1ST ANNUAL WORKSHOP OF SCOPES PROJECT No IZ74Z0_160453: DEVELOPING CAPACITY FOR HIGH-PRODUCTIVITY LARGE-SCALE COMPUTING

31.3.2016. - 10h - RC3 DMI

10:00-10:30

dr Srđan Škrbić

Faculty of Sciences, Novi Sad

Introduction

10:30-11:00

dr Viktor Kunčak

École Polytechnique Fédérale de Lausanne, Lausanne

Implicit Programming: Your Wish is My Command

ABSTRACT:

Programming is hard, and often requires explicitly elaborating conceptually simple steps. Among the goals of our research is to enable computers to process implicit descriptions of computational goals, such as properties or examples of the desired functionality. To make computers accept such implicit descriptions, we develop algorithms and tools for solving constraints over values that computers can process, such as numbers, sets, trees. We build on techniques from programming languages, formal methods, and automated reasoning. We hope to provide a framework to automatically glue general and domain-specific problem-solving knowledge to solve a given task. An ideal outcome are computers capable of listening to our wishes and acting upon them efficiently.

BIOGRAPHY:

Viktor Kuncak is an associate professor in the EPFL School of Computer and Communication Sciences, where, since 2007, he leads the Laboratory for Automated Reasoning and Analysis (http://lara.epfl.ch). He works in formal methods with emphasis on algorithms and tools, such as Leon (http://leon.epfl.ch). His proposal on Implicit Programming, aiming to bridge the gap between human goals and their computational realizations, was funded in 2012 by a European Research Council (ERC) starting grant. Viktor Kuncak received a PhD degree from the Massachusetts Institute of Technology (MIT) in 2007, a MSc from MIT in 2001 and a BSc degree from the University of Novi Sad, Faculty of Science, in 2000.

URL: http://lara.epfl.ch/~kuncak/

11:00-11:30

Nicolas Voirol

École Polytechnique Fédérale de Lausanne, Lausanne

Verifying Scala Programs

ABSTRACT:

The Leon framework supports contract-driven verification of pure functional programs given in a subset of the Scala language where contracts consist of quantifier-free expressions. Recent work extended Leon with support for higher-order functions, thus enabling precise handling of closures and free first-class function symbols along arbitrary program paths. However, expressiveness of

quantifier-free contracts severely limits the scope of interesting properties one can verify (or even specify!) on higher-order functions. We present two new Leon extensions that introduce support for first-order quantification and Algebraic Data Types (ADTs) with invariants.

We introduced first-order quantification into Leon through a heuristic-driven unification process that uses precise book-keeping to enable model finding for models where functions have finite range. We further identify a fragment of quantified expressions on which Leon is counter-example complete. ADT invariants are handled through synthetic contracts that ensure data types are well-formed. For non-enumerable types (such as functions), we inject the synthetic specifications at key points during reasoning to ensure all relevant invariants are enforced. Both extensions significantly increase the expressivity of contracts in Leon.

BIOGRAPHY:

Nicolas Voirol is a doctoral student in the School of Computer and Communication Sciences at EPFL. He works in the area of verification of functional programs. He completed his MSc and BSc degrees at EPFL.

Photo: http://lara.epfl.ch/~kuncak/staff/nicolas.jpg

11:30-12:00

Lidija Fodor

Faculty of Sciences, Novi Sad

Relite – R language performance improvement - challenges and problems

ABSTRACT:

The need for fast and reliable domain specific software is rapidly growing. As one of the most frequently used languages for statistical purposes, R exposes a need to follow this trend. Relite is a project, that strives to achieve significant performance improvement of R, by using modern Domain Specific Language (DSL) building principles, namely language virtualization and code generation with staging. It relies on an existing DSL, OptiML, that is built on top of Delite and Lightweight Modular Staging (LMS). We are describing the current state of the art, putting an emphasis on challenges and problems to be solved.

12:00-12:30

Željko Bajić

Faculty of Sciences, Novi Sad

DSL for information system specification - from idea to code

ABSTRACT

During the information system development process, a lot of time is spent for initial setup and development of basic functionalities. Furthermore, almost all information systems have some similarities in their structure: codebooks, basic entities and their relations, etc. For example, for all entities in an information system it is necessary to allow input and binding of data, modification of existing data and searching by various criteria. In case of large systems that contain hundreds of entities, this work takes an enormous amount of time. For these reasons there is a need for DSL, which will in an easy and simple way provide possibility to specify and generate basic source code framework of an information system.

12:30 Lunch