A proof of quiescence of a distributed algorithm, Version 4
Jayadev Misra
8/30/00

The following problem was communicated to me by Edsger W. Dijkstra. An undirected connected finite graph has a natural number initially associated with each node. There is a distinguished node, *anchor*. A non-anchor node can make a *move* only if its value differs from v, which is 1 + the minimum value over all its neighbors; in that case it sets its value to v. Anchor never makes a move. A *computation* is a sequence of moves starting from the initial state. Show that each computation is finite.

Lemma 0: The value of any node is bounded.

Proof: Let M be the maximum initial value of any node. We show that a node at distance k from the anchor has a value at most $M + k$ at any moment. Proof is by induction on k.

- $k = 0$: Anchor's initial value is at most M. It never makes a move; hence its value is at most $M + 0$ at all times.
- $k+1$: The initial value of this node is at most M, which satisfies the bound. It has a neighbor at distance k, and, from the induction hypothesis, this neighbor's value is at most $M + k$. Therefore, the minimum over all its neighbors is at most $M + k$, and any move of this node assigns it a value of at most $M + k + 1$. □

A move that assigns value n to a node is called a n-move; a move that assigns a value below n is a $\lhd n$-move.

Lemma 1: There is a finite number of $\lhd n$-moves, for any n.

Proof: Proof is by induction on n.

- $n = 0$: No move sets a node value to less than 0.
- $n + 1,\ n \geq 0$: From the induction hypothesis, there is a finite number of $<n$-moves. Consider the point in the computation, p, at which all such moves have been made. We claim that any node performs at most one n-move beyond p. Hence, there is a finite number of $\lhd(n + 1)$-moves.

  Let the first n-move of y beyond p be at q. The next move of y beyond q, if there is one, is not an n-move, because consecutive moves of the same node assign it different values; nor is the move a $<n$-move because all such moves have been completed. Therefore, this move assigns y a value exceeding n. Hence, every neighbor of y has a value at least n at this point. Subsequently, since there is no $<n$-move, value of each neighbor remains at least n. Therefore, all subsequent moves of y assign it values exceeding n; i.e., they are not n-moves. □

From Lemma 0, each node value is bounded. That is, there is a value B such that node values are always below B. Hence each move is a $<B$-move. From Lemma 1, there is a finite number of $<B$-moves. Hence the computation is finite.