

Este proyecto debe ser resuelto en grupos de 3 personas. La fecha límite para la resolución del proyecto es el 8 de Mayo de 2011 a las 23:55, y la entrega de la resolución debe hacerse a través del moodle de la materia.

Para cada uno de los ejercicios debe entregar el código fuente de los modelos, incluyendo las propiedades solicitadas. Además debe entregar un informe describiendo los modelos realizados, los detalles de las verificaciones llevadas a cabo y los resultados obtenidos.

Ejercicio 1. Un problema clásico en Concurrency es el *Problema de los Fumadores*. El mismo consiste de cuatro procesos, tres de los cuales son *fumadores*, y con un cuarto proceso que representa un *agente proveedor*. Los fumadores están continuamente armando cigarrillos y fumándolos. Para armar un cigarrillo, un fumador necesita tres ingredientes: tabaco, papel y fósforos, y cada uno de los tres fumadores cuenta una cantidad infinita de sólo uno de los ingredientes: uno de los fumadores tiene una cantidad infinita de fósforos, otro una cantidad infinita de papel, y el tercero una cantidad infinita de tabaco. El agente, en cambio, cuenta con una provisión infinita de los tres ingredientes. El sistema debe progresar repitiendo continuamente la siguientes tareas. Se comienza con el agente colocando dos de los ingredientes en la mesa; luego, el fumador que cuenta con el tercero de los ingredientes los toma, arma un cigarrillo, lo fuma, y comunica al agente al terminar de fumar.

Modele este problema en FSP, de manera tal de sincronizar adecuadamente los procesos para evitar deadlock. Describa en FSP la propiedad que indica que los tres fumadores consiguen fumar con infinita frecuencia, y verifique si la misma se cumple usando LTSA.

Ejercicio 2. Considere el conocido problema de calcular el máximo valor de un arreglo $A[0..n-1]$ de elementos. Una forma de resolver este problema de forma paralela es a través del algoritmo que se describe a continuación, en pseudo-código:

```
FAST-MAX(A)
  n ← length[A]
  m: array[0..n-1] of boolean
  for i ← 0 to n-1, in parallel do
    m[i] ← true
  for i ← 0 to n-1 and j ← 0 to n-1, in parallel do
    if A[i] < A[j] then
      m[i] ← false
  for i ← 0 to n-1, in parallel do
    if m[i] = true then
```

```
        max <- A[i]
    return max
```

Este algoritmo resuelve el problema de calcular el máximo valor del arreglo en $O(1)$, provisto que se cuenta con n^2 procesadores. Implemente el algoritmo en Promela, incluyendo un proceso que permita cargar valores en el arreglo a tratar, no determinísticamente. Complemente su modelo para poder expresar y verificar en Spin las siguientes propiedades:

- cualquiera sea el arreglo a tratar, el proceso eventualmente termina,
- una vez que concluye el proceso, el valor resultante es el máximo valor del arreglo.