Annotated bibliography on rely/guarantee conditions

Cliff B. Jones

School of Computing Science
Newcastle University, NE1 7RU, UK
e-mail: cliff.jones@ncl.ac.uk

Health warning This note makes no pretext of being balanced: it is a collection of references I have to hand and opinions on other people's approaches "may not be unbiased"!

It is also very hard (and I fail) to keep this note up to date. The most exciting current issue is understanding the link between rely/guarantee thinking and Separation Logic (see final section below).

For reasoning about sequential (non-concurrent) programs it is reasonably standard to take [Hoa69] as reference point (that is exactly what is done in the historical essay [Jon03b]); "Floyd/Hoare rules" for sequential programs are compositional in a way that means they can be used in design (as well as *post facto* verification).

It proved more challenging (cf. [Jon03a]) to achieve "compositionality" for concurrent programs: [dR01] provides a detailed discussion and extensive references up to the time of its publication.[1]

History

- In [AM71] the fact that two processes can interfere with each other's states is handled by constructing a single, equivalent, non-deterministic (sequential) program (which needs a number of assertions related exponentially to the size of the programs); the approach is in no way compositional because the correctness of the two processes cannot even be considered until their final code is available; furthermore, it implicitly makes an arbitrary decision about atomicity.
- I think it is fair to say that the "Owicki/Gries" approach [Owi75,OG76] is *not* compositional because, if the final *Einmischungsfrei* property does not hold, a whole development might have to be discarded (this approach also makes atomicity assumptions).

Rely/guarantee

- Rely/guarantee conditions [Jon81,MC81,Jon83b,Jon83a] offer one approach to recording and reasoning about interference in a way which recaptures

[1] The bulk of this discussion is in terms of shared-variable concurrency but the problem of "interference" is also present with communication-based concurrency (as is argued in [Jon03a]).

compositionality and puts atomicity decisions in the hands of the developer [Jon06]. Peter Aczel wrote a(n unpublished) note which cleaned up my notation considerably [Acz82].
- A useful early report on applying rely/guarantee conditions on an industrial problem is [WD88].
- After a significant pause, a significant number of PhD theses built on the rely/guarantee ideas [Stø90,Xu92,Col94,Din00,BS01,Bue00].
- Within the temporal logic framework, the issue of compositionality is tackled in [BKP84].
- Completeness was perhaps first tackled in an unpublished note by Ruurd Kuiper [Kui83]; it is treated thoroughly in [Sti86,Sti88].
- The FOCUS method [BS01] combines rely/guarantee ideas with earlier ideas of Manfred Broy.
- Other publications more or less related to rely/guarantee include [Sta86,CJ00,GNL91,dR85,HdR86,Zwi88]
- Searching for ways to *constrain* interference (and thus reduce the number of places where one has to use rely/guarantee conditions) led to a line of research ("POBL" or "$\pi o \beta \lambda$") on concurrent Object-Based Languages [Jon93a,Jon93b,Jon94,HJ96]
- A useful reference which summarises both rely/guarantee reasoning and the "POBL" approach is [Jon96].
- An examination of the notion of "atomicity" is given in [JLRW05] — see also [BJ05a,BJ05b,Bur04,CJ07b].
- Machine checked proofs of soundness of both the Owicki/Gries approach and that of rely/guarantee conditions are tackled in [Pre01,Pre03].
- Together with Joey Coleman, I have returned to proving the consistency of rely/guarantee condition proof rules in [CJ07a,Col08].
- The development of Simpson's "four slot" algorithm for "Asynchronous Communication Mechanisms" is tackled in [JP08] (which will hopefully be superseded by a journal version of [Jon09]).
- An examination of the role of auxiliary variables is to be given in a paper titled "The role of auxiliary variables in the formal development of concurrent programs" being written for [JRW09].

Connections with "Separation Logic"

This is an exciting development — the "must read" thesis is [Vaf07] which introduces "RG-Sep". Ken Pierce's Newcastle thesis tackles the connections between rely/guarantee thinking and

At slightly greater length:

- Development [Bur72,Rey00,Rey02,OP99,IO01,ORY01]
- Description [O'H07,Bro07,OYR09]
- Particularly relevant [PB05]
- As yet unpublished [BA08]

References

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

[AM71] E. A. Ashcroft and Z. Manna. Formalization of properties of parallel programs. In B. Meltzer and D. Michie, editors, *Machine Intelligence, 6*, pages 17–41. Edinburgh University Press, 1971.

[BA08] Richard Bornat and Hasan Amjad. Inter-process buffers in separation logic with rely-guarantee, 2008. (private communication) Submitted to Formal Aspects of Computing.

[BJ05a] J. I. Burton and C. B. Jones. Atomicity in system design and execution. *Journal of Universal Computer Science*, 11(5):634–635, 2005.

[BJ05b] J. I. Burton and C. B. Jones. Investigating atomicity and observability. *Journal of Universal Computer Science*, 11(5):661–686, 2005.

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

[BMJ88] R. Bloomfield, L. S. Marshall, and R. B. Jones, editors. *VDM'88: VDM – The Way Ahead*, volume 328 of *Lecture Notes in Computer Science*. Springer-Verlag, 1988.

[Bro07] S. D. Brookes. A semantics of concurrent separation logic. *Theoretical Computer Science (Reynolds Festschrift)*, 375(1-3):227–270, 2007. (Preliminary version appeared in CONCUR'04, LNCS 3170, pp16-34).

[BS01] Manfred Broy and Ketil Stølen. *Specification and Development of Interactive Systems*. Springer-Verlag, 2001.

[Bue00] Martin Buechi. *Safe Language Mechanisms for Modularization and Concurrency*. PhD thesis, Turku, 2000.

[Bur72] R.M. Burstall. Some techniques for proving correctness of programs which alter data structures. *Machine Intelligence*, 7:23–50, 1972.

[Bur04] J. Burton. *The Theory and Practice of Refinement-After-Hiding*. PhD thesis, University of Newcastle upon Tyne, 2004.

[CJ00] 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.

[CJ07a] 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.

[CJ07b] J.W. Coleman and C.B. Jones. Atomicity: A unifying concept in computer science. *Journal of Universal Computer Science*, 13(8):1042–1043, 2007.

[Col94] Pierre Collette. *Design of Compositional Proof Systems Based on Assumption-Commitment Specifications – Application to UNITY*. PhD thesis, Louvain-la-Neuve, June 1994.

[Col08] Joseph William Coleman. *Constructing a Tractable Reasoning Framework upon a Fine-Grained Structural Operational Semantics*. PhD thesis, Newcastle University, January 2008.

[Din00] Jürgen Dingel. *Systematic Parallel Programming*. PhD thesis, Carnegie Mellon University, 2000. CMU-CS-99-172.

[dR85] W.-P. de Roever. The quest for compositionality: A survey of assertion-based proof systems for concurrent programs: Part I: Concurrency based on shared variables. In *[NC85]*, pages 181–205, 1985.

[dR01] W. P. de Roever. *Concurrency Verification: Introduction to Compositional and Noncompositional Methods*. Cambridge University Press, 2001.

[GNL91] Peter Grønning, Thomas Qvist Nielsen, and Hans Henrik Løvengreen. Refinement and composition of transition-based rely-guarantee specifications with auxiliary variables. In K.V. Nori and C.E. Veni Madhavan, editors, *Foundations of Software Technology and Theoretical Computer Science*, volume 472 of *Lecture Notes in Computer Science*, pages 332–348. Springer-Verlag, 1991.

[HdR86] J. Hooman and W.-P. de Roever. The quest goes on: A survey of proof systems for partial correctness of CSP. In J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Current Trends in Concurrency*, volume 224 of *Lecture Notes in Computer Science*. Springer-Verlag, 1986.

[HJ96] 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, *Oject Orientation with Parallelism and Persistence*, pages 1–22. Kluwer Academic Publishers, 1996.

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

[IO01] S. Isthiaq and P. W. O'Hearn. BI as an assertion language for mutable data structures. In *28th POPL*, pages 36–49, 2001.

[JLRW05] C. B. Jones, D. Lomet, A. Romanovsky, and G. Weikum. The atomic manifesto. *Journal of Universal Computer Science*, 11(5):636–650, 2005.

[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.

[Jon83a] C. B. Jones. Specification and design of (parallel) programs. In *Proceedings of IFIP'83*, pages 321–332. North-Holland, 1983.

[Jon83b] C. B. Jones. Tentative steps toward a development method for interfering programs. *Transactions on Programming Languages and System*, 5(4):596–619, 1983.

[Jon93a] C. B. Jones. Constraining interference in an object-based design method. In M-C. Gaudel and J-P. Jouannaud, editors, *TAPSOFT'93*, volume 668 of *Lecture Notes in Computer Science*, pages 136–150. Springer-Verlag, 1993.

[Jon93b] C. B. Jones. Reasoning about interference in an object-based design method. In J. C. P. Woodcock and P. G. Larsen, editors, *FME'93*, volume 670 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 1993.

[Jon94] C. B. Jones. Process algebra arguments about an object-based design notation. In *A Classical Mind: Essays in Honour of C. A. R. Hoare*, chapter 14. Prentice-Hall, 1994.

[Jon96] 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.

[Jon03a] C. B. Jones. Wanted: a compositional approach to concurrency. In Annabelle McIver and Carroll Morgan, editors, *Programming Methodology*, pages 1–15. Springer Verlag, 2003.

[Jon03b] 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.

[Jon06] C. B. Jones. An approach to splitting atoms safely. *Electronic Notes in Theoretical Computer Science, MFPS XXI, 21st Annual Conference of Mathematical Foundations of Programming Semantics*, 155:43–60, 2006.

[Jon09] Cliff B Jones. Elucidating concurrent algorithms via layers of abstraction and reification. Technical Report CS-TR-1166, School of Computing Science, Newcastle University, 2009.

[JP08] Cliff B. Jones and Ken G. Pierce. Splitting atoms with rely/guarantee conditions coupled with data reification. In *ABZ2008*, volume LNCS 5238, pages 360–377, 2008.

[JRW09] Cliff Jones, Bill Roscoe, and Ken Wood, editors. *Reflections on the work of C.A.R.Hoare*. Springer, 2009.

[Kui83] Ruurd Kuiper. On completeness of an inference rule for parallel composition, 1983. (private communication) Manuscript, Manchester.

[MC81] J. Misra and K. M. Chandy. Proofs of networks of processes. *IEEE Transactions on Software Engineering*, 7:417–426, 1981.

[NC85] E. J. Neuhold and G. Chroust. *Formal Models in Programming*. North-Holland, 1985. Proceedings of the IFIP TC2 Working Conference on The Role of Abstract Models in Information Processing. Vienna, Austria, 30 January – 1 February 1985.

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

[O'H07] P. W. O'Hearn. Resources, concurrency and local reasoning. *Theoretical Computer Science (Reynolds Festschrift)*, 375(1-3):271–307, May 2007. Preliminary version appeared in CONCUR'04, LNCS 3170, 49–67.

[OP99] P. W. O'Hearn and D. J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, June 99.

[ORY01] P. O'Hearn, J. Reynolds, and H. Yang. Local reasoning about programs that alter data structures. In *15th CSL*, pp1–19, 2001.

[Owi75] S. Owicki. *Axiomatic Proof Techniques for Parallel Programs*. PhD thesis, Department of Computer Science, Cornell University, 1975.

[OYR09] P. W. O'Hearn, H. Yang, and J. C. Reynolds. Separation and information hiding. *ACM TOPLAS*, 31(3), April 2009. Preliminary version appeared in 31st POPL, pp268-280, 2004.

[PB05] Matthew Parkinson and Gavin Bierman. Separation logic and abstraction. In *POPL '05: Proceedings of the 32nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 247–258, New York, NY, USA, 2005. ACM.

[Pre01] Leonor Prensa Nieto. *Verification of Parallel Programs with the Owicki-Gries and Rely-Guarantee Methods in Isabelle/HOL*. PhD thesis, Institut für Informatic der Technischen Universitaet München, 2001.

[Pre03] Leonor Prensa Nieto. The rely-guarantee method in Isabelle/HOL. In *Proceedings of ESOP 2003*, volume 2618 of *LNCS*. Springer-Verlag, 2003.

[Rey00] J. C. Reynolds. Intuitionistic reasoning about shared mutable data structure. In Jim Davies, Bill Roscoe, and Jim Woodcock, editors, *Millennial Perspectives in Computer Science*, pages 303–321, Houndsmill, Hampshire, 2000. Palgrave.

[Rey02] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of 17th LICS*, pages 55–74. IEEE, 2002.

[Sta86] Eugene W. Stark. A proof technique for rely/guarantee properties. In S.N. Maheshwari, editor, *Foundations of Software Technology and Theoretical Computer Science*, volume 206 of *Lecture Notes in Computer Science*, pages 369–391. Springer-Verlag, 1986.

[Sti86] C. Stirling. A compositional reformulation of Owicki-Gries' partial correctness logic for a concurrent while language. In *ICALP'86*. Springer-Verlag, 1986. LNCS 226.

[Sti88] C. Stirling. A generalisation of Owicki-Gries's Hoare logic for a concurrent while language. *Theoretical Computer Science*, 58:347–359, 1988.

[Stø90] K. Stølen. *Development of Parallel Programs on Shared Data-Structures*. PhD thesis, Manchester University, 1990. Available as UMCS-91-1-1.

[Vaf07] Viktor Vafeiadis. *Modular fine-grained concurrency verification*. PhD thesis, University of Cambridge, 2007.

[WD88] J. C. P. Woodcock and B. Dickinson. Using VDM with rely and guarantee-conditions: Experiences of a real project. In *[BMJ88]*, pages 434–458, 1988.

[Xu92] Qiwen Xu. *A Theory of State-based Parallel Programming*. PhD thesis, Oxford University, 1992.

[Zwi88] J. Zwiers. *Compositionality, Concurrency and Partial Correctness: Proof theories for networks of processes, and their relationship*. PhD thesis, Technical University Eindhoven, 1988. Available as LNCS 321, Springer-Verlag.