1er. Evaluación Parcial – 28 de abril de 2008 Nombre: .....

Ingeniería de Software II

Ejercicio 1. Considere el problema de los filósofos comensales con la siguiente variante: los tenedores se encuentran en el centro de la mesa en una bandeja. Cuando un filósofo desea comer debe sentarse a la mesa y tomar los dos tenedores de la bandeja. Cuando termina de comer los devuelve a la bandeja.

(a) Suponga que hay N filósofos y M tenedores. Dé una condición sobre N y M para que el sistema no entre en deadlock. Asegurese que su condición sea lo más ajustada posible.

(b) Modele el sistema en FSP bajo la suposición de que la condición anterior no se cumple, modificando el sistema apropiadamente para evitar deadlock.

Ejercicio 2.

(a) Considere el alfabeto $\Sigma = \{a, b\}$. Determine si la propiedad $(b^*a)^*b^\omega$ es de *safety*, *liveness*, ambas, o ninguna. Justifique la respuesta.

(b) Si no es de *safety* extiéndala agregando el mínimo conjunto de palabras necesario de manera que lo sea. Use expresiones ω-regulares. Justifique que la extensión es de *safety* y mínima.

(c) Dé el autómata de Büchi que representa el lenguaje dado. Justifique.

Ejercicio 3. Dada dos fórmulas LTL ϕ y ψ, el término $\phi \curvearrowleft \psi$ se interpreta como "Siempre ocurre ϕ antes que ψ". Dé una definición de $\phi \curvearrowleft \psi$ usando los operadores LTL.

Ejercicio 4. Dé un algoritmo que demuestre que dos fórmulas LTL ϕ y ψ son equivalentes.

Ejercicio 5. Considere un sistema de administración de impresoras idénticas con dos colas, una para la impresión de textos cortos y la otra para la impresión de textos largos. Suponga que hay N impresoras y que la capacidad de cada cola es M. Las impresoras sólo pueden tomar trabajos de la cola de textos largos si la cola de textos cortos está vacía. Existen además K procesos que envían trabajos a imprimir a cada cola según sean textos cortos o largos.

(a) Modele el sistema utilizando FSP asegurándose de evitar *deadlocks* y proveyendo el diagrama de estructura del modelo. Si le conviene, utilice la siguiente definición de cola:

```
COLA(TAM=M) = COLA[0],
COLA[i:0..TAM] = ( when (i > 0) desencolar -> COLA[i-1]
                 | when (i < TAM) encolar -> COLA[i+1]
                 | when (i == 0) vacia -> COLA[i]).
```

(b) Represente en FSP la propiedad de safety que dice que nunca hay más de $2M + N$ trabajos dentro del sistema de impresión (ya sea imprimiendo o encolados). Asegúrese de que su modelo la cumpla.