nuXmv: exercises [*]

Patrick Trentin
patrick.trentin@unitn.it
http://disi.unitn.it/~trentin

Formal Methods Lab Class, May 12, 2015

UNIVERSITÀ DEGLI STUDI DI
TRENTO

Five philosophers sit around a circular table and spend their life alternatively thinking and eating. Each philosopher has a large plate of noodles and a fork on either side of the plate. The right fork of each philosopher is the left fork of his neighbor. Noodles are so slippery that **a philosopher needs two forks to eat it**. When a philosopher gets hungry, he tries to **pick up his left and right fork, one at a time**. If successful in acquiring two forks, he **eats for a while** (preventing both of his neighbors from eating), then **puts down the forks, and continues to think**.
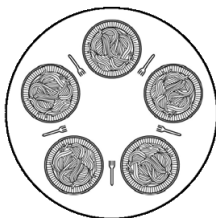
Exercise:

1. Implement in SMV a system that encodes the philosophers problem. Assume that when a philosopher gets hungry, he tries to pick up his left fork first and then the right one.
   Hint: you might consider an altruist philosopher, which can resign his fork in a deadlock situation.

2. Verify the correctness of the system, by specifiying and checking the following properties:
   - Never two neighboring philosophers eat at the same time.
   - No more than two philosophers can eat at the same time.
   - Somebody eats infinitely often.
   - If every philosopher holds his left fork, sooner or later somebody will get the opportunity to eat.

Exercise: mutual exclusion (Spin)

Exercise: A solution to **mutual exclusion** is based on message passing instead of shared variables.

Idea: two processes share a message channel and synchronize by reading and writing from/onto this channel.

- write a **Promela** model for mutual exclusion algorithm for two processes
- the only shared global data structure can be a channel
- check that following properties hold:
  - mutual exclusion
  - progress
  - lockout-freedom