

List of publications by Jan A. Bergstra (October 18, 2007)

Selected list of publications arranged by subject and per subject by date. Survey papers have been classified under miscellaneous; books and edited volumes have been listed first. (Complete information including unpublished works and submitted papers can be found on www.science.uva.nl/~janb. A different, independent and fairly complete source for journal papers is DBLP)

Books and edited volumes

- [1] J.A. Bergstra, J.W. Klop and A. Middeldorp. *Termherschrijfsystemen*. Kluwer Programmatuurkunde, Deventer, (1989), 170 pages. (In Dutch)
- [2] J.A. Bergstra, J. Heering and P. Klint (eds.) *Algebraic Specification*. Addison Wesley, ACM Press Frontier Series (1989).
- [3] 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.
- [4] 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.
- [5] J.A. Bergstra, A. Ponse and S. Smolka (eds.) *Handbook of Process Algebra*, Elsevier Science (2001).
- [6] Karl de Leeuw and Jan Bergstra (eds.) *The History of Information Security: A Comprehensive Handbook*, Elsevier, (2007)

Recursion theory

- [1] J.A. Bergstra. *Computability and continuity in finite types* Ph.D. Thesis Utrecht (January 1976).
- [2] 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.
- [3] J.A. Bergstra. The continuous functionals and 2E . In J.E. Fenstad, R.O. Gandy and G.E. Sacks eds. *Generalized Recursion Theory II*. North Holland (1978), p. 39-53.

Lambda calculus and rewrite systems

- [1] H.P. Barendregt, J.A. Bergstra, J.W. Klop and H. Volken. Representability in lambda algebras. *Indagationes 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.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.
- [9] 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.

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, J. Tiuryn and J.V. Tucker. Floyd's principle, correctness theories and program equivalence. *Theoretical Computer Science* 17 (1981), p. 112-149.
- [3] J.A. Bergstra and J.V. Tucker. Two theorems on the completeness of Hoare's logic. *Information Processing Letters* 15 (1982), p.143-149.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] J.A. Bergstra and J.V. Tucker. Hoare's logic and Peano's arithmetic. *Theoretical Computer Science* 22 (1983), p. 265-284.
- [8] 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.
- [9] J.A. Bergstra and J.W. Klop. Proving program inclusion using Hoare's logic. *Theoretical Computer Science* 30 (1984), p.1-48.
- [10] 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, 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 lecture Notes in Computer Science* 118 (1981), p. 193-204.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] J.A. Bergstra and J.V. Tucker. Algebraic specifications of computable and semi-computable data structures. *Theoretical Computer Science* 50(1987), p. 137-181.
- [11] J.A. Bergstra and J. Heering. Which data types have ω -complete initial algebra specifications? *Theoretical Computer Science*, 124 (1994) p. 149-168.
- [12] J.A. Bergstra and J.Heering. Homomorphism preserving algebraic specifications require hidden sorts. *Information and Computation*, 119(1) (1995) p. 119-123.

Specific algebras as data types

- [1] J.A. Bergstra and J.-J. Ch. Meyer. On specifying sets of integers. *Elektronische Informationsverarbeitung und Kybernetik* 20 10/11 (1984), p. 531-541.
- [2] 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.
- [3] J.A. Bergstra and J.V. Tucker. The data type variety of stack algebras. *Annals of Pure and Applied Logic*, 73 (1995) p.11–36.
- [4] J.A. Bergstra and J.V.Tucker. The rational numbers as an abstract datatype. *J. ACM*, vol. 54, Issue 2, Article no. 7 (April 2007) 25 pages.
- [5] J.A. Bergstra and J.V Tucker. Division Safe Calculation in Totalised Fields. *Theory of Computing Systems*, (15 pages) DOI 10.1007/s00224-007-9035-4
- [6] J.A. Bergstra. Elementary algebraic specifications of the rational function field. In A. Beckmann, U. Berger, B. Loewe, and J.V. Tucker (Eds.), *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006*, Swansea, UK, June 30-July 5, 2006, Proceedings (pp. 40-54). Springer-Verlag.
- [7] J.A. Bergstra and J.V.Tucker. An elementary algebraic specification of the rational complex numbers. In K. Futatsugi, J-P. Jouannaud, and J. Meseguer (Eds.), *Algebra, Meaning, and Computations*. Springer-Verlag (2006) p. 459-475.

Concrete process algebra

- [1] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Control, Vol. 60* 1/3 (1984), p. 109-137.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] J.A. Bergstra and J. Tiuryn. Process algebra semantics for queues. *Fundamenta Informaticae X* (1987), p. 213-224.
- [8] 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.
- [9] 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.
- [10] J.C.M. Baeten and J.A. Bergstra. Global renaming operators in concrete process algebra. *Information and Computation* 78 3 (1988), p. 205-245.
- [11] 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.
- [12] 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.

- [13] 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.
- [14] 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.
- [15] 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 *Springer Lecture Notes in Computer Science* (1994) p. 46–91.
- [16] J.A. Bergstra, I. Bethke, and A. Ponse. Process algebra with iteration and nesting. *The Computer Journal*, 37(4) (1994) p. 243–258.
- [17] J.C.M. Baeten and J.A. Bergstra. On sequential composition, action prefixes and process prefix. *Formal Aspects of Computing*, 6 (1994) p. 250–268.
- [18] J.C.M. Baeten and J.A. Bergstra. Process algebra with propositional signals. *Theoretical Computer Science*, 177(2) (1997) p. 381–405.
- [19] J.C.M. Baeten, J.A. Bergstra, and S.A. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities. *Information and Computation*, 121(2) (1995) p. 234–254.
- [20] 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, Springer Workshops in Computing*, (1994) p. 299–318, 1994.
- [21] 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*, CSLI Publications, Stanford (1995) p. 13–38.
- [22] 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.
- [23] Jan A. Bergstra and A. Ponse. Process algebra and conditional composition. *Information Processing Letters* 80 (2001), p. 41-49.
- [24] 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*, Springer-Verlag, (1999) p. 128–143.
- [25] J.A. Bergstra and A. Ponse. Bochvar-McCarthy logic and process algebra. *Notre Dame Journal of Formal Logic*, 39(4) (2000) p. 464–48.
- [26] J.A. Bergstra and C.A. Middelburg. Model theory of process algebra. In A. Middeldorp et. al. eds. *Processes, terms and cycles: steps on the road to infinity: Essays dedicated to Jan-Willem Klop* Springer (2005) p. 445-495.
- [27] J.A. Bergstra and C.A. Middelburg. Splitting bisimulations and retrospective conditions. *Information and Computation* 204 (7) (2007) p. 1083-1138.
- [28] J.A. Bergstra and C.A. Middelburg. Preferential choice and coordination conditions. *J. of Logic and Algebraic Programming* 70 (2007) p. 172-200.

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. Process 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. 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.
- [4] 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 Programming Concepts III*. North Holland (1987), p. 70-103.
- [5] 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.
- [6] 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.
- [7] J.A. Bergstra and J.W. Klop. A complete inference system 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.
- [8] Jan A. Bergstra and Alban Ponse. Non-regular iterators in process algebra. *Theoretical Computer Science 269* (2000), p. 203-229.
- [9] Jan A. Bergstra and Alban Ponse. Register-Machine Based Processes. *Journal of the ACM 48, 6* (2001), p. 1207-1241.
- [10] 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. *Formal aspects of computing 5* (1993), p.481-529.
- [3] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra. *Formal aspects of computing 8*, (1996), p. 286-305.
- [4] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra: absolute time, relative time and parametric time. *Fundamenta Informaticae*, 29 (1997) p. 51-76.
- [5] 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, MIT Press (2000) p. 535-569.
- [6] J.A. Bergstra and P. Klint. The discrete time toolbus - a software coordination architecture. *Science of Computer Programming*, 31 (1998) p. 205-229.
- [7] 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.
- [8] J. A. Bergstra and C.A. Middelburg. Located actions in process algebra with timing. *Fundamenta Informaticae*, 61(3/4) (2004) p. 183-211.
- [9] J. A. Bergstra and C.A. Middelburg. Process Algebra for Hybrid Systems. *Theoretical Computer Science 335* (2005), p. 215-280.
- [10] J. A. Bergstra and C.A. Middelburg. Continuity Controlled Hybrid Automata. *J. Logic and Algebraic Programming*, 68 (1-2), (2006), p. 5-53.

Algebras of modules, frames and networks

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

- [2] 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.
- [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, J.A. Hillebrand, and A. Ponse. Grid protocols based on synchronous communication. *Science of Computer Programming*, 29: (1997) p. 99–233.
- [5] 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)*, World Scientific (1995) p. 3–15.
- [6] 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*, CSLI Publications, Stanford (1995) p. 39–63.
- [7] J.A. Bergstra, W.J. Fokkink, and C.A. Middelburg. Algebra of timed frames. *International Journal of Computer Mathematics*, 61 (1996) p. 227–255.
- [8] Jan A. Bergstra and Inge Bethke. Network Algebra in Java. *J. Logic and Algebraic Programming*, 62 (2005) p. 247-271.

Program algebra and thread algebra

- [1] J.A. Bergstra and M.E. Loots. Program algebra for sequential code. *Journal of Logic and Algebraic Programming*, 51(2) (2002) p. 125–156.
- [2] J.A. Bergstra and I. Bethke. Molecular dynamics. *Journal of Logic and Algebraic Programming*, 51(2) (2002) p. 193–214.
- [3] 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.
- [4] Jan Bergstra and Pum Walters. Operator programs and operator processes. *Information and Software Technology* 45, (2003) p.681-689,
- [5] J.A. Bergstra and A. Ponse. Execution architectures for program algebra. *J. Appl. Logic* 5, (2007) p. 170-192.
- [6] Jan A. Bergstra and Inge Bethke. Polarized process algebra with reactive composition. *Theor. Comput. Sci.* 343 (3) (2005) p. 285-304.
- [7] 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 (2005) p. 1097-1106, .
- [8] J.A. Bergstra, I. Bethke and A. Ponse. Decision problems for push-down threads. *Acta Informatica* 44 (2007) p. 75-90.
- [9] J.A. Bergstra and C.A. Middelburg. Thread algebra for strategic interleaving. *Formal Aspects of Computing*, to appear.
- [10] J.A. Bergstra and C.A. Middelburg. A thread algebra with multi-level strategies. *Fundamenta Informaticae*, 71 (2/3), (2006) p. 153-182.

Miscellaneous

- [1] J.A. Bergstra and J.W. Klop. A proof rule for restoring logic circuits. *Integration, the VSLI Journal* 1 (1983), p.161-178.
- [2] 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.

- [3] 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.
- [4] J.A. Bergstra. Programmatuuronderzoek in de schaduw, oratie (11 november 1985, Amsterdam). *Informatie 27* 12 (1985), P.1054-1060.
- [5] J.A. Bergstra. Universitaire beroepsopleidingen in de informatica. *Hoger onderwijs, Jaargang 4* 4 (1986), p.133-140.
- [6] J.F.A.K. van Benthem and J.A.Bergstra. Logic of transition systems. *Journal of Logic Language and Information*, 3 (1995) p. 247–283.
- [7] J.A. Bergstra and P. Klint. About "trivial" software patents, the IsNot case. *SCP 64* (2007) p. 264-285.