



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

1.1. esValido

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

1.2. posicionesVivas

```
proc posicionesVivas (in t: toroide, out vivas: seq( $\mathbb{Z} \times \mathbb{Z}$ )) {  
    Pre {esToroideValido(t)}  
    Post {(∀i, j :  $\mathbb{Z}$ )(enRango(t, i, j) →L ((i, j) ∈ vivas ↔ estaViva(t[i][j])))}  
}
```

1.3. densidadPoblacion

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

1.4. evolucionDePosicion

```
proc evolucionDePosicion (in t: toroide, in posicion:  $\mathbb{Z} \times \mathbb{Z}$ , out result: Bool) {  
    Pre {esToroideValido(t) ∧L enRango(t, posicion0, posicion1)}  
    Post {result = valorLuegoDeEvolucion(t, posicion)}  
}
```

1.5. evolucionToroide

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

1.6. evolucionMultiple

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

1.7. esPeriodico

```

proc esPeriodico (in t: toroide, inout p:  $\mathbb{Z}$ , out result: Bool) {
  Pre {esToroideValido(t)}
  Post {result = true  $\leftrightarrow$  ( $\exists k : \mathbb{Z}$ )(esEvolucionMultiple(t, t, k)
     $\wedge_L$  esElMenorKAntesDeEvolucion(t, t, k))  $\wedge_L$  p = k}
}

```

1.8. primosLejanos

```

proc primosLejanos (in t1: toroide, in t2: toroide, out primos: Bool) {
  Pre {esToroideValido(t1)  $\wedge$  esToroideValido(t2)  $\wedge_L$  mismaDimension(t1, t2)}
  Post {primos = true  $\leftrightarrow$  (( $\exists k : \mathbb{Z}$ )(k > 0)
     $\wedge_L$  ((esEvolucionMultiple(t1, t2, k))  $\vee_L$  (esEvolucionMultiple(t2, t1, k))))}
}

```

1.9. seleccionNatural

```

proc seleccionNatural (in ts: seq<toroide>, out res:  $\mathbb{Z}$ ) {
  Pre {length(ts) > 0  $\wedge_L$  (todosToroidesValidos(ts)  $\wedge_L$  algunToroideMuere(ts))}
  Post {(0  $\leq$  res < length(ts))  $\wedge_L$  tieneElMayorTiempoDeMuerte(ts[res], ts)}
  pred todosToroidesValidos (ts: seq<toroide>) {
    ( $\forall t$ : toroide)((t  $\in$  ts)  $\rightarrow_L$  esToroideValido(t))
  }
  pred algunToroideMuere (ts: seq<toroide>) {
    ( $\exists t$ : toroide)((t  $\in$  ts)  $\wedge_L$  (muere(t)))
  }
  pred tieneElMayorTiempoDeMuerte (t: toroide, ts: seq<toroide>) {
    ( $\forall tx$ : toroide)((tx  $\in$  ts)
       $\rightarrow_L$  (cantidadDeTicksHastaMuerte(t)  $\geq$  cantidadDeTicksHastaMuerte(tx)))
  }
  aux cantidadDeTicksHastaMuerte (t: toroide) :  $\mathbb{Z}$  =
    if ( $\exists tx$ : toroide)(mismaDimension(t, tx)  $\wedge_L$  cantidadVivas(tx) = 0)
       $\wedge_L$  ( $\exists k : \mathbb{Z}$ )(k > 0  $\wedge_L$  esEvolucionMultiple(t, tx, k)  $\wedge_L$  esElMenorKAntesDeEvolucion(t, tx, k))
    then k else -1 fi;
  pred muere (t: toroide) {
    cantidadDeTicksHastaMuerte(t)  $\neq$  -1
  }
}

```

1.10. fusionar

```

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

```

1.11. vistaTrasladada

```

proc vistaTrasladada (in t1: toroide, in t2: toroide, out res: Bool) {
  Pre {(esToroideValido(t1) ∧ esToroideValido(t2)) ∧L mismaDimension(t1, t2)}
  Post {res = true ↔ esVistaTrasladada(t1, t2)}
}

```

1.12. enCrecimiento

```

proc enCrecimiento (in t: toroide, out res: Bool) {
  Pre {esToroideValido(t)}
  Post {res = true ↔ (∃te: toroide)(esEvolucion(t, te) ∧L crecioSuperficie(t, te))}
  pred crecioSuperficie (t: toroide, te: toroide) {
    (∃s, se: ℤ)((esMenorSuperficieDeToroide(t, s) ∧ esMenorSuperficieDeToroide(te, se))
    ∧L s < se)
  }
  pred esMenorSuperficieDeToroide (t: toroide, s: ℤ) {
    (∃x: toroide)((esVistaTrasladada(t, x) ∧L tieneLaMenorSuperficieDelToroide(x)))
  }
  pred tieneLaMenorSuperficieDelToroide (t: toroide) {
    (∀x: toroide)((x ≠ t) ∧L esVistaTrasladada(t, x) ∧L esSuperficieMayorOIgual(x, t))
  }
  pred esSuperficieMayorOIgual (t1: toroide, t2: toroide) {
    (∃s1, s2: ℤ)((esSuperficie(t1, s1) ∧ esSuperficie(t2, s2)) ∧L s1 ≥ s2)
  }
  pred esSuperficie (t: toroide, s: ℤ) {
    (∃rect: (ℤ × ℤ) × (ℤ × ℤ))(mantieneCantidadVivas(rect, t) ∧L esRectanguloDeToroide(rect, t)
    ∧L esElMenorRectangulo(rect, t) ∧L area(rect) = s)
  }
  pred mantieneCantidadVivas (rect: (ℤ × ℤ) × (ℤ × ℤ), t: toroide) {
    /* rect = (xStart, yStart) × (xEnd, yEnd) */
    contarVivasEnArea(t, rect) = cantidadVivas(t)
  }
  pred esRectanguloDeToroide (rect: (ℤ × ℤ) × (ℤ × ℤ), t: toroide) {
    /* rect = (xStart, yStart) × (xEnd, yEnd) */
    (0 ≤ rect00 ≤ rect10 ≤ columnas(t)) ∧ (0 ≤ rect01 ≤ rect11 ≤ filas(t))
  }
  pred esElMenorRectangulo (r1: (ℤ × ℤ) × (ℤ × ℤ), t: toroide) {
    (∀r2: (ℤ × ℤ) × (ℤ × ℤ))((r2 ≠ r1)
    →L (¬mantieneCantidadVivas(r2, t) ∨L area(r2) ≥ area(r1)))
  }
  /* rect = (xStart, yStart) × (xEnd, yEnd) */
  aux contarVivasEnArea (t: toroide, rect: (ℤ × ℤ) × (ℤ × ℤ)) : ℤ =
    ∑i=rect01rect11 ( ∑j=rect00rect10 if estaViva(t[i][j]) then 1 else 0 fi );
  aux area (rect: (ℤ × ℤ) × (ℤ × ℤ)) : ℤ =
    /* base × altura */
    (rect10 - rect00) × (rect11 - rect01);
}

```

2. Predicados y Auxiliares generales

```

pred noEsVacía (t: toroide) {
  (length(t) > 0)  $\wedge_L$  ( $\forall x : seq(Bool)$ ) ( $(x \in t) \longrightarrow_L (length(x) > 0)$ )
}
pred esMatriz (t: toroide) {
  ( $\forall x, y : seq(Bool)$ ) ( $(x, y \in t) \longrightarrow_L (length(x) = length(y))$ )
}
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 ( $0 \leq i < filas(t)$ )  $\wedge_L$  ( $0 \leq j < columnas(t)$ ) then true else false fi
}
aux cantidadVivas (t: toroide) :  $\mathbb{Z}$  =
   $\sum_{i=0}^{filas(t)-1} ( \sum_{j=0}^{columnas(t)-1} \text{if } estaViva(t[i][j]) \text{ then } 1 \text{ else } 0 \text{ fi} )$ ;
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:  $\mathbb{Z} \times \mathbb{Z}$ ) :  $\mathbb{Z}$  =
  ( $\sum_{i=-1}^1 \sum_{j=-1}^1 \text{if } valorPosicionNormalizada(t, (pos_0 + i, pos_1 + j)) = \text{true} \text{ then } 1 \text{ else } 0 \text{ fi}$ )
  - (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 + limite) else
  (if i  $\geq$  limite then (i - 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)  $\wedge_L$  ( $\forall i, j : \mathbb{Z}$ ) ( $enRango(t, i, j) \longrightarrow_L (te[i][j] = valorLuegoDeEvolucion(t, (i, j)))$ )
}
pred esEvolucionMultiple (t: toroide, te: toroide, k:  $\mathbb{Z}$ ) {
  ( $\exists ts : seq(toroide)$ ) ( $(length(ts) = k + 1) \wedge_L ts[0] = t \wedge_L ordenadaPorEvolucion(ts)$ 
 $\wedge_L te = ts[k]$ )
}

```

```

pred ordenadaPorEvolucion (ts: seq<toroide>) {
  (∀i : ℤ)((0 < i ≤ k) →L esEvolucion(ts[i - 1], ts[i]))
}
/* Indica si el toroide t llega a tener la forma de te por primera vez en el k-ésimo tick*/
pred esElMenorKAntesDeEvolucion (t: toroide, te: toroide, k: ℤ) {
  (∀i : ℤ)((0 < i < k) →L ¬esEvolucionMultiple(t, te, i))
}
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)))))
}

```

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.