Termination detection for diffusing computations.

by

Edsger W.Dijkstra and C.S.Scholten

The following seems to capture the quintessence of a situation that
is not unusual in distributed processing. Consider a finite, directed
graph. (If the graph contains an edge from node A to node B , we call
B "a successor of A " and A "a predecessor of B ".) One node is called
"the gate" and we may assume that each node is "reachable from the gate", a
concept defined by the (usual) postulates:
1)   the gate is "reachable from the gate"
2)   if A is "reachable from the gate", so are all successors of A
3)   only those nodes are "reachable from the gate" that are so on account
of 1) or 2) .
In addition, the gate has an extra incoming edge, leading to it, so to speak,
from "the environment" --a symbolic predecessor of the gate-- .

A so-called "diffusing computation" is started when, via that extra
edge, the environment injects a "message" into the gate. Prior to that, all
nodes are assumed to be in their "neutral" state; after reception of its first
message, a node is free to send messages to its successors. It is this
feature that inspired the name "diffusing computations".

We shall confine our attention to computations for which it can
be proved that each node will send only a finite number of messages. For
such a computation eventually each node will reach the situation in which it
neither receives nor sends any more messages; when all nodes have reached that
state, the whole graph is as dead as a doornail and the diffusing computation .
is defined to have terminated.

Our problem is the design of a signalling scheme --to be superimposed
upon the diffusing computation proper-- such that, when the diffusing compu-
tation proper has thus terminated, the gate will eventually signal the fact
of this completion back to the environment. Besides a node's ability to re-

ceive messages from its predecessors and to send messages to its successors, we assume each node also able to receive "signals" from its successors and to send "signals" to its predecessors; in other words, each edge is assumed to be able to accommodate two-way traffic, but only messages of the computation proper in the one direction and signals in the opposite direction. We shall impose that in the total computation --i.e. from the moment that the initial message was injected into the gate until the gate emits the completion signal towards the environment-- each edge will have carried as many messages in the one direction as it has carried signals in the opposite direction.

For each edge we define the "deficit" as the number of messages transmitted along that edge, minus the number of signals returned along it. In the signalling scheme we propose, each node keeps track of

D = the sum of the deficits of its outgoing edges

(initially and finally zero). A node sending a message to one of its successors increases its D by 1 ; upon receipt of a signal from one of its successors it decreases its D by 1 . Note that a node records neither to which successors it has sent messages, nor from which successors it has received signals.

Furthermore each node has what we have dubbed a "cornet" (initially and finally empty). The name "cornet" has been chosen because, like in a pointed bag, one element contained in it enjoys a special status: whereas in a traditional bag all elements contained in it enjoy the same status, one of the elements in a non-empty cornet occupies the special position of being "the oldest element". (Whereas a stack is characterized by "last in, first out", a cornet is characterized by the much weaker "very first in, very last out".)

Each reception by B of a message from A causes the name of A to be added to B's cornet, which by this mechanism can be filled with names of predecessors of B . Note that, because the directed graph may contain merging (and even cyclic) paths, the cornet of B may contain the name of B's predecessor A several times. When the name of A is added to B's empty cornet, this occurrence of A's name in B's cornet is marked as

"the oldest element". The transmission of a signal from B to its prede-
cessor A is accompanied by the removal of one occurrence of A's name
from B's cornet.

As reception of a message from a predecessor and transmission of a
signal to a predecessor correspond --in the way just described-- to the only
changes of the contents of a node's cornet, and because a node has to return
to each of its predecessors a signal for each message received from that
predecessor, the current contents of a node's cornet summarize its signal-
ling obligations. The choice of predecessor to send a signal to is, by de-
finition, constrained by the condition that the name of the predecessor chosen
occurs still at least once in the cornet of the signalling node (because
otherwise it would be impossible to remove an occurrence of that name from
that cornet). The additional constraint --which distinguishes a cornet from
a standard bag-- is that from a cornet the element marked as "the oldest
element" may only be removed provided it was the only element in the cornet
(which, as a result of the removal, then becomes empty).

We define for each node C to be the "size" of its cornet. i.e. the
number of elements contained in its cornet. Note that for each node

C = the sum of the deficits of its incoming edges .

A node's freedom of sending messages and signals is only constrained
by the obligation to keep

P1: $C > 0$ or $D = 0$

invariant (and by the obvious obligation not to be infinitely lazy). This
completes the description of the signalling scheme.

<center>*   *   *</center>

In its role of message receiver and signal sender, each node guarantees
by its structure for each of its incoming edges a non-negative deficit. Hence
all deficits are non-negative and we have for each node

P2: $D \geq 0$.

The definition of C implies for each node

P3: $C \geq 0$.

    The sending of a message keeps P1 invariant for all nodes: for the sending node by virtue of its construction, for the receiving successor by virtue of the fact that its C is increased by 1 , and for all other nodes because their C's and D's remain unaffected.

    Also the sending of a signal keeps P1 invariant for all nodes: for the sending node by virtue of its construction, for the receiving predecessor by virtue of the fact that the accompanying decrease of its D by 1 can never destroy the truth of D = 0 on account of P2, and for all other nodes because their C's and D's remain unchanged.

    Because the sending of a signal includes for the sender $C := C - 1$, the invariance of P1 and P3 requires, by the axiom of assignment, that the act of signalling be guarded by

G: $(C - 1 > 0$ or $D = 0)$ and $(C - 1 \geq 0)$

which is equivalent to

G: $C > 1$ or $(D = 0$ and $C = 1)$.

    When the computation proper has terminated, no C is increased anymore; the ensuing signalling, as guarded in each node by G , will terminate because each sending of a signal decreases the sum of all the C's over the graph, a sum that is bounded from below on account of P3 . Hence, when the computation proper has terminated, the system will reach the "ultimate state" in which neither messages nor signals are anymore sent. From the fact that no more signals are sent we conclude that in the ultimate state non G holds for each node, which under the truth of P2 and P3 reduces to

$$C = 0 \text{ or } (C = 1 \text{ and } D > 0) . \tag{1}$$

Because the C of each node equals the sum of the deficits of its incoming edges, and (1) implies $C \leq 1$, we conclude that for each node the sum of the deficits of its incoming edges is ≤ 1 . Therefore the maximum deficit is ≤ 1 and no two edges with a positive deficit can point to the same node.

Moreover, (1) tells us that a node with $C = 1$ has $D > 0$, hence an outgoing edge with deficit $= 1$. In short: in the ultimate state the target of an edge with deficit $= 1$ is the source of a similar edge. Because the graph is finite and no two such edges can have the same target, we conclude our

Lemma. In the ultimate state edges with a positive deficit have a deficit $= 1$ and form node-disjoint (directed) cycles; the nodes on these paths have $C = 1$ and $D = 1$.

Since the edge from the environment to the gate lies on no cycle at all, it has in the ultimate state a deficit $= 0$; hence we conclude

Theorem 1. When the diffusing computation has terminated, the gate will eventually have returned a signal to the environment.

Up till now we have not made use of the fact that a node has a cornet: for all of the above a standard bag would have been sufficient. We shall now exploit the difference between a bag and a cornet in order to prove that the gate will not return a signal too soon, i.e. before the diffusing computation has terminated.

We call a node without "oldest element", i.e. one with $C = 0$, a "neutral node"; because C equals the sum of the deficits of its incoming edges, its incoming edges have a deficit $= 0$; on account of P1 it has $D = 0$. i.e. also its outgoing edges have a deficit $= 0$, and, hence, a node never sends a message while being neutral. A node that is not neutral is called "engaged". We can now formulate the invariant relation

P4:   the set of edges, each of which leads to an engaged node from its predecessor named by the oldest element of that engaged node, form a rooted tree --the so-called "engagement tree"-- for which the environment acts as the root.

Because only the gate can have the environment's name as its oldest element, the gate is the only possible descendant of the environment. Relation P4 is certainly true at the beginning, when the environment injects a message into the gate. It obviously remains true when node B becomes

engaged: B does so by receiving a message from one of its predecessors,
node A say, but at that moment node A was certainly engaged, and the
engagement tree is extended with a branch from A to B . Finally we have
to show that, when B returns to neutral, it was a leaf of the engagement
tree. When B returns to neutral, it reduces its C to zero; as it keeps
P1 valid, it does so at a moment at which its D is zero, i.e. at a moment
without outgoing edges with a positive deficit and, a fortiori, without
outgoing branches of the engagement tree. Hence at that moment B was a
leaf of the engagement tree.

Having thus established the existence of the engagement tree, we are
ready to prove

Theorem 2. When the gate returns a signal to the environment, the diffusing
computation has terminated.

Proof. When the gate returns a signal to the environment, it itself returns
to the neutral state; as the gate was the environment's only descendant, the
engagement tree is now empty, i.e. all nodes are in the neutral state, i.e.
no more messages are sent. (End of proof.)

Remark. On account of the close similarity between termination of the diffusing
computation and deadlock, it seems likely that our solution can be adapted to
the purpose of deadlock detection in networks. As we are more in favour of
deadlock prevention we did not pursue this possibility. (End of remark.)

Plataanstraat 5   prof.dr.Edsger W.Dijkstra
5671 AL   NUENEN , The Netherlands   Burroughs Research Fellow

Philips Research Laboratories   drs.C.S.Scholten
5656 AA Eindhoven, The Netherlands   Scientific Adviser