

# Main Publications: Cliff B Jones<sup>1</sup>

Each section is listed in reverse date order.

**Books (including edited Proceedings)** [1]–[27]

**Articles** [28]–[163]

**Conference talks** (where no publication cited) [164]–[200]

**Technical reports** (where not preprints of articles<sup>2</sup>) [201]–[257]<sup>3</sup>

## References

- [1] Cliff B. Jones and John L. Lloyd, editors. *Dependable and Historic Computing*. Springer, 2011.
- [2] Jens Bendisposto, Cliff Jones, Michael Leuschel, and Alexander Romanovsky. Proceedings of the 11th international workshop on automated verification of critical systems. Technical Report CS-TR-1272, School of Computing Science, University of Newcastle, September 2011.
- [3] Cliff B. Jones, A. W. Roscoe, and Kenneth R. Wood, editors. *Reflections on the work of C.A.R. Hoare*. Springer, 2010.
- [4] Michael J. Butler, Cliff B. Jones, Alexander Romanovsky, and Elena Troubitsyna, editors. *Methods, Models and Tools for Fault-Tolerance*, volume 5454 of *Lecture Notes in Computer Science*. Springer, 2009.
- [5] Cliff B. Jones, Zhiming Liu, and Jim Woodcock, editors. *Theoretical Aspects of Computing – ICTAC 2007*. Number 4711 in *Lecture Notes in Computer Science*. Springer, 2007.
- [6] Cliff B. Jones, Zhiming Liu, and Jim Woodcock, editors. *Formal Methods and Hybrid Real-Time Systems: Essays in Honour of Dines Bjørner and Zhou Chaochen on the Occasion of Their 70th Birthdays*, volume 4700 of *Lecture Notes in Computer Science*. Springer, 2007.
- [7] M. Butler, C. Jones, A. Romanovsky, and E. Troubitsyna, editors. *Rigorous Development of Complex Fault-Tolerant Systems*, volume 4157 of *Lecture Notes in Computer Science*. Springer, 2006.
- [8] D. Besnard, C. Gacek, and C. B. Jones, editors. *Structure for Dependability: Computer-Based Systems from an Interdisciplinary Perspective*. Springer, 2006.

---

<sup>1</sup>Updated: 2012-02-13

<sup>2</sup>In industry, even significant items were not necessarily published externally: e.g. [239] was a complete definition of ALGOL 60 using “functional semantics”; [231] introduced the concept of “implementation bias”; [244] introduced the “exit” idea; and [235] was the key output of the denotational semantics definition of PL/I (which marks the start of VDM as a language definition method).

<sup>3</sup>[230] is my D.Phil. Thesis.

- [9] Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders, editors. *Communicating Sequential Processes: the First 25 years*, volume 3525 of *Lecture Notes in Computer Science*. Springer, 2005.
- [10] M. Butler, C. B. Jones, A. Romanovsky, and E. Troubitsyna. Proceedings of the workshop on rigorous engineering of fault-tolerant systems (REFT 2005). Technical Report CS-TR-915, ISSN 1368-1060, School of Computing Science, University of Newcastle, June 2005.
- [11] John Dobson, Cliff Jones, Rob Procter, Mark Rouncefield, and Robin Williams, editors. *Interdisciplinary approaches to the design of dependable computer systems*. ACM, Proceedings of the 2002 ACM symposium on Applied Computing, Madrid, 2002.
- [12] John Fitzgerald, Cliff B. Jones, and Peter Lucas, editors. *FME'97: Industrial Applications and Strengthened Foundations of Formal Methods*, volume 1313 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.
- [13] Burkhard Freitag, Cliff B. Jones, Christian Lengauer, and Hans-Jörg Schek, editors. *Object Orientation with Parallelism and Persistence*. Kluwer Academic Publishers, 1996.
- [14] C. B. Jones. *VDM: Une méthode rigoureuse pour le développement du logiciel*. Masson, Paris, 1993. Translation of [17].
- [15] C. B. Jones, R. C. Shaw, and T. Denvir, editors. *5th Refinement Workshop*. Springer-Verlag, 1992.
- [16] C. B. Jones, K. D. Jones, P. A. Lindsay, and R. Moore. *mural: A Formal Development Support System*. Springer-Verlag, 1991.
- [17] C. B. Jones. *Systematic Software Development using VDM*. Prentice Hall International, second edition, 1990.
- [18] C. B. Jones and R. C. F. Shaw, editors. *Case Studies in Systematic Software Development*. Prentice Hall International, 1990.
- [19] M. Broy and C. B. Jones, editors. *Programming Concepts and Methods*. North-Holland, April 1990.
- [20] C. A. R. Hoare and C. B. Jones. *Essays in Computing Science*. Prentice Hall International, Hemel Hempstead, UK, 1989.
- [21] D. Bjørner, C. B. Jones, M. Mac an Airchinnigh, and E. J. Neuhold, editors. *VDM – A Formal Definition at Work*, volume 252 of *Lecture Notes in Computer Science*. Springer-Verlag, 1987.
- [22] C. B. Jones. *Systematic Software Development Using VDM*. Prentice Hall International, 1986.
- [23] H. Bekič and C. B. Jones, editors. *Programming Languages and Their Definition: Selected Papers of H. Bekič*, volume 177 of *Lecture Notes in Computer Science*. Springer-Verlag, 1984.

- [24] C. B. Jones. *Konstruowanie oprogramowania metodą systematyczną*. Wydawnictwa Naukowo-Techniczne, Warsaw, 1984. Translation of [26].
- [25] Dines Bjørner and Cliff B. Jones, editors. *Formal Specification and Software Development*. Prentice Hall International, 1982.
- [26] C. B. Jones. *Software Development: A Rigorous Approach*. Prentice Hall International, Englewood Cliffs, N.J., USA, 1980.
- [27] D. Bjørner and C. B. Jones, editors. *The Vienna Development Method: The Meta-Language*, volume 61 of *Lecture Notes in Computer Science*. Springer-Verlag, 1978.
- [28] Cliff B. Jones. Turing’s ”Checking a Large Routine”. In S. Barry Cooper and Jan van Leeuwen, editors, *Alan Turing — His Work and Impact*, page ?? Elsevier, to appear April 2012.
- [29] Cliff B. Jones and Matthew J. Lovert. Semantic models for a logic of partial functions. *International Journal of Software and Informatics*, 5(1–2, Part I):55–76, 2011.
- [30] Cliff B. Jones and Ken G. Pierce. Elucidating concurrent algorithms via layers of abstraction and reification. *Formal Aspects of Computing*, 23(3):289–306, 2011.
- [31] Leo Freitas and Cliff B. Jones. Learning from an expert’s proof: AI4FM. In Tom Ball, Lenore Zuck, and Natarajan Shankar, editors, *UV10 (Usable Verification)*, November 2010.
- [32] C. B. Jones. The role of auxiliary variables in the formal development of concurrent programs. In Cliff B. Jones, A. W. Roscoe, and Kenneth Wood, editors, *Reflections on the work of C.A.R. Hoare*, chapter 8, pages 167–188. Springer, 2010.
- [33] C.B. Jones and A.W. Roscoe. Insight, inspiration and collaboration. In Cliff B. Jones, A. W. Roscoe, and Kenneth Wood, editors, *Reflections on the work of C.A.R. Hoare*, chapter 1, pages 1–32. Springer, 2010.
- [34] Alan Bundy, Gudmund Grov, and Cliff B. Jones. An outline of a proposed system that learns from experts how to discharge proof obligations automatically. In Jean-Raymond Abrial, Michael Butler, Rajeev Joshi, Elena Troubitsyna, and Jim C. P. Woodcock, editors, *Dagstuhl 09381: Refinement Based Methods for the Construction of Dependable Systems*, pages 38–42, 2009.
- [35] Cliff B. Jones. Abstraction is all we’ve got: auxiliary variables considered harmful. In Jean-Raymond Abrial, Michael Butler, Rajeev Joshi, Elena Troubitsyna, and Jim C. P. Woodcock, editors, *Dagstuhl 09381: Refinement Based Methods for the Construction of Dependable Systems*, pages 94–98, 2009.
- [36] Gudmund Grov, Alan Bundy, Cliff B. Jones, and Andrew Ireland. The ai4fm approach for automation within formal methods. In *Grand Challenges in Computing Research 2010*. UK Computing Research Committee, April 2010.

- [37] Cliff B. Jones. From problem frames to HJJ (and its known unknowns). In Bashar Nuseibeh and Pamela Zave, editors, *Software Requirements and Design: The Work of Michael Jackson*, chapter 16, pages 357–372. Good Friends Publishing Company, 2010.
- [38] Alan Bundy, Gudmund Grov, and Cliff B. Jones. Learning from experts to aid the automation of proof search. In Liam O’Reilly and Markus Roggenbach, editors, *AVoCS’09 – PreProceedings of the Ninth International Workshop on Automated Verification of Critical Systems*, Technical Report of Computer Science CSR-2-2009, pages 229–232. Swansea University, 2009.
- [39] J. S. Fitzgerald and C. B. Jones. The connection between two ways of reasoning about partial functions. *IPL*, 107(3–4):128–132, 2008.
- [40] J. S. Fitzgerald, J. W. Bryans, D. Greathead, C. B. Jones, and R. Payne. Animation-based validation of a formal model of dynamic virtual organisations. In P. Boca, J. Bowen, and P.G. Larsen, editors, *FACS 2007 Christmas Workshop: Formal Methods in Industry*, BCS London, UK, December 2008. BCS.
- [41] Cliff B. Jones. Some interdisciplinary observations about getting the “right” specification. In B. Meyer and J. P. C. Woodcock, editors, *Verified Software: Theories, Tools, Experiments*, number 4171 in Lecture Notes in Computer Science, pages 64–69. Springer, 2008.
- [42] Cliff B. Jones and Ken G. Pierce. Splitting atoms with rely/guarantee conditions coupled with data reification. In *ABZ2008*, number 5238 in Lecture Notes in Computer Science, pages 360–377. Springer, 2008.
- [43] John R. D. Hughes and C. B. Jones. Reasoning about programs via operational semantics: Requirements for a support system. *Automated Software Engineering*, 15(3–4):299–312, 2008.
- [44] Cliff B. Jones. Reflections on, and predictions for, support systems for the development of programs. In *Proceedings ASE 2008*, pages 7–8. IEEE/ACM, 2008.
- [45] Cliff B. Jones, Ian J. Hayes, and Michael A. Jackson. Deriving specifications for systems that are connected to the physical world. In Cliff B. Jones, Zhiming Liu, and Jim Woodcock, editors, *Formal Methods and Hybrid Real-Time Systems: Essays in Honour of Dines Bjørner and Zhou Chaochen on the Occasion of Their 70th Birthdays*, volume 4700 of *Lecture Notes in Computer Science*, pages 364–390. Springer Verlag, 2007.
- [46] Cliff B. Jones. Understanding programming language concepts via operational semantics. In Chris George, Zhiming Liu, and Jim Woodcock, editors, *Domain Modeling and the Duration Calculus*, volume 4710 of *Lecture Notes in Computer Science*, pages 177–235. Springer, 2007.
- [47] Cliff Jones. Discussion on the ideal of program correctness by Tony Hoare. *The Computer Journal*, 50(3):261–268, 2007.

- [48] Cliff B. Jones and Ken G. Pierce. What can the  $\pi$ -calculus tell us about the mondex purse system? In *ICECCS '07: Proceedings of the 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007)*, pages 300–306. IEEE, IEEE Computer Society, 2007.
- [49] Cliff B. Jones. Vdm "post-ludium". In D. Bjørner and M. Henson, editors, *Logics of Specification Languages*, pages 614–616. Springer, 2007.
- [50] J. W. Coleman and C. B. Jones. A structural proof of the soundness of rely/guarantee rules. *Journal of Logic and Computation*, 17(4):807–841, 2007.
- [51] C. B. Jones. Splitting atoms safely. *Theoretical Computer Science*, 375(1–3):109–119, 2007.
- [52] Cliff B. Jones and Brian Randell. The role of structure: a dependability perspective. In D. Besnard, C. Gacek, and C. B. Jones, editors, *Structure for Dependability: Computer-Based Systems from an Interdisciplinary Perspective*, chapter 1, pages 3–15. Springer, 2006.
- [53] Cliff B. Jones. The DIRC project as the context of this book. In *Trust in Technology: a Socio-Technical Perspective*, chapter 10, pages 217–221. Springer, 2006.
- [54] Cliff Jones, Peter O’Hearn, and Jim Woodcock. Verified software: a grand challenge. *IEEE Computer*, 39(4):93–95, 2006.
- [55] Cliff B. Jones. Reasoning about partial functions in the formal development of programs. In *Proceedings of AVoCS’05*, volume 145, pages 3–25. Elsevier, Electronic Notes in Theoretical Computer Science, 2006.
- [56] T. P. Clement and C. B. Jones. Model-oriented specifications. *FACS FACTS*, 2006-2:39–53, 2006.
- [57] C. B. Jones, D. Lomet, A. Romanovsky, and G. Weikum. The atomic manifesto: a story in four quarks. *ACM SIGMOD Record*, 34(1):63–69, 2005.
- [58] Cliff B. Jones. Specification before satisfaction: the case for research into obtaining the right specification. In Helen Treharne, Steve King, Martin Henson, and Steve Schneider, editors, *ZB-2005*, volume 3455 of *Lecture Notes in Computer Science*, pages 1–5. Springer Verlag, 2005.
- [59] D. Greathead and C. B. Jones. Dirc: an interdisciplinary attack on problems of dependability. In *EDCC-5, Budapest, Supplementary Volume*, pages 11–14, April 2005.
- [60] Joey Coleman, Cliff Jones, Ian Oliver, Alexander Romanovsky, and Elena Troubitsyna. RODIN (rigorous open development environment for complex systems). In *EDCC-5, Budapest, Supplementary Volume*, pages 23–26, April 2005.

- [61] C. B. Jones. Sequencing operations and creating objects. In *Proceedings Tenth IEEE International Conference on Engineering of Complex Computer Systems*, pages 33–35. IEEE Computer Society, 2005.
- [62] C. B. Jones, D. Lomet, A. Romanovsky, and G. Weikum. The atomic manifesto. *Journal of Universal Computer Science*, 11(5):636–650, 2005.
- [63] J. I. Burton and C. B. Jones. Atomicity in system design and execution. *Journal of Universal Computer Science*, 11(5):634–635, 2005.
- [64] J. I. Burton and C. B. Jones. Investigating atomicity and observability. *Journal of Universal Computer Science*, 11(5):661–686, 2005.
- [65] Joey W. Coleman and Cliff B. Jones. Examples of how to determine the specifications of control systems. In M. Butler, C. Jones, A. Romanovsky, and E. Troubitsyna, editors, *Proceedings of the Workshop on Rigorous Engineering of Fault-Tolerant Systems (REFT 2005)*, number CS-TR-915 in Technical Report Series, pages 65–73. University of Newcastle Upon Tyne, June 2005. see [www.cs.ncl.ac.uk/research/pubs/trs/papers/915.pdf](http://www.cs.ncl.ac.uk/research/pubs/trs/papers/915.pdf).
- [66] C. B. Jones. Panelist position statement: reasoning about the design of programs. *Royal Soc, Phil Trans R Soc A*, 363(1835):2395–2396, 2005.
- [67] C. B. Jones. An approach to splitting atoms safely: Extended abstract. *Electronic Notes in Theoretical Computer Science, MFPS XXI, 21st Annual Conference of Mathematical Foundations of Programming Semantics*, 155:43–60, 2006.
- [68] Tony Hoare, Cliff Jones, and Brian Randell. Extending the horizons of DSE. In *Grand Challenges*. UKCRC, 2004.
- [69] Cliff Jones and Brian Randell. Dependable pervasive systems. In Robin Mansell and Brian S. Collins, editors, *Trust and Crime in Information Societies*, chapter 3, pages 59–90. Edward Elgar, 2005. also visible at <http://www.foresight.gov.uk>.
- [70] Cliff Jones, Panos Periorellis, Alexander Romanovsky, and Ian Welch. Structured handling of on-line interface upgrades in integrating dependable systems of systems. In N. Guelfi, E. Astesiano, and G. Reggio, editors, *FIDGI 2002*, volume 2604 of *Lecture Notes in Computer Science*, pages 73–86. Springer Verlag, 2003.
- [71] Cliff B Jones. A formal basis for some dependability notions. In Bernhard K. Aichernig and Tom Maibaum, editors, *Formal Methods at the Crossroads: from Panacea to Foundational Support*, volume 2757 of *Lecture Notes in Computer Science*, pages 191–206. Springer Verlag, 2003.
- [72] Cliff B. Jones. The early search for tractable ways of reasoning about programs. *IEEE, Annals of the History of Computing*, 25(2):26–49, 2003.
- [73] Cliff B. Jones. Operational semantics: concepts and their expression. *Information Processing Letters*, 88(1-2):27–32, 2003.

- [74] Ian Hayes, Michael Jackson, and Cliff Jones. Determining the specification of a control system from that of its environment. In Keijiro Araki, Stefani Gnesi, and Dino Mandrioli, editors, *FME 2003: Formal Methods*, volume 2805 of *Lecture Notes in Computer Science*, pages 154–169. Springer Verlag, 2003.
- [75] C. B. Jones. Wanted: a compositional approach to concurrency. In Annabelle McIver and Carroll Morgan, editors, *Programming Methodology*, pages 5–15. Springer Verlag, 2003.
- [76] A. T. Lawrie and C. B. Jones. Goal-diversity in the design of dependable computer-based systems. In (in) *Proceedings of the Open Source Software Development Workshop, CS-TR-812*, 2002.
- [77] Cliff Jones, Alexander Romanovsky, and Ian Welch. A structured approach to handling on-line interface upgrades. In *COMPSAC 2002*, pages 1000–1005. IEEE, 2002.
- [78] C. B. Jones. The transition from VDL to VDM. *Journal of Universal Computer Science*, 7(8):631–640, 2001.
- [79] B. Randell, R. Stroud, and C. Jones. DSoS (Dependable Systems of Systems). In *Proceedings of the 2001 International Conference on Dependable Systems and Networks (DSN-2001), Gothenburg, Sweden, 1-4 July 2001*, pages D28–D31. IEEE Computer Society Press, 2001.
- [80] Pierre Collette and Cliff B. Jones. Enhancing the tractability of rely/guarantee specifications in the development of interfering operations. In Gordon Plotkin, Colin Stirling, and Mads Tofte, editors, *Proof, Language and Interaction*, chapter 10, pages 277–307. MIT Press, 2000.
- [81] C. B. Jones. Compositionality, interference and concurrency. In Jim Davies, Bill Roscoe, and Jim Woodcock, editors, *Milennial Perspectives in Computer Science*, pages 175–186. Macmillian Press, 2000.
- [82] C. B. Jones. Thinking tools for the future of computing science. In Reinhard Wilhelm, editor, *Informatics – 10 Years Back, 10 Years forward*, volume 2000 of *Lecture Notes in Computer Science*, pages 112–130. Springer-Verlag, 2000.
- [83] C. B. Jones. Dependability of computer-based systems. In *XIV Simposia Brasileiro de Engenharia de Software*, pages 16–20, 2000.
- [84] I. J. Hayes and C. B. Jones. Specifications are not (necessarily) executable. In *High-Integrity System Specification and Design*, FACIT, pages 563–581. Springer-Verlag, 1999.
- [85] C. B. Jones. Scientific decisions which characterize VDM. In *FM’99 – Formal Methods*, volume 1708 of *Lecture Notes in Computer Science*, pages 28–47. Springer-Verlag, 1999.
- [86] John Fitzgerald and Cliff Jones. A tracking system. In J. C. Bicarregui, editor, *Proof in VDM: Case Studies*, FACIT, pages 1–30. Springer-Verlag, 1998.

- [87] C. B. Jones. Some mistakes I have made and what I have learned from them. In *Fundamental Approaches to Software Engineering*, volume 1382 of *Lecture Notes in Computer Science*, pages 7–20. Springer-Verlag, 1998.
- [88] C. B. Jones. Whither formal methods? a plea to investigate new applications. In Michael G Hinchey and Shaoying Liu, editors, *Formal Engineering Methods*, page 5. IEEE Computer Society, 1997.
- [89] C. B. Jones. Accommodating interference in the formal design of concurrent object-based programs. *Formal Methods in System Design*, 8(2):105–122, March 1996.
- [90] Edmund M. Clarke, Jeannette M. Wing, Cliff B. Jones, et al. Formal methods: State of the art and future directions. *ACM, Computing Surveys*, 28(4):626–643, December 1996.
- [91] C. B. Jones. Some practical problems and their influence on semantics. In *ESOP'96*, volume 1058 of *Lecture Notes in Computer Science*, pages 1–17. Springer-Verlag, 1996.
- [92] J. R. Gurd and C. B. Jones. The global-yet-personal information system. In Ian Wand and Robin Milner, editors, *Computing Tomorrow*, pages 127–157. Cambridge University Press, 1996.
- [93] C. B. Jones, D. N. Jackson, and J. Wing. Formal methods light. *IEEE, Computer*, 29(4):20–22, 1996.
- [94] C. B. Jones. TANSTAAFL (with partial functions). In William Farmer, Manfred Kerber, and Michael Kohlhase, editors, *Proceedings of the CADE-13 Workshop on the Mechanization Of Partial Functions*, pages 53–64, 1996.
- [95] C. B. Jones. Describing the semantics of concurrent object-oriented languages. In William Farmer, Manfred Kerber, and Michael Kohlhase, editors, *Proceedings of the 3rd International Workshop on Foundations of Object-Oriented Languages (FOOL-3 with LICS'96)*, pages 24–25, 1996.
- [96] Steve J. Hodges and Cliff B. Jones. Non-interference properties of a concurrent object-based language: Proofs based on an operational semantics. In Burkhard Freitag, Cliff B. Jones, Christian Lengauer, and Hans-Jörg Schek, editors, *Object Orientation with Parallelism and Persistence*, pages 1–22. Kluwer Academic Publishers, 1996.
- [97] C. B. Jones. Granularity and the development of concurrent programs. In *Proceedings MFPS'95*, volume 1 of *Electronic Notes in Theoretical Computer Science*, pages 302–306. Elsevier, 1995.
- [98] C. B. Jones. Fixing the semantics of some concurrent object-oriented concepts. In *Proceedings MFPS'95*, volume 1 of *Electronic Notes in Theoretical Computer Science*, pages 307–312. Elsevier, 1995.
- [99] C.B. Jones. Partial functions and logics: A warning. *Information Processing Letters*, 54(2):65–67, 1995.

- [100] C. B. Jones. Process algebra arguments about an object-based design notation. In A. W. Roscoe, editor, *A Classical Mind*, chapter 14, pages 231–246. Prentice-Hall, 1994.
- [101] C.B. Jones and C.A. Middelburg. A typed logic of partial functions reconstructed classically. *Acta Informatica*, 31(5):399–430, 1994.
- [102] C. B. Jones. Constraining interference in an object-based design method. In M-C. Gaudel and J-P. Jouannaud, editors, *TAPSOFT'93: Theory and Practice of Software Development*, volume 668 of *Lecture Notes in Computer Science*, pages 136–150. Springer-Verlag, 1993.
- [103] C. B. Jones. Reasoning about interference in an object-based design method. In *FME'93: Industrial-Strength Formal Methods*, volume 670 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 1993.
- [104] I. J. Hayes, C. B. Jones, and J. E. Nicholls. Understanding the differences between VDM and Z. *ACM Software Engineering News*, 19(3):75–81, July 1994.
- [105] C. B. Jones. A pi-calculus semantics for an object-based design notation. In *CONCUR'93: 4th International Conference on Concurrency Theory*, volume 715 of *Lecture Notes in Computer Science*, pages 158–172. Springer-Verlag, 1993.
- [106] C. B. Jones. An object-based design method for concurrent programs. In S. Meldal and M. Haverdaen, editors, *4th Nordic Workshop on Program Correctness*, pages 27–52. University of Bergen, 1993. Report No. 78, ISSN 0333-3590.
- [107] J. H. Cheng and C. B. Jones. On the usability of logics which handle partial functions. In C. Morgan and J. C. P. Woodcock, editors, *3rd Refinement Workshop*, pages 51–69. Springer-Verlag, 1991.
- [108] C. B. Jones. Interference revisited. In J. E. Nicholls, editor, *Z User Workshop, Workshops in Computing*, pages 58–73. Springer-Verlag, 1991.
- [109] C. B. Jones. Does the O-O community need formal methods? In J. Bézivin and B. Meyer, editors, *TOOLS 4: Technology of Object-Oriented Languages and Systems*, pages 15–22. Prentice-Hall, 1991.
- [110] C. B. Jones. Can computer programs be trusted? In Nigel Calder, editor, *Scientific Euorope*, pages 304–309. Foundation Scientific Europe, 1991.
- [111] C. B. Jones. Interference resumed. In P. Bailes, editor, *Engineering Safe Software*, pages 31–56. Australian Computer Society, 1991.
- [112] C. B. Jones. A small language definition. In Jones and Shaw [18], chapter 9, pages 235–256.
- [113] J.S. Fitzgerald and C. B. Jones. Modularizing the formal description of a database system. In D. Bjørner, C. A. R. Hoare, and H. Langmaack, editors,

- VDM'90: VDM and Z – Formal Methods in Software Development*, volume 428 of *Lecture Notes in Computer Science*, pages 189–210. Springer-Verlag, 1990.
- [114] C. B. Jones. Consequences. In *Beauty is Our Business*, chapter 25. Springer-Verlag, 1990.
- [115] C. B. Jones. Data reification. In J. A. McDermid, editor, *The Theory and Practice of Refinement*, pages 79–89. Butterworths, 1989.
- [116] I. J. Hayes and C. B. Jones. Specifications are not (necessarily) executable. *IEE, Software Engineering Journal*, 4(6):330–338, November 1989.
- [117] C. B. Jones and P. A. Lindsay. A support system for formal reasoning: Requirements and status. In R. Bloomfield, L. Marshall, and R. Jones, editors, *VDM'88: VDM – The Way Ahead*, volume 328 of *Lecture Notes in Computer Science*, pages 139–152. Springer-Verlag, 1988.
- [118] C. B. Jones and R. Moore. Muffin: A user interface design experiment for a theorem proving assistant. In R. Bloomfield, L. Marshall, and R. Jones, editors, *VDM'88: VDM – The Way Ahead*, volume 328 of *Lecture Notes in Computer Science*, pages 337–375. Springer-Verlag, 1988.
- [119] C. B. Jones. Theorem proving and software engineering. *IEE, Software Engineering Journal*, 3(1):2, 1988.
- [120] C. B. Jones. VDM proof obligations and their justification. In Bjørner et al. [21], pages 260–286.
- [121] C. B. Jones. Program specification and verification in VDM. In M. Broy, editor, *Logic of Programming and Calculi of Discrete Design*, volume 36 of *NATO ASI Series F: Computer and Systems Sciences*, pages 149–184. Springer-Verlag, 1987.
- [122] C. B. Jones. Software development based on formal methods. In A. N. Habermann and U. Montanari, editors, *System Development and Ada*, volume 275 of *Lecture Notes in Computer Science*, pages 153–172. Springer-Verlag, 1987.
- [123] C. B. Jones. Proof obligations for data reification. In O. G. Folberth and C. Hackl, editors, *Der Informationsbegriff in Technik und Wissenschaft*, pages 77–96. Oldenbourg Verlag, 1986. Festschrift–Prof. Karl E. Ganzhorn.
- [124] C. B. Jones. Systematic program development. In J. W. de Bakker, M. Hazewinkel, and J. K. Lenstra, editors, *Mathematics and Computer Science*, pages 19–50. North-Holland, 1986.
- [125] C. B. Jones. Specifications and programs. In *Współczesne Kierunki Rozwoju Informatyki*, pages 33–48, 1986.
- [126] C. B. Jones. Specyfikacje a programy I. *Informatyka*, 10:1–4, 1987.
- [127] C. B. Jones. Specyfikacje a programy II. *Informatyka*, 11:2–6, 1987.

- [128] C. B. Jones. The role of proof obligations in software design. In *Formal Methods and Software Development*, volume 186 of *Lecture Notes in Computer Science*, pages 27–41. Springer-Verlag, 1985.
- [129] I. D. Cottam, C. B. Jones, T. Nipkow, A. C. Wills, M. I. Wolczko, and A. Yaghi. Project support environments for formal methods. In J. McDermid, editor, *Integrated Project Support Environments*, chapter 3. Peter Peregrinus Ltd., 1985.
- [130] C. B. Jones. Systematic program development. In N. Gehani and A. D. McGettrick, editors, *Software Specification Techniques*, pages 89–110. Addison-Wesley, 1985. revised version of [124].
- [131] C. B. Jones. Specification, verification and testing in software development. In T. Anderson, editor, *Software Requirements Specification and Testing*, pages 1–13. Blackwell Scientific Publications, 1985.
- [132] F. L. Morris and C. B. Jones. An early program proof by Alan Turing. *Annals of the History of Computing*, 6(2):139–143, 1984.
- [133] H. Barringer, J.H. Cheng, and C. B. Jones. A logic covering undefinedness in program proofs. *Acta Informatica*, 21(3):251–269, 1984.
- [134] C. B. Jones. Specification and verification. *IEEE, Transactions of Software Engineering*, SE-10(2):126–127, 1984.
- [135] C. B. Jones. Specification and design of (parallel) programs. In *Proceedings of IFIP’83*, pages 321–332. North-Holland, 1983.
- [136] C. B. Jones. Tentative steps toward a development method for interfering programs. *Transactions on Programming Languages and System*, 5(4):596–619, 1983.
- [137] Cliff B. Jones. The meta-language. In Bjørner and Jones [25], chapter 2, pages 25–46.
- [138] Cliff B. Jones. Modelling concepts of programming languages. In Bjørner and Jones [25], chapter 4, pages 85–124.
- [139] Cliff B. Jones. More on exception mechanisms. In Bjørner and Jones [25], chapter 5, pages 125–140.
- [140] Wolfgang Henhapl and Cliff B. Jones. ALGOL 60. In Bjørner and Jones [25], chapter 6, pages 141–174.
- [141] Cliff B. Jones. Compiler design. In Bjørner and Jones [25], chapter 8, pages 253–270.
- [142] Elizabeth Fielding and Cliff B. Jones. Program design by data refinement. In Bjørner and Jones [25], chapter 10, pages 323–352.
- [143] C. B. Jones. Formal methods in software development. In *Proceedings of State of the Art Review 1981*, pages 107–113. Infotech, 1981.

- [144] C. B. Jones. Towards more formal specifications. In C. Floyd and H. Kopetz, editors, *Software Engineering – Entwurf und Spezifikation*, pages 19–45. Teubner Verlag, 1981.
- [145] C. B. Jones. Specification as a design base. In *Trends in Information Processing*, volume 123 of *Lecture Notes in Computer Science*, pages 103–105. Springer-Verlag, 1981.
- [146] C. B. Jones. Models of programming language concepts. In *Abstract Software Specifications*, volume 86 of *Lecture Notes in Computer Science*, pages 100–143. Springer-Verlag, 1980.
- [147] C. B. Jones. The role of formal specifications in software development. In *State of the Art Report: Life-Cycle Management*, volume 8, pages 117–133. Infotech, 1980.
- [148] C. B. Jones. Constructing a theory of a data structure as an aid to program development. *Acta Informatica*, 11(2):119–137, 1979.
- [149] C. B. Jones. The Vienna Development Method: Examples of compiler development. In M. Amirchahy and D. Neel, editors, *Le Point sur la Compilation*, pages 89–114. IRIA-SEFI, 1979.
- [150] Cliff B. Jones. The meta-language: A reference manual. In Bjørner and Jones [27], pages 218–277.
- [151] Cliff B. Jones. Denotational semantics of goto: An exit formulation and its relation to continuations. In Bjørner and Jones [27], pages 278–304.
- [152] Wolfgang Henhagl and Cliff B. Jones. A formal definition of ALGOL 60 as described in the 1975 modified report. In Bjørner and Jones [27], pages 305–336.
- [153] C. B. Jones. Structured design and coding: Theory versus practice. *Informatie*, 19(6):311–319, June 1977.
- [154] C. B. Jones. Program specification and formal development. In E. Morlet and D. Ribbens, editors, *International Computing Symposium 1977*, pages 537–553. North-Holland, 1977.
- [155] C. B. Jones. Some requirements for specification languages. In *Programming Technology Symposium*, pages 13–18. IBM, 1976. RA 77.
- [156] C. B. Jones. Formal definition in program development. In *Programming Methodology*, volume 23 of *Lecture Notes in Computer Science*, pages 387–443. Springer-Verlag, 1975.
- [157] K. V. Hanford and C. B. Jones. Dynamic syntax: A concept for the definition of the syntax of programming languages. In *Annual Review in Automatic Programming*, volume 7, pages 115–140. Pergamon, 1973.

- [158] C. B. Jones. Formal development of correct algorithms: an example based on Earley’s recogniser. In *SIGPLAN Notices, Volume 7 Number 1*, pages 150–169. ACM, January 1972.
- [159] C. B. Jones and P. Lucas. Proving correctness of implementation techniques. In E. Engeler, editor, *A Symposium on Algorithmic Languages*, volume 188 of *Lecture Notes in Mathematics*, pages 178–211. Springer-Verlag, 1971.
- [160] W. Henhagl and C. B. Jones. A run-time mechanism for referencing variables. *Information Processing Letters*, 1(1):14–16, 1971.
- [161] S. N. Glassover, K. V. Hanford, and C. B. Jones. The syntax machine. In *Programming Symposium, Atlantic City, New Jersey*, pages 179–186. IBM, 1968.
- [162] K. V. Hanford and C. B. Jones. An approach to context dependency. In *Programming Symposium, Swampscott, Mass.*, pages 199–212. IBM, 1967. TR 00.1600 (Vol. II).
- [163] C. B. Jones. A special purpose compiler. *BCS Computer Journal*, 8(2):109–112, July 1965.
- [164] Cliff B. Jones. Mechanising LPF, December 2011. invited talk at HB60.
- [165] Cliff B. Jones. AI4FM – how to say why – in proofs, April 2011. invited talk at BCTCS.
- [166] Cliff B. Jones, Gudmund Grov, and Alan Bundy. Ideas for a high-level proof strategy language, July 2010. AFM Workshop at FLoC.
- [167] Cliff B. Jones. Abstractions before proofs, July 2010. invited talk at VERIFY Workshop at FLoC.
- [168] Cliff B. Jones. Rely/guarantee thinking and its role in splitting atoms safely, June 2009. PSY Workshop at CAV, Grenoble.
- [169] C. B. Jones. Which system are we looking at: a dependability perspective, November 2008. EAJ/RAEng Symposium, British Embassy, Tokyo.
- [170] Cliff Jones. Reflections on, and predictions for, support systems for the development of programs, September 2008. Keynote at ASE-08, L’Aquila, see [44] for extended abstract.
- [171] Cliff Jones. Developing programs by “splitting atoms”, October 2008. FMCO, Sophia Antipolis.
- [172] Cliff Jones. “splitting atoms” with (a bunch of ideas including) rely/guarantee conditions, data reification, . . . , September 2008. Keynote Talk at VSR/ABZ, see [42] for proceedings.
- [173] Cliff Jones. Dependability of computer-based systems: why interdisciplinarity is essential, September 2006. Keynote talk, SafeComp-06, Gdansk.

- [174] C. B. Jones. (Avoiding, describing and understanding) interference, May 2005. MFPS XXI.
- [175] Cliff Jones. The case for research into obtaining the right specification, June 2005. Presented at IEEE's ICECCS-2005, Shanghai.
- [176] Cliff Jones. Determining the specification of systems involving humans, July 2005. Presented at DSVIS-05.
- [177] Cliff Jones. Technical challenges for verification in design, July 2005. Presented at GC-6 Workshop, FM-05.
- [178] Cliff Jones. Technical challenges for formal methods, September 2005. Presented at MPI Kaiserslautern.
- [179] Cliff Jones. What can we do (technically) to get 'the right specification'?, October 2005. IFIP TC2 Working Conference, VSTTE, ETH Zurich, see [41] for proceedings.
- [180] Cliff Jones. Operational semantics: concepts and their expression, April 2003. ETAPS-WMT 2003, Warsaw.
- [181] Cliff Jones. How can design abstractions be used in testing?, September 2003. SoftTest, York.
- [182] C. B. Jones. Whither formal methods?, July 1997. Irish Workshop on Formal Methods, Dublin.
- [183] C. B. Jones. Describing the semantics concurrent object-oriented languages, July 1996. FOOL-3 (LICS) Workshop, Rutgers University.
- [184] C. B. Jones. Lessons from the development of a concurrent program: formality as a guide to intuition, May 1995. FMTA, Poland.
- [185] C. B. Jones. Some practical problems and their influence on semantics, July 1995. Newton Institute in mathematical Sciences, Cambridge.
- [186] C. B. Jones. What inhibits the use of formal methods?, April 1992. Mathematical Foundations of Programming Semantics, Oxford.
- [187] C. B. Jones. Support tools for reification, January 1990. talk at: Third Refinement Workshop, Hursley.
- [188] C. B. Jones. Computer-aided formal reasoning for software design, March 1989. talk at: TAPSOFT'89, Barcelona.
- [189] C. B. Jones. Recent developments in VDM, March 1987. talk at: VDM'87 Symposium.
- [190] C. B. Jones. VDM's notions of data type and data reification, April 1987. talk at: YoP Texas.

- [191] C. B. Jones. Software development based on formal methods; IPSE 2.5, May 1986. Two talks at: Innovative Software Factories and Ada, Capri.
- [192] C. B. Jones. Data reification; operation decomposition; a logic for partial functions; support environments for formal methods, July–August 1986. Four talks at Summer School, Marktoberdorf, Germany.
- [193] C. B. Jones. Computing science, mathematics, and the Alvey programme, September 1986. talk at annual meeting of British Association for the Advancement of Science, Bristol.
- [194] C. B. Jones. Specifications and programs, 6 November 1986. talks at: Współczesne Kierunki Rozwoju Informatyki, Mrągowo, Poland.
- [195] C. B. Jones. Rigorous design: Some experiments in top-down design, March 1983. BCS-FACS/SERC Conference on Program Specification and Verification, University of York, (Proceedings not published).
- [196] I. D. Cottam, C. B. Jones, T. Nipkow, and A. C. Wills. Mule: A support system for formal specification and rigorous software development, March 1983. BCS-FACS/SERC Conference on Program Specification and Verification, University of York, (Proceedings not published).
- [197] C. B. Jones. The rigorous method, September 1982. Three lectures at the Joint International Seminar on the Teaching of Computer Science – Formal Specification, Newcastle.
- [198] D. Bjørner and C. B. Jones. The Vienna Development Method: Software abstraction principles and software architecture, January–February 1979. Copenhagen Winter School on ‘Abstract Software Specification’.
- [199] C. B. Jones. Abstract data types, August 1979. Software Institute, UCSC, Santa Cruz, California.
- [200] C. B. Jones. Abstract data types, September 1979. International Seminar on Teaching Computer Science, University of Newcastle.
- [201] Jim Hall, C. B. Jones, R. Velykiene, et al. A semantic analysis of logics that cope with partial functions. Technical report, ITRC, January 2012.
- [202] C. B. Jones, M. J. Lovert, and L. J. Steggles. A semantic analysis of logics that cope with partial functions. Technical Report CS-TR-1310, Newcastle University, January 2012.
- [203] Manuel Mazzara, Cliff Jones, and Alexei Iliasov. Lessons from DEPLOYment. Technical Report CS-TR-1309, School of Computing Science, University of Newcastle, January 2012.
- [204] Roberta Velykiene and Cliff B. Jones. A fast track analysis of ict constraints on evolving physical infrastructure. Technical Report CS-TR-1282, School of Computing Science, University of Newcastle, September 2011.

- [205] Ian J. Hayes, Alan Burns, Brijesh Dongol, and Cliff B. Jones. Comparing models of nondeterministic expression evaluation. Technical Report CS-TR-1273, School of Computing Science, University of Newcastle, September 2011.
- [206] Cliff B. Jones, Gudmund Grov, and Alan Bundy. Ideas for a high-level proof strategy language. Technical Report EDI-INF-RR-1377, Informatics, Edinburgh University, 2010.
- [207] J. W. Bryans, J. S. Fitzgerald, D. Greathead, C. B. Jones, and R. J. Payne. A dynamic coalitions workbench: Final report. Technical Report CS-TR-1091, School of Computing Science, Newcastle University, April 2008.
- [208] J. S. Fitzgerald and C. B. Jones. The connection between two ways of reasoning about partial functions. Technical Report CS-TR-1044, School of Computing Science, Newcastle University, August 2007.
- [209] Nigel Gilbert, Cliff Jones, et al. Dilemmas of privacy and surveillance: Challenges of technological change. Technical report, Royal Academy of Engineering, March 2007.
- [210] C. B. Jones. Rodin: Intermediate report on methodology. Technical Report CS-TR-990, School of Computing Science, University of Newcastle, November 2006.
- [211] J. W. Coleman and C. B. Jones. Guaranteeing the soundness of rely/guarantee rules. Technical Report CS-TR-955, School of Computing Science, University of Newcastle, March 2006.
- [212] J. W. Bryans, J. S. Fitzgerald, C. B. Jones, and I. Mozolevsky. Formal modelling of dynamic coalitions, with an application in chemical engineering. In T. Margaria, A. Philippou, and B. Steffen, editors, *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006)*, pages 91–98. IEEE, 2006.
- [213] Gary T. Leavens, Jean-Raymond Abrial, Don Batory, Michael Butler, Alessandro Coglio, Kathi Fisler, Eric Hehner, Cliff Jones, Dale Miller, Simon Peyton-Jones, Murali Sitaraman, Douglas R. Smith, and Aaron Stump. Roadmap for enhanced languages and methods to aid verification. Technical Report 06-21, Iowa State University, Department of Computer Science, Ames, IA, July 2006.
- [214] Joey W. Coleman, Nigel P. Jefferson, and Cliff B. Jones. Comments on several years of teaching of modelling programming language concepts. In Raymond T. Boute and Jose N. Oliveira, editors, *Formal Methods in the Teaching Lab*, 2006.
- [215] J. Coleman, N. Jefferson, and C. B. Jones. Black tie optional: Modelling programming language concepts. Technical Report CS-TR-844, School of Computing Science, Newcastle University, May 2004. Submitted to Ghent Teaching Conference.
- [216] M.-C. Gaudel, V. Issarny, C. B. Jones, H. Kopetz, E. Marsden, N. Moffat, M. Paulitsch, D. Powell, B. Randell, A. Romanovsky, R.J. Stroud, and F. Taini.

- Final version of DSoS conceptual model. Technical Report CS-TR-782, School of Computing Science, Newcastle University, 2002.
- [217] A. Romanovsky, J. Arlat, J.-C. Fabre, V. Issarny, C. B. Jones, N. Levy, E. Marsden, P. Periorellis, M. Rodriguez, F. Tartanoglu, and I. Welch. Further results on architectures and dependability mechanisms for dependable SoSs. Technical Report CS-TR-779, School of Computing Science, Newcastle University, 2002.
- [218] J. S. Fitzgerald and C. B. Jones. Proof in the analysis of a model of a tracking system. Technical Report CS-TR-617, Newcastle University, 1997.
- [219] C. B. Jones. An object-based design method for concurrent programs. Technical Report UMCS-92-12-1, Manchester University, 1992.
- [220] C. B. Jones and A. M. McCauley. Formal methods – selected historical references. Technical Report UMCS-92-12-2, Manchester University, 1992.
- [221] C. B. Jones. Systematic software development using VDM (second edition) teaching notes. Technical report, Prentice Hall International, 1990.
- [222] C. B. Jones. Final report on SERC Grant No. GR/D/60294: IPSE 2.5. internal communication, July 1990.
- [223] C. B. Jones and R. Moore. An experimental user interface for a theorem proving assistant. Technical Report UMCS 87-12-8, Manchester University, December 1987.
- [224] I. D. Cottam, C. B. Jones, T. N. Nipkow, A. C. Wills, M. Wolczko, and A. Yaghi. Mule – an environment for rigorous software development (final report to SERC on grant number GR/C/05762). Technical report, Department of Computer Science, University of Manchester, 1986.
- [225] C. B. Jones. Teaching notes for systematic software development using VDM. Technical Report UMCS 86-4-2, University of Manchester, 1986.
- [226] C. B. Jones. Program specification and verification in VDM. Technical Report UMCS 86-10-5, University of Manchester, 1986. extended version of [121] (includes the full proofs).
- [227] C. B. Jones, P. Lindsay, and C. Wadsworth. Ipse 2.5 theorem proving concepts paper. Technical Report 060/00021/1.5, Manchester University and Rutherford Appleton Laboratory, June 1986.
- [228] B. T. Denvir, V. A. Downes, C. B. Jones, R. A. Snowdon, and M. K. Tordoff. Ipse 2.5 project proposal. Technical report, ICL/STC-IDEDEC/STL/University of Manchester, February 1985.
- [229] H. Barringer and C. B. Jones. A survey of verification techniques for parallel programs. Technical report, University of Manchester, September 1982.
- [230] C. B. Jones. *Development Methods for Computer Programs including a Notion of Interference*. PhD thesis, Oxford University, June 1981. Printed as: Programming Research Group, Technical Monograph 25.

- [231] C. B. Jones. Implementation bias in constructive specification of abstract objects. typescript, September 1977.
- [232] C. B. Jones. Formal definition in compiler development. Technical Report 25.145, IBM Laboratory Vienna, February 1976.
- [233] C. B. Jones. Program development using data abstraction. typescript, December 1976.
- [234] H. Bekič, H. Izbicki, C. B. Jones, and F. Weissenböck. Some experiments with using a formal language definition in compiler development. Technical Report LN 25.3.107, IBM Laboratory, Vienna, December 1975.
- [235] H. Bekič, D. Bjørner, W. Henhagl, C. B. Jones, and P. Lucas. A formal definition of a PL/I subset. Technical Report 25.139, IBM Laboratory Vienna, December 1974.
- [236] D. N. Chapman and C. B. Jones. Formal development: An example using FDSS. Technical Report TN 9007, IBM Laboratory, Hursley, 1973.
- [237] C. D. Allen and C. B. Jones. The formal development of an algorithm. Technical Report 12.110, IBM Laboratory Hursley, March 1973.
- [238] C. B. Jones. Formal development of programs. Technical Report 12.117, IBM Laboratory Hursley, June 1973.
- [239] C. D. Allen, D. N. Chapman, and C. B. Jones. A formal definition of ALGOL 60. Technical Report 12.105, IBM Laboratory Hursley, August 1972.
- [240] C. B. Jones. Operations and formal development. Technical Report TN 9004, IBM Laboratory, Hursley, September 1972.
- [241] C. B. Jones. Development of correct programs: An example based on earley's recogniser. Technical Report TN 9000, IBM Laboratory, Hursley, April 1971.
- [242] K. V. Hanford and C. B. Jones. Dynamic syntax: A concept for the definition of the syntax of programming languages. Technical Report 12.090, IBM Laboratory Hursley, June 1971.
- [243] C. B. Jones. Formal development of correct algorithms: an example based on Earley's recogniser. Technical Report 12.095, IBM Laboratory Hursley, December 1971.
- [244] W. Henhagl and C. B. Jones. On the interpretation of GOTO statements in the ULD. Technical Report LN 25.3.065, IBM Laboratory, Vienna, March 1970.
- [245] W. Henhagl and C. B. Jones. The block concept and some possible implementations, with proofs of equivalence. Technical Report 25.104, IBM Laboratory Vienna, April 1970.
- [246] C. B. Jones. A technique for showing that two functions preserve a relation between their domains. Technical Report LR 25.3.067, IBM Laboratory, Vienna, April 1970.

- [247] W. Henhagl and C. B. Jones. Some observations on the implementation of reference mechanisms for automatic variables. Technical Report LR 25.3.070, IBM Laboratory, Vienna, May 1970.
- [248] C. B. Jones and P. Lucas. Proving correctness of implementation techniques. Technical Report TR 25.110, IBM Laboratory Vienna, August 1970.
- [249] C. B. Jones. Yet another proof of the block concept. Technical Report LN 25.3.075, IBM Laboratory, Vienna, August 1970.
- [250] S. M. Glasover, K. V. Hanford, and C. B. Jones. The syntax machine: A description. Technical Report 12.078, IBM Laboratory Hursley, January 1969.
- [251] C. B. Jones. A proof of the correctness of some optimising techniques. Technical Report LN 25.3.051, IBM Laboratory, Vienna, June 1969.
- [252] C. B. Jones. A proof of the correctness of an interpreter based on axiom schema. Technical Report LN 25.3.053, IBM Laboratory, Vienna, June 1969.
- [253] C. B. Jones. Mathematical semantics of goto: Exit formulation and its relation to continuations. Technical report, IBM Laboratory Vienna, July 1968.
- [254] S. M. Glasover, K. V. Hanford, and C. B. Jones. The syntax machine: An introduction. Technical Report 12.077, IBM Laboratory Hursley, October 1968.
- [255] P. B. Schofield and C. B. Jones. A random generator for the automatic production of executable PL/I programs (APEX). Technical Report 12.062, IBM Laboratory Hursley, April 1966.
- [256] P. B. Schofield and C. B. Jones. APEX manual. IBM Laboratory, Hursley, April 1966.
- [257] C. B. Jones. A retrospective view of APEX (Automatic Production of Executable Programs). Technical Report 12.065, IBM Laboratory Hursley, October 1966.