

Algoritmos y Estructuras de Datos I - 2° cuatrimestre 2010

Práctico 4: Programación imperativa

Docentes: Javier Blanco, Mariana Badano, Renato Cherini, Mauricio Tellechea, Santiago Videla, Demetrio Vilela

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, a través de los ejercicios 1 y 2. 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 (ejercicios 3, 4, 5, 9 y 10). Además se presenta las noción de equivalencia y se busca reflexionar principalmente sobre las ventajas prácticas que estas nociones introducen a la hora de escribir y verificar programas (ejercicios 6, 7, 8).

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

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

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

- a) Dado un valor inicial positivo X , la variable r es *true* si X es par, y *false* en caso contrario.
- b) Dado un valor inicial positivo X , la variable r almacena la suma de los primeros X números pares.
- c) Dado un valor inicial positivo X , el valor de r determina si X es un cuadrado perfecto.
- d) Dado un valor inicial positivo X , el valor de r determina si X es un número primo.
- e) Dado dos pares de valores iniciales N_1, D_1 y N_2, D_2 que representan los números racionales $\frac{N_1}{D_1}$ y $\frac{N_2}{D_2}$ respectivamente, el valor de r_n y r_d representa el producto de dichas fracciones como la fracción $\frac{r_n}{r_d}$. El resultado debe estar simplificado tanto como sea posible.
- f) Para valores iniciales enteros X e Y , en r se almacena el resultado de multiplicar X por Y utilizando sólo sumas.
- g) Dado valores iniciales positivos *PESOS* y *CENTAVOS* que representan una suma de dinero, las variables $r1, r2, r3$ y $r4$ almacenan valores que representan la forma de dar cambio de la suma inicial en monedas de 50, 25, 10 y 5 centavos respectivamente. Además r almacena el (posible) resto que no puede devolverse.
- h) Dado un valor inicial positivo X , r es el resultado de multiplicar los dígitos de X .

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. a) Demostrá las siguientes equivalencias entre programas:

- 1) $x := x \simeq \mathbf{skip}$
- 2) $S; \mathbf{skip} \simeq S$ y simétricamente $\mathbf{skip}; S \simeq S$
- 3) $S; \mathbf{abort} \simeq \mathbf{abort}$ y simétricamente $\mathbf{abort}; S \simeq \mathbf{abort}$
- 4) $(S; T); U \simeq S; (T; U)$

b) Pensando a $;$ como un operador binario, y haciendo analogía con las propiedades del cálculo proposicional, ¿qué nombres podríamos darle a las propiedades (2), (3) y (4)?

7. a) Usando las propiedades del transformador de predicados *weakest precondition*, demostrá 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?

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) **if** $B_0 \rightarrow S$
 $\square B_1 \rightarrow S$
fi ii) S
- c) i) **if** $B_0 \rightarrow S; S_0; T$
 $\square B_1 \rightarrow S; S_1; T$
fi ii) **if** $B_0 \rightarrow S; S_0$
 $\square B_1 \rightarrow S; S_1$
fi;
 T iii) $S;$
if $B_0 \rightarrow S_0; T$
 $\square B_1 \rightarrow S_1; T$
fi

9. 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 $x \geq y \rightarrow \text{skip}$
 $\square x \leq y \rightarrow x, y := y, x$
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) $\{N \geq 0\}$
 $x, y := 0, 0;$
do $x \neq 0 \rightarrow x := x - 1$
 $\square y \neq N \rightarrow x, y := N, y + 1$
od
 $\{x = 0 \wedge y = N\}$

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

- i) $\{P\}$
do $x \neq 0 \rightarrow x := x - 1$
od
 $\{x = 0\}$
- ii) $\{P\}$
do $x \neq 0 \rightarrow x := x - 2$
od
 $\{x = 0\}$

- 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 precondition más débil?