



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

Juego de la vida toroidal

30 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 ↔ 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⟩)((∀i : ℤ)((0 < i < length(ks)) →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 ((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⟩)((∀i : ℤ)((i ∈ is) ∧L (0 < i < length(is)))
    →L (cantidadDeTicksHastaMuerte(ts[i - 1]) ≥ cantidadDeTicksHastaMuerte(ts[i]))) ∧L
    res = is[0]}
  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) :  $\mathbb{Z}$  =
  if ( $\exists tx : \text{toroide}$ )( $(\forall x \in tx) \longrightarrow_L \neg \text{estaViva}(x) \wedge_L (\exists is : \text{seq}(\mathbb{Z}))((\forall i : \mathbb{Z})((0 < i < \text{length}(is)) \longrightarrow_L ((is[i-1] < is[i]) \wedge_L \text{esEvolucionMultiple}(t, tx, is[i-1]))))$ ) then  $is[0]$ 
  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)  $\longrightarrow_L$  (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))  $\wedge_L$  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:  $\mathbb{Z}$ ) {
    ( $\exists ts : \text{seq}(\text{toroide})$ )(compuestoPorVistasTrasladadas(ts, t)  $\wedge$  ordenadoSuperficieAscendente(ts)  $\wedge_L$  s = ts[0])
  }
  pred compuestoPorVistasTrasladadas (ts: seq(toroide), t: toroide) {( $\forall tx : \text{toroide}$ )
    ((tx  $\in$  ts)  $\longrightarrow_L$  esVistaTrasladada(tx, t))}
  pred ordenadoSuperficieAscendente (ts: seq(toroide)) {( $\forall i : \mathbb{Z}$ )(0 < i < length(ts)  $\longrightarrow_L$  (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) {( $\exists rs : \text{seq}((\mathbb{Z} \times \mathbb{Z}) \times (\mathbb{Z} \times \mathbb{Z}))$ )
    (compuestoPorEquivalentes(rs, t)  $\wedge_L$  ( $\forall i : \mathbb{Z}$ )(0 < i < length(rs)  $\longrightarrow_L$ 
    ((0  $\leq$  rs[i]00  $\leq$  rs[i]10  $\leq$  columnas(t))  $\wedge$  (0  $\leq$  rs[i]01  $\leq$  rs[i]11  $\leq$  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)  $\longrightarrow_L$  (contarVivasEnArea(t, rect) = cantidadVivas(t))}
  /* rect = (xStart, yStart)  $\times$  (xEnd, yEnd) */
  aux contarVivasEnArea (t: toroide, rect: ( $\mathbb{Z} \times \mathbb{Z}$ )  $\times$  ( $\mathbb{Z} \times \mathbb{Z}$ )) :  $\mathbb{Z}$  =
     $\sum_{i=rect_{01}}^{rect_{11}} ( \sum_{j=rect_{00}}^{rect_{10}} \text{if } \text{estaViva}(t[i][j]) \text{ then } 1 \text{ else } 0 \text{ fi} );$ 
  aux area (rect: ( $\mathbb{Z} \times \mathbb{Z}$ )  $\times$  ( $\mathbb{Z} \times \mathbb{Z}$ )) :  $\mathbb{Z}$  =
    /* base  $\times$  altura */
    (rect10 - rect00)  $\times$  (rect11 - rect01);
}

```

2. Predicados y Auxiliares generales

```

pred noEsVacia (t: toroide) {(length(t) > 0) ∧L (∀x : seq⟨Bool⟩)((x ∈ t) →L (length(x) > 0))}
pred esMatriz (t: toroide) {(∀x, y : seq⟨Bool⟩)((x, y ∈ t) →L (length(x) = length(y)))}
pred esToroideValido (t: toroide) {(noEsVacia(t) ∧ 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: ℤ, j: ℤ) {if (0 ≤ i < filas(t)) ∧L (0 ≤ j < columnas(t)) then true else false fi}

aux cantidadVivas (t: toroide): ℤ = ∑i=0filas(t)-1 ( ∑j=0columnas(t)-1 if estaViva(t[i][j]) then 1 else 0 fi );

pred valorLuegoDeEvolucion (t: toroide, pos: ℤ × ℤ) {seMantieneViva(t, pos) ∨L vivePorReproduccion(t, pos)}
pred seMantieneViva (t: toroide, pos: ℤ × ℤ) {estaViva(t[posicion0][posicion1]) ∧L 2 ≤
vivasAdyacentes(t, posicion) ≤ 3}
pred vivePorReproduccion (t: toroide, pos: ℤ × ℤ) {(¬estaViva(t[posicion0][posicion1]) ∧L
vivasAdyacentes(t, posicion) = 3)}

aux vivasAdyacentes (t: toroide, pos: ℤ × ℤ): ℤ =
( ∑i=-11 ∑j=-11 if valorPosicionNormalizada(t, (pos0 + i, pos1 + j)) then 1 else 0 fi )
- (if estaViva(t, pos0, pos1) then 1 else 0 fi);

pred valorPosicionNormalizada (t: toroide, pos: ℤ × ℤ) {
t[normalizarIndice(filas(t), pos0)] [normalizarIndice(columnas(t), pos1)]
}

aux normalizarIndice (limite: ℤ, i: ℤ): ℤ = if i < 0 then (i + limite) else
if i ≥ limite then (i - limite) else i fi;

pred mismaDimension (t1: toroide, t2: toroide) {filas(t1) = filas(t2) ∧L columnas(t1) =
columnas(t2)}

pred esEvolucion (t: toroide, te: toroide) {mismaDimension(t, te) ∧L (∀i, j : ℤ)(enRango(t, i, j) →L
(te[i][j] = valorLuegoDeEvolucion(t, (i, j))))}

pred esEvolucionMultiple (t: toroide, te: toroide, k: ℤ) {(∃ts : seq⟨toroide⟩)((length(ts) =
k)(∀i : ℤ)((0 < i < k) →L (esEvolucion(ts[i - 1], ts[i])) ∧L te = ts[k - 1]))}

```

3. Decisiones tomadas

```

pred esVistaTrasladada (t1: toroide, t2: toroide) {(∃i, j : ℤ)(
(∀x, y : ℤ)(enRango(t1, x, y) →L (t1[x][y] = valorPosicionNormalizada(t2, (x + i, y + j))))))}

```