Trabajo practico 1

Especificacion y Weakest Precondition

13 de septiembre de 2024 — Algoritmos y Estructura de Datos

Grupo HIBTAYIGAFWKBCHZLZPJ

Integrante	LU	Correo electrónico
Dominguez, Mateo Felipe	923/24	matedominguez2@gmail.com
Morrone, Valentina	35/24	valenmorrone@hotmail.com
Moran, Juana Gala	119/24	juanagalamoran.u@gmail.com
Cuiña, Carolina	874/24	cuinacarolina43@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. Especificacion

1.1. grandesCiudades

`proc grandesCiudades` (in ciudades: $seq\langle Ciudad\rangle$) : $seq\langle Ciudad\rangle$
  `requiere` $\{\neg ciudadesRepetidas(ciudades)\}$
  `asegura` $\{\neg ciudadesRepetidas(res) \land (\forall i : \mathbb{Z})(0 \leq i < |res| \longrightarrow_L res[i] \in ciudades \land res[i][1] > 50{,}000)\}$

`pred ciudadesRepetidas` (s: $seq\langle Ciudad\rangle$) {
  $(\forall i : \mathbb{Z})(0 \leq i < |s| \longrightarrow_L (\exists j : \mathbb{Z})(0 \leq j < |s| \land i \neq j \land_L s[i] = s[j]))$
}

1.2. sumaDeHabitantes

`proc sumaDeHabitantes` (in menoresDeCiudades: $seq\langle Ciudad\rangle$, in mayoresDeCiudades: $seq\langle Ciudad\rangle$) : $seq\langle Ciudad\rangle$
  `requiere` $\{\neg ciudadesRepetidas(menoresDeCiudades) \land \neg ciudadesRepetidas(mayoresDeCiudades)\}$
  `requiere` $\{mismasCiudades(menoresDeCiudades, mayoresDeCiudades)\}$
  `asegura` $\{mismasCiudades(res, menoresDeCiudades)\}$
  `asegura` $\{\neg ciudadesRepetidas(res)\}$
  `asegura` $\{(\forall i, j : \mathbb{Z})(0 \leq i, j < |menoresDeCiudades| \land_L menoresDeCiudades[i][0] = mayoresDeCiudades[j][0] \longrightarrow_L$
  $(res[i][1] = menoresDeCiudades[i][1] + mayoresDeCiudades[j][1]))))\}$

`pred mismasCiudades` (s: $seq\langle Ciudad\rangle$, t: $seq\langle Ciudad\rangle$) {
  $(\forall i : \mathbb{Z})(0 \leq i < |s| \longrightarrow_L (\exists j : \mathbb{Z})(0 \leq j < |t| \land_L s[i][0] = t[j][0]))$
}

1.3. hayCamino

`proc hayCamino` (in distancias: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, in desde: $\mathbb{Z}$, in hasta: $\mathbb{Z}$) : Bool
  `requiere` $\{esCuadrada(distancias)\}$
  `requiere` $\{(\forall i, j : \mathbb{Z})(0 \leq i, j < |s| \longrightarrow_L s[i][j] = s[j][i] \land s[i][j] \geq 0)\}$
  `requiere` $\{0 \leq desde, hasta < |distancias|\}$
  `asegura` $\{res = True \iff (\exists\ camino : seq\langle \mathbb{Z}\rangle)(esCamino(distancias, camino, desde, hasta))\}$

`pred esCuadrada` (A : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
  $(\forall i : \mathbb{Z})(0 \leq i < |A| \longrightarrow_L |A[i]| = |A|)$
}
`pred esCamino` (s: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, t: $seq\langle \mathbb{Z}\rangle$, n: $\mathbb{Z}$, m: $\mathbb{Z}$) {
  $(2 \leq |t| \leq |s| \land_L t[0] = n \land_L t[|t| - 1] = m \land_L (\forall i : \mathbb{Z})(0 \leq i < |t| - 1 \longrightarrow_L 0 \leq t[i] < |s| \land s[t[i]][t[i + 1]] > 0))$
}

1.4. cantidadCaminosNSaltos

`proc cantidadCaminosNSaltos` (inout conexion: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, in n: $\mathbb{Z}$)
  `requiere` $\{conexion = C_0\}$
  `requiere` $\{esCuadrada(C_0)\}$
  `requiere` $\{(\forall i, j : \mathbb{Z})(0 \leq i, j < |C_0| \longrightarrow_L (C_0[i][j] = C_0[j][i] \land (C_0[i][j] = 0 \lor C_0[i][j] = 1)))\}$
  `requiere` $\{(\forall i, j : \mathbb{Z})(0 \leq i, j < |C_0| \land i = j \longrightarrow_L C_0[i][j] = 0)\}$
  `requiere` $\{n \geq 1\}$
  `asegura` $\{|conexion| = |C_0|\}$
  `asegura` $\{(\forall i, j : \mathbb{Z})(0 \leq i, j < |conexion| \longrightarrow_L conexion[i][j] = conexion[j][i])\}$
  `asegura` $\{(\exists t : seq\langle seq\langle seq\langle \mathbb{Z}\rangle\rangle\rangle)(|t| = n \land t[0] = C_0 \land_L (\forall i, j : \mathbb{Z})(0 \leq i < |t| - 1 \land 0 \leq j < |t| \longrightarrow_L |t[i]| = |t[j]| \land$
  $esCuadrada(t[j]) \land APorBEsC(t[i], C_0, t[i + 1])) \land conexion = t[n - 1])\}$

`pred APorBEsC` (A : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, B : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, C : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
  $(\forall i, j : \mathbb{Z})(0 \leq i, j < |C| \longrightarrow_L (C[i][j] = \sum_{k=0}^{|C|-1} A[i][k] * B[k][j]))$
}

1.5. caminoMinimo

`proc caminoMinimo` (`in` origen: $\mathbb{Z}$, `in` destino: $\mathbb{Z}$, `in` distancias: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) : $seq\langle\mathbb{Z}\rangle$

`requiere` $\{esCuadrada(distancias)\}$

`requiere` $\{(\forall i,j:\mathbb{Z})(0 \leq i,j < |s| \longrightarrow_L s[i][j] = s[j][i] \land s[i][j] \geq 0)\}$

`requiere` $\{0 \leq origen, destino < |distancias|\}$

`asegura` $\{res = \langle\rangle \iff \neg(\exists\ camino : seq\langle\mathbb{Z}\rangle)(esCamino(distancias, camino, origen, hasta))\}$

`asegura` $\{esCamino(distancias, res, origen, destino) \land esMinimo(distancias, res, origen, destino)\}$

`pred esMinimo` (s: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, t: $seq\langle\mathbb{Z}\rangle$, n: $\mathbb{Z}$, m: $\mathbb{Z}$) {

$(\forall w : seq\langle\mathbb{Z}\rangle)(esCamino(s,w,n,m) \longrightarrow_L sumaDistancias(s,t) \leq sumaDistancias(s,w))$

}

`aux sumaDistancias` (s: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, t: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= \sum_{i=0}^{|t|-2} s[t[i]][t[i+1]]$;

2. Demostraciones de correctitud

2.1. Demostrar que la implementacion es correcta con respecto a la especificacion

```
res = 0
i = 0
while (i < ciudades.length) do
    res = res + ciudades[i].habitantes
    i = i + 1
endwhile
```

Demostrar {P}S{Q}

Supongamos ciudades = [(a,10),(b,15)]

Iteracion	i	res
0	0	0
1	1	10
2	2	10+15=25

Tabla 1: Tabla de iteracion

$I \equiv 0 \leq i \leq |ciudades| \land res = \sum_{j=0}^{i-1} ciudades[j].habitantes$

$P_c \equiv \{res = 0 \land i = 0\}$

$Q_c \equiv \{res = \sum_{i=0}^{|c|-1} ciudades[i].habitantes\}$

$B = i < |ciudades|$

Paso 1

$P_c \implies I$?

$res = 0 \land i = 0 \implies 0 \leq i < |ciudades| \land res = \sum_{j=0}^{i-1} ciudades[j].habitantes$

Analizamos variable a variable:

- $i = 0 \implies 0 \leq i < |ciudades|$
- $res = \sum_{j=0}^{i-1} ciudades[j].habitantes = \sum_{j=0}^{-1} ciudades[j].habitantes = 0 \implies res = 0$

Paso 2

$\{I \land B\}S\{I\}$

$\{I \land B\} \implies wp(S, I)$?

- $\{I \land B\} = \{0 \leq i < |ciudades| \land \sum_{j=0}^{i-1} ciudades[j].habitantes\}$
- $wp(S_1; S_2, I) = wp(S_1, wp(S_2, I))$

1. $wp(S_2, I) = wp(i := i+i, 0 \leq i \leq |ciudades| \wedge res = \sum\limits_{j=0}^{i-1} ciudades[j].habitantes) \equiv$

   $def(i+1) \wedge_L 0 \leq i+1 \leq |ciudades| \wedge_L res = \sum\limits_{j=0}^{i} ciudades[j].habitantes \equiv$

   $0 \leq i+1 \leq |ciudades| \wedge res = \sum\limits_{j=0}^{i} ciudades[j].habitantes$

2. $wp(S_1, wp(S_2, I)) \equiv wp(res := res + ciudades[i].habitantes, 0 \leq i+1 \leq |ciudades| \wedge res = \sum\limits_{j=0}^{i} ciudades[j].habitantes) \equiv$

   $def(ciudades) \wedge def(ciudades[i].habitantes) \wedge_L 0 \leq i+1 \leq |ciudades| \wedge_L res + ciudades[i].habitantes = \sum\limits_{j=0}^{i} ciudades[j].habitant$

   $0 \leq i < |ciudades| \wedge 0 \leq i+1 \leq |ciudades| \wedge res + ciudades[i].habitantes = \sum\limits_{j=0}^{i} ciudades[j].habitantes \equiv$

   $0 \leq i \leq |ciudades| - 1 \wedge res + ciudades[i].habitantes = \sum\limits_{j=0}^{i} ciudades[j].habitantes \equiv$

   $0 \leq i \leq |ciudades| - 1 \wedge res = \sum\limits_{j=0}^{i} ciudades[j].habitantes - ciudades[i].habitantes \equiv$

   $0 \leq i \leq |ciudades| - 1 \wedge res = \sum\limits_{j=0}^{i-1} ciudades[j].habitantes$

Paso 3

$I \wedge \neg B \implies Q$?

1. $(0 \leq i < |ciudades| \wedge res = \sum\limits_{j=0}^{i-1} ciudades[j].habitantes \wedge i \geq |ciudades|) \implies res = \sum\limits_{j=0}^{|c|-1} ciudades[j].habitantes \equiv$

   $(i = |ciudades| \wedge res = \sum\limits_{j=0}^{i-1} ciudades[j].habitantes) \implies res = \sum\limits_{j=0}^{|c|-1} ciudades[j].habitantes \equiv$

   $res = \sum\limits_{j=0}^{|c|-1} ciudades[j].habitantes \implies res = \sum\limits_{j=0}^{|c|-1} ciudades[j].habitantes$

Teorema de terminacion

$f_v = |ciudades| - i$

Paso 1

$\{I \wedge B \wedge f_v = v_0\} S \{f_v < v_0\}$

$Verificamos\ \forall\ v_0,\ wp(S, f_v < v_0)$

1. $wp(S_1, wp(S_2, f_v < v_0)) \equiv$
   $wp(res := res + ciudades[i].habitantes, wp(i := i+1, |ciudades| - i < V_0)) \equiv$
   $wp(res := res + ciudades[i].habitantes, def(i) \wedge def(1)) \wedge_L |ciudades| - (i+1) < V_0 \equiv$
   $wp(res := res + ciudades[i].habitantes, |ciudades| - i - 1 < V_0) \equiv$
   $def(res) \wedge def(ciudades) \wedge 0 \leq i < |ciudades| \wedge_L |ciudades| - i - 1 < V_0 \equiv$
   $0 \leq i < |ciudades| \wedge_L |ciudades| - i - 1 < |ciudades| \equiv True$

Paso 2

$\{I \wedge f_v < 0\} \implies \neg B$?

1. $0 \leq i < |ciudades| \wedge res = \sum\limits_{j=0}^{i-1} ciudades.[i].habitantes \wedge |ciudades| - i < 0 \implies i \geq |ciudades| \equiv$

   $i < |ciudades| \wedge res = \sum\limits_{j=0}^{i-1} ciudades.[i].habitantes \wedge |ciudades| < i \implies i \geq |ciudades| \equiv$

   $i = |ciudades| \wedge res = \sum\limits_{j=0}^{i-1} ciudades.[i].habitantes \implies i \geq |ciudades| \equiv$

   $i = |ciudades| \implies i \geq |ciudades|$

2.2. Demostrar que el valor devuelto es mayor a 50.000

$P_c \equiv wp(S_1, Q_1)$
$S_1 \equiv res := res + ciudades[i].habitantes$
$Q_1 \equiv wp(S_2, Q_c)$
$S_2 \equiv i := i + 1$
$Q_c \equiv res = \sum_{j=0}^{|c|-1} ciudades[j].habitantes \wedge res > 50.000$

1. $wp(S_2, Q_c) \equiv wp(i := i + 1, res = \sum_{j=0}^{|c|-1} ciudades[j].habitantes \wedge res > 50.000) \equiv$

   $def(i) \wedge def(1) \wedge res = \sum_{j=0}^{|c|-1} ciudaades[j].habitantes \wedge res > 50.000 \equiv$

   $res = \sum_{j=0}^{|c|-1} ciudades[j].habitantes \wedge res > 50.000$

2. $wp(S_1, Q_1) \equiv wp(res = res + ciudades[i].habitantes, res = \sum_{j=0}^{|c|-1} ciudades[i].habitantes \wedge res > 50.000) \equiv def(i) \wedge$

   $def(c) \wedge (o \leq i < |c|) \wedge_L res + ciudades[i].habitantes = \sum_{j=0}^{|c|-1} ciudades[j].habitantes \wedge res + ciudades[i].habitantes >$
   $50.000 \equiv$
   $(0 \leq i < |c|) \wedge_L res + ciudades[i].habitantes = \sum_{j=0}^{|c|-1} ciudades[j].habitantes \wedge res + ciudades[i].habitantes > 50.000$

$Proponemos:$
$P_c \equiv res = 0 \wedge i = 0 \wedge \sum_{j=0}^{|c|-1} ciudades[j].habitantes > 50.000$

$Vemos:$
$P_c \implies Q_c$
$(res = 0 \wedge i = 0 \wedge \sum_{j=0}^{|c|-1} ciudades[j].habitantes > 50.000) \implies res = \sum_{j=0}^{|c|-1} ciudades[j].habitantes > 50.000$
$P_c \implies I$
$(res = 0 \wedge i = 0 \wedge \sum_{j=0}^{|c|-1} ciudades[j].habitantes > 50.000) \implies (0 \leq i \leq |c| \wedge res = \sum_{j=0}^{i-1} ciudades[j].habitantes)$

$Por\ ultimo, tenemos\ que:$
$P \equiv (\exists i : \mathbb{Z})(0 \leq i < |ciudades| \wedge_L ciudades[i].habitantes > 50.000) \wedge (\forall i : \mathbb{Z})(0 \leq i < |ciudades| \longrightarrow_L ciudades[i].habitantes \geq 0) \wedge (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < j < |ciudades| \longrightarrow_L ciudades[i].nombre = ciudades[j].nombre)$
$Veamos:$
$P_c \implies P\ ?$
$(res = 0 \wedge i = 0) \implies i = 0 \implies P$