

Algoritmos y Estructuras de Datos II

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

Trabajo Práctico 2

Grupo Omega

Integrante	LU	Correo electrónico
Candia, Matias	721/19	candia.matias2000@gmail.com
Caneva, Diego Gabriel	169/18	diego.g.caneva@gmail.com
Lin Zabala, Juan Ignacio	349/18	juanignacio.lin@gmail.com
Sarmiento, Matias Federico	741/18	matiasfsarmiento@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

1. TAD Casilla

TAD Casilla ES Tupla(Nat,Nat)

2. TAD Tablero

TAD Tablero

géneros tablero

usa Casilla

igualdad observacional

$$(\forall t1, t2 : \text{tablero}) \left(t1 =_{\text{obs}} t2 \iff \left(\begin{array}{l} \text{dimensiones}(t1) =_{\text{obs}} \text{dimensiones}(t2) \wedge \\ \text{salida}(t1) =_{\text{obs}} \text{salida}(t2) \wedge \\ \text{llegada}(t1) =_{\text{obs}} \text{llegada}(t2) \wedge_L \\ (\forall c:\text{casilla})(\text{enRango}(c, \text{dimensiones}(t1)) \\ \Rightarrow_L \text{hayFantasma}(t1, c) =_{\text{obs}} \\ \text{hayFantasma}(t2, c) \wedge \\ \text{hayPared}(t1, c) =_{\text{obs}} \text{hayPared}(t2, c)) \wedge \\ \text{hayChocolate}(t1, c) =_{\text{obs}} \text{hayChocolate}(t2, c) \end{array} \right) \right)$$

observadores básicos

dimensiones : tablero \longrightarrow tupla(nat,nat)

salida : tablero \longrightarrow casilla

llegada : tablero \longrightarrow casilla

hayFantasma : tablero $t \times$ casilla $c \longrightarrow$ bool {enRango(c,dimensiones(t))}

hayPared : tablero $t \times$ casilla $c \longrightarrow$ bool {enRango(c,dimensiones(t))}

hayChocolate : tablero $t \times$ casilla $c \longrightarrow$ bool {enRango(c,dimensiones(t))}

generadores

crearTablero : tupla(nat,nat) $d \times$ casilla $s \times$ casilla $ll \longrightarrow$ tablero {enRango(s,d) \wedge enRango(ll,d)}

agregarFantasma : tablero $t \times$ casilla $c \longrightarrow$ tablero {enRango(c,dimensiones(t)) \wedge \neg (salida(t)=c \vee llegada(t)=c) \wedge \neg (hayPared(t,c))}

agregarPared : tablero $t \times$ casilla $c \longrightarrow$ tablero {enRango(c,dimensiones(t)) \wedge \neg (salida(t)=c \vee llegada(t)=c) \wedge \neg (hayFantasma(t,c)) \wedge \neg (hayChocolate(t,c))}

agregarChocolate : tablero $t \times$ casilla $c \longrightarrow$ tablero {enRango(c,dimensiones(t)) \wedge_L \neg hayPared(t,c)}

otras operaciones

$\text{fantasmas} : \text{tablero} \longrightarrow \text{conj}(\text{casilla})$
 $\text{distancia} : \text{casilla} \times \text{casilla} \longrightarrow \text{nat}$
 $\text{casillaAsustada} : \text{casilla} \times \text{conj}(\text{casilla}) \longrightarrow \text{bool}$
 $\text{enRango} : \text{casilla} \times \text{tupla}(\text{nat}, \text{nat}) \longrightarrow \text{bool}$

axiomas

$\text{dimensiones}(\text{crearTablero}(d, s, ll)) \equiv d$
 $\text{dimensiones}(\text{agregarFantasma}(t, c)) \equiv \text{dimensiones}(t)$
 $\text{dimensiones}(\text{agregarPared}(t, c)) \equiv \text{dimensiones}(t)$
 $\text{dimensiones}(\text{agregarChocolate}(t, c)) \equiv \text{dimensiones}(t)$
 $\text{salida}(\text{crearTablero}(d, s, ll)) \equiv s$
 $\text{salida}(\text{agregarFantasma}(t, c)) \equiv \text{salida}(t)$
 $\text{salida}(\text{agregarPared}(t, c)) \equiv \text{salida}(t)$
 $\text{salida}(\text{agregarChocolate}(t, c)) \equiv \text{salida}(t)$
 $\text{llegada}(\text{crearTablero}(d, s, ll)) \equiv ll$
 $\text{llegada}(\text{agregarFantasma}(t, c)) \equiv \text{llegada}(t)$
 $\text{llegada}(\text{agregarPared}(t, c)) \equiv \text{llegada}(t)$
 $\text{llegada}(\text{agregarChocolate}(t, c)) \equiv \text{llegada}(t)$
 $\text{hayFantasma}(\text{crearTablero}(d, s, ll), c) \equiv \text{false}$
 $\text{hayFantasma}(\text{agregarFantasma}(t, k), c) \equiv \text{if } c=k \text{ then true else hayFantasma}(t, c) \text{ fi}$
 $\text{hayFantasma}(\text{agregarPared}(t, k), c) \equiv \text{hayFantasma}(t, c)$
 $\text{hayFantasma}(\text{agregarChocolate}(t, k), c) \equiv \text{hayFantasma}(t, c)$
 $\text{hayPared}(\text{crearTablero}(d, s, ll), c) \equiv \text{false}$
 $\text{hayPared}(\text{agregarFantasma}(t, k), c) \equiv \text{hayPared}(t, c)$
 $\text{hayPared}(\text{agregarPared}(t, k), c) \equiv \text{if } c=k \text{ then true else hayPared}(t, c) \text{ fi}$
 $\text{hayPared}(\text{agregarChocolate}(t, k), c) \equiv \text{hayPared}(t, c)$
 $\text{hayChocolate}(\text{crearTablero}(d, s, ll), c) \equiv \text{false}$
 $\text{hayChocolate}(\text{agregarFantasma}(t, k), c) \equiv \text{hayChocolate}(t, c)$
 $\text{hayChocolate}(\text{agregarPared}(t, k), c) \equiv \text{hayChocolate}(t, c)$
 $\text{hayChocolate}(\text{agregarChocolate}(t, k), c) \equiv \text{if } c=k \text{ then true else hayChocolate}(t, c) \text{ fi}$
 $\text{fantasmas}(\text{crearTablero}(d, s, ll)) \equiv \emptyset$
 $\text{fantasmas}(\text{agregarFantasma}(t, c)) \equiv \text{Ag}(c, \text{fantasmas}(t))$
 $\text{fantasmas}(\text{agregarPared}(t, c)) \equiv \text{fantasmas}(t)$
 $\text{fantasmas}(\text{agregarChocolate}(t, c)) \equiv \text{fantasmas}(t)$
 $\text{distancia}(c1, c2) \equiv |\pi_1(c1) - \pi_1(c2)| + |\pi_2(c1) - \pi_2(c2)|$
 $\text{casillaAsustada}(c, f) \equiv \text{if } \emptyset ?(f) \text{ then}$
 $\quad \text{false}$
 $\quad \text{else}$
 $\quad \quad \text{if distancia}(c, \text{dameUno}(f)) < 3 \text{ then}$
 $\quad \quad \quad \text{true}$
 $\quad \quad \text{else}$
 $\quad \quad \quad \text{casillaAsustada}(c, \text{sinUno}(f))$
 $\quad \quad \text{fi}$
 fi
 $\text{enRango}(x, y) \equiv \pi_1(y) > \pi_1(x) \wedge \pi_2(y) > \pi_2(x)$

Fin TAD

3. TAD Pacalgo

TAD Pacalgo

géneros pacalgo

usa TABLERO

igualdad observacional

$$(\forall p1, p2 : \text{pacalgo}) \quad p1 =_{\text{obs}} p2 \iff \left(\begin{array}{l} \text{jugador}(p1) =_{\text{obs}} \text{jugador}(p2) \wedge \\ \text{puntaje}(p1) =_{\text{obs}} \text{puntaje}(p2) \wedge \\ \text{inmunidad}(p1) =_{\text{obs}} \text{inmunidad}(p2) \wedge \\ \text{mostrarTablero}(p1) =_{\text{obs}} \\ \text{mostrarTablero}(p2) \wedge_{\text{L}} \\ (\forall c : \text{casilla}) \\ ((\text{enRango}(c, \text{dimensiones}(\\ \text{mostrarTablero}(p1))) \wedge_{\text{L}} \\ \text{hayChocolate}(\text{mostrarTablero}(p1), c)) \Rightarrow_{\text{L}} \\ \text{chocolateComido}(p1, c) =_{\text{obs}} \\ \text{chocolateComido}(p2, c)) \end{array} \right)$$

observadores básicos

jugador : pacalgo \longrightarrow casilla

mostrarTablero : pacalgo \longrightarrow tablero

puntaje : pacalgo \longrightarrow nat

inmunidad : pacalgo \longrightarrow nat

chocolateComido : pacalgo $p \times$ casilla $c \longrightarrow$ bool

$\{\text{enRango}(c, \text{dimensiones}(\text{mostrarTablero}(p))) \wedge_{\text{L}} \text{hayChocolate}(\text{mostrarTablero}(p), c)\}$

generadores

crearPacalgo : tablero \longrightarrow pacalgo

moverJugador : pacalgo $p \times$ casilla $d \longrightarrow$ pacalgo

$\{\neg (\text{perdió?}(p)) \wedge \neg (\text{ganó?}(p)) \wedge \text{esAledaña}(p, d)\}$

otras operaciones

ganó? : pacalgo \longrightarrow bool

perdió? : pacalgo \longrightarrow bool

esAledaña? : pacalgo \times casilla \longrightarrow bool

axiomas

```

mostrarTablero(crearPacalgo(t))  $\equiv$  t
mostrarTablero(moverJugador(p,d))  $\equiv$  mostrarTablero(p)
jugador(crearPacalgo(t))  $\equiv$  salida(t)
jugador(moverJugador(p,d))  $\equiv$  d
puntaje(crearPacalgo(t))  $\equiv$  0
puntaje(moverJugador(p,d))  $\equiv$  puntaje(p)+1
inmunidad(crearPacalgo(t))  $\equiv$  if hayChocolate(salida(t)) then 10 else 0 fi
inmunidad(moverJugador(p,d))  $\equiv$  if hayChocolate(d)  $\wedge$   $\neg$ (chocolateComido(p,d)) then
    10
    else
        if inmunidad(p) > 0 then inmunidad(p) -1 else 0 fi
    fi
chocolateComido(crearPacalgo(t),c)  $\equiv$  c=salida(t)
chocolateComido(moverJugador(p,d),c)  $\equiv$  d=c  $\vee$  chocolateComido(p,c)
ganó?(p)  $\equiv$  jugador(p)=llegada(mostrarTablero(p))
perdió?(p)  $\equiv$  casillaAsustada(jugador(p),fantasmas(mostrarTablero(p)))  $\wedge$  inmunidad(p)=0
esAledaña?(p,d)  $\equiv$  enRango(d,dimensiones(mostrarTablero(p)))  $\wedge$ 
     $\neg$ (hayPared(d,mostrarTablero(p)))  $\wedge$ 
    ((( $\pi_1$ (jugador(p))-1 =  $\pi_1$ (d)  $\vee$   $\pi_1$ (d) =  $\pi_1$ (jugador(p))+1)  $\wedge$  ( $\pi_2$ (jugador(p))
    =  $\pi_2$ (d))  $\vee$  ( $\pi_2$ (jugador(p))-1 =  $\pi_2$ (d)  $\vee$   $\pi_2$ (d) =  $\pi_2$ (jugador(p)) + 1)  $\wedge$ 
     $\pi_1$ (jugador(p)) =  $\pi_1$ (d))

```

Fin TAD

4. TAD Nombre

TAD Nombre ES String

5. TAD Ranking

TAD Ranking

usa NOMBRE

igualdad observacional

$$(\forall r1, r2 : \text{ranking}) \left(r1 =_{\text{obs}} r2 \iff \left(\begin{array}{l} \text{jugadores}(r1) =_{\text{obs}} \text{jugadores}(r2) \wedge_L \\ (\forall j : \text{nombre})(j \in \text{jugadores}(r1) \\ \Rightarrow_L \text{puntaje}(j, r1) =_{\text{obs}} \text{puntaje}(j, r2)) \\ \wedge \\ (\forall p : \text{nat})(0 < p \leq \#(\text{jugadores}(r)) \\ \Rightarrow_L p\text{-simo}(p, r1) =_{\text{obs}} p\text{-simo}(p, r2)) \end{array} \right) \right)$$

observadores básicos

jugadores : ranking \rightarrow conj(nombres)

puntaje : nombre $j \times$ ranking $r \rightarrow$ nat

p-simo : nat $p \times$ ranking $r \rightarrow$ nombre

$\{j \in \text{jugadores}(r)\}$
 $\{0 < p \leq \#(\text{jugadores}(r))\}$

generadores

$\emptyset : \rightarrow$ ranking

nuevoGanador : nombre $j \times$ nat \times ranking $r \rightarrow$ ranking

repetirGanador : nombre $j \times$ nat \times ranking $r \rightarrow$ ranking

$\{j \notin \text{jugadores}(r)\}$

$\{j \in \text{jugadores}(r)\}$

otras operaciones

suPuesto : nombre $j \times$ ranking $r \rightarrow$ nat

esSuPuesto? : nombre $j \times$ nat $n \times$ ranking $r \rightarrow$ bool

$\{j \in \text{jugadores}(r) \wedge 0 < n \leq \#(\text{jugadores}(r))\}$

suPuestoDesde : nombre $j \times$ nat $n \times$ ranking $r \rightarrow$ nat

$\{j \in \text{jugadores}(r) \wedge 0 < n \leq \#(\text{jugadores}(r))\}$

axiomas

```

jugadores( $\emptyset$ )  $\equiv \emptyset$ 
jugadores(nuevoGanador(j,n,r))  $\equiv \text{Ag}(j, \text{jugadores}(r))$ 
jugadores(repetirGanador(j,n,r))  $\equiv \text{jugadores}(r)$ 
puntaje(j,nuevoGanador(j',n,r))  $\equiv$  if  $j \neq j'$  then puntaje(j,r) else n fi
puntaje(j,repetirGanador(j',n,r))  $\equiv$  if  $j \neq j'$  then puntaje(j,r) else min(puntaje(j,r),n) fi
p-simo(p,nuevoGanador(j,n,r))  $\equiv$  if  $\emptyset?(\text{jugadores}(r))$  then
    j
else
    if puntaje(p-simo(p,r),r)  $\leq n$  then
        p-simo(p,r)
    else
        if  $p=1 \vee_L \text{puntaje}(p-simo(p-1,r)) \leq n$  then
            j
        else
            p-simo(p-1,r)
        fi
    fi
fi
p-simo(p,repetirGanador(j,n,r))  $\equiv$  if  $n \geq \text{puntaje}(p-simo(p,r),r) \vee n \geq \text{puntaje}(j,r)$  then
    p-simo(p,r)
else
    if  $p=1 \vee_L \text{puntaje}(p-simo(p-1,r),r) \leq n$  then
        j
    else
        p-simo(p-1,r)
    fi
fi
suPuesto(j,r)  $\equiv$  suPuestoDesde(j,1,r)
suPuestoDesde(j,n,r)  $\equiv$  if esSuPuesto?(j,n,r) then n else suPuestoDesde(j,n+1,r) fi
esSuPuesto?(j,n,r)  $\equiv$  p-simo(n,r)=j

```

Fin TAD

6. TAD Fichin

TAD Fichin

usa CASILLA TABLERO PACALGO NOMBRE RANKING

igualdad observacional

$$(\forall f1, f2 : \text{fichin}) \left(f1 =_{\text{obs}} f2 \iff \left(\begin{array}{l} \text{tablero}(f1) =_{\text{obs}} \text{tablero}(f2) \wedge \\ \text{ranking}(f1) =_{\text{obs}} \text{ranking}(f2) \wedge \\ \text{jugando?}(f1) =_{\text{obs}} \text{jugando?}(f2) \wedge_{\text{L}} \\ \text{jugador}(f1) =_{\text{obs}} \text{jugador}(f2) \wedge \\ \text{partida}(f1) =_{\text{obs}} \text{partida}(f2) \end{array} \right) \right)$$

usa tablero, pacalgo, ranking

observadores básicos

tablero : fichin \longrightarrow tablero
 jugando? : fichin \longrightarrow bool
 jugador : fichin $f \longrightarrow$ nombre {jugando?(f)}
 partida : fichin $f \longrightarrow$ pacalgo {jugando?(f)}
 ranking : fichin \longrightarrow ranking

generadores

crearFichin : tablero $t \longrightarrow$ fichin {\neg(perdió?(crearPacalgo(t)))}
 nuevaPartida : nombre $n \times$ fichin \longrightarrow fichin {\text{lon}(n) > 2}
 moverse : casilla $c \times$ fichin $f \longrightarrow$ fichin {\text{jugando?}(f) \wedge_{\text{L}} \text{esAledaña}(c, \text{actual}(\text{partida}(f)))}

otras operaciones

siguientePuntaje : fichin $f \longrightarrow$ tupla(nombre, nat)

 $\left\{ \begin{array}{l} \text{jugando?}(f) \wedge_{\text{L}} \text{jugador}(f) \in \text{jugadores}(\text{ranking}(f)) \wedge_{\text{L}} \\ \text{p-simo}(1, \text{ranking}(f)) \neq \text{jugador}(f) \end{array} \right\}$

axiomas

```

tablero(crearFichin(t))  $\equiv$  t
tablero(nuevaPartida(j,f))  $\equiv$  tablero(f)
tablero(moverse(c,f))  $\equiv$  tablero(f)
jugando?(crearFichin(t))  $\equiv$  false
jugando?(nuevaPartida(j,f))  $\equiv$  true
jugando?(moverse(c,f))  $\equiv$   $\neg$ (perdió?(moverJugador(c,partida(f)))  $\vee$ 
    ganó?(moverJugador(c,partida(f))))
jugador(nuevaPartida(j,f))  $\equiv$  j
jugador(moverse(c,f))  $\equiv$  jugador(f)
partida(nuevaPartida(j,f))  $\equiv$  crearPacalgo(tablero(f))
partida(moverse(c,f))  $\equiv$  moverJugador(c,partida(f))
ranking(crearFichin(t))  $\equiv$   $\emptyset$ 
ranking(nuevaPartida(j,f))  $\equiv$  ranking(f)
ranking(moverse(c,f))  $\equiv$  if c $\neq$ llegada(tablero(f)) then
    ranking(f)
else
    if jugador(f)  $\in$  jugadores(ranking(f)) then
        repetirGanador(jugador(f),puntaje(partida(f)),ranking(f))
    else
        nuevoGanador(jugador(f),puntaje(partida(f)),ranking(f))
    fi
fi
siguientePuntaje(f)  $\equiv$   $\langle$ p-simo(suPuesto(jugador(f))-1,ranking(f)),
    puntaje(p-simo(suPuesto(jugador(f))-1,ranking(f))) $\rangle$ 

```

Fin TAD