

# Tipos abstractos de datos básicos

Algoritmos y Estructuras de Datos II, DC, UBA.

## Índice

1. TAD PJ	2
2. TAD FANTASMA	2
3. TAD JUEGO	2
4. TAD ACCION	3
5. TAD DIRECCION	5

## 1. TAD PJ

## 2. TAD FANTASMA

## 3. 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) $pjs \times \text{secu}(accion) \text{ as} \rightarrow \text{juego}$ $\times \text{ubicacion } u \times \text{hab } h$			{esConexa?(h) ∧ ¬∅?(as) ∧ ¬∅?(pjs) ∧ esValida?(h, pos(u))}
proxPaso	: juego $j \times \text{pj } p \times \text{accion } a$	→	juego	{p ∈ jugadores(j) ∧ <sub>L</sub> vivePJ?(j, p) ∧ ¬termino?(j) ∧ ¬esMirar(a)}

**otras operaciones**

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

**Fin TAD**

## 4. TAD ACCION

### TAD ACCION

**géneros** accion

**exporta** observadores, generadores, genero, otras operaciones

**igualdad observacional**

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

secu(accion)

**observadores básicos**

esMover	: accion	→ bool
esMirar	: accion	→ bool
esDisparar	: accion	→ bool
esNada	: accion	→ bool
direccion	: accion $a$	→ direccion {esMirar(a) ∨ esMover(a)}

**generadores**

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

**otras operaciones**

ubicacionLuegoDe	: accion $a \times \text{hab } h \times \text{ubicacion } u$	→ conj(pos)	{esValida?(h, pos(u))}
posicionesAfectadasPor	: accion $a \times \text{hab } h \times \text{ubicacion } u$	→ conj(pos)	{esValida?(h, pos(u))}
$\neg \bullet$	: accion	→ accion	
invertir	: hab $h \times \text{ubicacion } u \times \text{secu(accion)}$	→ 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)	$\equiv \text{if esValida?}(h, \text{proxPosEnDir}(\text{dir}(u), \text{pos}(u)) \wedge_{\text{L}} \neg \text{estaOcupada?}(h, \text{proxPosEnDir}(\text{dir}(u), \text{pos}(u)))$ <b>then</b> $\text{Ag}(\text{proxPosEnDir}(\text{dir}(u), \text{pos}(u)),$ $\text{posicionesAfectadasPor}(\text{disparar}, h,$ $\langle \text{proxPosEnDir}(\text{dir}(u), \text{pos}(u)), \text{dir}(u) \rangle)$ <b>else</b> $\emptyset$ <b>fi</b>

invertir(h, u, as)	≡ <b>if</b> vacia?(as) <b>then</b> <> <b>else</b> invertir(h, ubicacionLuegoDe(prim(as), h, u), fin(as)) ∨ ¬(prim(as), h, u) <b>fi</b>
¬(mover(d), h, u)	≡ <b>if</b> pos(ubicacionLuegoDe(mover(d), h, u)) = pos(u) <b>then</b> mirar(opuesta(d)) <b>else</b> mover(opuesta(d)) <b>fi</b>
¬(mirar(d), h, u)	≡ mirar(opuesta(d))
¬(disparar, h, u)	≡ disparar
¬(nada, h, u)	≡ nada
ubicacionLuegoDe(nada, h, u)	≡ u
ubicacionLuegoDe(disparar, h, u)	≡ u
ubicacionLuegoDe(mirar(d), h, u)	≡ ⟨pos(u), d⟩
ubicacionLuegoDe(mover(d), h, u)	≡ ⟨(if esValida?(h, proxPosEnDir(d, pos(u))) ∧ ¬estaOcupada?(h, proxPosEnDir(d, pos(u))) <b>then</b> proxPosEnDir(d, pos(u)) <b>else</b> pos(u) <b>fi</b> ), d⟩
esMirar(mirar(d))	≡ true
esMirar(mover(d))	≡ false
esMirar(disparar)	≡ false
esMirar(nada)	≡ false
esMover(mirar(d))	≡ false
esMover(mover(d))	≡ true
esMover(disparar)	≡ false
esMover(nada)	≡ false
esDisparar(mirar(d))	≡ false
esDisparar(mover(d))	≡ false
esDisparar(disparar)	≡ true
esDisparar(nada)	≡ false
esNada(mirar(d))	≡ false
esNada(mover(d))	≡ false
esNada(disparar)	≡ false
esNada(nada)	≡ true
direccion(mirar(d))	≡ d
direccion(mover(d))	≡ d

**Fin TAD**

## 5. TAD DIRECCION

### TAD DIRECCION

**géneros**      direccion

**exporta**      observadores, generadores, otras operaciones

**igualdad observacional**

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

**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**

opuesta(arriba)	≡ abajo
opuesta(abajo)	≡ arriba
opuesta(izquierda)	≡ derecha
opuesta(derecha)	≡ izquierda
proxPosEnDir(arriba, p)	≡ ⟨Π <sub>1</sub> (p), Π <sub>2</sub> (p) + 1⟩
proxPosEnDir(abajo, p)	≡ ⟨Π <sub>1</sub> (p), Π <sub>2</sub> (p) - 1⟩
proxPosEnDir(izquierda, p)	≡ ⟨Π <sub>1</sub> (p) - 1, Π <sub>2</sub> (p)⟩
proxPosEnDir(derecha, p)	≡ ⟨Π <sub>1</sub> (p) + 1, Π <sub>2</sub> (p)⟩
esArriba(arriba)	≡ true
esArriba(abajo)	≡ false
esArriba(izquierda)	≡ false
esArriba(derecha)	≡ false
esAbajo(arriba)	≡ false
esAbajo(abajo)	≡ true
esAbajo(izquierda)	≡ false
esAbajo(derecha)	≡ false
esIzquierda(arriba)	≡ false

esIzquierda(abajo)	≡ false
esIzquierda(izquierda)	≡ true
esIzquierda(derecha)	≡ false
esDerecha(arriba)	≡ false
esDerecha(abajo)	≡ false
esDerecha(izquierda)	≡ false
esDerecha(derecha)	≡ true

**Fin TAD**