

1. Record Nr.	UNISA996464525403316
Titolo	Reversible computation : 13th international conference, RC 2021, virtual event, July 7-8, 2021 : proceedings // Shigeru Yamashita, Tetsuo Yokoyama (editors)
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2021] ©2021
ISBN	3-030-79837-2
Descrizione fisica	1 online resource (277 pages)
Collana	Lecture Notes in Computer Science ; ; v.12805
Disciplina	004
Soggetti	Reversible computing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Preface -- Organization -- Decision Diagrams and Reversible Computation (Abstract of Invited Talk) -- Contents -- Invited Talks -- How Can We Construct Reversible Turing Machines in a Very Simple Reversible Cellular Automaton? -- 1 Introduction -- 2 Reversible Cellular Automaton -- 2.1 Cellular Automaton and Its Reversibility -- 2.2 Triangular Partitioned Cellular Automaton (TPCA) -- 2.3 Elementary Triangular Partitioned Cellular Automaton (ETPCA), in Particular, ETPCA 0347 -- 3 Useful Patterns and Phenomena in the Reversible Cellular Space of ETPCA 0347 -- 3.1 Useful Patterns in ETPCA 0347 -- 3.2 Interacting Patterns in ETPCA 0347 to Find Useful Phenomena -- 4 Composing Reversible Logic Element with Memory -- 4.1 Reversible Logic Element with Memory (RLEM) -- 4.2 RLEM 4-31 -- 4.3 Universality of RLEMs -- 4.4 Composing RLEM 4-31 in the Reversible ETPCA 0347 -- 4.5 Comparing with the Method that Uses Reversible Logic Gates -- 5 Making Reversible Turing Machines in ETPCA 0347 -- 5.1 Reversible Turing Machine -- 5.2 Functional Modules Composed of RLEM 4-31 for RTMs -- 5.3 Constructing RTMs in ETPCA 0347 -- 6 Concluding Remarks -- References -- Variational Quantum Eigensolver and Its Applications -- 1 Introduction -- 2 Background -- 2.1 Quantum Circuits -- 2.2 The VQE Algorithm -- 3 The Proposed Problem-Specific PQCs -- 3.1 Overview of the Problem-Specific PQC -- 3.2 Problem-Specific PQCs for the TSP -- 4 Experimental Results -- 5 Conclusions

-- References -- Programming and Programming Languages --
Reversible Functional Array Programming -- 1 Introduction -- 2
Modifying for Reversibility -- 3 Parallel Implementation -- 3.1 The
Simple Cases -- 3.2 reduce -- 3.3 scanl -- 4 A Reversible Array
Programming Language -- 5 A Type System for Reversible Array
Programming -- 6 Examples -- 6.1 Inner Product.
6.2 Counting the Number of Elements that Satisfy a Predicate -- 6.3
Separation by Predicate -- 7 Conclusion and Future Work -- References
-- Compiling Janus to RSSA -- 1 Introduction -- 2 Preliminaries - Janus
and RSSA -- 2.1 Janus -- 2.2 The Reversible Intermediate Code RSSA
-- 3 Our Compiler -- 3.1 Compiler Front-End -- 3.2 Compiler Back-
End -- 3.3 Assignments and Expressions -- 3.4 If-Then-Else -- 3.5
Loops -- 3.6 Procedure Calls -- 3.7 Arrays and Stacks -- 3.8
Implementation -- 4 Results -- 5 Conclusions and Outlook --
References -- Causal-Consistent Debugging of Distributed Erlang
Programs -- 1 Introduction -- 2 Background -- 2.1 The Language
Syntax -- 2.2 The Language Semantics -- 2.3 A Reversible Semantics
-- 3 Distributed Reversible Semantics for Erlang -- 3.1 Distributed
System Semantics -- 3.2 Causality -- 3.3 Distributed Forward
Reversible Semantics -- 3.4 Distributed Backward Reversible Semantics
-- 3.5 Distributed Rollback Semantics -- 4 Distributed CauDER -- 5
Related Work and Conclusion -- References -- Towards a Unified
Language Architecture for Reversible Object-Oriented Programming --
1 Introduction -- 2 ROOPL -- 3 Design, Structure and Implementation
-- 3.1 System Architecture -- 3.2 Frontend Structure -- 3.3 Compiler
Structure -- 3.4 Program Inverters -- 3.5 Interpreter Design -- 4 Web
Interface -- 5 Conclusion -- A Binary Search Tree Example -- A.1
Example Program -- A.2 Source Code -- References -- Reversible
Concurrent Computation -- Towards a Truly Concurrent Semantics for
Reversible CCS -- 1 Introduction -- 2 CCS and Reversible CCS -- 3
Petri Nets, Unravel Nets and Reversible Unravel Nets -- 4 CCS Processes
as Unravel Nets -- 5 Conclusions and Future Works -- References --
Forward-Reverse Observational Equivalences in CCSK -- 1 Introduction
-- 2 CCSK -- 3 Syntactic Characterisation of Reachable Processes.
4 Revised Forward-Reverse Bisimilarity -- 5 Forward-Reverse Barbed
Congruence -- 6 Related Work -- 7 Conclusion and Future Work --
References -- Explicit Identifiers and Contexts in Reversible Concurrent
Calculus -- 1 Introduction: Filling the Blanks in Reversible Process
Algebras -- 2 Forward-Only Identified Calculus with Multiple Sums --
2.1 Preamble: Identifier Structures, Patterns, Seeds and Splitters -- 2.2
Identified CCS and Unicity Property -- 3 Reversible and Identified CCS
-- 3.1 Defining the Identified Reversible CCS -- 3.2 Properties: From
Concurrency to Causal Consistency and Unicity -- 3.3 Links to RCCS
and CCSK: Translations and Comparisons -- 4 Contexts, and How We
Do Not Have Congruences yet -- 5 Conclusion -- References --
Reversibility and Predictions -- 1 Introduction -- 2 Preliminary
Considerations -- 2.1 Informal Introduction to -- 2.2 State Simulation
-- 3 The Calculus -- 3.1 Syntax -- 3.2 Semantics -- 3.3 Oracle
Soundness -- 4 Weak Causal Consistency -- 5 Weak Causal
Consistency of -- 5.1 States Simulation -- 5.2 Causal Consistency of
the Traces Without G.Inform Reductions -- 5.3 Existence of a Trace
Free of G.Inform Reductions -- 6 Application: Sieve of Eratosthenes --
7 Conclusion and Future Work -- References -- Theory
and Foundations -- A Tangled Web of 12 Lens Laws -- 1 Introduction
-- 2 Summary of Lens Laws -- 2.1 Laws to Relate the get and put
Functions -- 2.2 Laws to Confine the put Function -- 3 Three Families
of Lens Laws -- 3.1 GetPut Family -- 3.2 PutGet Family -- 3.3 PutPut
Family -- 3.4 Separation of Three Families -- 4 Association Beyond

Families -- 5 Concluding Remark -- References -- Splitting Recursion Schemes into Reversible and Classical Interacting Threads -- 1 Introduction -- 2 The Driving Idea -- 3 From Recursion to Iteration -- 4 Future Work -- References.

Reversibility of Executable Interval Temporal Logic Specifications -- 1 Introduction -- 2 Background -- 3 Interval Temporal Logic -- 3.1 Interval -- 3.2 Syntax -- 3.3 Reflection -- 4 Executability, Reflection and Reversibility -- 4.1 Forward Executability -- 4.2 Backward Executability -- 4.3 Reversing the Effects of Bad Computations -- 5 Conclusion and Future Work -- References -- Circuit Synthesis -- Efficient Construction of Functional Representations for Quantum Algorithms -- 1 Introduction -- 2 Background -- 3 Representations for Quantum Algorithms -- 4 Proposed Approaches -- 4.1 General Scheme -- 4.2 Exploiting Repeating Structures -- 5 Experimental Evaluations -- 6 Conclusion -- References -- Finding Optimal Implementations of Non-native CNOT Gates Using SAT -- 1 Introduction -- 2 Background and Preliminaries -- 2.1 Quantum States and Circuits -- 2.2 SAT and Planning -- 3 Considered Problem -- 4 SAT Formulation -- 5 Experimental Results -- 5.1 Feasibility Studies -- 5.2 Non-native CNOTs on IBM Q Architectures -- 5.3 Effect on Circuit Transformation -- 6 Conclusions -- References -- Fast Swapping in a Quantum Multiplier Modelled as a Queuing Network -- 1 Introduction -- 2 The Multiplication Circuit -- 3 A Blocking Queuing Network -- 3.1 Modeling the Network -- 3.2 Modeling the Nodes -- 3.3 Non-boundary Nodes -- 3.4 Closed-Form Analysis -- 4 Discussion -- 5 Conclusion -- References -- OR-Toffoli and OR-Peres Reversible Gates -- 1 Introduction -- 2 Formalisms -- 3 Rewriting Rules -- 4 Closing Remarks -- References -- Author Index.
