

Algoritmos y Estructuras de Datos II

Recuperatorio del primer parcial – 24 de Junio de 2016

Aclaraciones

- El parcial es a **libro abierto**.
- Cada ejercicio debe entregarse **en hojas separadas**.
- Incluir en cada hoja el número de orden asignado, número de hoja, apellido y nombre.
- Al entregar el parcial, completar el resto de las columnas en la planilla.
- Cada ejercicio se calificará con **Promocionado**, **Aprobado**, **Regular**, o **Insuficiente**.
- El parcial completo está aprobado si el primer ejercicio tiene al menos **A**, y entre los ejercicios 2 y 3 hay al menos una **A**. Para más detalles, ver “Información sobre la cursada” en el sitio Web.

Ej. 1. Especificación

Se desea modelar mediante TADs el comportamiento de la “LA RED ANTISOCIAL”. Esta red está generada por un ente, llamado líder, que se encarga de convocar a distintos “seguidores” a la red. Cada uno de estos seguidores a la vez puede convocar otros seguidores para que se unan a la red. En cualquier caso, todo miembro fue convocado por algún miembro preexistente.

En cualquier momento puede ocurrir que un miembro (incluido el líder) sea enjuiciado por la red. En dicho juicio se conforma un tribunal que dictamina si hay que echarlo o no. El tribunal está constituido por tres miembros elegidos de alguna forma (que se decidirá luego en la etapa de diseño) entre todos los integrantes de la red.

El dictamen del tribunal se puede saber directamente cuando éste se conforma: si más de la mitad del tribunal “desciende” del enjuiciado, es decir, fue convocado por él o uno de sus convocados, entonces no se lo expulsará. De otra forma se expulsará no sólo al enjuiciado, sino también a todos los miembros a quienes el expulsado haya convocado y a los convocados de los convocados (y así recursivamente).

Una vez expulsado (ya sea directa o recursivamente), cada ex-miembro queda prohibido de volver a pertenecer a la red de por vida. Claro, de por vida es mucho tiempo, sobre todo para aquellos expulsados que no fueron enjuiciados. Entonces, la red tiene un esquema de absoluciones. Si un ex-miembro que fue expulsado es absuelto, entonces se elimina cualquier registro en la red acerca de que dicha persona fue alguna vez miembro de la red y entonces puede ser nuevamente convocado a unirse.

Interesa determinar, además, la cantidad total de seguidores que tiene cada miembro de la red.

Ej. 2. Inducción estructural

Dada la siguiente operación entre diccionarios y los siguientes axiomas:

intersección : $\text{dicc}(\alpha, \beta) \times \text{dicc}(\alpha, \beta) \longrightarrow \text{dicc}(\alpha, \beta)$

$i_0)$ intersección(vacío, d_1) \equiv vacío

$i_1)$ intersección(definir(c, v, d_1), d_2) \equiv **if** $\neg c \in \text{claves}(d_2)$ **then** intersección(d_1, d_2) **else** definir(c, v , intersección(d_1, d_2)) **fi**

$K_0)$ claves(vacío) $\equiv \emptyset$

$K_1)$ claves(definir(c, s, d)) $\equiv \text{Ag}(c, \text{claves}(d))$

$C_0)$ $\emptyset \cap c \equiv \emptyset$

$C_1)$ $\text{Ag}(a, c) \cap d \equiv$ **if** $a \in d$ **then** $\text{Ag}(a, c \cap d)$ **else** $c \cap d$ **fi**

$P_0)$ $a \in \emptyset \equiv \text{false}$

$P_1)$ $a \in \text{Ag}(b, c) \equiv (a = b) \vee (a \in c)$

Demuestre por inducción estructural la siguiente propiedad de diccionarios:

$$(\forall d_1, d_2 : \text{dicc}(\alpha, \beta)) (\text{claves}(\text{intersección}(d_1, d_2)) \equiv \text{claves}(d_1) \cap \text{claves}(d_2))$$

En caso de utilizar lemas auxiliares, plantearlos claramente y demostrarlos. Además, se pide:

a) Escribir el predicado unario. Luego escribir, completo, **el esquema de inducción** a utilizar.

En el esquema, marcar **claramente** CB(s), PI(s), HI(s), TI(s) y el alcance de cada cuantificador.

- b) Plantear el/los caso(s) base y resolverlo(s), justificando cada paso de la demostración.
- c) Plantear el/los paso(s) inductivo(s) y resolverlo(s), justificando cada paso de la demostración.

Ej. 3. Diseño

En esta oportunidad Laboratorios ROCHO[®] nos pidió terminar el diseño de su sistema de seguimiento de recetas v2.0. Acoplándose a la revolución del la alegría imperante, decidieron abolir los premios de su sistema pero no así el control sobre los médicos. Lo que quiere el mencionado laboratorio es controlar cuán popular son sus medicamentos y cuáles son los médicos que los recetan. A través del seguimiento de recetas en farmacias, Laboratorios ROCHO[®] puede conocer qué médico recetó qué medicamento en todo momento y así aumentar los precios de los remedios más populares. En una receta se prescribe al menos un fármaco y como mucho tres. Los mismos pueden ser o no elaborados por Laboratorios ROCHO[®].

TAD MÉDICO es NAT, **TAD** FÁRMACO es STRING, **TAD** RECETA es <ID, CONJ(FÁRMACO)> donde ID es NAT
TAD FECHA es <AÑO, MES, DÍA> donde AÑO es NAT, MES es ENUM{1...12}, DÍA es ENUM{1...31}

TAD ROCHO[®] tracking

géneros RT **exporta** generadores, observadores, otras operaciones

generadores

iniciar : conj(médico) $cm \times$ conj(fármaco) $cf \rightarrow$ RT $\{ \neg \emptyset(cm) \wedge \neg \emptyset(cf) \}$
 registrarReceta : receta $r \times$ fecha $f \times$ médico $m \times$ RT $rt \rightarrow$ RT
 $\{ \Pi_1(r) \notin \text{todosLosIDRecetas}(rt) \wedge m \in \text{médicos}(rt) \wedge 1 \leq \# \Pi_2(r) \leq 3 \}$

observadores básicos

\otimes médicos : RT \rightarrow conj(médico)
 \otimes vademecum : RT \rightarrow conj(fármaco)
 quéRecetó : médico $m \times$ fecha $f \times$ RT $rt \rightarrow$ conj(receta) $\{ m \in \text{médicos}(rt) \}$

otras operaciones

quéRecetóMes : médico $m \times$ fecha $f \times$ RT $rt \rightarrow$ conj(receta) $\{ m \in \text{médicos}(rt) \}$
 \otimes todosLosIDRecetas : RT \rightarrow conj(ID)

Nota: los axiomas con el símbolo \otimes no están especificados aquí.

axiomas

quéRecetó($m, f, \text{iniciar}(cm, cf)$) $\equiv \emptyset$
 quéRecetó($m, f, \text{registrarReceta}(r, f', m', rt)$) $\equiv \text{quéRecetó}(m, f, rt) \cup \text{if } m = m' \wedge f = f' \text{ then } \{r\} \text{ else } \emptyset \text{ fi}$

Fin TAD

Para representar el TAD ROCHO[®] TRACKING se decidió utilizar la siguiente estructura:

RT se representa con *estr*, donde

estr es tupla \langle *médicos*: *dicc*(médico, conj(ID)), *vademecum*: *conj*(fármaco))
todasLasRecetas: *dicc*(ID, \langle receta, fecha \rangle),
totalRecetadoRochoDelMes: *dicc*(médico, *secu*(\langle año, mes, nat \rangle)),
farmédico: *dicc*(fármaco, *conj*(médico)) \rangle

médico es nat, fármaco es string, receta es <ID, conj(fármaco)> donde ID es nat
 fecha es <año, mes, día> donde año es nat, mes es Enum{1...12}, día es Enum{1...31}

El propósito de la estructura es el siguiente: en *médicos* se guarda el histórico para cada médico de todos los ID de las recetas que recetó, en *todasLasRecetas* figuran todas las recetas de todos los médicos junto con la fecha en la que fue recetada, en *vademecum* figuran los fármacos del laboratorio ROCHO[®] y en *totalRecetadoRochoDelMes* están cuántos fármacos del laboratorio ROCHO[®] recetó cada médico en una fecha determinada (acumulado por día y mayor a cero) y en *farmédico* se encuentran para todos los fármacos del vademecum los médicos que alguna vez lo recetaron.

- a) Escribir en castellano el invariante de representación.
- b) Escribir formalmente el invariante de representación.
- c) Escribir formalmente la función de abstracción.