Algoritmos y Estructuras de Datos II

Primer Cuatrimestre de 2015

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

Trabajo Prï¿½ctico 1

Especificaciï¿½n

Integrante	LU	Correo electrónico
INTEGRANTE, 1	123/12	1@gmail.com
INTEGRANTE, 2	123/12	2@gmail.com
INTEGRANTE, 3	123/12	3@gmail.com
INTEGRANTE, 4	123/12	4@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

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 $\longrightarrow$ campus

hayHippie? : as a × pos p $\longrightarrow$ bool $\{posValida(campus(a), p)\}$

#capturas : as a × seg s $\longrightarrow$ nat $\{s \in seguridad(a)\}$

generadores

nueva : campus × conj(seguridad) $\longrightarrow$ as
{(∀ *segs*:e) posValida(c,pos(e)) ∧ (∀ *segs*:s,s1) id(s)!=id(s1) ⇒ pos(s)!=pos(s1)}

sacarEst : as a × pos p $\longrightarrow$ as $\{posValida(campus(a), p) \wedge_{\text{L}} hayEst?(a, p) \wedge posIngreso(a, p)\}$

otras operaciones

haySeg? : as a × pos p $\longrightarrow$ bool

proximasPosiciones : as a × conj(pos) *minPos* × pos *posAct* $\longrightarrow$ conj(pos)
$\{\neg(emptyset?(minPos)) \wedge_{\text{L}} posValida(campus(a), posAct) \wedge posicionesValidas(campus(a), minPos)\}$

axiomas

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

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

#capturas(sacarEst(a, p_1),s) $\equiv$ $\#capturas(a, s)$

#capturas(a,moverSeg(a, s, p_1)) $\equiv$ $\beta(posValida(campus(a), < \pi_1(p_1) + 1, \pi_2(p_1) >) \wedge_{\text{L}} (hayHippie?(a, < \pi_1(p_1) + 1, \pi_2(p_1) >) \wedge_{\text{L}} encerrado(a, < \pi_1(p_1) + 1, \pi_2(p_1) >))) +$
$\beta(posValida(campus(a), < \pi_1(p_1) - 1, \pi_2(p_1) >) \wedge_{\text{L}} (hayHippie?(a, < \pi_1(p_1) - 1, \pi_2(p_1) >) \wedge_{\text{L}} encerrado(a, < \pi_1(p_1) - 1, \pi_2(p_1) >))) +$
$\beta(posValida(campus(a), < \pi_1(p_1), \pi_2(p_1) + 1 >) \wedge_{\text{L}} (hayHippie?(a, < \pi_1(p_1), \pi_2(p_1) + 1 >) \wedge_{\text{L}} encerrado(a, < \pi_1(p_1), \pi_2(p_1) + 1 >))) +$
$\beta(posValida(campus(a), < \pi_1(p_1), \pi_2(p_1) - 1 >) \wedge_{\text{L}} (hayHippie?(a, < \pi_1(p_1), \pi_2(p_1) - 1 >) \wedge_{\text{L}} encerrado(a, < \pi_1(p_1), \pi_2(p_1) - 1 >))) +$
$\#capturas(a, s)$

moverTodos(a,segs) $\equiv$ **if** $(\emptyset?(segs))$ **then**
$\emptyset$
else
if $(hayHippies?(a))$ **then**
$Ag(moverTodos(a, sinUno(segs)),$
$moverSeg(a, dameUno(segs)),$
$dameUno(proxPosiciones$
$(hippiesMasCerca(a, dameUno(segs))))))$
else
$segs$
fi
fi

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
$\{1 \leq ancho \wedge 1 \leq alto \wedge (\forall\ p\text{:pos})\ p \in obst \Rightarrow_{\text{L}} posValida(c,p)\}$

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

minDistsPos : campus c $\times$ pos p $\times$ conj(pos) *posiciones* $\longrightarrow$ conj(pos) $\{\neg(\emptyset?(posiciones))\}$

minDist : campus c $\times$ pos p $\times$ conj(posiciones) *posiciones* $\longrightarrow$ nat $\{\neg(\emptyset?(posiciones))\}$

distMan : campus c $\times$ pos $p1$ $\times$ pos $p2$ $\longrightarrow$ nat

restaAbs : nat $\times$ nat $\longrightarrow$ nat

conjPos : campus $\times$ nat $\times$ nat $\longrightarrow$ conj(pos)

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_1) $\equiv$ $\pi_1(p_1) < ancho \wedge \pi_2(p_1) < alto$

adyacente(nuevo(*ancho*,*alto*,*obst*),p_1,p_2) $\equiv$ $(\pi_1(p_1) = \pi_1(p_2) - 1 \vee \pi_1(p_1) = \pi_1(p_2) + 1) \wedge (\pi_2(p_1) = \pi_2(p_2) - 1 \vee \pi_2(p_1) = \pi_2(p_2) + 1)$

posValida(nuevo(*ancho*,*alto*,*obst*),p_1) $\equiv$ $\pi_2(p_1) = alto - 1 \vee \pi_2(p_1) = 0$

minDistsPos(c,p,posiciones) $\equiv$ **if** $\emptyset?(sinUno(posiciones))$ **then**
$dameUno(posiciones)$
else
if $distMan(c,p,dameUno(posiciones)) \leq minDist(c,p,posiciones)$ **then**
$Ag(minDistsPos(c,sinUno(posiciones)), dameUno(posiciones))$
else
$minDistsPos(c,seg,sinUno(posiciones))$
fi
fi

minDist(c,p,posiciones) $\equiv$ **if** $\emptyset?(sinUno(posiciones))$ **then**
$distMan(c, p, dameUno(posiciones))$
else
if $distMan(c, p, dameUno(posiciones)) \leq minDist(c, pos/p, sinUno(posiciones))$
then
$distMan(c, p, dameUno(posiciones))$
else
$minDist(c, p, sinUno(posiciones))$
fi
fi

distMan(c,p_1,p_2) $\equiv$ $restaAbs(\pi_2(p_1), \pi_2(p_2))$+$restaAbs(\pi_1(p_1), \pi_1(p_2))$

restaAbs(n1,n2) $\equiv$ **if** $n2 > n1$ **then** $n2 - n1$ **else** $n1 - n2$ **fi**

conjPos(c,x,y) $\equiv$ **if** $x \geq ancho(c)$ **then**
$\emptyset$
else
if $y \geq alto(c)$ **then**
$conjPos(c, x + 1, 0)$
else
$ag(conjPos(c, x, y + 1), < x, y >)$
fi
fi

Fin TAD