Distributed Computing PWr, WIT, 2021

Prof. Mirosław Kutyłowski

3- Leader Election on a ring

Leader election

Termination condition:

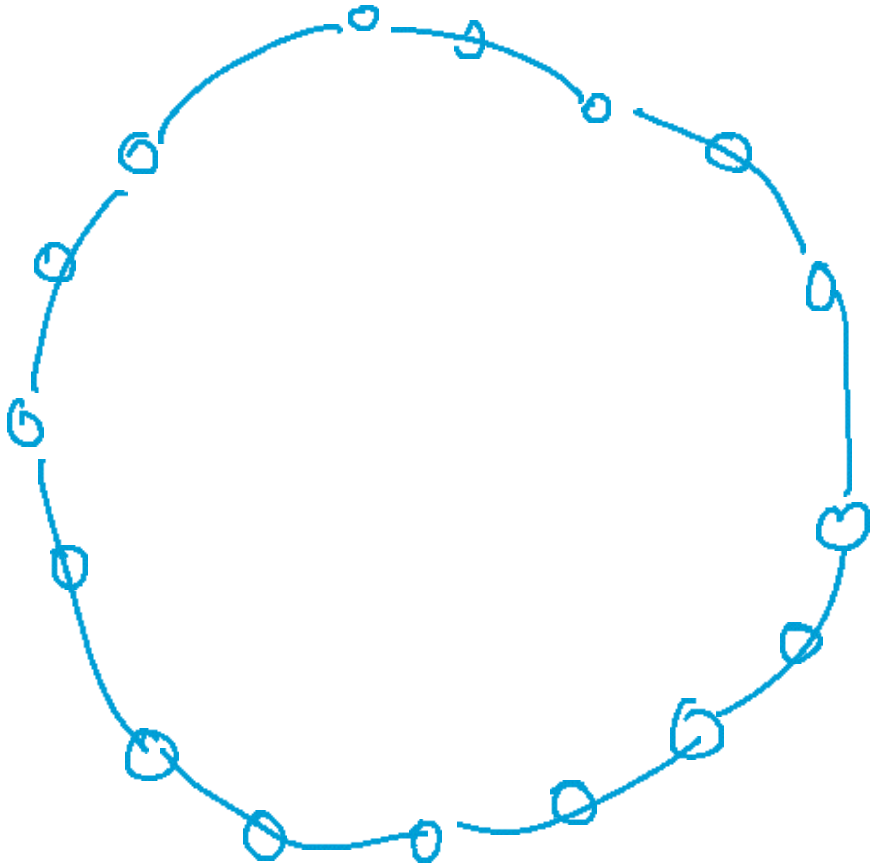
Exactly one node in the state “I am the leader”

Every other node: “I am not a leader”

Option

everybody knows the ID of the leader

Ring



Algorithm types

Anonymous: all nodes start with the same state

Uniform: number of nodes unknown

Non-uniform: number of nodes n is known to each node

Deterministic LE in a ring

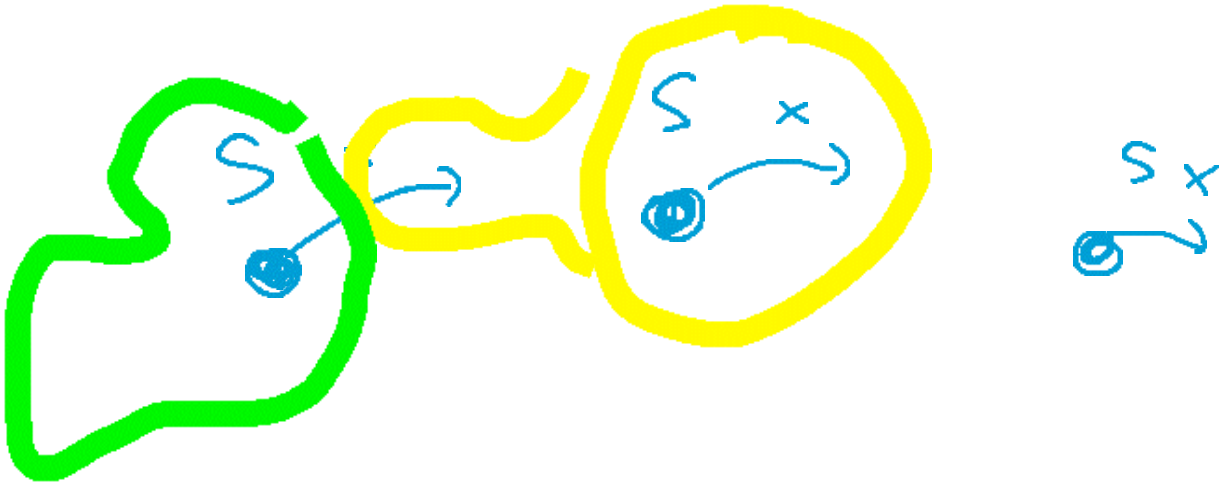
Theorem 3.5 (Anonymous Leader Election). *Deterministic leader election in an anonymous ring is impossible.*

Lemat Stan po t krokach jest wszędzie taki sam

Indukcja:

t = 0 ok

t+1:



Via highest ID

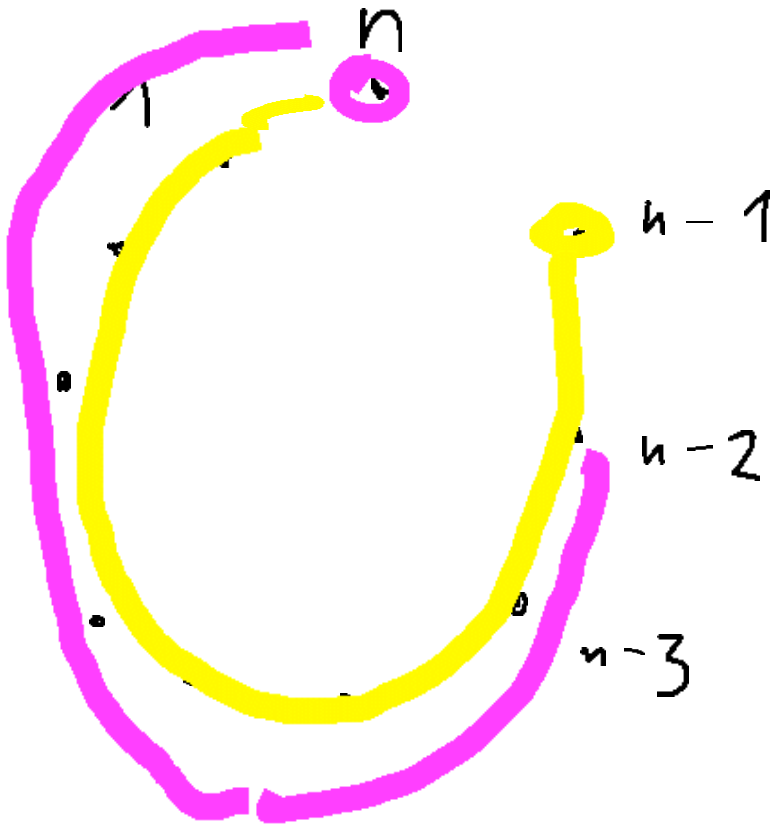
Algorithm 3.6 Clockwise Leader Election

1: **Each node** v executes the following code:
2: v sends a message with its identifier (for simplicity also v) to its clockwise neighbor.
3: v sets $m := v$ {the largest identifier seen so far}
4: **if** v receives a message w with $w > m$ **then**
5: v forwards message w to its clockwise neighbor and sets $m := w$
6: v decides not to be the leader, if it has not done so already.
7: **else if** v receives its own identifier v **then**
8: v decides to be the leader
9: **end if**

Complexity

time complexity: n

Message complexity: $O(n^2)$



Radius growth

Algorithm 3.8 Radius Growth

1: **Each node** v does the following:
2: Initially all nodes are *active*. {all nodes may still become leaders}
3: Whenever a node v sees a message w with $w > v$, then v decides to not be a leader and becomes *passive*.
4: Active nodes search in an exponentially growing neighborhood (clockwise and counterclockwise) for nodes with higher identifiers, by sending out *probe* messages. A probe message includes the ID of the original sender, a bit whether the sender can still become a leader, and a time-to-live number (TTL). The first probe message sent by node v includes a TTL of 1.
5: Nodes (active or passive) receiving a probe message decrement the TTL and forward the message to the next neighbor; if their ID is larger than the one in the message, they set the leader bit to zero, as the probing node does not have the maximum ID. If the TTL is zero, probe messages are returned to the sender using a *reply* message. The reply message contains the ID of the receiver (the original sender of the probe message) and the leader-bit. Reply messages are forwarded by all nodes until they reach the receiver.
6: Upon receiving the reply message: If there was no node with higher ID in the search area (indicated by the bit in the reply message), the TTL is doubled and two new probe messages are sent (again to the two neighbors). If there was a better candidate in the search area, then the node becomes passive.
7: If a node v receives its own probe message (not a reply) v decides to be the leader.

Complexity:

Time: O(n)

$$O(1) + O(2 \cdot 1) + O(2 \cdot 2 \cdot 1) + \dots$$

$$O(n) = O(n)$$

Message: O(n log n)

poprzednio $O(n^2)$

ile liderów lokalnych:

2^i - $n/2^{i+1}$

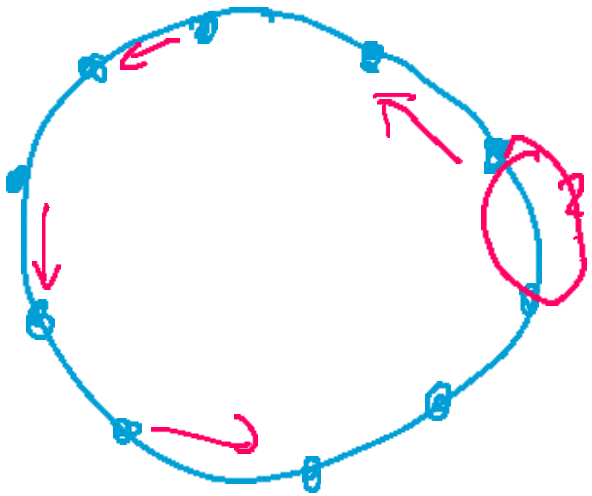
$\frac{n}{2^{i-1}} \cdot O(2^{i+2})$

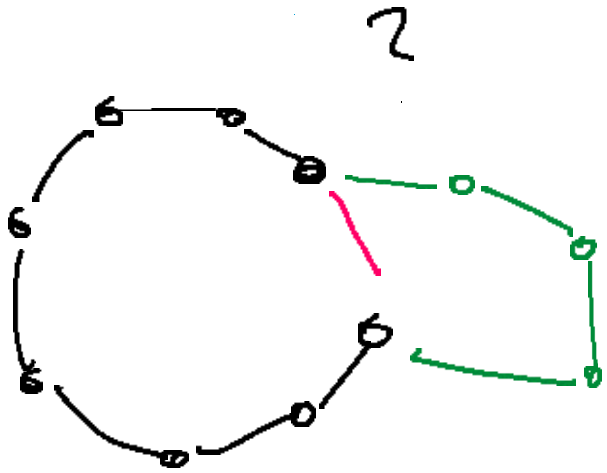
Lower bounds

Adversary: determines the order of delivering messages

Open schedule: at least one edge not used

Fact: the result still unknown if the schedule is open



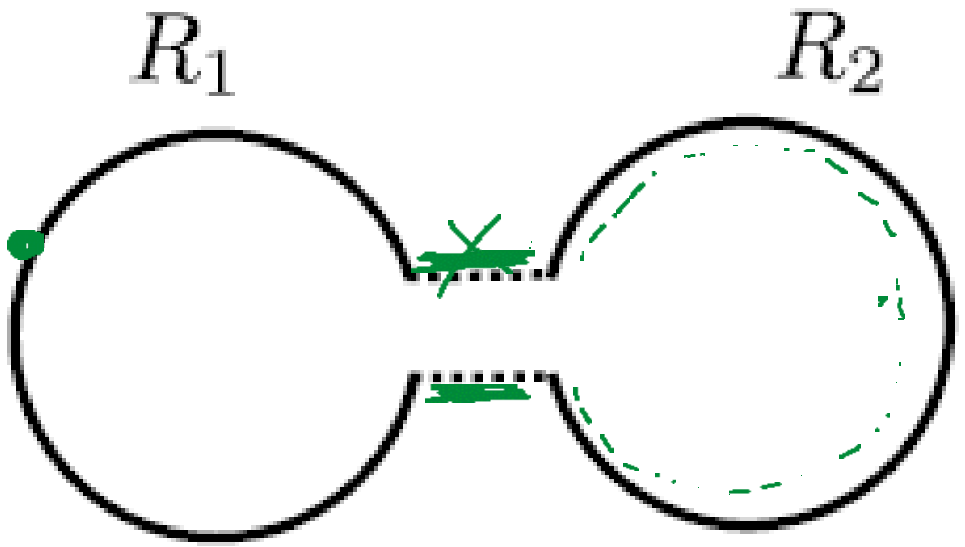


Lower bound result

Lemma 3.15. *Any uniform leader election algorithm for asynchronous rings has at least message complexity* $M(n) \geq \frac{n}{4}(\log n + 1)$.

proof

$$M(n) \overset{2.}{=} M\left(\frac{n}{2}\right) + \frac{n}{4}$$

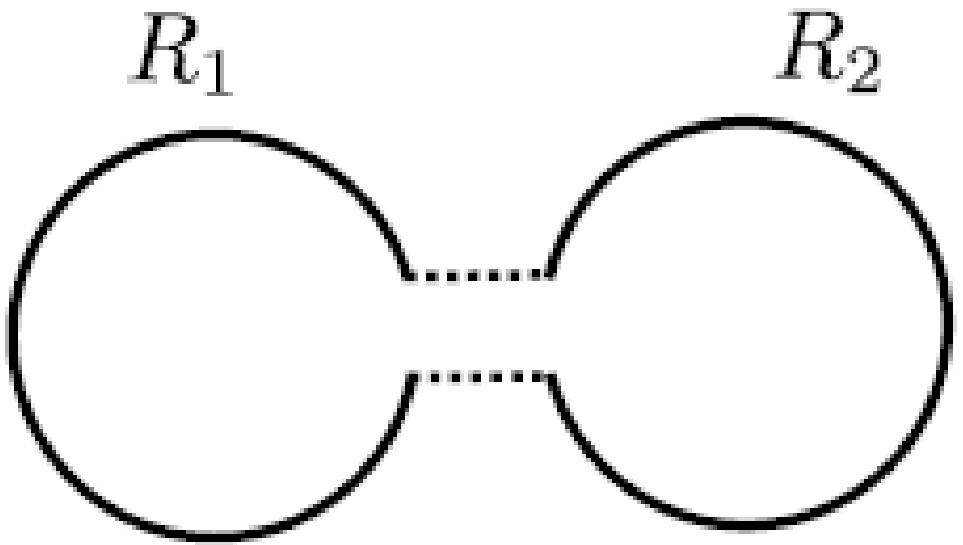


The rings R_1, R_2 are glued together at the

rozwiązanie

$$M(n) = O(n \log n)$$

proof



The rings R_1, R_2 are glued together at the

proof

proof

proof