



# Trabajo Practico de Especificación

## Buscaminas

23 de septiembre de 2022

Algoritmos y Estructuras de Datos I

### Grupo martinlau

Integrante	LU	Correo electrónico
Santillan, Lautaro	370/22	lautisantil@gmail.com
Campoverde, Axel	258/22	a.i.cpv3000@gmail.com
Menalled, Martín	339/22	martinmenalled@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. Definición de Tipos

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

# 2. Constantes

## Auxiliares

aux cantidadDeMinasEnJugadas (t:tablero, j:jugadas) :  $\mathbb{Z} = \sum_{i=0}^{|j|-1} \text{if } (t[x(j[i])][y(j[i])] = true) \text{ then } 1 \text{ else } 0 \text{ fi};$   
aux cantidadDeMinasTablero ( t : tablero) :  $\mathbb{Z} = \sum_{i=0}^{|t|-1} \sum_{j=0}^{|t|-1} \text{if } (t[i][j] = true) \text{ then } 1 \text{ else } 0 \text{ fi};$   
aux x (jugada: pos $\times\mathbb{Z}$ ) :  $\mathbb{Z} = (jugada_0)_0;$   
aux y (jugada: pos $\times\mathbb{Z}$ ) :  $\mathbb{Z} = (jugada_0)_1;$

## Predicados

pred esAdyacente (x: pos, p:pos) {  
    (*filaAdyacente*(x, p)  $\wedge$  ((*columnaAdyacente*(x, p)  $\vee$  *enMismaColumna*(x, p)))  $\vee$   
    (*enMismaFila*(x, p)  $\wedge$  *columnaAdyacente*(x, p))  
}  
pred enMismaFila (p : pos, q : pos) {  
     $x_0 = q_0$   
}  
pred enMismaColumna (p : pos, q : pos) {  
     $p_1 = q_1$   
}  
pred filaAdyacente (p : pos, q : pos) {  
     $|q_0 - p_0| = 1$   
}  
pred columnaAdyacente (p : pos, q : pos) {  
     $|q_1 - p_1| = 1$   
}  
pred posicionEsValida (t: tablero, p: pos) {  
     $0 \leq p_0 < |t| \wedge 0 \leq p_1 < |t|$   
}  
pred repiteJugada (j:jugadas) {  
     $(\forall i : \mathbb{Z})((0 \leq i < |j|) \longrightarrow_L (\exists k : \mathbb{Z})((0 \leq k \leq |j|) \wedge (i \neq k)) \wedge_L (x(j[i]) = x(j[k]) \wedge y(j[i]) = y(j[k])))$   
}  
pred predPerdio (t : tablero, j : jugadas) {  
    *cantidadDeMinasEnJugadas*(t, j) = 1  
}  
pred predGano (t : tablero, j : jugadas) {  
     $|t| * |t| - \text{cantidadDeMinasTablero}(t) = |j|$

```

}
pred tableroValido (t : tablero) {
  (∀i : ℤ)((0 ≤ i < |t|) →L (|t[i]| = |t|))
}
pred posEnJugadas (p: pos, j:jugadas) {
  (∃i : ℤ)((0 ≤ i < |j|) ∧L (x(j[i]) = p0 ∧ y(j[i]) = p1))
}
pred guardarPosYMinasEnJugadas (t:tablero, p: pos, j:jugadas) {
  (∃i : ℤ)((0 ≤ i < |j|) ∧L (x(j[i]) = p0 ∧ y(j[i]) = p1) ∧ minasAdyacentes(t, p) = j[i]1)
}
pred posEnBanderitas (p: pos, b:banderitas) {
  (∃i : ℤ)((0 ≤ i < |b|) ∧L ((b[i])0 = p0 ∧ (b[i])1 = p1))
}
pred caminoLibrePlus (t : tablero, p0 : pos, p1 : pos) {
  tableroValido(t) ∧ posicionValida(t, p0) ∧ posicionValida(t, p1) ∧
  ((∃s : seq(pos))
  |s| > 1 ∧L (p0 = s[0] ∧ p1 = s[|s| - 1] ∧ minasAdyacentes(t, p0) = 0 ∧
  ((∀i : ℤ)(1 ≤ i < |s| - 1 →L (minasAdyacentes(t, s[i]) = 0 ∧ esAdyacente(s[i], s[i + 1])))))
  )
}
pred jugadasAnterioresGuardadas (j: jugadas, k: jugadas) {
  (∀i : ℤ)((0 ≤ i < |k|) →L (k[i] ∈ j))
}
pred existePatron121 (j: jugadas) {
  (∃m, n, k : ℤ)(0 ≤ m, n, k < |j| ∧ m ≠ n ≠ k) ∧L
  (enMismaFila(j[m], j[n]) ∧ enMismaFila(j[n], j[k]) ∧ (columnaAdyacente(j[m], j[n]) ∧ columnaAdyacente(j[n], j[k]))) ∨
  (enMismaColumna(j[m], j[n]) ∧ enMismaColumna(j[n], j[k]) ∧ filaAdyacente(j[m], j[n]) ∧ filaAdyacente(j[n], j[k])) ∧
  (j[m]1 = 1 ∧ j[k]1 = 1 ∧ j[n]1 = 2)
  )
}
pred dosMinasEnPatron121 (j : jugadas, n : ℤ) {
  (∃m, k : ℤ)(0 ≤ m, k < |j| ∧ m ≠ k) ∧L
  ((enMismaFila(j[m], j[n]) ∧ enMismaFila(j[n], j[k]) ∧ (columnaAdyacente(j[m], j[n]) ∧ columnaAdyacente(j[n], j[k]))) ∨
  (enMismaColumna(j[m], j[n]) ∧ enMismaColumna(j[n], j[k])) ∧ (filaAdyacente(j[m], j[n]) ∧ filaAdyacente(j[n], j[k]))) ∧
  (j[m]1 = 1 ∧ j[k]1 = 1 ∧ j[n]1 = 2)
  )
}
pred tresEnMismaFila (p : pos, j : jugadas, n : ℤ) {
  (p1 = y(j[n]) + 1 ∨ p1 = y(j[n]) - 1) ∧ p0 = x(j[n])
}
pred tresEnMismaColumna (p : pos, j : jugadas, n : ℤ) {
  (p0 = x(j[n]) + 1 ∨ p0 = x(j[n]) - 1) ∧ p1 = y(j[n])
}

```

### 3. Problemas

#### Parte 1: Juego Básico

##### Ejercicio 1

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

##### Ejercicio 2

```
pred juegoValido (t: tablero, j: jugadas) {  
     $\neg(\text{repiteJugada}(j)) \wedge (0 \leq \text{cantidadDeMinas}(t, j) \leq 1) \wedge$   
     $(\forall i : \mathbb{Z})(0 \leq i < |j|) \longrightarrow_L (\text{posicionEsValida}(t, j[i]_0) \wedge (j[i]_1 = \text{minasAdyacentes}(t, j[i]_0)))$   
}
```

##### Ejercicio 3

```
proc plantarBanderita (in t:tablero, in j:jugadas, in p: pos, inout b: banderitas) {  
    Pre {tableroValido(t)  $\wedge$  posicionEsValida(t, p)  $\wedge$   $\neg(\text{posEnJugadas}(p, j) \wedge \neg(\text{posEnBanderitas}(p, b)) \wedge b = b_0)$ }  
    Post {posEnBanderitas(p, b)  $\wedge$   $(\forall i : \mathbb{Z})(0 \leq i < |b_0|) \longrightarrow_L \text{posEnBanderitas}(b_0[i], b)$ }  
}
```

##### Ejercicio 4

```
proc perdio (in t: tablero, in j: jugadas, out res: Bool) {  
    Pre {juegoValido(t, j)  $\wedge$  tableroValido(t)}  
    Post {res = true  $\leftrightarrow$  predPerdio(t, j)}  
}
```

##### Ejercicio 5

```
proc gano (in i: tablero, in j: jugadas, out res: Bool) {  
    Pre {juegoValido(t, j)  $\wedge$  tableroValido(t)  $\wedge$   $\neg(\text{predPerdio}(t, j))$ }  
    Post {res = true  $\leftrightarrow$  predGano(t, j)}  
}
```

##### Ejercicio 6

```
proc jugar (in t : tablero, in b : banderitas, in p : pos, inout j : jugadas) {  
    Pre {tableroValido(t)  $\wedge$  juegoValido(t, j)  $\wedge$   $\neg(\text{predPerdio}(t, j)) \wedge \neg(\text{predGano}(t, j)) \wedge \neg(\text{posEnJugadas}(p, j))$   
     $\wedge \neg(\text{posEnBanderitas}(p, b)) \wedge j = j_0$ }  
    Post {jugadasAnterioresGuardadas(j, j0)  $\wedge$  guardarPosYMinasEnJugadas(t, p, j)}  
}
```

## Parte 2: Despejar los vacios

### Ejercicio 7

```
pred caminoLibre (t : tablero, p0 : pos, p1 : pos) {  
  tableroValido(t) ∧ posicionValida(t, p0) ∧ posicionValida(t, p1) ∧  
  ((∃ s : seq(pos))  
  |s| > 1 ∧L (p0 = s[0] ∧ p1 = s[|s| - 1] ∧ minasAdyacentes(t, p0) = 0 ∧ minasAdyacentes(t, p1) >= 1) ∧  
  ((∀ i : ℤ)(1 ≤ i < |s| - 1 →L (minasAdyacentes(t, s[i]) = 0 ∧ esAdyacente(s[i], s[i + 1])))))  
}
```

### Ejercicio 8

*/\*Consideramos para este ejercicio que se deben descubrir todas las casilla con camino libre a p. Tengan estas minas adyacentes o no. Para eso armamos un predicado caminolibrePlus.\*/*

```
proc jugarPlus (in t:tablero, in b: banderitas, in p:pos, inout j: jugadas) {  
  Pre {tableroValido(t) ∧ juegoValido(t, j) ∧ ¬(predPerdio(t, j)) ∧ ¬(predGano(t, j)) ∧ ¬(posEnJugadas(p, j))  
  ∧ ¬(posEnBanderitas(p, b)) ∧ j = j0}  
  Post {jugadasAnterioresGuardadas(j, j0) ∧ guardarPosYMinasEnJugadas(t, p, j) ∧  
  ((∀ r : pos)((posicionValida(t, r) ∧ caminolibrePlus(t, p, r) ∧ ¬(posEnJugadas(r, j0)) ∧ ¬(posEnBanderitas(r, b)) →  
  guardarPosYMinasEnJugadas(t, r, j)))}  
}
```

### Ejercicio 9

```
proc sugerirAutomatico (in j : jugadas, in t : tablero, in b : banderitas, out p : pos) {  
  Pre {tableroValido(t) ∧ juegoValido(t, j) ∧ existePatron121(j)}  
  Post {¬(posEnBanderitas(p, b)) ∧ ¬(posEnJugadas(p, j)) ∧ posicionValida(t, p) ∧  
  (∃ n : ℤ)(0 ≤ n < |j| ∧L dosMinasEnPatron121(j, n) ∧ (tresEnMismaFila(p, j, n) ∨ tresEnMismaColumna(p, j, n))))}  
}
```