

# Algoritmos y Estructuras de Datos II

## Práctica 3 – Diseño: invariante de representación y función de abstracción

### Notas preliminares

- Los ejercicios marcados con el símbolo ★ constituyen un subconjunto mínimo de ejercitación. Sin embargo, aconsejamos fuertemente hacer todos los ejercicios.
- Para todos los ejercicios de esta parte, salvo indicación contraria, se pide definir interfaz imperativa (pre y post condición y aspectos de aliasing, no la complejidad), estructura e invariante de representación, función de abstracción y los algoritmos que implementen las operaciones.

### Ejercicio 1 ★

Diseñar el TAD COLA(NAT) y el TAD PILA(NAT) utilizando una secuencia como estructura de representación.

### Ejercicio 2

Diseñar el TAD CONJUNTO( $\alpha$ ) sobre secuencia.

### Ejercicio 3

El siguiente TAD modela una cola que puede contener a lo sumo  $n$  elementos.

#### TAD COLA ACOTADA( $\alpha$ )

##### observadores básicos

verCola :  $\text{cacotada}(\alpha) \rightarrow \text{cola}(\alpha)$   
 capacidad :  $\text{cacotada}(\alpha) \rightarrow \text{nat}$

##### generadores

vacía :  $\text{nat} \rightarrow \text{cacotada}(\alpha)$   
 encolar :  $\alpha \times \text{cacotada}(\alpha) \rightarrow \text{cacotada}(\alpha)$   $\{( \text{tamaño}(\text{verCola}(c)) < \text{capacidad}(c) )\}$

##### axiomas

$\text{verCola}(\text{vacía}(c)) \equiv \text{vacía}$   
 $\text{verCola}(\text{encolar}(a, c)) \equiv \text{encolar}(a, \text{verCola}(c))$   
 $\text{capacidad}(\text{vacía}(n)) \equiv n$   
 $\text{capacidad}(\text{encolar}(a, c)) \equiv \text{capacidad}(c)$

**Fin TAD**

Se propone la siguiente estructura de representación.

$\text{cacotada}$  se representa con  $\text{estr}$ , donde

$\text{estr}$  es tupla  
 $\langle \text{inicio: nat,}$   
 $\text{fn: nat,}$   
 $\text{elem: array}[0 \dots n] \text{ de } \alpha \rangle$

Se pide:

- Definir el invariante de representación y la función de abstracción.
- Escribir la interface completa y todos los algoritmos.

**Ejercicio 4 ★**

Se propone la siguiente estructura para representar a los polinomios que tienen a lo sumo grado  $n$  (ver TAD en Práctica 1):

polinomio **se representa con** *estr*, donde

*estr* es tupla

$\langle grado: nat,$   
 $coef: array[0 \dots n] \text{ de } nat \rangle$

Se pide:

- Definir el invariante de representación y la función de abstracción.
- Escribir la interface completa y el algoritmo para la función evaluar.

**Ejercicio 5**

Los palíndromos son aquellas palabras que pueden leerse al derecho o al revés. El siguiente TAD describe a los palíndromos:

**TAD** PALÍNDROMO( $\alpha$ )

**observadores básicos**

*ver* : palindromo( $\alpha$ )  $\longrightarrow$  secu( $\alpha$ )

**generadores**

*medio* :  $\alpha \longrightarrow$  palindromo( $\alpha$ )

*medioDoble* :  $\alpha \longrightarrow$  palindromo( $\alpha$ )

*agregar* :  $\alpha \times \text{palindromo}(\alpha) \longrightarrow \text{palindromo}(\alpha)$

**axiomas**

*ver*(*medio*( $a$ ))  $\equiv a \bullet \langle \rangle$

*ver*(*medioDoble*( $a$ ))  $\equiv a \bullet a \bullet \langle \rangle$

*ver*(*agregar*( $a, p$ ))  $\equiv a \bullet (\text{ver}(p) \circ a)$

**Fin TAD**

Se propone la siguiente estructura de representación:

polindromo **se representa con** *estr*, donde

*estr* es tupla

$\langle long: nat,$   
 $palabra: \text{secu}(\alpha) \rangle$

dónde *palabra* representa el palíndromo completo.

Se pide:

- Definir el invariante de representación y la función de abstracción.
- Escribir la interface y el algoritmo para la función *ver*.
- Rehacer los ítems anteriores si el campo *palabra* en lugar de la palabra completa guardamos sólo la mitad inicial de la palabra (redondeando hacia arriba).

**Ejercicio 6 ★**

Diseñar el TAD CONJUNTO DE NATURALES EN RANGO utilizando un arreglo. (El arreglo debe contener la mayor parte de la información, pero la estructura de representación puede contener la información adicional que sea necesaria).

#### TAD CONJUNTO DE NATURALES EN RANGO

**géneros**          `conjenrango`

##### observadores básicos

$\bullet \in \bullet : \text{nat} \times \text{conjenrango} \longrightarrow \text{bool}$   
`lower` : `conjenrango`  $\longrightarrow$  `nat`  
`upper` : `conjenrango`  $\longrightarrow$  `nat`

##### generadores

$\emptyset : \text{nat } n \times \text{nat } m \longrightarrow \text{conjenrango} \quad \{(n \leq m)\}$   
`Ag` : `nat a`  $\times$  `conjenrango c`  $\longrightarrow$  `conjenrango`  $\{(\text{lower}(c) \leq a \leq \text{upper}(c))\}$

##### axiomas

`lower`( $\emptyset(n, m)$ )  $\equiv n$   
`lower`(`Ag`(`a`, `c`))  $\equiv \text{lower}(c)$

`upper`( $\emptyset(n, m)$ )  $\equiv m$   
`upper`(`Ag`(`a`, `c`))  $\equiv \text{upper}(c)$

$a \in \emptyset(n, m) \equiv \text{false}$   
 $a \in \text{Ag}(b, c) \equiv (a =_{\text{nat}} b) \vee (a \in c)$

**Fin TAD**

#### Ejercicio 7 ★

Diseñar el TAD CONJUNTO AJUSTADO DE NATURALES teniendo en cuenta lo siguiente:

“Los elementos que serán representados por el diseño propuesto pertenecerán *mayoritariamente* al rango acotado que se indica al momento de crear el conjunto ajustado.”

Proponga una representación que tenga en cuenta este contexto de uso.

Considere la resolución del ejercicio anterior. ¿Se ajusta a lo pedido? Si es así, intente encontrar otra solución distinta; si no, ¿encuentra alguna forma de utilizarla?

#### TAD CONJUNTO AJUSTADO DE NATURALES

**géneros**          `conjajust`

##### observadores básicos

$\bullet \in \bullet : \text{nat} \times \text{conjajust} \longrightarrow \text{bool}$   
`lower` : `conjajust`  $\longrightarrow$  `nat`  
`upper` : `conjajust`  $\longrightarrow$  `nat`

##### generadores

$\emptyset : \text{nat } n \times \text{nat } m \longrightarrow \text{conjajust} \quad \{(n \leq m)\}$   
`Ag` : `nat a`  $\times$  `conjajust c`  $\longrightarrow$  `conjajust`

##### axiomas

`lower`( $\emptyset(n, m)$ )  $\equiv n$   
`lower`(`Ag`(`a`, `c`))  $\equiv \text{lower}(c)$

`upper`( $\emptyset(n, m)$ )  $\equiv m$   
`upper`(`Ag`(`a`, `c`))  $\equiv \text{upper}(c)$

$a \in \emptyset(n, m) \equiv \text{false}$   
 $a \in \text{Ag}(b, c) \equiv a = b \vee a \in c$

**Fin TAD**

**Ejercicio 8 ★**

Se propone la siguiente estructura para representar el TAD árbol binario:

ab **se representa con** puntero(estr), donde

estr es tupla  
 $\langle altura: nat,$   
 $izq: puntero(estr),$   
 $raiz: \alpha,$   
 $der: puntero(estr) \rangle$

Se pide:

- Definir el invariante de representación.
- Función de abstracción usando los observadores.
- Función de abstracción utilizando los generadores.

**Ayuda:** Tanto el invariante de representación como la función de abstracción pueden definirse recursivamente.

**Ejercicio 9**

Tomando como partida el TAD *rosetree*( $\alpha$ ) presentado en la práctica de tipos abstractos de datos, proponer:

- Una estructura de representación.
- Definir el invariante de representación.
- Definir la función de abstracción.

**Ejercicio 10**

Diseñar el TAD DICCIONARIO(String, Nat).

- Represente los diccionarios sobre secuencias.
- Ahora adapte el diseño a la siguiente información sobre el contexto: es habitual preguntarle al diccionario por la existencia de una clave para luego, en caso de existir, pedir su significado.

**Ejercicio 11**

Diseñe el TAD MULTICONJUNTO( $\alpha$ ), considerando un contexto de uso en el que los multiconjuntos suelen tener pocos elementos distintos y abundante cantidad de repeticiones de cada uno. Escriba los algoritmos para las siguientes operaciones:

- *Vacío*, que devuelve un multiconjunto sin elementos.
- *Agregar*, que agrega una sola repetición del elemento indicado.
- *Eliminar*, que elimina una sola repetición del elemento indicado.
- *#Repeticiones*, que devuelve la cantidad de repeticiones del elemento indicado que hay en el multiconjunto.

**Ejercicio 12 ★**

Se decidió utilizar la siguiente estructura para representar una fila en un banco (Ejercicio 10, Práctica 1):

banco **se representa con** estr, donde

estr es tupla  
 $\langle entraron: conj(persona),$   
 $fila: cola(persona),$   
 $colados: conj(persona),$   
 $atendidos: conjunto(persona) \rangle$

Donde:

- *Entraron* es un conjunto con todas las personas que alguna vez estuvieron en la fila.
- *Colados* son las personas que están actualmente en la fila y se colaron al llegar.
- *Atendidos* son las personas que fueron atendidas en el banco.

Escribir invariante de representación y función de abstracción.

### Ejercicio 13

Dado el siguiente TAD:

#### TAD SECUNDARIO

##### observadores básicos

alumnos	: secundario	→ conj(alumno)	
#faltas	: alumno $a \times$ secundario $s$	→ nat	$\{a \in \text{alumnos}(s)\}$
notas	: alumno $a \times$ secundario $s$	→ multiconj(nota)	$\{a \in \text{alumnos}(s)\}$

##### generadores

nuevo	: conj(alumno)	→ secundario	
nota	: alumno $a \times$ nota $n \times$ secundario $s$	→ secundario	$\{a \in \text{alumnos}(s)\}$
falta	: alumno $a \times$ secundario $s$	→ secundario	$\{a \in \text{alumnos}(s)\}$

**Fin TAD**

Se pide:

- Proponer una estructura de representación.
- Definir el invariante de representación y la función de abstracción.

### Ejercicio 14 (*Alta fiesta*) ★

El salón Alta Fiesta se encuentra últimamente en dificultades para coordinar el desarrollo de las reuniones que se realizan en sus facilidades. Hoy en día las fiestas se desarrollan de una manera muy particular. Los invitados llegan en grupos, por lo general numerosos, aunque también a veces de una sola persona. Que un invitado llegue sin un regalo está considerado una falta grave a las buenas costumbres. Tanto es así que a dichos individuos no se les permite el ingreso a las fiestas. Lo que sí se permite es que los invitados hagan regalos en grupo: los invitados que llegan en grupo traen un único regalo de parte de todos. Como es habitual, sólo se permite la entrada a la fiesta a aquellas personas que han sido invitadas.

Al ingresar un grupo de invitados a la fiesta, éste se identifica con un nombre: por ejemplo “Los amigos de la secundaria” o “La familia de la novia”. Este nombre se usa por ejemplo para los juegos grupales que van a hacer los animadores durante la fiesta. Igualmente, dado que el comportamiento de las personas en masa no siempre es civilizado, se quiere poder saber de manera eficiente cuál es el nombre del grupo más numeroso para poder seguirle el rastro.

Además, se desea tener un registro de todos los regalos, junto con el grupo de personas que lo hicieron, y se desea conocer en todo momento qué personas se encuentran ya en la fiesta.

Se nos ha encomendado realizar un sistema que permite llevar el control del estado de la fiesta en un momento dado. La cátedra ha realizado la especificación del sistema y ha armado una estructura de representación para el diseño del mismo.

altafiesta **se representa con** estr, donde

estr es tupla

$\langle$ invitados: conj(persona),  
 presentes: cola(persona),  
 grupoDe: dicc(grupo, conj(persona)),  
 regaloDeGrupo: dicc(grupo, regalo),  
 grupoMasNumeroso: grupo $\rangle$

grupo, persona y regalo **son** string

Informalmente, esta representación cumple las siguientes propiedades:

- En *invitados* están todos los invitados a la fiesta, incluyendo también a aquellos que ya llegaron.
- En *presentes* están los invitados que ya llegaron a la fiesta.
- En *grupoDe* se encuentra, para cada identificador de grupo  $i$ , las personas que al llegar agrupadas se identificaron como  $i$ .
- En *regaloDeGrupo* se encuentra qué regalo trajo cada grupo.
- *grupoMasNumeroso* contiene al identificador del grupo de más personas. En caso de empate, contiene al lexicográficamente menor de los empatados (se asume que la función  $<_{string}$  está definida).

Se pide:

- a) Realizar el invariante de representación del módulo. Escribirlo en lenguaje formal y en castellano. Para que quede claro cuáles enunciados informales hacen referencia a cuáles enunciados formales se recomienda numerar cada punto del invariante para luego poder hacer referencia al mismo.
- b) Escribir la función de abstracción. De considerarse necesario, explicarla en castellano.
- c) Escribir una versión imperativa de la función *llegaGrupo* marcando claramente los puntos de su programa en que alguna parte del invariante de representación se rompe, indicando a qué parte se refiere, y también aquellos puntos donde éste se reestablece.

### Ejercicio 15 (Planilla de actividades justificación)

Un consultor independiente desea mantener una planilla con las actividades que realiza cada mes en cada uno de los proyectos en los que participa. La planilla que desea mantener se describe con el siguiente TAD.

#### TAD PLANILLA

##### observadores básicos

actividades	: planilla	→ conjunto(actividad)	
proyectos	: planilla	→ conjunto(proyecto)	
proyecto	: actividad $a \times$ planilla $p$	→ proyecto	$\{(a \in \text{actividades}(p))\}$
mes	: actividad $a \times$ planilla $p$	→ mes	$\{(a \in \text{actividades}(p))\}$
horas	: actividad $a \times$ planilla $p$	→ horas	$\{(a \in \text{actividades}(p))\}$

##### generadores

nueva	:		→ planilla	
ag	: actividad $a \times$ proyecto $p \times$ mes $m \times$ horas $h \times$ planilla $q$	→ planilla		$\{a \notin \text{actividades}(q)\}$

##### otras operaciones

totProyxMes	: proyecto $p \times$ mes $m \times$ planilla $q$	→ planilla	$\{(p \in \text{proyectos}(q))\}$
proysMasHoras	: planilla	→ conj(proyecto)	
...			

Fin TAD

Se propone la siguiente estructura para representar dicho TAD

planilla se representa con *estr*, donde

*estr* es tupla

$\langle \text{detalle: dicc(actividad, tupla}\langle \text{proy: proyecto, mes: mes, horas: nat} \rangle \rangle,$   
 $\text{horasPorMes: dicc(proyecto, array[mes] de horas),}$   
 $\text{ConMasHoras: conj(proyectos)} \rangle$

mes es un entero en el rango  $1 \dots 12$

Se pide:

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

### Ejercicio 16 (*Oficina estatal*) ★

Considerar el siguiente TAD que modela el comportamiento de una oficina del Estado que procesa trámites. Cada trámite está identificado por un ID y se le asigna una categoría al momento de ingresar. Las categorías de la oficina no son fijas, y pueden agregarse nuevas categorías en cualquier momento. En cualquier momento se puede dar prioridad a una categoría. Todos los trámites pendientes que pertenecen a una categoría prioritaria se consideran prioritarios (Notar que en esta oficina, como buena dependencia estatal, un trámite nunca concluye):

#### TAD OFICINA

**géneros**            oficina

#### observadores básicos

categorias	:	oficina	→	conj(categoria)	
pendientes	:	oficina	→	secu(id)	
prioritarias	:	oficina	→	conj(categoria)	
catTram	:	id $i \times$ oficina $o$	→	categoria	$\{(i \in \text{pendientes}(o))\}$

#### generadores

nuevo	:		→	oficina	
nuevaCat	:	categoria $c \times$ oficina $o$	→	oficina	$\{(c \notin \text{categorias}(o))\}$
nuevoTram	:	id $i \times$ categoria $c \times$ oficina $o$	→	oficina	$\{(i \notin \text{pendientes}(o) \wedge c \in \text{categorias}(o))\}$
priorizar	:	categoria $c \times$ oficina $o$	→	oficina	$\{(c \in \text{categorias}(o))\}$

#### otras operaciones

pendPrioritarios	:	oficina	→	secu(id)	
filtrarPorCategorias	:	secu(id) $s \times$ conj(categoria) $\times$ oficina $o$	→	secu(id)	$\{((\forall i : \text{nat})(0 \leq i < \text{long}(s) \Rightarrow_L s[i] \in \text{pendientes}(o)))\}$

...

**Fin TAD**

Se decidió utilizar la siguiente estructura como representación:

oficina **se representa con** *estr*, donde

*estr* es tupla

$\langle \text{catPrioritarias: conj(categoria)},$   
*tramites*: *dicc*(id, categoria),  
*tramites*× *Cat*: *dicc*(categoria, conj(id)),  
*pendPrioritarios*: *secu*(id),  
*pendientes*: *secu*(id)  $\rangle$

Informalmente, *catPrioritarias* representa el conjunto de todas las categorías a las que se ha dado prioridad, *tramites* asocia a cada trámite su categoría mientras que *tramites*× *Cat* describe todos los trámites asociados a cada categoría. *pendPrioritarios* contiene la secuencia de trámites pendientes que tienen una categoría prioritaria mientras que *pendientes* contiene todos los trámites pendientes (incluso a los prioritarios).

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

### Ejercicio 17 (*Planta industrial*) ★

Considere la siguiente especificación que modela el funcionamiento de alarmas en una planta industrial. La planta cuenta con un conjunto de alarmas asociadas a distintos sensores. Cada sensor está asociado a varias alarmas y cada alarma puede tener varios sensores asociados. Una alarma está activa cuando la medición de

al menos uno de sus sensores asociados supera el valor umbral definido para ese sensor. Considere la siguiente especificación.

### TAD PLANTA

#### observadores básicos

esAlarma	:	planta $\times$ alarma	$\longrightarrow$	bool	
esSensor	:	planta $\times$ sensor	$\longrightarrow$	bool	
sensoresAlarma	:	planta $p \times$ alarma $a$	$\longrightarrow$	conj(sensor)	$\{esAlarma(p, a)\}$
umbral	:	planta $p \times$ sensor $s$	$\longrightarrow$	nat	$\{esSensor(p, s)\}$
medicion	:	planta $p \times$ sensor $s$	$\longrightarrow$	nat	$\{esSensor(p, s)\}$

#### generadores

crear	:		$\longrightarrow$	planta	
agregarAlarma	:	planta $p \times$ alarma $a$	$\longrightarrow$	planta	$\{\neg esAlarma(p, a)\}$
agregarSensor	:	planta $p \times$ sensor $s \times$ nat $n \times$ conj(alarma) $c$	$\longrightarrow$	planta	$\{\neg esSensor(p, s) \wedge (\forall a: alarma) (a \in c \Rightarrow esAlarma(p, a)) \wedge n > 0\}$
nuevaMedicion	:	planta $p \times$ sensor $s \times$ nat	$\longrightarrow$	planta	$\{esSensor(p, s)\}$

#### otras operaciones

encendida	:	planta $p \times$ alarma $a$	$\longrightarrow$	bool	$\{esAlarma(p, a)\}$
...					

**Fin TAD**

Se decidió utilizar la siguiente estructura como representación:

planta **se representa con** estr, donde

estr es tupla

$\langle alarmas: \text{dicc}(alarma, \text{conj}(\text{sensor})),$   
 $sensores: \text{dicc}(\text{sensor}, \text{tupla}(\langle alarmas: \text{conj}(alarma), umbral: \text{nat}, medición: \text{nat} \rangle)) \rangle$

donde *alarmas* es el diccionario que asocia a cada alarma el conjunto de los sensores asociados a la misma **que están encendidos**, y *sensores* es el diccionario que asocia a cada sensor el conjunto de alarmas que el mismo enciende, el umbral y el valor de la última medición.

Se pide:

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