

Validación y Verificación - Año 2019

Práctico 8 Parte 2

Análisis de Propiedades en FSP

Ejercicio 6. Una galería de arte abierta al público en general tiene dos puertas de entradas, una a la izquierda del salón y otra a la derecha del salón. El director de la galería está interesado en saber cuántas personas hay en la galería en un momento específico. Para ello se requiere un sistema de computadora que realice el control. Tenga en cuenta que las personas podrían entrar / salir por cualquiera de las dos puertas. Modele el problema en FSP. **Hint:** Puede utilizar la variable modelada en el ejercicio 2 y cerrojos para asegurar exclusión mutua a la misma.

CERROJO = (adquirir->liberar->CERROJO).

Ejercicio 7. Un problema clásico en concurrencia es el Problema de *Baño Unisex*. El mismo consiste de un baño, que cuenta con N lugares o compartimentos, y el cual es compartido por hombres y mujeres. Una persona, sea ésta hombre o mujer, puede encontrarse usando uno de los compartimentos del baño, esperando por usar el baño, o haciendo alguna otra cosa (alternan además entre hacer otras cosas y tener necesidad de ir al baño). Teniendo en cuenta que ninguna persona puede usar el baño en simultáneo con personas del sexo opuesto, se desea desarrollar una solución a este problema que evite deadlock y sea justo (fair) respecto del acceso al baño por parte de las personas que lo requieren.

Modele este problema en FSP. Describa en FSP la propiedad que indica que hombre y mujeres no pueden estar dentro del baño al mismo tiempo y verifique si la misma se cumple usando LTSA.

Ejercicio Adicional

El problema de los filósofos comensales (Dijkstra, 1968) es un problema clásico de concurrencia. El problema consiste en lo siguiente: 5 filósofos comparten una mesa circular. Cada filósofo pasa su vida pensando y comiendo spaghetti. Un filósofo necesita dos tenedores para comer spaghetti. Desafortunadamente, solo se cuenta con 5 tenedores ubicados a la derecha e Izquierda de cada filósofo. Los filósofos acordaron solo utilizar los tenedores que se encuentran a la izquierda y a la derecha de cada uno. En *Examples->chapter6->DiningPhilosophers.lts* se provee una solución al problema. Analice la solución planteada, utilice LTSA para dibujar los LTS y el animador. Chequee si el modelo planteado es libre de deadlock. En el caso de que no lo sea analice la traza reportada. En *DeadlockFreePhilosophers.lts* se provee una solución libre de deadlock. Analice dicha solución. ¿Cómo se solucionó el problema? Escriba un programa Java (usando Threads) que respete este modelo.