



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

Juego de la vida toroidal

25 de septiembre de 2018

Algoritmos y Estructuras de Datos I

Grupo: Java the Hutt;

Integrante	LU	Correo electrónico
Pomsztein, Vladimir	364/18	blastervla@gmail.com
Zinik, Luciano	290/17	lzinik@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. Problemas

```

proc esValido (in t: toroide, out result: Bool) {
  Pre {true}
  Post {result = true ↔ esToroideValido(t)}
}

proc posicionesVivas (in t: toroide, out vivas: seq⟨ℤ × ℤ⟩) {
  Pre {esToroideValido(t)}
  Post {(∀i, j : ℤ)(enRango(t, i, j) →L ((i, j) ∈ vivas ↔ estaViva(t[i][j])))}
}

proc densidadPoblacion (in t: toroide, out result: ℝ) {
  Pre {esToroideValido(t)}
  Post {result = cantidadVivas(t)/cantidadTotal(t)}
  aux cantidadTotal (t: toroide) : ℤ = filas(t) × columnas(t);
}

proc evolucionDePosicion (in t: toroide, in posicion: ℤ × ℤ, out result: Bool) {
  Pre {esToroideValido(t) ∧L enRango(t, posicion0, posicion1)}
  Post {result = true ↔ vivaLuegoDeEvolucion(t, posicion)}
}

proc evolucionToroide (inout t: toroide) {
  Pre {esToroideValido(t)}
  Post {t = te ↔ te = evolucionarToroide(t)}
}

proc evolucionMultiple (in t: toroide, in k: ℤ, out result: toroide) {
  Pre {esToroideValido(t) ∧ k ≥ 0}
  Post {result = tk ↔ tk = evolucionarToroideMultiple(t, k)}
}

proc esPeriodico (in t: toroide, inout p: ℤ, out result: Bool) {
  Pre {esToroideValido(t)}
  Post {(result = true) ↔ (∃ks : seq⟨ℤ⟩)((∀i : ℤ)((0 < i < length(ks) - 1) →L ((ks[i - 1] > 0) ∧L (ks[i - 1] < ks[i])) ∧L p = ks[0]))}
}

proc primosLejanos (in t1: toroide, in t2: toroide, out primos: Bool) {
  Pre {esToroideValido(t1) ∧ esToroideValido(t2)}
  Post {primos = true ↔ ((∃k : ℤ)((k > 0) ∧L ((t1 = evolucionarToroideMultiple(t2, k)) ∨L (t2 = evolucionarToroideMultiple(t1, k)))))}
}

proc seleccionNatural (in ts: seq⟨toroide⟩, out res: ℤ) {
  Pre {todosToroidesValidos(ts) ∧L algunToroideMuere(ts)}
  Post {res = is[0] ↔ (∃is : seq⟨ℤ⟩)((∀i : ℤ)((i ∈ is) ∧L (0 < i < length(is) - 1) →L (cantidadDeTicksHastaMuerte(ts[i-1]) ≥ cantidadDeTicksHastaMuerte(ts[i]))))}
  pred todosToroidesValidos (ts: seq⟨toroide⟩) {(∀t: toroide)((t ∈ ts) →L esToroideValido(t))}
  pred algunToroideMuere (ts: seq⟨toroide⟩) {(∃t: toroide)((t ∈ ts) ∧L (muere(t)))}
  aux cantidadDeTicksHastaMuerte (t: toroide) : ℤ =

```

```

    if ( $\exists i : \mathbb{Z}$ )(( $i \geq 0$ )  $\wedge_L$  cantidadVivas(evolucionarToroideMultiple( $t, i$ )) = 0) then  $i$ 
    else -1 fi;
  pred muere (t: toroide) {cantidadDeTicksHastaMuerte( $t$ )  $\neq$  -1}
}

proc fusionar (in t1: toroide, in t2: toroide, out res: toroide) {
  Pre {(esToroideValido( $t1$ ) $\wedge$ esToroideValido( $t2$ )) $\wedge_L$ (filas( $t1$ ) = filas( $t2$ ) $\wedge_L$ columnas( $t1$ ) =
    columnas( $t2$ ))}
  Post {res =  $tf \leftrightarrow (\exists tf : \text{toroide})((\text{filas}(tf) = \text{filas}(t1) \wedge_L \text{columnas}(tf) = \text{columnas}(t1)) \wedge_L$ 
    ( $\forall i, j : \mathbb{Z}$ )(enRango( $tf, i, j$ )  $\longrightarrow_L$  (estaViva( $tf, i, j$ )  $\leftrightarrow$  estaViva( $t1, i, j$ ) $\wedge_L$  estaViva( $t2, i, j$ ))))}
}

proc vistaTrasladada (in t1: toroide, in t2: toroide, out res: Bool) {
  Pre {(esToroideValido( $t1$ ) $\wedge$ esToroideValido( $t2$ )) $\wedge_L$ (filas( $t1$ ) = filas( $t2$ ) $\wedge_L$ columnas( $t1$ ) =
    columnas( $t2$ ))}
  Post {
    res = true  $\leftrightarrow (\exists i, j : \mathbb{Z})$ (
      ( $\forall x, y : \mathbb{Z}$ )(enRango( $t1, x, y$ )  $\longrightarrow_L$  ( $t1[x][y] = \text{valorPosicionNormalizada}(t2, (x +$ 
         $i, y + j))$ ))))}
}

proc enCrecimiento (in t: toroide, out res: Bool) {
  Pre {true}
  Post {true}
  aux Aux (i:  $\mathbb{Z}$ ) : Bool = true;
}

```

2. Predicados y Auxiliares generales

```

pred noEsVacia (t: toroide) {(length( $t$ ) > 0) $\wedge_L$ ( $\forall x : \text{seq}(\text{Bool})$ )( $(x \in t) \longrightarrow_L$  (length( $x$ ) > 0))}
pred esMatriz (t: toroide) {( $\forall x, y : \text{seq}(\text{Bool})$ )( $(x, y \in t) \longrightarrow_L$  (length( $x$ ) = length( $y$ )))}
pred esToroideValido (t: toroide) {(noEsVacia( $t$ )  $\wedge$  esMatriz( $t$ ))}
pred filas (t: toroide) {length( $t$ )}
pred columnas (t: toroide) {if filas( $t$ ) > 0 then length( $t[0]$ ) else 0 fi}
pred estaViva (x: Bool) {x = true}
pred enRango (t: toroide, i:  $\mathbb{Z}$ , j:  $\mathbb{Z}$ ) {if ( $0 \leq i < \text{filas}(t) \wedge_L (0 \leq j < \text{columnas}(t))$ ) then true else false fi}

aux cantidadVivas (t: toroide) :  $\mathbb{Z} = \sum_{i=0}^{\text{filas}(t)-1} ( \sum_{j=0}^{\text{columnas}(t)-1} \text{if } \text{estaViva}(t[i][j]) \text{ then } 1 \text{ else } 0 \text{ fi} )$ ;

pred vivaLuegoDeEvolucion (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) {seMantieneViva( $t, pos$ ) $\vee_L$ vivePorReproduccion( $t, pos$ )}
pred seMantieneViva (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) {estaViva( $t[\text{posicion}_0][\text{posicion}_1]$ )  $\wedge_L 2 \leq$ 
  vivasAdyacentes( $t, \text{posicion}$ )  $\leq 3$ }
pred vivePorReproduccion (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) {( $\neg$ estaViva( $t[\text{posicion}_0][\text{posicion}_1]$ ))  $\wedge_L$ 
  vivasAdyacentes( $t, \text{posicion}$ ) = 3}
aux vivasAdyacentes (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) :  $\mathbb{Z} =$ 
  ( $\sum_{i=-1}^1 \sum_{j=-1}^1$  if valorPosicionNormalizada( $t, (\text{pos}_0 + i, \text{pos}_1 + j)$ ) then 1 else 0 fi)
  - (if estaViva( $t, \text{pos}_0, \text{pos}_1$ ) then 1 else 0 fi);
pred valorPosicionNormalizada (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) {
   $t[\text{normalizarIndice}(\text{filas}(t), \text{pos}_0)][\text{normalizarIndice}(\text{columnas}(t), \text{pos}_1)]$ 
}

```

```

    aux normalizarIndice (limite:  $\mathbb{Z}$ , i:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  = if  $i < 0$  then  $(i + limite)$  else
if  $i \geq limite$  then  $(i - limite)$  else  $i$  fi fi;
    aux evolucionarToroide (t: toroide) : toroide = te  $\leftrightarrow$  (te : toroide) ( $\forall i, j : \mathbb{Z}$ ) (enRango(t, i, j)  $\rightarrow_L$ 
(te[i][j] = vivaLuegoDeEvolucion(t, (i, j)))));
    aux evolucionarToroideMultiple (t: toroide, k:  $\mathbb{Z}$ ) : toroide = ts[k-1]  $\leftrightarrow$  ( $\exists ts : seq\langle toroide \rangle$ ) ( $\forall i : \mathbb{Z}$ ) ( $(0 < i < k) \rightarrow_L (evolucionarToroide(ts[i - 1]) = ts[i])$ ));

```

3. Decisiones tomadas