Bibliograpy

The bibliography that follows was prepared originally with a software program called *refer* that is no longer available. As a result, the raw bibliography must be reproduced here.

%A S. K. Abdali %T A lambda-calculus model of programming languages— jumps and functions %J J. of Computer Languages %V 1 %D 1975 %P 303-320

%A H. Abelson %A G. Sussman %T Structure and Interpretation of Computer Programs %I MIT Press/McGraw-Hill %C New York %D 1984

%A S. Abramsky %T On semantic foundations for applicative multiprogramming %B LNCS 154: Proc. 10th ICALP %C Berlin %I Springer %D 1982 %P 1-14

%A S. Abramsky %T Experiments, powerdomains, and fully abstract models for applicative multiprogramming %B LNCS 158: Foundations of Computation Theory %I Springer %C Berlin %D 1983 %P 1-13

%A J. Adamek %A V. Koubek %T Least fixed point of a functor %J Journal of Computer and System Sciences %V 19 %D 1979 %P 163-178

%A A. Aho %A J. D. Ullman %T Principles of Compiler Design %I Addison-Wesley %C Reading, Mass. %D 1977

%A E. R. Anderson %A F. C. Belz %T Issues in the formal specification of programming languages %B Formal Description of Programming Concepts %E E. J. Neuhold %I North-Holland %C Amsterdam %D 1978

%A E. R. Anderson %A F. C. Belz %A E. K. Blum %T SEMANOL (73): A metalanguage for programming the semantics of programming languages %J Acta Informatica %V 6 %D 1976 %P 109-132

%A D. Andrews %A W. Henhapl %T Pascal %B Formal Specification and Software Development %E D. Bjørner and C. Jones %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1982 %P 175-252

%A A. Appel %T Semantics-directed code generation %B Proc. 12th ACM Symp. on Prin. of Prog. Lang. %C New Orleans %D 1985 %P 315-324

%A K. R. Apt %T Equivalence of operational and denotational semantics for a fragment of Pascal %B Formal Descriptions of Programming Concepts %E E. J. Neuhold %I North-Holland %C Amsterdam %D 1978 %P 141-163

%A K. R. Apt %T Ten years of Hoare's logic: a survey— part I %J ACM Trans. on Prog. Lang. and Systems %V 3 %D 1981 %P 431-483

%A K. R. Apt %T Ten years of Hoare's logic: a survey— part II: nondeterminism %J Theoretical Computer Science %V 28 %D 1984 %P 83-110

%A K. R. Apt %A A. Olderog %T Proof rules and transformations dealing with fairness %J Science of Comp. Programming %V 3 %D 1983 %P 65-100

%A K. Apt %A G. D. Plotkin %T A Cook's tour of countable nondeterminism %B LNCS 115: Proc. 9th ICALP %I Springer %C Berlin %D 1981 %P 479-494

%A M. A. Arbib %A E. G. Manes %T The pattern-of-calls expansion is the canonical fixpoint for recursive definitions %J Journal of the ACM %D 1982 %V 29 %P 577-602

%A E. Artesiano %A G. Costa %T Languages with reducing reflexive types %B LNCS 85: Proc. 7th ICALP %I Springer %C Berlin %D 1980

%A E. A. Ashcroft %A W. W. Wadge %T Lucid, a nonprocedural language with iteration %J Comm. of the ACM %V 20 %D 1977 %P 519-526

%A E. A. Ashcroft %A W. W. Wadge %T Prescription for semantics %J ACM Trans. on Prog. Lang. and Sys. %V 4 %D 1982 %P 283-194

%A L. Augustsson %T A compiler for lazy ML %B Proc. ACM Conf. on LISP and Functional Programming %C Austin, Texas %D 1984 %P 218-227

%A J. Backus %T Can programming be liberated from the von Neumann style? A functional style and its algebra of programs %J Comm. of the ACM %V 21 %D 1978 %P 613-641

%A J. W. deBakker %T Recursive programs as predicate transformers %B Formal Descriptions of Programming Concepts %E E.J. Neuhold %I North-Holland %C Amsterdam %D 1978 %P 165-181

%A J. W. deBakker %T Mathematical Theory of Program Correctness %I Prentice-Hall %D 1980 %C Englewood Cliffs, N.J.

%A H. Barendregt %T The type free lambda calculus %B Handbook of Mathematical Logic %E J. Barwise %I North-Holland %C Amsterdam %D 1977 %P 1091-1132

%A H. Barendregt %T The Lambda Calculus— Its Syntax and Semantics %I North-Holland %C Amsterdam %D 1981

%A H. Barendregt %T Introduction to lambda calculus %J Niew Archief Voor Wiskunde %V 4 %D 1984 %P 337-372

%A W. A. Barrett %A J. D. Couch %T Compiler Construction: Theory and Practice %I S.R.A. %C Chicago %D 1979

%A D. W. Barron %T An Introduction to the Study of Programming Languages %I Cambridge University Press %C Cambridge %D 1977

%A F. L. Bauer %A H. Wossner %T Algorithmic Language and Program Development %I Springer %C Berlin %D 1982

%A D. M. Berry %T Remarks on R. D. Tennent's language design methods based on semantic principles %J Acta Informatica %V 15 %D 1981 %P 83-98

%A D. M. Berry %T A denotational semantics for shared memory parallelism and nondeterminism %J Acta Informatica %V 21 %D 1985 %P 599-628

%A G. Berry %T Some syntactic and categorical constructions of lambda calculus models %J Report 80 %I INRIA %C Sophia Antipolis %D 1981

%A G. Berry %A J.-J. Levy %T A survey of some syntactic results of the lambda calculus %B LNCS 74: Proc. 8th Symp. Math. Foundations of Computer Science %I Springer %C Berlin %D 1979 %P 552-566

%A G. Berry %A P.-L. Curien %T Sequential algorithms on concrete data structures %J Theoretical Computer Science %V 20 %D 1982 %P 265-322

%A G. Berry %A P.-L. Curien %A J.-J. Levy %T Full abstraction for sequential languages: the state of the art %D 1982 %B Proc. French-American Seminar on Semantics %C Fontainebleau, France %D 1982

%A R. S. Bird %T Programs and Machines %I Wiley %C New York %D 1976

%A G. Birkhoff %T Lattice Theory, 3rd Edition %I American Mathematical Society %C Providence, R.I. %D 1967

%A D. Bjørner %T The Vienna development method %B LNCS 75: Mathematical studies of information processing %I Springer %C Berlin %D 1978 %P 326-359

%Q Bjørner, D., ed. %T LNCS 86: Abstract Software Specifications %I Springer %C Berlin %D 1980

%A Bjørner, D., ed. %T Formal Description of Programming Concepts II %I North-Holland %C Amsterdam %D 1983

%A D. Bjørner %A C. B. Jones, eds. %T LNCS 61: The Vienna Development Method: the Metalanguage %I Springer %C Berlin %D 1978

%A D. Bjørner %A C. B. Jones %T Formal Specification and Software Development %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1982

%A D. Bjørner %A O. N. Oest %T LNCS 98: Towards a formal description of Ada %I Springer %C Berlin %D 1980

%A A. Blikle %A A. Tarlecki %T Naive denotational semantics %B Proc. IFIP Congress 83 %I North-Holland %C Amsterdam %D 1983 %P 345-356

%A J. Bodwin %A L. Bradley %A J. Kanda %A D. Little %A U. Pleban %T Experience with an experimental compiler-compiler based on denotational semantics %B Proc. SIGPLAN 82 Symp. on Compiler Construction, SIGPLAN Notices %V 17 %D 1982 %P 216-229

%Q Bohm, C., ed. %T LNCS 37: Lambda-calculus and Computer Science Theory %I Springer %C Berlin %D 1975

%A P. Branquart %A G. Louis %A P. Wodon %T LNCS 128: An Analytical Description of CHILL, the CCITT High Level Language %I Springer %C Berlin %D 1982

%A S. D. Brookes %T A fully abstract semantics and a proof system for an ALGOL-like language with sharing %B LNCS: Proc. Workshop on Foundations of Programming Semantics %I Springer %C Berlin %D 1985a

%Q Brookes, S. D., A. W. Roscoe, and G. Winskel, eds. %T LNCS 197: Seminar on Concurrency %I Springer %C Berlin %D 1985

%A W. H. Burge %T Recursive Programming Techniques %I Addison-Wesley %C Reading, Mass. %D 1975

%A R. Burstall %T Proving properties of programs by structural induction %J Computer Journal %V 12 %P 41-48 %D 1969

%A R. Burstall %A J. Darlington %T A transformation system for developing recursive programs %J Journal of the ACM %V 24 %P 44-67 %D 1977

%A R. Burstall %A J. Goguen %T Putting theories together to make specifications %B Proc. 5th Int. Joint Conf. on Artificial Intelligence %C Cambridge, Mass. %D 1977 %P 1045-1058

%A R. Burstall %A J. Goguen %T Algebras, theories, and freeness: an introduction for computer scientists %B Proc. Marktoberdorf Summer School on Theoretical Foundations of Programming Methodology %D August 1981

%A R. M. Burstall %A D. B. MacQueen %A D. T. Sannella %T HOPE: an experimental applicative language %R CSR-62-80 %I Computer Science Dept., Edinburgh University %C Edinburgh, Scotland %D 1981

%A H. Christiansen %A N. D. Jones %T Control flow treatment in a simple semantics-directed compiler %B Formal Descriptions of Programming Concepts II %E D. Bjørner %I North-Holland %C Amsterdam %D 1983 %P 73-97

%A A. Church %T The Calculi of Lambda Conversion %I Princeton Univ. Press %C Princeton, N.J. %D 1951

%A T. J. Clarke %A P. J. Gladstone %A C. D. MacClean %A A. C. Norman %T SKIM— the S, K, I reduction machine %B Proc. ACM LISP conference %D 1980 %P 128-135

%A J. C. Cleaveland %T Mathematical specifications %J ACM SIGPLAN Notices %V 15-12 %D 1980 %P 31-42

%A J. C. Cleaveland %A R. C. Uzgalis %T Grammars for Programming Languages %I Elsevier %C New York %D 1977

%A W. Clinger %T Foundations of actor semantics %R Ph.D. thesis, AI lab report AI-TR-633 %I MIT %D 1981

%A W. Clinger %T Summary of the Scheme 311 compiler: an exercise in denotational semantics %B Proc. ACM Symp. on LISP and Functional Programming %C Austin, Texas %D August, 1984 %P 356-364

%A A. J. Cohn %T The equivalence of two semantic definitions: a case study in LCF %J SIAM J. of Computing %V 12 %D 1983 %P 267-285

%A P. M. Cohn %T Universal Algebra (rev. ed.) %I D. Reidel Pub. %C Boston %D 1981

%A B. Courcelle %A I. Guessarian %T On some classes of interpretations %J J. of Computer and System Sciences %V 17 %D 1978 %P 388-413

%A P. Cousot %A R. Cousot %T Abstract interpretation: a unified lattice model for static analysis of programs %B Poc. 4th ACM Symp. on Prin. of Prog. Lang. %C Los Angeles %D 1977 %P 238-252

%A P. Cousot %A R. Cousot %T Systematic design of program analysis frameworks %B Proc. 6th ACM Symp. on Prin. of Prog. Lang. %C San Antonio, Texas %D 1979 %P 269-282

%A P.-L. Curien %T Categorical combinatory logic %B LNCS 194: Proc. 12th ICALP %I Springer %C Berlin %D 1985 %P 130-139

%A H. B. Curry %A R. Feys %T Combinatory Logic, Vol. 1 %I North-Holland %C Amsterdam %D 1958

%A J. Darlington %A P. Henderson %A D. Turner, eds. %T Functional Programming and its Applications %I Cambridge Univ. Press %C Cambridge %D 1982

%A A. Demers %A J. Donohue %A G. Skinner %T Data types as values: polymorphism, type-checking, and encapsulation %B Proc. 5th ACM Symp. on Prin. of Prog. Lang. %D 1978 %C Tucson, Arizona %P 23-30

%A K. J. Devlin %T Fundamentals of Contemporary Set Theory %I Springer %C Berlin %D 1979

%A E. W. Dijkstra %T A Discipline of Programming %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1976

%A J. Donahue %T LNCS 42: Complementary Definitions of Programming Language Semantics %I Springer %C

Berlin %D 1976

%A J. Donahue %T Locations considered unnecessary %J Acta Informatica %V 8 %P 221-242 %D 1977

%A J. Donohue %T On the semantics of ‘data type’ %J SIAM J. of Computing %D 1979 %V 8 %P 546-560

%A V. Donzeau-Gouge %T On the formal description of Ada %B LNCS 94: Semantics-Directed Compiler Generation %I Springer %C Berlin %D 1980 %E N.D. Jones

%A V. Donzeau-Gouge %T Denotational definition of properties of program computations %B Program Flow Analysis: Theory and Applications %E S.S. Muchnick and N. D. Jones %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1982 %P 343-379

%A P. Dybjer %T Using domain algebras to prove the correctness of a compiler %B LNCS 182: Proc. 2nd Symp. on Theoretical Aspects of Comp. Sci. %I Springer %C Berlin %D 1985 %P 98-108

%A H. Enderton %T A Mathematical Introduction to Logic %I Academic Press %D 1974 %C New York

%A H. Enderton %T Elements of Set Theory %I Academic Press %C New York %D 1977

%A E. P. Ershov %T On the essence of compilation %B Formal Description of Programming Concepts %E E.J. Neuhold %I North-Holland %C Amsterdam %D 1978

%A E. P. Ershov %T Mixed computation: potential applications and problems for study %J Theoretical Computer Science %V 18 %D 1982 %P 41-68

%A J. Fairbairn %T PONDER and its type system %R Tech. rpt. 31 %I Computer Laboratory, University of Cambridge %D 1982

%A R. E. Filman %A D. E. Friedman %T Coordinated Computing %I McGraw-Hill %C New York %D 1984

%A D. Friedman %A D. S. Wise %T CONS should not evaluate its arguments %B Proc. 3rd ICALP %C Edinburgh %E S. Michaelson and R. Milner %P 257-284 %D 1976

%Q Friedman, D., M. Wand, C. Haynes, E. Kohlbecker, and W. Clinger %T Programming Languages: A Hitchhiker’s Guide to the Meta-Universe %R Course notes, Computer Science Dept. %I Indiana Univ. %D 1984

%A H. Ganzinger %A K. Ripken %A R. Wilhelm %T Automatic generation of optimizing multipass compilers %B Proc. IFIP Congress 77 %I North-Holland %C Amsterdam %D 1977 %P 535-540

%A H. Ganzinger %T Transforming denotational semantics into practical attribute grammars %B LNCS 94: Semantics-Directed Compiler Generation %E N.D. Jones %I Springer %C Berlin %D 1980 %P 1-69

%A H. Ganzinger %T Denotational semantics for languages with modules %B Formal Description of Programming Concepts II %E D. Bjørner %I North-Holland %C Amsterdam %D 1983 %P 3-23

%A H. Ganzinger %A R. Giegerich %A U. Moncke %A R. Wilhelm %T A truly generative semantics-directed compiler generator %B Proc. ACM SIGPLAN 82 Symp. on Compiler Construction %C Boston %D 1982 %P 172-184 %J ACM SIGPLAN Notices 17-6

%A M.-C. Gaudel %T Specification of compilers as abstract data type representations %B LNCS 94: Semantics-directed compiler generation %I Springer %C Berlin %D 1980 %P 140-164

%A M.-C. Gaudel %T Compiler definitions from formal definitions of programming languages: a survey %B LNCS 107: Formalization of Programming Concepts %I Springer %C Berlin %D 1981 %P 96-114

%A M. Georgeff %T Transformations and reduction strategies for typed lambda expressions %J ACM Trans. on Prog. Lang. and Sys. %V 6 %D 1984 %P 603-631

%A Gierz, G. %A K. H. Hoffmann %A K. Keimel %A J. D. Lawson %A M. Mislove %A D. S. Scott %T A Compendium of Continuous Lattices %I Springer %C Berlin %D 1980

%A H. Glaser %A C. Hankin %A D. Till %T Principles of Functional Programming %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1985

%A J. A. Goguen %T Some design principles and theory for OBJ-0 %B LNCS 75: Mathematical Studies of Information Processing %E E. Blum, M. Paul, and S. Takasu %I Springer %C Berlin %D 1979 %P 425-471

%A J. A. Goguen %A K. Parsaye-Ghomi %T Algebraic denotational semantics using parameterized abstract modules %B LNCS 107: Formalization of Programming Concepts %E J. Diaz %E I. Ramos %I Springer %C Berlin %D 1981 %P 292-309

%A Goguen, G. %A J. W. Thatcher %A E. G. Wagner %A J. B. Wright %T Initial algebra semantics and continuous algebras %J Journal of the ACM %V 24 %D 1977 %P 68-95

%A M. J. C. Gordon %T Models of Pure LISP %R Experimental Programming Reports 31 %I Machine Intelligence Dept., Edinburgh Univ. %C Scotland %D 1973

%A M. J. C. Gordon %T Towards a semantic theory of dynamic binding %R STAN-CS-75-507 %I Computer Science Dept., Stanford Univ. %D 1975

%A M. J. C. Gordon %B Proc. Int. Symp. on Proving and Improving Programs %C Arc-et-Senans, France %T Operational reasoning and denotational semantics %D 1978 %P 83-98

%A M. J. C. Gordon %T The Denotational Description of Programming Languages %I Springer %C Berlin %D 1979

%A M. J. C. Gordon %A R. Milner %A C. Wadsworth %T LNCS 78: Edinburgh LCF %I Springer %C Berlin %D 1979

%A G. Gratzer %T Universal Algebra, 2nd Edition %I Springer %C Berlin %D 1979

%A I. Grief %A A. Meyer %T Specifying the semantics of while-programs: a tutorial and critique of a paper by Hoare and Lauer %J ACM Trans. of Prog. Lang. and Sys. %V 3 %D 1981 %P 484-507

%A I. Guessarian %T LNCS 99: Algebraic Semantics %I Springer %C Berlin %D 1981

%A C. Gunter %T Profinite domains for recursive domain equations %R Tech. rpt. CMU-CS-85-107 %I Computer Science Dept., Carnegie-Mellon Univ. %C Pittsburgh %D 1985a

%A C. Gunter %T Comparing categories of domains %B LNCS: Proc. Workshop on Mathematical Foundations of Programming Semantics %I Springer %C Berlin %D 1985b

%A C. Gunter %T A universal domain technique for profinite posets %B LNCS 194: Proc. 12th ICALP %I Springer %C Berlin %D 1985c %P 232-243

%A P. Halmos %T Naive Set Theory %I Van Nostrand %C Princeton %D 1960

%A J. Halpern %A J. Williams %A E. Wimmers %A T. Winkler %T Denotational semantics and rewrite rules for FP %B Proc. 12th ACM Symp. on Princ. of Prog. Lang. %C New Orleans %D 1986 %P 108-120

%A P. Henderson %T Functional Programming %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1980

%A P. Henderson %A L. Morris %T A lazy evaluator %B 3rd ACM Symp. on Prin. of Prog. Lang. %D 1976 %P 95-103

%A W. Henhapl %A C. Jones %T ALGOL 60 %B Formal Specification and Software Development %E D. Bjørner and C. Jones %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1982 %P 141-174

%A M. Hennessy %A G. D. Plotkin %T Full abstraction for a simple parallel programming language %B LNCS 74: Proc. Math. Foundations of Comp. Sci. %I Springer %C Berlin %D 1979

%A M. C. Henson %A R. Turner %T Completion semantics and interpreter generation %B Proc. 9th ACM Symp. on Prin. of Prog. Lang. %C Albuquerque, N.M. %D 1982 %P 242-254

%A H. Herrlich %A G. E. Strecker %T Category Theory %I Allyn and Bacon %C Boston %D 1973

%A C. Hewitt %A H. Baker %T Actors and continuous functionals %B Formal Description of Programming Concepts %E E. J. Neuhold %I North-Holland %C Amsterdam %D 1978 %P 367-390

%A C. A. R. Hoare %T An axiomatic basis for computer programming %J Comm. of the ACM %V 12 %D 1969 %P 576-580

%A C. A. R. Hoare %T Proof of correctness of data representations %J Acta Informatica %V 1 %D 1972 %P 271-281

%A C. A. R. Hoare %T Recursive data structures %J Int. J. of Computer and Info. Sciences %V 4 %D 1975 %P 105-132

%A C. A. R. Hoare %A P. E. Lauer %T Consistent and complementary formal theories of the semantics of programming languages %J Acta Informatica %V 3 %D 1974 %P 135-153

%A C. A. R. Hoare %A N. Wirth %T An axiomatic definition of the programming language Pascal %J Acta Informatica %D 1973 %V 2 %P 335-355

%A C. M. Hoffman %A M. J. O'Donnell %T Programming with equations %J ACM Trans. Prog. Lang. and Systems %V 4 %D 1983 %P 83-112

%A J. Hopcroft %A J. Ullman %T Introduction to Automata Theory, Languages, and Computation %I Addison-Wesley %C Reading, Mass. %D 1979

%A P. Hudak %A D. Krantz %T A combinator-based compiler for a functional language %B Proc. 11th ACM Symp. on Prin. of Prog. Lang. %C Salt Lake City, Utah %D 1984 %P 122-132

%A G. Huet %A D. C. Oppen %T Equations and rewrite rules: a survey %B Formal Language Theory %E R. Book %I Academic Press %C New York %D 1980

%A R. J. Hughes %T Super combinators: a new implementation method for applicative languages %B Proc. ACM Symp. on LISP and Functional Programming %D 1982 %P 1-10

%A K. Jensen %T Connection between Dijkstra's predicate transformers and denotational continuation semantics %R Report DAIMI PB-86 %D 1978 %I Computer Science Dept., Aarhus Univ. %C Denmark

%A T. Johnsson %T Efficient compilation of lazy evaluation %B Proc. ACM SIGPLAN 84 Conference on Compiler Construction %J ACM SIGPLAN Notices %V 19-6 %D 1984 %P 58-69

%A J. B. Johnston %T The contour model of block structured processes %J ACM SIGPLAN Notices %V 6 %D 1971 %P 55-82

%A C. Jones %T Modelling concepts of programming languages %B Formal Specification and Software Development %E D. Bjørner and C. Jones %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1982a %P 85-124

%A C. Jones %T More on exception mechanisms %B Formal Specification and Software Development %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1982b %P 125-140 %E D. Bjørner and C. Jones

%Q Jones, N. D., ed. %T LNCS 94: Semantics-Directed Compiler Generation %I Springer %C Berlin %D 1980a

%A N. D. Jones %T Flow analysis of lambda expressions %B LNCS 85: Proc. 7th ICALP %I Springer %D 1980b %C Berlin

%A N. D. Jones %A S. S. Muchnick %T LNCS 66: TEMPO: A Unified Treatment of Binding Time and Parameter Passing Concepts in Programming Languages %I Springer %C Berlin %D 1978

%A N. D. Jones %A S. S. Muchnick %T A fixed-program machine for combinator expression evaluation %B Proc. ACM Conf. on LISP and Functional Programming %D 1982

%A N. D. Jones %A D. A. Schmidt %T Compiler generation from denotational semantics %B LNCS 94: Semantics-Directed Compiler Generation %I Springer %C Berlin %D 1980 %P 70-93

%A N. D. Jones %A P. Sestoft %A H. Sondergaard %T An experiment in partial evaluation: the generation of a compiler generator %J ACM SIGPLAN Notices %V 20-8 %D 1985 %P 82-87

%A G. Kahn %A D. B. MacQueen %A G. D. Plotkin, eds. %T LNCS 173: Semantics of Data Types %I Springer %C Berlin %D 1984

%A T. Kamimura %A A. Tang %T Algebraic relations and presentations %J Theoretical Comp. Science %V 27 %D 1983 %P 39-60

%A T. Kamimura %A A. Tang %T Effectively given spaces %J Theoretical Comp. Science %V 29 %D 1984a %P 155-166

%A T. Kamimura %A A. Tang %T Total objects of domains %J Theoretical Comp. Science %D 1984b

%A A. Kanda %T Data types as initial algebras %B 19th Symp. on Foundations of Comp. Science %D 1978

%A A. Kanda %T Effective solutions of recursive domain equations %R Ph.D. thesis %I University of Warwick %D 1979

%A K. Karlsson %A K. Petersson %T Notes from the Aspenas symposium on functional languages and computer architectures %J ACM SIGPLAN Notices %V 17-11 %D 1982 %P 14-23

%A V. Kini %A D. Martin %A A. Stoughton %T Testing the INRIA Ada formal definition: the USC-ISI formal semantics project %B Proc. ADATec Meeting %D 1982

%A S. C. Kleene %T Introduction to Metamathematics %I Van Nostrand %C Princeton, N.J. %D 1952

%A D. Knuth %T The semantics of context free languages %J Math. Systems Theory %V 2 %D 1968 %P 127-145 %O (Corrigenda, vol. 5, p. 95, 1971)

%A J. Lambek %T From lambda-calculus to cartesian closed categories %B To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism %E J. P. Seldin and J. R. Hindley %I Academic Press %C New York

%D 1980 %P 375-402

%A P. Landin %T The mechanical evaluation of expressions %J Computer Journal %V 6 %D 1964 %P 308-320

%A P. Landin %T A correspondence between ALGOL60 and Church's lambda notation %J Comm. of the ACM %V 8 %D 1965 %P 89-101, 158-165

%A P. Landin %A R. Burstall %T Programs and their proofs: an algebraic approach %B Machine Intelligence 4 %E D. Michie %I Edinburgh Univ. Press %D 1969 %P 17-44

%A H. Ledgard %T Ten mini-languages: a study of topical issues in programming languages %J ACM Computing Surveys %V 3 %D 1971 %P 115-146

%A J. A. N. Lee %T Computer Semantics %I Van Nostrand-Reinhold %C New York %D 1972

%A D. J. Lehmann %A M. B. Smyth %T Algebraic specification of data types: a synthetic approach %J Math. Systems Theory %V 14 %P 97-139 %D 1981

%A E. J. Lemmon %T Beginning Logic %I Thomas Nelson and Sons, Pub. %C London %D 1965

%A E. J. Lemmon %T Introduction to Axiomatic Set Theory %I Routledge and Kegan Paul, Ltd. %C London %D 1968

%A J.-J. Levy %T An algebraic interpretation of the lambda-beta-kappa-calculus and an application of a labelled lambda calculus %J Theoretical Computer Science %V 2 %D 1976 %P 97-114

%A G. T. Ligler %T A mathematical approach to language design %B Proc. 2nd ACM Symp. on Prin. of Prog. Lang. %C Palo Alto, Cal. %D 1975

%A G. T. Ligler %T Surface properties of programming language constructs %J Theoretical Comp. Science %V 2 %D 1976

%A P. Lucas %T Main approaches to formal semantics %B Formal Specification and Software Development %E D. Bjørner and C. Jones %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1982 %P 3-24

%A P. Lucas %A K. Walk %T On the formal definition of PL/1 %J Annual Review in Automatic Programming %I Pergammon Press %C London %D 1963 %V 6 %P 105-152

%A J. McCarthy %T Towards a mathematical science of computation %B Proc. IFIP Congress 63 %I North-Holland %D 1963 %C Amsterdam %P 21-28

%A J. McCarthy %A J. Painter %T The correctness of a compiler for arithmetic expressions %B Mathematical Aspects of Computer Science, Proc. Symp. Applied Math. 19 %I American Math. Society %D 1967 %P 33-41

%A N. McCracken %T The typechecking of programs with implicit type structure %B LNCS 173: Semantics of Data Types %I Springer %C Berlin %D 1984 %P 301-316

%A J. R. McGraw %T The Val language: description and analysis %J ACM Trans. on Prog. Lang. and Systems %V 4 %D 1982 %P 44-82

%A D. B. MacQueen %A R. Sethi %T A semantic model of types for applicative languages %B Proc. ACM Conf. on LISP and Functional Programming %D 1982 %C Pittsburgh %P 243-252

%A D. B. MacQueen %A G. Plotkin %A R. Sethi %T An ideal model for recursive polymorphic types %B Proc. 11th ACM Symp. on Princ. of Prog. Lang. %C Salt Lake City, Utah %D 1984 %P 165-174

%A M. Main %A D. Benson %T Functional behavior of nondeterministic and concurrent programs %J Information and Control %D 1984 %V 62 %P 144-189

%Q Manes, E. G., ed. %T LNCS 25: Category Theory Applied to Computation and Control %I Springer %C Berlin %D 1975

%A Z. Manna %T Mathematical Theory of Computation %I McGraw-Hill %C New York %D 1974

%A Z. Manna %A J. Vuillemin %T Fixpoint approach to the theory of computation %J Comm. of the ACM %D 1972 %V 15 %P 528-536

%A Z. Manna %A S. Ness %A J. Vuillemin %T Inductive methods for proving properties of programs %J ACM SIGPLAN Notices %V 7-1 %D 1972 %P 27-50

%A Z. Manna %A R. Waldinger %T The Logical Basis for Computer Programming, Vol. 1 %I Addison Wesley %C Reading, Mass. %D 1985

%A M. Marcotty %A H. F. Ledgaard %A G. V. Bochmann %T A sampler of formal definitions %J ACM Computing Surveys %V 8 %P 191-276 %D 1976

%A G. Markowski %T A motivation and generalization of Scott's notion of a continuous lattice %B LNM 871: Continuous Lattices %I Springer %C Berlin %D 1981 %P 298-307

%A G. Markowski %A B. K. Rosen %T Bases for chain-complete posets %J IBM J. or Research and Development %V 20 %D 1976 %P 138-147

%A M. Mauny %A P. L. Curien %A G. Cousineau %T The categorical abstract machine %B Proc. IFIP Conf. on Functional Programming Languages and Computer Architecture %C Nancy, France %D Sept. 1985

%A B. H. Mayoh %T Attribute grammars and mathematical semantics %J SIAM Journal of Computing %V 10 %D 1981 %P 503-518

%A A. C. Melton %A D. A. Schmidt %T A topological framework for cpos lacking bottom elements %B LNCS: Mathematical Foundations of Programming Semantics %I Springer %C Berlin %D 1986

%A A. Mazurkiewicz %T Proving algorithms by tail functions %J Information and Control %V 18 %D 1970 %P 220-226

%A A. R. Meyer %T What is a model of the lambda calculus? %J Information and Control %V 52 %D 1982 %P 87-122

%A A. R. Meyer %T Understanding ALGOL: a view of a recent convert to denotational semantics %B Proc. IFIP Congress 1983 %I North-Holland %C Amsterdam %D 1983 %P 951-962

%A A. Meyer %A M. Wand %T Continuation semantics in the typed lambda-calculus %B LNCS 193: Proc. Logics of Programs %I Springer %C Berlin %D 1985 %P 219-224

%A R. Milne %T Transforming predicate transformers %B Formal Description of Programming Concepts %E E.J. Neuhold %I North-Holland %C Amsterdam %D 1978

%A R. Milne %A C. Strachey %T A Theory of Programming Language Semantics %I Chapman and Hall %C London %D 1976

%A R. Milner %T Models of LCF %B Mathematical Centre Tracts 82: Foundations of Computer Science II, part 2 %D 1976a %P 49-63 %E J. de Bakker %E K. Apt %I Mathematisch Centrum %C Amsterdam

%A R. Milner %T Program semantics and mechanized proof %B Mathematical Centre Tracts 82: Foundations of Computer Science II, part 2 %D 1976b %P 3-44 %E J. de Bakker %E K. Apt %I Mathematisch Centrum %C Amsterdam

%A R. Milner %T Fully abstract models of typed lambda-calculi %J Theoretical Computer Science %V 4 %D 1977 %P 1-22

%A R. Milner %T A theory of type polymorphism in programming %J J. of Computer and System Sciences %V 17 %D 1978 %P 348-375

%A R. Milner %T LNCS 92: A Calculus of Communicating Systems %I Springer %C Berlin %D 1980

%A R. Milner %T Calculi for synchrony and asynchrony %J Theoretical Comp. Sci. %V 25 %D 1983 %P 267-310

%A R. Milner %T Lectures on a Calculus for Communicating Systems %B LNCS 197: Seminar on Concurrency %I Springer %C Berlin %D 1985 %P 197-220

%A D. Milos %A U. Pleban %A G. Loegel %T Direct implementation of compiler specifications or the Pascal P-code compiler revisited %B Proc. 11th ACM Symp. on Princ. of Prog. Lang. %C Salt Lake City, Utah %D 1984 %P 196-207

%A J. Mitchell %T Coercion and type inference %B Proc. 11th ACM Symp. on Pric. of Prog. Lang. %C Salt Lake City, Utah %D 1984 %P 175-185

%A F. L. Morris %T Advice on structuring compilers and proving them correct %B Proc. 1st ACM Symp. on Prin. of Prog. Lang. %C Boston %D 1973 %P 144-152

%A J. H. Morris %T Lambda-calculus models of programming languages %R Ph.D. thesis, Project MAC report TR-57 %I MIT %C Cambridge, Mass. %D 1968

%A J. H. Morris %A B. Wegbreit %T Subgoal induction %J Comm. of the ACM %V 20 %P 209-222 %D 1977

%A P. D. Mosses %T The mathematical semantics of Algol60 %R Tech. monograph PRG12 %I Programming Research Group, Oxford Univ., Oxford %D 1974

%A P. D. Mosses %T Mathematical semantics and compiler generation %R Ph.D. Thesis, Oxford University %D 1975

%A P. D. Mosses %T Compiler generation using denotational semantics %B LNCS 45: Proc. Math. Foundations of Comp. Science %I Springer %C Berlin %D 1976 %P 436-441

%A P. D. Mosses %T Making denotational semantics less concrete %B Workshop on semantics of programming languages %C Bad Honnef, Germany %D 1977 %O Univ. of Dortmund, FRG, tech. rpt. 41.

%A P. D. Mosses %T Modular denotational semantics %R draft 1979-3-17, Computer science dept., Aarhus Univ. %C Aarhus, Denmark %D 1979a

%A P. D. Mosses %T SIS— semantics implementation system: reference manual and user guide %R Report DAIMI MD-30 %I Computer Science Dept., Aarhus University %D 1979b

%A P. D. Mosses %T SIS— semantics implementation system: tested examples %R Report DAIMI MD-33 %I Computer Science Dept., Aarhus University %D 1979c

%A P. D. Mosses %T A constructive approach to compiler correctness %B LNCS 85: 7th ICALP %D 1980 %I

Springer %C Berlin

%A P. D. Mosses %T Abstract semantic algebras! %B Formal Description of Programming Concepts II %P 45-72 %E D. Bjørner %I North-Holland %C Amsterdam %D 1983a

%A P. D. Mosses %T A short course on denotational semantics %R Course notes, Facultad de Informatica, Universidad del Pais Vasco %C San Sebastian, Spain %D 1983b

%A P. D. Mosses %T A basic abstract semantic algebra %B LNCS 173: Semantics of data types %I Springer %D 1984 %C Berlin %P 87-108

%A S. S. Muchnick %A U. Pleban %T A semantic comparison of LISP and Scheme %B Proc. ACM Conf. on LISP and Functional Programming %D 1982 %P 56-64

%A K. Mulmuley %T Full abstraction and semantic equivalence %R Ph.D. thesis %I Computer Science Dept., Carnegie-Mellon University, Pittsburgh, PA %D 1985

%A A. Mycroft %T Abstract interpretation and optimizing transformations for applicative programs %R Ph.D. thesis %I Computer Science Dept., University of Edinburgh, Scotland %D 1981

%Q Naur, P., et. al. %T Revised report on the algorithmic language ALGOL60 %J Comm. of the ACM %V 6 %P 1-17 %D 1963

%A Nielsen, M. %A G. D. Plotkin %A G. Winskel %T Petri nets, event structures, and domains %J Theoretical Comp. Science %V 13 %D 1981 %P 85-108

%A Nielson, F. %T Compiler writing using denotational semantics %R Report DAIMI TR-10 %I Computer science dept., Aarhus Univ. %C Denmark %D 1979

%A Nielson, F. %T A denotational framework for data flow analysis %J Acta Informatica %V 18 %D 1983 %P 265-288

%A F. Nielson %T Program transformations in a denotational setting %J ACM Trans. on Prog. Lang. and Sys. %V 7 %D 1985 %P 359-379

%A M. Nivat %T On the interpretation of recursive polyadic program schemes %J Symp. Mathematica %V 15 %D 1975 %P 255-281

%A B. Nordstrom %A K. Petersson %T Types and specifications %B Proc. IFIP Congress 83 %I North-Holland %C Amsterdam %D 1983 %P 915-920

%A F. Oles %T Type algebras, functor categories, and block structure %B Algebraic Methods in Semantics %E M. Nivat and J. Reynolds %I Cambridge Univ. Press %C Cambridge %D in press

%A A. Ollengren %T Definition of Programming Languages by Interpreting Automata %I Academic Press %D 1974 %C New York

%A F. G. Pagan %T Semantics of Programming Languages: A Panoramic Primer %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1981

%A L. Paulson %T A semantics-directed compiler generator %B Proc. 9th ACM Symp. on Prin. of Prog. Lang. %D 1982 %P 224-233

%A L. Paulson %T Compiler generation from denotational semantics %B Methods and Tools for Compiler Construction %E B. Lorho %P Cambridge Univ. Press %D 1984 %P 219-250

%A D. Park %T Fixpoint induction and proofs of program properties %B Machine Intelligence %V 5 %D 1969 %P 59-78 %E Meltzer %E D. Michie %I Edinburgh Univ. Press %C Edinburgh

%A D. Park %T A predicate transformer for weak fair iteration %B Proc., 6th IBM Symp. on Math. Foundations of Computer Science %C Hakene, Japan %D 1981

%A U. Pleban %T Compiler prototyping using formal semantics %B Proc. SIGPLAM 84 Symp. on Compiler Construction %C Montreal %D June, 1984 %P 94-105 %J SIGPLAN Notices 19-6

%A G. D. Plotkin %T Call-by-name, call-by-value and the lambda calculus %J Theoretical Computer Science %V 1 %D 1975 %P 125-159

%A G. D. Plotkin %T A powerdomain construction %J SIAM J. of Computing %V 5 %D 1976 %P 452-487

%A G. D. Plotkin %T LCF considered as a programming language %J Theoretical Comp. Science %V 5 %D 1977 %P 223-255

%A G. D. Plotkin %T A structural approach to operational semantics %R Report DAIMI FN-19 %I Computer Science Dept., Aarhus Univ. %D 1981

%A G. D. Plotkin %T Dijkstra's predicate transformers and Smyth's powerdomains %B LNCS 86: Abstract Software Specifications %I Springer %C Berlin %D 1982a %P 527-553

%A G. D. Plotkin %T A powerdomain for countable nondeterminism %B LNCS 140: Proc. 9th ICALP %I Springer %D 1982b %P 412-428

%A G. D. Plotkin %T The category of complete partial orders: a tool for making meanings %R Postgraduate lecture notes, Computer Science Dept., Univ. of Edinburgh %C Edinburgh %D 1982c

%A W. Polak %T Program verification based on denotational semantics %B Proc. 8th ACM Symp. on Prin. of Prog. Lang. %D 1981a

%A W. Polak %T LNCS 124: Compiler Specification and Verification %I Springer %C Berlin %D 1981b

%A M. R. Raskovsky %T A correspondence between denotational semantics and code generation %R Ph.D. thesis %I Univ. of Essex %D 1982a

%A M. R. Raskovsky %T Denotational semantics as a specification of code generators %B Proc. ACM SIGPLAN 82 Symp. on Compiler Construction %C Boston %D 1982b %P 230-244 %J ACM SIGPLAN Notices 17-6

%A M. R. Raskovsky %A P. Collier %T From standard to implementation denotational semantics %B LNCS 94: Semantics-Directed Compiler Generation %E N.D. Jones %I Springer %C Berlin %D 1980 %P 94-139

%A J.-C. Raoult %A R. Sethi %T Properties of a notation for combining functions %B LNCS 140: Proc. 9th ICALP %I Springer %C Berlin %D 1982 %P 429-441

%A J.-C. Raoult %A R. Sethi %T The global storage needs of a subcomputation %B Proc. ACM Symp. on Prin. of Prog. Lang. %C Salt Lake City, Utah %D 1984 %P 148-157

%A J. C. Reynolds %T GEDANKEN— a simple typeless language based on the principle of completeness and the reference concept %J Comm. of the ACM %V 13 %D 1970 %P 308-319

%A J. C. Reynolds %T Definitional interpreters for higher order programming languages %B Proc. ACM Annual Conf. %D 1972a %P 717-740

%A J. C. Reynolds %T Notes on a lattice-theoretic approach to the theory of computation %R Report, Systems and Info. Sciences Dept., Syracuse Univ. %C Syracuse, N.Y. %D 1972b

%A J. C. Reynolds %T Towards a theory of type structure %B LNCS 19: Proc. Paris Programming Symp. %I Springer %C Berlin %D 1974a %P 408-425

%A J. C. Reynolds %T On the relation between direct and continuation semantics %B LNCS 14: Proc. 2nd ICALP %I Springer %C Berlin %D 1974b %P 157-168

%A J. C. Reynolds %T Semantics of the domain of flow diagrams %J J. of the ACM %V 24 %D 1977 %P 484-503

%A J. C. Reynolds %T The essence of Algol %B Int. Symp. on Algorithmic Languages %E J. de Bakker %E van Vliet %I North-Holland %C Amsterdam %D 1981 %P 345-372

%A J. C. Reynolds %T Types, abstraction, and parametric polymorphism %B Proc. IFIP Congress %E R.E.A. Mason %I North-Holland %C Amsterdam %D 1983 %P 513-524

%A J. C. Reynolds %T Three approaches to type structure %B LNCS 185: Mathematical Foundations of Software Development %I Springer %C Berlin %D 1985 %P 97-138

%A H. Richards %T An applicative programming bibliography %R Report %I Burroughs Corp., Austin Research Center %C Austin, Texas %D 1985

%A H. R. Rogers %T Theory of Recursive Functions and Effective Computability %I McGraw-Hill %C New York %D 1967

%A V. Royer %T Deriving stack semantics congruent to standard denotational semantics %B LNCS 182: Proc. 2nd Symp. on Theoretical Aspects of Comp. Sci. %I Springer %C Berlin %D 1985 %P 299-309

%A R. Rustin, ed. %T Formal Semantics of Programming Languages %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1972

%A L. E. Sanchis %T Data types as lattices: retractions, closures, and projections %J RAIRO Informatique theorique %V 11 %D 1977 %P 329-344

%A D. A. Schmidt %T State transition machines for lambda-calculus expressions %B LNCS 94: Semantics-Directed Compiler Generation %I Springer %C Berlin %D 1980 %P 415-440

%A D. A. Schmidt %T Detecting global variables in denotational specifications %J ACM Trans. on Prog. Lang. and Sys. %V 7 %D 1985a %P 299-310

%A D. A. Schmidt %T An implementation from a direct semantics definition %B LNCS: Proc. Workshop on Programs as Data Objects %I Springer %C Berlin %D 1985b

%A D. S. Scott %T The lattice of flow diagrams %B LNM 188: Semantics of Algorithmic Languages %E E. Engeler %I Springer %C Berlin %P 311-366 %D 1970

%A D. S. Scott %T Outline of a mathematical theory of computation %R Tech. monograph PRG-2 %I Programming Research Group, Univ. of Oxford %D 1971

%A D. S. Scott %T Continuous lattices %B LNM 274: Proc. Dahlhousie Conf. %I Springer %C Berlin %P 97-136 %D 1972

%A D. S. Scott %T Data types as lattices %J SIAM J. of Computing %V 5 %D 1976 %P 522-587

%A D. S. Scott %T Lectures on a mathematical theory of computation %R Report PRG-19, Programming Research Group, Univ. of Oxford %D 1980a

%T Relating theories of the lambda calculus %A D. S. Scott %B To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism %E J. P. Seldin %E J. R. Hindley %I Academic Press %C New York %D 1980b %P 403-450

%A D. S. Scott %T Domains for denotational semantics %B LNCS 140: Proc. 9th ICALP %I Springer %C Berlin %D 1982a %P 577-613

%A D. S. Scott %T Some ordered sets in computer science %B Ordered Sets %E I. Rival %I D. Reidel Pub. %D 1982b %P 677-718

%A D. S. Scott %A C. Strachey %T Towards a mathematical semantics for computer languages %R Tech. monograph PRG-6 %I Programming Research Group, Univ. of Oxford %D 1971

%A R. Sethi %T Control flow aspects of semantics-directed compiling %J ACM Trans. on Prog. Lang. and Systems %V 5 %D 1983 %P 554-596

%A R. Sethi %T Circular expressions: elimination of static environments %B LNCS 115: Proc. 8th ICALP %I Springer %D 1981 %C Berlin %P 378-392

%A R. Sethi %A A. Tang %T Transforming direct into continuation semantics for a simple imperative language %D 1978 %R Unpublished manuscript

%A R. Sethi %A A. Tang %T Constructing call-by-value continuation semantics %J J. of the ACM %V 27 %D 1980 %P 580-597

%A M. B. Smyth %T Effectively given domains %J Theoretical Comp. Science %V 5 %D 1977 %P 257-274

%A M. B. Smyth %T Powerdomains %D 1978 %P 23-36 %J J. of Computer and System Sciences

%A M. B. Smyth %T Power domains and predicate transformers: a topological view %B LNCS 154: Proc. 10th ICALP %I Springer %C Berlin %D 1982 %P 662-675

%A M. B. Smyth %T The largest cartesian closed category of domains %J Theoretical Computer Science %V 27 %D 1983 %P 109-120

%A M. B. Smyth %A G. D. Plotkin %T The category-theoretic solution of recursive domain equations %J SIAM J. of Computing %V 11 %D 1982 %P 761-783

%A G. L. Steele %T Debunking the ‘expensive procedure call’ myth %B Proc. ACM Annual Conf. %D 1977 %P 153-162

%A G. L. Steele %A G. J. Sussman %T LAMBDA: the ultimate imperative %R AI memo 353 %I AI Lab., MIT %D 1976a

%A G. L. Steele %A G. J. Sussman %T LAMBDA: the ultimate declarative %R AI Memo 379 %I AI Lab., MIT %D 1976b

%A G. L. Steele %A G. J. Sussman %T The revised report on SCHEME %R AI Memo 452 %I MIT %C Cambridge, Mass. %D 1978

%A A. Stoughton %R Ph.D. thesis %I Computer Science Dept., University of Edinburgh %C Edinburgh, Scotland %D 1986

%A J. E. Stoy %T Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory %I MIT Press %C Cambridge, Mass. %D 1977

%A J. E. Stoy %T The congruence of two programming language definitions %J Theoretical Comp. Science %V 13 %D 1981 %P 151-174

%A J. E. Stoy %T Some mathematical aspects of functional programming %B Functional Programming and its Applications %E J. Darlington, et. al. %I Cambridge Univ. Press %C Cambridge %D 1982 %P 217-252

%A C. Strachey %T Towards a formal semantics %B Formal Language Description Languages %I North-Holland %C Amsterdam %D 1966 %P 198-220 %E T.B. Steele

%A C. Strachey %T Fundamental concepts in programming languages %R Unpublished manuscript %I Programming Research Group, Univ. of Oxford %D 1968

%A C. Strachey %T The varieties of programming language %R Tech. monograph PRG-10 %I Programming Research Group, Univ. of Oxford %D 1973

%A C. Strachey %A C. P. Wadsworth %T Continuations: a mathematical semantics for handling full jumps %R Tech. monograph PRG-11 %I Programming Research Group, Univ. of Oxford %D 1974

%A R. D. Tennent %T Mathematical semantics of SNOBOL4 %B Proc. 1st ACM Symp. on Prin. of Prog. Lang. %C Boston %D 1973 %P 95-107

%A R. D. Tennent %T The denotational semantics of programming languages %J Comm. of the ACM %D 1976 %V 19 %P 437-452

%A R. D. Tennent %T A denotational definition of the programming language Pascal %R Tech. report 77-47 %I Department of Computing and Information Sciences, Queen’s Univ. %C Kingston, Ontario %D 1977a

%A R. D. Tennent %T Language design methods based on semantic principles %J Acta Informatica %V 8 %D 1977b %P 97-112

%A R. D. Tennent %T On a new approach to representation-independent data classes %J Acta Informatica %V 8 %D 1977c %P 315-324

%A R. D. Tennent %T Principles of Programming Languages %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1981

%A R. D. Tennent %T Semantics of inference control %B LNCS 140: Proc. 9th ICALP %I Springer %C Berlin %D 1982 %P 532-545

%A J. Thatcher %A E. Wagner %A J. Wright %T More on advice on structuring compilers and proving them correct %B LNCS 71: Proc. 6th ICALP %I Springer %C Berlin %D 1979 %P 596-615

%A D. Turner %T A new implementation technique for applicative languages %J Software Practice and Experience %V 9 %P 31-49 %D 1979

%A S. R. Vegdahl %T A survey of proposed architectures for the execution of functional languages %J IEEE Trans. on Computers %V c-33 %D 1984

%A T. Vickers %T Quokka: a translator generator using denotational semantics %R Report %I Computer Science Dept., University of New South Wales %C Kensington, Australia %D 1985

%A J. Vuillemin %T Correct and optimal implementations of recursion in a simple programming language %J J.

of Computer and System Sciences %D 1974 %V 9 %P 1050-1071

%A C. P. Wadsworth %T The relation between computational and denotational properties for Scott’s D∞-models of the lambda-calculus %J SIAM J. of Computing %V 5 %D 1976 %P 488-521

%A C. P. Wadsworth %T Approximate reductions and lambda calculus models %J SIAM J. of Computing %V 7 %D 1978 %P 337-356

%A C. P. Wadsworth %R Postgraduate lecture notes on domain theory %I Computer science dept., Univ. of Edinburgh %D 1978

%A M. Wand %T Fixed point constructions in order-enriched categories %J Theoretical Computer Science %V 8 %D 1979 %P 13-30

%A M. Wand %T Continuation-based program transformation strategies %J J. of the ACM %D 1980a %V 27 %P 164-180

%A M. Wand %T Induction, Recursion, and Programming %I Elsevier North Holland %C New York %D 1980b

%A M. Wand %T Semantics-directed machine architecture %B ACM Symp. on Prin. of Prog. Lang. %D 1982a %P 234-241

%A M. Wand %T Deriving target code as a representation of continuation semantics %J ACM Trans. on Prog. Lang. and Systems %V 4 %D 1982b %P 496-517

%A M. Wand %T Different advice on structuring compilers and proving them correct %I Computer Science Dept., Indiana University, Bloomington %D 1982c

%A M. Wand %T Loops in combinator-based compilers %J Information and Control %V 57 %D 1983 %P 148-164

%A M. Wand %T A types-as-sets semantics for Milner-style polymorphism %B Proc. ACM Conf. on Princ. of Prog. Lang. %C Salt Lake City, Utah %D 1984a %P 158-164

%A M. Wand %T A semantic prototyping system %B Proc. SIGPLAN ‘84 Symp. on Compiler Construction %C Montreal %D 1984b %P 213-221

%A M. Wand %T Embedding type structure in semantics %B Proc. 12th ACM Symp. on Princ. of Prog. Lang. %C New Orleans %D 1985a %P 1-6

%A M. Wand %T From interpreter to compiler: a representational derivation %B LNCS: Proc. Workshop on Programs as Data Objects %I Springer %C Berlin %D 1985b

%A S. Ward %T Functional domains of applicative languages %R Project MAC Report TR-136 %I MIT %C Cambridge, Mass. %D 1974

%A P. Wegner %T The Vienna Definition language %J ACM Computing Surveys %V 4 %D 1972a %P 5-63

%A P. Wegner %T Programming language semantics %B Formal Semantics of Programming Languages %E R. Rustin %I Prentice-Hall %C Englewood Cliffs, N.J. %D 1972b %P 149-248

%Q vanWijngaarden, A., et. al. %T Report on the algorithmic language ALGOL68 %J Numer. Math. %V 14 %D 1969 %P 79-218

%A G. Winskel %T Events in computation %R Ph.D. thesis, Univ. of Edinburgh %C Scotland %D 1980