

1. Record Nr.	UNINA9910592992603321
Titolo	A journey from process algebra via timed automata to model learning : essays dedicated to Frits Vaandrager on the occasion of his 60th birthday // edited by Nils Jansen, Marielle Stoelinga and Petra van den Bos
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2022] ©2022
ISBN	3-031-15629-3
Descrizione fisica	1 online resource (593 pages)
Collana	Lecture Notes in Computer Science ; ; v.13560
Disciplina	004.0151
Soggetti	Formal methods (Computer science) Computer science - Mathematics
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Preface -- Organization -- Contents -- Non-finite Axiomatisability Results via Reductions: CSP Parallel Composition and CCS Restriction -- 1 Introduction -- 2 Preliminaries -- 3 The Proof Strategy: Reduction Mappings -- 4 Axiomatisability Results for CSP Parallel Composition -- 4.1 The Languages BCCSPAp, BCCSPActp, BCCSPp and BCCSPpp -- 4.2 The Negative Result for BCCSPAp -- 4.3 The Case of BCCSPp and the Negative Result for BCCSPpp -- 4.4 The Case of BCCSPActp -- 5 The Case of Restriction -- 5.1 The Negative Result -- 5.2 The Reduction -- 6 Concluding Remarks -- References -- Operational Causality - Necessarily Sufficient and Sufficiently Necessary -- 1 Introduction -- 2 Preliminaries -- 3 Necessary and Sufficient Causes -- 3.1 Necessary Causes -- 3.2 Sufficient Causes -- 4 Finding Good Causes -- 4.1 Degrees of Sufficiency and Necessity -- 4.2 Weight-Minimal Necessary Causes -- 5 Conclusion -- References -- Axiomatizing Consciousness with Applications -- 1 Towards Consciousness -- 2 Consciousness as Discrete, Probabilistic-Computable Actor -- 3 Compound Consciousness -- 4 Self -- 5 Mindfulness: Mechanism and Application to ER -- 6 Suffering -- 6.1 Suffering as Pain -- 6.2 Suffering as Change -- 6.3 Suffering from Lack -- 7 Release: "3223379 Suffering and "3222378 Freedom -- 8 Freedom

Paradox -- 9 Layers of Consciousness -- 10 Conclusion -- References
-- Symmetric Transrationals: The Data Type and the Algorithmic Degree of its Equational Theory -- 1 Introduction -- 2 Basic Theory of Abstract Data Types -- 2.1 Expansions, Extensions, and Enlargements -- 3 The Symmetric Transfields -- 3.1 Algebra of Peripheral Numbers and their Equational Specification -- 3.2 Making a Symmetric Transfield -- 3.3 Making a Bounded Symmetric Transfield -- 4 Rationals and Transrationals -- 4.1 Transrationals and Bounded Transrationals. 4.2 Minimality of the Bounded Transrationals -- 5 One-One Degree of Equations in Symmetric Transrationals -- 5.1 The Diophantine Problem for the Rationals -- 5.2 Equational Theories -- 6 Concluding Remarks -- 6.1 Background on the Rationals and Their Enlargements -- 6.2 Practical Computer Arithmetics -- 6.3 Hybrid Systems -- References -- A Survey of Model Learning Techniques for Recurrent Neural Networks -- 1 Introduction -- 2 Recurrent Neural Networks as Language Acceptors -- 3 Model Learning for Recurrent Neural Networks -- 3.1 Answering Equivalence Queries Probably Approximately Correct -- 3.2 Equivalence Queries Inspired by Statistical Model Checking -- 3.3 An Abstraction Refinement Approach to Equivalence Queries -- 4 Explainability Beyond Regular Languages -- 5 Conclusion -- References -- Back-and-Forth in Space: On Logics and Bisimilarity in Closure Spaces -- 1 Introduction -- 2 Preliminaries -- 3 Bisimilarity for Closure Models -- 3.1 CM-bisimilarity -- 3.2 Logical Characterisation of CM-bisimilarity -- 4 CMC-bisimilarity for QdCMs -- 4.1 CMC-bisimilarity for QdCMs -- 4.2 Logical Characterisation of CMC-bisimilarity -- 5 CoPa-Bisimilarity for QdCM -- 5.1 CoPa-bisimilarity -- 5.2 Logical Characterisation of CoPa-bisimilarity -- 6 Conclusions -- References -- Better Automata Through Process Algebra -- 1 Introduction -- 2 Languages, Regular Expressions and Automata -- 2.1 Alphabets and Languages -- 2.2 Regular Expressions -- 2.3 Finite Automata -- 3 Kleene's Theorem -- 4 NFAs via Structural Operational Semantics -- 4.1 An Operational Semantics for Regular Expressions -- 4.2 Building Automata Using Acceptance and Transitions -- 4.3 Computing NFAs from Regular Expressions -- 5 Discussion -- 6 Conclusions and Directions for Future Work -- A SOS Rules for Acceptance and Transitions -- References.
Family-Based Fingerprint Analysis: A Position Paper -- 1 Introduction -- 2 Software Fingerprinting -- 2.1 Model Learning -- 2.2 A Methodology and Taxonomy for Formal Fingerprint Analysis -- 3 Family-Based Fingerprint Analysis -- 3.1 Family-Based Modeling and Analysis -- 3.2 A Framework for Family-Based Fingerprint Analysis -- 3.3 Practical and Theoretical Implications -- 4 Final Remarks -- References -- What's in School? - Topic Maps for Secondary School Computer Science -- 1 Introduction -- 2 Computer Science Learning and Teaching Standards -- 3 Topic Maps -- 4 Model Checking Unit -- 5 Discussion and Conclusion -- References -- Tree-Based Adaptive Model Learning -- 1 Introduction -- 2 Preliminaries -- 2.1 Learning with a Classification Tree -- 3 Learning Evolving Systems Incrementally -- 3.1 Correctness and Termination -- 4 Experiments -- 5 Conclusion -- A Omitted Incremental Subroutines -- B Additional Experiment Graphs -- B.1 Mutation Benchmark -- B.2 Feature-Add Benchmark -- References -- From Languages to Behaviors and Back -- 1 Introduction -- 2 Preliminaries for Behavioral Systems -- 2.1 A Semantic Point of View -- 2.2 A Technical Point of View -- 3 Systems of Procedural Transition Systems -- 3.1 A Semantic Point of View -- 3.2 A Technical Point of View -- 4 Learning Systems of Procedural Transition Systems -- 4.1 A Semantic Point of View -- 4.2 A Technical Point of View -- 5 On Behaviors and Reductions to Well-Matched Languages -- 6

Evaluation -- 6.1 Benchmark Setup -- 6.2 Results -- 7 Related Work --
8 Outlook and Future Work -- References -- The Quest for an
Adequate Semantic Basis of Dense-Time Metric Temporal Logic -- 1
Introduction -- 2 Signal Duration Calculus -- 3 Formulae
Differentiating Signal Classes -- 4 Conclusion -- References --
Equivalence Checking 40 Years After: A Review of Bisimulation Tools --
1 Introduction.
2 Retrospective of the 1980s -- 2.1 Bisimulation Tools Based on Term
Rewriting -- 2.2 Algorithms for Bisimulations on Finite-State Systems
-- 2.3 Early Bisimulation Tools -- 3 Retrospective of the 1990s -- 3.1
New Algorithms for Bisimulations -- 3.2 Algorithms for On-the-Fly
Verification -- 3.3 Algorithms for Symbolic Verification -- 3.4
Algorithms for Compositional Verification -- 3.5 Enhanced Bisimulation
Tools -- 3.6 New Bisimulation Tools -- 3.7 Bisimulation Tools for
Timed and Hybrid Systems -- 3.8 Bisimulation Tools for Probabilistic
and Stochastic Systems -- 3.9 Bisimulation Tools for Mobile Systems --
4 Retrospective of the 2000s -- 4.1 New Algorithm for Strong
Bisimulation -- 4.2 New Bisimulation Tools -- 4.3 Bisimulation Tools
Using On-the-Fly Verification -- 4.4 Bisimulation Tools Based on
Compositional Verification -- 4.5 Bisimulation Tools Based on
Parallel/Distributed Computing -- 4.6 Bisimulation Tools Based on
Symbolic Verification -- 4.7 Bisimulation Tools for Timed Systems --
4.8 Bisimulation Tools for Probabilistic and Stochastic Systems -- 5
Retrospective of the 2010s -- 5.1 New Bisimulation Tools -- 5.2
Bisimulation Tools for Probabilistic and Stochastic Systems -- 5.3
Bisimulation Tools for Mobile Systems -- 5.4 Bisimulation Tools Based
on Parallel/Distributed Computing -- 5.5 Bisimulation Tools Based on
Compositional Verification -- 5.6 Recent Results for Strong and
Branching Bisimulations -- 6 Conclusion -- References -- Apartness
and Distinguishing Formulas in Hennessy-Milner Logic -- 1
Introduction -- 2 Bisimulation and Apartness for LTSs -- 2.1
Hennessy-Milner Logic for Bisimulation -- 3 Weak Bisimulation and
Apartness for LTSs -- 3.1 Hennessy-Milner Logic for Weak Bisimulation
-- 4 Branching Bisimulation and Apartness for LTSs -- 4.1 Hennessy-
Milner Logic for Branching Bisimulation -- 4.2 Examples.
4.3 Related and Further Work -- References -- Playing Wordle with
Uppaal Stratego -- 1 Introduction -- 2 Wordle -- 3 Uppaal Stratego --
4 Wordle in Uppaal Stratego -- 5 Evaluation -- 6 Conclusion --
References -- Using the Parallel ATerm Library for Parallel Model
Checking and State Space Generation -- 1 Introduction -- 2 ATerm
Library -- 3 State Space Generation -- 4 Model Checking via Boolean
Equation Systems -- 5 Conclusion -- References -- Active Automata
Learning as Black-Box Search and Lazy Partition Refinement -- 1
Introduction -- 2 Regular Languages and Automata -- 3 MAT Learning
-- 4 The L Algorithm -- 5 Demonstration -- 6 Evaluation -- 7
Conclusion -- References -- A Reconstruction of Ewens' Sampling
Formula via Lists of Coins -- 1 Introduction -- 2 The Challenge -- 3
Going Probabilistic -- 4 Adding a Mutation Rate Parameter -- 5 The
Lengths of Coin Sequences -- 6 Multisets Instead of Lists -- References
-- .26em plus .1em minus .1emRooted Divergence-Preserving
Branching Bisimilarity is a Congruence: A Simpler Proof -- 1
Introduction -- 2 Finite-State CCS and Branching Bisimulation -- 3
Congruence for Finite-State Processes -- 4 Comparison with the
Congruence Proof for a Traditional Definition -- 5 Extending the Proof
to Full CCS? -- References -- Learning Language Intersections -- 1
Introduction -- 2 Strategies -- 2.1 The Word-by-Word Strategy -- 2.2
The Independent Strategy -- 2.3 The Machine-by-Machine Strategy --
3 Prototype Implementation -- 3.1 The Membership Oracle Signature

-- 3.2 An Adequate Equivalence Oracle -- 4 Empirical Evaluation -- 4.1
Setup -- 4.2 Summary -- 4.3 Detailed Results -- 5 Conclusion and
Outlook -- References -- Runtime Verification of Compound
Components with ComMA -- 1 Introduction -- 2 Interface Modeling
and Monitoring -- 2.1 ComMA Interface Modeling -- 2.2 Interface
Monitoring -- 3 Component Models.
3.1 Components with Functional Constraints.
