

Bibliography

- [Aba89] Martin Abadi. “The power of temporal proofs”. In: *Theoretical Computer Science* 65.1 (1989), pp. 35–83.
- [Aba90] Martin Abadi. “An axiomatization of Lamport’s temporal logic of actions”. In: *CONCUR ’90 Theories of Concurrency. Unification and Extension*. Ed. by J. C. M. Baeten and J. W. Klop. Vol. 458. Lecture Notes in Computer Science. Springer-Verlag, 1990, pp. 57–69.
- [Abr10] J.-R. Abrial. *Modeling in Event-B: System and Software Engineering*. New York, NY, USA: Cambridge University Press, 2010.
- [Abr96] J.-R. Abrial. *The B-Book: Assigning Programs to Meanings*. Cambridge University Press, 1996.
- [ACM20] ACM, ed. *Proceedings of the ACM on Programming Languages*. Vol. 4. HOPL. Association for Computing Machinery, 2020. DOI: 10.1145/3406494.
- [Acz82] P. H. G. Aczel. “A Note on Program Verification”. (private communication) Manuscript, Manchester. Jan. 1982.
- [Acz83] P. H. G. Aczel. “On An Inference Rule for Parallel Composition”. (private communication) Manuscript, Manchester. 1983.
- [Acz88] P. Aczel. *Non-Well-Founded Sets*. CSLI Lecture Notes 14. Stanford: Center for the Study of Language and Information, 1988.
- [AD22] Janet Abbate and Stephanie Dick, eds. *Abstractions and Embodiments: New Histories of Computing and Society*. Johns Hopkins University Press, 2022.
- [ADB88] Pierre America and Jaco De Bakker. “Designing equivalent semantic models for process creation”. In: *Theoretical Computer Science* 60.2 (1988), pp. 109–176.
- [AFd80] K. R. Apt, N. Francez, and W.-P. de Roever. “A Proof System for Communicating Sequential Processes”. In: *ACM Transactions on Programming Languages and Systems* 2.3 (July 1980), pp. 359–385. DOI: <https://doi.org/10.1145/357103.357110>.

- [AFDR80] Krzysztof R Apt, Nissim Francez, and Willem P De Roever. “A proof system for communicating sequential processes”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 2.3 (1980), pp. 359–385.
- [Agh85] Gul A Agha. *Actors: A model of concurrent computation in distributed systems*. Tech. rep. DTIC Document, 1985.
- [Agh+97] Gul A Agha, Ian A Mason, Scott F Smith, and Carolyn L Talcott. “A foundation for actor computation”. In: *Journal of functional programming* 7.1 (1997), pp. 1–72.
- [AH22] Krzysztof R. Apt and Tony Hoare, eds. *Edsger Wybe Dijkstra: his Life, Work and Legacy*. ACM, 2022.
- [AJ18] Troy K. Astarte and Cliff B. Jones. “Formal Semantics of ALGOL 60: Four Descriptions in their Historical Context”. In: *Reflections on Programming Systems - Historical and Philosophical Aspects*. Ed. by Liesbeth De Mol and Giuseppe Primiero. Springer Philosophical Studies Series, 2018, pp. 83–152. DOI: 10.1007/978-3-319-97226-8_4.
- [AJS05] Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders, eds. *Communicating Sequential Processes: The First 25 years*. Vol. 3525. Lecture Notes in Computer Science. Springer-Verlag, 2005. ISBN: 978-3-540-25813-1. DOI: 10.1007/b136154. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-25813-1>.
- [AL93] Martín Abadi and Leslie Lamport. “Composing specifications”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 15.1 (1993), pp. 73–132.
- [AL95a] Martín Abadi and Leslie Lamport. “Conjoining specifications”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 17.3 (1995), pp. 507–535.
- [AL95b] Martín Abadi and Leslie Lamport. “Conjoining specifications”. In: *ACM transactions on Programming Languages and Systems* 17.3 (1995), pp. 507–535.
- [Alb22] Gerard Alberts. “Philips and the fate of Electrologica: From single computer construction to manufacturing”. In: *Tales of Electrologica : Computers, Software and People*. Ed. by Alberts Gerard and Groote Jan Friso. History of Computing. Springer, 2022. ISBN: 9783031130328. URL: <https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,ip,shib,uid&db=nlebk&AN=3517627&site=ehost-live&scope=site&authtype=shib&custid=s8000044>.
- [Alp86] Bowen Alpern. “Proving temporal properties of concurrent programs: a non-temporal approach”. PhD thesis. Cornell University, 1986.
- [AM70] E. A. Ashcroft and Z. Manna. *Formalization of Properties of Parallel Programs*. Tech. rep. AIM-110. Published as [AM71b].

- Stanford Artificial Intelligence Project, Feb. 1970. URL: <https://apps.dtic.mil/sti/citations/AD0708740>.
- [AM71a] E. Ashcroft and Z. Manna. *The translation of 'go to' programs to 'while' programs*. Memo AIM-138 CS-71-188. Computer Science Dept., Stanford University, Jan. 1971.
- [AM71b] E. A. Ashcroft and Z. Manna. "Formalization of Properties of Parallel Programs". In: *Machine Intelligence, 6*. Ed. by B. Meltzer and D. Michie. Vol. 6. Edinburgh University Press, 1971, pp. 17–41.
- [AM79] E. Ashcroft and Z. Manna. "The translation of 'go to' programs to 'while' programs". In: *Classics in software engineering*. Yourdon Press, 1979, pp. 49–61.
- [Ame+86] P. America, J.W. de Bakker, J.N. Kok, and J.M. Rutten. "Operational semantics of a parallel object-oriented language". In: *Proceedings of the 13th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. 1986, pp. 194–208.
- [Ame89] P. America. "Issues in the Design of a Parallel Object-Oriented Language". In: *Formal Aspects of Computing 1.4* (1989), pp. 366–411.
- [Ame+89] Pierre America, Jaco de Bakker, Joost N Kok, and Jan Rutten. "Denotational semantics of a parallel object-oriented language". In: *Information and Computation 83.2* (1989), pp. 152–205. DOI: [https://doi.org/10.1016/0890-5401\(89\)90057-6](https://doi.org/10.1016/0890-5401(89)90057-6).
- [AO19] Krzysztof R. Apt and Ernst-Rüdiger Olderog. "Fifty years of Hoare's logic". In: *Formal Aspects of Computing 31.6* (2019), pp. 751–807.
- [AO91] Krzysztof R. Apt and Ernst-Rüdiger Olderog. *Verification of Sequential and Concurrent Programs*. Berlin, Heidelberg: Springer-Verlag, 1991.
- [Apt02] Krzysztof R. Apt. "Edsger Wybe Dijkstra (1930–2002): A Portrait of a Genius". In: *Formal Aspects of Computing 14* (2002), pp. 92–98.
- [AS79] J.-R. Abrial and S. A. Schuman. "Non-Deterministic System Specification". In: *Semantics of Concurrent Computation*. Ed. by G. Kahn. Vol. 70. Lecture Notes in Computer Science. Berlin: Springer-Verlag, 1979, pp. 34–50.
- [AS89] Bowen Alpern and Fred B Schneider. "Verifying temporal properties without temporal logic". In: *ACM Transactions on Programming Languages and Systems (TOPLAS) 11.1* (1989), pp. 147–167.
- [Ash70] Edward Anthony Ashcroft. "Mathematical logic applied to the semantics of computer programs". PhD thesis. University of London, 1970.

- [Ash75] Edward A Ashcroft. “Proving assertions about parallel programs”. In: *Journal of Computer and System Sciences* 10.1 (1975), pp. 110–135.
- [Ash+95] Edward A Ashcroft, Anthony A Faustini, William W Wadge, and R Jagannathan. *Multidimensional programming*. Oxford University Press on Demand, 1995.
- [Ast19] Troy K. Astarte. “Formalising Meaning: a History of Programming Language Semantics”. PhD thesis. Newcastle University, June 2019.
- [Ast20] Troy K. Astarte, ed. *HFM 2019 - History of Formal Methods Workshop*. In: *Formal Methods. FM 2019 International Workshops. Porto, Portugal, October 7–11, 2019, Revised Selected Papers, Part II*. Lecture Notes in Computer Science 12233. Springer-Verlag, 2020. DOI: 10.1007/978-3-030-54997-8.
- [Ast22] Troy Kaighin Astarte. ““Difficult things are difficult to describe”: The role of formal semantics in European computer science, 1960–1980”. In: *Abstractions and Embodiments: New Histories of Computing and Society*. Ed. by Janet Abbate and Stephanie Dick. Johns Hopkins University Press, Aug. 2022.
- [Ast23a] Troy K. Astarte. “From Monitors to Monitors: an Early History of Concurrency Primitives”. In: *Minds and Machines* (2023). URL: <https://doi.org/10.1007/s11023-023-09632-2>.
- [Ast23b] Troy Kaighin Astarte. *There will be a time-fight tomorrow: old problems, new logics*. History of Computing Collection Report. To appear. Swansea University, 2023.
- [Astbc] Troy Kaighin Astarte. “Tense & Temporality: Computers and the Logic of Time”. In: *tbc*. Ed. by Arianna Borrelli and Helena Durnova. remember to change! meson press, tbc.
- [AW76] Edward A Ashcroft and William W. Wadge. “Lucid—A formal system for writing and proving programs”. In: *SIAM Journal on Computing* 5.3 (1976), pp. 336–354.
- [BA10] Richard Bornat and Hasan Amjad. “Inter-process buffers in separation logic with rely-guarantee”. In: *Formal Aspects of Computing* 22.6 (2010), pp. 735–772. DOI: 10.1007/s00165-009-0141-8.
- [BA13] Richard Bornat and Hasan Amjad. “Explanation of two non-blocking shared-variable communication algorithms”. In: *Formal Aspects of Computing* 25.6 (2013), pp. 893–931. DOI: <http://dx.doi.org/10.1007/s00165-011-0213-4>.
- [BA90] Mordechai Ben-Ari. *Principles of Concurrent and Distributed Programming*. Prentice Hall International Series in Computer Science. Prentice Hall, 1990.
- [Bae05] Jos CM Baeten. “A brief history of process algebra”. In: *Theoretical Computer Science* 335.2-3 (2005), pp. 131–146.

- [Bae19] Jos CM Baeten. *Interview with Jos Baeten*. Conducted by Gracia Redder and You Hu. Jan. 2019.
- [Bae90] J. C. M. Baeten, ed. *Applications of Process Algebra*. Cambridge University Press, 1990.
- [Bae92] J. C. M. Baeten. *Informatica als wetenschap : formele specificatie en wiskundige verificatie [Computer science as a science: formal specification and mathematical verification]*. Intreerede [Inaugural address]. Technische Universiteit Eindhoven, Oct. 1992.
- [BAPM81] Mordechai Ben-Ari, Amir Pnueli, and Zohar Manna. “The Temporal Logic of Branching Time”. In: *Proceedings of the 8th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’81. New York, NY, USA: ACM, Jan. 1981, pp. 164–176. DOI: <https://doi.org/10.1145/567532.567551>.
- [BAPM83] Mordechai Ben-Ari, Amir Pnueli, and Zohar Manna. “The temporal logic of branching time”. In: *Acta informatica* 20.3 (1983), pp. 207–226.
- [BB06] Jos CM Baeten and Mario Bravetti. “A generic process algebra”. In: *Electronic Notes in Theoretical Computer Science* 162 (2006), pp. 65–71.
- [BBP16] Inge Bethke, Bert Bredeweg, and Alban Ponse, eds. *Liber Amicorum for Jan A. Bergstra*. Universiteit van Amsterdam Informatics Institute, Dec. 2016.
- [BCJ84] H. Barringer, J.H. Cheng, and C. B. Jones. “A Logic Covering Undefinedness in Program Proofs”. In: *Acta Informatica* 21.3 (1984), pp. 251–269. DOI: 10.1007/BF00264250. URL: <http://www.springerlink.com/content/x267t82951633456/>.
- [BDK01] Eike Best, Raymond Devillers, and Maciej Koutny. *Petri net algebra*. Springer Science & Business Media, 2001.
- [Bek71] H. Bekič. *Towards a Mathematical Theory of Processes*. Tech. rep. TR 25.125. IBM Lab. Vienna, Dec. 1971.
- [Bek+74] Hans Bekič, Dines Bjørner, Wolfgang Henhagl, Cliff B. Jones, and Peter Lucas. *A Formal Definition of a PL/I Subset*. Tech. rep. 25.139. IBM Laboratory Vienna, Dec. 1974. URL: <http://homepages.cs.ncl.ac.uk/cliff.jones/ftp-stuff/TR25139/>.
- [Ben82] M. Ben-Ari. *Principles of Concurrent Programming*. Englewood Cliffs, N. J.: Prentice Hall, 1982.
- [Ber19] Jan A. Bergstra. *Interview with Jan Bergstra*. Conducted by Gracia Redder. Jan. 2019.
- [BG96] Thomas J. Bergin and Richard G. Gibson, eds. *History of Programming Languages—II*. New York, NY, USA: ACM Press, 1996. ISBN: 0-201-89502-1.
- [BH72] Per Brinch Hansen. “Structured multiprogramming”. In: *Communications of the ACM* 15.7 (1972), pp. 574–578.

- [BH73] P. Brinch Hansen. *Operating System Principles*. Prentice Hall Series in Automatic Computation 4. Prentice Hall, 1973.
- [BH76a] P. Brinch Hansen. “The Programming Language Concurrent Pascal”. In: *Language Hierarchies and Interfaces*. Ed. by F. L. Bauer and K. Samelson. Vol. 46. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 1976, pp. 82–110.
- [BH76b] P. Brinch Hansen. “The Solo Operating System: A Concurrent Pascal Program”. In: *Software — Practice and Experience* 6.2 (1976), pp. 141–149.
- [BH78] P. Brinch Hansen. “Distributed processes: A concurrent programming concept”. In: *Communications of the ACM* 21.11 (Nov. 1978), pp. 934–941.
- [BH96] Per Brinch Hansen. “Monitors and Concurrent Pascal: a personal history”. In: *History of programming languages—II*. Ed. by Thomas J. Bergin and Richard G. Gibson. New York, NY, USA: ACM Press, 1996, pp. 121–172. ISBN: 0-201-89502-1.
- [BHG87] P. A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.
- [BHJ20] Alan Burns, Ian J. Hayes, and Cliff B. Jones. “Deriving specifications of control programs for cyber physical systems”. In: *The Computer Journal* 63.5 (2020), pp. 774–790. DOI: 10.1093/comjnl/bxz019.
- [BHR84] Stephen Brookes, Charles Anthony Richard Hoare, and Andrew William Roscoe. “A theory of communicating sequential processes”. In: *Journal of the ACM* 31.3 (July 1984), pp. 560–599. DOI: <https://doi.org/10.1145/828.833>.
- [BJ05a] J. I. Burton and C. B. Jones. “Atomicity in System Design and Execution”. In: *Journal of Universal Computer Science* 11.5 (2005), pp. 634–635. URL: http://www.jucs.org/jucs_11_5/atomicity_in_system_design/managing.html.
- [BJ05b] J. I. Burton and C. B. Jones. “Investigating Atomicity and Observability”. In: *Journal of Universal Computer Science* 11.5 (2005), pp. 661–686. DOI: 10.3217/jucs-011-05-0661. URL: http://www.jucs.org/jucs_11_5/investigating_atomicity_and_observability.
- [BJ82] Dines Bjørner and Cliff B. Jones, eds. *Formal Specification and Software Development*. Prentice Hall International, 1982. ISBN: 0-13-329003-4. URL: <http://homepages.cs.ncl.ac.uk/cliff.jones/ftp-stuff/BjornerJones1982>.
- [BJ84] Hans Bekič and Cliff B. Jones. *Programming Languages and Their Definition: Hans Bekič (1936-1982). Selected papers*. Vol. 177. Lecture Notes in Computer Science. Springer-Verlag, 1984.
- [BjØ+87] Dines Bjørner, C. B. Jones, M. Mac an Airchinnigh, and E. J. Neuhold, eds. *VDM – A Formal Definition at Work*. Vol. 252.

- Lecture Notes in Computer Science. Springer-Verlag, 1987. ISBN: 978-3-540-17654-1. DOI: 10.1007/3-540-17654-3. URL: <http://www.springer.com/computer/swe/book/978-3-540-17654-1>.
- [BK82] J.A. Bergstra and J.W. Klop. *Fixed point semantics in process algebras*. Tech. rep. IW 206/82. Mathematisch Centrum, Amsterdam (Netherlands), 1982.
- [BKP84] Howard Barringer, Ruurd Kuiper, and Amir Pnueli. “Now You May Compose Temporal Logic Specifications”. In: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*. STOC ’84. Association for Computing Machinery, 1984, pp. 51–63. DOI: 10.1145/800057.808665.
- [BKP85] H. Barringer, R. Kuiper, and A. Pnueli. “A compositional temporal approach to a CSP-like language”. In: *Formal models of programming*. Ed. by E. J. Neuhold and G. Chroust. North-Holland, 1985, pp. 207–227.
- [BM19] Jan A Bergstra and Cornelis A Middelburg. “Process algebra with strategic interleaving”. In: *Theory of Computing Systems* 63.3 (2019), pp. 488–505.
- [BO16] Stephen Brookes and Peter W O’Hearn. “Concurrent Separation Logic”. In: *ACM SIGLOG News* 3.3 (Aug. 2016), pp. 47–65. DOI: <https://doi.org/10.1145/2984450.2984457>.
- [BO80] D. Bjørner and O. N. Oest, eds. *Towards a Formal Description of Ada*. Vol. 98. Lecture Notes in Computer Science. Berlin: Springer-Verlag, 1980. DOI: 10.1007/3-540-10283-3. URL: <http://www.springerlink.com/content/t722165v4142/>.
- [BPS01] Jan A Bergstra, Alban Ponse, and Scott A Smolka, eds. *Handbook of process algebra*. Elsevier, 2001. ISBN: 9780444828309. DOI: <https://doi.org/10.1016/B978-0-444-82830-9.X5017-6>.
- [BR09] Wilfried Brauer and Wolfgang Reisig. “Carl adam Petri and ”Petri nets””. In: *Fundamental concepts in computer science* 3.5 (2009), pp. 129–139.
- [BR21] Steve Brookes and Bill Roscoe. “CSP: a practical process algebra”. In: *Theories of programming: the life and works of Tony Hoare*. Ed. by Cliff B. Jones and Jayadev Misra. ACM, 2021. Chap. 9. DOI: 10.1145/3477355.
- [BR83] S.D. Brookes and W.C. Rounds. “Behavioural Equivalence Relations Induced by Programming Logics”. In: *Automata, Languages and Programming. ICALP 1983. ICALP 1983*. Ed. by J. Diaz. Vol. 154. Lecture Notes in Computer Science. Springer-Verlag, 1983, pp. 97–108. DOI: 10.1007/BFb0036900. URL: <https://doi.org/10.1007/BFb0036900>.
- [BR84] S.D. Brookes and A. W. Roscoe. “An Improved Failures Model for Communicating Processes”. In: *Seminar on Concurrency*,

- Carnegie-Mellon University, Pittsburg, PA, USA, July 9-11, 1984*. Vol. 197. Lecture Notes in Computer Science. Springer-Verlag, 1984, pp. 281–305. DOI: 10.1007/3-540-15670-4_14. URL: https://doi.org/10.1007/3-540-15670-4_14.
- [Bri13] Per Brinch Hansen, ed. *The origin of concurrent programming: from semaphores to remote procedure calls*. 2nd. Springer Science & Business Media, June 2013.
- [Bri73] P. Brinch Hansen. “Concurrent Programming Concepts”. In: *ACM Computing Surveys* 5 (Dec. 1973), pp. 223–245.
- [Bri77] P. Brinch Hansen. *The Architecture of Concurrent Programs*. Englewood Cliffs, N. J.: Prentice Hall, 1977.
- [Bro02] S.D. Brookes. “Traces, Pomsets, Fairness and Full Abstraction for Communicating Processes”. In: *CONCUR 2002 - Proceedings of Concurrency Theory, 13th International Conference, Brno, Czech Republic, August 20-23, 2002*. Vol. 2421. Lecture Notes in Computer Science. Springer-Verlag, 2002, pp. 466–482. DOI: 10.1007/3-540-45694-5_31. URL: https://doi.org/10.1007/3-540-45694-5_31.
- [Bro05] Stephen Brookes. “Retracing the Semantics of CSP”. In: ed. by Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders. Vol. 3525. Lecture Notes in Computer Science. Springer-Verlag, 2005, pp. 1–14. ISBN: 978-3-540-25813-1. DOI: https://doi.org/10.1007/11423348_1. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-25813-1>.
- [Bro75] Frederick P. Brooks. *The mythical man-month : essays on software engineering*. eng. Reading, Mass.: Addison-Wesley Pub. Co., 1975. ISBN: 0201006502.
- [Bro83] Stephen D Brookes. “A Model for Communicating Sequential Processes”. printed as PRG Monograph PRG-35. PhD thesis. University of Oxford, 1983.
- [Bru67] Nicolaas Govert de Bruijn. “Additional comments on a problem in concurrent programming control”. In: *Communications of the ACM* 10.3 (1967), pp. 137–138.
- [BS01] Manfred Broy and Ketil Stølen. *Specification and Development of Interactive Systems*. Springer-Verlag, 2001.
- [BS14] Jos CM Baeten and Davide Sangiorgi. “Concurrency Theory: A Historical Perspective on Coinduction and Process Calculi.” In: *Computational Logic*. Vol. 9. 2014, pp. 399–442.
- [BS76] F. L. Bauer and K. Samelson. *Language Hierarchies and Interfaces*. Vol. 46. Lecture Notes in Computer Science. Springer-Verlag, 1976.
- [BST89] Henri E Bal, Jennifer G Steiner, and Andrew S Tanenbaum. “Programming languages for distributed computing systems”. In: *ACM Computing Surveys (CSUR)* 21.3 (1989), pp. 261–322.

- [Bur69] R. M. Burstall. “Formal Description of program Structure and Semantics in First Order Logic”. In: *Machine Intelligence 5*. Ed. by B. Meltzer and D. Michie. Edinburgh University Press, 1969, pp. 79–98. URL: <https://www.doc.ic.ac.uk/~s/mi/mi5.html>.
- [Bur74] R. M. Burstall. “Program Proving as Hand Simulation with a Little Induction”. In: *Information Processing, Proceedings of the 6th IFIP Congress 1974, Stockholm, Sweden, August 5-10, 1974*. Ed. by J. L. Rosenfeld. North-Holland, 1974, pp. 308–312.
- [BW71] H. Bekič and K. Walk. “Formalization of Storage Properties”. In: *Symposium on Semantics of Algorithmic Languages*. Ed. by E. Engeler. Lecture Notes in Mathematics 188. Springer-Verlag, 1971, pp. 28–61.
- [BW90] J. C. M. Baeten and W. P. Weijland. *Process Algebra*. Cambridge University Press, 1990.
- [BW98] R.-J. R. Back and J. von Wright. *Refinement Calculus: A Systematic Introduction*. New York: Springer-Verlag, 1998.
- [Cal+15] Cristiano Calcagno et al. “Moving Fast with Software Verification”. In: *NASA Formal Methods*. Vol. 9058. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 3–11. DOI: 10.1007/978-3-319-17524-9_1.
- [Car+80] William E Carlson, Larry E Druffel, David A Fisher, and William A Whitaker. “Introducing Ada”. In: *Proceedings of the ACM 1980 annual conference*. 1980, pp. 263–271.
- [CC92] Patrick Cousot and Radhia Cousot. “Abstract interpretation frameworks”. In: *Journal of logic and computation* 2.4 (1992), pp. 511–547.
- [CE81] Edmund M Clarke and E Allen Emerson. “Design and synthesis of synchronization skeletons using branching time temporal logic”. In: *Workshop on Logic of Programs*. Ed. by D. Kozen. Vol. 131. Lecture Notes in Computer Science. Springer-Verlag, 1981, pp. 52–71.
- [CG90] Bernard Carré and Jonathan Garnsworthy. “SPARK—an Annotated Ada Subset for Safety-critical Programming”. In: *Proceedings of the Conference on TRI-ADA '90*. TRI-Ada '90. ACM, 1990, pp. 392–402. DOI: 10.1145/255471.255563.
- [CH72] Maurice Clint and C. A. R. Hoare. “Program proving: Jumps and functions”. In: *Acta Informatica* 1.3 (1972), pp. 214–224.
- [CH74] Roy H Campbell and A Nico Habermann. “The specification of process synchronization by path expressions”. In: *Operating Systems*. Ed. by Kaiser C. Gelenbe E. Vol. 16. Lecture Notes in Computer Science. OS 1974. Springer, Berlin, Heidelberg, 1974, pp. 89–102.

- [CH79] Derek Coleman and Jane W. Hughes. “The clean termination of Pascal programs”. In: *Acta Informatica* 11.3 (1979), pp. 195–210.
- [Cha+81] Ashok Chandra, Joe Halpern, Albert Meyer, and Rohit Parikh. “Equations between regular terms and an application to process logic”. In: *Proceedings of the thirteenth annual ACM symposium on Theory of computing*. 1981, pp. 384–390.
- [CHP71] P.-J. Courtois, F. Heymans, and D. L. Parnas. “Concurrent Control with Readers and Writers”. In: *Communications of the ACM* 14.10 (Oct. 1971), pp. 667–668.
- [CHP72] Pierre-Jacques Courtois, F Heymans, and David Lorge Parnas. “Comments on “A comparison of two synchronizing concepts by PB Hansen””. In: *Acta Informatica* 1.4 (1972), pp. 375–376.
- [CJ00] Pierre Collette and Cliff B. Jones. “Enhancing the Tractability of Rely/Guarantee Specifications in the Development of Interfering Operations”. In: *Proof, Language and Interaction. Essays in Honour of Robin Milner*. Ed. by Gordon Plotkin, Colin Stirling, and Mads Tofte. MIT Press, 2000. Chap. 10, pp. 277–307.
- [CJ06] T. P. Clement and C. B. Jones. “Model-Oriented Specifications”. In: *FACS FACTS 2006-2* (Sept. 2006), pp. 39–53. URL: [/ Documents/CBJ-pubs/papers/find.pdf](#).
- [CJ07] J. W. Coleman and C. B. Jones. “Atomicity: A Unifying Concept in Computer Science”. In: *Journal of Universal Computer Science* 13.8 (2007), pp. 1042–1043. URL: [/ Documents/CBJ-pubs/papers/find.pdf](#).
- [CJ80] Edmund Clarke Jr. “Synthesis of resource invariants for concurrent programs”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 2.3 (1980), pp. 338–358.
- [CM84] K. Mani Chandy and Jayadev Misra. “The drinking philosopher’s problem”. In: *ACM transactions on programming languages and systems* 6.4 (1984), pp. 632–646.
- [CM85] K Mani Chandy and Jayadev Misra. “How processes learn”. In: *Proceedings of the fourth annual ACM symposium on Principles of Distributed Computing*. 1985, pp. 204–214.
- [CM88] K. M. Chandy and J. Misra. *Parallel Program Design: A Foundation*. Addison-Wesley, 1988.
- [Cod62] EF Codd. “Multiprogramming”. In: *Advances in Computers*. Ed. by F. Alt and M. Rubinoff. Vol. 3. Elsevier, 1962, pp. 77–153.
- [Col08] Joseph William Coleman. “Constructing a Tractable Reasoning Framework upon a Fine-Grained Structural Operational Semantics”. PhD thesis. Newcastle University School of Computer Science, Jan. 2008.

- [Con63a] Melvin E Conway. “A multiprocessor system design”. In: *Proceedings of the November 12-14, 1963, Fall Joint Computer Conference*. 1963, pp. 139–146.
- [Con63b] Melvin E Conway. “Design of a separable transition-diagram compiler”. In: *Communications of the ACM* 6.7 (1963), pp. 396–408.
- [Coo66] David C. Cooper. “The equivalence of certain computations”. In: *The Computer Journal* 9.1 (1966), pp. 45–52.
- [Cop20] B. Jack Copeland. “Arthur Prior”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2020. Metaphysics Research Lab, Stanford University, 2020.
- [Cou73] P. Cousot. *A Hoare-style Axiomatization of Burstall’s Intermittent Assertions Method for Non-Deterministic Programs*. Tech. rep. LRIM-83-04. France: Metz University, Sept. 1973.
- [Cri21] James E. Crimmins. “Jeremy Bentham”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2021. Metaphysics Research Lab, Stanford University, 2021.
- [CS14] R. Chapman and F. Schanda. “Are we there yet? 20 years of industrial theorem proving with SPARK”. In: *Interactive Theorem Proving. ITP 2014*. Ed. by G. Klein and R. Gamboa. Vol. 8558. Lecture Notes in Computer Science. Springer-Verlag, 2014, pp. 17–26.
- [DB76] John Darlington and Rod M Burstall. “A system which automatically improves programs”. In: *Acta Informatica* 6.1 (1976), pp. 41–60. DOI: <https://doi.org/10.1007/BF00263742>.
- [DBM75] JW De Bakker and Lambert G. L. T. Meertens. “On the completeness of the inductive assertion method”. In: *Journal of Computer and System Sciences* 11.3 (1975), pp. 323–357. DOI: [https://doi.org/10.1016/S0022-0000\(75\)80056-0](https://doi.org/10.1016/S0022-0000(75)80056-0).
- [DDH72] O.-J. Dahl, E. W. Dijkstra, and C. A. R. Hoare, eds. *Structured Programming*. Academic Press, 1972.
- [Dek22] Theodorus J. (Dirk) Dekker. “History of Dekker’s algorithm for mutual exclusion: a personal view”. In: *Tales of Electrologica : Computers, Software and People*. Ed. by Alberts Gerard and Groote Jan Friso. History of Computing. Springer, 2022. ISBN: 9783031130328. URL: <https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,ip,shib,uid&db=nlebk&AN=3517627&site=ehost-live&scope=site&authtype=shib&custid=s8000044>.
- [Deu76] L Peter Deutsch. *An interactive program verifier (reprint)*. Tech. rep. CSL-73-1. Xerox Palo Alto Research Center, July 1976.
- [DF88] Edsger W. Dijkstra and W. H. Feijen. *A Method of Programming*. 1st. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1988. ISBN: 0201175363.

- [DGKL80] V. Donzeau-Gouge, G. Kahn, and B. Lang. “On the formal definition of Ada. Proceedings of a Workshop, Aarhus, Denmark, January 1980”. In: *Semantics-Directed Compiler Generation. SDCG 1980. Proceedings of a Workshop, Aarhus, Denmark, January 1980*. Ed. by N.D. Jones. Vol. 94. Lecture Notes in Computer Science. Springer-Verlag, 1980, pp. 475–489. URL: http://dx.doi.org/10.1007/3-540-10250-7_34.
- [DH70] C. J. Date and P. Hopewell. *Functional Specifications for PL/I Data Base Support*. Technical Report TR12.? IBM Hursley Laboratory, 1970.
- [Dij01a] Edsger W. Dijkstra. “My recollections of operating system design”. EWD1303, Texas Archive. circulated privately. 2001. URL: <http://www.cs.utexas.edu/users/EWD/ewd13xx/EWD1303.PDF>.
- [Dij01b] Edsger Wybe Dijkstra. *What led to “Notes on Structured Programming”*. Held in the Dijkstra archive online. EWD 1308. June 2001. URL: <http://www.cs.utexas.edu/users/EWD/transcriptions/EWD13xx/EWD1308.html>.
- [Dij13] Edsger W Dijkstra. “Cooperating sequential processes”. In: *The origin of concurrent programming: from semaphores to remote procedure calls*. Ed. by P. Brinch Hansen. 2nd. Springer-Verlag, June 2013, pp. 65–138.
- [Dij62] E. W. Dijkstra. “Over de sequetialiteit van procesbeschrijvingen [On the sequentiality of process descriptions]”. Circulated privately, available in Texas Archive. EWD35. Date inferred. 1962. URL: <https://www.cs.utexas.edu/users/EWD/translations/EWD35-English.html>.
- [Dij63a] Edsger W. Dijkstra. “Multiprogramming en de X8 [Multiprogramming in the X8]”. Circulated privately, available in Texas Archive. EWD51. Date inferred. 1963. URL: <http://www.cs.utexas.edu/users/EWD/ewd00xx/EWD51.PDF>.
- [Dij63b] Edsger W. Dijkstra. “Over seinpalen [On semaphores]”. Circulated privately, available in Texas Archive. EWD74. Date inferred. 1963. URL: <http://www.cs.utexas.edu/users/EWD/ewd00xx/EWD74.PDF>.
- [Dij64] Edsger W. Dijkstra. “Een algoritme ter voorkoming van de dodelijke omarming [An algorithm to prevent the deadly embrace]”. Circulated privately, available in Texas Archive. EWD108. Date inferred. 1964. URL: <http://www.cs.utexas.edu/users/EWD/ewd01xx/EWD108.PDF>.
- [Dij65a] Edsger W. Dijkstra. “Cooperating sequential processes”. EWD123, Texas Archive. Published as [Dij13]. 1965. URL: <http://www.cs.utexas.edu/users/EWD/ewd01xx/EWD123.PDF>.
- [Dij65b] Edsger W. Dijkstra. “Documentatie over de communicatie apparatuur aan de EL X8 [Documentation on the communication

- apparatus of the EL X8]”. EWD 140. Available in Texas archive. 1965. URL: <http://www.cs.utexas.edu/users/EWD/ewd01xx/EWD140.PDF>.
- [Dij65c] EW Dijkstra. “Solution of a problem in concurrent programming control”. In: *Communications of the ACM* 8.9 (1965), p. 569.
- [Dij67] Edsger W. Dijkstra. “An effort towards structuring of programmed processes”. Circulated privately, available in Texas Archive. EWD198. Date inferred. 1967. URL: <http://www.cs.utexas.edu/users/EWD/ewd01xx/EWD198.PDF>.
- [Dij68a] E. W. Dijkstra. “Cooperating sequential processes”. In: *Programming Languages. NATO Advanced Study Institute: lectures given at a three weeks Summer School held in Villard-le-Lans, 1966*. Ed. by F. Genuys. Academic Press, New York, 1968, pp. 43–112.
- [Dij68b] E. W. Dijkstra. “Go to Statement Considered Harmful”. In: *Communications of the ACM* 11.3 (1968), pp. 147–148.
- [Dij68c] E. W. Dijkstra. *Stepwise program construction*. 1968.
- [Dij68d] Edsger W Dijkstra. “Letters to the editor: go to statement considered harmful”. In: *Communications of the ACM* 11.3 (1968), pp. 147–148.
- [Dij68e] Edsger W Dijkstra. “The structure of the “THE” multiprogramming system”. In: *The origin of concurrent programming*. Ed. by Hansen P.B. Springer, New York, NY, 1968, pp. 139–152.
- [Dij71] E. W. Dijkstra. “Hierarchical Ordering of Sequential Processes”. In: *Acta Informatica* 1 (1971), pp. 115–138.
- [Dij72a] Edsger W. Dijkstra. “Information streams sharing a finite buffer”. In: *Information Processing Letters* 1.5 (1972), pp. 179–180. ISSN: 0020-0190. DOI: [https://doi.org/10.1016/0020-0190\(72\)90034-8](https://doi.org/10.1016/0020-0190(72)90034-8). URL: <https://www.sciencedirect.com/science/article/pii/0020019072900348>.
- [Dij72b] EW Dijkstra. “Hierarchical ordering of sequential processes”. In: *Operating systems techniques*. Ed. by C.A.R. Hoare and R.H. Perrott. Academic Press London, 1972, pp. 72–98.
- [Dij75a] E. W. Dijkstra. “Guarded commands, nondeterminacy and formal derivation of programs”. In: *Communications of the ACM* 18 (1975), pp. 453–457.
- [Dij75b] Edsger W. Dijkstra. “After many a sobering experience”. circulated privately. 1975. URL: <http://www.cs.utexas.edu/users/EWD/ewd05xx/EWD500.PDF>.
- [Dij75c] EW Dijkstra. “Guarded commands, non-determinacy, and formal languages”. In: *Communications of the ACM* 18.8 (1975).
- [Dij76a] E. W. Dijkstra. *A Discipline of Programming*. Englewood Cliffs, N.J., USA: Prentice Hall, 1976.
- [Dij76b] Edsger W. Dijkstra. “A letter to Professor Zohar Manna”. Circulated privately, available in Texas archive. EWD574. July

1976. URL: <https://www.cs.utexas.edu/users/EWD/transcriptions/EWD05xx/EWD574.html>.
- [Dij+78] Edsger W Dijkstra, Leslie Lamport, Alain J. Martin, Carel S. Scholten, and Elisabeth FM Steffens. “On-the-fly garbage collection: an exercise in cooperation”. In: *Communications of the ACM* 21.11 (Nov. 1978), pp. 966–975. DOI: <https://doi.org/10.1145/359642.359655>.
- [Dij87] Edsger W. Dijkstra. “Twenty-eight years”. Circulated privately, available in Texas Archive. EWD1000. Jan. 1987. URL: <http://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1000.PDF>.
- [Din00] Jürgen Dingel. “Systematic Parallel Programming”. CMU-CS-99-172. PhD thesis. Carnegie Mellon University, 2000.
- [Dis+19] Dino Distefano, Manuel Fähndrich, Francesco Logozzo, and Peter W. O’Hearn. “Scaling Static Analyses at Facebook”. In: *Communications of the ACM* 62.8 (2019), pp. 62–70.
- [DKK06] Raymond Devillers, Hanna Klaudel, and Maciej Koutny. “A Petri Net Translation of π -calculus Terms”. In: *3rd International Colloquium on Theoretical Aspects of Computing’2006*. Vol. 4281. Lecture Notes in Computer Science. Springer Verlag, 2006, pp. 138–152. URL: <http://www.ibisc.univ-evry.fr/pub/basilic/OUT/Publications/2006/DKK06>.
- [DMB22] Liesbeth De Mol and Maarten Bullynck. “What’s in a name? Origins, transpositions and transformations of the triptych Algorithm – Code – Program”. In: *Abstractions and Embodiments: New Histories of Computing and Society*. Ed. by Janet Abbate and Stephanie Dick. Johns Hopkins University Press, 2022.
- [DMN68] O.-J. Dahl, B. Myhrhaug, and K. Nygaard. *SIMULA 67 Common Base Language*. Tech. rep. S-2. Norwegian Computing Center, Oslo, 1968.
- [DN66] Ole-Johan Dahl and Kristen Nygaard. “SIMULA: an ALGOL-based simulation language”. In: *Communications of the ACM* 9.9 (1966), pp. 671–678.
- [Dod+09] Mike Dodds, Xinyu Feng, Matthew Parkinson, and Viktor Vafeiadis. “Deny-Guarantee Reasoning”. In: *Programming Languages and Systems. ESOP 2009*. Ed. by Giuseppe Castagna. Vol. 5502. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, pp. 363–377. ISBN: 978-3-642-00589-3. URL: http://dx.doi.org/10.1007/978-3-642-00590-9_26.
- [Don76] James Edward Donahue. *Complementary Definitions of Programming Language Semantics*. Vol. 42. Lecture Notes in Computer Science. Springer-Verlag New York, Inc., 1976. ISBN: 038707628X.
- [Don80] V. Donzeau-Gouge. “Formal Definition of the Ada Programming Language”. PhD thesis. INRIA, 1980.
- [DR86] WP De Roever. “Questions to Robin Milner-A Responder’s Commentary”. In: *Information Processing 86: Proceedings of the*

- IFIP 10th World Computer Congress, Dublin, Ireland, September 1-5*. Vol. 10. IFIP Congress Series. North-Holland, 1986, pp. 515–518. ISBN: 0444700773.
- [DREB98] Willem-Paul De Roever, Kai Engelhardt, and Karl-Heinz Buth. *Data refinement: model-oriented proof methods and their comparison*. 1st ed. Cambridge Tracts in Theoretical Computer Science 47. Cambridge University Press, 1998.
- [DW19] N. Dershowitz and R. Waldinger. “Zohar Manna (1939–2018)”. In: *Formal Aspects of Computing* 31.6 (2019), pp. 643–660.
- [DY+13] Thomas Dinsdale-Young, Lars Birkedal, Philippa Gardner, Matthew Parkinson, and Hongseok Yang. “Views: compositional reasoning for concurrent programs”. In: ACM. 2013, pp. 287–300.
- [Ear70] Jay Earley. “An efficient context-free parsing algorithm”. In: *Communications of the ACM* 13.2 (1970), pp. 94–102.
- [EBJ79] P. van Emde Boas and T.M. V. Janssen. *The impact of Frege’s compositionality principle for the semantics of programming and natural languages*. Tech. rep. 79-07. Department of Mathematics, University of Amsterdam, 1979.
- [EC80a] E. Allen Emerson and Edmund M. Clarke. “Characterizing Correctness Properties of Parallel Programs as Fixpoints”. In: *Proc. Seventh Colloquium on Automata, Languages, and Programming*. Ed. by J. W. de Bakker and J. van Leeuwen. Vol. 85. Lecture Notes in Computer Science. Springer-Verlag, 1980, pp. 169–181.
- [EC80b] E Allen Emerson and Edmund M Clarke. “Characterizing correctness properties of parallel programs using fixpoints”. In: *International Colloquium on Automata, Languages, and Programming*. Springer-Verlag. 1980, pp. 169–181.
- [Egl07] Ron Eglash. “Broken metaphor: The master-slave analogy in technical literature”. In: *Technology and culture* 48.2 (2007), pp. 360–369.
- [EH83] E. Allen Emerson and Joseph Y. Halpern. ““Sometimes” and “Not Never” Revisited: On Branching versus Linear Time”. In: *Proc. Tenth Annual ACM Symposium on Principles of Programming Languages*. Austin, Texas: ACM, Jan. 1983, pp. 127–140.
- [EK81] Andrei P Ershov and Donald E Knuth, eds. *Algorithms in Modern Mathematics and Computer Science: Proceedings, Urgench, Uzbek SSR September 16–22, 1979*. Incs 122. Springer, 1981.
- [Eme08] E Allen Emerson. “The beginning of model checking: A personal perspective”. In: *25 Years of Model Checking*. Springer-Verlag, 2008, pp. 27–45.
- [Eme10] E. Allen Emerson. “Meanings of Model Checking”. In: *Concurrency, Compositionality, and Correctness: Essays in Honor of Willem-Paul de Roever*. Ed. by Dennis Dams, Ulrich Hanneemann, and Martin Steffen. Berlin, Heidelberg: Springer Berlin

- Heidelberg, 2010, pp. 237–249. DOI: [10.1007/978-3-642-11512-7_15](https://doi.org/10.1007/978-3-642-11512-7_15). URL: https://doi.org/10.1007/978-3-642-11512-7_15.
- [EN98] E Allen Emerson and Kedar S Namjoshi. “Verification of a parameterized bus arbitration protocol”. In: *International Conference on Computer Aided Verification*. Springer-Verlag, 1998, pp. 452–463.
- [Eng71] E. Engeler, ed. *Symposium on Semantics of Algorithmic Languages*. Lecture Notes in Mathematics 188. Springer-Verlag, 1971.
- [Esp10] Javier Esparza. “A false history of true concurrency: from Petri to tools”. In: *Model Checking Software. SPIN 2010*. Ed. by J. van de Pol and M. Weber. Vol. 6349. Lecture Notes in Computer Science. Springer-Verlag, 2010, pp. 180–186.
- [Est91] Gerald Estrin. “The WEIZAC years (1954-1963)”. In: *Annals of the History of Computing* 13.4 (1991), pp. 317–339.
- [Fen08] Xinyu Feng. “An open framework for certified system software”. PhD thesis. Yale University, 2008.
- [Fen09] Xinyu Feng. “Local rely-guarantee reasoning”. In: *Proceedings of the 36th annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’09. New York, NY, USA, 2009, pp. 315–327. ISBN: 978-1-60558-379-2. DOI: [10.1145/1480881.1480922](https://doi.org/10.1145/1480881.1480922). URL: <http://doi.acm.org/10.1145/1480881.1480922>.
- [FFS07] Xinyu Feng, Rodrigo Ferreira, and Zhong Shao. “On the relationship between concurrent separation logic and assume-guarantee reasoning”. In: *ESOP: Programming Languages and Systems*. Ed. by R. De Nicola. Springer-Verlag, 2007, pp. 173–188.
- [FIS64] A. D. Falkoff, K. E. Iverson, and E. H. Sussenguth. “A Formal Description of SYSTEM/360”. In: *IBM Systems Journal* 3.2 and 3 (June 1964), pp. 198–261.
- [Flo67] R. W. Floyd. “Assigning Meanings to Programs”. In: *Mathematical Aspects of Computer Science*. Ed. by J.T. Schwartz. Vol. 19. Proc. of Symposia in Applied Mathematics. American Mathematical Society, 1967, pp. 19–32.
- [Flo71] Robert W. Floyd. *Toward Interactive Design of Correct Programs*. Tech. rep. AIM-150. Floyd’s talk at IFIP 1971 (Ljubljana). Stanford Artificial Intelligence Project, Sept. 1971.
- [Flo79] Robert W Floyd. “The paradigms of programming”. In: *Communications of the ACM* 22.8 (1979). Turing Award acceptance speech, pp. 455–460.
- [FP78] N. Francez and A. Pnueli. “A Proof Method for Cyclic Programs”. In: *Acta Informatica* 9 (1978), pp. 133–157. DOI: <https://doi.org/10.1007/BF00289074>.

- [Fra76] Nissim Francez. “The analysis of cyclic programs”. PhD thesis. PhD thesis, Weizmann Institute of Science, 1976.
- [Fra86] N. Francez. *Fairness*. Monographs in Computer Science. New York: Springer-Verlag, 1986.
- [FS78] L. Flon and N. Suzuki. *Consistent and Complete Proof Rules for the Total Correctness of Parallel Programs*. Tech. rep. CSI-78-6. Xerox, Palo Alto, 1978.
- [Ga] *System/370 Principles of Operation*. 11th ed. GA-22-7000-10. IBM. Sept. 1987.
- [Gab+80] Dov Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. “On the temporal analysis of fairness”. In: *Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. 1980, pp. 163–173.
- [Ger84] Rob Gerth. “Transition logic: How to reason about temporal properties in a compositional way”. In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. STOC ’84. 1984, pp. 39–50.
- [GF64] Benrard A. Galler and Michael J. Fischer. “An improved equivalence algorithm”. In: *Communications of the ACM* 7.5 (1964), pp. 301–303. DOI: <http://doi.acm.org/10.1145/364099.364331>.
- [GH19] Rob Van Glabbeek and Peter Höfner. “Progress, Justness, and Fairness”. In: *ACM Comput. Surv.* 52.4 (2019). DOI: [10.1145/3329125](https://doi.org/10.1145/3329125).
- [Gil58] Stanley Gill. “Parallel programming”. In: *The computer journal* 1.1 (1958), pp. 2–10.
- [GJF22] Alberts Gerard and Groote Jan Friso, eds. *Tales of Electrológica : Computers, Software and People*. History of Computing. Springer, 2022. ISBN: 9783031130328. URL: <https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,ip,shib,uid&db=nlebk&AN=3517627&site=ehost-live&scope=site&authtype=shib&custid=s8000044>.
- [GMW79] M. Gordon, R. Milner, and C. Wadsworth. *Edinburgh LCF. A mechanised logic of computation*. Vol. 78. Lecture Notes in Computer Science. Springer-Verlag, 1979. ISBN: 978-3-540-09724-2. DOI: <https://doi.org/10.1007/3-540-09724-4>.
- [GN47] Herman H. Goldstine and John von Neumann. *Planning and Coding of Problems for an Electronic Computing Instrument*. Tech. rep. Institute of Advanced Studies, Princeton, 1947.
- [Goe06] Brian Goetz. *Java concurrency in practice*. Pearson Education, 2006.
- [Gor+78] Mike Gordon, Robin Milner, Lockwood Morris, Malcolm Newey, and Christopher Wadsworth. “A metalanguage for interactive proof in LCF”. In: *Proceedings of the 5th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*.

- POPL'78. New York, NY, USA, Jan. 1978, pp. 119–130. ISBN: 9781450373487. DOI: <https://doi.org/10.1145/512760.512773>.
- [Got68] C. C. Gotlieb. *Letter to Zohar Manna*. John McCarthy Papers (SC0524) Department of Special Collections and University Archives, Stanford University Libraries, Stanford, Calif. Sept. 1968.
- [Gri22] David Gries. “Development of Correct Programs”. In: *Edsger Wybe Dijkstra: His Life, Work, and Legacy*. Ed. by Krzysztof R Apt and Tony Hoare. ACM, 2022. Chap. 7, pp. 141–168.
- [Gri77] David Gries. “An Exercise in Proving Parallel Programs Correct”. In: *Communications of the ACM* 20.12 (1977), pp. 921–930.
- [Gri78] David Gries. *Programming Methodology: A Collection of Articles by Members of IFIP WG 2.3*. Berlin, Heidelberg: Springer-Verlag, 1978. ISBN: 0387903291.
- [Gue81] Pedro Guerreiro. “Relational semantics of strongly communicating sequential processes”. In: *Formalization of Programming Concepts*. Ed. by J. Díaz and I. Ramos. Lecture Notes in Computer Science. Springer-Verlag, 1981, pp. 310–326.
- [Gut75] J. V. Guttag. “The Specification and Application to Programming of Abstract Data Types”. CSRG-59. PhD thesis. University of Toronto, Computer Systems Research Group, Sept. 1975.
- [Han04] Per Brinch Hansen. *A PROGRAMMER'S STORY The Life of a Computer Pioneer*. 2004.
- [Han72] P. Brinch Hansen. “A Comparison of Two Synchronizing Concepts”. In: *Acta Informatica* 1.3 (1972), pp. 190–199.
- [Har10a] David Harel. *Amir Pnueli: A Gentle Giant, Lord of the ϕ 's and the ψ 's*. Talk given at FLoC. Video on YouTube. July 2010. URL: https://www.youtube.com/watch?v=KJ9WaUa_4m0.
- [Har10b] David Harel. “Amir Pnueli. A Gentle Giant: Lord of the ϕ 's and the ψ 's”. In: *Formal Aspects of Computing* 22.6 (2010), pp. 663–665.
- [Hay+13] Ian J. Hayes, Alan Burns, Brijesh Dongol, and Cliff B. Jones. “Comparing Degrees of Non-Determinism in Expression Evaluation”. In: *The Computer Journal* 56.6 (2013), pp. 741–755. DOI: 10.1093/comjnl/bxt005. URL: <http://comjnl.oxfordjournals.org/content/56/6/741>.
- [Hay16] I. J. Hayes. “Generalised rely-guarantee concurrency: An algebraic foundation”. In: *Formal Aspects of Computing* 28.6 (Nov. 2016), pp. 1057–1078. DOI: 10.1007/s00165-016-0384-0. URL: <http://dx.doi.org/10.1007/s00165-016-0384-0>.
- [Hay87] I. J. Hayes, ed. *Specification Case Studies*. Prentice Hall International, 1987.

- [HBR81] C.A.R. Hoare, S.D. Brookes, and A.W. Roscoe. *A theory of communicating sequential processes*. Tech. rep. Oxford, May 1981. URL: <https://www.cs.ox.ac.uk/publications/publication3738-abstract.html>.
- [HBS73] Carl Hewitt, Peter Bishop, and Richard Steiger. “A universal modular actor formalism for artificial intelligence”. In: *Proceedings of the 3rd international joint conference on Artificial intelligence*. 1973, pp. 235–245.
- [HC68] G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic*. New Accents. London, UK: Methuen, 1968. ISBN: 9780416294606. URL: <https://books.google.co.uk/books?id=CukMAQAAIAAJ>.
- [He05] Jifeng He. “Linking theories of concurrency”. In: *Communicating Sequential Processes. The First 25 Years*. Ed. by Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders. Vol. 3525. Lecture Notes in Computer Science. Springer-Verlag, 2005, pp. 61–74. ISBN: 978-3-540-25813-1. DOI: 10.1007/b136154. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-25813-1>.
- [Heh84] Eric CR Hehner. “Predicative programming part I”. In: *Communications of the ACM* 27.2 (1984), pp. 134–143.
- [Hen04] Neil Henderson. “Formal Modelling and Analysis of an Asynchronous Communication Mechanism”. PhD thesis. University of Newcastle upon Tyne, 2004.
- [Hen88] Matthew Hennessy. *Algebraic theory of processes*. The Foundations of Computing Series. MIT press, 1988. ISBN: 0262081717. DOI: <https://dl.acm.org/doi/book/10.5555/50497>. URL: <http://sro.sussex.ac.uk/id/eprint/1377>.
- [Her91] Maurice Herlihy. “Wait-free synchronization”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 13.1 (1991), pp. 124–149.
- [Hew06] Carl Hewitt. “What is commitment? physical, organizational, and social (revised)”. In: *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*. Springer. 2006, pp. 293–307.
- [Hew76] Carl Hewitt. *Viewing control structures as patterns of passing messages*. A.I. Memo 410. MIT, 1976, pp. 323–364.
- [Hew77] Carl Hewitt. “Viewing control structures as patterns of passing messages”. In: *Artificial intelligence* 8.3 (1977), pp. 323–364.
- [HH10] Jifeng He and C. A. R. Hoare. “CSP is a retract of CCS”. In: *Theoretical Computer Science* 411.11-13 (2010), pp. 1311–1337. DOI: 10.1016/j.tcs.2009.12.012.
- [HH75] Per Brinch Hansen and Alfred C Hartmann. *Sequential PASCAL report*. Tech. rep. California Institute of Technology, 1975.

- [HH83] E.C.R. Hehner and C.A.R. Hoare. “A More Complete Model of Communicating Processes”. In: *Theoretical Computer Science* 26.1–2 (Sept. 1983), pp. 105–120.
- [HJ16] C. A. R. Hoare and Cliff B. Jones. *Professor Sir Tony Hoare: ACM Turing Award Winner 1980*. Online. Interview. Mar. 2016.
- [HJ18] I. J. Hayes and C. B. Jones. “A guide to Rely/Guarantee thinking”. In: *Engineering Trustworthy Software Systems – Third International School, SETSS 2017*. Ed. by Jonathan Bowen, Zhiming Liu, and Zili Zhan. Vol. 11174. Lecture Notes in Computer Science. Springer-Verlag, 2018, pp. 1–38.
- [HK91] Iain Houston and Steve King. “CICS project report experiences and results from the use of Z in IBM”. In: *VDM’91 Formal Software Development Methods. VDM 1991*. Ed. by S Prehn and W. J. Toetenel. Vol. 551. Lecture Notes in Computer Science. Springer-Verlag, 1991, pp. 588–596.
- [HKP82] David Harel, Dexter Kozen, and Rohit Parikh. “Process logic: Expressiveness, decidability, completeness”. In: *Journal of computer and system sciences* 25.2 (1982), pp. 144–170.
- [HKSNO2] Ulf Hashagen, Reinhard Keil-Slawik, and Arthur Norberg, eds. *History of Computing: Software Issues: International Conference on the History of Computing, ICHC 2000 April 5–7, 2000 Heinz Nixdorf Museums Forum Paderborn, Germany*. Springer-Verlag, 2002.
- [HL20] C. A. R. Hoare and L. Lamport. *Virtual HLF 2020 – Dialogue: Sir C. Antony R. Hoare/Leslie Lamport*. Sept. 2020. URL: <https://www.youtube.com/watch?v=wQbFkAkThGk>.
- [HL74] C. A. R. Hoare and P. E. Lauer. “Consistent and complementary formal theories of the semantics of programming languages”. English. In: *Acta Informatica* 3.2 (1974), pp. 135–153. ISSN: 0001-5903. DOI: 10.1007/BF00264034. URL: <http://dx.doi.org/10.1007/BF00264034>.
- [HM80] M. Hennessy and R. Milner. “On observing nondeterminism and concurrency”. In: *Automata, Languages and Programming. Seventh Colloquium, Noordwijkerhout, July 1980*. Ed. by J. W. de Bakker and J. van Leeuwen. Vol. 85. Lecture Notes in Computer Science. Springer-Verlag, 1980, pp. 299–309. DOI: http://dx.doi.org/10.1007/3-540-10003-2_79.
- [HM85] M. Hennessy and R. Milner. “Algebraic Laws for Nondeterminism and Concurrency”. In: *Journal of the ACM* 32.1 (Jan. 1985), pp. 137–161. DOI: 10.1145/2455.2460. URL: <https://doi.org/10.1145/2455.2460>.
- [HMM83] Joseph Halpern, Zohar Manna, and Ben Moszkowski. “A hardware semantics based on temporal intervals”. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 1983, pp. 278–291.

- [HO83] C.A.R. Hoare and E.R. Olderog. “Specification-oriented Semantics for Communicating Processes”. In: *Automata Languages and Programming 10th Colloquium*. Vol. 154. Lecture Notes in Computer Science. Springer-Verlag, 1983, pp. 561–572.
- [Hoa05] Tony Hoare. “Process algebra: A unifying approach”. In: *Engineering Theories of Software Intensive Systems*. Springer-Verlag, 2005, pp. 257–283.
- [Hoa13] Charles Antony Richard Hoare. “Towards a theory of parallel programming”. In: *The origin of concurrent programming: from semaphores to remote procedure calls*. Ed. by Per Brinch Hansen. 2nd. Springer-Verlag, June 2013, pp. 231–244.
- [Hoa68] C.A.R. Hoare. “Record Handling”. In: *Programming Languages*. Ed. by F. Genuys. cbj has copy of Genuys’ book. Academic Press, 1968, pp. 291–347.
- [Hoa69] C. A. R. Hoare. “An axiomatic basis for computer programming”. In: *Communications of the ACM* 12.10 (1969), pp. 576–580.
- [Hoa71a] C. A. R. Hoare. “Procedures and Parameters: An Axiomatic Approach”. In: *Symposium On Semantics of Algorithmic Languages*. Ed. by E. Engeler. Vol. 188. LNM. Springer-Verlag, 1971, pp. 102–116.
- [Hoa71b] C. A. R. Hoare. “Proof of a Program: FIND”. In: *Communications of the ACM* 14.1 (Jan. 1971), pp. 39–45. URL: <http://doi.acm.org/10.1145/362452.362489>.
- [Hoa72a] C. A. Hoare. “Proof of Correctness of Data Representations”. In: *Acta Informatica* 1.4 (Dec. 1972). also have hard copy of a draft (for history), pp. 271–281. ISSN: 0001-5903. DOI: 10.1007/BF00289507. URL: <http://dx.doi.org/10.1007/BF00289507>.
- [Hoa72b] C. A. R. Hoare. “A Note on the FOR Statement”. In: *BIT* 12.3 (1972), pp. 334–341.
- [Hoa72c] C. A. R. Hoare. “Towards a Theory of Parallel Programming”. In: *Operating System Techniques*. Ed. by C. A. R. Hoare and R. Perrott. Academic Press, 1972, pp. 61–71.
- [Hoa72d] Charles Antony Richard Hoare. “Towards a theory of parallel programming”. In: *Operating System Techniques*. Ed. by C. A. R. Hoare and R. H. Perrott. Proceedings of a Seminar at Queen’s University, Belfast, August–September 1971. Academic Press, New York, 1972, pp. 61–71.
- [Hoa73] C.A.R. Hoare. “A Structured Paging System”. In: *BCS, Computer Journal* 16.3 (Aug. 1973), pp. 209–215.
- [Hoa74a] C. A. R. Hoare. “Monitors: An Operating System Structuring Concept”. In: *Communications of the ACM* 17.10 (1974), pp. 549–557.

- [Hoa74b] C.A.R. Hoare. “Hints on Programming Language Design”. In: *State of the Art Report 20: Computer Systems Reliability*. Ed. by C.J. Bunyan. Pergamon/Infotech, 1974, pp. 505–534.
- [Hoa75] Charles Antony Richard Hoare. “Parallel programming: an axiomatic approach”. In: *Computer Languages 1.2* (June 1975). also have hard copy, pp. 151–160.
- [Hoa76] C. A. R. Hoare. “Parallel Programming: An Axiomatic Approach”. In: *[BS76]*. Springer-Verlag, 1976, pp. 11–42.
- [Hoa78] C. A. R. Hoare. “Communicating Sequential Processes”. In: *Communications of the ACM 21.8* (1978), pp. 666–677.
- [Hoa80a] C.A.R. Hoare. “A Model for Communicating Sequential Processes”. In: *On the Construction of Programs*. Ed. by R.M. McKeag and A.M. MacNaughten. Cambridge University Press, 1980, pp. 229–254.
- [Hoa80b] Charles Antony Richard Hoare. “A model for communicating sequential process”. In: (1980). also have hard copy of PRG monograph (for history).
- [Hoa81a] C. A. R. Hoare. *A Model for Communicating Sequential Processes*. Tech. rep. PRG-22. Programming Research Group, Oxford University, 1981. URL: <http://www.cs.ox.ac.uk/publications/publication3766-abstract.html>.
- [Hoa81b] C.A.R. Hoare. “The Emperor’s Old Clothes: The ACM Turing Award Lecture”. In: *Communications of the ACM 24.2* (Feb. 1981), pp. 75–83.
- [Hoa81c] Charles Antony Richard Hoare. “A calculus of total correctness for communicating processes”. In: *Science of Computer Programming 1.1-2* (1981). also have hard copy of PRG monograph (for history), pp. 49–72.
- [Hoa85] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
- [Hoa91] C. A. R. Hoare. “The transputer and occam: A personal story”. In: *Concurrency: Practice and Experience 3.4* (1991), pp. 249–264. DOI: 10.1002/cpe.4330030403.
- [Hon94] K Honda. “A Study of ν -calculus and its Combinatory Representation”. Hard copy. PhD thesis. Phd Thesis in the Department of Computer Science, 1994.
- [Hor+74] James J Horning, Hugh C Lauer, PM Melliar-Smith, and Brian Randell. “A program structure for error detection and recovery”. In: *Operating Systems, Proceedings of an International Symposium*. 1974, pp. 171–187.
- [Hov19] Gauthier van den Hove. “New Insights from Old Programs: The Structure of the First ALGOL 60 System”. PhD thesis. University of Amsterdam, 2019.
- [How76a] John H Howard. “Proving monitors”. In: *Communications of the ACM 19.5* (1976), pp. 273–279.

- [How76b] John H. Howard. “Signaling in Monitors”. In: *Proceedings of the 2nd International Conference on Software Engineering*. ICSE ’76. IEEE Computer Society Press, 1976, pp. 47–52.
- [HS14] Tony Hoare and Stephan van Staden. “The laws of programming unify process calculi”. In: *Science of Computer Programming* 85 (2014), pp. 102–114.
- [HW87] Maurice P Herlihy and Jeannette M Wing. “Axioms for concurrent objects”. In: *Proceedings of the 14th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. 1987, pp. 13–26.
- [HW90] Maurice Herlihy and Jeannette M. Wing. “Linearizability: A Correctness Condition for Concurrent Objects.” In: *ACM Trans. Program. Lang. Syst.* 12.3 (1990), pp. 463–492.
- [IBM68] IBM. *System/360 Principles of Operation*. 8th ed. A22-6821-7. IBM. Sept. 1968.
- [Ich84] Jean Ichbiah. “ADA: Past, Present, Future”. In: *Communications of the ACM* 27.10 (Oct. 1984), pp. 990–997. DOI: 10.1145/358274.358278. URL: <https://doi.org/10.1145/358274.358278>.
- [ISO87] ISO/IEC. *Programming languages — Ada*. ISO Standard 8652:1987. JTC 1/SC 22, 1987.
- [Itu67] Renato Iturriaga. “Contributions to mechanical mathematics”. PhD thesis. Carnegie-Mellon University Pittsburgh, Pennsylvania, 1967.
- [Ive62] K. E. Iverson. *A Programming Language*. J. Wiley, 1962.
- [JA16] Cliff B. Jones and Troy K. Astarte. *An exegesis of four formal descriptions of ALGOL 60*. Technical Report CS-TR-1498. Newcastle University School of Computer Science, Sept. 2016.
- [JA18] Cliff B. Jones and Troy K. Astarte. “Challenges for semantic description: comparing responses from the main approaches”. In: *Proceedings of the Third School on Engineering Trustworthy Software Systems*. Ed. by Jonathan P. Bowen, Zili Zhang, and Zhiming Liu. Vol. 11174. Lecture Notes in Computer Science. 2018, pp. 176–217. DOI: 10.1007/978-3-030-02928-9_6.
- [Jac00] Michael Jackson. *Problem Frames: Analyzing and Structuring Software Development Problems*. Addison-Wesley, 2000.
- [Jac12] D. N. Jackson. *Software Abstractions: logic, language, and analysis*. MIT Press, 2012.
- [Jak20] David Jakobsen. “A.N. Prior and ‘The Nature of Logic’”. In: *History and Philosophy of Logic* 41.1 (2020), pp. 71–81. URL: <https://doi.org/10.1080/01445340.2019.1605479>.
- [JEB79] TMV Janssen and P van Emde-Boas. “The impact of Frege’s compositionality principle for the semantics of programming and natural languages”. In: *Proc. of the First Frege memorial conference*. 1979, pp. 110–129.

- [JH05] He Jifeng and CAR Hoare. “Linking theories of concurrency”. In: *International Colloquium on Theoretical Aspects of Computing*. Springer-Verlag, 2005, pp. 303–317.
- [JH16] Cliff B. Jones and Ian J. Hayes. “Possible values: Exploring a concept for concurrency”. In: *Journal of Logical and Algebraic Methods in Programming* 85.5 (2016), pp. 972–984. DOI: 10.1016/j.jlamp.2016.01.002.
- [JHC15] Cliff B. Jones, Ian J. Hayes, and Robert J. Colvin. “Balancing Expressiveness in Formal Approaches to Concurrency”. In: *Formal Aspects of Computing* 27.3 (2015), pp. 465–497. URL: <http://link.springer.com/article/10.1007/s00165-014-0310-2>.
- [JL71] C. B. Jones and P. Lucas. “Proving Correctness of Implementation Techniques”. In: *A Symposium on Algorithmic Languages*. Ed. by E. Engeler. Vol. 188. Lecture Notes in Mathematics. Springer-Verlag, 1971, pp. 178–211. DOI: 10.1007/BFb0059698. URL: <http://dx.doi.org/10.1007/BFb0059698>.
- [JM21a] Cliff B. Jones and Jayadev Misra. “Finding Effective Abstractions”. In: *Theories of programming: the life and works of Tony Hoare*. Ed. by Cliff B. Jones and Jayadev Misra. ACM, 2021. Chap. 2. DOI: 10.1145/3477355.
- [JM21b] Cliff B. Jones and Jayadev Misra, eds. *Theories of programming: the life and works of Tony Hoare*. ACM, 2021. DOI: 10.1145/3477355.
- [Jon01] C. B. Jones. “The Transition from VDL to VDM”. In: *Journal of Universal Computer Science* 7.8 (2001), pp. 631–640. DOI: 10.3217/jucs-007-08-0631. URL: http://www.jucs.org/jucs_7_8/the_transition_from_VDL.
- [Jon03] Cliff B. Jones. “The Early Search for Tractable Ways of Reasoning about Programs”. In: *IEEE Annals of the History of Computing* 25.2 (2003), pp. 26–49. DOI: 10.1109/MAHC.2003.1203057. URL: <http://doi.ieeecomputersociety.org/10.1109/MAHC.2003.1203057>.
- [Jon+05a] C. B. Jones, D. Lomet, A. Romanovsky, and G. Weikum. “The Atomic Manifesto”. In: *Journal of Universal Computer Science* 11.5 (2005), pp. 636–650. DOI: 10.3217/jucs-011-05-0636. URL: http://www.jucs.org/jucs_11_5/the_atomic_manifesto.
- [Jon+05b] C. B. Jones, D. Lomet, A. Romanovsky, and G. Weikum. *The Atomicity Manifesto*. 2005.
- [Jon07] C. B. Jones. “Splitting Atoms Safely”. In: *Theoretical Computer Science* 375.1–3 (2007), pp. 109–119. DOI: 10.1016/j.tcs.2006.12.029.
- [Jon09] Cliff B. Jones. *Rely/guarantee thinking and its role in splitting atoms safely*. PSY Workshop at CAV, Grenoble. June 2009.

- [Jon17] Cliff B. Jones. “Turing’s 1949 paper in context”. In: *Computability in Europe 2017*. Ed. by Jarkko Kari, Florain Manea, and Ion Petre. Vol. 10307. Lecture Notes in Computer Science. Springer-Verlag, 2017, pp. 21–41.
- [Jon20] Cliff B. Jones. *Understanding Programming Languages*. Springer-Verlag, 2020. URL: <https://doi.org/10.1007/978-3-030-59257-8>.
- [Jon23] Cliff B. Jones. “Three Early Formal Approaches to the Verification of Concurrent Programs”. In: *Minds and Machines (2023)*. URL: <https://doi.org/10.1007/s11023-023-09621-5>.
- [Jon70] C. B. Jones. *A Technique for Showing that Two Functions Preserve a Relation Between Their Domains*. Tech. rep. LR 25.3.067. IBM Laboratory, Vienna, Apr. 1970.
- [Jon71] C. B. Jones. *Formal Development of Correct Algorithms: an Example Based on Earley’s Recogniser*. Tech. rep. 12.095. IBM Laboratory Hursley, Dec. 1971. URL: <http://www.cs.newcastle.ac.uk/publications/inproceedings/papers/550.pdf>.
- [Jon72a] C. B. Jones. “Formal Development of Correct Algorithms: an Example Based on Earley’s Recogniser”. In: *SIGPLAN Notices*. Vol. 7:1. ACM, 1972, pp. 150–169. URL: http://portal.acm.org/ft_gateway.cfm?id=807083&type=pdf&coll=ACM&d1=ACM&CFID=23963294&CFTOKEN=25930840.
- [Jon72b] C. B. Jones. *Operations and Formal Development*. Tech. rep. TN 9004. IBM Laboratory, Hursley, Sept. 1972.
- [Jon73] C. B. Jones. *Formal Development of Programs*. Tech. rep. 12.117. IBM Laboratory Hursley, June 1973.
- [Jon76] C. B. Jones. *Formal Definition in Compiler Development*. Tech. rep. 25.145. IBM Laboratory Vienna, Feb. 1976.
- [Jon80a] C. B. Jones. *Software Development: A Rigorous Approach*. Englewood Cliffs, N.J., USA: Prentice Hall International, 1980. ISBN: 0-13-821884-6. URL: <http://portal.acm.org/citation.cfm?id=539771>.
- [Jon80b] N. D. Jones, ed. *Semantics-Directed Compiler Generation. SDCG 1980. Proceedings of a Workshop, Aarhus, Denmark, January 1980*. Vol. 94. Lecture Notes in Computer Science. Springer-Verlag, 1980.
- [Jon81] C. B. Jones. “Development Methods for Computer Programs including a Notion of Interference”. Printed as: Programming Research Group, Technical Monograph 25. PhD thesis. Oxford University, June 1981.
- [Jon83a] C. B. Jones. “Tentative Steps Toward a Development Method for Interfering Programs”. In: *Transactions on Programming Languages and System* 5.4 (1983), pp. 596–619. DOI: 10.1145/

- 69575.69577. URL: <http://doi.acm.org/10.1145/69575.69577>.
- [Jon83b] C. B. Jones. “Specification and Design of (Parallel) Programs”. In: *Proceedings of IFIP’83*. North-Holland, 1983, pp. 321–332.
- [Jon86] C. B. Jones. *Systematic Software Development Using VDM*. Prentice Hall International, 1986. ISBN: 0-13-880717-5.
- [Jon87] C. B. Jones. “Program Specification and Verification in VDM”. In: *Logic of Programming and Calculi of Discrete Design – NATO ASI Series F: Computer and Systems Sciences, Vol. 36*. Ed. by M. Broy. Vol. 36. Springer-Verlag, 1987, pp. 149–184.
- [Jon90] C. B. Jones. *Systematic Software Development Using VDM*. Second. Prentice Hall International, 1990. ISBN: 0-13-880733-7. URL: <http://homepages.cs.ncl.ac.uk/cliff.jones/ftp-stuff/Jones1990.pdf>.
- [Jon96] C. B. Jones. “Accommodating Interference in the Formal Design of Concurrent Object-Based Programs”. In: *Formal Methods in System Design* 8.2 (Mar. 1996), pp. 105–122. ISSN: 0925-9856. DOI: 10.1007/BF00122417. URL: <http://dx.doi.org/10.1007/BF00122417>.
- [JP08] Cliff B. Jones and Ken G. Pierce. “Splitting Atoms with Rely/Guarantee Conditions Coupled with Data Reification”. In: *ABZ2008*. Lecture Notes in Computer Science 5238. Springer-Verlag, 2008, pp. 360–377. DOI: 10.1007/978-3-540-87603-8_47. URL: <http://www.springerlink.com/content/d63746175654u503/fulltext.pdf>.
- [JP11] Cliff B. Jones and Ken G. Pierce. “Elucidating concurrent algorithms via layers of abstraction and reification”. In: *Formal Aspects of Computing* 23.3 (2011), pp. 289–306. DOI: 10.1007/s00165-010-0156-1. URL: <http://www.springerlink.com/content/e52509k41r31g880/>.
- [JR10] C.B. Jones and A.W. Roscoe. “Insight, inspiration and collaboration”. In: *Reflections on the work of C.A.R. Hoare*. Ed. by Cliff B. Jones, A. W. Roscoe, and Kenneth Wood. Springer-Verlag, 2010. Chap. 1, pp. 1–32. DOI: 10.1007/978-1-84882-912-1_1. URL: http://dx.doi.org/10.1007/978-1-84882-912-1_1.
- [JRW10] Cliff B Jones, A William Roscoe, and Kenneth R Wood, eds. *Reflections on the Work of C.A.R. Hoare*. Springer Science & Business Media, 2010.
- [JT22] C.B. Jones and M. Thomas. “The development and deployment of formal methods in the UK”. In: *Formal Aspects of Computing* 34.1 (2022), pp. 1–21. DOI: 10.1145/3522577.
- [JY15] Cliff B. Jones and Nisansala Yatapanage. “Reasoning about Separation Using Abstraction and Reification”. In: *Software Engineering and Formal Methods*. Ed. by Radu Calinescu and Bernhard Rumpe. Vol. 9276. Lecture Notes in Computer Science.

- Springer-Verlag, 2015, pp. 3–19. DOI: 10.1007/978-3-319-22969-0_1. URL: http://dx.doi.org/10.1007/978-3-319-22969-0_1.
- [JY19] Cliff B. Jones and Nisansala Yatapanage. “Investigating the limits of rely/guarantee relations based on a concurrent garbage collector example”. In: *Formal Aspects of Computing* 31.3 (2019). on-line April 2018, pp. 353–374. DOI: 10.1007/s00165-019-00482-3.
- [KA03] F. E. J. Kruseman Aretz. *The Dijkstra–Zonneveld ALGOL 60 compiler for the Electrologica X1 The Dijkstra–Zonneveld ALGOL 60 compiler for the Electrologica X1*. Technical Note SEN-N0301. CWI, June 2003.
- [Kah74] Gilles Kahn. “The semantics of a simple language for parallel programming”. In: *Information Process, Proceedings of the 6th IFIP Congress 1974, Stockholm, Sweden, August 5-10, 1974*. Ed. by J. L. Rosenfeld. Vol. 6. IFIP Congress Series. North-Holland, Aug. 1974, pp. 471–475.
- [Kah79] G. Kahn, ed. *Semantics of Concurrent Computation: Proceedings, Evian, France 1979*. Vol. 70. Lecture Notes in Computer Science. Berlin: Springer-Verlag, 1979.
- [Kam68] Hans Kamp. “Tense Logic and the Theory of Linear Order”. PhD thesis. University of California, Los Angeles, 1968.
- [Kam70] Hans Kamp. “Reviewed Works: ”And Next.” by Georg Henrik von Wright; ”And Then.” by G. H. von Wright; Quelques Remarques sur la Logique du Temps et les Systemes Modales by G. H. von Wright”. In: *The Journal of Symbolic Logic* 35.3 (1970), pp. 459–460. ISSN: 00224812. URL: <http://www.jstor.org/stable/2270720>.
- [Kay93] Alan C. Kay. “The Early History of Smalltalk”. In: *SIGPLAN Notices* 28.3 (Mar. 1993), pp. 69–95. DOI: 10.1145/155360.155364.
- [Kin69] J. C. King. “A Program Verifier”. PhD thesis. Department of Computer Science, Carnegie-Mellon University, 1969.
- [Kin71] J. C. King. “A Program Verifier”. In: *Information Processing, Proceedings of IFIP Congress 1971. Volume 1 - Foundations and Systems, Ljubljana, Yugoslavia, August 23-28, 1971*. Ed. by Charles V. Freiman, John E. Griffith, and Jack L. Rosenfeld. North-Holland, 1971, pp. 234–249.
- [Klo16] Jan Willem Klop. “The blue preprint, a serendipity thing”. In: ed. by Inge Bethke, Bert Bredeweg, and Alban Ponse. Universiteit van Amsterdam Informatics Institute, Dec. 2016.
- [Klo19] Jan Willem Klop. *Interview with Jan Willem Klop*. Conducted by Gracia Redder and Bart de Haan. Jan. 2019.

- [KM77] G. Kahn and D. MacQueen. “Coroutines and networks of parallel processes”. In: *Information Processing’77*. Ed. by B. Gilchrist. 1977, pp. 993–998.
- [Knu66] Donald E Knuth. “Additional comments on a problem in concurrent programming control”. In: *Communications of the ACM* 9.5 (1966), pp. 321–322.
- [Knu81] Donald E Knuth. “Algorithms in modern mathematics and computer science”. In: *Algorithms in Modern Mathematics and Computer Science: Proceedings, Urgench, Uzbek SSR September 16–22, 1979*. Ed. by Andrei P Ershov and Donald E Knuth. Incs 122. Springer. Springer, 1981, pp. 82–99.
- [Krö78] F Kröger. “A uniform logical basis for the description, specification and verification of programs”. In: *Proceedings, IFIP Working Conference on Formal Description of Programming Concepts, St. Andrews, Canada, August 1977*. North-Holland, Amsterdam, 1978, pp. 441–457.
- [Krö87] Fred Kröger. *Temporal Logic of Programs*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1987.
- [LA94] Leslie Lamport and Martin Abadi. “Decomposing specifications of concurrent systems”. In: *Programming Concepts, Methods and Calculi, Ernst-RÄ1/Ädiger Olderog editor. (Proceedings of the IFIP TC2/WG2. 1/WG2. 2/WG2. 3 Working Conference, Procomet’94, San Miniato, Italy.) North-Holland*. 1994, pp. 327–340.
- [Lai+19] Wenchen Lai, You Hu, Gracia Redder, and Bart de Haan. *On the crest of the waves: A history of Process Algebra*. Master’s students’ coursework. Jan. 2019.
- [Lam01] A. van Lamsweerde. “Goal-oriented requirements engineering: a guided tour”. In: *Proceedings Fifth IEEE International Symposium on Requirements Engineering*. 2001, pp. 249–262. DOI: 10.1109/ISRE.2001.948567.
- [Lam02] Leslie Lamport. *Specifying systems: the TLA+ language and tools for hardware and software engineers*. USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN: 978-0-321-14306-8. DOI: <https://dl.acm.org/doi/book/10.5555/579617>.
- [Lam74] Leslie Lamport. “A new solution of Dijkstra’s concurrent programming problem”. In: *Communications of the ACM* 17.8 (1974), pp. 453–455. DOI: <https://dl.acm.org/doi/10.1145/361082.361093>.
- [Lam77] L. Lamport. “Proving the Correctness of Mutiprocess Programs”. In: *IEEE Transactions on Software Engineering* 3 (Mar. 1977), pp. 125–143. DOI: <https://doi.org/10.1109/TSE.1977.229904>.
- [Lam80] Leslie Lamport. ““Sometime” is Sometimes “Not never”: On the Temporal Logic of Programs”. In: *Proceedings of the 7th ACM*

- SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. Las Vegas, Nevada, Jan. 1980, pp. 174–185. DOI: <https://doi.org/10.1145/567446.567463>.
- [Lam83] Leslie Lamport. “What Good is Temporal Logic”. In: *Information Processing '83*. Elsevier Science Inc., 1983, pp. 657–668. URL: <https://www.microsoft.com/en-us/research/publication/good-temporal-logic/>.
- [Lam90] L. Lamport. *A Temporal Logic of Actions*. Tech. rep. 57. DEC, SRC, 1990. URL: <https://lampport.azurewebsites.net/pubs/old-tla-src.pdf>.
- [Lam93] Leslie Lamport. “Verification and specification of concurrent programs”. In: *A Decade of Concurrency Reflections and Perspectives. REX 1993*. Lecture Notes in Computer Science 803. Workshop/School/Symposium of the REX Project (Research and Education in Concurrent Systems). Springer-Verlag, 1993, pp. 347–374.
- [Lam94] Leslie Lamport. “The temporal logic of actions”. In: *ACM Transactions on Programming Languages and Systems* 16.3 (May 1994), pp. 872–923. DOI: <https://doi.org/10.1145/177492.177726>.
- [Lam97] Leslie Lamport. “Composition: A way to make proofs harder”. In: *International Symposium on Compositionality*. Springer, 1997, pp. 402–423.
- [Lam99] Leslie Lamport. “Specifying Concurrent Systems with TLA⁺”. In: *Calculational System Design*. Ed. by M. Broy and R. Steinbrüggen. IOS Press, 1999, pp. 183–247.
- [Lau68] Peter E. Lauer. *Formal definition of ALGOL 60*. Tech. rep. 25.088. IBM Laboratory Vienna, Dec. 1968. URL: <http://homepages.cs.ncl.ac.uk/cliff.jones/publications/0CRd/Lau68.pdf>.
- [Lau71] P. E. Lauer. “Consistent Formal Theories of the Semantics of Programming Languages”. Printed as TR 25.121, IBM Lab. Vienna. PhD thesis. Queen’s University of Belfast, 1971.
- [Law05] Jonathan Lawrence. “Practical Applications of CSP and FDR to Software Design”. In: *Communicating Sequential Processes. The First 25 Years*. Ed. by Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders. Vol. 3525. Lecture Notes in Computer Science. Springer-Verlag, 2005, pp. 151–174. ISBN: 978-3-540-25813-1. DOI: 10.1007/b136154. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-25813-1>.
- [LB85] M. M. Lehman and L. A. Belady. *Program Evolution: Processes of Software Change*. APIC Studies in Data Processing 27. Academic Press, 1985.

- [Lee] Josh Lee. *Answer to "Why five dining philosophers?"*. URL: <https://softwareengineering.stackexchange.com/a/187921> (visited on 03/01/2013).
- [Lei10] K Rustan M Leino. "Dafny: An automatic program verifier for functional correctness". In: *International conference on logic for programming artificial intelligence and reasoning*. Springer, 2010, pp. 348–370.
- [Lep08] Jouni Leppäjärvi. "A pragmatic, historically oriented survey on the universality of synchronization primitives". MA thesis. University of Oulu, Dept. of Information Processing Science, 2008.
- [Lev72] K. N. Levitt. "The Application of Program-proving Techniques to the Verification of Synchronization Processes". In: *Proceedings of AFIPS Fall Joint Computer Conference*. AFIPS Press, 1972, pp. 33–47.
- [LFF12] Hongjin Liang, Xinyu Feng, and Ming Fu. "A rely-guarantee-based simulation for verifying concurrent program transformations". In: *Proceedings of the 39th annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '12. Philadelphia, PA, USA, 2012, pp. 455–468. ISBN: 978-1-4503-1083-3. DOI: 10.1145/2103656.2103711. URL: <http://doi.acm.org/10.1145/2103656.2103711>.
- [LG81] G. M. Levin and D. Gries. "A Proof Technique for Communicating Sequential Processes". In: *Acta Informatica* 15 (1981), pp. 281–302.
- [Lia14] Hongjin Liang. "Refinement Verification of Concurrent Programs and Its Applications". PhD thesis. USTC, China, 2014.
- [Loo59] B.J. Loopstra. "Input and output in the X-1 system". In: *Information processing: Proceedings of the International Conference on Information Processing*. Oldenbourg Verlag, 1959.
- [LPS81] Daniel Lehmann, Amir Pnueli, and Jonathan Stavi. "Impartiality, justice and fairness: The ethics of concurrent termination". In: *ICALP 1981*. Vol. 115. Lecture Notes in Computer Science. Springer-Verlag, 1981, pp. 264–277.
- [LPZ85] O. Lichtenstein, A. Pnueli, and L. Zuck. "The Glory of the Past". In: *Proceedings of the Workshop on Logics of Programs*. Ed. by R. Parikh. Vol. 193. Lecture Notes in Computer Science. Springer-Verlag, June 1985, pp. 196–218. DOI: https://doi.org/10.1007/3-540-15648-8_16.
- [LS84] Leslie Lamport and Fred B. Schneider. "The "Hoare Logic" of CSP, and All That". In: *ACM Trans. Program. Lang. Syst.* 6.2 (Apr. 1984), pp. 281–296. ISSN: 0164-0925. DOI: 10.1145/2993.357247. URL: <http://doi.acm.org/10.1145/2993.357247>.
- [LS89] Simon S Lam and A Udaya Shankar. "Refinement and projection of relational specifications". In: *Stepwise Refinement of Distributed Systems Models, Formalisms, Correctness*. REX 1989.

- Ed. by J. W. de Bakker, W.-P. de Roever, and G. Rozenberg. Vol. 430. Lecture Notes in Computer Science. Springer-Verlag, 1989, pp. 454–486.
- [Luc+68] P. Lucas et al. *Informal Introduction to the Abstract Syntax and Interpretation of PL/I*. Tech. rep. 25.083. IBM Laboratory Vienna, ULD Version II, June 1968.
- [Luc68] Peter Lucas. *Two constructive realisations of the block concept and their equivalence*. Tech. rep. TR 25.085. IBM Laboratory Vienna, June 1968.
- [LW69] Peter Lucas and Kurt Walk. “On the formal description of PL/I”. In: *Annual Review in Automatic Programming* 6 (1969), pp. 105–182.
- [Lyn+94] Nancy Lynch, Michael Merritt, William Weihl, and Alan Fekete. *Atomic Transactions*. MIT Press, 1994.
- [LZ74] B. Liskov and S. Zilles. *Programming with Abstract Data Types*. Computation Structures Group Memo 99. Massachusetts Institute of Technology, Mar. 1974.
- [Mac01] Donald MacKenzie. *Mechanizing Proof: Computing, Risk, and Trust*. MIT Press, 2001.
- [Man68a] Z. Manna. *Formalization of Properties of Programs*. Technical Memorandum AI-64. Stanford Artificial Intelligence Department, July 1968.
- [Man68b] Z. Manna. “Termination of Algorithms”. PhD thesis. Carnegie-Mellon University, Apr. 1968. URL: <https://apps.dtic.mil/dtic/tr/fulltext/u2/670558.pdf>.
- [Man69a] Z. Manna. *The Correctness of Non-Deterministic Programs*. Memo AI-95. Department of Computer Science, Stanford University, Aug. 1969.
- [Man69b] Z. Manna. “The Correctness of Programs”. In: *Journal of Computer and System Sciences* 3.2 (May 1969), pp. 119–127. DOI: [https://doi.org/10.1016/S0022-0000\(69\)80009-7](https://doi.org/10.1016/S0022-0000(69)80009-7).
- [Man69c] Zohar Manna. “Properties of programs and the first-order predicate calculus”. In: *Journal of the ACM* 16.2 (1969), pp. 244–255. DOI: <https://doi.org/10.1145/321510.321516>.
- [Man70] Zohar Manna. “The correctness of nondeterministic programs”. In: *Artificial Intelligence* 1.1-2 (1970), pp. 1–26. DOI: [https://doi.org/10.1016/0004-3702\(70\)90002-0](https://doi.org/10.1016/0004-3702(70)90002-0).
- [Man78] Zohar Manna. *Six Lectures on the Logic of Computer Programming*. Tech. rep. AIM-318. Stanford AI Lab, 1978.
- [Man80] Zohar Manna. *Lectures on the Logic of Computer Programming*. Vol. 31. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, 1980. ISBN: 1611970385, 9781611970388.
- [Man82] Zohar Manna. “Verification of sequential programs: Temporal axiomatization”. In: *Theoretical foundations of programming methodology*. Vol. 91. NATO Advanced Study Institute Series

- C – Mathematical and Physical Science. The Proceedings are in the "PRG" library - shelf mark 95F 31 BRO. Springer-Verlag, 1982, pp. 53–102. DOI: https://doi.org/10.1007/978-94-009-7893-5_6.
- [May21] David May. "CSP, occam and Inmos". In: *Theories of programming: the life and works of Tony Hoare*. Ed. by Cliff B. Jones and Jayadev Misra. ACM, 2021. Chap. 12. DOI: 10.1145/3477355.
- [May80] Kenneth O May. "Historiography: A perspective for computer scientists". In: *A History of Computing in the Twentieth Century*. Elsevier, 1980, pp. 11–18.
- [MC81] J. Misra and K. M. Chandy. "Proofs of Networks of Processes". In: *IEEE Transactions on Software Engineering* 7 (1981), pp. 417–426.
- [McC66] John McCarthy. "A formal description of a subset of ALGOL". In: *Formal Language Description Languages for Computer Programming*. North-Holland, 1966, pp. 1–12.
- [McC68a] John McCarthy. *Letter to A. P. Ershov*. Held in the Ershov archive (online). Mar. 1968. URL: <http://ershov.iis.nsk.su/en/node/779549>.
- [McC68b] John McCarthy. *Letter to C. C. Gotlieb*. John McCarthy Papers (SC0524) Department of Special Collections and University Archives, Stanford University Libraries, Stanford, Calif. Sept. 1968.
- [McT08] J Ellis McTaggart. "The unreality of time". In: *Mind* 17.68 (Oct. 1908), pp. 457–474.
- [Mey88] B. Meyer. *Object-oriented Software Construction*. Prentice Hall, 1988. ISBN: 0-13-629031-0.
- [Mey97] B. Meyer. *Object-oriented Software Construction*. Second. Prentice Hall, 1997. ISBN: 0-13-629031-0.
- [MF86] David A. Mundie and David A. Fisher. "Parallel Processing in Ada". In: *Computer (Long Beach, Calif.)* 19.8 (1986), pp. 20–25.
- [MH69] J. McCarthy and P. J. Hayes. "Some Philosophical Problems From the Standpoint of Artificial Intelligence". In: *Machine Intelligence 4*. Edinburgh University Press, 1969, pp. 463–502.
- [MHR80] N. Metropolis, J. Howlett, and Gian-Carlo Rota, eds. *A history of computing in the twentieth century*. Academic Press, 1980.
- [Mid90] C. A. Middelburg. "Syntax and Semantics of VVSL: A Language for Structured VDM Specifications". PhD thesis. PTT Research, Leidschendam, Department of Applied Computer Science, Sept. 1990.
- [Mid93] Cornelius A. Middelburg. *Logic and Specification: Extending VDM-SL for advanced formal specification*. Chapman and Hall, 1993.

- [Mil70] R. Milner. *A Formal Notion of Simulation Between Programs*. Memo 14. Hard copy. Department of Computer Science, University College of Swansea, Oct. 1970.
- [Mil71] R. Milner. *An Algebraic Definition of Simulation between Programs*. Tech. rep. CS-205. Hard copy. Computer Science Dept, Stanford University, Feb. 1971. DOI: <https://dl.acm.org/doi/book/10.5555/891902>.
- [Mil72] R. Milner. *Logic for Computable Functions Description of a Machine Implementation*. Tech. rep. STAN-CS-72-288. Hard copy. Computer Science Department, Stanford University, May 1972. DOI: <https://dl.acm.org/doi/book/10.5555/891954>. URL: purl.stanford.edu/tj064sd5929.
- [Mil73] R. Milner. “An Approach to the Semantics of Parallel Programs”. In: *Proceedings of the Convegno di Informatica Teorica*. Hard copy. 1973, pp. 283–302.
- [Mil75] Robin Milner. “Processes: a mathematical model of computing agents”. In: *Studies in Logic and the Foundations of Mathematics*. Vol. 80. Elsevier, 1975, pp. 157–173.
- [Mil77] George Johnstone Milne. “A mathematical model of concurrent computation”. PhD thesis. Edinburgh, 1977.
- [Mil78] Robin Milner. “Synthesis of communicating behaviour”. In: *Mathematical Foundations of Computer Science 1978*. Lecture Notes in Computer Science 64 (1978), pp. 71–83. DOI: https://doi.org/10.1007/3-540-08921-7_57.
- [Mil80] R. Milner. *A Calculus of Communicating Systems*. Vol. 92. Lecture Notes in Computer Science. Springer-Verlag, 1980. ISBN: 978-3-540-10235-9. DOI: <https://doi.org/10.1007/3-540-10235-3>.
- [Mil86] Robin Milner. “Process constructors and interpretations”. In: *IFIP Congress*. 1986, pp. 507–514.
- [Mil89] Robin Milner. *Communication and Concurrency*. Prentice Hall, Jan. 1989. ISBN: 978-0-13-114984-7. DOI: <https://dl.acm.org/doi/book/10.5555/534666>.
- [Mil93] Robin Milner. “Elements of Interaction”. In: *Communications of the ACM* 36.1 (1993), pp. 78–89. URL: <http://doi.acm.org/10.1145/151233.151240>.
- [Mil99] R. Milner. *Communicating and mobile systems: the pi calculus*. Cambridge University Press, 1999.
- [Mis94] Jayadev Misra. *A logic for concurrent programming: Safety and Progress*. Tech. rep. Technical report, The University of Texas at Austin, 1994.
- [MJ84] F. L. Morris and C. B. Jones. “An Early Program Proof by Alan Turing”. In: *Annals of the History of Computing* 6.2 (1984), pp. 139–143. DOI: 10.1109/MAHC.1984.10017.

- [MK05] Jeff Magee and Jeff Kramer. “Model-Based Design of Concurrent Programs”. In: *Communicating Sequential Processes. The First 25 Years*. Ed. by Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders. Vol. 3525. Lecture Notes in Computer Science. Springer-Verlag, 2005, pp. 211–219. ISBN: 978-3-540-25813-1. DOI: 10.1007/b136154. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-25813-1>.
- [MK06] Jeff Magee and Jeff Kramer. *Concurrency : state models & Java programs*. eng. 2nd. Wiley, 2006. ISBN: 9780470093559.
- [MK99] Jeff Magee and Jeff Kramer. *State Models and Java Programs*. Wiley, 1999.
- [MM79] George Milne and Robin Milner. “Concurrent processes and their syntax”. In: *Journal of the ACM* 26.2 (1979), pp. 302–321.
- [MM80] R. M. McKeag and A. M. Macnaghten. *On the Construction of Programs*. USA: Cambridge University Press, 1980.
- [MM83] Ben Moszkowski and Zohar Manna. “Reasoning in interval temporal logic”. In: *Workshop on Logic of Programs*. Springer, 1983, pp. 371–382.
- [MMN75] R. Milner, L. Morris, and M. Newey. “A Logic for Computable Functions with Reflexive and Polymorphic Types”. In: *Proceedings of Conference on Proving and Improving Programs*. IRIA-Laboria, 1975, pp. 371–394.
- [MN70] Zohar Manna and Steven Ness. “On the termination of Markov algorithms”. In: *Proceedings of the Third Hawaii International Conference on System Science*. 1970, pp. 789–792.
- [Mog88] Eugenio Moggi. *Computational Lambda-calculus and Monads*. Report ECS-LFCS-88-66. Edinburgh University LFCS, 1988.
- [Mon87] B. Q. Monahan. “A Type Model for VDM”. In: *VDM – A Formal Definition at Work*. Ed. by Dines Bjørner, C. B. Jones, M. Mac an Airchinnigh, and E. J. Neuhold. Vol. 252. Lecture Notes in Computer Science. Springer-Verlag, 1987, pp. 210–236. ISBN: 978-3-540-17654-1. DOI: 10.1007/3-540-17654-3. URL: <http://www.springer.com/computer/swe/book/978-3-540-17654-1>.
- [Moo19] J Strother Moore. “Milestones from the Pure Lisp theorem prover to ACL2”. In: *Formal Aspects of Computing* 31.6 (2019), pp. 699–732.
- [Mor88] Carroll Morgan. “The specification statement”. In: *ACM Trans. Program. Lang. Syst.* 10.3 (July 1988), pp. 403–419. ISSN: 0164-0925. DOI: 10.1145/44501.44503. URL: <http://doi.acm.org/10.1145/44501.44503>.
- [Mor90] Carroll Morgan. *Programming from Specifications*. Prentice Hall, 1990.

- [Mos21] Peter D Mosses. “Fundamental Constructs in Programming Languages”. In: *arXiv preprint arXiv:2107.10545* (2021). also in LNCS6415.
- [Mos83] Benjamin C Moszkowski. *Reasoning about digital circuits*. Tech. rep. STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1983.
- [Mos85] Ben Moszkowski. “Executing temporal logic programs. CONCURRENCY 1984”. In: *Seminar on Concurrency*. Ed. by Stephen D. Brookes, Andrew William Roscoe, and Glynn Winskel. Vol. 197. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1985, pp. 111–130. ISBN: 978-3-540-15670-3. DOI: 10.1007/3-540-15670-4_6. URL: http://dx.doi.org/10.1007/3-540-15670-4_6.
- [Mos86] B. C. Moszkowski. *Executing Temporal Logic Programs*. Cambridge University Press, 1986.
- [Mos92] Peter D. Mosses. *Action Semantics*. Cambridge Tracts in Theoretical Computer Science 26. Cambridge Press, 1992.
- [Mos94] Ben C Moszkowski. “Some very compositional temporal properties”. In: *Proceedings of the IFIP TC2/WG2. 1/WG2. 2/WG2. 3 Working Conference on Programming Concepts, Methods and Calculi*. North-Holland Publishing Co. 1994, pp. 307–326.
- [Mos96] Ben Moszkowski. “Using temporal fixpoints to compositionally reason about liveness”. In: *Proceedings of the BCS-FACS 7th Refinement Workshop 7*. 1996, pp. 1–28.
- [Mos97] Ben C Moszkowski. “Compositional reasoning using interval temporal logic and tempura. International Symposium, COMPOS’97 Bad Malente, Germany, September 8–12, 1997 Revised Lectures”. In: *Compositionality: The Significant Difference. International Symposium, COMPOS’97 Bad Malente, Germany, September 8–12, 1997 Revised Lectures*. Ed. by Willem-Paul de Roever, Hans Langmaack, and Amir Pnueli. Vol. 1536. Lecture Notes in Computer Science. Springer-Verlag, 1997, pp. 439–464. ISBN: 3540654933.
- [MP68] Z. Manna and A. Pnueli. “The Validity Problem of the 91-Function”. Rough draft. Aug. 1968. URL: <https://apps.dtic.mil/sti/citations/AD0785164>.
- [MP69] Zohar Manna and Amir Pnueli. “Formalization of properties of recursively defined functions”. In: *Proceedings of the first annual ACM symposium on Theory of computing*. STOC ’69. New York, NY, USA: Association for Computing Machinery, May 1969, pp. 201–210. DOI: <https://doi.org/10.1145/800169.805434>.
- [MP70] Zohar Manna and Amir Pnueli. “Formalization of properties of functional programs”. In: *Journal of the ACM* 17.3 (1970), pp. 555–569. DOI: <https://doi.org/10.1145/321592.321606>.

- [MP74] Zohar Manna and Amir Pnueli. “Axiomatic approach to total correctness of programs”. In: *Acta Informatica* 3.3 (Sept. 1974), pp. 243–263. DOI: <https://doi.org/10.1007/BF00288637>.
- [MP81a] Zohar Manna and Amir Pnueli. “Verification of Concurrent Programs: The Temporal Framework”. In: *The Correctness Problem in Computer Science*. Ed. by R. S. Boyer and J. S. Moore. New York: Academic Press, May 1981, pp. 215–273. DOI: 10.21236/ada106750.
- [MP81b] Zohar Manna and Amir Pnueli. “Verification of Concurrent Programs: Temporal Proof Principles”. In: *Proc. Workshop on Logics of Programs*. Ed. by Dexter Kozen. Vol. 131. Lecture Notes in Computer Science. Berlin: Springer-Verlag, June 1981, pp. 200–252. DOI: <https://dl.acm.org/doi/10.5555/648063.747433>.
- [MP82a] Z. Manna and A. Pnueli. “Verification of concurrent programs: a temporal proof system”. In: *Proceedings of the 4th School of Advanced Programming, Amsterdam, Holland*. Amsterdam, Holland, June 1982.
- [MP82b] Zohar Manna and Amir Pnueli. *Verification of concurrent programs: Proving eventualities by well-founded ranking*. Technical report STAN-CS-82-915. Stanford University Computer Science Dept, 1982. URL: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a132416.pdf>.
- [MP83] Zohar Manna and Amir Pnueli. *Verification of concurrent programs: A temporal proof system*. Tech. rep. STAN-CS-83-967. Department of Computer Science, Stanford University, 1983. DOI: <https://dl.acm.org/doi/book/10.5555/892296>.
- [MP87] Z. Manna and A. Pnueli. “Specification and Verification of Concurrent Programs by \forall -Automata”. In: *Conference Record of the 14th Annual ACM Symposium on Principles of Programming Languages Munich, West Germany 21–23 January*. ACM, 1987, pp. 1–12. DOI: <https://dl.acm.org/doi/pdf/10.1145/41625.41626>.
- [MP91] Zohar Manna and Amir Pnueli. “Completing the temporal picture”. In: *Theoretical Computer Science* 83.1 (June 1991), pp. 97–130. DOI: [https://doi.org/10.1016/0304-3975\(91\)90041-Y](https://doi.org/10.1016/0304-3975(91)90041-Y).
- [MP92] Zohar Manna and Amir Pnueli. *Temporal Logic of Reactive and Concurrent Systems*. New York: Springer-Verlag, 1992. ISBN: 978-0-387-97664-8. DOI: <https://dl.acm.org/doi/book/10.5555/128869>.
- [MP95] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer-Verlag, 1995. ISBN: 978-1-4612-4222-2. DOI: 10.1007/978-1-4612-4222-2.
- [MPW89] R. Milner, J. Parrow, and D. Walker. *A Calculus of Mobile Processes*. Tech. rep. (two parts) ECS-LFCS-89-85, 86. Labo-

- ratory for Foundations of Computer Science, Edinburgh University, 1989. URL: PartI:<https://www.lfcs.inf.ed.ac.uk/reports/89/ECS-LFCS-89-85/>, PartII:<https://www.lfcs.inf.ed.ac.uk/reports/89/ECS-LFCS-89-86/>.
- [MPW92] R. Milner, J. Parrow, and D. Walker. “A Calculus of Mobile Processes”. In: *Information and Computation* 100.1 (1992), pp. 1–77. DOI: PartI:[https://doi.org/10.1016/0890-5401\(92\)90008-4](https://doi.org/10.1016/0890-5401(92)90008-4), PartII:[https://doi.org/10.1016/0890-5401\(92\)90009-5](https://doi.org/10.1016/0890-5401(92)90009-5).
- [MS81] U Montanari and C Simonelli. *On distinguishing concurrency from nondeterminism*. Hard copy. Colleville sur mer, 1981.
- [MSS16] P. Müller, M. Schwerhoff, and A. J. Summers. “Viper: A Verification Infrastructure for Permission-Based Reasoning”. In: *Verification, Model Checking, and Abstract Interpretation (VMCAI)*. Ed. by B. Jobstmann and K. R. M. Leino. Vol. 9583. LNCS. Springer-Verlag, 2016, pp. 41–62. URL: https://doi.org/10.1007/978-3-662-49122-5_2.
- [MW71] Zohar Manna and Richard J Waldinger. “Toward automatic program synthesis”. In: *Communications of the ACM* 14.3 (Mar. 1971), pp. 151–165. DOI: <https://doi.org/10.1145/362566.362568>.
- [MW76] Zohar Manna and Richard Waldinger. *Is “sometime” sometimes better than “always”? : intermittent assertions in proving program correctness*. Memo AIM-281, STAN-CS-76-558. Stanford Artificial Intelligence Laboratory, June 1976.
- [MW77] Z. Manna and R. Waldinger. *Synthesis: Dreams → Programs*. Tech. rep. STAN-CS-77-630. Department of Computer Science, Stanford University, Nov. 1977.
- [MW78] Zohar Manna and Richard Waldinger. “Is “sometime” sometimes better than “always”? : intermittent assertions in proving program correctness”. In: *Communications of the ACM* 21.2 (1978), pp. 159–172. DOI: <https://doi.org/10.1145/359340.359353>.
- [MW84] Zohar Manna and Pierre Wolper. “Synthesis of communicating processes from temporal logic specifications”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 6.1 (1984), pp. 68–93. DOI: <https://doi.org/10.1145/357233.357237>.
- [Nau66] Peter Naur. “Proof of algorithms by general snapshots”. In: *BIT Numerical Mathematics* 6.4 (1966), pp. 310–316.
- [ND78] Kristen Nygaard and Ole-Johan Dahl. “The Development of the SIMULA Languages”. In: *History of Programming Languages*. Ed. by Richard L. Wexelblat. New York, NY, USA: Association for Computing Machinery, 1978, pp. 439–480. URL: <https://doi.org/10.1145/800025.1198392>.

- [NK13] Tobias Nipkow and Gerwin Klein. *Concrete Semantics. A Proof Assistant Approach*. Springer-Verlag, 2013.
- [NL16] Linus Nyman and Mikael Laakso. “Notes on the History of Fork and Join”. In: *IEEE Annals of the History of Computing* 38.3 (2016), pp. 84–87.
- [NN92] H. R. Nielson and F. Nielson. *Semantics with Applications: A Formal Introduction*. Wiley, 1992.
- [NPA14] David Nofre, Mark Priestley, and Gerard Alberts. “When Technology Became Language: The Origins of the Linguistic Conception of Computer Programming, 1950–1960”. In: *Technology and Culture* 55.1 (Jan. 2014), pp. 40–75.
- [OG76a] S. S. Owicki and D. Gries. “An axiomatic proof technique for parallel programs I”. In: *Acta Informatica* 6 (1976), pp. 319–340.
- [OG76b] Susan Owicki and David Gries. “Verifying properties of parallel programs: An axiomatic approach”. In: *Communications of the ACM* 19.5 (May 1976), pp. 279–285.
- [O’H07] P. W. O’Hearn. “Resources, Concurrency and Local Reasoning”. In: *Theoretical Computer Science* 375.1-3 (May 2007), pp. 271–307.
- [ØH07] Peter Øhrstrøm and Per Hasle. *Temporal logic: from ancient ideas to artificial intelligence*. Vol. 57. Studies in Linguistics and Philosophy. Springer Science & Business Media, 2007.
- [OL82] Susan Owicki and Leslie Lamport. “Proving Liveness Properties of Concurrent Programs”. In: *ACM Transaction on Programming Languages and Systems* 4, 3 (July 1982). Hard copy, pp. 455–495. DOI: <https://doi.org/10.1145/357172.357178>.
- [Old91] E. R. Olderog. *Nets, Terms and Formulas: Three Views of Concurrent Processes and their Relationship*. Vol. 23. Cambridge Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press, 1991. DOI: [10.1017/CB09780511526589](https://doi.org/10.1017/CB09780511526589).
- [Owi75] S. S. Owicki. “Axiomatic Proof Techniques for Parallel Programs”. Published as technical report 75-251. PhD thesis. Department of Computer Science, Cornell University, 1975.
- [Par10] Matthew Parkinson. “The Next 700 Separation Logics”. In: *Verified Software: Theories, Tools, Experiments*. Ed. by Gary Leavens, Peter O’Hearn, and Sriram Rajamani. Vol. 6217. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, pp. 169–182. ISBN: 978-3-642-15056-2. URL: http://dx.doi.org/10.1007/978-3-642-15057-9_12.
- [Par72] David L Parnas. “On the criteria to be used in decomposing systems into modules”. In: *Pioneers and Their Contributions to Software Engineering*. Springer-Verlag, 1972, pp. 479–498.

- [Par78] David L Parnas. “The non-problem of nested monitor calls”. In: *ACM SIGOPS Operating Systems Review* 12.1 (1978), pp. 12–18.
- [Pet62a] C. A. Petri. “Fundamentals of a Theory of Asynchronous Information Flow.” In: *IFIP Congress*. North-Holland, 1962, pp. 386–390. URL: <http://dblp.uni-trier.de/db/conf/ifip/ifip1962.html#Petri62>.
- [Pet62b] C. A. Petri. “Kommunikation mit Automaten”. PhD thesis. University of Darmstadt, 1962.
- [Pet76] C. A. Petri. *Nichtsequentielle Prozesse*. Tech. rep. ISF-76-6. Hard copy. GMD, Bonn, 1976.
- [Pet81] G.L. Peterson. “Myths about the Mutual Exclusion Problem”. In: *Information Processing Letters* 12.3 (1981), pp. 115–116.
- [Pic95] Andrew Pickering. *The Mangle of Practice: Time, Agency, and Science*. University of Chicago Press, 1995.
- [Pie09] Ken Pierce. “Enhancing the Useability of Rely-Guarantee Conditions for Atomicity Refinement”. PhD thesis. Newcastle University, 2009.
- [PJ01] Simon Peyton Jones. “Tackling the awkward squad: monadic input/output, concurrency, exceptions, and foreign-language calls in Haskell”. In: *NATO SCIENCE SERIES SUB SERIES III COMPUTER AND SYSTEMS SCIENCES* 180 (2001), pp. 47–96.
- [PJGF96] Simon Peyton Jones, Andrew Gordon, and Sigbjorn Finne. “Concurrent haskell”. In: *POPL*. Vol. 96. 1996, pp. 295–308.
- [Plo04a] Gordon D. Plotkin. “A structural approach to operational semantics”. In: *Journal of Logic and Algebraic Programming* 60–61 (2004), pp. 17–139. ISSN: 1567-8326. DOI: [doi:10.1016/j.jlap.2004.03.002](https://doi.org/10.1016/j.jlap.2004.03.002). URL: <http://www.elsevier.com/locate/jlap>.
- [Plo04b] Gordon D. Plotkin. “The origins of structural operational semantics”. In: *Journal of Logic and Algebraic Programming* 60–61 (2004), pp. 3–15. ISSN: 1567-8326. DOI: [doi:10.1016/j.jlap.2004.03.009](https://doi.org/10.1016/j.jlap.2004.03.009). URL: <http://www.elsevier.com/locate/jlap>.
- [Plo81] G. D. Plotkin. *A Structural Approach to Operational Semantics*. Tech. rep. DAIMI FN-19. Aarhus University, 1981.
- [Pnu77a] A. Pnueli. “The temporal logic of programs”. In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*. SFCS ’77. USA: IEEE Computer Society, 1977, pp. 46–57. DOI: [10.1109/SFCS.1977.32](https://doi.org/10.1109/SFCS.1977.32).
- [Pnu77b] A. Pnueli. *The Temporal Semantics of Concurrent Programs*. Tech. rep. Tel-Aviv University. University of Pennsylvania and Tel-Aviv University, 1977.
- [Pnu79] A. Pnueli. “The Temporal Semantics of Concurrent Programs”. In: *Semantics of Concurrent Computation*. Ed. by G. Kahn. Vol. 70. Lecture Notes in Computer Science. Heidelberg: Springer-

- Verlag, 1979, pp. 1–20. DOI: <https://doi.org/10.1007/BFb0022460>.
- [Pnu81] A. Pnueli. “The Temporal Semantics of Concurrent Programs”. In: *Theoretical Computer Science* 13.1 (1981), pp. 45–60. DOI: [https://doi.org/10.1016/0304-3975\(81\)90110-9](https://doi.org/10.1016/0304-3975(81)90110-9).
- [Pon19] Alban Ponse. *Interview with Alban Ponse*. Conducted by Gracia Redder and Wenchen Lai. Jan. 2019.
- [Pra65] Dag Prawitz. *Natural Deduction: a Proof-Theoretical Study*. Dover Publications, 1965.
- [Pra76] Vaughan R Pratt. “Semantical consideration on Floyd-Hoare logic”. In: *17th Annual Symposium on Foundations of Computer Science (SFCS 1976)*. IEEE. 1976, pp. 109–121.
- [Pre01] Leonor Prensa Nieto. “Verification of Parallel Programs with the Owicki-Gries and Rely-Guarantee Methods in Isabelle/HOL”. PhD thesis. Institut für Informatik der Technischen Universität München, 2001.
- [Pri11] Mark Priestley. *A Science of Operations: Machines, Logic and the Invention of Programming*. History of Computing. Springer-Verlag, London, 2011.
- [Pri20] Mark Priestley. “Flow Diagrams, Assertions, and Formal Methods”. In: *HFM 2019 - History of Formal Methods Workshop. In: Formal Methods. FM 2019 International Workshops. Porto, Portugal, October 7–11, 2019, Revised Selected Papers, Part II*. Ed. by Troy K. Astarte. Lecture Notes in Computer Science 12233. Springer-Verlag, 2020, pp. 15–34. ISBN: 978-3-030-54997-8. DOI: 10.1007/978-3-030-54997-8.
- [Pri57] A. N. Prior. *Time and Modality: Being the John Locke Lectures for 1955-6 delivered the University of Oxford*. Oxford University Press, 1957.
- [Pri67] A. N. Prior. *Past, Present and Future*. Oxford University Press, 1967.
- [PS19] Damien Pous and Davide Sangiorgi. “Bisimulation and coinduction enhancements: A historical perspective”. In: *Formal Aspects of Computing* 31.6 (2019), pp. 733–749.
- [PST00] G. Plotkin, C. Stirling, and M. Tofte, eds. *Proof, Language, and Interaction: Essays in Honour of Robin Milner*. MIT Press, 2000. ISBN: 978-0-262-16188-6.
- [QS82] Jean-Pierre Queille and Joseph Sifakis. “Specification and verification of concurrent systems in CESAR”. In: *International Symposium on programming*. Springer-Verlag. 1982, pp. 337–351.
- [Ray86] M. Raynal. *Algorithms for Mutual Exclusion*. Cambridge, Mass.: MIT Press, 1986.
- [RB81] W.C. Rounds and S.D. Brookes. “Possible Futures, Acceptances, Refusals, and Communicating Processes”. In: *22nd Annual Symposium on Foundations of Computer Science, Nashville, Ten-*

- nessee, USA, 28-30 October 1981. IEEE Computer Society, 1981, pp. 140–149. DOI: 10.1109/SFCS.1981.36. URL: <https://doi.org/10.1109/SFCS.1981.36>.
- [Rei13] Wolfgang Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer-Verlag, 2013.
- [Rei85] Wolfgang Reisig. *Petri Nets: an Introduction*. Vol. 4. Monographs in Theoretical Computer Science. Springer Science & Business Media, 1985.
- [Res66] Nicholas Rescher. *The Logic of Commands*. Routledge & Kegan Paul and Dover Publications, 1966.
- [Rey02a] John C. Reynolds. “Separation Logic: A Logic for Shared Mutable Data Structures”. In: *Proceedings of 17th LICS*. IEEE, 2002, pp. 55–74.
- [Rey02b] John H Reynolds. “Linda arouses a sleeping barber”. In: *Proceedings of the Winter Simulation Conference*. Vol. 2. IEEE, 2002, pp. 1804–1808.
- [Rey03] John. C. Reynolds. *Separation Logic*. Slides from three mini-course lectures at BRICS. Nov. 2003.
- [Rey04] John. C. Reynolds. *Towards a Grainless Semantics for Shared Variable Concurrency*. Slides from talk at POPL’04. Jan. 2004.
- [RH07] Barbara G. Ryder and Brent Hailpern, eds. *HOPL III: Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages*. San Diego, California, 2007. ISBN: 978-1-59593-766-7.
- [RH80] Olivier Roubine and Jean-Claude Heliard. “Parallel Processing in Ada”. In: *On the Construction of Programs*. USA: Cambridge University Press, 1980.
- [RLP98] Willem-Paul de Roever, Hans Langmaack, and Amir Pnueli, eds. *Compositionality: The Significant Difference. International Symposium, COMPOS’97 Bad Malente, Germany, September 8–12, 1997 Revised Lectures*. Vol. 1536. Lecture Notes in Computer Science. Springer Science & Business Media, 1998. ISBN: 3540654933.
- [Rob65] J. A. Robinson. “A Machine-Oriented Logic Based on the Resolution Principle”. In: *Journal of the ACM* 12 (1 1965), pp. 23–41.
- [Roc55] Nathaniel Rochester. “The computer and its peripheral equipment”. In: *Proceedings of the Eastern Joint AIEE-IRE Computer Conference: Computers in business and industrial systems*. 1955, pp. 64–69.
- [Roe+01] Willem-Paul de Roever et al. *Concurrency Verification: Introduction to Compositional and Noncompositional Methods*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001. ISBN: 9780521169325.

- [Ros05] AW Roscoe. “Seeing beyond divergence”. In: *Communicating Sequential Processes. The First 25 Years*. Ed. by Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders. Vol. 3525. Lecture Notes in Computer Science. Springer-Verlag, 2005, pp. 15–35. ISBN: 978-3-540-25813-1. DOI: 10.1007/b136154. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-25813-1>.
- [Ros10a] A. W. Roscoe. *Understanding Concurrent Systems*. Texts in Computer Science. Springer-Verlag, 2010. ISBN: 978-1-84882-257-3. DOI: 10.1007/978-1-84882-258-0.
- [Ros10b] Andrew William Roscoe. *Understanding concurrent systems*. Springer Science & Business Media, 2010.
- [Ros74] B. K. Rosen. *Correctness of Parallel Programs: the Church-Rosser Approach*. IBM Research Report RC5107. Yorktown Heights, N.Y.: T. J. Watson Research Center, 1974.
- [Ros82] Andrew William Roscoe. “A mathematical theory of communicating processes”. PhD thesis. University of Oxford, 1982. URL: <https://ora.ox.ac.uk/objects/uuid:ed4e2289-f817-4851-868a-00e2c4f48db5>.
- [Ros97] Bill Roscoe. *The theory and practice of concurrency*. Prentice Hall Series in Computer Science. USA: Prentice Hall PTR, 1997. ISBN: 0-13-674409-5.
- [RR64] Brian Randell and Lawford J. Russell. *Algol 60 Implementation*. Academic Press, Inc., 1964.
- [RU71] N. Rescher and A. Urquart. *Temporal Logic*. New York: Springer-Verlag, 1971.
- [Rya05] Peter Ryan. “Shedding Light on Haunted Corners of Information Security”. In: *Communicating Sequential Processes. The First 25 Years*. Ed. by Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders. Vol. 3525. Lecture Notes in Computer Science. Springer-Verlag, 2005, p. 264. ISBN: 978-3-540-25813-1. DOI: 10.1007/b136154. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-25813-1>.
- [San09] Davide Sangiorgi. “On the origins of bisimulation and coinduction”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 31.4 (2009), p. 15.
- [Sat75] Edwin H. Satterthwaite. “Source Language Debugging Tools”. PhD thesis. Stanford University, 1975.
- [SC85] A Prasad Sistla and Edmund M Clarke. “The complexity of propositional linear temporal logics”. In: *Journal of the ACM (JACM)* 32.3 (1985), pp. 733–749.
- [Sch+14] Gerhard Schellhorn, Bogdan Tofan, Gidon Ernst, Jörg Pfähler, and Wolfgang Reif. “RGITL: A temporal logic framework for compositional reasoning about interleaved programs”. In: *An-*

- nals of Mathematics and Artificial Intelligence* 71.1-3 (2014), pp. 131–174.
- [Sch75] Jerald Schwarz. *Event Based Reasoning – A system for proving correct termination of programs*. Research report 12. Edinburgh University, Department of A.I., 1975.
- [Sch97] Fred B. Schneider. *On Concurrent Programming*. Texts in Computer Science. Springer-Verlag, 1997. ISBN: 978-0-387-94942-0. DOI: 10.1007/978-1-4612-1830-2.
- [Sco13] Michael L. Scott. *Shared-Memory Synchronization*. Morgan & Claypool Publishers, 2013. ISBN: 978-1-60845-956-8.
- [Sco72] Dana S. Scott. *Letter to Robin Milner*. Christopher Strachey Collection, Bodleian Library, Oxford. Box 276, C.241. Feb. 1972.
- [SD05] Steve Schneider and Rob Delicata. “Verifying Security Protocols: An Application of CSP”. In: *Communicating Sequential Processes. The First 25 Years*. Ed. by Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders. Vol. 3525. Lecture Notes in Computer Science. Springer-Verlag, 2005, pp. 243–263. ISBN: 978-3-540-25813-1. DOI: 10.1007/b136154. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-25813-1>.
- [Sim90] H. R. Simpson. “Four-slot fully asynchronous communication mechanism”. In: *Computers and Digital Techniques, IEE Proceedings E* 137.1 (1990), pp. 17–30.
- [Sit74] R. L. Sites. “Proving that Computer Programs Terminate Cleanly”. Printed as STAN-CS-74-418. PhD thesis. Computer Science Department, Stanford University, 1974.
- [Sou84] Neelam Soundararajan. “Axiomatic semantics of communicating sequential processes”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 6.4 (1984), pp. 647–662.
- [Spi89] J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall International, 1989.
- [Sta96] Mark G Staskauskas. “An experience in the formal verification of industrial software”. In: *Communications of the ACM* 39.12es (1996), 256–es.
- [Sti86] C. Stirling. “A Compositional Reformulation of Owicki-Gries’s Partial Correctness Logic for a Concurrent While Language”. In: *Automata, Languages and Programming: ICALP’86*. Ed. by L. Kott. Vol. 226. Lecture Notes in Computer Science. Springer-Verlag, 1986, pp. 407–415.
- [Sti88] C. Stirling. “A Generalisation of Owicki-Gries’s Hoare Logic for a Concurrent While Language”. In: *Theoretical Computer Science* 58.1–3 (1988), pp. 347–359.

- [Sto77] Joseph E. Stoy. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*. Cambridge, MA, USA: MIT Press, 1977. ISBN: 0262191474.
- [Sto80] Joseph E. Stoy. “Foundations of denotational semantics”. In: *Abstract Software Specifications: 1979 Copenhagen Winter School January 22 – February 2, 1979 Proceedings*. Ed. by Dines Bjørner. Springer Berlin Heidelberg, 1980, pp. 43–99. URL: http://dx.doi.org/10.1007/3-540-10007-5_35.
- [Stø90] K. Stølen. “Development of Parallel Programs on Shared Data-Structures”. Published as technical report UMCS-91-1-1, now available on-line <https://breibakk.no/kst/PhD-thesis.htm>. PhD thesis. Manchester University, 1990.
- [Str00] Christopher Strachey. “Fundamental concepts in programming languages”. In: *Higher-order and symbolic computation* 13.1-2 (2000), pp. 11–49.
- [Str21] Georg Struth. “Trimming the Hedges: An Algebra to Tame Concurrency”. In: *Theories of programming: the life and works of Tony Hoare*. Ed. by Cliff B. Jones and Jayadev Misra. ACM, 2021. Chap. 14. DOI: 10.1145/3477355.
- [Str66] Christopher Strachey. “System analysis and programming”. In: *Scientific American* 215 (1966), pp. 112–124.
- [Suf21] Bernard Sufrin. “Teaching at Belfast and Oxford”. In: *Theories of programming: the life and works of Tony Hoare*. Ed. by Cliff B. Jones and Jayadev Misra. ACM, 2021. Chap. 10. DOI: 10.1145/3477355.
- [SW01] Davide Sangiorgi and David Walker. *The π -Calculus: A Theory of Mobile Processes*. Cambridge, United Kingdom: Cambridge University Press, 2001. ISBN: 0-521-78177-9. URL: https://assets.cambridge.org/97805215/43279/frontmatter/9780521543279_frontmatter.pdf.
- [Szy88] Boleslaw K Szymanski. “A simple solution to Lamport’s concurrent programming problem with linear wait”. In: *Proceedings of the 2nd International Conference on Supercomputing*. 1988, pp. 621–626.
- [TBJ15] Arnold Thackray, David C. Brock, and Rachel Jones. *Moore’s Law: The Life of Gordon Moore, Silicon Valley’s Quiet Revolutionary*. Basic Books, 2015.
- [Tea06] Team SPARK. *SPARK Examiner: The SPARK Ravenscar Profile*. Tech report. 2006.
- [Tof+13] Bogdan Tofan, Gerhard Schellhorn, Gidon Ernst, Jörg Pfahler, and Wolfgang Reif. “Compositional verification of a lock-free stack with RGITL”. In: *Electronic Communications of the EASST Automated Verification of Critical Systems 2013* (2013).
- [Top79] R. W. Topor. “The correctness of the Schorr-Waite list marking algorithm”. In: *Acta Informatica* 11.3 (Sept. 1979), pp. 211–

221. ISSN: 1432-0525. DOI: 10.1007/BF00289067. URL: <https://doi.org/10.1007/BF00289067>.
- [Toy47] Arnold Toynbee. *A study of history*. Abridged by D. C. Somervell. New York: Oxford University Press, 1947.
- [Tur49] A. M. Turing. “Checking a Large Routine”. In: *Report of a Conference on High Speed Automatic Calculating Machines*. University Mathematical Laboratory, Cambridge, June 1949, pp. 67–69.
- [TZ] JV Tucker and JI Zucker. “Origins of our Theory of Computation on Abstract Data Types at the Mathematical Centre, Amsterdam, 1979-80”. In: *Liber Amicorum: Jaco de Bakker*. Ed. by F de Boer, M van der Heijden, P Klint, and J Rutten, pp. 197–221.
- [TZ88] J.V. Tucker and J.I. Zucker. *Program Correctness over Abstract Data Types, with Error-State Semantics*. CWI Monograph. North-Holland, 1988. ISBN: 0-444-70340-3. URL: <https://ir.cwi.nl/pub/22755>.
- [Vaf07] Viktor Vafeiadis. “Modular fine-grained concurrency verification”. PhD thesis. University of Cambridge, 2007.
- [Vaf11] Viktor Vafeiadis. “Concurrent Separation Logic and Operational Semantics”. In: *Electronic Notes in Theoretical Computer Science* 276 (2011). Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVII), pp. 335–351. ISSN: 1571-0661. DOI: <https://doi.org/10.1016/j.entcs.2011.09.029>. URL: <https://www.sciencedirect.com/science/article/pii/S1571066111001204>.
- [Vin90] Walter G. Vincenti. *What Engineers Know and How They Know It*. John Hopkins University Press, 1990.
- [VP07] Viktor Vafeiadis and Matthew Parkinson. “A Marriage of Rely/Guarantee and Separation Logic”. In: *CONCUR 2007 – Concurrency Theory*. Ed. by Luís Caires and Vasco Vasconcelos. Vol. 4703. Lecture Notes in Computer Science. Springer-Verlag, 2007, pp. 256–271. ISBN: 978-3-540-74406-1. URL: http://dx.doi.org/10.1007/978-3-540-74407-8_18.
- [Wal02] Kurt Walk. “Roots of Computing in Austria: Contributions of the IBM Vienna Laboratory and Changes of Paradigms and Priorities in Information Technology”. In: *Human Choice and Computers*. Springer-Verlag, 2002, pp. 77–87.
- [Wal+67] K. Walk et al. *Abstract Syntax and Interpretation of PL/I (Draft for Version 2)*. Technical Report TR 25.082. IBM Laboratory Vienna, Dec. 1967.
- [Wal67] Kurt Walk. *Minutes of the 1st meeting of IFIP WG 2.2 on Formal Language Description Languages*. Kept in the van Wijngaarden archive. Held in Porto Conte, Alghero, Sardinia. Chaired by T. B. Steel. Sept. 1967.

- [Wal68] Kurt Walk. *Minutes of the 2nd meeting of IFIP WG 2.2 on Formal Language Description Languages*. Held in Vedbaek-Copenhagen, Denmark. Chaired by T. B. Steel. July 1968.
- [Wal69] Kurt Walk. *Minutes of the 4th meeting of IFIP WG 2.2 on Formal Language Description Languages*. Held in Colchester, Essex, England. Chaired by T. B. Steel. Sept. 1969.
- [WD71] Arne Wang and Ole-Johan Dahl. "Coroutine sequencing in a block structured environment". In: *BIT Numerical Mathematics* 11.4 (1971), pp. 425–449.
- [Wei+06] Gerhard Weikum, Cliff B. Jones, David Lomet, and Alexander Romanovsky. "Executive Summary – Atomicity: A Unifying Concept in Computer Science". In: *Atomicity: A Unifying Concept in Computer Science*. Dagstuhl Seminar Proceedings 06121. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2006. URL: <http://drops.dagstuhl.de/opus/volltexte/2006/835>.
- [Wei71] Gerald M. Weinberg. *The Psychology of Computer Programming*. Van Norstrand, 1971.
- [Wex81] Richard L. Wexelblat, ed. *History of Programming Languages*. Academic Press, 1981.
- [WH66] N. Wirth and C.A.R. Hoare. "A Contribution to the Development of ALGOL". In: *Communications of the ACM* 9.6 (June 1966), pp. 413–432.
- [Whi76] James E White. "A high-level framework for network-based resource sharing". In: *Proceedings of the June 7-10, 1976, national computer conference and exposition*. 1976, pp. 561–570.
- [Whi96] William A. Whitaker. "ADA—the Project: The DoD High Order Language Working Group". In: *History of Programming Languages—II*. New York, NY, USA: Association for Computing Machinery, 1996, pp. 173–232. URL: <https://doi.org/10.1145/234286.1057816>.
- [Wij66] Adriaan van Wijngaarden. "Numerical analysis as an independent science". In: *BIT Numerical Mathematics* 6.1 (1966), pp. 66–81.
- [Wir07] Niklaus Wirth. "Modula-2 and Oberon". In: *Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages*. HOPL III. San Diego, California, 2007. URL: <https://doi.org/10.1145/1238844.1238847>.
- [Wir69] Niklaus Wirth. *The Programming Language Pascal and its Design Criteria*. Presented at the NATO Conference on Software Engineering, Rome, 27. – 31.10.69. July 1969.
- [Wir71a] Niklaus Wirth. "Program development by stepwise refinement". In: *cacm* (1971), pp. 545–569.

- [Wir71b] Niklaus Wirth. “The programming language Pascal”. In: *Acta informatica* 1.1 (1971), pp. 35–63.
- [Wir77a] Niklaus Wirth. “Modula: A language for modular multiprogramming”. In: *Software: Practice and Experience* 7.1 (1977), pp. 1–35.
- [Wir77b] Niklaus Wirth. “Toward a discipline of real-time programming”. In: *Communications of the ACM* 20.8 (1977), pp. 577–583.
- [Wir80] Niklaus Wirth. *MODULA-2*. Tech. rep. ETH Institut für Informatik, Mar. 1980.
- [Wol83] Pierre Wolper. “Temporal logic can be more expressive”. In: *Information and control* 56.1-2 (1983), pp. 72–99.
- [Woo78] M. Woodger. “A History of IFIP WG 2.3: Programming Methodology”. In: *Programming Methodology: A Collection of Articles by Members of IFIP WG 2.3*. Ed. by David Gries. Monographs in Computer Science. Berlin, Heidelberg: Springer-Verlag, 1978, pp. 1–6. ISBN: 0387903291.
- [Wri65] Georg Henrik von Wright. “And next”. In: *Acta Philosophica Fennica* 18 (1965), pp. 293–304.
- [Wri66] Georg Henrik von Wright. “And then”. In: *Commentationes Physico-mathematicae* 32.7 (1966).
- [Wri68] Georg Henrik von Wright. “Always”. In: *Theoria* 34 (1968), pp. 208–221.
- [WV01] Gerhard Weikum and Gottfried Vossen. *Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. ISBN: 1-55860-508-8.
- [WWF87] David A Watt, Brian A Wichmann, and William Findlay. *Ada language and methodology*. Prentice Hall International (UK) Ltd., 1987.
- [XRH97] Q. Xu, W.-P. de Roever, and J. He. “The rely-guarantee method for verifying shared variable concurrent programs”. In: *Formal Aspects of Computing* 9.2 (1997), pp. 149–174.
- [Xu92] Qiwen Xu. “A theory of state-based parallel programming”. PhD thesis. Oxford University Computing Laboratory, 1992.
- [Yon90] Akinori Yonezawa, ed. *ABCL: An Object-Oriented Concurrent System*. ISBN 0-262-24029-7. MIT Press, 1990.
- [Zil74] S. N. Zilles. *Abstract Specifications for Data Types*. Tech. rep. 11. M.I.T. Progress Report, 1974, pp. 52–58.
- [Zuc19] Lenore Zuck. *Amir Pnueli - A.M. Turing Award Laureate*. 2019. URL: https://amturing.acm.org/award_winners/pnueli_4725172.cfm.
- [Zwi88] J. Zwiers. “Compositionality, Concurrency and Partial Correctness: Proof theories for networks of processes, and their relationship”. Available as lncs 321, Springer-Verlag. PhD thesis. Technical University Eindhoven, 1988.