

1. Record Nr.	UNISA996418283503316
Autore	Goubault-Larrecq Jean
Titolo	Foundations of Software Science and Computation Structures [[electronic resource]] : 23rd International Conference, FOSSACS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25–30, 2020, Proceedings // edited by Jean Goubault-Larrecq, Barbara König
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2020
ISBN	3-030-45231-X
Edizione	[1st ed. 2020.]
Descrizione fisica	1 online resource (XV, 644 p. 1 illus.)
Collana	Theoretical Computer Science and General Issues, , 2512-2029 ; ; 12077
Disciplina	511.3
Soggetti	Mathematical logic Computer science—Mathematics Discrete mathematics Compilers (Computer programs) Computer programming Logic programming Computer engineering Computer networks Mathematical Logic and Foundations Discrete Mathematics in Computer Science Compilers and Interpreters Programming Techniques Logic in AI Computer Engineering and Networks
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di contenuto	Intro -- ETAPS Foreword -- Preface -- Organization -- Contents -- Neural Flocking: MPC-based Supervised Learning of Flocking Controllers -- 1 Introduction -- 2 Background -- 2.1 Model-Predictive Control -- 2.2 Declarative Flocking -- 3 Additional Control Objectives

-- 4 Neural Flocking -- 4.1 Training Distributed Flocking Controllers

-- 5 Experimental Evaluation -- 5.1 Preliminaries -- 5.2 Results for Basic Flocking -- 5.3 Results for Obstacle and Predator Avoidance -- 5.4 DNC Generalization Results -- 5.5 Statistical Model Checking Results -- 6 Related Work -- 7 Conclusions -- References -- On Well-Founded and Recursive Coalgebras -- 1 Introduction -- 2 Preliminaries -- 2.1 Algebras and Coalgebras. -- 2.2 Preservation Properties. -- 2.3 Factorizations. -- 2.4 Chains. -- 3 Recursive Coalgebras -- 4 The Next Time Operator and Well-Founded Coalgebras -- 5 The General Recursion Theorem and its Converse -- 6 Closure Properties of Well-founded Coalgebras -- 7 Conclusions -- References -- Timed Negotiations* -- 1 Introduction -- 2 Negotiations: Definitions and Brexit example -- 3 Timed Negotiations -- 4 High level view of the main results -- 5 Deterministic Negotiations -- 6 Sound Negotiations -- 7 k-Layered Negotiations -- 7.1 Algorithmic properties -- 7.2 Minimal Execution Time -- 8 Conclusion -- References -- Cartesian Difference Categories -- 1 Introduction -- 2 Cartesian Differential Categories -- 2.1 Cartesian Left Additive Categories -- 2.2 Cartesian Differential Categories -- 3 Change Action Models -- 3.1 Change Actions -- 3.2 Change Action Models -- 4 Cartesian Difference Categories -- 4.1 Infinitesimal Extensions in Left Additive Categories -- 4.2 Cartesian Difference Categories -- 4.3 Another look at Cartesian Differential Categories -- 4.4 Cartesian Difference Categories as Change Action Models -- 4.5 Linear Maps and -Linear Maps.

5 Examples of Cartesian Difference Categories -- 5.1 Smooth Functions -- 5.2 Calculus of Finite Differences -- 5.3 Module Morphisms -- 5.4 Stream calculus -- 6 Tangent Bundles in Cartesian Difference Categories -- 6.1 The Tangent Bundle Monad -- 6.2 The Kleisli Category of T -- 7 Conclusions and Future Work -- References -- Contextual Equivalence for Signal Flow Graphs -- 1 Introduction -- 2 Background: the Affine Signal Flow Calculus -- 2.1 Syntax -- 2.2 String Diagrams -- 2.3 Denotational Semantics and Axiomatisation -- 2.4 Affine vs Linear Circuits -- 3 Operational Semantics for Affine Circuits -- 3.1 Trajectories -- 4 Contextual Equivalence and Full Abstraction -- 4.1 From Polynomial Fractions to Trajectories -- 4.2 Proof of Full Abstraction -- 5 Functional Behaviour and Signal Flow Graphs -- 6 Realisability -- 7 Conclusion and Future Work -- References -- Parameterized Synthesis for Fragments of First-Order Logic over Data Words -- 1 Introduction -- 2 Preliminaries -- 3 Parameterized Synthesis Problem -- 4 FO[] and Parameterized Vector Games -- 4.1 Satisfiability and Normal Form for FO[] -- 4.2 From Synthesis to Parameterized Vector Games -- 5 Results for FO[] via Parameterized Vector Games -- 6 Conclusion -- References -- Controlling a random population -- 1 Introduction -- 2 The stochastic control problem -- 3 The sequential ow problem -- 4 Reduction of the stochastic control problem to the sequential flow problem -- 5 Computability of the sequential ow problem -- 6 Conclusions -- Acknowledgements -- References -- Decomposing Probabilistic Lambda-Calculi -- 1 Introduction -- 1.1 Related Work -- 2 The Probabilistic Event - Calculus λ^{PE} -- 3 Properties of Permutative Reduction -- 4 Confluence -- 4.1 Parallel Reduction and Permutative Reduction -- 4.2 Complete Reduction -- 5 Strong Normalization for Simply-Typed Terms. 6 Projective Reduction -- 7 Call-by-value Interpretation -- 8 Conclusions and Future Work -- Acknowledgments -- References -- On the k-synchronizability of Systems -- 1 Introduction -- 2 Preliminaries -- 3 k-synchronizable Systems -- 4 Decidability of Reachability for k-synchronizable Systems -- 5 Decidability of k-

synchronizability for Mailbox Systems -- 6 k-synchronizability for Peer-to-Peer Systems -- 7 Concluding Remarks and Related works -- References -- General Supervised Learning as Change Propagation with Delta Lenses -- 1 Introduction -- 2 Background: Update propagation and delta lenses -- 2.1 Why deltas. -- 2.2 Consistency restoration via update propagation: An Example -- 2.3 Update propagation and update policies -- 2.4 Delta lenses -- 3 Asymmetric Learning Lenses with Amendments -- 3.1 Does Bx need categorical learning? -- 3.2 Ala-lenses -- 4 Compositionality of ala-lenses -- 4.1 Compositionality of update policies: An example -- 4.2 Sequential composition of ala-lenses -- 4.3 Parallel composition of ala-lenses -- 4.4 Symmetric monoidal structure over ala-lenses -- 4.5 Functoriality of learning in the delta lens setting -- 5 Related work -- 6 Conclusion -- A Appendices -- A.1 Category of parameterized functors pCat -- A.2 Ala-lenses as categorification of ML-learners -- References -- Non-idempotent intersection types in logical form -- Introduction -- 1 Notations and preliminary definitions -- 2 The relational model of the λ -calculus -- 3 The simply typed case -- 3.1 Why do we need another system? -- 3.2 Minimal LJ(I) -- 3.3 Basic properties of LJ(I) -- 3.4 Relation between intersection types and LJ(I) -- 4 The untyped Scott case -- 4.1 Formulas -- 5 Concluding remarks and acknowledgments -- References -- On Computability of Data Word Functions Defined by Transducers -- 1 Introduction -- 2 Data Words and Register Transducers.

2.1 Register Transducers -- 2.2 Technical Properties of Register Automata -- 3 Functionality, Equivalence and Composition of NRT -- 4 Computability and Continuity -- 5 Test-free Register Transducers -- References -- Minimal Coverability Tree Construction Made Complete and Efficient -- 1 Introduction -- 2 Covering abstractions -- 2.1 Petri nets: reachability and covering -- 2.2 Abstraction and acceleration -- 3 A coverability tree algorithm -- 3.1 Specification and illustration -- 3.2 Correctness Proof -- 4 Tool and benchmarks -- 5 Conclusion -- References -- Constructing Infinitary Quotient-Inductive Types -- 1 Introduction -- 2 QW-types -- 3 Quotient-inductive types -- 3.1 General QIT schemas -- 4 Construction of QW-types -- 5 Conclusion -- References -- Relative full completeness for bicategorical cartesian closed structure -- 1 Introduction -- 2 Cartesian closed bicategories -- 2.1 Bicategories -- 2.2 fp-Bicategories -- 2.3 Cartesian closed bicategories -- 3 Bicategorical glueing -- 4 Cartesian closed structure on the glueing bicategory -- 4.1 Finite products in $gl(J)$ -- 4.2 Exponentials in $gl(J)$ -- 5 Relative full completeness -- References -- A duality theoretic view on limits of finite structures -- 1 Introduction -- 2 Preliminaries -- 2.1 Stone-Priestley duality -- 2.2 Stone duality and logic: type spaces -- 2.3 Duality and logic on words -- 3 The space -- 3.1 The algebraic structure on \mathcal{M} -- 3.2 The retraction $[0, 1]$ -- 4 Spaces of measures valued in \mathcal{M} and in $[0, 1]$ -- 5 The \mathcal{M} -valued Stone pairing and limits of finite structures -- 5.1 The \mathcal{M} -valued Stone pairing and logic on words -- 5.2 Limits in the spaces of measures -- 6 The logic of measures -- Conclusion -- References -- Correctness of Automatic Differentiation via Diffeologies and Categorical Gluing -- 1 Introduction -- 2 A simple forward-mode AD translation.

3 Semantics of differentiation -- 4 Extending the language: variant and inductive types -- 5 Categorical analysis of forward AD and its correctness -- 6 A continuation-based AD algorithm -- 7 Discussion and future work -- References -- Deep Induction: Induction Rules for (Truly) Nested Types* -- 1 Introduction -- 2 The Key Idea -- 3 Extending to Nested Types -- 4 Theoretical Foundations -- 4.1 Categorical Preliminaries -- 4.2 Syntax and Semantics of ADTs -- 4.3

Induction Rules for ADTs -- 4.4 Syntax and Semantics of Nested Types -- 5 The General Methodology -- 6 Related Work and Directions for Further Investigation -- References -- Exponential Automatic Amortized Resource Analysis* -- 1 Introduction -- 2 Language and Cost Semantics -- 3 Automatic Amortized Resource Analysis -- 4 Exponential Potential -- 5 Mixed Potential -- 6 Exponentials, Polynomials, and Logarithms -- 7 Conclusion and Future Work -- References -- Concurrent Kleene Algebra with Observations: from Hypotheses to Completeness -- 1 Introduction -- 2 Preliminaries -- 3 Pomset contexts -- 4 Concurrent Kleene Algebra with Hypotheses -- 4.1 Reification -- 4.2 Factoring the exchange law -- 4.3 Lifting -- 5 Instantiation to CKA with Observations -- 6 Discussion -- References -- Graded Algebraic Theories -- 1 Introduction -- 2 Preliminaries -- 2.1 Enriched Category Theory -- 2.2 Graded Monads -- 2.3 Day Convolution -- 2.4 Categories Enriched in a Presheaf Category -- 3 Graded Algebraic Theories -- 3.1 Equational Logic -- 3.2 Free Models -- 3.3 Examples -- 4 Graded Lawvere Theories -- 5 Equivalence -- 5.1 Graded Algebraic Theories and Graded Lawvere Theories -- 5.2 Graded Lawvere theories and Finitary Graded Monads -- 6 Combining Effects -- 6.1 Sums -- 6.2 Tensor Products -- 7 Related Work -- 8 Conclusions and Future Work -- References.

A Curry-style Semantics of Interaction: From untyped to second-order lazy -calculus.

Sommario/riassunto

This open access book constitutes the proceedings of the 23rd International Conference on Foundations of Software Science and Computational Structures, FOSSACS 2020, which took place in Dublin, Ireland, in April 2020, and was held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020. The 31 regular papers presented in this volume were carefully reviewed and selected from 98 submissions. The papers cover topics such as categorical models and logics; language theory, automata, and games; modal, spatial, and temporal logics; type theory and proof theory; concurrency theory and process calculi; rewriting theory; semantics of programming languages; program analysis, correctness, transformation, and verification; logics of programming; software specification and refinement; models of concurrent, reactive, stochastic, distributed, hybrid, and mobile systems; emerging models of computation; logical aspects of computational complexity; models of software security; and logical foundations of data bases.
