DOI 10.1007/s00165-016-0408-9

Formal Aspects of Computing (2017) 29: 1–2

Formal Aspects of Computing

Editorial

Maurizio Proietti[1], Hirohisa Seki[2], and Jim Woodcock[3]

[1] CNR-IASI, Via dei Taurini 19, 00185 Rome, Italy
[2] Nagoya Institute of Technology, Showa-ku, Nagoya, 466-8555 Japan
[3] University of York, York, YO10 5DD, UK

A wide range of formal methods based on logical theories of programming are used in many areas of software engineering for automating the program development process and improving the programmer's productivity. Among these methods, program synthesis, program transformation and abstract interpretation techniques take advantage of formal definitions of program semantics to guarantee the correctness of programs by construction.

This special issue includes a collection of papers that are extended and revised versions of the ones presented at the 24th International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR 2014). The LOPSTR symposia cover various topics related to logic-based program manipulation techniques and, more in general, logical theories and formal techniques for software development. LOPSTR 2014 was held in September 2014 in Canterbury, UK, co-located with the 16th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP 2014). The LOPSTR 2014 post-proceedings have been published as Volume 8981 in Springer's Lecture Notes in Computer Science (LNCS).

A total of eight papers that had been presented at LOPSTR 2014 were invited to submit significantly extended and revised versions for inclusion in this special issue. Each submission was reviewed by at least three reviewers and a final selection of six papers was made for this collection. These papers focus on various aspects of logic-based program development.

Maximal Incompleteness as Obfuscation Potency by Roberto Giacobazzi, Isabella Mastroeni and Mila Dalla Preda deals with obfuscating transformations, whose goal is to make code hard to reverse engineer and understand, with the purpose of preventing information extraction by attackers. The authors propose a formal model for specifying and understanding the strength of these transformations with respect to a given attack model. The idea is to consider the attacker as an abstract interpreter willing to extract information about the program's semantics. The authors show in a formal way that obfuscating code is making the analysis imprecise, namely making the corresponding abstract domain incomplete.

Partial Evaluation of String Obfuscations for Java Malware Detection by Aziem Chawdhary, Ranjeet Singh and Andy King shows how to apply partial evaluation to remove obfuscations that are often applied by hackers to circumvent detection of malware by Anti Virus (AV) software. Popular obfuscation techniques for Java include string obfuscation and applying reflection to hide method calls; two techniques that can either be used together or independently. The paper presents a partial evaluator for Jimple, which is an intermediate language for JVM bytecode designed for optimisation and program analysis, and demonstrates how partially evaluated Jimple code, when transformed back into Java, improves the detection rates of a number of commercial AV products.

On Proving Confluence Modulo Equivalence for Constraint Handling Rules by Henning Christiansen and Maja H. Kirkeby proposes a method for proving confluence module equivalence for Constraint Handling Rules (CHR), where the notion of confluence is generalized so that two CHR states that are not identical but equivalent according to a given equivalence relation are considered joinable. A new CHR operational semantics and a formal meta-language, MetaCHR, are developed to handle non-logical built-in predicates.

Correspondence and offprint requests to: M. Proietti, Email: maurizio.proietti@iasi.cnr.it

Constraint Logic Programming with a Relational Machine by Emilio Jesús Gallego Arias, James Lipton and Julio Mariño presents a declarative framework for the translation of constraint logic programs into variable free relational theories, which are then executed by rewriting. In this setting, term rewriting not only provides an operational semantics for logic programs, but also a simple framework for reasoning about program execution. The authors prove the translation sound, and the rewriting system complete with respect to the traditional operational semantics of constraint logic programs based on resolution.

Abstract Conjunctive Partial Deduction for the Analysis and Compilation of Coroutines by Vincent Nys and Danny De Schreye focuses on the formal analysis of the computational behavior of coroutines in logic programs. Coroutining logic programs allows the evaluation of an atom to be delayed until a specific execution context occurs. The authors show that Abstract Conjunctive Partial Deduction, a framework for (conjunctive) partial deduction based on abstract interpretation, provides an appropriate framework for the analysis and the automatic transformation of computations into new programs not requiring any support for coroutines.

Proving Completeness of Logic Programs with the Cut by Włodzimierz Drabent presents a sufficient condition for the completeness of definite logic programs with the cut. The cut is an important construct of the Prolog programming language. It prunes part of the search space, which may result in a loss of completeness. The author gives a formal definition of the operational semantics of definite programs with the cut, and shows that the proposed sufficient condition guarantees completeness with respect to this semantics.

We are very grateful to Formal Aspects of Computing for publishing this special issue, and in particular to the managing editor John Cooke for his continuous support and advice. We would like to thank all the authors who responded to our invitation. Our deep gratitude also goes to all the reviewers who did an excellent work in commenting on the submissions and helped us in the selection process. Their constructive suggestions have contributed significantly to improve the quality of the articles.

Maurizio Proietti
Hirohisa Seki
Jim Woodcock