ICT COST Action IC1405

WG1 Year-End Report
COST Action IC1405 Reversible Computation

Editors: Iain Phillips and Michael Kirkedal Thomsen

June 2017

List of contributors

Kamila Barylska, Maciej Koutny, Łukasz Mikulski, Marcin Piatkowski
Tim Boykett
Robert Glück
Robin Kaarsgaard
Ivan Lanese
Torben Ægidius Mogensen
Mohammad Reza Mousavi
Rajagopal Nagarajan
Iain Phillips
Kyriaki Psara
Irek Ulidowski
German Vidal
Thomas Worsch

Contents

1 Introduction

This report covers research carried out with the aid of COST Action IC1405 on Reversible Computation during the second grant period GP2 (May 2016 to April 2017); and in particular research relating to the topics covered by Working Group WG1 Foundations. We have mostly followed the structure of the State of the Art report for WG1, but we have added the topics of 'Algebra of Reversible Circuits' and 'Theory of Programming Languages.' Note that work on Programming Languages is also to be found in the Year-End Report of WG2 Software and Systems.

Iain Phillips and Michael Kirkedal Thomsen

2 Finite-State Computing Models

Work continues but there are no new publications to report for the second grant period GP2.

3 Reversible Cellular Automata

> [MU16] Daniel Morrison and Irek Ulidowski. Direction-reversible self-timed cellular automata for delay-insensitive circuits. *J. Cellular Automata*, 12(1-2):101–120, 2016

We introduce in this paper a new Self-Timed Cellular Automaton capable of simulating reversible delay-insensitive circuits. In addition to a number of reversibility and determinism properties, our STCA exhibits direction-reversibility, where reversing the direction of a signal and running a circuit forwards is equivalent to running the circuit in reverse. We define also several extensions of the STCA which allow us to realise three larger classes of delay-insensitive circuits, including parallel circuits. We then show which of the reversibility, determinism and direction-reversibility properties hold for these classes of circuits.

Irek Ulidowski

There is now work going on (by Ville Salo and Thomas Worsch) considering block rules for asynchronous cellular automata (ACA). As a first step fully asynchronous CA are considered where in each transition an update happens at only exactly position. If one further restricts the updating to sequential sweeps from left to right (or right to left) by a reversible block rule then it is now known that a reversible synchronous CA can be realized this way if and only if its global average movement slants to the left resp. right.

For CA where any order of updates (not only strict left-to-right or right-to-left) leads to the same result we can show that surjectivity implies injectivity of the global rule (this not the case for synchronous CA in general). There has been defined a notion of a "homotopy" between two CA which has some good properties of classical homotopies in algebraic topology, and is naturally connected to asynchronicity.

Thomas Worsch

4 Algebra of Reversible Circuits

> [BKS16] Tim Boykett, Jarkko Kari, and Ville Salo. *Strongly Universal Reversible Gate Sets*, pages 239–254. Springer International Publishing, Cham, 2016

> [BKS17] Tim Boykett, Jarkko Kari, and Ville Salo. Finite generating sets for reversible gate sets under general conservation laws. *Theoretical Computer Science*, 2017. In press

It is well-known that the Toffoli gate and the negation gate together yield a universal gate set, in the sense that every permutation of $\{0,1\}^n$ can be implemented as a composition of these gates. Since every bit operation that does not use all of the bits performs an even permutation, we need to use at least one auxiliary bit to perform every permutation, and it is known that one bit is indeed enough. Without auxiliary bits, all even permutations can be implemented. We generalize these results to non-binary logic: If A is a finite set of odd cardinality then a finite gate set can generate all permutations of A^n for all n, without any auxiliary symbols. If the cardinality of A is even then, by the same argument as above, only even permutations of A^n can be implemented for large n, and we show that indeed all even permutations can be obtained from a finite universal gate set. We also consider the conservative case, that is, those permutations of A^n that preserve the weight of the input word. The weight is the vector that records how many times each symbol occurs in the word. It turns out that no finite conservative gate set can, for all n, implement all conservative even permutations of A^n without auxiliary bits. But we provide a finite gate set that can implement all those conservative permutations that are even within each weight class of A^n.

[BKS17] extends and generalises the conference paper [BKS16]. These papers answer a conjecture and develop an alternative proof for one result in the following paper, that is in the refereeing process for a mathematics journal.

> [Boy15] Tim Boykett. Closed systems of invertible maps. *CoRR*, abs/1512.06813, 2015

We generalise clones, which are sets of functions $f : A^n \to A$, to sets of maps $f : A^n \to A^m$. We formalise this and develop language that we can use to speak about such maps. In particular we look at bijective mappings, which model the logical gates of reversible computation. Reversible computation is important for physical (e.g. quantum computation) as well as engineering (e.g. heat dissipation) reasons. We generalise Toffoli's seminal work on reversible computation to multiple valued logics. In particular, we show that some restrictions Toffoli found for reversible computation on alphabets of order 2 do not apply for odd order alphabets. For A odd, we can create all invertible mappings from the Toffoli 1- and 2-gates, demonstrating that we can realise all reversible mappings from four generators. We discuss various forms of closure, corresponding to various systems of permitted manipulations. This leads, amongst other things, to discussions about ancilla bits in quantum computation.

Tim Boykett

5 Theory of Programming Languages

> [GY17] Robert Glück and Tetsuo Yokoyama. A minimalist's reversible while language. *IEICE Transactions on Information and Systems*, E100-D, 2017

The paper presents a small reversible language R-CORE, a structured imperative programming language with symbolic tree-structured data (S-expressions). The language is reduced to the core of a reversible language, with a single command for reversibly updating the store, a single reversible control-flow operator, a limited number of variables, and data with a single atom and a single constructor. Despite its extreme simplicity, the language is reversibly universal, which means that it is as powerful as any reversible language can be, while it is linear-time self-interpretable, and it allows reversible programming with dynamic data structures. The four-line program inverter for R-CORE is among the shortest existing program inverters, which demonstrates the conciseness of the language. The translator to R-CORE, which is used to show the formal properties of the language, is clean and modular, and it may serve as a model for related reversible translation problems. The goal is to provide a language that is sufficiently concise for theoretical investigations. Owing to its simplicity, the language may also be used for educational purposes.

Robert Glück

> [Mog16] Torben Ægidius Mogensen. *RSSA: a reversible SSA form*, pages 203–217. Springer, 2016

The SSA form (Static Single Assignment form) is used in compilers as an intermediate language as an alternative to traditional three-address code because code in SSA form is easier to analyse and optimize using data-flow analysis such as common-subexpression elimination, value numbering, register allocation and so on.

We introduce RSSA, a reversible variant of the SSA form suitable as an intermediate language for reversible programming languages that are compiled to reversible machine language. The main issues in making SSA reversible are the unsuitability for SSA of the reversible updates and exchanges that are traditional in reversible languages and the need for ϕ-nodes on both joins and splits of control-flow. The first issue is handled by making selected uses of a variable destroy the variable and the latter by adding parameters to labels.

We show how programs in the reversible intermediate language RIL can be translated into RSSA and discuss copy propagation, constant propagation and register allocation in the context of RSSA.

Torben Ægidius Mogensen

6 Model-based Testing

> [HMTT17] Robert M. Hierons, Mohammad Reza Mousavi, Michael Kirkedal Thomsen, and Uraz Cengiz Türker. Hardness of deriving invertible sequences from finite state machines. In *SOFSEM 2017: Theory and Practice of Computer Science - 43rd International Conference on Current Trends in Theory and Practice of Computer Science, Limerick, Ireland, January 16-20, 2017, Proceedings*, volume 10139 of *Lecture Notes in Computer Science*, pages 147–160. Springer, 2017

We have performed research on test-case generation for finite state machines. Our focus has been on making the test-case generation and more efficient by using reversible subset of transitions in a finite state machine. In particular, we have focused on developing efficient algorithms for state verification using unique input-output sequences.

Mohammad Reza Mousavi

7 Term Rewriting

Some work has been done on reversibility in term rewriting, and a paper has been submitted to a journal. This work will be described in the next Year-End Report.

German Vidal

8 Categorical models and semantics

> [KAG17] Robin Kaarsgaard, Holger Bock Axelsen, and Robert Glück. Join inverse categories and reversible recursion. *Journal of Logical and Algebraic Methods in Programming*, 87:33–50, 2017

Recently, a number of reversible functional programming languages have been proposed. Common to several of these is the assumption of totality, a property that is not necessarily desirable, and certainly not required in order to guarantee reversibility. In a categorical setting, however, faithfully capturing partiality requires handling it as additional structure. Recently, Giles studied inverse categories as a model of partial reversible (functional) programming. In this paper, we show how additionally assuming the existence of countable joins on such inverse categories leads to a number of properties that are desirable when modeling reversible functional programming, notably morphism schemes for reversible recursion, a †-trace, and algebraic ω-compactness. This gives a categorical account of reversible recursion, and, for the latter, provides an answer to the problem posed by Giles regarding the formulation of recursive data types at the inverse category level.

Robin Kaarsgaard

> [GPY17] E. Graversen, I.C.C. Phillips, and N. Yoshida. Towards a categorical representation of reversible event structures. In *Proceedings of the International Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software (PLACES 16)*, volume 246 of *EPTCS*, pages 49–60, 2017

We study categories for reversible computing, focussing on reversible forms of event structures. Event structures are a well-established model of true concurrency. There exist a number of forms of event structures, including prime event structures, asymmetric event structures, and general event structures. More recently, reversible forms of these types of event structures have been defined. We formulate corresponding categories and functors between them. We show that products and co-products exist in many cases. In most work on reversible computing, including reversible process calculi, a cause-respecting condition is posited, meaning that the cause of an event may not be reversed before the event itself. Since reversible event structures are not assumed to be cause-respecting in general, we also define cause-respecting subcategories of these event structures. Our longer-term aim is to formulate event structure semantics for reversible process calculi.

Iain Phillips

9 Petri Nets

> [BMP$^+$16] Kamila Barylska, Lukasz Mikulski, Marcin Piatkowski, Maciej Koutny, and Evgeny Erofeev. Reversing transitions in bounded Petri nets. In *Proceedings of the 25th International Workshop on Concurrency, Specification and Programming, Rostock, Germany, September 28-30, 2016*, volume 1698 of *CEUR Workshop Proceedings*, pages 74–85. CEUR-WS.org, 2016

> [BKMP16] Kamila Barylska, Maciej Koutny, Lukasz Mikulski, and Marcin Piatkowski. Reversible computation vs. reversibility in Petri nets. In *Reversible Computation - 8th International Conference, RC 2016, Bologna, Italy, July 7-8, 2016, Proceedings*, volume 9720 of *Lecture Notes in Computer Science*, pages 105–118. Springer, 2016

Papers [BMP^{+}16] and [BKMP16] contain preliminary results on the possibility of reversing the effect of the execution of Petri net transitions without retaining the previous stable states of the system or preserving the exact execution order of the transitions (but maintaining their partial order).

In [BMP^{+}16], the undecidability of the problem of maintaining the set of reachable states of a p/t-net unchanged (related to the reachability problem) has been proven. Not only this cannot always be done, but no universal method can be proposed for checking if it is feasible. At the same time, it is possible to verify the invariability of executable computations (such a problem is related to the coverability problem in p/t-nets).

In [BKMP16], the case of bounded p/t-nets has been considered. It was shown that the problem of maintaining the set of reachable states unchanged is not only decidable, but - with the use of finite sets of inverse transitions - for each instance of this problem the answer is positive. The proof is constructive, and the proposed procedure based on the original p/t-net increases (at most twice) the number of places. Potentially, the number of the constructed reverses might be large when using the proposed naive algorithm (in fact, it can be as big as the number of occurrences of a reversed transition in the reachability graph of the original p/t-net). However, the procedure can be optimised, and one can always compute a minimal set of reverses.

The above results have been obtained thanks to two successful Short Term Scientific Missions of the COST action.

Kamila Barylska, Maciej Koutny, Łukasz Mikulski, Marcin Piatkowski

[PP17] Anna Philippou and Kyriaki Psara. Reversible computation in Petri nets. Technical report, University of Cyprus, 2017

We have conducted a research that studies reversible computation in the context of Petri Nets and in particular explores the modeling of the three main strategies for reversing computation. Our aim has been to address the challenges of capturing the notions of backtracking, causal reversibility and out-of-causal-order reversibility within the Petri Net framework, thus proposing a novel, graphical methodology for studying reversible models where actions can be executed in either direction. Our approach is based on the introduction of memories for transitions as well as a special treatment of tokens, that requires them to be persistent and to retain individuality in order to allow the reversal of transitions in or out-of-causal order. The expressive power and visual nature offered by Petri Nets coupled with reversible computation has the potential of providing an attractive setting for analyzing systems. Indeed, good models that can be easily understood and simulated, even by scientists with expertise outside Computer Science, can prove very useful to understand complex systems. Furthermore, the applicability of our framework has been illustrated with an example of a biochemical system and an example of a transaction-processing system that naturally embed reversible behavior.

Kyriaki Psara

10 Process Calculi

[GLMT17] Elena Giachino, Ivan Lanese, Claudio Antares Mezzina, and Francesco Tiezzi. Causal-consistent rollback in a tuple-based language. *J. Log. Algebr. Meth. Program.*, 88:99–120, 2017

[GLMT15] Elena Giachino, Ivan Lanese, Claudio Antares Mezzina, and Francesco Tiezzi. Causal-consistent reversibility in a tuple-based language. In *PDP*, pages 467–475. IEEE Computer Society, 2015

We have studied the definition of a rollback operator in the coordination language muKlaim [GLMT17], continuing the work undertaken in [GLMT15]. The main new result is that such an operator satisfies a simple intuitive specification, namely that it is the smallest causal-consistent set of backward moves undoing the target action.

Ivan Lanese

[BDLd15] Franco Barbanera, Mariangiola Dezani-Ciancaglini, Ivan Lanese, and Ugo de'Liguoro. Retractable contracts. In *PLACES*, volume 203 of *EPTCS*, pages 61–72, 2015

[BLd17] Franco Barbanera, Ivan Lanese, and Ugo de'Liguoro. Retractable and speculative contracts. In *COORDINATION*, LNCS. Springer, 2017. to appear

[BdL16] Franco Barbanera and Ugo de' Liguoro. A game interpretation of retractable contracts. In Alberto Lluch-Lafuente and José Proença, editors, *COORDINATION*, volume 9686 of *Lecture Notes in Computer Science*, pages 18–34. Springer, 2016

[BDLd15] Franco Barbanera, Mariangiola Dezani-Ciancaglini, Ivan Lanese, and Ugo de'Liguoro. Retractable contracts. In *PLACES*, volume 203 of *EPTCS*, pages 61–72, 2015

We have continued the study of retractable contracts started in [BDLd15]. Main results [BLd17] are that both compliance and the subcontract relation are decidable in polynomial time, and that the dual of a contract always exists and has a simple syntactic characterization. Furthermore, the same results apply to a novel model of contracts featuring a speculative choice: all the options of the choice are explored concurrently, and the computation succeeds if at least one of the options is successful. In [BdL16], instead, we have proposed a three-party game-theoretic interpretation of retractable session contracts, namely the retractable contracts of [BDLd15]. In such an interpretation a client is retractable-compliant with a server if and only if there exists a winning strategy for a particular player in a game-theoretic model of contracts. Such a player can be looked at as a mediator, driving the choices in the retractable points.

Ivan Lanese

11 Formal Verification of Quantum Systems

> [WN16] David Windridge and Rajagopal Nagarajan. Quantum bootstrap aggregation. In *Quantum Interaction - 10th International Conference, QI 2016, San Francisco, CA, USA, July 20–22, 2016, Revised Selected Papers*, pages 115–121, 2016

We set out a strategy for quantizing attribute bootstrap aggregation to enable variance-resilient quantum machine learning. To do so, we utilise the linear decomposability of decision boundary parameters in the Rebentrost et al. Support Vector Machine to guarantee that stochastic measurement of the output quantum state will give rise to an ensemble decision without destroying the superposition over projective feature subsets induced within the chosen SVM implementation. We achieve a linear performance advantage, $O(d)$, in addition to the existing $O(\log(n))$ advantages of quantization as applied to Support Vector Machines. The approach extends to any form of quantum learning giving rise to linear decision boundaries.

Rajagopal Nagarajan

References

[BdL16] Franco Barbanera and Ugo de' Liguoro. A game interpretation of retractable contracts. In Alberto Lluch-Lafuente and José Proença, editors, *COORDINATION*, volume 9686 of *Lecture Notes in Computer Science*, pages 18–34. Springer, 2016.

[BDLd15] Franco Barbanera, Mariangiola Dezani-Ciancaglini, Ivan Lanese, and Ugo de'Liguoro. Retractable contracts. In *PLACES*, volume 203 of *EPTCS*, pages 61–72, 2015.

[BKMP16] Kamila Barylska, Maciej Koutny, Lukasz Mikulski, and Marcin Piatkowski. Reversible computation vs. reversibility in Petri nets. In *Reversible Computation - 8th International Conference, RC 2016, Bologna, Italy, July 7-8, 2016, Proceedings*, volume 9720 of *Lecture Notes in Computer Science*, pages 105–118. Springer, 2016.

[BKS16] Tim Boykett, Jarkko Kari, and Ville Salo. *Strongly Universal Reversible Gate Sets*, pages 239–254. Springer International Publishing, Cham, 2016.

[BKS17] Tim Boykett, Jarkko Kari, and Ville Salo. Finite generating sets for reversible gate sets under general conservation laws. *Theoretical Computer Science*, 2017. In press.

[BLd17] Franco Barbanera, Ivan Lanese, and Ugo de'Liguoro. Retractable and speculative contracts. In *COORDINATION*, LNCS. Springer, 2017. to appear.

[BMP$^+$16] Kamila Barylska, Lukasz Mikulski, Marcin Piatkowski, Maciej Koutny, and Evgeny Erofeev. Reversing transitions in bounded Petri nets. In *Proceedings of the 25th International Workshop on Concurrency, Specification and Programming, Rostock, Germany, September 28-30, 2016*, volume 1698 of *CEUR Workshop Proceedings*, pages 74–85. CEUR-WS.org, 2016.

[Boy15] Tim Boykett. Closed systems of invertible maps. *CoRR*, abs/1512.06813, 2015.

[GLMT15] Elena Giachino, Ivan Lanese, Claudio Antares Mezzina, and Francesco Tiezzi. Causal-consistent reversibility in a tuple-based language. In *PDP*, pages 467–475. IEEE Computer Society, 2015.

[GLMT17] Elena Giachino, Ivan Lanese, Claudio Antares Mezzina, and Francesco Tiezzi. Causal-consistent rollback in a tuple-based language. *J. Log. Algebr. Meth. Program.*, 88:99–120, 2017.

[GPY17] E. Graversen, I.C.C. Phillips, and N. Yoshida. Towards a categorical representation of reversible event structures. In *Proceedings of the International Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software (PLACES 16)*, volume 246 of *EPTCS*, pages 49–60, 2017.

[GY17] Robert Glück and Tetsuo Yokoyama. A minimalist's reversible while language. *IEICE Transactions on Information and Systems*, E100-D, 2017.

[HMTT17] Robert M. Hierons, Mohammad Reza Mousavi, Michael Kirkedal Thomsen, and Uraz Cengiz Türker. Hardness of deriving invertible sequences from finite state machines. In *SOFSEM 2017: Theory and Practice of Computer Science - 43rd International Conference on Current Trends in Theory and Practice of Computer Science, Limerick, Ireland, January 16-20, 2017, Proceedings*, volume 10139 of *Lecture Notes in Computer Science*, pages 147–160. Springer, 2017.

[KAG17] Robin Kaarsgaard, Holger Bock Axelsen, and Robert Glück. Join inverse categories and reversible recursion. *Journal of Logical and Algebraic Methods in Programming*, 87:33–50, 2017.

[Mog16] Torben Ægidius Mogensen. *RSSA: a reversible SSA form*, pages 203–217. Springer, 2016.

[MU16] Daniel Morrison and Irek Ulidowski. Direction-reversible self-timed cellular automata for delay-insensitive circuits. *J. Cellular Automata*, 12(1-2):101–120, 2016.

[PP17] Anna Philippou and Kyriaki Psara. Reversible computation in Petri nets. Technical report, University of Cyprus, 2017.

[WN16] David Windridge and Rajagopal Nagarajan. Quantum bootstrap aggregation. In *Quantum Interaction - 10th International Conference, QI 2016, San Francisco, CA, USA, July 20–22, 2016, Revised Selected Papers*, pages 115–121, 2016.