Class Scribe

16/03/21

Ashish G kempwad (2019201091)
Naman Baheti (2018101099)
Kripa Anne Tharakan (20171159)

Chandy-Misra-Haas Distributed Deadlock Detection Algorithm

If a process makes a request for a resource which fails or times out, the process generates a *probe message* and sends it to each of the processes holding one or more of its requested resources.

Each probe message contains the following information:

- the *id* of the process that is blocked *(the one that initiates the probe message)*;
- the *id* of the process is sending this particular version of the probe message; and
- the *id* of the process that should receive this probe message.

When a process receives a probe message, it checks to see if it is also waiting for resources. If not, it is currently using the needed resource and will eventually finish and release the resource. If it *is* waiting for resources, it passes on the probe message to all processes it knows to be holding resources it has itself requested. The process first modifies the probe message, changing the sender and receiver ids.If a process receives a probe message that it recognizes as having initiated,it knows there is a cycle in the system and thus, **deadlock**.

Q) If a process is stuck in a deadlock, will the Chandy-Misra-Haas algorithm always tell that the process is in a deadlock?

Ans : in short, No! If a process is in deadlock but is not part of a cycle, then it will not be able to get the probe back. Hence Chandy-Misra-Haas fails.

Q) How do we know that algorithm is terminated?

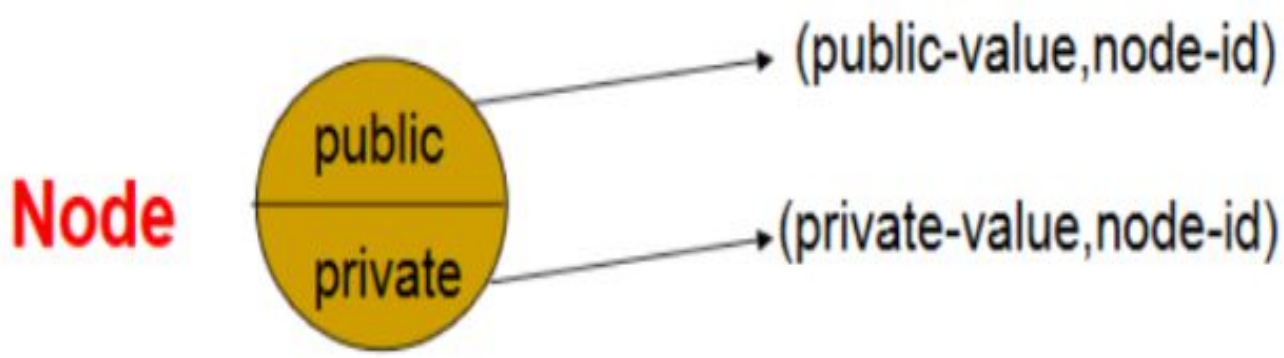
Ans: It could be that the termination detection algorithm has completed, the other way could be to see if the process comes to know that it is part of the cycle and hence involved in a deadlock(by receiving back the probe it sent), thereby releasing the resources it is holding.

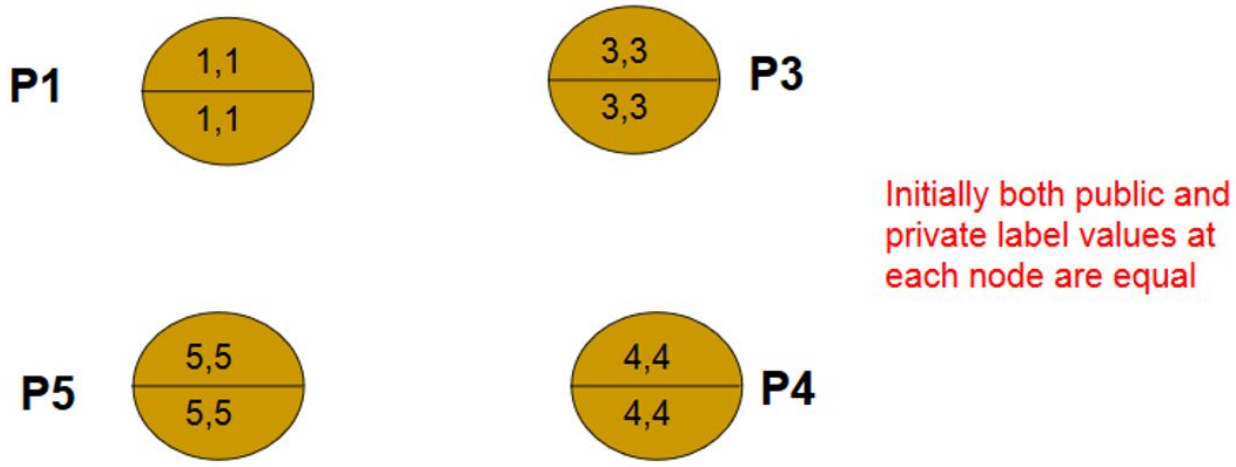
Mitchell and Merritt's Algorithm

The algorithm by Mitchell and Merritt [1984] presented in this section is as simple as the deadlock model for which it was defined. It is an edge-chasing algorithm in which probes are sent in the opposite directions of the edges of the WFG.
In the simplest case, a probe consists of a single natural number that is unique to the nodes in the WFG. When the probe comes back to its initiator, the initiator declares **deadlock**.

Intro

Each part of node forms the tuple of (public-value, node-id) & (private-value,node-id).





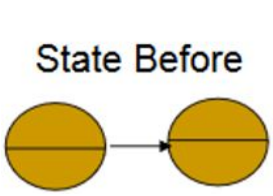
Algorithm :

There are four kinds of state transitions that happen along the span of algorithm namely, Block, Transmit, Detect, Activate.

1. BLOCK

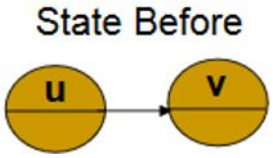
1)Block state happens when the one process requests for a resource that is being held by some other process and thus gets blocked.
2)Label change occurs in this step for the waiting process.
3)Both the public and the private value of the waiting process gets changed to a value that is greater than its previous value and greater than the public label of the process being waited on.

2. ACTIVATE

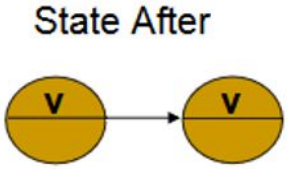


1) The process that was initially blocked for a resource gets the resource, then this state takes place.
2)The only change that happens is that the edge between the process is completely removed.

3. TRANSMIT

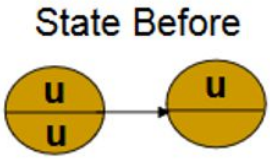


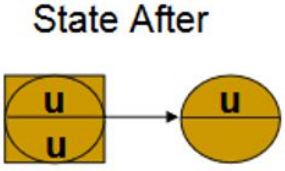
If u < v



1)Waiting process finds that the process it was waiting on has a public value greater than it.
2)Then,the waiting process will change its public label equal to the public label of the process upon whom it is waiting.
3)Waiting process's private label remains unchanged.

4. DETECT



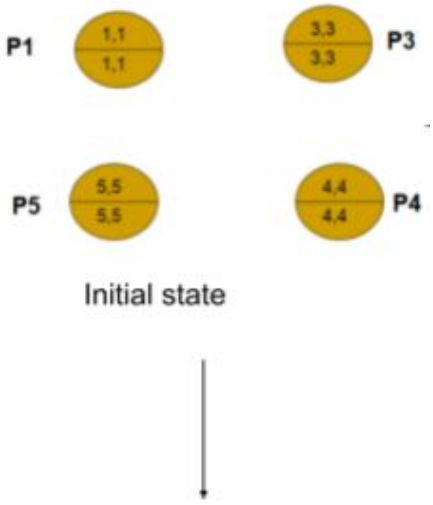


1)This state occurs when the process sees its own public label come back to itself.
2) When a process reads a public label of waiting upon process and finds that the public level value of waiting upon process is equal to its own public label value then it determines that a cycle exists and declares the deadlock.

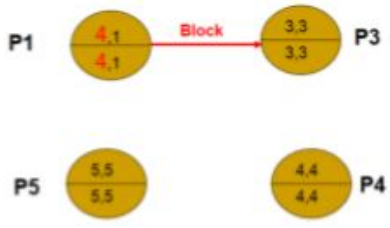
Q: The values keep on increasing, can there be a bounding error?

Ans : The algorithm is silent about that. Let's understand that a bit by going through an example.

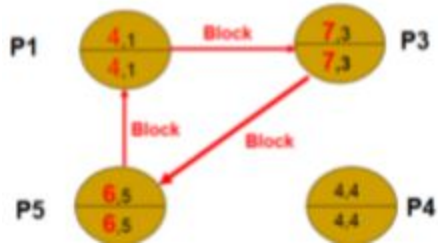
Step 1)



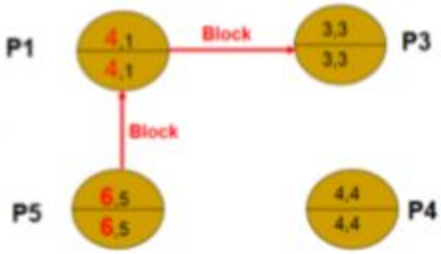
Step 2) P1 waiting on P2, hence block.



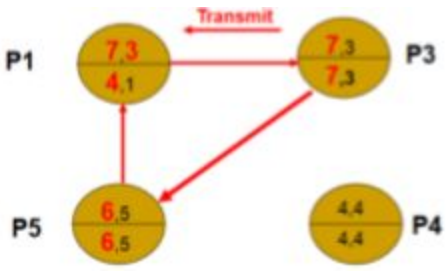
Step 3) P5 waiting on P1, hence block.



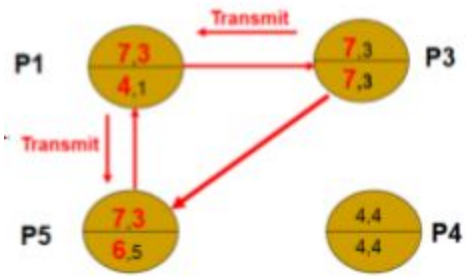
Step 4) P3 waiting on P5, hence block.



Step 5)P3 has higher public value than P1, hence P3 transmits public label to P1.



Step 6)P1 has higher public value than P5, hence P1 transmits public label to P5.



Here P3 finds that it has the same public label as P5, this leads to cycle and hence deadlock is detected.