

Práctica 1

Algo II1) Duplicar : $\text{secu}(\alpha) \xrightarrow{\alpha} \text{secu}(\alpha)$ $\text{Duplicar}(xs) = \text{if } xs \neq \text{vac}\alpha? (xs) \text{ then } <> \text{ else } \text{prim}(xs) \circ \text{prim}(xs) \circ \text{Duplicar}(\text{fin}(xs))$ b) \leq : $\text{secu}(\alpha) \times \text{secu}(\alpha) \xrightarrow{\alpha} \text{bool}$ $x_s \leq y_s = \text{if } \text{vac}\alpha? (xs) \text{ then } \text{true} \text{ else}$ if $\text{vac}\alpha? (ys) \text{ then false else}$ if $\text{prim}(xs) \equiv \text{prim}(ys) \text{ then } \text{fin}(xs) \leq \text{fin}(ys) \text{ else}$ $\text{prim}(xs) < \text{prim}(ys)$

fi

fi

fi

c) Reverso : $\text{secu}(\alpha) \xrightarrow{\alpha} \text{secu}(\alpha)$ $\text{Reverso}(xs) = \text{if } xs \equiv <> \text{ then } xs \text{ else}$ $\text{Reverso}(\text{fin}(xs)) \circ \text{prim}(xs)$

fi

d) Capicúa : $\text{secu}(\alpha) \xrightarrow{\alpha} \text{Bool}$ $\text{Capicúa}(xs) = \text{if } \text{vac}\alpha? (xs) \vee \text{vac}\alpha? (\text{fin}(xs)) \text{ then true else}$ $\text{cab}(xs) \equiv \text{ult}(xs) \wedge \text{Capicúa}(\text{com}(\text{fin}(xs)))$

fi

e) EsPrefijo? : $\text{secu}(\alpha) \times \text{secu}(\alpha) \xrightarrow{\alpha} \text{bool}$ $\text{EsPrefijo}(xs, ys) = \text{if } \text{long}(xs) > \text{long}(ys) \text{ then false else}$ if $\text{vac}\alpha? (ys) \text{ then false else}$ if $\text{prim}(xs) \equiv \text{prim}(ys) \wedge \text{EsPrefijo}(\text{fin}(xs), \text{fin}(ys))$

t) Buscar: $\text{secu}(\alpha) \times s \times \text{secu}(\alpha) \times s \rightarrow \text{Ent}$

Buscar(x_s, y_s, n) = if $\text{isPrefixo}(x_s, y_s)$ then n else Buscar($x_s, \text{fin}(y_s), n+1$) fi
// Solo sirve si le pases 0

Buscar: $\text{secu}(\text{secu}(\alpha)) \times s \times \text{secu}(\alpha) \times s \rightarrow \text{Ent}$

Buscar(x_s, x_s) = if $x_s \in \text{prim}(x_s)$ then 0 else $1 + \text{Buscar}(x_s, \text{fin}(x_s))$ fi

g) EstaOrdenada?: $\text{secu}(\alpha) \times s \rightarrow \text{bool}$

EstaOrdenada?(x_s) = if vaca?(x_s) \vee_L vaca?($\text{fin}(x_s)$) then true else
 $\text{prim}(x_s) < \text{prim}(\text{fin}(x_s)) \wedge \text{EstaOrdenada?}(\text{fin}(x_s))$
fi

h) InsertarOrdenada: $\text{secu}(\alpha) \times s \times \alpha \xrightarrow{x} \text{secu}(\alpha) \{ \dots \}$

InsertarOrdenada(x_s, x) = if vaca?(x_s) \vee_L $x \leq \text{prim}(x_s)$ then $x \cdot x_s$ else
 $\text{prim}(x_s) \cdot \text{InsertarOrdenada}(\text{fin}(x_s), x)$
fi

i) CantidadApariciones: $\text{secu}(\alpha) \times s \times \alpha \times x \rightarrow \text{Ent}$

CantidadApariciones(x_s, x) = if vaca?(x_s) then 0 else
if $x \equiv \text{prim}(x_s)$ then $1 + \text{CantidadApariciones}(\text{fin}(x_s), x)$
else CantidadApariciones($\text{fin}(x_s), x$)
fi

j) EsPermutacion?: $\text{secu}(\alpha) \times s - x \text{ secu}(\alpha) \ y s \rightarrow \text{bool}$

$\text{EsPermutacion?}(xs, ys) = \text{if } \neg (\text{long}(xs) = \text{long}(ys)) \text{ then false else}$

$\text{if } \text{vacia?}(xs) \text{ then true else } \text{cantApariciones}(xs, \text{prim}(xs)) = \text{cantApariciones}(ys, \text{prim}(xs)) \wedge$

$\text{EsPermutacion}(\text{fin}(xs), \text{sacar}(ys, \text{prim}(xs))) \circ \text{fin}(xs, \text{prim}(xs))$

fi

$\text{sacar}: \text{secu}(\alpha) \times s - x \times x \rightarrow \text{secu}(\alpha)$

$\text{sacar}(xs, x) = \text{if } \text{prim}(xs) = x \text{ then } \text{fin}(xs) \text{ else } \text{prim}(xs) \circ \text{sacar}(\text{fin}(xs), x)$

k) Combinar: $\text{secu}(\alpha) \times s \times \text{secu}(\alpha) \ y s \rightarrow \text{secu}(\alpha)$

$\text{Combinar}(xs, ys) = \text{if } \text{vacio?}(xs) \text{ then } ys \text{ else}$

$\text{if } \text{vacio?}(ys) \text{ then } xs \text{ else}$

$\text{if } \text{prim}(xs) \leq \text{prim}(ys) \text{ then } \text{Combinar}(\text{fin}(xs), ys) \circ \text{prim}(xs)$

$\text{else } \text{Combinar}(xs, \text{fin}(ys)) \circ \text{prim}(ys)$

fi

fi

fi

2) $\# \text{Hojas} : ab(\alpha) \rightarrow \text{Nat}$

$\# \text{Hojas}(\alpha) = \text{if } \alpha \text{ nil?}(\alpha) \text{ then } 0 \text{ else if}$

if $\text{EsHoja?}(\alpha)$ then 1 else $\# \text{Hojas}(\text{izq}(\alpha)) + \# \text{Hojas}(\text{der}(\alpha))$
fi

b) $\text{DegeneradoAIzquierda} : ab(\alpha) \alpha \rightarrow \text{bool}$

$\text{DegeneradoAIzquierda}(\alpha) = \text{if } \alpha \text{ nil?}(\alpha) \text{ then true else}$

if $\neg(\text{der}(\alpha) \equiv \text{nil})$ then false else

$\text{DegeneradoAIzquierda}(\text{izq}(\alpha))$

fi

fi

c) $\text{ZigZag} : ab(\alpha) \alpha \rightarrow \text{bool}$

$\text{ZigZag}(\alpha) = \text{if } \alpha \text{ nil?}(\alpha) \text{ then true else}$

$\neg(\text{nil?}(\text{izq}(\alpha)) \Rightarrow_L (\neg \text{nil?}(\text{der}(\alpha)) \wedge \text{ZigZag}(\text{izq}(\alpha)))) \vee \wedge$
 $(\text{nil?}(\text{der}(\alpha)) \Rightarrow_L (\neg \text{nil?}(\text{izq}(\alpha)) \wedge \text{ZigZag}(\text{der}(\alpha))))$

fi

d) $\text{UltimoNivelCompleto} : ab(\alpha) \alpha \rightarrow \text{Ent}$

$\text{UltimoNivelCompleto}(\alpha) = \text{if } \alpha \text{ nil?}(\alpha) \text{ then } 0 \text{ else } [\text{if } \text{pot}(2, \text{altura}(\alpha) - 1) \equiv \# \text{Hojas}(\alpha)$

then $\text{altura}(\alpha)$ else $\text{UltimoNivelCompleto}(\text{limpiarUltimoNivel}(\alpha))$ fi]

fi

$\text{pot} : \text{Nat} b \times \text{Nat} e \rightarrow \text{Nat}$

$\text{pot}(b, e) = \text{if } e \equiv 0 \text{ then } 1 \text{ else } b \times \text{pot}(b, e-1)$

$\text{LimpiarUltimoNivel} : ab(\alpha) \alpha \rightarrow ab(\alpha)$

$\text{LimpiarUltimoNivel}(\alpha) =$

e) Espejo: $\text{ab}(\alpha) \rightarrow \text{ab}(\alpha)$

$\text{Espejo}(\alpha) = \text{if } \text{nil?}(\alpha) \text{ then nil else bin}(\text{espejo}(\text{der}(\alpha)), \text{raiz}(\alpha), \text{espejo}(\text{izq}(\alpha)))\text{fi}$

f) EsSimetrico?: $\text{ab}(\alpha) \rightarrow \text{bool}$

$\text{EsSimetrico?}(\alpha) = \alpha \equiv \text{espejo}(\alpha)$

3) PartesDe: $\text{conj}(\alpha) \rightarrow \text{conj}(\text{conj}(\alpha))$

$\text{PartesDe}(\alpha) = \text{if } \emptyset?(\alpha) \text{ then } \emptyset \text{ else }$

AgregarAT (PartesDe (sinUno(α)), DameUno(xs)) U
PartesDe (sinUno(xs))

fi

4)

5) a) NivelNormal? : $\text{Bt}(\alpha) \times \text{Nat}_n \rightarrow \text{Bool}$

$\text{NivelNormal?}(a, n) = \text{if } \text{nil?}(a) \text{ then false else }$

$\text{if } k \geq 0 \text{ then false else }$

$\text{if } k = 1 \text{ then } (\neg \text{nil?}(\text{izq}(a)) \wedge \neg \text{nil?}(\text{med}(a)) \wedge \neg \text{nil?}(\text{der}(a)))$

$\text{else } [\text{NivelNormal}(\text{izq}(a), n-1) \wedge \text{NivelNormal}(\text{med}(a), n-1) \wedge$

$\text{NivelNormal}(\text{der}(a), n-1)]$

fi

fi

fi

Parte 2)

POLINOMIO7) ($\forall a, n : \text{Nat}$), $p_1, p_2 : \text{polinomio}$):

$$\text{Evaluar}(\text{Cte}(n), a) = n$$

$$\text{Evaluar}(X, a) = a$$

$$\text{Evaluar}(p_1 + p_2, a) = \text{Evaluar}(p_1, a) + \text{Evaluar}(p_2, a)$$

$$\text{Evaluar}(p_1 \cdot p_2, a) = \text{Evaluar}(p_1, a) \cdot \text{Evaluar}(p_2, a)$$

EsRaiz? : polinomio $\times \text{Nat} \rightarrow \text{Bool}$

$$\text{EsRaiz?}(\text{Cte}(n), a) = n = 0?$$

$$\text{EsRaiz?}(X, n) = n = 0?$$

$$\text{EsRaiz?}(p_1 + p_2, n) = \text{Evaluar}(p_1 + p_2, n) = 0?$$

$$\text{EsRaiz?}(p_1 \cdot p_2, n) = \text{Evaluar}(p_1 \cdot p_2, n) = 0?$$

8)

ROBOT

$$\text{Trayectoria}(\text{Ubicar}(c)) = c \cdot \langle \rangle$$

$$\text{Trayectoria}(\text{Arriba}(r)) = \text{Trayectoria}(r) \circ \text{SubeY}(\text{Ult}(\text{Trayectoria}(r)))$$

$$\text{Trayectoria}(\text{Abajo}(r)) = \text{Trayectoria}(r) \circ \text{BajaY}(\text{Ult}(\text{Trayectoria}(r)))$$

$$\text{Trayectoria}(\text{Derecha}(r)) = \text{Trayectoria}(r) \circ \text{SubeX}(\text{Ult}(\text{Trayectoria}(r)))$$

$$\text{Trayectoria}(\text{Izquierda}(r)) = \text{Trayectoria}(r) \circ \text{BajaX}(\text{Ult}(\text{Trayectoria}(r)))$$

$$\text{PosicionActual}(r) = \text{Ult}(\text{Trayectoria}(r))$$

$$\text{CuantasVecesPasó}(c, r) = \text{CantidadApariciones}(\text{Trayectoria}(r), c)$$

$$\text{MásALaDerecha}(r) = \text{MaxX}(\text{Trayectoria}(r))$$

SubeY: coordenada $c \rightarrow$ coordenada | SubeX: coordenada $c \rightarrow$ coordenada

$$\text{SubeY}(c) = \langle \pi_1(c), \pi_2(c)+1 \rangle$$

$$\text{SubeX}(c) = \langle \pi_1(c)+1, \pi_2(c) \rangle$$

BajaY: coordenada $c \rightarrow$ coordenada | BajaX: coordenada $c \rightarrow$ coordenada

$$\text{BajaY}(c) = \langle \pi_1(c), \pi_2(c)-1 \rangle$$

$$\text{BajaX}(c) = \langle \pi_1(c)-1, \pi_2(c) \rangle$$

MaxX: Secu(coordenadas) cs \rightarrow coordenada

MaxX(cs) = if vacio? (fin(cs)) then prim(cs) else
if $\pi_1(\text{prim}(cs)) \geq \pi_2(\text{prim}(\text{fin}(cs)))$ then MaxX
MaxX(prim(cs) o fin(fin(cs)))
else MaxX(fin(cs))
fi
fi

// Otro

MásAlaDerecha(Derecha(r)) = PosicionActual(Derecha(r))

MásAlaDerecha(Izquierda(r)) = MásAlaDerecha(r)

MásAlaDerecha(Abajo(r)) = MásAlaDerecha(r)

MásAlaDerecha(Arriba(r)) = MásAlaDerecha(r)

MásAlaDerecha(Ubicar(r, c)) = c

CINTA

• 9) igualdad observational:

$$(\forall c_1, c_2, c: \text{cinta}) \left(c_1 = c_2 \Leftrightarrow \begin{array}{l} \#\text{Celdas}(c_1) = \#\text{Celdas}(c_2) \wedge \\ (\forall n: \text{Nat}, n < \#\text{Celdas}(c_1)) \text{CeldaOcupada?}(n, c_1) \equiv \text{CeldaOcupada?}(n, c_2) \wedge \\ \text{CeldaActual}(c_1) = \text{CeldaActual}(c_2) \wedge \\ \#\text{Giros} \leftarrow (c_1) = \#\text{Giros} \leftarrow (c_2) \wedge \\ \#\text{Giros} \rightarrow (c_1) = \#\text{Giros} \rightarrow (c_2) \end{array} \right)$$

($\forall n, m: \text{Nat}, c: \text{cinta}$):

$\#\text{Celdas}(\text{Arrancar}(n)) = n$

$\text{CeldaActual}(\text{Arrancar}(n)) = 0$

$\#\text{Celdas}(\text{PonerElem}(c)) = \#\text{Celdas}(c)$

$\text{CeldaActual}(\text{PonerElem}(c)) = \text{CeldaActual}(c)$

$\#\text{Celdas}(\text{SacarElem}(c)) = \#\text{Celdas}(c)$

$\text{CeldaActual}(\text{SacarElem}(c)) = \text{CeldaActual}(c)$

$\#\text{Celdas}(\leftarrow(c)) = \#\text{Celdas}(c)$

$\text{CeldaActual}(\leftarrow(c)) = \begin{cases} \text{CeldaActual}(c) & \text{if } \text{CeldaActual}(c) = 0 \\ \#\text{Celdas}(c) - 1 & \text{else} \\ \text{CeldaActual}(c) + 1 & \end{cases}$

$\#\text{Celdas}(\rightarrow(c)) = \#\text{Celdas}(c)$

$\text{CeldaActual}(\rightarrow(c)) = \begin{cases} \text{CeldaActual}(c) - \#\text{Celdas}(c) + 1 & \text{if } \text{CeldaActual}(c) = \#\text{Celdas}(c) - 1 \\ 0 & \text{else} \\ \text{CeldaActual}(c) + 1 & \end{cases}$

NOTA

$$\# \text{Giros} \leftarrow (\text{Arrancar}(n)) = 0$$

$$\# \text{Giros} \leftarrow (\text{PonerElem}(c)) = \# \text{Giros}(c)$$

$$\# \text{Giros} \leftarrow (\text{SacarElem}(c)) = \# \text{Giros}(c)$$

$$\# \text{Giros} \leftarrow (\leftarrow(c)) = 1 + \# \text{Giros}(c)$$

$$\# \text{Giros} \leftarrow (\rightarrow(c)) = \# \text{Giros}(c)$$

$$\# \text{Giros} \rightarrow (\text{Arrancar}(n)) = 0$$

$$\# \text{Giros} \rightarrow (\text{PonerElem}(c)) = \# \text{Giros} \rightarrow (c)$$

$$\# \text{Giros} \rightarrow (\text{SacarElem}(c)) = \# \text{Giros} \rightarrow (c)$$

$$\# \text{Giros} \rightarrow (\leftarrow(c)) = \# \text{Giros} \rightarrow (c)$$

$$\# \text{Giros} \rightarrow (\rightarrow(c)) = 1 + \# \text{Giros} \rightarrow (c)$$

$$\text{CeldaOcupada?}(m, \text{Arrancar}(n)) = \text{false}$$

$$\text{CeldaOcupada?}(m, \text{PonerElem}(c)) = \text{if } \text{CeldaActual}(c) \in m \text{ then true else } \text{CeldaOcupada?}(c, \text{fi})$$

$$\text{CeldaOcupada?}(m, \text{SacarElem}(c)) = \text{if } \text{CeldaActual}(c) \in m \text{ then false else } \text{CeldaOcupada?}(c, \text{fi})$$

$$\text{CeldaOcupada?}(m, \leftarrow(c)) = \text{CeldaOcupada?}(c)$$

$$\text{CeldaOcupada?}(m, \rightarrow(c)) = \text{CeldaOcupada?}(c)$$

$$\text{CeldaActualOcupada?}(c) = \text{CeldaOcupada?}(\text{CeldaActual}(c))$$

$$\# \text{Elem}(\text{Arrancar}(n)) = 0$$

$$\# \text{Elem}(\text{PonerElem}(c)) = 1 + \# \text{Elem}(c)$$

$$\# \text{Elem}(\text{SacarElem}(c)) = \# \text{Elem}(c) - 1$$

$$\# \text{Elem}(\leftarrow(c)) = \# \text{Elem}(c)$$

$$\# \text{Elem}(\rightarrow(c)) = \# \text{Elem}(c)$$

ELECTRODOMÉN

($\forall C : \text{cinta}, e : \text{electrodomén}$) :

$\text{Cinta}(\text{Arrancar}(c)) = c$

$\text{Cinta}(\text{Prender}(e)) = \text{if CeldaActualOcupada?}(e) \text{ then SacarElem}(\text{Cinta}(e))$
 $\text{else Cinta}(e) \text{ fi}$

$\text{Cinta}(\text{Apagar}(e)) = \text{if ImánCargado?}(e) \text{ then PonerElem}(\text{Cinta}(e)) \text{ else}$
 $\text{Cinta}(e) \text{ fi}$

$\text{Cinta}(\leftarrow(e)) = \leftarrow(\text{Cinta}(e))$

$\text{Cinta}(\rightarrow(e)) = \rightarrow(\text{Cinta}(e))$

$\text{ImánPrendido?}(\text{Arrancar}(c)) = \text{false}$

$\text{ImánPrendido?}(\text{Prender}(e)) = \text{true}$

$\text{ImánPrendido?}(\text{Apagar}(e)) = \text{false}$

$\text{ImánPrendido?}(\leftarrow(e)) = \text{ImánPrendido?}(e)$

$\text{ImánPrendido?}(\rightarrow(e)) = \text{ImánPrendido?}(e)$

$\text{ImánCargado?}(\text{Arrancar}(c)) = \text{false}$

$\text{ImánCargado?}(\text{Prender}(e)) = \text{CeldaActualOcupada?}(e)$

$\text{ImánCargado?}(\text{Apagar}(e)) = \text{false}$

$\text{ImánCargado?}(\leftarrow(e)) = \text{ImánCargado?}(e) \vee (\text{ImánPrendido?}(e) \wedge$

$(\text{ImánPrendido?}(e) \wedge \text{CeldaActualOcupada?}(\leftarrow(e)))$

$\text{ImánCargado?}(\rightarrow(e)) = \text{ImánCargado?}(e) \vee (\text{ImánPrendido?}(e) \wedge \text{CeldaActualOcupada?}(\rightarrow(e)))$

$\text{CeldaActualOcupada?}(\text{Arrancar}(c)) = \text{CeldaActualOcupada?}(c)$

$\text{CeldaActualOcupada?}(\text{Prender}(e)) = \text{false}$

$\text{CeldaActualOcupada?}(\text{Apagar}(e)) = \text{ImánCargado?}(e) \vee \text{CeldaActualOcupada?}(\text{Cinta}(e))$

$\text{CeldaActualOcupada?}(\leftarrow(e)) = \text{CeldaActualOcupada?}(\leftarrow(\text{Cinta}(e)))$

$\text{CeldaActualOcupada?}(\rightarrow(e)) = \text{CeldaActualOcupada?}(\rightarrow(\text{Cinta}(e)))$

$\# \text{Giros} \leftarrow (\text{Arranclar}(c)) = 0$

$$\# \text{Giro} \leftarrow (\text{Prender}(e)) = \# \text{Giro} \leftarrow (e)$$

$$\#Ginos \leftarrow (\text{Apagir}(e)) = \#Gitos \leftarrow (e)$$

$$\#Gicos^{\leftarrow}(\leftarrow(e)) = 1 + \#Gicos^{\leftarrow}(e)$$

$$\# \text{Grost}^{\leftarrow}(\rightarrow(e)) = \# \text{Grost}^{\leftarrow}(e)$$

Giros < (Arrancar(c)) = 0

$$\#G_{\text{iros}} \rightarrow (\text{Prender}(e)) = \#G_{\text{iros}} \rightarrow (e)$$

$$\# \text{Gros} \rightarrow (\text{Aperge}(e)) = \# \text{Gros} \rightarrow (e)$$

$$\# \text{Giros} \rightarrow (\leftarrow(e)) = \# \text{Giros} \rightarrow (e)$$

$$\# \text{Girós}(\rightarrow(e)) = \# \text{Girós}(e) + 1$$

10) Quiero:

| FILA

• Distinguir clientes, sabe si alguno en particular está esperando

• Tener los ordenados → Atiendo al primero → Sale de la fila

• Longitud de la fila

~~Accio~~?

TAD FILA

igualdad observacional:

($\forall f_1, f_2 : \text{fila}$) ($f_1 = f_2 \iff (\forall p : \text{personas}) [\text{Esperando}(p, f_1) \equiv \text{Esperando}(p, f_2)] \wedge_L$)

exporta

observadores básicos

Esperando: persona \times fila \rightarrow bool

Posición: persona p \times fila f \rightarrow nat

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

Axiomas

(A)

Esperando(p , AbrirVentanilla) = falso

Esperando(p , Llegar(p', f)) = if ($p \equiv p'$) then true else Esperando(p, f) fi

Esperando(p , ColarseAdelanteDe(q, r, f)) = if $p \equiv q$ then true else

Esperando(p, f)

fi

Posición($p', Llegar(p, f)$) = if $p' \equiv p$ then Longitud(f) + 1 else Posición(p, f) fi

Posición($p, ColarseAdelanteDe(q, r, f)$) = if ($p \equiv q$) then Posición(r, f) else

if ($p \equiv r$) then Posición(r, f) + 1

else if Llegar(p, f)

if EstaAntes(p, r, f) then

Posición(p, f) else

1 + Posición(p, f)

fi

fi

fi

SeColó?($p', Llegar(p, f)$) = $\neg(p \equiv p')$ \wedge SeColó?(p', f)

SeColó?($p, ColarseAdelanteDe(q, r, f)$) = $(p \equiv q) \vee$ SeColó?(p, f)

Atender(Llegar(p, f)) = if vacía(f) then f else Llegar($p, Atender(f)$) fi

Atender(ColarseAdelanteDe(p, q, f)) = if Longitud(f) = 1 then f else

ColarseAntesDe($p, q, Atender(f)$)

fi

Vacia(AbrirVentanilla) = true

Vacia(Llegar(p, f)) = false

Vacia(ColarseAdelanteDe(p, q, f)) = false

Longitud(AbrirVentanilla) = 0

Longitud(Llegar(p,f)) = 1 + Longitud(f)

Longitud(ColarseAdelanteDe(p,q,f)) = 1 + Longitud(f)

Retirarse(p',Llegar(p,f)) = if ($p' \equiv p$) then f else Retirarse(p',f) fi

Retirarse(p, ColarseAdelanteDe(q,r,f)) = if ($p \equiv q$) then f else
if ($p \equiv r$) then
if Posición(r,f) = Longitud(f)
then Llegar(q,Retirarse(p,f))
else

ColarseAdelanteDe(q,ElDeAtrás(f,
Retirarse(r,f)))

fi

else

ColarseAdelanteDe(q,r,Retirarse(p,f))

fi

fi

auxiliares (deberían ir en otras op)

EstáAntes: persona p × persona q × fila f

. { Esperando(p,f) ∧ Esperando(q,f) }

EstáAntes(p,q,Llegar(p',f)) = $\neg(p \equiv p') \wedge (q \equiv p')$ ∨L EstáAntes(p,q,f)

EstáAntes(p,q, ColarseAdelanteDe(r,s,f)) = if ($p \equiv r$) then true

if ($q \equiv s$) then true

else EstáAntes(s,q,f)

fi

else

if ($p \equiv s$) then

if ($q \equiv r$) then false

else EstáAntes(p,q,f)

else

EstáAntes(p,q,f)

E|DeAtrás: persona $p \rightarrow$ fila $f \rightarrow$ persona

{ Esperando(p, f) \wedge Posición(p, f) < Longitud(f) + 1 }

E|DeAtrás($p, Llegar(q, Llegar(r, f))$) = if ($p \equiv r$) then q

else E|DeAtrás($p, Llegar(r, f)$)
fi.

E|DeAtrás($p, ColorseAdelanteDe(q, r, f)$) = if ($p \equiv q$) then r

else E|DeAtrás(p, f)
fi

10/b

Observadores básicos

generadores

Esperando, Posición, SeColó? AbrirVentanilla, Llegar, ColarseAdelanteDe, Retirarse
otras op.

Atender, Vacía, Longitud, ElDeAtrás, ElÚltimo

Esperando(p,AbrirVentanilla) = false

Esperando(p,Llegar(p',f)) = (p = p') \vee Esperando(p,f)Esperando(p, ColarseAdelanteDe(q,r,f)) = (p = q) \vee Esperando(p,f)Esperando(p, Retirarse(p',f)) = $\neg(p = p')$ \wedge Esperando(p,f)

Posición(p,Llegar(p',f)) = if (p = p') then Longitud(f)+1 else Posición(p,f) fi

Posición(p, ColarseAdelanteDe(q,r,f)) = if (p = q) then Posición(r,f)

else if (p = r) then Posición(s,f) + 1

else if EstáAntes(p,r,f) then Posición(p,f)

else Posición(p,f) + 1

fi

fi

fi

Posición(p,Retirarse(p',f)) = Posición(p,f)

SeColó?(p,Llegar(p',f)) = $\neg(p = p')$ \wedge SeColó?(p,f)SeColó?(p, ColarseAdelanteDe(q,r,f)) = (p = q) \vee SeColó?(p,f)

SeColó?(p, Retirarse(p',f)) = SeColó?(p,f)

Atender(Llegar(p,f)) = if Vacía(f) then f else Llegar(p,Atender(f))

Atender(ColarseAdelanteDe(p,q,f)) = if Longitud(f) = 1 then f else

ColarseAdelanteDe(p,q,Atender(f))

fi

Atender(Retirarse(p,f)) = if Posición(p,f) \neq 1 then Retirarse(ElDeAtrás(p,f),Retirarse(p,f))

else Retirarse(p,Atender(f))

fi

Vacia(AbrirVentanilla) = true

Vacia(Llegar(p,f)) = false

Vacia(ColarseAdelanteDe(p,q,f)) = false

Vacia(Retirarse(p,f)) = Longitud(f) = 1

Longitud(AbrirVentanilla) = 0

Longitud(Llegar(p,f)) = Longitud(f) + 1

Longitud(ColarseAdelanteDe(p,q,f)) = Longitud(f) + 1

Longitud(Retirarse(p,f)) = Longitud(f) - 1

ElDeAtrás(p,Llegar(p',f)) = if ($p \equiv p'$) then ElÚltimo(f) else ElDeAtrás(p,f)

ElDeAtrás(p,ColarseAdelanteDe(q,r,f)) = if ($p \equiv q$) then r else ElDeAtrás(p,f) fi

ElDeAtrás(p,Retirarse(p',f)) = if $p' = \text{ElDeAtrás}(p,f)$ then ElDeAtrás(p',f)
else ElDeAtrás(p,f)

ElÚltimo(Llegar(p,f)) = p

ElÚltimo(ColarseAdelanteDe(p,q,f)) = q

ElÚltimo(Retirarse(p,f))

BIBLIOTECA

II) a

TAD BIBLIOTECA

observadores básicos

LibrosDeAutor : autor $a \rightarrow$ con $\{(\text{libro}) \mid a \in \text{Autores}(b)\}$

Autores : biblioteca $b \rightarrow$ con $\{(\text{autor})\}$
generador

Nueva Biblioteca : \rightarrow biblioteca

Ag.Libro : libro $l \times$ autor $a \times$ biblioteca $b \rightarrow$ biblioteca

otras operaciones

DeQuienEs : libro $l \times$ biblioteca $b \rightarrow$ autor

$\{(\exists a \in \text{Autores}(b)) \mid l \in \text{LibrosDeAutor}(a, b)\}$

LibrosDeAutor : $(a, \text{Ag.Libro}(l, a, b)) =$ if Vacia?(b)

then if ($a \equiv a'$)

then $\{\cdot\}$

else \emptyset

else if ($a \equiv a'$)

then Ag(\cdot , LibrosDeAutor(a, b))

else LibrosDeAutor(a, b)

fi

fi

DeQuienEs($l, \text{Ag.Libro}(l, a, b)$) = ~~if ($l \equiv l'$) then a else DeQuienEs(l, b) fi~~ if ($l \equiv l'$) then a else DeQuienEs(l, b) fi

Autores(NuevaBiblioteca) = \emptyset

Autores(Ag.Libro(l, a, b)) = Ag(A, Autores(b))

b) TAD BIBLIOTECA

genero biblioteca

observadores

LibrosDeAutor : Autor a \times biblioteca b \rightarrow conj(libro) $\{ a \in Autores(b) \}$

Autores : biblioteca b \rightarrow conj(autor)

ISBNLibro : biblioteca b \rightarrow diccionario(ISBN, libro)

generadores

NuevaBiblioteca : \rightarrow biblioteca

AglLibro : libro l \times ISBN i \times autor a \times biblioteca b \rightarrow biblioteca

otras operaciones

DeQuentes : ISBN i \times biblioteca b \rightarrow autor $\{ i \in ISBNLibro(b) \}$

expresiones

Autores(NuevaBiblioteca) = \emptyset

Autores(AglLibro(l, i, a, b)) = Agl(A, Autores(b))

ISBNLibro(NuevaBiblioteca) = vacio

ISBNLibro(AglLibro(l, i, a, b)) = definir (i, l, ISBNLibro(b))

LibrosDeAutor(a, AglLibro(l, i, a', b)) = if (b = NuevaBiblioteca)

then if (a = a')

then $\{ \}$

else \emptyset

• fi

else if (a = a')

then Agl(a, LibrosDeAutor(a, b))

else LibrosDeAutor(a, b)

fi