CI2611 - Algoritmos y estructuras I

Parcial 2

Daniel Delgado

Abr-Jul 2025

Contenido

1 Resumen Parcial 2

1.1 Técnicas de derivación de Invariantes

1. Eliminar un predicado de una conjunción.

   A partir de una postcondición de la forma: $P \wedge Q$

   Es posible elegir:

   Inv: P

   Guarda: $\neg Q$

   El programa resultante.

   $\{Inv : P\}$

   $do \neg Q \rightarrow$

   S

   od

   $\{P \wedge Q\}$

2. Reemplazo de constantes por variables.

   A partir de una postcondición de la forma: $r = F(N)$

   Es posible elegir:

   Inv: $r = F(x) \wedge 0 \leq x \leq N$

   El programa resultante.

   $\{Inv : r = F(x) \wedge 0 \leq x \leq N\}$

   S

   $\{r = F(N)\}$

3. Fortalecimiento de invariantes.

   A partir de una postcondición de la forma: $r = F(N)$

   Es posible elegir:

   Inv: $r = F(x)$

   Cuando la función $F(x)$ es difícil de calcular en una sola iteración se debe determinar una función $G(x)$ tal que cumpla.

$r = F(x) \otimes G(x)$, donde $\otimes$ representa la operación que une a las expresiones F y G.

Y podemos fortalecer el invariante con un nuevo predicado.

$s = G(x)$

De manera que r se vuelve recursiva.

$r = r \otimes s$

Finalmente el invariante queda.

$r = F(x) \wedge 0 \leq x < N \wedge s = G(x)$

En este punto si $G(x)$ sigue siendo difícil de computar en una sola iteración, entonces se debe aplicar el mismo procedimiento de forma recurrente hasta alcanzar una expresión fácil de calcular en una iteración.

1.2 Arreglos

Notación declaración de arreglos:

$a : array\ [p..q)\ of\ t$, donde t es el tipo de dato de cada elemento

$a : array\ [p..q)\ of\ array\ [r..s)\ of\ t$, Matríz

$a : array\ [p..q) \times [r..s)\ of\ t$, Matríz

Regla de asignación.

$\{P\}\ a[E] := F\ \{Q\}$, con $a : array[p..q)\ of\ t$ se traduce matemáticamente en:

$$a[i : E][j] = \begin{cases} a[j] & \text{si } i \neq j \\ E & \text{si } i = j \end{cases} \tag{1}$$

Regla de correctitud para la asignación.

$\{P\}\ a[E] := F\ \{Q\} \iff [P \Rightarrow p \leq E < q \wedge Q(a := a(E : F))]$

Regla de correctitud para el intercambio de elementos en un arreglo.

$\{P\}\ Intercambio(a, E, F)\ \{Q\} \iff [P \Rightarrow p \leq E < q \wedge p \leq F < q \wedge Q(a := a(E, F : a[F], a[E]))]$

1.3 Procedimientos

Notación:

proc $< nombre\_procedimiento > (< parametro1 >; < parametro2 >; \ldots; < parametroN >)$

$\{precondicion : P\}$

$\{postcondicion : Q\}$

S

Donde, $< parametro >=< tipo\ parametro > < nombre\_variable : tipo >$

y $< tipo\ parametro >$ puede ser: in (entrada), in-out (entrada-salida), out (salida).

Definición procedimiento general.

proc $p\ (in\ x;\ in-out\ y;\ out\ z)$

$\{P_{def}\}$

$\{Q_{def}\}$

S

Correctitud de la llamada a procedimientos.

Nota: En este resumen solo se plasmará el caso general que en la literatura es conocido como el caso B2.

Con el procedimiento p definido anteriormente se desea probar la siguiente tripleta.

$$\{P_{llam}\}\ p(E, a, b)\ \{Q_{llam}\}$$

Para lo cual se deben cumplir los siguientes predicados.

1.$[P_{llam} \Rightarrow P_{def}(x, y := E, a)]$

2.$[P_{llam}(a, b := A, B) \wedge Q_{def}(x, y_0, y, z := E, A, a, b) \Rightarrow Q_{llam}]$

Donde A, B son los valores iniciales de los argumentos de entrada-salida y salida respectivamente y son constantes cualquieras dentro de la demostración.

Dado que un procedimiento puede tener múltiples parámetros de entrada, entrada-salida y salida, tanto los parámetros x, y, z como los argumentos E, a, b represnetan una lista y no una única variable, bajo la misma justificación las constantes A, B represnetan una lista de constantes de los estados de los argumentos de entradas-salida y salida, respectivamente.

1.4 Funciones

Notación:

func $f(< param1 >; < param2 >; \ldots; < paramN >) \rightarrow tipoRetorno$

$\{precondicion : P\}$

$\{postcondicion : Q\}$

[

$I;$

$>> E$

]

Donde I representa el conjunto de instrucciones que ejecuta la función y E representa el valor retornado.

Correctitud de funciones.

Para probar una función es necesario traducir la función a un procedimiento.

proc $pf(< param1 >; < param2 >; \ldots; < paramN >; out\ \theta : tipoRetorno)$

$\{precondicion : P\}$

$\{postcondicion : Q\}$

[

$I;$

$\theta := E$

]

Donde, θ es el parámetro que representa a una variable en el cuerpo del programa principal que pasará como argumento durante la llamada del nuevo procedimiento y recibirá el valor de retorno de la función definida originalmente.

Algunos ejemplos.

Función	**Procedimiento**
$x := 3 * (f(a) + b)$	$pf(a, \phi);\ x := 3 * (\phi + b)$
	$pf(a, \phi); pf(b, \omega)$
$if\ f(a) \geq 0 \rightarrow b$	$if\ \phi \geq 0 \rightarrow b$
$[\,]f(b) \leq 0 \rightarrow c$	$[\,]\omega \leq 0 \rightarrow c$
	$pf(i, \theta);$
$do\ n - f(i) > 0 \rightarrow \ldots$	$do\ n - \theta > 0 \rightarrow \ldots$
$\ldots$	$\ldots; pf(i, \theta)$
od	od

— PROBLEMAS RESUELTOS —

2 Derivación de Invariantes

2.1 Ejercicios sección 4.3 Kaldewaij

Derivar programa.

2.1.1 Ejercicio 0

[

$const N : int$

$const A : array\ [0..N)\ of\ int$

$var\ r : int$

$\{N \geq 1\}$

S

$\{r = (\max\ p, q |\ 0 \leq p < q < N : A[p] - A[q])\}$

]

Solución

1. Se define el siguiente invariante:

$$\text{P0: } r = (\max\ p, q |\ 0 \leq p < q < x : A[p] - A[q])$$

$$\text{P1: } 0 \leq x \leq N$$

$$x = 0 \Rightarrow (\max\ p, q |\ 0 \leq p < q < 0 : A[p] - A[q]) = -\infty$$

2. Guarda $x < N$

3. Se verifica la actualización de r en $x + 1$

$$(\max\ p, q |\ 0 \leq p < q < x + 1 : A[p] - A[q])$$

$$\equiv \quad \langle Ultimo\ termino :\ q = x \rangle$$

$$(\max\ p, q |\ 0 \leq p < q < x : A[p] - A[q])\ \max\ (\max\ p |\ 0 \leq p < x : A[p] - A[x])$$

$$\equiv \quad \langle P0 \rangle$$

$$r\ \max\ (\max\ p |\ 0 \leq p < x : A[p] - A[x])$$

$$\equiv \quad \langle\ \text{Distributividad - sobre max}\ \rangle$$

$$r\ \max\ ((\max\ p |\ 0 \leq p < x : A[p]) - A[x])$$

4. Introducimos una nueva variable

$$s = (\max\ p |\ 0 \leq p < x : A[p])$$

De esta manera se fortalece el invariante con el predicado.

$$Q : s = (\max\ p |\ 0 \leq p < x : A[p])$$

$$x = 0 \Rightarrow s = (\max\ p |\ 0 \leq p < 0 : A[p]) = -\infty$$

La actualización de r queda como sigue.

$$r = r\ \max\ (s - A[x])$$

5. Ahora evaluamos la actualizacion de s en $x + 1$

$$(\max\ p |\ 0 \leq p < x + 1 : A[p])$$

$$\equiv \quad \langle Ultimo\ termino \rangle$$

$$(\max\ p |\ 0 \leq p < x : A[p])\ \max\ A[x]$$

$\equiv \quad \langle Q \rangle$

$s \ \max \ A[x]$

La actualización de s queda:

$$s = s \ \max \ A[x]$$

Programa:

[

$const\ N : int;$

$const\ A : array\ [0..N)\ of\ int;$

$var\ r, s, x : int;$

$\{N \geq 1\}$

$x, r, s := 0, -\infty, -\infty$

$\{Inv : r = (\max\ p, q |\ 0 \leq p < q < x : A[p] - A[q]) \wedge 0 \leq x \leq N \wedge s = (\max\ p |\ 0 \leq p < x : A[p])\}$

$\{Cota : N - x\}$

do $x < N \rightarrow$

$r = r \ \max\ (s - A[x]);$

$s = s \ \max \ A[x];$

$x = x + 1$

od

$\{r = (\max\ p, q |\ 0 \leq p < q < N : A[p] - A[q])\}$

]

2.1.2 Ejercicio 4

[

$const N : int;$

$const A : array\ [0..N)\ of\ bool;$

$var\ r : bool$

$\{N \geq 0\}$

S

$\{r \equiv (\exists\ p |\ 0 \leq p \leq N : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < N : \neg A[i]))\}$

]

Solución

1. Se define el siguiente invariante:

   $P0 : r \equiv (\exists\, p |\, 0 \leq p \leq x : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < x : \neg A[i]))$

   $P1 : 0 \leq x \leq N$

   $x = 0 \Rightarrow r \equiv$

   $(\exists\, p |\, 0 \leq p \leq 0 : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < 0 : \neg A[i]))$

   $\equiv$ $\langle 0 \leq p \leq 0 \equiv p = 0 \rangle$

   $(\exists\, p |\, p = 0 : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < 0 : \neg A[i]))$

   $\equiv$ $\langle Logica \rangle$

   $(\forall i \mid 0 \leq i < 0 : A[i]) \wedge (\forall i \mid 0 \leq i < 0 : \neg A[i])$

   $\equiv$ $\langle$ Rango vacío$\rangle$

   $true$

2. Guarda $x \neq N$

3. Se verifica la actualización de r en x+1.

   $(\exists\, p |\, 0 \leq p \leq x + 1 : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < x + 1 : \neg A[i]))$

   $\equiv$ $\langle$ Ultimo termino en $\forall\, i = x \rangle$

   $(\exists\, p |\, 0 \leq p \leq x + 1 : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < x : \neg A[i]) \wedge \neg A[x])$

   $\equiv$ $\langle$ Dado $\neg ocurrelibre('p', \neg A[x]) \equiv true \rangle$

   $(\exists\, p |\, 0 \leq p \leq x + 1 : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < x : \neg A[i])) \wedge \neg A[x]$

   $\equiv$ $\langle$ Ultimo termino en $\exists\, p = x + 1 \rangle$

   $((\exists\, p |\, 0 \leq p \leq x : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < x : \neg A[i])) \vee (\forall i \mid 0 \leq i < x + 1 : A[i]) \wedge (\forall i \mid x + 1 \leq i < x : \neg A[i])) \wedge \neg A[x]$

   $\equiv$ $\langle x + 1 \leq i < x \equiv false \rangle$

   $((\exists\, p |\, 0 \leq p \leq x : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < x : \neg A[i])) \vee (\forall i \mid 0 \leq i < x + 1 : A[i]) \wedge (\forall i \mid false : \neg A[i])) \wedge \neg A[x]$

   $\equiv$ $\langle$ Rango vacío $\rangle$

   $((\exists\, p |\, 0 \leq p \leq x : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < x : \neg A[i])) \vee (\forall i \mid 0 \leq i < x + 1 : A[i]) \wedge true) \wedge \neg A[x]$

$\equiv$ $\langle P0 \rangle$

$(r \vee (\forall i \mid 0 \leq i < x + 1 : A[i])) \wedge \neg A[x]$

$\equiv$ $\langle$ Ultimo termino en $\forall\ i = x \rangle$

$(r \vee ((\forall i \mid 0 \leq i < x : A[i]) \wedge A[x])) \wedge \neg A[x]$

4. Introducimos una nueva variable.

   $s \equiv (\forall i \mid 0 \leq i < x : A[i])$

   $x = 0 \Rightarrow s \equiv (\forall i \mid 0 \leq i < 0 : A[i]) = true$

   Fortalecemos el invariante con el predicado.

   $$Q : s \equiv (\forall i \mid 0 \leq i < x : A[i])$$

   De esta manera, la actualización de r queda como sigue.

   $$r \equiv (r \vee (s \wedge A[x])) \wedge \neg A[x]$$

5. Verificacmos la actualización de s en x+1

   $(\forall i \mid 0 \leq i < x + 1 : A[i])$

   $\equiv$ $\langle$ Ultimo termino $i = x \rangle$

   $(\forall i \mid 0 \leq i < x : A[i]) \wedge A[x]$

   $\equiv$ $\langle Q \rangle$

   $s \wedge A[x]$

   La actualización de s queda como sigue.

   $$s \equiv s \wedge A[x]$$

Programa:

[

$const\ N : int;$

$const\ A : array\ [0..N)\ of\ bool;$

$var\ r, s, x : bool$

$\{N \geq 0\}$

$x, r, s := 0, true, true$

$\{Inv : r \equiv (\exists\ p|\ 0 \leq p \leq x : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < x : \neg A[i])) \wedge s \equiv (\forall i \mid 0 \leq i < x : A[i]) \wedge 0 \leq x \leq N\}\{Cota : N - x\}$

do $x \neq N \rightarrow$

$r \equiv (r \vee (s \wedge A[x])) \wedge \neg A[x];$

$s \equiv s \wedge A[x];$

$x = x + 1$

od

$\{r \equiv (\exists\ p|\ 0 \leq p \leq N : (\forall i \mid 0 \leq i < p : A[i]) \wedge (\forall i \mid p \leq i < N : \neg A[i]))\}$

]

2.2 Parcial 2 Ene-Mar 2025

2.2.1 Ejercicio 4

Sea las funciones F y G.

$$F(n) = \begin{cases} -10 & \text{si } n = 0 \\ 5 - F(n-1) + G(n-1) & \text{si } n > 0 \end{cases} \tag{2}$$

$$G(n) = \begin{cases} 0 & \text{si } n = 0 \\ G(n-1) + 7 & \text{si } n > 0 \end{cases} \tag{3}$$

[

$const\ N : int;$

$var\ r : int$

$\{N \geq 0\}$

S

$\{r = 2F(N)\}$

]

Solución

1. Se define el siguiente invariante.

$$P0 : r = 2F(n)$$

$$P1 : 0 \leq n \leq N$$

$$n = 0 \Rightarrow r = 2F(0) = -10$$

2. Guarda: $x < N$

3. Actualización de r en $n + 1$

$2F(n + 1)$

$=$ $\langle$ Definición de F $\rangle$

$2(5 - F(n + 1 - 1) + G(n + 1 - 1))$

$=$ $\langle$ Aritmética $\rangle$

$10 - 2F(n) + 2G(n)$

4. Se introduce una nueva variable.

$s = G(n)$

$n = 0 \Rightarrow s = G(0) = 0$

La actualización de r queda.

$$r = 10 - 2r + 2s$$

Se fortalece el invariante con el predicado.

$$Q : s = G(n)$$

5. Se verifica la actualización de s en $n + 1$.

$G(n + 1)$

$=$ $\langle$ Definición de G $\rangle$

$G(n + 1 - 1) + 7$

$=$ $\langle$ Aritmética $\rangle$

$G(n) + 7$

$=$ $\langle Q \rangle$

$s + 7$

Actualización de s.

$$s = s + 7$$

Programa:

[

$const\ N : int;$

$var\ r, s, n : int$

$\{N \geq 0\}$

$r, s, n := -10, 0, 0;$

$\{inv : r = 2F(n) \wedge s = G(n) \wedge 0 \leq n \leq N\}\{cota : N - n\}$

$do\ x < N \rightarrow$

$r = 10 - 2r + 2s;$

$s = s + 7;$

$n = n + 1$

od

$\{r = 2F(N)\}$

]

2.3 Tarea 6 2015

Derive programa.

2.3.1 Ejercicio 1-b

[

$const N : int;$

$var\ s : int$

$\{N > 0\}$

$Programa$

$\{s = (\sum i | 0 \leq i < N : 2^i)\}$

]

Solución

1. Se propone las siguientes proposiciones como invariantes.

$$P0 : s = (\sum i | 0 \leq i < x : 2^i)$$

$$P1 : 0 \leq x \leq N$$

Si $x = 0 \Rightarrow s = (\sum i | 0 \leq i < 0 : 2^i) = 0$

2. Guarda $x < N$.

3. Se verifica la actualización de s en $x + 1$.

$(\sum i | 0 \leq i < x + 1 : 2^i)$

$\equiv$ $\langle$ Último término $i = x \rangle$

$(\sum i | 0 \leq i < x : 2^i) + 2^x$

$\equiv$ $\langle P0 \rangle$

$s + 2^x$

4. Dado que el calculo de 2^x no es posible hacerlo en una solo iteración se propone la siguiente expresión.

$r = 2^x$

Si $x = 0 \Rightarrow r = 1$

De esta manera se puede fortalecer el invariante con el predicado a continuación.

$$Q : r = 2^x$$

Luego, la actualización de s queda como sigue.

$$s = s + r$$

5. Verificamos la actualización de r en x+1.

   2^{x+1}

   $\equiv$ $\langle$ Definición de exponente de la suma $\rangle$

   $2^x \cdot 2$

   $\equiv$ $\langle Q \rangle$

   $r \cdot 2$

   De esta manera, la actualización de r es la siguiente.

$$r = 2r$$

Programa final.

[

$const N : int;$

$var\ s, r, x : int$

$\{N > 0\}$

$x, s, r := 0, 0, 1;$

$\{inv : s = (\sum i | 0 \leq i < x : 2^i) \wedge r = 2^x \wedge 0 \leq x \leq N\}\{cota : N - x\}$

$do\ x < N \rightarrow$

$s, r = s + r, 2r$

od

$\{s = (\sum i | 0 \leq i < N : 2^i)\}$

]

2.3.2 Ejercicio 1-c

[

$const\ N : int;$

$const\ X : float;$

$var\ s : float$

$\{N > 0\}$

$Sumatoria$

$\{s = (\sum i | 0 < i < N : \frac{X^i}{i!})\}$ Cambié el enunciado para no incluir el 0

]

Solución

1. Se propone el siguiente invariante.

$$P0 : s = (\sum i | 0 < i < n : \frac{X^i}{i!})$$

$$P1 : 0 < n \leq N$$

$n = 1 \Rightarrow s = (\sum i | 0 < i < 1 : \frac{X^i}{i!}) = 0$

2. Guarda $n < N$.

3. Dado que esta expresión del invariante no es fácil de calcular en una iteración se verifica la actualización de s en $n + 1$ para verificar si podemos hallar una función G(n) que sea más fácil de calcular.

$(\sum i | 0 \leq i < n + 1 : \frac{X^i}{i!})$

$\equiv$ $\langle$ Último término $i = n \rangle$

$(\sum i | 0 \leq i < n : \frac{X^i}{i!}) + \frac{X^n}{n!}$

$\equiv$ $\langle P0 \rangle$

$s + \frac{X^n}{n!}$

4. La expresión anterior es una expresión que no resulta fácil de calcular en una iteración dado que tiene un factorial y una potencia del iterador. Se introduce la siguiente variable.

$r = \frac{X^n}{n!}$

$n = 1 \Rightarrow r = \frac{X^1}{1!} = X$

Se fortalece el invariante con la siguiente proposición.

$$Q : r = \frac{X^n}{n!}$$

Con lo cual la actualización de s queda como sigue.

$$s = s + r$$

5. Evaluamos la actualización de r en $n + 1$.

   $\frac{X^{n+1}}{(n+1)!}$

   $\equiv$ $\langle$ Aritmética $\rangle$

   $\frac{X \cdot X^n}{(n+1) \cdot n!}$

   $\equiv$ $\langle$ Aritmética $\rangle$

   $\frac{X}{(n+1)} \cdot \frac{X^n}{n!}$

   $\equiv$ $\langle Q \rangle$

   $\frac{X}{(n+1)} \cdot r$

   Se tiene que la expresión anterior resulta fácil de calcular en una iteración por lo tanto no tenemos que continuar fortaleciendo el invariante.

   La actualización de r queda como se muestra.

$$r = \frac{X}{(n+1)} \cdot r$$

Programa derivado.

[

$const\ N : int;$

$const\ X : float;$

$var\ s, r : float$

$var\ n : int$

$\{N > 0\}$

$\{inv : s = (\sum i | 0 \leq i < n : \frac{X^i}{i!}) \wedge r = \frac{X^n}{n!} \cdot r \wedge 0 < n \leq N\}\{cota : N - n\}$

$do\ n < N \rightarrow$

$s, r = s + r, \frac{X}{(n+1)} \cdot r$

od

$\{s = (\sum i | 0 < i < N : \frac{X^i}{i!})\}$

]

2.3.3 Ejercicio 1-f

[

const

$N : int;$

$S : array[0..N)\ of\ int;$

var $r : int$

$\{N \geq 0\}$

Programa

$\{r = (\#i, j | 0 \leq i < j < N : S[i] \leq 0 \wedge S[j] \geq 0)\}$

]

Solución

1. Se propone el siguiente invariante.

$$P0 : r = (\#i, j | 0 \leq i < j < x : S[i] \leq 0 \wedge S[j] \geq 0)$$

$$P1 : 0 \leq x \leq N$$

$x = 0 \Rightarrow r = (\#i, j | 0 \leq i < j < 0 : S[i] \leq 0 \wedge S[j] \geq 0) = 0$

2. Guarda $x < N$ y cota $N - x > 0$

3. Veamos la actualización de r en x+1

$(\#i, j | 0 \leq i < j < x + 1 : S[i] \leq 0 \wedge S[j] \geq 0)$

$\equiv$ $\langle$ Último término $j = x \rangle$

$(\#i, j | 0 \leq i < j < x : S[i] \leq 0 \wedge S[j] \geq 0) + (\#i | 0 \leq i < x : S[i] \leq 0 \wedge S[x] \geq 0)$

$\equiv$ $\langle$ P0 $\rangle$

$r + (\#i | 0 \leq i < x : S[i] \leq 0 \wedge S[x] \geq 0)$

Dado que en el cuerpo del cuantificador se tiene la presencia de la variable x en una expresión lógica, se debe analizar por casos.

$\equiv$ $\langle$ Casos $\rangle$

$$\begin{cases} r & \text{si } S[x] < 0 \\ r + (\#i | 0 \leq i < x : S[i] \leq 0) & \text{si } s[x] \geq 0 \end{cases} \tag{4}$$

4. Se agrega una nueva variable s.

   $s = (\#i|0 \leq i < x : S[i] \leq 0)$

   $x = 0 \Rightarrow s = (\#i|0 \leq i < 0 : S[i] \leq 0) = 0$

   Por lo tanto, se fortalece el invariante con el predicado siguiente.

$$Q : s = (\#i|0 \leq i < x : S[i] \leq 0)$$

   De esta manera, la actualización de r queda.

$$\begin{cases} r & \text{si } S[x] < 0 \\ r + s & \text{si } s[x] \geq 0 \end{cases} \tag{5}$$

5. Se verifica la actualización de s en x+1.

   $(\#i|0 \leq i < x + 1 : S[i] \leq 0)$

   $\equiv$ $\langle$ Último término con $i = x\rangle$

   $(\#i|0 \leq i < x : S[i] \leq 0) + (\#i|i = x : S[i] \leq 0)$

   $\equiv$ $\langle$ Por casos $\rangle$

$$\begin{cases} (\#i|0 \leq i < x : S[i] \leq 0) & \text{si } S[x] > 0 \\ (\#i|0 \leq i < x : S[i] \leq 0) + 1 & \text{si } S[x] \leq 0 \end{cases} \tag{6}$$

   $\equiv$ $\langle$ Q $\rangle$

$$\begin{cases} s & \text{si } S[x] > 0 \\ s + 1 & \text{si } s[x] \leq 0 \end{cases} \tag{7}$$

   Finalmente la actualización de s está dada por.

$$s = \begin{cases} s & \text{si } S[x] > 0 \\ s + 1 & \text{si } s[x] \leq 0 \end{cases} \tag{8}$$

Programa final.

[

const

$N : int;$

$S : array[0..N)\ of\ int;$

var $r, s, x : int$

$\{N \geq 0\}$

$r, s, x := 0, 0, 0;$

$\{inv : r = (\#i, j | 0 \leq i < j < x : S[i] \leq 0 \wedge S[j] \geq 0) \wedge s = (\#i | 0 \leq i < x : S[i] \leq 0) \wedge 0 \leq x \leq N\}\{cota : N - x\}$

do $x < N \rightarrow$

if $S[x] < 0 \rightarrow$

$r = r;$ // Es posible eliminar

$s = s + 1;$

[ ] $S[x] > 0 \rightarrow$

$r = r + s;$

$s = s;$ // Es posible eliminar

[ ] $S[x] = 0 \rightarrow$

$r = r + s;$

$s = s + 1;$

fi

od

$\{r = (\#i, j | 0 \leq i < j < N : S[i] \leq 0 \wedge S[j] \geq 0)\}$

]

3 Matríces

3.1 Ejercicios Prof. Chang

3.1.1 Ejercicio 5: Iteración en matrices. Recorridos

Dada una matriz A de números enteros, de dimensiones [0..M) x [0..N), se desea numerar cada casilla comenzando en A[M - 1][0] y comenzando con el número 0 (A[M - 1][0] := 0), siguiendo estrictamente el recorrido que se muestra a continuación (tomando como ejemplo una matriz 5 x 6).

$$A = \begin{bmatrix} 4 & 5 & 14 & 15 & 24 & 25 \\ 3 & 6 & 13 & 16 & 23 & 26 \\ 2 & 7 & 12 & 17 & 22 & 27 \\ 1 & 8 & 11 & 18 & 21 & 28 \\ 0 & 9 & 10 & 19 & 20 & 29 \end{bmatrix}$$

Solución

```
[
  const N, M : int;
  var A : array [0..M)x[0..N) of int;
  var i, j, enum : int;
  i, j, enum := M, 0, 0
  do j < N →
    i := i + (−1) ∗ ∗(j + 1)
    do i ! = −1 ∧ i ! = M →
      A[i][j] := enum;
      enum := enum + 1;
      i := i + (−1) ∗ ∗(j + 1)
    od
    j := j + 1
  od
]
```

En caso de no poder usar la potenciación.

```
[
  const N, M : int;
  var A : array [0..M)x[0..N) of int;
  var i, j, enum : int;
  j, enum := 0, 0
  do j < N →
    if j  mod 2 = 0 →
      i = M − 1
      do i >= 0 →
        A[i][j] := enum;
        enum := enum + 1;
        i := i − 1
      od
    [ ] j  mod 2 ! = 0 →
      i = 0
      do i < M →
        A[i][j] := enum;
        enum := enum + 1;
        i := i + 1
      od
    if;
    j = j + 1;
  od
]
```

3.1.2 Ejercicio 6: Recorrido en Arreglos y Matríces, Procedimientos y Funciones.

Parte a. Procedimiento:

Programe correctamente el procedimiento:

$move\_in\_The\_Matrix(in-out\ i,\ j:\ int\ ,\ in\ N,\ M:\ int,\ in\ TM:\ array\ [0..N)x[0..M)\ of\ int,\ in\ p:\ bool)$

tal que, posicionados en la fila i y en la columna j ($con\ N, M > 0; 0 \leq i < Ny0 \leq j < M$) de la matriz TM, se determine una nueva posición en la matriz, de la siguiente manera:

Si p es true, cuando $TM\ [i-1][j] \leq TM[i][j]$ nos moveremos una fila hacia arriba en la matriz. En caso que $TM[i-1][j] > TM[i][j]$, nos moveremos una fila hacia abajo.

Si p es false, cuando $TM[i][j-1] \leq TM[i][j]$ nos moveremos una columna hacia la izquierda en la matriz. En caso que $TM[i][j-1] > TM[i][j]$ nos moveremos una columna a la derecha.

La matriz es circular, esto es, si i = 0, se considera que $i-1 = N-1$ y si $i = N-1$, entonces $i+1 = 0$. De forma similar, si j = 0, se considera que $j-1 = M-1$ y si $j = M-1$, entonces $j+1 = 0$. De esta manera, no es posible desplazarse a posiciones inexistentes de la matriz. Claramente, actualizar correctamente los valores de i y j es parte de lo que se debe programar en el procedimiento. En resumen, el procedimiento actulizará el valor de i o de j, dependiendo del valor de p y de ciertos valores de T M.

3.2 Ejercicios Tarea 3 Sep-Dic 2013

3.2.1 Ejercicio 15

Mostrar la correctitud.

$\{A[i] = X \wedge A[j] = Y\}A[i] := A[i] + A[j]\{A[i] = X + Y\}$

Solución

Se debe probar la asignación en arreglos.

$[P \Rightarrow p \leq i < q \wedge Q(A := A(i : (A[i] + A[j])))]$

Sustituyendo P y Q.

$A[i] = X \wedge A[j] = Y \Rightarrow p \leq i < q \wedge p \leq j < q \wedge (A[i] = X + Y)(A := A(i : (A[i] + A[j])))$

Por suposición del antecedente empezando por el consecuente.

$p \leq i < q \wedge p \leq j < q \wedge (A[i] = X + Y)(A := A(i : (A[i] + A[j])))$

$\equiv$ $\langle$ Sustitución textual $\rangle$

$p \leq i < q \wedge p \leq j < q \wedge A(i : (A[i] + A[j]))[i] = X + Y$

$\equiv$ $\langle$ Asignación de Arreglo $i = j \Rightarrow a[i : E][j] = E\rangle$

$p \leq i < q \wedge p \leq j < q \wedge A[i] + A[j] = X + Y$

$\equiv$ $\langle$ Hipótesis $A[i] = X \wedge A[j] = Y \equiv true \Rightarrow p \leq i < q \wedge p \leq j < q\rangle$

$true \wedge true \wedge A[i] + A[j] = X + Y$

$\equiv$ $\langle$ Hipótesis $A[i] = X \wedge A[j] = Y$ $Sustitucion\rangle$

$X + Y = X + Y$

$\equiv$ $\langle q = q = true\rangle$

$true$

■

3.2.2 Ejercicio 16

Mostrar la correctitud.

$\{(\forall i : 0 \leq i < k : A[i] = 2^i) \wedge 0 \leq k \leq N \wedge k \neq N\}$

$A[k] := 2^k$

$\{(\forall i : 0 \leq i < k : A[i] = 2^i) \wedge 0 \leq k \leq N\}$

Solución

Se debe demostrar por regla de la asignación de arreglos.

$P \Rightarrow 0 \leq k < N \wedge Q(A := A(k : 2^k))$

Sustituyendo los predicados P y Q.

$(\forall i : 0 \leq i < k : A[i] = 2^i) \wedge 0 \leq k \leq N \wedge k \neq N \Rightarrow 0 \leq k < N \wedge ((\forall i : 0 \leq i < k : A[i] = 2^i) \wedge 0 \leq k \leq N)(A := A(k : 2^k))$

Por suposición del antecedente y empezando por el consecuente para alcanzar true.

$0 \leq k < N \wedge ((\forall i : 0 \leq i < k : A[i] = 2^i) \wedge 0 \leq k \leq N)(A := A(k : 2^k))$

$\equiv$ $\langle$ Hipótesis $0 \leq k \leq N \wedge k \neq N \equiv 0 \leq k < N \equiv true\rangle$

$true \wedge ((\forall i : 0 \leq i < k : A[i] = 2^i) \wedge 0 \leq k \leq N)(A := A(k : 2^k))$

$\equiv$ $\langle$ Sustitución Textual $\rangle$

$(\forall i : 0 \leq i < k : A(k : 2^k)[i] = 2^i) \wedge 0 \leq k \leq N$

$\equiv$ $\langle$ Definición Asignación de arreglos $k \neq i \Rightarrow A(k : 2^k)[i] = A[i]\rangle$

$(\forall i : 0 \leq i < k : A[i] = 2^i) \wedge 0 \leq k \leq N$

$\equiv$ $\langle$ Hipótesis $(\forall i : 0 \leq i < k : A[i] = 2^i) \wedge 0 \leq k \leq N\rangle$

$true$

■

3.2.3 Ejercicio 17

Mostrar la correctitud.

$\{(\forall i : 0 \leq i < k : S[i] = V[i] \cdot V[i]) \wedge 0 \leq k \leq N \wedge k \neq N\}$

$S[k], k := V[k] \cdot V[k], k + 1$

$\{(\forall i : 0 \leq i < k : S[i] = V[i] \cdot V[i]) \wedge 0 \leq k \leq N\}$

Solución

Aplicando la regla de correctitud para la asignación de arreglos, se tiene.

$P \Rightarrow 0 \leq k < N \wedge Q(S, k := S(k : V[k] \cdot V[k]), k + 1)$

Sustituyendo los predicados P y Q.

$(\forall i : 0 \leq i < k : S[i] = V[i] \cdot V[i]) \wedge 0 \leq k \leq N \wedge k \neq N \Rightarrow$

$0 \leq k < N \wedge ((\forall i : 0 \leq i < k : S[i] = V[i] \cdot V[i]) \wedge 0 \leq k \leq N)(S, k := S(k : V[k] \cdot V[k]), k + 1)$

Sustitución del antecedente, empezando por el consecuente para alcanzar true.

$0 \leq k < N \wedge ((\forall i : 0 \leq i < k : S[i] = V[i] \cdot V[i]) \wedge 0 \leq k \leq N)(S, k := A(k : V[k] \cdot V[k]), k + 1)$

$\equiv$ $\langle$ Hipótesis $0 \leq k \leq N \wedge k \neq N \equiv 0 \leq k < N \equiv true\rangle$

$true \wedge ((\forall i : 0 \leq i < k : S[i] = V[i] \cdot V[i]) \wedge 0 \leq k \leq N)(S, k := S(k : V[k] \cdot V[k]), k + 1)$

$\equiv$ $\langle$ Sustitución Textual $\rangle$

$(\forall i : 0 \leq i < k + 1 : S(k : V[k] \cdot V[k])[i] = V[i] \cdot V[i]) \wedge 0 \leq k + 1 \leq N$

$\equiv$ $\langle$ Sacando último término $i = k\rangle$

$(\forall i : 0 \leq i < k : S(k : V[k] \cdot V[k])[i] = V[i] \cdot V[i]) \wedge S(k : V[k] \cdot V[k])[k] = V[k] \cdot V[ik] \wedge 0 \leq k + 1 \leq N$

$\equiv$ $\langle$ Definición Asignación de Arreglos $i = k \Rightarrow a[k : F][i] = F \wedge i \neq k \Rightarrow a[k : F][i] = a[i]\rangle$

$(\forall i : 0 \leq i < k : S[i] = V[i] \cdot V[i]) \wedge V[k] \cdot V[k] = V[k] \cdot V[k] \wedge 0 \leq k + 1 \leq N$

$\equiv$ $\langle$ Hipótesis $(\forall i : 0 \leq i < k : S[i] = V[i] \cdot V[i]);\ q = q = true\rangle$

$true \wedge true \wedge 0 \leq k + 1 \leq N$

$\Leftarrow$ $\langle k \geq 0 \Rightarrow k + 1 \geq 0;\ k \leq N \wedge k \neq N \equiv k < N \Rightarrow k + 1 \leq N\rangle$

$0 \leq k \leq N \wedge k \neq N$

$\equiv$ $\langle$ Hipótesis $0 \leq k \leq N \wedge k \neq N\rangle$

$true$

■

4 Procedimientos y Funciones

4.1 Caso general B2 (Teoría)

$\neg ocurreLibre('a, b', E) \equiv false$

$proc\ p\ (entrada\ x;\ entrada - salida\ y;\ salida\ z)$

$\{P_{def}\}$

$\{Q_{def}\}$

S

Llamada.

$\{P_{llam}\}\ p(E, a, b)\ \{Q_{llam}\}$

Demostraciones:

1. $[P_{llam} \Rightarrow P_{def}(x, y := E, a)]$
2. $[P_{llam}(a, b := A, B) \wedge Q_{def}(x, y_0, y, z := E(a, b := A, B), A, a, b) \Rightarrow Q_{llam}]$

4.2 Ejercicio Practica Ene-Mar 2025

Dado el procedimiento.

$proc\ sumar\ (entrada\ i, d, N : int; entrada - salida\ a :\ array[0..N)\ de\ int)$

$\{P_{def} : d > 0 \land N > 0 \land 0 \leq i < N\}$

$\{Q_{def} : a[i] = a_0[i] + d\}$

Demostrar la correctitud.

$\{P_{llam} : a \geq M \geq 73 \land p[0] = 0\}$

$sumar(p[0], a, M, p)$

$\{Q_{llam} : p[0] \geq 73\}$

Demostración:

Entradas: $p[0](p),\ a,\ M$

Entradas-salidas: p

Se debe demostrar (Definición):

1. $[P_{llam} \Rightarrow P_{def}(x, y := E, a)]$
2. $[P_{llam}(a, b := A, B) \land Q_{def}(x, y_0, y, z := E(a, b := A, B), A, a, b) \Rightarrow Q_{llam}]$

Donde:

$('parametros' \rightarrow\ 'argumentos')$

$x =\ 'i, d, N' \rightarrow\ 'p[0], a, M'$

$y =\ 'a' \rightarrow\ 'p'$

$z =''$

Las sustituciones a realizar:

1. $[P_{llam} \Rightarrow P_{def}(i, d, N, a := p[0], a, M, p)]$
2. $[P_{llam}(p := P) \land Q_{def}(i, d, N, a_0, a := (p[0])(p := P), a, M, P, p) \Rightarrow Q_{llam}]$

Demostraando las expresiones:

1. $[P_{llam} \Rightarrow P_{def}(i, d, N, a := p[0], a, M, p)]$

   $a \geq M \geq 73 \land p[0] = 0 \Rightarrow (d > 0 \land N > 0 \land 0 \leq i < N)(i, d, N, a := p[0], a, M, p)$

   Suposición del antecedente empezando por consecuente para alcanzar true.

   $(d > 0 \land N > 0 \land 0 \leq i < N)(i, d, N, a := p[0], a, M, p)$

$\equiv$ $\langle S.T \rangle$

$a > 0 \land M > 0 \land 0 \leq p[0] < M$

$\equiv$ $\langle$ Hipótesis $p[0] = 0 \rangle$

$a > 0 \land M > 0 \land 0 \leq 0 < M$

$\equiv$ $\langle$ Hipótesis $M \geq 73 \Rightarrow M > 0 \equiv true \land a \geq M \geq 73 \Rightarrow a > 0 \equiv true \rangle$

$true \land true \land 0 \leq 0 < M$

$\equiv$ $\langle$ Aritmética $0 \leq 0 < M \equiv 0 \leq 0 \land 0 < M \rangle$

$0 \leq 0 \land 0 < M$

$\equiv$ $\langle$ Aritmética $0 \leq 0 \equiv true < M \rangle$

$0 < M$

$\equiv$ $\langle$ Hipótesis $M \geq 73 \Rightarrow M > 0 \equiv true \rangle$

$true$

2. $[P_{llam}(p := P) \land Q_{def}(i, d, N, a_0, a := (p[0])(p := P), a, M, P, p) \Rightarrow Q_{llam}]$

   $(a \geq M \geq 73 \land p[0] = 0)(p := P) \land Q_{def}(i, d, N, a_0, a := (p[0])(p := P), a, M, P, p) \Rightarrow p[0] \geq 73$

   Debilitamiento.

   $(a \geq M \geq 73 \land p[0] = 0)(p := P) \land (a[i] = a_0[i] + d)(i, d, N, a_0, a := (p[0])(p := P), a, M, P, p)$

   $\equiv$ $\langle S.T \rangle$

   $a \geq M \geq 73 \land P[0] = 0 \land p[P[0]] = P[P[0]] + a$

   $\equiv$ $\langle$ Sustitución $P[0] = 0 \rangle$

   $a \geq M \geq 73 \land P[0] = 0 \land p[0] = P[0] + a$

   $\equiv$ $\langle$ Sustitución $P[0] = 0 \rangle$

   $a \geq M \geq 73 \land P[0] = 0 \land p[0] = 0 + a$

   $\equiv$ $\langle$ Sustitución $p[0] = a \rangle$

   $p[0] \geq M \geq 73 \land P[0] = 0 \land p[0] = a$

   $\Rightarrow$ $\langle$ Transitividad $a \geq b \geq c \Rightarrow a \geq c \rangle$

   $p[0] \geq 73 \land P[0] = 0 \land p[0] = a$

   $\Rightarrow$ $\langle$ Debilitamiento $p \land q \Rightarrow p \rangle$

$p[0] \geq 73$

■

4.2.1 Ejercicio Examen Ene-Mar 2025

Dado el procedimiento.

$proc\ intercambiar\ \ (entrada\ i, N : int;\ entrada - salida\ a, b :\ array[0..N)\ de\ int)$

$\{Pre : N > 0 \land 0 \leq i < N\}$

$\{Post : a[i] = b_0[i] \land b[i] = a_0[i]\}$

Demostrar la correctitud.

$\{M > 5 \land p[0] > -1 \land a[0] > 3 \land p[0] + a[0] = X \land X < 3 \land a[x] = C\}$

$intercambiar(p[0] + a[0], M, p, a)$

$\{p[X] = C\}$

Demostración:

$1.[P_{llam} \Rightarrow P_{def}(x, y := E, a)]$

$2.m[P_{llam}(a, b := A, B) \land Q_{def}(x, y_0, y, z := E, A, a, b) \Rightarrow Q_{llam}]$

$parameter \rightarrow argumentos$

Entradas: $i, N \rightarrow p[0] + a[0], M$

Entrada-Salida: $a, b \rightarrow p, a$

Valores iniciales: b_0, a_0

Las pruebas a realizar quedan:

$1.[P_{llam} \Rightarrow P_{def}(i, N, a, b := p[0] + a[0], M, p, a)]$

$2.[P_{llam}(p, a := P, A) \land Q_{def}(i, N, a_0, b_0, a, b := (p[0] + a[0])(p, a := P, A), M, P, A, p, a) \Rightarrow Q_{llam}]$

Sustituyendo los predicados.

$1.[M > 5 \land p[0] > -1 \land a[0] > 3 \land p[0] + a[0] = X \land X < 3 \land a[x] = C \Rightarrow$
$(N > 0 \land 0 \leq i < N)(i, N, a, b := p[0] + a[0], M, p, a)]$

$2.[(M > 5 \land p[0] > -1 \land a[0] > 3 \land p[0] + a[0] = X \land X < 3 \land a[x] = C)(p, a := P, A) \land (a[i] = b_0[i] \land b[i] = a_0[i])(i, N, a_0, b_0, a, b := (p[0] + a[0])(p, a := P, A), M, P, A, p, a) \Rightarrow p[X] = C]$

Demostración 1.

$M > 5 \land p[0] > -1 \land a[0] > 3 \land p[0] + a[0] = X \land X < 3 \land a[x] = C \Rightarrow$
$(N > 0 \land 0 \leq i < N)(i, N, a, b := p[0] + a[0], M, p, a)$

Aplicando sustitución textual.

$M > 5 \land p[0] > -1 \land a[0] > 3 \land p[0] + a[0] = X \land X < 3 \land a[x] = C \Rightarrow M > 0 \land 0 \leq p[0] + a[0] < M$

Por debilitamiento.

$M > 5 \land p[0] > -1 \land a[0] > 3 \land p[0] + a[0] = X \land X < 3 \land a[x] = C$

$\Rightarrow$ $\langle a > b \wedge c > d \Rightarrow a + c > b + d \rangle$

$M > 5 \wedge p[0] + a[0] > 2 \wedge p[0] + a[0] = X \wedge X < 3 \wedge a[x] = C$

$\equiv$ $\langle$Sustitución$p[0] + a[0] = X\rangle$

$M > 5 \wedge X > 2 \wedge p[0] + a[0] = X \wedge X < 3 \wedge a[x] = C$

$\Rightarrow$ $\langle$Debilitamiento$p \wedge q \Rightarrow p\rangle$

$X > 2 \wedge X < 3$

$\equiv$ $\langle$Rango vacío$\rangle$

$false$

$\equiv$ $\langle false \Rightarrow p \equiv true\rangle$

$M > 0 \wedge 0 \leq p[0] + a[0] < M$

■

Demostración 2.

$(M > 5 \wedge p[0] > -1 \wedge a[0] > 3 \wedge p[0] + a[0] = X \wedge X < 3 \wedge a[X] = C)(p, a := P, A) \wedge (a[i] = b_0[i] \wedge b[i] = a_0[i])(i, N, a_0, b_0, a, b := (p[0] + a[0]))(p, a := P, A), M, P, A, p, a) \Rightarrow p[X] = C$

Aplicando debilitamiento.

$(M > 5 \wedge p[0] > -1 \wedge a[0] > 3 \wedge p[0] + a[0] = X \wedge X < 3 \wedge a[X] = C)(p, a := P, A) \wedge (a[i] = b_0[i] \wedge b[i] = a_0[i])(i, N, a_0, b_0, a, b := (p[0] + a[0]))(p, a := P, A), M, P, A, p, a)$

$\equiv$ $\langle$ S.T $p, a := P, A\rangle$

$M > 5 \wedge P[0] > -1 \wedge A[0] > 3 \wedge P[0] + A[0] = X \wedge X < 3 \wedge A[X] = C \wedge (a[i] = b_0[i] \wedge b[i] = a_0[i])(i, N, a_0, b_0, a, b := (p[0] + a[0]))(p, a := P, A), M, P, A, p, a)$

$\equiv$ $\langle$ S.T $\rangle$

$M > 5 \wedge P[0] > -1 \wedge A[0] > 3 \wedge P[0] + A[0] = X \wedge X < 3 \wedge A[X] = C \wedge p[P[0] + A[0]] = A[P[0] + A[0]] \wedge a[P[0] + A[0]] = P[P[0] + A[0]]$

$\equiv$ $\langle$ S.T $P[0] + A[0] = X\rangle$

$M > 5 \wedge P[0] > -1 \wedge A[0] > 3 \wedge P[0] + A[0] = X \wedge X < 3 \wedge A[X] = C \wedge p[X] = A[X] \wedge a[X] = P[X]$

$\equiv$ $\langle$ S.T $A[X] = C\rangle$

$M > 5 \wedge P[0] > -1 \wedge A[0] > 3 \wedge P[0] + A[0] = X \wedge X < 3 \wedge A[X] = C \wedge p[X] = C \wedge a[X] = P[X]$

$\Rightarrow$ $\langle$ Debilitamiento $p \wedge q \Rightarrow p\rangle$

$p[X] = C$

■

4.3 Ejercicios Prof. Chang

4.3.1 Ejercicio 7: Corrección de llamadas a procedimientos. Euclides

Dado el siguiente procedimiento:

proc euclides(in b, c :int ; out q,r :int)

$\{P : b \geq 0 \wedge c > 0\}$

$\{Q : b = q \cdot c + r \wedge 0 \leq r < c\}$

Tenga en cuenta que los operadores div y mod se definen en base a esta descomposición de b:

$$\text{b div c = q , y b mod c = r}$$

Si la siguiente tripleta de Hoare es correcta, demuéstrela formalmente. En caso de ser incorrecta, explique por qué:

$\{q = 5 \wedge 0 \leq r < 3\}$ $euclides(q + r, q - r, b, c)$ $\{(b = 1 \vee b = 2) \wedge 0 \leq c < 3\}$

Solución

Se usa el caso general B2. Por lo que se debe demostrar lo siguiente.

1.$[P_{llam} \Rightarrow P_{def}(x, y := E, a)]$

2.$[P_{llam}(a, b := A, B) \wedge Q_{def}(x, y_0, y, z := E(a, b := A, B), A, a, b) \Rightarrow Q_{llam}]$

Para este problema particular se tienen las siguientes variables:

$$parametros \rightarrow argumentos$$

Entradas (x): $b, c \rightarrow (q + r), (q - r)$

Salidas (z): $q, r \rightarrow b, c$

Se debe probar:

1.$[P_{llam} \Rightarrow P_{def}(b, c := q + r, q - r)]$

2.$[P_{llam} \wedge Q_{def}(b, c, q, r := q + r, q - r, b, c) \Rightarrow Q_{llam}]$

Prueba 1.

$q = 5 \wedge 0 \leq r < 3 \Rightarrow q + r \geq 0 \wedge q - r > 0$

Aplicando método de suposición del antecedente.

$true$

$\equiv$ $\langle$ Hipótesis $0 \leq r < 3 \rangle$

$0 \leq r < 3$

$\equiv$ $\langle$ Aritmética $\rangle$

$0 \leq r \wedge r < 3$

$\equiv$ $\langle$ Aritmética sumar 5 a ambos lados de $0 \leq r \rangle$

$5 \leq r + 5 \wedge r < 3$

$\equiv$ $\langle$ Aritmética resto 5 a ambos lados de $r < 3\rangle$

$5 \leq r + 5 \wedge r - 5 < 3 - 5$

$\equiv$ $\langle$ Aritmética $\rangle$

$5 \leq r + 5 \wedge r - 5 < -2$

$\equiv$ $\langle$ Aritmética: multiplicar -1 a ambos lados de $r - 5 < -2\rangle$

$5 \leq r + 5 \wedge 5 - r > 2$

$\Rightarrow$ $\langle$ Hipótesis y sustitución Leibniz $q = 5\rangle$

$5 \leq r + q \wedge q - r > 2$

$\Rightarrow$ $\langle 5 \leq a \Rightarrow 0 \leq a\rangle$

$0 \leq r + q \wedge q - r > 2$

$\Rightarrow$ $\langle a > 2 \Rightarrow a > 0\rangle$

$0 \leq r + q \wedge q - r > 0$

$\equiv$ $\langle$ Simetría $a > b \equiv b < a\rangle$

$r + q \geq 0 \wedge q - r > 0$

■

Prueba 2.

$q = 5 \wedge 0 \leq r < 3 \wedge q + r = b \cdot (q - r) + c \wedge 0 \leq c < q - r \Rightarrow (b = 1 \vee b = 2) \wedge 0 \leq c < 3$

Contraejemplo

Hipótesis:

$q = 5 \wedge r = 1 \wedge 0 \leq c < q - r \Rightarrow q = 5 \wedge r = 1 \wedge 0 \leq c < 5 - 1 \Rightarrow q = 5 \wedge r = 1 \wedge 0 \leq c < 4 \Rightarrow q = 5 \wedge r = 1 \wedge c = 3$

Tomemos la siguiente hipótesis para sustituir los valores particulares de q, r, c:

$q + r = b \cdot (q - r) + c \Rightarrow 5 + 1 = b \cdot (5 - 1) + 3 \equiv 6 = b \cdot 4 + 3 \equiv 3 = b \cdot 4 \Rightarrow b = \frac{3}{4}$

Lo cual no corresponde con el consecuente que indica que b es 1 o 2, de igual manera indica que $c < 3$.

4.3.2 Ejercicio 8: Corrección de llamadas a procedimientos. Productos Notables

Dado el siguiente procedimiento:

proc productoNotableLimitado(in a, b :int ; out z :int)

$\{P : b \geq 0 \wedge b > a\}$

$\{Q : z = a^2 + 2ab + b^2\}$

Si la siguiente tripleta de Hoare es correcta, demuéstrela formalmente. En caso de ser incorrecta, explique por qué:

$\{0 < b < 5 \wedge z = -1 \wedge c = 1\}$ $productoNotableLimitado(b + z, b + c, b)$ $\{2^2 \leq b \leq 2^6\}$

Solución

Se implementará el caso general B2, cuyas pruebas se muestran a continuación.

1.$[P_{llam} \Rightarrow P_{def}(x, y := E, a)]$

2.$[P_{llam}(a, b := A, B) \wedge Q_{def}(x, y_0, y, z := E(a, b := A, B), A, a, b) \Rightarrow Q_{llam}]$

Para este problema se tiene lo siguiente:

parámetros $\rightarrow$ argumentos

Entradas (x): $a, b \rightarrow b + z, b + c$

Salidas (z): $z \rightarrow b$

Valor inicial (out b): $b \rightarrow B$

De esta manera las pruebas a realizar son las siguientes.

1.$[P_{llam} \Rightarrow P_{def}(a, b := b + z, b + c)]$

2.$[P_{llam}(b := B) \wedge Q_{def}(a, b, z := (b + z)(b := B), (b + c)(b := B), b) \Rightarrow Q_{llam}]$

Prueba 1.

$0 < b < 5 \wedge z = -1 \wedge c = 1 \Rightarrow b + c \geq 0 \wedge b + c > b + z$

Por debilitamiento.

$0 < b < 5 \wedge z = -1 \wedge c = 1$

$\equiv$ $\langle$ Aritmetica $\rangle$

$0 < b \wedge b < 5 \wedge z = -1 \wedge c = 1$

$\equiv$ $\langle$ Aritmetica sumando 1 a ambos lados de $0 < b\rangle$

$1 < b + 1 \wedge b < 5 \wedge z = -1 \wedge c = 1$

$\Rightarrow$ $\langle a > 1 \Rightarrow a > 0\rangle$

$b + 1 > 0 \wedge b < 5 \wedge z = -1 \wedge c = 1$

$\Rightarrow$ $\langle a > 0 \Rightarrow a \geq 0 \wedge a > 0\rangle$

$b + 1 \geq 0 \wedge b + 1 > 0 \wedge b < 5 \wedge z = -1 \wedge c = 1$

$\Rightarrow \quad \langle a > 0 \Rightarrow a + 1 > 0 \Rightarrow a + 1 > a - 1 \rangle$

$b + 1 \geq 0 \wedge b + 1 > b - 1 \wedge b < 5 \wedge z = -1 \wedge c = 1$

$\Rightarrow \quad \langle$ Sustitucion Leibniz $c = 1; z = -1 \rangle$

$b + c \geq 0 \wedge b + c > b - z$

■

Prueba 2.

$(0 < b < 5 \wedge z = -1 \wedge c = 1)(b := B) \wedge (z = a^2 + 2ab + b^2)(a, b, z := (b + z)(b := B), (b + c)(b := B), b) \Rightarrow 2^2 \leq b \leq 2^6$

$\equiv \quad \langle$ Sustitución $b := B \rangle$

$0 < B < 5 \wedge z = -1 \wedge c = 1 \wedge (z = a^2 + 2ab + b^2)(a, b, z := (B + z), (B + c), b) \Rightarrow 2^2 \leq b \leq 2^6$

$\equiv \quad \langle$ Sustitución $\rangle$

$0 < B < 5 \wedge z = -1 \wedge c = 1 \wedge b = (B + z)^2 + 2(B + z)(B + c) + (B + c)^2 \Rightarrow 2^2 \leq b \leq 2^6$

Debilitamiento.

$0 < B < 5 \wedge z = -1 \wedge c = 1 \wedge b = (B + z)^2 + 2(B + z)(B + c) + (B + c)^2$

$\Rightarrow \quad \langle$ Sustitución Leibniz $z = -1; c = 1 \rangle$

$0 < B < 5 \wedge b = (B - 1)^2 + 2(B - 1)(B + 1) + (B + 1)^2 \Rightarrow 2^2 \leq b \leq 2^6$

$\equiv \quad \langle$ Aritmética $\rangle$

$0 < B < 5 \wedge b = B^2 - 2B + 1 + 2B^2 - 2 + B^2 + 2B + 1$

$\equiv \quad \langle$ Aritmética $\rangle$

$0 < B < 5 \wedge b = 4B^2$

$\equiv \quad \langle$ Aritmética $\rangle$

$0 < B^2 < 25 \wedge b = 4B^2$

$\equiv \quad \langle$ Aritmética $\rangle$

$0 < 4B^2 < 100 \wedge b = 4B^2$

$\Rightarrow \quad \langle$ Sustitución Leibniz $\rangle$

$0 < b < 100$

$\equiv \quad \langle$ Aritmética $\rangle$

$0 < b \wedge b < 5 \cdot 2^2$

$\Rightarrow \quad \langle b > 0 \Rightarrow b \geq 2^2;\ b < 5 \cdot 2^2 \Rightarrow b < 2^6 \rangle$

$2^2 \leq b \wedge b < 2^6$

$\equiv \quad \langle a < b \equiv a \leq b \wedge a \neq b \rangle$

$2^2 \leq b \wedge b \leq 2^6 \wedge b \neq 2^6$

$\Rightarrow \quad \langle p \wedge q \Rightarrow p \rangle$

$2^2 \leq b \wedge b \leq 2^6$

$\equiv$ $\langle$ Aritmética $\rangle$

$2^2 \leq b \leq 2^6$

■

4.4 Ejercicios Tarea 3 Sep-Dic 2013

4.4.1 Ejercicio 12

Mostrar la correctitud.

$\{x > 0\}$

$P(x)$

$\{x \mod 2 = 1\}$

proc $P(in - out\ a : entero)$

$\{pre : a \geq 0\}$

$\{post : a = 2a_0 + 1\}$

Solución

Vamos a aplicar el caso general B2, para lo cual se debe demostrar lo siguiente.

1.$[P_{llam} \Rightarrow P_{def}(a := x)]$

2.$[P_{llam}(x := X) \wedge Q_{def}(a_0, a := X, x) \Rightarrow Q_{llam}]$

Prueba 1.

$x > 0 \Rightarrow x \geq 0$

Fortalecimiento.

$x \geq 0$

$\equiv \quad \langle a \geq b \equiv a > b \vee a = b \rangle$

$x > 0 \vee x = 0$

$\Leftarrow \quad \langle p \Rightarrow p \vee q \rangle$

$x > 0$

■

Prueba 2.

$(x > 0)(x := X) \wedge (a = 2a_0 + 1)(a_0, a := X, x) \Rightarrow x \mod 2 = 1$

Debilitamiento.

$(x > 0)(x := X) \wedge (a = 2a_0 + 1)(a_0, a := X, x)$

$\equiv \quad \langle S.T \rangle$

$X > 0 \wedge x = 2X + 1$

$\equiv \quad \langle a \mod 2 = 1 \equiv \exists\, b > 0 | a = 2b + 1 \rangle$

$X > 0 \wedge x \mod 2 = 1$

$\Rightarrow \quad \langle p \wedge q \Rightarrow p \rangle$

$x \mod 2 = 1$

■

4.4.2 Ejercicio 13

Mostrar la correctitud.

$\{n \geq m \wedge m \geq 0\}$

$Pfactorial(n, c)$

$\{c = (\prod i | 1 \leq i \leq n : i)\}$

proc $Pfactorial(in\ x : entero;\ out\ f : entero)$

$\{pre : x \geq 0\}$

$\{post : f = (\prod i | 1 \leq i \leq x : i)\}$

Solución

Vamos a usar el caso general B2, para lo cual se debe demostrar lo siguiente.

1.$[P_{llam} \Rightarrow P_{def}(x := n)]$

2.$[P_{llam}(c := C) \wedge Q_{def}(x, f := n, c) \Rightarrow Q_{llam}]$

Prueba 1.

$n \geq m \wedge m \geq 0 \Rightarrow n \geq 0$

Debilitamiento.

$n \geq m \wedge m \geq 0$

$\Rightarrow \quad \langle Transitividad \rangle$

$n \geq 0$

■

Prueba 2.

$n \geq m \wedge m \geq 0 \wedge (f = (\prod i | 1 \leq i \leq x : i))(x, f := n, c) \Rightarrow c = (\prod i | 1 \leq i \leq n : i)$

Debilitamiento.

$n \geq m \wedge m \geq 0 \wedge (f = (\prod i | 1 \leq i \leq x : i))(x, f := n, c)$

$\equiv \quad \langle S.T \rangle$

$n \geq m \wedge m \geq 0 \wedge c = (\prod i | 1 \leq i \leq n : i)$

$\Rightarrow \quad \langle p \wedge q \Rightarrow p \rangle$

$c = (\prod i | 1 \leq i \leq n : i)$

■

4.4.3 Ejercicio 14

Mostrar la correctitud de la siguiente terna de Hoare.

$\{0 \leq k \leq N \wedge s = (\sum i | 0 \leq i < k : (\prod i | 0 \leq i \leq k : x^i)/(\prod i | 1 \leq i \leq k : i))\}$

$s, k := s + potencia(x, k)/factorial(k), k + 1$

$\{s = (\sum i | 0 \leq i < k : (\prod i | 0 \leq i \leq k : x^i)/(\prod i | 1 \leq i \leq k : i))\}$

De donde se tiene,

func $potencia(y : real, n : entero) \rightarrow real$

$\{pre : n \geq 0\}$

$\{post : potencia = (\prod i | 0 \leq i < n : y^i)\}$

func $factorial(x : entero) \rightarrow entero$

$\{pre : x \geq 0\}$

$\{post : factorial = (\prod i | 1 \leq i \leq x : i)\}$

Solución

Se definen las siguientes variables para las llamadas de las funciones potencia y factorial.

ϕ : Retorno de potencia.

θ : Retorno de factorial.

Los procedimientos quedan definidos como siguen.

proc $proc\_potencia(in\ y : real, n : entero;\ out\ p : real)$

$\{pre : n \geq 0\}$

$\{post : p = (\prod i | 0 \leq i < n : y^i)\}$

proc $proc\_factorial(in\ x : entero;\ out\ f : entero)$

$\{pre : x \geq 0\}$

$\{post : f = (\prod i | 1 \leq i \leq x : i)\}$

La tripleta nos queda.

$\{0 \leq k \leq N \wedge s = (\sum i | 0 \leq i < k : (\prod i | 0 \leq i \leq k : x^i)/(\prod i | 1 \leq i \leq k : i))\}$

$proc\_potencia(x, k, \phi)$;

$proc\_factorial(k, \theta)$;

$s, k := s + \phi/\theta, k + 1$

$\{s = (\sum i | 0 \leq i < k : (\prod i | 0 \leq i \leq k : x^i)/(\prod i | 1 \leq i \leq k : i))\}$

Se determina la precondición más débil.

$\{0 \leq k \leq N \wedge s = (\sum i | 0 \leq i < k : (\prod i | 0 \leq i \leq k : x^i)/(\prod i | 1 \leq i \leq k : i))\}$

$proc\_potencia(x, k, \phi)$;

$proc\_factorial(k, \theta)$;

$\{s + \phi/\theta = (\sum i | 0 \leq i < k + 1 : (\prod i | 0 \leq i \leq k + 1 : x^i)/(\prod i | 1 \leq i \leq k + 1 : i))\}$

$s, k := s + \phi/\theta, k + 1$

$\{s = (\sum i | 0 \leq i < k : (\prod i | 0 \leq i \leq k : x^i)/(\prod i | 1 \leq i \leq k : i))\}$

4.4.4 Ejercicio 18

Mostrar la correctitud.

$\{i \geq 0 \wedge i < N-1 \wedge A[i] = X \wedge A[i+1] = Y\}$

$Intercambio(N, A, i)$

$\{A[i] = Y \wedge A[i+1] = X\}$

proc $Intercambio(in\ M : entero;\ in-out\ A : arreglo\ [0..M)\ de\ enteros; in\ r : entero)$

$\{pre : 0 \leq r < M-1\}$

$\{post : A[r] = A_0[r+1] \wedge A[r+1] = A_0[r]\}$

Solución

Aplicando la regla general B2 tenemos lo siguiente.

$$parameters \rightarrow argumentos$$

In: $M, r \rightarrow N, i$

In-out: $A \rightarrow A$. Valor inicial: A_0

Se debe demostrar lo siguiente.

1.$P_{llam} \Rightarrow P_{def}(M, r, A := N, i, A)$

2.$P_{llam}(A := AA) \wedge Q_{def}(M, r, A_0, A := N, i, AA, A) \Rightarrow Q_{llam}$

Prueba 1.

$i \geq 0 \wedge i < N-1 \wedge A[i] = X \wedge A[i+1] = Y \Rightarrow (0 \leq r < M-1)(M, r, A := N, i, A)$

Suposición del antecedente, empezando por el consecuente para alcanzar true.

$(0 \leq r < M-1)(M, r, A := N, i, A)$

$\equiv$ $\langle S.T \rangle$

$0 \leq i < N-1$

$\equiv$ $\langle$ Aritmética $\rangle$

$0 \leq i \wedge i < N-1$

$\equiv$ $\langle$ Hipótesis $i \geq 0 \equiv true;\ i < N-1 \equiv true \rangle$

$true \wedge true$

$\equiv$ $\langle$ Identidad $true \wedge true \equiv true \rangle$

$true$

■

Prueba 2.

$(i \geq 0 \wedge i < N-1 \wedge A[i] = X \wedge A[i+1] = Y)(A := AA) \wedge$

$(A[r] = A_0[r+1] \wedge A[r+1] = A_0[r])(M, r, A_0, A := N, i, AA, A) \Rightarrow A[i] = Y \wedge A[i+1] = X$

Debilitamiento.

$(i \geq 0 \wedge i < N-1 \wedge A[i] = X \wedge A[i+1] = Y)(A := AA) \wedge$

$(A[r] = A_0[r+1] \wedge A[r+1] = A_0[r])(M, r, A_0, A := N, i, AA, A)$

$\equiv$ $\langle$ Sustitución textual $A := AA \rangle$

$i \geq 0 \wedge i < N-1 \wedge AA[i] = X \wedge AA[i+1] = Y \wedge$

$(A[r] = A_0[r+1] \wedge A[r+1] = A_0[r])(M, r, A_0, A := N, i, AA, A)$

$\equiv$ $\langle$ Sustitución textual $\rangle$

$i \geq 0 \wedge i < N-1 \wedge AA[i] = X \wedge AA[i+1] = Y \wedge A[i] = AA[i+1] \wedge A[i+1] = AA[i]$

$\equiv$ $\langle$ Sustitución $AA[i] = X \wedge AA[i+1] = Y \rangle$

$i \geq 0 \wedge i < N-1 \wedge AA[i] = X \wedge AA[i+1] = Y \wedge A[i] = Y \wedge A[i+1] = X$

$\Rightarrow$ $\langle p \wedge q \Rightarrow p \rangle$

$A[i] = Y \wedge A[i+1] = X$

■