1. Record Nr. UNISA996247344303316

Autore	NOVARETTI, Simona
Titolo	Che il passato serva il presente : tutela giuridica dei beni culturali e partecipazione pubblica nella Repubblica popolare cinese / Simona Novaretti
Pubbl/distr/stampa	Napoli : Edizioni scientifiche italiane, 2017
ISBN	978-88-495-3479-5
Descrizione fisica	343 p. ; 24 cm
Collana	Collana di studi sino-italiani ; 2
Disciplina	344.51094
Soggetti	Beni culturali - Tutela - Cina
Collocazione	XXIV.3.S. 213
Lingua di pubblicazione	Italiano
Formato	Materiale a stampa
Livello bibliografico	Monografia

2. Record Nr. UNISA996503466903316

Titolo	Trends in functional programming : 23rd International Symposium, TFP 2022, virtual event, March 17-18, 2022, revised selected papers / / edited by Wouter Swierstra, Nicolas Wu
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2022] ©2022
ISBN	3-031-21314-9
Descrizione fisica	1 online resource (200 pages)
Collana	Lecture Notes in Computer Science ; ; v.13401
Disciplina	910.5
Soggetti	Functional programming (Computer science)
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 -- Project Paper: Embedding Generic Monadic Transformer into Scala -- 1 Introduction -- 2 Embedding Generic Monadic Cps Transform into Scala -- 2.1 Monads Parametrization -- 2.2 Translation of Higher-Order Functions -- 2.3 Call-Chain Substitutions -- 2.4 Automatic Coloring -- 3 Related Work -- 4 Conclusion and Further Work -- References -- Towards a Language for Defining Reusable Programming Language Components -- 1 Introduction -- 2 CS by Example -- 2.1 Data Types and Functions -- 2.2 Effects and Handlers -- 2.3 Order of Evaluation, Suspension, and Enactment -- 2.4 Modules and Imports -- 2.5 Composable Data Types and Functions -- 3 Defining Reusable Language Components in CS -- 3.1 A Signature for Reusable Components -- 3.2 A Language Component for Arithmetic Expressions -- 3.3 Implementing Functions as a Reusable Effect -- 3.4 Example Usage -- 4 Related Work -- 5 Future Work -- 6 Conclusion -- References -- Deep Embedding with Class -- 1 Introduction -- 1.1 Research Contribution -- 2 Deep Embedding -- 3 Shallow Embedding -- 3.1 Tagless-Final Embedding -- 4 Lifting the Backends -- 5 Existential Data Types -- 5.1 Unbraiding the Semantics from the Data -- 6 Transformation Semantics -- 6.1 Convolution -- 6.2 Pattern Matching -- 6.3 Chaining Semantics -- 7 Generalised Algebraic Data Types -- 8 Conclusion -- 9 Related Work -- A Appendix -- A.1 Data Type Definitions -- A.2 Smart Constructors --

A.3 Semantics Classes and Data Types -- A.4 GDict instances -- A.5 Evaluator Instances -- A.6 Printer Instances -- A.7 Optimisation Instances -- References -- Understanding Algebraic Effect Handlers via Delimited Control Operators -- 1 Introduction -- 2 S0: A Calculus of Shift0 and Dollar -- 3 h: A Calculus of Effect Handlers -- 4 Adding and Removing the Return Clause -- 4.1 Translating Between Dollar and Reset0.
4.2 Translating Between Handlers with and Without Return Clause -- 4.3 Correctness -- 5 Deriving a CPS Translation -- 5.1 CPS Translation of Shift0 and Dollar -- 5.2 CPS Translation of Effect Handlers -- 5.3 Macro Translation from h to S0 -- 5.4 Composing Macro and CPS Translations -- 6 Deriving a Type System from the CPS Translation -- 6.1 Type System of Shift0 and Dollar -- 6.2 Type System of Effect Handlers -- 6.3 Applying the CPS Approach -- 6.4 Soundness -- 7 Related Work -- 8 Conclusion and Future Perspectives -- References -- Reducing the Power Consumption of IoT with Task-Oriented Programming -- 1 Introduction -- 1.1 Research Contribution -- 2 Task-Oriented Programming -- 2.1 mTask -- 3 Energy Efficient IoT Nodes -- 4 Scheduling Tasks Efficiently -- 4.1 Evaluation Interval -- 4.2 Basic Refresh Rates -- 4.3 Deriving Refresh Rates -- 4.4 User Defined Refresh Rates -- 5 Running Tasks on Interrupts -- 6 Implementing Refresh Rates -- 7 Scheduling Tasks -- 8 Resulting Power Reductions -- 8.1 Power Saving -- 9 Related Work -- 10 Conclusion -- References -- Semantic Equivalence of Task-Oriented Programs in TopHat -- 1 Introduction -- 2 The TopHat Language -- 2.1 Editors -- 2.2 Sequential Composition -- 2.3 Parallel Composition -- 2.4 Observations -- 2.5 Semantics -- 3 Contextual Equivalence -- 4 Expression Equivalence -- 5 Task Conditions -- 5.1 Failing Tasks -- 5.2 Finished Tasks -- 5.3 Stuck Tasks -- 5.4 Running Tasks -- 6 Task Equivalence -- 7 Laws on Tasks -- 8 Conclusions -- A Appendix -- A.1 Failing -- A.2 Inputs -- A.3 Value -- References -- Algorithm Design with the Selection Monad -- 1 Introduction -- 2 Selection Functions -- 2.1 Pairs of Selection Functions -- 2.2 Password Example -- 2.3 Iterated Product of Selection Functions -- 2.4 Dependently Typed Version -- 2.5 Extended Password Example -- 2.6 Selection Functions Form a Monad.
2.7 History-Dependent Product of Selection Functions -- 2.8 Efficiency Drawbacks of This Implementation -- 3 Greedy Algorithms -- 4 Examples of Greedy Algorithms -- 4.1 Password Example -- 4.2 Prim's Algorithm -- 4.3 Greedy Graph Walking -- 5 Correctness -- 6 Limited Lookahead -- 6.1 Graph Example -- 7 Iterated Product -- 7.1 Examples -- 8 Conclusion and Future Work -- A Proof that Product Equals Sequence -- References -- Sound and Complete Type Inference for Closed Effect Rows -- 1 Introduction -- 2 Motivation -- 2.1 Algebraic Effects and Handlers -- 2.2 Naive Hindley-Milner Type Inference with Algebraic Effects and Handlers -- 2.3 Type Inference with open and close -- 3 Implicitly Typed Calculus for Algebraic Effects and Handlers -- 3.1 Syntax -- 3.2 Declarative Type Inference Rules -- 4 Syntax-Directed Type Inference Rules -- 4.1 Type Substitution -- 4.2 Type Ordering -- 4.3 Inference Rules with open and close -- 4.4 Principles on the Use of [let] Rule -- 4.5 Fragility of [let] Rule -- 5 Type Inference Algorithm -- 5.1 Auxiliary Functions -- 5.2 Unification Algorithm -- 5.3 Type Inference Algorithm -- 6 Related Work -- 7 Conclusion and Future Work -- A System F + restrict -- B Type-Directed Translation to System F+restrict -- References -- Towards Efficient Adjustment of Effect Rows -- 1 Introduction -- 2 Overview -- 2.1 Effect Handlers -- 2.2 Evidence Passing Semantics and Row-Based Effect System -- 2.3 Effect Type Adjustment for Function Types -- 2.4