Description Logics and Disjunctive Datalog — The Story so Far

Ullrich Hustadt
University of Liverpool
Liverpool, UK
U.Hustadt@csc.liv.ac.uk

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

Abstract

In this paper we present an overview of our recent work on the relationship between description logics and disjunctive datalog. In particular, we reduce satisfiability and instance checking in $\mathcal{SHIQ}$ to corresponding problems in disjunctive datalog. This allows us to apply practically successful deductive database optimization techniques, such as magic sets. Interestingly, the reduction also allows us to obtain novel theoretical results on description logics. In particular, we show that the *data complexity* of reasoning in $\mathcal{SHIQ}$ is in NP, and we define a fragment called Horn-$\mathcal{SHIQ}$ for which the data complexity is in P. Finally, the reduction provides a basis for query answering in an extension of $\mathcal{SHIQ}$ with so-called *DL-safe* rules.

1 Introduction

Although the main reasoning problems for $\mathcal{SHIQ}$ are ExpTime-complete, practical tableaux algorithms [11] and numerous heuristics [9] were developed that make reasoning over $\mathcal{SHIQ}$ knowledge bases feasible in practice. These were implemented in state-of-the-art DL reasoners, such as FaCT [10] and Racer [8], and have successfully been applied in numerous practical applications.

Modern DL reasoners are optimized primarily for TBox reasoning. However, some applications, such as metadata management in the Semantic Web, rely mainly on query answering (i.e. instance checking), and not on concept satisfiability or subsumption testing. Existing DL reasoning techniques often do not exhibit satisfactory performance in such cases. We have identified three main causes for that. Firstly, tableaux algorithms provide a refutation procedure, and not a query answering algorithm so, to retrieve all ABox individuals which belong to a given concept w.r.t. a knowledge base, one potentially needs to run the tableaux algorithm for each individual in the ABox. Secondly, it is difficult to guide the application of tableaux inference rules to compute only consequences relevant to the query. Thirdly, tableaux algorithms are non-deterministic, so efficient indexing structures are difficult to develop. Namely, due to backtracking, the benefit of indexing is often outweighed by the overhead of constantly updating the indices[1].

These drawbacks have been addressed to a certain extent in (disjunctive) deductive databases. Such systems usually compute query answers "in one pass", either bottom-up or top-down. Furthermore, techniques for binding propagation, such as (disjunctive) magic sets, have been developed to restrict reasoning only to the part of the ABox relevant to the query. Finally, query answering algorithms (e.g. bottom-up) are deterministic and allow for efficient indexing.

Above mentioned techniques have shown to be very useful on large databases, so applying them to description logics promises to enable efficient reasoning with large ABoxes. Motivated by this idea, in this paper we show how to reduce a $\mathcal{SHIQ}$ knowledge base KB to a disjunctive datalog program $\mathsf{DD}(KB)$, such that $\mathsf{DD}(KB)$ and KB entail the same set of ground facts. Thus, instance checking in KB is reduced to query answering in $\mathsf{DD}(KB)$, which can be performed using available techniques for disjunctive datalog. This reduction is relatively complex, so due to space constraints we only provide a high-level overview; for details, please refer to [13, 14]. In [12] we have presented extensions of this reduction to description logics with datatypes, which we omit here for clarity.

As reported in [15], the reduction allows us to establish a tighter bound on the complexity of our reasoning algorithms. Namely, if the TBox of a knowledge base is much smaller than its ABox, a better estimate of the complexity of instance checking is provided by the so-called *data complexity*, which is measured in the size of the ABox alone. We show that satisfiability checking by our algorithms is data complete for NP. Furthermore, for a Horn fragment of $\mathcal{SHIQ}$, which does not require disjunctive reasoning, our reduction produces non-disjunctive programs, so data complexity becomes polynomial.

Finally, extending DLs with rule formalisms has been an important issue in the development of ontology languages for the Semantic Web. We show that our approach can be used as a basis for reasoning with the decidable *DL-safe* fragment of the Semantic Web Rule Language (SWRL) [17].

---

[1] Private communication with Ralf Möller.

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

Mapping Concepts to FOL	
$\pi_y(\top, X) = \top$	$\pi_y(\bot, X) = \bot$
$\pi_y(A, X) = A(X)$	$\pi_y(\neg C, X) = \neg \pi_y(C, X)$
$\pi_y(C \sqcap D, X) = \pi_y(C, X) \wedge \pi_y(D, X)$	$\pi_y(C \sqcup D, X) = \pi_y(C, X) \vee \pi_y(D, X)$
$\pi_y(\forall R.C, X) = \forall y : R(X, y) \rightarrow \pi_x(C, y)$	$\pi_y(\exists R.C, X) = \exists y : R(X, y) \wedge \pi_x(C, y)$
$\pi_y(\leq n\, R.C, X) = \forall y_1, \ldots, y_{n+1} : \bigwedge R(X, y_i) \wedge \bigwedge \pi_x(C, y_i) \rightarrow \bigvee y_i \approx y_j$	
$\pi_y(\geq n\, R.C, X) = \exists y_1, \ldots, y_n : \bigwedge R(X, y_i) \wedge \bigwedge \pi_x(C, y_i) \wedge \bigwedge y_i \not\approx y_j$	
Mapping Axioms to FOL	
$\pi(C \sqsubseteq D) = \forall x : \pi_y(C, x) \rightarrow \pi_y(D, x)$	$\pi(R \sqsubseteq S) = \forall x, y : R(x, y) \rightarrow S(x, y)$
$\pi(\mathsf{Trans}(R)) = \forall x, y, z : R(x, y) \wedge R(y, z) \rightarrow R(x, z)$	
$\pi(C(a)) = \pi_y(C, a)$	$\pi(a \approx b) = a \approx b$
$\pi(R(a, b)) = R(a, b)$	$\pi(a \not\approx b) = a \not\approx b$
Mapping KB to FOL	
$\pi(R) = \forall x, y : R(x, y) \leftrightarrow R^-(y, x)$	
$\pi(KB) = \bigwedge_{R \in N_R} \pi(R) \wedge \bigwedge_{\alpha \in KB} \pi(\alpha)$	

X is a meta variable and is substituted by the actual variable. π_x is obtained from π_y by simultaneously substituting $x_{(i)}$ for all $y_{(i)}$, respectively, and π_y for π_x.

2 Reducing $\mathcal{SHIQ}$ to Disjunctive Datalog

Our reduction to disjunctive datalog is based on translating a $\mathcal{SHIQ}$ knowledge base KB into a first-order formula $\pi(KB)$, by the operator π from Table 1. It is well-known that KB is satisfiable with respect to the standard direct model-theoretic semantics of $\mathcal{SHIQ}$ iff $\pi(KB)$ is satisfiable in first-order logic [2].

A crucial issue in translating $\pi(KB)$ into a rule-based formalism is how to handle existential quantifiers. Usually, existentially quantified variables are skolemized, i.e. replaced by function symbols which represent the missing individuals. This poses problems for known query evaluation algorithms, which do not necessarily terminate if rules contain function symbols and are cyclic. Therefore, our goal is to produce a (disjunctive) datalog program $\mathsf{DD}(KB)$ *without* function symbols, but which entails exactly the same set of ground facts as KB. Thus, any existing technique can be used for query answering in $\mathsf{DD}(KB)$.

We now overview the algorithm for computing $\mathsf{DD}(KB)$, which is schematically shown in Figure 1. In the first step, we encode KB into an equisatisfiable

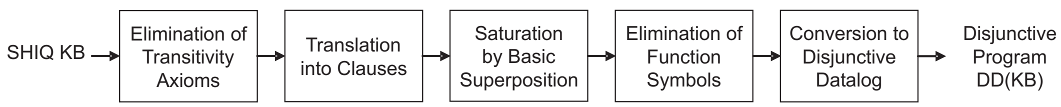

Figure 1: Algorithm for Reducing $\mathcal{SHIQ}$ to Datalog Programs

$\mathcal{ALCHIQ}$ knowledge base $\Omega(KB)$ by eliminating transitivity axioms from KB similarly as this was done for modal logics in [19].

To translate $\Omega(KB)$ into first-order clauses, we apply the well-known *structural transformation* [18] to $\pi(\Omega(KB))$. We thus avoid a potential exponential blowup of the size of the resulting clause set, and also ensure that the structure of clauses is not destroyed. We denote the obtained set of clauses by $\Xi(KB)$.

We next saturate the TBox clauses of $\Xi(KB)$ by basic superposition [1]—a clausal calculus optimized for theorem proving with equality. In this key step, we compute all non-ground clauses derivable from KB by basic superposition. The saturation terminates because all clauses derived in the saturation contain at most one variable and all functional terms are of depth at most two. This yields an exponential bound on the number of clauses we can compute, and an exponential time complexity bound for our algorithm so far.

The saturated set of clauses is next transformed into a function-free clause set $\mathsf{FF}(KB)$. This is possible since the clauses are of a certain restricted form, which enables simulating functional terms with fresh constants. It is possible to show that all inference steps of basic superposition in $\Xi(KB)$ can be simulated in $\mathsf{FF}(KB)$, and vice versa, so $\mathsf{FF}(KB)$ is equisatisfiable to $\Xi(KB)$.

The clauses in $\mathsf{FF}(KB)$ do not contain functional terms and are safe (i.e. each variable in a clause occur in a negative literal), so they can be rewritten into disjunctive rules. We denote the result of such rewriting with $\mathsf{DD}(KB)$.

The following theorem summarizes the properties of our algorithm ($\models_c$ denotes cautions entailment in disjunctive datalog, which coincides on ground facts with first-order entailment for positive datalog programs [7]):

Theorem 1 *For KB a $\mathcal{SHIQ}$ knowledge base and $|KB|$ the length of KB with numbers in number restrictions coded in unary, (*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 an atomic concept, 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 non-atomic concept, and Q a new atomic concept; and (*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 in $\mathcal{SHIQ}$

If the TBox of a knowledge base is much smaller than its ABox, then the *data complexity*, measured in the size of the ABox alone, provides a much better performance estimate than the combined complexity usually considered. The reduction presented in Section 2 immediately provides an upper bound for the data complexity of reasoning in $\mathcal{SHIQ}$. Namely, a closer inspection of the reduction algorithm reveals that $|\mathsf{DD}(KB)|$ is exponential in $|KB|$ only due to the rules obtained by saturating clauses corresponding to TBox axioms. If we assume a bound on the size of the TBox, then the size of the rules in $\mathsf{DD}(KB)$ is bounded as well, and the size of the facts in $\mathsf{DD}(KB)$ is linear in the size of the ABox. Therefore, data complexity of checking satisfiability of $\mathsf{DD}(KB)$ follows from data complexity of positive disjunctive datalog [4], i.e. it is in NP. Intuitively, $\mathcal{SHIQ}$ has the nice property that TBox reasoning does not "interfere" with ABox reasoning, so all non-ground consequences of KB can be computed without taking the ABox into account. Combined with the lower bound from [5], we obtain the following result:

Theorem 2 *Let KB be a knowledge base in any logic between $\mathcal{ALC}$ and $\mathcal{SHIQ}$ containing only atomic concepts in the ABox. 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.

Although still intractable, this is much better than the exponential combined complexity of the same problem. Furthermore, it is easy to see that intractability is due to the fact that $\mathsf{DD}(KB)$ is a disjunctive program. Dealing with disjunction requires reasoning-by-cases, which requires "guessing" — an obvious source of non-determinism. Therefore, to provide a better estimate in the non-disjunctive case, we propose a Horn fragment of $\mathcal{SHIQ}$, where disjunction is traded for P-complete data complexity. In Horn-$\mathcal{SHIQ}$, apart from inverse roles and role inclusion axioms, 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 $\leq 1\, R$. In [15] we give a more general, but also a rather technical definition, which takes into account that some axioms which initially are not of the form above can easily be transformed into the required form by standard equivalences. For a Horn-$\mathcal{SHIQ}$ knowledge base KB, $\mathsf{DD}(KB)$ is a non-disjunctive program, so it can be evaluated in polynomial time, giving a polynomial upper bound for the data complexity. The lower bound follows easily by a reduction from the Boolean circuit value problem, and we get the following result:

Theorem 3 *For KB a Horn knowledge base in a logic between $\mathcal{ALC}$ and $\mathcal{SHIQ}$ containing only atomic concepts in the ABox, deciding KB (un)satisfiability, and deciding whether $KB \models (\neg)C(a)$ with $|C|$ bounded, is data complete for* P.

Due to polynomial data complexity, Horn-$\mathcal{SHIQ}$ is interesting for practical applications. In particular, our experience shows that it can be used to model ontologies in metadata management applications, which are often lightweight and consist mainly of a class hierarchy and simple, possibly functional, relationships. Furthermore, Horn-$\mathcal{SHIQ}$ extends DL-lite [3]—a logic aiming to capture most constructs of ER and UML formalisms—with qualified existential quantification, conditional functionality and role inclusions.

Whereas results from Theorems 2 and 3 are encouraging, $\mathsf{DD}(KB)$ is still potentially exponential in the size of the TBox of KB. Hence, it makes sense to look for ways of restricting this exponential blowup as much as possible. One possible approach might be based on the fact that tableaux algorithms for DLs with acyclic TBoxes are self-terminating; similarly, rules obtained from $\pi(KB)$ are acyclic, so algorithms such as bottom-up evaluation terminate as well. Therefore, it is interesting to investigate whether the reduction from Section 2 can be applied only to the cyclic part of the TBox, as this would introduce a blowup which is exponential only in the size of the maximal TBox cycle. To work out the details is part of our future research.

4 Extending $\mathcal{SHIQ}$ with Rules

Extending description logics with rules to obtain a hybrid knowledge representation system has been an important research issue in the past. Along the lines of [16, 6], in [17] we introduce the notion of *DL-safe* rules. The DL $\mathcal{SHIQ}$ and function-free rules are integrated as usual, by allowing concepts and roles to occur in rules as unary and binary predicates, respectively. Contrary to previous proposals, we allow concepts and roles to occur in rule heads; however, to achieve decidability, we require each variable in the rule to occur in a body literal with a predicate outside of the DL knowledge base.

For example, if *Person*, *livesAt*, and *worksAt* are concepts and roles from KB, the rule $\mathit{Homeworker}(x) \leftarrow \mathit{Person}(x), \mathit{livesAt}(x,y), \mathit{worksAt}(x,y)$ is not DL-safe, since the variables x and y occur only in concepts and roles from KB. The rule can be made DL-safe by appending literals $\mathcal{O}(x)$ and $\mathcal{O}(y)$, where $\mathcal{O}$ is a special predicate used to enumerate all individuals in KB. Semantically, this makes the rule applicable only to explicitly named individuals, and not to individuals introduced by existential quantification.

DL-safety ensures decidability of the hybrid logic, since the rules are applicable only to finitely many explicitly named individuals. Notice, however, that it does not require a closed interpretation domain: unnamed individuals can still be introduced and reasoned with in the DL part of the hybrid knowledge base.

As it turns out, the reduction from Section 2 provides a basis for query answering with DL-safe rules. Namely, for a hybrid knowledge base consisting of a $\mathcal{SHIQ}$ knowledge base KB and a finite set of DL-safe rules P, we have shown that $\pi(KB) \cup P \models \alpha$ iff $\mathsf{DD}(KB) \cup P \models \alpha$, for a ground atom α not involving a simple role. Therefore, reasoning with rules can be performed using well-known techniques from deductive databases. Furthermore, by assuming a bound on the arity of the predicates in P, query answering can be performed in single exponential time, and in double exponential time otherwise.

5 Conclusion

In this paper we present an overview of our recent work on novel algorithms for reasoning in description logics. For a $\mathcal{SHIQ}$ knowledge base KB, we compute a disjunctive datalog program $\mathsf{DD}(KB)$ which is equisatisfiable to KB and entails the same set of ground facts as KB. This allows reusing deductive database optimization techniques, such as magic sets, which, we hope, will make handling large ABoxes feasible in practice. Furthermore, these algorithms exhibit better data complexity: for $\mathcal{SHIQ}$ they run in non-deterministic polynomial time and, if disjunctions are not used, in deterministic polynomial time in the size of the ABox. Finally, the reduction algorithm provides a basis for query answering in an extension of $\mathcal{SHIQ}$ with so-called DL-safe rules.

The main focus of our future work will be to validate the practicability of these algorithms. Therefore, we are currently implementing KAON2[2], a new DL reasoner. Since the implementation has not been finished yet, we are unable to provide a comprehensive performance comparison; however, our initial results are encouraging.

References

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

[2] Alexander Borgida. On the Relative Expressiveness of Description Logics and Predicate Logics. *Artificial Intelligence*, 82(1-2):353–367, 1996.

[3] 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.

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

[5] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Deduction in Concept Languages: From Subsumption to Instance Checking. *Journal of Logic and Computation*, 4(4):423–452, 1994.

[6] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. AL-log: Integrating Datalog and Description Logics. *J. of Intelligent Information Systems*, 10(3):227–252, 1998.

[2]http://kaon2.semanticweb.org/

[7] T. Eiter, G. Gottlob, and H. Mannila. Disjunctive Datalog. *ACM Transactions on Database Systems*, 22(3):364–418, 1997.

[8] V. Haarslev and R. Möller. High Performance Reasoning with Very Large Knowledge Bases: A Practical Case Study. In *Proc. IJCAI 2001*, pages 161–168. Morgan Kaufmann Publishers, 2001.

[9] I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, 1997.

[10] I. Horrocks. Using an Expressive Description Logic: FaCT or Fiction? In *Proc. KR '98*, pages 636–647, Trento, Italy, June 1998. Morgan Kaufmann.

[11] I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.

[12] U. Hustadt, B. Motik, and U. Sattler. Reasoning in Description Logics with a Concrete Domain in the Framework of Resolution. In *Proc. ECAI 2004*, pages 353–357, Valencia, Spain, August 2004.

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

[14] U. Hustadt, B. Motik, and U. Sattler. A Decomposition Rule for Decision Procedures by Resolution-based Calculi. In *Proc. LPAR 2004*, pages 21–35. Springer, 2005.

[15] U. Hustadt, B. Motik, and U. Sattler. Data Complexity of Reasoning in Very Expressive Description Logics. In *Proc. IJCAI 2005*, 2005. to appear.

[16] A. Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104(1-2):165–209, 1998.

[17] B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with Rules. In *Proc. ISWC 2004*, pages 549–563, Japan, 2004. Springer.

[18] 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.

[19] R. A. Schmidt and U. Hustadt. A Principle for Incorporating Axioms into the First-Order Translation of Modal Formulae. In *Proc. CADE-19*, pages 412–426. Springer, 2003.