Tipos abstractos de datos básicos

Cátedra de Algoritmos y Estructuras de Datos II
Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

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
$\text{suc}(n) = 0?$	$\equiv$	false
$\text{pred}(\text{suc}(n))$	$\equiv$	n
$n + m$	$\equiv$	**if** $m = 0?$ **then** n **else** $\text{suc}(n + \text{pred}(m))$ **fi**
$n - m$	$\equiv$	**if** $m = 0?$ **then** n **else** $\text{pred}(n) - \text{pred}(m)$ **fi**
$n \times m$	$\equiv$	**if** $m = 0?$ **then** 0 **else** $n \times \text{pred}(m) + n$ **fi**
$n < m$	$\equiv$	$\neg(m = 0?) \wedge_{\text{L}} (n = 0? \vee_{\text{L}} \text{pred}(n) < \text{pred}(m))$
$n \leq m$	$\equiv$	$n < m \vee n = m$
$\text{mín}(n, m)$	$\equiv$	**if** $m < n$ **then** m **else** n **fi**
$\text{máx}(n, m)$	$\equiv$	**if** $m < n$ **then** n **else** m **fi**

Fin TAD

3. TAD Int

TAD INT

géneros int

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

usa NAT

igualdad observacional

$(\forall n, m : \text{int})\ \ (n =_{\text{obs}} m \iff (\text{negativo?}(n) =_{\text{obs}} \text{negativo}(m) \wedge |n| =_{\text{obs}} |m|))$

observadores básicos

negativo?	:	int	$\longrightarrow$	bool
$\mid\bullet\mid$	:	int	$\longrightarrow$	nat

generadores

$+\bullet$	:	nat	$\longrightarrow$	int	
$-\bullet$	:	nat n	$\longrightarrow$	int	$\{\neg(n = 0?)\}$

otras operaciones

$\bullet + \bullet$	:	int $\times$ int	$\longrightarrow$	int
$\bullet - \bullet$	:	int $\times$ int	$\longrightarrow$	int
$\bullet \times \bullet$	:	int $\times$ int	$\longrightarrow$	int
$\bullet < \bullet$	:	int $\times$ int	$\longrightarrow$	bool
$\bullet \leq \bullet$	:	int $\times$ int	$\longrightarrow$	bool
mín	:	int $\times$ int	$\longrightarrow$	int
máx	:	int $\times$ int	$\longrightarrow$	int

axiomas $\forall\ n, m$: nat, $\forall\ x, y$: int

negativo?($+n$)	$\equiv$	false
negativo?($-n$)	$\equiv$	true
$\mid +n \mid$	$\equiv$	n
$\mid -n \mid$	$\equiv$	n
$+n + +m$	$\equiv$	$+(\text{n} + \text{m})$
$+n + -m$	$\equiv$	**if** $m \leq n$ **then** $+(n - m)$ **else** $-(m - n)$ **fi**
$-n + +m$	$\equiv$	**if** $n \leq m$ **then** $+(m - n)$ **else** $-(n - m)$ **fi**
$-n + -m$	$\equiv$	$-(\text{n} + \text{m})$
$x - +n$	$\equiv$	**if** $n = 0?$ **then** x **else** $x + -n$ **fi**
$x - -n$	$\equiv$	$x + +n$
$+n \times +m$	$\equiv$	$+(n \times m)$
$+n \times -m$	$\equiv$	**if** $n = 0?$ **then** $+0$ **else** $-(n \times m)$ **fi**
$-n \times +m$	$\equiv$	**if** $m = 0?$ **then** $+0$ **else** $-(n \times m)$ **fi**
$-n \times -m$	$\equiv$	$+(n \times m)$
$+n < +m$	$\equiv$	$n < m$
$+n < -m$	$\equiv$	false
$-n < +m$	$\equiv$	true
$-n < -m$	$\equiv$	$m < n$
$x \leq y$	$\equiv$	$x < y \vee x = y$
mín(x, y)	$\equiv$	**if** $x < y$ **then** x **else** y **fi**
máx(x, y)	$\equiv$	**if** $x < y$ **then** y **else** x **fi**

Fin TAD

4. TAD Tupla(α_1, ..., α_n)

TAD TUPLA(α_1, ..., α_n)

igualdad observacional

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

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 $\forall\ a_1$: 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

5. 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 $\forall\ s, t$: 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

6. 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

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

generadores

$\emptyset$	:		$\longrightarrow$	conj(α)
Ag	:	$\alpha \times \text{conj}(\alpha)$	$\longrightarrow$	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	
$\bullet - \{\bullet\}$	:	$\text{conj}(\alpha) \times \alpha$	$\longrightarrow$	conj(α)	
$\bullet \cup \bullet$	:	$\text{conj}(\alpha) \times \text{conj}(\alpha)$	$\longrightarrow$	conj(α)	
$\bullet \cap \bullet$	:	$\text{conj}(\alpha) \times \text{conj}(\alpha)$	$\longrightarrow$	conj(α)	
dameUno	:	conj(α) c	$\longrightarrow$	α	$\{\neg\emptyset?(c)\}$
sinUno	:	conj(α) c	$\longrightarrow$	conj(α)	$\{\neg\emptyset?(c)\}$
$\bullet \subseteq \bullet$	:	$\text{conj}(\alpha) \times \text{conj}(\alpha)$	$\longrightarrow$	bool	
$\bullet - \bullet$	:	$\text{conj}(\alpha) \times \text{conj}(\alpha)$	$\longrightarrow$	conj(α)	

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

$a \in \emptyset$	$\equiv$	false
$a \in \text{Ag}(b,\ c)$	$\equiv$	$(a = b) \lor (a \in c)$
$\emptyset?(\emptyset)$	$\equiv$	true
$\emptyset?(\text{Ag}(b,\ c))$	$\equiv$	false
vacio?($\emptyset$)	$\equiv$	$\emptyset?(\emptyset)$
vacio?(Ag(b, c))	$\equiv$	$\emptyset?(\text{Ag}(b,\ c))$
$\#(\emptyset)$	$\equiv$	0
$\#(\text{Ag}(a,\ c))$	$\equiv$	$1 + \#(c$ - $\{\ a\ \}\ )$
$\{a_1,\ \ldots,\ a_n\}$	$\equiv$	$\text{Ag}(a_n,\ \ldots,\ \text{Ag}(a_1,\ \emptyset))$
$c - \{a\}$	$\equiv$	$c - \text{Ag}(a,\ \emptyset)$
$\emptyset \cup c$	$\equiv$	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
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

7. TAD Multiconjunto(α)

TAD MULTICONJUNTO(α)

igualdad observacional

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

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)	
$\bullet \cap \bullet$	: $\text{multiconj}(\alpha) \times \text{multiconj}(\alpha)$	$\longrightarrow$ 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
$\#(a,\ \text{Ag}(b,\ c))$	$\equiv$	**if** $a = b$ **then** 1 **else** 0 **fi** $+\ \#(a,\ c)$
$a \in c$	$\equiv$	$\#(a,\ c) > 0$
$\emptyset?(\emptyset)$	$\equiv$	true
$\emptyset?(\text{Ag}(a,\ c))$	$\equiv$	false
$\#(\emptyset)$	$\equiv$	0
$\#(\text{Ag}(a,\ c))$	$\equiv$	$1 + \#(c)$
$\emptyset - \{a\}$	$\equiv$	$\emptyset$
$\text{Ag}(a,\ c) - \{b\}$	$\equiv$	**if** $a = b$ **then** c **else** $\text{Ag}(a,\ c - \{b\})$ **fi**
$\emptyset \cup c$	$\equiv$	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** $\text{Ag}(a,\ c \cap (d - \{a\}))$ **else** $c\ \cap\ d$ **fi**
$\text{dameUno}(c) \in c$	$\equiv$	true
$\text{sinUno}(c)$	$\equiv$	$c - \{\text{dameUno}(c)\}$

Fin TAD

8. 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	{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)	$\equiv$	$n = m$ $\vee$ definido?(a, m)
(a [n] $\leftarrow$ e) [m]	$\equiv$	**if** $n = m$ **then** e **else** a [m] **fi**

Fin TAD

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

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

10. TAD Cola(α)

TAD COLA(α)

igualdad observacional

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

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

11. TAD Árbol binario(α)

TAD ÁRBOL BINARIO(α)

igualdad observacional

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

parámetros formales

géneros α

géneros ab(α)

exporta ab(α), generadores, observadores, 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)
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)) $\wedge$ nil?(der(a))) **fi**

Fin TAD

12. 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 × 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) ≡ ∅

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

Fin TAD

13. 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 $a, b : \alpha$