Algoritmos y Estructuras de Datos II
Primer parcial – Miércoles 16 de Noviembre 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, número total de hojas, apellido, nombre y número de Libreta Universitaria**.
- Al entregar el parcial, completar el resto de las columnas en la planilla.
- Cada ejercicio se calificará con **P**romocionado, **A**probado, **R**egular, o **I**nsuficiente.
- 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

En un campo de *paintball* compiten en un gran torneo diversos equipos de jugadores. Se desea construir un sistema que mantenga cierta información sobre los equipos que todavía no han sido eliminados. Cada tanto ingresan al predio grupos de integrantes de un mismo equipo (pudiendo haber jugadores del equipo entrante ya presentes en el campo).

Algunos jugadores son más habilidosos que otros. El nivel de habilidad de un jugador se representa con un número natural donde 0 significa "jugador principiante". Dado un equipo, su habilidad total se define como la suma de las habilidades de cada uno de sus jugadores actuales. Durante la batalla sucede que si la cantidad de jugadores de un equipo es inferior a la cantidad de jugadores de otro equipo de habilidad total mayor, entonces el primer equipo es (dolorosamente) eliminado de la batalla de manera inmediata. Dado esto, notar que cuando un grupo de jugadores ingresa al campo, varios equipos pueden resultar eliminados, incluido el equipo del grupo que está ingresando. Por cuestiones comerciales y de balance de la competencia, los organizadores del encuentro prohíben ingresar un grupo de jugadores si eso causara que más de la mitad de la cantidad total de jugadores que había en el campo sea eliminada. Un jugador abandona el campo de juego únicamente cuando su equipo es eliminado: nunca se eliminan jugadores de un equipo individualmente.

El sistema debe poder contestar cuáles son los equipos que compiten actualmente en el campo de juego y cuántos jugadores hay de un equipo determinado.

Ej. 2. Inducción estructural

Dadas las siguientes funciones sobre árboles binarios ya presentadas en la práctica 2 y el apunte de TADs básicos:

nil? : ab(α) $\longrightarrow$ bool

n_1) nil?(nil) $\equiv$ true

n_2) nil?(bin(i, e, d)) $\equiv$ false

h : ab(α) $\longrightarrow$ nat

h_1) h(nil) $\equiv$ 0

h_2) h(bin(i, e, d)) $\equiv$ máx(h(i), h(d)) + 1

#Hojas : ab(α) $\longrightarrow$ nat

$\#h_1$) #Hojas(nil) $\equiv$ 0

$\#h_2$) #Hojas(bin(i, e, d)) $\equiv$ **if** nil?(i) $\wedge$ nil?(d) **then**
 1
else
 #Hojas(i) + #Hojas(d)
fi

#Internos : ab(α) $\longrightarrow$ nat

$\#i_1$) #Internos(nil) $\equiv$ 0

$\#i_2$) #Internos(bin(i, e, d)) $\equiv$ **if** nil?(i) $\wedge$ nil?(d) **then**
 0
else
 #Internos(i) + #Internos(d) + 1
fi

Se quiere probar por inducción estructural la siguiente propiedad:

$$(\forall a\colon \text{ab}(\alpha))(\neg\ \text{nil?}(a) \ \Rightarrow\ \#\text{Internos}(a) \leq \#\text{Hojas}(a) \times (\text{h}(a) - 1)\ )$$

a) Escribir el predicado unario a utilizar en la demostración.
b) Dar el esquema de inducción a utilizar.
c) Plantear el/los caso/s base/s y resolverlo, justificando cada paso de la demostración.
d) Plantear el/los paso/s inductivo/s, marcando claramente la hipótesis, tesis inductiva y el alcance de los cuantificadores. Resolver justificando cada paso de la demostración.

Ej. 3. Diseño - Rep & Abs

En la siguiente especificación se ilustra el comportamiento de una carrera de bicicletas. En una carrera participan n ciclistas, numerados de 1 a n. Las bicis largan todas juntas y recorren la pista, hasta eventualmente cruzar la meta. Durante la carrera pueden ocurrir choques, en cuyo caso los participantes del choque (que pueden ser cualquier cantidad de bicicletas) finalizan su participación en la carrera. A medida que las bicis cruzan la meta, se les asigna su posición. Todas las bicis accidentadas comparten la posición n.

TAD CarreraBicicletas

observadores básicos

participantes : carreraBicis $\longrightarrow$ nat
enCarrera : carreraBicis $\times$ nat $\longrightarrow$ bool
cruzaronMeta : carreraBicis $\longrightarrow$ nat
posicion : carreraBicis c $\times$ nat n $\longrightarrow$ nat $\{1 \leq n \leq$ participantes(c) $\land \neg$ enCarrera$(c,n)\}$

generadores

nuevaCarrera : nat n $\longrightarrow$ carreraBicis $\{n > 2\}$
cruzarMeta : carreraBicis c $\times$ nat n $\longrightarrow$ carreraBicis $\{$enCarrera$(c,n)\}$
choque : carreraBicis c $\times$ conj(nat) p $\longrightarrow$ carreraBicis $\{\forall n, n \in p \Rightarrow_{\mathrm{L}}$ enCarrera$(c,n)\}$

axiomas

participantes(nuevaCarrera(p)) $\equiv$ p
participantes(cruzarMeta(c, n)) $\equiv$ participantes(c)
participantes(choque(c, p)) $\equiv$ participantes(c)
enCarrera(nuevaCarrera(p), n) $\equiv$ $1 \leq n \land n \leq p$
enCarrera(cruzarMeta(c,n'), n) $\equiv$ enCarrera(c,n) $\land n \neq n'$
enCarrera(choque(c,p), n) $\equiv$ enCarrera(c,n) $\land n \notin p$
cruzaronMeta(nuevaCarrera(p)) $\equiv$ 0
cruzaronMeta(cruzarMeta(c,n')) $\equiv$ 1 + cruzaronMeta(c)
cruzaronMeta(choque(c,p)) $\equiv$ cruzaronMeta(c)
posicion(cruzarMeta(c, n), n') $\equiv$ **if** $n = n'$ **then** 1 + cruzaronMeta(c) **else** posicion(c,n') **fi**
posicion(choque(c,p), n) $\equiv$ **if** $n \in p$ **then** participantes(c) **else** posicion(c, n) **fi**

Fin TAD

Se decidió utilizar la siguiente estructura para representar el TAD.

carreraBicis **se representa con** estr, donde

estr es tupla $\langle$ *posiciones*: conj(tupla $\langle$*bici*: nat, *posición*: nat$\rangle$),
chocadas: conj(nat)
enCarrera: conj(nat) $\rangle$

donde *posiciones* contiene tuplas indicando qué bicis cruzaron la meta y en qué posición; *chocadas* es el conjunto de bicis que chocaron; y *enCarrera* las bicis que quedan en competencia.

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