DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Especificación y WP

15 de Septiembre de 2024 — Algoritmos y Estructuras de Datos / ex Algo 2

Grupo uwuntu

Integrante	LU	Correo electrónico
Mendez Ayala, Lautaro Evaristo	799/23	lemayala@dc.uba.ar
Garces, Facundo	1044/23	pacusgarces@gmail.com
Monges Luces, Rafael Martin	888/23	rafaml2003@gmail.com
Gallardo, Ignacio Joaquin	409/23	nachoqmt@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. Especificación

1.1. grandesCiudades

proc grandesCiudades (in ciudades : $seq\langle Ciudad\rangle$) : $seq\langle Ciudad\rangle$
requiere $\{habitantesPositivos(ciudades) \wedge_L noHayRepetidos(ciudades)\}$
asegura $\{ciudades = [\,] \longrightarrow_L res = [\,]\}$
asegura {
$(\forall ciudad : Ciudad)\ (ciudad_{habitantes} > 50000 \wedge pertenece(ciudad, ciudades)) \Leftrightarrow pertenece(ciudad, res)$
}
asegura $\{noHayRepetidos(res)\}$

1.2. sumaDeHabitantes

proc sumaDeHabitantes (in menoresDeCiudades : $seq\langle Ciudad\rangle$, in mayoresDeCiudades : $seq\langle Ciudad\rangle$) : $seq\langle Ciudad\rangle$
requiere {
$habitantesPositivos(menoresDeCiudades) \wedge_L noHayRepetidos(menoresDeCiudades) \wedge_L$
$habitantesPositivos(mayoresDeCiudades) \wedge_L noHayRepetidos(mayoresDeCiudades) \wedge_L$
$|menoresDeCiudades| = |mayoresDeCiudades| \wedge_L$
$mismasCiudades(menoresDeCiudades, mayoresDeCiudades)$
}
asegura $\{menoresDeCiudades = [\,] \longrightarrow_L res = [\,]\}$
asegura {
$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |menoresDeCiudades| \longrightarrow_L$
$(menoresDeCiudades[i]_{nombre} = mayoresDeCiudades[j]_{nombre}) \longrightarrow_L$
$\langle menoresDeCiudades[i]_{nombre}, menoresDeCiudades[i]_{habitantes} + mayoresDeCiudades[j]_{habitantes}\rangle \in res)$
}
asegura $\{mismasCiudades(res, menoresDeCiudades)\}$
asegura $\{noHayRepetidos(res)\}$
pred mismasCiudades (ciudad1 : $seq\langle Ciudad\rangle$, ciudad2 : $seq\langle Ciudad\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |ciudad1| \longrightarrow_L (\exists j : \mathbb{Z})\ (0 \leq j < |ciudad1| \wedge_L (ciudad1[i]_{nombre} = ciudad2[j]_{nombre})))$
}

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 {
$matrizValida(distancias) \wedge_L$
$existeEnMatriz(distancias, desde) \wedge_L$
$existeEnMatriz(distancias, hasta) \wedge_L$
$noElemNeg(distancias)$
}
asegura {
$res = \text{true} \Leftrightarrow (\exists camino : seq\langle \mathbb{Z}\rangle)\ (|camino| > 0 \wedge_L$
$noHayMasElementos(camino, distancias) \wedge_L$
$noHayElemPorFuera(camino, distancias)) \wedge_L$
$(estan2a2(camino, distancias) \wedge (camino[0] = desde \wedge camino[|camino| - 1] = hasta))$
}

1.4. cantidadCaminosNSaltos

proc cantidadCaminosNSaltos (inout conexion : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, in n : $\mathbb{Z}$)
requiere $\{conexion = C_0\}$
requiere $\{matrizValida(conexion)\}$
requiere $\{esMatrizOrden1(conexion)\}$
requiere $\{n \geq 1\}$
asegura $\{n = 1 \longrightarrow conexion = C_0\}$
asegura $\{|conexion| = |C_0| \wedge_L (\forall i : \mathbb{Z})\ (0 \leq i < |C_0| \longrightarrow_L |conexion[i]| = |C_0[i]|)\}$
asegura {
$(\exists s : seq\langle seq\langle seq\langle \mathbb{Z}\rangle\rangle\rangle)\ (|s| = n \wedge s[0] = C_0 \wedge_L$
$(\forall i : \mathbb{Z})\ (1 \leq i < |s| \longrightarrow_L (esMatrizCuadrada(s[i]) \wedge_L |s[i-1]| = |s[i]| \wedge_L$

$esProducto(s[i], s[i-1], C_0) \wedge_L conexion = s[|s|-1])))$
}
pred esMatrizOrden1 (conexion : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$matrizValida(conexion) \wedge_L$
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < |conexion| \wedge 0 \leq j < |conexion| \longrightarrow_L (conexion[i][j] = 0 \vee conexion[i][j] = 1))$
}
pred esProducto (m: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, n: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, o: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |o| \longrightarrow_L m[i][j] = \sum_{r=1}^{|o|} o[i][r] * n[r][i])$
}

1.5. caminoMínimo

proc caminoMínimo (in origen:$\mathbb{Z}$, in destino:$\mathbb{Z}$, in distancias: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) : $seq\langle \mathbb{Z}\rangle$
requiere {
$matrizValida(distancias) \wedge_L$
$existeEnMatriz(distancias, origen) \wedge_L$
$existeEnMatriz(distancias, destino)$
$\wedge_L noElemNeg(distancias)$
}
asegura {
$(res = \varnothing \wedge noHayCamino(origen, destino, distancias)) \vee$
$(\forall camino : seq\langle \mathbb{Z}\rangle)(|camino| > 0 \rightarrow$
$(esCamino(camino, origen, destino, distancias) \wedge esCamino(res, origen, destino, distancias) \wedge$
$(distTotal(res, distancias) \leq distTotal(camino, distancias))))$
}
pred noHayCamino (origen: $\mathbb{Z}$, destino: $\mathbb{Z}$, distancias: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall camino : seq\langle \mathbb{Z}\rangle)(\neg esCamino(camino, origen, destino, distancias))$
}
aux distTotal (camino : $seq\langle \mathbb{Z}\rangle$, distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) : $\mathbb{Z} = \sum_{i=0}^{|camino|-2} distancias[camino[i]][camino[i+1]]$;

1.6. Predicados y Auxiliares usados en multiples ejercicios

pred habitantesPositivos (ciudades : $seq\langle Ciudad\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i < |ciudades|) \longrightarrow_L ciudades[i]_{habitantes} \geq 0)$
}
pred noHayRepetidos (ciudades : $seq\langle Ciudad\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |ciudades| \longrightarrow (\forall j : \mathbb{Z})\ (0 \leq j < |ciudades| \longrightarrow (ciudades[i]_{nombre} ! = ciudades[j]_{nombre})))$
}
pred noEsReflexiva (distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i < |distancias|) \longrightarrow_L distancias[i][i] = 0)$
}
pred esMatrizCuadrada (distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i < |distancias|) \longrightarrow_L |distancias| = |distancias[i]|)$
}
pred esSimetrica (distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((1 \leq i < |distancias|) \longrightarrow_L (\forall j : \mathbb{Z})\ ((0 \leq j < i) \longrightarrow_L distancias[i][j] = distancias[j][i]))$
}
pred matrizValida (distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$esMatrizCuadrada(distancias) \wedge_L noEsReflexiva(distancias) \wedge esSimetrica(distancias)$
}
pred existeEnMatriz (matriz : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, elemento : $\mathbb{Z}$) {
$0 \leq elemento < |matriz|$
}
pred estan2a2 (camino : $seq\langle \mathbb{Z}\rangle$, distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i < |camino| - 1) \longrightarrow_L distancias[camino[i]][camino[i+1]] \neq 0)$
}
pred noHayElemPorFuera (camino : $seq\langle \mathbb{Z}\rangle$,distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i \leq |camino|) \longrightarrow_L 0 \leq camino[i] < |distancias|)$
}
pred noHayMasElementos (camino : $seq\langle \mathbb{Z}\rangle$,distancias : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$0 \leq |camino| \leq |distancias|$

}
pred esCamino (camino : $seq\langle\mathbb{Z}\rangle$, origen : $\mathbb{Z}$, destino : $\mathbb{Z}$, distancias : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$noHayElemPorFuera(camino, distancias) \land$
$noHayMasElementos(camino, distancias) \land$
$estan2a2(camino, distancias) \land$
$(camino[0] = origen \land camino[|camino| - 1] = destino)$
}
pred noElemNeg (distancias : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i < |distancias|) \longrightarrow_L (\forall j : \mathbb{Z})\ ((0 \leq j < (|distancias|)) \longrightarrow_L distancias[i][j] > 0))$
}

2. Demostraciones de correctitud

2.1. Punto 1

Se nos pide demostrar que la implementacion es correcta. Luego, debemos demostrar, con los conceptos de precondicion mas debil y con los teoremas de Invariante y de Terminacion de Ciclo, que el ciclo finaliza, y que es correcta la implementacion.

Comencemos demostrando la correctitud parcial del ciclo. Para ello, debemos probar que la tripla de Hoare $\{P_c\}S\{Q_c\}$ es valida. Proponemos, entonces, la precondicion P_c y postcondicion Q_c de ciclo.

$P_c \equiv$
$(\exists i : \mathbb{Z})(0 \leq i < |ciudades| \land_L ciudades[i]_{habitantes} > 50000) \land$
$(\forall i : \mathbb{Z})(0 \leq i < |ciudades| \longrightarrow_L ciudades[i]_{habitantes} \geq 0) \land$
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < j < |ciudades| \longrightarrow_L ciudades[i]_{nombre} \neq ciudades[j]_{nombre}) \land$
$res = 0 \land i = 0$
$Q_c \equiv res = \sum_{i=0}^{|ciudades|-1} ciudades[i]_{habitantes}$

Ahora formalizemos la guarda del ciclo B y el invariante I.

$B \equiv (i < |ciudades|)$
$I \equiv (0 \leq i \leq |ciudades| \land res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes})$

Ahora comenzemos con la demostracion. Para ello, debemos demostrar que son validas las siguientes proposiciones:

1. $P_c \implies I$
2. $\{I \land B\}S\{I\}$
3. $(I \land \neg B) \implies Q_c$

Comenzemos demostrando la primer proposicion. Primero, debemos notar que podemos elegir ignorar los cuantificadores de P_c, ya que las expresiones que comparten P_c e I son las que definen a res e i.

$P_c \implies I$

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

Luego, es verdadera la proposicion. Sigamos con la segunda proposicion.

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

- $wp(S, I)$
$\equiv wp(res := res + ciudades[i]_{habitantes}, wp(i := i + 1, 0 \leq i \leq |ciudades| \land res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes}))$
$\equiv wp(res := res + ciudades[i]_{habitantes}, 0 \leq i + 1 \leq |ciudades| \land res \sum_{j=0}^{i} ciudades[j]_{habitantes})$
$\equiv 0 \leq i < i + 1 \leq |ciudades| \land res + ciudades[i]_{habitantes} = \sum_{j=0}^{i} ciudades[j]_{habitantes}$
$\equiv 0 \leq i < i + 1 \leq |ciudades| \land res = \sum_{j=0}^{i} ciudades[j]_{habitantes} - ciudades[i]_{habitantes}$
$\equiv 0 \leq i < i + 1 \leq |ciudades| \land res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes}$
$\equiv 0 \leq i \leq |ciudades| \land res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes}$
$\equiv I$

Esclarezcamos un par de cosas.

- La desigualdad $i < i + 1$ proviene de que todo natural es menor estricto que su sucesor $S(i)$.

- Al ser $ciudades[i]_{habitantes}$ el ultimo termino de la sumatoria $res = \sum_{j=0}^{i} ciudades[j]_{habitantes}$, restarselo es equivalente a reducir el indice por 1.

Procedamos con el ultimo punto del teorema del Invariante.

$(I \wedge \neg B) \implies Q_c$

- $(I \wedge \neg B)$

  $\equiv 0 \leq i \leq |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes} \wedge i \geq |ciudades|$

  $\equiv i = |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes}$

  Como $i = |ciudades|$, reemplazo en la sumatoria res:

  $\equiv res = \sum_{j=0}^{|ciudades|-1} ciudades[j]_{habitantes}$

  Como realmente el nombre que le demos al indice de la sumatoria es indiferente, podemos afirmar que

  $\equiv Q_c$

Con lo cual queda demostrada la correctitud parcial del ciclo. Sigamos con la demostracion del teorema de Terminacion. Para ello, proponemos la funcion variante $f_v = \mathbb{V} \longrightarrow \mathbb{Z}$, $f_v = |ciudades| - i$. Resta demostrar ambos puntos del teorema:

1. $\{I \wedge B \wedge f_v = v_0\} S \{fv, v_0\}$
2. $I \wedge f_v \leq 0 \implies \neg B$

Sigamos con la demostracion de la primer proposicion. Definamos primero a $I \wedge B \wedge f_v = v_0$.

- $I \wedge B \wedge f_v = v_0$

  $\equiv 0 \leq i \leq |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes} \wedge i < |ciudades| \wedge |ciudades| - i = v_0$

  $\equiv 0 \leq i < |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes} \wedge |ciudades| - i = v_0$

  Ahora veamos si la tripla de Hoare es valida. Para eso, verifiquemos el valor de verdad de la proposicion $I \wedge B \wedge f_v = v_0 \implies wp(S, f_v < v_0)$

- $wp(S, f_v < v_0)$

  $\equiv wp(res := res + ciudades[i]_{habitantes}, wp(i := i + 1, |ciudades| - i < v_0))$

  $\equiv wp(res := res + ciudades[i]_{habitantes}, |ciudades| - i - 1 < v_0)$

  Podemos ignorar la asignacion de res al no estar presente en $|ciudades| - i - 1 < v_0$. Por lo tanto,

  $\equiv |ciudades| - i - 1 < v_0$

  Entonces, queremos ver que

  $0 \leq i < |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes} \wedge |ciudades| - i = v_0 \implies |ciudades| - i - 1 < v_0$

  Lo cual es verdadero, pues la proposicion

  $|ciudades| - i = v_0 \implies |ciudades| - i - 1 < v_0$

  lo es.

Sigamos con el ultimo punto del teorema de Terminacion.

- $I \wedge f_v \leq 0$

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

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

  $\equiv i = |ciudades| \wedge res = \sum_{j=0}^{i-1} ciudades[j]_{habitantes}$

  Entonces, veamos si la proposicion $I \wedge f_v \leq 0 \implies \neg B$ es verdadera.

- $I \wedge f_v \leq 0 \implies \neg B$

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

  Entonces, si tomamos unicamente las comparaciones de i con $|ciudades|$

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

  Lo cual es verdadero.

Entonces, queda demostrado que el ciclo inicia y finaliza correctamente.

Ahora veamos el primer bloque de codigo restante:

```
1 |   res := 0;
2 |   i := 0;
```

A este bloque de codigo lo llamaremos S_1. Ahora, queremos probar que $P \implies wp(S_1, P_c)$. En particular:
$wp(S_1, P_c)$
$\equiv wp(res := 0; i := 0, P_c)$
$\equiv wp(res := 0, wp(i := 0, P_c))$
$\equiv wp(res := 0; P_{c0}^{i})$
$\equiv P_{c0|0}^{i|res}$
$\equiv P_c$
Que es verdadero, pues $P \implies P_c$. Luego, la implementacion es correcta.

2.2. Punto 2

Planteamos las siguientes formulas:

$$\psi = (\exists i : \mathbb{Z})(0 \leq i < |ciudades| \wedge_L ciudades[i]_{habitantes} > 50000)$$

$$\phi = (\forall i : \mathbb{Z})(0 \leq i < |ciudades| \longrightarrow_L ciudades[i]_{habitantes} \geq 0)$$

Por otro lado, viendo la especificacion del procedimiento, notamos que el parametro *ciudades* es de entrada(in). Aunque sería posible agregar las formulas ψ y ϕ tanto en la precondicion del ciclo P_c como en el invariante I, y demostrar formalmente que estas precondiciones siguen siendo válidas al terminar el programa, no es necesario porque el programa no modifica este parámetro ni ninguno de sus elementos. Por lo tanto, al finalizar el programa va a seguir existiendo un k con $0 \leq k < |ciudades|$ y que $ciudades[k]_{habitantes} > 50000$.

Además, sabemos que el número de habitantes de cada ciudad es positivo pues $ciudades[j]_{habitantes} \geq 0$ para todo j, $0 \leq j < |ciudades|$, y dado que res es la suma de todos ellos, podemos concluir que $res > 50000$.