

# Algoritmos y Estructuras de Datos II

## Recuperatorio del primer parcial – 3 de julio 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 MB, B, R o M, y podrá recuperarse independientemente de los demás. La cursada podrá aprobarse con hasta 1 (un) ejercicio R (regular) en cada parcial (post-recuperatorios), siempre y cuando ninguno de ellos sea un ejercicio 1. Para más detalles, ver “Información sobre la cursada” en el sitio Web.

### Ej. 1. Especificación

Antes de cada parcial, los docentes de la materia con más onda de cierto departamento de computación organizan las compras para la picada. Durante la organización la gente va proponiendo nuevos ingredientes a comprar, junto con sus precios por unidad. A la vez, quienes desean colaborar para la picada se van anotando, informando cuánto dinero pueden aportar y cuál de los ingredientes propuestos hasta ese momento (que todavía no haya sido comprado) preferirían que se adquiriera.

Como la fecha del parcial se acerca rápidamente, los organizadores están un poco ansiosos por concretar las compras cuanto antes. Por eso, ni bien un ingrediente comprable tiene al menos 3 votos y además su cantidad de votos supera estrictamente el 50 % del total de votos recibidos hasta el momento (contando todos los votos no descartados), inmediatamente se compra todo lo que se pueda comprar de dicho ingrediente, y éste pasa al registro de ingredientes ya adquiridos. Nótese que una compra podría desencadenar otras.

Los votos correspondientes a ingredientes ya comprados se descartan (se tiran a la basura), junto con el precio que tenían dichos ingredientes al momento de ser comprados. Sólo interesa saber qué ingredientes se compraron y en qué cantidad. Con respecto al dinero, sólo interesa saber cuánto de lo recaudado sobra hasta el momento.

Como esta modalidad genera interminables discusiones entre los involucrados, se pide modelar formalmente el proceso utilizando tipos abstractos de datos.

### Ej. 2. Inducción estructural

Considerar el siguiente tipo abstracto de datos y las operaciones asociadas:

| TAD CONTROLADORDEPRECIO |                          |             |                |
|-------------------------|--------------------------|-------------|----------------|
| observadores básicos    |                          |             |                |
| piso                    | : cdp                    | → nat       |                |
| techo                   | : cdp                    | → nat       |                |
| actual                  | : cdp                    | → nat       |                |
| estuvoEn                | : cdp                    | → conj(nat) |                |
| generadores             |                          |             |                |
| crear                   | : nat $p \times$ nat $t$ | → cdp       | $\{p \leq t\}$ |
| variar                  | : cdp $\times$ bool      | → cdp       |                |
| avanzar                 | : cdp                    | → cdp       |                |
| Fin TAD                 |                          |             |                |

piso : cdp → nat  
 $p_0$ ) piso(crear( $p, t$ )) ≡  $p$   
 $p_1$ ) piso(variar( $c, d$ )) ≡ piso( $c$ )  
 $p_2$ ) piso(avanzar( $c$ )) ≡ piso( $c$ )

techo : cdp → nat  
 $t_0$ ) techo(crear( $p, t$ )) ≡  $t$   
 $t_1$ ) techo(variar( $c, d$ )) ≡ techo( $c$ )  
 $t_2$ ) techo(avanzar( $c$ )) ≡ techo( $c$ )

actual : cdp → nat  
 $a_0$ ) actual(crear( $p, t$ )) ≡  $p$   
 $a_1$ ) actual(variar( $c, d$ )) ≡ if  $d$  then  
                                     mín(actual( $c$ ) + 1, techo( $c$ ))  
                                     else  
                                     máx(actual( $c$ ) - 1, piso( $c$ ))  
                                     fi  
 $a_2$ ) actual(avanzar( $c$ )) ≡ actual( $c$ )

mín : nat  $\times$  nat → nat  
 $min_0$ ) mín( $a, b$ ) ≡ if  $a \leq b$  then  
                                      $a$   
                                     else  
                                      $b$   
                                     fi

estuvoEn : cdp → conj(nat)  
 $e_0$ ) estuvoEn(crear( $p, t$ )) ≡  $\emptyset$   
 $e_1$ ) estuvoEn(variar( $c, d$ )) ≡ Ag(actual( $c$ ), estuvoEn( $c$ ))  
 $e_1$ ) estuvoEn(avanzar( $c$ )) ≡ Ag(actual( $c$ ), estuvoEn( $c$ ))

máx : nat  $\times$  nat → nat  
 $max_0$ ) máx( $a, b$ ) ≡ if  $a \geq b$  then  
                                      $a$   
                                     else  
                                      $b$   
                                     fi

Interesa demostrar por inducción estructural la siguiente propiedad:

$$(\forall c : \text{cdp}) (\text{piso}(c) \leq \text{actual}(c) \leq \text{techo}(c))$$

- Escribir el predicado unario. Luego escribir, completo, **el esquema de inducción** a utilizar.  
En el esquema, marcar **claramente** todos los 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.

En caso de utilizar lemas, plantearlos claramente y **demostrarlos**.

### Ej. 3. Diseño

La fiambrería “La Longa Niza”, que provee los salamines para las picadas organizadas (ver ??), ha decidido incorporar promociones como una estrategia de marketing. Para esto nos ha pedido que diseñemos un sistema que maneje las promociones de los ingredientes que venden. Cada promoción afecta a un conjunto de ingredientes y determina un descuento a aplicar sobre ellos expresado como un porcentaje sobre el precio final. Si un ingrediente es afectado por varias promociones, los descuentos de todas ellas se suman, aunque en ningún caso el descuento sobre un ingrediente podrá superar el 40 %.

Para cada ingrediente se desea conocer su precio real y su precio con descuentos. Además se desea poder agregar nuevas promociones y consultar los ingredientes de una determinada promoción en todo momento.

Los ingredientes se identificarán con un nombre de hasta 20 letras, y las promociones con un número natural.

Se provee a continuación la especificación del sistema.

**TAD** PROMOCIÓN, DESCUENTO, PRECIO **ES** NAT  
**TAD** INGREDIENTE **ES** STRING[20]  
**TAD** LALONGANIZA

#### observadores básicos

|                     |   |                                      |   |                   |                                    |
|---------------------|---|--------------------------------------|---|-------------------|------------------------------------|
| promociones         | : | longaniza                            | → | conj(promoción)   |                                    |
| ingredientes        | : | longaniza                            | → | conj(ingrediente) |                                    |
| promosQueAfectan    | : | longaniza $l \times$ ingrediente $i$ | → | conj(promoción)   | $\{i \in \text{ingredientes}(l)\}$ |
| descuentosOfrecidos | : | longaniza $l \times$ promoción $p$   | → | descuento         | $\{p \in \text{promociones}(l)\}$  |
| Precio              | : | longaniza $l \times$ ingrediente $i$ | → | precio            | $\{i \in \text{ingredientes}(l)\}$ |

#### generadores

|               |   |  |   |           |   |
|---------------|---|--|---|-----------|---|
| longaniza     |   |  |   |           |   |
| crear         | : |  | → | longaniza |   |
| agIngrediente | : | longaniza $l \times$ ingrediente $i \times$ precio $v$                               | → | longaniza | $\{i \notin \text{ingredientes}(l)\}$   |
| agPromocion   | : | longaniza $l \times$ promoción $p \times$ conj(ingrediente) $c \times$ descuento $v$ | → | longaniza | $\left\{ \begin{array}{l} c \subseteq \text{ingredientes}(l) \wedge p \notin \text{promociones}(l) \wedge \\ (\forall i' : \text{ingrediente})(i' \in c \Rightarrow \text{sumaDeDescuentos}(l, i') + v \leq 40) \end{array} \right\}$ |

#### otras operaciones

|                          |   |                                      |   |                   |                                    |
|--------------------------|---|--------------------------------------|---|-------------------|------------------------------------|
| sumaDeDescuentos         | : | longaniza $l \times$ ingrediente $i$ | → | descuento         | $\{i \in \text{ingredientes}(l)\}$ |
| precioFinal              | : | longaniza $l \times$ ingrediente $i$ | → | precio            | $\{i \in \text{ingredientes}(l)\}$ |
| ingredientesAfectadosPor | : | longaniza $l \times$ promoción $p$   | → | conj(ingrediente) | $\{p \in \text{promociones}(l)\}$  |

**Fin TAD**

La cátedra ha armado la siguiente estructura de representación para el diseño del sistema:

LaLongaNiza **se representa con** estr,

donde estr es tupla  $\langle \text{promociones} : \text{conj}(\text{promoción}), \text{promosXIngrediente} : \text{dicc}(\text{promoción}, \text{infoPromocion}),$   
 $\text{datosIngredientes} : \text{dicc}(\text{ingrediente}, \text{infoIngrediente}) \rangle$

infoPromocion es tupla  $\langle \text{descuento} : \text{nat}, \text{ingredientesAfectados} : \text{conj}(\text{ingrediente}) \rangle$

infoIngrediente es tupla  $\langle \text{precioFinal} : \text{nat}, \text{promosQueAfectan} : \text{conj}(\text{promoción}), \text{descuentos} : \text{multiconj}(\text{nat}) \rangle$

Se pide:

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