Properties

- An *assertion* is a statement about a global state, e.g., at most one process is accessing the printer.
- An assertion is *invariant* if it holds in *every* global state of *every* run.
- Proving that an assertion is invariant usually requires *induction* on the number of rounds.
Environment

- Processes can receive *inputs* and generate *outputs*, but who supplies them?
- add the environment process *env*
- *env* can be used also generate *failure patterns*

We say that one system *simulates* another if, using the same *env*, the two systems produce the same outputs (at the same rounds.) Simulation relations are also usually proven by induction on the number of rounds.
**Complexity Measures**

- **Time complexity**: measures in terms of the number of rounds it takes until all outputs are produced OR until all processes halt. If the system allows variable *wake-up* times, time is measured from the round of the first *wakeup*.

- **Communication Complexity** measures either in terms of the number of messages sent until all outputs are produced (or all processes halt), or in number of total bits sent.
Processes *do not know* their ids. They each have a variable *status*. A *solution* to the problem is a protocol for the processes such that eventually one, and only one, process has (and outputs) *status*=leader.
Versions and Impossibility

- All non-leaders eventually output non-leader;
- The number $n$ is known/unknown;
- The ring is uni/bi-directional;
- Processes are identical or have unique ids
Versions and Impossibility

- All non-leaders eventually output non-leader;
- The number $n$ is known/unknown;
- The ring is uni/bi-directional;
- Processes are identical or have unique ids

**Theorem:** If the ring is bidirectional and all processes are identical, there is no deterministic solution to leader election

$\implies$ no solution for unidirectional rings
**Possible and Impossible**

- All non-leaders eventually output non-leader;
- The number $n$ is known/unknown;
- The ring is uni/bi-directional;
- Processes are identical or have unique ids

**Theorem:** If the ring is bidirectional and all processes are identical, there is no deterministic solution to leader election

$\implies$ no solution for unidirectional rings

$\implies$: A solution must **break the symmetry**, by process ids or randomization

Restrict to non-randomized solutions, and assume that each process $i$ has a unique id $u_i$
The LCR Algorithm

Assumption: The ring is unidirectional; Only leader produces output; \( n \) is unknown

Solution: At each round, each process sends the maximal id it had seen that it didn’t send before. If it receives own id, it becomes leader

A state of process \( i \) consists of:

- \( \max_i \) an id, initially \( u_i \)
- \( \text{status}_i \in \{\text{unknown, leader}\} \) initially \( \text{unknown} \)

The transition function of process \( i \) is:

- (send) \( \text{send}(\max_i) \)
- (receive) \( \text{receive}(u) \)
- (local) \( \text{if } u > \max_i \text{ then } \max_i := u \)
  \( \text{if } \max_i = u_i \text{ then } \text{status} := \text{leader} \)
Let $i^*$ be the process with maximal id $u^*$. Then:

**Claim 1:** For every $r = 0, \ldots, n - 1$, after $r$ rounds $msg_{i^*+r} = u^*$

**Proof:** By induction on $r$. Base case is trivial. Assume true for $r' \in \{0..n - 2\}$. Then at round $r' + 1$ process $i^* + r'$ sends $u^*$. Since $u^*$ is the maximal in the ring, after $r' + 1$ rounds $msg_{i^*+r'+1} = u^*$

**Corollary:** After $n$ rounds, $status_{i^*} = \text{leader}$
Let $i^*$ be the process with maximal id $u^*$. Then:

**Claim 1:** For every $r = 0, \ldots, n - 1$, after $r$ rounds $msg_{i^* + r} = u^*$

**Corollary:** After $n$ rounds, $status_{i^*} = \text{leader}$

**Claim 2:** For every $j \neq i^*$, its always the case that $status_j \neq \text{leader}$

**Proof:** By induction on the rounds, prove that for every $r = 0, \ldots, n - 1$, after $r$ rounds, for no $j$ it is the case that $msg_j$ is the $r^{th}$ smallest id among $u_1, \ldots, u_n$. Consequently, the only id that reaches its owner is $u^*$
Complexity and Variants

- The time complexity is $n$ rounds until a leader is announced. The message complexity is $O(n^2)$.
- The algorithm is correct when processes send only messages they didn’t previously send. Worst time complexity remains, but best time improves.
- If halting is required, then leader can send a leader message around the ring, costing $n$ rounds (and $n$ message, leaving message complexity intact.)
- If variable start times are assumed, then algorithm still works, with possible cost of $n - 1$ rounds.
- The algorithm is not resilient to failures.
Assumptions: LCR with bidirectionality

Protocol consists of phases

phase $p$ takes $2 \cdot 2^p$ rounds

in a phase, active processes send their id in both directions to their $2^{p-1}$ consecutive neighbours

ids get relayed only if $> \text{highest id seen}$

A process whose id returns from both directions ($2^{p-1}$ steps) remains active

A process that receives its own id in outbound message becomes leader
Communication Complexity:

- First phase has at most $4n$ messages
- After phase $p$, only one of $2^{p-1} + 1$ processes “survives”, and generates at most $4 \cdot 2^p$ messages
- Thus, at each phase $p > 1$, there are at most
  \[
  4 \left( 2^p \cdot \left\lfloor \frac{n}{2^{p-1} + 1} \right\rfloor \right)
  \]
  messages
- There are at least $1 + \lceil \log n \rceil$ rounds
- Thus, upper bound of $8n(1 + \lceil \log n \rceil) = O(n \log n)$
Analysis of Protocol

<table>
<thead>
<tr>
<th>Phase</th>
<th>Out rounds start</th>
<th>In rounds start</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>22</td>
</tr>
<tr>
<td>...</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>$k$</td>
<td>$2^k - 2$</td>
<td>$3 \cdot 2^{k-1} - 2$</td>
</tr>
</tbody>
</table>

- A message need only contain a process id
- If $P_i$ receives $u$ in round $k$, it knows what to do with it
- $P_i$ becomes leader when it receives its outbound message (as an outbound message) (it will receive it from both directions at the same round)
Time Complexity

Each phase $p$, but the last one, takes $2 \cdot 2^{p-1}$ rounds. The last phase takes $n$ rounds. There are at most $1 + \lceil \log n \rceil$ phases, and therefore

$$\sum_{p=1}^{\lceil \log n \rceil} 2^p + n$$ rounds. Since

$$\sum_{p=1}^{\lceil \log n \rceil} 2^p = 2^{\lceil \log n \rceil + 1} - 1$$

total time complexity is:

$$n + 2^{\lceil \log n \rceil + 1} - 1 \approx n + 2 \cdot 2^{\log n} = 3 \cdot n$$
A Randomized Solution

Solution from Itai and Rodeh, in The Lord of the Ring or Probabilistic Methods for Breaking Symmetry in Distributed Networks, *Tech. report RJ 3110, IBM San Jose, 1981*

Previous solutions broke symmetry was broken by ids

Here processes are *fully symmetric*

$n$ (the size of the ring) in known;

There exists a number $K$ shared by all the processes (we can take $K = n$)

Ring is unidirectional
Initially, all processes are *initiating*

At the beginning of each phase, an initiating process $P_i$ randomly draws $K_i \in [1..K]$ with equal probability which is its id for phase. Non-initiating process have id 0

Each process sends its id along the ring, and relays any id it receives

At the end of $n$ rounds, each process inspects the $n$ ids; If there is some id $u > 0$ that is held by a single process, the process who generated the maximal such id in this phase is *leader*

Otherwise, only processes who generated the maximal (non-unique!) id are *initiating*

There are executions with no *leader*
An Example of a Run

\( n = 5 \):

- At \textit{first} phase:
  \[ u_1 = u_2 = 3; \quad u_3 = 2; \quad u_4 = u_5 = 1; \]
  Only \( P_1 \) and \( P_2 \) remain initiating

- Assume that at the \textit{second} phase:
  \[ u_1 = u_2 = 3; \quad (u_3 = u_4 = u_5 = 0) \]

\( P_1 \) and \( P_2 \) remain only initiators

- The probability of the event that both processes draw the same numbers is \( \frac{1}{5} \). The probability that they repeatedly do with \( \ell \) consecutive times is \( \left( \frac{1}{5} \right)^\ell \). Thus, the protocol terminates with probability 1
Complexity of IR Protocol

- If the probabilities of the random choices are all \( p = \frac{1}{2} \), then at each phase we expect \( p \) of the processes to lose.

- Expected number of rounds is \( \mathcal{O}(n \log n) \), and the expected number of messages is \( \mathcal{O}(n^2 \log n) \).

- Complexity can be improved E.g., limit range of values to number of initiators. Expected number of phases is \( \mathcal{O}(3 \cdot n) \), leading to \( \mathcal{O}(n^2) \) messages.
Comments on Leader Election

- If there is a leader, then $n$ can be computed. If $n$ is known, then a leader can be elected. But can a leader be elected and $n$ computed in a ring where neither is à-priori known?

- How much do we depend on the lock-step assumptions?

- The non-randomized protocols are based only on comparison between ids. The IR protocol also uses counting, has better complexity. Can comparison based protocols do better?