



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# TP de Especificación

## Juego de la vida toroidal

1 de octubre 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 = valorLuegoDeEvolucion(t, posicion)}
}

proc evolucionToroide (inout t: toroide) {
  Pre {esToroideValido(t) ∧L t = T0}
  Post {esEvolucion(T0, t)}
}

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

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

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

proc seleccionNatural (in ts: seq⟨toroide⟩, out res: ℤ) {
  Pre {todosToroidesValidos(ts) ∧L algunToroideMuere(ts)}
  Post {(∃is : seq⟨ℤ⟩)(length(is) = length(ts)
    ∧L ordenadaDeMayorAMenorPorCantidadDeTicksHastaMuerte(ts, is))}
}

```

```

 $\wedge_L res = is[0]\}$ 
pred todosToroideValidos (ts: seq<toroide>) { $(\forall t: \text{toroide})((t \in ts) \rightarrow_L esToroideValido(t))$ }
pred algunToroideMuere (ts: seq<toroide>) { $(\exists t: \text{toroide})((t \in ts) \wedge_L (muere(t)))$ }
pred ordenadaDeMayorAMenorPorCantidadDeTicksHastaMuerte (ts: seq<toroide>, is: seq< $\mathbb{Z}$ >)
  { $(\forall i: \mathbb{Z})(((i \in is) \wedge_L (0 < i < length(is)))$ 
 $\rightarrow_L (cantidadDeTicksHastaMuerte(ts[i-1]) \geq cantidadDeTicksHastaMuerte(ts[i])))$ }
aux cantidadDeTicksHastaMuerte (t: toroide) :  $\mathbb{Z}$  =
  if  $(\exists tx: \text{toroide})(mismaDimension(t, tx) \wedge_L cantidadVivas(tx) = 0) \wedge_L (\exists ks: seq<\mathbb{Z}>)(length(ks) >$ 
 $0 \wedge_L ks[0] > 0 \wedge_L ordenadaDeMenorAMayor(is) \wedge_L esEvolucionMultiple(t, tx, ks[i -$ 
 $1]))$  then  $ks[0]$ 
  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 { $res = tf \leftrightarrow (\exists tf: \text{toroide})((filas(tf) = filas(t1) \wedge_L columnas(tf) = columnas(t1)) \wedge_L$ 
 $(\forall i, j: \mathbb{Z})((enRango(tf, i, j)) \rightarrow_L (estaViva(tf, i, j) \leftrightarrow estaViva(t1, i, j) \wedge_L estaViva(t2, i, j))))$ }
}

proc vistaTrasladada (in t1: toroide, in t2: toroide, out res: Bool) {
  Pre { $(esToroideValido(t1) \wedge esToroideValido(t2)) \wedge_L mismaDimension(t1, t2)$ }
  Post {
 $res = true \leftrightarrow esVistaTrasladada(t1, t2)$ 
}

proc enCrecimiento (in t: toroide, out res: Bool) {
  Pre { $esToroideValido(t)$ }
  Post { $res = true \leftrightarrow$ 
 $(\exists te: \text{toroide})(esEvolucion(t, te) \wedge_L crecio(t, te))$ }
  pred crecio (t: toroide, te: toroide) { $(\exists s, se: \mathbb{Z})$ 
 $((esMenorSuperficie(t, s) \wedge esMenorSuperficie(te, se)) \wedge_L s < se)$ }
  pred esMenorSuperficie (t: toroide, s:  $\mathbb{Z}$ ) {
 $(\exists ts: seq<\text{toroide}>)((compuestoPorVistasTrasladadas(ts, t) \wedge_L ordenadoSuperficieAscendente(ts))$ 
 $\wedge_L s = ts[0])$ }
  pred compuestoPorVistasTrasladadas (ts: seq<toroide>, t: toroide) { $(\forall tx: \text{toroide})$ 
 $((tx \in ts) \rightarrow_L esVistaTrasladada(tx, t))$ }
  pred ordenadoSuperficieAscendente (ts: seq<toroide>) { $(\forall i: \mathbb{Z})((0 < i < length(ts))$ 
 $\rightarrow_L (esSupMayorOIgual(ts[i], ts[i-1])))$ }
  pred esSupMayorOIgual (t1: toroide, t2: toroide) { $(\exists s1, s2: \mathbb{Z})$ 
 $((esSuperficie(t1, s1) \wedge esSuperficie(t2, s2)) \wedge_L s1 \geq s2)$ }
  pred esSuperficie (t: toroide) { $(\exists rs: seq<(\mathbb{Z} \times \mathbb{Z}) \times (\mathbb{Z} \times \mathbb{Z})>)$ 
 $(compuestoPorEquivalentes(rs, t) \wedge_L ((\forall i: \mathbb{Z})((0 < i < length(rs)) \rightarrow_L$ 
 $((0 \leq rs[i]_{0_0} \leq rs[i]_{1_0} \leq columnas(t)) \wedge (0 \leq rs[i]_{0_1} \leq rs[i]_{1_1} \leq filas(t)))$ 
 $\wedge_L (area(rs[i]) \geq area(rs[i-1]))))$ }
  pred compuestoPorRangosEquivalentes (rs: seq< $(\mathbb{Z} \times \mathbb{Z}) \times (\mathbb{Z} \times \mathbb{Z})$ >, t: toroide) { $(\forall rect:$ 
 $(\mathbb{Z} \times \mathbb{Z}) \times (\mathbb{Z} \times \mathbb{Z}))((rect \in rs) \rightarrow_L (contarVivasEnArea(t, rect) = cantidadVivas(t)))$ }
  / *  $rect = (xStart, yStart) \times (xEnd, yEnd)$  */
  aux contarVivasEnArea (t: toroide, rect:  $(\mathbb{Z} \times \mathbb{Z}) \times (\mathbb{Z} \times \mathbb{Z})$ ) :  $\mathbb{Z}$  =

```

```

      
$$\sum_{i=rect_{0_1}}^{rect_{1_1}} \left( \sum_{j=rect_{0_0}}^{rect_{1_0}} \text{ if } estaViva(t[i][j]) \text{ then } 1 \text{ else } 0 \text{ fi} \right);$$

    aux area (rect:  $(\mathbb{Z} \times \mathbb{Z}) \times (\mathbb{Z} \times \mathbb{Z})$ ) :  $\mathbb{Z} =$ 
      / * base  $\times$  altura * /
       $(rect_{1_0} - rect_{0_0}) \times (rect_{1_1} - rect_{0_1});$ 
  }

```

## 2. Predicados y Auxiliares generales

```

    pred noEsVacía (t: toroide) { (length(t) > 0)  $\wedge_L$  ( $\forall x : seq\langle Bool \rangle$ ) ( $(x \in t) \rightarrow_L (\text{length}(x) > 0)$ ) }
    pred esMatriz (t: toroide) { ( $\forall x, y : seq\langle Bool \rangle$ ) ( $(x, y \in t) \rightarrow_L (\text{length}(x) = \text{length}(y))$ ) }
    pred esToroideValido (t: toroide) { (noEsVacía(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) :  $\mathbb{Z} =$ 
      
$$\sum_{i=0}^{\text{filas}(t)-1} \left( \sum_{j=0}^{\text{columnas}(t)-1} \text{ if } estaViva(t[i][j]) \text{ then } 1 \text{ else } 0 \text{ fi} \right);$$


    aux valorLuegoDeEvolucion (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) : Bool =
    if seMantieneViva(t, pos)  $\vee_L$  vivePorReproduccion(t, pos) then true else false fi;

    pred seMantieneViva (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) { estaViva(t[posicion_0][posicion_1])  $\wedge_L$   $2 \leq$ 
    vivasAdyacentes(t, posicion)  $\leq$  3 }
    pred vivePorReproduccion (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) { ( $\neg estaViva(t[posicion_0][posicion_1])$ )  $\wedge_L$ 
    vivasAdyacentes(t, posicion) = 3 }

    aux vivasAdyacentes (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) :  $\mathbb{Z} =$ 
    
$$\left( \sum_{i=-1}^1 \sum_{j=-1}^1 \text{ if } valorPosicionNormalizada(t, (pos_0 + i, pos_1 + j)) = \text{true} \text{ then } 1 \text{ else } 0 \text{ fi} \right)$$

    - (if estaViva(t, pos_0, pos_1) then 1 else 0 fi);

    aux valorPosicionNormalizada (t: toroide, pos:  $\mathbb{Z} \times \mathbb{Z}$ ) : Bool =
    t[normalizarIndice(filas(t), pos_0)][normalizarIndice(columnas(t), pos_1)]
  ;

    aux normalizarIndice (limite:  $\mathbb{Z}$ , i:  $\mathbb{Z}$ ) :  $\mathbb{Z} =$  if  $i < 0$  then  $(i + \text{limite})$  else
    (if  $i \geq \text{limite}$  then  $(i - \text{limite})$  else  $i$  fi) fi;

    pred mismaDimension (t1: toroide, t2: toroide) { filas(t1) = filas(t2)  $\wedge_L$  columnas(t1) =
    columnas(t2) }

    pred esEvolucion (t: toroide, te: toroide) { mismaDimension(t, te)  $\wedge_L$  ( $\forall i, j : \mathbb{Z}$ ) ( $\text{enRango}(t, i, j) \rightarrow_L$ 
     $(te[i][j] = \text{valorLuegoDeEvolucion}(t, (i, j)))$ ) }

    pred esEvolucionMultiple (t: toroide, te: toroide, k:  $\mathbb{Z}$ ) { ( $\exists ts : seq\langle toroide \rangle$ ) ( $(\text{length}(ts) =$ 
     $k + 1) \wedge_L ts[0] = t \wedge_L \text{ordenadaPorEvolucion}(ts) \wedge_L te = ts[k]$ ) }

    pred ordenadaPorEvolucion (ts:  $seq\langle toroide \rangle$ ) { ( $\forall i : \mathbb{Z}$ ) ( $(0 < i \leq k) \rightarrow_L$ 
     $\text{esEvolucion}(ts[i-1], ts[i])$ ) }

    pred ordenadaDeMenorAMayor (ks:  $seq\langle \mathbb{Z} \rangle$ ) { ( $\forall i : \mathbb{Z}$ ) ( $(0 < i < \text{length}(ks)) \rightarrow_L (ks[i-1] <$ 
     $ks[i])$ ) }

    pred esVistaTrasladada (t1: toroide, t2: toroide) { ( $\exists i, j : \mathbb{Z}$ ) ( $(\forall x, y : \mathbb{Z}) (\text{enRango}(t1, x, y) \rightarrow_L$ 
     $(t1[x][y] = \text{valorPosicionNormalizada}(t2, (x + i, y + j))))$ ) }

```

### **3. Decisiones tomadas**

Intuimos que una posición tiene 8 adyacentes independientemente del tamaño del toroide, implicando esto que dentro de las adyacentes a una posición se pueden contar posiciones repetidas