Rewriting $\mathcal{ALC}$-TBoxes of Depth One via Resolution

Fabio Papacchini

Department of Computer Science, University of Liverpool, Fabio.Papacchini@liverpool.ac.uk

Abstract: Former studies show that there exists a class of $\mathcal{ALC}$-TBoxes of depth one (i.e., with no nested role restrictions) such that answering conjunctive queries is in PTIME in data complexity. We propose ideas on how to rewrite such TBoxes into Horn-$\mathcal{ALC}$-TBoxes of depth one via a resolution-based approach.

1 Introduction

Ontology-based data access (OBDA) has received particular attention from the research community in the last years. When enhancing query answering by the use of an ontology, one of the most natural questions arising is 'how does the ontology affect the complexity of query answering?'. Studies focusing on a wide variety of ontology languages and queries show that query answering becomes easily intractable. This is the case, for example, for the $\mathcal{ALC}$ language, where the worst-case data complexity it is known to be CONP-COMPLETE when answering conjunctive queries (CQs). It is, however, possible to identify $\mathcal{ALC}$-TBoxes for which CQ answering is tractable by adopting a fine-grained approach.

[2] shows that for $\mathcal{ALC}$-TBoxes of depth one (i.e., with no nested role restrictions) there exists a P/CONP dichotomy, and it presents a semantic characterisation for TBoxes where CQ answering is in PTIME. Specifically, [2] proves the equivalence between materialisability, unravelling tolerance and the ABox disjunction property for $\mathcal{ALC}$-TBoxes of depth one, and that CQ answering in the presence of these properties is in PTIME for data complexity. For the purpose of this abstract we give only the formal definition of the ABox disjunction property, and we refer to [2] for the other definitions. A TBox $\mathcal{T}$ has the *ABox disjunction property* if for all ABoxes $\mathcal{A}$ and $\mathcal{EL}$ concepts $C_1, \ldots, C_n$, it follows from $(\mathcal{T}, \mathcal{A}) \models C_1(a_1) \vee \ldots \vee C_n(a_n)$ that $(\mathcal{T}, \mathcal{A}) \models C_i(a_i)$ for some $i \leq n$. We recall that an $\mathcal{EL}$ concept is any concept built from $\exists$, $\sqcap$, $\top$, and any concept name A.

Recent results of our research show that for $\mathcal{ALC}$-TBoxes of depth one the ABox disjunction property can be made stronger by requiring the C_i to be of at most depth one, and that it is enough to check the ABox disjunction property over ABoxes that can be expressed as a depth one $\mathcal{EL}$ assertion. For example, the ABox $\{A(a), r(a,b), B(b), B'(b)\}$ can be represented as $(A \sqcap \exists r.(B \sqcap B'))(a)$.

2 Resolution Calculus

The idea of adopting a resolution-based method as a rewriting procedure is not new, e.g. [1].

We indicate literals by L, where $L := A \mid \neg A$, a possibly empty disjunction of literals by $\mathcal{C}_\sqcup$ or $\mathcal{D}_\sqcup$, a possibly empty conjunction of literals by $\mathcal{C}_\sqcap$ or $\mathcal{D}_\sqcap$. In this context, arbitrary concepts C and D are of the form $C := L \mid \forall r.\mathcal{C}_\sqcup \mid \exists r.\mathcal{C}_\sqcap$. Finally, we indicate the complement by an overline (e.g., $\overline{L}$ is the complement of a literal, $\overline{\mathcal{C}_\sqcap}$ is the complement of a conjunction of literals).

As the procedure is a resolution-based procedure, the input is a set of DL clauses. The set of clauses is computed using the common CNF transformation, with the proviso that $\forall r$ is distributed over conjunctions and $\exists r$ is distributed over disjunctions. Given the aforementioned definitions, a clause is just a disjunction of arbitrary concepts, and we indicate clauses by $\mathcal{C}$ and $\mathcal{D}$. This normal form and representation of clauses is possible because we assume the input to be an $\mathcal{ALC}$-TBox of depth one.

To simplify the calculus we assume that any clause in the calculus does not present any repetition, neither in a disjunction nor in a conjunction. Table 1 shows the rules of the resolution calculus, which we call Res.

Subsumption between clauses can be defined on a completely syntactic level. First, let us define an ordering between role restricted concepts. We say that $\forall r.\mathcal{C}_\sqcup \preceq \forall r.\mathcal{D}_\sqcup$ if $\mathcal{C}_\sqcup \subseteq \mathcal{D}_\sqcup$, and that $\exists r.\mathcal{C}_\sqcap \preceq \exists r.\mathcal{D}_\sqcap$ if $D_\sqcap \subseteq \mathcal{C}_\sqcap$. We indicate that a clause $\mathcal{C}$ *subsumes* a clause $\mathcal{D}$ by $\mathcal{C} \preceq \mathcal{D}$. $\mathcal{C} \preceq \mathcal{D}$ holds if either

- for all $L \in \mathcal{C}$ it is the case that $L \in \mathcal{D}$,
- for all $\exists r.\mathcal{C}_\sqcap \in \mathcal{C}$ there exists $\exists r.\mathcal{D}_\sqcap \in \mathcal{D}$ s.t. $\exists r.\mathcal{C}_\sqcap \preceq \exists r.\mathcal{D}_\sqcap$, and
- for all $\forall r.\mathcal{C}_\sqcup \in \mathcal{C}$ there exists $\forall r.\mathcal{D}_\sqcup \in \mathcal{D}$ s.t. $\forall r.\mathcal{C}_\sqcup \preceq \forall r.\mathcal{D}_\sqcup$,

or,

$\mathcal{C} = L_1 \sqcup \ldots \sqcup L_n$ and $\forall r.(L_1 \sqcup \ldots \sqcup L_m) \in \mathcal{D}$ where $m \geq n$.

A clause is *redundant* if it is subsumed or it is a tautology.

Clause simplification is applied during the saturation process, and we indicate the exhaustive application of simplification rules to a clause $\mathcal{C}$ by SIMP($\mathcal{C}$). Apart from the common simplification rules (e.g., removing $\exists r.\bot$ from a clause), two rules need to be illustrated. First,

$$(\mathbf{Cond})\ \frac{\mathcal{C} \sqcup \forall r.\mathcal{C}_\sqcup \sqcup \forall r.\mathcal{C}'_\sqcup}{\mathcal{C} \sqcup \forall r.\mathcal{C}'_\sqcup}\ \ \mathcal{C}_\sqcup \subseteq \mathcal{C}'_\sqcup\,.$$

The $(\mathbf{Cond})$ rule represents a condensation step. The first-order translation helps to illustrate the purpose of the rule.

$$(BR)\ \frac{\mathcal{C} \sqcup C \qquad \overline{C} \sqcup \mathcal{D}}{\mathcal{C} \sqcup \mathcal{D}} \qquad (\forall\bot)\ \frac{\mathcal{C} \sqcup \forall r.\bot \qquad \exists r.\mathcal{D}_\sqcap \sqcup \mathcal{D}}{\mathcal{C} \sqcup \mathcal{D}}$$

$$(BR\forall)\ \frac{\mathcal{C}_\sqcup \sqcup L \qquad \mathcal{D} \sqcup \forall r.(\overline{L} \sqcup \mathcal{D}_\sqcup)}{\mathcal{D} \sqcup \forall r.(\mathcal{C}_\sqcup \sqcup \mathcal{D}_\sqcup)} \qquad (BR\exists)\ \frac{\bigsqcup_i L_i \sqcup L \qquad \mathcal{C} \sqcup \exists r.(\overline{L} \sqcap \mathcal{C}_\sqcap)}{\mathcal{C} \sqcup \bigsqcup_i \exists r.(L_i \sqcap \mathcal{C}_\sqcap)}$$

$$(\forall\forall)\ \frac{\mathcal{C} \sqcup \forall r.(\mathcal{C}_\sqcup \sqcup L) \qquad \mathcal{D} \sqcup \forall r.(\overline{L} \sqcup \mathcal{D}_\sqcup)}{\mathcal{C} \sqcup \mathcal{D} \sqcup \forall r.(\mathcal{C}_\sqcup \sqcup \mathcal{D}_\sqcup)} \qquad (\forall\exists)\ \frac{\mathcal{C} \sqcup \forall r.(\bigsqcup_i L_i \sqcup L) \qquad \mathcal{D} \sqcup \exists r.(\overline{L} \sqcap \mathcal{D}_\sqcap)}{\mathcal{C} \sqcup \mathcal{D} \sqcup \bigsqcup_i \exists r.(L_i \sqcap \mathcal{D}_\sqcap)}$$

Table 1: Rules of the resolution calculus Res

Consider as an example the clause $\forall r.(A \sqcup B) \sqcup \forall r.(A \sqcup B \sqcup C)$, its first-order translation is as follows.

$$\neg r(x,y) \vee A(y) \vee B(y) \vee \neg r(x,z) \vee A(z) \vee B(z) \vee C(z)$$

A first-order prover would substitute y with z, and then apply factoring as much as possible. The resulting clause would be $\neg r(x,z) \vee A(z) \vee B(z) \vee C(z)$, which subsumes the original one. The (**Cond**) rule performs exactly the same simplification.

Second,

$$(\mathbf{Taut})\ \frac{\mathcal{C} \sqcup \forall r.\mathcal{C}_\sqcup \sqcup \exists r.\mathcal{C}_\sqcap}{\top}\ \ \overline{\mathcal{C}_\sqcap} \subseteq \mathcal{C}_\sqcup\ .$$

The (**Taut**) rule recognises particular tautologies, and it is necessary for the second step of our procedure. The idea behind this rule is that a clause such as $\forall r.(A \sqcup B) \sqcup \exists r.\neg A$ is equivalent to the TBox axiom $\exists r.(\neg A \sqcap \neg B) \sqsubseteq \exists r.\neg A$, where its tautological nature is more evident.

Given a set $\mathcal{N}$ of DL clauses, we indicate by $Res^*_\preceq(\mathcal{N})$ the saturation of $\mathcal{N}$ w.r.t. Res and the redundancy criterion.

Claim 1. *Let $\mathcal{N}$ be a set of DL clauses of an $\mathcal{ALC}$-TBox of depth one. Then $Res^*_\preceq(\mathcal{N})$ contains all non-redundant consequences of $\mathcal{N}$ with depth less or equal one.*

Soundness of the calculus can be easily proved, and refutational completeness of $Res^*_\preceq$ follows from Claim 1.

Claim 2. *$\mathcal{N}$ is unsatisfiable iff $\bot \in Res^*_\preceq(\mathcal{N})$.*

Claim 3. *$Res^*_\preceq(\mathcal{N})$ terminates for any set $\mathcal{N}$ of clauses. (There are only exponentially many non redundant consequences of depth one.)*

3 Procedure

Let us define a function POS as follows.

$$\text{POS}(C) = \begin{cases} \exists r.\bigsqcap A_i & \text{if } C = \exists r.\mathcal{C}_\sqcap \text{ and } A_i \in \mathcal{C}_\sqcap \\ C & \text{otherwise} \end{cases}$$

The POS function can be extended to clauses.

Let $\mathcal{N}$ be the set of DL clauses resulting from an $\mathcal{ALC}$-TBox of depth one. The procedure is divided into two parts. First, the resolution calculus Res is exhaustively applied to $\mathcal{N}$, resulting in the set $S = Res^*_\preceq(\mathcal{N})$. Second, let $S^+ = \{\mathcal{C} \mid \mathcal{C} = \text{SIMP}(\text{POS}(\mathcal{C}')), \mathcal{C}' \in S\}$, and let $S^+_\preceq$ be the result of applying redundancy elimination to S^+.

Claim 4. *Let $\mathcal{T}$ be an $\mathcal{ALC}$-TBox of depth one. $\mathcal{T}$ has the ABox disjunction property iff any clause $\mathcal{C} \in S^+_\preceq$ is a Horn-clause.*

The intuition for Claim 4 is as follows. First, any clause in $S^+_\preceq$ can be rewritten as a TBox axiom $C \sqsubseteq D$ where C is an $\mathcal{EL}$ concept. ($\Rightarrow$) Take a non-Horn clause in $S^+_\preceq$, build an ABox $\mathcal{A}$ from the left hand-side of its TBox representation, and prove that the ABox disjunction property fails on $\mathcal{A}$. ($\Leftarrow$) If the ABox disjunction property fails on an ABox $\mathcal{A}$, then there exists a non-Horn clause in $S^*_\preceq$.

Claim 5. *Let $\mathcal{T}$ be an $\mathcal{ALC}$-TBox of depth one. If $\mathcal{T}$ has the ABox disjunction property, then $S^+_\preceq$ is a Horn rewriting preserving CQ-answering.*

4 Conclusion

The ideas proposed in this abstract represent a possible syntactic characterisation for $\mathcal{ALC}$-TBoxes of depth one for which CQ answering is tractable. Due to the normal form transformation and the size of the resulting set of clauses, the procedure is double exponential in the size of the original TBox. This result agrees with the upper bound we obtained via a different approach. Evaluations of BioPortal ontologies[1], however, shows that the normal form transformation does not usually result in a substantial increase in size.

Acknowledgements

All the ideas in this abstract would have not been possible without the help and discussions with André Hernich, Boris Konev and Frank Wolter.

References

[1] M. Kaminski, Y. Nenov, and B. Cuenca Grau. Computing datalog rewritings for disjunctive datalog programs and description logic ontologies. In *Proc. of RR'14*, volume 8741 of *LNCS*, pages 76–91. Springer, 2014.

[2] C. Lutz and F. Wolter. Non-uniform data complexity of query answering in description logics. In *Proc. of KR'12*. AAAI Press, 2012.

[1] http://bioportal.bioontology.org