Modular Construction of Models

Towards a Consistency Proof for the Foundational Ontology DOLCE

Oliver KUTZ [a], Dominik LÜCKE [a], and Till MOSSAKOWSKI [a,b]
[a] *SFB/TR 8 Spatial Cognition, Bremen, Germany*
[b] *DFKI Lab Bremen, Germany*
{okutz,luecke}@informatik.uni-bremen.de
Till.Mossakowski@dfki.de

Abstract We discuss the problem of consistency proofs for large and complex first-order theories originating from the realm of ontologies. In particular, we argue that 'standard' automated reasoning methods are often insufficient for proving such consistency results.

We advocate an approach where a global model of a theory is built from smaller models together with amalgamability properties between such models. To illustrate the feasibility of this technique, we have constructed a modular version (a so-called architectural specification) of the first-order version of the foundational ontology DOLCE.

Keywords. (Relative) Consistency, (Foundational) Ontologies, Modular Reasoning

The field of formal ontology may be subdivided into the study of *domain ontologies*, devoted to specific application areas, and *foundational ontologies*, axiomatising fundamental and domain-independent concepts. Foundational ontologies, such as SUMO [Niles and Pease, 2001] and DOLCE [Gangemi et al., 2002], are typically specified in some variant of first-order logic[1], and their first-order theories tend to be rather large (DOLCE consists of a few hundred axioms, and SUMO of several thousand).

Automated and semi-automated theorem proving systems have successfully been applied to reasoning about foundational ontologies. In particular, using automated provers, a number of inconsistencies in SUMO have been found [Voronkov, 2006], and SUMO has been corrected accordingly. The problem of proving the *consistency* of ontologies, however, is much harder in general.

[1] There are, however, also versions of DOLCE that use modal logic and second-order constructs.

There are several model finders for first-order logic available. Some of them search for finite models by a translation to propositional logic (and then using SAT solvers) (e.g. `Isabelle-refute` [Weber, 2005]), some of them use more advanced methods like the model evolution calculus (e.g. `Darwin` [Baumgartner and Tinelli, 2003; Baumgartner et al., 2004]), or resolution via detecting a saturated set of clauses (e.g. SPASS [Weidenbach et al., 2002]). However, these techniques currently only suffice to find models for relatively small first-order theories—they do not scale to DOLCE, let alone SUMO.[2]

The complexity of the DOLCE ontology stems from the fact that it combines several (non-trivial) formalised ontological theories into one theory, viz. the theories of essence and identity, parts and wholes (mereology), dependence, composition and constitution, as well as properties and qualities.

In this work, we propose to construct models not in a monolithic, but in a structured way. That is, the task of constructing a model of a (large) theory should be decomposed into subtasks of finding models for smaller theories. Of course, such a decomposition is not accomplished easily. We propose to use a set of operations for model construction that have been introduced in the context of software specification under the name of architectural specifications. Basically, an architectural specification consists of a sequence of declarations of units (which are just named models), declarations of unit functions (mapping models of a smaller theory to models of an extended theory), and definition of units by unit terms. A unit term may refer to named units, apply unit functions to other units, take reducts of units, and amalgamate units to larger units. Finally, an overall result unit term yields the overall model that is provided by the architectural specification.

The semantics of architectural specifications ensures that this is done in such a way that any realisation of the declared units leads to a model corresponding to the result unit term. In particular, this means that appropriate shar-

[2] We have experimented with several model-finders to find models for the sub-theories 'classical extensional parthood' (CEP) and 'constitution' (CON) of DOLCE. These experiments were largely disillusioning. `Darwin` as well as `Paradox` are able to find trivial models with exactly one atom for both theories; here, models are generated by the atoms which themselves have no proper (temporal or spatial) parts. But this is where their power ends. Trying to find a model with 4 atoms for CEP made `Darwin` as well as `Paradox` calculate for two days at which point they terminated with error messages. We then tried `Isabelle-refute` with the SAT solver `zChaff` on CEP trying to find a model with 4 atoms. This even worked out with an enlarged variable pool for `Isabelle-refute` and setting the model size to the required one. `Isabelle-refute` was able to present a model with these settings after 5 to 10 minutes depending on the speed of the computer. Trying to find a model with 4 atoms for CON just led `Isabelle-refute` to consume about 14.8 GB of memory on a 16 GB machine while preparing the input for a SAT solver. Similarly, `Mace4` worked on this theory for several days without finding a model with 4 atoms.

ing conditions are checked; namely, if two (or more) units are amalgamated, then the shared symbols must originate from the same declared unit.

In this way, the consistency of large theories can be reduced to the consistency of a number of unit declarations. The latter amounts to consistency of smaller theories (in case of simple units) or to conservativity of theory extensions (in the case of parametrised units). Consistency of small theories can be checked with the means discussed above, while for conservativity of theory extensions, several options are available: the fact that an extension is merely an extension by definition can be checked automatically, using syntactic criteria. But often, the extension carries some looseness, allowing for several expansions of a given model of the smaller theory to the larger one. In this case, we can either try to use SMT solvers or QBF provers to prove conservativity, or we can try to describe the model expansion explicitly by a theory extension that is known to be conservative by syntactic criteria. The latter is a kind of (localised) relative consistency proof.

We have carefully analysed the DOLCE theory and have designed an architectural specification for it. In the process of this design, we had to re-arrange the architectural decomposition several times in order to find an optimal decomposition. The forces to be balanced out are the following:

- the theories of the individual units (and theory extensions, for parametrised units) should be small enough in order to keep the consistency and conservativity checks feasible;
- the theory extensions of the parametrised units must be large enough to make the conservativity checks work (that is, if a new symbol is introduced, the theory extension should contain all essential constraints for that symbol);
- the theory extensions must be large enough to guarantee the amalgamability conditions.

Indeed, the check of the amalgamability conditions has been implemented as part of the Heterogeneous Tool Set HETS [Klin et al., 2001; Mossakowski et al., 2007]. This is of great help when designing an architectural decomposition for DOLCE. Full technical details as well as the corresponding architectural specification (in notation of the Common Algebraic Specification Language CASL [CoFI (The Common Framework Initiative), 2004]) can be found in Kutz et al. [2008].

References

BAUMGARTNER, P., FUCHS, A. AND TINELLI, C. (2004). Darwin: A Theorem Prover for the Model Evolution Calculus. In S. Schulz, G. Sutcliffe and T. Tammet, eds., *IJCAR Workshop on Empirically Successful First Order Reasoning (ESFOR (aka S4))*, Electronic Notes in Theoretical Computer Science.

BAUMGARTNER, P. AND TINELLI, C. (2003). The Model Evolution Calculus. In F. Baader, ed., *CADE-19 – The 19th International Conference on Automated Deduction*, volume 2741 of *Lecture Notes in Artificial Intelligence*. Springer.

COFI (THE COMMON FRAMEWORK INITIATIVE) (2004). CASL *Reference Manual*. LNCS Vol. 2960 (IFIP Series). Springer. Available at `http://www.cofi.info`.

GANGEMI, A., GUARINO, N., MASOLO, C., OLTRAMARI, A. AND SCHNEIDER, L. . (2002). Sweetening Ontologies with DOLCE. In A. Gómez-Pérez and V. R. Benjamins, eds., *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1–4, 2002*, LNCS Vol. 2473. Springer.

KLIN, B., HOFFMAN, P., TARLECKI, A., SCHRÖDER, L. AND MOSSAKOWSKI, T. (2001). Checking Amalgamability Conditions for Architectural Specifications. In *MFCS-01: Proc. of the 26th International Symposium on Mathematical Foundations of Computer Science*. London, UK: Springer.

KUTZ, O., LÜCKE, D. AND MOSSAKOWSKI, T. (2008). Modular Construction of Models—The Consistency of DOLCE. Technical report, University of Bremen, Bremen, Germany.

MOSSAKOWSKI, T., MAEDER, C. AND LÜTTICH, K. (2007). The Heterogeneous Tool Set. In O. Grumberg and M. Huth, eds., *TACAS 2007*, volume 4424 of *Lecture Notes in Computer Science*. Springer, Heidelberg.

NILES, I. AND PEASE, A. (2001). Towards a Standard Upper Ontology. In *FOIS-01: Proc. of the International Conference on Formal Ontology in Information Systems*. New York, NY, USA: ACM.

VORONKOV, A. (2006). Inconsistencies in Ontologies. In *JELIA-06*.

WEBER, T. (2005). Bounded model generation for Isabelle/HOL. volume 125 of *Electronic Notes in Theoretical Computer Science*.

WEIDENBACH, C., BRAHM, U., HILLENBRAND, T., KEEN, E., THEOBALT, C. AND TOPIC, D. (2002). SPASS version 2.0. In A. Voronkov, ed., *Automated Deduction – CADE-18*, volume 2392 of *LNCS Vol.* . Springer. ISBN 3-540-43931-5. ISSN 0302-9743.