DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Descripción del tp

Subtítulo del tp

11 de septiembre de 2024 — Materia de la carrera

Grupo 42

Integrante	LU	Correo electrónico
Krivonosoff, Thiago	310/24	thiagokribas@gmail.com
Pelli, Agustin	002/01	email2@dominio.com
Miguel, Facundo	003/01	email3@dominio.com
Montenegro, Ulises	477/24	ulinicolasmonte@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. Ejemplo de sección

1.1. Subsección: ambientes comunes de LaTeX

Lo principal: las fórmulas. Se puede poner en una linea, como $x_i = x_{i-1} + x_{i-2}$, o ponerse más grande:

$$\sum_{i=0}^{n} i \tag{1}$$

Y se pueden citar ecuaciones con `\eqref{nombreDeEq}`: (1)
Ejemplo de itemizado:

- Item 1
- Item 2
- Item 3

Ejemplo de enumerado con menor distancia entre items:

1. Item 1
2. Item 2
3. Item 3

Podemos escribir mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto.
Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo.

Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos.

La tabla 1 es un ejemplo de cómo se hace una tabla.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Tabla 1: Ejemplo de tabla

La figura 2 es un ejemplo de cómo se agrega una imagen.

Figura 1: Ejemplo de figura

```
res := 0;
i := 0;
while (i < s.size()) do
    res := res + s[i];
    i := i + 1
endwhile
```

Código 1: Ejemplo de código (usando los estilos de la cátedra, ver las macros para más detalles)

Si se pone un label al `lstlisting`, se puede referenciar: Código 1.

(a) Logo de LaTeX

(b) Logo de TeX

Figura 2: Ejemplo para poner dos figuras juntas. Y citarlas por separado a (a) y (b).

1.2. Macros de la cátedra para especificar

`proc nombre` (in paramIn : N, inout paramInout : $seq\langle Z\rangle$) : tipoRes
`requiere` $\{expresionBooleana1\}$
`asegura` $\{expresionBooleana2\}$
`aux auxiliar1` (parametros) : tipoRes $=$ $expresion$;
`pred pred1` (parametros) {+5em $expresion$ }

`aux auxiliarSuelto` (parametros) : tipoRes $=$ $expresion$;
`pred predSuelto` (parametros) {+2em $(\forall variable : tipo)\ (algo \longrightarrow_L expresion)$ }
`pred predSuelto` (parametros) {+2em $(\exists variable : tipo)\ (algo \land_L expresion)$ }

A partir de aca empieza el TP.

2. Problema 1

`proc grandesCiudades` (in ciudades : $seq\langle Ciudad\rangle$) : $seq\langle Ciudad\rangle$
`requiere` $\{noRepetidos(ciudades) \land noHabitantesNegativos(ciudades)\}$
`asegura` $\{|res| \leq |ciudades|\}$
`asegura` $\{(\forall elem : Ciudad)\ ((elem.habitantes > 50000 \land elem \in ciudades) \longrightarrow elem \in res)\}$

`pred noRepetidos` (in ciudades: $seq\langle Ciudad\rangle$) {+2em $(\forall i : Z)\ (0 \leq i < |ciudades|) \longrightarrow_L (\forall j : Z)\ (0 \leq j < |ciudades|) \longrightarrow_L (i \neq j \longrightarrow ciudades[i].nombre \neq ciudades[j].nombre)$ }
`pred noHabitantesNegativos` (in ciudades: $seq\langle Ciudad\rangle$) {+2em $(\forall i : Z)\ (0 \leq i < |ciudades|) \longrightarrow_L ciudades[i].habitantes >= 0$ }

3. Problema 2

`proc sumaDeHabitantes` (in menoresDeCiudades: $seq\langle Ciudad\rangle$, in mayoresDeCiudades: $seq\langle Ciudad\rangle$) : $seq\langle Ciudad\rangle$
`requiere` $\{noRepetidos(menoresDeCiudades) \land noRepetidos(mayoresDeCiudades)\}$
`requiere` $\{noHabitantesNegativos(menoresDeCiudades) \land noHabitantesNegativos(mayoresDeCiudades)\}$
`requiere` $\{mismosElementos(menoresDeCiudades, mayoresDeCiudades)\}$
`requiere` $\{|menoresDeCiudades| = |mayoresDeCiudades|\}$
`asegura` $\{|res| = |mayoresDeCiudades|\}$
`asegura` $\{(\forall elem : Ciudad)\ ((elem \in res) \longrightarrow$
$(\forall i : Z)\ (0 \leq i < |res|) \longrightarrow_L (\exists j : Z)\ (0 \leq j < |res|)) \land_L$
$mayoresDeCiudades[i].nombre = menoresDeCiudades[j].nombre \longrightarrow$
$res[i].nombre = mayoresDeCiudades[i].nombre \land$
$res[i].habitantes = mayoresDeCiudades[i].habitantes + menoresDeCiudades[j].habitantes\}$

pred `mismosElementos` (in s1: $seq\langle Ciudad\rangle$, in s2: $seq\langle Ciudad\rangle$) {+2em $(\forall i : Z)\ (0 \leq i < |s1|) \longrightarrow_L (\exists j : Z)\ (0 \leq j < |s1|) \wedge_L s1[i].$
}

4. Problema 3

proc `hayCamino` (in distancia: $seq\langle seq\langle Z\rangle\rangle$, in desde: Z, in hasta: Z) : Bool
requiere $\{esCuadrada(distancia)\}$
requiere $\{0 \leq desde < |distancia|\}$
requiere $\{0 \leq hasta < |distancia|\}$
requiere $\{filaIgualColumna(distancia)\}$
requiere $\{matrizTodosPositivos(distancia)\}$
asegura $\{(\exists sec : seq\langle Z\rangle)\ ((\forall i : Z)\ (0 \leq i < |sec|)) \longrightarrow_L 0 \leq sec[i] < |distancia| \wedge$
$sec[0] = desde \wedge sec[|sec| - 1] = hasta \wedge$
$todosConexionAnterior(sec, distancia)\}$

pred `todosConexionAnterior` (in sec: $seq\langle Z\rangle$, in mat: $seq\langle seq\langle Z\rangle\rangle$) {+2em $(\forall j : Z)\ (1 \leq i < |sec| \longrightarrow_L mat[sec[i]][sec[i-1]] \neq 0)$ }
pred `esCuadrada` (in mat: $seq\langle seq\langle Z\rangle\rangle$) {+2em $(\forall i : Z)\ (0 \leq i < |mat|) \longrightarrow_L |mat| = |mat[i]|$ }
pred `filaIgualColumna` (in mat: $seq\langle seq\langle Z\rangle\rangle$) {+2em $(\forall i : Z)\ (0 \leq i < |mat|) \longrightarrow_L (\forall j : Z)\ (0 \leq j < |mat|) \longrightarrow_L mat[i][j] = mat[j][i]$ }
pred `matrizTodosPositivos` (in mat: $seq\langle seq\langle Z\rangle\rangle$) {+2em $(\forall i : Z)\ (0 \leq i < |mat|) \longrightarrow_L (\forall j : Z)\ (0 \leq j < |mat|) \longrightarrow_L mat[i][j] >= 0$ }

5. Problema 4

proc `cantidadCaminosNSaltos` (inout conexion: $seq\langle seq\langle Z\rangle\rangle$, in n: Z)
requiere $\{1 \leq n\}$
requiere $\{esCuadrada(conexion)\}$
requiere $\{filaIgualColumna(conexion)\}$
requiere $\{(\forall i : Z)\ (0 \leq i < |conexion|) \longrightarrow_L (\forall j : Z)\ (0 \leq j < |conexion|) \longrightarrow_L conexion[i][j] \in [0, 1]\}$
requiere $\{conexion = C_0\}$
asegura $\{(\exists sec : seq\langle seq\langle seq\langle Z\rangle\rangle\rangle)\ (|sec| = n) \wedge sec[0] = C_0 \wedge$
$(\forall i : Z)\ (1 \leq i < |sec|) \longrightarrow_L sec[i] = multiplicarMatrices(C_0, sec[i-1]) \longrightarrow conexion = sec[|sec| - 1]\}$

aux `inversa` (in mat: $seq\langle seq\langle Z\rangle\rangle$) : $seq\langle seq\langle Z\rangle\rangle$ =
$(\forall i : Z)\ (0 \leq i < |mat|) \longrightarrow_L (\forall j : Z)\ (0 \leq j < |mat|) res[j][i] = mat[i][j]$;
aux `multiplicarMatrices` (in mat1: $seq\langle seq\langle Z\rangle\rangle$, in mat2: $seq\langle seq\langle Z\rangle\rangle$) : $seq\langle seq\langle Z\rangle\rangle$ =
$(\forall i : Z)\ (0 \leq i < |mat1|) \longrightarrow_L (\forall j : Z)\ (0 \leq j < |mat1[i]|) \longrightarrow_L$
$res[i][j] = productoEscalar(mat1[i], inversa(mat2)[j])$;
aux `productoEscalar` $(\in fila : seq\langle Z\rangle, \in col : seq\langle Z\rangle)$: Z =
$(\forall i : Z)\ (0 \leq i < |fila|) \longrightarrow_L (\forall j : Z)\ (0 \leq j < |col|) \longrightarrow_L res = \sum_{i=0}^{|fila|-1} fila[i] * fila[j]$;

6. Punto 2.1

Demostramos que la implementación es correcta con correcta con respecto a la especificación dada mediante teorema de invariante y teorema de terminación.

Por teorema del invariante primero debemos demostrar los siguientes puntos:
$\mathrm{I} = 0 \leq i \leq |\mathrm{ciudades}| \wedge \sum_{j=0}^{i-1} \mathrm{ciudades}[j].habitantes = res$

- $P_c \longrightarrow I$
- $\{I \wedge B\}\ S\ \{I\}$
- $I \wedge \neg B \longrightarrow Q_c$

Primer paso Probamos primero la implicación de la precondición del ciclo hacia el invariante:
$\mathrm{P}_c \longrightarrow I$

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

$$0 \leq 0 \leq |\mathrm{ciudades}| \wedge \sum_{j=0}^{0-1} \mathrm{ciudades}[j].habitantes = 0$$

$$True \wedge True$$

$$True$$

Segundo paso Ahora probamos que vale la siguiente tripla de Hoare:
{I $\wedge B$} S $\{I\}$
Hacemos uso del axioma.

$$wp(S, I) \cong wp(S_1, wp(S_2, I))$$

$$wp(S_2, I) \cong def(S_2) \wedge_L I_i^{i:=i+1}$$

$$wp(S_2, I) \cong True \wedge 0 \leq i+1 \leq |\text{ciudades}| \wedge_L \sum_{j=0}^{i} \text{ciudades}[j].habitantes = res$$

Terminamos de definir el wp de S2:

$$wp(S_2, I) \cong 0 \leq i+1 \leq |\text{ciudades}| \wedge_L \sum_{j=0}^{i} \text{ciudades}[j].habitantes = res$$

Ahora definimos el wp de S1:

$$wp(S_1, wp(S_2, I)) \cong wp(res = res + \text{ciudades}[i].habitantes, 0 \leq i+1 \leq |\text{ciudades}| \wedge_L \sum_{j=0}^{i} \text{ciudades}[j].habitantes = res)$$

$$wp(S_1, wp(S_2, I)) \cong def(res = res + \text{ciudades}[i].habitantes) \wedge_L I_res^{res+\text{ciudades}[i].habitantes}$$

$$wp(S_1, wp(S_2, I)) \cong True \wedge_L 0 \leq i+1 \leq |\text{ciudades}| \wedge_L \sum_{j=0}^{i} \text{ciudades}[j].habitantes = res + \text{ciudades}[i].habitantes$$

$$wp(S_1, wp(S_2, I)) \cong 0 \leq i+1 \leq |\text{ciudades}| \wedge_L \sum_{j=0}^{i} \text{ciudades}[j].habitantes = res + \text{ciudades}[i].habitantes$$

Ahora queda definido el wp de S:

$$wp(S, I) \cong 0 \leq i < |\text{ciudades}| \wedge_L \sum_{j=0}^{i-1} \text{ciudades}[j].habitantes = res$$

Ahora veo la implicacion del invariante y la guarda hacia wp(S,I):

$$\{I \wedge B\} \longrightarrow wp(S, I) \cong (sigueabajo)$$

$$0 \leq i < |\text{ciudades}| \wedge 0 \leq i \leq |\text{ciudades}| \wedge_L \sum_{j=0}^{i-1} \text{ciudades}[j].habitantes = res \longrightarrow (sigueabajo)$$

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

Se cancelan los terminos y queda:

$$0 \leq i \leq |\text{ciudades}| \longrightarrow True$$

$$True$$

Tercer paso Ahora probamos que vale la siguiente implicación: $I \wedge \neg B \longrightarrow Q_c$

$$0 \leq i \leq |\text{ciudades}| \wedge_L \sum_{j=0}^{i-1} \text{ciudades}[j].habitantes = res \wedge i \geq |\text{ciudades}| (sigueabajo)$$

$$\longrightarrow \sum_{j=0}^{|\text{ciudades}|-1} \text{ciudades}[j].habitantes = res \wedge i = |\text{ciudades}|$$

Se cancelan las sumatorias y la igualdad

$$0 \leq i \leq |\text{ciudades}| \longrightarrow True$$

$$True$$

Ahora por teorema de terminación ahora debemos demostrar que la ejecucion del ciclo siempre termina, nuestra función variante:

$\mathbf{F}_v = |\mathbf{ciudades}| - i - 1$

- $\{I \wedge B \wedge F_v = v_0\}S\{F_v < v_0\}$
- $I \wedge F_v \leq 0 \longrightarrow \neg B$

Primer paso Probamos la primera implicación:

$\{$ **I** $\wedge B \wedge F_v = v_0\}S\{F_v < v_0\}$

$$\{0 \leq i \leq |\mathbf{ciudades}| \wedge_L \sum_{j=0}^{i-1} \mathbf{ciudades}[j].habitantes = res \wedge 0 \leq i \leq |\mathbf{ciudades}| \wedge F_v = |\mathbf{ciudades}| - i - 1\} \longrightarrow (sigueabajo)$$

$$wp(S, F_v < v_0) =$$

$$wp(S1, wp(S2, F_v < v_0)) =$$

$$wp(S1, wp(i := i + 1, F_v < v_0)) =$$

$$wp(S1, True \wedge F_v = |\mathbf{ciudades}| - i - 2 < v_0) =$$

$$wp(res := res + \mathbf{ciudades}[i].habitantes, True \wedge F_v = |\mathbf{ciudades}| - i - 2 < v_0) =$$

$$True \wedge F_v = |\mathbf{ciudades}| - i - 1 < v_0 =$$

$$F_v = |\mathbf{ciudades}| - i - 1 < v_0$$

Finalmente la implicación nos queda de la forma

$$\{0 \leq i \leq |\mathbf{ciudades}| \wedge_L \sum_{j=0}^{i-1} \mathbf{ciudades}[j].habitantes = res \wedge 0 \leq i \leq |\mathbf{ciudades}| \wedge F_v = |\mathbf{ciudades}| - i - 1\} \longrightarrow (sigueabajo)$$

$$|\mathbf{ciudades}| - i - 1 < |\mathbf{ciudades}| - i$$

Se cancelan los terminos y queda

$$\{0 \leq i \leq |\mathbf{ciudades}| \wedge_L \sum_{j=0}^{i-1} \mathbf{ciudades}[j].habitantes = res \wedge_L F_v = |\mathbf{ciudades}| - i - 1\} \longrightarrow True$$

$$True$$

Segundo paso Probamos la segunda implicación:

I $\wedge F_v \leq 0 \longrightarrow \neg B$

$$0 \leq i \leq |\mathbf{ciudades}| \wedge_L res = \sum_{j=0}^{i-1} \mathbf{ciudades}[j].habitantes \wedge_L |\mathbf{ciudades}| - i - 1 \leq 0 \longrightarrow i \geq |\mathbf{ciudades}|$$

$$0 \leq i \leq |\mathbf{ciudades}| \wedge |\mathbf{ciudades}| \leq i + 1 \longrightarrow i \geq |\mathbf{ciudades}|$$

$$i = |\mathbf{ciudades}| \longrightarrow i \geq |\mathbf{ciudades}|$$

$$True$$

7. Ejercicio 2.2

Teniendo en cuenta la correctitud del programa demostrada en el 2.1, sabemos que el valor devuelto por el programa, siempre que se cumpla la precondición, tendrá la pinta de:

$$\textbf{res} = \textbf{sumatoria}$$

En el procedimiento, se especifica que el parámetro de entrada es de tipo IN, por lo tanto en la precondición y la postcondición, el parámetro *ciudades* mantendrá las mismas características, entre ellas que sus elementos son todos mayores o iguales a 0. Por lo tanto, al tomar un elemento de *ciudades*, obligatoriamente será menor o igual a la sumatoria del total de elementos de *ciudades*, es decir:

$$\boldsymbol{ciudades}[i].habitantes \leq \textbf{sumatoriacon algún i}0 \leq i < |\boldsymbol{ciudades}|$$

Ahora bien, según la postcondición sabemos que res = la sumatoria de todos sus elementos, entre los cuales sabemos que al menos alguno de ellos es mayor a 50.000, por lo tanto:

$$50{,}000 < \boldsymbol{ciudades}[k].habitantes \quad \textbf{con algún k} \quad 0 \leq k < |\boldsymbol{ciudades}|$$

Podemos afirmar entonces que:

$$50{,}000 < \boldsymbol{ciudades}[k].habitantes \leq \textbf{sumatoria} \quad \textbf{con algún k} \quad 0 \leq k < |\boldsymbol{ciudades}|$$

Y por transitividad concluimos que:

$$50{,}000 < \textbf{sumatoria}$$