Intuition before Formalism
References

June 23, 2015

Relational post conditions: [Flo67, dBS69, Hoa69, Jon73, Jon80, Acz82, Jon86]
Data abstraction/reification: [Luc68, Jon70, Jon80, Jon86, Mar86, Nip86, Jon07]
Tool support: [Kin69, JJLM91]
LPF: [GS96, Kle52, BCJ84, JM94, JLS13]
Concurrency: [BJ84, OG76, Jon77, Jon81, BKP84, Jon10, JHC15, San99, Jon07, JY15]
Semantics: [Jon01, JL71, Jon99, ACJ72, BBH$^+$74, BJ78, BJ82, Jon78, Jon82]
History resources: http://homepages.cs.ncl.ac.uk/cliff.jones/semantics-library/
History: [MJ84, Jon03]

References

[ACJ72] 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.

[Acz82] P. Aczel. A note on program verification. (private communication) Manuscript, Manchester, January 1982.

[BBH$^+$74] H. Bekič, D. Bjørner, W. Henhapl, C. B. Jones, and P. Lucas. A formal definition of a PL/I subset. Technical Report 25.139, IBM Laboratory Vienna, December 1974.

[BCJ84] H. Barringer, J.H. Cheng, and C. B. Jones. A logic covering undefinedness in program proofs. *Acta Informatica*, 21(3):251–269, 1984.

[BJ78] 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.

[BJ82] Dines Bjørner and Cliff B. Jones, editors. *Formal Specification and Software Development*. Prentice Hall International, 1982.

[BJ84] 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.

[BKP84] H. Barringer, R. Kuiper, and A. Pnueli. Now you can compose temporal logic specification. In *Proceedings of 16th ACM STOC*, Washington, May 1984.

[dBS69] J. W. de Bakker and D. Scott. A theory of programs. Manuscript notes for IBM Seminar, Vienna, August 1969.

[Flo67] R. W. Floyd. Assigning meanings to programs. In *Proc. Symp. in Applied Mathematics, Vol.19: Mathematical Aspects of Computer Science*, pages 19–32. American Mathematical Society, 1967.

[GS96] David Gries and Fred B. Schneider. *A Logical Approach to Discrete Math*. Springer-Verlag, second edition, 1996.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 583, October 1969.

[JHC15] Cliff B. Jones, Ian J. Hayes, and Robert J. Colvin. Balancing expressiveness in formal approaches to concurrency. *Formal Aspects of Computing*, 27:475–497, 2015.

[JJLM91] C. B. Jones, K. D. Jones, P. A. Lindsay, and R. Moore. *mural: A Formal Development Support System*. Springer-Verlag, 1991.

[JL71] 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.

[JLS13] C. B. Jones, M. J. Lovert, and L. J. Steggles. Revising basic theorem proving algorithms to cope with the logic of partial functions. *Science of Computer Programming*, 2013.

[JM94] C.B. Jones and C.A. Middelburg. A typed logic of partial functions reconstructed classically. *Acta Informatica*, 31(5):399–430, 1994.

[Jon70] 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.

[Jon73] C. B. Jones. Formal development of programs. Technical Report 12.117, IBM Laboratory Hursley, June 1973.

[Jon77] C. B. Jones. Implementation bias in constructive specification of abstract objects. typescript, September 1977.

[Jon78] Cliff B. Jones. Denotational semantics of goto: An exit formulation and its relation to continuations. In Bjørner and Jones [BJ78], pages 278–304.

[Jon80] C. B. Jones. *Software Development: A Rigorous Approach*. Prentice Hall International, Englewood Cliffs, N.J., USA, 1980.

[Jon81] 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.

[Jon82] Cliff B. Jones. More on exception mechanisms. In Bjørner and Jones [BJ82], chapter 5, pages 125–140.

[Jon86] C. B. Jones. *Systematic Software Development Using VDM*. Prentice Hall International, 1986.

[Jon99] 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.

[Jon01] C. B. Jones. The transition from VDL to VDM. *Journal of Universal Computer Science*, 7(8):631–640, 2001.

[Jon03] Cliff B. Jones. The early search for tractable ways of reasonning about programs. *IEEE, Annals of the History of Computing*, 25(2):26–49, 2003.

[Jon07] C. B. Jones. Splitting atoms safely. *Theoretical Computer Science*, 375(1–3):109–119, 2007.

[Jon10] 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.

[JY15] Cliff B. Jones and Nisansala Yatapanage. Reasoning about separation using abstraction and reification. 2015. submitted to SEFM-2015.

[Kin69] J. C. King. *A Program Verifier*. PhD thesis, Department of Computer Science, Carnegie-Mellon University, 1969.

[Kle52] S. C. Kleene. *Introduction to Metamathematics*. Van Nostrad, 1952.

[Luc68] P. Lucas. Two constructive realizations of the block concept and their equivalence. Technical Report TR 25.085, IBM Laboratory Vienna, June 1968.

[Mar86] L.S. Marshall. *A Formal Description Method for User Interfaces*. PhD thesis, University of Manchester, 1986.

[MJ84] 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.

[Nip86] T. Nipkow. Non-deterministic data types: Models and implementations. *Acta Informatica*, 22:629–661, 1986.

[OG76] S. S. Owicki and D. Gries. An axiomatic proof technique for parallel programs I. *Acta Informatica*, 6:319–340, 1976.

[San99] Davide Sangiorgi. Typed π-calculus at work: a correctness proof of Jones's parallelisation transformation on concurrent objects. *Theory and Practice of Object Systems*, 5(1):25–34, 1999.