Ricart-Agrawala Algorithm for Mutual Exclusion

program mutex
define m: msg
try,want,in: boolean
N: integer (number of acknowledgements)
t: timestamp
A: array [0..n-1] of boolean

initially try=false (turns true when process wants to enter CS)
want=false (turns false when process exits CS)
N=0
A[k] = false for all k: (0 ≤ k ≤ n - 1)

do try m := (i, req, t);
forall j : j NOT = i :: sendmtoj;
try := false; want := true

G: (m.type = request) AND ( NOT want OR m.st < t') send (i, ack, t1) to m.sender
G: (m.type = request) AND (want AND m.ts > t) A[sender] := true
G: m.type = ack N = N + 1;
G: N = N - 1 in := true;
(process enters CS)
want := false
G: in AND NOT want in := false; N := 0;
for all k : A[k] := send (i, ack, t1) to k;
for all k : A[k] := A[k] := false:

od