

# Algoritmos y Estructuras de Datos I - 1º cuatrimestre 2011

## Práctico 3: Programación imperativa

Docentes: Javier Blanco, Silvia Pelozo, Natalia Bidart, Demetrio Vilela, Walter Alini

El objetivo general de este práctico es afianzar los conceptos elementales de la programación imperativa. Por un lado, se pretende reforzar la noción de programa como **transformador de estados**, de manera intuitiva. Por otro lado, priorizando la interpretación de los programas como **transformadores de predicados**, se busca consolidar la noción de corrección de programa, a través de la verificación de programas simples. Además se presenta la noción de equivalencia y se busca reflexionar principalmente sobre las ventajas prácticas que estas nociones introducen a la hora de escribir y verificar programas.

1. ¿Cuál es el valor que se computa en  $r$ , en cada uno de los siguientes programas?

- |  |   |   |
|--|---|---|
| <p>(a) <math>x := y;</math><br/> <b>if</b> <math>x = y \rightarrow r := true</math><br/> <math>\square x \neq y \rightarrow r := false</math><br/> <b>fi</b></p> | <p>(b) <math>x := y;</math><br/> <b>if</b> <math>x \leq y \rightarrow r := true</math><br/> <math>\square x \geq y \rightarrow r := false</math><br/> <b>fi</b></p> | <p>(c) <math>x, r := 2 * y, true;</math><br/> <b>do</b> <math>x &gt; 0 \rightarrow x, r := x - 1, \neg r</math><br/> <b>od</b></p>                              |
| <p>(d) <b>do</b> <math>r &lt; 0 \rightarrow r + 1</math><br/> <math>\square r &gt; 0 \rightarrow r - 1</math><br/> <b>od</b></p>                                 | <p>(e) <math>x, r := 0, true;</math><br/> <b>do</b> <math>x &gt; 0 \wedge r \rightarrow r := false</math><br/> <b>od</b><br/> <math>r := \neg r</math></p>          | <p>(f) <math>x, r := 2 * y, true;</math><br/> <b>do</b> <math>x &gt; 0 \rightarrow x, r := x + 1, \neg r</math><br/> <b>od</b><br/> <math>r := false</math></p> |

2. a) Encontrá (informalmente) un programa imperativo que resuelva cada una de las siguientes especificaciones descriptas en lenguaje natural:

- 1) Dado un valor inicial positivo  $X$ , la variable  $r$  es *true* si  $X$  es par, y *false* en caso contrario.
- 2) Dado un valor inicial positivo  $X$ , la variable  $r$  almacena la suma de los primeros  $X$  números pares.
- 3) Dado un valor inicial positivo  $X$ , el valor de  $r$  determina si  $X$  es un cuadrado perfecto.
- 4) Dado un valor inicial positivo  $X$ , el valor de  $r$  determina si  $X$  es un número primo.
- 5) Para valores iniciales enteros  $X$  e  $Y$ , en  $r$  se almacena el resultado de multiplicar  $X$  por  $Y$  utilizando sólo sumas.
- 6) Dado un valor inicial positivo  $X$ ,  $r$  es el resultado de multiplicar los dígitos de  $X$ .

b) Pasá cada una de estas funciones a C, tomando los valores por pantalla e imprimiendo los resultados en la salida estándar. La traducción de los programas tiene que ser **directa**, es decir, el programa en C tiene que ser el fiel reflejo de lo que escribiste en papel.

3. *Calculá* una precondition  $P$  de modo que sean correctos los siguientes programas anotados, suponiendo 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)  $\{P\} a := a \equiv b; b := a \equiv b; a := a \equiv b \{(a \equiv B) \wedge (b \equiv A)\}$

4. Calculá las expresiones  $E$  y  $F$  (intuitivamente) más simples de modo que las siguientes ternas de Hoare sean correctas:

- a)  $\{A = q * B + r\} q := E; r := r - B \{A = q * B + r\}$   
 b)  $\{x * y + p * q = N\} x := x - p; q := F \{x * y + p * q = N\}$

5. Considerá los siguientes programas que intercambian los valores de dos variables  $x$  e  $y$  de tipo  $Int$ :

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

- a) Especificá la pre y postcondición, y verificá los tres programas.  
 b) Decimos que dos programas  $S$  y  $T$  son *equivalentes*, denotado por  $S \simeq T$ , si y solo si  $wp.S.Q \equiv wp.T.Q$  para cualquier predicado  $Q$ . Demostrá que el primer programa es equivalente al tercero, valiéndote de las siguientes propiedades de sustitución sintáctica en predicados:
- $(Q(x := E))(x := F) \equiv Q(x := E(x := F))$   
 donde  $x$  es una (lista de) variable(s), y  $E$  y  $F$  son (listas de) expresiones bien definidas.
  - $Q(x := E) \equiv Q(x, y := E, y)$   
 donde  $x$  e  $y$  son variables distintas.
- c) ¿Es el segundo programa equivalente a los demás? En caso negativo, ¿como podemos relajar la definición de equivalencia, de modo que sea satisfecha por este programa respecto a cualquiera de los otros dos?

6. Demostrá las siguientes equivalencias entre programas:

- a)  $x := x \simeq \mathbf{skip}$   
 b)  $S; \mathbf{skip} \simeq S$  y simétricamente  $\mathbf{skip}; S \simeq S$   
 c)  $S; \mathbf{abort} \simeq \mathbf{abort}$  y simétricamente  $\mathbf{abort}; S \simeq \mathbf{abort}$   
 d)  $(S; T); U \simeq S; (T; U)$

7. Usando las propiedades del transformador de predicados *weakest precondition*, demostrá las siguientes propiedades:

- a)  $\{P\} S \{Q\} \wedge [P_0 \Rightarrow P] \Rightarrow \{P_0\} S \{Q\}$   
 b)  $\{P\} S \{Q\} \wedge [Q \Rightarrow Q_0] \Rightarrow \{P\} S \{Q_0\}$   
 c)  $\{P\} S \{Q\} \wedge \{P\} S \{R\} \equiv \{P\} S \{Q \wedge R\}$   
 d)  $\{P\} S \{Q\} \wedge \{R\} S \{Q\} \equiv \{P \vee R\} S \{Q\}$

8. 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 demostralo, en caso negativo dá un contraejemplo (instanciando los programas y las guardas).

- a) i)  $x := E;$   
 $y := F;$   
 ii)  $y := F;$   
 $x := E;$   
 b) i)  $\mathbf{if} B_0 \rightarrow S$   
 $\square B_1 \rightarrow S$   
 $\mathbf{fi}$   
 ii)  $S$

c) i)  $\begin{array}{l} \mathbf{if} B_0 \rightarrow S; S_0; T \\ \square B_1 \rightarrow S; S_1; T \\ \mathbf{fi} \end{array}$       ii)  $\begin{array}{l} \mathbf{if} B_0 \rightarrow S; S_0 \\ \square B_1 \rightarrow S; S_1 \\ \mathbf{fi}; \\ T \end{array}$       iii)  $\begin{array}{l} S; \\ \mathbf{if} B_0 \rightarrow S_0; T \\ \square B_1 \rightarrow S_1; T \\ \mathbf{fi} \end{array}$

9. Para los siguientes programas, donde las variables  $x, y$  son de tipo  $Int$ , y  $a, b$  de tipo  $Bool$ :

a)  $\begin{array}{l} \{true\} \\ \mathbf{if} x \geq 1 \rightarrow x := x + 1 \\ \square x \leq 1 \rightarrow x := x - 1 \\ \mathbf{fi} \\ \{x \neq 1\} \end{array}$       b)  $\begin{array}{l} \{true\} \\ \mathbf{if} x \geq y \rightarrow \mathbf{skip} \\ \square x \leq y \rightarrow x, y := y, x \\ \mathbf{fi} \\ \{x \geq y\} \end{array}$

c)  $\begin{array}{l} \{true\} \\ x, y := y * x, x * y; \\ \mathbf{if} x \geq y \rightarrow x := x - y \\ \square x \leq y \rightarrow y := y - x \\ \mathbf{fi} \\ \{x \geq 0 \wedge y \geq 0\} \end{array}$       d)  $\begin{array}{l} \{true\} \\ \mathbf{if} \neg a \vee b \rightarrow a := \neg a \\ \square a \vee \neg b \rightarrow b := \neg b \\ \mathbf{fi} \\ \{a \vee b\} \end{array}$

e)  $\begin{array}{l} \{N \geq 0\} \\ x := 0; \\ \mathbf{do} x \neq N \rightarrow x := x + 1 \\ \mathbf{od} \\ \{x = N\} \end{array}$       f)  $\begin{array}{l} \{N \geq 0\} \\ x, y := 0, 0; \\ \mathbf{do} x \neq 0 \rightarrow x := x - 1 \\ \square y \neq N \rightarrow x, y := N, y + 1 \\ \mathbf{od} \\ \{x = 0 \wedge y = N\} \end{array}$

- Demostrá que los siguientes programas anotados son correctos.
- Pasá cada una de estas funciones a C, tomando los valores por pantalla e imprimiendo los resultados en la salida estándar. La traducción de los programas tiene que ser **directa**, es decir, el programa en C tiene que ser el fiel reflejo de lo que escribiste en papel.

10. Considerá los siguientes programas anotados, donde la variable  $x$  es de tipo  $Int$ :

i)  $\begin{array}{l} \{P\} \\ \mathbf{do} x \neq 0 \rightarrow x := x - 1 \\ \mathbf{od} \\ \{x = 0\} \end{array}$       ii)  $\begin{array}{l} \{P\} \\ \mathbf{do} x \neq 0 \rightarrow x := x - 2 \\ \mathbf{od} \\ \{x = 0\} \end{array}$

- a) Determiná en cada caso una precondition  $P$ , la más débil que encuentres, de manera que se satisfaga la corrección de las anotaciones.
- b) El predicado  $P$  encontrado, ¿es la *precondición más débil*?
- c) Pasá cada una de estas funciones a C, tomando el/los valore/s que hagan falta por pantalla, e imprimiendo los resultados en la salida estándar. La traducción de los programas tiene que ser **directa**, es decir, el programa en C tiene que ser el fiel reflejo de lo que escribiste en papel.