



TP de Especificación

Análisis Habitacional Argentino

26 de Agosto de 2019

Algoritmos y Estructuras de Datos I

Grupo 25

Integrante	LU	Correo electrónico
Yulita, Federico	351/17	fyulita@dc.uba.ar
Chanes, Mauricio	226/19	mchanes@dc.uba.ar



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 esImagenValida (in A : imagen , out result : Bool) {
  Pre {true}
  Post {esValida(A) ∧ esBinaria(A) ↔ result = true}
}

pred esValida (A: imagen) {(∀i : Z)(0 ≤ i < filas(A)) →L (|A[i]| > 0 ∧ (∀j : Z)(0 ≤ j < filas(A) →L
|A[i]| = |A[j]|))}

pred esBinaria (A: imagen) {(∀s : seq(datos))(s ∈ A → (∀d : dato)(d ∈ s → ((d = 1) ∨ (d = 0))))}

proc sonPixelesConectados (in A : imagen, in p0 : pixel, in p1 : pixel, in k : Z, out result : Bool){
  Pre {esImagenValida(A) ∧ pixelEnImagen(p0, A) ∧ pixelEnImagen(p1, A) ∧ (k = 4 ∨ k = 8)}
  Post {result = true ↔ (∃s : sqPixel)(secuenciaValida(s, A) ∧ secuenciaConectada(s, k, A) ∧ p0 ∈ s ∧ p1 ∈ s)}
}

pred secuenciaConectada (s : sqPixel, k : Z, A : imagen) {sqPixelesActivado(s, A) ∧ (∀i : Z)(1 ≤ i < |s| - 1
→L sonVecinos(s[i], s[i + 1], k))}

proc esFormaConvexa (in A : imagen, in k : Z, out result : Bool) {
  Pre {esImagenValida(A) ∧ (k = 4 ∨ k = 8)}
  Post {result = true ↔ ¬(∃p, q : pixel)(pixelActivado(p, A) ∧ pixelActivado(q, A)
  ∧ (∃r : sqPixel)(esRectaDe(r, p, q, A) ∧ ¬sqPixelesActivados(r, A)))}
}

pred esRectaDe (r : sqPixel, p : pixel, q : pixel, A : imagen) {secuenciaValida(r, A) ∧ (p ∈ r) ∧ (q ∈ r) ∧ (∀z : pixel)
(z ∈ r →L (∃x : R)((z[1] ≤ x < z[1] + 1) ∧ (z[0] ≤ pendiente(p, q) * x + ordenada(p, q) < z[0] + 1)))}

aux pendiente (p : pixel, q : pixel) : R = (p[0] - q[0]) / (p[1] - q[1]);

aux ordenada (p : pixel, q : pixel) : R = q[1];

proc devolverPromedioAreas (in A: imagen, in k: Z, out prom: R) {
  Pre {esImagenValida(A) ∧ (k = 4 ∨ k = 8)}
  Post {(noHayRegion(A) → prom = 0) ∨ (¬noHayRegion(A) → prom =
  sumaPixelesActivados(A) / cantidadDeRegiones(A, convertirImagenSecuencia(A), k))}
}

pred noHayRegion (A: Imagen, k: Z) {¬(∃s : sqPixel)(esRegion(s, A, k))}

aux sumaPixelesActivados (A: imagen) : R =  $\sum_{i=0}^{|A|-1} \sum_{j=0}^{|A[i]|-1} A[i][j]$ ;

aux sumaDeRegiones (A: imagen, s: sqPixel, k: Z) : R =  $\sum_{i=0}^{|s|} \text{if } ((\forall j : Z)(0 \leq j < |s| - 1)$ 
   $(\text{sonVecinos}(s[i], s[j], k))) \text{ then } 1 \text{ else } 0 \text{ fi}$ ;

proc calcularContorno (in A : imagen, in k : Z, out edge : seq (sqPixel)){
  Pre {regionesValidas(A, k) ∧ regionImagen(A, k) ∧ (k = 4 ∨ k = 8)}
  Post {(∀s : sqPixel)((s ∈ edge → esBorde(s, A, k)) ∧ ¬(∃l : sqPixel)((|l| > largoTotal(edge)) ∧ esBorde(l, A, k)))}
}

pred esBorde (s : sqPixel, A : imagen, k : Z) {sqPixelesActivados(s, A) ∧ (∀p : pixel)(p ∈ s → (∃q : pixel)
(sonVecinos(p, q, k) ∧ ¬pixelActivado(q, A)))}

```

```

pred regionesValidas (A : imagen, k :  $\mathbb{Z}$ )  $\{(\forall s : sqPixel)(esRegion(s, A, k) \longrightarrow |s| \geq 2)\}$ 

pred regionImagen (A : imagen, k :  $\mathbb{Z}$ )  $\{(\exists! s : sqPixel)(esRegion(s, A, k))\}$ 

proc cerrarForma (in A : imagen, in b : imagen, out c : seq  $\langle sqPixel \rangle$ ) {
  Pre  $\{esImagenValida(A) \wedge (|A| > 0) \wedge esElemEstruc(b)\}$ 
  Post  $\{(\forall i : \mathbb{Z})(0 \leq i < |c| - 1 \longrightarrow_L closing(c[i - 1], c[i], b))\}$ 
}

pred closing (si : sqPixel, so : sqPixel, b : imagen)  $\{(\exists sa : sqPixel)(erosion(dilatacion(si, sa, b), so, b))\}$ 

pred erosion (si : sqPixel, so : sqPixel, b : imagen)  $\{(\forall p : pixel)$ 
   $(pixelActivado(p, elemEstrucCentrado(so[|so| - 1], b)) \longrightarrow p \in si)\}$ 

proc obtenerRegionConectada (A: imagen, in semilla: pixel, out seq: seq  $\langle Imagen \rangle$ ) {
  Pre  $\{esImagenValida(A) \wedge pixelActivado(semilla, A)\}$ 
  Post  $\{(\forall p : pixel)((\exists B : imagen)(esImagenValida(B) \wedge \neg esActiva(B) \wedge |B| = |A|)(\neg sonVecinos(semilla, p, 8) \longrightarrow$ 
     $seq = \langle A, dePixelAImagen(B, semilla) \rangle \vee ((\exists s : sqPixel)(\forall i : \mathbb{Z})(0 \leq i < |s| - 1 \wedge_L sonVecinos(semilla, s[i], 8))$ 
     $\longrightarrow seq = \langle A, desqPixelAImagen(B, \langle semilla \rangle + + s) \rangle)\}$ 
}

aux dePixelAImagen (A: imagen, p: pixel) : Imagen = setAt(A[p[0]], A[p[0]] [p[1]], 1;

aux desqPixelAImagen (A: imagen, s: sq  $\langle Pixel \rangle$ ) : imagen =  $(\forall i : \mathbb{Z})(0 \leq i < |s| - 1 \longrightarrow_L dePixelAImagen(A, s[i]));$ 

proc esMaximoDiscoEsqueleto (in A : imagen, in S : imagen, in Dz : imagen, out res : Bool) {
  Pre  $\{esImagenValida(A) \wedge esElemEstruc(D_z) \wedge esImagenValida(S)\}$ 
  Post  $\{esqueletiza(A, S, D_z) \wedge \neg(\exists D : imagen)(esqueletiza(A, S, D_z) \wedge (\exists l, l_z : seq \langle sqPixel \rangle)$ 
     $(largoTotal(convertirImagenSecuencia(D, l)) > largoTotal(convertirImagenSecuencia(D_z, l_z))))\}$ 
}

pred esqueletiza (A : imagen, S : imagen, Dz : imagen)  $\{(\exists si, so : sqPixel)$ 
   $(S = interseccion(A, convertirImagenSecuencia(dilatacion(convertirImagenSecuencia(A, si), so, Dz))))\}$ 

pred convertirImagenSecuencia (A : imagen, s : sqPixel)  $\{secuenciaEnImagen(s, A) \wedge \neg(\exists l : sqPixel)$ 
   $(secuenciaEnImagen(l, A) \wedge |l| > |s|)\}$ 

pred interseccion (A : imagen, S: imagen, b : imagen)  $\{(\exists p : pixel)((\forall q : pixel)$ 
   $(pixelActivado(q, S) \longrightarrow pixelActivado(q, A) \wedge pixelActivado(q, elemEstrucCentrado(p, b))))\}$ 

```

2. Predicados y Auxiliares generales

```

aux filas (m : A) :  $\mathbb{Z} = |m|;$ 

aux columnas (m : A) :  $\mathbb{Z} = \text{if } filas(A) > 0 \text{ then } |m[0]| \text{ else } 0 \text{ fi};$ 

pred pixelEnImagen (p : pixel, A : imagen)  $\{(0 \leq p[0] < filas(A)) \wedge (0 \leq p[1] < columnas(A))\}$ 

pred secuenciaEnImagen (s : sqPixel, A : imagen)  $\{(\forall p : pixel)(p \in s \longrightarrow pixelEnImagen(p, A))\}$ 

pred secuenciaValida (s : sqPixel, A : imagen)  $\{esImagenValida(A) \wedge secuenciaEnImagen(s, A)\}$ 

pred sqPixelesActivados (s : sqPixel, A : imagen)  $\{secuenciaEnImagen(s, A) \wedge (\forall p : pixel)(p \in s \longrightarrow pixelActivado(p, A))\}$ 

pred pixelActivado (p : pixel, A : imagen)  $\{pixelEnImagen(p, A) \wedge A[p[0]][p[1]] = 1\}$ 

```

```

pred sonVecinos (p : pixel, q : pixel, k :  $\mathbb{Z}$ )  $\{((k = 4) \wedge (|p[0] - q[0]| + |p[1] - q[1]| = 1)) \vee ((k = 8) \wedge (1 \leq |p[0] - q[0]| + |p[1] - q[1]| \leq 2))\}$ 

pred esRegion (s : sqPixel, A : Imagen, k :  $\mathbb{Z}$ )  $\{(\forall p, q : pixel)((p \in s \wedge q \in s) \longrightarrow sonPixelesConectados(A, p, q, k)) \wedge \neg(\exists l : sqPixel)(esSubsecuencia(s, l) \wedge (\forall p, q : pixel)((p \in l \wedge q \in l) \longrightarrow sonPixelesConectados(A, p, q, k)))\}$ 

pred esSubsecuencia (s : sqPixel, l : sqPixel)  $\{(|s| < |l|) \wedge (\forall p : pixel)(p \in s \longrightarrow p \in l)\}$ 

pred esElemEstruc (b : imagen)  $\{esCuadradaImpar(b) \wedge esActiva(b)\}$ 

pred esCuadradaImpar (b : imagen)  $\{(\forall s : seq(dato))(s \in b \longrightarrow (|b| = |s|)) \wedge (|b| \bmod 2 = 1) \wedge (|b| > 0)\}$ 

pred esActiva (b : imagen)  $\{(\forall p : pixel)(pixelEnImagen(p, b) \longrightarrow pixelActivado(p, b))\}$ 

aux largoTotal (s : seq(sqPixel)) :  $\mathbb{Z} = \sum_{i=0}^{|s|-1} |s[i]|;$ 

pred dilatacion (si : sqPixel, so : sqPixel, b : imagen)  $\{(\exists p : pixel)((p \in si) \wedge (pixelActivado(p, elemEstrucCentrado(so[|so| - 1], b))))\}$ 

pred elemEstrucCentrado (p : pixel, b : imagen)  $\{(p[0] = (|b| - 1)/2) \wedge (p[1] = (|b| - 1)/2)\}$ 

```

3. Decisiones tomadas

- Para el predicado *secuenciaConectada* se tomó implícitamente que la secuencia va a considerarse conectada sólo si además de tener un conjunto de pixeles que están conectados en la secuencia están en orden. Esto lo hicimos ya que usamos la definición de trayectoria de secuencia del pdf.
- Tomamos que los únicos valores válidos de adyecencia k son $k = 4$ y $k = 8$ ya que fueron los únicos descriptos en el pdf.
- Al procedimiento 3 (*esFormaConvexa*) le agregamos el input k ya que dependiendo de qué forma de adyecencia querramos las rectas se definen de forma distinto y, por lo tanto, las regiones convexas se definen distinto.
- Para el predicado *esSubsecuencia* consideramos que una secuencia no es una subsecuencia de sí misma por motivos prácticos.
- Para los predicados dilatación, erosión y closing se tomó que los pixeles por los que se va trasladando el elemento estructurante están en orden.