Debugging of $\mathcal{ALC}$-Ontologies via Minimal Model Generation

Fabio Papacchini Renate A. Schmidt

School of Computer Science, The University of Manchester, {papacchf,schmidt}@cs.man.ac.uk

Abstract: This short abstract revises a procedure for the generation of minimal models for propositional modal logics, and explains how it can be used for $\mathcal{ALC}$-ontology debugging.

1 Introduction

Model generation and minimal model generation are useful for computer science tasks such as fault analysis, model checking and debugging of logical specifications [6]. For this reason, there have been studies on minimal model generation for classical and non-classical logics [1, 4, 5, 2].

[5] suggests the possibility of using minimal model generation techniques as a complementary debugging notion to the more common notion of ontology debugging such as the one discussed in [7, 3], where an ontology is considered bugged if it is incoherent. The idea in [5] is that even a coherent ontology can be considered faulty when it does not model properly the domain of interest. This can be because aspects and properties expected to hold for the domain of interest do not follow from the ontology. Formally, given an ontology $\mathcal{O}$ and a property γ that $\mathcal{O}$ is suppose to have, if $\mathcal{O} \not\models \gamma$ then $\mathcal{O}$ is faulty. In this context procedures for the generation of minimal models, similarly to test-driven software development paradigms, complement the notion of ontology debugging and help to model correctly the domain of interest. Minimal model generation procedures can be used to check whether these properties hold at any stage of the life cycle of the ontology, and then corrected based on the returned models.

This abstract presents a minimal model generation procedure for the description logic $\mathcal{ALC}$.

2 Logic and Minimality Criterion

Syntax and semantics of the description logic $\mathcal{ALC}$ are defined as usual. The minimality criterion used is the same as in [5]. The minimality criterion establishes the minimality of a model by comparing it with other models by means of a relation called *subset-simulation*. To ease the presentation of subset-simulation, we define a valuation function V as follows.

$$V(a) = \{A \in N_C \mid a \in A^{\mathcal{I}}\}$$

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{I}' = (\Delta'^{\mathcal{I}'}, \cdot'^{\mathcal{I}'})$ be two models of an $\mathcal{ALC}$ formula ϕ. A *subset-simulation* is a binary relation $S \subseteq \Delta^{\mathcal{I}} \times \Delta'^{\mathcal{I}'}$ such that for any two $a \in \Delta^{\mathcal{I}}$ and $a' \in \Delta'^{\mathcal{I}'}$, if aSa' then the following hold.

- $V(a) \subseteq V'(a')$, and
- if $(a, b) \in r^{\mathcal{I}}$, then there exist a $b' \in \Delta'^{\mathcal{I}'}$ such that $(a', b') \in r'^{\mathcal{I}'}$ and bSb'.

If there is a full subset-simulation from a model $\mathcal{I}$ to a model $\mathcal{I}'$, then $\mathcal{I}'$ subset-simulates $\mathcal{I}$ (i.e., $\mathcal{I} \leq \mathcal{I}'$).

Subset-simulation is a preorder on models, and it can be used to define the following minimality criterion. A model $\mathcal{I}$ of an $\mathcal{ALC}$ formula ϕ is *minimal modulo subset-simulation* iff for any model $\mathcal{I}'$, if $\mathcal{I}' \leq \mathcal{I}$, then $\mathcal{I} \leq \mathcal{I}'$.

To have a visual idea of the minimality criterion, Figure 1 shows two possible models of a formula. The subset-simulation relationship is represented by directed dashed line. Given the definition of the minimality criterion, the model on the left is considered minimal because it is subset-simulated by the model on the right.

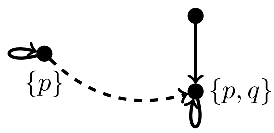

Figure 1: Example of minimality w.r.t. subset-simulation

As subset-simulation is not anti-symmetric, there can be models that subset-simulate each other, resulting in a symmetry class w.r.t. the preorder. As a result, infinitely many models minimal modulo subset-simulation can belong to the same symmetry class. To avoid the generation of all such models, and because they entail the same positive formulae, we consider a procedure to be minimal model complete if it generates at least one minimal model for each symmetry class of minimal models.

Models minimal modulo subset-simulation spread the positive information into several domain elements, while minimising the valuation function. We believe that this results in minimal models that are easier to understand.

3 Minimal Model Generation Procedure

The procedure for the generation of models minimal modulo subset-simulation for the description logic $\mathcal{ALC}$ is an adaptation of the one proposed in [5]. It is composed of a tableau calculus and a minimality test. For reasons of space we focus only on the modification of the tableau calculus. This is because while the calculus needs to be adapted to handle $\mathcal{ALC}$ ontologies, the minimality test is exactly as in [5]. Table 1 shows the rules of the calculus.

The input of the calculus is assumed to be in negation normal form and all the TBox axioms are supposed to have the form $\neg C \sqcup D$. Φ represents a disjunction of formulae, Φ^+_α a disjunction with no negated basic concept,

Table 1: Rules of the tableau calculus

$$(\forall)\ \frac{r(a,b) \qquad (\forall r.C)(a)}{C(b)}$$

$$(\alpha)\ \frac{(C_1 \sqcap \ldots \sqcap C_n)(a) \vee \Phi^+_\alpha}{\begin{array}{c} C_1(a) \vee \Phi^+_\alpha \\ \vdots \\ C_1(a) \vee \Phi^+_\alpha \end{array}}$$

$$(TBox)\ \frac{\neg C \sqcup D}{(\neg C \sqcup D)(a)}$$

where a appears on the branch.

$$(\vee)\ \frac{(C_1 \sqcup \ldots \sqcup C_n)(a) \vee \Phi}{C_1(a) \vee \ldots \vee C_n(a) \vee \Phi}$$

$$(\beta)\ \frac{\mathcal{C}(a) \vee \Phi^+}{\begin{array}{c|c} \mathcal{C}(a) & \Phi^+ \\ neg(\Phi^+) & \end{array}}$$

where $\mathcal{C}$ is of the form $\exists r.C$, $\forall r.C$, or A, and $neg(\Phi^+) = \{(\neg A)(a) \mid A(a)$ is a disjunct of $\Phi^+\}$.

$$(\exists)\ \frac{(\exists r.C)(a)}{\begin{array}{c} r(a,b) \\ C(b) \end{array}}$$

where b is fresh.

$$(SBR)\ \frac{A_1(a_1)\ \ldots\ A_n(a_n)\ r_1(b_1,c_1)\ \ldots\ r(b_m,c_m) \qquad u : (\neg A_1)(a_1) \vee \ldots \vee \neg r(b_m,c_m) \vee \Phi^+_\alpha}{\Phi^+_\alpha}$$

and Φ^+ is as Φ^+_α except that no conjunction is allowed as a disjunct.

The $(\forall)$ and $(\exists)$ rules are the common rules for description logic formulae under role restrictions. The $(TBox)$ rule is used to instantiate TBox axioms with the domain elements constituting the current domain. The (α) rule performs a lazy clausification step and, if $\Phi^+_\alpha = \top$, expands a conjunction. Due to the possibility that the property that the ontology is supposed to have can be a set of ABox statements, the calculus needs to be able of handling formulae where concepts belong to different domain elements (for example, $A(a) \sqcap B(b)$). For this reason and to have a unified way to deal with such formulae, the $(\vee)$ rule distributes the domain element over a disjunction of concepts. The (β) rule is a complement splitting rule that aims to close a branch from which a non-minimal model can be extracted. The (SBR) rule is a selection-based resolution rule. It is the only closure rule of the calculus, aims to remove all the negative information from a disjunction and, together with the Φ^+ and Φ^+_α restrictions in other rules, reduces the number of inferences resulting in non-minimal models. It is possible to note that the (SBR) rule allows the main premise to have negated role instances. This does not mean that we cover a logic where negation over roles is allowed, but that the calculus needs to handle negated role instances resulting from the negation of ABox statements such as $r(a,b)$.

The calculus in Table 1 is sound and refutationally complete. When augmented with the minimality test, we obtain a procedure that is minimal model sound and complete. Termination can be achieved by dynamic ancestor equality blocking, as proved in [5].

4 Conclusion

The abstract presented a procedure for the generation of models minimal modulo subset-simulation for $\mathcal{ALC}$ ontologies. This procedure is just a first step of a work in progress that aims to implement the procedure and its generalisations to more expressive description logics such as $\mathcal{SHI}$.

References

[1] F. Bry and A. Yahya. Positive unit hyperresolution tableaux and their application to minimal model generation. *J. Automated Reasoning*, 25(1):35–82, 2000.

[2] J. Hintikka. Model minimization - an alternative to circumscription. *J. Automated Reasoning*, 4(1):1–13, 1988.

[3] M. Horridge, B. Parsia, and U. Sattler. Extracting justifications from bioportal ontologies. *Proc. ISWC'12*, volume 7650 of *LNCS*, pages 287–299. Springer, 2012.

[4] F. Papacchini and R. A. Schmidt. A tableau calculus for minimal modal model generation. *ENTCS*, 278(3):159–172, 2011.

[5] F. Papacchini and R. A. Schmidt. Terminating minimal model generation procedures for propositional modal logics. *Proc. IJCAR'14*, volume 8562 of *LNAI*, pages 381–395. Springer, 2014.

[6] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.

[7] S. Schlobach, Z. Huang, R. Cornet, and F. van Harmelen. Debugging incoherent terminologies. *J. Automated Reasoning*, 39(3):317–349, 2007.