The Analysis of Multiprogrammed Systems of Unspecified Degree of Parallelism.

Multiprogrammed systems have been tackled by considering them as finite state machines with a considerable amount of non-determinacy, reflecting unspecified speed ratios. This is all right as far as it goes (see [1]), but it does not go very far if the reasoning forces us to choose beforehand the number of parallel processes. It is particularly unattractive if the amount of reasoning -or computing if the proof procedure is mechanized- rises steeply with the number of participating processes. The following example shows how even for general "n", a finite state analysis can suffice, provided that what we call "states" are carefully chosen.

It is the by now classical example of the readers and the writers, where the critical reader activity excludes the critical writer activity and where in addition the critical activity of any one writer excludes the same activity of any other writer. In short, in this example we have an unknown number of participants of two different types.

We have chosen a rather unusual solution, not because we have any reason for favouring it, but in order to show the power of our technique for analysing such systems. The chosen solution is rather sophisticated in granting a rather strong priority to writers and in trying to minimize the execution of synchronizing primitives and tests.

The common state space is controlled by four integer variables "ar, aw, br, bw" (short for "active or blocked readers or writers") and three semaphores, called R, W and mutex. Their initial values are

ar = aw = br = bw = R = W = 0   and mutex = 1.

The programs for the readers and the writers -for explanation of the notation see below, after the program texts- are as follows.

[1] Gilbert, P, and Chandler, W.J., Interference Between Communicating Parallel Processes. Comm.ACM 15, 6 (June 1972), 427 - 437

```
READER:
cycle non critical reader activity;
            P(mutex);
            if aw = 0 and bw = 0
               then ar plus 1; V(mutex)
               else br plus 1; V(mutex);
      empty statement 1;
                    P(R);
                    br minus 1; ar plus 1;
                    if br > 0
                       then V(R)
                       else V(mutex)
                    fi
            fi;
      critical reader activity;
            P(mutex); ar minus 1;
            if ar = 0 and bw > 0
               then V(W)
               else V(mutex)
            fi
elcyc

WRITER:
cycle non critical writer activity;
            P(mutex);
            if ar = 0 and aw = 0
               then aw plus 1; V(mutex)
               else bw plus 1; V(mutex);
      empty statement 2;
                    P(W);
                    bw minus 1; aw plus 1; V(mutex)
            fi;
      critical writer activity;
            P(mutex); aw minus 1;
            if bw > 0
               then V(W)
               else if br > 0
                       then V(R)
                       else V(mutex)
                    fi
            fi
elcyc
```

Note 1. As conditional clause we have used

if B then statement list else statement list fi ,

thus doing away with begin's and end's; the fi has been written right underneath the corresponding if, the enclosed then and else have been slightly more indented.

Note 2. We have used "ar plus 1" instead of "ar:= ar + 1" etc.

Note 3. The two empty statements have been inserted for the purpose of the discussion that is to follow.

Note 4. Instead of "aw plus 1" the program could have prescribed "aw:= 1" because whenever it is executed, aw = 0 holds initially; similarly "aw minus 1" could have been replaced by "aw:= 0". It is easier not to do so at the start, but to regard this as a possible optimization, permissible when the correctness proof has been given.

Note 5. The P-operation decreases the semaphore concerned by 1, the V-operation increases it by 1; semaphore values are by definition non-negative and therefore the P-operations represent the potential delays. When a V-operation takes place on a semaphore prohibiting at that moment the progress of more than one (other) process, we assume that the blocked processes are served on a FIFO basis (or any weaker assumption excluding the danger of individual starvation at that level.)

We start our considerations by drawing a number of (rather obvious, but vital) conclusions on account of the topology of the programs.

1) The programs are such that between two successive non-indented statements, control is such that first a P-operation is executed and finally one V-operation. As a result $0 \leq R + W + \text{mutex} \leq 1$, the non-indented ones are the only ones that -as far as the semaphores are concerned - can take place simultaneously, while at most one process can be engaged in the execution of its indented statements: they, together, form a sort of generalized critical section. It means that modification and inspection of the four common variables can take place without the danger of uncontrolled interference by any of the other processes.

2) The topology of the reader program is such that each critical reader activity is preceded by one execution of "ar plus 1", after which no further P-operation can postpone its critical activity; moreover each critical activity -which is taken to last until the following P-operation takes place- is then immediately followed by "ar minus 1". This, together with the initial value of ar and the mutual exclusion of the indented statements, tells us that we are entitled to interpret the value of ar as equal to (or temporarily slightly larger than) the number of critically active readers.

3) Similarly we can interpret aw as the number of critically active writers (see Note 4 above).

4) Similarly we can interpret br, the number of blocked readers as the number of readers engaged in "empty statement 1" and bw, the number of blocked writers, as the number of writers engaged in "empty statement 2". As a consequence, at least one reader can proceed when R = 1, and at least one writer when W = 1.

After the above observation we verify that outside critical sections -i.e. R + W + mutex = 1- will be in one of the states A through E, characterized by the following table.

ar	≥ 0	> 0	$= 0$	≥ 0	$= 0$
aw	$= 0$	$= 0$	$= 1$	$= 0$	$= 0$
br	$= 0$	≥ 0	≥ 0	> 0	≥ 0
bw	$= 0$	> 0	≥ 0	$= 0$	> 0
R	$= 0$	$= 0$	$= 0$	$= 1$	$= 0$
W	$= 0$	$= 0$	$= 0$	$= 0$	$= 1$
mutex	$= 1$	$= 1$	$= 1$	$= 0$	$= 0$
	A	B	C	D	E

It is easily verified that each pair of states -classes of states, if we want to be very precise- has an empty intersection: D and E have nothing in common with any other on account of the semaphore values, among the remaining three, C is the only one with aw = 1 and A and B are distinguished on account of bw = 0 or bw > 0. The system is started in state A, possible transitions are from A to A, B or C

from B to B or E

from C to A, C, D or E

from D to A or D

from E to C .

Furthermore it is now easily checked that the existence of at least one reader or at least one writer ensures that in each state at least one of the transitions is possible, excluding the case of general deadlock. Assuming FIFO (Note 5), the writers are free of the danger of individual starvation; the readers are exposed to this danger, but that is an essential consequence of the strong priority granted to the writers.

15th May 1973