

# 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>4</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  $a \times \text{pos } p \rightarrow \text{bool}$

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

posSeg : as  $a \times \text{seg } s \rightarrow \text{pos}$

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

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

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

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

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

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

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

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

**axiomas**  $\forall dc:dcnet, \forall r:red, \forall p_1, p_2:paqueteID, \forall c_1, c_2, c_3:compuID,$   
 $\forall camino:secu(tupla(compuID, interfaz),$   
 $\forall cpag:conj(paqueteID)$

red(nueva(r))

$$\equiv r$$

**Fin TAD**

## 2. TAD CAMPUS

### TAD CAMPUS

**géneros**      campus

**usa**            CAMPUS

**exporta**

**observadores básicos**

alto : campus  $\longrightarrow$  nat

ancho : campus  $\longrightarrow$  nat

obstaculos : campus  $\longrightarrow$  conj(pos)

**generadores**

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

**otras operaciones**

adyacente : as *a*  $\times$  pos *pe*  $\times$  pos *pd*  $\longrightarrow$  bool  $\{posValida(c, pe) \wedge posValida(c, pd)\}$

posValida : as *a*  $\times$  pos *p*  $\longrightarrow$  bool

posIngreso : as *a*  $\times$  pos *p*  $\longrightarrow$  bool

**axiomas**       $\forall alto:nat, \forall ancho:nat, \forall obst:conj (pos)$   
 $\forall p_1:pos \forall p_2:pos$

alto(nuevo(*ancho*, *alto*, *obst*))  $\equiv alto$

ancho(nuevo(*ancho*, *alto*, *obst*))  $\equiv ancho$

obstaculos(nuevo(*ancho*, *alto*, *obst*))  $\equiv obst$

posValida(nuevo(*ancho*, *alto*, *obst*), *p*<sub>1</sub>)  $\equiv \pi_1(p_1) < ancho \wedge \pi_2(p_1) < alto$

adyacente(nuevo(*ancho*, *alto*, *obst*), *p*<sub>1</sub>, *p*<sub>2</sub>)  $\equiv (\pi_1(p_1) = \pi_1(p_2) - 1 \vee \pi_1(p_1) = \pi_1(p_2) + 1) \wedge$   
 $(\pi_2(p_1) = \pi_2(p_2) - 1 \vee \pi_2(p_1) = \pi_2(p_2) + 1)$

posValida(nuevo(*ancho*, *alto*, *obst*), *p*<sub>1</sub>)  $\equiv \pi_2(p_1) = alto - 1 \vee \pi_2(p_1) = 0$

**Fin TAD**