Tipos abstractos de datos bï¿½sicos

Algoritmos y Estructuras de Datos II, DC, UBA.

Índice

1. TAD Bool

TAD Bool

géneros bool

exporta bool, generadores, $\neg$, $\vee$, $\wedge$, $\Rightarrow$, **if** $\bullet$ **then** $\bullet$ **else** $\bullet$ **fi**

igualdad observacional

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

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

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

$x \Rightarrow y$ $\equiv$ $\neg x \vee y$

if true **then** a **else** b **fi** $\equiv$
a

if false **then** a **else** b **fi** $\equiv$
b

Fin TAD

2. TAD Nat

TAD Nat

géneros nat

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

usa Bool

igualdad observacional

$$(\forall n, m : \text{nat}) \left(n =_{\text{obs}} m \Longleftrightarrow \begin{pmatrix} (n = 0? =_{\text{obs}} m = 0?) \wedge \\ (\neg(n = 0?) \Rightarrow (\text{pred}(n) =_{\text{obs}} \text{pred}(m))) \end{pmatrix} \right)$$

observadores básicos

$\bullet = 0?$: nat $\longrightarrow$ bool pred : nat n $\longrightarrow$ nat $\{\neg(n = 0?)\}$

generadores

0 : $\longrightarrow$ nat

suc : nat $\longrightarrow$ nat

otras operaciones

$\bullet + \bullet$: nat $\times$ nat $\longrightarrow$ nat

$\bullet - \bullet$: nat n $\times$ nat m $\longrightarrow$ nat $\{m \leq_{\text{nat}} n\}$

$\bullet \times \bullet$: nat $\times$ nat $\longrightarrow$ nat

$\bullet < \bullet$: nat $\times$ nat $\longrightarrow$ bool

$\bullet \leq \bullet$: nat $\times$ nat $\longrightarrow$ bool

mín : nat $\times$ nat $\longrightarrow$ nat

máx : nat $\times$ nat $\longrightarrow$ nat

axiomas $\forall\, n, m$: nat

0 = 0? $\equiv$ true

suc(n)= 0? $\equiv$ false

pred(suc(n)) $\equiv$ n

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

$n - m$ $\equiv$ **if** $m = 0?$ **then** n **else** pred(n) $-$ pred(m) **fi**

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

$n < m$ $\equiv$ $\neg(m = 0?) \wedge (n = 0? \vee$ pred(n) $<$ pred(m))

$n \leq m$ $\equiv$ $n < m \vee n = m$

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

máx(n, m) $\equiv$ **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

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

parámetros formales

gï¿½neros α_1, ..., α_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

$$(\forall s, s' : \text{secu}(\alpha)) \left(s =_{\text{obs}} s' \iff \begin{pmatrix} \text{vacï¿½a?}(s) =_{\text{obs}} \text{vacï¿½a?}(s') \wedge \\ (\neg\ \text{vacï¿½a?}(s) \Rightarrow (\text{prim}(s) =_{\text{obs}} \text{prim}(s') \wedge \text{fin}(s) =_{\text{obs}} \\ \text{fin}(s')) \end{pmatrix} \right)$$

parámetros formales

gï¿½neros α

géneros secu(α)

exporta secu(α), generadores, observadores, &, ∘, ult, com, long, estï¿½?

usa Bool, Nat

observadores básicos

vacï¿½a? : secu(α) $\longrightarrow$ bool

prim : secu(α) s $\longrightarrow$ α $\{\neg$ vacï¿½a?$(s)\}$

fin : secu(α) s $\longrightarrow$ secu(α) $\{\neg$ vacï¿½a?$(s)\}$

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 $\{\neg$ vacï¿½a?$(s)\}$

com : secu(α) s $\longrightarrow$ secu(α) $\{\neg$ vacï¿½a?$(s)\}$

long : secu(α) $\longrightarrow$ nat

estï¿½? : $\alpha \times$ secu(α) $\longrightarrow$ bool

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

vacï¿½a?(<>) $\equiv$
true

vacï¿½a?(e $\bullet$ s) $\equiv$
false

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

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

$s \circ e$	$\equiv$	**if** vacï¿½a?(s) **then** e • <> **else** prim(s) • (fin(s) $\circ$ e) **fi**
s & t	$\equiv$	**if** vacï¿½a?(s) **then** t **else** prim(s) • (fin(s) & t) **fi**
ult(s)	$\equiv$	**if** vacï¿½a?(fin(s)) **then** prim(s) **else** ult(fin(s)) **fi**
com(s)	$\equiv$	**if** vacï¿½a?(fin(s)) **then** <> **else** prim(s) • com(fin(s)) **fi**
long(s)	$\equiv$	**if** vacï¿½a?(s) **then** 0 **else** 1 + long(fin(s)) **fi**
estï¿½?(e, s)	$\equiv$	$\neg$ vacï¿½a?(s) $\wedge$ (e = prim(s) $\vee$ estï¿½?(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

gï¿½neros α

géneros conj(α)

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

usa Bool, Nat

observadores básicos

• $\in$ •	: $\alpha \times$ conj(α)	$\longrightarrow$	bool

generadores

$\emptyset$	:	$\longrightarrow$	conj(α)
Ag	: $\alpha \times$ conj(α)	$\longrightarrow$	conj(α)

otras operaciones

$\emptyset$?	: conj(α)	$\longrightarrow$	bool	
#	: conj(α)	$\longrightarrow$	nat	
• − {•}	: conj(α) $\times$ α	$\longrightarrow$	conj(α)	
• $\cup$ •	: conj(α) $\times$ conj(α)	$\longrightarrow$	conj(α)	
• $\cap$ •	: conj(α) $\times$ conj(α)	$\longrightarrow$	conj(α)	
dameUno	: conj(α) c	$\longrightarrow$	α	$\{\neg\emptyset?(c)\}$
sinUno	: conj(α) c	$\longrightarrow$	conj(α)	$\{\neg\emptyset?(c)\}$
• $\subseteq$ •	: conj(α) $\times$ conj(α)	$\longrightarrow$	bool	
• − •	: conj(α) $\times$ conj(α)	$\longrightarrow$	conj(α)	

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

$a \in \emptyset$	$\equiv$	false
$a \in$ Ag(b, c)	$\equiv$	$(a = b) \vee (a \in c)$
$\emptyset$?($\emptyset$)	$\equiv$	true
$\emptyset$?(Ag(b, c))	$\equiv$	false
#($\emptyset$)	$\equiv$	0

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

Fin TAD

6. TAD Multiconjunto(α)

TAD Multiconjunto(α)

igualdad observacional

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

parámetros formales

gï¿½neros α

géneros multiconj(α)

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

usa Bool, Nat

observadores básicos

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

generadores

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

otras operaciones

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

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

parámetros formales

gï¿½neros 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	{definido?(a, n)}

generadores

crearArreglo	:	nat	$\longrightarrow$ ad(α)	
$\bullet\,[\,\bullet\,] \leftarrow \bullet$	:	ad(α) a $\times$ nat n $\times$ α	$\longrightarrow$ ad(α)	$\{n <_{\mathrm{nat}} \mathrm{tam}(a)\}$

axiomas $\forall$ a: ad(α), $\forall$ e: α, $\forall$ n, m: nat

$\mathrm{tam}(\mathrm{crearArreglo}(n)) \equiv n$

$\mathrm{tam}(a\ [\ n\ ] \leftarrow e) \equiv \mathrm{tam}(a)$

$\mathrm{definido}(\mathrm{crearArreglo}(n),\ m)) \equiv$ false

$\mathrm{definido}(a\ [\ n\ ] \leftarrow e,\ m) \equiv n = m \vee \mathrm{definido?}(a,\ m)$

$(a\,[\,n\,] \leftarrow e)\,[\,m\,] \quad \equiv$ **if** $n = m$ **then** e **else** $a\,[\,m\,]$ **fi**

Fin TAD

8. TAD Pila(α)

TAD Pila(α)

igualdad observacional

$$(\forall p, p' : \text{pila}(\alpha)) \left(p =_{\text{obs}} p' \iff \begin{pmatrix} \text{vacï¿½a?}(p) =_{\text{obs}} \text{vacï¿½a?}(p')) \land (\neg\ \text{vacï¿½a?}(p) \Rightarrow \\ (\text{tope}(p) =_{\text{obs}} \text{tope}(p') \land \text{desapilar}(p) =_{\text{obs}} \text{desapilar}(p')) \end{pmatrix} \right)$$

parámetros formales

gï¿½neros α

géneros pila(α)

exporta pila(α), generadores, observadores, tamaï¿½o

usa Bool, Nat

observadores básicos

vacï¿½a? : pila(α) $\longrightarrow$ bool

tope : pila(α) p $\longrightarrow$ α $\{\neg$ vacï¿½a?(p)$\}$

desapilar : pila(α) p $\longrightarrow$ pila(α) $\{\neg$ vacï¿½a?(p)$\}$

generadores

vacï¿½a : $\longrightarrow$ pila(α)

apilar : $\alpha \times$ pila(α) $\longrightarrow$ pila(α)

otras operaciones

tamaï¿½o : pila(α) $\longrightarrow$ nat

axiomas $\forall$ p: pila(α), $\forall$ e: α

vacï¿½a?(vacï¿½a) $\equiv$ true

vacï¿½a?(apilar(e,p)) $\equiv$ false

tope(apilar(e,p)) $\equiv$ e

desapilar(apilar(e,p)) $\equiv$ p

tamaï¿½o(p) $\equiv$ **if** vacï¿½a?(p) **then** 0 **else** 1 + tamaï¿½o(desapilar(p)) **fi**

Fin TAD

9. TAD Cola(α)

TAD Cola(α)

igualdad observacional

$$(\forall c, c' : \text{cola}(\alpha)) \left(c =_{\text{obs}} c' \iff \begin{pmatrix} \text{vacï¿½a?}(c) =_{\text{obs}} \text{vacï¿½a?}(c') \land \\ (\neg\text{vacï¿½a?}(c) \Rightarrow (\text{prï¿½ximo}(c) =_{\text{obs}} \text{prï¿½ximo}(c') \land \\ \text{desencolar}(c) =_{\text{obs}} \text{desencolar}(c')) \end{pmatrix} \right)$$

parámetros formales

gï¿½neros α

géneros cola(α)

exporta cola(α), generadores, observadores, tamaï¿½o

usa Bool, Nat

observadores básicos

vacï¿½a? : cola(α) $\longrightarrow$ bool

prï¿½ximo : cola(α) c $\longrightarrow$ α {¬ vacï¿½a?(c)}

desencolar : cola(α) c $\longrightarrow$ cola(α) {¬ vacï¿½a?(c)}

generadores

vacï¿½a : $\longrightarrow$ cola(α)

encolar : α × cola(α) $\longrightarrow$ cola(α)

otras operaciones

tamaï¿½o : cola(α) $\longrightarrow$ nat

axiomas ∀ c: cola(α), ∀ e: α

vacï¿½a?(vacï¿½a) ≡ true

vacï¿½a?(encolar(e,c)) ≡ false

prï¿½ximo(encolar(e,c)) ≡ **if** vacia?(c) **then** e **else** prï¿½ximo(c) **fi**

desencolar(encolar(e,c)) ≡ **if** vacï¿½a?(c) **then** vacï¿½a **else** encolar(e, desencolar(c)) **fi**

tamaï¿½o(c) ≡ **if** vacï¿½a?(c) **then** 0 **else** 1 + tamaï¿½o(desencolar(c)) **fi**

Fin TAD

10. TAD ï¿½rbol binario(α)

TAD ï¿½rbol binario(α)

igualdad observacional

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

parámetros formales

gï¿½neros α

géneros ab(α)

exporta ab(α), generadores, observadores, tamaï¿½o, altura, tamaï¿½o, inorder, preorder, postorder

usa Bool, Nat, Secuencia(α)

observadores básicos

nil? : ab(α) $\longrightarrow$ bool

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

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

der : ab(α) a $\longrightarrow$ ab(α) {¬ 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

tamaï¿½o : 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**

tamaï¿½o(a) $\equiv$ **if** nil?(a) **then** 0 **else** 1 + tamaï¿½o(izq(a)) + tamaï¿½o(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

$$(\forall d, d' : \text{dicc}(\kappa,\sigma)) \left(d =_{\text{obs}} d' \iff \begin{pmatrix}(\forall c : \kappa)(\text{def?}(c,d) =_{\text{obs}} \text{def?}(c,d') \land \\ (\text{def?}(c,d) \ \Rightarrow\ \text{obtener}(c,d) =_{\text{obs}} \text{obtener}(c,d')))\end{pmatrix}\right)$$

parámetros formales

gï¿½neros clave, significado

géneros dicc(clave, significado)

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

usa Bool, Nat, Conjunto(clave)

observadores básicos

def? : clave $\times$ dicc(clave, significado) $\longrightarrow$ bool

obtener : clave c $\times$ dicc(clave, significado) d $\longrightarrow$ significado {def?(c, d)}

generadores

vacï¿½o : $\longrightarrow$ dicc(clave, significado)

definir : clave $\times$ significado $\times$ dicc(clave, significado) $\longrightarrow$ dicc(clave, significado)

otras operaciones

borrar	: clave c × dicc(clave, significado) d	$\longrightarrow$	dicc(clave, significado)	{def?(c,d)}
claves	: dicc(clave, significado)	$\longrightarrow$	conj(clave)	

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

def?(c,vacï¿½o)	≡	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(vacï¿½o)	≡	∅
claves(definir(c,s,d)	≡	Ag(c, claves(d))

Fin TAD

12. TAD Cola de prioridad(α)

TAD Cola de prioridad(α)

igualdad observacional

$$(\forall c, c' : \text{colaPrior}(\alpha)) \left(c =_{\text{obs}} c' \iff \left(\begin{array}{l} \text{vacï¿½a?}(c) =_{\text{obs}} \text{vacï¿½a?}(c') \wedge \\ (\neg \text{vacï¿½a?}(c) \Rightarrow (\text{prï¿½ximo}(c) =_{\text{obs}} \text{prï¿½ximo}(c') \\ \wedge \\ \qquad \text{desencolar}(c) =_{\text{obs}} \text{desencolar}(c')) \end{array} \right) \right)$$

parámetros formales

gï¿½neros α

operaciones $\bullet < \bullet$: $\alpha \times \alpha \longrightarrow$ bool — Relaciï¿½n de orden total estricto[1]

géneros colaPrior(α)

exporta colaPrior(α), generadores, observadores

usa Bool

observadores básicos

vacï¿½a?	: colaPrior(α)	$\longrightarrow$	bool	
prï¿½ximo	: colaPrior(α) c	$\longrightarrow$	α	{(¬ vacï¿½a?(c))}
desencolar	: colaPrior(α) c	$\longrightarrow$	colaPrior(α)	{(¬ vacï¿½a?(c))}

generadores

vacï¿½a	:	$\longrightarrow$	colaPrior(α)
encolar	: α × colaPrior(α)	$\longrightarrow$	colaPrior(α)

axiomas ∀ c: colaPrior(α), ∀ e: α

[1] Una relaciï¿½n es un orden total estricto cuando se cumple:

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

Antisimetrï¿½a: $(a < b \Rightarrow \neg\, b < a)$ para todo $a, b : \alpha$

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

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

vacï¿½a?(vacï¿½a) $\equiv$ true

vacï¿½a?(encolar(e, c)) $\equiv$ false

prï¿½ximo(encolar(e, c)) $\equiv$ **if** vacï¿½a?(c) $\vee$ proximo(c) $< e$ **then** e **else** prï¿½ximo(c) **fi**

desencolar(encolar(e, c)) $\equiv$ **if** vacï¿½a?(c) $\vee$ proximo(c) $< e$ **then** c **else** encolar(e, desencolar(c)) **fi**

Fin TAD