

Práctico 4: Introducción al Cálculo de Programas Imperativos

Algoritmos y Estructuras de Datos I

2^{do} cuatrimestre 2018

1. Para cada uno de los siguientes programas, calcule la precondition más débil y las anotaciones intermedias. Para ello utilice el transformador de predicados wp .

a) $\text{Var } x, y : \text{Num};$
 $\{ \quad \}$
 $x := x + y$
 $\{x = 6 \wedge y = 5\}$

b) $\text{Var } x : \text{Num};$
 $\{ \quad \}$
 $x := 8$
 $\{x = 8\}$

c) $\text{Var } x : \text{Num};$
 $\{ \quad \}$
 $x := 8$
 $\{x = 7\}$

d) $\text{Var } x, y : \text{Num};$
 $\{ \quad \}$
 $x, y := y, x$
 $\{x = B \wedge y = A\}$

e) $\text{Var } x, y, a : \text{Num};$
 $\{ \quad \}$
 $a, x := x, y;$
 $\{ \quad \}$
 $y := a$
 $\{x = B \wedge y = A\}$

f) $\text{Var } x, y : \text{Num};$
 $\{ \quad \}$
if $x \geq y \rightarrow$
 $\{ \quad \}$
 $x := 0$
 $\{ \quad \}$
 $\square x \leq y \rightarrow$
 $\{ \quad \}$
 $x := 2$
 $\{ \quad \}$
fi
 $\{(x = 0 \vee x = 2) \wedge y = 1\}$

2. Demuestre que las siguientes ternas de Hoare son correctas. En todos los casos las variables x, y son de tipo Int , y a, b de tipo Bool .

a) $\{True\}$
if $x \geq 1 \rightarrow x := x + 1$
 $\square x \leq 1 \rightarrow x := x - 1$
fi
 $\{x \neq 1\}$

b) $\{x \neq y\}$
if $x > y \rightarrow \text{skip}$
 $\square x < y \rightarrow x, y := y, x$
fi
 $\{x > y\}$

c) $\{True\}$
 $x, y := y * y, x * x;$
if $x \geq y \rightarrow x := x - y$
 $\square x \leq y \rightarrow y := y - x$
fi
 $\{x \geq 0 \wedge y \geq 0\}$

d) $\{True\}$
if $\neg a \vee b \rightarrow a := \neg a$
 $\square a \vee \neg b \rightarrow b := \neg b$
fi
 $\{a \vee b\}$

e) $\{N \geq 0\}$
 $x := 0;$
do $x \neq N \rightarrow x := x + 1$
od
 $\{x = N\}$

f) $\{True\}$
 $r := N;$
do $r \neq 0 \rightarrow$
if $r < 0 \rightarrow r := r + 1$
 $\square r > 0 \rightarrow r := r - 1$
fi
od
 $\{r = 0\}$

3. Para cada uno de los siguientes programas, elija valores para las expresiones **E** y **F** de modo que las ternas de Hoare sean correctas.

- | | |
|--|---|
| <p>a) Var $x, y : Nat$;
 $\{True\}$
 $x, y := x + 1, \mathbf{E}$
 $\{y = x + 1\}$</p> | <p>b) Var $a, q, c, w : Num$;
 $\{q = a * c \wedge w = c^2\}$
 $a, q := a + c, \mathbf{E}$
 $\{q = a * c\}$</p> |
| <p>c) Const $A, B : Nat$;
 Var $q, r : Nat$;
 $\{A = q * B + r\}$
 $q := \mathbf{E}; r := r - B$
 $\{A = q * B + r\}$</p> | <p>d) Const $N : Num$;
 Var $x, y, p, q : Num$;
 $\{x * y + p * q = N\}$
 $x := x - p$;
 $q := \mathbf{F}$
 $\{x * y + p * q = N\}$</p> |

4. Especifique los siguientes problemas, enunciando pre y postcondición, y luego derive programas imperativos a partir las especificaciones.

- a) Calcular el mínimo de dos valores.
b) Calcular el valor absoluto de un número.

5. Demuestre que si la terna de Hoare (a) es correcta, entonces la terna (b) también lo es:

- | | |
|---|--|
| <p>a) $\{P\}$
 if $B_0 \rightarrow S_0$
 $\square B_1 \rightarrow S_1$
 fi
 $\{Q\}$</p> | <p>b) $\{P\}$
 if $B_0 \rightarrow S_0$
 $\square \neg B_0 \rightarrow S_1$
 fi
 $\{Q\}$</p> |
|---|--|

¿Qué utilidad tiene esta propiedad cuando se programa en lenguaje C?

6. Analice los siguientes programas anotados. En cada caso, describa en lenguaje natural la postcondición, y decida si el programa efectivamente valida las anotaciones.

- | | |
|---|--|
| <p>a) Const $N : Int, A : array[0, N) of Num$;
 Var $s : Num, i : Int$;
 $\{N \geq 0\}$
 $i, s := 0, 0$;
 do $i \neq N \rightarrow$
 $s := s + A.i$
 od
 $\{s = \langle \sum k : 0 \leq k < N : A.k \rangle\}$</p> | <p>b) Const $N : Int, A : array[0, N) of Num$;
 Var $s : Num, i : Int$;
 $\{N \geq 0\}$
 $i, s := 0, 0$;
 do $i \neq N \rightarrow$
 $i := i + 1$;
 $s := s + A.i$
 od
 $\{s = \langle \sum k : 0 \leq k < N : A.k \rangle\}$</p> |
| <p>c) Const $N : Int, A : array[0, N) of Num$;
 Var $s : Num, i : Int$;
 $\{N \geq 0\}$
 $i, s := -1, 0$;
 do $i \neq N \rightarrow$
 $i := i + 1$;
 $s := s + A.i$
 od
 $\{s = \langle \sum k : 0 \leq k < N : A.k \rangle\}$</p> | <p>d) Const $E : Num, N : Int, A : array[0, N) of Num$;
 Var $i : Int, r : Bool$;
 $\{N \geq 0\}$
 $i, r := 0, False$;
 do $i \neq N \wedge \neg r \rightarrow$
 if $A.i = E \rightarrow r := True$
 $\square A.i \neq E \rightarrow \mathbf{skip}$
 fi ;
 $i := i + 1$
 od
 $\{\langle \exists k : 0 \leq k < N : A.k = E \rangle \Rightarrow A.i = E\}$</p> |

7. Decida si los siguientes predicados son invariantes válidos del ciclo del programa 6(b). Justifique.

- a) $\{1 \leq i\}$
- b) $\{0 \leq i\}$
- c) $\{0 \leq i \leq N\}$
- d) $\{s = \langle \sum k : 0 \leq k < N : A.i \rangle\}$
- e) $\{0 \leq s \leq \langle \sum k : 0 \leq k < N : A.i \rangle\}$

8. *Swap*: Considere los siguientes programas que intercambian los valores de dos variables x e y de tipo *Int*:

$$\begin{array}{lll}
 x, y := y, x & z := x; & x := x - y; \\
 & x := y; & y := x + y; \\
 & y := z & x := y - x
 \end{array}$$

Especifique el *swap* (con pre y postcondición), y verifique que los programas satisfacen la especificación.

9. Derive un programa para calcular el máximo común divisor entre dos enteros positivos. Utilice la siguiente especificación:

```

Const X, Y : Int;
Var x, y : Int;
{X > 0 ∧ Y > 0 ∧ x = X ∧ y = Y}
S
{x = mcd.X.Y}

```

Utilice como invariante $\{I : x > 0 \wedge y > 0 \wedge \text{mcd}.x.y = \text{mcd}.X.Y\}$.

Para la derivación serán de utilidad las siguientes propiedades del *mcd*:

- a) $\text{mcd}.x.x = x$
- b) $\text{mcd}.x.y = \text{mcd}.y.x$
- c) $x > y \Rightarrow \text{mcd}.x.y = \text{mcd}.(x - y).y$
- d) $y > x \Rightarrow \text{mcd}.x.y = \text{mcd}.x.(y - x)$

10. Considere las siguientes definiciones recursivas de la función de exponenciación $\text{exp}.x.y$, especificada como $\text{exp}.x.y = x^y$:

a) Definición de complejidad lineal:

$$\text{exp}.x.y = (\begin{array}{l} y = 0 \rightarrow 1 \\ \square y \neq 0 \rightarrow x * \text{exp}.x.(y - 1) \end{array})$$

b) Definición de complejidad logarítmica:

$$\text{exp}.x.y = (\begin{array}{l} y = 0 \rightarrow 1 \\ \square y \neq 0 \rightarrow (\begin{array}{l} y \text{ mód } 2 = 0 \rightarrow \text{exp}.(x * x).(y \div 2) \\ \square y \text{ mód } 2 = 1 \rightarrow x * \text{exp}.x.(y - 1) \end{array}) \end{array})$$

Derive **dos** programas imperativos que calculen la exponenciación, cada uno utilizando una de las definiciones recursivas. Utilice la siguiente especificación:

```

Const X, Y : Int;
Var x, y, r : Int;
{x = X ∧ y = Y ∧ x ≥ 0 ∧ y ≥ 0}
S
{r = XY}

```

Utilice como invariante $\{I : y \geq 0 \wedge r * x^y = X^Y\}$.

Ejercicios extra

1. Dado $n > 0$, derive un programa que devuelva en la variable k la mayor potencia de 2 menor o igual que n , utilizando el invariante dado.

Precondición R : $\{n > 0\}$

Postcondición Q : $\{0 < k \leq n \wedge n < 2 * k \wedge \langle \exists j : 0 \leq j : k = 2^j \rangle\}$

Invariante I : $\{0 < k \leq n \wedge \langle \exists j : 0 \leq j : k = 2^j \rangle\}$

2. Calcule las precondiciones más débiles (*weakest preconditions*) P de modo que sean correctos los siguientes programas anotados. Agregue además las anotaciones intermedias en caso que haya sentencias compuestas con “;”. Asuma que las variables x, y, z, q, r son de tipo *Int*, las variables i, j de tipo *Nat* y las variables a, b de tipo *Bool*:

a) $\{P\} x := 8 \{x = 8\}$

b) $\{P\} x := 8 \{x \neq 8\}$

c) $\{P\} x := 9 \{x = 7\}$

d) $\{P\} x := x + 1; y := y - 2 \{x + y = 0\}$

e) $\{P\} x := x + 1; y := y - 1 \{x * y = 0\}$

f) $\{P\} x := x + 1; y := y - 1 \{x + y + 10 = 0\}$

g) $\{P\} z := z * y; x := x - 1 \{z * y^x = C\}$

h) $\{P\} x, y, z := 1, d, c \{x * x^y = c^d\}$

i) $\{P\} i, j := i + i, j; j := j + i \{i = j\}$

j) $\{P\} x := (x - y) * (x + y) \{x + y^2 = 0\}$

k) $\{P\} q, r := q + 1, r - y \{q * y + r = x\}$

l) $\{P\} a := a \equiv b; b := a \equiv b; a := a \equiv b \{(a \equiv B) \wedge (b \equiv A)\}$

3. Demuestre que si el programa

$$\begin{array}{l} \{P\} \\ \mathbf{if} B \rightarrow S \\ \mathbf{fi} \\ \{Q\} \end{array} \quad \text{es correcto, entonces también lo es} \quad \begin{array}{l} \{P\} \\ S \\ \{Q\} \end{array}$$

4. Sean S, S_0, S_1, T programas cualesquiera, B_0, B_1 guardas cualesquiera, E, F expresiones cualesquiera. En cada caso, ¿son equivalentes los programas i, ii e iii ? En caso afirmativo demuéstralo, en caso negativo dá un contraejemplo (instanciando los programas y las guardas).

a) $i) \quad x := E;$
 $\quad y := F;$

$ii) \quad y := F;$
 $\quad x := E;$

b) $i) \quad \mathbf{if} B_0 \rightarrow S$
 $\quad \square B_1 \rightarrow S$
 $\quad \mathbf{fi}$

$ii) \quad S$

c) $i) \quad \mathbf{if} B_0 \rightarrow S; S_0; T$
 $\quad \square B_1 \rightarrow S; S_1; T$
 $\quad \mathbf{fi}$

$ii) \quad \mathbf{if} B_0 \rightarrow S; S_0$
 $\quad \square B_1 \rightarrow S; S_1$
 $\quad \mathbf{fi};$
 $\quad T$

$iii) \quad S;$
 $\quad \mathbf{if} B_0 \rightarrow S_0; T$
 $\quad \square B_1 \rightarrow S_1; T$
 $\quad \mathbf{fi}$