

Algoritmos y Estructura de Datos 2

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Exorcismo Extremo

TP1

Integrante	LU	Correo electrónico
Rosinov, Gaston Einan	37/18	<code>grosinov@gmail.com</code>
Schuster, Martin Ariel		
Panichelli, Manuel		

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. TAD JUEGO	3
2. TAD HABITACION	4
3. TAD ACCION	5
4. TAD DIRECCION	7
5. EXTENSIONES Y RENOMBRES	8

1. TAD JUEGO

TAD JUEGO

géneros juego

exporta TODO

usa HABITACION

igualdad observacional

$$(\forall j, j' : \text{juego}) \left(j =_{\text{obs}} j' \iff \left((n = 0? =_{\text{obs}} m = 0?) \wedge_L (\neg(n = 0?) \Rightarrow_L (\text{pred}(n) =_{\text{obs}} \text{pred}(m))) \right) \right)$$

igualdad observacional

$$(\forall j, j' : \text{juego}) \left(j =_{\text{obs}} j' \iff \left(\begin{array}{l} (\text{accionesPJs}(j) =_{\text{obs}} \text{accionesPJs}(j')) \wedge \\ (\text{accionesFan}(j) =_{\text{obs}} \text{accionesFan}(j')) \wedge \\ (\text{localizarJugadores}(j) =_{\text{obs}} \text{localizarJugadores}(j')) \wedge \\ (\text{hab}(j) =_{\text{obs}} \text{hab}(j')) \wedge \\ ((\forall p : \text{pj}) (\text{vivePJ?}(j, p) =_{\text{obs}} \text{vivePJ?}(j', p))) \wedge \\ ((\forall f : \text{fantasma}) ((\text{viveFan?}(j, p) =_{\text{obs}} \text{viveFan?}(j', p)) \wedge \\ (\text{ubicacionInicialFan}(j, f) =_{\text{obs}} \text{ubicacionInicialFan}(j', f)))) \end{array} \right) \right)$$

observadores básicos

accionesPJs	: juego	→	dicc(pj, secu(accion))	
accionesFan	: juego	→	dicc(pj, secu(accion))	
hab	: juego	→	hab	
vivePJ?	: juego $j \times \text{pj } p$	→	bool	{p ∈ jugadores(j)}
viveFan?	: juego $j \times \text{fantasma } f$	→	bool	{f ∈ fantasmas(j)}
ubicacionInicialFan	: juego $j \times \text{fantasma } f$	→	ubicacion	{f ∈ fantasmas(f)}
localizarJugadores	: juego	→	dicc(pj, ubicacion)	

generadores

iniciar	: conj(pj) $pjs \times \text{secu}(accion) \text{ as} \times \text{ubicacion } u \times \text{hab } h$	→	juego	{esConexa?(h) ∧ ¬∅?(as) ∧ ¬∅?(pjs) ∧ esValida?(h, pos(u))}
proxPaso	: juego $j \times \text{pj } p \times \text{accion } a$	→	juego	{p ∈ jugadores(j) ∧ _L vivePJ?(j, p) ∧ ¬termino?(j) ∧ ¬esMirar(a)}

otras operaciones

axiomas $\forall n, m : \text{nat}$
 $0 = 0? \quad \equiv \text{true}$

Fin TAD

2. TAD HABITACION

TAD HABITACION

géneros hab

exporta hab, observadores, generadores, esConexa?

usa POSICION, BOOL, NAT

igualdad observacional

$$(\forall h, h' : \text{hab}) \left(h =_{\text{obs}} h' \iff \left((\forall p : \text{posicion}) (\text{esValida?}(p, h) =_{\text{obs}} \text{esValida?}(p, h') \wedge_{\text{L}}) \right) \right)$$

observadores básicos

esValida? : hab × posicion → bool

estaOcupada? : hab h × posicion p → bool {esValida?(h, p)}

generadores

nueva : nat n → hab {n>1}

ocupar : hab h × posicion p → hab {esValida?(h, p) ∧_L ¬ estaOcupada?(h, p)}

otras operaciones

esConexa? : hab → bool

tamano : hab → nat

posiciones : hab → conj(posicion)

posicionesLibres : hab h × conj(posicion) ps → conj(posicion) {ps ⊆ posiciones(h)}

verificarAlcance : hab h × conj(posicion) ps → bool {ps ⊆ posiciones(h)}

verificarAlcancePos : hab h × conj(posicion) ps × posicion p → bool {ps ⊆ posiciones(h) ∧ p ∈ posiciones(h)}

axiomas ∀ h: hab ∀ ps: conj(posicion) ∀ p: posicion ∀ n, k, tam: nat

esValida?(nueva(n), p) ≡ 0 ≤ Π₁(p) < n ∧ 0 ≤ Π₂(p) < n

esValida?(ocupar(h, p'), p) ≡ p = p' ∨_L esValida?(h, p)

estaOcupada?(nueva(n), p) ≡ false

estaOcupada?(ocupar(h, p'), p) ≡ p = p' ∨ estaOcupada?(h, p)

tamano(nueva(n)) ≡ n

tamano(ocupar(h, p)) ≡ tamano(h)

esConexa?(h) ≡ verificarAlcance(h, posicionesLibres(posiciones(h)))

posicionesLibres(h, ps) ≡ **if** ∅?(ps) **then** ∅ **else** (if estaOcupada?(h, dameUno(ps)) **then** ∅ **else** {dameUno(ps)} **fi**) ∪ posicionesLibres(h, sinUno(ps)) **fi**

verificarAlcance(h, ps) ≡ **if** ∅?(ps) **then** true **else** verificarAlcancePos(h, ps, dameUno(ps)) ∧ verificarAlcance(h, p) **fi**

```

verificarAlcancePos(h, ps, p)  ≡ if  $\emptyset?(ps)$ 
                                then true
                                else
                                  esAlcancePos(h, p, dameUno(ps))  $\wedge$  verificarAlcancePos(h, p, sinUno(ps))
                                fi
posiciones(h)                  ≡ darPosiciones(h, tamano(h) - 1, tamano(h) - 1, tamano(h) - 1)
darPosiciones(h, n, k, tam)    ≡ if  $n = 0? \wedge k = 0?$ 
                                then  $\emptyset$ 
                                else if  $k = 0?$ 
                                  then Ag((n,k), darPosiciones(h, n - 1, tam, tam))
                                  else Ag((n,k), darPosiciones(h, n, k - 1, tam))
                                fi
                                fi

```

Fin TAD

3. TAD ACCION

TAD ACCION

géneros accion

exporta observadores, generadores, genero, otras operaciones

igualdad observacional

$$(\forall a, a' : \text{accion}) \left(a =_{\text{obs}} a' \iff \left(\begin{array}{l} \text{esNada}(a) =_{\text{obs}} \text{esNada}(a') \wedge \\ \text{esDisparar}(a) =_{\text{obs}} \text{esDisparar}(a') \wedge \\ \text{esMover}(a) =_{\text{obs}} \text{esMover}(a') \wedge \\ \text{esMirar}(a) =_{\text{obs}} \text{esMirar}(a') \wedge \\ ((\text{esMover}(a) \vee \text{esMirar}(a)) \Rightarrow_{\text{L}} \text{direccion}(a) =_{\text{obs}} \text{direccion}(a')) \end{array} \right) \right)$$

secu(accion)

observadores básicos

esMover	: accion	→ bool
esMirar	: accion	→ bool
esDisparar	: accion	→ bool
esNada	: accion	→ bool
direccion	: accion a	→ direccion {esMirar(a) \vee esMover(a)}

generadores

mover	: direccion	→ accion
mirar	: direccion	→ accion
disparar	:	→ accion
nada	:	→ accion

otras operaciones

ubicacionLuegoDe	: accion a \times hab h \times ubicacion u	→ conj(pos) {esValida?(h, pos(u))}
posicionesAfectadasPor	: accion a \times hab h \times ubicacion u	→ conj(pos) {esValida?(h, pos(u))}
$\neg \bullet$: accion	→ accion

invertir : hab $h \times$ ubicacion $u \times$ secu(accion) \longrightarrow secu(accion) {esValida?(h, pos(u))}

axiomas $\forall n, m: \text{nat}, \forall u: \text{ubicacion}, \forall a: \text{habitacion}$

posicionesAfectadasPor(mover(d), h, u) $\equiv \emptyset$

posicionesAfectadasPor(mirar(d), h, u) $\equiv \emptyset$

posicionesAfectadasPor(nada, h, u) $\equiv \emptyset$

posicionesAfectadasPor(disparar, h, u) \equiv **if** esValida?(h, proxPosEnDir(dir(u), pos(u)) \wedge_L \neg estaOcupada?(h, proxPosEnDir(dir(u), pos(u)))
then
Ag(proxPosEnDir(dir(u), pos(u)), posicionesAfectadasPor(disparar, h, \langle proxPosEnDir(dir(u), pos(u)), dir(u)))
else \emptyset
fi

invertir(h, u, as) \equiv **if** vacia?(as)
then $\langle \rangle$
else
invertir(h, ubicacionLuegoDe(prim(as), h, u), fin(as)) \vee \neg (prim(as), h, u)
fi

\neg (mover(d), h, u) \equiv **if** pos(ubicacionLuegoDe(mover(d), h, u)) = pos(u)
then mirar(opuesta(d))
else mover(opuesta(d))
fi

\neg (mirar(d), h, u) \equiv mirar(opuesta(d))

\neg (disparar, h, u) \equiv disparar

\neg (nada, h, u) \equiv nada

ubicacionLuegoDe(nada, h, u) \equiv u

ubicacionLuegoDe(disparar, h, u) \equiv u

ubicacionLuegoDe(mirar(d), h, u) $\equiv \langle \text{pos}(u), d \rangle$

ubicacionLuegoDe(mover(d), h, u) $\equiv \langle$ (**if** esValida?(h, proxPosEnDir(d, pos(u))) \wedge_L \neg estaOcupada?(h, proxPosEnDir(d, pos(u)))
then proxPosEnDir(d, pos(u))
else pos(u)
fi), d \rangle

esMirar(mirar(d)) \equiv true

esMirar(mover(d)) \equiv false

esMirar(disparar) \equiv false

esMirar(nada) \equiv false

esMover(mirar(d)) \equiv false

esMover(mover(d)) \equiv true

esMover(disparar) \equiv false

esMover(nada) \equiv false

esDisparar(mirar(d)) \equiv false

esDisparar(mover(d)) \equiv false

esDisparar(disparar) \equiv true

esDisparar(nada) \equiv false

esNada(mirar(d)) \equiv false

esNada(mover(d)) \equiv false

esNada(disparar)	\equiv false
esNada(nada)	\equiv true
direccion(mirar(d))	\equiv d
direccion(mover(d))	\equiv d

Fin TAD

4. TAD DIRECCION

TAD DIRECCION**géneros** direccion**exporta** observadores, generadores, otras operaciones**igualdad observacional**

$$(\forall d, d' : \text{direccion}) \left(d =_{\text{obs}} d' \iff \left(\begin{array}{l} \text{esArriba}(d) =_{\text{obs}} \text{esArriba}(d') \wedge \\ \text{esAbajo}(d) =_{\text{obs}} \text{esAbajo}(d') \wedge \\ \text{esIzquierda}(d) =_{\text{obs}} \text{esIzquierda}(d') \wedge \\ \text{esDerecha}(d) =_{\text{obs}} \text{esDerecha}(d') \end{array} \right) \right)$$

observadores básicos

esArriba	: direccion	\longrightarrow bool
esAbajo	: direccion	\longrightarrow bool
esIzquierda	: direccion	\longrightarrow bool
esDerecha	: direccion	\longrightarrow bool

generadores

arriba	:	\longrightarrow direccion
abajo	:	\longrightarrow direccion
izquierda	:	\longrightarrow direccion
derecha	:	\longrightarrow direccion

otras operaciones

opuesta	: direccion	\longrightarrow direccion
proxPosEnDir	: direccion \times posicion	\longrightarrow posicion

axiomas

opuesta(arriba)	\equiv abajo
opuesta(abajo)	\equiv arriba
opuesta(izquierda)	\equiv derecha
opuesta(derecha)	\equiv izquierda
proxPosEnDir(arriba, p)	$\equiv \langle \Pi_1(p), \Pi_2(p) + 1 \rangle$
proxPosEnDir(abajo, p)	$\equiv \langle \Pi_1(p), \Pi_2(p) - 1 \rangle$
proxPosEnDir(izquierda, p)	$\equiv \langle \Pi_1(p) - 1, \Pi_2(p) \rangle$
proxPosEnDir(derecha, p)	$\equiv \langle \Pi_1(p) + 1, \Pi_2(p) \rangle$
esArriba(arriba)	\equiv true
esArriba(abajo)	\equiv false

esArriba(izquierda)	\equiv false
esArriba(derecha)	\equiv false
esAbajo(arriba)	\equiv false
esAbajo(abajo)	\equiv true
esAbajo(izquierda)	\equiv false
esAbajo(derecha)	\equiv false
esIzquierda(arriba)	\equiv false
esIzquierda(abajo)	\equiv false
esIzquierda(izquierda)	\equiv true
esIzquierda(derecha)	\equiv false
esDerecha(arriba)	\equiv false
esDerecha(abajo)	\equiv false
esDerecha(izquierda)	\equiv false
esDerecha(derecha)	\equiv true

Fin TAD

5. EXTENSIONES Y RENOMBRES

TAD FANTASMA ES NAT**TAD PJ ES NAT****TAD POSICION ES TUPLA(NAT, NAT)****TAD NAT extiende NAT****otras operaciones** $\bullet \% \bullet : \text{nat} \times \text{nat} \longrightarrow \text{nat}$ **axiomas** $\forall n, m: \text{nat}$ $n \% m \equiv \text{if } n < m \text{ then } n \text{ else } (n - m) \% m \text{ fi}$ **Fin TAD****TAD UBICACION extiende TUPLA(POSICION, DIRECCION)****otras operaciones** $\text{pos} : \text{ubicacion} \longrightarrow \text{posicion}$ $\text{dir} : \text{ubicacion} \longrightarrow \text{direccion}$ **axiomas** $\forall u: \text{ubicacion}$ $\text{pos}(u) \equiv \Pi_1(u)$ $\text{dir}(u) \equiv \Pi_2(u)$ **Fin TAD****TAD SECUENCIA extiende SECUENCIA****otras operaciones**

$\bullet[\bullet] : \text{secu}(\alpha) \times \text{nat } i \longrightarrow \alpha$ $\{i < \text{long}(s)\}$

axiomas $\quad \forall s: \text{secu}(\alpha), \forall i: \text{nat}$

$s[i] \equiv \text{if } i = 0? \text{ then } \text{prim}(s) \text{ else } \text{fin}(s)[i - 1] \text{ fi}$

Fin TAD