https://doi.org/10.1007/s00165-017-0448-9
BCS © 2017
Formal Aspects of Computing (2018) 30: 1

Formal Aspects of Computing

Editorial

Ewen Denney[1], Perdita Stevens[2] and Andrzej Wąsowski [3]

[1] NASA Ames Research Center, Mountain View, CA, USA
[2] University of Edinburgh, Edinburgh, Scotland
[3] IT University of Copenhagen, Copenhagen, Denmark

Formal approaches underlie progress on many of the most fundamental aspects of software engineering. This special section contains revised and extended versions of a few of the best papers that were presented at the *19th International Conference on Fundamental Aspects of Software Engineering* (FASE) held as part of the *European Joint Conferences on Theory and Practice of Software* (ETAPS) in Eindhoven, The Netherlands, in April 2016.

FASE is well-established as one of the best software engineering conferences in the world; its position within ETAPS makes it a very attractive venue to which authors send their best work. While its scope is not limited to *formal* approaches to software engineering (the 'F' in the name deliberately denotes *fundamental*, not formal), the majority of accepted papers do involve precise definitions and formal proofs of key results. This is also the case for the five papers included in this special issue:

1. *A Formal Verification Technique for Behavioural Model-To-Model Transformations*, by de Putter and Wijs, shows how to verify that a system of rules for transforming labelled transition systems preserves a given property, for any input. Correctness is proved with the help of the Coq theorem prover.
2. *ProFeat Feature-oriented Engineering for Family-based Probabilistic Model Checking*, by Chrszon, Dubslaff, Klüppelholz, and Baier, introduces a tool that allows easier analysis of families of probabilistic systems, built on top of PRISM and extending its input language.
3. *Model-Based Testing of Probabilistic Systems*, by Gerhold and Stoelinga, shows how to derive probabilistic test cases from a requirements model and, using a new notion of conformance, proves soundness and completeness results.
4. *Cut Branches Before Looking for Bugs: Certifiably Sound Verification on Relaxed Slices*, by Léchenet, Kosmatov, and Le Gall, studies the delicate issues that arise in the use of program slicing for verification problems, and uses the Coq theorem prover to guarantee correctness.
5. *Variability-based Model Transformation: Formal Foundation and Application*, by Strüber, Rubin, Arendt, Chechik, Taentzer, and Plöger, shows how to encode a collection of similar rules into so-called variability-based rules, providing a formal basis in category theory.
6. *A Semantics Comparison Workbench for Concurrent, Asynchronous, Distributed Programs*, by Corrodi, Heußner, and Poskitt, presents a framework for comparing the effective semantics, under different execution models, of programs in a language such as SCOOP that are concurrent, asynchronous, and distributed.

We are grateful to Formal Aspects of Computing for allowing us to publish this collection of papers, and in particular Priyanka Ganesh for her editorial support in preparing the issue. We are grateful to all the Programme Committee members of FASE 2016 and to everyone else who reviewed either for the conference or for this issue. Their guidance has greatly improved the papers you read here. Last but not least, we thank the authors.

Ewen Denney
Perdita Stevens
Andrzej Wąsowski

Correspondence and offprint requests to: E. Denney, E-mail:ewen.w.denney@nasa.gov