Representing CASL in a Proof-Theoretical Logical Framework

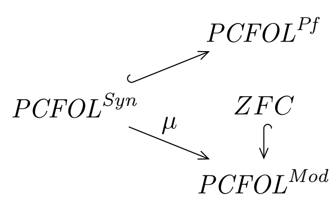
Mihai Codescu[1], Fulya Horozal[2], Iulia Ignatov[2], and Florian Rabe[2]

[1] DFKI GmbH Bremen, Germany
[2] Computer Science, Jacobs University Bremen, Germany

The Common Algebraic Specification Language CASL [9] provides a standard language for algebraic specification, expressive enough to subsume and thus unify many of the existing languages. At the basic level, the CASL logic is multi-sorted partial first-order logic with subsorting and induction. Moreover, sublogics can be obtained by restricting one or many of the features of the logics (e.g. partiality, subsorting) and the design of the CASL logic allows for language extensions (e.g. higher-order or coalgebraic extensions) which can be integrated easily. An extension of CASL to *heterogeneous* multi-logic specifications is supported by the Heterogeneous Tools Set Hets [8].

The Logic Atlas and Integrator (LATIN) project [1] develops a foundationally unconstrained framework for the representation of logics and translations between them [11]. It integrates proof-theoretical frameworks (such as LF [4]) and model-theoretic frameworks (such as institutions [3]) by giving a general definition of a logical framework, called the LATIN meta-framework [2].

The LATIN meta-framework follows a "logics-as-theories and translations-as-morphisms" approach, providing a generic construction of a logic from a theory graph in a logical framework. In particular, this has the advantage that we can use declarative logic representations in the LATIN meta-framework to automatically give rise to implementations of these logics. Thus, Hets can be extended conveniently not only on the developer's side but also directly by the user, in a significantly simplified manner.



In this work, we represent CASL as a logic in the instantiation of the LATIN meta-framework with LF [4] as a concrete representation language. Our representation follows closely the definition of the CASL logic in [9]. Firstly, we represent the syntax, proof theory and model theory of the multi-sorted version of CASL (*PCFOL*) in LF, which results in the diagram of LF theories and theory morphisms on the right. $PCFOL^{Syn}$ encodes the syntax by declaring one LF symbol for every class of expressions in CASL (e.g., sorts, functions, predicates, terms, etc.). This representation includes declaration patterns for CASL signatures, a new notion we add to LF that represents the different kind of symbol declarations allowed in signatures of a logic or a related declarative language [5]. $PCFOL^{Pf}$ encodes the proof-theory by adding one LF constant for every proof rule. $PCFOL^{Mod}$ encodes CASL models as theories of set theory ZFC, and μ interprets the CASL syntax by mapping each CASL concept in $PCFOL^{Syn}$ to a CASL model.

Secondly, we represent only the syntax of subsorting explicitly in a signature $CASL^{Syn}$ and obtain its semantics and proof theory via a translation of subsorting in *PCFOL*. We then use a logic program in the Twelf [10] implementation of LF combined with a pattern-based functor to implement the signature and expression translation from $CASL^{Syn}$ to $PCFOL^{Syn}$. The Twelf meta-theory guarantees that this translation preserves typing. The Twelf sources of our representation are available in the LATIN Logic Atlas [6].

This representation of CASL has two major benefits. Firstly, CASL extensions or variants can now be easily defined in LF and added to Hets automatically. Secondly, we can apply knowledge management services [7] (such as search, change management, querying, presentation, etc.) via the LATIN framework to CASL specifications.

References

1. M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, and F. Rabe. Project Abstract: Logic Atlas and Integrator (LATIN). In J. Davenport, W. Farmer, F. Rabe, and J. Urban, editors, *Intelligent Computer Mathematics*, volume 6824 of *Lecture Notes in Computer Science*, pages 287–289. Springer, 2011.
2. M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, F. Rabe, and K. Sojakova. Towards Logical Frameworks in the Heterogeneous Tool Set Hets. In T. Mossakowski and H. Kreowski, editors, *Recent Trends in Algebraic Development Techniques 2010*, volume 7137 of *Lecture Notes in Computer Science*, pages 139–159. Springer, 2012.
3. J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.
4. R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinery*, 40(1):143–184, 1993.
5. F. Horozal. Logic translations with declaration patterns. `https://svn.kwarc.info/repos/fhorozal/pubs/patterns.pdf`, 2012.
6. M. Kohlhase, T. Mossakowski, and F. Rabe. The LATIN Project, 2009. see `https://trac.omdoc.org/LATIN/`.
7. M. Kohlhase, F. Rabe, and V. Zholudev. Towards MKM in the Large: Modular Representation and Scalable Software Architecture. In S. Autexier, J. Calmet, D. Delahaye, P. Ion, L. Rideau, R. Rioboo, and A. Sexton, editors, *Intelligent Computer Mathematics*, volume 6167 of *Lecture Notes in Computer Science*, pages 370–384. Springer, 2010.
8. T. Mossakowski, C. Maeder, and K. Lüttich. The Heterogeneous Tool Set. In O. Grumberg and M. Huth, editor, *TACAS 2007*, volume 4424 of *Lecture Notes in Computer Science*, pages 519–522, 2007.
9. Peter D. Mosses, editor. *CASL Reference Manual.* Number 2960 in LNCS. Springer Verlag, 2004.
10. F. Pfenning and C. Schürmann. System description: Twelf - a meta-logical framework for deductive systems. *Lecture Notes in Computer Science*, 1632:202–206, 1999.
11. F. Rabe. A Logical Framework Combining Model and Proof Theory. see `http://kwarc.info/frabe/Research/rabe_combining_10.pdf`, 2010.