Conferinta

Joi 27 nov 2014, ora 14, Amfiteatru Stoilow

Technical presentation:

Semantics-based Program Verification using K and Matching logic
sustinuta de

Grigore Rosu,
University of Illinois at Urbana-Champaign, US

Abstract:

One of the long-lasting dreams of the programming language community is to have one formal semantic definition of a target programming language and from it to derive all the tools needed to execute and analyse programs written in that language: parsers, interpreters, compilers, symbolic execution engines, model checkers, deductive program verifiers, and so on. We believe that this is not a dream anymore. In this talk we will give an overview of K, a rewrite-based executable semantic framework in which programming languages can be defined using configurations, computations and rules. Configurations organize the state in units called cells, which are labelled and can be nested. Computations carry computational meaning as special nested list structures sequentializing computational tasks, such as fragments of program. Computations extend the original language abstract syntax.

K (rewrite) rules make it explicit which parts of the term they read-only, write-only, read-write, or do not care about. This makes K suitable for defining truly concurrent languages even in the presence of sharing. Computations are like any other terms in a rewriting environment: they can be matched, moved from one place to another, modified, or deleted. This makes K suitable for defining control-intensive features such as abrupt termination, exceptions or call/cc. The K framework is designed to allow implementing a variety of generic tools that can be used with any language defined in K. Currently, it includes the following: a parser, interpreters, symbolic execution engines (connected to the Z3 SMT solver), state-space exploration via a configurable semantic debugger, a model checker, and a deductive program verifier. The latter is based on matching logic for expressing static properties, which generalizes separation logic, and on reachability logic for expressing dynamic properties, which generalizes Hoare logic.
Non-technical presentation:

**Studying and doing research at the University of Illinois and in the US**

I will also present a few slides about the graduate program in computer science at UIUC (deadline for PhD applications is December 15!), as we as some facts about Urbana in particular and about studying in the USA in general. This part is expected to be quite informal, being driven by questions from the audience.

Short Bio:

Grigore Rosu is a professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign (UIUC), where he leads the Formal Systems Laboratory (FSL). His research interests encompass both theoretical foundations and system development in the areas of formal methods, software engineering and programming languages. Before joining UIUC in 2002, he was a research scientist at NASA Ames. He obtained his Ph.D. at the University of California at San Diego in 2000 and his M.S. at the University of Bucharest, Romania, in 1996. He was offered the CAREER award by the NSF, the Dean's award for excellence in research by the College of Engineering at UIUC in 2014, and the outstanding junior award by the Computer Science Department at UIUC in 2005. He won an ACM SIGSOFT distinguished paper award at ASE 2008 and the best software science paper award at ETAPS 2002. He was ranked a UIUC excellent teacher in Spring 2013, Fall 2012, Spring 2008 and Fall 2004.