

# Práctico 3

## Introducción a la programación imperativa

### Algoritmos y Estructuras de Datos I

1. Anote con los estados posibles y señale las trazas de los siguientes programas:

- |  |   |   |
|--|---|---|
| <p>a) <b>Var</b> <math>x, y : Num;</math><br/> <math>\llbracket \sigma_0 : (x \mapsto 1, y \mapsto 5) \rrbracket</math><br/> <math>x := x + y;</math><br/> <math>\llbracket \sigma_1 : \quad \quad \quad \rrbracket</math></p>   | <p>b) <b>Var</b> <math>x, y, a : Num;</math><br/> <math>\llbracket \sigma_0 : (x \mapsto 1, y \mapsto 5, a \mapsto 0) \rrbracket</math><br/> <math>a, x := x, y;</math><br/> <math>\llbracket \sigma_1 : \quad \quad \quad \rrbracket</math><br/> <math>y := a</math><br/> <math>\llbracket \sigma_2 : \quad \quad \quad \rrbracket</math></p>  | <p>c) <b>Var</b> <math>x, y : Num;</math><br/> <math>\llbracket \sigma_0 : (x \mapsto 1, y \mapsto 1) \rrbracket</math><br/> <b>if</b> <math>x \geq y \rightarrow</math><br/> <math>\llbracket \sigma_1 : \quad \quad \quad \rrbracket</math><br/> <math>x := 0</math><br/> <math>\llbracket \sigma_2 : \quad \quad \quad \rrbracket</math><br/> <math>\square x \leq y \rightarrow</math><br/> <math>\llbracket \sigma'_1 : \quad \quad \quad \rrbracket</math><br/> <math>x := 2</math><br/> <math>\llbracket \sigma'_2 : \quad \quad \quad \rrbracket</math><br/> <b>fi</b><br/> <math>\llbracket \sigma_3 : \quad \quad \quad , \sigma'_3 : \quad \quad \quad \rrbracket</math></p> |
| <p>d) <b>Var</b> <math>i : Int;</math><br/> <math>\llbracket \sigma_0 : (i \mapsto 4) \rrbracket</math><br/> <b>do</b> <math>i \neq 0 \rightarrow</math><br/> <math>\llbracket \sigma_1^1 : \quad , \sigma_1^2 : \quad , \dots \rrbracket</math><br/> <math>i := i - 1</math><br/> <math>\llbracket \sigma_2^1 : \quad , \sigma_2^2 : \quad , \dots \rrbracket</math><br/> <b>od</b><br/> <math>\llbracket \quad \quad \quad \rrbracket</math></p> | <p>e) <b>Var</b> <math>i : Int;</math><br/> <math>\llbracket \sigma_0 : (i \mapsto -1) \rrbracket</math><br/> <b>do</b> <math>i \neq 0 \rightarrow</math><br/> <math>\llbracket \sigma_1^1 : \quad , \sigma_1^2 : \quad , \dots \rrbracket</math><br/> <math>i := i - 1</math><br/> <math>\llbracket \sigma_2^1 : \quad , \sigma_2^2 : \quad , \dots \rrbracket</math><br/> <b>od</b><br/> <math>\llbracket \quad \quad \quad \rrbracket</math></p> | <p>f) <b>Var</b> <math>r : Int;</math><br/> <math>\llbracket \sigma_0 : (r \mapsto 3) \rrbracket</math><br/> <b>do</b> <math>r &lt; 0 \rightarrow</math><br/> <math>\llbracket \quad \quad \quad \rrbracket</math><br/> <math>r := r + 1</math><br/> <math>\llbracket \quad \quad \quad \rrbracket</math><br/> <math>\square r &gt; 0 \rightarrow</math><br/> <math>\llbracket \quad \quad \quad \rrbracket</math><br/> <math>r := r - 1</math><br/> <math>\llbracket \quad \quad \quad \rrbracket</math><br/> <b>od</b><br/> <math>\llbracket \quad \quad \quad \rrbracket</math></p>  |

2. Anotar con predicados los programas del ejercicio anterior.

3. Anotar con predicados los siguientes programas utilizando el transformador de predicados  $wp$ . A partir del resultado en los ejercicios 3d, 3e y 3f verifique que los resultados del ejercicio anterior correspondientes fueron correctos.

- |  |   |  |
|--|---|--|
| <p>a) <b>Var</b> <math>x, y : Num;</math><br/> <math>\{ \quad \quad \quad \}</math><br/> <math>x := x + y;</math><br/> <math>\{x = 6 \wedge y = 5\}</math></p> | <p>b) <b>Var</b> <math>x : Num;</math><br/> <math>\{ \quad \quad \}</math><br/> <math>x := 8</math><br/> <math>\{x = 8\}</math></p>   | <p>c) <b>Var</b> <math>x : Num;</math><br/> <math>\{ \quad \quad \}</math><br/> <math>x := 8</math><br/> <math>\{x = 7\}</math></p>  |
| <p>d) <b>Var</b> <math>x, y : Num;</math><br/> <math>\{ \quad \quad \}</math><br/> <math>x, y := y, x</math><br/> <math>\{x = B \wedge y = A\}</math></p>      | <p>e) <b>Var</b> <math>x, y, a : Num;</math><br/> <math>\{ \quad \quad \quad \}</math><br/> <math>a, x := x, y</math><br/> <math>\{ \quad \quad \quad \}</math><br/> <math>y := a</math><br/> <math>\{x = B \wedge y = A\}</math></p> | <p>f) <b>Var</b> <math>x, y : Num;</math><br/> <math>\{ \quad \quad \}</math><br/> <b>if</b> <math>x \geq y \rightarrow</math><br/> <math>\{ \quad \quad \}</math><br/> <math>x := 0</math><br/> <math>\{ \quad \quad \quad \}</math><br/> <math>\square x \leq y \rightarrow</math><br/> <math>\{ \quad \quad \}</math><br/> <math>x := 2</math><br/> <math>\{ \quad \quad \quad \}</math><br/> <b>fi</b><br/> <math>\{(x = 0 \vee x = 2) \wedge y = 1\}</math></p> |

4. Especifique y encuentre un programa que intercambia el valor de dos variables utilizando solo asignaciones simples. Luego verifique su corrección.

5. Demostrá que los siguientes programas anotados son correctos. 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)  $\{True\}$   
**if**  $x \geq y \rightarrow \mathbf{skip}$   
 $\square$   $x \leq y \rightarrow x, y := y, x$   
**fi**  
 $\{x \geq y\}$

c)  $\{True\}$   
 $x, y := y * x, x * x;$   
**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)  $\{N \geq 0\}$   
 $r := N;$   
**do**  $r < 0 \rightarrow r := r + 1$   
 $\square$   $r > 0 \rightarrow r := r - 1$   
**od**  
 $\{r = 0\}$

6. Calculá las expresiones  $\mathbf{E}$  y  $\mathbf{F}$  de modo que los siguientes programas sean correctos:

a)  $\mathbf{Var} \ a, q, c, w : Num;$   
 $\{q = a * c \wedge w = c^2\}$   
 $a, q := a + c, \mathbf{E}$   
 $\{q = a * c\}$

b)  $\mathbf{Var} \ q, r : Nat;$   
 $\mathbf{Const} \ A, B : Nat;$   
 $\{A = q * B + r\}$   
 $q := \mathbf{E}; r := r - B$   
 $\{A = q * B + r\}$

c)  $\mathbf{Var} \ x, y : Nat;$   
 $\{True\}$   
 $x, y := x + 1, \mathbf{E}$   
 $\{y = x + 1\}$

d)  $\mathbf{Var} \ x, y, p, q : Num;$   
 $\mathbf{Const} \ N : Num;$   
 $\{x * y + p * q = N\}$   
 $x := x - p;$   
 $q := \mathbf{F};$   
 $\{x * y + p * q = N\}$

7. Especifique los siguientes problemas con ternas de Hoare y luego derive un programa que los verifique.

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

8. Demostrar.

a) Si el programa

$\{P\}$ <b>if</b> $B_0 \rightarrow S_0$ $\square$ $B_1 \rightarrow S_1$ <b>fi</b> $\{Q\}$	es correcto, entonces también lo es	$\{P\}$ <b>if</b> $B_0 \rightarrow S_0$ $\square$ $\neg B_0 \rightarrow S_0$ <b>fi</b> $\{Q\}$
---	-------------------------------------	--

b) Si el programa

$\{P\}$ <b>if</b> $B \rightarrow S$ <b>fi</b> $\{Q\}$	es correcto, entonces también lo es	$\{P\}$ $S$ $\{Q\}$
--	-------------------------------------	---------------------------

9. Decir si son correctos los siguientes programas. En caso negativo explicar que hace el programa.

a) **Var**  $s, i : Num;$   
 $a : array[0, N) \text{ of } Num;$   
**Const**  $N : Num;$   
 $\{N \geq 0\}$   
 $i, s := 0, 0$   
**do**  $i \neq N \rightarrow$   
 $s := s + a.i$   
**od**  
 $\{s = \langle \sum k : 0 \leq k < N : a.i \rangle\}$

b) **Var**  $s, i : Num;$   
 $a : array[0, N) \text{ of } Num;$   
**Const**  $N : Num;$   
 $\{N \geq 0\}$   
 $i, s := 0, 0$   
**do**  $i \neq N \rightarrow$   
 $i := i + 1$   
 $s := s + a.i$   
**od**  
 $\{s = \langle \sum k : 0 \leq k < N : a.i \rangle\}$

c) **Var**  $s, i : Num;$   
 $a : array[0, N) \text{ of } Num;$   
**Const**  $N : Num;$   
 $\{N \geq 0\}$   
 $i, s := -1, 0$   
**do**  $i \neq N \rightarrow$   
 $i := i + 1$   
 $s := s + a.i$   
**od**  
 $\{s = \langle \sum k : 0 \leq k < N : a.i \rangle\}$

d) **Var**  $i : Num; r : Bool;$   
 $a : array[0, N) \text{ of } Num;$   
**Const**  $N : Num;$   
 $\{N \geq 0\}$   
 $i, r := 0, False$   
**do**  $i \neq N \wedge \neg r \rightarrow$   
**if**  $a.i = e \rightarrow r := True$   
 $\square a.i \neq e \rightarrow \text{skip}$   
**fi**  
 $i := i + 1$   
**od**  
 $\{\langle \exists k : 0 \leq k < N : a.k = e \rangle \Rightarrow a.i = e\}$

10. a) Usando las propiedades del transformador de predicados *weakest precondition*, demostrar las siguientes propiedades:

- 1)  $\{P\} S \{Q\} \wedge [P_0 \Rightarrow P] \Rightarrow \{P_0\} S \{Q\}$
- 2)  $\{P\} S \{Q\} \wedge [Q \Rightarrow Q_0] \Rightarrow \{P\} S \{Q_0\}$
- 3)  $\{P\} S \{Q\} \wedge \{P\} S \{R\} \equiv \{P\} S \{Q \wedge R\}$
- 4)  $\{P\} S \{Q\} \wedge \{R\} S \{Q\} \equiv \{P \vee R\} S \{Q\}$

b) Desde un punto de vista práctico, ¿qué aportan las propiedades anteriores a la hora de verificar la corrección de un programa?

11. Derivar los siguientes programas con bucles:

a) Desarrollar un algoritmo para calcular el máximo común divisor entre dos enteros positivos, especificado como:

**Var**  $x, y : Int;$   
**Const**  $X, Y : Int;$   
 $\{X > 0 \wedge Y > 0 \wedge x = X \wedge y = Y\}$   
 $S$   
 $\{x = mcd.X.Y\}$

Para derivar  $S$  serán de utilidad las propiedades del *mcd*:

- (1)  $mcd.x.x = x$
- (2)  $mcd.x.y = mcd.y.x$
- (3)  $x > y \Rightarrow mcd.x.y = mcd.(x - y).y$   
 $y > x \Rightarrow mcd.x.y = mcd.x.(y - x)$

b) Derivar dos programas que calculen  $r = X^Y$  a partir de cada una de las siguientes definiciones de la función exponencial:

$$(a) \quad \text{exp}(x, y) = \left( \begin{array}{l} y = 0 \rightarrow 1 \\ \square y \neq 0 \rightarrow x * \text{exp}(x, y - 1) \end{array} \right)$$

$$(b) \quad \text{exp}(x, y) = \left( \begin{array}{l} y = 0 \rightarrow 1 \\ \square y \neq 0 \rightarrow \left( \begin{array}{l} y \bmod 2 = 0 \rightarrow \text{exp}(x * x, y \div 2) \\ \square y \bmod 2 = 1 \rightarrow x * \text{exp}(x, y - 1) \end{array} \right) \end{array} \right)$$

Diseñar los dos programas a partir de:

$$\text{Precondición } R: \quad \{x = X \wedge y = Y \wedge x \geq 0 \wedge y \geq 0\}$$

$$\text{Postcondición } Q: \quad \{r = X^Y\}$$

$$\text{Invariante } P: \quad \{y \geq 0 \wedge r * x^y = X^Y\}$$

Para cada programa usar una de las definiciones. Tener en cuenta las mismas a la hora de decidir la manera de achicar la cota.

c) Dado  $n > 0$ , desarrollar un programa que devuelva en la variable  $k$  la mayor potencia de 2 menor o igual que  $n$ .

$$\text{Precondición } R: \quad \{n > 0\}$$

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

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