Validación y Verificación - Año 2019

Práctico 9

Model Checking: Spin

Ejercicio 1. Instale iSpin (http://spinroot.com/spin/Man/README.html) en la página www.spinroot.com encontrarás información acerca de cómo usar la herramienta.

Ejercicio 2. Aquí se encuentra un manual para familiarizarce con el lenguaje de modelado ProMeLa: http://spinroot.com/spin/Man/Manual.html

****Ejercicio 3**. Cuatro soldados que están gravemente heridos, intentan huir de la guerra y en medio de la noche llegan a un puente (frontera) que deben cruzar. El puente ha sido dañado y solo puede llevar a dos soldados a la vez. Además, se han colocado varias minas en el puente y se necesita una linterna para evitar pisar las minas. El enemigo está en su camino, por lo que los soldados saben que solo tienen 60 minutos para cruzar el puente. Los soldados tienen una sola linterna y no están igualmente heridos. Los siguientes son los tiempos que tarda cada soldado en cruzar el puente (de ida):

Soldado S0 5 minutos
Soldado S1 10 minutos
Soldado S2 20 minutos
Soldado S3 25 minutos

¿Existe un *Schedule* que permite que los 4 soldados estén del lado seguro del puente en 60 minutos?

Modele este problema en ProMeLa. Utilice Spin para encontrar algún camino que permita que los 4 soldados terminan del lado seguro en 60 minutos.

Tips: al menos deberá definir los siguientes procesos y las estructuras necesarias (el uso de canales sincronicos puede ser de ayuda).

```
active proctype Unsafe(){

      -Inicia con todos los soldados del lado inseguro

      do ::
            -Selecciona dos soldados que este del lado inseguro
            -envía los dos soldados seleccionados al lado seguro
            -Si ya no hay soldados del lado inseguro termina el proceso
            -Recibe el soldado que  llega desde el lado seguro
            -Agrega al tiempo total, el tiempo de este último soldado
      od
}



active  proctype Safe()  {
```

```
      -Inicia sin ningún soldado del lado inseguro

      do ::
            -Recibe los dos soldados que fueron enviados del lado inseguro
            -Agrega al tiempo total el tiempo del soldado más lento recibido
            -Si ya estan los 4 soldados del lado seguro termina el proceso
            -Selecciona un soldado que esté del lado seguro
            -Envía el soldado seleccionado al lado inseguro
      od
}
```

****Ejercicio 5.** Considere el 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 (*in parallel: cada proceso realiza un acceso al arreglo*):

```
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 deterministicamente. Complemente su modelo para poder expresar y verificar en Spin las siguientes propiedades:

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

**** Deberá entregar alguno (a elección) de estos ejercicio como parte del Trabajo Práctico Obligatorio 2**