Fundamenta Informaticae 173 (2020) i-ii
DOI 10.3233/FI-2020-1923
IOS Press

Special Issue on the 27th International Symposium on Logic-based Program Synthesis and Transformation: LOPSTR 2017

Preface

This special issue contains revised and extended versions of papers presented at the 27th International Symposium on Logic-based Program Synthesis and Transformation – LOPSTR 2017 – which was held in Namur, Belgium on 10-12 October, 2017. The authors of a selection of the best papers from the conference were invited by the programme committee to submit a paper for this special issue of Fundamenta Informaticae.

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. Overall, the papers in this special issue provide examples of the way in which logical analysis is the basis for approaches to implementing practical tools for specifying and verifying program properties. The three papers in the special issue are the following.

- Thom Frühwirth. *Justifications in Constraint Handling Rules for Logical Retraction in Dynamic Algorithms: Theory, Implementations, and Complexity*. This paper presents a way to introduce justifications into Constraint Handling Rules (CHR) through a program transformation scheme. Justifications in a knowledge representation language track dependencies between derived information and the information from which it is derived. The scheme presented in this paper allows CHR programs to be applied to dynamic constraint satisfaction problems, where one might wish to withdraw certain conclusions by retracting their justifications.

- Michael Hanus. *Combining Static and Dynamic Contract Checking for Curry*. Contracts on functions are logical conditions relating the pre- and post-conditions that the function is required to satisfy. The paper presents an approach in which contracts are statically checked for the functional-logic programming language Curry. Dynamic checks are retained only for those conditions that are not successfully checked statically.

- Stephen Skeirik, Andrei Stefanescu and José Meseguer. *A Constructor-Based Reachability Logic for Rewrite Theories*. Reachability properties of programs are often critical to their correctness. Rewrite theories provide a generic logical framework for specifying programming languages and systems, including distributed systems. This paper builds a framework for automatic verification of reachability properties in rewrite theories, and discusses its relation to Hoare logic and LTL.

We would like to thank the editorial office of Fundamenta Informaticae, in particular editor-in-chief Damian Niwiński and Alberto Pettorossi from the Editorial Board for their help and support in hosting this special issue. Thanks are also due to the programme committee and local organisers of LOPSTR 2017 and to the reviewers of the papers for their support and hard work.

Special issue editors:

Fabio Fioravanti
University of Chieti-Pescara, Pescara, Italy

John P. Gallagher
Roskilde University, Roskilde, Spain

Maurizio Proietti
IASI-CNR, Rome, Italy

December 2019