Reducing $\mathcal{SHIQ}^-$ Description Logic to Disjunctive Datalog Programs

Ullrich Hustadt
University of Liverpool, UK, ullrich@csc.liv.ac.uk
Joint work with Boris Motik, FZI Karlsruhe, Germany
and Ulrike Sattler, University of Manchester, UK

Introduction. In recent years description logics have found their application in various fields of computer science, including data integration, knowledge representation and ontology modelling for the Semantic Web. Some description logic reasoners have been built and applied to practical problems. Experience shows that such systems perform well when computing subsumption hierarchies: they use tableau algorithms [6] which perform much better on these tasks than their ExpTime worst-case computational complexity suggests.

However, new applications, such as metadata management in the Semantic Web, require efficient query answering over large ABoxes. Query answering can be reduced to ABox consistency checking, just as the computation of subsumption hierarchies, which can then be solved by employing above mentioned tableau algorithms. From a theoretical point of view, this approach is quite elegant, but from a practical point of view it has a significant drawback: as the number of ABox individuals increases, the performance may become quite poor. We believe that there are two main reasons for this. Firstly, tableau algorithms treat all individuals separately, that is, they do not group individuals together depending on common properties. Secondly, to answer a query, one usually does not need to consider all ABox information. Rather, only a small subset of the ABox usually suffices to compute the query answer. It is difficult to modify the tableau search strategy to take into account the query in the search. These deficiencies have been acknowledged by the research community, and certain optimisation techniques for instance retrieval have already been developed [5]. However, the performance of query answering is still often not satisfactory.

Since techniques for reasoning in deductive databases are highly successful, we believe it makes sense to examine how to apply them to improve ABox reasoning in description logics. We have recently developed a novel technique for reducing $\mathcal{SHIQ}^-$ knowledge bases to disjunctive datalog programs, while preserving the semantics of the knowledge base. $\mathcal{SHIQ}^-$, a fragment of $\mathcal{SHIQ}$, is a very expressive description logics which is at the core of OWL-DL [10], which in turn is a fragment of the Ontology Web Language (OWL), the current standard for ontology modelling in the Semantic Web.

A $\mathcal{SHIQ}$ knowledge base KB is a triple of the form $(KB_\mathcal{R}, KB_\mathcal{T}, KB_\mathcal{A})$, where $KB_\mathcal{R}$ is an RBox, $KB_\mathcal{T}$ is a TBox, and $KB_\mathcal{A}$ is an ABox.

Given a set N_R of role names, the set of $\mathcal{SHIQ}$ *roles* is the set $N_R \cup \{R^- | R \in N_R\}$, where R^- denotes the *inverse* of R. For $R \in N_R$, $\mathsf{Inv}(R)$ denotes R^- and $\mathsf{Inv}(R^-)$ denotes R. An *RBox* $\mathcal{R}$ over N_R is a finite set of transitivity axioms Trans(R) and role inclusion axioms $R \sqsubseteq S$, where R and S are roles, such that, if $R \sqsubseteq S \in \mathcal{R}$, then $\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) \in \mathcal{R}$ as well. A role R is *transitive* iff $\mathsf{Trans}(S) \in \mathcal{R}$ or $\mathsf{Trans}(\mathsf{Inv}(S)) \in \mathcal{R}$ for some S with $S \sqsubseteq^* R$ and $R \sqsubseteq^* S$; R is *simple* iff there is no transitive role S such that $S \sqsubseteq^* R$ and S; R is *complex* iff it is not simple.

A *TBox* $\mathcal{T}$ is a finite set of *concept inclusion axioms* $C \sqsubseteq D$. Given a set N_C of *atomic* concept names, the set of $\mathcal{SHIQ}$ *concepts* over N_C and N_R is defined inductively as the smallest set for which the following holds: $\top$ and $\bot$ are $\mathcal{SHIQ}$ concepts, each atomic concept name $A \in N_C$ is a $\mathcal{SHIQ}$ concept, if C and D are $\mathcal{SHIQ}$ concepts and R is a role, then $\neg C$ (*complement*), $C \sqcap D$ (*intersection*), $C \sqcup D$ (*union*), $\exists R.C$ (*existential restriction*), $\forall R.C$ (*universal restriction*) are also $\mathcal{SHIQ}$ concepts, and, if C is a $\mathcal{SHIQ}$ concept, R a simple role and n an integer, then $\leq n\, R.C$ and $\geq n\, R.C$ (*number restrictions*) are $\mathcal{SHIQ}$ concepts.

Given a set N_I of *individual names*, an *ABox* $\mathcal{A}$ is a set of concept and role *membership axioms* $C(a)$ and $R(a, b)$, and *(in)equality axioms* $a \approx b$ and $a \not\approx b$, where C is a $\mathcal{SHIQ}$ concept, R a role, and a and b are individuals.

$\mathcal{SHIQ}^-$ differs from $\mathcal{SHIQ}$ [6] in the additional restriction that number restrictions are allowed only for roles not having subroles.

Our translation of $\mathcal{SHIQ}^-$ knowledge bases into disjunctive datalog programs uses a basic superposition decision procedure for $\mathcal{SHIQ}^-$, which is interesting in its own right. Many resolution decision procedures for various classes of logics have already been devised, e.g. for the DL* class [8] or for the (loosely) guarded fragment with equality [3]. However, the standard semantics-based translation of number restrictions does not result in first-order formulae belonging to any of those decidable classes.

We use disjunctive datalog programs as target of our translation for two reasons. Firstly, disjunctive datalog "naturally" handles sets of individuals since it deduces new ground facts simultaneously, which enables various optimisation techniques based on database statistics such as join order optimisation, which were shown to be very effective in practice [1]. Secondly, we can use the magic sets transformation for the identification of the relevant information subset for a conclusion to be derived. The magic sets transformation for disjunctive programs has been presented in [4], along with the empirical evidence of its usefulness. We hope that join order optimisation and magic sets transformation will help us to provide efficient query answering over large ABoxes.

Basic superposition. Basic superposition [2, 7] is an optimised version of superposition (a calculus for equational theories) which prohibits superposition into terms introduced by previous unification steps, thus reducing the number of clauses generated. Its inferences rules are formalised by distinguishing two parts of a clause: (i) the *skeleton* clause C and (ii) the *substitution* σ representing the cumulative effects of previous unifications. The calculus requires two parameters. The first is an *admissible* ordering on terms

Rule	Conditions
Positive superposition: $\dfrac{(C \vee s \approx t) \cdot \rho \;\; (D \vee w \approx v) \cdot \rho}{(C \vee D \vee w[t]_p \approx v) \cdot \theta}$	(*i*) $\sigma = \text{MGU}(s, w\vert_p)$, (*ii*) $\theta = \rho\sigma$, (*iii*) $t\theta \not\succeq s\theta$ and $v\theta \not\succeq w\theta$, (*iv*) $(s \approx t) \cdot \theta$ is SES, (*v*) $(w \approx v) \cdot \theta$ is SES, (*vi*) $s\theta \approx t\theta \not\succeq w\theta \approx v\theta$, (*vii*) $w\vert_p$ is not a variable.
Negative superposition: $\dfrac{(C \vee s \approx t) \cdot \rho \;\; (D \vee w \not\approx v) \cdot \rho}{(C \vee D \vee w[t]_p \not\approx v) \cdot \theta}$	(*i*) $\sigma = \text{MGU}(s, w\vert_p)$, (*ii*) $\theta = \rho\sigma$, (*iii*) $t\theta \not\succeq s\theta$ and $v\theta \not\succeq w\theta$, (*iv*) $(s \approx t) \cdot \theta$ is SES, (*v*) $(w \not\approx v) \cdot \theta$ is ER, (*vi*) $w\vert_p$ is not a variable.
Reflexivity resolution: $\dfrac{(C \vee s \not\approx t) \cdot \rho}{C \cdot \theta}$	(*i*) $\sigma = \text{MGU}(s, t)$, (*ii*) $\theta = \rho\sigma$, (*iii*) $(s \not\approx t) \cdot \theta$ is ER.
Equality factoring: $\dfrac{(C \vee s \approx t \vee s' \approx t') \cdot \rho}{(C \vee t \not\approx t' \vee s' \approx t') \cdot \theta}$	(*i*) $\sigma = \text{MGU}(s, s')$, (*ii*) $\theta = \rho\sigma$, (*iii*) $t\theta \not\succeq s\theta$ and $t'\theta \not\succeq s'\theta$, (*iv*) $(s \approx t) \cdot \theta$ is ES.
Ordered resolution: $\dfrac{(C \vee A) \cdot \rho \;\; (D \vee \neg B) \cdot \rho}{(C \vee D) \cdot \theta}$	(*i*) $\sigma = \text{MGU}(A\rho, B\rho)$, (*ii*) $\theta = \rho\sigma$, (*iii*) $A \cdot \theta$ is SES, (*iv*) $\neg B \cdot \theta$ is ER.

Notes:

(i) $L \cdot \sigma$ is *maximal* in $C \cdot \sigma$ if there is no $L' \in C \setminus \{L\}$ such that $L'\sigma \succ L\sigma$.

(ii) $L \cdot \sigma$ is *strictly maximal* in $C \cdot \sigma$ if there is no $L' \in C \setminus \{L\}$ such that $L'\sigma \succeq L\sigma$.

(iii) A literal $L \cdot \theta$ is *(strictly) eligible for superposition* ((S)ES) in a closure $(C \vee L) \cdot \theta$ if nothing is selected in $(C \vee L) \cdot \theta$ and $L \cdot \theta$ is (strictly) maximal in $C \cdot \theta$.

(iv) A literal $L \cdot \theta$ is *eligible for resolution* (ER) in a closure $(C \vee L) \cdot \theta$ if it is selected in $(C \vee L) \cdot \theta$ or nothing is selected in $(C \vee L) \cdot \theta$ and $L \cdot \theta$ is maximal in $C \cdot \theta$.

Table 1. Inference Rules of Basic Superposition

$\succ$, i.e., a *reduction ordering* total on ground terms. Such an ordering is then extended to literals. The second parameter of the calculus is a *selection function* which selects an arbitrary set of negative literals in a closure.

Given these parameters, the basic superposition calculus, $\mathcal{BS}$ for short, consists of the inference rules given in Table 1.

Reducing $\mathcal{SHIQ}^-$ to Disjunctive Datalog. The first step in our reduction is to eliminate transitivity axioms from KB, translate KB to first-order logic with equality and to transform it into clausal form. To eliminate transitivity axioms we use a similar transformation to the one presented in [11]. We use the standard semantics-based translation π of $\mathcal{SHIQ}^-$ to first-order logic. In order to avoid the exponential blow-up by direct clausification of $\pi(KB)$, we apply *structural transformation* [9] to all non-literal subformulae of $\pi(KB)$. The result of translating and clausifying KB, denoted by $\Xi(KB)$, is satisfiability equivalent to KB, and can be computed in polynomial time.

The second step is to saturate $\Xi(KB)$ which will also decide its satisfiability. To do so, we use the calculus $\mathcal{BS}$ parameterised as follows: (i) the term ordering $\succ$ is a *lexicographic path ordering* (LPO) induced by a total precedence $>_P$ on function, constant and predicate symbols, such that, for any function symbol f, constant symbol c, and predicate symbol p, we have $f >_P c >_P p >_P \top$; and (ii) the selection function selects, in each closure $C \cdot \sigma$, every negative binary literal. We use $\mathcal{BS}_{DL}$ to denote this particular instance of $\mathcal{BS}$. We have shown that any derivation from $\Xi(KB)$ by $\mathcal{BS}_{DL}$ terminates and that the saturated set of clauses can be computed in exponential time given unary coding of numbers in KB. By the soundness and completeness of $\mathcal{BS}$, $\Xi(KB)$ is unsatisfiable iff the saturation of $\Xi(KB)$ contains the empty clause. This procedure in itself would already implement query answering in KB.

For the reduction to disjunctive datalog we only apply $\mathcal{BS}_{DL}$ to the translation of the TBox and RBox of KB, eliminate clauses from the result which are irrelevant for query answering, add the translation of the ABox of KB, and transform the set of clauses into a disjunctive datalog program. The number of rules in the resulting disjunctive datalog program $\mathsf{DD}(KB)$ is exponential in the size of KB and answering a query, given by a single ground atom, with respect to $\mathsf{DD}(KB)$ requires exponential time in the size of the knowledge base KB.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic Paramodulation. *Information and Computation*, 121(2):172–192, 1995.
3. H. Ganzinger and H. de Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proc. of the 14th IEEE Symposium on Logic in Computer Science*, pages 295–305. IEEE Computer Society Press, 1999.
4. S. Greco. Binding Propagation Techniques for the Optimization of Bound Disjunctive Queries. *IEEE Trans. on Knowledge and Data Engineering*, 15(2):717–736, 2003.
5. V. Haarslev and R. Möller. Optimization Strategies for Instance Retrieval. In *Proc. DL-2002*, Toulouse, France, 2002.
6. I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
7. R. Nieuwenhuis and A. Rubio. Theorem Proving with Ordering and Equality Constrained Clauses. *Journal of Logic and Computation*, 19(4):312–351, 1995.
8. H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-Based Methods for Modal Logics. *Logic Journal of the IGPL*, 8(3):265–292, 2000.
9. 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.
10. P. F. Patel-Schneider, P. Hayes, I. Horrocks, and F. van Harmelen. OWL Web Ontology Language; Semantics and Abstract Syntax, W3C Candidate Recommendation. http://www.w3.org/TR/owl-semantics/, November 2002.
11. R. A. Schmidt and U. Hustadt. A Principle for Incorporating Axioms into the First-Order Translation of Modal Formulae. In F. Baader, editor, *Automated Deduction—CADE-19*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 412–426. Springer, 2003.