22 de marzo de 2007

1. Signatura de los TADs

TAD USORECURSOS

géneros usoRecursos

exporta usoRecursos, generadores, observadores, menorConsumo

usa Nat, Tarea, Recursos, Bool

igualdad observacional

$$(\forall a, b : \text{workflow}) \left(a =_{\text{obs}} b \Longleftrightarrow \begin{pmatrix} \text{tiposDeRecursos } (a) =_{\text{obs}} \text{tiposDeRecursos } (b) \wedge_{\text{L}} (\forall\ r\text{: re-} \\ \text{curso}) \wedge_{\text{L}} \text{verTotal } (a,r) =_{\text{obs}} \text{verTotal } (b,r) \wedge \text{disponibili-} \\ \text{dad } (a,r) =_{\text{obs}} \text{disponibilidad } (b,r) \end{pmatrix} \right)$$

observadores básicos

tiposDeRecursos	: usoRecursos	$\longrightarrow$ nat	
verTotal	: usoRecursos u $\times$ recurso r	$\longrightarrow$ nat	{r < tiposDeRecuros(u)}
disponibilidad	: usoRecurs u $\times$ recurso r	$\longrightarrow$ nat	{r < tiposDeRecuros(u)}

generadores

generar	: recursos	$\longrightarrow$ usoRecursos	
nuevoConsumo	: usoRecursos u $\times$ recurso r $\times$ nat n	$\longrightarrow$ usoRecursos	{r < tipoDeRecursos(u) $\wedge$ n verTotal(u,r)}

otras operaciones

menorConsumo	: usoRecursos u	$\longrightarrow$ recurso	{tipoDeRecursos(u) > 0}
actualizarConsumo	: usoRecursos u $\times$ recursos $mconj$	$\longrightarrow$ usoRecursos	{($\forall$ r: recurso) $r \in$ mconj $\Rightarrow_{\text{L}}$ ($r <$ tiposDeRecursos $u \wedge_{\text{L}} \#$ (r, $mconj$) < verTotal (u, i))}
actualizarUso	: usoRecursos u $\times$ arregloDimensionable (nat) ar	$\longrightarrow$ usoRecursos	{tiposDeRecursos (u) == tam (ar) $\wedge$ ($\forall$ i: nat) $i <$ tam (ar) $\Rightarrow_{\text{L}}$ $ar[i] <$ verTotal (u, i)}
alcanzanLosRecursos	: usoRecursos u $\times$ recursos $mconj$	$\longrightarrow$ bool	{($\forall$ r: recurso) $r \in$ mconj $\Rightarrow_{\text{L}}$ ($r <$ tiposDeRecursos(u))}
multiconjuntizar	: usoRecursos u	$\longrightarrow$ recursos	
disponiblesEnMultiConj	usoRecursos u	$\longrightarrow$ recursos	
actualizarUsoAux	: usoRecursos u $\times$ arregloDimensionable (nat) ar $\times$ nat n	$\longrightarrow$ usoRecursos	{tiposDeRecursos (u) == tam (ar) $\wedge$ ($\forall$ i: nat) $i <$ tam (ar) $\Rightarrow_{\text{L}}$ $ar[i] <$ verTotal (u, i)}
multiconjuntizarAux	: usoRecursos u $\times$ nat n	$\longrightarrow$ recursos	
agregarMuchos	: recursos $mconj$ $\times$ recurso r $\times$ nat n	$\longrightarrow$ recursos	
disponiblesEnMultiConjAux	: usoRecursos u $\times$ nat n	$\longrightarrow$ recursos	

axiomas

$\forall$ u: usoRecursos, $\forall$ $rs, mconj$: recursos, $\forall$ $r1, r$: recurso, $\forall$ n, i: nat

tiposDeRecursos (crear (rs) $\equiv$ cantidadDeRec(recursos)
tiposDeRecursos (nuevoConsumo (u,r,n) $\equiv$ tiposDeRecursos(u)

verTotal (crear (rs),r) $\equiv$ (r,rs)
verTotal (nuevoConsumo (u,r,n),$r1$) $\equiv$ tipoDeRecursos(u)

disponibilidad(crear (rs),r) $\equiv$ (r,rs)

disponibilidad (nuevoConsumo (u,r,n),$r1$) $\equiv$ **if** r=dameUno($r1$) **then** n **else** disponibilidad (u) **fi**

menorConsumo (u) $\equiv$ dameElMinimo (u,0,0)

dameElMinimo (u,n,min) $\equiv$ **if** $n ==$ cantElemDistintos(u) **then** min **else if** ((verTotal(u,n) - disponibilidad (u,n)< min) **then** dameElMinimo (u,n+1,(verTotal(u,n) - disponibilidad (u,n)) **else** dameElMinimo (u,n+1,min

actualizarConsumo (u, $mconj$) $\equiv$ **if** vacio? ($mconj$) **then** u **else** actualizarConsumo (nuevoConsumo (u, dameUno ($mconj$), # (dameUno ($mconj$), $mconj$)), $mconj$ \ dameUno $mconj$) **fi**

actualizarUso (u, ar) $\equiv$ actualizarUsoAux ($u, ar, 0$)

actualizarUsoAux (u, ar, n) $\equiv$ **if** $n ==$ tiposDeRecur

â

sos (u) **then** u **else** actualizarUsoAux (nuevoConsumo (u, n, $ar[n]$), ar, n+1)

alcanzanLosRecursos (u, $mconj$) $\equiv$ **if** vacio? ($mconj$) **then** true **else** disponibilidad (u, dameUno ($mconj$)) $\geq$ # (dameUno ($mconj$), $mconj$)) *wedge* alcanzanLosRecursos (u, $mconj$ \ dameUno $mconj$) **fi**

multiconjuntizar (u) $\equiv$ multiconjuntizarAux (u,0)

multiconjuntizarAux (u,n) $\equiv$ **if** $n ==$ tiposDeRecursos (u) **then** vacio **else** agregarMuchos (multiconjuntizarAux (u, $n+1$),n, $verTotal$(u,n)

disponiblesEnMulticonj (u) $\equiv$ disponiblesEnMulticonjAux (u,0)

disponiblesEnMulticonjAux (u,n) $\equiv$ **if** $n ==$ tiposDeRecursos (u) **then** vacio **else** agregarMuchos (disponiblesEnMulticonjAux (u, $n+1$),n, $disponibilidad$(u,n)

agregarMuchos ($mconj$,r,n) $\equiv$ **if** $n == 0$ (u) **then** $mconj$ **else** agregarMuchos (Ag (r, $mconj$), r, $n-1$)

Fin TAD

TAD ITERADOR

géneros Iterador

exporta iterador, generadores, observadores, otras operaciones

usa Nat, Bool

igualdad observacional

$$(\forall a, b : \text{iterador}) \left(a =_{\text{obs}} b \iff \begin{pmatrix} \text{secuOriginal } (a) =_{\text{obs}} \text{secuOriginal } (b) \wedge \text{HayMas } (a) =_{\text{obs}} \\ \text{HayMas } (b) \wedge \text{Actual } (a) =_{\text{obs}} \text{Actual } (b) \wedge \text{Avanzar } (a) \\ =_{\text{obs}} \text{Avanzar } (b) \end{pmatrix} \right)$$

observadores básicos

secuOriginal	: iterador(α) i	$\longrightarrow$ secu(α)	
HayMas?	: iterador(α) i	$\longrightarrow$ bool	
Actual	: iterador(α) i	$\longrightarrow$ nat	{HayMas?(i)}
Avanzar	: iterador(α) i	$\longrightarrow$ nat	{HayMas?(i)}

generadores

crearIt : secu(α) i secu(α) s $\longrightarrow$ iterador(α)

otras operaciones

agregarAtrasDeIt	: iterador(α) i $\times$ nat n	$\longrightarrow$ iterador(α)
agregarAdelanteDeIt	: iterador(α) i $\times$ nat n	$\longrightarrow$ iterador(α)
retrocederAlPrincipio	: iterador(α) i	$\longrightarrow$ iterador(α)
borrarActual	: iterador(α) i	$\longrightarrow$ iterador(α)
hastaIt	: secu(α) i $\times$ secu(α) s	$\longrightarrow$ Secu(α)
tomar	: secu(α)i $\times$ nat n	$\longrightarrow$ Secu(α)

axiomas

$\forall$ i,s: secu(α), $\forall$ n:nat

secuOriginal (crearIt (i,s)) $\equiv$ s
HayMas (crearIt (i,s)) $\equiv$ Vacia?(s)
Actual (crearIt (i,s)) $\equiv$ prim(s)
Avanzar (crearIt (i,s)) $\equiv$ crearIt($i \circ$ prim(s),s)
Actual (crearIt (i),s) $\equiv$ prim(s)
AgregarAtrasDeIt (crearIt (i,s)),n $\equiv$ crearIt(hastaIt(i,s)&& ($i \circ n$),s)
AgregarAdelanteDeIt (crearIt (i,s)),n $\equiv$ crearIt(hastaIt(i,s)&& ($n \bullet i$),s)
retrocederAlPrincio (crearIt (i,s)) $\equiv$ crearIt (<>,s)
borrarActual (crearIt (i,s)) $\equiv$ crearIt (hastaIt(i,s)&&fin(i)),s)
hastaIt(i, s) $\equiv$ tomar(i,long(s)$-$long(i))
tomar(i,n) $\equiv$ **if** n=0 **then** <> **else** prim(i) $\bullet$ tomar (fin(i), n$-$1)

Fin TAD

TAD COLA

géneros cola

exporta generadores, observadores, otras operaciones

usa Nat, Bool, Secu (α)

igualdad observacional

$$(\forall a, b : \text{cola}) \left(a =_{\text{obs}} b \iff \begin{pmatrix} \text{vacia } (a) =_{\text{obs}} \text{vacia } (b) \wedge \text{proxima } (a) =_{\text{obs}} \text{proxima } (b) \wedge \text{pri-} \\ \text{oridad } (a) =_{\text{obs}} \text{prioridad } (b) \wedge_{\text{L}} (\forall\ t\text{: tarea}) \text{ ordenes } (a,t) =_{\text{obs}} \\ \text{ordenes } (b,t) \wedge_{\text{L}} (\forall\ n\text{: nat}) \text{ sacarOrden } (a,t,n) =_{\text{obs}} \text{sacarOrden} \\ (b,t,n) \end{pmatrix} \right)$$

observadores básicos

vacia?	: cola c	$\longrightarrow$ bool		
proxima	: cola c	$\longrightarrow$ actividad		$\{\neq$ vacia?(c)$\}$
prioridad	: cola c	$\longrightarrow$ nat		$\{\neq$ vacia?(c)$\}$
ordenes	: cola $c \times$ tarea t	$\longrightarrow$ secu(Orden)		
sacarOrden	: cola $c \times$ tarea $t \times$ nat n	$\longrightarrow$ cola		$\{$esta?$(c,t) \wedge_{\text{L}}$ esta?$(n,$ ordenes $(c,t)\}$

generadores

vacia	:	$\longrightarrow$	cola
encolar	: cola $c \times$ tarea $t \times$ prioridad $p \times$ nat n	$\longrightarrow$	cola

otras operaciones

Esta?	: cola $c \times$ tarea t	$\longrightarrow$	bool
encolarSecuOrdenada	: cola $c \times$ secu<actividad $\times$ nat> s	$\longrightarrow$	cola

axiomas

$\forall\ i,s$: secu(α), $\forall\ n$:nat

vacia? (vacia) $\equiv$ true
vacia? (encolar (c,t,p,n)) $\equiv$ false

proxima (encolar (c,t,p,n)) $\equiv$ actividad (n,t)

prioridad (encolar (c,t,p,n)) $\equiv$ p)

ordenes (vacia) $\equiv$ <>
ordenes (encolar (c,t,p,n)) $\equiv$ ordenes (c) $\circ$ n

sacarOrden (encolar (c,t,p,n), $t1$, o) $\equiv$ **if** $t == t1 \wedge$ n $== o$ **then** c **else** encolar (ordenes (c), t, p, n)

Fin TAD