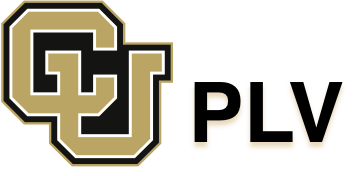
Be inspired.

Programming Languages and Verification

PhD Defense

****

**­**

Flexible Goal-Directed Abstraction

Sam Blackshear  
University of Colorado Boulder

ECAE 199

Thursday, May 28, 2015

1:00 p.m. to 2:00 p.m.

Static program analysis is a powerful technique for bug-finding, verification, and program understanding. Yet static analyses remain conspicuously absent from the toolbox of the average developer because they must abstract away details about concrete program behavior. Designing effective abstractions is tricky business: imprecise abstractions are inexpensive, but can be useless because they lose too much information about the program, whereas conventional wisdom states that precise abstractions are too expensive to scale.

This dissertation considers the problem of goal-directed static analysis, an intriguing domain where there is hope for defying the conventional wisdom about precise and scalable abstractions. In contrast to more traditional whole-program approaches that must apply precise abstractions to the entire program, goal-directed approaches have the potential to be much more tractable because they can focus the effort of the analysis on a single goal query.

There are two fundamental challenges in goal-directed analysis: (A) designing abstractions that are as flexible as possible so they can be specialized to the needs of the query and (B) using this flexibility wisely to achieve a better precision/scalability tradeoff in practice. This dissertation addresses these challenges by introducing goal-directed abstraction coarsening, a new approach to goal-directed analysis. Our approach works backward from the goal query using an abstraction that is as precise as possible by default, but can be coarsened in order to narrow the focus of the analysis and improve scalability. We meet Challenge A by introducing flexible coarsening-based techniques for store abstraction and control-flow abstraction. Our store abstraction precisely represents a small view of the store relevant to the query by combining a separation logic-based representation with a flow-insensitive points-to analysis. We present a general framework for flexible control-flow abstraction that allows the framework to coarsen the control-flow abstraction by jumping over irrelevant code. Finally, we meet Challenge B by presenting goal-directed analyses incorporating our flexible abstractions. Our analyses use programmers’ invariant-based reasoning and the structure of event-driven programs to recognize opportunities for coarsening that are likely to enhance scalability without losing precision. We have implemented these analyses and shown that they can achieve precise and scalable results for a variety of client analyses than cannot be handled effectively by previous techniques.

**Committee**: Bor-Yuh Evan Chang, Mayur Naik, Sriram Sankaranarayanan, Manu Sridharan, and Pavol Cerny.

**Sam Blackshear** earned a BA in Computer Science and Philosophy from Williams College in 2010. He is interested in static program analysis, verification, and type systems. He has interned at Microsoft Research India, Microsoft Research Redmond, and Facebook London. He is a recipient of the 2013-2014 CU CS Outstanding Research Award, the 2015 Ralph J. Slutz Excellence Award, and a 2014-2015 Facebook Fellowship. He will work on static analysis and developer tools at Facebook after graduating.

**Hosted** by Bor-Yuh Evan Chang



*­*

**