

Two starvation-free solutions of a general exclusion problem.

In the canonical problem of the five dining philosophers, the philosophers, each of which alternately "thinks" and "eats", are arranged cyclicly, and no two neighbours may eat simultaneously. This constraint can be represented by placing the philosophers at the edges of a regular pentagon, each edge representing a pair-wise exclusion constraint between the two philosophers situated at its ends. This representation suggests a generalization: N philosophers, each placed at a vertex of an undirected graph of N vertices, and philosophers placed at directly connected vertices --so-called "neighbours"-- being not allowed to eat simultaneously. In the sequel we shall give two starvation-free solutions.

First solution.

In this solution we associate with each edge a three-valued variable, represented by an undirected edge and the two directions of an arrow along that edge. Initially all philosophers are thinking and all edges are undirected. We use angle brackets to denote mutually exclusive actions. The life of a philosopher can then be described as a repetition of:

```

think;
L0:   < direct outgoing arrows towards all your non-thinking
      neighbours --i.e. all your neighbours that are hungry or eating--,
      stop thinking and become hungry > ;
      wait until all your outgoing arrows have disappeared;
L1:   < stop being hungry and start eating > ;
      eat;
L2:   < remove all your incoming arrows, stop eating and start thinking >

```

Because L0 is the only arrow-creating action, each arrow is created by its source, and we conclude from this and the initial state

P1: Only hungry philosophers have outgoing arrows.

Furthermore it follows from L0 that being non-thinking is a necessary condition for receiving incoming arrows; on account of the initial state we can therefore conclude:

P2: Only non-thinking philosophers have incoming arrows.

Corollary 1. A thinking philosopher has no arrows.

But the inverse of Corollary 1 also holds:

P3: A pair of non-thinking neighbours implies an arrow between them.

Relation P3 is left invariant by L0 --which creates an arrow between each pair of non-thinking neighbours it introduces-- , and is also left invariant by L2 --which destroys a pair of non-thinking neighbours for each arrow it removes-- .

From P1 and P3 we conclude that of each pair of non-thinking neighbours, at least one must be hungry. Hence, eating neighbours are excluded. (Argument Bulterman.)

P4: There is no cyclic path of arrows.

Relation P4 could only be violated by L0, because L0 is the only arrow-creating action. But action L0 can never close a cyclic path, because the source of the arrows it creates is initially thinking, and --from Corollary 1-- hence has no incoming arrows.

Theorem If N is finite, and each eating period for each philosopher is finite, for each philosopher each hungry period is finite.

This theorem expresses the absence of the danger of individual starvation (and, hence, the absence of the danger of deadlock).

Proof. Because during a hungry period of philosopher C --a period, which ends when C has no outgoing arrows-- no new outgoing arrows for C are created, infinite hunger of C implies infinite existence of at least one outgoing arrow of C. Corollary 1 tells us that philosopher D at the target of that arrow is not thinking; D is not eating either, for then that arrow would disappear within a finite period of time. Hence, D is also infinitely long hungry. Repeating the argument we conclude the existence of an arrow from D to an infinitely long hungry E, etc., and, because the graph is finite, we conclude that infinite hunger of one philosopher implies that the arrows form at least one cyclic path, which contradicts P4. (End of proof.)

The first solution has been recorded now --I know it already much longer-- because, at last, I can show a correctness proof that fully satisfies me. Prior to the explicit formulation of P3, mutual exclusion used to be "argued" by means of an operational argument --usually even with a reductio ad absurdum!-- ; all that was very ugly, and Bulterman's argument does away with all that mess.

From a more abstract point of view, our first solution is perhaps not attractive: the implementation of the actions L0 requires exactly the same form of mutual exclusion as the eating actions! No two neighbours may perform their action L0 simultaneously. (This observation does not imply that under all circumstances, our first solution is useless. If L0 is very short in time compared with the eating action, and we have a mechanism that ensures that over the whole network only one copy of a given action is active, we can use that mechanism for the implementation of the actions L0, while it may be too restrictive to use it directly for the eating actions.) Our second solution overcomes these objections.

Second solution.

Our second solution does not assume a central daemon that in one way or another ensures mutual exclusion on a global scale. We associate with each edge a binary semaphore mutex, and the program for a philosopher has now the following form --initially all semaphores are = 1--

```

think;
P(mutexi); ... ; P(mutexj);
eat;
V(mutexi); ... ; V(mutexj)

```

where the P- and V-operations are performed, one for each edge meeting in the philosopher's vertex. Starvation and deadlock are effectively prevented by a very simple --although to my taste ugly-- trick: give an arbitrary numbering to the edges and deal in the sequence of P-operations with the semaphores in question in the order of increasing number.

The argument is simple. Assume that a philosopher is held up indefinitely; then there must exist a semaphore --with number p, say-- that is = 0 indefinitely. This can only be, because one of its neighbours has decreased it by 1, but is held up indefinitely by another semaphore --q, say-- . But because

on account of our ordering convention we can assert that $q > p$, repetition of the argument leads on a finite graph to an obvious contradiction.

The second solution is so ugly, because the ordering of the semaphores is so arbitrary. I can only accept it, provided the total time spent in the excluded actions is such a negligible fraction of real time that actual delay in a P-operation is, indeed, extremely rare and always very short. The second solution could under circumstances perhaps be used to implement the actions L0 of the first solution, which has the very nice property that neighbours are admitted to the dining table on the basis of first-come-first-served.

Plataanstraat 5
5671 AL NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow