Computing Maximally Satisfiable Terminologies for the Description Logic $\mathcal{ALC}$ with GCIs

Kevin Lee and Thomas Meyer
National ICT Australia and University of New South Wales,
Sydney, Australia
{kevin.lee, thomas.meyer}@nicta.com.au

Jeff Z. Pan
Department of Computing Science, University of Aberdeen,
Aberdeen, UK
jpan@csd.abdn.ac.uk

Introduction

Existing description logic reasoners provide the means to detect logical errors in ontologies, but lack the capability to resolve them. We present a tableau-based algorithm for computing maximally satisfiable terminologies in $\mathcal{ALC}$. Our main contribution is the ability of the algorithm to handle GCIs, using a refined blocking condition that ensures termination is achieved at the right point during the expansion process. Our work is closely related to that of [1], which considered the same problem for assertional (Abox) statements only, and [2], which deals only with unfoldable terminologies for $\mathcal{ALC}$.

Computing Maximally Satisfiable Terminologies

The algorithm receives as input a Tbox $\mathcal{T}$ and a concept name A, represented in $\mathcal{ALC}$, and returns as output the A-MSSs of $\mathcal{T}$. Each sentence in $\mathcal{T}$ is labelled with a unique propositional atom, hence $\mathcal{T}$ will be a set of labelled axioms of the form $(C \sqsubseteq D)^p$ and $(C \doteq D)^p$. Our algorithm starts by creating an Abox $\mathcal{A}$ containing only the labelled assertion $A(x)^\top$, where x is an individual name. It then proceeds by continuously applying the expansion rules below, first to $\mathcal{A}$, and then to the Aboxes created subsequently, until none of the rules are applicable. Note that the algorithm does not teminate even when a clash is detected, also we assume that all concept assertions are in negation normal form. The expansion

$\sqcap$-rule	**if** $C_1 \sqcap C_2(x)^\phi \in \mathcal{A}$, and either $C_1(x)^\phi$ or $C_2(x)^\phi$ is $\mathcal{A}$-insertable, **then** $\mathcal{A}' := (\mathcal{A} \oplus C_1(x)^\phi) \oplus C_2(x)^\phi$.
$\sqcup$-rule	**if** $C_1 \sqcup C_2(x)^\phi \in \mathcal{A}$, and both $C_1(x)^\phi$ and $C_2(x)^\phi$ are $\mathcal{A}$-insertable, **then** $\mathcal{A}' := \mathcal{A} \oplus C_1(x)^\phi$, $\mathcal{A}'' := \mathcal{A} \oplus C_2(x)^\phi$.
$\exists$-rule	**if** $\exists R.C(x)^\phi \in \mathcal{A}$, x is not blocked, and either $R(x,y)^\phi$ or $C(y)^\phi$ is $\mathcal{A}$-insertable, **then** $\mathcal{A}' := (\mathcal{A} \oplus R(x,y)^\phi) \oplus C(y)^\phi$, where y is a new individual name and $y > y'$ for all individual names y' in $\mathcal{A}$.
$\forall$-rule	**if** $\{\forall R.C(x)^\phi, R(x,y)^\psi\} \subseteq \mathcal{A}$, and $C(y)^{\phi \wedge \psi}$ is $\mathcal{A}$-insertable, **then** $\mathcal{A}' := \mathcal{A} \oplus C(y)^{\phi \wedge \psi}$.
$\sqsubseteq$-rule	**if** $(C \sqsubseteq D)^\phi \in \mathcal{T}$, $(\neg C \sqcup D)(x)^\phi$ is $\mathcal{A}$-insertable for some individual name x, **then** $\mathcal{A}' := \mathcal{A} \oplus (\neg C \sqcup D)(x)^\phi$.
$\doteq$-rule	**if** $(C \doteq D)^\phi \in \mathcal{T}$, $(\neg C \sqcup D) \sqcap (C \sqcup \neg D)(x)^\phi$ is $\mathcal{A}$-insertable for some individual name x, **then** $\mathcal{A}' := \mathcal{A} \oplus (\neg C \sqcup D) \sqcap (C \sqcup \neg D)(x)^\phi$.

rules make use of the following abbreviations and definitions: $\mathcal{A} \oplus C(x)^\phi$ stands for: $(\mathcal{A} \setminus \{C(x)^\psi\}) \cup \{C(x)^{\phi \vee \psi}\}$ if $C(x)^\psi \in \mathcal{A}$, and $\mathcal{A} \cup \{C(x)^\phi\}$ otherwise. Similarly, $\mathcal{A} \oplus R(x,y)^\phi$ stands for: $(\mathcal{A} \setminus \{R(x,y)^\psi\}) \cup \{R(x,y)^{\phi \vee \psi}\}$ if $R(x,y)^\psi \in \mathcal{A}$, and $\mathcal{A} \cup \{R(x,y)^\phi\}$ otherwise. We refer to a labelled concept assertion $C(x)^\phi$ as $\mathcal{A}$-*insertable* iff, whenever there is a ψ such that $C(x)^\psi \in \mathcal{A}$, then $\phi \not\models \psi$. After constructing an expansion tree using the above expansion rules, we compute a propositional formula called the *clash-resolve* formula as follows: Suppose $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are the complete Aboxes obtained from the expansion. A particular clash $\{C(x)^{\phi_1}, \neg C(x)^{\phi_2}\} \subseteq \mathcal{A}_i$ is expressed by the propositional formula $\phi_1 \wedge \phi_2$. Now suppose there are k_i clashes in $\mathcal{A}_i$, and let $\psi_{i,1}, \ldots, \psi_{i,k_i}$ be the formulas expressing all the clashes in $\mathcal{A}_i$. The clash-resolve formula associated with $\mathcal{T}$ and A is: $\bigvee_{i=1}^{n} \bigwedge_{j=1}^{k_i} \neg\psi_{i,j}$. To compute the A-MSSs , simply find the prime implicants of the clash-resolve formula. Each prime implicant is of the form $(\neg p_1 \wedge \ldots \wedge \neg p_m)$, and the corresponding A-MSS can be found by removing from the original set of axioms $\mathcal{T}$ the axioms whose labels are $p_1, \ldots, p_m$. Our result also show that classical blocking does not always block correctly at the right point, hence it will not always yield the desired results. The reason is that the labels associated with sentences are not taken into account when blocking is performed. Therefore, we define the *refined* blocking condition as follows: An individual y is *blocked* by x iff $y > x$ and for every $C(y)^\psi \in \mathcal{A}$, it is the case that $C(x)^\phi \in \mathcal{A}$ for some ϕ such that $\psi \models \phi$.

Conclusion and Future Work

We presented an algorithm for computing maximally satisfiable for description logic represented in $\mathcal{ALC}$. Unlike the existing algorithms, our proposed algorithm could also handle GCIs. We outlined some limitations with the classic subset blocking in the labelled satifiablity algorithm and addressed these issues by proposing a refined blocking condition. For future work, we will investigate on extending our algorithm to more expressive description logics, such as $\mathcal{SI}$ and $\mathcal{ALCN}$.

References

[1] Franz Baader and Bernhard Hollunder. Embedding Defaults into Terminological Knowledge Representational Formalisms. *Journal of Automated Reasoning*, 14:149–180, 1995.

[2] Stefan Schlobach. Diagnosing terminologies. In *Proceedings of AAAI05*, pages 670–675, 2005.