

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

Description Logics (DLs) have become a prominent paradigm for representing knowledge bases in a variety of application areas. Central to leveraging them for corresponding systems is the provision of a favourable balance between expressivity of the knowledge representation formalism on the one hand, and runtime performance of reasoning algorithms on the other. Due to this, Horn description logics (Horn DLs) have attracted attention since their (worst-case) data complexities are in general lower than their overall (i.e. combined) complexities, which makes them attractive for reasoning with large sets of instance data (ABoxes). However, the natural question whether Horn DLs also provide advantages for schema (TBox) reasoning has hardly been addressed so far. In this paper, we therefore provide a thorough and comprehensive analysis of the combined complexities of Horn DLs. While the combined complexity for many Horn DLs studied herein turns out to be the same as for their non-Horn counterparts, we identify subboolean DLs where Hornness simplifies reasoning. We also provide convenient normal forms for Horn DLs.

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 availability of expressive language features for the knowledge modelling process, while at the same time striving for the most inexpensive languages in terms of computational complexity. A particularly prominent

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 the Deutsche Forschungsgemeinschaft (DFG) under the ExpresST project, and by the National Science Foundation under award 1017225 *III: Small: TROn—Tractable Reasoning with Ontologies*.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2011 ACM 1529-3785/2011/0700-0001 \$5.00

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 [Grosz 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 of the World Wide Web consortium (W3C) adopted several of them as designated important fragments of OWL [Motik et al. 2009].

At the same time, Horn DLs have been introduced [Grosz et al. 2003; Hustadt et al. 2005], which 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 formula. Many kinds of rules in logic programming, and especially Datalog rules, thus correspond to Horn clauses. In terms of Datalog, the restriction to Horn clauses disallows disjunctions in the head of rules, and thus allows for deterministic evaluation strategies. This simplification is also visible in terms of computational complexities: inferencing in Datalog is EXPTIME-complete w.r.t. the size of the program, while it is NEXPTIME-complete in disjunctive Datalog. Similar differences are found when considering *data complexity*, the complexity of inferencing w.r.t. the number of ground facts of the program, which increases from P to (co-)NP when adding disjunctions.

This has motivated the study of cases where DL knowledge bases can be reduced to Datalog in such a way that it results in non-disjunctive Datalog programs, i.e. in sets of Horn clauses, and the corresponding description logics have been dubbed *Horn description logics* accordingly. The first and most prominent such DL was Horn-*SHIQ*, which was obtained naturally from the KAON2 system [Motik and Sattler 2006], but other well-known DLs such as \mathcal{EL}^{++} [Baader et al. 2005] also share characteristics of Horn DLs. Since, under these approaches, ABox facts can usually be directly rewritten into Datalog facts, Horn description logics necessarily allow standard inference tasks to be solved in polynomial time w.r.t. the size of the ABox that contains no complex concepts (i.e., in terms of data complexity). It turned out that this useful property of Horn DLs can also be exploited in inferencing algorithms that do not rely on reductions to Datalog, e.g., in [Motik et al. 2009; Kazakov 2009].

In this paper, we generalise the definition of Horn-*SHIQ* to arbitrary DLs that are fragments of *SROIQ*, and we provide a comprehensive analysis of the worst-case complexities of the resulting logics. While low data complexity is a characteristic (and well-known) feature of Horn DLs, our results show that the complexity of inferencing w.r.t. the overall size of the knowledge base is not necessarily lower in the Horn case. However, we are able to identify restricted DLs for which inferencing is significantly harder than for their Horn versions.

Our observations also highlight the close connections of Horn DLs to *Description Logic Programs* (DLP) [Grosz et al. 2003] that have been proposed as an “inter-

section” of Horn and description logic. Indeed basic DLP languages are interesting simple formalisms that allow for straightforward rule-based implementations. This was one of the central motivations for the definition of the OWL 2 RL ontology language [Motik et al. 2009] which we can also relate to a suitable Horn DL below.

The paper is structured as follows. In Section 2 we recall some preliminaries required throughout the paper. In Section 3 we define Horn- $SROIQ^{\text{free}}$ as a large Horn DL that provides the framework for defining the more specific logics that are considered herein. Increasingly expressive fragments of Horn- $SROIQ^{\text{free}}$ are studied in subsequent sections. Section 4 introduces the tractable Horn- \mathcal{FL}_0 , Section 5 shows that reasoning for all DLs between Horn- \mathcal{FL}^- and Horn- \mathcal{FLOH}^- is PSPACE-complete, and Section 6 establishes EXPTIME-completeness for all DLs between Horn- \mathcal{FLE} and Horn- \mathcal{SHIQ} . An overview of related work is provided in Section 7, and the results are discussed in Section 8.

This article is a significantly rewritten and extended compilation of [Krötzsch et al. 2006; Krötzsch et al. 2006; Krötzsch et al. 2007].

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. [Baader et al. 2007] provides introductory and advanced material on many aspects of DL research, while a textbook introduction to description logics in the context of Semantic Web technologies can be found in [Hitzler et al. 2009].

Because it will make our content more easily accessible, we first define a very general description logic, called $SROIQ^{\text{free}}$, and then specialize this definition, throughout the paper, as needed for introducing other DLs.

2.1 Syntax

$SROIQ^{\text{free}}$ 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}_s and *non-simple roles* \mathbf{N}_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}_n or \mathbf{N}_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 $\mathcal{S} = \langle \mathbf{I}, \mathbf{A}, \mathbf{N} \rangle$ with $\mathbf{N} = \mathbf{N}_s \cup \mathbf{N}_n$. The set \mathbf{R} of $\mathcal{SROIQ}^{\text{free}}$ role expressions (or simply roles) for \mathcal{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 $\mathbf{R}_s \subseteq \mathbf{R}$ of all *simple role expressions* is defined to contain all role expressions that contain no non-simple role names. The set \mathbf{R}_n of *non-simple role expressions* is $\mathbf{R}_n := \mathbf{R} \setminus \mathbf{R}_s$. A bijective function $\text{Inv} : \mathbf{R} \rightarrow \mathbf{R}$ is defined by setting $\text{Inv}(R) := R^-$, $\text{Inv}(R^-) := R$, and $\text{Inv}(U) := U$ for all $R \in \mathbf{N}$.

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

$$\mathbf{C} ::= \top \mid \perp \mid \mathbf{A} \mid \{\mathbf{I}\} \mid \exists \mathbf{R}.\text{Self} \mid \neg \mathbf{C} \mid (\mathbf{C} \sqcap \mathbf{C}) \mid (\mathbf{C} \sqcup \mathbf{C}) \mid \forall \mathbf{R}.\mathbf{C} \mid \exists \mathbf{R}.\mathbf{C} \mid \geq n \mathbf{R}.\mathbf{C} \mid \leq n \mathbf{R}.\mathbf{C}$$

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.

Parentheses are typically omitted if the exact structure of a given concept expression is clear or irrelevant. Also, we will commonly assume a signature and according 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 $\mathcal{SROIQ}^{\text{free}}$ RBox axiom is an expression of one of the following forms:

- $R_1 \circ \dots \circ R_n \sqsubseteq R$ where $R_1, \dots, R_n, R \in \mathbf{R}$ and where $R \notin \mathbf{R}_n$ only if $n = 1$ and $R_1 \in \mathbf{R}_s$,
- $\text{Ref}(R)$ (reflexivity), $\text{Tra}(R)$ (transitivity), $\text{Irr}(R)$ (irreflexivity), $\text{Dis}(R, R')$ (role disjointness), $\text{Sym}(R)$ (symmetry), $\text{Asy}(R)$ (asymmetry), where $R, R' \in \mathbf{R}$.

A $\mathcal{SROIQ}^{\text{free}}$ TBox axiom is an expression of the form $C \sqsubseteq D$ or $C \equiv D$ with $C, D \in \mathbf{C}$. A $\mathcal{SROIQ}^{\text{free}}$ 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 $R_1 \circ \dots \circ R_n \sqsubseteq R$ are also known as *role inclusion axioms* (RIAs), and a RIA is said to be *complex* if $n > 1$. Expressions such as $\text{Ref}(R)$ are called *role characteristics*. 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 $U \in \mathbf{R}_s$. 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 (cf. Definition 2.4) without leading to undecidability or increased worst-case complexity of reasoning [Rudolph et al. 2008b]. 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*

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

Fig. 1. Nomenclature for important DL features

(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 $\exists S.\text{Self}$, $\geq n S.E$, or $\leq n S.E$ are such that $S \in \mathbf{R}_s$ 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:

$$R \circ R \sqsubseteq R, \quad \text{Inv}(R) \sqsubseteq R,$$

$$R_1 \circ \dots \circ R_n \sqsubseteq R, \quad R \circ R_1 \circ \dots \circ R_n \sqsubseteq R, \quad R_1 \circ \dots \circ R_n \circ R \sqsubseteq R$$

such that $R, R_1, \dots, R_n \in \mathbf{R}$, and $R_i \prec R$ for $i = 1, \dots, n$.

A \mathcal{SROIQ} RBox is a regular $\mathcal{SROIQ}^{\text{free}}$ RBox that contains role characteristics of the forms $\text{lrr}(S)$, $\text{Dis}(S, T)$, and $\text{Asy}(S)$ only for simple role names $S, T \in \mathbf{N}_s$. 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. For example,

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	\perp	\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	$\{a\}$	$\{a^{\mathcal{I}}\}$
univ. restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \langle x, y \rangle \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\}$
exist. restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \text{for some } y \in \Delta^{\mathcal{I}}, \langle x, y \rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
Self concept	$\exists S.\text{Self}$	$\{x \in \Delta^{\mathcal{I}} \mid \langle x, x \rangle \in S^{\mathcal{I}}\}$
qualified number	$\leq n S.C$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid \langle x, y \rangle \in S^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\}$
restriction	$\geq n S.C$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid \langle x, y \rangle \in S^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\}$

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

\mathcal{SHIQ} is the fragment of \mathcal{SROIQ} that does not allow nominals, and that restricts to RBox axioms of the form $\text{Tra}(R)$, $\text{Sym}(R)$, and $S \sqsubseteq R$. 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 $\mathcal{S} = \langle \mathbf{I}, \mathbf{A}, \mathbf{N} \rangle$ 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 \mathcal{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.

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 $\mathcal{I} \models \text{KB}$, 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:

Axiom α	Condition for $\mathcal{I} \models \alpha$
$R_1 \circ \dots \circ R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
$\text{Tra}(R)$	if $R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
$\text{Ref}(R)$	$\langle x, x \rangle \in R^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$
$\text{Irr}(S)$	$\langle x, x \rangle \notin S^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$
$\text{Dis}(S, T)$	if $\langle x, y \rangle \in S^{\mathcal{I}}$ then $\langle x, y \rangle \notin T^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
$\text{Sym}(R)$	if $\langle x, y \rangle \in R^{\mathcal{I}}$ then $\langle y, x \rangle \in R^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
$\text{Asy}(S)$	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 on the right-hand side 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 2.7. 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 $\text{KB} \models \text{KB}'$, 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 subsumption*: Given concepts C, D , does $\text{KB} \models C \sqsubseteq D$ hold?
- Instance checking*: Given a concept C and individual name a , does $\text{KB} \models C(a)$ hold?
- Concept unsatisfiability*: Given a concept C , is there no model $\mathcal{I} \models \text{KB}$ such that $C^{\mathcal{I}} \neq \emptyset$?

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 $A \sqcap C \sqsubseteq \perp$.*

PROOF. We find that KB is inconsistent if the concept \top is unsatisfiable. C is unsatisfiable in KB if $\text{KB} \models C \sqsubseteq \perp$. Given a fresh individual name a , we obtain $\text{KB} \models C \sqsubseteq D$ if $\text{KB} \cup \{C(a)\} \models D(a)$. For a fresh concept name A , $\text{KB} \models C(a)$

if $\text{KB} \cup \{A(a), A \sqcap C \sqsubseteq \perp\}$ is inconsistent. This cyclic reduction shows that all reasoning problems can be reduced to one another. \square

2.3 Simplifications and Normal Forms

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

Every $\text{SROIQ}^{\text{free}}$ 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:

- Whenever a DL features counting quantifiers, we use $\geq 1 R.C$ instead of $\exists R.C$, and $\leq 0 R.\neg C$ 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 .

These conventions reduce the amount of cases that need to be considered in definitions.

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 $\leq n R.D$ 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 2.9. A $\text{SROIQ}^{\text{free}}$ concept expression C is in *positive negation normal form* (pNNF) if

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

C	$\text{pNNF}(C)$
$\top, \perp, A, \neg A, \{a\}, \neg\{a\}, \exists R.\text{Self}, \neg\exists R.\text{Self}$	C
$\neg\top$	\perp
$\neg\perp$	\top
$\neg\neg D$	$\text{pNNF}(D)$
$D_1 \sqcap D_2$	$\text{pNNF}(D_1) \sqcap \text{pNNF}(D_2)$
$D_1 \sqcup D_2$	$\text{pNNF}(D_1) \sqcup \text{pNNF}(D_2)$
$\neg(D_1 \sqcap D_2)$	$\text{pNNF}(\neg D_1) \sqcup \text{pNNF}(\neg D_2)$
$\neg(D_1 \sqcup D_2)$	$\text{pNNF}(\neg D_1) \sqcap \text{pNNF}(\neg D_2)$
$\leq_n R.D$	$\leq_n R.\neg\text{pNNF}(\neg D)$
$\neg\leq_n R.D$	$\geq_{(n+1)} R.\text{pNNF}(\neg D)$
$\geq_n R.D$	$\geq_n R.\text{pNNF}(D)$
$\neg\geq_0 R.D$	\perp
$\neg\geq_n R.D \quad (n > 1)$	$\leq_{(n-1)} R.\neg\text{pNNF}(\neg D)$

Fig. 4. Negation normal form transformations for DL concept expressions. A a concept name, a an individual name, R a role name, $D_{(i)}$ concept expressions.

—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 $\neg\exists R.\text{Self}$.

Every concept expression C can be transformed into a semantically equivalent concept expression $\text{pNNF}(C)$ 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.

Role expressions and RBox axioms also allow for a number of simplifications. $\text{Sym}(R)$ and $\text{Tra}(R)$ are equivalent to $R^- \sqsubseteq R$ and $R \circ R \sqsubseteq R$, respectively. $\text{Ref}(R)$ is equivalent to $\top \sqsubseteq \exists R.\text{Self}$ but the latter is not admissible in \mathcal{SROIQ} if R is not simple. As an alternative, $\text{Ref}(R)$ can be expressed by $\{\top \sqsubseteq \exists S.\text{Self}, S \sqsubseteq R\}$ where S is a fresh simple role name. Irreflexivity $\text{Irr}(S)$ and asymmetry $\text{Asy}(S)$ are again equivalently expressed by $\exists S.\text{Self} \sqsubseteq \perp$ and $\text{Dis}(S, \text{Inv}(S))$, respectively. In summary, $\text{Dis}(S, T)$ 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 $C(a)$, $R(a, b)$, $a \approx b$ 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 $\mathcal{SROIQ}^{\text{free}}$ that 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 \mathcal{S} is a set of $\mathcal{SROIQ}^{\text{free}}$ axioms which are*

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

$$\begin{array}{l} \mathbf{C}_1 ::= \mathbf{C}_0 \mid \mathbf{A} \mid \{\mathbf{I}\} \mid \exists \mathbf{R}.\text{Self} \mid \leq 0 \mathbf{R}.\neg \mathbf{C}_1 \mid \leq 1 \mathbf{R}.\neg \mathbf{C}_0 \mid \geq n \mathbf{R}.\mathbf{C}_1 \mid \mathbf{C}_1 \sqcap \mathbf{C}_1 \mid \mathbf{C}_1 \sqcup \mathbf{C}_0 \\ \mathbf{C}_0 ::= \top \mid \perp \mid \neg \mathbf{A} \mid \neg \{\mathbf{I}\} \mid \neg \exists \mathbf{R}.\text{Self} \mid \leq 0 \mathbf{R}.\neg \mathbf{C}_0 \mid \mathbf{C}_0 \sqcap \mathbf{C}_0 \mid \mathbf{C}_0 \sqcup \mathbf{C}_0 \end{array}$$

Fig. 5. Horn- $SR\mathcal{O}IQ^{\text{free}}$ concept expressions in positive negation normal form

$$\begin{array}{l} \overline{C|_\epsilon} = C \quad \text{pol}(C, \epsilon) = 1 \\ \overline{(\neg C)|_{1p}} = C|_p \quad \text{pol}(\neg C, 1p) = -\text{pol}(C, p) \\ \overline{(C_1 \sqcap C_2)|_{ip}} = C_i|_p \quad \text{pol}(C_1 \sqcap C_2, ip) = \text{pol}(C_i, p) \quad \text{for } \square \in \{\sqcap, \sqcup\}, i \in \{1, 2\} \\ \overline{\leq n R.C|_{3p}} = C|_p \quad \text{pol}(\leq n R.C, 3p) = -\text{pol}(C, p) \\ \overline{\geq n R.C|_{3p}} = C|_p \quad \text{pol}(\geq n R.C, 3p) = \text{pol}(C, p) \end{array}$$

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

D	$\text{pl}^+(D)$	$\text{pl}^-(D)$
\perp	0	0
\top	0	0
A	1	0
$\neg C$	$\text{pl}^-(C)$	$\text{pl}^+(C)$
$\sqcap C_i$	$\max_i \text{sgn}(\text{pl}^+(C_i))$	$\sum_i \text{sgn}(\text{pl}^-(C_i))$
$\sqcup C_i$	$\sum_i \text{sgn}(\text{pl}^+(C_i))$	$\max_i \text{sgn}(\text{pl}^-(C_i))$
$\geq n R.C$	1	$\frac{n(n-1)}{2} + n \text{sgn}(\text{pl}^-(C))$
$\leq n R.C$	$\frac{n(n+1)}{2} + (n+1) \text{sgn}(\text{pl}^-(C))$	1

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

- GCI s $C \sqsubseteq D$ over \mathcal{S} such that $\text{pNNF}(\neg C \sqcup D)$ is a \mathbf{C}_1 concept as defined in Fig. 5, or
- $ABox$ axioms $C(a)$ where the $\text{pNNF}(C)$ is a \mathbf{C}_1 concept as defined in Fig. 5.

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- $SR\mathcal{O}IQ^{\text{free}}$ below.

The original definition of Horn- $SHIQ$ 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 order to 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* $\text{pol}(C, p)$ 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 3.2. Let pl^+ and pl^- denote mutually recursive functions that map a $SHIQ$ concept D to a non-negative integer as specified in Fig. 7 where $\text{sgn}(0) = 0$

and $\text{sgn}(n) = 1$ 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:

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

A concept C is *Horn* if $\text{pl}(C, p) \leq 1$ 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.

The corresponding Definition 1 in [Hustadt et al. 2005] refers to $\mathcal{ALCHI}Q^1$ instead of \mathcal{SHIQ} since an 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 3.2 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 3.2 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 3.2, we first observe that Hornness is not affected by transformation to positive negation normal form.

LEMMA 3.3. *A \mathcal{SHIQ} concept C is Horn according to Definition 3.2 iff its positive negation normal form $\text{pNNF}(C)$ is Horn according to this definition.*

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^+ . 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. The claim clearly holds if C is a concept name, \top , or \perp . Consider the case that $C = \neg(D_1 \sqcap D_2)$. Then $\text{pl}^+(C) = \text{sgn}(\text{pl}^-(D_1)) + \text{sgn}(\text{pl}^-(D_2)) = \text{sgn}(\text{pl}^+(\neg D_1)) + \text{sgn}(\text{pl}^+(\neg D_2))$. By the induction hypothesis this equals $\text{sgn}(\text{pl}^+(\text{pNNF}(\neg D_1))) + \text{sgn}(\text{pl}^+(\text{pNNF}(\neg D_2))) = \text{pl}^+(\text{pNNF}(\neg(D_1 \sqcap D_2)))$, as required. The other cases of the induction are similar. \square

PROPOSITION 3.4. *A \mathcal{SHIQ} concept C is Horn according to Definition 3.2 iff it is Horn according to Definition 3.1.*

PROOF. “ \Leftarrow ” We need to show that $\text{pNNF}(D) \in \mathbf{C}_1$ ($\text{pNNF}(D) \in \mathbf{C}_0$) implies $\text{pl}^+(D) \leq 1$ ($\text{pl}^+(D) = 0$). 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 3.3, it suffices to show that $D \in \mathbf{C}_1$ ($D \in \mathbf{C}_0$) implies $\text{pl}^+(D) \leq 1$ ($\text{pl}^+(D) = 0$). This can be established with some easy inductions over the structure of \mathbf{C}_0 and \mathbf{C}_1 , where all cases follow

¹ $\mathcal{ALCHI}Q$ is \mathcal{SHIQ} without transitivity declarations for roles.

by straightforward calculation of pl^+ , applying the induction hypothesis to obtain results for subexpressions.

“ \Rightarrow ” By Lemma 3.3, we can again restrict attention to concepts in positive negation normal form. We first show that, whenever D in pNNF is such that $\text{pl}^+(D) = 0$, we find that $D \in \mathbf{C}_0$. The contrapositive – if $D \notin \mathbf{C}_0$ then $\text{pl}^+(D) \neq 0$ – can be shown by induction over the structure of D . The result is immediate for $D \in \mathbf{A}$, and follows by simple calculation in all other cases. As an example, consider $D = \leq n R. \neg D'$. If $n > 0$, then $\text{pl}^+(D) \leq 1$ is immediate. If $n = 0$ then $D' \notin \mathbf{C}_0$ and $\text{pl}^+(D') = \text{sgn}(\text{pl}^+(D'))$, where the later is 1 by the induction hypothesis.

To establish the claim, we can now show that, whenever D in pNNF is such that $\text{pl}^+(D) \leq 1$, we find that $D \in \mathbf{C}_1$. The required induction is similar to the \mathbf{C}_0 case, so we omit the details. \square

The previous result shows that Definition 3.1 is indeed a generalisation of the original definition of Horn-*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-*SR₀IQ^{free}* as such in the sequel, but we will rather consider various fragments of this logic. Recall the following definitions of subboolean description logics from [Baader et al. 2007]:

Definition 3.5. Consider a *SR₀IQ^{free}* concept expression C .

- C is an $\mathcal{FL}\mathcal{E}$ concept if it uses only the constructors \top , \perp , \sqcap , \exists , and \forall .
- C is an \mathcal{FL}^- concept if it is an $\mathcal{FL}\mathcal{E}$ concept and all of its existential role restrictions have the form $\exists R. \top$.
- C is an \mathcal{FL}_0 concept if it is an \mathcal{FL}^- concept that does not contain existential role restrictions.

The description logics $\mathcal{FL}\mathcal{E}$, \mathcal{FL}^- , and \mathcal{FL}_0 allow for arbitrary GCIs and concept assertions that contain only concept expressions of the respective type. RBox axioms are not supported.

When defining the Horn variant of each of these description logics, it is relevant whether GCIs or globally valid concept expressions are considered when applying the syntactic restrictions. For example, the GCI $A \sqcap B \sqsubseteq C$ is in \mathcal{FL}_0 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 Horn-*SHIQ* – but then concepts such as $\forall R. \neg A \sqcup \forall S. B$ would be expressible, whereas the corresponding GCI $\exists R. A \sqsubseteq \forall S. B$ cannot be expressed in \mathcal{FL}_0 . Therefore, we apply restrictions on the level of GCIs and do not include concept unions, thus ensuring that all Horn- \mathcal{FL}_0 knowledge bases are also expressible in \mathcal{FL}_0 . Note that the normal form transformations that were used in Definition 3.1 are not affected by such considerations, since Horn restrictions are invariant under negation normal form transformations as illustrated in Lemma 3.3.

Definition 3.6. The description logic Horn- $\mathcal{FL}\mathcal{E}$ (Horn- \mathcal{FL}^- , Horn- \mathcal{FL}_0) allows for the following axioms:

- GCIs $C \sqsubseteq D$ such that the concepts C, D are in $\mathcal{FL}\mathcal{E}$ (\mathcal{FL}^- , \mathcal{FL}_0) and we find that $\text{pNNF}(\neg C \sqcup D) \in \mathbf{C}_1$, or

—concept assertions $C(a)$ such that the concept C is in $\mathcal{FL}\mathcal{E}$ (\mathcal{FL}^- , \mathcal{FL}_0) and $\rho\text{NNF}(C) \in \mathbf{C}_1$,

where \mathbf{C}_1 is defined as in Fig. 5.

These basic Horn DLs form the basis of our subsequent investigations, and it will turn out that they have very different computational properties in spite of the rather similar syntax. We will also extend the previously defined Horn DLs to include further features of Horn- $\mathcal{SROIQ}^{\text{free}}$ that are not included yet. For example, we will consider the logic Horn- $\mathcal{FL}\mathcal{OH}^-$ that extends Horn- \mathcal{FL}^- with nominals and role hierarchies.

4. A LIGHT-WEIGHT HORN DL: HORN- \mathcal{FL}_0

The description logic \mathcal{FL}_0 is indeed very simple: \top , \perp , \sqcap , and \forall are the only operators allowed. Yet, checking the satisfiability of \mathcal{FL}_0 knowledge bases is already EXPTIME-complete [Baader et al. 2005]. It is not hard to see that Horn- \mathcal{FL}_0 is in P, and thus is much simpler than its non-Horn counterpart.

PROPOSITION 4.1. *The standard reasoning tasks for Horn- \mathcal{FL}_0 are P-complete.*

PROOF. An axiom of Horn- \mathcal{FL}_0 is in normal form if it is of one of the following forms: $A \sqsubseteq C$, $A \sqcap B \sqsubseteq C$, $A \sqsubseteq \perp$, $\top \sqsubseteq C$, $A \sqsubseteq \forall R.C$, $C(a)$, $R(a, b)$, where A, B, C are concept names, R is a role name, and a, b are individual names. Now it is easy to see that every Horn- \mathcal{FL}_0 knowledge base KB is equisatisfiable to a Horn- \mathcal{FL}_0 knowledge base in normal form that can be computed in linear time w.r.t the size of KB. An according normal form transformation is detailed for Horn- $\mathcal{FL}\mathcal{OH}^-$ in Lemma 5.6, and the transformation for Horn- \mathcal{FL}_0 is an easy special case thereof, with the only difference that GCIs $\{a\} \sqsubseteq C$ must be written as $C(a)$ in Horn- \mathcal{FL}_0 .

It is easy to see that every Horn- \mathcal{FL}_0 knowledge base in normal form can be translated to a semantically equivalent Datalog program.² Indeed, this translation is obtained by applying the standard transformation of \mathcal{SROIQ} axioms to first-order logic with equality as can be found, e.g., in [Hitzler et al. 2009, Chapter 5]. For example, the axiom $A \sqsubseteq \forall R.B$ is transformed to $A(x) \wedge R(x, y) \rightarrow B(y)$ (as usual, we omit the ubiquitous universal quantifiers when writing Datalog formulae). Since all of the rules that are obtained by translating normal form axioms have at most three variables, the result follows from the fact that satisfiability checking is P-complete for Datalog programs with a bounded number of variables per rule [Dantsin et al. 2001]. Moreover, we also note that the reductions of standard reasoning problems to satisfiability checking are possible in Horn- \mathcal{FL}_0 as well. \square

This simple result could be established even when extending Horn- \mathcal{FL}_0 with further expressive features. In particular, this is the case for all features of \mathcal{SROIQ} for which the standard first-order translation would lead to Datalog axioms, possibly even with equality. This includes nominals, inverse roles, role chains,³ local reflexivity (Self), and the universal role. Moreover, role conjunctions and concept

²Here, *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.

³RIAs must be normalised to obtain RIAs with at most three roles, e.g., transforming $R_1 \circ R_2 \circ R_3 \sqsubseteq R$ to $R_1 \circ R_2 \sqsubseteq S$ and $S \circ R_3 \sqsubseteq R$, where S is a fresh role name.

products as discussed in [Rudolph et al. 2008a; 2008b] are easily integrated into this setting as well, even without restricting to simple roles. Description logics that can faithfully be expressed in Datalog have been called *Description Logic Programs* (DLP) [Grosz et al. 2003].

Two additional features – disjunction and qualified functionality restrictions of the form $\leq 1 R.C$ – are of interest for us to obtain a Horn DL that is more closely related to the OWL RL profile [Motik et al. 2009]. Considering Definition 3.1 and Fig. 5, we observe that Horn DLs allow for at most one \mathbf{C}_1 concept in each disjunction. Every GCI that is Horn in this sense can therefore be expressed in a form where said \mathbf{C}_1 concept constitutes the right-hand side of the concept inclusion axiom, while all other disjunctions occur on the left-hand side. Such disjunctions on the left-hand side of GCIs, however, can easily be eliminated during normal form transformation since $A \sqcup B \sqsubseteq C$ is equivalent to $\{A \sqsubseteq C, B \sqsubseteq C\}$. Therefore, the addition of Horn-disjunction does not increase the expressiveness of the DL.

Qualified functionality restrictions in turn are only allowed in \mathbf{C}_1 expressions of the form $\leq 1 R.\neg C$ with $C \in \mathbf{C}_0$. Such expressions can be simplified by replacing $\neg C$ with a fresh concept name A while introducing a new axiom $\neg C \sqsubseteq A$ (this is Horn since $C \in \mathbf{C}_0$). In addition, it is easy to see that axioms of the form $B \sqsubseteq \leq 1 R.A$ are translated to Datalog rules $B(x) \wedge R(x, y_1) \wedge A(y_1) \wedge R(x, y_2) \wedge A(y_2) \rightarrow y_1 \approx y_2$, so they can safely be included into an extension of Horn- \mathcal{FL}_0 where equality \approx is axiomatised as usual. Summing up the above discussion, we obtain the following result:

PROPOSITION 4.2. *Let Horn-SROIQ(\sqcap)^{free} be Horn-SROIQ^{free} extended with arbitrary conjunctions of roles, and let \mathcal{RL} denote the fragment of the Horn DL Horn-SROIQ(\sqcap)^{free} comprising all knowledge bases that contain no maximality restrictions for numbers other than 1, no existential restrictions, and no minimality restrictions. The standard reasoning problems for \mathcal{RL} are P-complete.*

PROOF. It has been sketched in the above discussion how to extend the normal form transformation to cover Horn disjunction of concepts and qualified functionality restrictions on the right-hand side of GCIs. A suitable normal form for GCIs is defined by requiring all left-hand sides to be of the forms \top , A or $A \sqcap B$, and all right-hand sides to be of the form \perp , A , $\forall R.A$, or $\leq 1 R.A$, where A and B are concept names, nominals, or expressions $\exists S.\text{Self}$, and where R, S are role names. A normal form of RIAs allows only axioms of the form $R \sqsubseteq T$, $R \circ S \sqsubseteq T$, and $R \sqcap S \sqsubseteq T$ (see [Rudolph et al. 2008b] for an introduction to role conjunctions), where R, S, T are role names, inverses of role names, or the universal role. Clearly, any \mathcal{RL} knowledge base is equisatisfiable to an \mathcal{RL} knowledge base in normal form that can be computed in polynomial time – in fact, since all transformations can be accomplished in a single pass, it is even possible to achieve the normalisation in LOGSPACE.

A polynomial-time inferencing algorithm is obtained by further translating normalised \mathcal{RL} knowledge bases into Datalog programs with a bounded number of variables per rule, as in Proposition 4.1. \square

The reason why the rather exotic description logic \mathcal{RL} is specifically mentioned here is that it includes essentially all features of the OWL 2 RL ontology language

which are not related to datatypes [Motik et al. 2009]. Adding datatypes is no major difficulty but requires extended preliminary discussions that are beyond the scope of this work. The only syntactic feature of OWL RL that \mathcal{RL} is missing are existential quantifiers on the left-hand side of GCIs which do not increase expressiveness but which syntactically extend OWL RL. Horn DLs do not restrict the use of existentials, so introducing them to \mathcal{RL} would require additional constraints that do not fit well into the framework of Horn DLs. In contrast, restrictions on the use of existentials appear naturally when studying DLP [Krötzsch et al. 2010]. This indicates that Horn DLs are based on first-order Horn logic with functions, while DLP refers to the function-free fragment Datalog.

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

Horn- \mathcal{FL}^- is the Horn fragment of \mathcal{ALC} that allows \top , \perp , \sqcap , \sqcup , and unqualified \exists , i.e. concept expressions of the form $\exists R.\top$. Although Horn- \mathcal{FL}^- is only a very small extension of Horn- \mathcal{FL}_0 , we will see that it is PSPACE-complete. Moreover, not all of the extensions that could be added to Horn- \mathcal{FL}_0 can also be added to Horn- \mathcal{FL}^- without further increasing the complexity. The extension of \mathcal{FL}^- that we will consider below is defined as follows.

Definition 5.1. The description logic \mathcal{FLOH}^- is the extension of \mathcal{FL}^- with nominals, and role hierarchies. The logic Horn- \mathcal{FLOH}^- is the restriction of \mathcal{FLOH}^- that contains only GCIs $C \sqsubseteq D$ and concept assertions $E(a)$ such that $\text{pNNF}(\neg C \sqcup D) \in \mathbf{C}_1$ and $\text{pNNF}(E) \in \mathbf{C}_1$.

In the following sections, we show that all logics between Horn- \mathcal{FL}^- and Horn- \mathcal{FLOH}^- are PSPACE-complete.

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 5.2. A *deterministic Turing machine* (TM) \mathcal{M} is a tuple (Q, Σ, Δ, q_0) where

- Q is a finite set of states,
- Σ is a finite *alphabet* that includes a *blank* symbol \square ,
- $\Delta \subseteq (Q \times \Sigma) \times (Q \times \Sigma \times \{l, r\})$ 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) $\alpha = w_l q \sigma$, $\alpha' = w_l \sigma' q' \square$, 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 the successor configurations of α are 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 5.3. 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 5.2, 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 $\text{KB}_{\mathcal{M},w}$ and show that w is accepted by \mathcal{M} iff $\text{KB}_{\mathcal{M},w}$ is unsatisfiable. Intuitively, the elements of an interpretation domain of $\text{KB}_{\mathcal{M},w}$ represent possible configurations of \mathcal{M} , encoded by the following concept names

- A_q for $q \in Q$: the TM is in state q ,
- H_i for $i = 0, \dots, p(|w|) - 1$: the TM is at position i on the storage tape,
- $C_{\sigma,i}$ with $\sigma \in \Sigma$ and $i = 0, \dots, p(|w|) - 1$: 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 :

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

where σ_i denotes the symbol at the i th 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 introduce a new concept F that states that a state is *not* accepting, and to propagate this assumption forwards along the runs to provoke an inconsistency as soon as an accepting configuration is reached. Thus we arrive at the axioms given in Fig. 8.

Next we need to investigate the relationship between elements of an interpretation that satisfies $\text{KB}_{\mathcal{M},w}$ and configurations of \mathcal{M} . Given an interpretation \mathcal{I} of

<p>(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 $\delta = (q, \sigma, q', \sigma', l)$, $i > 0$</p>	
<p>(2) Memory: $H_j \sqcap C_{\sigma,i} \sqsubseteq \forall S. C_{\sigma,i} \quad i \neq j$</p>	
<p>(3) Failure: $F \sqcap A_q \sqsubseteq \perp \quad q \in Q_A$</p>	<p>(4) Propagation of failure: $F \sqsubseteq \forall S. F$</p>

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

Fig. 8. Knowledge base $\text{KB}_{\mathcal{M},w}$ simulating a polynomially space-bounded TM

$\text{KB}_{\mathcal{M},w}$, 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 $j \in \{0, \dots, p(|w|) - 1\}$, $e \in C_{\sigma,j}^{\mathcal{I}}$ whenever

$$j \leq m \text{ and } \sigma = \sigma_m \quad \text{or} \quad j > m \text{ and } \sigma = \square.$$

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 5.4. *Consider some interpretation \mathcal{I} that satisfies $\text{KB}_{\mathcal{M},w}$. 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 . \square

LEMMA 5.5. *A word w is accepted by \mathcal{M} iff $\{I_w(i), F(i)\} \cup \text{KB}_{\mathcal{M},w}$ is unsatisfiable, where i is a new constant symbol.*

PROOF. Let \mathcal{I} be a model of $\{I_w(i), F(i)\} \cup \text{KB}_{\mathcal{M},w}$. \mathcal{I} being a model for $I_w(i)$, $i^{\mathcal{I}}$ clearly represents the initial configuration of \mathcal{M} with input w . By Lemma 5.4, for any further configuration reached by \mathcal{M} during computation, $i^{\mathcal{I}}$ has a (not necessarily direct) $S^{\mathcal{I}}$ successor representing that configuration.

Since \mathcal{I} satisfies $F(i)$ and axiom (4) of Fig. 8, a simple induction argument shows that $F^{\mathcal{I}}$ contains all $S^{\mathcal{I}}$ successors of $i^{\mathcal{I}}$. 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, $\{I_w(i), F(i)\} \cup \text{KB}_{\mathcal{M},w}$ 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 $\{I_w(i), F(i)\} \cup \text{KB}_{\mathcal{M},w}$. To this end, 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

$$\frac{A \sqsubseteq C \quad \top \sqsubseteq C \quad A \sqsubseteq \forall R.C \quad R \sqsubseteq S}{A \sqcap B \sqsubseteq C \quad A \sqsubseteq \perp \quad R(c, d)}$$

Fig. 9. Normal forms for Horn- \mathcal{FLOH}^- with $A, B, C \in \mathbf{B}$ basic concepts (including existential restrictions), R, S role names, and c, d individual names

$$\frac{\begin{array}{l} \hat{C} \sqsubseteq \hat{D} \mapsto \{\hat{C} \sqsubseteq X, X \sqsubseteq \hat{D}\} \\ \perp \sqsubseteq C \mapsto \emptyset \\ C \sqsubseteq \top \mapsto \emptyset \end{array} \quad \begin{array}{l} \hat{C} \sqcap A \sqsubseteq B \mapsto \{\hat{C} \sqsubseteq X, X \sqcap A \sqsubseteq B\} \\ A \sqsubseteq C \sqcap D \mapsto \{A \sqsubseteq C, A \sqsubseteq D\} \\ A \sqsubseteq \forall R.\hat{C} \mapsto \{A \sqsubseteq \forall R.X, X \sqsubseteq \hat{C}\} \end{array}}{\begin{array}{l} A, B \text{ basic concept expressions, } \top, \text{ or } \perp; X \text{ a fresh concept name;} \\ C, D \text{ concept expressions; } \hat{C}, \hat{D} \text{ concept expressions that are not basic} \end{array}}$$

Fig. 10. Normal form transformation for Horn- \mathcal{FLOH}^-

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 $(\alpha, \alpha') \in S^M$ iff there is a transition δ from α to α' . F^M is defined to be the set of all configurations that are reached during the run of \mathcal{M} on w .

It is easy to see that M satisfies the axioms (1), (2), and (3) of Fig. 8. Axiom (4) is satisfied since, by our initial assumption, none of the configurations reached by \mathcal{M} is in an accepting state. \square

Thus checking satisfiability of Horn- \mathcal{FL}^- knowledge bases is PSPACE-hard.

5.2 Containment

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 attention to simple forms of axioms. Afterwards, we present the tableau construction and show its correctness, and demonstrate that it can be executed in polynomial space.

To simplify notation, we define a \mathcal{FLOH}^- concept expression C to be *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 the underlying signature is irrelevant or clear from the context.

LEMMA 5.6. *Every Horn- \mathcal{FLOH}^- knowledge base KB is equisatisfiable to a Horn- \mathcal{FLOH}^- knowledge base that contains only axioms in the normal form given in Fig. 9, and that can be computed in linear time with respect to the size of KB.*

PROOF. ABox axioms $C(a)$ can clearly be expressed as GCIs $\{a\} \sqsubseteq C$. To express arbitrary GCIs, we exhaustively apply the transformation rules in Fig. 10, where each rule application consists in replacing the axiom on the left-hand side by 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 the duplication of A still leads to only a linear increase in size. \square

Next, we are going to present a procedure for checking satisfiability of Horn- \mathcal{FL}^-

knowledge bases. In the following we assume without loss of generality that the DL signature in consideration has at least one individual name.

Definition 5.7. Consider a Horn- \mathcal{FLCH}^- 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

$$\text{cl}(\text{KB}) = \mathbf{B} \cup \{\forall R.C \mid R \in \mathbf{R}, C \in \mathbf{B}\} \cup \{\top, \perp\}.$$

Given a set I of individual names, a set \mathcal{T}_I of ABox expressions is defined as follows:

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

For a set $T \subseteq \mathcal{T}_I$ and individuals $e, f \in I$, we use $T_{e \mapsto f}$ to denote the set

$$\{C(f) \mid C(e) \in T\} \cup \{R(f, g) \mid R(e, g) \in T, g \in I\} \cup \{R(g, f) \mid R(g, e) \in T, g \in I\}.$$

For the special case that $e = f$, we use the abbreviation $T_e := T_{e \mapsto e}$. 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 following conditions hold:

- (1) if $e \in I$, then $\top(e) \in T$ and, if $e \in \mathbf{I}$, $\{e\} \in T$,
- (2) if $A(e) \in \text{KB}$ ($R(e, f) \in \text{KB}$), then $A(e) \in T$ ($R(e, f) \in T$),
- (3) 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 $C \in \text{cl}(\text{KB})$, $R \in \mathbf{R}$, and $g \in I$,
- (4) if $A \sqsubseteq C \in \text{KB}$ and $A(e) \in T$, then $C(e) \in T$,
- (5) if $A \sqcap B \sqsubseteq C \in \text{KB}$, $A(e) \in T$, and $B(e) \in T$, then $C(e) \in T$,
- (6) if $R \sqsubseteq S \in \text{KB}$ and $R(e, f) \in T$, then $S(e, f) \in T$,
- (7) $\exists R.\top(e) \in T$ iff $R(e, f) \in T$ for some $f \in I$,
- (8) if $\forall R.C(e) \in T$, then $C(f) \in T$ for all $f \in I$ with $R(e, f) \in T$,

A *tableau* is said contain a *clash* if it contains a statement of the form $\perp(e)$.

PROPOSITION 5.8. *A Horn- \mathcal{FLCH}^- 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 3 in Definition 5.7, 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 $e^\mathcal{I} = [e]_\sim$, $e^\mathcal{I} \in C^\mathcal{I}$ iff $C(e) \in T$, 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 C , 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 $C \in \text{cl}(\text{KB})$, and R some role name. Again, it is easy to see that this meets the conditions of Definition 5.7. \square

As is evident by the Turing machine construction in the previous section, some Horn- \mathcal{FLCH}^- knowledge bases may require a model to contain an exponential number of individuals, even within single paths of the computation. Detecting clashes in polynomial space thus requires special care. In particular, a standard

- (T1) $T := T \cup \{\top(e)\}$
- (T2) if $e \in \mathbf{I}$ is an individual name, $T := T \cup \{\{e\}(e)\}$
- (T3) for each $A(e) \in \text{KB}$, $T := T \cup \{A(e)\}$
- (T4) for each $R(e, f) \in \text{KB}$, $T := T \cup \{R(e, f)\}$
- (T5) for each $\{f\}(e) \in T$
 - (T5a) for each $C(f) \in T$, $T := T \cup \{C(e)\}$,
 - (T5b) for each $g \in I$ and each $R(f, g) \in T$, $T := T \cup \{R(e, g)\}$; $R(e, g)$ is marked inactive,
 - (T5c) for each $g \in I$ and each $R(g, f) \in T$, $T := T \cup \{R(g, e)\}$; $R(g, e)$ is marked inactive,
 - (T5d) for each $C(f) \in T$, $T := T \cup \{C(f)\}$,
 - (T5e) for each $g \in I$ and each $R(e, g) \in T$, $T := T \cup \{R(f, g)\}$; $R(f, g)$ is marked inactive,
 - (T5f) for each $g \in I$ and each $R(g, e) \in T$, $T := T \cup \{R(g, f)\}$; $R(g, f)$ is marked inactive
- (T6) for each $A \sqsubseteq C \in \text{KB}$, if $A(e) \in T$ then $T := T \cup \{C(e)\}$
- (T7) for each $A \sqcap B \sqsubseteq C \in \text{KB}$, if $A(e) \in T$ and $B(e) \in T$ then $T := T \cup \{C(e)\}$
- (T8) for each $R \sqsubseteq S \in \text{KB}$, do the following:
 - (T8a) for each $f \in I$, if $R(e, f) \in T$ and $R(e, f)$ is not inactive, then $T := T \cup \{S(e, f)\}$,
 - (T8b) if $\exists R.\top(e) \in T$ then $T := T \cup \{\exists S.\top(e)\}$
- (T9) for each $f \in I$ and $R(e, f) \in T$ with $R(e, f)$ not inactive, $T := T \cup \{\exists R.\top(e)\}$
- (T10) 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 $T := T \cup \{C(f)\}$
- (\exists) 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. 11. Constructing tableaux for Horn- \mathcal{FLCH}^- knowledge bases

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 5.9. An algorithm that computes a tableau-like structure $\langle I, T \rangle$ is defined as follows. Initially, we set $I := \mathbf{I}$, and $T := \emptyset$. The algorithm executes the steps:

- (1) Iterate over all individuals $e \in I$. To each such e , apply rules (T1) to (T10) of Fig. 11.
- (2) If T was changed in the previous step, go to (1).
- (3) Apply rule (\exists) of Fig. 11 to all existing elements $e \in I$.
- (4) If T was changed by the previous step, go to (1).
- (5) Halt.

While most rules should be obvious, some require explanations. The rules (T5) 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 (T8a),

(T9), or (T10), 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 (T5).

Rule (T8) is also special. In principle, one could omit (T8b), and use rules (T8a) and (T9) 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 (T8b), the class expressions containing e can be computed without considering any role successor, and rule (T9) is essential only when role expressions have been inferred from ABox statements. In combination with the delayed application of rule (\exists) , this ensures that concepts are indeed inferred by (T8b) rather than by (T8a)+(T9), which will be exploited in the proof of Lemma 5.13 below.

Also note that the algorithm of Definition 5.9 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. It is easy to see that there is a correspondence between the rules of Fig. 11 and the conditions of Definition 5.7, so that the limit structure will indeed meet all the requirements imposed on a tableau. For a given knowledge base KB, we write $\langle \bar{I}_{\text{KB}}, \bar{T}_{\text{KB}} \rangle$ to denote the *canonical tableau* constructed by the above algorithm from KB, where the subscripts are omitted when understood. It is easy to see that, whenever the canonical tableau contains a clash, this must be the case for all possible tableaux.

The algorithm of Definition 5.9 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.⁴ 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. The definition uses notation introduced in Definition 5.7.

DEFINITION 5.10. *Consider a Horn- $\mathcal{FL}\mathcal{O}\mathcal{H}^-$ 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 $T := \{\perp(g)\}$.*

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

- (N1) *Given any $X \in \mathcal{T}_I$, set $T := T \cup \{X\}$. If X is a role statement, decide non-deterministically whether X is marked inactive.*
- (N2) *If there is some individual $e \in I$ and $X \in T$ such that X can be derived from $T \setminus \{X\}$ using one of the rules (T1) to (T10) in Fig. 11, set $T := T \setminus \{X\}$. Rules (T5b), (T5c), (T5e), and (T5f) can only be used if X is marked inactive.*

⁴For 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.

- (N3) If $T_a = \{R(e, a)\}$ for some $e \in I \setminus \{a\}$ such that $\exists R.\top(e) \in T$, set $T := T \setminus T_a$.
(N4) If $T_a = \emptyset$, set $T := (T \cup T_{b \rightarrow a}) \setminus T_b$.
(N5) If $T = \emptyset$, return “unsatisfiable.”

LEMMA 5.11. *The algorithm of Definition 5.10 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 $\text{cl}(\text{KB})$ and thus T . \square

LEMMA 5.12. *If there is a sequence of choices such that the algorithm of Definition 5.10 returns “unsatisfiable” after some finite time, KB is indeed unsatisfiable.*

PROOF. Intuitively, the non-deterministic algorithm applies rules of the algorithm in Definition 5.9 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. For a formal proof, assume that the algorithm terminates within finitely many steps, and, without loss of generality, that each step involves a successful application of one of the rules (N1) to (N5). 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 $e, f \in \bar{I}$, such that $T_{a \rightarrow e, b \rightarrow f}^n \subseteq \bar{T}$. 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 $e, f \in \bar{I}$.

For the induction step, assume that $T_{a \rightarrow e, b \rightarrow f}^n \subseteq \bar{T}$. To show the claim for T^{n+1} , we distinguish by the transformation rule that was applied to obtain T^n from T^{n+1} :

- (N1) Since $T^{n+1} \subset T^n$, we conclude $T_{a \rightarrow e, b \rightarrow f}^{n+1} \subseteq \bar{T}$.
(N2) $T^{n+1} = T^n \cup \{X\}$, where X can be derived from T^n by one of the rules (T1) to (T10). Since those rules have been applied exhaustively in \bar{T} , we find $T_{a \rightarrow e, b \rightarrow f}^{n+1} \subseteq \bar{T}$.
(N3) We find $T_a^n = \emptyset$ and, for some $g \in I \setminus \{a\}$ and $R \in \mathbf{R}$, $T^{n+1} = T^n \cup \{R(g, a)\}$ and $\exists R.\top(g) \in T^n$. Define $g' := f$ if $g = b$, and $g' = g$ otherwise. We conclude that $\exists R.\top(g') \in \bar{T}$ and thus there is some individual $e' \in \bar{I}$ with $R(g', e')$. We conclude that $T_{a \rightarrow e', b \rightarrow f}^{n+1} \subseteq \bar{T}$.
(N4) This rule merely exchanges b with (the unused) a , so we have $T_{a \rightarrow f, b \rightarrow e}^{n+1} \subseteq \bar{T}$.

Applying the above induction to the initial state $\{\perp(g)\}$, we find $\{\perp(g)\}_{a \rightarrow e, b \rightarrow f} \in \bar{T}$. Hence \bar{T} indeed contains a clash and KB is unsatisfiable. \square

LEMMA 5.13. *Whenever KB is unsatisfiable, there is a sequence of choices such that the algorithm of Definition 5.10 returns “unsatisfiable” after some finite time.*

PROOF. We first specify a possible sequence of choices, and then show its correctness. If KB is unsatisfiable, there is some element $e \in \bar{I}$ in the canonical tableau such that $\perp(e) \in \bar{T}$. Pick one such e . We use a' and b' to denote the elements of \bar{I} that are currently simulated by a and b . Initially, we set $a' = b' = \square$ for some element $\square \notin \bar{I}$. Rule (N1) of the algorithm will repeatedly be used to close T under relevant inferences that are \prec -smaller than some statement X . Given $X \in \bar{T}$, we define:

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

This selects all elements in \bar{T} that can be represented using the elements from I with the current representation of a' as a , and b' as b . Throughout the below computation, the following property will be preserved:

$$T_{a \mapsto a', b \mapsto b'} \subseteq \bar{T} \quad (\dagger)$$

Now if $e \in \mathbf{I}$, set $a' := e$. Using the non-deterministic initialisation and rule (N1), the algorithm of Definition 5.10 can now compute $T = \downarrow \{\perp(e)\}$. The algorithm now repeatedly executes steps according to the following choice strategy.

Single Step Choice Strategy. If T_a is non-empty, let X be the \prec -largest element of T_a . Else, let X be the \prec -largest element of T . By property (\dagger) , there is some $X' \in \bar{T}$ with $\{X\}_{a \mapsto a', b \mapsto b'} = \{X'\}$. Applying rule (N1), the algorithm first computes $T := T \cup \downarrow X$ (*). The algorithm non-deterministically guesses the rule of Fig. 11 that was used to infer X' , and proceeds accordingly:

- If X' was inferred by one of the rules (T1), (T2), (T3), (T4), (T6), (T7), (T8a), (T8b), and (T9), 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 X , i.e. which are represented by I with the current choice of a' and b' . Hence the algorithm can apply rule (N2) to reduce X .
- If X' was inferred by one of the rules of (T5), then one of the premises used was of the form $\{f\}(e)$, and thus $f \in \mathbf{I}$. Since inactive roles are not generated by any of the given choices, rules (T5b), (T5c), (T5e), and (T5f) are not relevant. If X' was inferred by rule (T5a) then X can directly be reduced by applying rule (N2). The existence of the premises in T follows again from (*).
If X' was inferred by rules (T5d), then X' is of the form $C(f)$ and thus $T_a = \emptyset$. If the individual e in the premise is in \mathbf{I} , then X again can be reduced by rule (N2). If $e \notin \mathbf{I}$, set $a' = e$ and use rule (N1) to compute $T_a = \{\{f\}(e), C(e)\}$. Apply (N2) to reduce X .
- If X' was inferred by rule (T10), then $X' = C(g)$ 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 $X = C(g)$ and $T_a = \emptyset$. Set $a' = e$ and use rule (N1) to compute $T_a = \{\forall R.C(a), R(a, g)\}$. Use rule (N2) to reduce X .
 - If $g \notin \mathbf{I}$, then $X = C(a)$ and $e \neq a'$. If $e \in \mathbf{I} \cup \{b'\}$, then $\{\forall R.C(e), R(e, a)\} \subseteq T$ by (*). Use rule (N2) to reduce X . If $e \notin \mathbf{I} \cup \{b'\}$, then $b' = \square$ and $T_b = \emptyset$, as we will show below. Set $b' = e$ and use rule (N1) to compute $T_b = \{\forall R.C(b), R(b, g)\}$. Use rule (N2) to reduce X .
We claimed that $b' = \square$ whenever it is not equal to the predecessor e . This is so, since $a' \notin \mathbf{I}$ is ensured by each step of the algorithm, and since elements that are not in \mathbf{I} are involved in active role statements of exactly one predecessor (the one which generated a'). This is easily verified by inspecting the rules that can create role statements.

—If X' was inferred by rule (\exists) , we have $X' = R(e, g)$ for some newly introduced element $g \notin \mathbf{I}$. Thus X is of the form $R(e', a)$, and, since X was selected to be \prec -maximal, $T_a = \{X\}$. Thus we can apply rule (N3) to reduce X . In addition, the algorithm applies rule (4) to copy b to the (now empty) a , and we set $a' := b'$ and $b' := \square$.

With the above choices, the algorithm instantiates elements a on demand, and repeatedly reduces the statements of those elements. The individual rules show that this reduction might require another (predecessor) individual b to be considered, but that no further element is needed. Also note that rule (T8b) is required to ensure that all concept expressions in T_a can be reduced without generating any role successors for a . Hence, it is evident that the above choice strategy ensures that exactly one of the above reductions is applicable in each step.

Finally, we need to show that the algorithm terminates. This claim 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 X is deleted, and possibly replaced by one or more elements that are strictly smaller than X . \square

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 5.14. *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 5.11, 5.12, and 5.13 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 [Papadimitriou 1994]. \square

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

THEOREM 5.15. *The standard reasoning problems for any description logic between Horn- \mathcal{FL}^- and Horn- \mathcal{FLOH}^- are PSpace-complete.*

PROOF. Combine Lemma 5.5 and Theorem 5.14. \square

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

$\mathcal{FL}\mathcal{E}$ further extends \mathcal{FL}^- by allowing arbitrary existential role quantifications, which turns out to raise the complexity of standard reasoning tasks for Horn- $\mathcal{FL}\mathcal{E}$ to EXPTIME, thus establishing EXPTIME-completeness of Horn- \mathcal{SHIQ} . Note that inclusion in EXPTIME is obvious since $\mathcal{FL}\mathcal{E}$ is a fragment of \mathcal{SHIQ} which is also in EXPTIME [Tobies 2001]. To show that Horn- $\mathcal{FL}\mathcal{E}$ 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 6.1. 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* \square ,
- $\Delta \subseteq (Q \times \Sigma) \times (Q \times \Sigma \times \{l, r\})$ 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) $\alpha = w_l q \sigma$, $\alpha' = w_l \sigma' q' \square$, 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 6.2. 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 6.3. *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-*SHIQ* by polynomially reducing the halting problem of ATMs with a polynomially bounded storage space to inferencing in Horn-*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- $\mathcal{FL}\mathcal{E}$

In the following, we consider a fixed ATM \mathcal{M} denoted as in Definition 6.1, and a polynomial p that defines a bound for the required space. For any word $w \in \Sigma^*$, we construct a Horn- $\mathcal{FL}\mathcal{E}$ knowledge base $\text{KB}_{\mathcal{M},w}$ and show that acceptance of w by the ATM \mathcal{M} can be decided by inferencing over this knowledge base.

In detail, $\text{KB}_{\mathcal{M},w}$ depends on \mathcal{M} and $p(|w|)$, and has an empty ABox.⁵ 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 $\text{KB}_{\mathcal{M},w}$ 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 $i = 0, \dots, p(|w|) - 1$: the ATM is at position i on the storage tape,
- $C_{\sigma,i}$ with $\sigma \in \Sigma$ and $i = 0, \dots, p(|w|) - 1$: 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 6.1. In this way, acceptance is 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. We therefore may focus on the task of propagating properties to successor configurations, while not taking care of disallowing additional statements to hold. Our encoding ensures that, whenever the initial configuration is not accepting, there is at least one “minimal” model that reflects this.

⁵The RBox is empty for $\mathcal{FL}\mathcal{E}$ anyway.

<p>(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 $\delta = (q, \sigma, q', \sigma', r)$, $i < p(w) - 1$ $A_q \sqcap H_i \sqcap C_{\sigma,i} \sqsubseteq \exists S_\delta.(A_{q'} \sqcap H_{i-1} \sqcap C_{\sigma',i})$ with $\delta = (q, \sigma, q', \sigma', l)$, $i > 0$</p>	<p>(3) Existential acceptance: $A_q \sqcap \exists S_\delta.A \sqsubseteq A$ for all $q \in E$</p>
<p>(2) Memory: $H_j \sqcap C_{\sigma,i} \sqsubseteq \forall S_\delta.C_{\sigma,i}$ $i \neq j$</p>	<p>(4) Universal acceptance: $A_q \sqcap H_i \sqcap C_{\sigma,i} \sqcap \prod_{\delta \in \tilde{\Delta}} (\exists S_\delta.A) \sqsubseteq A$</p>

$q \in U, x \in \{r \mid i < p(|w|) - 1\} \cup \{l \mid i > 0\}$
 $\tilde{\Delta} = \{(q, \sigma, q', \sigma', x) \in \Delta\}$

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

Fig. 12. Knowledge base $\text{KB}_{\mathcal{M},w}$ simulating a polynomially space-bounded ATM

After this informal introduction, consider the knowledge base $\text{KB}_{\mathcal{M},w}$ given in Fig. 12. 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 expression I_w :

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

where σ_i denotes the symbol at the i th 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 $\text{KB}_{\mathcal{M},w}$. The following is obvious from the characterisation given in Definition 3.1.

LEMMA 6.4. $\text{KB}_{\mathcal{M},w}$ and $I_w \sqsubseteq A$ are in Horn- $\mathcal{FL}\mathcal{E}$.

Next we need to investigate the relationship between elements of an interpretation that satisfies $\text{KB}_{\mathcal{M},w}$ and configurations of \mathcal{M} . Given an interpretation \mathcal{I} of $\text{KB}_{\mathcal{M},w}$, 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 $j \in \{0, \dots, p(|w|) - 1\}$, $e \in C_{\sigma_j}^{\mathcal{I}}$ whenever

$$j \leq m \text{ and } \sigma = \sigma_m \quad \text{or} \quad j > m \text{ and } \sigma = \square.$$

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 6.5. Consider some interpretation \mathcal{I} that satisfies $\text{KB}_{\mathcal{M},w}$. 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 . \square

LEMMA 6.6. A word w is accepted by \mathcal{M} iff $I_w \sqsubseteq A$ is a consequence of $\text{KB}_{\mathcal{M},w}$.

PROOF. Consider an arbitrary interpretation \mathcal{I} that satisfies $\text{KB}_{\mathcal{M},w}$. 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 6.5, 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 6.5, 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 $\text{KB}_{\mathcal{M},w}$ 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. 12, it is easy to see that M satisfies $\text{KB}_{\mathcal{M},w}$. Now if the initial configuration is not accepting, $I_w^M \not\subseteq A^M$ by construction. Thus M is a counterexample for $I_w \subseteq A$ which thus is not a logical consequence. \square

We can summarise our results as follows.

THEOREM 6.7. *The standard reasoning problems for any description logic between Horn- $\mathcal{FL}\mathcal{E}$ 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 6.6 shows that the word problem for polynomially space-bounded ATMs can be reduced to checking concept subsumption in $\text{KB}_{\mathcal{M},w}$. By Lemma 6.4, $\text{KB}_{\mathcal{M},w}$ is in Horn- $\mathcal{FL}\mathcal{E}$. The reduction is polynomially bounded due to the restricted number of axioms: there are at most $2 \times |Q| \times p(|w|) \times |\Sigma| \times |\Delta|$ axioms of type (1), $p(|w|)^2 \times |\Sigma| \times |\Delta|$ of type (2), $|Q| \times |\Sigma|$ of type (3), and $|Q| \times p(|w|) \times |\Sigma|$ of type (4). \square

Note that, even in Horn logics, it is straightforward to reduce knowledge base satisfiability to the entailment of the concept subsumption $\top \sqsubseteq \perp$. 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- $\mathcal{FL}\mathcal{E}$.

THEOREM 6.8. *Consider the description logics*

- (a) \mathcal{ELF} obtained by extending \mathcal{EL} with number restrictions of the form $\leq 1 R.\top$,
 (b) $\mathcal{FL}\circ^-$ obtained by extending \mathcal{FL}^- with composition of roles while restricting to regular *RBoxes*, and
 (c) \mathcal{FLI}^- obtained by extending \mathcal{FL}^- with inverse roles,
 and let *Horn- \mathcal{ELF}* , *Horn- $\mathcal{FL}\circ^-$* , and *Horn- \mathcal{FLI}^-* denote the respective Horn DLs defined as in Definition 3.6.

Horn- $\mathcal{FL}\circ^-$ is EXPTIME-hard. *Horn- \mathcal{ELF}* and *Horn- \mathcal{FLI}^-* are EXPTIME-complete.

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

- (a) Replace axioms (2) in Fig. 12 with the following statements:

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

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

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

Any occurrence of concept A is replaced by $\exists R_A.\top$, with R_A a new role. Moreover, we introduce roles $R_{A\delta}$ for each transition δ , and replace any occurrence of $\exists S_\delta.A$ with $\exists R_{A\delta}.\top$. Finally, the following axioms are added:

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

- (c) Axioms (1) are replaced as in (b). Any occurrence of $\exists S_\delta.A$ is now 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. Inclusion results for *Horn- \mathcal{ELF}* and *Horn- \mathcal{FLI}^-* are immediate from their inclusion in *SHIQ*. \square

EXPTIME-completeness of \mathcal{ELF} was shown in [Baader et al. 2005] (where it was called $\mathcal{EL}^{\leq 1}$), but the above theorem sharpens this result to the Horn case, and provides a more direct proof. Theorems 6.7 and 6.8 thus can be viewed as sharpenings of the hardness results on extensions of \mathcal{EL} .

7. RELATED WORK

Horn-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-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-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. 2006; Krötzsch et al. 2006; Krötzsch et al. 2007]. While the data complexity of *Horn-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{SHIQ} can be performed in EXPTIME [Eiter et al. 2008], whereas this problem is known to be 2EXPTIME-complete for \mathcal{SHIQ} [Glimm et al. 2008]. 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; 2007]. 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 provided by 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 [Groszof 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]. Notable exceptions occur when considering Horn description logics as discussed herein [Hustadt et al. 2005; Kazakov 2006] or the language ELP [Krötzsch et al. 2008]. However, not all approaches lead to non-disjunctive Datalog when applied to Horn DLs, as illustrated by the reduction in [Rudolph et al. 2008d; 2008c] that requires disjunctions to encode binary decision diagrams.

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].

8. CONCLUSIONS

In this paper, we have generalised the well-known definition of Horn- \mathcal{SHIQ} to Horn- $\mathcal{SROIQ}^{\text{free}}$, and derived 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. 13.

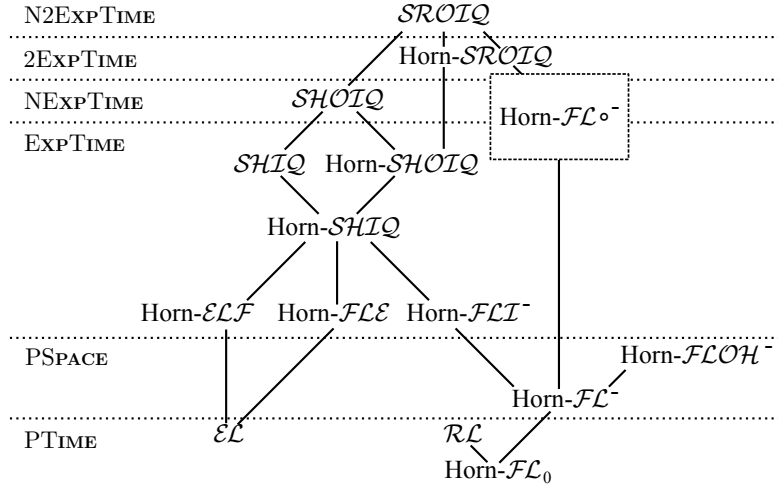


Fig. 13. Reasoning complexities of Horn DLs; the exact position of $\text{Horn-}\mathcal{FL}^{\circ-}$ is not known

Some non-Horn DLs – \mathcal{EL} , \mathcal{RL} , SHIQ , SHOIQ , and 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-SHOIQ and Horn-SROIQ do not follow from this work: they have been established only recently [Ortiz et al. 2010].

The entry for $\text{Horn-}\mathcal{FL}^{\circ-}$ in Fig. 13 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 $\text{Horn-}\mathcal{FL}^-$ does no longer admit reasoning in PSPACE .⁶ The 2EXPTIME upper bound for the complexity follows from the according result for Horn-SROIQ [Ortiz et al. 2010]. Further checks are needed to determine the exact complexity of $\text{Horn-}\mathcal{FL}^{\circ-}$. But when considering the fact that no Horn DL is known to be complete for a non-deterministic complexity class, it seems to be extremely 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 $\text{Horn-}\mathcal{FL}^-$ and $\text{Horn-}\mathcal{FLOH}^-$ 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 therefore consists in showing that such an extremely limited view suffices for complete reasoning in the considered logics. As opposed to $\text{Horn-}\mathcal{FL}_0$, the addition of nominals to $\text{Horn-}\mathcal{FL}^-$ significantly complicates reasoning procedures, although it does not lead to increased worst-case complexities. Due to a high amount of unguided non-determinism, the tableau algorithm for $\text{Horn-}\mathcal{FLOH}^-$ 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

⁶Unless $\text{PSPACE} = \text{EXPTIME}$.

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.

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., GOTTLÖB, 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.
- EITER, T., GOTTLÖB, G., ORTIZ, M., AND SIMKUS, M. 2008. Query answering in the description logic Horn-*SHIQ*. See Hölldobler et al. [2008], 166–179.
- GLIMM, B., LUTZ, C., HORROCKS, I., AND SATTLER, U. 2008. Answering conjunctive queries in the *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.
- HÖLDOBLER, S., LUTZ, C., AND WANSING, H., Eds. 2008. *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA'08)*. LNAI, vol. 5293. Springer.
- HUSTADT, U., MOTIK, B., AND SATTLER, U. 2005. Data complexity of reasoning in very expressive description logics. See Kaelbling and Saffiotti [2005], 466–471.
- KAEHLING, 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. *RIQ* and *SROIQ* are harder than *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.
- ACM Transactions on Computational Logic, Vol. V, No. N, August 2011.

- KAZAKOV, Y. 2009. Consequence-driven reasoning for horn *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. 2006. 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. 2006. 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 International Conference on Web Reasoning and Rule Systems (RR'10)*. LNCS, vol. 6333. Springer, 88–102.
- 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 Intelligence, and Reasoning (LPAR'01)*, M. Hermann and A. Voronkov, Eds. LNCS, vol. 4246. Springer, 227–241.
- MOTIK, B., SHEARER, R., AND HORROCKS, I. 2007. Optimized reasoning in description logics using hypertableaux. In *Proceedings of the 21st Conference on Automated Deduction (CADE'07)*, F. Pfenning, Ed. LNAI, vol. 4603. Springer, 67–83.
- 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 SIMKUS, 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.
- 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/>.
- PAPADIMITRIOU, C. H. 1994. *Computational Complexity*. Addison Wesley.
- 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. See Hölldobler et al. [2008], 362–374.
- RUDOLPH, S., KRÖTZSCH, M., AND HITZLER, P. 2008c. Description logic reasoning with decision diagrams: Compiling *SHIQ* to disjunctive datalog. See Sheth et al. [2008], 435–450.
- RUDOLPH, S., KRÖTZSCH, M., AND HITZLER, P. 2008d. Terminological reasoning in *SHIQ* with ordered binary decision diagrams. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI'08)*. AAAI Press, 529–534.

- 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.