

Ingeniería de Software I - Recuperatorio del segundo parcial

Jueves 29 de noviembre

Lea cuidadosamente los enunciados antes de resolver los ejercicios. El parcial es a libro abierto. Justifique sus respuestas y explícite cualquier suposición que tome.

Ejercicio 1 (FSM)

El club *Defensores del Catenaccio* nos ha solicitado modelar su mecanismo de venta de entradas. El club tiene dos puntos de ventas de entradas: uno por ventanillas en el estadio y otro online por medio de su página web. Las localidades son populares o plateas. Ninguna de ellas es numerada, lo cual simplifica mucho el proceso de venta. Los interesados pueden comprar hasta un máximo de 4 entradas cada vez que se presenten en ventanilla o vía el sitio web. No faltan los “vivos” que hacen la cola en la ventanilla o entran al sitio más de una vez para evadir estas restricciones. Sin embargo, el presidente del club manifestó que no le preocupan esos “vivos”.

El estadio cuenta con dos ventanillas para venta de entradas. Obviamente, cada ventanilla sólo atiende a una persona por vez. Si bien el estadio tiene capacidad para 45.000 personas (20.000 plateas y 25.000 populares), al llegar a un límite de 30.000 localidades vendidas, se envía un mail a un conjunto de personas, que se han registrado para recibir este tipo de alertas. La administración de los datos de estas personas registradas escapa al alcance de nuestro proyecto y esperamos recibir una lista actualizada cada vez que se necesite. Además, para intentar balancear la cantidad de plateas vendidas en ventanillas vs. las ventas online, cada cinco minutos se chequea la cantidad de ventas de plateas por web. Si en tal período hubo más de 10 plateas vendidas, la venta de plateas online se detiene por 10 minutos.

Modele, utilizando FSMs, el comportamiento descripto.

Ejercicio 2 (Testing funcional)

Se tiene una función que decide, para un CUIL y un crédito solicitado de cierto monto, si el crédito se aprueba o no. Es decir, dado el CUIL de una persona y un monto solicitado, se retorna si se aprueba o no tal crédito. Existe una lista con los CUIL de personas morosas. Si el CUIL está en la lista entonces se rechaza el crédito. Si el monto solicitado es menor o igual a los \$10.000, y el CUIL que lo solicita no está en la lista de morosos, entonces siempre se aprueba el crédito. Para pedidos de más de \$10.000 la aprobación depende de la historia crediticia para ese CUIL. La historia crediticia es simplemente una lista de montos que representa los créditos otorgados con anterioridad para ese CUIL. El pedido se aprueba sólo si la historia crediticia es no vacía y el monto solicitado no supera el 110% del monto máximo de su historia crediticia.

Proponga una partición de categorías y un conjunto de tests para validar la implementación de la funcionalidad pedida. El conjunto de tests debe ser *adecuado*, es decir que debe contemplar todos los casos interesantes de ser testeados.

Ejercicio 3 (Testing estructural)

Considere la siguiente función:

```
1 void foo(int a, int b, int c) {  
2     int d = 10;  
3     if (d == 10 && (a == 10 || b == 10))  
4         d = a + b + d;  
5         if (d < 20 && (a + b) == 10)  
6             printf("Va por menos de 20!");  
7     if (c > 0)  
8         d += c;  
9     else  
10        d -= c;  
11    if (d < 0)  
12        printf("Dio negativo!");  
13    else  
14        printf("Dio positivo!");  
15 }
```

- Usando los números de líneas como nodos, construya el Control Flow Graph correspondiente al código.
- Describe coloquialmente cuál es la salida de la función en relación a sus parámetros de entrada.
- Dar un conjunto de casos de test que cubra cada condición atómica. Utilice semántica de corto-circuito. En caso de haber condiciones imposibles de cubrir, justifique por qué es así.