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) and 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.) Upon reception of a message from one of its predecessors, a node reacts typically in one of two ways:

a) either it absorbs the message, or

b) it sends a message to each of its successors.

A node, originally without successors, is modeled as a node with at least one successor --for instance: itself!-- that always opts for reaction a) . It is the possibility of reaction b) that inspired the name "diffusing computations". From now onwards we shall confine our attention to such computations for which it can be proved that each node will opt for reaction b) a bounded number of times. For such computations eventually each node will reach the state that all messages it had opted to transmit to its successors have been transmitted, each node is "waiting" for a next message that doesn't come and the whole graph is as dead as a doornail; when this stable state has been reached 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 computation proper has thus terminated, the gate will eventually signal the fact of this completion back to the environment. Besides a node's ability to receive messages from its predecessors and to send messages to its successors, we assume each node also able to receive "signals" from its successors and 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.

The implementation of the signalling scheme we propose requires for each node an integer variable called "count", initially and finally zero, and 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 contained 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 predecessor 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 signalling obligations. The choice of predecessor to receive a signal is, by definition, 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).

In order to determine when a node will return a signal to one of its predecessors the variable "count", which is initially equal to zero, is manipulated in the following fashion. Let a node receive a message and, as a result, send a message to x of its successors: in reaction a) $x = 0$, in reaction b) $x = k$ if k denotes the number of its successors. (We can regard reactions a) and b) as extreme cases and admit values of x such that $0 < x < k$ as well, i.e. the reaction in which a node sends a message to some of its successors.) Whenever a node reacts thus upon a message received from one of its predecessors, it increases its count by $k - x$; when it receives a signal from one of its successors, it increases its count by 1 . These are the only mechanisms that increase the count . The count is only decreased in multiples of k , more precisely, when the count $\geq k$ and the cornet is non-empty, the count will be decreased by k and the node will return a signal to a predecessor, the name of which is removed from the node's cornet as described in the preceding paragraph.

This completes the description of the signalling scheme. Note that a node neither records to which successors it has sent messages, nor from which successors it has received signals.

* * *

For each edge we define the "deficit" as the number of messages transmitted along that edge, minus the number of signal returned along it. In its role of message receiver and signal sender, the node to which an edge points guarantees that the deficit of that edge is non-negative. Furthermore, for each node the relation

P1: the size of its cornet = the sum of the deficits of its incoming edges which is true to start with, is left invariant: when the node receives a message both sides are increased by 1 , when the node sends a signal both sides are decreased by 1 .

For each node we define the "stock" as the number of messages, the transmission of which it has decided but not yet performed; by definition also stocks are non-negative. This concept enables us to formulate for each node the relation

P2: k *(the size of its cornet) = the sum of the deficits of its k outgoing edges + count + stock

which is clearly true to start with; of its invariance we can convince ourselves via the following table of increments, in which the rows correspond to the quantities incremented and the columns to the events:

	I	II	III	IV
k *(the size of its cornet)	k	0	0	-k
sum of the deficits	0	+1	-1	0
count	k - x	0	+1	-k
stock	x	-1	0	0

with I: receiving a message
II: sending a message
III: receiving a signal
IV: sending a signal.

From P2 we conclude, stocks and deficits being non-negative and $k > 0$, that count $\geq k$ implies that the cornet is non-empty. Hence, a count $\geq k$ is, all by itself, a sufficient condition for the transmission of a signal to a predecessor. Hence, when the diffusing computation proper has died out --i.e. all stocks = 0 -- , each count will eventually become less than the k of its node. (If a count is $\geq k$, it will be decreased by k and, because also a signal is sent, the count of one of its predecessors is increased by 1 ; this process, however, is bound to terminate, because each time the sum of all the deficits is decreased by 1 .) Let us call the state in which all stocks are zero and all counts less than the corresponding k's --i.e. the state in which neither messages, nor signals are anymore sent-- the "ultimate state". From P2 we conclude that in the ultimate state --because then we have count $< k$ and stock $= 0$ --

k *(the size of its cornet) $<$ the sum of the deficits of its k
outgoing edges + k

holds for each node. Because the right-hand side is at most equal to

k *(the maximum deficit of its outgoing edges + 1)

we can now, thanks to P1 and because $k > 0$, formulate our

Lemma. In the ultimate state each node has at least one outgoing edge with a deficit at least equal to the sum of the deficits of its incoming edges.

Up till now we have not made use of the fact that a node has a cornet: for the above a standard bag instead of a cornet would have been sufficient. We shall now exploit the difference between a cornet and a bag.

We call a node without "oldest element", i.e. one with an empty cornet, a "neutral node". Relation P1 tells us that the deficits of the incoming edges of a neutral node are all $= 0$, relation P2 tells us that the deficits of the outgoing edges of a neutral node, and its count and its stock are all $= 0$. A node that is not neutral is called "engaged". We can now formulate the invariant relation

P3: 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 P3 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, B was a leaf of the engagement tree. As long as B remains engaged, its cornet contains an occurrence of A's name as "the oldest element" and the deficit of the edge from A to B remains positive. When that deficit returns to zero, also the size of B's cornet returns to zero; from P2 we conclude that at that moment all B's outgoing edges have a deficit equal to zero. Because branches of the engagement tree correspond to edges with a positive deficit, none of B's outgoing edges can correspond at that moment to a branch of the engagement tree, and, hence, B was at that moment a leaf.

Having thus established the existence of the engagement tree, we are, at last, ready to prove our two theorems.

Theorem 1. When the gate returns a signal to the environment, the diffusing computation has died out.

Proof. When the gate returns a signal to the environment, it returns itself 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. (End of proof.)

Theorem 2. When the diffusing computation has died out, the gate will eventually return a signal to the environment.

Proof. We have already shown that, when the diffusing computation has died out, the system will reach the ultimate state. Let in the ultimate state y be the maximum value of all the deficits on the whole graph. Consider an edge with deficit y. Its target can have no other incoming edges with a positive deficit, for then the sum of the deficits of its incoming edges would exceed y and, according to our Lemma, it would have an outgoing edge with a deficit exceeding y, something which is impossible because y has been defined to be the maximum deficit. According to our Lemma and on account of our definition of y, that target has at least one outgoing edge with a deficit $= y$. As a result edges with deficit $= y$ form at least one cyclic path, such that none of the nodes on that path has other incoming edges with a positive deficit. Therefore, no node of such a cyclic path can be reached from the environment via branches of the engagement tree, which all have a positive deficit. Hence the nodes on such a cyclic path are all neutral, y therefore equals zero, therefore all nodes are neutral, and therefore the environment has received the signal from the gate when the ultimate state has been reached. (End of proof.)

Remark. On account of the close similarity 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.)

Acknowledgements. We are indebted to the members of The Tuesday Afternoon Club. In one session a particular diffusing computation and the problem of the detection of its termination were discussed; in its next session our first draft was subjected to a scrutiny that improved the final presentation. Finally we are indebted to Dr.P.M.Merlin from the Technion-Israel Institute of Technology who, in a lecture he gave at the University of Newcastle-upon-Tyne, inspired the problem.

Plataanstraat 5
5671 AL NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
BURROUGHS Research Fellow

Philips Research Laboratories
5656 AA EINDHOVEN
The Netherlands

drs.C.S.Scholten
Scientific Adviser