DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Especificación y wp

En búsqueda del camino

6 de septiembre de 2024 — Algoritmos y estructuras de datos

Grupo datapath

Integrante	LU	Correo electrónico
Andreone, Joaquin	122/24	jandreone06@gmail.com
Comerci, Lucas	818/24	lukicomerci@gmail.com
Luis, Theo	130/23	theoluis44@gmail.com
Zea, Marcos	405/09	zea.marcos@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. Especificaciones

proc grandesCiudades (in ciudades : $seq\langle Ciudad\rangle$) : $seq\langle Ciudad\rangle$
requiere {$sinRepetidos(ciudades)$}
asegura {$|res| \leq |ciudades| \wedge sinRepetidos(res)$}
asegura {$(\forall i : \mathbb{Z})\ (0 \leq i < |ciudades| \wedge_L ciudades[i]_1 \geq 50000 \longrightarrow_L ciudades[i] \in res)$}

proc sumaDeHabitantes (in menoresDeCiudades: $seq\langle Ciudad\rangle$, in mayoresDeCiudades: $seq\langle Ciudad\rangle$) : $seq\langle Ciudad\rangle$
requiere {$sinRepetidos(menoresDeCiudades), sinRepetidos(mayoresDeCiudades),$
$mismasCiudades(menoresDeCiudades, mayoresDeCiudades)$}
asegura {$|res| = |menoresDeCiudades|$}
asegura {$(\forall i, j : \mathbb{Z})\ ((0 \leq i < |menoresDeCiudades| \wedge 0 \leq j < |mayoresDeCiudades|) \wedge_L$
$menoresDeCiudades[i]_0 = mayoresDeCiudades[j]_0 \longrightarrow_L (\exists k : \mathbb{Z})\ (0 \leq k < |res| \wedge$
$(res[k]_0 = menoresDeCiudades[k]_0 \wedge res[k]_1 = menoresDeCiudades[i]_1 + mayoresDeCiudades[j]_1)))$}

proc hayCamino (in distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, in desde: $\mathbb{Z}$, in hasta: $\mathbb{Z}$) : Bool
requiere {$0 \leq desde, hasta < |distancias| \wedge esSimetrica(distancias) \wedge tieneDiagonalNula(distancias)$}
asegura {$res = \text{true} \leftrightarrow (\exists camino : seq\langle \mathbb{Z}\rangle)\ (esCamino(distancias, desde, hasta, camino))$}

proc cantidadCaminosNSaltos (inout conexion : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, in n: $\mathbb{Z}$)
requiere {$esMatrizDeConexion(conexion) \wedge n \geq 1 \wedge conexion = Conexion_0$}
asegura {$mismasDimensiones(conexion, Conexion_0)$}
asegura {$(\exists potenciasDeConexion : seq\langle seq\langle seq\langle \mathbb{Z}\rangle\rangle\rangle)\ (|potenciasDeConexion| = n \wedge$
$potenciasDeConexion[0] = Conexion_0 \wedge todasMismasDimensiones(potenciasDeConexion, Conexion_0) \wedge$
$sonTodasPotenciasDe(potenciasDeConexion, Conexion_0) \wedge conexion = potenciasDeConexion[n-1])$}

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 {$0 \leq origen < |distancias| \wedge 0 \leq destino < |distancias| \wedge esSimetrica(distancias)$}
asegura {$esCamino(distancias, origen, destino, res) \vee res = seq\langle\rangle$}
asegura {$(\forall camino : seq\langle \mathbb{Z}\rangle)\ (esCamino(distancias, origen, destino, camino) \longrightarrow_L$
$longitudDeCamino(res, distancias) \leq longitudDeCamino(camino, distancias))$}

1.1. Predicados y funciones auxiliares globales

pred sinRepetidos (lista: $seq\langle Ciudad\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |lista| \longrightarrow_L (\forall j : \mathbb{Z})\ (0 \leq j < |lista| \wedge Ciudad[i] = Ciudad[j] \longrightarrow_L j = i))$
}
pred esSimetrica (matriz:$seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |matriz| \longrightarrow_L |matriz[i]| = |matriz|) \wedge (\forall i, j : \mathbb{Z})\ (0 \leq i, j < |matriz| \wedge_L matriz[i][j] = matriz[j][i])$
}
pred tieneDiagonalNula (matriz: $seq\langle seq\langle\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |matriz| \wedge i = j \longrightarrow_L matriz[i][j] = 0)$
}
pred mismasCiudades (in ciudades: $seq\langle Ciudad\rangle$, in otrasCiudades: $seq\langle Ciudad\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |ciudades| \longrightarrow_L (\exists j : \mathbb{Z})\ (0 \leq j < |otrasCiudades| \wedge_L ciudades[i]_0 = otrasCiudades[j]_0))$
}
pred esCamino (distancias: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, desde: $\mathbb{Z}$, hasta: $\mathbb{Z}$, camino: $seq\langle \mathbb{Z}\rangle$) {
$desde \in camino \wedge hasta \in camino \wedge desde \neq hasta \wedge (\forall i : \mathbb{Z})\ (0 \leq i < |camino| \longrightarrow_L 0 \leq camino[i] < |distancias|) \wedge$
$(\forall i : \mathbb{Z})\ (0 \leq i < |camino| - 1 \longrightarrow_L distancias[camino[i]][camino[i+1]] \neq 0)$
}
pred esMatrizDeConexion (matriz: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$esSimetrica(matriz) \wedge tieneDiagonalNula(matriz) \wedge (\forall i, j : \mathbb{Z})\ (0 \leq i, j < |matriz| \wedge i \neq j \longrightarrow_L$
$(matriz[i][j] = 1 \vee matriz[i][j] = 0))$
}
pred esProductoDe (matrizA: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, matrizB: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, matriz: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$mismasDimensiones(matrizA, matrizB) \wedge mismasDimensiones(matrizA, matriz) \wedge$
$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |matriz| \longrightarrow_L matriz[i][j] = \sum_{k=0}^{|A|-1} matrizA[i][k] \cdot matrizB[k][j])$
}
pred mismasDimensiones (matrizA: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, matrizB: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$|matrizA| = |matrizB| \wedge (\forall i : \mathbb{Z})\ (0 \leq i < |matrizA| \longrightarrow_L |matrizA[i]| = |matrizB[i]|)$
}
pred todasMismasDimensiones (matrices: $seq\langle seq\langle seq\langle \mathbb{Z}\rangle\rangle\rangle$, matriz: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |matrices| \longrightarrow_L mismasDimensiones(matrices[i], matriz))$

}
pred sonTodasPotenciasDe (matrices: $seq\langle seq\langle seq\langle \mathbb{Z}\rangle\rangle\rangle$, matriz: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (1 \leq i < |matrices| \longrightarrow esProductoDe(matriz, matrices[i-1], matrices[i]))$
}
aux longitudDeCamino (camino: $seq\langle \mathbb{Z}\rangle$, distancias: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) : $\mathbb{Z}$ =
$\sum_{i=0}^{|camino|-2} distancias[camino[i]][camino[i+1]]$;

2. Demostraciones de correctitud

```
1 S1: res := 0;
2 S2: i := 0;
3 S3: while (i < ciudades.length) do
4 S4:       res := res + ciudades[i].habitantes
5 S5:       i = i + 1
6     endwhile
```

Como vemos en la implementacion de la especificacion el programa consta solo de un ciclo while, por lo que para demostrar su correctitud debemos usar el "Teorema del Invariantez el "Teorema de terminacion de un ciclo". Tienen que cumplir estas condiciones:

1. $P_c \Longrightarrow I$
2. $\{I \wedge B\}S\{I\}$
3. $I \wedge \neg B \Longrightarrow Q_c$
4. $\{I \wedge B \wedge v_0 = fv\}S\{fv < v_0\}$
5. $I \wedge fv \leq 0 \Longrightarrow \neg B$

Basado en la especificacion y la implementacion propongo:

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $B \equiv \{i < |ciudades|\}$
- $S_1 \equiv \{res := res + ciudades[i]_1\}$
- $S_2 \equiv \{i := i + 1\}$
- $Q_c \equiv \{res = \sum_{j=0}^{|ciudades|-1} ciudades[j]_1 \wedge res > 50000\}$
- $I \equiv \{0 \leq i \leq |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_1\}$
- $fv = |ciudades| - i$

2.1. $P_c \Longrightarrow I$

Reemplazo:

$$res = 0 \wedge i = 0 \Longrightarrow 0 \leq i \leq |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_1$$

$$0 \leq i \leq |ciudades| \equiv 0 \leq 0 \leq |ciudades| \equiv True$$

$$res = \sum_{j=0}^{i-1} ciudades[j]_1 \equiv 0 = \sum_{j=0}^{0-1} ciudades[j]_1 = 0 \equiv True$$

Por lo tanto se cumple $P_c \Longrightarrow I$

2.2. $\{I \wedge B\}S\{I\}$

Para que la tripla de Hoare sea valida, la precondicion debe implicar la "Weakest precondition"del codigo y la postcondicion. Es decir:

$$\{I \wedge B\}S\{I\} \iff \{I \wedge B\} \Longrightarrow_L wp(S, I)$$

$$wp(S, I) \equiv wp(S_1, wp(S_2, I)) \equiv wp(S_1, wp(i := i + 1, 0 \leq i \leq |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_1))$$

$$\equiv wp(S_1, def(i + 1) \wedge_L 0 \leq i + 1 \leq |ciudades| \wedge res = \sum_{j=0}^{i+1-1} ciudades[j]_1)$$

$$\equiv wp(res := res + ciudades[i]_1, 0 \leq i + 1 \leq |ciudades| \wedge res = \sum_{j=0}^{i} ciudades[j]_1)$$

$$\equiv def(res) \wedge_L def(ciudades[i]_1) \wedge 0 \leq i + 1 \leq |ciudades| \wedge res + ciudades[i]_1 = \sum_{j=0}^{i} ciudades[j]_1$$

$$\equiv 0 \leq i < |ciudades| \wedge_L 0 \leq i + 1 \leq |ciudades| \wedge res + ciudades[i]_1 = \sum_{j=0}^{i} ciudades[j]_1$$

Combino los intervalos de i y resto en ambas partes de la igualdad $ciudades[i]_1$ para transformar el rango de la sumatoria

$$\equiv 0 \leq i < |ciudades| \wedge_L res = \sum_{j=0}^{i-1} ciudades[j]_1$$

Como vemos $I \wedge B \equiv wp(S, I)$, por tanto $\{I \wedge B\} \Longrightarrow wp(S, I)$ que era lo que necesitabamos para probar que la tripla de Hoare sea valida.

2.3. $I \wedge \neg B \Longrightarrow Q_c$

Reemplazo:

$$I \wedge \neg B \equiv 0 \leq i \leq |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_1 \wedge i \geq |ciudades|$$

Si $i \geq |ciudades| \wedge i \leq |ciudades|$ entonces $i = |ciudades|$

$$i = |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_1 \Longrightarrow res = \sum_{j=0}^{|ciudades|-1} ciudades[j]_1 \wedge res > 50000 \equiv Q_c$$

Y esto se cumple ya que al por especificacion al menos una de ellas es grande, es decir, supera los 50.000 habitantes.