Science of Computer Programming 111 (2015) 363–364

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Preface of the special issue on Foundational and Practical Aspects of Resource Analysis (FOPARA) 2009 & 2011

This special issue collects selected articles from the second edition of the workshop "Foundational and Practical Aspects of Resource Analysis" (FOPARA), which was held in Madrid, May 19, 2011 (co-located with the 12th Trends in Functional Programming Symposium), and the first edition, held in Eindhoven, in June 2009 (as a satellite event of the 16th Formal Methods Symposium). The aim of the FOPARA workshop series is to serve as a forum for presenting original research results that are relevant to the analysis of resource (time, space) consumption by computer programs. Moreover it aims to bring together the researchers that work on foundational issues with the researchers that focus more on practical results. The series of FOPARA started from collaboration between researchers in *implicit computational complexity* (ICC) and researchers working in applied resource analysis of *existing programming languages* such as Java or functional languages. In fact the foundational part of resource analysis, consisting in measuring and verifying time and memory consumption through formal methods and abstract computational models, is a research field inside ICC.

The exchange of ideas and methodologies inside the first two editions of FOPARA was very fruitful, the community increased, producing interesting results, in both the foundational and applied direction. This special issue, arising from an open call after the second edition of FOPARA, is a witness of the vitality of this research field.

This issue contains six articles, the first two in the foundational line, and the other four in the application-oriented line.

The first article is titled "Light Combinators for Finite Field Arithmetic". The authors advocate the idea that a functional programming approach can be efficient for implementing arithmetic. They have chosen as a test-bed the arithmetic over binary extension of finite fields, which has many important applications in cryptography. To fulfil their objectives the authors deploy a variant of the system "Dual Light Affine Logic" called Typeable Functional Assembly (TFA) having in mind programming patterns specific for the arithmetic. The typing discipline assures polynomial-time complexity.

In the second article, titled "On Bounding Space Usage of Streams Using Interpretation Analysis", the authors extend the semantic interpretation method to lazy first-order functional programs over streams. They provide interpretation criteria useful to ensure upper bounds on the size of each output stream element expressed in term of the maximal size of the input stream elements. Moreover, the authors provide the input/output upper bounds criteria for relations between the number of elements read from the input stream and the number of elements produced on the output stream.

The next two articles are examples of the application-oriented usage of methodologies developed in ICC for foundational aims. As in the previous work, the interpretation method is deployed. The content of the research is described by titles of the articles: "Space Consumption Analysis by Abstract Interpretation: Inference of Recursive Functions" and "Space Consumption Analysis by Abstract Interpretation: Reductivity Properties". The authors present an abstract-interpretation-based static analysis for inferring heap and stack memory consumption in an eager first-order functional language "Safe". Its memory management is based on heap regions instead of the more conventional approach of having a garbage collector. In the first paper the authors give abstract interpretation rules for non-recursive function definitions, and then show how the memory consumption of recursive ones is approximated. In the second paper the researchers show that the inferred bounds are reductive under certain reasonable conditions. This means that by iterating the analysis using as input the prior inferred bound, one can get tighter and tighter bounds, all of them correct. In some cases, even the exact bound is obtained.

In a certain sense, the article "A Practical Comparator of Cost Functions and Its Applications" goes in line with the reductivity results of the previous paper since it addresses the comparison of resource analysis results. Compared resource bounds are presented in the form of Cost Functions (CF) that are non-recursive expressions composed of a number of basic expressions. One is interested in automated means for comparing CF in order to prove that a CF is tighter (smaller) than or equal to another CF for all input values of interest. The authors focus on comparing CFs by exploiting their syntactic properties.

The last article "Certifying Execution Time in Multicores" can be (rather conditionally) viewed as the most application-oriented one. It presents a semantics-based program verification framework for embedded real-time critical systems that uses the worst-case execution time (WCET). The verification is performed on devices that have low computational resources.

http://dx.doi.org/10.1016/j.scico.2015.08.008

The authors propose feasible verification of programs running on multicore architectures. They extend the notion of certificates used in the abstract-carrying code (ACC) framework with a verification mechanism for linear programming (LP), using the duality to check the WCET estimates.

The design of a certifying WCET fixed-point algorithm, targeted for the ARM9 hardware platform, is presented as a particular instantiation of a compositional data-flow framework that uses abstract interpretation. The computational and algebraic properties of the data-flow framework provide program transformations that significantly increase resource utilization efficiency.

In order to create this special issue, an open call was published and distributed. Furthermore, the call was specifically sent to the authors of the 18 papers that were published in the FOPARA 2009 and 2011 proceedings. Both from FOPARA 2009 and FOPARA authors were explicitly invited to submit for this special issue an revised, extended version of their article with at least 25% new material. Six papers were received, two from FOPARA2009, two from FOPARA20011 and two open submissions. All papers were reviewed by multiple reviewers on the criteria of relevance, elegance, readability, originality and the scientific contribution. The process was very thorough and it took relatively much time incorporating, upon the request of the reviewers, multiple substantial revisions for almost every paper.

We feel that the resulting selection of articles that is the result of this process, can well illustrate the state of the art of the research in resource analysis, in particular the fruitful exchange between foundational methods and applied results. The activity of the research group is in continuous evolution, in fact further editions of FOPARA have been organized in Bertinoro, in 2013 and in London, in 2015. This last edition was co-located with the workshop “Developments in Implicit Computational Complexity” (DICE).

Finally, our first words of thanks go the authors of the papers for improving, extending, submitting, revising, resubmitting and re-revising their work again. It was well worth it!

Last but not least, we want to acknowledge the efforts of the anonymous reviewers that were willing to invest their time so extensively and repeatedly performing the reviewing process in such a very diligent manner.

Dr. Olha Shkaravska
Meertens Institute, Amsterdam, The Netherlands

Prof. dr. Simona Ronchi Della Rocca
Università di Torino, Italy

Prof. dr. Marko van Eekelen
Open Universiteit and Radboud University, The Netherlands