INF646 Métodos Formales
2017-2

PREPARACIÓN PARA EL EXAMEN 1

1. Consider the following two processes, A and B, to be run concurrently in a shared memory (all variables are shared between the two processes).

PROCESS A:	PROCESS B:
1 $x := 2x$;	1 $x := 1$;
2 $x := 2x$;	2 $x := x + 1$;

Assume that load (read) and store (write) of the single shared register x are atomic, x is initialized to 0, and x must be loaded into a register before being incremented. What are all the possible values for x after both processes have terminated?

Use, como el modelo del incremento de una variable x en Promela, la secuencia de dos asignaciones: `temp = x; x = temp + 1`.

2. (Bridge Crossing Problem) Three people begin on the same side of a bridge. You must help them across to the other side. It is night. There is one flashlight. A maximum of two people can cross at a time. Any party who crosses, either one or two people, must have the flashlight to see. The flashlight must be walked back and forth, it cannot be thrown, etc. Each person walks at a different speed. A pair must walk together at the rate of the slower person's pace, based on this information: Person 0 takes t0 = 1 minutes to cross, and the other persons take t1 = 2 minutes, and t2 = 5 minutes to cross, respectively.

```
$ cat bridge.pml

...

#define max(a,b) ((a>b) -> a : b)
#define N        3

byte a[N] = 0    /* crossing times of N persons */

active proctype Bridge() {
...
}
```

Construya el modelo en Promela para encuentrar todas las soluciones posibles.

3. The *frogs puzzle* (from `jspin-4-7/jspin-examples/frogs.pml`) La descripción se puede encontrar en:

http://www.hellam.net/maths2000/frogs.html

```
$ cat frogs.pml
/*
  Frogs puzzle:
    Seven stones
    Three male frogs at right facing left
    Three female frogs at left facing right
    F-> F-> F-> [EMPTY] <-M <-M <-M

  Frogs can move in the direction it is facing to an empty stone:
    That is adjacent
    That is reached by jumping over a frog on an adjacent stone

  Is there a sequence of moves that will exchange the positions
    of the male and female frogs?
  Solution: try to Verify/Safety []!success;
    when it fails the trail gives the set of moves.
  Local variables ":init:" and "at" can be excluded.
*/

#define STONES 7

/* Verify acceptance of []!success */
#define success (\
    (stones[0]==female) && \
    (stones[1]==female) && \
    (stones[2]==female) && \
    (stones[4]==male)   && \
    (stones[5]==male)   && \
    (stones[6]==male)      \
    )

mtype = { none, male, female }
mtype stones[STONES];

ltl { []!success }

proctype mF(byte at) {
end:do
    :: atomic {
            (at < STONES-1) &&
            (stones[at+1] == none) ->
                stones[at] = none;
                stones[at+1] = male;
                at = at + 1;
        }
    :: atomic {
            (at < STONES-2) &&
            (stones[at+1] != none) &&
            (stones[at+2] == none) ->
                stones[at] = none;
                stones[at+2] = male;
                at = at + 2;
        }
    od
}
```

```
proctype fF(byte at) {
end:do
    :: atomic {
            (at > 0) &&
            (stones[at-1] == none) ->
                stones[at] = none;
                stones[at-1] = female;
                at = at - 1;
        }
    :: atomic {
            (at > 1) &&
            (stones[at-1] != none) &&
            (stones[at-2] == none) ->
                stones[at] = none;
                stones[at-2] = female;
                at = at - 2;
        }
    od
}

init {
    atomic {
        stones[STONES/2] = none;
        byte I = 0;
        do
        :: I == STONES/2 -> break;
        :: else ->
            stones[I] = male;
            run mF(I);
            stones[STONES-I-1] = female;
            run fF(STONES-I-1);
            I++
        od
    }
}
```

6 de octubre del 2017