Complexity of the Description Logic $\mathcal{ALCM}$

Mónica Martínez  and Edelweis Rohrer  
Instituto de Computación, Facultad de Ingeniería, 
Universidad de la República, Uruguay

Paula Severi  
Department of Computer Science, 
University of Leicester, England

Abstract

In this paper we show that the problem of deciding the consistency of a knowledge base in the Description Logic $\mathcal{ALCM}$ is ExpTime-complete. The $\mathcal{M}$ stands for meta-modelling as defined by Motz, Rohrer and Severi. To show our main result, we define an ExpTime Tableau algorithm as an extension of an algorithm for $\mathcal{ALC}$ by Nguyen and Szalas.

1 Introduction

The main motivation of the present work is to study the complexity of meta-modelling as defined in (Motz, Rohrer, and Severi 2014; 2015). No study of complexity has been done so far for this approach and we would like to analyze if it increases the complexity of a given description logic. The standard tableau algorithm for $\mathcal{ALC}$ which builds completion trees, e.g. see (Baader et al. 2003), can be extended with the expansion rules for meta-modelling of (Motz, Rohrer, and Severi 2015). However, it has a high (worse case) complexity, namely NExpTime, and cannot be used to prove that the consistency problem for $\mathcal{ALCM}$ is ExpTime-complete. In our approach we define a flexible syntax to equate an individual to a concept, with a strong semantics which ensures that the interpretation of the individual coincides with that of the concept. So, the domain of an interpretation can no longer consist of only basic objects, but it has to be a well-founded set. Unlike other approaches in the literature, in our approach, general translations to the consistency of a DL without metamodelling does not work because our reasoner has to check inside the algorithm that the canonical model is well-founded.

It is well-known that consistency of a (general) knowledge base in $\mathcal{ALC}$ is ExpTime-complete (Schild 1991; De Giacomo and Lenzerini 1996). The main contribution of this paper is to show that the consistency problem for $\mathcal{ALCM}$ is ExpTime-complete too. So, complexity does not change when moving from $\mathcal{ALC}$ to $\mathcal{ALCM}$. For proving our result, we define a tableau algorithm for checking consistency as an extension of an algorithm for $\mathcal{ALC}$ by (Nguyen and Szalas 2009), and prove that it is ExpTime. Hardness follows trivially from the fact that $\mathcal{ALCM}$ is an extension of $\mathcal{ALC}$ since any algorithm that decides consistency of a knowledge base in $\mathcal{ALCM}$ can be used for a knowledge base in $\mathcal{ALC}$.

Details of our ExpTime algorithm for $\mathcal{ALCM}$ along with proofs of correctness and the complexity result can be found in (Martínez, Rohrer, and Severi 2015).

2 A Flexible Meta-modelling Approach

A knowledge base in $\mathcal{ALCM}$ contains an $\text{Mbox}$ besides of a $\text{Tbox}$ and an $\text{Abox}$. An $\text{Mbox}$ is a set of equalities of the form $a =_m A$ where $a$ is an individual and $A$ is a concept (Motz, Rohrer, and Severi 2015). Figure 1 shows an example of two ontologies separated by a horizontal line, where concepts are denoted by large ovals and individuals by bullets. The two ontologies conceptualize the same entities at different levels of granularity. In the ontology above the horizontal line, rivers and lakes are formalized as individuals while in the one below the line they are concepts. If we want to integrate these ontologies into a single ontology it is necessary to interpret the individual $\text{river}$ and the concept $\text{River}$ as the same real object. Similarly for $\text{lake}$ and $\text{Lake}$. The $\text{Mbox}$ for this example is:

$$\text{river} =_m \text{River} \quad \text{lake} =_m \text{Lake}$$

These equalities are called meta-modelling axioms and in this case, we say that the ontologies are related through meta-modelling. In Figure 1, meta-modelling axioms are represented by dashed edges. After adding the meta-modelling axioms, the concept $\text{HydrographicObject}$ is now also a meta-concept because it is a concept that contains an individual which is also a concept.

This kind of meta-modelling can be expressed in the undecidable logic of OWL Full (Motik 2007) but it cannot be expressed in OWL DL.

OWL 2 DL has a very restricted form of meta-modelling called punning where the same identifier can be used as an individual and as a concept (Hitzi, Krötzsch, and Rudolph 2009). We next illustrate two examples where OWL would not detect inconsistencies because the identifiers, though they look syntactically equal, are treated as different objects.

Example 1 If we introduce an axiom expressing that $\text{HydrographicObject}$ is a subclass of River, then OWL’s reasoner will not detect that the interpretation of River is not a well founded set (it is a set that belongs to itself).
Example 2 We add two axioms, the first one says that river and lake as individuals are equal and the second one says that the classes River and Lake are disjoint. Then OWL’s reasoner does not detect that there is a contradiction.

In order to detect these inconsistencies, river and River should be made semantically equal, i.e. the interpretations of the individual river and the concept River should be the same. The domain $\Delta$ can no longer consist of only basic objects and cannot be an arbitrary set either. We require that the domain be a well-founded set. The reason for this is explained as follows. Suppose we have a domain $\Delta^2 = \{X\}$ where $X = \{X\}$. Intuitively, $X$ is the set $\{\{\ldots\}\}$ which is the solution of a recursive equation obtained by unfolding it an infinite number of times. Clearly, a set like $X$ cannot represent any real object from our usual applications in Semantic Web. The well-foundedness of our model is guaranteed by the reasoner which checks for circularities.

Our approach allows the user to have any number of levels or layers (meta-concepts, meta meta-concepts and so on). The user does not have to write or know the layer of the concept because the reasoner will infer it for him. In this way, axioms can also naturally mix elements of different layers and the user has the flexibility of changing the status of an individual at any point without having to make any substantial change to the ontology.

3 The Description Logic $\text{ALCM}$

In this section, we extend the description logic $\text{ALC}$ (Schmidt-Schauß and Smolka 1991; Baader et al. 2003) with meta-modelling (Motz, Rohrer, and Severi 2014; 2015).

A knowledge base $\mathcal{K}$ in $\text{ALCM}$ is a triple $(\mathcal{T}, \mathcal{A}, \mathcal{M})$ where $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{M}$ are a Tbox, Abox and an Mbox respectively. An Mbox $\mathcal{M}$ is a finite set of meta-modelling axioms. A meta-modelling axiom is a statement of the form $a =_m A$ where $a$ is an individual and $A$ is an atomic concept. Figure 2 shows the Tbox, Abox and Mbox of the knowledge base that corresponds to Figure 1.

In our approach it is specially important to include expressions of the form $a = b$ and $a \neq b$ in the Abox. Individuals with meta-modelling represent new concepts. Since we can express equality and difference between concepts, we also need to be able to express equality and difference between the corresponding individuals. If we have an equality $A \equiv B$ between concepts then $a$ and $b$ should be equal. So, without equalities, the language lacks expressibility for doing inferences of the form $\mathcal{K} \models a = b$. Similarly, if we have that $A$ and $B$ are different, (i.e. there exists an element in $A$ that is not in $B$, since inequalities cannot be expressed by axioms in the Tbox), then $a$ and $b$ should be different.

In our semantics this “correspondence” is in both directions (from individuals to concepts and vice versa), it is what we call Equality Transference (Motz, Rohrer, and Severi 2015).

Definition 1 (Model of a Knowledge Base in $\text{ALCM}$) An interpretation $I$ is a model of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{M})$ in $\text{ALCM}$ if the following holds:

1. the domain $\Delta$ of the interpretation is a subset of some $S_n$ where $S_n$ is defined by starting from an arbitrary set $S_0$ of atomic objects and by giving as inductive step $S_{n+1} = S_n \cup \mathcal{P}(S_n)$.
2. $I$ is a model of $(\mathcal{T}, \mathcal{A})$ in $\text{ALC}$.
3. $a^I = A^I$ for all $a =_m A \in \mathcal{M}$.

In the first part of Definition 1 we restrict the domain of an interpretation in $\text{ALCM}$ to be a subset of $S_n$, which can now contain sets since the set $S_n$ is defined recursively. It is easy to prove that $S_n$ is well-founded for all $n \in \mathbb{N}$. The second part of Definition 1 refers to the $\text{ALC}$-knowledge base without the Mbox axioms.

The third part of the definition restricts the interpretation of an individual that has a corresponding concept through meta-modelling to be equal to the concept interpretation. Figure 3 shows a model for the knowledge base of Figure 2.

Definition 2 (Consistency in $\text{ALCM}$) We say that a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{M})$ is consistent (satisfiable) if there exists a model of $\mathcal{K}$.

An algorithm for checking consistency gives only one model amongst many that, depending on the choices, e.g. the application of the or rule, may or may not be well-founded. A (general) reduction from the consistency of a DL with meta-modelling to the consistency of a DL without meta-modelling does not work. So, checking for circularities has to be done inside the algorithm. To prove that the consistency problem for $\text{ALCM}$ is ExpTime-complete (not greater than $\text{ALC}$), we define an algorithm for $\text{ALCM}$ that is ExpTime.

4 A Tableau Calculus for $\text{ALCM}$

We extend the Tableau Calculus given by (Nguyen and Salsas 2009) to handle meta-modelling axioms. This calculus
uses structures called and-or graphs where both branches of a non-deterministic choice introduced by disjunction are explicitly represented. Satisfiability of the branches is propagated bottom-up and if it reaches an initial node, we can be sure that a model exists. A global catching of nodes and a proper rule-application strategy is used to guarantee the exponential bound on the size of the graph.

We make several changes to this algorithm to accommodate meta-modelling. Basically, the key feature of our extension is given by four new rules (Figure 4) for the expansion of the nodes of the graph, which are the mechanism to handle meta-modelling. Some of these rules adds new axioms to the Tbox and others modify the Mbox, so we add TBox and MBox axioms to the labels of the nodes in the and-or graph. Besides checking for contradictions, it is necessary to check for circularities w.r.t. the membership relation. This is done by means of the predicate circular(\(A, M\)) in (\(\bot_3\)) which guarantees that the canonical model is well-founded. We say that circular(\(A, M\)) holds if there is a sequence of meta-modelling axioms \(a_1 =_m A_1, a_2 =_m A_2, \ldots, a_n =_m A_n\) all in \(M\) such that \(A_1(a_2), A_2(a_3), \ldots, A_n(a_1)\) are in \(A\). In that case we say that \(A\) has a circularity w.r.t. \(M\).

Intuitively, the algorithm works as follows:

- First of all, an and-or graph is built applying a global catching of nodes and a proper rule-application strategy.
- If the graph does not contain any node with label \(\bot\), the algorithm returns that the knowledge base is consistent.
- If not, a bottom-up exploration of the graph is done, starting from the node with label \(\bot\), which is marked as inconsistent.

At the end of this process, if the root node was not marked as inconsistent, the algorithm returns that the knowledge base is consistent and otherwise inconsistent.

Let \(n\) be the size of the knowledge base in ALCM. The complexity of the predicate circular is linear on \(n\) since it amounts to detecting cycles in a directed graph (Sedgewick and Wayne 2011). The complexity of the whole algorithm is determined by the size of the constructed graph which is exponential on \(n\). From this, we show the main new result of this paper:

**Theorem 1 (Complexity of ALCM)** Consistency of a (general) knowledge base in ALCM is ExpTime-complete.
preliminary tests show (Vidal 2015). We plan to study the complexity of more expressive logics with meta-modelling, including cardinality restrictions, role hierarchies and nominals (Tobies 2001; Nguyen and Golinska-Pilarek 2014). We will also study the incorporation of meta-modelling to the automata approach (Calvanese, De Giacomo, and Lenzerini 1999). Furthermore, it is also possible to show Pspace-completeness for ALCM under certain conditions of unfoldable Tboxes. The details will appear in a separate report.

7 Acknowledgements

The third author would like to acknowledge a Daphne Jackson fellowship sponsored by EPSRC and the University of Leicester. We would also like to thank Alfredo Viola for some excellent suggestions.

References


