

Contents

Model checking and Boolean graphs	1
<i>H.R. Andersen</i>	
SIGNAL as a model for real-time and hybrid systems	20
<i>A. Benveniste, M. Le Borgne, P. Le Guernic</i>	
Towards an adequate notion of observation	39
<i>G. Bernot, M. Bidoit, T. Knapik</i>	
Proving safety of speculative load instructions at compile time	56
<i>D. Bernstein, M. Rodeh, M. Sagiv</i>	
Typed norms	73
<i>A. Bossi, N. Cocco, M. Fabris</i>	
Compositional refinements in multiple blackboard systems	93
<i>X.J. Chen, C. Montangero</i>	
Fully persistent arrays for efficient incremental updates and voluminous reads	110
<i>T.-R. Chuang</i>	
Back to direct style	130
<i>O. Danvy</i>	
Extraction of strong typing laws	151
<i>K.-G. Doh, D.A. Schmidt</i>	
Detecting determinate computations by bottom-up abstract interpretation	167
<i>R. Giacobazzi, L. Ricci</i>	
ELIOS-OBJ - Theorem proving in a specification language	182
<i>I. Gnaedig</i>	
Incremental garbage collection without tags	200
<i>B. Goldberg</i>	
Approximate fixed points in abstract interpretation	219
<i>C. Hankin, S. Hunt</i>	
Dynamic typing	233
<i>F. Henglein</i>	
Automatic parallelization of lazy functional programs	254
<i>G. Hogen, A. Kindler, R. Loogen</i>	
Reversing abstract interpretations	269
<i>J. Hughes, J. Launchbury</i>	
A theory of qualified types	287
<i>M.P. Jones</i>	

A semantics for multiprocessor systems	307
<i>P. Krishnan</i>	
Interprocedural type propagation for object-oriented languages	321
<i>J.-M. Larcheveque</i>	
Using the Centaur system to design SIMD parallel languages and programming a case study	341
<i>J.L. Levaire</i>	
The tensor product in Wadler's analysis of lists	351
<i>F. Nielson, H.R. Nielson</i>	
Basic superposition is complete	371
<i>R. Nieuwenhuis, A. Rubio</i>	
Observers for linear types	390
<i>M. Odersky</i>	
Type inference for partial types is decidable	408
<i>P. O'Keefe, M. Wand</i>	
A provably correct compiler generator.	418
<i>J. Palsberg</i>	
An adequate operational semantics for sharing in lazy evaluation	435
<i>S. Purushothaman, J. Seaman</i>	
Modules for an object-oriented specification language (Invited lecture)	451
<i>A. Tarlecki</i>	
Typing references by effect inference	473
<i>A. K. Wright</i>	