A Framework for Developing Distributed Protocols with Event-B/Rodin:

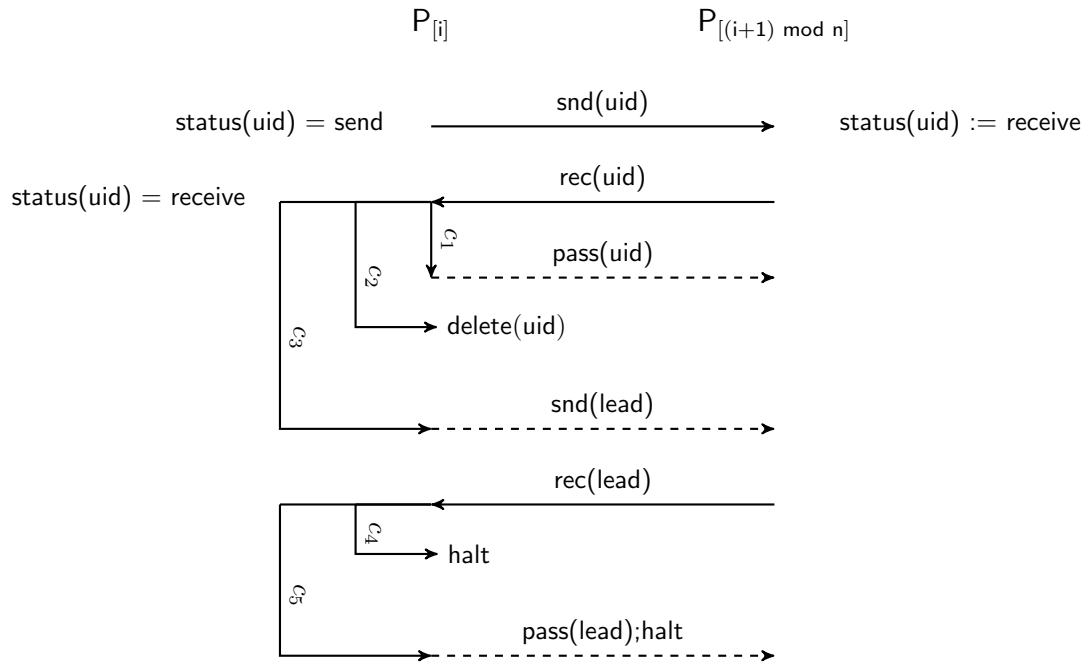
Graphical Language Notes

Paulius Stankaitis and Alexei Iliasov

Leader Election Protocol (Synchronous Ring)

In the first example we use (modified) Le Lann, Chang and Roberts (LCR) algorithm to solve leader election problem. The algorithm uses only unidirectional communication and does not rely on knowledge of the size of the ring. Only the leader performs an output. The algorithm uses only comparison operations on the unique identifiers [A. Lynch: Distributed Algorithms, p. 27].

- All processes send identifier around the ring.
- Upon receiving an incoming identifier, it compares to its own.
- If identifier is greater than its own, keep passing the identifier.
- If received identifier is smaller, discard identifier.
- If identifier is equal to its own, declare itself the leader.
- Leader sends a report message around the ring.
- Upon receiving a report message process halts.



Everyone Eventually Halts; Leader Elected

$c_1 = [\mathsf{rec(uid)} > \mathsf{snd(uid)}]$ $c_2 = [\mathsf{rec(uid)} < \mathsf{snd(uid)}]$ $c_3 = [\mathsf{rec(uid)} = \mathsf{snd(uid)}]$
$c_4 = [\mathsf{f_{uid}(lead)} = \mathsf{f_p(uid)}]$ $c_5 = [\mathsf{f_{uid}(lead)} \neq \mathsf{f_p(uid)}]$

Comments:

- $P_{[i]}$ always sends a message to $\mathsf{P}_{[(i+1) \bmod n]}$ where n is the number of processes.
- In contrary to other protocols (e.g. distr. resource allocation) the source of received message is irrelevant (same destination) for this protocol.
- Message types patterns [when, where, status] message;
- Initiating messages such as snd(uid) typically don't have when part.
- A reply message has when guard part.
- Status variable for reasoning about protocols.
- $\mathsf{status}(\mathsf{p}_1) := \mathsf{start} \rightarrow \cdot \rightarrow \mathsf{status}(\mathsf{p}_1) := \mathsf{consensus}$
- $\mathsf{status}(\mathsf{p}_1) := \mathsf{state}_1 \rightarrow \mathsf{Guards} \rightarrow \mathsf{status}(\mathsf{p}_1) := \mathsf{state}_2$
- Use flow plug-in to verify process state transition.
- Capturing processes knowledge;
  - message content;
  - topology;

Dinning Philosophers Protocol

Distributed Resource Allocation Protocol