g \in \mathbf{I}$ with $\{\{g\}(e), \{g\}(f)\} \subseteq T$. The domain $I_\sim$ of $\mathcal{I}$ is the set of equivalence classes of $\sim$. The interpretation function is defined by setting $e^\mathcal{I} = [e]_\sim$, $e^\mathcal{I} \in A^\mathcal{I}$ iff $A(e) \in T$, and $(e^\mathcal{I}, f^\mathcal{I}) \in R^\mathcal{I}$ iff $R(e,f) \in T$, for all elements $e, f \in I$, concept names A, and role names R. It is easy to see that $\mathcal{I}$ satisfies KB.

For the converse, assume that KB is satisfiable, and that it thus has some model $\mathcal{I}$. We define a tableau $\langle I, T\rangle$ where I is the domain of $\mathcal{I}$. Further, we set $C(e) \in T$ iff $e \in C^\mathcal{I}$, and $R(e,f) \in T$ iff $(e,f) \in R^\mathcal{I}$, where $C \in \mathsf{cl}(\text{KB})$, and R some role name. Again, it is easy to see that this meets the conditions of Definition 5.13. □

As is evident by the Turing machine construction in the previous section, some Horn-$\mathcal{FLOH}^-$ knowledge bases may require models to contain an exponential number of individuals in a single relational path. Indeed, a polynomially space-bounded TM might require exponentially many steps in every accepting run, e.g., it could use its tape to store bounded-length binary number, increment the number by 1 in each step, and accept the input when an overflow occurs. The TM encoding of

(T1) $T := T \cup \{\top(e)\}$
(T2) if $e \in \mathbf{I}$ is an individual name, $T := T \cup \{\{e\}(e)\}$
(T3) for each $A \sqsubseteq C \in$ KB, if $A(e) \in T$ then $T := T \cup \{C(e)\}$
(T4) for each $A \sqcap B \sqsubseteq C \in$ KB, if $A(e) \in T$ and $B(e) \in T$ then $T := T \cup \{C(e)\}$
(T5) for each $R \sqsubseteq S \in$ KB, do the following:
- (T5a) for each $f \in I$, if $R(e, f) \in T$ and $R(e, f)$ is not inactive, then $T := T \cup \{S(e, f)\}$,
- (T5b) if $\exists R.\top(e) \in T$ then $T := T \cup \{\exists S.\top(e)\}$

(T6) for each $\{f\}(e) \in T$
- (T6a) for each $C(f) \in T$, $T := T \cup \{C(e)\}$,
- (T6b) for each $g \in I$ and each $R(f, g) \in T$, $T := T \cup \{R(e, g)\}$; $R(e, g)$ is marked inactive,
- (T6c) for each $g \in I$ and each $R(g, f) \in T$, $T := T \cup \{R(g, e)\}$; $R(g, e)$ is marked inactive,
- (T6d) for each $C(e) \in T$, $T := T \cup \{C(f)\}$,
- (T6e) for each $g \in I$ and each $R(e, g) \in T$, $T := T \cup \{R(f, g)\}$; $R(f, g)$ is marked inactive,
- (T6f) for each $g \in I$ and each $R(g, e) \in T$, $T := T \cup \{R(g, f)\}$; $R(g, f)$ is marked inactive

(T7) for each $f \in I$ and $R(e, f) \in T$ with $R(e, f)$ not inactive, $T := T \cup \{\exists R.\top(e)\}$
(T8) for each $\forall R.C(e) \in T$ and each $f \in I$ with $R(e, f) \in T$, if $R(e, f)$ is not inactive, then $T := T \cup \{C(f)\}$

(T∃) for each $\exists R.\top(e) \in T$, if $R(e, f) \notin T$ for all $f \in I$ then $I := I \cup \{g\}$ and $T := T \cup \{R(e, g)\}$, where g is a fresh individual

Fig. 13. Rules for constructing canonical tableaux for Horn-$\mathcal{FLOH}^-$ knowledge bases KB

Fig. 10 would then lead to models that represent this exponential run in an exponentially long path of successor elements. Detecting clashes in polynomial space thus requires special care. In particular, a standard tableau procedure with blocking does not execute in polynomial space. Therefore, we first provide a procedural description of a *canonical tableau* which will form the basis for our below decision algorithm.

Definition 5.16. An algorithm that computes a tableau-like structure $\langle I, T\rangle$, where every role statement in T is marked *active* or *inactive*, is defined as follows. Initially, we set $I := \mathbf{I}$ and $T := \emptyset$. The algorithm executes the following steps:

(1) Iterate over all individuals $e \in I$. To each such e, apply rules (T1) to (T8) of Fig. 13. All inferences that are not explicitly marked inactive are active.
(2) If T was changed in the previous step, go to (1).
(3) Apply rule (T∃) of Fig. 13 to all existing elements $e \in I$, where all inferences are active.
(4) If T was changed in the previous step, go to (1).
(5) Halt.

Observe that the numbered rules in Fig. 13 correspond to the conditions in Fig. 12, where the "if" direction of condition (C7) is captured by (T∃). Most of the rules should thus be intuitive to understand.

The rules (T6) are used to ensure that individuals e satisfying a nominal class are synchronised with the respective named individual $f \in \mathbf{I}$. The six sub-rules are needed since one generally cannot add $\{e\}(f)$ to T as e might not be an element of $\mathbf{I}$. However, role statements that are inferred in this way need not be taken into account as premises in other deduction rules, since they are guaranteed to have an active original. Whatever could be inferred using copied role statements and rules (T5a), (T7), or (T8), can as well be inferred via the active original from which the inactive role was initially created. Note that this argument involves an induction over the number of applications of rule (T6).

Rule (T5) is also special. In principle, one could omit (T5b), and use rules (T5a) and (T7) instead. This inference, however, is the only case where a role-successor of some individual e might contribute to the classes inferred for e. By providing rule (T5b), the class expressions containing e can be computed without considering any role successor, and rule (T7) is essential only when role expressions have been inferred from ABox statements. In combination with the delayed application of rule (T∃), this ensures that concepts are indeed inferred by (T5b) rather than by (T5a)+(T7), which will be exploited in the proof of Lemma 5.24 below.

Also note that the algorithm of Definition 5.16 is not a decision procedure, since we do not require the algorithm to halt. What we are interested in, however, is the (possibly infinite) tableau that the algorithm constructs in the limit. The existence of this limit is evident from the fact that all completion rules are finitary, and that each rule monotonically increases the size of the computed structure. For a given knowledge base KB, we write $\langle \bar{I}_{\mathrm{KB}}, \bar{T}_{\mathrm{KB}} \rangle$ to denote the *canonical tableau* constructed by the above algorithm from KB, where the subscripts are omitted when understood.

Example 5.17. The tableau $\langle I, T \rangle$ in Example 5.14 is a canonical tableau for KB, which is therefore finite in this case.

PROPOSITION 5.18. *The canonical tableau $\langle \bar{I}_{\mathrm{KB}}, \bar{T}_{\mathrm{KB}} \rangle$ is a tableau in the sense of Definition 5.13. Moreover,* KB *has a clash-free tableau iff $\langle \bar{I}_{\mathrm{KB}}, \bar{T}_{\mathrm{KB}} \rangle$ is clash-free.*

PROOF. The first part of the claim is easy to verify based on the correspondence between the conditions of Fig. 12 and the rules of Fig. 13. This also shows the "if" direction of the second part of the claim.

For the "only if" direction, we show the contrapositive: if $\langle \bar{I}_{\mathrm{KB}}, \bar{T}_{\mathrm{KB}} \rangle$ contains a clash, then every tableau $\langle I, T \rangle$ contains a clash. To this end, consider an arbitrary tableau $\langle I, T \rangle$ for KB. We construct a mapping $\pi : \bar{I}_{\mathrm{KB}} \to I$ such that, for all $e, f \in \bar{I}_{\mathrm{KB}}$, $C(e) \in \bar{T}_{\mathrm{KB}}$ implies $C(\pi(e)) \in T$ and $R(e, f) \in \bar{T}_{\mathrm{KB}}$ implies $R(\pi(e), \pi(f)) \in T$ $(*)$. Thus, π is a homomorphism from $\langle \bar{I}_{\mathrm{KB}}, \bar{T}_{\mathrm{KB}} \rangle$ to $\langle I, T \rangle$.

We construct π iteratively by following the construction of $\langle \bar{I}_{\mathrm{KB}}, \bar{T}_{\mathrm{KB}} \rangle$, and we show that property $(*)$ holds throughout the construction. We initialise π by setting $\pi(c) := c$ for every $c \in \mathbf{I}$. This is possible since, by Definitions 5.13 and 5.16, $\mathbf{I} \subseteq \bar{I}_{\mathrm{KB}}$ and $\mathbf{I} \subseteq I$. Since $\bar{T}_{\mathrm{KB}}$ is initialised to $\emptyset$, property $(*)$ holds.

For the induction step, we consider each rule in Fig. 13. For (T1), $(*)$ follows directly from condition (C1). Case (T2) is similar. For (T3), we have $A(e) \in \bar{T}_{\mathrm{KB}}$, so $A(\pi(e)) \in T$ by the induction hypothesis, and $(*)$ follows from (C3). The cases (T4), (T5a), (T6), (T7), and (T8) are similar.

For (T5b), we have $\exists R.\top(e) \in \bar{T}_{\text{KB}}$, so $\exists R.\top(\pi(e)) \in T$ by the induction hypothesis. By (C7) there is some $f \in I$ with $R(\pi(e), f) \in T$. Thus, by (C5), $S(\pi(e), f) \in T$ and, again by (C7), $\exists S.\top(\pi(e)) \in T$ as required.

For (T∃), we have $\exists R.\top(\pi(e)) \in T$ by the induction hypothesis, so there is some $f' \in I$ with $R(\pi(e), f') \in T$. Define $\pi(g) := f'$ for the new individual g introduced in (T∃). Thus $R(\pi(e), \pi(g)) \in T$ as required.

This finishes the induction. By (∗), we find that $\bot(e) \in \bar{T}_{\text{KB}}$ implies $\bot(e) \in T$. Since $\langle I, T\rangle$ was arbitrary, this establishes the claim. □

The algorithm of Definition 5.16 can be viewed as a "breadth-first" construction of a canonical tableau. Due to the explicit procedural description of tableau rules, any role and class expression of the canonical tableau is first computed after a well-defined number of computation steps.[2] Accordingly, we define a total order $\prec$ on $\bar{T}$ by setting $F \prec G$ iff F is computed before G.

The canonical tableau and the order $\prec$ are the main ingredients for showing the correctness of the following non-deterministic decision algorithm. Its definition uses the following notation.

Definition 5.19. Consider a set $T \subseteq \mathcal{T}_I$ with $\mathcal{T}_I$ defined for some set of individuals I and knowledge base KB as in Definition 5.13.

—Given an element $e \in I$, the set $T[e]$ consists of all ABox expressions in T that contain e, i.e., $T[e] := \{\tau \in T \mid e \text{ occurs in } \tau\}$.

—Given elements $e, f \in I$, the set $T_{e\mapsto f}$ is obtained from T by replacing all occurrences of e by f. To simplify notation, $(T_{e\mapsto f})_{e'\mapsto f'}$ is denoted as $T_{e\mapsto f, e'\mapsto f'}$.

Definition 5.20. Consider a Horn-$\mathcal{FLOH}^-$ knowledge base KB with canonical tableau $\langle \bar{I}, \bar{T}\rangle$. A set of individuals I is defined as $I := \mathbf{I} \cup \{a, b\}$, where a, $b \notin \bar{I}$. Non-deterministically select one element $g \in I$, and initialise $T \subseteq \mathcal{T}_I$ by setting $T := \{\bot(g)\}$.

The algorithm repeatedly modifies T by non-deterministically applying one of the following rules:

(N1) Given any $\tau \in \mathcal{T}_I$, set $T := T \cup \{\tau\}$. If τ is a role statement, decide non-deterministically whether τ is marked inactive.

(N2) If there is $\tau \in T$ such that τ can be derived from $T \setminus \{\tau\}$ using one of the rules (T1) to (T8) in Fig. 13, set $T := T \setminus \{\tau\}$. Rules (T6b), (T6c), (T6e), and (T6f) can only be used if τ is marked inactive.

(N3) If $T[a] = \{R(e, a)\}$ for some $e \in I \setminus \{a\}$ such that $\exists R.\top(e) \in T$, set $T := (T \setminus T[a])_{b\mapsto a}$.

(N4) If $T = \emptyset$, return "unsatisfiable."

Intuitively, the non-deterministic algorithm applies rules of the algorithm in Definition 5.16 in reverse order, deleting a conclusion whenever it can be derived from the remaining statements. The anonymous individuals a and b are used to dynamically represent (various) elements from the canonical tableau. Step (N3) is based

[2]For this to be true, one must also specify the order for the involved iterations, e.g., by ordering elements lexicographically and adopting a naming scheme for newly introduced elements. We assume that such an order was chosen.

$\{\bot(a)\}$	initialisation
$\{\bot(a), C(a), D(a), E(a)\}$	(N1) × 3
$\{C(a), D(a), E(a)\}$	(N2): (T3), (8)
$\{C(a), D(a)\}$	(N2): (T3), (7)
$\{C(a), D(a), S(b,a), \exists S.\top(b), \forall S.C(b), \forall S.D(b), B(b)\}$	(N1) × 5
$\{D(a), S(b,a), \exists S.\top(b), \forall S.C(b), \forall S.D(b), B(b)\}$	(N2): (T8)
$\{S(b,a), \exists S.\top(b), \forall S.C(b), \forall S.D(b), B(b)\}$	(N2): (T8)
$\{\exists S.\top(a), \forall S.C(a), \forall S.D(a), B(a)\}$	(N3)
$\{\forall S.C(a), \forall S.D(a), B(a)\}$	(N2): (T3), (4)
$\{\forall S.D(a), B(a)\}$	(N2): (T3), (5)
$\{B(a)\}$	(N2): (T3), (6)
$\{B(a), R(c,a), \exists R.\top(c), \forall R.B(c), A(c), \{c\}(c)\}$	(N1) × 5
$\{R(c,a), \exists R.\top(c), \forall R.B(c)\}$	(N2): (T8)
$\{\exists R.\top(c), \forall R.B(c), A(c), \{c\}(c)\}$	(N3)
$\{\forall R.B(c), A(c), \{c\}(c)\}$	(N2): (T3), (2)
$\{A(c), \{c\}(c)\}$	(N2): (T3), (3)
$\{\{c\}(c)\}$	(N2): (T3), (1)
$\{\}$	(N2): (T2)

Fig. 14. A possible application of the algorithm of Definition 5.20 to the knowledge base in Example 5.14

on rule (T∃). The condition that this rule introduces a new element is reflected by the requirement that $R(e,a)$ is the only statement about a when applying (N3). Thereafter, a is no longer used, and the statements about b are copied to a.

Example 5.21. Figure 14 shows an application of the algorithm of Definition 5.20 to the knowledge base in Example 5.14. Each line specifies the value of the set T of the algorithm, followed by the rule that was used to obtain it. For each application of (N2), we use (T1)–(T8) to specify the respective rule of Fig. 13, and (1)–(7) to refer to the axioms in Example 5.14 that were used. The algorithm thus computes $T = \emptyset$ and can terminate with rule (N4).

LEMMA 5.22. *The algorithm of Definition 5.20 can be executed in polynomially bounded space.*

PROOF. Since $|I|$, $|\mathbf{B}|$, and $|\mathbf{R}|$ are polynomially bounded by the size of the knowledge base, so is $\mathsf{cl}(\mathrm{KB})$ and thus T. □

LEMMA 5.23. *If there is a sequence of choices such that the algorithm of Definition 5.20 returns "unsatisfiable" after some finite time,* KB *is indeed unsatisfiable.*

PROOF. Assume that the algorithm terminates within finitely many steps, where each step involves an application of one of the rules (N1) to (N4). We use T^n to denote the state of the algorithm n steps before termination. In particular, $T^0 = \emptyset$.

We claim that for each T^n there are individuals $a', b' \in \bar{I}$, such that $T^n_{a\mapsto a',\, b\mapsto b'} \subseteq \bar{T}$. This is verified by induction over the number of steps executed by the algorithm. Since $T^0 = \emptyset$, the claim for T^0 holds for any $a', b' \in \bar{I}$.

For the induction step, assume that $T^n_{a\mapsto a',\, b\mapsto b'} \subseteq \bar{T}$. To show the claim for T^{n+1}, we distinguish cases based on the transformation rule that was applied to obtain T^n from T^{n+1}:

(N1) Since $T^{n+1} \subset T^n$, we conclude $T^{n+1}_{a\mapsto a',\, b\mapsto b'} \subseteq \bar{T}$.

(N2) $T^{n+1} = T^n \cup \{\tau\}$, where τ can be derived from T^n by one of the rules (T1) to (T8). Since those rules have been applied exhaustively in $\bar{T}$, we find $T^{n+1}_{a\mapsto a',\, b\mapsto b'} \subseteq \bar{T}$.

(N3) $T^{n+1} = T^n_{a\mapsto b} \cup \{R(e,a)\}$ for some $e \in I\setminus\{a\}$ and $R \in \mathbf{R}$, where $\exists R.\top(e) \in T^n$. By the induction hypothesis, there are $a', b' \in \bar{I}$ such that $T^n_{a\mapsto a',\, b\mapsto b'} \subseteq \bar{T}$, and thus $T^{n+1} \setminus \{R(e,a)\}_{b\mapsto a'} \subseteq \bar{T}$. Moreover, $\exists R.\top(e') \in \bar{T}$ where $e' := b'$ if $e = b$, and $e' := e$ otherwise. Thus, by rule (T∃), there is an individual $f \in \bar{I}$ with $R(e', f) \in \bar{T}$, and we find $\{R(e,a)\}_{a\mapsto f} \subseteq \bar{T}$. Thus, $T^{n+1}_{a\mapsto f, b\mapsto a'} \subseteq \bar{T}$.

Applying the above induction to the initial state $\{\bot(g)\}$, we find $\{\bot(g)\}_{a\mapsto a',\, b\mapsto b'} \in \bar{T}$. Hence $\bar{T}$ does indeed contain a clash and KB is unsatisfiable. □

LEMMA 5.24. *Whenever* KB *is unsatisfiable, there is a sequence of choices such that the algorithm of Definition 5.20 returns "unsatisfiable" after some finite time.*

PROOF. It is not hard to see that active role statements in the canonical tableau form a tree among the elements that are not in $\mathbf{I}$. The following related properties are relevant to our proof:

(P1) If there are active role statements $R(e, f), S(e', f) \in \bar{T}$ with $f \in \bar{I} \setminus \mathbf{I}$, then $e = e'$.

(P2) If there is an active role statement $R(e, f) \in \bar{T}$ with $e \in \bar{I} \setminus \mathbf{I}$, then $f \in \bar{I} \setminus \mathbf{I}$ and $f \neq e$.

These properties can be verified by an easy induction over the rules of Fig. 13. Active role statements can only be derived through (T5a) and (T∃), and it is clear that the required properties are preserved in these cases.

To establish the claim, we specify a possible sequence of choices and show its correctness. If KB is unsatisfiable, there is some element $e \in \bar{I}$ in the canonical tableau such that $\bot(e) \in \bar{T}$. Pick one such e. We use a' and b' to denote the elements of $\bar{I}$ that are currently simulated by a and b. To initialise a' and b', consider some element $\star \notin \bar{I}$. If $e \in \mathbf{I}$, set $a' := b' := \star$; if $e \notin \mathbf{I}$, set $a' := e$ and $b' := \star$. Now the algorithm of Definition 5.20 initialises the set T by setting $T := \{\bot(e)\}_{a'\mapsto a}$. Throughout the computation below, the following properties will be preserved:

(P3) $T_{a\mapsto a',\, b\mapsto b'} \subseteq \bar{T}$ and all role statements in $T_{a\mapsto a',\, b\mapsto b'}$ are active in $\bar{T}$.

(P4) $a', b' \notin \mathbf{I}$, i.e., $a', b' \in \{\star\} \cup \bar{I} \setminus \mathbf{I}$.

(P5) If $T[b] \neq \emptyset$, then $R(b, a) \in T[a] \cap T[b]$ for some $R \in \mathbf{R}$. In particular, $T[b] \neq \emptyset$ implies $T[a] \neq \emptyset$.

In other words, a and b are used to represent anonymous elements that are directly related with an active role statement, and, if used at all, b is necessarily the predecessor of a. Clearly, (P3) to (P5) hold initially.

Rule (N1) of the algorithm will be used repeatedly to close T under relevant inferences that are $\prec$-smaller than some statement τ. Given $\tau \in \bar{T}$, we define:

$$\downarrow\tau = \Big\{C(f) \in \bar{T} \mid C(f) \preceq \tau,\ f \in \mathbf{I} \cup \{a', b'\}\Big\}_{a' \mapsto a,\, b' \mapsto b} \quad \cup$$
$$\Big\{R(f,g) \in \bar{T} \mid R(f,g) \text{ not inactive}, R(f,g) \preceq \tau,\ f,g \in \mathbf{I} \cup \{a', b'\}\Big\}_{a' \mapsto a,\, b' \mapsto b}.$$

This selects *all* active elements in $\bar{T}$ that are $\prec$-smaller than τ and that can be represented using the elements from I with the current representation of a' as a, and b' as b. The algorithm now repeatedly executes steps according to the following choice strategy.

Single Step Choice Strategy. If $T[a]$ is non-empty, let τ' be the $\prec$-largest element of $T[a]_{a \mapsto a'}$. Else, let τ' be the $\prec$-largest element of $T_{a \mapsto a', b \mapsto b'}$. By property (P3), there is some $\tau \in T$ such that $\{\tau\}_{a \mapsto a',\, b \mapsto b'} = \{\tau'\}$ and τ is not inactive. Applying rule (N1), the algorithm first computes $T := T \cup \downarrow\tau$ $(*)$. The algorithm nondeterministically guesses the rule of Fig. 13 that was used to infer τ', and proceeds accordingly:

—If τ' was inferred by one of the rules (T1), (T2), (T3), (T4), (T5a), (T5b), and (T7), the premises of a respective rule application in T have been computed in $(*)$. This is so since the required premises are $\prec$-smaller and not inactive, and since they only involve individuals that are also found in τ, i.e., which are represented by I with the current choice of a' and b'. Hence the algorithm can apply rule (N2) to reduce τ.

—If τ' was inferred by one of the rules of (T6), then one of the premises used was of the form $\{f\}(e)$, and thus $f \in \mathbf{I}$. Since τ' is not inactive, rules (T6b), (T6c), (T6e), and (T6f) are not relevant. We distinguish two cases:
  - —If τ' was inferred by rule (T6a) then τ can directly be reduced by applying rule (N2). The existence of the premises in T follows again from $(*)$.
  - —If τ' was inferred by rules (T6d), then τ' is of the form $C(f)$ and thus $T[a] = \emptyset$. If $e \in \mathbf{I}$, then τ can again be reduced by rule (N2). If $e \notin \mathbf{I}$, set $a' := e$ and use rule (N1) to compute $T[a] = \{\{f\}(a), C(a)\}$. Apply (N2) to reduce τ.

—If τ' was inferred by rule (T8), then $\tau' = C(g)$ for some element g, and there is some element e such that $\{\forall R.C(e), R(e,g)\} \subseteq \bar{T}$. We distinguish two cases:
  - —If $g \in \mathbf{I}$, then $\tau = C(g)$ and, by (P2), $e \in \mathbf{I}$. Thus τ can again be reduced by rule (N2).
  - —If $g \notin \mathbf{I}$, then $g \in \{a', b'\}$. Thus, by (P3), $T[a] \neq \emptyset$ or $T[b] \neq \emptyset$, and therefore $T[a] \neq \emptyset$ by (P5). Hence τ was chosen from $T[a]$, so $\tau = C(a)$ and $g = a'$. By (P2), $e \neq a'$.
    If $e \in \mathbf{I}$, then $\{\forall R.C(e), R(e,a)\} \subseteq T$ by $(*)$. Use rule (N2) to reduce τ.
    Now consider the case $e \notin \mathbf{I}$. If $b' = \star$ then set $b' := e$ and use rule (N1) to compute $T[b] = \{\forall R.C(b), R(b,a)\}$. Use rule (N2) to reduce τ. Otherwise, if $b' \neq \star$, then there is some active role statement $S(b', a') \in \bar{T}$ by (P5). Thus, $b' = e$ by (P1), and rules (N1) and (N2) can be applied as before to reduce τ.

—If τ' was inferred by rule (T$\exists$), we have $\tau' = R(e,g)$ for some newly introduced element $g \notin \mathbf{I}$. Thus $g \in \{a', b'\}$. By (P3), $T[a] \neq \emptyset$ or $T[b] \neq \emptyset$, and therefore $T[a] \neq \emptyset$ by (P5). Hence τ was chosen from $T[a]$, i.e., $e = a'$ or $g = a'$. If $e = a'$, then $g \notin \mathbf{I}$ by (P2), so $g = b'$, which by (P5) contradicts (P1) or (P2). Thus, $g = a'$.

Since $\tau' = R(e, a')$ is the first statement using a' in the canonical tableau, it is the $\prec$-smallest element in $T[a]_{a \mapsto a'}$. Since τ' was chosen to be the $\prec$-largest element of $T[a]_{a \mapsto a'}$, this implies $T[a]_{a \mapsto a'} = \{\tau'\}$ and thus $T[a] = \{\tau\}$. Thus we can apply rule (N3) to reduce τ, and we set $a' := b'$ and $b' := \star$.

It is easy to check that properties (P3) to (P5) are preserved in each of these steps. Due to (P3), it is also clear that one of the above cases must be applicable as long as $T \neq \emptyset$ (all ABox statements in $\bar{T}$ were derived by some rule).

Finally, we need to show that the algorithm terminates. This is established by defining a well-founded termination order. For details on such approaches and the related terminology, see [Baader and Nipkow 1998]. Now considering T as a multiset, the multiset-extension of the well-founded order $\prec$ is a suitable termination order, which is easy to see since in every reduction step, the element τ is deleted, and possibly replaced by one or more elements that are strictly smaller than τ. □

The above lemmata establish an NPSpace decision procedure for detecting unsatisfiability of Horn-$\mathcal{FLOH}^-$ knowledge bases. But NPSpace is known to coincide with PSpace, and we can conclude the main theorem of this section.

Theorem 5.25. *Unsatisfiability of a Horn-$\mathcal{FLOH}^-$ knowledge base* KB *can be decided in space that is polynomially bounded by the size of* KB.

Proof. Combine Lemma 5.22, 5.23, and 5.24 to obtain a non-deterministic time-polynomial decision procedure for detecting unsatisfiability. Apply *Savitch's Theorem* to show the existence of an according PSpace algorithm [Savitch 1970]. □

Summing up the result from the previous two sections, we obtain the following.

Theorem 5.26. *The standard reasoning problems for any description logic between Horn-$\mathcal{FL}^-$ and Horn-$\mathcal{FLOH}^-$ are* PSpace*-complete.*

Proof. Combine Theorem 5.10 and Theorem 5.25. □

6. HORN-$\mathcal{SHIQ}$ AND OTHER EXPTIME-COMPLETE HORN DLS

Horn-$\mathcal{ALC}$ further extends Horn-$\mathcal{FL}^-$ by allowing arbitrary existential role quantifications instead of only unqualified ones. Note that the step from Horn-$\mathcal{FL}^-$ to Horn-$\mathcal{ALC}$ is less significant than that from $\mathcal{FL}^-$ to $\mathcal{ALC}$ since the imposed Hornness restricts the usage of the operators $\sqcup$ and $\neg$ available in $\mathcal{ALC}$.

Definition 6.1. A concept C in pNNF is a *Horn-$\mathcal{ALC}$ concept* if

—C is in $\mathbf{C_1}$ of Fig. 5, and

—C contains only concept constructors $\top$, $\bot$, $\sqcap$, $\sqcup$, $\neg$, $\forall$ and $\exists$.

The description logic Horn-$\mathcal{ALC}$ supports the following axioms:

—TBox axioms $C \sqsubseteq D$ such that $\mathsf{pNNF}(\neg C \sqcup D)$ is a Horn-$\mathcal{ALC}$ concept,

—ABox axioms $C(a)$, $R(a, b)$, and $a \approx b$ such that $\mathsf{pNNF}(C)$ is a Horn-$\mathcal{ALC}$ concept and R is a $\mathcal{SROIQ}^{\text{free}}$ role.

This moderate syntactic extension turns out to raise the complexity of standard reasoning tasks for Horn-$\mathcal{ALC}$ to ExpTime, thus establishing ExpTime-completeness of Horn-$\mathcal{SHIQ}$. Note that inclusion in ExpTime is obvious since

$\mathcal{ALC}$ is a fragment of $\mathcal{SHIQ}$ which is also in ExpTime [Tobies 2001]. To show that Horn-$\mathcal{ALC}$ is ExpTime-hard, we reduce the halting problem of polynomially space-bounded alternating Turing machines, defined next, to the concept subsumption problem.

6.1 Alternating Turing Machines

Definition 6.2. An *alternating Turing machine* (ATM) $\mathcal{M}$ is a tuple (Q, Σ, Δ, q_0) where

—$Q = U \dot\cup E$ is the disjoint union of a finite set of *universal states* U and a finite set of *existential states* E,

—Σ is a finite *alphabet* that includes a *blank* symbol $\Box$,

—$\Delta \subseteq (Q \times \Sigma) \times (Q \times \Sigma \times \{l, r\})$ is a *transition relation*, and

—$q_0 \in Q$ is the *initial state*.

A (universal/existential) *configuration* of $\mathcal{M}$ is a word $\alpha \in \Sigma^* Q \Sigma^*$ ($\Sigma^* U \Sigma^* / \Sigma^* E \Sigma^*$). A configuration α' is a *successor* of a configuration α if one of the following holds:

(1) $\alpha = w_l q \sigma \sigma_r w_r$, $\alpha' = w_l \sigma' q' \sigma_r w_r$, and $(q, \sigma, q', \sigma', r) \in \Delta$,

(2) $\alpha = w_l q \sigma$, $\alpha' = w_l \sigma' q' \Box$, and $(q, \sigma, q', \sigma', r) \in \Delta$,

(3) $\alpha = w_l \sigma_l q \sigma w_r$, $\alpha' = w_l q' \sigma_l \sigma' w_r$, and $(q, \sigma, q', \sigma', l) \in \Delta$,

where $q \in Q$ and $\sigma, \sigma', \sigma_l, \sigma_r \in \Sigma$ as well as $w_l, w_r \in \Sigma^*$. Given some natural number s, the possible *transitions in space s* are defined by additionally requiring that $|\alpha'| \leq s + 1$.

The set of *accepting configurations* is the least set which satisfies the following conditions. A configuration α is accepting iff

—α is a universal configuration and all its successor configurations are accepting, or

—α is an existential configuration and at least one of its successor configurations is accepting.

Note that universal configurations without any successors here play the rôle of accepting final configurations, and thus form the basis for the recursive definition above.

$\mathcal{M}$ *accepts* a given word $w \in \Sigma^*$ (in space s) iff the configuration $q_0 w$ is accepting (when restricting to transitions in space s).

This definition is inspired by the complexity classes NP and co-NP, which are characterised by non-deterministic Turing machines that accept an input if either at least one or all possible runs lead to an accepting state. An ATM can switch between these two modes and indeed turns out to be more powerful than classical Turing machines of either kind. In particular, ATMs can solve ExpTime problems in polynomial space [Chandra et al. 1981].

Definition 6.3. A language L is accepted by a polynomially space-bounded ATM iff there is a polynomial p such that, for every word $w \in \Sigma^*$, $w \in L$ iff w is accepted in space $p(|w|)$.

FACT 6.4. *The complexity class* APSPACE *of languages accepted by polynomially space-bounded ATMs coincides with the complexity class* EXPTIME.

We thus can show EXPTIME-hardness of Horn-$\mathcal{SHIQ}$ by polynomially reducing the halting problem of ATMs with a polynomially bounded storage space to inferencing in Horn-$\mathcal{SHIQ}$. In the following, we exclusively deal with polynomially space-bounded ATMs, and so we omit additions such as "in space s" when clear from the context.

6.2 Simulating ATMs in Horn-$\mathcal{ALC}$

In the following, we consider a fixed ATM $\mathcal{M}$ denoted as in Definition 6.2, and a polynomial p that defines a bound for the required space. For any word $w \in \Sigma^*$, we construct a Horn-$\mathcal{ALC}$ knowledge base $\mathrm{KB}_{\mathcal{M},w}$ and show that acceptance of w by the ATM $\mathcal{M}$ can be decided by inferencing over this knowledge base.

In detail, $\mathrm{KB}_{\mathcal{M},w}$ depends on $\mathcal{M}$ and $p(|w|)$, and has an empty ABox.[3] Acceptance of w by the ATM is reduced to checking concept subsumption, where one of the involved concepts directly depends on w. Intuitively, the elements of an interpretation domain of $\mathrm{KB}_{\mathcal{M},w}$ represent possible configurations of $\mathcal{M}$, encoded by the following concept names:

—A_q for $q \in Q$: the ATM is in state q,
—H_i for $i = 0, \ldots, p(|w|) - 1$: the ATM is at position i on the storage tape,
—$C_{\sigma,i}$ with $\sigma \in \Sigma$ and $i = 0, \ldots, p(|w|) - 1$: position i on the storage tape contains symbol σ,
—A: the ATM accepts this configuration.

This approach is pretty standard, and it is not too hard to axiomatise a successor relation S and appropriate acceptance conditions in $\mathcal{ALC}$ (see, e.g., [Lutz and Sattler 2005]). But this reduction is not applicable in Horn-$\mathcal{SHIQ}$, and it is not trivial to modify it accordingly.

One problem that we encounter is that the acceptance condition of existential states is a (non-Horn) disjunction over possible successor configurations. To overcome this, we encode individual transitions by using a distinguished successor relation for each transition in Δ. This allows us to explicitly state which conditions must hold for a particular successor without requiring disjunction. For the acceptance condition, we use a recursive formulation as employed in Definition 6.2. Acceptance is thus propagated backwards from the final accepting configurations.

In the case of $\mathcal{ALC}$, acceptance of the ATM is reduced to concept satisfiability, i.e., one checks whether an accepting initial configuration can exist. This requires that acceptance is faithfully propagated to successor states, so that any model of the initial concept encodes a valid trace of the ATM. Axiomatising this requires many exclusive disjunctions, such as "The ATM always is in *exactly* one of its states H_i." Since it is not clear how to model this in a Horn DL, we take a dual approach: reducing acceptance to concept subsumption, we require the initial state to be accepting in *all* possible models. Under this approach, we do not need to ensure that every element of every model represents a unique configuration of the

[3]The RBox is empty for $\mathcal{ALC}$ anyway.

(1) Left and right transition rules:	
$A_q \sqcap H_i \sqcap C_{\sigma,i} \sqsubseteq \exists S_\delta.(A_{q'} \sqcap H_{i+1} \sqcap C_{\sigma',i})$	with $\delta = (q,\sigma,q',\sigma',r)$, $i < p(\lvert w\rvert) - 1$
$A_q \sqcap H_i \sqcap C_{\sigma,i} \sqsubseteq \exists S_\delta.(A_{q'} \sqcap H_{i-1} \sqcap C_{\sigma',i})$	with $\delta = (q,\sigma,q',\sigma',l)$, $i > 0$
(2) Memory:	**(3) Existential acceptance:**
$H_j \sqcap C_{\sigma,i} \sqsubseteq \forall S_\delta.C_{\sigma,i} \quad i \neq j$	$A_q \sqcap \exists S_\delta.A \sqsubseteq A$ for all $q \in E$
(4) Universal acceptance:	
$A_q \sqcap H_i \sqcap C_{\sigma,i} \sqcap \bigsqcap_{\delta\in\tilde{\Delta}}(\exists S_\delta.A) \sqsubseteq A$	$q \in U$, $x \in \{r \mid i < p(\lvert w\rvert) - 1\} \cup \{l \mid i > 0\}$
	$\tilde{\Delta} = \{(\tilde{q},\tilde{\sigma},q',\sigma',x) \in \Delta \mid \tilde{q} = q \text{ and } \tilde{\sigma} = \sigma\}$
Rules are instantiated for all $q,q' \in Q$, $\sigma,\sigma' \in \Sigma$, $i,j \in \{0,\dots,p(\lvert w\rvert)-1\}$, and $\delta \in \Delta$.	

Fig. 15. Knowledge base $\mathrm{KB}_{\mathcal{M},w}$ simulating a polynomially space-bounded ATM

ATM. We merely require that every element relates to the necessary successor configurations of all of the configurations it represents. Our encoding ensures that, whenever the initial configuration is not accepting, there is at least one "minimal" model that reflects this.

After this informal introduction, consider the knowledge base $\mathrm{KB}_{\mathcal{M},w}$ given in Fig. 15. The roles S_δ ($\delta \in \Delta$) describe a configuration's successors using the transition δ. The initial configuration for a word w is described by the concept I_w:

$$I_w := A_{q_0} \sqcap H_0 \sqcap C_{\sigma_0,0} \sqcap \dots \sqcap C_{\sigma_{\lvert w\rvert-1},\lvert w\rvert-1} \sqcap C_{\Box,\lvert w\rvert} \sqcap \dots \sqcap C_{\Box,p(\lvert w\rvert)-1},$$

where σ_i denotes the symbol at the ith position of w. We will show that checking whether the initial configuration is accepting is equivalent to checking whether $I_w \sqsubseteq A$ follows from $\mathrm{KB}_{\mathcal{M},w}$. The following is obvious from the characterisation given in Definition 3.1.

Lemma 6.5. $\mathrm{KB}_{\mathcal{M},w}$ *and* $I_w \sqsubseteq A$ *are in Horn-$\mathcal{ALC}$.*

Next we need to investigate the relationship between elements of an interpretation that satisfies $\mathrm{KB}_{\mathcal{M},w}$ and configurations of $\mathcal{M}$. Given an interpretation $\mathcal{I}$ of $\mathrm{KB}_{\mathcal{M},w}$, we say that an element e of the domain of $\mathcal{I}$ *represents* a configuration $\sigma_1 \dots \sigma_{i-1} q \sigma_i \dots \sigma_m$ if $e \in A_q^{\mathcal{I}}$, $e \in H_i^{\mathcal{I}}$, and, for every $j \in \{0,\dots,p(\lvert w\rvert)-1\}$, $e \in C_{\sigma,j}^{\mathcal{I}}$ whenever

$$j \leq m \text{ and } \sigma = \sigma_j \qquad \text{or} \qquad j > m \text{ and } \sigma = \Box.$$

Note that we do not require uniqueness of the above, so that a single element might in fact represent more than one configuration. As we will see below, this does not affect our results. If e represents a configuration as above, we will also say that e has state q, position i, symbol σ_j at position j etc.

Lemma 6.6. *Consider some interpretation $\mathcal{I}$ that satisfies* $\mathrm{KB}_{\mathcal{M},w}$. *If some element e of $\mathcal{I}$ represents a configuration α and some transition δ is applicable to α, then e has an $S_\delta^{\mathcal{I}}$-successor that represents the (unique) result of applying δ to α.*

Proof. Consider an element e, state α, and transition δ as in the claim. Then one of the axioms (1) applies, and e must also have an $S_\delta^{\mathcal{I}}$-successor. This successor represents the correct state, position, and symbol at position i of e, again by the axioms (1). By axiom (2), symbols at all other positions are also represented by all $S_\delta^{\mathcal{I}}$-successors of e. $\Box$

LEMMA 6.7. *A word w is accepted by $\mathcal{M}$ iff $I_w \sqsubseteq A$ is a consequence of* $\mathrm{KB}_{\mathcal{M},w}$.

PROOF. Consider an arbitrary interpretation $\mathcal{I}$ that satisfies $\mathrm{KB}_{\mathcal{M},w}$. We first show that, if any element e of $\mathcal{I}$ represents an accepting configuration α, then $e \in A^{\mathcal{I}}$.

We use an inductive argument along the recursive definition of acceptance. If α is a universal configuration then all successors of α are accepting, too. By Lemma 6.6, for any δ-successor α' of α there is a corresponding $S_\delta^{\mathcal{I}}$-successor e' of e. By the induction hypothesis for α', e' is in $A^{\mathcal{I}}$. Since this holds for all δ-successors of α, axiom (4) implies $e \in A^{\mathcal{I}}$. Especially, this argument covers the base case where α has no successors.

If α is an existential configuration, then there is some accepting δ-successor α' of α. Again by Lemma 6.6, there is an $S_\delta^{\mathcal{I}}$-successor e' of e that represents α', and $e' \in A^{\mathcal{I}}$ by the induction hypothesis. Hence axiom (3) applies and also conclude $e \in A^{\mathcal{I}}$.

Since all elements in $I_w^{\mathcal{I}}$ represent the initial configuration of the ATM, this shows that $I_w^{\mathcal{I}} \subseteq A^{\mathcal{I}}$ whenever the initial configuration is accepting.

It remains to show the converse: if the initial configuration is not accepting, there is some interpretation $\mathcal{I}$ such that $I_w^{\mathcal{I}} \not\subseteq A^{\mathcal{I}}$. To this end, we define a canonical interpretation M of $\mathrm{KB}_{\mathcal{M},w}$ as follows. The domain of M is the set of all configurations of $\mathcal{M}$ that have size $p(|w|) + 1$ (i.e., that encode a tape of length $p(|w|)$, possibly with trailing blanks). The interpretations for the concepts A_q, H_i, and $C_{\sigma,i}$ are defined as expected so that every configuration represents itself but no other configuration. Especially, I_w^M is the singleton set containing the initial configuration. Given two configurations α and α', and a transition δ, we define $(\alpha, \alpha') \in S_\delta^M$ iff there is a transition δ from α to α'. A^M is defined to be the set of accepting configurations.

By checking the individual axioms of Fig. 15, it is easy to see that M satisfies $\mathrm{KB}_{\mathcal{M},w}$. Now if the initial configuration is not accepting, $I_w^M \not\subseteq A^M$ by construction. Thus M is a counterexample for $I_w \sqsubseteq A$, which thus is not a logical consequence. □

We can summarise our results as follows.

THEOREM 6.8. *The standard reasoning problems for any description logic between Horn-$\mathcal{ALC}$ and Horn-$\mathcal{SHIQ}$ are* EXPTIME*-complete.*

PROOF. Inclusion is obvious as Horn-$\mathcal{SHIQ}$ is a fragment of $\mathcal{SHIQ}$, for which these problems are in EXPTIME [Tobies 2001]. Regarding hardness, Lemma 6.7 shows that the word problem for polynomially space-bounded ATMs can be reduced to checking concept subsumption in $\mathrm{KB}_{\mathcal{M},w}$. The other standard reasoning problems can be reduced to satisfiability checking by Proposition 2.8. By Lemma 6.5, $\mathrm{KB}_{\mathcal{M},w}$ is in Horn-$\mathcal{ALC}$. The reduction is polynomially bounded due to the restricted number of axioms: there are at most $p(|w|) \times |\Delta|$ axioms of type (1), $p(|w|)^2 \times |\Sigma| \times |\Delta|$ of type (2), $|Q| \times |\Delta|$ of type (3), and $|Q| \times p(|w|) \times |\Sigma|$ of type (4). □

The proof that was used to establish the previous result is suitable for obtaining further complexity results for logical fragments that are not above Horn-$\mathcal{ALC}$.

THEOREM 6.9. *Consider the description logics*

(a) $\mathcal{ELF}$*, the fragment of* $\mathcal{SROIQ}^{\text{free}}$ *that supports ABoxes and TBoxes using the constructors* $\top$, $\exists$, $\sqcap$, *and number restrictions of the form* $\leqslant 1\, R.\top$,

(b) Horn-$\mathcal{FLE}$ *obtained by extending Horn-*$\mathcal{FL}^-$ *with GCIs of the form* $\exists R.A \sqsubseteq B$,

(c) Horn-$\mathcal{FL}\circ^-$ *obtained by extending Horn-*$\mathcal{FL}^-$ *with composition of roles while restricting to regular RBoxes, and*

(d) Horn-$\mathcal{FLI}^-$ *obtained by extending Horn-*$\mathcal{FL}^-$ *with inverse roles.*

Concept subsumption is EXPTIME*-hard for Horn-*$\mathcal{FL}\circ^-$*, and* EXPTIME*-complete for* $\mathcal{ELF}$*, Horn-*$\mathcal{FLE}$*, and Horn-*$\mathcal{FLI}^-$*.*

PROOF. The results are established by modifying the knowledge base $\mathrm{KB}_{\mathcal{M},w}$ to suit the given fragment. We restrict to providing the required modifications; the full proofs are analogous to the proof for Horn-$\mathcal{ALC}$.

(a) Replace axioms (2) in Fig. 15 with the following statements:

$$\top \sqsubseteq \leq 1\, S_\delta.\top \qquad H_j \sqcap C_{\sigma,i} \sqcap \exists S_\delta.\top \sqsubseteq \exists S_\delta.C_{\sigma,i},\ i \neq j.$$

(b) Replace axioms (1) with axioms of the form

$$A_q \sqcap H_i \sqcap C_{\sigma,i} \sqsubseteq \exists S_\delta.\top \sqcap \forall S_\delta.(A_{q'} \sqcap H_{i\pm 1} \sqcap C_{\sigma',i}).$$

All occurrences of existential restrictions $\exists R.C$ on the left-hand side of GCIs are replaced by a fresh concept name X for which the axiom $\exists R.C \sqsubseteq X$ is added.

(c) Axioms (1) are replaced as in (b). We introduce roles $R_{A\delta}$ for each transition δ, and replace every occurrence of $\exists S_\delta.A$ with $\exists R_{A\delta}.\top$. Moreover, every remaining occurrence of concept A is replaced with $\exists R_A.\top$, with R_A a new role. Finally, the following axioms are added:

$$S_\delta \circ R_A \sqsubseteq R_{A\delta} \quad \text{for each } \delta \in \Delta.$$

(d) Axioms (1) are replaced as in (b). Every occurrence of $\exists S_\delta.A$ is replaced with a new concept name $A_{S\delta}$, and the following axioms are added:

$$A \sqsubseteq \forall S_\delta^{-1}.A_{S\delta} \quad \text{for each } \delta \in \Delta.$$

It is easy to see that those changes allow for analogous reductions. The membership results for $\mathcal{ELF}$, Horn-$\mathcal{FLE}$, and Horn-$\mathcal{FLI}^-$ are immediate from their inclusion in $\mathcal{SHIQ}$. □

EXPTIME-completeness of $\mathcal{ELF}$ was shown in [Baader et al. 2005] (where it was called $\mathcal{EL}^{\leq 1}$), but the above theorem provides a more direct proof.

7. RELATED WORK

Horn-$\mathcal{SHIQ}$ has originally been introduced in [Hustadt et al. 2005] where it has been defined as discussed in Section 3 but with additional implicit restrictions related to the presence of transitivity. The latter was caused by a method of transitivity elimination that creates non-Horn axioms of the form $\forall R.A \sqsubseteq \forall R.\forall R.A$ for transitive roles R which must be taken into account when defining Horn-$\mathcal{SHIQ}$.

As discussed in Section 3, this problem can be avoided by encoding transitivity (and other RIAs) by means of automata encoding techniques as used in [Demri and Nivelle 2005] which have first been applied to DLs in [Kazakov 2008]. Taking this into account, our formulation of Horn-$\mathcal{SHIQ}$ is slightly more general than the one in [Hustadt et al. 2005] and than the formulations used in precursors to this work [Krötzsch et al. 2006a; Krötzsch et al. 2006b; Krötzsch et al. 2007]. While the data complexity of Horn-$\mathcal{SHIQ}$ has been one of the main motives for defining it in [Hustadt et al. 2005], the combined complexity result reported herein is new. Recent investigations revealed that even entailment of conjunctive queries for Horn-$\mathcal{SHOIQ}$ can be performed in ExpTime [Ortiz et al. 2011], whereas this problem is known to be 2ExpTime-complete for $\mathcal{SHIQ}$ [Glimm et al. 2008] and even co-N2ExpTime-hard for $\mathcal{ALCOIF}$ [Glimm et al. 2011]. Another recent result established the exact reasoning complexity of Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ to be ExpTime and 2ExpTime, respectively [Ortiz et al. 2010].

The lower data complexity of reasoning in Horn-$\mathcal{SHIQ}$ has first been exploited by the KAON2 system as described in [Motik 2006; Motik and Sattler 2006]. Further algorithms and implementations have since been able to exploit the simpler structure of Horn knowledge bases to achieve tangible performance gains. An example is the *hypertableau* reasoner HermiT that can handle arbitrary $\mathcal{SROIQ}$ (OWL 2) knowledge bases [Motik et al. 2009]. The "consequence-driven" reasoning method of [Kazakov 2009] is restricted to Horn-$\mathcal{SHIQ}$, but shows outstanding performance for practically relevant ontologies that fall into that fragment. The restriction of consequence-driven reasoning to Horn DLs has recently been relaxed [Simančík et al. 2011].

Other notable examples of Horn DLs are light-weight description logics. Indeed, disjunctive information makes reasoning NP-hard in all DLs that support conjunction and GCIs, and hence it is excluded from DLs that allow for polynomial-time reasoning. Thus, it is no surprise to find that $\mathcal{EL}^{++}$ [Baader et al. 2005; 2008] and various versions of DL-Lite [Calvanese et al. 2007] are Horn DLs in the sense of this paper. The same is true for various formulations of DLP [Grosof et al. 2003; Volz 2004], as has already been observed in Section 4.

Reducing inference problems of DL to inference problems of corresponding Datalog programs has been considered in a number of approaches. Examples include resolution-based approaches for $\mathcal{EL}$ [Kazakov 2006], for its extension ELP [Krötzsch et al. 2008] and for $\mathcal{SHIQ}$ [Hustadt et al. 2005; Motik 2006], as well as approaches for $\mathcal{SHIQ}$ based on ordered binary decision diagrams [Rudolph et al. 2008d; 2008c]. In many of these cases, disjunctive Datalog is required [Motik 2006; Rudolph et al. 2008d; 2008c]. Some encodings naturally lead to Datalog without disjunctions when applied to Horn DLs [Hustadt et al. 2005; Kazakov 2006; Krötzsch et al. 2008], while others use disjunctions in this case as well [Rudolph et al. 2008d; 2008c].

The description logic $\mathcal{FL}^-$ dates back to [Brachman and Levesque 1984] where it was introduced as a presumably tractable variant of the frame language $\mathcal{FL}$. While subsumption of *individual concept expressions* can indeed be decided in polynomial time, the subsumption problem for $\mathcal{FL}^-$ and even in $\mathcal{FL}_0$ is ExpTime-hard in the presence of arbitrary $\mathcal{FL}^-$ TBoxes, as was first shown by McAllester in an unpublished manuscript of 1991 [Donini et al. 1996].

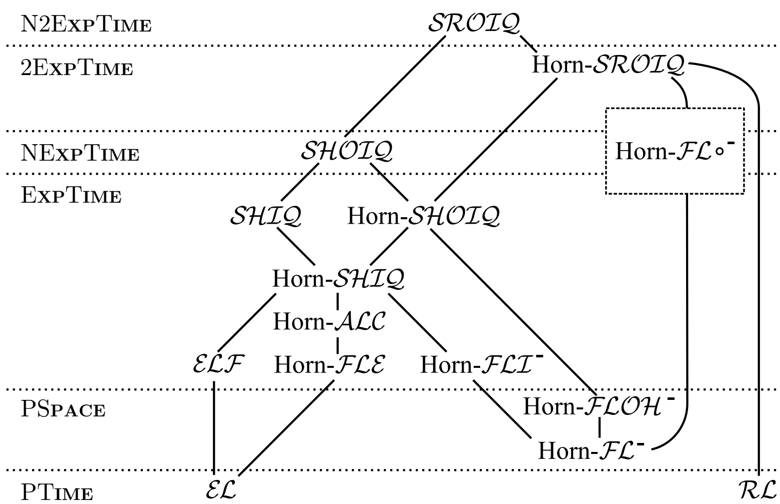

Fig. 16. Reasoning complexities of Horn DLs; the exact position of Horn-$\mathcal{FL}\circ^-$ is not known

8. CONCLUSIONS

In this paper, we have generalised the well-known definition of Horn-$\mathcal{SHIQ}$ to Horn-$\mathcal{SROIQ}^{\text{free}}$, the Horn fragment of $\mathcal{SROIQ}$ without structural restrictions on regularity or simplicity. This also led us to a simplified characterisation of Horn DLs based on a formal grammar. We have then studied a number of increasingly expressive Horn description logics that are obtained as fragments of Horn-$\mathcal{SROIQ}^{\text{free}}$ w.r.t. their worst-case inferencing complexities. The reported results are summarised in Fig. 16. Some non-Horn DLs – $\mathcal{SHIQ}$, $\mathcal{SHOIQ}$, and $\mathcal{SROIQ}$ – are also displayed in this context, while $\mathcal{FL}_0$ and $\mathcal{FL}^-$ (both ExpTime) are omitted to simplify the presentation. The complexity results for Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ do not follow from this work: they have been established by Ortiz et al. [2010].

The entry for Horn-$\mathcal{FL}\circ^-$ in Fig. 16 is displayed in a dotted box to indicate that its exact position is not certain. We have established ExpTime hardness, which suffices to demonstrate that this extension of Horn-$\mathcal{FL}^-$ does no longer admit reasoning in PSpace.[4] The 2ExpTime upper bound for the complexity follows from the according result for Horn-$\mathcal{SROIQ}$ [Ortiz et al. 2010]. Further checks are needed to determine the exact complexity of Horn-$\mathcal{FL}\circ^-$. But when considering the fact that no Horn DL is known to be complete for a non-deterministic complexity class, it seems to be very unlikely that this DL is complete for NExpTime. Indeed, we conjecture that this avoidance of non-determinism is inherent to Horn DLs.

A tableau algorithm for reasoning in description logics between Horn-$\mathcal{FL}^-$ and Horn-$\mathcal{FLOH}^-$ has been devised to show the upper complexity bound for reasoning in these logics. In essence, this algorithm achieves its goal in polynomial space by storing only very small portions of the constructed tableau, corresponding to very restricted "local" environments in the according model. The main result there-

[4]Unless PSpace = ExpTime.

fore consists in showing that such an extremely limited view suffices for complete reasoning in the considered logics. As opposed to $\mathcal{RL}$, the addition of nominals to Horn-$\mathcal{FL}^-$ significantly complicates reasoning procedures, although it does not lead to increased worst-case complexities. Due to a high amount of unguided nondeterminism, the tableau algorithm for Horn-$\mathcal{FLOH}^-$ is clearly unsuitable for practical implementation.

Another important theme in this paper was to establish hardness results that require only a minimal amount of logical expressivity, and which can therefore be useful to derive hardness results for many other DLs as well. This was achieved by directly simulating Turing machine computations in terms of DL inferencing, where polynomially space-bounded Alternating Turing Machines have been found a convenient tool for showing ExpTime hardness. The versatility of this approach was illustrated by deriving a number of additional hardness results for extensions of $\mathcal{EL}$ and $\mathcal{FL}^-$ which extended or strengthened existing results.

Generally, over the last years, the ongoing investigation of Horn fragments of widely adopted DLs has produced many new insights in terms of the syntactic description of such fragments and the computational complexity of reasoning tasks, but it has also provided valuable stimuli for practical developments, such as the definition of tractable ontology formalisms and the implementation of reasoning engines which benefit from the fact that many practically occurring ontologies are at least "almost Horn."

Acknowledgements. We would like to thank the anonymous reviewers who have provided detailed and insightful comments on this work and its precursors.

REFERENCES

Baader, F., Brandt, S., and Lutz, C. 2005. Pushing the $\mathcal{EL}$ envelope. See Kaelbling and Saffiotti [2005], 364–369.

Baader, F., Brandt, S., and Lutz, C. 2008. Pushing the $\mathcal{EL}$ envelope further. In *Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, K. G. Clark and P. F. Patel-Schneider, Eds. CEUR Workshop Proceedings, vol. 496. CEUR-WS.org.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., Eds. 2007. *The Description Logic Handbook: Theory, Implementation, and Applications*, Second ed. Cambridge University Press.

Baader, F. and Nipkow, T. 1998. *Term Rewriting and All That.* Cambridge University Press.

Brachman, R. J. and Levesque, H. J. 1984. The tractability of subsumption in frame-based description languages. In *Proceedings of the 4th National Conference on Artificial Intelligence (AAAI'84)*, R. J. Brachman, Ed. AAAI Press, 34–37.

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

Chandra, A. K., Kozen, D. C., and Stockmeyer, L. J. 1981. Alternation. *Journal of the ACM 28,* 1, 114–133.

Dantsin, E., Eiter, T., Gottlob, G., and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys 33,* 3, 374–425.

Demri, S. and Nivelle, H. 2005. Deciding regular grammar logics with converse through first-order logic. *Journal of Logic, Language and Information 14,* 3, 289–329.

Donini, F. M., Lenzerini, M., Nardi, D., and Schaerf, A. 1996. Reasoning in description logics. In *Principles of Knowledge Representation*, G. Brewka, Ed. Studies in Logic, Language, and Information. CLSI Publications, 193–238.

GLIMM, B., KAZAKOV, Y., AND LUTZ, C. 2011. Status $\mathcal{QIO}$: An update. In *Proceedings of the 24th International Workshop on Description Logics (DL'11)*, R. Rosati, S. Rudolph, and M. Zakharyaschev, Eds. CEUR Workshop Proceedings, vol. 745. CEUR-WS.org.

GLIMM, B., LUTZ, C., HORROCKS, I., AND SATTLER, U. 2008. Answering conjunctive queries in the $\mathcal{SHIQ}$ description logic. *Journal of Artificial Intelligence Research 31*, 150–197.

GROSOF, B. N., HORROCKS, I., VOLZ, R., AND DECKER, S. 2003. Description logic programs: combining logic programs with description logic. In *Proceedings of the 12th International Conference on World Wide Web (WWW'03)*. ACM, 48–57.

HITZLER, P., KRÖTZSCH, M., AND RUDOLPH, S. 2009. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC.

HUSTADT, U., MOTIK, B., AND SATTLER, U. 2005. Data complexity of reasoning in very expressive description logics. See Kaelbling and Saffiotti [2005], 466–471.

KAELBLING, L. AND SAFFIOTTI, A., Eds. 2005. *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*. Professional Book Center.

KAZAKOV, Y. 2006. Saturation-based decision procedures for extensions of the guarded fragment. Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany.

KAZAKOV, Y. 2008. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, G. Brewka and J. Lang, Eds. AAAI Press, 274–284.

KAZAKOV, Y. 2009. Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In *Proceedings of the 21st International Conference on Artificial Intelligence (IJCAI'09)*, C. Boutilier, Ed. IJCAI, 2040–2045.

KRÖTZSCH, M. 2011. Efficient rule-based inferencing for OWL EL. See Walsh [2011], 2668–2673.

KRÖTZSCH, M., HITZLER, P., VRANDEČIĆ, D., AND SINTEK, M. 2006a. How to reason with OWL in a logic programming system. In *Proceedings of the 2nd International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML'06)*. IEEE Computer Society Press.

KRÖTZSCH, M., RUDOLPH, S., AND HITZLER, P. 2006b. On the complexity of Horn description logics. In *Proceedings of the 2nd Workshop on OWL: Experiences and Directions*, B. Cuenca Grau, P. Hitzler, C. Shankey, and E. Wallace, Eds. CEUR WS Proceedings, vol. 216. CEUR-WS.org.

KRÖTZSCH, M., RUDOLPH, S., AND HITZLER, P. 2007. Complexity boundaries for Horn description logics. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI'07)*. AAAI Press, 452–457.

KRÖTZSCH, M., RUDOLPH, S., AND HITZLER, P. 2008. ELP: Tractable rules for OWL 2. See Sheth et al. [2008], 649–664.

KRÖTZSCH, M., RUDOLPH, S., AND SCHMITT, P. H. 2010. On the semantic relationship between Datalog and description logics. In *Proceedings of the 4th Interational Conference on Web Reasoning and Rule Systems (RR'10)*. LNCS, vol. 6333. Springer, 88–102.

KRÖTZSCH, M., SIMANČÍK, F., AND HORROCKS, I. 2012. A description logic primer. *CoRR abs/1201.4089*.

LUTZ, C. AND SATTLER, U. 2005. Description Logics. Lecture at the ICCL Summer School 2005, Dresden, Germany. Slides available at `http://www.computational-logic.org/content/events/iccl-ss-2005/`.

MOTIK, B. 2006. Reasoning in description logics using resolution and deductive databases. Ph.D. thesis, Universität Karlsruhe (TH), Germany.

MOTIK, B., CUENCA GRAU, B., HORROCKS, I., WU, Z., FOKOUE, A., AND LUTZ, C., Eds. 27 October 2009. *OWL 2 Web Ontology Language: Profiles*. W3C Recommendation. Available at `http://www.w3.org/TR/owl2-profiles/`.

MOTIK, B. AND SATTLER, U. 2006. A comparison of reasoning techniques for querying large description logic ABoxes. In *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligencen, and Reasoning (LPAR'01)*, M. Hermann and A. Voronkov, Eds. LNCS, vol. 4246. Springer, 227–241.

Motik, B., Shearer, R., and Horrocks, I. 2009. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research 36*, 165–228.

Ortiz, M., Rudolph, S., and Šimkus, M. 2010. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR'10)*, F. Lin, U. Sattler, and M. Truszczynski, Eds. AAAI Press, 269–279.

Ortiz, M., Rudolph, S., and Šimkus, M. 2011. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. See Walsh [2011], 1039–1044.

OWL Working Group, W. 27 October 2009. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation. Available at http://www.w3.org/TR/owl2-overview/.

Rudolph, S., Krötzsch, M., and Hitzler, P. 2008a. All elephants are bigger than all mice. In *Proceedings of the 21st International Workshop on Description Logics (DL'08)*, F. Baader, C. Lutz, and B. Motik, Eds. CEUR Workshop Proceedings, vol. 353. CEUR-WS.org.

Rudolph, S., Krötzsch, M., and Hitzler, P. 2008b. Cheap Boolean role constructors for description logics. In *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA'08)*, S. Hölldobler, C. Lutz, and H. Wansing, Eds. LNAI, vol. 5293. Springer, 362–374.

Rudolph, S., Krötzsch, M., and Hitzler, P. 2008c. Description logic reasoning with decision diagrams: Compiling $\mathcal{SHIQ}$ to disjunctive Datalog. See Sheth et al. [2008], 435–450.

Rudolph, S., Krötzsch, M., and Hitzler, P. 2008d. Terminological reasoning in $\mathcal{SHIQ}$ with ordered binary decision diagrams. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI'08)*. AAAI Press, 529–534.

Savitch, W. J. 1970. Relationship between nondeterministic and deterministic tape complexities. *Journal of Computer and System Science 4*, 177–192.

Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., and Thirunarayan, K., Eds. 2008. *Proceedings of the 7th International Semantic Web Conference (ISWC'08)*. LNCS, vol. 5318. Springer.

Simančík, F., Kazakov, Y., and Horrocks, I. 2011. Consequence-based reasoning beyond Horn ontologies. See Walsh [2011], 1093–1098.

Tobies, S. 2001. Complexity results and practical algorithms for logics in knowledge representation. Ph.D. thesis, RWTH Aachen, Germany.

Volz, R. 2004. Web ontology reasoning with logic databases. Ph.D. thesis, Universität Karlsruhe (TH), Germany.

Walsh, T., Ed. 2011. *Proceedings of the 22nd International Conference on Artificial Intelligence (IJCAI'11)*. AAAI Press/IJCAI.