University of Utah
UNDERGRADUATE RESEARCH JOURNAL

LEVERAGING COMPILER INTERMEDIATE REPRESENTATION FOR MULTI- AND CROSS-LANGUAGE VERIFICATION

Jack J Garzella, Marek Baranowski, Shaobo He, (Zvonimir Rakamaric)
Department of Computer Science

Developers nowadays regularly use numerous programming languages with different characteristics and trade-offs. Unfortunately, implementing a software verifier for a new language from scratch is a large and tedious undertaking, requiring expert knowledge in multiple domains, such as compilers, verification, and constraint solving. Hence, only a tiny fraction of the used languages has readily available soft- ware verifiers to aid in the development of correct programs. In the past decade, there has been a trend of leveraging popular compiler intermediate representations (IRs), such as LLVM IR, when implementing soft- ware verifiers. Processing IR promises out-of-the-box multi- and cross- language verification since, at least in theory, a verifier ought to be able to handle a program in any programming language (and their combination) that can be compiled into the IR. In practice though, to the best of our knowledge, nobody has explored the feasibility and ease of such integration of new languages. In this paper, we provide a procedure for adding support for a new language into an IR-based verification toolflow. Using our procedure, we extend the SMACK verifier with prototypical support for 6 additional languages. We assess the quality of our extensions through several case studies, and we describe our experience in detail to guide future efforts in this area.