

## 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.  $M > N$
- (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  $(b^*a)^*b^w$  es de *safety*, *liveness*, ambas, o ninguna. Justifique la respuesta.  $a^w$
- (b) Si no es de *safety* extiéndala agregando el mínimo conjunto de palabras necesario de manera que lo sea. Use expresiones  $w$ -regulares. Justifique que la extensión es de *safety* y mínima.  $(b^*a)^*b^w + a^w$
- (c) Dé el autómata de Büchi que representa el lenguaje dado. Justifique.  $(a+b)^*(b^w + a^w)$

Ejercicio 3. Dada dos fórmulas LTL  $\phi$  y  $\psi$ , el término  $\phi \frown \psi$  se interpreta como "Siempre ocurre  $\phi$  antes que  $\psi$ ". Dé una definición de  $\phi \frown \psi$  usando los operadores LTL.  $\Box \neg (\neg \phi \cup \psi)$

Ejercicio 4. Dé un algoritmo que demuestre que dos fórmulas LTL  $\phi$  y  $\psi$  son equivalentes.

$$L(A\phi) \cap L(A\psi) = \emptyset \quad \text{y} \quad L(A\neg\phi) \cap L(A\psi)$$

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 provéyendo 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.