

- man 84]: **H.Abelson, G.J.Sussman**: Structure and Interpretation of Computer Programs. MIT mress **1984**.
- [Addison et al 65]: **J.Addison, L.Henkin, A.Tarski (Eds)**: Symposium on the Theory of Models. North-Holland **1965**.
- [Amit 84]: **N.Amit**: A Different solution to improving readability of deeply-nested if-then-else control structures. SIGPLAN 19,1,Jan **1984**, P.24-29.
- [Apt 83] **K.R.Apt**: A Static Analysis of CSP Programs. in [LNCS v.164] P.1-17, **1983**.
- [Arsac 79]: **J.Arsac** : Syntactic Source to Source Program Transformations and Program Manipulation. Comm. A.C.M. **1979** P.43-54.
- [Arsac 82]: **J.Arsac** : Transformation of Recursive Procedures. in [Neel 82] P.211-265 **1982**.
- [Back 80]: **R.J.R.Back**: Correctness Preserving Program Refinements: Proof Theory and Applications. Mathematical Centre Tracts **131**. Mathematisch Centrum **1980**.
- [Back 80a]: **R.J.R.Back**: Semantics of Unbounded Nondeterminism. in [LNCS v85] p.51 **1980**.
- [Barron 68]: **D.W.Barron**: Recursive Techniques in Programming. Macdonald Computer Monographs **1968**.
- [Barstow et al 84]: **R.Barstow, H.E.Shrobe, E.Sandwall**: Interactive Programming Environments McGraw-Hill **1984**.
- [Bauer 76]: **F.L.Bauer**: Programming as an Evolutionary Process. in [LNCS v46] P.153-182 **1976**.
- [Bauer 76a]: **F.L.Bauer**: A Philosophy of Programming. in [LNCS v46] P.194-241 **1976**.
- [Bauer 79]: **F.L.Bauer**: Program Development By Stepwise Transformations -the Project CIP. in [LNCS v69] P.237-266 **1979**.
- [Bauer et al 79]: **F.L.Bauer, M.Broy, H.Partsch, P.Pepper, H.Wossner**: Systematics of Transformation Rules in [LNCS v69] P.273-289 **1979**.
- [Bauer & Wossner 82]: **F.L.Bauer, H.Wossner**: Algorithmic Language and Program Development. Springer-Verlag **1982**.
- [Bernstein 66]: **A.J.Bernstein**: Analysis of Programs for Parallel Processing. IEEE trans. Electronic Computers 15, P.757-763 **1966**.
- [Bird 77]: **R.Bird**: Notes on Recursion Removal. Comm. A.C.M. 20,6,June **1977**, P.434-439.
- [Bochmann 73]: **G.V.Bochmann** Multiple exits from a loop without the goto. Comm. A.C.M. 16,7,July **1973**, P.443-444.
- [Bohm & Jacopini 66]: **C.Bohm, G.Jacopini**: Flow Diagrams, Turing Machines and Languages with only Two Formation Rules. Comm. A.C.M. 9,5, May **1966**, P.366-371
- [Brinch Hansen 78]: **P. Brinch Hansen**: Distributed Processes: A Concurrent Programming Concept Comm. A.C.M. 21,11, Nov **1978**, P.934-941.
- [Brooks 83]: **S.D.Brooks**: A Model for Communicating Sequential Processes. D.Phil Thesis. Oxford University **1983**.

- [Brooks et al 84]: **S.D.Brookes, C.A.R.Hoare, A.W.Roscoe**: A Theory of Communicating Sequential Processes. Forthcoming **1984**.
- [Brooks 87]: **F.P.Brooks**: No Silver Bullet. IEEE Computer Apr **1987**, P.10-19.
- [Broy et al 79]: **M.Broy, R.Gnatz, M.Wirsig**: Semantics of Nondeterminism and Noncontinuous Constructs. in [LNCS v.69] P.563-592, **1979**.
- [Broy & Schmidt 82]: **M.Broy, G.Schmidt (Eds)**: Theoretical Foundations of Programming Methodology. Int. Summer School, Marktoberdorf 1981, Dordrecht: Reidel **1982**.
- [Broy 82]: **M.Broy** : A Fixed Point Approach to Applicative Multiprogramming. in [Broy & Schmidt 82], P.565-622, **1982**.
- [Buhr 85] **P.A.Buhr**: A Case for Teaching Multi-exit Loops to Beginning Programmers. SIGPLAN 20,11,Nov **1985**, P.14-22.
- [Burstall 69]: **R.M.Burstall**: Proving Properties of Programs by Structural Induction. Computer Journal 12, P.41-48, **1969**.
- [Burstall & Darlington 75]: **R.M.Burstall, J.Darlington**: Some Transformations for Developing Recursive Programs in Proc. 1975 Int. Conf. on Reliable Software, 21st-23rd April, ACM SIGPLAN Notices 10,6 June **1975** P.465-472.
- [Burstall & Darlington 77]: **R.M.Burstall, J.Darlington**: A Transformation System for Developing Recursive Programs J.A.C.M. 24,1 Jan **1977** P.44-67.
- [Calliss 88]: **F.W.Calliss**: Problems with Automatic Restructurers. University of Durham Computer Science Technical Report 88/6, **1988**.
- [Church 51]: **A.Church**: The Calculi of Lambda Conversion. Annals of Mathematical Studies, 6, **1951**.
- [Cooper 66]: **D.C.Cooper**: The Equivalence of Certain Computations. Computer Journal 9, P.45-52, **1966**.
- [Correll 78]: **C.H.Correll**: Proving Programs Correct through Refinement. Acta Informatica 9 P.121-132 Springer-Verlag **1978**.
- [Covington 84]: **M.A.Covington** A Pedagogical Disadvantage of REPEAT and WHILE. SIGPLAN 19,8,Aug **1984**, P.85-86.
- [Dahl et al 71]: **O.J.Dahl, E.W.Dijkstra, C.A.R.Hoare**: Structured Programming. Academic Press **1971**.
- [Darlington & Burstall 76]: **J.Darlington, R.M.Burstall**: A System which Automatically Improves Programs. in Acta Informatica 6,1, P.41-60 Springer-Verlag,**1976**.
- [Davey & Priestley 89]: **B.A.Davey, H.A.Priestley**: Partition-Induced Natural Dualities for Varieties of Distributive p-Algebras. to be submitted to Discrete Mathematics. **1989**.
- [den Hartog 83] **E.H.den Hartog**: DO-SELECT revisited. SIGPLAN 18,3,March **1983**, P.32-35.
- [de Bakker 76]: **J.W.de Bakker**: Semantics and Termination of Nondeterministic Programs. in [Michaelson & Milner 76] P.436-477, **1976**.

- [de Bakker & van Vilet 81]: **J.W.de Bakker, J.C.van Vilet:** Algorithmic Languages. North-Holland **1981**.
- [de Roever 78]: **W.P.de Roever:** On Backtracking and Greatest Fixpoints. in [Newhold 78] P.621-636 **1978**.
- [Dijkstra 60]: **E.W.Dijkstra:** Recursive Programming. Nuerische Math. 2, P.312-318 **1960**.
- [Dijkstra 68]: **E.W.Dijkstra:** Go to Statement Considered Harmful. Comm. A.C.M. 11,3, P.147-148, 538, 541. **1968**.
- [Dijkstra 71]: **E.W.Dijkstra:** Notes on Structured Programming. in [Dahl et al 71] **1971**.
- [Dijkstra 72]: **E.W.Dijkstra:** The Humble Programmer. Comm. A.C.M. 15,10, P.859-866. **1972**.
- [Dijkstra 76]: **E.W.Dijkstra:** A Discipline of Programming. Prentice-Hall **1976**.
- [Dijkstra 79]: **E.W.Dijkstra:** On the Interplay Between Mathematics and Programming. in [LNCS v.69] P.35-46 **1979**.
- [Dijkstra et al 76]: **E.W.Dijkstra, L.Lamport, A.J.Martin, C.S.Scholten, E.F.M.Steffens:** On-the-Fly Garbage Collection: an Exercise in Cooperation. in [LNCS v.46] P.43-53, **1976**.
- [Earley 71]: **J.Earley:** Towards an Understanding of Data Structures. Comm. A.C.M. 14,10,Oct **1971**, P.617-627.
- [Engeler 68]: **E.Engeler:** Formal Languages: Automata and Structures. Markham. Chicago, **1968**.
- [Feiler & Medine-Mora 80]: **P.Feiler, R. Medine-Mora:** An Incremental Programming Environment. Technical Report CMU-C5-80-126 Comp. Sci. Div. Carnegie-Mellon Univ. Pittsburgh April **1980**.
- [Francez 77]: **N.Francez:** Another Advantage of Keyword Notation for Parameter Communication with Subprograms. Comm. A.C.M. **1977** P.604.
- [Gill 65]: **S.Gill:** Automatic Computing: Its Problems and Prizes. Comp. J. 8, P.177-189, **1965**.
- [Gries 77] **D.Gries:** An Exercise in Proving Parallel Programs Correct. Comm. A.C.M. **1977** P.237-266.
- [Gries 79]: **D.Gries:** The Schorr-Waite Graph Marking Algorithm. in [LNCS v.69] P.58-64 **1979**. and Acta Informatica 11, P.223-232 1979.
- [Gries 79a]: **D.Gries:** Is Sometimes Ever Better than Always? in [LNCSv69], P.113-124, **1979**.
- [Gries 81]: **D.Gries:** The Science of Programming. Springer-Verlag **1981**.
- [Griffiths 76]: **M.Griffiths:** Program Production by Successive Transformation. in [LNCS v.46] P.125-152 **1979**.
- [Griffiths 79]: **M.Griffiths:** Development of the Schorr-Waite Algorithm. in [LNCS v.69] P.464-471 **1979**.
- [Gutttag 77]: **J.V.Gutttag:** Abstract Data Types and the Development of Data Structures. Comm. A.C.M. 20,6,June **1977**, P.396-404.

- [Gutttag 79]: **J.V.Gutttag**: Notes on Type Abstraction. in [LNCS v.69], P.593-616, **1979**.
- [Gutttag & Horning 78]: **J.V.Gutttag, J.J.Horning**: The Algebraic Specification of Abstract Data Types. Acta Informatica 10,1, P.27-52, **1978**.
- [Haskell 75]: **R.Haskell**: Efficient Implementation of a Class of Recursively Defined Functions. Comp. J. 18,1, Jan **1975**, P.23-29.
- [Hayes 87]: **I.J.Hayes**: Specification Case Studies. Prentice-Hall International, **1987**.
- [Henderson 84]: **P.Henderson (Ed.)**: Practical Software Development Environments. Proc.A.C.M. SIGSOFT/SOPLAN Software Engineering Symp. SIGPLAN Notices 19,5 May **1984**.
- [Henderson & Snowdon 72]: **P.Henderson, R.Snowdon**: An Experiment in Structured Programming. BIT 12,1, **1972**, P.38-52.
- [Hill 82] **G.P.Hill**: An Improvement over Deeply-Nested if-then-else Control Structures. SIGPLAN 17,8,Aug **1982**, P.52-56.
- [Hoare 69]: **C.A.R.Hoare**: An Axiomatic Basis for Computer Programming. Comm. A.C.M. **1969** P.576-580.
- [Hoare 72]: **C.A.R.Hoare**: Proof of Correctness of Data Representations. Acta Informatica 1, P.271-278, Springer-Verlag **1972**.
- [Hoare 73]: **C.A.R.Hoare**: Hints on Programming Language Design. Invited Address at SIGACT/SIGPLAN Symp. on Principles of Programming Languages. Boston, Oct 1-3 **1973**.
- [Hoare 75]: **C.A.R.Hoare**: Recursive Data Structures. Int. J. Comp. Inf. Sci. 4, P.105-132, **1975**.
- [Hoare 81]: **C.A.R.Hoare**: A Model for Communicating Sequential Processes. Oxford University Programming Research Group Technical Monograph **PRG-22 1981**.
- [Hoare 81b]: **C.A.R.Hoare**: The Emperor's Old Clothes. The 1980 Turing Award Lecture, Comm. A.C.M. 24, 2, Feb **1981**, P.75-83.
- [Hoare et al 87]: **C.A.R.Hoare, I.J.Hayes, H.E.Jifeng, C.C.Morgan, A.W.Roscoe, J.W.Sanders, I.H.Sorensen, J.M.Spivey, B.A.Sifren**: Laws of Programming. Comm. A.C.M. 30, 8, Aug **1987**. P.672-686.
- [Hoare & Wirth 73]: **C.A.R.Hoare, N.Wirth**: An Axiomatic Definition of the Programming language PASCAL. Acta Informatica 2, **1973**. P.335-355.
- [Irluk 77]: **J.Irluk**: Translating Some Recursive Procedures into Iterative Schemes. Programmation, B.Robinet Ed., Dunod, Paris, **1977** P.31-52.
- [Jones 86]: **C.B.Jones**: Systematic Software Development using VDM. Prentice-Hall International **1986**.
- [Karp 64]: **C.R.Karp**: Languages with Expressions of Infinite Length. North-Holland **1964**.
- [Kernighan & Plauser 74]: **B.W.Kernighan, P.J.Plauser**: Programming Style: Examples and Counterexamples. Computing Surveys 6,4 Dec **1974**, P.303-319.
- [Kernighan & Plauser 76]: **B.W.Kernighan, P.J.Plauser**: Software Tools. Addison-Wesley **1976**.

- [Kibler et al 77]: **D.F.Kibler, J.M.Neighbors, T.A.Standish**, Program Manipulation via an Efficient Production System. SIGPLAN Notices 12,8, Aug **1977**, P.163-173.
- [Knuth 74]: **D.E.Knuth**: Structured Programming with the GOTO Statement. Computing Surveys 6,4 P.261-301 **1974**.
- [Kosaraju 74]: **S.R.Kosaraju**: Analysis of Structured Programs. J. of Comp. and Sys. Sci. 9,3, Dec **1974**, P.232-255.
- [Kowalski 79]: **R.Kowalski**: Algorithm = Logic + Control. Comm. A.C.M. 22,7, July **1979**, P.424-436.
- [Kowlatowski 79]: **T.Kowlatowski**: Data Structure and Correctness of Programs. J.A.C.M. 26,2, **1979**, P.283-301.
- [Kuiper 81]: **R.Kuiper**: An Operational Semantics for Bounded Nondeterminism Equivalent to a Denotational one. in [de Bakker & van Vilet 81] P.373-398, **1981**.
- [Landin 66]: **P.J.Landin**: The Next 700 Programming Languages. Comm. A.C.M. 9,3, March **1966**, P.157-166.
- [Langmaack & Olderog 80]: **H.Langmaack, E.R.Olderog**: Present Day Hoare-Like Systems for Programming Languages with Procedures: Power, Limits and most likely Extensions in [LNCS v.85] **1980**.
- [Leintz & Swanson 80]: **B.P.Leintz, E.B.Swanson**: Software Maintenance Management: Addison-Wesley **1980**.
- [Ledgard & Marcotty 75]: **H.F.Ledgard, M.Marcotty**: A Genealogy of Control Structures. Comm. A.C.M. 18,11, Nov **1975**, P.629-639.
- [Liskov et al 77]: **B.Liskiv, A.Snyder, R.Atkinson, C.Schaffert**: Abstraction Mechanisms in CLU. Comm. A.C.M. 20,8, **1977**, P.54-579.
- [LNCS v.46]: **F.L.Bauer, K.Samelson (Eds)**: Language Hierarchies and Interfaces. Lecture Notes in Computer Science. Volume 46. Springer Verlag **1976**.
- [LNCS v.52]: **A.Salomaa, M.Steinby (Eds)**: Automata Languages and Programming. Lecture Notes in Computer Science. Volume 52. Springer Verlag **1977**.
- [LNCS v.69]: **G.Goos, H.Hartmanis (Eds)**: Program Construction. Lecture Notes in Computer Science. Volume 69. Springer Verlag **1979**.
- [LNCS v.85]: **J.W.Bakker, J.van Leeuwen (Eds.)**: Automata Languages and Programming. Lecture Notes in Computer Science. Volume 85. Springer Verlag **1980**.
- [LNCS v.86]: **D.Bjorner(Ed.)**: Abstract Software Specification. Lecture Notes in Computer Science. Volume 86. Springer Verlag **1980**.
- [LNCS v.137]: **M.Dezami-Ciancaglini, U.Montanari**: International Symposium on Programming. Lecture Notes in Computer Science. Volume 137. Springer Verlag **1982**.
- [Loveman 77]: **D.B.Loveman**: Program Improvement by Source-to -Source Transformation. J.A.C.M. 24,1 Jan **1977**, P.121-145.

- [Majester 77]: **M.E.Majester**: Limits of the “Algebraic” Specification of Abstract Data Types. SIGPLAN Notices 12,10, Oct **1977**, P.37-42.
- [Manna & McCarthy 69]: **Z.Manna, J.McCarthy**: Properties of Programs and Partial Function Logic. in [Michie 69] P.27-37, **1969**.
- [Manna et al 73]: **Z.Manna, S.Ness, J.Vuillemin**: Inductive Methods for Proving Properties of Programs. Comm. A.C.M. 16, P.491-502, **1973**.
- [Manna & Waldinger 78]: **Z.Manna, R.Waldinger**: Is “Sometime” Sometimes Better than “Always”? Intermittent Assertions in Proving Program Correctness. Comm. A.C.M. 21, P.159-172, **1978**.
- [McCarthy 60]: **J.McCarthy**: Recursive Functions of Symbolic Expressions and their Computation by Machine. Comm. A.C.M. 3, P.184, **1960**.
- [McKeag & Macnaughton 80]: **R.M.McKeag, A.M.Macnaughten**: On the Construction of Programs. Cambridge University Press **1980**.
- [Michaelson & Milner 76]: **S.Michaelson, R.Milner**: Automata, Languages and Programming, Proceedings. Edinborough Univ. Press **1976**
- [Michie 69]: **D.Michie**: Machine Intelligence Vol. 5 Edinburgh: Edinburgh University Press **1969**.
- [Milner 80]: **R.Milner**: A Calculus of Communicating Systems. Lecture Notes in Computer Science. Volume 92. Springer Verlag **1980**.
- [Morris 71]: **J.H.Morris**: Another Recursion Induction Principle. Comm. A.C.M. 14,5, May **1971**, P.351-354.
- [Morris 82]: **J.H.Morris**: A Proof of the Schorr-Waite Algorithm. in [Broy 82] **1982**.
- [Naur 66]: **P.Naur**: Proof of Algorithms by General Snapshots. BIT 6,4, **1966**, P.310-316.
- [Naur 69]: **P.Naur**: Programming by Action Clusters. BIT 9, P.250-258, **1969**.
- [Neel 82]: **D.Neel (Ed)**: Tools and Notations for Program Construction. Cambridge University Press **1982**.
- [Neuhold 78]: **E.J.Neuhold (Ed.)**: Formal Description of Programming Constructs. North-Holland **1978**.
- [Newman 83] **P.S.Newman**: If-then-else again. SIGPLAN 18,12,Dec **1983**, P.106-111.
- [Owicki 75]: **S.Owicki**: Axiomatic Proof Techniques for Parallel Programs. Ph.D. Thesis TR 75-251 Dept. Compnr. Sci., Cornell U., Ithaca, N.Y., July **1975**.
- [Owicki & Gries 75]: **S.Owicki, D.Gries**: An Axiomatic Proof Technique for Parallel Programs. Cornell Univ. Sept **1975**.
- [Parnas 72]: **D.L.Parnas**: A Technique for Software Module Specification with Examples. Comm. A.C.M. 15,5, May **1972**, P.350-336.
- [Parnas 79]: **D.L.Parnas**: Designing Software. IEEE Trans. Soft. Eng. 5,2, Mar **1979**, P.128-138.
- [Partsch & Pepper 76]: **H.Partsch, P.Pepper**: A Family of Rules for Recursion Removal. Inf. Processing Letters 5,6, Dec **1976**, P.174-177.

- [Pepper 79]: **P.Pepper**: A Study on Transformational Semantics. in [LNCS v.69] P.232-405, **1979**.
- [Pepper 84]: **P.Pepper**: Program Transformation and Programming Environments. NATO ASI (Advanced Studies Institute) July **1984**.
- [Peterson et al 73]: **W.W.Peterson, T.Kasami, N.Tokura**: On the Capabilities of While, Repeat and Exit Statements. Comm. A.C.M. 16,8, Aug **1973**, P.503-512.
- [Plotkin 76]: **G.D.Plotkin**: A Powerdomain Construction. SIAM J. Computing 5,3, P.452-487, Sept **1976**.
- [Plotkin 80]: **G.D.Plotkin**: Dijkstra's Predicate Transformers and Smyth's Powerdomains. in [LNCS v.85], P.527-553, **1980**.
- [Post 43]: **E.L.Post**: Formal Reduction of the General Combinatorial Decision Problem. Am. J. Math. 65, P.197-215, **1943**.
- [Rice 65]: **H.G.Rice**: Recursion and Iteration. Comm. A.C.M. 8,2, Feb **1965**, P.114-115.
- [Robinson & Levitt 77]: **L.Robinson, K.N.Levitt**: Proof Techniques for Hierarchically Structured Programs. Comm. A.C.M. 20,4, April **1977**.
- [Roscoe 82]: **A.W.Roscoe**: Notes on Distributed Computing. Unpublished Lecture Notes **1982**.
- [Roscoe 83]: **A.W.Roscoe**: A Denotational Semantics for occam. Programming Research Group Monograph, Oxford **1983**.
- [Roscoe 84]: **A.W.Roscoe**: Modeling Parallel Computation. Seminar. Programming Research Group. Oxford University. **1984**.
- [Schorr & Waite 67]: **H.Schorr W.M.Waite**: An Efficient Machine-Independent Procedure for Garbage Collection in Various List Structures. Comm. A.C.M. Aug. **1967**.
- [Scott 65]: **D.Scott**: Logic with Denumerably Long Formulas and Finite Strings of Quantifiers. in [Addison et al 65] P.329-341. **1965**.
- [Shamir & Wadge 77]: **A.Shamir W.W.Wadge**: Data Types as Objects. in [LNCS v.52] P.465-479. **1977**.
- [Smyth 78]: **M.B.Smyth**: Power Domains. J. of Comp. and Sys. Sci. 16,1, P.23-36, **1978**.
- [Solway et al 83]: **E.Solway, J.Bonar, K.Ehrlich**: Cognitive Strategies and Looping Constructs: An Empirical Study. Comm. A.C.M. 26,11,Nov **1983**, P.853-860.
- [Stoy 77]: **J.E.Stoy**: Denotational Semantics: the Scott-Strachy Approach to Programming Language Theory. Cambridge Mass.:MIT Press **1977**.
- [Strong 71]: **H.R.Strong**: Translating Recursion Equations into Flow Charts. J.Computer Systems Sci. 5, **1971**, P.254-285.
- [Taylor 84]: **D.Taylor**: An Alternative to Current Looping Syntax. SIGPLAN Notices 19,12, P.48-53, Dec **1984**.
- [Teitelbaum & Reps 81]: **T.Teitelbaum, T.Reps**: The Cornell Program Synthesizer. Comm. A.C.M. 24,9, **1981**, P.563-573.

[Topor 79]:

- .W.**Topor**: The Correctness of the Schorr-Waite List Marking Algorithm. Acta Informatica 11, **1979**, P.211-221.
- [Townley 82]: **J.A.Townley**: The use of Transformations to Implement an Algorithm. in [LNCS v.137], P.381-406, **1982**.
- [Vuillemin 74]: **J.Vuillemin**: Correct and Optimal Implementations of Recursion in a Simple Programming Language. J.Computer Systems Sci. 9 P.332-354, **1974**.
- [Ward 88]: **M.Ward**: Transforming a Program into a Specification. University of Durham Computer Science Technical Report 88/1, **1988**.
- [Ward 89a]: **M.Ward**: Constructive Specifications and Program Transformation. University of Durham Computer Science Technical Report 89/2, **1989**.
- [Ward 89b]: **M.Ward**: Development of a Program: FIND. University of Durham Computer Science Technical Report 89/3, **1989**.
- [Ward 89c]: **M.Ward**: Data Types and Prototypes. University of Durham Computer Science Technical Report 89/4, **1989**.
- [Wegbreit 76]: **B.Wegbreit**: Goal-Directed Program Transformations. IEEE Trans. on Software Engineering 2,2 June **1976**, P.69-80.
- [Wharton 83]: **R.M.Wharton**: A Note on Types and Prototypes. SIGPLAN 18,12,Dec **1983**, P.122-126.
- [Whitelaw 85]: **M.W.Whitelaw**: Some Ramifications of the exit statement in Loop Control. SIGPLAN 20,8,Aug **1985**, P.99-106.
- [Williams 84]: **D.O.Williams**: Structured Transfer of Control. SIGPLAN 19,10,Oct **1984**, P.46-51.
- [Wirth 71]: **N.Wirth**: Program Development by Stepwise Refinement. Comm. A.C.M. **1971** P.221-227.
- [Wirth 73]: **N.Wirth**: Systematic Programming. Prentice-Hall **1973**.
- [Wosser et al 79]: **H.Wosser, P.Pepper, H.Partsch, F.L.Bauer**: Special Transformation Techniques.in [LNCS v.69] P.290-321 **1979**.
- [Yelowitz & Duncan 77]: **L.Yelowitz, A.G.Duncan**: Abstractions, Instantiations and Proofs of Marking Algorithms. Proc. Symp. Artificial Intelligence and Programming Languages, SIGPLAN 12,8, **1977**, P.13-21.