Algoritmos y Estructuras de Datos II
Primer parcial – 08/05/2010

Ej. 1. Especificación

Se desea especificar una generalización del problema de los fumadores. Este problema consiste en un grupo de fumadores compulsivos que cohabitan en una oficina. Un fumador sólo puede fumar cuando tiene en su poder papel, tabaco y fósforo. Lamentablemente, ningún fumador ha podido comprarse todos estos elementos, y a todos les falta exactamente un elemento (por simplicidad, consideraremos que cada uno tiene una cantidad ilimitada de los elementos que posee). Dado que estos fumadores son extremadamente egoístas, no comparten ni intercambian elementos entre sí. En consecuencia, ninguno podría fumar. Sin embargo, existe en esta oficina un jefe que se divierte observando el comportamiento de estos fumadores y cada tanto ofrece una ración de alguno de los tres elementos, por ejemplo una ración de tabaco, una hojita de papel, o un fósforo. Cada vez que el jefe ofrece una ración, uno de los fumadores a los que les falta el elemento ofrecido comenzará a fumar inmediatamente (estos fumadores son muy civilizados y siguen una política secreta muy precisa para determinar quien debe tomar la ración). Si no hubiere fumadores esperando por este elemento, el jefe se lo llevará. Cuando un fumador termina su cigarrillo, retorna a su actividad principal: esperar a que el jefe ofrezca el elemento que le falta para poder fumar.

Se desea conocer en todo momento la identidad de los fumadores que están fumando y la cantidad de veces que cada fumador logró comenzar a fumar.

Dar una especificación del problema descripto utilizando TADs. Puede asumir que el tipo RECURSO se encuentra definido como una enumeración de las constantes *papel*, *tabaco* y *fósforo*.

Ej. 2. Inducción Estructural

Considere la siguiente definición de la operación *inorder* que permite obtener la secuencia inorder de los nodos de un árbol binario:

inorder : ab(α) $\longrightarrow$ secu(α)

i_1) inorder(nil) $\equiv$ $<>$

i_2) inorder(bin(i, r, d)) $\equiv$ inorder(i) & (r $\bullet$ inorder(d))

y la siguiente operación *recorrido* que puede ser utilizada también para recorrer inorder los nodos de un árbol

recorrido : ab(α) $\times$ secu(α) $\longrightarrow$ secu(α)

r_1) recorrido(nil,w) $\equiv$ w

r_2) recorrido(bin(i, r, d),w) $\equiv$ recorrido(i, r $\bullet$ recorrido(d,w))

Se solicita demostrar por inducción estructural la siguiente propiedad:

$$(\forall a : \mathrm{ab}(\alpha))(\forall w : \mathrm{secu}(\alpha))(\mathrm{inorder}(a) \& w \equiv \mathrm{recorrido}(a, w))$$

Para ello,

1. Dar el esquema de inducción.
2. Plantear el caso base y resolverlo, justificando cada paso de la demostración.
3. Plantear el/los paso/pasos inductivos, marcando claramente hipótesis, tesis inductiva y alcance de los cuantificadores. Resolver, justificando cada paso de la demostración.

Nota: Si se necesitaran lemas auxiliares, enúncielos, plantee su esquema de demostración y demuéstrelos.

Ej. 3. Invariante de Representación y Función de Abstracción

Considere la siguiente especificación que describe un sistema de monitoreo de una planta industrial que cuenta con un conjunto de alarmas asociadas a distintos sensores. Cada sensor está asociado a una única alarma pero cada una de estas puede ser activada por distintos sensores. Una alarma está activa cuando la medición de al menos uno de sus sensores asociados supera un valor umbral definido para ese sensor.

El siguiente TAD especifica el comportamiento de las alarmas.

TAD PLANTA

géneros planta

observadores básicos

esAlarma : planta × alarma $\longrightarrow$ bool

esSensor : planta × sensor $\longrightarrow$ bool

alarmaSensor : planta p × sensor s $\longrightarrow$ alarma {esSensor(p, s)}

umbral : planta p × sensor s $\longrightarrow$ nat {esSensor(p, s)}

medicion : planta p × sensor s $\longrightarrow$ nat {esSensor(p, s)}

generadores

crear : conj(alarma) $\longrightarrow$ planta

agSensor : planta p × sensor s × nat u × alarma a $\longrightarrow$ planta {$\neg$esSensor(p, s) $\wedge$ esAlarma(p, a) $\wedge$ $u > 0$}

nuevaMedicion : planta p × sensor s × nat $\longrightarrow$ planta {esSensor(p, s)}

otras operaciones

encendidosPorAlarma : planta p × alarma a $\longrightarrow$ conj(sensor) {esAlarma(p, a)}

encendida : planta p × alarma a $\longrightarrow$ bool {esAlarma(p, a)}

axiomas

esAlarma(crear(c), a) $\equiv$ $a \in c$

esAlarma(agSensor(p, s, u, a'), a) $\equiv$ esAlarma(p, a)

esAlarma(nuevaMedicion(p, s, n), a) $\equiv$ esAlarma(p, a)

esSensor(crear(c), s) $\equiv$ false

esSensor(agSensor(p, s', u, a), s) $\equiv$ $s = s'$ $\vee$ esSensor(p, s)

esSensor(nuevaMedicion(p, s', n), s) $\equiv$ esSensor(p, s)

alarmaSensor(agSensor(p, s', u, a), s) $\equiv$ **if** $s = s'$ **then** a **else** alarmaSensor(p, a) **fi**

alarmaSensor(nuevaMedicion(p, s', n), s) $\equiv$ alarmaSensor(p, s)

umbral(agSensor(p, s', u, a), s) $\equiv$ **if** $s = s'$ **then** u **else** umbral(p, a) **fi**

umbral(nuevaMedicion(p, s', n), s) $\equiv$ umbral(p, s)

medicion(agSensor(p, s', u, a), s) $\equiv$ **if** $s = s'$ **then** 0 **else** medicion(p, a) **fi**

medicion(nuevaMedicion(p, s', n), s) $\equiv$ **if** $s = s'$ **then** n **else** medicion(p, a) **fi**

encendidosPorAlarma(crear(c), a) $\equiv$ $\emptyset$

encendidosPorAlarma(agSensor(p, s, u, a'), a) $\equiv$ encendidosPorAlarma(p, a)

encendidosPorAlarma(nuevaMedicion(p, s, n), a) $\equiv$ **if** alarmaSensor$(p, s) = a$ $\wedge$ $n >$ umbral(p, s) **then**
  Ag(s, encendidosPorAlarma(p, a))
else
  encendidosPorAlarma(p, a) $-$ $\{s\}$
fi

encendida(p, a) $\equiv$ #encendidosPorAlarma$(p, a) > 0$

Fin TAD

Se decidió utilizar la siguiente estructura como representación:

PLANTA **se representa con** estr, donde
estr es tupla $\langle$*alarmaSensor*: dicc(alarma, conj(sensor))
sensores: dicc(sensor, tupla$\langle$umbral: nat, medicion:nat$\langle\rangle$,
alarmasEncendidas: secu(alarmas)$\rangle$

Donde, *alarmaSensor* representa los sensores asociados a cada alarma, *sensores* asocia a cada sensor con su umbral y su medición actual, y *alarmasEncendidas* mantiene una secuencia de todas las alarmas que están encendidas.

1. Escribir en castellano y formalmente el invariante de representación.
2. Escribir formalmente la función de abstracción.