

Algoritmos y Estructuras de Datos II

Recuperatorio del primer parcial — 10/12/2008

Aclaraciones

- El parcial **NO** es a libro abierto.
- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas.
- Incluya el número de orden asignado (léalo cuando circule la planilla), apellido y nombre en cada hoja.
- Al entregar el parcial complete los datos faltantes en la planilla.
- Cada ejercicio se calificará con B, R ó M. Una B no significa que el ejercicio está “perfecto”, sino que cumple con los requisitos necesarios para aprobar. En los parciales promocionados se asignará una nota numérica más precisa a cada ejercicio.
- Para aprobar el parcial debe obtenerse B en el ejercicio 1 y en alguno de los ejercicios 2 y 3. Un parcial se considera promocionado si está aprobado y su puntaje es 70 o superior.

Ej. 1. Especificación (Servicio de Impresión)

La empresa ASD brinda un servicio de impresión a terceros. ASD ofrece dos tipos de calidad de impresión: estándar y superior. El precio de cada uno de estos servicios viene dado por página de impresión, siendo el estándar el más económico. Cada vez que el cliente solicita un trabajo de impresión indica, además de la cantidad de páginas que requiere, la calidad de impresión que contrata. ASD se compromete a entregar la impresión solicitada con una calidad igual o superior a la solicitada por el cliente, manteniendo el precio contratado por el cliente.

La empresa cuenta con dos impresoras, una para calidad estándar y otra para calidad superior. Cada impresora puede manejar una cantidad máxima de trabajos en espera. Cada vez que llega un nuevo pedido, se trata de asignar el mismo a la impresora de la calidad solicitada. Si esto no es posible, se asigna a la impresora de mayor calidad. Si ésto no fuese posible, el trabajo no se acepta.

Se desea modelar el funcionamiento de la empresa ASD a través de TADs. Debe ser posible conocer en todo momento:

- el estado de cada una de las impresoras, si esta imprimiendo o no y la cantidad de trabajos en espera.
- el dinero que ha “perdido” ASD. Se considera que ASD ha perdido dinero cada vez que ha impreso una página con una calidad superior a la acordada con el cliente.

Ej. 2. Inducción estructural

Considere las siguientes operaciones sobre secuencia: (i) *suma*, que calcula la suma de todos los elementos de una secuencia no vacía de naturales; (ii) *cantApar* que devuelve la cantidad de ocurrencias de un determinado elemento en una secuencia; y (iii) *remp* que reemplaza todas las ocurrencias de un elemento por otro, que están definidas de la siguiente manera:

$$\begin{array}{ll} \text{suma} : \text{secu}(\text{nat}) \longrightarrow \text{nat} & (\neg(\text{vacía?}(s))) \\ s_1) \quad \text{suma}(s) & \equiv \text{if } \text{vacía?}(\text{fin}(s)) \text{ then } \text{prim}(s) \text{ else } \text{prim}(s) + \text{suma}(\text{fin}(s)) \text{ fi} \end{array}$$

$$\begin{array}{ll} \text{cantApar} : \alpha \times \text{secu}(\alpha) \longrightarrow \text{nat} & \\ c_1) \quad \text{cantApar}(a, \langle \rangle) & \equiv 0 \\ c_2) \quad \text{cantApar}(a, b \bullet s) & \equiv (\text{if } a = b \text{ then } 1 \text{ else } 0 \text{ fi}) + \text{cantApar}(a, s) \end{array}$$

$$\begin{array}{ll} \text{remp} : \alpha \times \alpha \times \text{secu}(\alpha) \longrightarrow \text{secu}(\alpha) & \\ r_1) \quad \text{remp}(a, b, \langle \rangle) & \equiv \langle \rangle \\ r_2) \quad \text{remp}(a, b, c \bullet s) & \equiv (\text{if } a = c \text{ then } b \text{ else } c \text{ fi}) \bullet \text{remp}(a, b, s) \end{array}$$

Se desea probar la siguiente propiedad:

$$(\forall a, b : \text{nat})(\forall s : \text{secu}(\text{nat}))(\neg \text{vacía?}(s) \Rightarrow \text{suma}(\text{remp}(a, b, s)) = \text{suma}(s) + (b - a) * \text{cantApar}(a, s))$$

1. Demuéstrelo por inducción estructural.

- a) Plantear y resolver el caso base.
- b) Plantear el paso inductivo, marcando claramente hipótesis, tesis inductiva y alcance de los cuantificadores.
- c) Resolver el paso inductivo. Justificar cada paso de la demostración.

IMPORTANTE: En caso de necesitar lemas auxiliares, enúncielos, plantee su esquema de demostración y demuéstrellos.

Ej. 3. Invariante de Representación y Función de Abstracción

Diego SanCarmen es un famoso criminal, buscado en todo el mundo. Los agentes de ACME tienen como misión su captura, pero el problema es que Diego tiene muchos secuaces que interfieren en el trabajo de ACME. Por eso, los agentes de ACME han decidido capturar primero a los virtualmente infinitos secuaces de Diego. Cada agente de ACME opera de la siguiente manera:

- Cuando se reporta a la agencia, situada en Indochina, se le encomienda un caso, asignándole al mismo un monto fijo para gastos.
- Para sus investigaciones, el agente debe recorrer distintas ciudades, y para eso debe realizar viajes. Cada viaje tiene un cierto costo, que depende únicamente de las ciudades de origen y destino (es decir, si se puede ir de A a B, entonces también se puede ir de B a A por el mismo costo).
- El caso termina cuando el agente captura al sospechoso o se queda sin dinero. En cualquiera de estos casos, regresa a Indochina y se le asigna nuevamente el monto inicial.

El comportamiento de los agentes fue especificado de la siguiente manera:

TAD CARMEN**observadores básicos**

dineroInicial : carmen \rightarrow monto
 dineroActual : carmen \rightarrow monto
 itinerario : carmen \rightarrow secu(ciudad)
 costo : carmen $c \times$ ciudad $ori \times$ ciudad $dest \rightarrow$ monto
 $(ori \in ciudades(c) \wedge dest \in ciudades(c)) \wedge_L hayRuta(c, ori, dest)$
 hayRuta : carmen $c \times$ ciudad $ori \times$ ciudad $dest \rightarrow$ bool $ori \in ciudades(c) \wedge dest \in ciudades(c)$
 ciudades : carmen $c \rightarrow$ conj(ciudad)

generadores

nuevo : conj(ciudad) $c \times$ dicc(conj(ciudad) \times monto) $d \times$ monto $m \rightarrow$ carmen
 $indochina \in c \wedge (\forall p : conj(ciudad))(p \in claves(d) \Rightarrow (\#p = 2 \wedge p \subseteq c))$
 viajar : carmen $c \times$ ciudad $dest \rightarrow$ carmen $hayRuta(c, ubicacion(c), dest)$
 capturar : carmen \rightarrow carmen

otras operaciones

alcanzaPlata : carmen $c \times$ ciudad $u \rightarrow$ bool $u \in ciudades(c)$
 ubicacion : carmen \rightarrow ciudad

axiomas

dineroInicial(nuevo(c, d, m)) $\equiv m$
 dineroInicial(viajar(c, d)) \equiv dineroInicial(c)
 dineroInicial(capturar(c)) \equiv dineroInicial(c)
 dineroActual(nuevo(c, d, m)) $\equiv m$
 dineroActual(viajar(c, d)) \equiv **if** \neg alcanzaPlata(c, d) **then** dineroInicial(c) **else** dineroActual(c) - costo($c, ubicacion(c), d$) **fi**
 dineroActual(capturar(c)) \equiv dineroInicial(c)
 itinerario(nuevo(c, d, m)) \equiv indochina $\bullet \langle \rangle$
 itinerario(viajar(c, d)) \equiv **if** \neg alcanzaPlata(c, d) **then** indochina $\bullet \langle \rangle$ **else** itinerario(c) $\circ d$ **fi**
 itinerario(capturar(c)) \equiv indochina $\bullet \langle \rangle$
 costo(nuevo(c, d, m), $ori, dest$) \equiv obtener($Ag(ori, Ag(dest, \emptyset)), d$)
 costo(viajar(c, d), $ori, dest$) \equiv costo($c, ori, dest$)
 costo(capturar(c), $ori, dest$) \equiv costo($c, ori, dest$)
 hayRuta(nuevo(c, d, m), $ori, dest$) \equiv $ag(ori, ag(dest, vacio)) \in claves(d)$
 hayRuta(viajar(c, d), $ori, dest$) \equiv hayRuta($c, ori, dest$)
 hayRuta(capturar(c), $ori, dest$) \equiv hayRuta($c, ori, dest$)
 ciudades(nuevo(c, d, m), $ori, dest$) $\equiv c$
 ciudades(viajar(c, d), $ori, dest$) \equiv ciudades(c)
 ciudades(capturar(c), $ori, dest$) \equiv ciudades(c)
 alcanzaPlata(c, d) \equiv costo($c, ubicacion(c), d$) \geq dineroActual(c)
 ubicacion(c) \equiv ult(itinerario(c))

Fin TAD

Para el diseño, se decidió utilizar la siguiente estructura:

estr es tupla

\langle dineroInicial : monto
 dineroActual : monto
 costos : dicc($\langle ori : ciudad \times dest : ciudad \rangle$, costo)
 ubicacion : ciudad
 ciudades : conj(ciudad)
 itinerario : secu(ciudad) \rangle

Se pide:

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