On the Borowsky-Gafni Simulation Algorithm

(Brief Announcement)

Nancy Lynch * Sergio Rajsbaum †

Consider a read/write asynchronous shared memory system. In [1], Borowsky and Gafni describe an algorithm that allows a set of $f+1$ processes, any f of which may exhibit stopping failures, to "simulate" a larger number n of processes, also with at most f failures. This simulation algorithm is used in [1] to convert an arbitrary k-fault-tolerant n-process solution for the k-set-agreement problem into a wait-free $k+1$-process solution for the same problem. Since the $k+1$-process k-set-agreement problem has been shown to have no wait-free solution (e.g. [1]), this transformation implies that there is no k-fault-tolerant solution to the n-process k-set-agreement problem, for any n.

This and other initial examples suggest that the Borowsky-Gafni simulation can become a powerful tool for proving solvability and unsolvability results for fault-prone asynchronous systems. However, in order for this to happen, it must be clear exactly what the simulation algorithm is, and what it guarantees.

We began this research with the modest aim of stating and proving precise correctness guarantees for the Borowsky-Gafni simulation algorithm, using the I/O automaton model (e.g. [2]) and standard proof techniques. However, the job turned out to be more difficult than we expected, because the description in [1] is brief and informal, and does not include a careful specification of what the algorithm provides to its users; it leaves some ambiguities that we needed to resolve. [1] The final product of our work is a complete and careful description of a version of the Borowsky-Gafni simulation algorithm, plus a careful description of what it accomplishes, plus a proof of correctness.

In order to specify what the simulation accomplishes, we define a notion of *fault-tolerant reducibility* between decision problems, and show that the algorithm implements this reducibility, in a precise sense. We give some examples of pairs of decision problems that do and do not satisfy this reducibility. In contrast, [1] deals only with the use of the simulation in the set agreement problem.

The presentation has a great deal of interesting modularity, expressed by I/O automaton composition and both forward and backward simulation relations. Composition is used to include a *safe agreement* module, a simplification of one in [1], as a subroutine. Forward and backward simulation relations are used to view the algorithm as implementing a *multi-try snapshot* strategy. The most interesting part of the proof is the safety argument, which is handled by the forward and backward simulation relations; once that is done, the liveness argument is straightforward.

*lynch@theory.lcs.mit.edu. Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. Supported by Air Force Contract AFOSR F49620-92-J-0125, NSF contract 9225124CCR, and DARPA contracts N00014-92-J-4033 and F19628-95-C-0118.

†rajsbaum@servidor.unam.mx. Instituto de Matemáticas, U.N.A.M., Ciudad Universitaria, D.F. 04510, México. Supported by DGAPA projects.

PODC'96, Philadelphia PA, USA
© 1996 ACM 0-89791-800-2/96/05..$3.50

References

[1] E. Borowsky and E. Gafni, "Generalized FLP impossibility result for t-resilient asynchronous computations," in *Proceedings of the 1993 ACM Symposium on Theory of Computing*, May 1993.

[2] N.A. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers, Inc. 1996.

[1] For example, the simulation does not work in the read/write shared memory model that we believe is intended in [1]; we instead use an atomic snapshot memory.