

Interfaz

se explica con: MAPA

géneros: mapa.

Operaciones básicas de mapa

NUEVOMAPA(*in largo*: nat, *in alto*: nat, *in inicio*: coordenada, *in llegada*: coordenada, *in fantasmas*: conj(coordenada), *in paredes*: conj(coordenada), *in chocolates*: conj(coordenada), *out m*: mapa) $\rightarrow res$: bool
Pre $\equiv \{\text{true}\}$
Post $\equiv \{(res = (inicio \neq llegada \wedge todosEnRango(paredes \cup fantasmas \cup chocolates \cup \{inicio, llegada\}, largo, alto) \wedge \{inicio, llegada\} \cap (fantasmas \cup paredes) = \emptyset \wedge disjuntosDeAPares(paredes, fantasmas, chocolates))) \wedge_L res \Rightarrow _L m =_{\text{obs}} nuevoMapa(largo, alto, inicio, llegada, paredes, fantasmas, chocolates))\}$
Complejidad: $O(chocolates + fantasmas + paredes)$
Descripción: Genera un nuevo mapa
Aliasing: -

ESCASILLEROPELIGROSO(*in m*: mapa, *in posicin*: coordenada) $\rightarrow res$: res
Pre $\equiv \{\text{true}\}$
Post $\equiv \{res = \text{distConFantasmasMásCercano}(fantasmas(m), posición) \leq 3\}$
Complejidad: $O(1)$
Descripción: Devuelve true si el casillero es peligroso, es peligroso si existe un fantasma con distancia ≤ 3 respecto a la posición
Aliasing: -

ENRANGO(*in m*: mapa, *in posicin*: coordenada) $\rightarrow res$: bool
Pre $\equiv \{\text{true}\}$
Post $\equiv \{res = \text{enRango}(posición, largo(m), alto(m))\}$
Complejidad: $O(1)$
Descripción: Devuelve true si la posición se encuentra en rango
Aliasing: -

CANTCHOCOLATES(*in map*: mapa) $\rightarrow res$: nat
Pre $\equiv \{\text{true}\}$
Post $\equiv \{res = \#(chocolates(map))\}$
Complejidad: $O(c)$
Descripción: Devuelve la cantidad de chocolates en el mapa
Aliasing: -

ESPARED(*in map*: mapa, *in posicin*: coordenada) $\rightarrow res$: bool
Pre $\equiv \{\text{true}\}$
Post $\equiv \{res = \text{true} \iff posicin \in paredes(map)\}$
Complejidad: $O(1)$
Descripción: Devuelve el conjunto de paredes
Aliasing: -

INICIO(*in map*: mapa) $\rightarrow res$: coordenada
Pre $\equiv \{\text{true}\}$
Post $\equiv \{res = inicio(map)\}$
Complejidad: $O(1)$
Descripción: Devuelve la coordenada de inicio del mapa
Aliasing: -

LLEGADA(*in map*: mapa) $\rightarrow res$: coordenada
Pre $\equiv \{\text{true}\}$
Post $\equiv \{res = llegada(map)\}$
Complejidad: $O(1)$
Descripción: Devuelve la coordenada de llegada del mapa
Aliasing: -

Representación

mapa se representa con *mp*

donde *casillero* es tupla(*fantasma*: bool,
peligrosa: bool,
pared: bool,
idChocolate: int)

donde *columna* es array[0...*largo*] de casillero

donde *mp* es tupla(*matriz*: array[0...*alto*] de columna,
#chocolates: nat,
alto: nat,
largo: nat,
inicio: coordenada,
llegada: coordenada)

Invariante de representación

Rep : mp \rightarrow boolean

Rep(*e*) \equiv True \iff ($0 \leq e.inicio_1 \leq e.largo \wedge 0 \leq e.inicio_2 \leq e.alto$) \wedge
 $(0 \leq e.llegada_1 \leq e.largo \wedge 0 \leq e.llegada_2 \leq e.alto) \wedge$
 $(e.inicio \neq e.llegada) \wedge$
 $(\forall i: \text{nat})(0 \leq i \leq e.largo) \Rightarrow_L ($
 $(\forall j: \text{nat})(0 \leq j \leq e.alto) \Rightarrow_L ($
 $(\beta(e.matriz[i][j].pared) +$
 $\beta(e.matriz[i][j].fantasma) +$
 $\beta(0 \leq e.matriz[i][j].idChocolate < e.\#chocolates) \leq 1) \wedge$
 $((\forall n: \mathbb{N})(0 \leq n < e.\#chocolates)$
 $(\exists! i, j: \text{nat})(0 \leq i \leq e.largo \wedge 0 \leq j \leq e.alto) \wedge_L$
 $(e.matriz[i][j].chocolates = n)) \wedge$
 $(e.matriz[i][j].peligrosa \Rightarrow_L (\exists n, m: \text{nat})(0 \leq n \leq e.largo \wedge 0 \leq m \leq e.alto) \wedge_L$
 $(e.matriz[n][m].fantasma \wedge \text{distancia}(\langle i, j \rangle, \langle n, m \rangle) \leq 3)))$
 $)$

Función de abstracción

Abs : mp *e* \rightarrow Mapa

{Rep(*e*)}

($\forall e: \text{mp}$) Abs(*e*) =_{obs} *m*: mapa | largo(*m*) = *e.largo* \wedge

alto(*m*) = *e.alto* \wedge

#(chocolates(*m*)) = *e.\#chocolates* \wedge

e.inicio = inicio(*m*) \wedge

e.llegada = llegada(*m*) \wedge

($\forall i: \text{nat})(0 \leq i \leq e.largo) \Rightarrow_L ($

($\forall j: \text{nat})(0 \leq j \leq e.alto) \Rightarrow_L ($

e.matriz[*i*][*j*].fantasma $\iff \langle i, j \rangle \in \text{fantasmas}(m) \wedge$

e.matriz[*i*][*j*].paredes $\iff \langle i, j \rangle \in \text{paredes}(m) \wedge$

$0 \leq e.matriz[i][j].chocolate < e.\#chocolates \iff \langle i, j \rangle \in \text{chocolates}(m) \wedge$

e.matriz[*i*][*j*].peligrosa $\iff \text{distConFantasmasMsCercano}(\text{fantasmas}(m), \langle i, j \rangle) \leq 3)$)

Interfaz

se explica con: PARTIDA

géneros: partida.

Operaciones básicas de partida

NUEVAPARTIDA(in m : mapa) $\rightarrow res$: partida

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res = \text{nuevaPartida}(m)\}$

Complejidad: $O(c)$, $c = \#chocolates$

Descripción: Genera una nueva partida

Aliasing: -

MOVER(in/out p : partida, in d : dirección)

Pre $\equiv \{p_0 = p\}$

Post $\equiv \{p = \text{mover}(p_0, d)\}$

Complejidad: $O(1)$

Descripción: Mueva la posición del jugador un casillero

Aliasing: -

GANÓ?(in p : partida) $\rightarrow res$: bool

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res = \text{ganó?}(p)\}$

Complejidad: $O(1)$

Descripción: Devuelve true si el jugador gana la partida

Aliasing: -

PERDIÓ?(in p : partida) $\rightarrow res$: bool

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res = \text{perdió?}(p)\}$

Complejidad: $O(1)$

Descripción: Devuelve true si el jugador perdio la partida

Aliasing: -

JUGADOR(in p : partida) $\rightarrow res$: coordenada

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res = \text{perdió?}(p)\}$

Complejidad: $O(1)$

Descripción: Devuelve la posición del jugador

Aliasing: -

CANTMOV(in p : partida) $\rightarrow res$: nat

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res = \text{cantMov}(p)\}$

Complejidad: $O(1)$

Descripción: Devuelve la cantidad de movimientos del jugador

Aliasing: -

Representación

partida se representa con pt

donde pt es tupla($mapa$: mp,
jugador: coordenada,
chocolates: array[0...c] de bool,
cantMov: nat,
inmunidad: nat,
gano: bool,
perdio: bool)

Funciones auxiliares

distancia : coordenada \times coordenada \rightarrow nat

distanciaMinima : coordenada \times conj(coordenada) \rightarrow bool

chocolatesSinComer : pt \rightarrow conj(coordenada)

distancia(x, y) $\equiv |x_1 - y_1| + |x_2 - y_2|$

distanciaMinima(j, c) \equiv **if** $\#(c) = 1$ **then**

distancia($j, dameUno(c)$)

else

mín(distancia($j, dameUno(c)$), distanciaMinima($j, sinUno(c)$))

fi

distanciaMinima(e) \equiv Esta función devuelve el conjunto de coordenadas de chocolates en la $e.mapa.matriz$ cuyo Id en el array de la partida ($e.chocolates$) aun estan en true

Invariante de representación

Rep : pt \rightarrow boolean

Rep(e) \equiv True $\iff (e.mapa.EnRango(e.mapa, e.jugador) \wedge$

$long(e.chocolates) = e.mapa.\#chocolates) \wedge_L$

$e.cantMov = 0 \Rightarrow$

$e.mapa.inicio = jugador \wedge e.chocolates = e.mapa.chocolates - jugador \wedge$

if $e.chocolates[e.mapa.matriz[e.jugador_1][e.jugador_2].chocolate]$ **then**

$e.inmunidad = 10$

else

$e.inmunidad = 0$

fi \wedge

$\neg(e.chocolates[e.mapa.matriz[e.jugador_1][e.jugador_2].chocolate]) \wedge$

$e.inmunidad = 10 \Rightarrow 0 \leq e.mapa.matriz[e.jugador_1][e.jugador_2].chocolate < e.mapa.\#chocolates \wedge$

$e.inmunidad \leq 10 - distanciaMinima(jugador, chocolatesSinComer(e)) \wedge$

$e.gano \iff jugador = e.mapa.llegada \wedge$

$e.perdio \iff e.inmunidad = 0 \wedge e.mapa.distConFantasmasMásCercano(e.mapa, e, jugador) \leq 3$

Función de abstracción

Abs : pt $e \rightarrow$ partida

{Rep(e)}

$(\forall e : pt) Abs(e) =_{obs} p : partida \mid mapa(p) = e.mapa \wedge$

$jugador(p) = e.jugador \wedge$

$chocolates(p) = chocolatesSinComer(e) \wedge$

$cantMov(p) = e.cantMov \wedge$

$inmunidad(p) = e.inmunidad \wedge$

$ganó?(p) = e.gano \wedge$

$perdió?(p) = e.perdio$

Interfaz

se explica con: FICHÍN

géneros: fichin.

Operaciones básicas de fichin

NUEVOFICHIN(**in** m : mapa) $\rightarrow res$: fichin

Pre $\equiv \{true\}$

Post $\equiv \{res = nuevoFichin(m)\}$

Complejidad: $O(?)$

Descripción: Genera un fichín

Aliasing: -

NUEVAPARTIDA(**in/out** f : fichin, **in** j : jugador) $\rightarrow res$: bool

Pre $\equiv \{f_0 = f\}$

Post $\equiv \{res = \neg alguienJugando?(f) \wedge_L res \Rightarrow_L f = nuevaPartida(f_0, j)\}$

Complejidad: $O(c)$

Descripción: Inicia una nueva partida

Aliasing: -

MOVER(**in/out** f : fichin, **in** d : dirección) $\rightarrow res$: bool

Pre $\equiv \{f_0 = f\}$

Post $\equiv \{res = alguienJugando?(f) \wedge_L res \Rightarrow_L f = mover(f_0, d)\}$

Complejidad: $O(|J|)$

Descripción: Mueve en la dirección indicada

Aliasing: -

VERRANKING(**in** f : fichin) $\rightarrow res$: ranking

Pre $\equiv \{true\}$

Post $\equiv \{res = ranking(f)\}$

Complejidad: $O(1)$

Descripción: Devuelve el ranking del fichin

Aliasing: -

OBJETIVO(**in** f : fichin, **out** o : tupla<jugador, nat>) $\rightarrow res$: res

Pre $\equiv \{true\}$

Post $\equiv \{(res = alguienJugando?(f) \wedge def?(jugadorActual(f), ranking(f))) \wedge_L res \Rightarrow_L o = objetivo(f)\}$

Complejidad: $O(\log_2 n)$

Descripción: Devuelve una tupla con el oponente y su puntaje

Aliasing: -

Representación

partida se representa con fch

donde fch es tupla($mapa$: mp,
 $alguienJugando$: bool,
 $jugadorActual$: string,
 $partidaActual$: pt,
 $ranking$: dicc(string, nat))

Invariante de representación

$Rep : fch \longrightarrow \text{boolean}$

$Rep(e) \equiv \text{True} \iff$
 $(e.alguienJugando \iff (\text{longitud}(e.jugadorActual) > 0 \wedge \neg e.partida.gano \wedge \neg e.partida.perdio)) \wedge$
 $(e.pt.gano \Rightarrow$
 $\text{def?}(e.jugadorActual, e.ranking) \wedge_L$
 $\text{obtener}(e.jugadorActual, e.ranking) \leq e.partida.cantMov) \wedge$
 $(e.mapa = e.pt.mapa)$

Función de abstracción

$Abs : fch \times e \longrightarrow \text{partida}$

$\{Rep(e)\}$

$(\forall e : fch) Abs(e) =_{\text{obs}} f : \text{fichin} \mid \text{mapa}(f) = e.mapa \wedge$
 $\text{alguienJugando}(f) = e.alguienJugando \wedge$
 $\text{ranking}(f) = e.ranking \wedge$
 $e.alguienJugando \Rightarrow_L \text{partidaActual}(f) = e.partidaActual \wedge$
 $e.alguienJugando \Rightarrow_L \text{jugadorActual}(f) = e.jugadorActual$