Semantics and Types in Functional Logic Programming  p. 1
Polytypic Programming With Ease (Extended Abstract)  p. 21
Type Inference for Overloading without Restrictions, Declarations or Annotations  p. 37
Partial Evaluation and Non-interference for Object Calculi  p. 53
Lazy Lexing is Fast  p. 68
A Functional-Logic Perspective of Parsing  p. 85
Implementing Encapsulated Search for a Lazy Functional Logic Language  p. 100
Comparison of Deforestation Techniques for Functional Programs and for Tree Transducers  p. 114
Automatic Verification Based on Abstract Interpretation  p. 131
A Transformation System for Lazy Functional Logic Programs  p. 147
Termination Analysis of Tabled Logic Programs Using Mode and Type Information  p. 163
On Quasi-Reductive and Quasi-Simplifying Deterministic Conditional Rewrite Systems  p. 179
An Interval Lattice-Based Constraint Solving Framework for Lattices  p. 194
Higher Order Matching for Program Transformation  p. 209
Automated Generalisation of Function Definitions  p. 225
An Extensional Characterization of Lambda-Lifting and Lambda-Dropping  p. 241
Using Types as Approximations for Type Checking Prolog Programs  p. 251
Typed Static Analysis: Application to Groundness Analysis of Prolog and ?Prolog  p. 267
A Space Efficient Engine for Subsumption-Based Tabled Evaluation of Logic Programs  p. 284

On Reducing the Search Space of Higher-Order Lazy Narrowing..  p. 319
Typed Higher-Order Narrowing without Higher-Order Strategies  p. 335
A Semantics for Program Analysis in Narrowing-Based Functional Logic Languages  p. 353
Author Index  p. 369

Table of Contents provided by Blackwell's Book Services and R.R. Bowker. Used with permission.