Chandy-Misra-Haas Algorithm for the AND Model

- Chandy-Misra-Haas's distributed deadlock detection algorithm for AND model is based on edge-chasing.
- The algorithm uses a special message called *probe*, which is a triplet (i, j, k), denoting that it belongs to a deadlock detection initiated for process P_i and it is being sent by the home site of process P_j to the home site of process P_k.
- A probe message travels along the edges of the global WFG graph, and a deadlock is detected when a probe message returns to the process that initiated it.

- A process P_j is said to be *dependent* on another process P_k if there exists a sequence of processes P_j, P_{i1}, P_{i2}, ..., P_{im}, P_k such that each process except P_k in the sequence is blocked and each process, except the P_j, holds a resource for which the previous process in the sequence is waiting.
- Process P_j is said to be *locally dependent* upon process P_k if P_j is dependent upon P_k and both the processes are on the same site.

Data Structures

- Each process P_i maintains a boolean array, *dependent*$_i$, where *dependent*$_i$(j) is true only if P_i knows that P_j is dependent on it.
- Initially, *dependent*$_i$(j) is false for all i and j.

Algorithm

The following algorithm determines if a blocked process is deadlocked:

- if P_i is locally dependent on itself then declare a deadlock else for all P_j and P_k such that
  1. P_i is locally dependent upon P_j, and
  2. P_j is waiting on P_k, and
  3. P_j and P_k are on different sites,send a probe (i, j, k) to the home site of P_k
- On the receipt of a probe (i, j, k), the site takes the following actions: if
  1. P_k is blocked, and
  2. $dependent_k$(i) is false, and
  3. P_k has not replied to all requests P_j,

then

begin

$dependent_k$(i) = true;

if k=i

then declare that P_i is deadlocked

else for all P_m and P_n such that

(a') P_k is locally dependent upon P_m, and

(b') P_m is waiting on P_n, and

(c') P_m and P_n are on different sites,

send a probe (i, m, n) to the home site of P_n

end.

- A probe message is continuously circulated along the edges of the global WFG graph and a deadlock is detected when a probe message returns to its initiating process.

Performance Analysis

- One probe message (per deadlock detection initiation) is sent on every edge of the WFG which that two sites.
- Thus, the algorithm exchanges at most $m(n-1)/2$ messages to detect a deadlock that involves m processes and that spans over n sites.
- The size of messages is fixed and is very small (only 3 integer words).
- Delay in detecting a deadlock is O(n).