

Algoritmos y Estructuras de Datos II

Primer parcial – Sábado 19 de Septiembre de 2015

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, Insuficiente o No Entregó. Notar que **I** es distinto de **N**.
- 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** y a lo sumo una **I**. Para más detalles, ver “Información sobre la cursada” en el sitio Web.

Ej. 1. Especificación

Escaleras al cielo

Hay una dama que está segura de que todo lo que reluce es oro, y está comprando una escalera al cielo, y cuando llega allí sabe si las tiendas están cerradas, con una palabra puede conseguir lo que vino a buscar.

Nos piden que ayudemos a conseguir una escalera al cielo (que como sabemos está medio indeterminada su altura, luego la construiremos de la altura que nos pidan).

La gente, para organizarse un poco, va subiendo cada escalón al unísono (o turno), cosa de pisarse lo menos posible y sólo pueden comenzar en alguno de los 10 primeros escalones (se comienza en el 1 y el piso se cuenta como el 0). Obviamente nunca tendremos dos almas en un mismo escalón.

Llegar al cielo no es fácil. Podremos tener trampas (máximo una por escalón), que pueden hacerte caer una determinada cantidad de escalones. También puede pasar que una persona se le haga escuchar un pájaro cantor que lo hará dudar por un tiempo (es decir que por algunos turnos no podrá moverse).

Si varias almas caen en el mismo peldaño, una sola queda en él. El resto cae hacia abajo hasta que quede una en cada escalón. Las almas que caen al piso quedan indefectiblemente en el purgatorio de por vida. No podrán acceder al cielo y por ende nos dejan de interesar. Obviamente pueden haber varias almas en el purgatorio.

En cualquier momento, una persona puede escuchar el sonido del gaitero. Si alguien lo escucha, en el próximo turno, si hay alguna trampa, mágicamente no caerá en ella.

Obviamente, los que llegan al cielo no siguen subiendo ni bajando.

Nos piden especificar en qué posición está cada una de las almas, la persona que menos trampas agarró (de las que están en la escalera) y quiénes están en el purgatorio.

HINT: Piense en las caídas como reacciones en cadena de arriba hacia abajo.

Ej. 2. Inducción estructural

Este TAD modela un mazo de cartas del famoso juego “CARS 2”. Donde el rayo mac queen salva a sus amigos de poderosos camiones maliciosos.

TAD CARTA es NAT					
TAD MAZO CARS					
generadores					
mazoCars					
nuevoMazo :			→	mazoCars	
agAlFrente :	carta × mazoCars		→	mazoCars	
observadores básicos					
vacío?	: mazoCars		→	bool	
frente	: mazoCars m		→	carta	{tamano(m) > 0}
resto	: mazoCars m		→	mazoCars	{tamano(m) > 0}
otras operaciones					
tamano	: mazoCars		→	nat	
pelear	: mazoCars m × mazoCars m'		→	mazoCars	{tamano(m)= tamano(m')}
axiomas $\forall c : carta, \forall m, m' : mazoCars$					
eV0)	vacío?(nuevoMazo)	≡ true	eT0)	tamano(nuevoMazo)	≡ 0
eV1)	vacío?(agAlFrente(c, m))	≡ false	eT1)	tamano(agAlFrente(c, m))	≡ 1+tamano(m)
eF0)	frente(agAlFrente(c, m))	≡ c	eP0)	pelear(m, nuevoMazo)	≡ nuevoMazo
eR0)	resto(agAlFrente(c, m))	≡ m	eP1)	pelear(m, agAlFrente(c, m'))	≡ agAlFrente(c−frente(m) , pelear(resto(m), m'))
Fin TAD					

Interesa demostrar por inducción estructural la siguiente propiedad:

$$(\forall m, m' : mazoCars) ((tamano(m) = tamano(m')) \Rightarrow pelear(m, m') = pelear(m', m))$$

Si lo desea puede usar que $(\forall c, c' : carta) (|c - c'| = |c' - c|)$ [Lema 1]

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

Ej. 3. Diseño

Nos interesa diseñar un ranking de ajedrez nacional. Cada jugador podrá jugar con otro. Obtiene 1 punto por ganar y -1 por perder (no contabilizaremos tablas o empates). Asimismo, un jugador puede tener un conjunto de jugadores que son sus clásicos (recíprocamente). Si un jugador gana un clásico, se adjudicarán 2 puntos.

A su vez se determinará el *Gran Maestro* del torneo como uno de los que más puntos tengan.

TAD PARTIDA ES TUPLA \langle blancas: jugador, negras: jugador, ganador: jugador \rangle			
TAD AJEDREZ			
generadores			
nuevo	: conj(jugador)	\rightarrow	ajedrez
cargarResultado	: jugador ganador \times jugador perdedor \times blancas jug \times ajedrez aj	\rightarrow	ajedrez
			$\{ganador \in jugadores(aj) \wedge perdedor \in jugadores(aj) \wedge jug \in \{ganador, perdedor\}\}$
generarClasico	: jugador j1 \times jugador j2 \times ajedrez aj	\rightarrow	ajedrez
			$\{j1 \in jugadores(aj) \wedge j2 \in jugadores(aj) \wedge j2 \notin clasicos(j1, aj) \wedge j1 \notin clasicos(j2, aj)\}$
observadores básicos			
jugadores	: ajedrez	\rightarrow	conj(jugador)
clasicos	: jugador j \times ajedrez aj	\rightarrow	conj(jugador)
			$\{j \in jugadores(aj)\}$
partidas	: ajedrez	\rightarrow	conj(partida)
otras operaciones			
posiciones	: ajedrez	\rightarrow	secu(jugador)
granMaestro	: ajedrez	\rightarrow	jugador
Fin TAD			

A la hora del diseño se elige la siguiente estructura de representación:

ajedrez **se representa con** estr, donde

estr es tupla \langle posiciones: secu(jugador), granMaestro: jugador),
 clásicos: dicc(jugador, conj(jugador)),
 partidos: conj(partida) \rangle

en la cual *posiciones* es la tabla de posiciones de los jugadores; *granMaestro* es algún jugador que de los que más puntos tienen; *clásicos* indica, dado un jugador, quiénes son sus clásicos; y *partidos* es el conjunto de partidos jugados.

Se pide:

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