



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico de Especificación

Algoritmos y Estructuras de Datos II - Grupo 36

Integrante	LU	Correo electrónico
Sebastián Silvera	680/17	sebaok2011@gmail.com
Luciano Zinik	290/17	lzinik@gmail.com
Martín Funes	342/16	martinfunesfunes@gmail.com
Fernando Regert	282/15	fernandostds9@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

1. Decisiones tomadas y Notas

EN ESTE TRABAJO, TOMAMOS LA SIGUIENTE DECISIÓN:

- Definimos el mapa como una tupla de *nats* desde la celda $\langle 0, 0 \rangle$, de modo que sus valores puedan aumentar hasta el infinito en forma de damero.

NOTAS:

- En el **TAD MAPA**:
 - El generador «nuevoMapa» recibe como parámetro un `dicc(nat x bool)`, este diccionario define los ríos, que, citando el enunciado: «*corren en líneas infinitas horizontales o verticales*». Los parámetros de este diccionario definen lo siguiente: el segundo indica si el río se encuentra en una fila o una columna, y el primero indica el número de la fila o columna.:
 - TRUE: Horizontal.
 - FALSE: Vertical.

Entonces, si es $\langle 3, \text{true} \rangle$, lo que nos dice es que toda la **fila** 3 de nuestra grilla, esta ocupada por un río. En cambio, si es $\langle 3, \text{false} \rangle$, nos dice que toda la **columna** 3 de nuestra grilla, estará ocupada por un río.

- La «otra operación» **casasADistManhattan** está axiomatizada sobre los generadores. Nos permitimos esto, ya que la información que nos brinda se puede deducir únicamente de los observadores, por lo cual podemos asegurar que no hay riesgo de romper la congruencia.

2. TADs

TAD Celda es $\langle \text{nat}, \text{nat} \rangle$

TAD Terreno es String

TAD Mapa

géneros mapa
usa conj, dicc, nat
exporta observadores, generadores
igualdad observacional

$$(\forall m, m' : \text{mapa}) \left(m =_{\text{obs}} m' \iff \left(\begin{array}{l} (\forall c : \text{celda}) \\ (\text{queHay?}(m, c) =_{\text{obs}} \text{queHay?}(m', c) \wedge_{\text{L}} \\ \neg (\text{queHay?}(m, c) \neq \text{río}) \Rightarrow_{\text{L}} \text{nivel}(m, c) =_{\text{obs}} \text{nivel}(m', c)) \end{array} \right) \right)$$

observadores básicos

<code>queHay?</code>	: mapa × celda	→ terreno	
<code>nivel</code>	: mapa <i>m</i> × celda <i>c</i>	→ nat	{ <code>queHay?(m, c) ≠ río</code> }

generadores

<code>nuevoMapa</code>	: dicc(nat × bool)	→ mapa	
<code>agregarCasa</code>	: mapa <i>m</i> × celda <i>c</i>	→ mapa	{ <code>queHay?(m, c) = baldío</code> }
<code>agregarComercio</code>	: mapa <i>m</i> × celda <i>c</i>	→ mapa	{ <code>queHay?(m, c) = baldío</code> }
<code>aumentarNivelEnUno</code>	: mapa	→ mapa	

otras operaciones

<code>hayCasa?</code>	: mapa × celda	→ bool	
<code>distanciaManhattan</code>	: celda × celda	→ nat	
<code>casasADistManhattan</code>	: mapa × celda	→ conj(celda)	
<code>maxNivel</code>	: mapa × conj(celda)	→ nat	
<code>nivelManhattan</code>	: mapa <i>m</i> × celda <i>c</i>	→ nat	{ <code>queHay?(m, c)</code> }

axiomas $\forall c, c1: \text{celda}, \forall A: \text{conjunto}(\text{celdas}), \forall m: \text{mapa}, \forall h: \text{bool}$	
$\text{queHay?}(\text{nuevoMapa}(\text{dicc}(n, h)), c)$	$\equiv \text{if } h \text{ then}$ $\quad \text{if } \text{prim}(c)=n \text{ then } \text{río} \text{ else } \text{baldío} \text{ fi}$ $\quad \text{else}$ $\quad \text{if } \text{seg}(c)=n \text{ then } \text{río} \text{ else } \text{baldío} \text{ fi}$ fi
$\text{queHay?}(\text{agregarCasa}(m, c1), c2)$	$\equiv \text{if } c1=c2 \text{ then } \text{casa} \text{ else } \text{queHay?}(m, c2) \text{ fi}$
$\text{queHay?}(\text{agregarComercio}(m, c1), c2)$	$\equiv \text{if } c1=c2 \text{ then } \text{comercio} \text{ else } \text{queHay?}(m, c2) \text{ fi}$
$\text{queHay?}(\text{aumentarNivelEnUno}(m), c)$	$\equiv \text{queHay?}(m, c)$
$\text{nivel}(\text{nuevoMapa}(d), c)$	$\equiv 0$
$\text{nivel}(\text{agregarCasa}(m, c), c1)$	$\equiv \text{if } c=c1 \text{ then } 0 \text{ else } \text{nivel}(m, c1) \text{ fi}$
$\text{nivel}(\text{agregarComercio}(m, c), c1)$	$\equiv \text{if } c=c1 \text{ then}$ $\quad \text{nivelManhattan}(m, c1)$ $\quad \text{else}$ $\quad \text{nivel}(m, c1)$ fi
$\text{nivel}(\text{aumentarNivelEnUno}(m), c)$	$\equiv \text{nivel}(m, c) + 1$
$\text{hayCasa?}(m, c)$	$\equiv \text{queHay?}(m, c) = \text{casa}$
$\text{distanciaManhattan}(c1, c2)$	$\equiv \max(\pi_1(c1), \pi_1(c2)) - \min(\pi_1(c1), \pi_1(c2))$ $\quad +$ $\quad \max(\pi_2(c1), \pi_2(c2)) - \min(\pi_2(c1), \pi_2(c2))$
$\text{casasADistManhattan}(\text{nuevoMapa}(\text{dicc}(n, h)), c)$	$\equiv \emptyset$
$\text{casasADistManhattan}(\text{agregarCasa}(m, c), c1)$	$\equiv \text{if } 0 < \text{distanciaManhattan}(c, c1) \leq 3 \text{ then}$ $\quad \text{Ag}(c, \text{casasADistManhattan}(m, c1))$ $\quad \text{else}$ $\quad \text{casasADistManhattan}(m, c1)$ fi
$\text{casasADistManhattan}(\text{agregarComercio}(m, c), c1)$	$\equiv \text{casasADistManhattan}(m, c1)$
$\text{casasADistManhattan}(\text{aumentarNivelEnUno}(m), c)$	$\equiv \text{casasADistManhattan}(m, c)$
$\text{maxNivel}(m, A)$	$\equiv \text{if } \emptyset?(A) \text{ then}$ $\quad 0$ $\quad \text{else}$ $\quad \max(\text{nivel}(m, \text{dameUno}(A)),$ $\quad \text{maxNivel}(m, \text{sinUno}(A)))$ fi
$\text{nivelManhattan}(m, c)$	$\equiv \text{maxNivel}(m, \text{casasADistManhattan}(m, c))$

Fin TAD

TAD Juego

géneros juego

igualdad observacional

$$(\forall j, j' : \text{juego}) \left(j =_{\text{obs}} j' \iff \left(\begin{array}{l} \text{turno}(j) =_{\text{obs}} \text{turno}(j') \wedge \\ \text{mapa?}(j) =_{\text{obs}} \text{mapa?}(j') \wedge \\ \text{hayConstNuevas?}(j) =_{\text{obs}} \text{hayConstNuevas?}(j') \end{array} \right) \right)$$

observadores básicos

turno	: juego	→ nat
mapa?	: juego	→ mapa
hayConstNuevas?	: juego	→ bool

generadores

nuevoJuego	: mapa	→ juego	
nuevaCasa	: juego j × celda c	→ juego	{queHay?(mapa?(j), c) = baldío}
nuevoComercio	: juego j × celda c	→ juego	{queHay?(mapa?(j), c) = baldío}
siguienteTurno	: juego j	→ juego	{hayConstNuevas?(j)}

axiomas $\forall c: \text{celda}, \forall j: \text{juego}, \forall m: \text{mapa}$

turno(nuevoJuego(m))	≡ 0
turno(nuevaCasa(j, c))	≡ turno(j)
turno(nuevoComercio(j, c))	≡ turno(j)
turno(siguienteTurno(j))	≡ turno(j) + 1

mapa?(nuevoJuego(m))	≡ m
mapa?(nuevaCasa(j, c))	≡ agregarCasa(mapa?(j), c)
mapa?(nuevoComercio(j, c))	≡ agregarComercio(mapa?(j), c)
mapa?(siguienteTurno(j))	≡ aumentarNiveles(mapa?(j))

hayConstNuevas?(nuevoJuego(m))	≡ false
hayConstNuevas?(nuevaCasa(j, c))	≡ true
hayConstNuevas?(nuevoComercio(j, c))	≡ true
hayConstNuevas?(siguienteTurno(j))	≡ false

Fin TAD