DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TPE - “Juego de la vida toroidal”

Ejercicios de LaTeX

13/09/20 Algoritmos y Estructuras de Datos I

Grupo 4

Integrante	LU	Correo electrónico
Giansiracusa, Magali	[redacted]	[redacted]
Oca, Mariano Agustín	[redacted]	[redacted]

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 con su Resolución Respectiva

Especificar los siguientes predicados y funciones auxiliares dado el renombre de tipos:
type toroide $= seq\langle seq\langle Bool\rangle\rangle$

1. Ejercicio 1 : *pred* $esValido(t : toroide)$

   Que dado una secuencia de secuencias verifique si modela un toroide válido.

   `aux filas` (t: toroide) : $\mathbb{Z} = |t|$;

   `aux columnas` (t: toroide) : $\mathbb{Z}$ = if $filas(t) > 0$ then $|t[0]|$ else 0 fi ;

   /* el aux columnas se va a utilizar más adelante */

   `pred esValido` (t: $seq\langle seq\langle Bool\rangle\rangle$) {
   $filas(t) \geq 3 \wedge (\forall i : \mathbb{Z})(0 \leq i < filas(t) \longrightarrow_L |t[i]| \geq 3 \wedge$
   $(\forall j : \mathbb{Z})(0 \leq j < filas(t) \longrightarrow_L |t[i]| = |t[j]|)\ )$
   }

2. 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.

   /*la sumatorias recorren las filas de las columnas, donde t[i][j] representa cada cuadrado*/

   `aux cuantasVivas` (t:toroide) : $\mathbb{Z} = \sum_{i=0}^{|t|-1} \sum_{j=0}^{|t[i]|-1}$ if $t[i][j] = True$ then 1 else 0 fi ;

   `pred toroideMuerto` (t: toroide) {
   $cuantasVivas(t) = 0$
   }

3. Ejercicio 3 : *pred* $posicionesVivas(t : toroide, vivas : seq\langle \mathbb{Z}x\mathbb{Z}\rangle)$

   Que dado un toroide válido t y una secuencia de posiciones, devuelve true si la secuencia contiene todas las celdas vivas del toroide, y ninguna otra celda.

   /* si la posición vecina se cae afuera de la matriz 2D, generamos una posición equivalente para simular un toroide evitando que se caiga */

   `aux nuevaF` (t : toroide, f : $\mathbb{Z}$) : $\mathbb{Z} = f \mod filas(t)$;

   `aux nuevaC` (t : toroide, c : $\mathbb{Z}$) : $\mathbb{Z} = c \mod columnas(t)$;

   /* asumimos que en "casillas" pueden haber tuplas fuera de la matriz pero que son equivalentes en el toroide, como se ejemplifica más abajo */

   `pred estanVivas` (t : toroide, casillas :$seq\langle \mathbb{Z}x\mathbb{Z}\rangle$) {
   $|casillas| = (\sum_{i=0}^{|casillas|-1}$ if $t[\ nuevaF(t, casillas[i]_0\ )\ ][\ nuevaC(t, casillas[i]_1\ )\ ] = True$ then 1 else 0 fi)
   }

   /* asumimos que en "vivas" no hay tuplas repetidas ni equivalentes (equivalentes serían por ejemplo: (4,1) y (1,1) en un toroide de 3 x 3) */

   `pred posicionesVivas` (t: toroide, vivas: $seq\langle \mathbb{Z}x\mathbb{Z}\rangle$) {
   $(cuantasVivas(t) = |vivas|) \wedge estanVivas(t, vivas)$
   }

4. 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.

   `aux tamaño` (t : toroide) : $\mathbb{Z} = filas(t) * columnas(t)$;

`aux densidadPoblacion` (t : toroide) : $\mathbb{R}$ $= \frac{cuantasVivas(t)}{tamaño(t)}$;

5. 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.

   `aux cantVecinosVivos` (t: toroide, f: $\mathbb{Z}$, c: $\mathbb{Z}$) : $\mathbb{Z}$ $=$

   $$(\sum_{i=f-1}^{f+1} \sum_{j=c-1}^{c+1} \text{if } t[\ nuevaF(t,i)\ ][\ nuevaC(t,j)\ ] = True \text{ then } 1 \text{ else } 0 \text{ fi }) -$$
   $$\text{if } t[f][c] = True \text{ then } 1 \text{ else } 0 \text{ fi };$$

6. Ejercicio 6 : $pred\ evolucionDePosicion(t : toroide, posicion : \mathbb{Z}x\mathbb{Z})$

   Que dado un toroide válido t y una posición válida del mismo, indique si dicha posición estaría viva luego de un tick.

   /* con 3 vecinos sigue viviendo o se reproduce, y si está viva y tiene 2 vecinos sigue viviendo, si está muerta con 2 vecinos, no llega a reproducirse */

   `pred evolucionDePosicion` (t: toroide, posicion: $\mathbb{Z}$x$\mathbb{Z}$) {
   $cantVecinosVivos(t, posicion_0, posicion_1) = 3 \lor$
   $(\ cantVecinosVivos(t, posicion_0, posicion_1) = 2 \land t[posicion_0][posicion_1] = True\ )$
   }

7. 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.

   `pred evolucionToroide` (t1: toroide, t2: toroide) {
   $(\ filas(t1) = filas(t2) \land columnas(t1) = columnas(t2)\ ) \land_L$
   $(\forall i : \mathbb{Z})(0 \leq i < filas(t1) \longrightarrow_L (\forall j : \mathbb{Z})(0 \leq j < columnas(t1) \longrightarrow_L$
   $evolucionDePosicion(t1, (i,j)\ ) = t2[i][j]))$
   }

2. Ejercicios con su Resolución Respectiva

8. 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.

   `proc evolucionMultiple` (in t: toroide, in k: $\mathbb{Z}$, out res : $\mathbb{Z}$) {
   `Pre` $\{k \geq 1 \land esValido(t)\}$
   `Post` $\{pasanNTiks(t, k, result)\}$
   }

   `pred pasanNTiks` (t: toroide, n: $\mathbb{Z}$, res: toroide) {
   $(\exists s : seq\langle toroide\rangle)(\ |s| = n{+}1 \land_L s[0] = t \land (\forall i : \mathbb{Z})(0 \leq i < n \longrightarrow_L evolucionToroide(s[i], s[i{+}1]) \land res = s[n])\ )$
   }

9. 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.

   `proc esPeriodico` (t: toroide, inout p: $\mathbb{Z}$, out result: Bool) {
   `Pre` $\{esValido(t)\}$
   `Post` $\{\ (\ esCiclico(t) \land result = true \land esElMenorP(p,t)\ ) \lor\ (\ \neg esCiclico(t) \land result = false\ )\ \}$
   }

pred esCiclico (t: toroide) {
$(\exists n : \mathbb{Z})(n \geq 1 \wedge_L pasanNTiks(t, n, t))$
}

pred esElMenorP (p: $\mathbb{Z}$, t: toroide) {
$p \geq 1 \wedge_L (pasanNTiks(t, p, t) \wedge \neg(\exists i : \mathbb{Z})(1 \leq i < p \wedge_L pasanNTiks(t, i, t)))$
}

10. 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.

proc primosLejanos (in t1: toroide, in t2: toroide, out primos: Bool) {
Pre $\{esValido(t1) \wedge esValido(t2)\}$
Post $\{ primos = true \leftrightarrow (\exists n : \mathbb{Z})(\ n \geq 1 \wedge_L (pasanNTiks(t1, n, t2) \vee pasanNTiks(t2, n, t1))\ )\ \}$
}

11. 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.

proc seleccionNatural (in ts: $seq\langle toroide\rangle$, out res: $\mathbb{Z}$) {
Pre $\{|ts| > 0 \wedge (\forall i : \mathbb{Z})(0 \leq i < |ts| \longrightarrow_L esValido(ts[i]))\}$
Post $\{0 \leq res < |ts|\ \wedge_L$
$(\exists n : \mathbb{Z})(\ (n \geq 1 \wedge \neg esCiclico(ts[res]))\ \wedge_L\ ticksHastaMorir(ts[res], n) \wedge esElMayor(ts, n)\ )\}$
}

pred esElMayor (ts : $seq\langle toroide\rangle$, n : $\mathbb{Z}$) {
$\neg(\exists k, i : \mathbb{Z})(\ (0 \leq i < |ts| \wedge k > n) \wedge_L ticksHastaMorir(ts[i], k)\ )$
}

/* n es la cantidad de ticks que da el toroide t hasta morir */
pred ticksHastaMorir (t : toroide, n : $\mathbb{Z}$) {
$(\exists s : seq\langle toroide\rangle)$
/* "armo" una secuencia de toroides tal que haya 1 tick de diferencia entre cada uno */
$(\ |t| = n + 1\ \wedge_L\ s[0] = t\ \wedge\ (\forall i : \mathbb{Z})(\ 0 \leq i < n \longrightarrow_L (\ evolucionToroide(s[i], s[i+1])\ \wedge$
/* finalmente me aseguro que muera en n chequeando que seguía vivo en n-1 */
$\neg toroideMuerto(s[n-1]) \wedge toroideMuerto(s[n]))\ )\ )$
}

12. 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.

proc fusionar (in t1: toroide, in t2: toroide, out res: toroide) {
/* todo tiene el mismo tamaño */
Pre $\{(esValido(t1) \wedge esValido(t2)) \wedge_L mismaDimensión(t1, t2)\}$
Post $\{mismaDimensión(res, t2) \wedge_L$
/* nueva fusionada */
$(\forall i, j : \mathbb{Z})(\ (0 \leq i < filas(t1) \wedge 0 \leq j < columnas(t1)) \longrightarrow_L$
$((\ t1[i][j] = true\ \wedge\ t2[i][j] = true\ \wedge\ res[i][j] = true\ )\ \vee$
$(\ \neg(t1[i][j] = true\ \wedge\ t2[i][j] = true)\ \wedge\ res[i][j] = false\ ))\ )$
}
}

```
pred mismaDimensión (t1: toroide, t2: toroide) {
```
$filas(t1) = filas(t2) \land columnas(t1) = columnas(t2)$

}

13. 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.

    proc vistaTrasladada (in t1: toroide, in t2: toroide, out res: Bool) {

    Pre $\{(esValido(t1) \land esValido(t2)) \land_L mismaDimensión(t1, t2)\}$

    Post $\{res = true \leftrightarrow esTrasladade(t1, t2)\}$

    }

    /* consideramos que si nos dan 2 toroides iguales, cuentan como vista trasladada */

    pred esTrasladade (t1: toroide , t2: toroide) {

    $(\exists n, k : \mathbb{Z})(\forall i, j : \mathbb{Z})((0 \leq i < filas[t1] \land 0 \leq j < columnas[t1]) \longrightarrow_L$
    $t2[i][j] = t1[nuevaF(t1, i + n)][nuevaC(t1, j + k)])$

    }

14. 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.

    proc menorSuperficieViva (in t: toroide, out res: $\mathbb{Z}$) {

    Pre $\{esValido(t)\}$

    Post {/* existe un toroide trasladado que cumple todas las condiciones */
    $(\exists s : toroide)(\ esTrasladade(t, s) \land_L$
    $(\exists f, c : \mathbb{Z})(res = f * c\ \land\ menorFPosible(s, f)\ \land\ menorCPosible(s, c))\ ) \land$
    /* y no existe un toroide trasladado que cumple todas las condiciones y, además, tenga un res más chico */
    $\neg(\exists s : toroide)(\ esTrasladade(t, s) \land_L$
    $(\exists f, c, o : \mathbb{Z})(o < res \land o = f * c\ \land\ menorFPosible(s, f)\ \land\ menorCPosible(s, c))\ )$
    }

    }

    pred menorCPosible (t: toroide, c: $\mathbb{Z}$) {
    $(\exists a, b : \mathbb{Z})(0 \leq a, b < columnas[t] \land_L (aYbSonCorrectas(t, a, b) \land c = a - b)) \land$
    $\neg(\exists a, b, d : \mathbb{Z})(d < c\ \land\ 0 \leq a, b < columnas[t] \land_L (aYbSonCorrectas(t, a, b) \land d = a - b))$
    }

    pred aYbSonCorrectas (t: toroide, a,b: $\mathbb{Z}$) {
    /* a es la viva más alta */
    $(\exists i : \mathbb{Z})(0 \leq i < filas[t] \land_L t[i][a] = true) \land$
    $(\forall i, j : \mathbb{Z})(\ (0 \leq i < filas[t]\ \land\ a < j < columnas[t]) \longrightarrow_L t[i][j] = false) \land$
    /* b es la viva más baja */
    $(\exists i : \mathbb{Z})(0 \leq i < filas[t] \land_L t[i][b] = true) \land$
    $(\forall i, j : \mathbb{Z})(\ (0 \leq i < filas[t]\ \land\ 0 \leq j < b) \longrightarrow_L t[i][j] = false)$
    }

    pred menorFPosible (t: toroide, f: $\mathbb{Z}$) {
    $(\exists i, d : \mathbb{Z})(0 \leq i, d < filas[t] \land_L (iYdSonCorrectas(t, i, d) \land f = d - i)) \land$
    $\neg(\exists i, d, a : \mathbb{Z})(a < f\ \land\ 0 \leq i, d < filas[t] \land_L (iYdSonCorrectas(t, i, d) \land a = d - i)\ )$
    }

    pred iYdSonCorrectas (t: toroide, i,d: $\mathbb{Z}$) {
    /* i es la viva más a la izquierda */

$(\exists j : \mathbb{Z})(0 \leq j < columnas[t] \wedge_L t[i][j] = true) \wedge$
$(\forall j, k : \mathbb{Z})(\ (0 \leq k < i \wedge 0 \leq j < columnas[t]) \longrightarrow_L t[k][j] = false) \wedge$
/* d es la viva más a la derecha */
$(\exists j : \mathbb{Z})(0 \leq j < columnas[t] \wedge_L t[d][j] = true) \wedge$
$(\forall j, k : \mathbb{Z})(\ (d < k < filas[t] \wedge 0 \leq j < columnas[t]) \longrightarrow_L t[k][j] = false)$
}