Mechanizing Meta-Theory in Beluga*

Brigitte Pientka

School of Computer Science, McGill University
Montreal, Canada
bpientka@cs.mcgill.ca

Abstract

Mechanizing formal systems, given via axioms and inference rules, together with proofs about them plays an important role in establishing trust in formal developments. In this talk, I will survey the proof environment Beluga. To specify formal systems and represent derivations within them, Beluga provides a sophisticated infrastructure based on the logical framework LF; in particular, its infrastructure not only supports modelling binders via binders in LF, but extends and generalizes LF with first-class contexts to abstract over a set of assumptions, contextual objects to model derivations that depend on assumptions, and first-class simultaneous substitutions to relate contexts. These extensions allow us to directly support key and common concepts that frequently arise when describing formal systems and derivations within them.

To reason about formal systems, Beluga provides a dependently typed functional language for implementing inductive proofs about derivations as recursive functions on contextual objects following the Curry-Howard isomorphism. Recently, the Beluga system has also been extended with a totality checker which guarantees that recursive programs are well-founded and correspond to inductive proofs and an interactive program development environment to support incremental proof / program construction. Taken together these extensions enable direct and compact mechanizations. To demonstrate Beluga's strength, we develop a weak normalization proof using logical relations. The Beluga system together with examples is available from http://complogic.cs.mcgill.ca/beluga/.

1998 ACM Subject Classification D.3.1 Formal Definitions and Languages, F.3.2 Semantics of Programming Languages, F.3.1 Specifying and Verifying and Reasoning about Programs

Keywords and phrases Type systems, Dependent Types, Logical Frameworks

Digital Object Identifier 10.4230/OASIcs.WPTE.2015.1

Category Invited Talk

Acknowledgements. Many of the ideas I describe in this talk are joint work with Andrew Cave. Over the past 6 years, several undergraduate students, graduate students, and postdocs have contributed to the implementation of Beluga: M. Boespflug, A. Cave, S. Cooper,A. Marchildon, F. Ferreira, O. Savary Belanger, D. Thibodeau, T. Xue.

* This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and Fonds de Recherche Nature et technologies Quebec (FQRNT).

2nd International Workshop on Rewriting Techniques for Program Transformations and Evaluation (WPTE'15).
Editors: Yuki Chiba, Santiago Escobar, Naoki Nishida, David Sabel, and Manfred Schmidt-Schauß; pp. 1–1

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany