

## TP Algo2landia

Algoritmos y Estructuras de Datos II, DC, UBA.

## 1. TAD ALGO2LANDIA

**TAD OBJETIVO ES TUPLA(TUPLA(COLOR  $\times$  CASILLERO)  $\times$  TUPLA(COLOR  $\times$  CASILLERO))**

TAD COLOR ES STRING

**TAD** CASILLERO ES TUPLA(NAT  $\times$  NAT)

TAD Algo2landia

igualdad observacional

$$(\forall al_1, al_2 : \text{algo2landia}) \quad \left( al_1 =_{\text{obs}} al_2 \iff \begin{pmatrix} \text{grilla}(al_1) =_{\text{obs}} \text{grilla}(al_2) \wedge \\ (\forall a: \text{agente}) (\text{posición}(a, al_1) =_{\text{obs}} \\ \text{posición}(a, al_2) \wedge \\ \text{cantMovimientos}(a, al_1) =_{\text{obs}} \\ \text{cantMovimientos}(a, al_2) \wedge \\ \text{objetivosDisponibles}(a, al_1) =_{\text{obs}} \\ \text{objetivosDisponibles}(a, al_2) \wedge \\ \text{objetivosCompletados}(a, al_1) =_{\text{obs}} \\ \text{objetivosCompletados}(a, al_2)) \end{pmatrix} \right)$$

g neros      algo2landia

**exporta** algo2landia, generadores, observadores, sePuedeMover?, esFinalObjetivo?

**usa** DIRECCIÓN, AGENTE, GRILLA, CONJ(OBJETIVO)

## observadores básicos

grilla	: algo2landia	→ grilla
posición	: agente × algo2landia	→ casillero
cantMovimientos	: agente × algo2landia	→ nat
objetivosDisponibles	: agente × algo2landia	→ conj(objetivo)
#objetivosCompletados	agente × algo2landia	→ nat

generadores

$$\begin{array}{lll}
\text{comenzar} & : \text{agente} \times \text{conj}(\text{objetivo}) \times \text{grilla} & \longrightarrow \text{algo2landia} \\
\text{mover} & : \text{agente } a \times \text{direccion } d \times \text{algo2landia } al & \longrightarrow \text{algo2landia} \\
& & \{ \text{haySiguienteCasillero?}(\text{grilla}(al), \text{posición}(a, al), d) \} \\
\text{agregarObjetivo} & : \text{objetivo } ob \times \text{agente} \times \text{algo2landia } al & \longrightarrow \text{algo2landia} \\
& & \{ \text{hayObjeto?}(\pi_2(\pi_1(ob)), \text{grilla}(al)) \wedge \pi_1(\pi_1(ob)) = o \wedge \pi_1(\pi_2(ob)) = c \}
\end{array}$$

otras operaciones

$$\begin{aligned} \text{sePuedeMover?} &: \text{agente} \times \text{dirección} \times \text{algo2landia} \longrightarrow \text{bool} \\ \text{esFinalObjetivo?} &: \text{agente} \times \text{dirección} \times \text{algo2landia} \longrightarrow \text{bool} \end{aligned}$$
$$\text{axiomas} \quad \forall d: \text{dirección}, \forall a, a': \text{agente}, \forall ob: \text{objetivo}, \forall co: \text{conj}(\text{objetivo}), \forall g: \text{grilla}, \forall al: \text{algo2landia} \\ \text{grilla}(\text{comenzar}(a, co, g)) \equiv g$$

<code>grilla(mover(<math>a</math>, <math>d</math>, <math>al</math>))</code>	$\equiv$ <code>grilla(<math>al</math>)</code>
<code>posición(<math>a</math>, comenzar(<math>a'</math>, <math>co</math>, <math>g</math>))</code>	$\equiv$ <code>posición(<math>a</math>, <math>al</math>)</code>
<code>posición(<math>a</math>, mover(<math>a'</math>, <math>d</math>, <math>al</math>))</code>	$\equiv$ <b>if</b> $a = a'$ <b>then</b> <b>if</b> sePuedeMover?( $a$ , $d$ , $al$ ) <b>then</b> siguienteCasillero(posición( $a$ , $al$ ), $al$ ) <b>else</b> posición( $a$ , $al$ ) <b>else</b> posición( $a$ , $al$ ) <b>fi</b> <b>fi</b>
<code>cantMovimientos(<math>a</math>, comenzar(<math>a'</math>, <math>g</math>))</code>	$\equiv$ <b>if</b> $a = a'$ <b>then</b> 0 <b>else</b> cantMovimientos( $a$ , $al$ ) <b>fi</b>
<code>cantMovimientos(<math>a</math>, mover(<math>a'</math>, <math>d</math>, <math>al</math>))</code>	$\equiv$ <b>if</b> $a = a'$ <b>then</b> cantMovimientos( $a$ , $al$ ) + 1 <b>else</b> cantMovimientos( $a$ , $al$ ) <b>fi</b>
<code>objetivosDisponibles(<math>a</math>, comenzar(<math>a'</math>, <math>co</math>, <math>g</math>))</code>	$\equiv$ <b>if</b> $a = a'$ <b>then</b> $co$ <b>else</b> objetivosDisponibles( $a$ , $al$ )
<code>objetivosDiponibles(<math>a</math>, mover(<math>a'</math>, <math>d</math>, <math>al</math>))</code>	$\equiv$ <b>if</b> $a = a'$ <b>then</b> <b>if</b> sePuedeMover?( $a$ , $d$ , $al$ ) $\wedge$ esFinalObjetivo?( $a$ , $d$ , $al$ ) <b>then</b> sinUno(objetivosDiponibles( $a$ , $al$ )) <b>else</b> objetivosDiponibles( $a$ , $al$ ) <b>fi</b> <b>else</b> objetivosDiponibles( $a$ , $al$ ) <b>fi</b>
<code>objetivosDisponibles(<math>a</math>, agregarObjetivo(<math>ob</math>, <math>a'</math>, <math>al</math>))</code>	$\equiv$ <b>if</b> $a = a'$ <b>then</b> Ag( $ob$ , objetivosDisponibles( $a$ , $al$ )) <b>else</b> objetivosDisponibles( $a$ , $al$ )
<code>#objetivosCompletados(<math>a</math>, comenzar(<math>a'</math>, <math>co</math>, <math>g</math>))</code>	$\equiv$ <b>if</b> $a = a'$ <b>then</b> 0 <b>else</b> objetivosCompletados( $a$ , $al$ )
<code>#objetivosCompletados(<math>a</math>, mover(<math>a'</math>, <math>d</math>, <math>al</math>))</code>	$\equiv$ <b>if</b> $a = a'$ <b>then</b> <b>if</b> sePuedeMover?( $a$ , $d$ , $al$ ) $\wedge$ esFinalObjetivo?( $a$ , $d$ , $al$ ) <b>then</b> 1 + #objetivosCompletados( $a$ , $al$ ) <b>else</b> #objetivosCompletados( $a$ , $al$ ) <b>fi</b> <b>else</b> #objetivosCompletados( $a$ , $al$ ) <b>fi</b>
<code>#objetivosCompletados(<math>a</math>, agregarObjetivo(<math>ob</math>, <math>a'</math>, <math>al</math>))</code>	$\equiv$ #ObjetivosCompletados( $a$ , $al$ )
<code>sePuedeMover?(<math>a</math>, <math>d</math>, <math>al</math>)</code>	$\equiv$ (esPiso?(grilla( $al$ ), posición( $a$ , $al$ )) $\wedge$ esElevación?(siguienteCasillero(posición( $a$ , $al$ ), $d$ ))) $\vee$ (hayObjeto?(posición( $a$ , $al$ ), grilla( $al$ )) $\wedge$ hayObjeto?(siguienteCasillero(posición( $a$ , $al$ ), $d$ ), grilla( $al$ )))
<code>esFinalObjetivo?(<math>a</math>, <math>d</math>, <math>al</math>)</code>	$\equiv$ hayObjeto?(posición( $a$ , $al$ ), grilla( $al$ )) $\wedge_L$ colorObjeto(posición( $a$ , $al$ ), grilla( $al$ )) = $\pi_1(\pi_1(\text{dameUno}(\text{objetivosDiponibles}(a, al)),$ grilla( $al$ ))) $\wedge$ $\pi_1(\pi_2(\text{dameUno}(\text{objetivosDiponibles}(a, al)))) =$ colorCasillero(siguienteCasillero(posicion( $a$ , $al$ ), $d$ ))

**Fin TAD**

## 2. TAD GRILLA

### TAD Grilla

#### igualdad observacional

$$(\forall g_1, g_2 : \text{grilla}) \left( g_1 =_{\text{obs}} g_2 \iff \left( \begin{array}{l} \text{alto}(g_1) =_{\text{obs}} \text{alto}(g_2) \wedge \text{ancho}(g_1) =_{\text{obs}} \text{ancho}(g_2) \\ \wedge \text{pisos}(g_1) =_{\text{obs}} \text{pisos}(g_2) \wedge \text{rampas}(g_1) =_{\text{obs}} \text{rampas}(g_2) \\ \wedge \text{rampas}(g_2) \wedge \text{elevaciones}(g_1) =_{\text{obs}} \text{elevaciones}(g_2) \wedge \\ \text{casillerosColoridos}(g_1) =_{\text{obs}} \text{casillerosColoridos}(g_2) \wedge \\ \text{casillerosConObjetos}(g_1) =_{\text{obs}} \text{casillerosConObjetos}(g_2) \end{array} \right) \right)$$

**géneros** grilla

**exporta** grilla, generadores, observadores, esCasilleroVálido?, haySiguienteCasillero?, siguienteCasillero, esPiso?, esRampa?, esElevación?, hayObjeto?, tieneColorCas?, colorObjeto, colorCasillero

**usa** BOOL, NAT, CASILLERO, DIRECCIÓN, COLOR, CONJ(CASILLERO), DICC(CASILLERO, COLOR)

#### observadores básicos

alto : grilla  $\rightarrow$  nat  
 ancho : grilla  $\rightarrow$  nat  
 pisos : grilla  $\rightarrow$  conj(casillero)  
 rampas : grilla  $\rightarrow$  conj(casillero)  
 elevaciones : grilla  $\rightarrow$  conj(casillero)  
 casillerosColoridos : grilla  $\rightarrow$  dicc(casillero, color)  
 casillerosConObjetos : grilla  $\rightarrow$  dicc(casillero, color)

#### generadores

crear : conj(casillero)  $p \times$  conj(casillero)  $r \times$  conj(casillero)  $e \times$  dicc(casillero, color)  $cc \times$  dicc(casillero, color)  $co$   
 $\times$  nat  $alt \times$  nat  $anch$   
 $\left\{ \begin{array}{l} (alt > 0 \wedge anch > 0 \wedge p \cap r = \emptyset \wedge p \cap e = \emptyset \wedge r \cap e = \emptyset) \wedge \\ (\#p + \#r + \#e = alt \times anch) \wedge_L \\ (\forall r_i : \text{casillero})(r_i \in r \rightarrow_L (\exists d : \text{dirección})(\exists p_i : \text{casillero})(p_i \in p \wedge_L \text{siguienteCasillero}(r_i, d) = p_i)) \\ \wedge (\exists d' : \text{dirección})(\exists e_i : \text{casillero})(e_i \in e \wedge_L \text{siguienteCasillero}(r_i, d') = e_i)) \wedge \\ \text{claves}(cc) \subseteq (p \cup r \cup e) \wedge \text{claves}(co) \subseteq (p \cup r \cup e) \wedge \\ (\forall c_i, c_j : \text{casillero})(((c_i \neq c_j \wedge (c_i \in \text{claves}(cc) \wedge c_j \in \text{claves}(cc))) \rightarrow_L \text{obtener}(c_i, cc) \neq \\ \text{obtener}(c_j, cc)) \wedge (\forall c_k, c_l : \text{casillero})(((c_k \neq c_l \wedge (c_k \in \text{claves}(co) \wedge c_l \in \text{claves}(co))) \rightarrow_L \\ \text{obtener}(c_k, co) \neq \text{obtener}(c_l, co)) \end{array} \right\}$

#### otras operaciones

esCasilleroVálido? : casillero  $\times$  grilla  $\rightarrow$  bool  
 haySiguienteCasillero? : casillero  $c \times$  dirección  $\times$  grilla  $g \rightarrow$  bool {esCasilleroVálido?( $c, g$ )}  
 siguienteCasillero : casillero  $c \times$  dirección  $d \rightarrow$  casillero  
 $\{(\forall g : \text{grilla})(\text{esCasilleroVálido}(c, g) \rightarrow_L \text{haySiguienteCasillero?}(c, d, g))\}$   
 esPiso? : casillero  $\times$  grilla  $\rightarrow$  bool  
 esRampa? : casillero  $\times$  grilla  $\rightarrow$  bool  
 esElevación? : casillero  $\times$  grilla  $\rightarrow$  bool  
 hayObjeto? : casillero  $\times$  grilla  $\rightarrow$  bool  
 tieneColorCas? : casillero  $\times$  grilla  $\rightarrow$  bool  
 colorObjeto : casillero  $c \times$  grilla  $g \rightarrow$  color {hayObjeto?( $c, g$ )}  
 colorCasillero : casillero  $c \times$  grilla  $g \rightarrow$  color {tieneColorCas?( $c, g$ )}

<b>axiomas</b>	$\forall c: \text{casillero}, \forall d: \text{dirección}, \forall alt, anch: \text{nat}, \forall p, r, e: \text{conj}(\text{casillero}), \forall cc, co: \text{dicc}(\text{casillero}, \text{color})$ $\forall g: \text{grilla}$
<code>alto(crear(<i>p, r, e, cc, co, alt, anch</i>))</code>	$\equiv alt$
<code>ancho(crear(<i>p, r, e, cc, co, alt, anch</i>))</code>	$\equiv anch$
<code>pisos(crear(<i>p, r, e, cc, co, alt, anch</i>))</code>	$\equiv p$
<code>rampas(crear(<i>p, r, e, cc, co, alt, anch</i>))</code>	$\equiv r$
<code>elevaciones(crear(<i>p, r, e, cc, co, alt, anch</i>))</code>	$\equiv e$
<code>casillerosColoridos(crear(<i>p, r, e, cc, co, alt, anch</i>))</code>	$\equiv cc$
<code>casillerosConObjetos(crear(<i>p, r, e, cc, co, alt, anch</i>))</code>	$\equiv co$
<code>esCasilleroVálido(<i>c, g</i>)?</code>	$\equiv c \in \text{pisos}(g) \vee c \in \text{rampas}(g) \vee c \in \text{elevaciones}(g)$
<code>haySiguienteCasillero?(<i>c, d, g</i>)</code>	$\equiv (d = \text{arriba} \wedge \pi_1(c) > 0) \vee (d = \text{abajo} \wedge \pi_1(c) < \text{alto}(g)) \vee (d = \text{derecha} \wedge \pi_2(c) > \text{ancho}(g)) \vee (d = \text{izquierda} \wedge \pi_2(c) > 0)$
<code>siguienteCasillero(<i>c, d</i>)</code>	$\equiv$ <b>if</b> <i>d</i> = arriba <b>then</b> $\langle \pi_1(c) + 1, \pi_2(c) \rangle$ <b>else</b> <b>if</b> <i>d</i> = abajo <b>then</b> $\langle \pi_1(c) - 1, \pi_2(c) \rangle$ <b>else</b> <b>if</b> <i>d</i> = derecha <b>then</b> $\langle \pi_1(c), \pi_2(c) + 1 \rangle$ <b>else</b> $\langle \pi_1(c), \pi_2(c) - 1 \rangle$ <b>fi</b> <b>fi</b> <b>fi</b>
<code>esPiso?(<i>c, g</i>)</code>	$\equiv c \in \text{pisos}(g)$
<code>esRampa?(<i>c, g</i>)</code>	$\equiv c \in \text{rampas}(g)$
<code>esElevación?(<i>c, g</i>)</code>	$\equiv c \in \text{elevaciones}(g)$
<code>hayObjeto?(<i>c, g</i>)</code>	$\equiv c \in \text{casillerosConObjetos}(g)$
<code>tieneColorCas?(<i>c, g</i>)</code>	$\equiv c \in \text{casillerosColoridos}(g)$
<code>colorObjeto(<i>c, g</i>)</code>	$\equiv \text{obtener}(c, \text{casillerosConObjetos}(g))$
<code>colorCasillero(<i>c, g</i>)</code>	$\equiv \text{obtener}(c, \text{casillerosColoridos}(g))$

**Fin TAD**