Data Complexity of Reasoning in Very Expressive Description Logics

Ullrich Hustadt
Dept. of Computer Science
Univ. of Liverpool, UK
U.Hustadt@csc.liv.ac.uk

Boris Motik
Forschungszentrum Informatik
Univ. of Karlsruhe, Germany
motik@fzi.de

Ulrike Sattler
Dept. of Computer Science
Univ. of Manchester, UK
sattler@cs.man.ac.uk

Abstract

Data complexity of reasoning in description logics (DLs) estimates the performance of reasoning algorithms measured in the size of the ABox only. We show that, even for the very expressive DL $\mathcal{SHIQ}$, satisfiability checking is data complete for NP. For applications with large ABoxes, this can be a more accurate estimate than the usually considered *combined complexity*, which is EXPTIME-complete. Furthermore, we identify an expressive fragment, Horn-$\mathcal{SHIQ}$, which is data complete for P, thus being very appealing for practical usage.

1 Introdcution

Description logics (DLs) [Baader *et al.*, 2003] are state-of-the-art knowledge representation formalisms with applications in many areas of computer science. Very expressive DLs such as $\mathcal{SHIQ}$ are interesting mainly due to their high expressivity combined with the clearly defined model-theoretic semantics and known formal properties, such as the computational complexity of reasoning. In particular, the *combined complexity* of checking satisfiability of a $\mathcal{SHIQ}$ knowledge base KB is EXPTIME-complete in $|KB|$ [Schild, 1991; Tobies, 2001].

EXPTIME-completeness is a rather discouraging result since $|KB|$ can be large in practice. However, similar to a database, KB consists of a *schema* part $\mathcal{T}$, called the *TBox*, and a data or fact part $\mathcal{A}$, called the *ABox*. For applications with a fixed schema but varying data part, *data complexity*, measured in the size of $\mathcal{A}$ only, provides a more precise performance estimate.

Here, we assume that the ABox of KB is *extensionally reduced*, i.e. it involves only roles and (possibly negated) atomic concepts. Thus, the terminological knowledge is strictly separated from assertional knowledge, so $|\mathcal{A}|$ is the measure of "raw" data. For such a KB, we show that checking KB satisfiability is NP-complete in $|\mathcal{A}|$, and that instance checking is co-NP-complete in $|\mathcal{A}|$. Since this might still lead to intractability, we identify Horn-$\mathcal{SHIQ}$, a fragment of $\mathcal{SHIQ}$ analogous to the Horn fragment of first-order logic. Namely, Horn-$\mathcal{SHIQ}$ provides existential and universal quantifiers, but does not provide means to express disjunctive information. We show that, for Horn-$\mathcal{SHIQ}$, the basic reasoning problems are P-complete in $|\mathcal{A}|$. To develop an intuition and to provide a more detailed account of these results, we compare them with similar results for (variants of) datalog [Dantsin *et al.*, 2001].

Our results are important since they provide a formal justification for hoping to provide tractable algorithms for very expressive description logics. Furthermore, Horn-$\mathcal{SHIQ}$ subsumes DL-lite [Calvanese *et al.*, 2004], a logic aiming to capture most constructs of ER and UML formalisms, while providing polynomial algorithms for satisfiability checking and conjunctive query answering (assuming an unbounded knowledge base, but a bound on the query size). Horn-$\mathcal{SHIQ}$ additionally allows qualified existential quantification, conditional functionality and role inclusions, while allowing for a reasoning algorithm that runs in time polynomial in the size of data.

In [Schaerf, 1994], the complexity of concept subsumption was contrasted with combined and data complexity of instance checking for various fragments of $\mathcal{ALC}$. This work provides a lower bound for the data complexity of reasoning in $\mathcal{SHIQ}$.

2 Preliminaries

Description Logics. Given a set of role names N_R, a $\mathcal{SHIQ}$ *role* is either some $R \in N_R$ or an *inverse role* R^- for $R \in N_R$. A $\mathcal{SHIQ}$ *RBox* $\mathcal{R}$ is a finite set of role inclusion axioms $R \sqsubseteq S$ and transitivity axioms $\mathsf{Trans}(R)$, for R and S $\mathcal{SHIQ}$ roles. For $R \in N_R$, we set $\mathsf{Inv}(R) = R^-$ and $\mathsf{Inv}(R^-) = R$, and assume that $R \sqsubseteq S \in \mathcal{R}$ ($\mathsf{Trans}(R) \in \mathcal{R}$) implies $\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) \in \mathcal{R}$ ($\mathsf{Trans}(\mathsf{Inv}(R)) \in \mathcal{R}$). A role R is said to be *simple* if $\mathsf{Trans}(S) \notin \mathcal{R}$, for each $S \sqsubseteq^* R$, where $\sqsubseteq^*$ is the reflexive-transitive closure of $\sqsubseteq$.

Given a set of *concept names* N_C, the set of $\mathcal{SHIQ}$ concepts is the minimal set such that each $A \in N_C$ is a $\mathcal{SHIQ}$ concept and, for C and D $\mathcal{SHIQ}$ concepts, R a role, S a simple role and n a positive integer, then $\neg C$, $C \sqcap D$, $\forall R.C$ and $\geq n\, S.C$ are also $\mathcal{SHIQ}$ concepts. We use $\top$, $\bot$, $C_1 \sqcup C_2$, $\exists R.C$, and $\leq n\, S.C$ as abbreviations for $A \sqcup \neg A$, $A \sqcap \neg A$, $\neg(\neg C_1 \sqcap \neg C_2)$, $\neg\forall R.\neg C$, and $\neg(\geq (n+1)\, S.C)$, respectively. Concepts that are not concept names are called *complex*. A *literal* concept is a possibly negated concept name.

A TBox $\mathcal{T}$ is a finite set of concept inclusion axioms of the form $C \sqsubseteq D$. An ABox $\mathcal{A}$ is a finite set of axioms $C(a)$, $R(a, b)$, and (in)equalities $a \approx b$ and $a \not\approx b$. A knowledge

Table 1: Semantics of $\mathcal{SHIQ}$ by Mapping to FOL

Translating Concepts to FOL
$\pi_y(A, X) = A(X)$
$\pi_y(C \sqcap D, X) = \pi_y(C, X) \wedge \pi_y(D, X)$
$\pi_y(\neg C, X) = \neg\pi_y(C, X)$
$\pi_y(\forall R.C, X) = \forall y : R(X, y) \rightarrow \pi_x(C, y)$
$\pi_y(\geq n\, S.C, X) =$
$\exists y_1, \ldots, y_n : \bigwedge S(X, y_i) \wedge \bigwedge \pi_x(C, y_i) \wedge \bigwedge y_i \not\approx y_j$
Translating Axioms to FOL
$\pi(C \sqsubseteq D) = \forall x : \pi_y(C, x) \rightarrow \pi_y(D, x)$
$\pi(R \sqsubseteq S) = \forall x, y : R(x, y) \rightarrow S(x, y)$
$\pi(\mathsf{Trans}(R)) = \forall x, y, z : R(x, y) \wedge R(y, z) \rightarrow R(x, z)$
$\pi(C(a)) = \pi_y(C, a)$
$\pi(R(a, b)) = R(a, b)$
$\pi(a \circ b) = a \circ b$ for $\circ \in \{\approx, \not\approx\}$
Translating KB to FOL
$\pi(R) = \forall x, y : R(x, y) \leftrightarrow R^-(y, x)$
$\pi(KB) = \bigwedge_{R \in N_R} \pi(R) \wedge \bigwedge_{\alpha \in \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}} \pi(\alpha)$
X is a meta variable and is substituted by the actual variable. π_x is obtained from π_y by simultaneously substituting in the definition $x_{(i)}$ for all $y_{(i)}$, respectively, and π_y for π_x.

base KB is a triple $(\mathcal{R}, \mathcal{T}, \mathcal{A})$. KB is *extensionally reduced* if all ABox axioms of KB contain only literal concepts.

The semantics of KB is given by translating it into first-order logic by the operator π from Table 1. The main inference problem is checking KB satisfiability, i.e. determining if a first-order model of $\pi(KB)$ exists. An individual a is an *instance* of a concept C w.r.t. KB if $\pi(KB) \models \pi_y(C, a)$, which is the case iff $KB \cup \{\neg C(a)\}$ is unsatisfiable.

The logic $\mathcal{ALCHIQ}$ is obtained by disallowing transitivity axioms in $\mathcal{SHIQ}$ RBoxes, and $\mathcal{ALC}$ is obtained by disallowing RBoxes, inverse roles and number restrictions. A logic $\mathcal{L}$ is *between* logics $\mathcal{L}_1$ and $\mathcal{L}_2$ if it contains at least the primitives from $\mathcal{L}_1$ and at most the primitives from $\mathcal{L}_2$.

We measure the *size* of concepts by their length, and assume *unary coding of numbers*, i.e. $|\leq n\, R.C| = n+1+|C|$, and use $|R(a, b)| = |(\neg)A(a)| = 3$.

Disjunctive Datalog. A *datalog term* is a constant or a variable, and a *datalog atom* has the form $A(t_1, \ldots, t_n)$ or $t_1 \approx t_2$, where t_i are datalog terms. A *disjunctive datalog program with equality* P is a finite set of rules of the form $A_1 \vee \ldots \vee A_n \leftarrow B_1, \ldots, B_m$ where A_i and B_j are datalog atoms. Each rule is required to be *safe*, i.e. each variable occurring in the rule must occur in at least one body atom. A *fact* is a rule with $m = 0$. For the semantics, we take a rule to be equivalent to a clause $A_1 \vee \ldots \vee A_n \vee \neg B_1 \vee \ldots \vee \neg B_m$. We consider only Herbrand models, and say that a model M of P is *minimal* if there is no model M' of P such that $M' \subsetneq M$. A ground literal A is a *cautious answer* of P (written $P \models_c A$) if A is true in all minimal models of P. First-order entailment coincides with cautious entailment for positive ground atoms.

Reducing KB to Disjunctive Datalog. The results in this paper are based on our algorithm from [Hustadt *et al.*, 2004b]. For a $\mathcal{SHIQ}$ knowledge base KB, this algorithm computes a positive disjunctive datalog program with equality $\mathsf{DD}(KB)$ which is equisatisfiable with KB.

Table 2: Types of $\mathcal{ALCHIQ}$-clauses

1	$\neg R(x, y) \vee \mathsf{Inv}(R)(y, x)$
2	$\neg R(x, y) \vee S(x, y)$
3	$\mathbf{P^f}(x) \vee R(x, f(x))$
4	$\mathbf{P^f}(x) \vee R(f(x), x)$
5	$\mathbf{P_1}(x) \vee \mathbf{P_2}(\mathbf{f}(x)) \vee \bigvee f_i(x) \approx / \not\approx f_j(x)$
6	$\mathbf{P_1}(x) \vee \mathbf{P_2}(g(x)) \vee \mathbf{P_3}(\mathbf{f}(g(x))) \vee \bigvee t_i \approx / \not\approx t_j$ where t_i and t_j are of the form $f(g(x))$ or x
7	$\mathbf{P_1}(x) \vee \bigvee \neg R(x, y_i) \vee \mathbf{P_2}(\mathbf{y}) \vee \bigvee y_i \approx y_j$
8	$\mathbf{R}(\mathbf{a}, \mathbf{b}) \vee \mathbf{P}(\mathbf{t}) \vee \bigvee t_i \approx / \not\approx t_j$ where $t_{(i)}$ are a constant b or a functional term $f_i(a)$

A minor obstacle in computing $\mathsf{DD}(KB)$ are the transitivity axioms which, in their clausal form, do not contain a literal in which all variables of a clause occur. Such clauses are known to be difficult to handle, so KB is preprocessed into an equisatisfiable $\mathcal{ALCHIQ}$ knowledge base $\Omega(KB)$. Roughly speaking, a transitivity axiom $\mathsf{Trans}(S)$ is replaced with axioms of the form $\forall R.C \sqsubseteq \forall S.(\forall S.C)$, for each R with $S \sqsubseteq^* R$ and C a concept occurring in KB. This transformation is polynomial, so in the rest of this paper w.l.o.g. we assume KB to be an $\mathcal{ALCHIQ}$ knowledge base.

The next step is to translate $\Omega(KB)$ into clausal first-order logic. Assuming π is defined as in Table 1, $\pi(\Omega(KB))$ is transformed into a set of clauses $\Xi(KB)$ using *structural transformation* to avoid an exponential blowup [Nonnengart and Weidenbach, 2001]. Roughly speaking, the structural transformation introduces a new name for each complex subformula of $\pi(\Omega(KB))$. A specialized version of the structural transformation is presented in detail in Section 4.

A core property of $\Xi(KB)$ is that it only contains clauses of one of the forms given in Table 2; such clauses are called $\mathcal{ALCHIQ}$-clauses. For a term t, $\mathbf{P}(t)$ denotes a disjunction of the form $(\neg)P_1(t) \vee \ldots \vee (\neg)P_n(t)$, and $\mathbf{P}(\mathbf{f}(x))$ denotes a disjunction of the form $\mathbf{P_1}(f_1(x)) \vee \ldots \vee \mathbf{P_n}(f_m(x))$ (notice that this allows each $\mathbf{P_i}(f_i(x))$ to contain positive and negative literals).

Next, the RBox and TBox clauses of $\Xi(KB)$ are saturated by basic superposition [Bachmair *et al.*, 1995; Nieuwenhuis and Rubio, 1995]—a clausal equational theorem proving calculus. Due to space limitations, we are unable to present the rules of basic superposition; it can be considered to be an optimized version of the well-known paramodulation calculus. Let Γ be the saturated set of clauses. In this key step, all non-ground consequences of KB are computed. In [Hustadt *et al.*, 2004b], we have shown the following key property: (♠) an application of a basic superposition inference rule to $\mathcal{ALCHIQ}$-clauses produces an $\mathcal{ALCHIQ}$-clause. The proof examines all inference rules and clause types.

Furthermore, by examining the types of clauses from Table 2, one can show the following property: (♣) for a finite KB, the number of $\mathcal{ALCHIQ}$-clauses unique up to variable renaming is exponential in $|KB|$. The proof is a straightforward counting exercise since the number of variables and the depth of functional terms in $\mathcal{ALCHIQ}$-clauses are bounded.

Each inference step can be carried out in polynomial time,

so by (♠) and (♣), after at most exponentially many steps, all $\mathcal{ALCHIQ}$-clauses are derived, and saturation terminates.

Satisfiability of $\Xi(KB)$ can be decided by further saturating $\Gamma \cup \Xi(\mathcal{A})$ by basic superposition. Since Γ contains all non-ground consequences of $\Xi(KB)$, all remaining inferences will produce only ground clauses, and will not involve clauses of type 4 and 6. These inferences can be simulated in a disjunctive datalog program by transforming Γ into a function-free clause set, and by introducing new constants playing the role of ground functional terms, as described next.

We define an operator λ transforming Γ as follows: (*i*) each functional term $f(a)$ is replaced with a new, globally unique constant a_f; (*ii*) each term $f(x)$ is replaced with a new, globally unique variable x_f; (*iii*) for each variable in a clause introduced in step (*ii*), λ appends a literal $\neg S_f(x, x_f)$, where S_f is a new predicate unique for the function symbol f; (*iv*) if some variable x occurs in a positive, but not in a negative literal in a clause, then the literal $\neg HU(x)$ is appended to the clause; (*v*) for each function symbol f and each constant a from $\Xi(KB)$, the facts $S_f(a, a_f)$, $HU(a)$ and $HU(a_f)$ are added. The set of (function-free) clauses obtained by applying λ to $\Gamma \cup \Xi(\mathcal{A})$ is denoted with $\mathsf{FF}(KB)$. An example of applying λ to a clause of type 5 is shown below.

$$\neg C(x) \vee D(f(x)) \Rightarrow^{\lambda} \neg S_f(x, x_f) \vee \neg C(x) \vee D(x_f)$$

Now each remaining ground inference by basic superposition in $\Gamma \cup \Xi(\mathcal{A})$ can be simulated by a sound inference step in $\mathsf{FF}(KB)$, and vice versa [Hustadt *et al.*, 2004b], so KB and $\mathsf{FF}(KB)$ are equisatisfiable. Since $\mathsf{FF}(KB)$ does not contain functional terms and all its clauses are safe, each clause can be rewritten into a positive disjunctive rule; let $\mathsf{DD}(KB)$ be the resulting set of rules. The following theorem summarizes the properties of $\mathsf{DD}(KB)$:

Theorem 1 ([Hustadt *et al.*, 2004b])**.** *For KB an $\mathcal{ALCHIQ}$ knowledge base, the following claims hold: (*i*) KB is unsatisfiable iff $\mathsf{DD}(KB)$ is unsatisfiable; (*ii*) $KB \models \alpha$ iff $\mathsf{DD}(KB) \models_c \alpha$, for α of the form $A(a)$ or $S(a, b)$, A a concept name, and S a simple role; (*iii*) $KB \models C(a)$ iff $\mathsf{DD}(KB \cup \{C \sqsubseteq Q\}) \models_c Q(a)$, for C a complex concept, and Q a new concept name; (*iv*) the number of rules in $\mathsf{DD}(KB)$ is at most exponential in $|KB|$, the number of literals in each rule is at most polynomial in $|KB|$, and $\mathsf{DD}(KB)$ can be computed in time exponential in $|KB|$.*

3 Data Complexity of Reasoning in $\mathcal{SHIQ}$

For an extensionally reduced $\mathcal{SHIQ}$ knowledge base KB, an upper bound for the data complexity follows from the reduction of KB to $\mathsf{DD}(KB)$. Before presenting the details, we first discuss the intuition behind this result.

By Theorem 1, $|\mathsf{DD}(KB)|$ is exponential in $|KB|$. However, a closer inspection of the reduction algorithm reveals that the exponential blowup is caused by the rules obtained by saturating $\mathcal{ALCHIQ}$-clauses of types 1 – 7, which correspond to TBox and RBox, but not to ABox clauses. Hence, the size of the rules of $\mathsf{DD}(KB)$ is exponential in the size of TBox and RBox; however, the size of the facts in $\mathsf{DD}(KB)$ is linear in the size of the ABox. Therefore, data complexity of checking satisfiability of $\mathsf{DD}(KB)$ corresponds to data complexity of checking satisfiability of a positive disjunctive datalog program, and is thus in NP. Intuitively, this is due to nice property of $\mathcal{SHIQ}$ that TBox and RBox reasoning does not "interfere" with ABox reasoning, i.e. all non-ground consequences of KB can be computed without taking the ABox into account. Notice that this result holds even for binary number coding.

Lemma 1 (Membership)**.** *For KB an extensionally reduced $\mathcal{SHIQ}$ knowledge base, satisfiability of KB can be decided in non-deterministic polynomial time in $|\mathcal{A}|$.*

Proof. Let c be the number of constants, f the number of function symbols in the signature of $\Xi(KB)$, and s the number of facts in $\Xi(KB)$ (which is equal to the number of facts in $\mathcal{A}$). By definition of λ, the number of constants in $\mathsf{DD}(KB)$ is bounded by $\ell_1 = c + cf$ (cf accounts for constants of the form a_f), and the number of facts in $\mathsf{DD}(KB)$ is bounded by $\ell_2 = s + c + 2cf$ (c accounts for facts of the form $HU(a)$, one cf accounts for facts of the form $S_f(a, a_f)$, and the other cf accounts for facts of the form $HU(a_f)$). All function symbols are introduced by skolemizing TBox concepts $\exists R.C$ and $\geq n\, R.C$. Since $|\mathcal{T}|$ and $|\mathcal{R}|$ are constant, f is also a constant, so both ℓ_1 and ℓ_2 are linear in $|\mathcal{A}|$.

Hence, $|\mathsf{DD}(KB)|$ can be exponential in $|KB|$ only because the non-ground rules in $\mathsf{DD}(KB)$ are obtained from exponentially many clauses of types 1 – 7. Since these clause types do not contain ABox clauses, the number of clauses obtained after saturation is obviously exponential in $|\mathcal{T}| + |\mathcal{R}|$ only. Since we assume that the latter is constant, both the number of rules in $\mathsf{DD}(KB)$ and their length are bounded by constants, so $|\mathsf{DD}(KB)|$ is polynomial in $|\mathcal{A}|$, and can be computed from KB in time polynomial in $|\mathcal{A}|$.

As KB and $\mathsf{DD}(KB)$ are equisatisfiable, the data complexity of checking satisfiability of KB follows from the data complexity of checking satisfiability of $\mathsf{DD}(KB)$, which is NP-complete [Dantsin *et al.*, 2001]: assuming $\mathsf{DD}(KB)$ contains r rules and at most v variables in a rule, the number of literals in a ground instantiation $\mathsf{ground}(\mathsf{DD}(KB))$ is bounded by $r \cdot \ell_1^v + \ell_2$ (in each rule, each variable can be replaced in ℓ_1 possible ways). Assuming r and v are constants, $\wp = |\mathsf{ground}(\mathsf{DD}(KB))|$ is polynomial in $|\mathcal{A}|$. Satisfiability of $\mathsf{ground}(\mathsf{DD}(KB))$ can be checked by nondeterministically generating an interpretation of size at most $\wp$, and then checking whether it is a model. Both tasks can be performed in polynomial time, so the overall algorithm is obviously non-deterministically polynomial in $|\mathcal{A}|$. □

The hardness of the satisfiability checking problem follows from [Schaerf, 1994, Lemma 4.2.7]. Actually, the lemma shows co-NP-hardness of instance checking, by a reduction of satisfiability of 2-2-CNF propositional formulae. The reduction produces an extensionally reduced ABox and a single TBox axiom, so it applies in our case as well. Hence, we immediately obtain the following result:

Theorem 2. *Let KB be an extensionally reduced knowledge base in any logic between $\mathcal{ALC}$ and $\mathcal{SHIQ}$. Then (*i*) deciding KB satisfiability is data complete for* NP *and (*ii*) deciding whether $KB \models (\neg)C(a)$ with $|C|$ bounded is data complete for co-*NP.

4 A Horn Fragment of $\mathcal{SHIQ}$

Horn logic is a well-known fragment of first-order logic where formulae are restricted to clauses containing at most one positive literal. The main limitation of Horn logic is its inability to represent disjunctive information; however, its main benefit is the existence of practical refutation procedures. Furthermore, data complexity of query answering in Horn logic without function symbols is P-complete [Dantsin *et al.*, 2001], which makes it appealing for practical usage.

Following this idea, in this section we identify a Horn fragment of $\mathcal{SHIQ}$, where disjunction is traded for P-complete data complexity. Roughly speaking, in Horn-$\mathcal{SHIQ}$, only axioms of the form $\bigsqcap C_i \sqsubseteq D$ are allowed, where each C_i has the form A or $\exists R.A$, and D has the form A, $\bot$, $\exists R.A$, $\forall R.A$, $\geq n\, R.A$ or $\leq 1\, R$. Whereas such a definition succinctly demonstrates the expressivity of the fragment, in general it is too restricting: e.g., the axiom $A_1 \sqcup A_2 \sqsubseteq \neg B$ is not Horn, but it is equivalent to Horn axioms $A_1 \sqcap B \sqsubseteq \bot$ and $A_2 \sqcap B \sqsubseteq \bot$. Similarly, a non-Horn axiom $A \sqsubseteq \exists R.(\exists R.B)$ can be transformed into Horn axioms $A \sqsubseteq \exists R.Q$ and $Q \sqsubseteq \exists R.B$ by introducing a new name Q for the subconcept $\exists R.B$. To avoid dependency on such obvious syntactic transformations, we give a rather technical definition of Horn-$\mathcal{SHIQ}$.

We first adapt the notions of positions and polarity in first-order formulae to DL. A *position* p is a finite sequence of integers; the empty position is denoted with ϵ. If a position p_1 is a proper prefix of a position p_2, then and p_1 is *above* p_2, and p_2 is *below* p_1. For a concept α, the subterm at a position p, written $\alpha|_p$, is defined as follows: $\alpha|_\epsilon = \alpha$; $(\neg D)|_{1p} = D|_p$; $(D_1 \circ D_2)|_{ip} = D_i|_p$ for $\circ \in \{\sqcap, \sqcup\}$ and $i \in \{1, 2\}$; $\alpha|_1 = R$ and $\alpha|_{2p} = D|_p$ for $\alpha = \Diamond R.D$ and $\Diamond \in \{\exists, \forall\}$; and $\alpha|_1 = n$, $\alpha|_2 = R$ and $\alpha|_{3p} = D|_p$ for $\alpha = \bowtie n\, R.D$ and $\bowtie \in \{\leq, \geq\}$. A *replacement* of a subterm of α at position p with a term β is defined in the standard way and is denoted as $\alpha[\beta]_p$. For a concept α and a position p such that $\alpha|_p$ is a concept, the *polarity* of $\alpha|_p$ in α, denoted as $\mathsf{pol}(\alpha, p)$, is defined as follows:

$$
\begin{aligned}
\mathsf{pol}(C, \epsilon) &= 1;\\
\mathsf{pol}(C_1 \circ C_2, ip) &= \mathsf{pol}(C_i, p) \text{ for } \circ \in \{\sqcap, \sqcup\},\ i \in \{1, 2\};\\
\mathsf{pol}(\Diamond R.C, 2p) &= \mathsf{pol}(C, p) \text{ for } \Diamond \in \{\exists, \forall\};\\
\mathsf{pol}(\geq n\, R.C, 3p) &= \mathsf{pol}(C, p);\\
\mathsf{pol}(\neg C, 1p) &= -\mathsf{pol}(C, p);\\
\mathsf{pol}(\leq n\, R.C, 3p) &= -\mathsf{pol}(C, p).
\end{aligned}
$$

Intuitively, $\mathsf{pol}(\alpha, p)$ equals 1 if $\alpha|_p$ occurs in α under an even number of explicit and implicit negations, and -1 otherwise.

Definition 1. *In Table 3, we define two mutually recursive functions* pl^+ *and* pl^-*, where* $\mathsf{sgn}(0) = 0$ *and* $\mathsf{sgn}(n) = 1$ *for* $n > 0$*. For a concept* C *and a position* p *of a subconcept in* C*, let* $\mathsf{pl}(C, p) = \mathsf{pl}^+(C|_p)$ *if* $\mathsf{pol}(C, p) = 1$*, and let* $\mathsf{pl}(C, p) = \mathsf{pl}^-(C|_p)$ *if* $\mathsf{pol}(C, p) = -1$*.*

A concept C *is a* Horn *concept if* $\mathsf{pl}(C, p) \leq 1$ *for each position* p *of a subconcept in* C *(including the empty position* ϵ*). An extensionally reduced* $\mathcal{ALCHIQ}$ *knowledge base* KB *is* Horn *if, for each axiom* $C \sqsubseteq D \in KB$*, the concept* $\neg C \sqcup D$ *is Horn. An extensionally reduced* $\mathcal{SHIQ}$ *knowledge base* KB *is* Horn *if* $\Omega(KB)$ *is Horn.*

Table 3: Definitions of pl^+ and pl^-

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

It is easy to see that, for a concept C without complex subconcepts, $\mathsf{pl}^+(C)$ yields the maximal number of positive literals in clauses obtained by clausifying $\forall x : \pi_y(C, x)$. To clausify a concept C containing a complex subconcept at a position p, one should consider if $C|_p$ occurs in C under positive or negative polarity. E.g., in $\neg(\neg A \sqcap \neg B)$ the concepts A and B occur effectively positive, and $\sqcap$ is effectively $\sqcup$. Hence, $\mathsf{pl}^+(C|_p)$ ($\mathsf{pl}^-(C|_p)$) counts the number of positive literals used to clausify $C|_p$, provided that $C|_p$ occurs in C under positive (negative) polarity. The function $\mathsf{sgn}(\cdot)$ takes into account that $C|_p$ will be replaced in C by structural transformation with only one concept name, even if clausification of $C|_p$ produces more than one positive literal. E.g., to clausify $C = \forall R.(D_1 \sqcup D_2)$, the structural transformation replaces $D_1 \sqcup D_2$ with an new concept name Q, yielding $C' = \forall R.Q$; then clausifying C' produces a clause with only one positive literal. Now a concept C is Horn if the maximal number of positive literals obtained by clausifying subconcepts of C is at most one.

If a concept C has a complex subconcept at position p, special care has to be taken in introducing a new name α for $C|_p$. Consider the Horn concept $C = \forall R.D_1 \sqcup \forall R.\neg D_2$. To apply structural transformation to C, one might replace $\forall R.D_1$ and $\forall R.\neg D_2$ with new concept names Q_1 and Q_2, yielding concepts $\neg Q_1 \sqcup \forall R.D_1$, $\neg Q_2 \sqcup \forall R.\neg D_2$ and $Q_1 \sqcup Q_2$. The problem with such a straight-forward application of structural transformation is that a Horn concept C was reduced to a non-Horn concept $Q_1 \sqcup Q_2$, so the structural transformation destroyed Horn-ness. To remedy this, we modify the structural transformation to replace each $C|_p$ with a literal concept α such that clausifying α and $C|_p$ requires the same number of positive literals. In the above example, this would mean that $\forall R.D_1$ should be replaced with Q_1, but $\forall R.\neg D_2$ should be replaced with $\neg Q_2$, yielding concepts $\neg Q_1 \sqcup \forall R.D_1$, $Q_2 \sqcup \forall R.\neg D_2$ and $Q_1 \sqcup \neg Q_2$, which are all Horn.

Although transitivity axioms are translated by π into Horn clauses, recall that the algorithm from Section 2 replaces them with axioms of the form $\forall R.C \sqsubseteq \forall S.(\forall S.C)$. Now $\mathsf{pl}^+(\exists R.\neg C \sqcup \forall S.(\forall S.C)) = 1 + \mathsf{pl}^+(C)$, so if $\mathsf{pl}^+(C) > 0$, $\Omega(KB)$ is not a Horn knowledge base. Hence, the presence of transitivity axioms can make a knowledge base non-Horn.

Definition 2. *Let* C *be a concept and* Λ *a function mapping* C *to the set of positions* $p \neq \epsilon$ *of subconcepts of* C *such that* $C|_p$ *is not a literal concept and, for all positions* q *below* p*,*

$C|_q$ is a literal concept. Then $\mathsf{Def}(C)$ is defined recursively as follows, where $\alpha = Q$ if $\mathsf{pl}(C,p) > 0$, and $\alpha = \neg Q$ if $\mathsf{pl}(C,p) = 0$, with Q a new atomic concept, and $\neg(\neg Q) = Q$:

- $\mathsf{Def}(C) = \{C\}$ *if* $\Lambda(C) = \emptyset$, *or*
- *if* $\Lambda(C) \neq \emptyset$, *then choose some* $p \in \Lambda(C)$ *and let*

$$\mathsf{Def}(C) = \begin{cases} \{\neg\alpha \sqcup C|_p\} \cup \mathsf{Def}(C[\alpha]_p) & \text{if } \mathsf{pol}(C,p) = 1 \\ \{\neg\alpha \sqcup \neg C|_p\} \cup \mathsf{Def}(C[\neg\alpha]_p) & \text{if } \mathsf{pol}(C,p) = -1 \end{cases}$$

Let $\mathsf{Cls}(\varphi)$ denote the set of clauses obtained by clausifying a formula φ in the standard way and let

$$\mathsf{Cls}(C) = \bigcup_{D \in \mathsf{Def}(C)} \mathsf{Cls}(\forall x : \pi_y(D,x)).$$

*For an $\mathcal{ALCHIQ}$ knowledge base KB, $\Xi(KB)$ is the smallest set of clauses such that: (*i*) for each role name $R \in N_{R_a}$, $\mathsf{Cls}(\pi(R)) \subseteq \Xi(KB)$; (*ii*) for each RBox or ABox axiom α in KB, $\mathsf{Cls}(\pi(\alpha)) \subseteq \Xi(KB)$; (*iii*) for each TBox axiom $C \sqsubseteq D$ in KB, $\mathsf{Cls}(\neg C \sqcup D) \subseteq \Xi(KB)$.*

By [Nonnengart and Weidenbach, 2001], $\forall x : \pi_y(C,x)$ and $\bigwedge_{D \in \mathsf{Def}(C)} \forall x : \pi_y(D,x)$ are equisatisfiable, so $\Xi(KB)$ and $\pi(KB)$ are equisatisfiable as well.

Lemma 2. *For a Horn-$\mathcal{SHIQ}$ knowledge base KB, each clause from $\Xi(KB)$ contains at most one positive literal.*

Proof. We first show the following property (*): for a Horn concept C, all concepts in $\mathsf{Def}(C)$ are Horn concepts. The proof is by induction on the recursion depth in Def. The induction base for $\Lambda(C) = \emptyset$ is obvious. Consider an application of $\mathsf{Def}(C)$, where C is a Horn concept and p a position of a subconcept of C, such that $C|_p$ is not a literal concept and, for each position q below p, $C|_q$ is a literal concept. Observe that in all cases, we have $\mathsf{pl}^+(\alpha) = \mathsf{pl}(C,p)$ and $\mathsf{pl}^+(\neg\alpha) = 1 - \mathsf{pl}(C,p)$. We now consider two cases, depending on $\mathsf{pol}(C,p)$:

- $\mathsf{pol}(C,p) = 1$. Now we have $\mathsf{pl}^+(\neg\alpha \sqcup C|_p) = \mathsf{pl}^+(\neg\alpha) + \mathsf{pl}^+(C|_p) = \mathsf{pl}^+(\neg\alpha) + \mathsf{pl}(C,p) = 1$. Furthermore, $\mathsf{pl}(C,p) = \mathsf{pl}(C[\alpha]_p, p)$, so $C[\alpha]_p$ is Horn.
- $\mathsf{pol}(C,p) = -1$. Now we have $\mathsf{pl}^+(\neg\alpha \sqcup \neg C|_p) = \mathsf{pl}^+(\neg\alpha) + \mathsf{pl}^-(C|_p) = \mathsf{pl}^+(\neg\alpha) + \mathsf{pl}(C,p) = 1$. Furthermore, $\mathsf{pl}(C,p) = \mathsf{pl}(C[\neg\alpha]_p, p)$, so $C[\neg\alpha]_p$ is Horn.

Hence, each application of the operator Def decomposes a Horn concept C into two simpler Horn concepts, so (*) holds. Furthermore, for each $C|_p$ or $\neg C|_p$ in the definition of Def, each immediate subconcept is a literal.

For $D \in \mathsf{Def}(C)$, by definition of π from Table 1, it is easy to see that $\mathsf{pl}^+(D)$ gives the maximal number of positive literals occurring in a clause from $\mathsf{Cls}(\forall x : \pi_y(D,x))$. Thus, if C is a Horn concept, all clauses from $\mathsf{Cls}(C)$ contain at most one positive literal. Finally, clauses obtained by translating RBox and ABox axioms of $\Omega(KB)$ also contain at most one positive literal. □

As stated by the following lemma, a basic superposition inference, when applied to Horn premises, produces a Horn conclusion. The full proof of the lemma is given in [Hustadt *et al.*, 2004a].

Lemma 3. *If all premises of an inference by basic superposition contain at most one positive literal, then inference conclusions also contain at most one positive literal.*

Proof. (*Sketch*) Consider a resolution inference with clauses $A \vee C$ and $\neg B \vee D$, where all literals in C are negative and at most one literal in D is positive. Obviously, the number of positive literals in the conclusion $C\sigma \vee D\sigma$ is equal to the number of positive literals in D. Similarly, consider a paramodulation inference from a clause $s \approx t \vee C$ into a clause $A \vee D$, where all literals in C and D are negative. Obviously, the conclusion $A\sigma[t\sigma]_p \vee C\sigma \vee D\sigma$ has only one positive literal. Similar considerations hold for a paramodulation into a negative literal. □

By Lemma 2 and 3, if KB is a Horn-$\mathcal{SHIQ}$ knowledge base, then $\mathsf{DD}(KB)$ is a non-disjunctive program. This is enough for the following result:

Theorem 3. *For KB an extensionally reduced Horn knowledge base in any logic between $\mathcal{ALC}$ and $\mathcal{SHIQ}$, deciding KB (un)satisfiability, and deciding whether $KB \models (\neg)C(a)$ with $|C|$ bounded, is* P*-complete in $|\mathcal{A}|$.*

Proof. Membership in P is a consequence of the fact that $\mathsf{DD}(KB)$ is a non-disjunctive program, whose satisfiability can be checked in polynomial time [Dantsin *et al.*, 2001].

For hardness, consider the well-known P-complete problem of deciding whether a path from a node a_1 to a node a_n in a graph G exists [Papadimitriou, 1993]. For a graph G, let KB_G be a knowledge base containing the assertions $edge(a,b)$ and $edge(b,a)$ for each edge $\langle a,b\rangle$ in G, the axioms $C(a_1)$ and $\neg C(a_n)$, and the TBox axiom $C \sqsubseteq \forall edge.C$. Obviously, a_1 is reachable from a_n if and only if KB_G is unsatisfiable, thus implying P-completeness of unsatisfiability checking. The other inference problems can be reduced to unsatisfiability as usual. □

5 Discussion

To better understand the results from the previous two sections, we contrast them with well-known results for (disjunctive) datalog [Dantsin *et al.*, 2001]. Since datalog has been successfully applied in practice, this analysis gives interesting insights into the practical applicability of DLs.

Interestingly, the data complexity of datalog variants and of corresponding $\mathcal{SHIQ}$ fragments coincide. Namely, without disjunctions, a $\mathcal{SHIQ}$ knowledge base and a datalog program always have at most one model, which can be computed in polynomial time. With disjunctions, several models are possible, and this must be dealt with using reasoning-by-cases. Intuitively, one needs to "guess" a model, which increases data complexity to NP.

The key difference between datalog and DLs is revealed by considering the effects that various parameters have on the complexity. For a datalog program P and a ground atom α, checking whether $P \models \alpha$ can be performed in time $\mathcal{O}(|P|^v)$, where v is the maximal number of distinct variables in a rule of P [Vardi, 1995]. Namely, the problem can be solved by grounding P, i.e. by replacing, in each rule of P, all variables with individuals from P in all possible ways. The size of the

grounding is bounded by $|P|^v$, and propositional Horn logic is P-complete, giving the above estimate. Now in general, v is linear in $|P|$, so the size of the grounding is exponential; thus, the combined complexity of datalog coincides with the combined complexity of $\mathcal{SHIQ}$. However, in practical applications v is usually small, so it makes sense to assume it is bounded. Under this assumption, datalog actually exhibits polynomial behavior.

By an analogy, one might try to limit the length of concepts in axioms or the number of variables. For the former, structural transformation can be used to polynomially reduce "long" axioms with complex concepts to "short" axioms with just elementary concepts. For the latter, we note that DLs are closely related to the two-variable fragment of first-order logic: e.g., $\mathcal{ALC}$ concepts correspond to first-order formulae with only two variables regardless of nesting (see, e.g., [Baader *et al.*, 2003, ch. 4]). Therefore, the number of variables in DL axioms is "intrinsically" bounded (assuming a bound on the numbers occurring in number restrictions). Hence, neither restriction actually reduces complexity.

We summarize our discussion as follows: assuming a bound on the axiom length, but not on the number of axioms, satisfiability checking in datalog is (non-deterministically) polynomial, but in DLs it is exponential. The reason for this is that DLs such as $\mathcal{ALC}$ provide existential and universal quantification and general inclusion axioms, which can be used to succinctly encode models with paths of exponential length. The saturation step eliminates function symbols introduced by existential quantification, but it also incurs an exponential blowup in the program size to account for such paths. Hence, although combined complexity of both datalog and DLs is exponential, the reasons for this are different.

In [Baader *et al.*, 2003, ch. 5] two sources of complexity in DLs have been identified: OR-branching caused by the existence of several possible models, and AND-branching caused by the existence of paths within a model. Our results show that OR-branching is not so "bad" as AND-branching: the former incurs "only" an increase to NP, whereas the latter incurs an increase in complexity to EXPTIME.

6 Conclusion

In many application of DLs, the TBox can be assumed to be rather stable — like a database schema — whereas the ABox is varying and possibly very large — like a database extension. Hence, we study the complexity of reasoning in expressive DLs measured in the size of the ABox. In particular, we show that checking satisfiability of a $\mathcal{SHIQ}$ knowledge base is NP-complete, and that checking unsatisfiability and instance checking are co-NP-complete in the size of the ABox. Furthermore, we identify Horn-$\mathcal{SHIQ}$, a fragment of $\mathcal{SHIQ}$ which, analogously to Horn logic, does not allow to represent disjunctive knowledge, and for which the basic reasoning problems are P-complete in the size of theABox.

Our results indicate that reasoning with large ABoxes may be feasible if the TBox is not "too" large and if we stay within Horn-$\mathcal{SHIQ}$. To verify these assumptions, we are currently implementing our algorithms and plan to conduct a thorough performance analysis.

Acknowledgments

We thank Enrico Franconi for a stimulating discussion which lead us to the results presented in this paper.

References

[Baader *et al.*, 2003] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[Bachmair *et al.*, 1995] L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic Paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[Calvanese *et al.*, 2004] D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, and G. Vetere. DL-Lite: Practical Reasoning for Rich DLs. In *Proc. DL 2004*, volume 104 of *CEUR Workshop Proceedings*, 2004.

[Dantsin *et al.*, 2001] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Comp. Surveys*, 33(3):374–425, 2001.

[Hustadt *et al.*, 2004a] U. Hustadt, B. Motik, and U. Sattler. Reasoning in Description Logics with a Concrete Domain in the Framework of Resolution. Technical Report 2-8-1/04, FZI, Karlsruhe, Germany, February 2004. http://www.fzi.de/wim/publikationen.php?id=1144.

[Hustadt *et al.*, 2004b] U. Hustadt, B. Motik, and U. Sattler. Reducing $\mathcal{SHIQ}^-$ Description Logic to Disjunctive Datalog Programs. In *Proc. KR2004*, pages 152–162. AAAI Press, 2004.

[Nieuwenhuis and Rubio, 1995] R. Nieuwenhuis and A. Rubio. Theorem Proving with Ordering and Equality Constrained Clauses. *Journal of Logic and Computation*, 19(4):312–351, April 1995.

[Nonnengart and Weidenbach, 2001] A. Nonnengart and C. Weidenbach. Computing Small Clause Normal Forms. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 6, pages 335–367. Elsevier Science, 2001.

[Papadimitriou, 1993] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1993.

[Schaerf, 1994] A. Schaerf. *Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity, and Semantic Issues*. PhD thesis, Università degli studi di Roma "La Sapienza", 1994.

[Schild, 1991] K. Schild. A correspondence theory for terminological logics: preliminary report. In *Proc. IJCAI '91*, pages 466–471, Sidney, Australia, 1991.

[Tobies, 2001] S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany, 2001.

[Vardi, 1995] M. Y. Vardi. On the Complexity of Bounded-Variable Queries. In *Proc. PODS 1995*, pages 266–276. ACM Press, 1995.