Project 1: Leader Election

Program Verification Course 12/13

1 LE in a Ring

Consider N processes connected in a ring so that process No. i is connected to process No. $i{+}1$ **mod** N. One directional connection in the indicated direction is sufficient. Importantly: *a process can only send messages to the process it is connected to.*

Every process has a unique integer ID which is not necessarily the same as its number in the ring. The goal of leader election is to choose one process as a 'leader'. The *DLER* algoritihm goes as follows:

1. Every process i sends its ID to its sucessor in the ring.
2. When a process i with ID x_i receives an id x' from its predecessor, it does the following:
   (a) If $x' < x_i$, it discards x'.
   (b) If $x' > x_i$, it forwards x' to its sucessor.
   (c) If $x' = x_i$ the process wins and declares itself as the leader.

Furthermore, the winning process also needs to let the other process knows that he is the leader. I leave it to you to decide how this is done.

2 Base Task, max. 8pt

Verify the *DLER* algorithm by model checking. The specification to check is:

1. The algorithm terminates.
2. When it terminates, every process agrees who the leader is.

You will need to express the algorithm in Promela first. Furthermore, model checking can only deal with finite state space. So, you can only verify the algorithm up to some concrete N.

See if you can express the specification both *with* and *without* LTL.

3 Extension, max. 2pt

The *DLER* is 'deterministic' in the sense that it always chooses the process with the highest ID as the leader. Can you come up with a modification that makes it non-deterministic?

Note: what can be a challenge here is for the processes to detect when it is safe to terminate.

4 To deliver

1. The source code of your model along with any LTL formula you use.
2. A short report. It should contain:
   (a) Your model of the algorithm and its explanation.
   (b) How you express the above specification in SPIN.
3. Your verification results (e.g. with screenshots)