UBA – Facultad de Ciencias Exactas y Naturales – Departamento de Computación

Algoritmos y Estructura de Datos I

Segundo cuatrimestre de 2020

Versión: 23 de septiembre de 2020

TPE - "Juego de la vida toroidal"

Fecha de entrega - Primera+Segunda parte: Viernes 2 de octubre de 2020 hasta las 22hs
Devolución: Viernes 9 de octubre de 2020

1. INTRODUCCIÓN

El juego de la vida de Conway [1] (no confundir con el juego de la vida que jugamos todos en nuestra niñez) [2] es un autómata celular diseñado por el matemático John Horton Conway en 1970. Este juego tiene lugar sobre un tablero de posiciones que pueden estar *vivas* o *muertas*, y que evoluciona a lo largo de unidades de tiempo discretas, llamadas ticks, respetando las siguientes reglas:

- Cualquier posición viva con menos de 2 vecinas vivas, muere (por *soledad*)
- Cualquier posición viva con 2 o 3 vecinos vivos, vive.
- Cualquier posición viva con más de 3 vecinas vivas, muere (por *superpoblación*)
- Cualquier posición muerta con exactamente 3 vecinos vivos, pasa a vivir (por *reproducción*)

Cada posición tiene 8 posiciones *vecinas*: la de arriba, la de abajo, la de la izquierda, la de la derecha, y las 4 en diagonal. Para este TP consideraremos nuestro tablero como un *toroide*[3] de NxM, donde $N \geq 3$ y $M \geq 3$. Toroide es la palabra matemáticamente elegante para describir una *donut*.

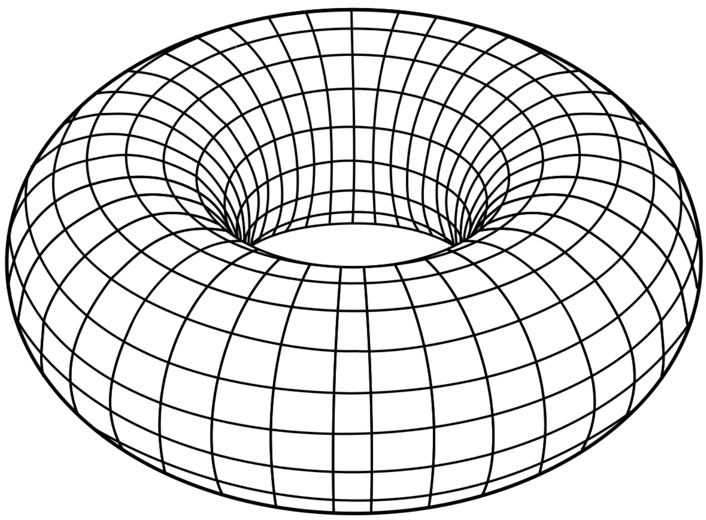


Figura 1: Un tablero toroidal sin posiciones vivas

Representaremos un toroide mediante una secuencia de secuencias de booleanos. La secuencia principal contendrá las filas. Cada fila se representará con otra secuencia que modela cada una de las columnas de la fila en cuestión. El valor *True* indicará que una posición se encuentra *viva*, mientras que el valor *False* representará a las posiciones *muertas*. El toroide de la siguiente figura está representado por la siguiente secuencia: `[[True, True, False],[False, False, False],[False, False, False]]`

[1] `https://en.wikipedia.org/wiki/Conway's_Game_of_Life`
[2] `https://en.wikipedia.org/wiki/The_Game_of_Life`
[3] `https://es.wikipedia.org/wiki/Toroide`

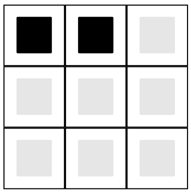


Figura 2: Un tablero toroidal de 3x3 con 2 posiciones vivas

2. EJERCICIOS - Primera parte

Especificar los siguientes predicados y funciones auxiliares dado el **renombre** de tipos:

type *toroide*= $seq\langle seq\langle \mathsf{Bool}\rangle\rangle$

Ejercicio 1 : pred esValido(t: *toroide*)
Que dado una secuencia de secuencias verifique si modela un toroide válido.

Ejercicio 2 : pred toroideMuerto(t: *toroide*)
Que dado un toroide t indica si está muerto. Diremos que un toroide está muerto cuando no tiene ninguna celda viva. Asumir que este predicado se usará en el contexto en el cual t cumple ser un toroide válido.

Ejercicio 3 : pred posicionesVivas(t: *toroide*, vivas : $seq\langle \mathbb{Z} \times \mathbb{Z}\rangle$)
Que dado un toroide válido t y una secuencia de posiciones, devuelve true sii la secuencia contiene todas las celdas vivas del toroide, y ninguna otra celda.

Ejercicio 4 : aux densidadPoblacion(t: *toroide*) = $\mathbb{R}$
Que dado un toroide válido t devuelva su densidad de población, es decir, la relación entre la cantidad de posiciones vivas y la cantidad total de posiciones.

Ejercicio 5 : aux cantVecinosVivos(t: *toroide*, f: $\mathbb{Z}$, c: $\mathbb{Z}$) = $\mathbb{Z}$
Que dado un toroide válido t, y una posición del mismo representada por dos enteros f y c (los cuales representan una fila y una columna en el rango del toroide), devuelve la cantidad de vecinos vivos de dicha posición.

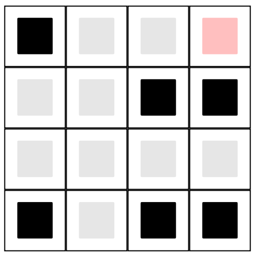


Figura 3: La posición (0,3) se encuentra muerta y tiene 6 de sus 8 vecinos vivos

Ejercicio 6 : pred evolucionDePosicion(t: *toroide*, posicion : $\mathbb{Z} \times \mathbb{Z}$)
Que dado un toroide válido t y una posición válida del mismo, indique si dicha posición estaría viva luego de un tick.

Ejercicio 7 : pred evolucionToroide(t1: *toroide*, t2: *toroide*)
Que dados dos toroides válidos $t1$ y $t2$, indique si $t2$ es la evolución de $t1$ luego de transcurrido un tick.

3. EJERCICIOS - Segunda parte

Ejercicio 8 *:* **proc evolucionMultiple(in t:** ***toroide*****, in k:** $\mathbb{Z}$**, out result:** ***toroide*****)**
Que dado un toroide t y un natural k, devuelva el toroide resultante de evolucionar t por k ticks.

Ejercicio 9 *:* **proc esPeriodico(in t:** ***toroide*****, inout p:** $\mathbb{Z}$**, out result:** Bool**)**
Que dado un toroide devuelva si el mismo es periódico o no. En caso de serlo, se debe devolver en p la mínima cantidad de ticks en la cual se repite el patrón. Decimos que un toroide es periódico si pasada cierta cantidad de ticks, vuelve a tener exactamente la misma configuración que tenía originalmente.
Por ejemplo, en el siguiente caso el toroide es periódico, y $p = 2$.

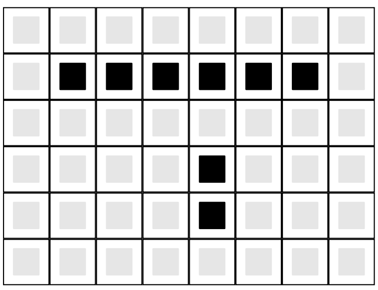
Figura 4: Toroide con período 2

Ejercicio 10 *:* **proc primosLejanos(in t1:** ***toroide*****, in t2:** ***toroide*****, out primos:** Bool**)**
Que dados dos toroides, devuelva si uno es la evolución múltiple del otro.

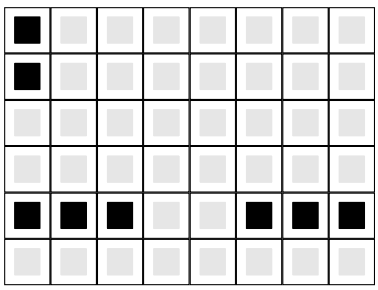
Ejercicio 11 *:* **proc seleccionNatural(in ts:** $seq\langle$***toroide***$\rangle$**, out res:** $\mathbb{Z}$**)**
Que dada una secuencia de toroides, devuelva el índice de aquel toroide que más ticks tardará en morir. Se considera que un toroide muere cuando no tiene posiciones vivas.

Ejercicio 12 *:* **proc fusionar(in t1:** ***toroide*****, in t2:** ***toroide*****, out res:** ***toroide*****)**
Que dados dos toroides de la misma dimensión, devuelva otro (de la misma dimensión) que tenga vivas solo aquellas posiciones que estaban vivas en ambos toroides.

Ejercicio 13 *:* **proc vistaTrasladada(in t1:** ***toroide*****, in t2:** ***toroide*****, out res:** Bool**)**
Que dados dos toroides de la misma dimensión, indica si uno es el resultado de trasladar la vista en el otro. Es decir, que moviendo el centro del eje de coordenadas de uno de los toroides en alguna dirección, se obtiene el otro. Por ejemplo, los toroides que se ven a continuación cumplen esta propiedad, ya que si se aplica un desplazamiento de 4 lugares a la derecha en el eje X y de 3 lugares hacia abajo en el eje Y al toroide de la izquierda, se obtiene el toroide de la derecha:



(a) Un toroide



(b) El mismo toroide trasladado $\rightarrow$4 en horizontal y $\downarrow$3 en vertical

Figura 5: Ejemplo de traslación de toroides

Ejercicio 14 *:* **proc menorSuperficieViva(in t:** ***toroide*****, out res:** $\mathbb{Z}$**)**
Que dado un toroide t, devuelva el valor del área del rectángulo más chico que contiene a todas las posiciones vivas. Tener en cuenta que el rectángulo más chico del toroide podría encontrarse en una de sus vistas trasladadas, como muestra la siguiente figura.

Figura 6: Un tablero toroidal (izquierda) de 4x4 con 4 posiciones vivas, cuya menor superficie viva tiene área 6 (tamaño 2x3). Esto se ilustra en un de sus vistas trasladadas (derecha): dos lugares hacia la derecha en el eje X y un lugar hacia abajo en el eje Y.

Términos y condiciones

El trabajo práctico debe realizarse de manera grupal y todos los integrantes del grupo deben conocer como se resuelven los ejercicios. Para aprobar el trabajo se necesita:

- Que todos los ejercicios estén resueltos.
- Que las soluciones sean correctas.
- Que el lenguaje de especificación esté bien utilizado.
- Que las soluciones sean prolijas: evitar repetir especificaciones innecesariamente y usar adecuadamente las funciones y predicados auxiliares.
- Que no haya casos de sub-especificación ni sobre-especificación.

Pautas de Entrega

El trabajo debe ser subido al campus en la sección Trabajos Prácticos en la fecha estipulada

Importante: se admitirá un único envío por grupo, sin excepción alguna. Por favor planifiquen el trabajo para llegar a tiempo con la entrega.