Towards Verification of Product Lines*
(Abstract)

Don Batory

Department of Computer Science
The University of Texas at Austin
Austin, Texas, USA
batory@cs.utexas.edu

Abstract. Although mechanized proof assistants are powerful verification tools, proof development can still be difficult and time-consuming. It becomes even more challenging when proofs are needed for product lines. A *product line* is a family of similar programs. Each program is constructed by a distinct (linear) combination of features, where a *feature* or *feature module* encapsulates program fragments that are added to a program to introduce a new capability or functionality to that program.

The first part of my presentation reviews basic concepts on product lines and how programs are synthesized using feature modules. The second part addresses product line verification. I explain how proofs for product lines can be engineered using feature modules. Each module contains proof fragments which are composed to build a complete proof of correctness for each product. A product line of programming languages is considered, where each variant includes metatheory proofs verifying the correctness of its syntax and semantic definitions. The approach has been realized in the Coq proof assistant, where the proofs of each feature are independently certifiable by Coq. Proofs are composed for each language variant, where Coq mechanically verifies that the composite proofs are correct. As validation, a core calculus for Java in Coq was formalized which can be extended with any combination of casts, interfaces, or generics.

Acknowledgements. Delaware, Cook and Batory are supported by the NSF's Science of Design Project CCF 0724979.

Reference

1. Delaware, B., Cook, W.R., Batory, D.: Theorem Proving for Product Lines. Tech. Rep. TR-11-25, University of Texas at Austin, Dept. of CS (May 2011)

* This is joint work with Benjamin Delaware and William Cook [1].

M. Van Eekelen et al. (Eds.): ITP 2011, LNCS 6898, p. 1, 2011.