Proceedings, Fifteenth International Conference on
Principles of Knowledge Representation and Reasoning (KR 2016)

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

Mónica Martinez 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 analyse 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 (Martinez, Rohrer, and Severi 2015).

2 A Flexible Meta-modelling Approach

A knowledge base in $\mathcal{ALCM}$ contains an *Mbox* besides of a Tbox and an Abox. An *Mbox* is a set of equalities of the form $a =_{\mathsf{m}} A$ 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 $river$ and the concept $River$ as the same real object. Similarly for $lake$ and $Lake$. The Mbox for this example is:

$$river =_{\mathsf{m}} River \qquad lake =_{\mathsf{m}} Lake$$

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 $HydrographicObject$ 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 (Hitzler, 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* 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).*

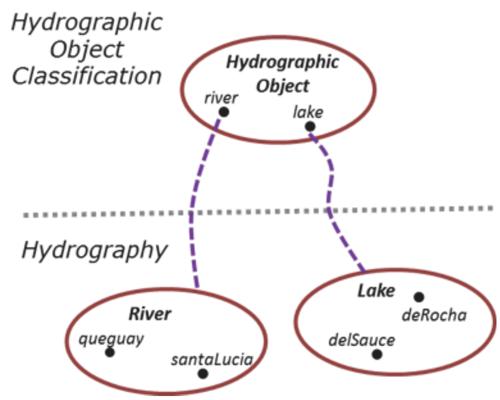

Figure 1: Two ontologies on Hydrography

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 Δ 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 $\Delta^{\mathcal{I}} = \{X\}$ 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-foundness 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 $\mathcal{ALCM}$

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

A *knowledge base* $\mathcal{K}$ *in* $\mathcal{ALCM}$ 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 $a =_{\mathsf{m}} A$ 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 now concepts. Since we can express equality and difference between concepts, we

Tbox
$River \sqcap Lake \sqsubseteq \bot$

Mbox
$river =_{\mathsf{m}} River$
$lake =_{\mathsf{m}} Lake$

Abox
$HydrographicObject(river)$
$HydrographicObject(lake)$
$River(queguay)$
$River(santaLucia)$
$Lake(deRocha)$
$Lake(delSauce)$

Figure 2: Tbox, Abox and Mbox for Figure 1

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 viceversa), it is what we call *Equality Transference* (Motz, Rohrer, and Severi 2015).

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

1. *the domain Δ 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. *$\mathcal{I}$ is a model of $(\mathcal{T}, \mathcal{A})$ in $\mathcal{ALC}$.*
3. *$a^{\mathcal{I}} = A^{\mathcal{I}}$ for all $a =_{\mathsf{m}} A \in \mathcal{M}$.*

In the first part of Definition 1 we restrict the domain of an interpretation in $\mathcal{ALCM}$ 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 $\mathcal{ALC}$-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 $\mathcal{ALCM}$) *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 $\mathcal{ALCM}$ is ExpTime-complete (not greater than $\mathcal{ALC}$), we define an algorithm for $\mathcal{ALCM}$ that is ExpTime.

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

We extend the Tableau Calculus given by (Nguyen and Szalas 2009) to handle meta-modelling axioms. This calculus

$$S_0 = \{queguay, santaLucia, deRocha, delSauce\}$$

$$river^{\mathcal{I}} = River^{\mathcal{I}} = \{queguay, santaLucia\}$$
$$lake^{\mathcal{I}} = Lake^{\mathcal{I}} = \{deRocha, delSauce\}$$

$$HydrographicObject^{\mathcal{I}}$$
$$= \{river^{\mathcal{I}}, lake^{\mathcal{I}}\}$$
$$= \{\{queguay, santaLucia\}, \{deRocha, delSauce\}\}$$

Figure 3: Model for the knowledge base of Figure 2

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 $\mathsf{circular}(\mathcal{A}, \mathcal{M})$ in $(\bot_3)$ which guarantees that the canonical model is well-founded. We say that $\mathsf{circular}(\mathcal{A}, \mathcal{M})$ holds if there is a sequence of meta-modelling axioms $a_1 =_{\mathsf{m}} A_1$, $a_2 =_{\mathsf{m}} A_2$, …, $a_n =_{\mathsf{m}} A_n$ all in $\mathcal{M}$ such that $A_1(a_2)$, $A_2(a_3), \ldots, A_n(a_1)$ are in $\mathcal{A}$. In that case we say that $\mathcal{A}$ has a circularity w.r.t. $\mathcal{M}$.

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 $\mathcal{ALCM}$. The complexity of the predicate $\mathsf{circular}$ 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 $\mathcal{ALCM}$) *Consistency of a (general) knowledge base in $\mathcal{ALCM}$ is ExpTime-complete.*

$$(\bot_3)\frac{(\mathcal{T}, \mathcal{A}, \mathcal{M})}{\bot}\ \mathsf{circular}(\mathcal{A}, \mathcal{M})$$

$$(\mathsf{close})\frac{(\mathcal{T}, \mathcal{A}, \mathcal{M})}{(\mathcal{T}, \mathcal{A}', \mathcal{M}') \vee (\mathcal{T}, \mathcal{A}'', \mathcal{M})}\ \begin{array}{l}\{a, b\} \subseteq \mathsf{dom}(\mathcal{M})\\ a \neq b \notin \mathcal{A}\ \ a \neq b\end{array}$$

where $\mathcal{A}' = \mathcal{A}[a/b]$, $\mathcal{M}' = \mathcal{M}[a/b]$,$\mathcal{A}'' = \mathcal{A} \cup \{a \neq b\}$

$$(=)\frac{(\mathcal{T}, \mathcal{A}, \mathcal{M} \cup \{a =_{\mathsf{m}} A, a =_{\mathsf{m}} B\})}{(\mathcal{T}', \mathcal{A} \cup \mathcal{A}', \mathcal{M} \cup \{a =_{\mathsf{m}} A\})}$$

where $\mathcal{T}' = \mathcal{T} \cup \{(A \sqcup \neg B), (B \sqcup \neg A)\}$ and

$\mathcal{A}' = \{((A \sqcup \neg B) \sqcap (B \sqcup \neg A))(d) \mid$
$d \in \mathsf{dom}(\mathcal{A}) \cup \mathsf{dom}(\mathcal{M}) \cup \{a\}\}$

$$(\neq)\frac{(\mathcal{T}, \mathcal{A} \cup \{a \neq b\}, \mathcal{M} \cup \{a =_{\mathsf{m}} A, b =_{\mathsf{m}} B\})}{(\mathcal{T}, \mathcal{A}', \mathcal{M} \cup \{a =_{\mathsf{m}} A, b =_{\mathsf{m}} B\})}\mathsf{cond}_{(\neq)}$$

where $\mathsf{cond}_{(\neq)}$ means that

there is no d such that $(A \sqcap \neg B \sqcup \neg A \sqcap B)(d) \in \mathcal{A}$

$\mathcal{A}' = \mathcal{A} \cup \{a \neq b\} \cup \mathcal{A}''$

$\mathcal{A}'' = \{(A \sqcap \neg B \sqcup \neg A \sqcap B)(d_0)\} \cup \{C(d_0) \mid C \in \mathcal{T}\}$

for a new individual d_0

Figure 4: New Tableau Rules for Meta-modelling

5 Related Work

ExpTime tableau algorithms for checking satisfiability w.r.t. a general Tbox are shown in (De Giacomo, Donini, and Massacci 1996; Donini and Massacci 2000), which globally cache only unsatisfiable sets. An ExpTime Tableau algorithm for checking satisfiability of a concept in $\mathcal{ALC}$ w.r.t. a general Tbox that can globally cache satisfiable and unsatisfiable sets is showed in (Goré and Nguyen 2013). In (Nguyen and Szalas 2009), an ExpTime algorithm for checking consistency of a knowledge base, including a Tbox and an Abox in $\mathcal{ALC}$ is presented.

In the literature of Description Logic, there are other approaches to meta-modelling (Motik 2007; Pan, Horrocks, and Schreiber 2005; Glimm, Rudolph, and Völker 2010; Jekjantuk, Gröner, and Pan 2010; Giacomo, Lenzerini, and Rosati 2011; Homola et al. 2013; 2014; Lenzerini, Lepore, and Poggi 2014). The approaches which define fixed layers or levels of meta-modelling (Pan, Horrocks, and Schreiber 2005; Jekjantuk, Gröner, and Pan 2010; Homola et al. 2013; 2014) impose a very strong limitation to the ontology engineer. The key feature in our semantics is to interpret a and A as the same object when a and A are connected through meta-modelling, i.e., if $a =_{\mathsf{m}} A$ then $a^{\mathcal{I}} = A^{\mathcal{I}}$. This allows us to detect inconsistencies in the ontologies which is not possible under the Hilog semantics (Motik 2007; Giacomo, Lenzerini, and Rosati 2011; Homola et al. 2013; 2014; Lenzerini, Lepore, and Poggi 2014; Kubincová, Kluka, and Homola 2015).

6 Conclusions and Future Work

The current algorithm was chosen only for the theoretical purpose of proving complexity. The standard tableau algorithm with meta-modelling presented in (Motz, Rohrer, and Severi 2015) is likely to work better in practice, as some 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 $\mathcal{ALCM}$ 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

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 1999. Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI 1999)*, 84–89.

De Giacomo, G., and Lenzerini, M. 1996. TBox and ABox Reasoning in Expressive Description Logics. In *Proceedings of Description Logic Workshop*, 37–48.

De Giacomo, G.; Donini, F. M.; and Massacci, F. 1996. Exptime tableaux for ALC. In *Proceedings of Description Logic Workshop*, 107–110.

Donini, F. M., and Massacci, F. 2000. EXPTIME tableaux for ALC. *Artificial Intelligence* 124(1):87–138.

Giacomo, G. D.; Lenzerini, M.; and Rosati, R. 2011. Higher-order description logics for domain metamodeling. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011*. AAAI Press.

Glimm, B.; Rudolph, S.; and Völker, J. 2010. Integrated metamodeling and diagnosis in OWL 2. In *International Semantic Web Conference (1)*, 257–272.

Goré, R., and Nguyen, L. A. 2013. Exptime tableaux for ALC using sound global caching. *Journal of Automated Reasoning* 50(4):355–381.

Hitzler, P.; Krötzsch, M.; and Rudolph, S. 2009. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC.

Homola, M.; Kluka, J.; Svátek, V.; and Vacura, M. 2013. Towards typed higher-order description logics. In *Proceedings of the 26th International Workshop on Description Logic*, 221–233.

Homola, M.; Kluka, J.; Svátek, V.; and Vacura, M. 2014. Typed higher-order variant of SROIQ - why not? In *Informal Proceedings of the 27th International Workshop on Description Logics*, 567–578.

Jekjantuk, N.; Gröner, G.; and Pan, J. Z. 2010. Modelling and reasoning in metamodelling enabled ontologies. *International Journal Software and Informatics* 4(3):277–290.

Kubincová, P.; Kluka, J.; and Homola, M. 2015. Towards expressive metamodelling with instantiation. In *Proceedings of the 28th International Workshop on Description Logics*.

Lenzerini, M.; Lepore, L.; and Poggi, A. 2014. Making metaquerying practical for Hi(DLLiteR) knowledge bases. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences-Confederated International Conferences: CoopIS, and ODBASE 2014*, volume 8841 of *Lecture Notes in Computer Science*. Springer. 580–596.

Martinez, M.; Rohrer, E.; and Severi, P. 2015. Complexity of the Description Logic ALCM. http://arxiv.org/abs/1511.03749.

Motik, B. 2007. On the properties of metamodeling in OWL. *Journal of Logic and Computation* 17(4):617–637.

Motz, R.; Rohrer, E.; and Severi, P. 2014. Reasoning for *ALCQ* extended with a flexible meta-modelling hierarchy. In *4th Joint International Semantic Technology Conference, JIST 2014*, volume 8943 of *Lecture Notes in Computer Science*, 47–62.

Motz, R.; Rohrer, E.; and Severi, P. 2015. The description logic *SHIQ* with a flexible meta-modelling hierarchy. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*.

Nguyen, L. A., and Golinska-Pilarek, J. 2014. An ExpTime tableau method for dealing with nominals and qualified number restrictions in deciding the description logic SHOQ. *Fundamenta Informaticae - Concurrency Specification and Programming 2013* 135(4):433–449.

Nguyen, L. A., and Szalas, A. 2009. ExpTime tableaux for checking satisfiability of a knowledge base in the description logic ALC. In *First International Conference, ICCCI 2009*, volume 5796 of *Lecture Notes in Computer Science*, 437–448.

Pan, J. Z.; Horrocks, I.; and Schreiber, G. 2005. OWL FA: A metamodeling extension of OWL DL. In *OWLED*, volume 188 of *CEUR Workshop Proceedings*.

Schild, K. 1991. A correspondence theory for terminological logics: Preliminary report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 466–471.

Schmidt-Schauß, M., and Smolka, G. 1991. Attributive concept descriptions with complements. *Artificial Intelligence* 48(1):1–26.

Sedgewick, R., and Wayne, K. 2011. *Algorithms, 4th Edition*. Addison-Wesley.

Tobies, S. 2001. *Complexity results and practical algorithms for logics in knowledge representation*. Ph.D. Dissertation, LuFG Theoretical Computer Science, RWTH-Aachen, Germany.

Vidal, I. 2015. The OWL reasoner Pellet extended with Meta-modelling. Preliminary tests. http://www.cs.le.ac.uk/people/ps56/pelletM.xml.