Examine the algorithm presented in the 5th lecture (pp. 5-21–5-22).

**TicketME algorithm (informal):** A shared variable holds a pair \((next, granted)\) of values in \(\{1, \ldots, n\}\), initially \((1, 1)\). The \(next\) component represents the next “ticket” into the critical section, while the \(granted\) component represents the last “ticket” that has been granted permission to enter the critical section. When a process enters the trying section, it “takes a ticket,” that is, it copies and increments the \(next\) component modulo \(n\). When the ticket of a process is equal to the \(granted\) component, it goes to the critical region. When process exits the critical section, it increments the \(granted\) component modulo \(n\).


Model check the following properties with Maria:

1. Two processes will never be in the critical section at the same time
2. Process 1 never gets to enter the critical section
3. Process 1 is able enter the critical section
4. Process 2 can exit the critical section

In what different ways can the properties 1 and 2 be checked? Try at least two different ways.

Examine the counterexamples that you get. Were the algorithm used as a part of a real system, would the situations that the counterexamples represent be sensible?

Use the fairness assumptions in the model (switch \(-DFAIR\)) and check the properties 3 and 4 again.