Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Converting Instance Checking to Subsumption: A Rethink for Object Queries over Practical Ontologies

Jia Xu, Ubbo Visser, Mansur Kabuka
University of Miami, Coral Gables, FL 33146, USA
Associated report in http://goo.gl/BOfo1F

Abstract

Instance checking is considered a central service for data retrieval from description logic (DL) ontologies. In this paper, we propose a revised most specific concept (MSC) method for DL $\mathcal{SHI}$, which converts instance checking into subsumption problems. This revised method can generate small concepts that are specific-enough to answer a given query, and allow reasoning to explore only a subset of the ABox data to achieve efficiency. Experiments show effectiveness of our proposed method in terms of concept size reduction and the improvement in reasoning efficiency.

Introduction

One of the core tasks in Description Logic (DL) systems is to provide an efficient way to query the assertional knowledge (i.e. ABox $\mathcal{A}$) in a DL ontology; and DL systems are expected to scale well with respect to (w.r.t.) the fast growing ABox data.

Instance checking that tests whether an individual is an instance of a given concept, is considered the most basic service for data retrieving from ontology ABoxes. A common intuition about realizing instance checking is the so-called most specific concept (MSC) method (Donini and Era 1992) that computes the MSC of a given individual and reduces instance checking of this individual into a *subsumption test* (i.e. test if one concept is more general than the other). More precisely, once the most specific concept C of an individual a is known, to check if a is a member of any given concept D, it is sufficient to test if C is subsumed by D w.r.t. the terminological part (i.e. TBox $\mathcal{T}$) of the ontology.

The computation of a MSC, however, could be difficult when qualified existential restrictions (i.e. $\exists R.C$) are supported by DLs. For example, when computing the MSC for individual a given assertion $R(a, a)$, there may not exist a finite representation of the concept. Most importantly, the computation may involve assertions of other individuals that are connected to the given one through role assertions, which may consequently make the resulting MSC a large concept and reasoning with it degenerated into a prohibitively expensive procedure.

In this paper, we propose a revised MSC method that solves the above mentioned problems by using nominals (Donini and Era 1992) and applying a *call-by-need* strategy together with optimizations. The revised method takes into consideration only the related ABox information and computes a concept for each individual that is only *specific enough* to answer the *current* query w.r.t. the TBox. Based on this strategy, the revision allows the method to generate much simpler and smaller concepts than the original MSC's by ignoring irrelevant ABox assertions. On the other hand, the complexity reduction comes with the price of re-computation for every new query if no optimization is applied. Nevertheless, as shown in our experimental evaluation, the achieved reduction could be significant in many practical ontologies, and the overhead is thus negligible comparing with the reasoning efficiency gained for instance checking. Moreover, due to the re-computations, the ABox data is amenable to frequent modifications, which is in contrast to the original MSC method where a relatively static ABox is assumed.

The Revised MSC Method

Without loss of generality, we assume every concept in a given ontology is in *simple-form* with maximum level of nested quantifiers less than 2. The original MSC of an individual a preserves *complete* information of a w.r.t. the ABox $\mathcal{A}$, denoted $\mathrm{MSC}(\mathcal{A}, a)$. To apply the call-by-need strategy, we abandon this completeness and compute a concept that is only *specific enough* to determine if individual a can be classified into current query concept D. A simple way to realize this strategy is to assign a fresh name A every time to a (complex) query concept D by adding the axiom $A \equiv D$ to $\mathcal{T}$,[1] and to concentrate only on ABox assertions that would (probably) classify individual a into A w.r.t. $\mathcal{T}$.

Theoretically, a sufficient and necessary condition for a role assertion $R(a, b)$ to derive individual classification $A(a)$ is that, the class term behind $R(a, b)$ conjuncted with other essential information of a should be subsumed by concept A w.r.t. $\mathcal{T}$ (Donini and Era 1992). More precisely, this condition can be expressed as:

$$\mathcal{T} \models \exists R.B \sqcap A_0 \sqsubseteq A, \tag{1}$$

[1] Notice that, to obey the simple-form concept restriction, multiple axioms may be added.

where $b \in B$, and concept A_0 summarizes other information that is also essential for this classification, with $A_0 \not\sqsubseteq A$. As shown in (Xu et al. 2013), for (1) to hold when A is a named concept, there must exist some role restriction $\exists R'.C$ with $R \sqsubseteq R'$ used in the TBox for concept definition; otherwise $\exists R.B$ is not comparable (w.r.t. subsumption) with other named concepts (except $\top$ and its equivalents). This syntactic premise is formally indicated as follows.

Proposition 1 ((Xu et al. 2013)). *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{SHI}$ ontology with simple-form concepts only, $\exists R.B$ and A_0 be $\mathcal{SHI}$ concepts, and A a named concept. If*

$$\mathcal{K} \models \exists R.B \sqcap A_0 \sqsubseteq A$$

with $A_0 \not\sqsubseteq A$, there must exist formulae in $\mathcal{T}$ in the form as:

$$\exists R'.C_1 \bowtie C_2 \sqsubseteq C_3 \tag{2}$$

where $R \sqsubseteq R'$ and $\bowtie$ is a place holder for $\sqcup$ and $\sqcap$. Also note the following equivalence:

$$\begin{array}{lcl} \exists R.C \sqsubseteq D & \equiv & \neg D \sqsubseteq \forall R.\neg C \\ \exists R.C \sqsubseteq D & \equiv & C \sqsubseteq \forall R^{-}.D \end{array}$$

This proposition is proven in (Xu et al. 2013). It states in fact a syntactic premise in $\mathcal{SHI}$ for a role assertion to be essential for some individual classification. That is, if $R(a, b)$ is essential for derivation of $A(a)$, there must exist some related axiom in $\mathcal{T}$ in the form of (2) for $R \sqsubseteq R'$. This condition can be further optimized by consider the following two cases for axiom (2):

1. if there is any concept B_0 in explicit class assertions of individual b, such that $\mathcal{K} \models B_0 \sqsubseteq \neg C_1$, or
2. if there is any concept A_0 in explicit class assertions of individual a, such that $\mathcal{K} \models A_0 \sqsubseteq \neg(C_3 \sqcup \neg C_2)$ or $\mathcal{K} \models A_0 \sqsubseteq \neg C_3$, respectively for $\bowtie$ standing for $\sqcap$ or $\sqcup$.

Either one of the above cases happening, that particular axiom in fact makes no contribution to the derivation of $A(a)$, unless the ABox is *inconsistent* where MSC's are always $\bot$. Thus, a revised condition requires not only the existence of related axiom (2) but also with none of the above cases happening. We denote this condition as SYN_COND*, and use it to rule out assertions that are irrelevant to the current query. A algorithm for computation of a specific-enough "MSC" (denoted $\text{MSC}_{\mathcal{T}}$) is given in the associated report.

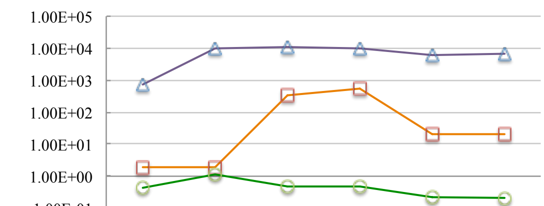
Empirical Evaluation

We tested our method on a set of well-known ontologies with large ABoxes: benchmark LUBM (LM), *extended* DBpedia (DP), and realistic biomedical ontologies AT and CE. More details can be found in the associated report.

For evaluation and comparison, we also implemented the original MSC method. We compute the $\text{MSC}_{\mathcal{T}}$ for each individual in every ontology using the two methods respectively, and measure the complexity of the resulted concepts in terms of the maximum and the average depth of nested quantifiers (see Table 1). We report in Figure (1) the reasoning efficiency achieved when using the revised MSC method for instance checking, comparing with a *complete* ABox reasoning using DL reasoner HermiT (Motik, Shearer, and Horrocks 2009), which implements various optimizations for the reasoning algorithm. We also compared our method with the *modular* reasoning reported in (Xu et al. 2013). Efficiency in the initialization stage (e.g. ontology loading and reasoner initialization) can also be achieved using the $\text{MSC}_{\mathcal{T}}$ method, as it only needs to load a TBox while a compete reasoning requires loading of both the TBox and the big ABox.

Table 1: Quantification depth of $\text{MSC}_{\mathcal{T}}$'s

	Original		Revised	
Ontology	Max.	Avg.	Max.	Avg.
LM1	5,103	4,964.68	2	1.48
LM2	23,015	22,654.01	2	1.51
AT	2,605	2,505.50	8	3.02
CE	3,653	3,553.4	8	2.76
DP1	3,906	3,070.80	4	1.13
DP2	3,968	3,865.60	5	1.12



	LM1	LM2	AT	CE	DP1	DP2
Complete	733.21	9,839.22	11,378.63	10,336.61	6,187.01	6,948.20
Modular	1.93	1.91	344.53	542.6	19.88	20.2
MSC_T	0.45	1.13	0.49	0.49	0.22	0.21

Figure 1: Average time (ms) on instance checking.

Conclusion

We proposed a revised MSC method for efficient instance checking in DL $\mathcal{SHI}$. This method allows the ontology reasoning to explore only a subset of ABox data that is relevant to a given instance checking problem, thus being able to achieve great efficiency and to solve the limitation of current memory-based reasoning techniques. It can be particularly useful for answering object queries over those large *non-Horn* DL ontologies, where existing optimization techniques may fall short and answering object queries may demand thousands or even millions of instance checking tasks. Due to the independence between $\text{MSC}_{\mathcal{T}}$'s, scalability for query answering over huge ontologies (e.g. semantic webs) can also be achieved by parallelizing the computations.

References

Donini, F., and Era, A. 1992. Most specific concepts for knowledge bases with incomplete information. In *Proceedings of CIKM*, 545–551.

Motik, B.; Shearer, R.; and Horrocks, I. 2009. Hypertableau reasoning for description logics. *JAIR* 36(1):165–228.

Xu, J.; Shironoshita, P.; Visser, U.; John, N.; and Kabuka, M. 2013. Extract ABox Modules for Efficient Ontology Querying. *ArXiv e-prints* arXiv:1305.4859 [cs.AI].