Assignment 6

Deadline: Friday **15.12.2023** 23:55

Notes:

- Solve the assignment **on your own** — no groups allowed.
- Hand-written solutions will not be accepted, except for graphs and diagrams.
- If you hand in non-pdf files or multiple files, name your submission as `stla23_06_SURNAME.zip`, replacing `SURNAME` with your surname. Otherwise `stla23_06_SURNAME.pdf`. Also include your full name in the submitted PDF.
- Submit your solution via **Ilias**.

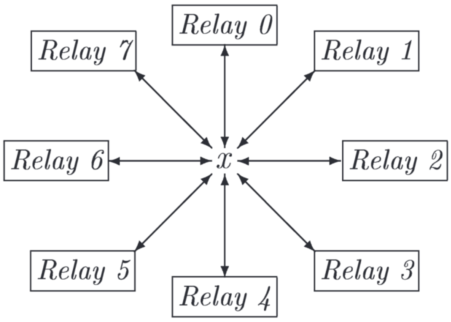

Figure 1: A relay ring for $N = 8$.

Exercise 6.1 Relay Ring 20 Points

In order to control the access to a shared resource one can implement a relay ring. In a relay ring the access is parsed to the processes in a round-robin fashion. The action which is executed by process i on access to the shared resource is called $Relay(i)$. An example for a relay ring is shown in Figure 1. A possible sequence of actions starting at process 5 would be:

$$Relay(5) \rightarrow Relay(6) \rightarrow Relay(7) \rightarrow Relay(0) \rightarrow Relay(1) \rightarrow Relay(2) \ldots$$

A specification of the relay ring called *RelayRing* is attached in the file `RelayRing.tla`.

a) Come up with a specification *TokenPass* that implements *RelayRing* using a single wire between each pair of processes. The wire joining process i and $(i + 1) \bmod N$ will be set by process i alternately to 1 and 0. For $i > 0$, process i's action will be enabled when the value on wire i is different from the value on wire $i - 1$. Process 0's action will be enabled when the value on wire 0 equals the value on wire $N - 1$. Verify, using TLC that *TokenPass* implements *RelayRing* under an appropriate refinement mapping. **(12 points)**

b) Formulate a liveness property *Liveness* that states that whenever a process doesn't have access to the shared resource, it will eventually get access. Verify, using TLC, that *TokenPass* satisfies *Liveness*. If it doesn't use the weakest fairness condition that is necessary to make it satisfy *Liveness*. Explain why you choose this fairness condition in contrast to no fairness condition or a stronger fairness condition. **(8 points)**

c) Formulate a safety property *Safety* that ensures mutual exclusion. Verify, using TLC, that *TokenPass* satisfies *Safety*. **(4 bonus points)**

Hand in all relevant files (`.tla`, `.cfg`, `.out`).

Total: 20 Points