

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	grosinov@gmail.com
Schuster, Martin Ariel	208/18	m.a.schuster98@gmail.com
Panichelli, Manuel	72/18	panicmanu@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. EXTENSIONES Y RENOMBRES	3
2. TAD JUEGO	4
3. TAD HABITACION	7
4. TAD ACCION	9
5. TAD DIRECCION	11

1. 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

direccion

$\text{pos} : \text{ubicacion} \longrightarrow \text{posicion}$

$\text{dir} : \text{ubicacion} \longrightarrow \text{direccion}$

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

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

Fin TAD

2. TAD JUEGO

TAD JUEGO

géneros	juego
exporta	juego, observadores, generadores, puntaje, ronda
usa	SECUENCIA, NAT, CONJUNTO, HABITACION, UBICACION, PJ, ACCION, DIRECCION, FANTASMA, PJ, DICCIONARIO, BOOL

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 $\{p \in \text{jugadores}(j)\}$
viveFan?	: juego $j \times \text{fantasma } f$	→	bool $\{f \in \text{fantasmas}(j)\}$
ubicacionInicialFan	: juego $j \times \text{fantasma } f$	→	ubicacion $\{f \in \text{fantasmas}(f)\}$
localizarJugadores	: juego	→	dicc(pj, ubicacion)

generadores

iniciar	: conj(pj) $pjs \times \text{secu}(accion) as \times \text{ubicacion } u$	→	juego $\{ \text{esConexa?}(h) \wedge \neg \emptyset?(as) \wedge \neg \emptyset?(pjs) \wedge \text{esValida?}(h, \text{pos}(u)) \}$
proxPaso	: juego $j \times \text{pj } p \times \text{accion } a$	→	juego $\{ p \in \text{jugadores}(j) \wedge \text{vivePJ?}(j, p) \wedge \neg \text{termino?}(j) \wedge \neg \text{esMirar}(a) \}$

otras operaciones

jugadores	: juego	→	conj(pj)
fantasmas	: juego	→	conj(fantasma)
nombreSiguienteFan	: juego	→	fantasma
puntaje	: juego	→	nat
ronda	: juego	→	nat
paso	: juego	→	nat
cantAcciones	: juego $\times \text{conj}(pj)$	→	nat
terminaRonda	: juego $j \times \text{pj } p \times \text{accion}$	→	bool $\{p \in \text{jugadores}(j)\}$
fantasmaEspecial	: juego	→	fantasma
termino?	: juego	→	bool
estanVivos	: juego $\times \text{conj}(pj) pjs$	→	bool $\{pjs \subseteq \text{jugadores}(j)\}$

ubicacionInicialPJ	: juego $j \times$ pj p	\longrightarrow ubicacion	$\{p \in \text{jugadores}(j)\}$
ubicacionPJ	: juego $j \times$ pj p	\longrightarrow ubicacion	$\{p \in \text{jugadores}(j)\}$
ubicacionFan	: juego $j \times$ fantamsa f	\longrightarrow ubicacion	$\{f \in \text{fantasmas}(j)\}$
deducirUbicacion	: juego $j \times$ ubicacion $u \times$ acciones	\longrightarrow ubicacion	$\{\text{esValida?}(\text{hab}(j), \text{pos}(u))\}$
moriraFantasma	: juego $j \times$ pj $p \times$ accion \times fantasma f	\longrightarrow bool	$\{p \in \text{jugadores}(j) \wedge$ $f \in \text{fantasmas}(j) \wedge_L$ $\text{viveFan?}(j, f) \wedge$ $\text{vivePJ?}(j, p)\}$
moriraPJ	: juego $j \times$ conj(fantasma) $fs \times$ pj $p \times$ accion	\longrightarrow bool	$\{p \in \text{jugadores}(j) \wedge$ $fs \subseteq \text{fantasmas}(j) \wedge_L$ $\text{vivePJ?}(j, p)\}$
moriraPJPorFan	: juego $j \times$ fantasma $f \times$ pj $p \times$ accion	\longrightarrow bool	$\{p \in \text{jugadores}(j) \wedge$ $f \in \text{fantasmas}(j) \wedge_L$ $\text{vivePJ?}(j, p) \wedge$ $\text{viveFan?}(j, f)\}$
accionFan	: juego $j \times$ fantasma f	\longrightarrow accion	$\{f \in \text{fantasmas}(j) \wedge_L$ $\text{viveFan?}(j, f)\}$
inicializarAcciones	: conj(pj)	\longrightarrow dicc(pj, secu(accion))	
agregarFantasma	: juego $j \times$ ubicacion $u \times$ dicc(fantasma \times secu(accion)) \times fantasma \times secu(accion)	\longrightarrow dicc(fantasma, secu(accion))	$\{\text{esValida?}(\text{hab}(j), \text{pos}(u))\}$
generarAccionesFan	: juego $j \times$ ubicacion $u \times$ secu(accion)	\longrightarrow secu(accion)	$\{\text{esValida?}(\text{hab}(j), \text{pos}(u))\}$
axiomas	$\forall p: \text{pj},$ $\forall pjs: \text{conj}(\text{pj}),$ $\forall f: \text{fantasma},$ $\forall fs: \text{conj}(\text{fantasma}),$ $\forall j: \text{juego},$ $\forall h: \text{hab},$ $\forall u, uInicialPJ: \text{ubicacion},$ $\forall a: \text{accion},$ $\forall as: \text{secu}(\text{accion})$		
accionesPJs(iniciar(pjs, as, u, h))	\equiv inicializarAcciones(pjs)		
accionesPJs(proxPaso(j, p, a))	\equiv if \neg terminaRonda(j, p, a) then definir(p, obtener(p, accionesPJs(j)) \circ a, accionesPJs(j)) else inicializarAcciones(jugadores(j)) fi		
accionesFan(iniciar(pjs, as, u, h))	\equiv definir(nombreSiguienteFan(j), as, vacio)		
accionesFan(proxPaso(j, p, a))	\equiv if \neg terminaRonda(j, p, a) then accionesFan(j) else agregarFantasma(j, ubicacionInicialPJ(j, p), accionesFan(j), nombreSiguienteFan(j), obtener(p, accionesPJs(j)) \circ a) fi		
hab(iniciar(pjs, as, u, h))	\equiv h		
hab(proxPaso(j, p, a))	\equiv hab(j)		
vivePJ?(iniciar(pjs, as, u, h), p')	\equiv true		

vivePJ?(proxPaso(j, p, a), p')	≡ terminaRonda?(j, p, a) ∨ if p = p' then ¬ moriraPJ(j, fantasmas(j), p, a) else vivePJ?(j, p') ∧ _L ¬ moriraPJ(j, fantasmas(j), p, a) fi
viveFan?(iniciar(pjs, as, u, h), f)	≡ true
viveFan?(proxPaso(j, p, a), f)	≡ terminaRonda?(j, p, a) ∨ (viveFan?(j, f) ∧ _L ¬ moriraFantasma(j, p, a, f))
ubicacionInicialFan(iniciar(pjs, as, u, h))	≡ u
ubicacionInicialFan(proxPaso(j, p, a))	≡ if f ∈ fantasmas(j) then ubicacionInicialFan(j, f) else ubicacionInicialPJ(j, p) fi
jugadores(j)	≡ claves(accionesPJs(j))
fantasmas(j)	≡ claves(accionesFan(j))
nombreSiguienteFan(j)	≡ #(claves(accionesFan(j))) + 1
puntaje(j)	≡ ronda(j) - 1
ronda(j)	≡ #(fantasmas(j))
paso(j)	≡ cantAcciones(j, jugadores(j))
cantAcciones(j, pjs)	≡ if ∅?(pjs) then 0 else long(obtener(dameUno(pjs), accionesPJs(j))) + cantAcciones(j, sinUno(pjs)) fi
termino?(j)	≡ ¬ estanVivos(j, jugadores(j))
estanVivos(j, pjs)	≡ if ∅?(pjs) then true else vivePJ?(j, dameUno(pjs)) ∧ estanVivos(j, sinUno(pjs)) fi
fantasmaEspecial(j)	≡ #(claves(accionesFan(j)))
ubicacionInicialPJ(j, p)	≡ obtener(p, localizarJugadores(j))
ubicacionPJ(j, p)	≡ deducirUbicacion(j, ubicacionInicialPJ(j, p), obtener(p, accionesPJs(j)))
ubicacionFan(j, f)	≡ deducirUbicacion(j, ubicacionInicialFan(j, f), obtener(f, accionesFan(j)))
deducirUbicacion(j, u, as)	≡ if vacia?(as) then u else deducirUbicacion(j, ubicacionLuegoDe(prim(as), hab(j), u), fin(as)) fi
agregarFantasma(j, uInicialPJ, accionesFantasmas, f, as)	≡ definir(f, generarAccionesFantasma(j, uInicialPJ, as), accionesFantasmas)
generarAccionesFan(j, uInicialPJ, as)	≡ as & (nada • nada • nada • nada • nada) & invertir(hab(j), uInicialPJ, as)
inicializarAcciones(pjs)	≡ if ∅?(pjs) then vacio else definir(dameUno(pjs), <>, inicializarAcciones(sinUno(pjs))) fi

terminaRonda(j, p, a)	\equiv moriraFantasma(j, p, a, fantasmaEspecial(j))
moriraFantasma(j, p, a, f)	\equiv pos(ubicacionFan(j, f)) \in posicionesAfectadasPor(a, hab(j), ubicacionPJ(j, p))
moriraPJ(j, fs, p, a)	\equiv if $\emptyset?(fs)$ then false else (viveFan?(j, dameUno(fs)) \wedge_L moriraPJPorFan(j, dameUno(fs), p, a)) \vee_L moriraPJ(j, sinUno(fs), p, a) fi
moriraPJPorFan(j, f, p, a)	\equiv \neg moriraFantasma(j, p, a, f) \wedge (pos(ubicacionLuegoDe(a, hab(j), ubicacionPJ(j, p))) \in posicionesAfectadasPor(accionFan(j, f), hab(j), ubicacionFan(j, f)))
accionFan(j, f)	\equiv obtener(accionesFan(j), f)[paso(j) % obtener(accionesFan(j), f)]

Fin TAD

3. TAD HABITACION

TAD HABITACION

géneros hab**exporta** hab, observadores, generadores, esConexa?**usa** POSICION, BOOL, NAT**igualdad observacional**

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

$$\left(\text{esValida?}(p, h) \Rightarrow_L \text{estaOcupada?}(p, h) =_{\text{obs}} \text{estaOcupada?}(p, h') \right) \right)$$

observadores básicos

esValida?	: hab \times posicion	\longrightarrow bool	
estaOcupada?	: hab $h \times$ posicion p	\longrightarrow bool	$\{\text{esValida?}(h, p)\}$

generadores

nueva	: nat n	\longrightarrow hab	$\{n > 1\}$
ocupar	: hab $h \times$ posicion p	\longrightarrow hab	$\{\text{esValida?}(h, p) \wedge_L \neg \text{estaOcupada?}(h, p)\}$

otras operaciones

esConexa?	: hab	\longrightarrow bool	
tamano	: hab	\longrightarrow nat	
posiciones	: hab	\longrightarrow conj(posicion)	
posicionesLibres	: hab $h \times$ conj(posicion) ps	\longrightarrow conj(posicion)	$\{ps \subseteq \text{posiciones}(h)\}$
verificarAlcance	: hab $h \times$ conj(posicion) ps	\longrightarrow bool	$\{ps \subseteq \text{posiciones}(h)\}$
verificarAlcancePos	: hab $h \times$ conj(posicion) $ps \times$ posicion p	\longrightarrow bool	$\{ps \subseteq \text{posiciones}(h) \wedge p \in \text{posiciones}(h)\}$

axiomas $\forall h : \text{hab} \forall ps : \text{conj}(\text{posicion}) \forall p : \text{posicion} \forall n, k, tam : \text{nat}$

$$\text{esValida?}(nueva(n), p) \equiv 0 \leq \Pi_1(p) < n \wedge 0 \leq \Pi_2(p) < n$$

```

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  $\emptyset?(ps)$ 
                                then  $\emptyset$ 
                                else
                                (if estaOcupada?(h, dameUno(ps)) then  $\emptyset$  else {dameUno(ps)} fi)
                                ∪ posicionesLibres(h, sinUno(ps))
                                fi
verificarAlcance(h, ps)       ≡ if  $\emptyset?(ps)$ 
                                then true
                                else
                                verificarAlcancePos(h, ps, dameUno(ps)) ∧ verificarAlcance(h, p)
                                fi
verificarAlcancePos(h, ps, p) ≡ if  $\emptyset?(ps)$ 
                                then true
                                else
                                esAlcanzable(h, p, dameUno(ps)) ∧ 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

4. TAD ACCION

TAD ACCION

géneros accion

exporta accion, observadores, generadores, genero, otras operaciones

usa DIRECCION, POSICION, UBICACION, BOOL, CONJUNTO, HABITACION, SECUENCIA

igualdad observacional

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

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

axiomas $\forall d$: direccion $\forall u$: ubicacion, $\forall a$: habitacion, $\forall as$: secu(accion)

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) \rangle) else \emptyset fi

<code>invertir(h, u, as)</code>	\equiv if <code>vacía?(as)</code> then <code><></code> else <code>invertir(h, ubicacionLuegoDe(prim(as), h, u), fin(as))</code> \circ $\neg(\text{prim}(as), h, u)$ fi
$\neg(\text{mover}(d), h, u)$	\equiv if <code>pos(ubicacionLuegoDe(mover(d), h, u)) = pos(u)</code> then <code>mirar(opuesta(d))</code> else <code>mover(opuesta(d))</code> fi
$\neg(\text{mirar}(d), h, u)$	\equiv <code>mirar(opuesta(d))</code>
$\neg(\text{disparar}, h, u)$	\equiv <code>disparar</code>
$\neg(\text{nada}, h, u)$	\equiv <code>nada</code>
<code>ubicacionLuegoDe(nada, h, u)</code>	\equiv <code>u</code>
<code>ubicacionLuegoDe(disparar, h, u)</code>	\equiv <code>u</code>
<code>ubicacionLuegoDe(mirar(d), h, u)</code>	\equiv <code>< pos(u), d ></code>
<code>ubicacionLuegoDe(mover(d), h, u)</code>	\equiv <code>< (if esValida?(h, proxPosEnDir(d, pos(u))) \wedge_L \negestaOcupada?(h, proxPosEnDir(d, pos(u))) then proxPosEnDir(d, pos(u)) else pos(u) fi), d ></code>
<code>esMirar(mirar(d))</code>	\equiv true
<code>esMirar(mover(d))</code>	\equiv false
<code>esMirar(disparar)</code>	\equiv false
<code>esMirar(nada)</code>	\equiv false
<code>esMover(mirar(d))</code>	\equiv false
<code>esMover(mover(d))</code>	\equiv true
<code>esMover(disparar)</code>	\equiv false
<code>esMover(nada)</code>	\equiv false
<code>esDisparar(mirar(d))</code>	\equiv false
<code>esDisparar(mover(d))</code>	\equiv false
<code>esDisparar(disparar)</code>	\equiv true
<code>esDisparar(nada)</code>	\equiv false
<code>esNada(mirar(d))</code>	\equiv false
<code>esNada(mover(d))</code>	\equiv false
<code>esNada(disparar)</code>	\equiv false
<code>esNada(nada)</code>	\equiv true
<code>direccion(mirar(d))</code>	\equiv <code>d</code>
<code>direccion(mover(d))</code>	\equiv <code>d</code>

Fin TAD

5. TAD DIRECCION

TAD DIRECCION

géneros direccion

exporta direccion, observadores, generadores, otras operaciones

usa BOOL, POSICION, NAT

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	→ bool
esAbajo	: direccion	→ bool
esIzquierda	: direccion	→ bool
esDerecha	: direccion	→ bool

generadores

arriba	:	→ direccion
abajo	:	→ direccion
izquierda	:	→ direccion
derecha	:	→ direccion

otras operaciones

opuesta	: direccion	→ direccion
proxPosEnDir	: direccion × posicion	→ posicion

axiomas $\forall p$: posicion

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

<code>esIzquierda(arriba)</code>	\equiv <code>false</code>
<code>esIzquierda(abajo)</code>	\equiv <code>false</code>
<code>esIzquierda(izquierda)</code>	\equiv <code>true</code>
<code>esIzquierda(derecha)</code>	\equiv <code>false</code>
<code>esDerecha(arriba)</code>	\equiv <code>false</code>
<code>esDerecha(abajo)</code>	\equiv <code>false</code>
<code>esDerecha(izquierda)</code>	\equiv <code>false</code>
<code>esDerecha(derecha)</code>	\equiv <code>true</code>

Fin TAD