(Ex.1) - LEONARDO SERICCI 26/09/2021

(A1) Yes.

Look at the case where the processor i is woken up by a message in it's input buffer; Because of NON-DETERMINISM, it can decide to execute firstly the M_i guard, setting M to the received value. this is the case where L_i guard will never be execute.

(A2) No.

If happen the case described in (A1), L_i guard will never be executed, so the Processor i will never set $M=i$, so it won't compete for the ELECTION.

(A3) Yes. In the case it's woken up by a msg $J \leq i$.

(A4) LOCAL VARIABLES INITIALIZATION: PARTECIPANT = FALSE

L_i: { PARTECIPANT == FALSE }

    PARTECIPANT = TRUE
    SEND <i> TO NEXT IN THE RING

M_i: { MESSAGE HOLDING $J \in N$ IS READY }

    RECEIVE MESSAGE
    if (J>i)
        PARTECIPANT = TRUE
        SEND <J> TO NEXT IN THE RING
    if ((J<i) AND (PARTECIPANT = FALSE))
        PARTECIPANT = TRUE
        SEND <i> TO NEXT IN THE RING
    if (J==i)
        PARTECIPANT = FALSE
        "I'M THE LEADER"

(A5) the difference between the two can be seen in the situation described in (A1) in the first algorithm, if Processor i is woken up by J, it can set $M=J$, forward the value <J>, then $M \neq 0$ and the guard L_i: {$M == \emptyset$} won't realize at all so the ID i won't candidate.

In the second algorithm, if processor i doesn't wake up spontaneously, it will compare it's ID i with the value J received, sending the greatest.

We can conclude that in CASE 1 may be elected a node which ID is not maximum, but isn't a problem because every node will agree that it is the leader. ⇒ BOTH ALGORITHMS solve the LEADER ELECTION problem.

(EX 2) · The Alghoritm works, but it can end-up in a loop.

· Let's analyze the number of rounds and messages in the BEST, WORST and AVERAGE cases.

· We <u>won't</u> consider messages sendt in the PROCLAMATION PHASE.

(1) WORST CASE: if each node keeps having the same output in the coin toss the algorithm continously repeat the same round.

$$\boxed{\begin{array}{l} \rightarrow \#ROUND_{WORST} = \infty \\ \rightarrow \#MESSAGES_{WORST} = \infty \end{array}}$$

(2) BEST CASE: if just one node output 1 in the first coin toss.

$$\boxed{\begin{array}{l} \rightarrow \#ROUND_{BEST} = 1 \\ \rightarrow \#MESSAGES_{BEST} = \underset{|V|}{n}\,\underset{|N(v)|}{(n-1)} = \Theta(n^2) \end{array}}$$

// each node send one message to each of it's n-1 neighbors

(3) AVERAGE CASE: if we assume that in each round the number of nodes that succeed is the ROUNDED DOWN HALF, you will terminate in $\log_2(n)$

$$\boxed{\begin{array}{l} \rightarrow \#ROUND_{AVG} = \log_2(n) \\ \rightarrow \#MESSAGES_{AVG} \leq \sum_{i=1}^{\log_2 n} \left(n/2^{i-1}\right)(n-1) \end{array}}$$

· The number of msgs is less equal because we assume the ROUND DOWN