Tipos abstractos de datos básicos

Algoritmos y Estructuras de Datos II, DC, UBA.

Índice

1. TAD Bool

TAD Bool

géneros bool

exporta bool, generadores, observadores, $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\vee_L$, $\wedge_L$, $\Rightarrow_L$, β

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

if $\bullet$ **then** $\bullet$ **else** $\bullet$ **fi**	:	bool $\times\ \alpha \times \alpha$	$\longrightarrow \alpha$
$\neg\ \bullet$	:	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
$\bullet \vee_L \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool
$\bullet \wedge_L \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool
$\bullet \Rightarrow_L \bullet$	:	bool $\times$ bool	$\longrightarrow$ bool
$\beta\ (\bullet)$	:	bool	$\longrightarrow$ nat

axiomas $\forall\ x, y$: bool, $\forall\ a, b$: α

if true **then** a **else** b **fi**	$\equiv a$
if false **then** a **else** b **fi**	$\equiv b$
$\neg\ x$	$\equiv$ **if** x **then** false **else** true **fi**
$x \vee y$	$\equiv$ **if** x **then** (**if** y **then** true **else** true **fi**) **else** y **fi**
$x \wedge y$	$\equiv$ **if** x **then** y **else** (**if** y **then** false **else** false **fi**) **fi**
$x \Rightarrow y$	$\equiv \neg x \vee y$
$x \vee_L y$	$\equiv$ **if** x **then** true **else** y **fi**
$x \wedge_L y$	$\equiv$ **if** x **then** y **else** false **fi**
$x \Rightarrow_L y$	$\equiv \neg x \vee_L y$
$\beta\ (x)$	$\equiv$ **if** x **then** 1 **else** 0 **fi**

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 $\longrightarrow$ 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 $\{m \leq n\}$

$\bullet \times \bullet$: 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**

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

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

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

$$(\forall t, t' : \text{tupla}(\alpha_1, \ldots, \alpha_n)) \;\; (t =_{\text{obs}} t' \iff (\pi_1(t) =_{\text{obs}} \pi_1(t') \wedge \ldots \wedge \pi_n(t) =_{\text{obs}} \pi_n(t')))$$

parámetros formales

géneros 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: α_1 ...$\forall$ a_n: α_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	$\longrightarrow\ \alpha$	$\{\neg$ 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 $\forall\ s, t$: secu(α), $\forall\ e$: α

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

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

generadores

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

otras operaciones

$\emptyset$?	:	conj(α)	$\longrightarrow$ bool

vacio?	:	conj(α)	$\longrightarrow$ bool	
$\{\bullet, \ldots, \bullet\}$	:	$\alpha \times \ldots \times \alpha$	$\longrightarrow$ conj(α)	
#	:	conj(α)	$\longrightarrow$ nat	
$\bullet - \{\bullet\}$	:	conj(α) $\times$ α	$\longrightarrow$ conj(α)	
$\bullet \cup \bullet$	:	conj(α) $\times$ conj(α)	$\longrightarrow$ conj(α)	
$\bullet \cap \bullet$	:	conj(α) $\times$ conj(α)	$\longrightarrow$ conj(α)	
dameUno	:	conj(α) c	$\longrightarrow$ α	$\{\neg\emptyset?(c)\}$
sinUno	:	conj(α) c	$\longrightarrow$ conj(α)	$\{\neg\emptyset?(c)\}$
$\bullet \subseteq \bullet$	:	conj(α) $\times$ conj(α)	$\longrightarrow$ bool	
$\bullet - \bullet$	:	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
vacio?($\emptyset$)	$\equiv$	$\emptyset?(\emptyset)$
vacio?(Ag(b, c))	$\equiv$	$\emptyset?$(Ag(b, c))
$\#(\emptyset)$	$\equiv$	0
#(Ag(a, c))	$\equiv$	$1 + \#(c$ - { a })
$\{a_1, \ldots, a_n\}$	$\equiv$	Ag(a_n, ..., Ag(a_1, $\emptyset$))
$c - \{a\}$	$\equiv$	$c -$ Ag(a, $\emptyset$)
$\emptyset \cup c$	$\equiv$	c
Ag(a, c) $\cup\ d$	$\equiv$	Ag(a, $c \cup d$)
$\emptyset \cap c$	$\equiv$	$\emptyset$
Ag(a, c) $\cap\ d$	$\equiv$	**if** $a \in d$ **then** Ag(a, $c \cap d$) **else** $c \cap d$ **fi**
dameUno(c) $\in c$	$\equiv$	true
sinUno(c)	$\equiv$	$c - \{$dameUno(c)$\}$
$c \subseteq d$	$\equiv$	$c \cap d = c$
$\emptyset - c$	$\equiv$	$\emptyset$
Ag(a, c) $-\ d$	$\equiv$	**if** $a \in d$ **then** $c - d$ **else** 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

$\#$: $\alpha \times$ multiconj(α) $\longrightarrow$ nat

generadores

$\emptyset$: $\longrightarrow$ multiconj(α)

Ag : $\alpha \times$ multiconj(α) $\longrightarrow$ multiconj(α)

otras operaciones

$\bullet \in \bullet$: $\alpha \times$ multiconj(α) $\longrightarrow$ bool

$\emptyset$? : multiconj(α) $\longrightarrow$ bool

$\#$: multiconj(α) $\longrightarrow$ nat

$\bullet - \{\bullet\}$: multiconj(α) $\times\ \alpha$ $\longrightarrow$ multiconj(α)

$\bullet \cup \bullet$: multiconj(α) $\times$ multiconj(α) $\longrightarrow$ 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

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

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	{definido?(a, n)}

generadores

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

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

tam(crearArreglo(n))	$\equiv$	n
tam(a [n] $\leftarrow$ e)	$\equiv$	tam(a)
definido(crearArreglo(n), m))	$\equiv$	false
definido(a [n] $\leftarrow$ e, m)	$\equiv$	$n = m$ $\lor$ definido?(a, m)
(a [n] $\leftarrow$ e) [m]	$\equiv$	**if** $n = m$ **then** e **else** a [m] **fi**

Fin TAD

8. TAD Pila(α)

TAD Pila(α)

igualdad observacional

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

parámetros formales

géneros α

géneros pila(α)

exporta pila(α), generadores, observadores, tamaño

usa Bool, Nat

observadores básicos

vacía?	: pila(α)	$\longrightarrow$	bool	
tope	: pila(α) p	$\longrightarrow$	α	$\{\neg$ vacía?(p)$\}$
desapilar	: pila(α) p	$\longrightarrow$	pila(α)	$\{\neg$ vacía?(p)$\}$

generadores

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

otras operaciones

tamaño	: pila(α)	$\longrightarrow$	nat

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

vacía?(vacía)	$\equiv$	true
vacía?(apilar(e,p))	$\equiv$	false
tope(apilar(e,p))	$\equiv$	e
desapilar(apilar(e,p))	$\equiv$	p
tamaño(p)	$\equiv$	**if** vacía?(p) **then** 0 **else** 1 + tamaño(desapilar(p)) **fi**

Fin TAD

9. TAD Cola(α)

TAD Cola(α)

igualdad observacional

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

parámetros formales
géneros α

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

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

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 $\{\neg\, \text{nil?}(a)\}$

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

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

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)

esHoja? : ab(α) $\longrightarrow$ bool

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**
esHoja?(a) $\equiv$ **if** nil?(a) **then** false **else** (nil?(izq(a)) & nil?(der(a))) **fi**

Fin TAD

11. TAD Diccionario(clave, significado)

TAD Diccionario(clave, significado)

igualdad observacional

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

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

generadores

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

otras operaciones

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

axiomas $\forall$ d: dicc(clave, significado), $\forall$ c, k: clave, $\forall$ s: significado

def?(c,vacío) $\equiv$ false
def?(c, definir(k, s, d) $\equiv$ $c = k \vee$ def?(c, d)
obtener(c, definir(k, s, d)) $\equiv$ **if** $c = k$ **then** s **else** obtener(c, d) **fi**
borrar(c, definir(k, s, d)) $\equiv$ **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) $\equiv$ $\emptyset$
claves(definir(c,s,d)) $\equiv$ 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) $\vee_{\text{L}}$ proximo(c) $< e$ **then** e **else** próximo(c) **fi**

desencolar(encolar(e, c)) $\equiv$ **if** vacía?(c) $\vee_{\text{L}}$ 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 $a, b : \alpha$