

# Algoritmos y Estructuras de Datos II

Primer Cuatrimestre de 2015

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

## Trabajo Práctico 1

Especificación

Integrante	LU	Correo electrónico
BENITEZ, Nelson	945/13	nelson.benitez92@gmail.com
ROIZMAN, Violeta	273/11	violeroizman@gmail.com
VÍZQUEZ, Jéssica	318/13	jesis_93@hotmail.com
ZAVALLA, Agustín	670/13	nkm747@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

## Índice

<b>1. TAD AS</b>	<b>3</b>
<b>2. TAD CAMPUS</b>	<b>5</b>

# 1. TAD AS

## TAD AS

**géneros** as

**igualdad observacional**

$$(\forall dc, dc' : \text{dcnet}) (dc =_{\text{obs}} dc' \iff ())$$

**usa** CAMPUS

**exporta**

**observadores básicos**

campus : as  $\rightarrow$  campus

seguridad : as  $\rightarrow$  conj(seguridad)

hayEst? : as  $a \times \text{pos } p \rightarrow \text{bool}$

$$\{posValida(campus(a), p)\}$$

hayHippie? : as  $a \times \text{pos } p \rightarrow \text{bool}$

$$\{posValida(campus(a), p)\}$$

#capturas : as  $a \times \text{seg } s \rightarrow \text{nat}$

$$\{s \in seguridad(a)\}$$

#sanciones : as  $a \times \text{seg } s \rightarrow \text{nat}$

$$\{s \in seguridad(a)\}$$

**generadores**

nueva : campus  $\times$  conj(seguridad)  $\rightarrow$  as

$$\{(\forall segs:e) posValida(c, pos(e)) \wedge (\forall segs:s, s1) id(s) \neq id(s1) \Rightarrow pos(s) \neq pos(s1)\}$$

moverEst : as  $a \times \text{pos } pe \times \text{pos } pd \rightarrow \text{as}$

$$\left\{ \begin{array}{l} posValida(campus(a), pe) \wedge_L hayEst?(a, pe) \wedge adyacente(campus(a), pe, pd) \wedge \\ posValidaPersona(as, pd) \end{array} \right\}$$

nuevoHippie : as  $a \times \text{pos } p \rightarrow \text{as}$

$$\{posIngreso(campus(a), p) \wedge posValidaPersona(a, p)\}$$

nuevoEst : as  $a \times \text{pos } p \rightarrow \text{as}$

$$\{posIngreso(campus(a), p) \wedge posValidaPersona(a, p)\}$$

sacarEst : as  $a \times \text{pos } p \rightarrow \text{as}$

$$\{posValida(campus(a), p) \wedge_L hayEst?(a, p) \wedge posIngreso(a, p)\}$$

**otras operaciones**

haySeg? : as  $a \times \text{pos } p \rightarrow \text{bool}$

posValidaPersona : as  $a \times \text{pos } p \rightarrow \text{bool}$

posIngreso : as  $a \times \text{pos } p \rightarrow \text{bool}$

moverTodos : as  $a \times \text{conj(seguridad) } segs \rightarrow \text{conj(seguridad)}$

moverSeg : as  $a \times \text{seguridad } seg \times \text{pos } posSig \rightarrow \text{seguridad}$

proximasPosiciones : as  $a \times \text{conj(pos) } minPos \times \text{pos } posAct \rightarrow \text{conj(pos)}$

$$\{\neg(emptyset?(minPos)) \wedge_L posValida(campus(a), posAct) \wedge posicionesValidas(campus(a), minPos)\}$$

hippiesMasCerca : as  $a \times \text{seguridad } seg \rightarrow \text{conj(pos)}$

$$\{seg \in seguridad(a) \wedge hayHippies(a)\}$$

**axiomas**

$$campus(nueva(c, segs)) \equiv c$$

$$campus(moverEst(a, p_1, p_2)) \equiv campus(a)$$

$$campus(nuevoEst(a, p_1)) \equiv campus(a)$$

$$campus(nuevoHippie(a, p_1)) \equiv campus(a)$$

$$campus(sacarEst(a, p_1)) \equiv campus(a)$$

$$seguridad(nueva(c, segs)) \equiv segs$$

$$seguridad(moverEst(a, p_1, p_2)) \equiv moverTodos(seguridad(a))$$

seguridad(nuevoEst( $a, p_1$ ))	$\equiv$ <i>campus</i> ( $a$ )
seguridad(nuevoHippie( $a, p_1$ ))	$\equiv$ <i>campus</i> ( $a$ )
seguridad(sacarEst( $a, p_1$ ))	$\equiv$ <i>campus</i> ( $a$ )
hayEst?(nueva( $c, segs$ ), $p$ )	$\equiv$ <i>False</i>
hayEst?(nuevoEst( $a, p_1$ ), $p$ )	$\equiv$ <b>if</b> $p_1 = p$ <b>then</b> <i>True</i> <b>else</b> <i>hayEst?</i> ( $a, p$ ) <b>fi</b>
hayEst?(moverEst( $a, p_1, p_2$ ), $p$ )	$\equiv$ <b>if</b> $p_1 = p$ <b>then</b> <i>False</i> <b>else</b> <b>if</b> $p_2 = p$ <b>then</b> <i>True</i> <b>else</b> <i>hayEst?</i> ( $a, p$ ) <b>fi</b> <b>fi</b>
hayEst?(nuevoHippie( $a, p_1$ ), $p$ )	$\equiv$ <i>hayEst?</i> ( $a, p$ )
hayEst?(sacarEst( $a, p_1$ ), $p$ )	$\equiv$ <b>if</b> $p_1 = p$ <b>then</b> <i>False</i> <b>else</b> <i>hayEst?</i> ( $a, p$ ) <b>fi</b>
hayHippie?(nueva( $c, segs$ ), $p$ )	$\equiv$ <i>False</i>
hayHippie?((nuevoHippie( $a, p_1$ ), $p$ )	$\equiv$ <b>if</b> $p_1 = p$ <b>then</b> <i>True</i> <b>else</b> <i>hayHippie?</i> ( $a, p$ ) <b>fi</b>
#capturas(nueva( $c, segs$ ), $p$ )	$\equiv$ 0
#sanciones(nueva( $c, segs$ ), $p$ )	$\equiv$ 0
moverTodos( $a, segs$ )	$\equiv$ <b>if</b> ( $\emptyset?(segs)$ ) <b>then</b> $\emptyset$ <b>else</b> <b>if</b> ( <i>hayHippies?</i> ( $a$ )) <b>then</b> <i>Ag</i> ( <i>moverTodos</i> ( $a, sinUno(segs)$ ), <i>moverSeg</i> ( $a, dameUno(segs)$ , <i>dameUno</i> ( <i>proxPosiciones</i> <i>(hippiesMasCerca</i> ( $a, dameUno(segs)$ )))))) <b>else</b> $segs$ <b>fi</b> <b>fi</b>
moverSeg( $a, seg, nPos$ )	$\equiv$ <b>if</b> ( <i>distMan</i> ( <i>campus</i> ( $a$ ), $\pi_2(seg)$ , $nPos$ ) $\geq$ 2 $\vee \neg(posValida(campus(a), nPos))$ ) <b>then</b> $seg$ <b>else</b> $< \pi_1(seg), nPos >$ <b>fi</b>



$\text{alto} : \text{campus} \rightarrow \text{nat}$   
 $\text{ancho} : \text{campus} \rightarrow \text{nat}$   
 $\text{obstaculos} : \text{campus} \rightarrow \text{conj}(\text{pos})$

**generadores**

$\text{nuevo} : \text{nat } ancho \times \text{nat } alto \times \text{conj}(\text{pos}) \text{ obst} \rightarrow \text{campus}$   
 $\{1 \leq ancho \wedge 1 \leq alto \wedge (\forall p:\text{pos}) p \in \text{obst} \Rightarrow_L \text{posValida}(c, p)\}$

**otras operaciones**

$\text{adyacente} : \text{as } a \times \text{pos } pe \times \text{pos } pd \rightarrow \text{bool} \quad \{\text{posValida}(c, pe) \wedge \text{posValida}(c, pd)\}$   
 $\text{posValida} : \text{as } a \times \text{pos } p \rightarrow \text{bool}$   
 $\text{posIngreso} : \text{as } a \times \text{pos } p \rightarrow \text{bool}$   
 $\text{minDistsPos} : \text{campus } c \times \text{pos } p \times \text{conj}(\text{pos}) \text{ posiciones} \rightarrow \text{conj}(\text{pos}) \quad \{\neg(\emptyset?(posiciones))\}$   
 $\text{minDist} : \text{campus } c \times \text{pos } p \times \text{conj}(\text{posiciones}) \text{ posiciones} \rightarrow \text{nat} \quad \{\neg(\emptyset?(posiciones))\}$   
 $\text{distMan} : \text{campus } c \times \text{pos } p1 \times \text{pos } p2 \rightarrow \text{nat}$   
 $\text{abs} : \text{nat} \times \text{nat} \rightarrow \text{nat}$

**axiomas**  $\forall alto:\text{nat}, \forall ancho:\text{nat}, \forall obst:\text{conj}(\text{pos})$   
 $\forall p1:\text{pos} \forall p2:\text{pos}$

$\text{alto}(\text{nuevo}(\text{ancho}, \text{alto}, \text{obst})) \equiv alto$   
 $\text{ancho}(\text{nuevo}(\text{ancho}, \text{alto}, \text{obst})) \equiv ancho$   
 $\text{obstaculos}(\text{nuevo}(\text{ancho}, \text{alto}, \text{obst})) \equiv obst$   
 $\text{posValida}(\text{nuevo}(\text{ancho}, \text{alto}, \text{obst}), p1) \equiv \pi_1(p1) < ancho \wedge \pi_2(p1) < alto$   
 $\text{adyacente}(\text{nuevo}(\text{ancho}, \text{alto}, \text{obst}), p1, p2) \equiv (\pi_1(p1) = \pi_1(p2) - 1 \vee \pi_1(p1) = \pi_1(p2) + 1) \wedge$   
 $(\pi_2(p1) = \pi_2(p2) - 1 \vee \pi_2(p1) = \pi_2(p2) + 1)$   
 $\text{posValida}(\text{nuevo}(\text{ancho}, \text{alto}, \text{obst}), p1) \equiv \pi_2(p1) = alto - 1 \vee \pi_2(p1) = 0$   
 $\text{minDistsPos}(c, p, \text{posiciones}) \equiv \text{if } \emptyset?(\text{sinUno}(\text{posiciones})) \text{ then}$   
 $\quad \text{dameUno}(\text{posiciones})$   
 $\text{else}$   
 $\quad \text{if } \text{distMan}(c, p, \text{dameUno}(\text{posiciones})) \leq$   
 $\quad \text{minDist}(c, p, \text{posiciones}) \text{ then}$   
 $\quad \quad \text{Ag}(\text{minDistsPos}(c, \text{sinUno}(\text{posiciones})),$   
 $\quad \quad \text{dameUno}(\text{posiciones}))$   
 $\quad \text{else}$   
 $\quad \quad \text{minDistsPos}(c, \text{seg}, \text{sinUno}(\text{posiciones}))$   
 $\quad \text{fi}$   
 $\text{fi}$   
 $\text{minDist}(c, p, \text{posiciones}) \equiv \text{if } \emptyset?(\text{sinUno}(\text{posiciones})) \text{ then}$   
 $\quad \text{distMan}(c, p, \text{dameUno}(\text{posiciones}))$   
 $\text{else}$   
 $\quad \text{if } \text{distMan}(c, p, \text{dameUno}(\text{posiciones})) \leq$   
 $\quad \text{minDist}(c, \text{pos}/p, \text{sinUno}(\text{posiciones}))$   
 $\quad \text{then}$   
 $\quad \quad \text{distMan}(c, p, \text{dameUno}(\text{posiciones}))$   
 $\quad \text{else}$   
 $\quad \quad \text{minDist}(c, p, \text{sinUno}(\text{posiciones}))$   
 $\quad \text{fi}$   
 $\text{fi}$   
 $\text{distMan}(c, p1, p2) \equiv \text{restaAbs}(\pi_2(p1), \pi_2(p2)) + \text{restaAbs}(\pi_1(p1), \pi_1(p2))$   
 $\text{restaAbs}(n1, n2) \equiv \text{if } n2 > n1 \text{ then } n2 - n1 \text{ else } n1 - n2 \text{ fi}$

**Fin TAD**