Operating Systems Review 16(4) October 1982

Reply to "On Proof Rules for Monitors"

John H. Howard

IBM Research Laboratory
5600 Cottle Rd
San Jose, California

May 5, 1982

A paper by Adams and Black[1] recently printed in OSR contains a serious error. As a result its authors claim that there are flaws in the previously-proposed proof rules for monitors' cond.wait and cond.signal operations. I hope in this note to clarify the resulting confusion.

The monitor proof rules proposed by Hoare[2] are:

{ J } each procedure body { J }
{ J } cond.wait { J & B }
{ J & B } cond.signal { J }

In order to prove things about the presence of waiting processes, these rules can be extended as follows:[3,4]

{ J & E } each procedure body { J & E }
{ J & E } cond.wait { J & B }
{ J & B } cond.signal { J & E }

The meaning of the various predicates ("invariants") in these rules is:

J asserts that the monitor's data is in a consistent state, but does not deal with pending or needed signals. For the semaphore monitor[3], J is np<=min(na,nv). Intuitively this means that the number of completed P operations does not exceed either the number of attempted ones or the number of V operations.

B asserts that the condition variable associated with B should be signaled. (there is a separate B for each condition variable.) For the semaphore example, B is np < na & np = nv-1. Intuitively this says that there exists an attempted P operation which is not complete (that is, there is a queue at the semaphore) and that there has been exactly one more V than there have been P's. This will happen only during execution of the V procedure, and means that a signal is required.

E asserts that none of the condition variables need signaling. For the semaphore example, E is np>=min(na,nv).

Habermann's invariant for semaphores[5] is np = min(na,nv), which is readily seen to be the conjunction J & E. This is used as the monitor's entry/exit invariant as well as the precondition of cond.wait and postcondition of cond.signal

Adams and Black's misconception is to use the Habermann invariant for J rather than for J & E. This leads to problems since it includes the predicate np>=min(na,nv) in the precondition of cond.signal. Since this predicate says that either nobody is waiting or else there are no excess V operations, signals can be performed only when they are not needed. Adams and Black correctly point out that this (wrong) J cannot be used as the precondition for cond.signal, leading to a "flawed" proof. The real flaw, however, is in their selection of the wrong predicate for J.

The remainder of the paper addresses technical details such as the (implicit) effects of cond.signal and cond.wait operations on queues. This area was touched upon briefly in [3], and covered thoroughly in [4], which is not cited by Adams and Black.

In summary, the paper is based on an erroneous premise and misses some relevant prior work. Significant new knowlege about monitor semantics is more likely to be discovered by stepping back and taking the broad view than by fiddling with details. A reasonable approach for further work would be to search for a way to avoid the ambiguities and complexities described in [4].

References

[1] Adams, J.M. & Black, A.P. "On Proof Rules for Monitors", ACM Operating Systems Review 16,2 (April 1982), 18.

[2] Hoare, C.A.R. "Monitors: an operating system structuring concept", Comm ACM 17,10 (Oct 1974), 549.

[3] Howard, J. "Proving Monitors", Comm ACM 19,5 (May 1976), 273.

[4] Howard, J. "Signaling in Monitors", Second International Conference on Software Engineering, IEEE Catalog #76CH1125-4C (Oct 1976), 47.

[5] Habermann, A.N. "Synchronization of communicating processes", Comm ACM 15,3 (March 1972), 171.