

# Práctico 3: Introducción a la programación imperativa

## Algoritmos y Estructuras de Datos I 2<sup>do</sup> cuatrimestre 2015

1. En cada uno de los siguientes programas, anotá los valores que las variables toman a medida que se ejecutan.

(a) **Var**  $x : Int$ ;  
 $\llbracket \sigma_0 : (x \mapsto 1) \rrbracket$   
 $x := 5$   
 $\llbracket \sigma_1 : \quad \quad \quad \rrbracket$

(b) **Var**  $x, y : Int$ ;  
 $\llbracket \sigma_0 : (x \mapsto 2, y \mapsto 5) \rrbracket$   
 $x := x + y$ ;  
 $\llbracket \sigma_1 : \quad \quad \quad \rrbracket$   
 $y := y + y$   
 $\llbracket \sigma_2 : \quad \quad \quad \rrbracket$

(c) **Var**  $x, y : Int$ ;  
 $\llbracket \sigma_0 : (x \mapsto 2, y \mapsto 5) \rrbracket$   
 $y := y + y$ ;  
 $\llbracket \sigma_1 : \quad \quad \quad \rrbracket$   
 $x := x + y$   
 $\llbracket \sigma_2 : \quad \quad \quad \rrbracket$

(d) **Var**  $x, y : Int$ ;  
 $\llbracket \sigma_0 : (x \mapsto 3, y \mapsto 1) \rrbracket$   
**if**  $x \geq y \rightarrow$   
 $\llbracket \sigma_1 : \quad \quad \quad \rrbracket$   
 $x := 0$   
 $\llbracket \sigma_2 : \quad \quad \quad \rrbracket$   
 $\square x \leq y \rightarrow$   
 $\llbracket \sigma'_1 : \quad \quad \quad \rrbracket$   
 $x := 2$   
 $\llbracket \sigma'_2 : \quad \quad \quad \rrbracket$   
**fi**  
 $\llbracket \sigma_3 : \quad \quad \quad \rrbracket$

(e) **Var**  $x, y : Int$ ;  
 $\llbracket \sigma_0 : (x \mapsto -100, y \mapsto 1) \rrbracket$   
**if**  $x \geq y \rightarrow$   
 $\llbracket \sigma_1 : \quad \quad \quad \rrbracket$   
 $x := 0$   
 $\llbracket \sigma_2 : \quad \quad \quad \rrbracket$   
 $\square x \leq y \rightarrow$   
 $\llbracket \sigma'_1 : \quad \quad \quad \rrbracket$   
 $x := 2$   
 $\llbracket \sigma'_2 : \quad \quad \quad \rrbracket$   
**fi**  
 $\llbracket \sigma'_3 : \quad \quad \quad \rrbracket$

(f) **Var**  $x, y : Int$ ;  
 $\llbracket \sigma_0 : (x \mapsto 1, y \mapsto 1) \rrbracket$   
**if**  $x \geq y \rightarrow$   
 $\llbracket \sigma_1 : \quad \quad \quad \rrbracket$   
 $x := 0$   
 $\llbracket \sigma_2 : \quad \quad \quad \rrbracket$   
 $\square x \leq y \rightarrow$   
 $\llbracket \sigma'_1 : \quad \quad \quad \rrbracket$   
 $x := 2$   
 $\llbracket \sigma'_2 : \quad \quad \quad \rrbracket$   
**fi**  
 $\llbracket \sigma_3 : \quad \quad \quad , \sigma'_3 : \quad \quad \quad \rrbracket$

(g) **Var**  $x, y : Int$ ;  
 $\llbracket \sigma_0 : (x \mapsto 2, y \mapsto 5) \rrbracket$   
 $y, x := y + y, x + y$   
 $\llbracket \sigma_1 : \rrbracket$

(h) **Var**  $i : Int$ ;  
 $\llbracket \sigma_0 : (i \mapsto 4) \rrbracket$   
**do**  $i \neq 0 \rightarrow$   
 $\llbracket \sigma_1^1 : \quad , \sigma_1^2 : \quad , \dots \rrbracket$   
 $i := i - 1$   
 $\llbracket \sigma_2^1 : \quad , \sigma_2^2 : \quad , \dots \rrbracket$   
**od**  
 $\llbracket \sigma_3 : \quad \quad \quad \rrbracket$

(i) **Var**  $i : Int$ ;  
 $\llbracket \sigma_0 : (i \mapsto 400) \rrbracket$   
**do**  $i \neq 0 \rightarrow$   
 $\llbracket \sigma_1^1 : \quad , \sigma_1^2 : \quad , \dots \rrbracket$   
 $i := 0$   
 $\llbracket \sigma_2^1 : \quad , \sigma_2^2 : \quad , \dots \rrbracket$   
**od**  
 $\llbracket \sigma_3 : \quad \quad \quad \rrbracket$

(j) **Var**  $i : Int$ ;  
 $\llbracket \sigma_0 : (i \mapsto 4) \rrbracket$   
**do**  $i < 0 \rightarrow$   
 $\llbracket \sigma_1^1 : \quad , \sigma_1^2 : \quad , \dots \rrbracket$   
 $i := i - 1$   
 $\llbracket \sigma_2^1 : \quad , \sigma_2^2 : \quad , \dots \rrbracket$   
**od**  
 $\llbracket \sigma_3 : \quad \quad \quad \rrbracket$

(k) **Var**  $i : Int$ ;  
 $\llbracket \sigma_0 : (i \mapsto 0) \rrbracket$   
**do**  $i \leq 0 \rightarrow$   
 $\llbracket \sigma_1^1 : \quad , \sigma_1^2 : \quad , \dots \rrbracket \star$   
 $i := i - 1$   
 $\llbracket \sigma_2^1 : \quad , \sigma_2^2 : \quad , \dots \rrbracket \star'$   
**od**  
 $\llbracket \sigma_3 : \quad \quad \quad \rrbracket$

(l) **Var**  $r : Int$ ;  
 $\llbracket \sigma_0 : (r \mapsto 3) \rrbracket$   
**do**  $r \neq 0 \rightarrow$   
 $\llbracket \quad \quad \quad \rrbracket$   
**if**  $r < 0 \rightarrow$   
 $\llbracket \quad \quad \quad \rrbracket$   
 $r := r + 1$   
 $\llbracket \quad \quad \quad \rrbracket$   
 $\square r > 0 \rightarrow$   
 $\llbracket \quad \quad \quad \rrbracket$   
 $r := r - 1$   
 $\llbracket \quad \quad \quad \rrbracket$   
**fi**  
 $\llbracket \quad \quad \quad \rrbracket$   
**od**  
 $\llbracket \quad \quad \quad \rrbracket$

2. Respondé a las siguientes preguntas respecto a los programas del ejercicio anterior:

- Considerá los programas de los ítems 1b, 1c y 1g. ¿Qué tienen en común? ¿Cuáles son las diferencias en la ejecución de cada uno de ellos?
- ¿Se puede escribir el programa del ítem 1c con sólo una asignación múltiple? Justificá.
- ¿Qué sucedería si en el ítem 1f se utilizaran las guardas “ $x > y$ ” y “ $x < y$ ” respectivamente?
- Con respecto al programa del ítem 1k; sin recurrir a enumerar todos los estados posibles ¿cómo caracterizarías los valores que toma la variable  $i$  cuando el programa se encuentra en los puntos de ejecución marcados por  $\star$  y  $\star'$ , independientemente de su valor inicial? ¿cuales son las ventajas? ¿las desventajas respecto a la enumeración de estados?

3. En cada uno de los siguientes programa, anotá los valores de las expresiones que se mencionan en la tabla respectiva, de acuerdo a cómo cambian los estados a medida que se ejecutan. Describí qué hace cada programa, en los términos más generales que encuentres.

(a) Const  $A : \text{array}[0, 4] \text{ of } \text{Int};$

Var  $i, s : \text{Int};$

$\llbracket \sigma_0 : (i \mapsto -3, s \mapsto 5, A \mapsto [2, 10, 10, -1]) \rrbracket$

$i, s := 0, 0; (\star)$

$\llbracket \sigma'_0 \rrbracket$

do  $i < 4 \rightarrow$

$\llbracket \sigma_1^0, \dots, \sigma_1^3 \rrbracket$

$s, i := s + A.i, i + 1$

$\llbracket \sigma_2^0, \dots, \sigma_2^3 \rrbracket$

od

$\llbracket \sigma_3 \rrbracket$

Estado	$i$	$s$
$\sigma_0$		
$\sigma'_0$		
$\sigma_1^0$		
$\sigma_2^0$		
$\sigma_1^1$		
$\sigma_2^1$		
$\sigma_1^2$		
$\sigma_2^2$		
$\sigma_1^3$		
$\sigma_2^3$		
$\sigma_3$		

(b) Const  $A : \text{array}[0, 4] \text{ of } \text{Int};$

Var  $i, c : \text{Int};$

$\llbracket \sigma_0 : (i \mapsto 3, c \mapsto 12, A \mapsto [12, -9, 10, -1]) \rrbracket$

$i, c := 0, 0;$

$\llbracket \sigma'_0 \rrbracket$

do  $i < 4 \rightarrow$

$\llbracket \sigma_1^0, \dots, \sigma_1^3 \rrbracket$

if  $A.i > 0 \rightarrow$

$c := c + 1$

□  $A.i \leq 0 \rightarrow$

skip

fi

$\llbracket \sigma_2^0, \dots, \sigma_2^3 \rrbracket$

$i := i + 1 (\star)$

$\llbracket \sigma_3^0, \dots, \sigma_3^3 \rrbracket$

od

$\llbracket \sigma_4 \rrbracket$

Estado	$i$	$A.i$	$c$
$\sigma_0$			
$\sigma'_0$			
$\sigma_1^0$			
$\sigma_2^0$			
$\sigma_3^0$			
$\sigma_1^1$			
$\sigma_2^1$			
$\sigma_3^1$			
$\sigma_1^2$			
$\sigma_2^2$			
$\sigma_3^2$			
$\sigma_1^3$			
$\sigma_2^3$			
$\sigma_3^3$			
$\sigma_4$			

4. Respondé las siguientes preguntas sobre los programas del ejercicio anterior:

- En el ítem 3a ¿qué sucede si eliminamos la línea de código marcada con  $\star$ ? Las asignaciones de este tipo tienen una función específica ¿cómo se las denomina?
- ¿Cómo modificarías el programa 3a para que calcule el promedio de los valores del arreglo  $A$ ?
- En el ítem 3b ¿qué sucede si movemos línea de código marcada con  $\star$  al comienzo del ciclo?
- ¿Cómo modificarías el programa 3b para que cuente los valores negativos que hay en el arreglo  $A$ ?
- ¿Cómo modificarías el programa del 3b para que sólo tenga en cuenta las posiciones pares del arreglo  $A$ ? ¿Y para que tenga en cuenta las posiciones impares?

5. Escribí un programa que calcule  $N!$ , donde  $N$  es una constante de tipo  $\text{Nat}$ .

6. Escribí dos programas que calculen el valor  $\langle \sum i : 0 \leq i \leq N : i \rangle$  (con  $N$  una constante de tipo  $\text{Nat}$ ) y lo almacenen en una variable  $r$ : uno debe usar un ciclo, y el otro basarse en la propiedad

$$\langle \sum i : 0 \leq i \leq N : i \rangle = N * (N + 1) / 2$$

Comparálos en términos de *eficiencia* y de *facilidad* para su comprensión.

7. Escribí un programa que cuente, por un lado, la cantidad de valores positivos, y por otro, la cantidad de valores negativos que ocurren en un arreglo  $A$  de tamaño  $N$ .
8. Escribí un programa que calcule *simultáneamente*, pero por separado, la sumatoria de los valores positivos y la sumatoria de los valores negativos de un arreglo  $A$  de tamaño  $N$ .
9. Decidí cuáles de las siguientes anotaciones son correctas y cuáles incorrectas.

- |   |  |  |
|---|--|--|
| <p>(a) <math>\text{Var } x : \text{Int};</math><br/> <math>\{x &gt; 0\}</math><br/> <math>x := x * x</math><br/> <math>\{\text{True}\}</math></p> | <p>(b) <math>\text{Var } x : \text{Int};</math><br/> <math>\{x \neq 100\}</math><br/> <math>x := x * x</math><br/> <math>\{x \geq 0\}</math></p> | <p>(c) <math>\text{Var } x : \text{Int};</math><br/> <math>\{x &gt; 0 \wedge x &lt; 100\}</math><br/> <math>x := x * x</math><br/> <math>\{x \geq 0\}</math></p> |
| <p>(d) <math>\text{Var } x : \text{Int};</math><br/> <math>\{x &gt; 0\}</math><br/> <math>x := x * x</math><br/> <math>\{x &lt; 0\}</math></p>    | <p>(e) <math>\text{Var } x : \text{Int};</math><br/> <math>\{x &lt; 0\}</math><br/> <math>x := x * x</math><br/> <math>\{x &lt; 0\}</math></p>   | <p>(f) <math>\text{Var } x : \text{Int};</math><br/> <math>\{\text{True}\}</math><br/> <math>x := x * x</math><br/> <math>\{x \geq 0\}</math></p>                |

10. Respondé las siguientes preguntas sobre el ejercicio anterior:

- (a) ¿Cuál es la utilidad de las anotaciones del ítem 9a?
- (b) De entre las anotaciones correctas ¿cuál tiene la precondition más fuerte? ¿cuál tiene la precondition más débil? ¿se te ocurre alguna otra precondition aún más débil?
- (c) ¿qué implica para un programa (cualquiera) tener “ $\{\text{True}\}$ ” como precondition? ¿y como postcondition?

11. Decidí si las siguientes anotaciones son correctas. En caso que no lo sean, corregilas. (Notar que el primer programa es el mismo que el del ejercicio 3, pero el segundo está levemente modificado.)

- |  |   |
|--|---|
| <p>(a) <math>\text{Const } A : \text{array}[0, 4) \text{ of Int};</math><br/> <math>\text{Var } i, s : \text{Int};</math><br/> <math>\{i = -3 \wedge s = 5\}</math><br/> <math>i, s := 0, 0;</math><br/> <math>\{i = -3 \wedge s = 5\}</math><br/> <b>do</b> <math>i &lt; 4 \rightarrow</math><br/> <math>\{0 \leq i &lt; 4 \wedge s = \langle \sum j : 0 \leq j &lt; i : A.j \rangle\}</math><br/> <math>s, i := s + A.i, i + 1</math><br/> <math>\{0 \leq i &lt; 4 \wedge s = \langle \sum j : 0 \leq j &lt; i : A.j \rangle\}</math><br/> <b>od</b><br/> <math>\{i = 3 \wedge s = \langle \sum j : 0 \leq j &lt; i : A.j \rangle\}</math></p> | <p>(b) <math>\text{Const } A : \text{array}[0, 4) \text{ of Int};</math><br/> <math>\text{Var } i, s : \text{Int};</math><br/> <math>\{i = 3 \wedge c = 12\}</math><br/> <math>i, c := 0, 0;</math><br/> <math>\{i = 0 \wedge c = 0\}</math><br/> <b>do</b> <math>i &lt; 4 \rightarrow</math><br/> <math>\{0 \leq i &lt; 4 \wedge c = 0\}</math><br/> <b>if</b> <math>A.i &gt; 0 \rightarrow</math><br/> <math>c, i := c + 1, i + 1</math><br/> <math>\square</math> <math>A.i \leq 0 \rightarrow</math><br/> <math>i := i + 1</math><br/> <b>fi</b><br/> <math>\{0 \leq i &lt; 4 \wedge c = \langle \text{N } j : 0 \leq j &lt; i : A.j &gt; 0 \rangle\}</math><br/> <b>od</b><br/> <math>\{i = 4 \wedge c = \langle \text{N } j : 0 \leq j &lt; i : A.j &gt; 0 \rangle\}</math></p> |
|--|---|

12. Respondé las siguientes preguntas sobre los programas anteriores:

- (a) ¿Por qué en las anotaciones no es necesario mencionar los valores del arreglo  $A$ ?
- (b) Compará las postcondiciones de cada programa con la descripción informal que realizaste en el ejercicio 3 ¿cómo se relacionan?