

1

Sección

TAD. Secuencia extendida ( $\alpha$ )  
extiende secuencia

OTRAS OPERACIONES

duplicar:  $sec(\alpha) \rightarrow sec(\alpha)$

menor(IgualLexi:  $sec(\alpha) \times sec(\beta) \rightarrow Bool$

Reverso:  $sec(\alpha) \rightarrow sec(\alpha)$

Copiar:  $sec(\alpha) \rightarrow Bool$

esPrefijo:  $sec(\alpha)A \times sec(\beta)B \rightarrow Bool$

Buscar:  $sec(\alpha)A \times sec(\beta)B \rightarrow Bool$

EstaOrdenada:  $sec(\alpha) \rightarrow Bool$

Insertar Ordenado:  $sec(\alpha)SO \times \alpha \rightarrow sec(\alpha)$

esPermutacion:  $sec(\alpha) \times sec(\beta) \rightarrow Bool$

Combinar:  $sec(\alpha)A \times sec(\beta)B \rightarrow sec(\alpha)$

CANT. A permutacion:  $sec(\alpha) \times e \rightarrow NOT$

Orden total?  
Lexicografico?

$\{ \log(A) \leq \log(B) \}$   
 $\{ \log(A) \leq \log(B) \}$

esta  
Ordenada(SO)

$\{ \text{esta ordenada}(A) \wedge \text{esta ordenada}(B) \}$

AXIOMAS  $\forall s, t \text{ sec}(\alpha), \forall e, d$

duplicar(S)  $\equiv$  IF vacio?(S) then  $\langle \rangle$  else Prim(S) \* Prim(S) \* duplicar(fin(S))  
F

menor(IgualLexi(S, t)  $\equiv$  IF esVacio(S)  $\wedge$   $\neg$  esVacio(t) then True  
else IF Prim(S)  $\leq$  Prim(t) then NOT(fin(S)) fin  
else False Fi Fi

Reverso(S)  $\equiv$  IF vacio?(S) then  $\langle \rangle$  else  
ult(S) \* Reverso(com(S)) Fi

Copiar(S)  $\equiv$  IF vacio?(S) then True else  
(Prez-lex) IF Prim(S) = ult(S) then Copiar(com(fin(S)))  
else False Fi Fi

esPrefijo(S, t)  $\equiv$  IF vacio(S) then True else  
IF Prim(S) = Prim(t) then (esPrefijo(com(S), com(t))  
else False Fi Fi

Buscar(S, t)  $\equiv$  IF esPrefijo(S, t) then 1 else 1 + Buscar(S, fin(t))

estaOrdenada(S)  $\equiv$  IF  $\log(S) = 1$  then true else  
IF Prim(S)  $\leq$  Prim(fin(S)) then estaOrdenada(fin(S))  
else False Fi Fi

Insertar Ordenado(S, e)  $\equiv$  IF  $e \leq$  Prim(S) then  $e * S$   
else Prim(S) \* InsertarOrdenado(fin(S), e)  
IF Prim(S)  $\leq$  Prim(fin(S)) then

comf. Aporcion(S, e)  $\equiv$  IF Prim(S) = e then 1 + conf. Aporcion(fin(S), e)  
else conf. Aporcion(fin(S), e)

esPermutacion(s, t)  $\equiv$  IF  $vacio(s) \wedge vacio(t)$  then

True

else

IF  $long(s) \neq long(t) \wedge combinator(fin(s), fin(t)) = combinator(fin(s), fin(t))$

then esPermutacion(fin(s), fin(t))

else

False

Fi Fi

combiner(s, t)  $\equiv$  esOrdenada(s  $\circ$  t)



②

TAD ARBOL BINARIO extendido ( $\alpha$ )

EXTIENDE ARBOL BINARIO

OTRAS OPERACIONES

eshoja?  $ab(\alpha) \rightarrow \text{Bool}$

#hojas:  $ab(\alpha) \rightarrow \text{NOT}$

Degenerado  $ab(\alpha) \rightarrow \text{Bool}$

Zigzag:  $ab(\alpha) \rightarrow \text{Bool}$

Ultimo Nivel Completo:  $ab(\alpha) \rightarrow \text{NOT}$

Espajo:  $ab(\alpha) \rightarrow ab(\alpha)$

ES Simetrico?  $ab(\alpha) \rightarrow \text{Bool}$

Axiomas:

$\forall a, b: ab(\alpha), \forall a: \alpha$

eshoja?  $\equiv$  IF  $\text{nil}(a)$  then False else  $(\text{nil}(\text{Izg}(a)) \wedge \text{nil}(\text{der}(a)))$  Fi

#hojas  $\equiv$  IF  $\text{nil}(a)$  then 0  
else

IF eshoja?  $a$  then 1

else #hojas( $\text{Izg}(a)$ ) + #hojas( $\text{der}(a)$ )

fi

DAI( $a$ )  $\equiv$  IF eshoja( $a$ ) then True

else

IF  $\text{tamaño}(\text{Izg}(a)) \geq 1 \wedge \text{tamaño}(\text{der}(a)) = 0$   
then DAI( $\text{Izg}(a)$ )

else

False

fi

zigzag( $a$ )  $\equiv$  IF eshoja( $a$ ) then True

else

IF ( $\text{tamaño}(\text{Izg}(a)) \geq 1 \wedge \text{tamaño}(\text{der}(a)) = 0$ )  
then zigzag( $\text{Izg}(a)$ )

else

IF ( $\text{tamaño}(\text{der}(a)) \geq 1 \wedge \text{tamaño}(\text{Izg}(a)) = 0$ )  
then zigzag( $\text{der}(a)$ )

else

False

fi

fi

fi

$UNC(a) \equiv$  IF  $Nil(a) \vee Nil(izq(a)) \vee Nil(der(a))$  then 0

~~else~~  $1 + \max(UNC(izq(a)), UNC(der(a)))$

$1 + \max(UNC(izq(a)), UNC(der(a)))$

else

$UNC(izq(a)) + UNC(der(a))$

fi

fi

$Espajo(a) \equiv$  IF  $Nil(a)$  then a

else

$Bin$

$(Espajo(der(a)), raiz(a), Espajo(izq(a)))$  fi

$esSimetrico(a) \equiv$  IF  $esEspajo(a) \equiv a$  then True else False

④

3)  $\forall c: \text{conj}(\alpha), \forall k: \text{not}$  de TAD conjuntos  $\alpha$

2)  $\text{conj}(\alpha) \rightarrow \text{conj}(\alpha)$   
Partes De:  $\text{conj}(\text{nat}) \rightarrow \text{conj}(\text{conj}(\text{nat}))$   
Partes De (C)  $\equiv$

IF es vacia (c)? then <> else  
Aq(Prim(c), <>)  $\cup$  Partes de (Fin(c))  
 $\hookrightarrow$  union

5) Combinaciones De k:  $\text{conj}(\alpha) \rightarrow \text{conj}(\alpha)$

Combinaciones De k (C, k)  $\equiv$

IF es vacia? (c) then <> else

IF  $\# \text{Partes de } C(c) = k$  then

De una (Partes de C(c))  $\cup$  Combinaciones De k (Sin Uno de  
else  
Combinaciones De k (Sin Uno de C)

fi

fi

TAD (Secuencia Extendida 2( $\alpha$ ))

Extende  
secuencias extendida

Uso

conj

OTRAS OPERACIONES

NTN:  $\text{conj}(\text{seq}(\alpha)) \times \text{seq}(\alpha) \rightarrow \text{conj}(\text{seq}(\alpha))$

domo Sub-Secuencia:  $\text{conj}(\text{seq}(\alpha)) \times \text{seq}(\alpha) \rightarrow \text{conj}(\text{seq}(\alpha))$

es Sub-Secuencia:  $\text{seq}(\alpha) \times \text{seq}(\alpha) \rightarrow \text{Bool}$

sin Prefijos:  $\text{conj}(\text{seq}(\alpha)) \times \text{conj}(\text{seq}(\alpha)) \rightarrow \text{conj}(\text{seq}(\alpha))$

es Prefijo de Otra:  $\text{seq}(\alpha) \times \text{conj}(\text{seq}(\alpha)) \rightarrow \text{Bool}$

else

NTN (Sin uno)

$\rightarrow$

Sigue OTRO modo



Axiomas

$NTN(C, S) \equiv \text{Sin Prefijos}(\text{dome Subsecuencia}(C, S), C)$

$\text{dome Subsecuencia}(C, S) \equiv$

If  $\text{vacuo}(C)$  then  $\langle \rangle$  else

If  $\text{es Subsecuencia}(\text{dome Uno}(C), S)$

Ag( $\text{dome Uno}$ ,  $\text{dome Subsecuencia}(\text{sin Uno}(C), S)$ )

else

$\text{dome Subsecuencia}(\text{sin Uno}(C), S)$

fi

fi

$\text{es Subsecuencia}(S, T) \equiv$

If  $\text{vacuo}(S) \vee \text{vacuo}(T)$  then  $\text{vacuo}(S)$

else

If  $\text{Prim}(S) = \text{Prim}(T)$  then

$\text{es Subsecuencia}(\text{fin}(S), \text{fin}(T))$

else

$\text{es Subsecuencia}(S, \text{fin}(T))$

fi

fi

$\rightarrow \text{cot}(\text{seg})$

$\text{Sin Prefijos}(C, d) \equiv$

If  $\text{vacuo}(C)$  then  $\langle \rangle$  else

If  $\text{es Prefijo}^{\text{Dextra}}(\text{dome Uno}(C), d)$  then

$\text{Sin Prefijos}(\text{sin Uno}(C), d)$

else

Ag( $\text{dome Uno}(C)$ ,  $\text{Sin Prefijos}(\text{sin Uno}(C), d)$ )

fi

fi

$\text{es PrefijoDextra}(S, d) \equiv$

If  $\text{vacuo}$  then

false

else

$(S \neq \text{dome Uno}(C) \wedge \text{es Prefijo}(S, \text{dome Uno}(C))) \vee$   
 $\text{es PrefijoDextra}(S, \text{sin Uno}(C))$

fi

5  
a)  $\forall \alpha \in A, \forall d \in K$

NivelNormal( $\alpha, d$ )  $\rightarrow$  Bool

NivelNormal( $A, K$ )  $\equiv$

IF  $K=0$  then  
 $\neg Nil(A) \wedge \neg Nil(izq(A)) \wedge \neg Nil(der(A)) \wedge \neg Nil(der(A))$

ELSE  
 $NivelNormal(izq(A), K-1) \wedge NivelNormal(der(A), K-1) \wedge$   
 $NivelNormal(der(A), K-1)$

fi

Se recorren la base sobre el parametro  $K$ , descomponiendo u recorriendo  
 1 hasta llegar a 0 para llegar al nivel que queremos chequear  
 c/llamada recursiva dispone 3 nuevas 'nomas' de recursión  
 xq q llamo a NivelNormal pero c/b de la 3 hijos

b) Acolado( $\alpha, d$ )  $\rightarrow$  Bool

Acolado( $A, K$ )  $\equiv$

IF  $K=0$  then Nil( $A$ )

else

$\#nodes(A) \leq K \wedge Acolado(izq(A), K) \wedge Acolado(der(A), K)$   
 $\wedge Acolado(der(A), K)$

fi

$\#nodes(\alpha) \rightarrow$  not

$\#nodes(A) =$  IF  $(\neg Nil(der(A)) \wedge \neg Nil(izq(A)) \wedge \neg Nil(der(A)))$  then 3  
 else

IF  $(\neg Nil(der(A)) \wedge \neg Nil(izq(A))) \vee$

$(\neg Nil(der(A)) \wedge \neg Nil(der(A))) \vee$

$(\neg Nil(izq(A)) \wedge \neg Nil(der(A)))$  then 2

else

IF  $(\neg Nil(der(A)) \vee \neg Nil(izq(A)) \vee \neg Nil(der(A)))$

then 1

else

fi 0

fi

## TAD Rose Tree Extendido (A)

Extiende

Rosetree(A)

Otros Operaciones

hoja? : Rosetree(A)  $\rightarrow$  Bool

Altura : Rosetree(A)  $\rightarrow$  Nat

Altura De Hijos :  $secu(rosetree(A)) \rightarrow$  Nat  $\{ \neg vacio(s) \}$

# hojas : rosetree(A)  $\rightarrow$  Nat

# hojas De Hijos :  $secu(rosetree(A)) \rightarrow$  Nat  $\{ \neg vacio(s) \}$

Podar : rosetree(A)  $\rightarrow$  Rosetree(A)  $\wedge$  Altura(A)  $\geq 1$

Podar Hijos :  $secu(rosetree(A)) \rightarrow secu(rosetree(A))$

Axiomas

hoja?(A)  $\equiv$  vacio?(hijos(A))

Altura(A)  $\equiv 1 +$  IF hoja?(A) then 0 else altura dehijos(hijos(A)) fi

altura dehijos(s)  $\equiv$  IF vacio?(fin(s)) then  
altura(prim(s))

else  $\max(altura(prim(s)), altura dehijos(fin(s)))$  fi

# hojas(A)  $\equiv$  IF hoja?(A) then 1 else # hojas dehijos(hijos(A)) fi

# hojas dehijos(s)  $\equiv$  # hojas(prim(s)) + # hojas De hijos(fin(s))

Podar(A)  $\equiv$  rose(raiz(A), Podarhijos(A))

Podarhijos  $\equiv$  IF vacio?(s) then  $\epsilon$  else

IF hoja?(prim(s)) then podarhijos(fin(s)) else

podar(prim(s)) o podarhijos(fin(s)) fi

fi

FINTAD



⑧

TAD ROBOT

AXIOMAS

$\forall r: \text{ROBOT } c: \text{coordenadas}$

$T: \text{Seq}(\text{coordenadas}) = \text{Trayectorias}$

$\text{PosiciónActual}(r) \equiv$

$\text{Prim}(\text{Trayectoria}(r))$

$\text{Cuentas veces Paso}(c, r) \equiv$

IF  $\text{Loco}(c) ?$  then 0 else

IF  $\text{Prim}(\text{Trayectoria}(r)) = c$  then

$1 + \text{Cuentas veces Paso}(\text{fin}(c), r)$

else

$\text{Cuentas veces Paso}(\text{fin}(c), r)$

fi

fi

$\text{Mas a la Derecha}(r) \equiv$

IF  $\text{Vacio}(\text{Trayectoria}(r))$  then 1 else

IF  $\text{Prim}(T) \geq \text{Mas a la Derecha}(\text{fin}(T))$  then  $\text{Prim}(T)$

else  $\text{Mas a la Derecha}(\text{fin}(T)) + 1$

SINTAXIS

⑨

Axiomas  $\forall e: \text{electroimen}, c: \text{curto}$

$\text{CeldaAOcupada}_e(e) \equiv$

IF  $\neg \text{Imarcayado}(e) \vee \neg \text{ImenPrendido}(e)$

then True else false fi

$\# \text{Giros} \leftarrow (e) \equiv \# \overleftarrow{\text{Giros}}(\text{CmtA}(e))$

$\# \text{Giros} \rightarrow (e) \equiv \# \overrightarrow{\text{Giros}}(\text{CmtB}(e))$

Axiomas

$\text{CeldaAOcupada}_c(c) \equiv$

$\text{CeldaAOcupada}(\text{Arrobar}(c))$

$\#E \text{ lemn?}$

o Banco decide analizar el comportamiento de los  
filas de clientes (Fila)

- a) \* APERTURA Ventanilla
- \* Llegada de nuevo cliente a la fila
  - \* Atención del Banco del cliente que estuvo en 1 lugar (agras a la fila tambien)
  - \* Vacar(fila)
  - \*  $long(fila) \equiv \# \text{clientes}$
  - \* cliente determinado  $\in$  fila (si este esperando o no)  
→ Posición en caso afirmativo.
- Posición 1 el Primer en llegar.

• Escribir el Tipo FILA y distinguir

TAD Fila

genera Fila

Exporta Abrir Ventanilla, Llegar, Atender, Esperando, Vacar,  
Posición Longitud

Use secuencia

Igualdad

$(\forall f, f': \text{fila}) (f = f' \iff \begin{cases} \text{vacar}(f) = \text{vacar}(f') \wedge \\ \text{Posición}(p, f) = \text{Posición}(p, f') \\ \text{Esperando}(p, f) = \text{Esperando}(p, f') \\ \text{Longitud}(p, f) = \text{Longitud}(p, f') \end{cases})$

OBSERVADORES BASICO

$\text{vacar: fila} \rightarrow \text{Bool}$

$\text{Posición: persona} \times \text{fila} \rightarrow \text{nat}$

$\text{longitud: fila} \rightarrow \text{nat}$

$\text{Esperando: persona} \times \text{fila} \rightarrow \text{Bool}$

$\{ \text{Esperando}(p, f) \}$

OTRAS OPERACIONES

$\text{Llegar: persona} \times \text{fila} \rightarrow \text{fila}$

$\text{Atender: fila} \rightarrow \text{fila}$

generadores

$\text{Abrir Ventanilla} \rightarrow \text{fila}$

$\{ \text{esperando}(p, f) \}$

$\{ \text{vacar}(f) \}$



## AXIOMAS

HP, file, HP persona

Vacio? ( $< >$ )  $\equiv$  True

Posicion (P, F)  $\equiv$

IF Prim(F) = P then 1 else 1 + Posicion (P, Fin(F)) fi

Longitud (F)  $\equiv$

IF Vacio?(F) then 0 else 1 + longitud (Fin(F)) fi

Españado (P, F)  $\equiv$

IF esta (P, F) then True else False fi

llagar (P, F)  $\equiv$

IF Vacio?(F) then P  $\bullet$   $< >$  else Prim(S)  $\bullet$  llagar (P, Fin(F)) fi

Alembic (F)  $\equiv$  fin(F)

## FIN TAD

- o donde se donantes de ser Alembic
- o donde colidese
- o agujeros y conos a la malochas

Se colat persona P x filo f  $\rightarrow$  Bobl {esperando (P, f)}

WOBs BASICO

Retirarse persona P x filo f  $\rightarrow$  filo {esperando (P, f)}

Coloca Alembic de persona P x persona q x filo f  $\rightarrow$  filo

{esperando (P, f)  $\wedge$  esperando (q, f)}

OTRAS OPERACIONES

## AXIOMAS

Retirarse (P, F)  $\equiv$

IF Vacio?(F) then  $< >$  else

IF P  $\neq$  Prim(F) then Prim(F)  $\bullet$  Retirarse (P, Fin(F))

else Retirarse (P, Fin(F))

fi

fi

Coloca Alembic de (P, q, f)  $\equiv$

IF Vacio?(f) then  $< >$  else

IF Prim(f) = q then P  $\bullet$  (f) else Prim(P)  $\bullet$  CAD (P, q, Fin(f))

fi fi

IF

© (10)

si el cliente  
o Ingreso (esperado, hego sido atendido o se hego retirado) en esto  
y si una persona determinada habra sido atendido durante  
el dia.

Axiomas

Entro? (P, F)  $\equiv$

Esperando (P, F)  $\underline{V}$

Fue Atendido (P, F)  $\underline{V}$

$(\neg \text{Fue Atendido (P, F)} \wedge \neg \text{Esperando (P, F)})$

OPS BASICO

Fue Atendido: PERSONA X F, L  $\rightarrow$  Bool

$\neg \text{Esperando (P, F)}$

11

TAD STOCK

Igualdad obs

stock = lotusol

$(\forall p, p', \text{Producto})$   $\left( \begin{array}{l} R_{\text{CONPRA}}(p) = R_{\text{CAPRA}}(p') \\ R_{\text{VENTA}}(p) = R_{\text{VENTA}}(p') \\ \text{Tiene Sustituto}(p) = \text{Tiene Sustituto}(p') \\ \text{Sustituto}(p) = \text{Sustituto}(p') \end{array} \right)$

generar stock

EXPORTA

OBS BASICOS

- Sustituto: Producto  $\rightarrow$  Producto

Tiene Sustituto: Producto  $\rightarrow$  Bool ✓

RCONPRA: Producto  $\rightarrow$  stock ✓

RVENTA: Producto  $\rightarrow$  stock ✓

generadores

Stock FINITO: Producto  $\rightarrow$  stock ✓

Operaciones Basicas

Stock Dstgo:  $\text{conj}(\text{conj}(\text{Producto})) \rightarrow \text{conj}(\text{Producto})$

Axiomas

$\text{conj}(C) = \{c\}$

#  $\text{conj}(C) = \{ \{c_1, c_2, \dots, c_n\}, \{c_1, c_2\}, \{c_1, c_3\}, \dots, \{c_{n-1}, c_n\} \}$   
Cada c / tiene un producto  
igual en su mismo conj

Sustituido Productos  $\& \text{conj}(\text{Producto}) \rightarrow \text{conj}(\text{Producto})$



IF  $\text{vacuo?}(c)$  then  $\emptyset$  else

IF  $[\# \text{doneUno}(c) \leq \text{stockMinimo}(\text{doneUno}(c))$   
 $\wedge \neg \text{Tiene Sustituto}(\text{doneUno}(\text{doneUno}(c)))]$

$\vee [\# \text{SustituyedoProducts}(\text{doneUno}(c)) +$   
 $\# \text{doneUno}(c) \leq \text{stockMinimo}(\text{doneUno}(\text{doneUno}(c)))]$

then  $\text{Ag}(\text{doneUno}(c), \text{stock}(\text{doneUno}(c)))$

else

$\text{stock}(\text{doneUno}(c))$

fi

fi

$cp: \text{copy to}(\text{products})$

$\text{SustituyeProducts}(cp) \equiv$

IF  $\text{vacuo?}(cp)$  then  $\emptyset$  else

IF  $\text{TieneSustituto}(\text{doneUno}(cp))$  then

$\text{Ag}(\text{Sustituto}(\text{doneUno}(cp)), \text{SustituyeProducts}(\text{doneUno}(cp)))$

else

$\text{SustituyeProducts}(\text{doneUno}(cp))$

fi

fi