

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

(A1) Yes.

Look at the case where the processor i is woken up by a message in its 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 executed.

(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 $S \leq i$.

(A4) LOCAL VARIABLES INITIALIZATION: $\text{PARTICIPANT} = \text{FALSE}$

$L_i : \{ \text{PARTICIPANT} = \text{FALSE} \}$

$\text{PARTICIPANT} = \text{TRUE}$

$\text{SEND } <i> \text{ TO NEXT IN THE RING}$

$M_i : \{ \text{MESSAGE HOLDING SET } N \text{ IS READY} \}$

RECEIVE MESSAGE

$\{ S > i \}$

$\text{PARTICIPANT} = \text{TRUE}$

$\text{SEND } <S> \text{ TO NEXT IN THE RING}$

$\{ (S < i) \text{ AND } (\text{PARTICIPANT} = \text{FALSE}) \}$

$\text{PARTICIPANT} = \text{TRUE}$

$\text{SEND } <i> \text{ TO NEXT IN THE RING}$

$\{ S = i \}$

$\text{PARTICIPANT} = \text{FALSE}$

"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 S , it can set $M = S$, forward the value $<S>$, then $M = 0$ 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 its ID i with the value j received, sending the greatest.

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

(EX 2) The Algorithm 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 won't consider messages sent in the PROCLAMATION PHASE.

(1) WORST CASE: If each mode keeps having the same output in the coin toss the algorithm continuously repeat the same round.

$$\rightarrow \# \text{ROUND}_{\text{WORST}} = \infty$$

$$\rightarrow \# \text{MESSAGES}_{\text{WORST}} = \infty$$

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

$$\rightarrow \# \text{ROUND}_{\text{BEST}} = 1$$

$$\rightarrow \# \text{MESSAGES}_{\text{BEST}} = m(m-1) = \Theta(m^2)$$

// each mode send one message to each of its $m-1$ neighbors
 N $|N(V)|$

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

$$\rightarrow \# \text{ROUND}_{\text{AVG}} = \log_2(m)$$

$$\rightarrow \# \text{MESSAGES}_{\text{AVG}} \leq \sum_{i=1}^{\log_2 m} \left(\frac{m}{2^{i-1}} \right) (m-1)$$

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