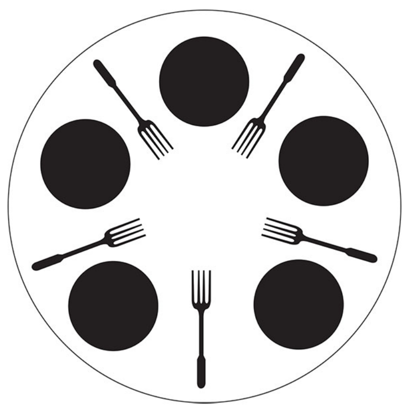
TP3 MVFA: LTL and the Dining Philosophers

Let us consider the Dining Philosopher's problem. A table holds five forks (arranged as pictured) and a bowl of noodles. Five philosophers sit around the table. A philosopher shares his right fork with his right neighbor and his left fork with his left neighbor. Each philosopher cycles through three states: thinking, hungry and eating (in that order). After thinking for a while, a philosopher gets hungry. In order to eat, he needs to hold his left and right forks (the noodles are very slippery). A philosopher can only pick one fork at a time. When he is done eating, he goes back to thinking. Since the philosophers are sharing forks, they need to find a method for them not to starve to death. More can be found at `https://fr.wikipedia.org/wiki/D%C3%AEner_des_philosophes` (French), `https://en.wikipedia.org/wiki/Dining_philosophers_problem` (English)

The objective of the exercise is to complete the program, and check that the given requirements are satisfied. You will have to formalize the transition relation for the existing variables, possibly introduce additional variables and definitions, formalize the requirements in temporal logic, and make sure that the design satisfies them.



1. Read section "LTL specification" of the NuSMV manual `http://nusmv.fbk.eu/NuSMV/userman/v26/nusmv.pdf` to know how to verify a LTL property and to understand the syntax of LTL properties. You can see that NuSMV allows LTL properties that specify the past, but we will limit ourselves to LTL properties that specify the present and the future.
2. We will first specify the models we have seen so far. Try to specify the LTL properties for `semaphore.smv` and `TP1.smv`. What happens when we omit `FAIRNESS running`?
3. Using NuSMV, build a model for this problem. You can use the file `filo.smv` on the MVFA website. Since we do not want any philosopher to starve, you will need to add variables to the proposed model. A simple solution is to use a round system, allowing a philosopher to eat only during his round. Another solution is to use semaphors, such that when a philosophe starts picking up his forks, he prevents his neighbors from doing the same.

4. Using `LTLSPEC` and `INVARSPEC`, insure that your model verify the following properties:

   - If a philosopher is hungry, then he will eat at some point in the future.
   - When a philosopher eats, none of his neighbors eat.
   - Each fork is put on the table infinitely often.
   - When a philosopher is thinking, he is not holding any fork.
   - If a philosopher is eating, then his next action is to think.
   - A philosopher can only pick one fork at a time.

   You can check other properties to ensure that your method works as intended.

5. If you have some time left, look up `COMPASSION` in the manual. Can you find an implementation using `COMPASSION` and using only the variables introduced in `filo.smv`? (You might need to change arguments of `philosopher`)

Send the solution as source files, to `victor.roussanaly@irisa.fr` with subject "[MVFA] TP3 - Names" with Names containing the names of the students involved.