In reaction to Ernest Chang's "Deadlock Detection".

We consider a finite directed graph, without edges of which both ends coincide, and introduce the following terminology.

Let there be an edge directed from node A to node B ; then
-- this edge will be denoted as "edge AB "
-- it is an "outgoing edge" of node A
-- it is an "incoming edge" of node B
-- node A is a "predecessor" of node B
-- node B is a "successor" of node A .

Let "node B is reachable from node A" mean "node B is node A <u>or</u> node B is a successor of a node that is reachable from node A " .

Let "nodes A and B belong to the same strong component" mean "node A is reachable from node B <u>and</u> node B is reachable from node A ".

Let a "knot" be a strong component of more than one node, such that all successors of all nodes of the knot belong to the knot as well.

*   *   *

In each node the identities of its predecessors and of its successors are recorded; a node contains no further information about the shape of the graph. Furthermore each node can send messages to its successors and to its predecessors --note that, in spite of its being directed, each edge accommodates two-way traffic-- .

We can now ask ourselves the question whether we can design a diffusing computation, to be fired by an arbitrary node D , that will enable node D to detect whether or not it belongs to a knot. Because the answer is clearly "No" when node D has no successors --a circumstance that would be recorded in node D itself-- we can restrict ourselves to the case that D has at

least one successor. In the latter case the statement "node D belongs to
a knot" is equivalent to the statement "node D is reachable from each node
that is reachable from node D ".

From the last formulation it follows quite clearly that the determination
that node D does belong to a knot involves <u>all</u> nodes that are reachable from
node D . To establish for each node whether it is reachable from node D or
not, is the first task to which we turn our attention.

<div align="center">

*   *   *
*

</div>

For the sake of brevity we introduce the predicate Q1, defined as

$Q1(X)$ = "node X is, reachable from node D " .

In order to record for each node whether Q1 has been established, we
associate with each node X a variable --of type "natural number"-- that we
denote by Xn ; the "significance" of its value follows from the relation

$P1(X)$: $Xn < 1$ <u>or</u> $Q1(X)$,

a relation that the algorithm shall keep invariant for each node X . Be-
cause, to start with, nothing about reachability from node D has been es-
tablished, we assume the --in view of P1 : only safe!-- initialization
$Xn = 0$ for each node.

Nodes can send to their successors so-called M1-messages, meaning
"the recipient of an M1 message is reachable from node D ". We can jus-
tify this meaning by guarding the sending of an M1-message from node A
to node B by $An \geq 1$, because this guard implies $Q1(A)$ on account of
$P1(A)$. In order to suppress (logically harmless, but superfluous) further
transmission of M1-messages along that edge, the guard is further strengthen-
ed by a boolean --associated with that edge and denoted by M1AB -- which
is reset to false with the first transmission of an M1-message along that
edge. For each edge AB , the boolean variable M1AB is assumed to be stored
in node A and to be initialized at false.

The sending of an M1-message from node A to node B can now be

represented by the guarded command

$$An \geq 1 \text{ } \underline{and} \text{ } M1AB \rightarrow \{Q1(A) \text{ } \underline{and} \text{ } Q1(B)\}$$

$$M1AB := false;$$

$$\underline{if} \text{ } Bn \geq 1 \rightarrow skip$$

$$[] \text{ } Bn = 0 \rightarrow Bn := 1; \text{ } (\underline{A} \text{ } X: X \text{ is a successor of } B: M1BX := true)$$

$$\underline{fi} \text{ } \{P1(B)\}$$

The first two lines describe the sending; the following alternative construct describes the reception.

Note that --thanks to our initialization and the second alternative-- we have also the invariant

$$An \geq 1 \text{ } \underline{or} \text{ } \underline{non} \text{ } M1AB$$

for each edge AB (and that, as a result, the sending of an M1-message from node A to node B could have been guarded by M1AB all by itself).

The diffusing computation can be started by node D performing

$$Dn := 1; \text{ } (\underline{A} \text{ } X: X \text{ is a successor of } D : M1DX := true)$$

an act, which maintains all invariants. Because the graph is finite and each edge carries at most one message, the diffusing computation will die out; from the fact that then no more messages are sent, we conclude that in the final state we have again $\underline{non}$ M1AB for each edge AB . Observing for the edge AB the invariant

$$An = 0 \text{ } \underline{or} \text{ } Bn \geq 1 \text{ } \underline{or} \text{ } M1AB$$

we conclude that in the final state we have for each edge AB

$$An = 0 \text{ } \underline{or} \text{ } Bn \geq 1 \quad .$$

In the final state we can now conclude for each node X

$$Xn \geq 1 \text{ } \underline{or} \text{ } \underline{non} \text{ } Q1(X)$$

as follows. For a node X such that $\underline{non}$ Q1(X) holds it is obviously true. For a node X such that Q1(X) holds, there exists a path from D to X , and our previous result tells us that on that path a node with $n \geq 1$ can only be followed on that path by a successor with $n \geq 1$ as well. Hence,

from $Dn \geq 1$ we conclude $Xn \geq 1$, thereby establishing our latest result. Combining our latest result with P1 we conclude that upon termination we have for all nodes X

$$(Xn \geq 1) = Q1(X) \qquad .$$

<div align="center">* * *</div>

Recalling our interest in the truth of "node D is reachable from each node that is reachable from node D " and having dealt with the reachability from node D , we now turn our attention to the question whether node D is reachable from a node.

For the sake of brevity we introduce the predicate Q2 , defined as

$Q2(X) = $ "node D is reachable from node X "

and because we are only interested in recording for nodes whether Q2 has been established provided Q1 holds for them as well, we can use for each node X the same variable Xn by virtue of the relation

P2(X): $Xn < 2$ <u>or</u> Q2(X)   .

a relation that the algorithm shall keep invariant for each node. In view of our initialization $Xn = 0$ relation P2 holds to start with for each node. Because our algorithm as developed so far only assigns the value 1 to Xn's , that earlier part of the algorithm leaves P2 invariant. We change, however, the way in which node D starts the diffusing computation into

  $Dn := 2$; (<u>A</u> X: X is a successor of D : M1DX:= true)   ,

i.e. at the start node D records that node D is reachable from itself.

Nodes can send to their predecessors so-called M2-messages, meaning "node D is reachable from the recipient of an M2-message". We can justify this meaning by guarding the sending of an M2-message from node B to its predecessor A by $Bn \geq 2$, because that guard implies Q2(B), from which Q2(A) follows on account of the existence of the edge AB . In order to suppress logically harmless but superfluous transmissions of M2-messages, the guard is further strengthened by a boolean --associated with that edge and denoted by M2BA-- which is reset to false with the first transmission

of an M2-message along that edge. For edge AB the boolean variable M2BA is assumed to be stored in node B and to be initialized at false. The boolean M2BA is also used to suppress a superfluous M2-message from node B to a predecessor A for which $Q1(A)$ has not (yet) been established. Because the establishement of $Q1(A)$ takes place in node A and M2BA is stored in node B , we change in the transmission of an M1-message from node A to node B the reaction of the recipient by including the setting of M2BA.

The transmission of an M1-message from node A to node B can now be represented by the guarded command

    $An \geq 1$ <u>and</u> M1AB →

        M1AB:= false;

        M2BA:= true;

        <u>if</u> $Bn \geq 1$ → skip

        [] $Bn = 0$ → Bn:= 1; (<u>A</u> X: X is a successor of B: M1BX:= true)

        <u>fi</u>   •

The transmission of an M2-message from node B to node A can now be represented by the guarded command

    $Bn \geq 2$ <u>and</u> M2BA →

        M2BA:= false;

        <u>if</u> $An \geq 2$ → skip

        [] $An = 1$ → An:= 2

        <u>fi</u>   •

The above traffic keeps clearly $P1(X)$ <u>and</u> $P2(X)$ invariant for all nodes. Because the graph is finite and each edge transmits at most one M1-message and hence at most one M2-message, the diffusing computation will die out; from the fact that then no more messages are sent we conclude that in the final state we have for each edge AB

    <u>non</u> M1AB <u>and</u> ($Bn < 2$ <u>or</u> <u>non</u> M2BA)   •

Observing for the edge from node A to node B the invariant

    $An \neq 1$ <u>or</u> M1AB <u>or</u> M2BA   '

we conclude, combining the last two relations, that in the final state we have

$$An \neq 1 \text{ or } Bn < 2$$

for each edge AB . In the final state we can now conclude

$$Xn \geq 2 \text{ or non } (Q1(X) \text{ and } Q2(X))$$

as follows. For a node X such that non $(Q1(X)$ and $Q2(X))$ holds it is obviously true. For a node X such that $(Q1(X)$ and $Q2(X))$ holds, there exists a path from X to D , all nodes of which have $n \geq 1$. Our latest result tells us that on that path a node with $n \geq 2$ can only be preceded by a node with $n \neq 1$, hence $n \geq 2$. Because the path ends in node D with $Dn \geq 2$, we conclude $Xn \geq 2$, thereby establishing our latest result. Combining our latest result with $P2$ we conclude for the final state

$$\text{non } Q1(X) \text{ or } ((Xn \geq 2) = Q2(X)) \quad .$$

Summing up: in the final state we have
a node with $Xn = 0$ satisfies non $Q1(X)$,
a node with $Xn = 1$ satisfies $Q1(X)$ and non $Q2(X)$
a node with $Xn = 2$ satisfies $Q1(X)$ and $Q2(X)$.

Note that after firing --i.e. autonomously setting its Dn to 2 and the $M1DX$'s for its outgoing edges to true-- node D acts just like any other node of the graph. Note also that all the ways in which the $M1-$ and $M2-$message traffic may be mixed in time --a node X with $Xn = 2$ may still receive $M1$-messages-- did not need to be mentioned in our argument.

$$* \qquad * \qquad *$$
$$*$$

We now turn our attention to the third part of the algorithm, viz. the detection that D is reachable from all nodes that are reachable from D . For the sake of the inductive argument required we introduce a rooted tree T with node D as its root, with edges from the given graph and subspanning all nodes reachable from D . (The definition of "reachable from node D " implies the existence of such a tree.) Its edges will be called the "T-edges".

Let "the subtree of node X " be defined as that subtree of T of which node X is the root. For the sake of brevity we introduce the predicate Q3, defined as

Q3(X) = "node X is reachable from node D and node D is reachable from
all nodes of the subtree of node X " .

Because the subtree of node D is T itself, and T subspans by definition all nodes reachable from node D , the value of Q3(D) is the answer we are looking for. Because Q3(X) => (Q1(X) and Q2(X)) , we can use for the recording whether Q3 has been established for each node the same variable Xn by virtue of the relation

P3(X):   Xn < 3 or Q3(X)

which is clearly an invariant of our algorithm developed thus far.

Nodes can send to their predecessors so-called M3-messages meaning "Q3 holds for the sender of this M3-message". A node will send an M3-message with this meaning at most once, and only along its one and only incoming T-edge. With each T-edge we associate a boolean variable that will be used to record whether it has carried an M3-message. More precisely, with each T-edge AB we associate a boolean variable that is denoted by M3BA and assumed to be stored in node A . For the T-edge AB we shall keep the relation

$$\text{non } M3BA \text{ or } (Bn \geq 3) \tag{1}$$

invariant; it is initially true because M3BA is initialized with the value false.

For a node X we may conclude that Q3(X) holds, provided
1)   Q3(Y) holds for all "T-successors Y of X" --where "Y is a T-successor of X" means "the edge XY is a T-edge"-- and
2)   Q1(X) and Q2(X) holds --note that X could be a leaf of T !--

The sending of an M3-message from B to A along its one an only incoming T-edge can now be represented by the guarded command

$$Bn = 2 \text{ and } (A\ X: X \text{ is a T-successor of } B: M3XB) \rightarrow$$
$$Bn := 3;$$
$$M3BA := \text{true} \quad .$$

For node D , which has no incoming T-edge, this reduces to recording that Q3(D) has been established:

$$Dn = 2 \text{ and } (A\ X: X \text{ is a T-successor of } D: M3XD) \rightarrow$$
$$Dn := 3 \quad .$$

It is clear that with this addition the algorithm leaves P1(X) and P2(X) and P3(X) invariant for each node X . Note that for the M3BA associated with the T-edge AB we have the invariant stronger than (1):

$$M3BA = (Bn \geq 3) \quad .$$

The proof that upon completion we have

$$(Dn \geq 3) = Q3(D)$$

is left to the reader.

<p align="center">*   *   *</p>

We are now only left with the task of choosing the rooted tree T .
We choose for the T-edges those edges that carry an M1-message that is accompanied by the transition from n = 0 to n = 1 for its recipient.
It is clear that each node X with Q1(X) is reachable from D via such edges; it is also clear that each X with Q1(X) that differs from D has exactly one such incoming edge. Hence they satisfy all properties required from the edges of the rooted tree T .

With this choice for the tree T , the component of the guard controlling the M3-traffic

$$(A\ X: X \text{ is a T-successor of } B: M3XB)$$

presents some implementation problems, because the identities of their T-successors are not recorded in nodes. We therefore associate also a boolean variable M3BA with each non-T-edge AB , and replace the above guard component by

$$(A\ X: X \text{ is a successor of } B: M3XB)$$

and introduce a further mechanism that can set the M3BA to true for each
non-T-edge AB . Because upon arrival of an M1-message its recipient can
decide whether it received the M1-message via a T-edge or via a non-T-edge,
we can give each recipient of an M1-message via a non-T-edge the obligation
to return along that edge an M3-message meaning "the edge along which this
M3-message is carried is not a T-edge". The recipient of that M3-message
need not distinguish between the two meanings of an M3-message: in both
cases it sets the M3BA for that edge to true. With this protocol we would
eventually establish for each non-T-edge AB

$$\underline{non}\ Q1(A)\ \underline{or}\ (M3BA = (Bn \geq 1))\ .$$

In the case Q3(D) each edge that has carried an M1-message has to
carry an M2- and an M3-message as well. In the case $\underline{non}$ Q3(D) --charac-
terized by the existence of a node X with Q1(X) $\underline{and}$ $\underline{non}$ Q2(X)-- it would
be permissible to suppress all M3-traffic, and, from the point of efficiency,
the more we suppress, the better. We can achieve suppression by strengthening
the guards controlling the M3-signalling via T-edges by establishing for
a non-T-edge AB eventually

$$\underline{non}\ Q1(A)\ \underline{or}\ (M3BA = (Bn \geq 2))\ .$$

Also in the case Q3(D) this strengthening of the guards is permissible: in
that case --characterized by the $\underline{absence}$ of a node X with Q1(X) $\underline{and}$ $\underline{non}$
Q2(X)-- each M3BA of an outgoing non-T-edge of a reachable node will be
set to true, and our initially strengthened component of the guard control-
ling the M3-traffic along T-edges becomes equivalent to the original

$$(\underline{A}\ X\colon X\text{ is a T-successor of }\mathbf{B}\colon M3XB)\ .$$

This means, however, that along non-T-edges the M2- and M3-messages can
be combined into M23-messages.

Incorporating this optimization we get the following solution. For
a non-T-edge it maintains the invariant

$$An = 0\ \underline{or}\ (M3BA = (Bn \geq 2\ \underline{and}\ \underline{non}\ M1AB\ \underline{and}\ \underline{non}\ M2BA)$$

The start of the computation by node D :

    $Dn := 2$; $(\underline{A}\ X:\ X\ is\ a\ successor\ of\ D:\ M1DX := true)$

The sending of an M1-message from A to its successor B :

    $An \geq 1$ $\underline{and}$ $M1AB \rightarrow$

        $M1AB := false$;

        $M2BA := true$;

        $\underline{if}$ $Bn \geq 1 \rightarrow skip$

        $[\!]$ $Bn = 0 \rightarrow$ "record that AB is B's incoming T-edge";

                $Bn := 1$; $(\underline{A}\ X:\ X\ is\ a\ successor\ of\ B:\ M1BX := true)$

        $\underline{fi}$

The sending of an M2-message from B to A via B's incoming T-edge:

    $Bn \geq 2$ $\underline{and}$ $M2BA \rightarrow$

        $M2BA := false$;

        $\underline{if}$ $An \geq 2 \rightarrow skip$

        $[\!]$ $An = 1 \rightarrow An := 2$

        $\underline{fi}$

The sending of an M23-message from B to A via an incoming non-T-edge:

    $Bn \geq 2$ $\underline{and}$ $M2BA \rightarrow$

        $M2BA := false$;

        $\underline{if}$ $An \geq 2 \rightarrow skip$

        $[\!]$ $An = 1 \rightarrow An := 2$

        $\underline{fi}$; $M3BA := true$

The sending of an M3-message from B to A via B's incoming T-edge:

    $Bn = 2$ $\underline{and}$ $(\underline{A}\ X:\ X\ is\ a\ successor\ of\ B:\ M3XB) \rightarrow$

        $Bn := 3$;

        $M3BA := true$

For node D , which has no incoming T-edge, this reduces to recording that $Q3(D)$ has been established:

$$Dn = 2 \text{ } \underline{and} \text{ } (\underline{A} \text{ } X: X \text{ is a successor of } D: M3XD) \rightarrow$$
$$Dn := 3$$

Note that we did not prescribe that along T-edges the M3-messages
are preceded by the M2-messages.

*   *   *

The algorithm·described above is a diffusing computation that dies out.
In the case action from D is only required if it belongs to a knot, it
suffices --although it leaves "residues" in all other nodes-- . Alternatively
we can superimpose the detection mechanism for diffusing computations [1]:
upon receipt of the termination signal in D , the truth of

$$(Dn \geq 3) = Q3(D)$$

is guaranteed. (Note that each directed edge of the above, carrying messages
of the diffusing computation possibly in both directions, corresponds to two
edges --in opposite directions-- in the terminology of [1]. Note further
that node D of the above, which can receive messages, is $\underline{not}$ the "environ-
ment" of [1], and that a (virtual) "environment" has to be added to the graph
that fires D . Note further that the "engagement edges" from [1] are not
restricted to the T-edges from the above.) I shall not carry out the super-
position.

*   *   *

The above has been prompted by Ernest Chang [2], in which the author
--following R.C.Holt-- reduces deadlock detection to the question whether a
node --in a bipartite directed graph of "processes" and "resources"-- belongs
to a knot. Chang restricts himself to a diffusing computation of which ter-
mination is not necessarily detected in the case $\underline{non}$ Q3(D). (Chang's algorithm
--if correct-- recognizes $\underline{non}$ Q3(D) explicitly in the trivial case when
there exists a node X with Q1(X) , but without successors at all.)

The main incentive for writing the above was that while reading [2]
I could not decide whether its author had solved the problem or not; neither
could I decide --if he has solved it-- whether all complexity of his solution
is really required.

Warning. Those that would like to study Chang's [2] should be warned
that in his programs he attaches --without explaining his conventions-- a
syntactic role to indentation. (End of Warning.)

After spending a number of hours studying [2] I had placed mentally
so many question marks in margine that I decided that there was only one way
to understand his problem: I had to put his text aside and try to solve the
problem myself. Hence this text.

After solving the problem (without pencil and paper, within a few hours)
I spent two days on its first presentation, which was rejected because --though
much less so than [2]-- parts of my argument were still too operational. (I
had also made the mistake of borrowing some of Chang's --operationally de-
fined!-- terminology, such as "primary" and "secondary" edges.) The writing
of the above text --my second effort-- took me another three full days, days
that I consider as extremely well-spent. It forced me to separate my four
main concerns --Q1(X), Q2(X), Q3(X), and the definition of the rooted tree
T -- even more rigorously than I had done before; but even after that had
been achieved, finding the proper notations and deciding upon the proper
amount of essential detail turned out to be a major challenge. (I hope that
I have met it well.)

Note. For the operationally minded I point out that there is no objection
to regarding the tree T as completely defined, all through the computation.
The objection that, at the beginning of the computation, T is not "known"
yet, is totally irrelevant: with a modest amount of clairvoyance it is. (End
of Note.)

I must admit that, in the meantime I have lost all incentive to return
to [2] in order to try, once again, whether I can convince myself that Chang's
solution is correct. If I had to referee the paper, I would give the author
on this point the benefit of the doubt, but would recommend unconditional re-
jection of the paper in its current form, because it is much faster to solve
the problem yourself than to try to understand his text. I am also not tempted

to investigate to what extent the complexity of his program is due to a clumsy coding or to further optimizations that I did not incorporate --such as the combination of an M2- and an M3-message into a single M23-message along a T-edge, whenever possible-- , but he did not disentangle.

Acknowledgement. I am grateful to the members of the Tuesday Afternoon Club for explaining with refreshing clarity to me why my first version had to be rejected.

[1]   Dijkstra, Edsger W. and Scholten, C.S., Termination detection for diffusing computations. (EWD687a, submitted to ACTA INFORMATICA)

[2]   Chang, Ernest, Decentralized Deadlock Detection in Distributed Systems. (University of Toronto)

21 February 1979

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