



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

1 de octubre de 2020

Algoritmos y Estructuras de Datos I

Grupo: 15

Integrante	LU	Correo electrónico
Mauricio David Toranzo	63/20	david-toranzo@hotmail.com
Matias Federico Sarmiento	741/18	mfsarmiento@gmail.com
Victor Manuel Asmad Murga	760/19	victorasmad2@gmail.com
Marco Antonio Avila Tapia	412/20	marco6267@hotmail.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. Ejercicios - Primera Parte

```
pred esValido (t : toroide) {
  (∀i : ℤ)((0 ≤ i < |t| ∧ |t| ≥ 3) →L (|t[i]| ≥ 3 ∧ |t[0]| = |t[i]|))
}

pred toroideMuerto (t : toroide) {
  (∀i : ℤ)((∀j : ℤ)((0 ≤ i < |t| ∧L 0 ≤ j < |t[i]|) →L (t[i][j] = false)))
}

pred posicionesVivas (t : toroide, vivas : seq(ℤxℤ)) {
  |vivas| > 0 ∧L (∀i : ℤ)((0 ≤ i < |vivas|) →L
  ((0 ≤ vivas[i]0 < |t| ∧L 0 ≤ vivas[i]1 < |t[0]|) ∧L (t[vivas[i]0][vivas[i]1] = true)))
}

aux densidadPoblacion (t : toroide) : ℤ =
  (∑i=0|t|-1 (∑j=0|t[i]|-1 if (t[i][j] = true) then 1 else 0 fi)) / (|t| * |t[0]|);

aux cantVecinosVivos (t : toroide, f : ℤ, c : ℤ) : ℤ =
  (∑i=f-1f+1 (∑j=c-1c+1 if (i ≠ f ∧ j ≠ c ∧ (t[imod|t|][jmod|t[0]|])) then 1 else 0 fi));

pred evolucionDePosicion (t : toroide, posicion : ℤxℤ) {
  0 ≤ posicion0 < |t| ∧ 0 ≤ posicion1 < |t[0]| ∧
  if t[posicion0][posicion1] then 2 ≤ cantVecinosVivos(t, posicion0, posicion1) ≤ 3
  else cantVecinosVivos(t, posicion0, posicion1) = 3 fi
}

pred evolucionToroide (t1 : toroide, t2 : toroide) {
  |t1| = |t2| ∧ |t1[0]| = |t2[0]| ∧L
  (∀i : ℤ)(0 ≤ i < |t1| ∧L (∀j : ℤ)(0 ≤ j < |t1[0]| →L (evolucionDePosicion(t1, (i, j)) = t2[i][j])))
}
```

2. Ejercicios - Segunda Parte

```
proc evolucionMultiple (in t: toroide, in k: ℤ, out result: toroide) {
  Pre {esValido(t) ∧ k > 0}
  Post {coincideTamanio(t1, result) ∧ esKesimaEvolucion(t, k, result)}
}

proc esPeriodico (in t: toroide, inout p: ℤ, out result: Bool) {
  Pre {esValido(t) ∧ p > 0}
  Post {result = true ↔ (∃k : ℤ)(k > 0 → (esKesimaEvolucion(t, k, t) ∧ p = k))}
}

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

proc seleccionNatural (in ts: seq(toroide), out res: ℤ) {
  Pre {|ts| > 0 ∧ noHayToroidesInmortales(ts)}
  Post {0 ≤ res < |ts| ∧L toroideTardaMasEnMorir(ts, res)}
}

proc fusionar (in t1: toroide, in t2: toroide, out res: toroide) {
  Pre {esValido(t1) ∧ esValido(t2) ∧ coincideTamanio(t1, t2)}
  Post {coincideTamanio(t1, res) ∧ coincideTamanio(t2, res) → contieneVivas(t1, t2, result)}
}
```

```

proc vistaTrasladada (in t1: toroide, in t2: toroide, out res: Bool) {
  Pre {esValido(t1) ∧ esValido(t2) ∧ coincideTamanio(t1, t2)}
  Post {res = true ↔ esTraslado(t1, t2)}
}

proc menorSuperficieViva (in t: toroide, out res: ℤ) {
  Pre {esValido(t) ∧ ¬toroideMuerto(t)}
  Post {
    (∃ts : seq⟨toroide⟩)(esListaDeTraslados(ts, t) ∧ (∃tMenor : toroide)
      (tMenor ∈ ts ∧ (∀tItem ∈ ts)(tieneSuperficieMasChica(tMenor, t, res))))))
}

proc enCrecimiento (in t: toroide, out res: Bool) {
  Pre {esValido(t)}
  Post {res = true ↔ (∃tEvo : toroide)
    (coincideTamanio(t, tEvo) ∧ evolucionToroide(t, tEvo)
      ∧ evolucionTieneSuperficieMayor(tEvo, t))}
}

```

3. Funciones y Predicados Auxiliares:

```

pred esKesimaEvolucion (t:toroide, k:ℤ, result: toroide) {
  (∃ ts : seq < toroide >)
  (|ts| = k ∧L ts[0] = t ∧ ts[k - 1] = result ∧ (∀i : ℤ)
  (0 ≤ i < |ts| - 1 →L evolucionToroide(ts[i], ts[i + 1]))))}

pred coincideTamanio (t:toroide, tAux:toroide) {|t| = |tAux| ∧L |t[0]| = |tAux[0]|}
pred muerteEnTicks (t:toroide, k:ℤ) {
  (∃tm : toroide)(|tm| = |t| ∧L |tm[0]| = |t[0]| ∧ toroideMuerto(tm) ∧L esKesimaEvolucion(t, k, tm))}

pred noHayToroidesInmortales (ts: seq⟨toroide⟩) {(∀i : ℤ)(0 ≤ i < |ts| →L esValido(ts[i]) ∧ (∃k : ℤ)(k > 0 ∧
muerteEnTicks(ts[i], k))}
pred toroideTardaMasEnMorir (ts: seq⟨toroide⟩, res: ℤ) {(∀i : ℤ)(0 ≤ i < |ts| →L
(∃k, w : ℤ)(k > w ∧ k > 0 ∧ w > 0 ∧
muerteEnTicks(ts[res], k) ∧ muerteEnTicks(ts[i], w)))}

pred contieneVivas (t1:toroide, t2:toroide, result:toroide) {
  (∀i : ℤ)(0 ≤ i < |t1| ∧L (∀j : ℤ)(0 ≤ j < |t1[i]| →L (t1[i][j] = true ∧ t2[i][j] = true ∧ result[i][j] = true))))}

pred esTraslado (t1:toroide, t2:toroide) {
  (∃k : ℤ)(0 ≤ k < |t1| ∧L (∃l : ℤ)(0 ≤ l < |t1[0]|
→L (∀i : ℤ)(0 ≤ i < |t1| ∧L (∀j : ℤ)(0 ≤ j < |t1[0]|
→L (t1[(i + k) mod |t1|][(j + l) mod |t1[0]|] = t2[(i + k) mod |t1|][(j + l) mod |t1[0]|])))))
}

pred tieneSuperficieMasChica (tMenor: toroide, tComparado:toroide, res:ℤ) {
  (∃matrizMenor, matrizComparada : seq⟨seq⟨Bool⟩⟩)(
estaEnRango(matrizMenor, tMenor) ∧
estaEnRango(matrizComparada, tComparado) ∧
cantVivas(matrizMenor) = cantVivas(tMenor) ∧
cantVivas(matrizComparada) = cantVivas(tComparado) ∧
estaContenido(matrizMenor, tMenor) ∧
estaContenido(matrizComparada, tComparado) ∧
superficieTotal(matrizMenor) = res ∧
superficieTotal(matrizMenor) ≤ superficieTotal(matrizComparada)
)}

pred esListaDeTraslados (ts:seq⟨toroide⟩, t:toroide) {
  (∀i : ℤ)(0 ≤ i < |ts| →L esValido(t) ∧ |ts[i]| = |t| ∧ |ts[i][0]| = |t[0]| ∧ esTraslado(t, ts[i]))
}

```

```
pred estaEnRango (m:seq⟨seq⟨Bool⟩⟩, t:toroide) {0 < |m| ≤ |t| ∧ 0 < |m[0]| ≤ |t[0]|}
```

```
pred estaContenido (m:seq⟨seq⟨Bool⟩⟩, tAux:toroide) {
|m| ≤ |tAux| ∧L |m[0]| ≤ |tAux[0]| → (∀i : ℤ)(0 ≤ i < |m| ∧L (∀j : ℤ)(0 ≤ j < |m[0]| →L m[i][j] = tAux[i][j]))}
```

```
aux cantVivas (t:toroide) : ℤ =
(∑i=0|t|-1 (∑j=0|t[0]|-1 if (t[i mod |t|] [j mod |t[0]|]) then 1 else 0 fi));
```

```
aux superficieTotal (m:seq⟨seq⟨Bool⟩⟩) : ℤ = |m| * |m[0]|;
```

```
pred evolucionTieneSuperficieMayor (tEvo:toroide, t:toroide) {
(∃trasladoTInicial, trasladoEvolucion : toroide)
(esTraslado(t, trasladoTInicial) ∧
esTraslado(tEvo, trasladoEvolucion) ∧
¬(∃k : ℤ)(tieneSuperficieMasChica(trasladoTInicial, trasladoEvolucion, k)))
}
```

4. Decisiones tomadas

Usamos la primer fila en nuestras funciones y predicados pues los toroides son matrices (todas sus filas tienen el mismo largo y sus columnas el mismo alto), por lo tanto no cambia si usamos la primer o la i-esima fila.