Reversible Computing: Foundations and Software

Preface of Special Issue

Robert Glück¹ · Tetsuo Yokoyama²

Published online: 26 July 2018

© Ohmsha, Ltd. and Springer Japan KK, part of Springer Nature 2018

Reversible computing is an emerging field of computer science that has received increasing attention during the past years. The articles in this special issue reflect the broad spectrum of research on reversible computing, including topics such as programming languages and semantics, methods and algorithms, computation models and theoretical foundations. Reversible computing has derived from fundamental questions on inverting programs and computations. Inverse problems arise frequently in mathematics, science and engineering, but the corresponding problems in computer science are yet to be fully understood. This special issue aims at contributing to this long-term endeavor by providing a forum in which answers to important questions are presented in detail.

The fully reviewed articles in this special issue are arranged alphabetically and represent the latest research on topics ranging from theoretical investigations to practical studies and applications. Here is a brief summary of the six articles.

Jarkko Kari [1] surveys results of and recent developments in research on reversible cellular automata, a massively parallel computation model. The article presents techniques to construct reversible cellular automata and discusses undecidability issues and types of universality.

Kazutaka Matsuda and Meng Wang [2] tackle the classic problem of constructing parsers using grammar-based inversion as well as program composition and specialization techniques for generating efficient parsers for context-free languages from unparsers, also known as pretty printers.

Torben Ægidius Mogensen [3] investigates dynamic memory management, a central system component for supporting dynamic data structures and pattern matching in reversible functional programming languages, and experimentally evaluates a heap manager algorithm for a reversible intermediate language.

✉ Robert Glück
glueck@acm.org

¹ DIKU, Department of Computer Science, University of Copenhagen, 2100 Copenhagen, Denmark

² Department of Software Engineering, Nanzan University, Nagoya-shi 466-8673, Japan

Luca Paolini, Mauro Piccolo, and Luca Roversi [4] present a foundational study of reversible computing by exploring recursion-theoretic characterizations of primitive and partial recursive functions to eventually encompass higher-order functional programming languages.

Markus Schordan, Tomas Oppelstrup, David Jefferson, and Peter D. Barnes, Jr. [5] elaborate on an automatic source-to-source transformation to program pairs for incremental forward checkpointing and reverse execution for implementing fast parallel discrete event simulations on supercomputers.

Irek Ulidowski, Iain Phillips, and Shoji Yuen [6] survey the modeling of reversibility in concurrent computation, in particular the extension of two forms of event structures and the resulting theoretical properties, and illustrate their approach to reversibility using various examples.

We would like to thank the 14 authors for their contributions; their research is the justification for this special issue. We are also grateful to the 20 reviewers, who helped us in fully reviewing all submissions and ensuring the high quality of the selected articles. We are thankful to the past and present Editors-in-Chief of New Generation Computing, Masami Hagiya and Masayuki Numao, for the invitation to edit this special issue and to Springer for the efficient handling of the publication process and the editorial assistance, and particularly to Mio Sugino at Springer Japan. The support given by EU COST Action IC1405 is acknowledged.

References

1. Kari, J.: Reversible cellular automata: from fundamental classical results to recent developments. *New Gener. Comput.* **36**(3) (2018). <https://doi.org/10.1007/s00354-018-0034-6>
2. Matsuda, K., Wang, M.: FlipPr: a system for deriving parsers from pretty-printers. *New Gener. Comput.* **36**(3) (2018). <https://doi.org/10.1007/s00354-018-0033-7>
3. Mogensen, T.Æ.: Reversible garbage collection for reversible functional languages. *New Gener. Comput.* **36**(3) (2018). <https://doi.org/10.1007/s00354-018-0037-3>
4. Paolini, L., Piccolo, M., Roversi, L.: On a class of reversible primitive recursive functions and its Turing-complete extensions. *New Gener. Comput.* **36**(3) (2018). <https://doi.org/10.1007/s00354-018-0039-1>
5. Schordan, M., Oppelstrup, T., Jefferson, D., Barnes Jr., P.D.: Generation of reversible C++ code for optimistic parallel discrete event simulation. *New Gener. Comput.* **36**(3) (2018). <https://doi.org/10.1007/s00354-018-0038-2>
6. Ulidowski, I., Phillips, I., Yuen, S.: Reversing event structures. *New Gener. Comput.* **36**(3) (2018). <https://doi.org/10.1007/s00354-018-0040-8>