

Table of Contents

Synthesis

- A Formal Framework for Synthesis and Verification of Logic Programs 1
Alessandro Avellone, Mauro Ferrari and Camillo Fiorentini
- Protocols between Programs and Proofs 18
Iman Poernomo and John N. Crossley
- A Technique for Modular Logic Program Refinement 38
Robert Colvin, Ian Hayes and Paul Strooper

Transformation

- Higher-Order Transformation of Logic Programs 57
Silvija Seres and Michael Spivey

Analysis

- Non-transformational Termination Analysis of Logic Programs,
Based on General Term-Orderings 69
Alexander Serebrenik and Danny De Schreye

Specialisation

- A Model for Inter-module Analysis and Optimizing Compilation 86
*Francisco Bueno, María García de la Banda, Manuel Hermenegildo,
Kim Marriott, Germán Puebla and Peter J. Stuckey*
- Measuring the Effectiveness of Partial Evaluation
in Functional Logic Languages 103
Elvira Albert, Sergio Antoy and Germán Vidal
- Automated Strategies for Specializing Constraint Logic Programs 125
Fabio Fioravanti, Alberto Pettorossi and Maurizio Proietti

Abstract Interpretation

- Measuring the Precision of Abstract Interpretations 147
Alessandra Di Pierro and Herbert Wiklicky

Debugging

Specifying Prolog Trace Models with a Continuation Semantics 165
Erwan Jahier, Mireille Ducassé and Olivier Ridoux

Author Index 183