Operating Systems Review 17(1) Jan 1983

LETTERS

Dear Dr Waite,

In his recent paper *Reply to "On Proof Rules for Monitors"* [4], John Howard remarks that [1] contains a serious error. In that he is less than correct; it actually contains two! However, it does not follow that the previously proposed proof rules for monitors are therefore perfect, and we fear that [4] provides little clarification.

Habermann's invariant has been accepted as the definition of the counting semaphore for over ten years. Hoare's proof rules for monitors have been similarly accepted since 1974. It has been widely supposed that Hoare's rules are sufficient to prove that the monitor *S1* given in the Introduction to [1] (see Figure 1) is equivalent to a counting semaphore with a constant of zero. It was while one of us was teaching this topic to a class of students that it became apparent that Hoare's rules are not sufficient for this task. Howard's letter does not dispute that Hoare's rules cannot be used to prove that the monitor *S1* is equivalent to a semaphore.

Note that there is no implication that Hoare's rules are *wrong*. They are right in the sense that the statements they make about *cond.wait* and *cond.signal* are true. Hoare's rules do not, however, capture all of the semantics he informally attributed to the *wait* and *signal* operations. In order to formalise the fact that *cond.signal* is a null operation if no processes are waiting on *cond*, Hoare's rules require a rather strong precondition for *cond.signal*: the monitor invariant must be true as well as the resource condition.

Howard's rules, first stated in [2], *can* be used to prove that *S1* simulates a semaphore. Our claim in [1] that they could not is wrong, and we are grateful to Howard for pointing this out. The applicability of Howard's rules is not obvious from the text of [2], where the proof of the semaphore monitor does not use the general proof rules, but is given before they are developed.

Our mistake occurred because we failed to realise that Howard's rules can be interpreted in two different ways. As normally written, Howard's rules replace Hoare's monitor invariant J by the expression J & E, and leave the precondition for *signal* unchanged as J & B. We regarded this as strengthening the invariant. It may also be viewed as a weakening of the precondition for *signal*: the whole of the monitor invariant is no longer required as a pre-condition for *signal*, but only a part of it (J). It is necessary to take this second point of view in order to realise that Howard's rules can be applied to *S1*. Our "erroneous premise" was to take only the first point of view, and not the second.

The second point of view has its own difficulties, however. Hoare's *signal* operation is clearly stated to be a null operation in the case where there is no outstanding *wait*. However, permitting the precondition of *signal* (J & B) not to imply the postcondition (the monitor invariant J & E) means that *cond.signal* can never be allowed to be null.

```
monitor S1;
var
        na, np, nv : integer;
        cond : condition;
        { Invariant is np = min(na, nv) }

procedure P;
begin
        na := na + 1;
        if na > nv then cond.wait;
        np := np + 1
end P;

procedure V;
begin
        nv := nv + 1;
        if na > np then cond.signal
end V;

begin {initialization}
        na := np := nv := 0
end S1
```

Figure 1

```
monitor S2;
var
        {np,} na, nv : integer;
        cond : condition;
        { Invariant is np = min(na, nv) }

procedure P;
begin
        na := na + 1;
        if na > nv then cond.wait
        { np := np + 1 }
end P;

procedure V;
begin
        nv := nv + 1;
        cond.signal
end V

begin {initialisation}
        {np := } na := nv := 0
end S2
```

Figure 2

Thus Howard's rule for signal cannot be applied when there are no processes waiting on *cond*, i.e. when ¬ *cond.queue*.

This property of Howard's rules is clearly stated in [2]. For this reason we considered Howard's rules to define the semantics of a new construct for monitor synchronisation that is different from Hoare's. Howard's subsequent paper [3] deals exclusively with signaling primitives that require the existence of at least one waiting process. This is one of the reasons why we did not refer to this paper in [1]: we had no wish to minimise Howard's contribution, but simply saw our paper as addressing a different topic.

Howard's *cond.signal* primitive is strictly weaker than Hoare's; it has a smaller domain of applicability, but where both primitives are applicable they have the same effect. Thus it is quite correct to apply Howard's rules to *S1*, because *cond.queue* is true before every call of *cond.signal*. However, Howard's rules cannot be applied to a monitor in which this condition does not hold: Figure 2 (also shown in Section 3 of [1]) illustrates such a monitor *S2*. According to the informal description Hoare gave of the semantics of his *cond.signal* operation, *S2* is certainly a valid semaphore, although this cannot be demonstrated using his proof rules. Can Howard's rules help us in this case? No: they apply to a different *signal* operation which cannot be used in *S2*. This was the problem that led us to propose the new rules described in [1], which are adequate to prove both *S1* and *S2*.

The second error in [1] occurs in the final paragraph. We state that the rule of assignment does not help us to establish the meaning of *cond.queue* because operations on *cond* in one procedure affect the value of *cond.queue* in other textually unrelated ones. This misses the essential property of a monitor, that only one procedure can be active at a time. Because of this property, *wait* and *signal* operations *can* be treated as assignments which affect the number of waiting processes. The resulting proof rules are similar in principle to those Howard gives in [3] involving queue lengths, but naturally differ in that signaling of a condition is always permitted.

In our view, the details of the semantics of monitors and their synchronisation are well worth "fiddling with". Proof rules which capture all of the semantics of Hoare's *wait* and *signal* operations have yet to be published, even though the monitor construct has appeared in a number of programming languages and has been used in the construction of several operating systems. A paper which steps back and takes the broad view must get the details right too. We feel that both Howard's work and our own offer useful insights into this topic, and that both will assist in the preparation of such a paper.

References

[1] Adams, J. M. and Black, A. P. *On Proof Rules for Monitors*, Operating Systems Review **16**, 2 (April 1982) pp 18–27.

[2] Howard, J. H. *Proving Monitors*, Comm ACM **19**, 5 (May 1976) pp 273–279.

[3] Howard, J. H. *Signaling in Monitors*, Proc. Second Int. Conf. Softw. Eng. (October 1976) pp 47–52.

[4] Howard, J. H. *Reply to "On Proof Rules for Monitors"*, Operating Systems Review **16**, 5 (October 1982) pp 8–9.

- J. Mack Adams
Andrew P. Black
Dept. of Computer Science, FR-35
University of Washington
Seattle, WA 98195