Abstract

Compilers perform semantics-preserving optimizations to improve the efficiency of the input code. Among them, peephole optimizations that perform local algebraic simplifications are a persistent source of bugs. This talk will present domain specific languages (DSL) in the Alive-NJ tool kit to verify both integer and floating point based peephole optimizations in the LLVM compiler. The talk will briefly highlight the challenges in handling the ambiguities in the LLVM’s semantics. A transformation expressed in these DSLs is shown to be correct by automatically encoding it as constraints whose validity is checked with help of a satisfiability modulo theories (SMT) solver. Furthermore, an optimization expressed in the DSL can be automatically translated into C++ code that is suitable for inclusion in an LLVM optimization pass. The talk will also highlight our recent results and future directions on data driven precondition inference for these optimizations, which will be useful while debugging an incorrect optimization. I will conclude the talk by briefly describing other active projects on profiling task parallel programs, approximations with Hadoop/Spark frameworks, and reasoning about cryptography software.

Bio

Santosh Nagarakatte is an Assistant Professor of Computer Science at Rutgers University. He obtained his PhD from the University of Pennsylvania in 2012. His research interests are in Hardware-Software Interfaces spanning Programming Languages, Compilers, Software Engineering, and Computer Architecture. His papers have been selected as IEEE MICRO TOP Picks papers of computer architecture conferences in 2010 and 2013. He has received the NSF CAREER Award in 2015, ACM SIGPLAN PLDI 2015 Distinguished Paper Award, and ACM SIGSOFT ICSE 2016 Distinguished Paper Award for his research on LLVM compiler verification. His papers have been selected as the 2016 SIGPLAN Research Highlights Paper and 2017 Communication of the ACM Research Highlights Paper.