Fundamenta Informaticae 177 (2020) i-iii
DOI 10.3233/FI-2020-1987
IOS Press

Special Issue on the 26th International Symposium on Logic-Based Program Synthesis and Transformation: LOPSTR 2016

Preface

This special issue contains the revised and extended versions of selected papers presented at the 26th International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR 2016), held during September 6–8, 2016, at the University of Edinburgh, Scotland, UK. The aim of the LOPSTR symposium series is to stimulate and promote international research and collaboration on logic-based program development in any language paradigm, including all stages of the software life cycle.

The authors of a selection of the papers presented at the conference were invited to submit an improved, extended version to this special issue. The papers they submitted went through a careful review by qualified international referees, to whom we express our deep gratitude. The papers that were finally accepted lie in the areas of analysis, verification, synthesis, specification, transformation, and testing of software. Here is the list of the authors, titles, and overviews of the papers included in this special issue.

- Elvira Albert, Nikolaos Bezirgiannis, Frank de Boer and Enrique Martín-Martín. *A Formal, Resource Consumption-Preserving Translation from Actors with Cooperative Scheduling to Haskell*. This paper presents a formal translation of a resource-aware extension of the Abstract Behavioral Specification (ABS) language to the functional language Haskell, together with a correctness proof of the translation by means of a simulation relation between a formal semantics of the source language and a high-level operational semantics of the target language, i.e., a subset of Haskell. The experiments presented confirm resource preservation over a set of benchmarks featuring different asymptotic costs.

- María Alpuente, Daniel Pardo and Alicia Villanueva. *Abstract Contract Synthesis and Verification in the Symbolic K Framework*. This paper presents a symbolic technique for automatically inferring software contracts from programs written in a non-trivial fragment of C that supports

pointer-based structures and heap manipulation. This abstract symbolic technique axiomatically explains the execution of any (modifier) C function by using other (observer) routines in the same program. Some synthesized axioms that cannot be guaranteed to be correct by construction due to abstraction can finally be verified in the proposed setting with little effort.

- Manuel Bichler, Michael Morak and Stefan Woltran. *lpopt: A Rule Optimization Tool for Answer Set Programming*. This paper introduces lpopt, a tool that decomposes large logic programming rules into smaller rules that are easier to handle for current ASP solvers. The tool is specifically tailored to handle the standard syntax of the ASP language (ASP-Core) and makes it easier for users to write efficient and intuitive ASP programs, which would otherwise often require significant hand-tuning by expert ASP engineers.

- María Alpuente, Ángel Cuenca-Ortega, Santiago Escobar and José Meseguer. *Order-sorted Homeomorphic Embedding modulo Combinations of Associativity and/or Commutativity Axioms*. This paper generalizes the symbolic homeomorphic embedding relation to order-sorted rewrite theories that may contain various combinations of associativity and/or commutativity axioms for different binary operators. The authors systematically measure the performance of different, increasingly efficient formulations of the homeomorphic embedding relation modulo axioms that they implement in Maude.

- Moreno Falaschi, Maurizio Gabbrielli, Carlos Olarte and Catuscia Palamidessi. *Dynamic slicing for Concurrent Constraint Languages*. This paper defines a dynamic slicer for Concurrent Constraint Programming, as well as other language variants, and shows it to be a useful companion tool for the existing debugging techniques. The slicer allows marking part of the state of the computation and assists the user in eliminating most of the redundant information in order to highlight errors.

- Fred Mesnard, Étienne Payet and Germán Vidal. *Selective Unification in* (*Constraint*) *Logic Programming*. This paper builds on a previous adaptation of concolic testing to logic programming, based on *selective unification*. The authors show that the existing algorithm for selective unification is not complete in the presence of non-linear atoms, and prove soundness and completeness for a restricted version of the problem where some atoms are required to be linear. They also extend the notion of selective unification in the context of constraint logic programming.

- Paul Tarau. *Deriving Efficient Sequential and Parallel Generators for Closed Simply-Typed Lambda Terms and Normal Forms*. This paper introduces methods for deriving progressively faster sequential Prolog programs that generate and/or count the set of *closed simply-typed lambda terms* and *closed simply-typed normal forms* of sizes up to 14, greatly improving over previously known approaches. The author also devises several parallel execution algorithms, based on generating code to be uniformly distributed among the available cores, which push the counts for simply-typed terms up to size 15 and simply-typed normal forms up to size 16.

We would like to thank the editorial office of Fundamenta Informaticae, and in particular the Editor-in-Chief Damian Niwiński for his help and support in hosting this special issue. Thanks are also due to the programme committee and local organisers of LOPSTR 2016 and to the reviewers of the papers for their support and hard work.

October 2020

Guest editors:

Manuel Hermenegildo
IMDEA Software Institute and
Universidad Politécnica de Madrid (UPM)
Madrid, Spain
manuel.hermenegildo@imdea.org

Pedro López-García
IMDEA Software Institute and
Spanish Council for Scientific Research (CSIC)
Madrid, Spain
pedro.lopez@imdea.org

Alberto Pettorossi
Dipartimento di Ingegneria Civile
e Ingegneria Informatica
Università degli Studi di Roma "Tor Vergata"
Roma, Italy
pettorossi@info.uniroma2.it

Maurizio Proietti
Istituto di Analisi dei Sistemi ed Informatica
"A. Ruberti", Consiglio Nazionale delle Ricerche
Roma, Italy
maurizio.proietti@iasi.cnr.it