DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# TP de Especificación

**Buscaminas**

30 de Marzo de 2022 — Algoritmos y Estructuras de Datos I


| Integrante | LU | Correo electrónico |
|---|---|---|
| Aguirre, Victoria Inés | 626/19 | - |
| Bongiovanni, Franco Alessio | 344/19 | - |
| Begalli, Juan Martin | 139/22 | - |
| Fernández Lilli, Ignacio | 221/22 | - |






**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. Definición de Tipos

type $pos = \mathbb{Z} \times \mathbb{Z}$

type $tablero = seq\langle seq\langle \mathsf{Bool} \rangle \rangle$

type $jugadas = seq\langle pos \times \mathbb{Z} \rangle$

type $banderitas = seq\langle pos \rangle$

# 2. Auxiliares Utilizadas

`aux fila` (x: $pos \times \mathbb{Z}$) : $\mathbb{Z} = (x_0)_0$ ;

`aux columna` (x: $pos \times \mathbb{Z}$) : $\mathbb{Z} = (x_0)_1$ ;

# 3. Problemas

## Ejercicio 1

`aux minasAdyacentes` (t: *tablero*, p: *pos*) : $\mathbb{Z} =$
$\sum_{i=p_0-1}^{p_0+1} \sum_{j=p_1-1}^{p_1+1}$ if $((0 \leq i < |t| \land 0 \leq j < |t|) \land_L ((i,j) \neq (p_0,p_1) \land t[i][j] = true))$ then 1 else 0 fi ;

## Ejercicio 2

`pred juegoValido` (t: *tablero*, j: *jugadas*) {

$tableroValido(t) \land posicionesDeJugadasValidas(t,j) \land noHayJugadasRepetidas(j) \land$
$(cantidadDeMinasJugadas(t,j) \leq 1) \land minasAdyacentesAJugadasSonCorrectas(j,t)$

}

`pred minasAdyacentesAJugadasSonCorrectas` (j:*jugadas*, t:*tablero*) {

$(\forall i : \mathbb{Z})(0 \leq i < |j| \longrightarrow_L minasAdyacentes(t, j[i]_0) = j[i]_1)$

}

`pred posicionesDeJugadasValidas` (t: *tablero*, j: *jugadas*) {

$(\forall i : \mathbb{Z})(0 \leq i < |j| \longrightarrow_L (0 \leq fila(j[i]) < |t| \land 0 \leq columna(j[i]) < |t|))$

}

`pred noHayJugadasRepetidas` (j: *jugadas*) {

$(\forall i,k : \mathbb{Z})((0 \leq i < |j| \land 0 \leq k < |j| \land i \neq k) \longrightarrow_L (j[i])_0 \neq (j[k])_0)$

}

`aux cantidadDeMinasJugadas` (t: *tablero*, j: *jugadas*) : $\mathbb{Z} =$
$\sum_{i=0}^{|j|-1}$ if $t[fila(j[i])][columna(j[i])] = true$ then 1 else 0 fi ;

`pred tableroValido` (t:*tablero*) {

$|t| > 0 \land (\forall i : \mathbb{Z})(0 \leq i < |t| \longrightarrow_L |t[i]| = |t|)$

}

## Ejercicio 3

proc plantarBanderita (in t: *tablero*, in j: *jugadas*, in p: *pos*, inout b: *banderitas*) {

Pre $\{juegoValido(t, j) \land b = b_0 \land tableroValido(t) \land casilleroValida(p, b_0, j, t) \land banderitasValidas(b_0, t, j)\}$

Post $\{|b| = |b_0| + 1 \land \neg noHayBanderita(b, p) \land contenido(b, b_0)\}$

}

pred casilleroValido (p: *pos*, b: *banderitas*, j: *jugadas*, t: *tablero*) {

$noHayBanderita(b, p) \land noSeJugo(p, j) \land posicionValida(p, t)$

}

pred noHayBanderita (b: *banderitas*, p: *pos*) {

$(\forall i : \mathbb{Z})(0 \leq i < |b| \longrightarrow_L b[i] \neq p)$

}

pred noSeJugo (p: *pos*, j: *jugadas*) {

$(\forall i : \mathbb{Z})(0 \leq i < |j| \longrightarrow_L [j[i]]_0 \neq p)$

}

pred posicionValida (p:*pos*, t:*tablero*) {

$0 \leq p_0 < |t| \land 0 \leq p_1 < |t|$

}

pred banderitasValidas (b: *banderitas*, t : *tablero*, j:*jugadas*) {

$noHayBanderitasRepetidas(b) \land (\forall i : \mathbb{Z})(0 \leq i < |b| \longrightarrow_L (posicionValida(b[i], t) \land noSeJugo(b[i], j)))$

}

pred noHayBanderitasRepetidas (b: *banderitas*) {

$(\forall i, k : \mathbb{Z})((0 \leq i < |b| \land 0 \leq k < |b| \land i \neq k) \longrightarrow_L b[i] \neq b[k])$

}

pred contenido (b: *banderitas*, $b_0$:*banderitas*) {

$(\forall k : \mathbb{Z})(0 \leq k < |b_0| \longrightarrow_L (\exists j : \mathbb{Z})(0 \leq j < |b| \land_L b[j] = b_0[k]))$

}

## Ejercicio 4

proc Perdió (in t: *tablero*, in j: *jugadas*, out res: Bool) {

Pre $\{|j| > 0 \land juegoValido(t, j)\}$

Post $\{res = true \leftrightarrow jugoMina(t, j)\}$

}

pred jugoMina (t:*tablero*, j: *jugadas*) {

$(\exists h : \mathbb{Z})(0 \leq h < |j| \land_L t[fila(j[h])][columna(j[h])] = \text{true})$

}

## 3.1. Ejercicio 5

proc gano (in t: *tablero*, in j: *jugadas*, out res: Bool) {

Pre $\{|j| > 0 \land juegoValido(t, j)\}$

Post $\{res = true \leftrightarrow ganoElJuego(t, j)\}$

}

pred ganoElJuego (t:*tablero*, j: *jugadas*) {

$\neg jugoMina(t, j) \land |j| = |t| * |t| - cantidadDeMinas(t)$

}

aux cantidadDeMinas (t: *tablero*) : $\mathbb{Z} = \sum_{i=0}^{|t|-1} \sum_{j=0}^{|t|-1}$ if $t[i][j] = true$ then 1 else 0 fi ;

## Ejercicio 6

proc jugar (in t: *tablero*, in b: *banderitas*, in p: *pos*, inout j: *jugadas*) {

Pre $\{j = j_0 \land banderitasValidas(b, t, j) \land juegoValido(t, j) \land$
$\neg ganoElJuego(t, j) \land \neg jugoMina(t, j) \land casilleroValido(p, b, j_0, t)\}$

Post $\{|j| = |j_0| + 1 \land seAgregoPosicionJugada(j, p, t) \land seActualizoJugadas(j, j_0)\}$

}

pred seActualizoJugadas (j:jugadas, $j_0$:jugadas) {

$(\forall i : \mathbb{Z})(0 \leq i < |j_0| \longrightarrow_L (\exists k : \mathbb{Z})(0 \leq k < |j| \land_L j[k] = j_0[i])$

}

pred seAgregoPosicionJugada (j:*jugadas*, p:*pos*, t:*tablero*) {

$(\exists i : \mathbb{Z})(0 \leq i < |j| \land_L j[i] = (p, minasAdyacentes(t, p)))$

}

## Ejercicio 7

Aclaracion: En este ejercicio el camino puede contener banderitas porque como banderitas no esta considerado (en la consigna), en CaminoLibre no lo chequeamos. Por la misma razon tampoco considero la lista de jugadas, por lo que podrian aparecer posiciones ya jugadas.

pred CaminoLibre (t:*tablero*, p: *pos*, p': *pos*) {

$minasAdyacentes(t, p) = 0 \land minasAdyacentes(t, p') \geq 1 \land existeCamino(t, p, p')$

}

pred existeCamino (t:*tablero*, p: *pos*, p': *pos*) {

$(\exists c : seq\langle pos\rangle)(esCaminoValido(c, t, p, p'))$

}

pred esCaminoValido (c: $seq\langle pos\rangle$,t:*tablero*, p: *pos*, p': *pos*) {

$\neg sonAdyacentes(p, p') \land_L (secuenciaDeAdyacentes(c) \land sinMinasAdyacentes(c, t) \land sonAdyacentes(p, c[0]) \land$
$sonAdyacentes(p', c[|c| - 1]) \land posicionesValidas(c, t))$

}

pred sonAdyacentes (p:*pos*,p':*pos*) {

$sonAdyacentesArriba(p, p') \lor sonAdyacentesAbajo(p, p') \lor sonAdyacentesAlosCostados(p, p')$

}

pred sonAdyacentesAbajo (p:*pos*,p':*pos*) {

$(p'_0 = p_0 + 1) \land (p_1 - 1 \leq p'_1 \leq p_1 + 1)$

}

pred sonAdyacentesArriba (p:*pos*,p':*pos*) {

$(p'_0 = p_0 - 1) \land (p_1 - 1 \leq p'_1 \leq p_1 + 1)$

}

pred sonAdyacentesAlosCostados (p:*pos*,p':*pos*) {

$(p'_0 = p_0) \land (p_1 - 1 = p'_1 \lor p_1 + 1 = p'_1)$

}

pred secuenciaDeAdyacentes (c: $seq\langle pos \rangle$) {

$(\forall i : \mathbb{Z})(0 \leq i < |c| - 1 \longrightarrow_L sonAdyacentes(c[i], c[i+1]))$

}

pred sinMinasAdyacentes (c: $seq\langle pos \rangle$,t:*tablero*) {

$(\forall i : \mathbb{Z})(0 \leq i < |c| \longrightarrow_L minasAdyacentes(c[i], t) = 0)$

}

pred posicionesValidas (c: $seq\langle pos \rangle$,t:*tablero*) {

$(\forall i : \mathbb{Z})(0 \leq i < |c| \longrightarrow_L (posicionValida(c[i], t) \land t[c[i]_0][c[i]_1] = false))$

}

## Ejercicio 8

Aclaracion: La consigna nos pide que se descubran solamente los casilleros SIN minas adyacentes. Por lo tanto, no se deberian descubrir las casillas que tienen por lo menos 1 mina adyacente. Aunque esto no es lo que usualmente ocurre en los buscaminas (donde se descubre hasta la casilla con por lo menos 1 mina inclusive) nosotros seguimos lo que decia la consigna.

proc jugarPlus (in t: *tablero*, in b: *banderitas*, in p: *pos*, inout j: *jugadas*) {

Pre $\{j = j_0 \land banderitasValidas(b, t, j) \land juegoValido(t, j) \land \neg ganoElJuego(t, j) \land \neg jugoMina(t, j) \land casilleroValido(p, b, j_0, t)\}$

Post $\{|j| > |j_0| \land seAgregoPosicionJugada(j, p, t) \land seActualizoJugadas(j, j_0) \land juegoValido(t, j) \land seDescubrenCaminos(t, j, p, b) \land noHayJugadasDeMás(j, j_0, p, b, t)\}$

}

pred seDescubrenCaminos (t: *tablero*, j: *jugadas*, p: *pos*, b: *banderitas*) {

$descubreCasillasSinMinasAdy(t, j, p, b) \longrightarrow_L (minasAdyacentes(t, p) = 0)$

}

pred descubreCasillasSinMinasAdy (t: *tablero*, j: *jugadas*, p: *pos*, b:*banderitas*) {

$(\forall p' : pos)(esFinalDeCamino(p', b, t) \longrightarrow_L agregaCaminoLibreAJugadas(j, t, p, p', b))$

}

pred esFinalDeCamino (p': *pos*, b: *banderitas*, t: *tablero*) {

$noHayBanderita(b, p') \land posicionValida(p', t) \land minasAdyacentes(t, p') \geq 1$

}

pred agregaCaminoLibreAJugadas (j: *jugadas*, t: *tablero*, p: *pos*, p': *pos*, b: *banderitas*) {

$(\forall c : seq\langle pos \rangle)(esCaminoValido(c, t, p, p') \longrightarrow_L todoElCaminoEstaEnJugadas(j, c, t, b))$

}

pred todoElCaminoEstaEnJugadas (j: *jugadas*, c: $seq\langle pos \rangle$, t: *tablero*, b: *banderitas*) {

$(\forall i : \mathbb{Z})((0 \leq i < |c| \land casilleroValido(c[i], b, j, t)) \longrightarrow_L seAgregoPosicionJugada(j, c[i], t))$

}

pred noHayJugadasDeMás (j: *jugadas*, $j_0$ : *jugadas*, p: *pos*, b: *banderitas*, t: *tablero*) {

$(\forall k : pos \times \mathbb{Z})(k \in j \longrightarrow_L (k \in j_0 \lor k_0 = p \lor esParteDeUnCaminoValido(j, k_0, p, b, t)))$

}

pred esParteDeUnCaminoValido (j: *jugadas*, i: *pos*, p: *pos*, b: *banderitas*, t: *tablero*) {

$(\exists p' : pos)(esFinalDeCamino(p', b, t) \wedge_L (\exists c : seq\langle pos\rangle)(i \in c \wedge_L esCaminoValido(c, t, p, p')))$

}

## Ejercicio 9

proc sugerirAutomatico121 (in t: *tablero*, in b: *banderitas*, in j: *jugadas*, out p: *pos*) {

Pre $\{juegoValido(t, j) \wedge banderitasValidas(b, t, j) \wedge$
$(\exists i, m, k : \mathbb{Z})((0 \leq i < |j| \wedge 0 \leq m < |j| \wedge 0 \leq k < |j| \wedge i \neq m \neq k) \wedge_L hayPatron121(j, i, m, k))\}$

Post $\{casilleroValido(p, b, j, t) \wedge esPosicionAdyacenteAlPatron(j, p)\}$

}

pred hayPatron121 (j: *jugadas*, i, m, k: $\mathbb{Z}$) {

$patron121Horizontal(j, i, m, k) \vee_L patron121Vertical(j, i, m, k)$

}

pred patron121Horizontal (j: *jugadas*, i, m, k : $\mathbb{Z}$) {

$mismaFila(j[i], j[m], j[k]) \wedge_L columnasConsecutivas(j[i], j[m], j[k])) \wedge_L ((j[i])_1 = 1 \wedge (j[m])_1 = 2 \wedge (j[k])_1 = 1)$

}

pred mismaFila (x, y, z : $pos \times \mathbb{Z}$) {

$fila(x) = fila(y) \wedge fila(y) = fila(z)$

}

pred columnasConsecutivas (x, y, z : $pos \times \mathbb{Z}$) {

$columna(x) = columna(y) - 1 \wedge columna(y) = columna(z) - 1$

}

pred patron121Vertical (j: *jugadas*, i, m, k : $\mathbb{Z}$) {

$mismaColumna(j[i], j[m], j[k]) \wedge_L filasConsecutivas(j[i], j[m], j[k]) \wedge_L ((j[i])_1 = 1 \wedge (j[m])_1 = 2 \wedge (j[k])_1 = 1)$

}

pred mismaColumna (x, y, z : $pos \times \mathbb{Z}$) {

$columna(x) = columna(y) \wedge columna(y) = columna(z)$

}

pred filasConsecutivas (x, y, z : $pos \times \mathbb{Z}$) {

$fila(x) = fila(y) - 1 \wedge fila(y) = fila(z) - 1$

}

pred esPosicionAdyacenteAlPatron (j: *jugadas*, p: *pos*) {

$(\exists i, m, k : \mathbb{Z})((0 \leq i < |j| \wedge 0 \leq m < |j| \wedge 0 \leq k < |j| \wedge i \neq m \neq k) \wedge_L hayPatron121(j, i, m, k) \wedge_L$
$jugadaSugeridaValida(p, (j[i])_0, (j[m])_0, (j[k])_0) \wedge_L \neg esDiagonalAl2(p, (j[m])_0))$

}

pred jugadaSugeridaValida (p: *pos*, x, y, z: *pos*) {

$sonAdyacentes(p, x) \vee sonAdyacentes(p, y) \vee sonAdyacentes(p, z)$

}

pred esDiagonalAl2 (p: *pos*, x: *pos*) {

$(p_0 = x_0 - 1 \wedge (p_1 = x_1 - 1 \vee p_1 = x_1 + 1)) \vee (p_0 = x_0 + 1 \wedge (p_1 = x_1 - 1 \vee p_1 = x_1 + 1))$

}