Prof. O. Nierstrasz
Jorge Ressia, Jan Kurš

Serie 3 - Safety and Synchronization

Exercise 1

Answer the following questions:

1. Provide a definition of *multiprogramming*, *multiprocessing* and *distributed processing*.
2. In a multiprogramming system implemented on top of a multiprocessor architecture, do we need to implement a synchronization system if we know that all the threads use their own variables and they don't have to communicate with other threads?
3. What are safety properties? How are they modeled in FSP?
4. Is the busy-wait mutex protocol fair? Deadlock-free? Justify your answer.
5. If we have to implement a simple batch operating system on a single processor architecture would have sense to use path expressions o handle the concurrency? (Motivate your answer!!!)
6. What happens if you call wait or notify outside a synchronized method or block?
7. How would you manually check a safety property?
8. Can you ensure safety in concurrent programs without using locks?
9. what is the *finite progress assumption*?

Exercise 2

Consider the following process definitions:

```
P = (a->b->P).
Q = (c->b->Q).
||S1 = (P || Q).

S2 = (a->c->b->S2|c->a->b->S2).
```

Show that S1 and S2 describe the same behavior.

Exercise 3

Russian roulette is the practice of placing one bullet in a gun, spinning the cylinder and closing it into the gun without looking, aiming the revolver at one's own head in a suicidal fashion, and pulling the trigger.

In Figure 1 we show the Finite State Process (FSP) description of the Labeled Transition System (LTS) graph for a simulation of the Russian Roulette game. To simplify the number of possible states, the gun in our version has only two slots for holding bullets. Initially the gun is loaded with one bullet.

Write the FSP process definitions corresponding to Figure 1 and start the processes.

For the solution we expect you to have 4 parallel processes. You should define a Player Process and create two instances labeled *Michael* and *Charlton* respectively. They share the table resource where the loaded gun is placed at the start of the game. Both *Michael* and *Charlton* compete for the gun. They are too impatient to take turns. You will need to define a process to describe the *LoadedGun* with two slot positions for the bullet. To play the game, a player can take the gun from the table, press the trigger and if he is lucky, he survives to play again. If he is unlucky, he shoots himself.

Check your FSP process description by generating the corresponding state machine using the LTS-analysis tool.

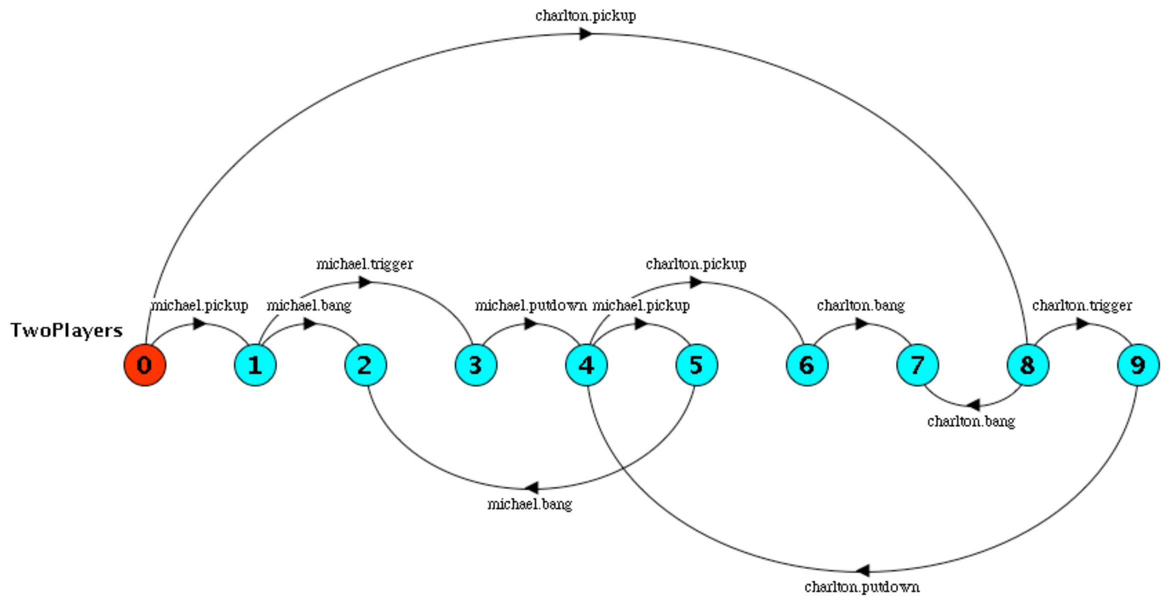

Figure 1: LTS for Russian Roulette.