List of publications by Jan A. Bergstra (September 15, 2005)

List of publications arranged by subject and per subject by date. Survey papers have been classified under miscellaneous. Submitted reports are listed in their subject category unless publication is not intended at this stage. Such reports and documents end up in the final section.

Recursion theory

[1] J.A. Bergstra. *Computability and continuity in finite types* Ph.D. Thesis Utrecht (Januari 1976).

[2] J.A .Bergstra. 1-envelopes of continuous functionals. *Recursive Function Theory: Newsletter*, 16, (August 1977).

[3] J.A. Bergstra and S.S. Wainer. The "real ordinal" of the 1-section of a continuous functional. *Abstract in J. Symbolic Logic*, 42(3), 1977.

[4] J.A. Bergstra. The continuous functionals and ^{2}E. In J.E. Fenstad, R.O. Gandy and G.E. Sacks eds. *Generalized Recursion Theory II.* North Holland (1978), p. 39-53.

[5] J.A. Bergstra. Datatypen bezien vanuit de recursietheorie. In J. van Vliet, ed. *MC-syllabus 37, Colloquium capita datastructuren* (1978), p. 157-170.

[6] J.A. Bergstra. Degrees of partial functions. *Notre Dame J. of Formal Logic, Vol. 19*, 1 (1978), p.152-154.

[7] J.A. Bergstra. Effective transformations on probabilistic data. *Zeitschrift fuer Mathematik und Grundlagen der Mathematik*, Bd. 25 (1979), p. 219-226.

[8] J.A. Bergstra. Recursion theory on processes. *International J. of Computer Mathematics, section A, vol. 7*, 2 (1979), p.119-128.

[9] J.A. Bergstra. Dynamic recursion theory. Abstract in *J. Symbolic Logic, vol. 45*, 2 (1980), p. 387.

Lambda calculus and rewrite systems

[1] H.P. Barendregt, J.A. Bergstra, J.W. Klop and H. Volken. Representability in lambda algebras. *Indigationes Mathematicae, vol. 79* (5) (1976), p.377-387.

[2] J.A. Bergstra and J.W. Klop. Church-Rosser strategies in the lambda calculus. *Theoretical Computer Science 9* (1979), p. 27-38.

[3] J.A. Bergstra and J.W. Klop. Invertible terms in the lambda calculus. *Theoretical Computer Science 11* (1980), p.19-38.

[4] J.A. Bergstra and J.W. Klop. Strong normalisation and perpetual reduction in the lambda calculus. *Elektronische Informationsverarbeitung und Kybernetik 18* 7/8 (1982), p. 403-417.

[5] J.A. Bergstra and J.W. Klop. Conditional rewrite rules: confluence and termination. *J. of Comp. & Syst. Sci, Vol 32* 3 (1986), p. 323-362.

[6] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. Term rewriting systems with priorities. In P. Lescanne, ed. *Proc. Conf. on Rewriting Techniques and Applications.* Bordeaux 1987, Springer Lecture Notes in Computer Science 256 (1987), p. 83-94.

[7] J.C.M. Baeten, J.A. Bergstra, J.W. Klop and W.P. Weijland. Term rewriting with rule priorities. *Theoretical Computer Science 67* (1989), p. 283-301.

[8] J.A. Bergstra and J.W. Klop. *Semi-complete termherschrijfsystemen.* Kluwer Programmatuurkunde, Deventer (1987), 123 blz. (In Dutch)

[9] J.A. Bergstra, J.W. Klop and A. Middeldorp. *Termherschrijfsystemen.* Kluwer Programmatuurkunde, Deventer, (1989), 170 blz. (In Dutch)

Program verification

[1] K.R. Apt, J.A. Bergstra and L.G.L.T. Meertens. Recursive assertions are not enough - or are they? *Theoretical Computer Science 8* (1979), p.73-88.

[2] J.A. Bergstra and J.V. Tucker. Algebraically specified programming systems and Hoare's logic. In S.Even and O.Kariv, eds. *Proceedings of ICALP'81. Springer Lecture Notes in Computer Science 115* (1981), p.348-362.

[3] J.A. Bergstra, J. Tiuryn and J.V. Tucker. Floyd's principle, correctness theories and program equivalence. *Theoretical Computer Science 17* (1981), p. 112-149.

[4] J.A. Bergstra and J. Terlouw. A characterization of program equivalence in terms of Hoare's logic. In W.Brauer, ed.*The proceedings of the G-I Tagung. Springer Lecture Notes in Computer Science 123* (1981), p. 72-77.

[5] J.A. Bergstra and J. Terlouw. A propositional version of Hoare's logic. *Bulletin of the EATCS, no 14* (1981).

[6] J.A. Bergstra. A propositional version of Hoare's logic with modal operators. *Bulletin of the EATCS, no 16* (1982).

[7] J.A. Bergstra and J.V. Tucker. The field of algebraic numbers fails to possess even a nice sound, if relatively incomplete, Hoare-like logic for its while programs. *Theoretical Computer Science 17* (1982), p.303-315

[8] J.A. Bergstra and J.V. Tucker. Two theorems on the completeness of Hoare's logic. *Information Processing Letters 15* (1982), p.143-149.

[9] J.A. Bergstra and J.V. Tucker. On the refinement of specifications and the stability of Hoare's logic. In D.Kozen,ed. *Logic of programs. Springer lecture Notes in Computer Science 131* (1982), p.24-36.

[10] J.A. Bergstra and J.W. Klop. A formalized proof system for total correctness of while programs. In *Proceedings of ISI '82. Springer Lecture Notes in Computer Science 137* (1982)

[11] J.A. Bergstra, A. Chmielienska and J. Tiuryn. Hoare's logic is incomplete when it does not have to be. In D. Kozen, ed. Logic of programs. Springer Lecture Notes in Computer Science 131 (1982), p. 9-23.

[12] J.A. Bergstra and J.W. Klop. Formal proof systems for program equivalence. In D.Bjoerner, ed *Formal description of programming concepts II.* North Holland (1983), p. 289-302.

[13] J.A. Bergstra, A. Chmielienska and J. Tiuryn. Another incompleteness theorem of Hoare's logic. *Information and Control, Vol. 52* 2(1983), p. 159-171.

[14] J.A. Bergstra and J.V. Tucker. Expressiveness and the completeness of Hoare's logic. *J. Computer and Systems Sciences 25* (1983), p. 267-284.

[15] J.A. Bergstra and J.V. Tucker. Hoare's logic and Peano's arithmetic. *Theoretical Computer Science 22* (1983), p. 265-284.

[16] J.A. Bergstra and J.V. Tucker. Hoare's logic for programming languages with two datatypes. *Theoretical Computer Science 28* 1/2 (1984), p. 215-222.

[17] J.A. Bergstra and J. Tiuryn. PC-compactness, a necessary condition for the existence of sound and complete logics of partial correctness. In *Logic of programs. Springer Lecture Notes in Computer Science 164* (1984).

[18] J.A. Bergstra and J.W. Klop. Proving program inclusion using Hoare's logic. *Theoretical Computer Science 30* (1984), p.1-48.

[19] J.A. Bergstra and J.V. Tucker. The axiomatic semantics of while programs based on Hoare's logic. *Acta Informatica 21* (1984), p. 293-320.

Abstract data types

[1] J.A. Bergstra What is an abstract data type? *Information Processing Letters, Vol. 7* 1 (1978), p. 42-43.

[2] J.A. Bergstra and J.V. Tucker. On the adequacy of finite equational methods for data type specification. *ACM Sigplan Notices 14* 11 (1979), p. 13-18.

[3] J.A. Bergstra and J.V. Tucker. A natural data type with a finite equational final semantics specification, but no effective equational initial specification. *Bulletin of the EATCS 11* (1980), p. 23-33.

[4] J.A. Bergstra and J.V. Tucker. A characterization of computable data types by means of a finite, equational specification mehod. In J.W. de Bakker and J. van Leeuwen, eds. *Proceedings of ICALP'80. Springer lecture Notes in Computer Science 85* (1980), p. 76-90.

[5] J.A. Bergstra, M. Broy, J.V. Tucker and M. Wirsing. On the power of algebraic specifications . In J. Gruska and M. Chytil, eds. *Proceedings of MFCS'81. Springer lecure Notes in Computer Science 118* (1981), p. 193-204.

[6] J.A. Bergstra and J.-J. Ch. Meyer. An equational class of I/O-computable data structures that contains no recursive data structure. *Bulletin of the EATCS, no 13* (1981).

[7] J.A. Bergstra and J.-J. Ch. Meyer. I/O-computable data structures. *ACM Sigplan Notices 16* 4 (1981), p. 27-32.

[8] J.A. Bergstra. On the use of conditional data for data type specifications. In J. Bormann and W. Wechler, eds. *Vortraege zu Abstrakten Datatypen.* T.U. Dresden (1981), p. 15-18.

[9] J.A. Bergstra and J.-J. Ch. Meyer. Small specifications for large finite data structures. *International J. of Computer Mathematics, Vol. 9* 4 (1981), p. 305-320.

[10] J.A. Bergstra and J.-J. Ch. Meyer. Equational specifications of finite minimal unoids, using unary hidden functions only. *Fundamenta Informaticae, Vol. 5* 2 (1982), p. 143-170.

[11] J.A. Bergstra and J.-J. Ch. Meyer. A simple transfer lemma for abstract data type specifications. *Information Processing Letters 14* 2, p. 80-85.

[12] J.A. Bergstra and J.W. Klop. Algebraic specifications for parametrized data types with minimal parameter and target algebras. In *Proceedings of ICALP'82. Springer Lecture Notes in Computer Science 140* (1982), p. 23-34.

[13] J.A. Bergstra and J.V. Tucker. The completeness of the algebraic specification methods for computable data types. *Information and Control, Vol. 54* 3 (1983), p. 186-200.

[14] J.A. Bergstra and J.V. Tucker. Initial and final algebra semantics for data type specifications, two characterization theorems. *SIAM J. of Computing, Vol. 12* 2 (1983), p. 366-387.

[15] J.A. Bergstra and J.W. Klop Initial algebra specifications for parametrized data types. *Elektronische Informationsverarbeitung und Kybernetik 19* 1/2 (1983), p. 17-31.

[16] J.A. Bergstra and J.-J. Ch. Meyer. On specifying sets of integers. *MC IW 237/83 Elekronische Informationsverarbeitung und Kybernetik 20* 10/11 (1984), p. 531-541.

[17] J.A. Bergstra and J.V. Tucker. Algebraic specifications of computable and semi-computable data structures. *Theoretical Computer Science 50*(1987), p. 137-181.

[18] J.A. Bergstra, S. Mauw and F. Wiedijk. Uniform algebraic specifications of finite sorts with equality. *Int. J. Found. Comp. Sci. Vol. 2* 1 (1991), p. 43-65.

[19] J.A. Bergstra and J. Heering. Which data types have ω-complete initial algebra specifications? *Theoretical Computer Science*, 124:149–168, 1994.

[20] J.A. Bergstra and J.Heering. Homomorphism preserving algebraic specifications require hidden sorts. *Information and Computation*, 119(1):119–123, 1995.

[21] J.A. Bergstra and J.V. Tucker. The data type variety of stack algebras. *Annals of Pure and Applied Logic*, 73 p.11–36, 1995.

[22] J.A. Bergstra and J.V. Tucker. Equational specifications, complete term rewriting systems, and computable and semicomputable algebras. *Journal of the Association for Computing Machinery*, 42 (6) (1996) p. 1194-1230.

[23] J.A. Bergstra, T.B. Dinesh, J. Field, and J. Heering. Toward a complete transformational toolkit for compilers. In H.R. Nielson, editor, *Programming Languages and Systems (ESOP'96)*, volume 1058 of *Lecture Notes*, pages 92–107. Springer Verlag, 1996.

[24] J.A. Bergstra, T.B. Dinesh, J. Field, and J. Heering. Toward a complete transformational toolkit for compilers. *ACM TOPLAS 19*, 5 (1997) p. 639-684.

[25] J.A. Bergstra and J.V.Tucker. The rational numbers as an abstract datatype. Technical Report CSR-12-2005, Dept. of Comp. Sci. University of Wales at Swansea, 2005.

[26] J.A. Bergstra and J.V.Tucker. An elementary algebraic specification of the rational complex numbers. Technical Report CSR-xxx, Dept. of Comp. Sci. University of Wales at Swansea, 2005.

Process algebra

[1] J.A. Bergstra and J.W. Klop. The algebra of recursively defined processes and the algebra of regular processes. *Proceedings of ICALP'84. Springer Lecture Notes in Computer Science 172* (1984), p. 82-94. (Reprinted in A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, editors, *Algebra of Communicating Processes, Utrecht 1994*, pages 1–25, London Berlin Heidelberg, 1994. Workshops in Computing, Springer-Verlag.)

[2] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Control, Vol. 60* 1/3 (1984), p. 109-137.

[3] J.A. Bergstra, J.W. Klop and J.V. Tucker. Algebraic tools for system construction. In E.Clarke and D. Kozen, eds. *Logic of Programs. Springer Lecture Notes in Computer Science 164* (1984), p.34-45.

[4] J.A. Bergstra and J.V. Tucker. Top-down design and the algebra of communicating processes. *Science of Computer Programming 5* 2 (1984), p. 171-199.

[5] J.A. Bergstra, J.W. Klop and J.V. Tucker. Process algebra with asynchronous communication mechanisms. In S.D. Brookes, A.W. Roscoe and G. Winskel, eds. Proceedings Seminar on Concurrency. Springer Lecture Notes in Computer Science 197 (1985), p. 76-95.

[6] J.A. Bergstra and J.W. Klop. Process Theory based on Bisimulation Semantics. In J.W. de Bakker, J.W. de Roever and G. Rozenberg, eds. *Linear Time, Branching Time and partial Order Logics and Models for Concurrency. Springer lecture Notes in Computer Science 354* (1989), p. 50-122.

[7] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae IX* 2 (1986), p.127-168.

[8] J.A. Bergstra and J. Tiuryn. Process algebra semantics for queues. em Fundamenta Informaticae X (1987), p. 213-224.

[9] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. Ready trace semantics for concrete process algebra with priority operator. *British Computer Journal 30* 6 (1987), p. 498-506.

[10] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. In J.W. de Bakker, A.J. Nijman and P.C. Treleaven, eds. *Proceedings PARLE, Vol. II Parallel Languages. Springer Lecture Notes in Computer Science 259* (1987), p. 94-113.

[11] J.A. Bergstra. ACP with signals. In J. Grabowski, P. Lescanne and W. Wechler, eds. *Algebraic and Logic programming. Springer Lecture Notes in Computer Science 343* (1988), p. 11-20.

[12] J.A. Bergstra, J.W. Klop and E.-R. Olderog. Readies and failures in the algebra of communicating processes. *SIAM J. Computing, vol. 7* 6 (1988), p. 1134-1177.

[13] J.C.M. Baeten and J.A. Bergstra. Global renaming operators in concrete process algebra. *Information and Computation 78* 3 (1988), p. 205-245.

[14] J.C.M. Baeten and J.A. Bergstra. Recursive process definitions with the state operator. *Theoretical Computer Science 82* (1991), p. 285-302.

[15] J.A. Bergstra. A process creation mechanism in process algebra. In J.C.M. Baeten, ed. Applications of Process Algebra. Cambridge Tracts in Theoretical Computer Science 17. Cambridge University press (1990), p. 81-89.

[16] J.C.M. Baeten and J.A. Bergstra. Process algebra with a zero object. In J.C.M. Baeten and J.W. Klop, eds. *CONCUR'90 Theories of Concurrency, Unification and Extension. Springer lecture Notes in Computer Science 458* (1990), p. 83-98.

[17] J.C.M. Baeten and J.A. Bergstra. Process algebra with signals and conditions. In M. Broy, ed. *Programming and mathematical Method, Marktoberdorf Summer School 1990* NATO-ASI series, Springer-Verlag (1991), p. 273-323.

[18] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *Journal of the Association for Computing Machinery, Vol. 40* No. 3 (1993), p. 653-682.

[19] J.A. Bergstra, A. Ponse, and J.J. van Wamel. Process algebra with backtracking. In J.W. de Bakker, W.P. de Roever, and G. Rozenberg, editors, *Proceedings of the REX Symposium "A Decade of Concurrency: Perspectives"*, volume 803 of *Lecture Notes in Computer Science*, pages 46–91, London Berlin Heidelberg, 1994. Springer-Verlag.

[20] J.A. Bergstra, I. Bethke, and A. Ponse. Process algebra with combinators. In E. Börger, Y. Gurevich, and K. Meinke, editors, *Proceedings CSL '93, Swansea*, volume 832 of *Lecture Notes in Computer Science*, pages 36–65, London Berlin Heidelberg, 1994. Springer-Verlag.

[21] J.A. Bergstra, I. Bethke, and A. Ponse. Process algebra with iteration and nesting. *The Computer Journal*, 37(4):243–258, 1994.

[22] J.C.M. Baeten and J.A. Bergstra. On sequential composition, action prefixes and process prefix. *Formal Aspects of Computing*, 6:250–268, 1994.

[23] J.C.M. Baeten and J.A. Bergstra. Process algebra with propositional signals. *Theoretical Computer Science*, 177(2):381–405, 1997.

[24] J.C.M. Baeten, J.A. Bergstra, and S.A. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities. *Information and Computation*, 121(2):234–254, 1995.

[25] J.C.M. Baeten and J.A. Bergstra. Graph isomorphism models for non interleaving process algebra. In A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, editors, *Algebra of Communicating Processes, Utrecht 1994*, pages 299–318, London Berlin Heidelberg, 1994. Workshops in Computing, Springer-Verlag.

[26] J.C.M. Baeten and J.A. Bergstra. Process algebra with partial choice. In B. Jonsson and J. Parrow, editors, *Proceedings CONCUR 94,* Uppsala, Sweden, volume 836 of *Lecture Notes in Computer Science*, pages 401–416, London Berlin Heidelberg, 1994. Springer-Verlag.

[27] J.A. Bergstra and Gh. Ştefănescu. Processes with multiple entries and exits. In H. Reichel, editor, *FCT'95*, volume 965 of *Lecture Notes in Computer Science*, pages 136–145. Springer-Verlag, 1995.

[28] J.A. Bergstra and Gh. Stefanescu. Processes with multiple entries and exits modulo isomorphism and modulo bisimulation. *Fundamenta Informaticae*, 27(1):37–56, 1996.

[29] J.C.M. Baeten, J.A. Bergstra, and Gh. Ştefănescu. Process algebra with feedback. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra*, volume 53 of *CSLI Lecture Notes*, pages 13–38. CSLI Publications, Stanford, 1995.

[30] J.C.M. Baeten and J.A. Bergstra. Bounded bags, stacks and queues. In J. Winkowski A. Mazurkiewicz, editor, *Proceedings CONCUR 1997*, volume 1243 of *LNCS*, pages 104–118. Springer-Verlag, 1997.

[31] J.C.M. Baeten and J.A. Bergstra. Deadlock behaviour in split and ST bismulation semantics. *Electronic Notes in Theoretical Computer Science* 16 2 (1998) 14 p.

[32] Jan A. Bergstra and A. Ponse. Process algebra and conditional composition. *Information Processing Letters 80* (2001), p. 41-49.

[33] Jan A. Bergstra, Wan Fokkink and A. Ponse. Process Algebra with Recursive Operators. In J.A. Bergstra, A. Ponse and S. Smolka eds. *Handbook of Process Algebra, Elsevier Science* (2001), p. 333-389.

Process algebra with silent steps

[1] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science 37* 1 (1985), p. 77-121.

[2] J.A. Bergstra and J.W. Klop. Algebra, specification and verification in bisimulation semantics. In M. Hazewinkel, J.K. Lenstra and L.G.L.T. Meertens, eds. *Proceedings of the CWI Symposium Mathematics and Computer Science. CWI Monograph series no. 4.* North Holland (1986), p. 61-94.

[3] J.A. Bergstra and J.W. Klop. ACP - a universal axiom sytem for process specification. *CWI-Newsletter, no. 15.* Amsterdam (1987), p.3-23.

[4] J.A. Bergstra and J.W. Klop. ACP - a universal axiom sytem for process specification. In M. Wirsing and J.A. Bergstra, eds. *Algebraic Methods: Theory, Tools and Applications. Springer lecture Botes in Computer Science 394* (1989), p. 447-462.

[5] J.A. Bergstra and J.W. Klop. Verification of an alternating bit protocol by means of process algebra. In W.Bibel and K.P. Jantke, eds. *Mathematical Methods of Specification and Synthesis of Software Systems 1985. Math. Research 31.* Akademie-Verlag Berlin (1986), p.9-23.

[6] J.A. Bergstra, J.W. Klop and E.-R. Olderog. Failure semantics without chaos: a new process model for fair abstraction. In M. Wirsing, ed. Formal Description of Porgramming Concepts III. North Holland (1987), p. 70-103.

[7] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. Conditional axioms and α/β- calculus in process algebra. In M. Wirsing, ed. Formal Description of Programming Concepts III. North Holland (1987), p. 53-75.

[8] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. On the consistency of Koomen's fair abstraction rule. *Theoretical Computer Science 51* 1/2 (1987), p. 129-176.

[9] J.A. Bergstra and J.W. Klop. A complete inference sytem for regular processes with silent moves. In F.R. Drake and J.K. Truss, eds. Proceedings Logic Colloquium 1986. North Holland (1988), p. 21-81.

[10] Jan A. Bergstra and Alban Ponse. Non-regular iterators in process algebra. *Theoretical Computer Science 269* (2000), p. 203-229.

[11] Jan A. Bergstra and Alban Ponse. Register-Machine Based Processes. *Journal of the ACM 48,* 6 (2001), p. 1207-1241.

[12] Jan A. Bergstra, Alban Ponse, and Mark B. van der Zwaag. Branching time and orthogonal bisimulation equivalence. *Theoretical Computer Science 309* 1/3 (2003), p. 313-355.

Process algebra with time and space

[1] J.C.M. Baeten and J.A. Bergstra. Real-time process algebra. *Formal Aspects of Computing 3* 2 (1991), p. 472-492.

[2] J.C.M. Baeten and J.A. Bergstra. Real space process algebra. In J.C.M. Baeten and J.F, Groote, eds. *Proceedings of CONCUR'91. Springer Lecture Notes in Computer Science 527* (1991), p.96-110.

[3] J.C.M. Baeten and J.A. Bergstra. Asynchronous communication in real space process algebra. In J.Vytopil, ed. Proceedings of Formal techniques in real time and fault tolerant systems. Springer Lecture Notes in Computer Science 527 (1991), p.473-492.

[4] J.C.M. Baeten and J.A. Bergstra. The state operator in real time process algebra. In: J.W. de Bakker, W.P. de Roever and G. Rozenberg, eds. *Proceedings REX Workshop Real time: theory in practice. Springer Lecture Notes in Computer Science XXXX* (1992), p

[5] J.C.M. Baeten and J.A. Bergstra. Real space process algebra. *Formal aspects of computing 5* (1993), p.481-529.

[6] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra. *Formal aspects of computing*, 8:286–305, 1996.

[7] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra, absolute time, relative time and parametric time. *Fundamenta Informaticae*, 29:51–76, 1997.

[8] J.C.M. Baeten, J.A. Bergstra, and R.N. Bol. A real time process logic. In D.M. Gabbay and H.J. Ohlbach, editors, *Proceedings ICTL 94*, volume 827 of *LNAI*, pages 30–47, London Berlin Heidelberg, 1994. Springer-Verlag.

[9] J.C.M. Baeten, J.A. Bergstra, and M.A. Reniers. Discrete time process algebra with silent step. In G. Plotkin, C. Stirling, and M. Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*, Foundations of Computing, pages 535–569. MIT Press, 2000.

[10] J.C.M. Baeten and J.A. Bergstra. Real time process algebra with infinitesimals. In A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, editors, *Algebra of Communicating Processes, Utrecht 1994*, pages 148–187, London Berlin Heidelberg, 1994. Workshops in Computing, Springer-Verlag.

[11] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra with abstraction. In H. Reichel, editor, *FCT'95*, volume 965 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 1995

[12] J.A. Bergstra and P. Klint. The discrete time toolbus. In M. Wirsing and M. Nivat, editors, *Algebraic Methodology and Software Technology (AMAST), LNCS 1101*, pages 286–305. Springer-Verlag, 1996.

[13] J.A. Bergstra and P. Klint. The toolbus coordination architecture. In P. Ciancarini and C. Hankin, editors, *Coordination Languages and Models (COORDINATION), LNCS 1061*, pages 75–88. Springer-Verlag, 1996.

[14] J.A. Bergstra and P. Klint. The discrete time toolbus - a software coordination architecture. *Science of Computer Programming*, 31:205–229, 1998.

[15] J. A. Bergstra, C.A. Middelburg and Y.S. Usenko. Discrete Time Process Algebra and the semantics of SDL. In J.A. Bergstra, A. Ponse and S. Smolka eds. *Handbook of Process Algebra, Elsevier Science* (2001), p. 1209-1268.

[16] J. A. Bergstra and C.A. Middelburg. Located actions in process algebra with timing. *Fundmenta Informaticae*, 61(3/4):183–211, 2004.

[17] J. A. Bergstra and C.A. Middelburg. Process Algebra for Hybrid Systems. *Theoretical Computer Science 335* (2005), p. 215-280.

[18] J. A. Bergstra and C.A. Middelburg. Continuity Controlled Hybrid Automata. Tech. Univ. Eindhoven, Dept. of Mathematics and Comp. Science, CS-Report 04/11 (2004). To appear in JLAP.

Logics and languages for processes and data

[1] J.A. Bergstra and J. Terlouw. Standard model semantics for DSL, a data type specification language. *Acta Informatica 19* 1 (1983), p. 97-113.

[2] J.A. Bergstra, J. Heering and P. Klint. Module algebra. *Journal of the Association for Computing Machinery, Vol. 37* 2 (1990), p. 335-372.

[3] J.A. Bergstra, J. Heering and P. Klint. The algebraic specification formalism ASF. In J.A. Bergstra, J. Heering and P. Klint,eds. *Algebraic Specification.* Addison Wesley, ACM Press Frontier Series (1989), p. 67-132.

[4] J.A. Bergstra, J. Heering and P. Klint. A simple programming language and its implementation. In J.A. Bergstra, J. Heering and P. Klint,eds. *Algebraic Specification.* Addison Wesley, ACM Press Frontier Series (1989), p. 67-131.

[5] J.C.M. Baeten, J.A. Bergstra, S. Mauw and G. Veltink. A process specification formalism based on static COLD. In J.A. Bergstra, L.M.G. Feys, eds. *Algebraic methods II: Theory, Tools and Applications. Springer Lecture Notes in Computer Science 490* (1991), p. 341-362.

[6] J.C.Baeten and J.A. Bergstra. Design of a specification language by abstract syntax engineering. In J.A. Bergstra, L.M.G. Feys, eds. *Algebraic methods II: Theory, Tools and Applications. Springer lecture Notes in Computer Science 490* (1991), p.363-394.

[7] J.A. Bergstra and A. Ponse. Kleene's three-valued logic and process algebra. *Information Processing Letters*, 67:95–103, 1998.

[8] J.A. Bergstra and A. Ponse. Process algebra with five-valued logic. In C.S Calude and M.J. Dinneen, editors, *Proceedings of DMTCS'99 and CATS'99*, volume 21 of *Australian Computer Science Communications*, pages 128–143, Auckland, 1999. Springer-Verlag.

[9] J.A. Bergstra and A. Ponse. Bochvar-McCarthy logic and process algebra. *Notre Dame Journal of Formal Logic*, 39(4):464–484, 2000.

[10] J.A. Bergstra and A. Ponse. Process algebra with four-valued logic. *Journal of Applied Non-Classical Logics*, 10(1):27–53, 2000.

[11] J.F.A.K. van Benthem and J.A.Bergstra. Logic of transition systems. *Journal of Logic Language and Information*, 3:247–283, 1995.

[12] J.A. Bergstra, I. Bethke, and P.H. Rodenburg. A propositional logic with four values: true, false, divergent and meaningless. *Journal of Applied Non-Classical Logics*, 5:199–217, 1995.

[13] J.A. Bergstra and C.A. Middelburg. Model theory of process algebra. TU Eindhoven Technical Report 04/24, to appear in SLNCS (2004)

[14] J.A. Bergstra and C.A. Middelburg. Splitting bisimulations and retrospective conditions. TU Eindhoven Technical Report 05/03, to appear in Information and Computation (2005)

[15] J.A. Bergstra and C.A. Middelburg. Process algebra with conditionals in the presence of epsilon. TU Eindhoven Technical Report 05/15 (2005)

[16] J.A. Bergstra and C.A. Middelburg. Strong splitting bisimulation equivalence. TU Eindhoven Technical Report 05/04, to appear in Proc. CALCO 2005 (2005)

[17] J.A. Bergstra and C.A. Middelburg. Preferential choice and coordination conditions. TU Eindhoven Technical Report 05/14 (2005)

Algebras of modules, frames and networks

[1] J.A. Bergstra, J. Heering, and J.W Klop. Object-oriented algebraic specification: Proposal for a notation and 12 examples. In K.R. Apt, A. Schrijver, and N.M. Temme, editors, *From Universal Morphisms to Megabytes – a Baayen Space Odyssey*, pages 187–213. CWI, Amsterdam, December 1994.

[2] J.A. Bergstra, J. Heering and P. Klint. Module algebra. *Journal of the Association for Computing Machinery, Vol. 37* 2 (1990), p. 335-372.

[3] J.A. Bergstra, C.A. Middelburg, and Gh. Ştefănescu. Network algebra for asynchronous dataflow. *International Journal of Computer Mathematics*, 65 (1997) p. 57-88.

[4] J.A. Bergstra and A. Ponse. Grid protocol specifications. In *Prospects for Hardware Foundations*, volume 1546 of *LNCS*, pages 278–309. Springer-Verlag, 1998.

[5] J.A. Bergstra, J.A. Hillebrand, and A. Ponse. Grid protocols based on synchronous communication. *Science of Computer Programming*, 29:199–233, 1997.

[6] J.A. Bergstra and A. Ponse. Frame algebra with synchronous communication. In R.J. Wieringa and R.B. Feenstra, editors, *Information Systems - Correctness and Reusability (Selected papers from the IS-CORE Workshop)*, pages 3–15. World Scientific, 1995.

[7] J.A. Bergstra and A. Ponse. Frame-based process logics. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra*, volume 53 of *CSLI Lecture Notes*, pages 39–63. CSLI Publications, Stanford, 1995.

[8] J.A. Bergstra and Gh. Stefanescu. Network algebra with demonic relation operators. *Revue Roumaine de Mathematiques pure et Appliquees*, Tome XLIII(5-6):57–88, 1998.

[9] J.A. Bergstra and Gh. Ştefănescu. Bisimulation is two-way simulation. *Information Processing Letters*, 52:285–287, 1994.

[10] J.A. Bergstra, W.J. Fokkink, and C.A. Middelburg. Algebra of timed frames. *International Journal of Computer Mathematics*, 61:227–255, 1996.

[11] Jan A. Bergstra and Inge Bethke. Network Algebra in Java. *J. Logic and Algebraic Programming*, 62:247-271, 2005.

Program algebra and thread algebra

[1] J.A. Bergstra and M.E. Loots. Program algebra and coprogram calculus. In F.L. Bauer and R. Steinbrüggen, editors, *Foundations of Secure Computation*, volume 175 of *NATO Science Series: Computer & Systems Sciences*, pages 61–88. IOS Press, 2000.

[2] J.A. Bergstra and M.E. Loots. Program algebra for component code. *Formal Aspects of Computing*, 12:1–17, 2000.

[3] J.A. Bergstra and M.E. Loots. Program algebra for sequential code. *Journal of Logic and Algebraic Programming*, 51(2):125–156, 2002.

[4] J.A. Bergstra and A. Ponse. Combining programs and state machines. *Journal of Logic and Algebraic Programming*, 51(2):175–192, 2002.

[5] J.A. Bergstra and I. Bethke. Molecular dynamics. *Journal of Logic and Algebraic Programming*, 51(2):193–214, 2002.

[6] Jan A. Bergstra and Inge Bethke. Polarized process algebra and program equivalence. In J.C.M. Baeten, J.K. Lenstra, J. Parrow, G.J. Woeginger, eds. *Proceedings of ICALP 2003, Springer Lecture Notes in Computer Science 2719*,(2003) p. 1-21.

[7] Jan Bergstra and Pum Walters. Operator programs and operator processes. *Information and Software Technology 45*, (2003) p.681-689,

[8] J.A. Bergstra and A. Ponse. Execution architectures for program algebra. *Utrecht University LGPS 230, to appear in J. Appl. Logic*, (2004).

[9] J.A. Bergstra and I. Bethke. Linear projective program syntax. Logic Group Preprint Series No. 233, October 2004, Department of Philosophy, Utrecht University, 2004.

[10] J.A. Bergstra and A. Ponse. A bypass of Cohen's impossibility result . Eds. P.M.A. Sloot et. al. *Advances in Grid computing*, LNCS 3470:1097-1106, (2005).

[11] J.A. Bergstra, I. Bethke and A. Ponse. Decidability problems for push-down threads. UvA, IvI El. report PRG0502 , (2005).

[12] J.A. Bergstra and C.A. Middelburg. Thread algebra for strategic interleaving. TU Eindhoven Technical Report CS-05-14, (2005).

[13] J.A. Bergstra and C.A. Middelburg. A thread algebra with multi-level strategic interleaving. In S. Barry Cooper, Benedikt Löwe, Leen Torenvliet, eds. *New Computational Paradigms: First Conference on Computability in Europe, CiE 2005, Amsterdam, The Netherlands, Springer Lecture Notes in Computer Science 3526*, (2005) p. 35-48.

[14] J.A. Bergstra and C.A. Middelburg. Maurer computers with single thread control. TU Eindhoven Technical Report CS-05-17, (2005).

Miscellaneous

[1] J.A. Bergstra. Discourse between processes. *Notre Dame J. of Formal Logic, Vol. 21* 1 (1980), p.131-134.

[2] J.A. Bergstra. Decision problems concerning parallel programming. In *Proceedings of MFCS78. Springer Lecture Notes in Computer Science 64* (1978), p.134-142.

[3] J.A. Bergstra and Th.P. van der Weide. Process semantics of algebraic data types. In *Proceedings of Colloquia Mathematicae Societaties Janos Bolyai, 26, Mathematical Logic in Computer Science 1978*. North Holland (1981), p. 167-180.

[4] J.A. Bergstra, A. Ollengren and Th.P. van der Weide. Process semantics of algebraic data types. In Proceedings of FCT77, Springer Notes in Computer Science 56 (1978), p. 33-38.

[5] J.A. Bergstra, H.J.M. Goeman, A. Ollengren, G.A. Terpstra and Th.P. van der Weide. Axioms for multilevel objects. *Fundamenta Informaticae, Vol. III* 2 (1980), p. 171-180.

[6] J.A. Bergstra and J. Tiuryn. Implicit definability of algebraic structurs by means of program properties. In L.Budach, ed. Proceedings of FCT79. Akademie-Verlag, Berlin (1979), p. 59-63.

[7] J.A. Bergstra and J. Tiuryn. Implicit definability of algebraic structures by means of program properties. *Fundamenta Informaticae, Vol. IV* 3(1981), p. 661-676.

[8] J.A. Bergstra and J. Tiuryn. Regular extensions of iterative algebras and metric interpretations. *Fundamenta Informaticae, Vol. IV* 4 (1981), p. 997-1014.

[9] J.A. Bergstra and J. Tiuryn. Algorithmic degrees of algebraic structures. *Fundamenta Informaticae, Vol. IV* (1981), p. 851-862.

[10] J.A. Bergstra and J.-J.Ch. Meyer. On the quantifier free fragment of Logic of Effective Definitions. *Fundamenta Informaticae Vol IV* 4 (1981), p. 935-956.

[11] J.A. Bergstra and J.-J. Ch. Meyer. On the elimination of iteration quantifiers in a fragment of algorithmic logic. *Theoretical Computer science 21* (1982), p. 269-279.

[12] J.A. Bergstra. Disk allocation for cartesian product files, a counterexample. *EATCS Bulletin 20* June (1983).

[13] J.A. Bergstra and J.W. Klop. A proof rule for restoring logic circuits. *Integration, the VSLI Journal 1* (1983), p.161-178.

[14] J.W. de Bakker, J.A. Bergstra, J.-J. Ch. Meyer and J.W. Klop. Linear time and branching time semantics for recursion with merge. *Theoretical Computer Science 34* (1984), p. 135-156.

[15] J.A. Bergstra and J. Smeets. Formele ontwikkeling van schermverloopschema's en interactieve programma's. *Informatie 26* 4 (1984), p.318-324.

[16] J.A. Bergstra, H.B.M. Jonkers and J.H. Obbink. A software development model for method engineering. Proc. ESPRIT Technical Week (1984)

[17] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. An operational semantics for process algebra. *CWI report CS-R8522, 1985*. In *Proceedings Banach semester Warsaw* (1985). North Holland (198?)

[18] J.A. Bergstra. Programmatuuronderzoek in de schaduw, oratie(11 november 1985, Amsterdam). *Informatie 27* 12 (1985), P.1054-1060.

[19] J.A. Bergstra. Universitaire beroepsopleidingen in de informatica. *Hoger onderwijs, Jaargang 4* 4 (1986), p.133-140.

[20] J.A. Bergstra and J.W. Klop. Algebra of communicating processes. In J.W. de Bakker, M. Hazewinkel, J.K. Lenstra, eds. *Proceedings of the CWI Symposium Mathematics and Computer Science. CWI Monograph series no. 3*. North Holland (1986), p. 89-138.

[21] J.A. Bergstra. *Terminologie van algebraische specificaties*. Kluwer Programmatuurkunde, Deventer (1987), 122 blz.

[22] J.C.M. Baeten and J.A. Bergstra. Processen en procesexpressies. *Informatie 30* 3(1988), p.214-222.

[23] J.A. Bergstra and J.W. Klop. An introduction to process algebra. In J.C.M. Baeten,ed. *Applications of Process Algebra. Cambridge Tracts in Theoretical Computer Science 17*. Cambridge University press (1990), p.1-23.

[24] J.A. Bergstra. Kerninformatica en toekomst. *Informatie 31*(1989), p.470-476.

BRdL89 J.A. Bergstra en G.R. Renardel de Lavalette. De plaats van formele specificaties in de software technologie. *Informatie 31* (1989), p. 480-494.

[25] J.A. Bergstra en G.R. Renardel de Lavalette. Onderzoek en ontwikkeling terzake formele specificatietalen. *Informatie 32* (1990), p.1-104.

[26] J.A. Bergstra, W.J. Fokkink, W.M.T. Mennen, and S.F.M. van Vlijmen. *Spoorweglogica via EURIS*, volume XII of *Quaestiones Infinitae*. Zeno-institute for Philosophy, Utrecht-Leiden, 1997.

[27] J.A. Bergstra and S.F.M. van Vlijmen. *Theoretische software-engineering*, volume XXVIII of *Quaestiones Infinitae*. Zeno, The Leiden-Utrecht Research Institute of Philosophy, 1998.

[28] J.A. Bergstra and P. Klint. Informatica moet eigen weg kiezen. *Automatisering Gids*, 3 april 1998.

[29] J.A. Bergstra and P. Klint. Theorie en logica eruit, het bedrijfsleven erin. *Automatisering Gids*, 20 februari 1998.

[30] Jan Bergstra. Provably correct programming and beyond. *NWO Huygenslezing by C.A.R. Hoare, commentary part*, NWO, The Hague (1998) p. 38-50.

[31] J.A. Bergstra. Molecule-oriented programming in Java. *Information and Software Technology*, 44:617–638, 2002.

[32] J.A. Bergstra and P. Klint. About "trivial" software patents, the IsNot case. In Proc HOSC 2005; www.homepages.cwi.nl/ paulk/patents, 2005

Technical Reports (no publication forthcoming)

[1] H.P. Barendregt, J.A. Bergstra, J.W. Klop and H. Volken. *Degrees, reductions and representability in the Lambda-Calculus*. University of Utrecht, Department of Mathematics, Preprint 22 (1976).

[2] J.A. Bergstra. Dynamic recursion theory. University of Leiden, *TWI. Report* 79-19 (1979)

[3] J.A. Bergstra and J.-J. Ch. Meyer. *On axiomatizing finite data structures*. Leiden TWI report 80-18 (1980)

[4] J.A. Bergstra and J. Terlouw. Four equivalent proof systems for recursive procedures. *University of Utrecht, Department of Mathematics, preprint no. 262* (1982).

[5] J.A. Bergstra and J. Terlouw. A completeness result for algebraic Hoare logics. *MC IW 237/83* (1983)

[6] J.A. Bergstra, B. Biebow, M.C. Gaudel, J. Hagelstein, H.B.M. Jonkers, J.W. Klop, J.L. Lacour, J.H. Obbink, P. Walton and P. Wodon. FAST: Requirements on a software development model. *Intermediate report for Esprit pilot project FAST* February (1984).

[7] J.A. Bergstra, B. Biebow, M.C. Gaudel, J. Hagelstein, H.B.M. Jonkers, J.W. Klop, J.L. Lacour, J.H. Obbink, P. Walton and P. Wodon. FAST: towards METEOR. *Final report on the Esprit pilot project FAST* October (1984).

[8] J.A. Bergstra. Put and get, primitives for synchronous unreliable message passing. *CIF report LGPS 3*. RUU (1985)

[9] J.A. Bergstra and J.W. Klop. A convergence theorem in process algebra. *CWI Report CS-R8733 (revised version of MC report IW206)* (1982)

[10] J.A. Bergstra. Module algebra for relational specifications. *CIF rapport LGPS 16, RUU* (1986).

[11] J.A. Bergstra. A representation of addition and deletion lists using module algebra. *University of Amsterdam, Programming Research Group Report, P8903* (1989)

[12] J.A. Bergstra and J.W. Klop. BMACP. In J.W. Klop, J.-J. Ch. Meyer and J.J.M.M. Rutten, eds. *J.W. de Bakker, 25 jaar semantiek*. Liber amicorum, CWI. Also: Programming Research Group Report, P8905 (1989).

[13] J.A. Bergstra and J.V. Tucker. The inescapable stack, an exercise in algebraic specification with total functions. *University of Amsterdam, Programming Research Group Report P8804* (1988, revised version P8804b, 1990)

[14] J.A. Bergstra. Process algebra for synchronous communication and observation. *University of Amsterdam, Programming research Group report P8815* (1988)/P8815b. Second revised version: J.C.M. Baeten and J.A. Bergstra: Process algebra with signals and conditions: Report P9008.

[15] J.A. Bergstra and C.A. Middelburg. Process algebra semantics of φSDL. In A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, editors, *De proceedings: ACP'95*, pages 309–346, 1995.

[16] J.C.M. Baeten and J.A. Bergstra. Process algebra with propositional signals. In A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, editors, *De proceedings: ACP'95*, pages 213–228, 1995.

[17] J.A. Bergstra. Algebra of states and transitions. *University of Amsterdam Programming Research Group Report, P8909* (1989).

[18] J.A. Bergstra and A. Ponse. Process Algebra Primitives for File Transfer. Technical Report P9725, University of Amsterdam, Programming Research Group, 1997. Available by anonymous ftp from ftp.wins.uva.nl, file pub/programming-research/reports/1997/P9725.ps.Z.

[19] J.A. Bergstra and M.P.A. Sellink. Sequential data algebra primitives. Technical Report P9602, University of Amsterdam, 1996.

[20] J.A. Bergstra and M.P.A. Sellink. Sequential data algebra primitives (revised version of P9602). Technical Report P9602b, University of Amsterdam, 1996.

[21] J.A. Bergstra and M.P.A. Sellink. An Arithmetical Module for Rationals and Reals. Technical Report P9709, University of Amsterdam, Programming Research Group, 1997. Available by anonymous ftp from ftp.wins.uva.nl, file pub/programming-research/reports/1997/P9709.ps.Z.

[22] J.A. Bergstra and M.E. Loots. Software mechanics for java multi-threading. Technical Report P9906, University of Amsterdam, Programming Research Group, 1999. Available by anonymous ftp from ftp.wins.uva.nl, file pub/programming-research/reports/1999/P9906.ps.gz.

[23] J.A. Bergstra and M.J.G. van den Brand. Syntax and semantics of a high-level intermediate representation for ASF+SDF. Report SEN-R0030 November 2000.

[24] J.C.M. Baeten and J.A. Bergstra. Mode transfer in process algebra. Technical Report CSR 00-01, TUE, Eindhoven, 2000.

[25] J.A. Bergstra, I. Bethke, and M.E. Loots. A first course on C program reading. Technical Report P0002, UvA, Amsterdam, 2000.