Tipos abstractos de datos básicos

Algoritmos y Estructuras de Datos II, DC, UBA.

Índice

1. TAD Bool

TAD Bool

géneros bool

exporta bool, generadores, $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\vee_{\text{L}}$, $\wedge_{\text{L}}$, $\Rightarrow_{\text{L}}$

igualdad observacional

$$((true =_{\text{obs}} true) \wedge (false =_{\text{obs}} false) \wedge \neg(true =_{\text{obs}} false) \wedge \neg(false =_{\text{obs}} true))$$

bool

generadores

true	:		$\longrightarrow$ bool
false	:		$\longrightarrow$ bool

otras operaciones

$\neg\ \bullet$	:	bool	$\longrightarrow$ bool
$\bullet \vee \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool
$\bullet \wedge \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool
$\bullet \Rightarrow \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool
$\bullet \vee_{\text{L}} \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool
$\bullet \wedge_{\text{L}} \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool
$\bullet \Rightarrow_{\text{L}} \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool

axiomas $\forall\ x, y$: bool

$\neg$ true	$\equiv$	false
$\neg$ false	$\equiv$	true
true $\vee\ x$	$\equiv$	true
false $\vee\ x$	$\equiv$	x
true $\wedge\ x$	$\equiv$	x
false $\wedge\ x$	$\equiv$	false
$x \Rightarrow y$	$\equiv$	$\neg x \vee y$
$x \wedge_{\text{L}} y$	$\equiv$	**if** x **then** y **else** false **fi**
$x \vee_{\text{L}} y$	$\equiv$	**if** x **then** true **else** y **fi**
$x \Rightarrow_{\text{L}} y$	$\equiv$	$\neg x \vee_{\text{L}} y$

Fin TAD

2. TAD Nat

TAD Nat

géneros nat

exporta nat, generadores, observadores, $+$, $-$, $\times$, $<$, $\leq$, mín, máx

usa Bool

igualdad observacional

$$(\forall n, m : \text{nat}) \left(n =_{\text{obs}} m \iff \begin{pmatrix} (n = 0? =_{\text{obs}} m = 0?) \wedge_{\text{L}} \\ (\neg(n = 0?) \Rightarrow_{\text{L}} (\text{pred}(n) =_{\text{obs}} \text{pred}(m))) \end{pmatrix} \right)$$

observadores básicos

$\bullet = 0?$: nat ⟶ bool

pred : nat n ⟶ nat $\{\neg(n = 0?)\}$

generadores

0 : ⟶ nat

suc : nat ⟶ nat

otras operaciones

$\bullet + \bullet$: nat × nat ⟶ nat

$\bullet - \bullet$: nat n × nat m ⟶ nat $\{m \leq n\}$

$\bullet \times \bullet$: nat × nat ⟶ nat

$\bullet < \bullet$: nat × nat ⟶ bool

$\bullet \leq \bullet$: nat × nat ⟶ bool

mín : nat × nat ⟶ nat

máx : nat × nat ⟶ nat

axiomas $\forall$ n, m: nat

$0 = 0?$ ≡ true

suc(n)= 0? ≡ false

pred(suc(n)) ≡ n

$n + m$ ≡ **if** $m = 0?$ **then** n **else** suc(n + pred(m)) **fi**

$n - m$ ≡ **if** $m = 0?$ **then** n **else** pred(n) − pred(m) **fi**

$n \times m$ ≡ **if** $m = 0?$ **then** 0 **else** n × pred(m) + n **fi**

$n < m$ ≡ $\neg(m = 0?) \wedge_{\text{L}} (n = 0? \vee_{\text{L}}\ \text{pred}(n) < \text{pred}(m))$

$n \leq m$ ≡ $n < m \vee n = m$

mín(n, m) ≡ **if** $m < n$ **then** m **else** n **fi**

máx(n, m) ≡ **if** $m < n$ **then** n **else** m **fi**

Fin TAD

3. TAD Tupla(α_1, ..., α_n)

TAD Tupla(α_1, ..., α_n)

igualdad observacional

$$(\forall t, t' : \text{tupla}(\alpha_1, \ldots, \alpha_n))\ (t =_{\text{obs}} t' \iff (\Pi_1(t) =_{\text{obs}} \Pi_1(t') \wedge \ldots \wedge \Pi_n(t) =_{\text{obs}} \Pi_n(t')))$$

parámetros formales

géneros $\alpha_1, \ldots, \alpha_n$

géneros tupla($\alpha_1, \ldots, \alpha_n$)

exporta tupla, generadores, observadores

observadores básicos

Π_1 : tupla$(\alpha_1, \ldots, \alpha_n)$ $\longrightarrow$ α_1

$\vdots$

Π_n : tupla$(\alpha_1, \ldots, \alpha_n)$ $\longrightarrow$ α_n

generadores

$\langle \bullet, \ldots, \bullet \rangle$: $\alpha_1 \times \ldots \times \alpha_n$ $\longrightarrow$ tupla$(\alpha_1, \ldots, \alpha_n)$

axiomas $\forall\ a_1 \colon \alpha_1 \ldots \forall\ a_n \colon \alpha_n$

$\Pi_1(\langle a_1, \ldots, a_n \rangle) \equiv a_1$

$\vdots \equiv \vdots$

$\Pi_n(\langle a_1, \ldots, a_n \rangle) \equiv a_n$

Fin TAD

4. TAD Secuencia(α)

TAD Secuencia(α)

igualdad observacional

$$(\forall s, s' : \text{secu}(\alpha)) \left(s =_{\text{obs}} s' \iff \begin{pmatrix} \text{vacía?}(s) =_{\text{obs}} \text{vacía?}(s') \wedge_{\text{L}} \\ (\neg\ \text{vacía?}(s) \Rightarrow_{\text{L}} (\text{prim}(s) =_{\text{obs}} \text{prim}(s') \wedge \text{fin}(s) =_{\text{obs}} \\ \text{fin}(s')) \end{pmatrix} \right)$$

parámetros formales

géneros α

géneros secu(α)

exporta secu(α), generadores, observadores, &, $\circ$, ult, com, long, está?

usa Bool, Nat

observadores básicos

vacía? : secu(α) $\longrightarrow$ bool

prim : secu(α) s $\longrightarrow$ α $\{\neg$ vacía?$(s)\}$

fin : secu(α) s $\longrightarrow$ secu(α) $\{\neg$ vacía?$(s)\}$

generadores

<> : $\longrightarrow$ secu(α)

$\bullet \bullet \bullet$: $\alpha \times$ secu(α) $\longrightarrow$ secu(α)

otras operaciones

$\bullet \circ \bullet$: secu(α) $\times\ \alpha$ $\longrightarrow$ secu(α)

$\bullet \,\&\, \bullet$: secu(α) $\times$ secu(α) $\longrightarrow$ secu(α)

ult : secu(α) s $\longrightarrow$ α $\{\neg$ vacía?$(s)\}$

com : secu(α) s $\longrightarrow$ secu(α) $\{\neg$ vacía?$(s)\}$

long : secu(α) $\longrightarrow$ nat

está? : $\alpha \times$ secu(α) $\longrightarrow$ bool

axiomas $\forall\ s, t \colon$ secu(α), $\forall\ e \colon \alpha$

vacía?(<>) $\equiv$ true

vacía?(e $\bullet$ s) $\equiv$
false

prim(e $\bullet$ s) $\equiv$ e

fin(e $\bullet$ s) $\equiv$ s

$s \circ e$ $\equiv$ **if** vacía?(s) **then** e $\bullet$ <> **else** prim(s) $\bullet$ (fin(s) $\circ$ e) **fi**

s & t $\equiv$ **if** vacía?(s) **then** t **else** prim(s) $\bullet$ (fin(s) & t) **fi**

ult(s) $\equiv$ **if** vacía?(fin(s)) **then** prim(s) **else** ult(fin(s)) **fi**

com(s) $\equiv$ **if** vacía?(fin(s)) **then** <> **else** prim(s) $\bullet$ com(fin(s)) **fi**

long(s) $\equiv$ **if** vacía?(s) **then** 0 **else** 1 + long(fin(s)) **fi**

está?(e, s) $\equiv$ $\neg$ vacía?(s) $\wedge_{\text{L}}$ (e = prim(s) $\vee$ está?(e, fin(s))

Fin TAD

5. TAD Conjunto(α)

TAD Conjunto(α)

igualdad observacional

$(\forall c, c' : \text{conj}(\alpha))\ \ (c =_{\text{obs}} c' \iff ((\forall a : \alpha)(a \in c =_{\text{obs}} a \in c')))$

parámetros formales

géneros α

géneros conj(α)

exporta conj(α), generadores, observadores, $\emptyset$?, $\cup$, $\cap$, #, $\bullet - \{\bullet\}$, dameUno, sinUno, $\subseteq$, $\bullet - \bullet$

usa Bool, Nat

observadores básicos

$\bullet \in \bullet$: $\alpha \times \text{conj}(\alpha) \longrightarrow \text{bool}$

generadores

$\emptyset$: $\longrightarrow \text{conj}(\alpha)$

Ag : $\alpha \times \text{conj}(\alpha) \longrightarrow \text{conj}(\alpha)$

otras operaciones

$\emptyset$? : $\text{conj}(\alpha) \longrightarrow \text{bool}$

: $\text{conj}(\alpha) \longrightarrow \text{nat}$

$\bullet - \{\bullet\}$: $\text{conj}(\alpha) \times \alpha \longrightarrow \text{conj}(\alpha)$

$\bullet \cup \bullet$: $\text{conj}(\alpha) \times \text{conj}(\alpha) \longrightarrow \text{conj}(\alpha)$

$\bullet \cap \bullet$: $\text{conj}(\alpha) \times \text{conj}(\alpha) \longrightarrow \text{conj}(\alpha)$

dameUno : $\text{conj}(\alpha)\ c \longrightarrow \alpha$ $\{\neg\emptyset?(c)\}$

sinUno : $\text{conj}(\alpha)\ c \longrightarrow \text{conj}(\alpha)$ $\{\neg\emptyset?(c)\}$

$\bullet \subseteq \bullet$: $\text{conj}(\alpha) \times \text{conj}(\alpha) \longrightarrow \text{bool}$

$\bullet - \bullet$: $\text{conj}(\alpha) \times \text{conj}(\alpha) \longrightarrow \text{conj}(\alpha)$

axiomas $\forall$ c, d: conj(α), $\forall$ a, b: α

$a \in \emptyset$	$\equiv$	false
$a \in \text{Ag}(b, c)$	$\equiv$	$(a = b) \vee (a \in c)$
$\emptyset?(\emptyset)$	$\equiv$	true
$\emptyset?(\text{Ag}(b, c))$	$\equiv$	false
$\#(\emptyset)$	$\equiv$	0
$\#(\text{Ag}(a, c))$	$\equiv$	$1 + \#(c$ - $\{ a \})$
$c - \{a\}$	$\equiv$	$c - \text{Ag}(a, \emptyset)$
$\emptyset \cup c$	$\equiv$	c
$\text{Ag}(a, c) \cup d$	$\equiv$	$\text{Ag}(a, c \cup d)$
$\emptyset \cap c$	$\equiv$	$\emptyset$
$\text{Ag}(a, c) \cap d$	$\equiv$	**if** $a \in d$ **then** $\text{Ag}(a, c \cap d)$ **else** $c \cap d$ **fi**
$\text{dameUno}(c) \in c$	$\equiv$	true
$\text{sinUno}(c)$	$\equiv$	$c - \{\text{dameUno}(c)\}$
$c \subseteq d$	$\equiv$	$c \cap d = c$
$\emptyset - c$	$\equiv$	$\emptyset$
$\text{Ag}(a, c) - d$	$\equiv$	**if** $a \in d$ **then** $c - d$ **else** $\text{Ag}(a, c - d)$ **fi**

Fin TAD

6. TAD Multiconjunto(α)

TAD Multiconjunto(α)

igualdad observacional

$(\forall c, c' : \text{multiconj}(\alpha))\ \left(c =_{\text{obs}} c' \iff ((\forall a : \alpha)(\#(a, c) =_{\text{obs}} \#(a, c')))\right)$

parámetros formales

géneros α

géneros multiconj(α)

exporta multiconj(α), generadores, observadores, $\in$, $\emptyset?$, $\#$, $\cup$, $\cap$, $\in$, $\bullet - \{ \bullet \}$, dameUno, sinUno

usa Bool, Nat

observadores básicos

$\#$	: $\alpha \times \text{multiconj}(\alpha)$	$\longrightarrow$	nat

generadores

$\emptyset$	:	$\longrightarrow$	multiconj(α)
Ag	: $\alpha \times \text{multiconj}(\alpha)$	$\longrightarrow$	multiconj(α)

otras operaciones

$\bullet \in \bullet$	: $\alpha \times \text{multiconj}(\alpha)$	$\longrightarrow$	bool
$\emptyset?$	: multiconj(α)	$\longrightarrow$	bool
$\#$	: multiconj(α)	$\longrightarrow$	nat
$\bullet - \{\bullet\}$	: $\text{multiconj}(\alpha) \times \alpha$	$\longrightarrow$	multiconj(α)
$\bullet \cup \bullet$	: $\text{multiconj}(\alpha) \times \text{multiconj}(\alpha)$	$\longrightarrow$	multiconj(α)

$\bullet \cap \bullet$	: multiconj(α) $\times$ multiconj(α)	$\longrightarrow$	multiconj(α)	
dameUno	: multiconj(α) c	$\longrightarrow$	α	$\{\neg\emptyset?(c)\}$
sinUno	: multiconj(α) c	$\longrightarrow$	multiconj(α)	$\{\neg\emptyset?(c)\}$

axiomas $\forall\, c, d$: multiconj(α), $\forall\, a, b$: α

$\#(a,\, \emptyset) \equiv 0$

$\#(a,\, \mathrm{Ag}(b,\, c)) \equiv$ **if** $a = b$ **then** 1 **else** 0 **fi** $+ \#(a,\, c)$

$a \in c \equiv \#(a,\, c) > 0$

$\emptyset?(\emptyset) \equiv$ true

$\emptyset?(\mathrm{Ag}(a,\, c)) \equiv$ false

$\#(\emptyset) \equiv 0$

$\#(\mathrm{Ag}(a,\, c)) \equiv 1 + \#(c)$

$\emptyset - \{a\} \equiv \emptyset$

$\mathrm{Ag}(a,\, c) - \{b\} \equiv$ **if** $a = b$ **then** c **else** $\mathrm{Ag}(a,\, c - \{b\})$ **fi**

$\emptyset \cup c \equiv c$

$\mathrm{Ag}(a,\, c) \cup d \equiv \mathrm{Ag}(a,\, c \cup d)$

$\emptyset \cap c \equiv \emptyset$

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

$\mathrm{dameUno}(c) \in c \equiv$ true

$\mathrm{sinUno}(c) \equiv c - \{\mathrm{dameUno}(c)\}$

Fin TAD

7. TAD Arreglo dimensionable(α)

TAD Arreglo dimensionable(α)

igualdad observacional

$$(\forall a, a' : \mathrm{ad}(\alpha)) \left(a =_{\mathrm{obs}} a' \iff \begin{pmatrix} \mathrm{tam}(a) =_{\mathrm{obs}} \mathrm{tam}(a') \wedge \\ (\forall n : \mathrm{nat})(\mathrm{definido?}(a, n) =_{\mathrm{obs}} \mathrm{definido?}(a', n) \wedge \\ (\mathrm{definido?}(a, n) \Rightarrow a[n] =_{\mathrm{obs}} a'[n])) \end{pmatrix} \right)$$

parámetros formales
géneros α

géneros ad(α)

exporta ad(α), generadores, observadores

usa Bool, Nat

observadores básicos

tam	: ad(α)	$\longrightarrow$	nat	
definido?	: ad(α) $\times$ nat	$\longrightarrow$	bool	
$\bullet\,[\,\bullet\,]$	: ad(α) a $\times$ nat n	$\longrightarrow$	α	$\{\mathrm{definido?}(a,\, n)\}$

generadores

crearArreglo	: nat	$\longrightarrow$	ad(α)

$\bullet\,[\,\bullet\,] \leftarrow \bullet$: ad(α) a × nat n × α ⟶ ad(α) — $\{n < \text{tam}(a)\}$

axiomas ∀ a: ad(α), ∀ e: α, ∀ n, m: nat

tam(crearArreglo(n)) ≡ n

tam(a [n] ← e) ≡ tam(a)

definido(crearArreglo(n), m)) ≡ false

definido(a [n] ← e, m) ≡ $n = m$ ∨ definido?(a, m)

(a [n] ← e) [m] ≡ **if** $n = m$ **then** e **else** a [m] **fi**

Fin TAD

8. TAD Pila(α)

TAD Pila(α)

igualdad observacional

$$(\forall p, p' : \text{pila}(\alpha)) \left(p =_{\text{obs}} p' \iff \begin{pmatrix} \text{vacía?}(p) =_{\text{obs}} \text{vacía?}(p')) \wedge_{\text{L}} (\neg\, \text{vacía?}(p) \Rightarrow_{\text{L}} \\ (\text{tope}(p) =_{\text{obs}} \text{tope}(p') \wedge \text{desapilar}(p) =_{\text{obs}} \text{desapilar}(p')) \end{pmatrix} \right)$$

parámetros formales

géneros α

géneros pila(α)

exporta pila(α), generadores, observadores, tamaño

usa Bool, Nat

observadores básicos

vacía? : pila(α) ⟶ bool

tope : pila(α) p ⟶ α — $\{\neg$ vacía?(p)$\}$

desapilar : pila(α) p ⟶ pila(α) — $\{\neg$ vacía?(p)$\}$

generadores

vacía : ⟶ pila(α)

apilar : α × pila(α) ⟶ pila(α)

otras operaciones

tamaño : pila(α) ⟶ nat

axiomas ∀ p: pila(α), ∀ e: α

vacía?(vacía) ≡ true

vacía?(apilar(e,p)) ≡ false

tope(apilar(e,p)) ≡ e

desapilar(apilar(e,p)) ≡ p

tamaño(p) ≡ **if** vacía?(p) **then** 0 **else** 1 + tamaño(desapilar(p)) **fi**

Fin TAD

9. TAD Cola(α)

TAD Cola(α)

igualdad observacional

$$(\forall c, c' : \text{cola}(\alpha)) \left(c =_{\text{obs}} c' \iff \begin{pmatrix} \text{vacía?}(c) =_{\text{obs}} \text{vacía?}(c') \land_{\text{L}} \\ (\neg\text{vacía?}(c) \Rightarrow_{\text{L}} (\text{próximo}(c) =_{\text{obs}} \text{próximo}(c') \land \\ \text{desencolar}(c) =_{\text{obs}} \text{desencolar}(c')) \end{pmatrix} \right)$$

parámetros formales
géneros α

géneros cola(α)

exporta cola(α), generadores, observadores, tamaño

usa Bool, Nat

observadores básicos

vacía?	: cola(α)	$\longrightarrow$	bool	
próximo	: cola(α) c	$\longrightarrow$	α	$\{\neg$ vacía?(c)$\}$
desencolar	: cola(α) c	$\longrightarrow$	cola(α)	$\{\neg$ vacía?(c)$\}$

generadores

vacía	:	$\longrightarrow$	cola(α)
encolar	: $\alpha \times$ cola(α)	$\longrightarrow$	cola(α)

otras operaciones

tamaño	: cola(α)	$\longrightarrow$	nat

axiomas $\forall$ c: cola(α), $\forall$ e: α

vacía?(vacía)	$\equiv$	true
vacía?(encolar(e,c))	$\equiv$	false
próximo(encolar(e,c))	$\equiv$	**if** vacia?(c) **then** e **else** próximo(c) **fi**
desencolar(encolar(e,c))	$\equiv$	**if** vacía?(c) **then** vacía **else** encolar(e, desencolar(c)) **fi**
tamaño(c)	$\equiv$	**if** vacía?(c) **then** 0 **else** 1 + tamaño(desencolar(c)) **fi**

Fin TAD

10. TAD Árbol binario(α)

TAD Árbol binario(α)

igualdad observacional

$$(\forall a, a' : \text{ab}(\alpha)) \left(a =_{\text{obs}} a' \iff \begin{pmatrix} \text{nil?}(a) =_{\text{obs}} \text{nil?}(a') \land_{\text{L}} (\neg\ \text{nil?}(a) \Rightarrow_{\text{L}} (\text{raiz}(a) =_{\text{obs}} \text{raiz}(a') \\ \land\ \text{izq}(a) =_{\text{obs}} \text{izq}(a') \land \text{der}(a) =_{\text{obs}} \text{der}(a')) \end{pmatrix} \right)$$

parámetros formales

géneros α

géneros ab(α)

exporta ab(α), generadores, observadores, tamaño, altura, tamaño, inorder, preorder, postorder

usa Bool, Nat, Secuencia(α)

observadores básicos

nil? : ab(α) $\longrightarrow$ bool

raiz : ab(α) a $\longrightarrow$ α $\{\neg$ nil?(a)$\}$

izq : ab(α) a $\longrightarrow$ ab(α) $\{\neg$ nil?(a)$\}$

der : ab(α) a $\longrightarrow$ ab(α) $\{\neg$ nil?(a)$\}$

generadores

nil : $\longrightarrow$ ab(α)

bin : ab(α) $\times$ α $\times$ ab(α) $\longrightarrow$ ab(α)

otras operaciones

altura : ab(α) $\longrightarrow$ nat

tamaño : ab(α) $\longrightarrow$ nat

inorder : ab(α) $\longrightarrow$ secu(α)

preorder : ab(α) $\longrightarrow$ secu(α)

postorder : ab(α) $\longrightarrow$ secu(α)

axiomas $\forall$ a, b: ab(α), $\forall$ e: α

nil?(nil) $\equiv$ true

nil?(bin(a,e,b)) $\equiv$ false

raiz(bin(a,e,b)) $\equiv$ e

izq(bin(a,e,b)) $\equiv$ a

der(bin(a,e,b)) $\equiv$ b

altura(a) $\equiv$ **if** nil?(a) **then** 0 **else** 1 + máx(altura(izq(a)), altura(der(a))) **fi**

tamaño(a) $\equiv$ **if** nil?(a) **then** 0 **else** 1 + tamaño(izq(a)) + tamaño(der(a)) **fi**

inorder(a) $\equiv$ **if** nil?(a) **then** <> **else** inorder(izq(a)) & (raiz(a) $\bullet$ inorder(der(a))) **fi**

preorder(a) $\equiv$ **if** nil?(a) **then** <> **else** (raiz(a) $\bullet$ preorder(izq(a))) & preorder(der(a)) **fi**

postorder(a) $\equiv$ **if** nil?(a) **then** <> **else** postorder(izq(a)) & (postorder(der(a)) $\circ$ raiz(a)) **fi**

Fin TAD

11. TAD Diccionario(clave, significado)

TAD Diccionario(clave, significado)

igualdad observacional

$$(\forall d, d' : \mathrm{dicc}(\kappa, \sigma)) \left(d =_{\mathrm{obs}} d' \iff \begin{pmatrix} (\forall c : \kappa)(\mathrm{def?}(c, d) =_{\mathrm{obs}} \mathrm{def?}(c, d') \wedge_{\mathrm{L}} \\ (\mathrm{def?}(c, d) \Rightarrow_{\mathrm{L}} \mathrm{obtener}(c, d) =_{\mathrm{obs}} \mathrm{obtener}(c, d'))) \end{pmatrix} \right)$$

parámetros formales

géneros clave, significado

géneros dicc(clave, significado)

exporta dicc(clave, significado), generadores, observadores, borrar, claves

usa Bool, Nat, Conjunto(clave)

observadores básicos

def? : clave × dicc(clave, significado) ⟶ bool

obtener : clave c × dicc(clave, significado) d ⟶ significado {def?(c, d)}

generadores

vacío : ⟶ dicc(clave, significado)

definir : clave × significado × dicc(clave, significado) ⟶ dicc(clave, significado)

otras operaciones

borrar : clave c × dicc(clave, significado) d ⟶ dicc(clave, significado) {def?(c,d)}

claves : dicc(clave, significado) ⟶ conj(clave)

axiomas ∀ d: dicc(clave, significado), ∀ c, k: clave, ∀ s: significado

def?(c,vacío) ≡ false

def?(c, definir(k, s, d) ≡ $c = k$ ∨ def?(c, d)

obtener(c, definir(k, s, d)) ≡ **if** $c = k$ **then** s **else** obtener(c, d) **fi**

borrar(c, definir(k, s, d)) ≡ **if** $c = k$ **then**
  if def?(c,d) **then** borrar(c,d) **else** d **fi**
else
  definir(k, s, borrar(c, d))
fi

claves(vacío) ≡ $\emptyset$

claves(definir(c,s,d)) ≡ Ag(c, claves(d))

Fin TAD

12. TAD Cola de prioridad(α)

TAD Cola de prioridad(α)

igualdad observacional

$$(\forall c, c' : \text{colaPrior}(\alpha)) \left(c =_{\text{obs}} c' \iff \begin{pmatrix} \text{vacía?}(c) =_{\text{obs}} \text{vacía?}(c') \wedge_{\text{L}} \\ (\neg\text{vacía?}(c) \Rightarrow_{\text{L}} (\text{próximo}(c) =_{\text{obs}} \text{próximo}(c') \wedge \\ \text{desencolar}(c) =_{\text{obs}} \text{desencolar}(c')) \end{pmatrix} \right)$$

parámetros formales

géneros α

operaciones $\bullet < \bullet : \alpha \times \alpha \longrightarrow$ bool Relación de orden total estricto[1]

géneros colaPrior(α)

exporta colaPrior(α), generadores, observadores

usa Bool

observadores básicos

vacía? : colaPrior(α) $\longrightarrow$ bool

próximo : colaPrior(α) c $\longrightarrow$ α $\{\neg$ vacía?(c)$\}$

desencolar : colaPrior(α) c $\longrightarrow$ colaPrior(α) $\{\neg$ vacía?(c)$\}$

generadores

vacía : $\longrightarrow$ colaPrior(α)

encolar : $\alpha \times$ colaPrior(α) $\longrightarrow$ colaPrior(α)

axiomas $\forall$ c: colaPrior(α), $\forall$ e: α

vacía?(vacía) $\equiv$ true

vacía?(encolar(e, c)) $\equiv$ false

próximo(encolar(e, c)) $\equiv$ **if** vacía?(c) $\vee_L$ proximo(c) $< e$ **then** e **else** próximo(c) **fi**

desencolar(encolar(e, c)) $\equiv$ **if** vacía?(c) $\vee_L$ proximo(c) $< e$ **then** c **else** encolar(e, desencolar(c)) **fi**

Fin TAD

[1] Una relación es un orden total estricto cuando se cumple:

Antirreflexividad: $\neg\, a < a$ para todo $a : \alpha$

Antisimetría: $(a < b \Rightarrow \neg\, b < a)$ para todo $a, b : \alpha$, $a \neq b$

Transitividad: $((a < b \wedge b < c) \Rightarrow a < c)$ para todo $a, b, c : \alpha$

Totalidad: $(a < b \vee b < a)$ para todo $a, b : \alpha$