Práctico 3: Introducción a la Programación Imperativa

Algoritmos y Estructuras de Datos I
2$^{\text{do}}$ cuatrimestre 2023

Esta guía tiene como objetivo 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 ***anotación de programa****, y en particular de* ***precondición*** *y* ***postcondición****. Además se presenta las noción de corrección de programas anotados.*

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

a) Var $x : Int$;
$[\![\sigma_0 : (x \mapsto 1)]\!]$
$x := 5$
$[\![\sigma_1 :\quad\quad]\!]$

b) Var $x, y : Int$;
$[\![\sigma_0 : (x \mapsto 2, y \mapsto 5)]\!]$
$x := x + y;$
$[\![\sigma_1 :\quad\quad]\!]$
$y := y + y$
$[\![\sigma_2 :\quad\quad]\!]$

c) Var $x, y : Int$;
$[\![\sigma_0 : (x \mapsto 2, y \mapsto 5)]\!]$
$y := y + y;$
$[\![\sigma_1 :\quad\quad]\!]$
$x := x + y$
$[\![\sigma_2 :\quad\quad]\!]$

d) Var $x, y : Int$;
$[\![\sigma_0 : (x \mapsto 2, y \mapsto 5)]\!]$
$y, x := y + y, x + y$
$[\![\sigma_1 :\quad\quad]\!]$

e) Var $x, y : Int$;
$[\![\sigma_0 : (x \mapsto 3, y \mapsto 1)]\!]$
if $x \geq y \rightarrow$
  $[\![\sigma_1 :\quad\quad]\!]$
  $x := 0$
  $[\![\sigma_2 :\quad\quad]\!]$
□ $x \leq y \rightarrow$
  $[\![\sigma'_1 :\quad\quad]\!]$
  $x := 2$
  $[\![\sigma'_2 :\quad\quad]\!]$
fi
$[\![\sigma_3 :\quad\quad]\!]$

f) Var $x, y : Int$;
$[\![\sigma_0 : (x \mapsto -100, y \mapsto 1)]\!]$
if $x \geq y \rightarrow$
  $[\![\sigma_1 :\quad\quad]\!]$
  $x := 0$
  $[\![\sigma_2 :\quad\quad]\!]$
□ $x \leq y \rightarrow$
  $[\![\sigma'_1 :\quad\quad]\!]$
  $x := 2$
  $[\![\sigma'_2 :\quad\quad]\!]$
fi
$[\![\ \sigma'_3 :\quad\quad]\!]$

g) Var $x, y : Int$;
$[\![\sigma_0 : (x \mapsto 1, y \mapsto 1)]\!]$
if $x \geq y \rightarrow$
  $[\![\sigma_1 :\quad\quad]\!]$
  $x := 0$
  $[\![\sigma_2 :\quad\quad]\!]$
□ $x \leq y \rightarrow$
  $[\![\sigma'_1 :\quad\quad]\!]$
  $x := 2$
  $[\![\sigma'_2 :\quad\quad]\!]$
fi
$[\![\sigma_3 :\quad\quad,\ \sigma'_3 :\quad\quad]\!]$

h) Var $i : Int$;
$[\![\sigma_0 : (i \mapsto 4)]\!]$
do $i \neq 0 \rightarrow$
  $[\![\sigma_1^1 :\quad, \sigma_1^2 :\quad, \cdots]\!]$
  $i := i - 1$
  $[\![\sigma_2^1 :\quad, \sigma_2^2 :\quad, \cdots]\!]$
od
$[\![\sigma_3 :\quad\quad]\!]$

i) Var $i : Int$;
$[\![\sigma_0 : (i \mapsto 400)]\!]$
do $i \neq 0 \rightarrow$
  $[\![\sigma_1^1 :\quad, \sigma_1^2 :\quad, \cdots]\!]$
  $i := 0$
  $[\![\sigma_2^1 :\quad, \sigma_2^2 :\quad, \cdots]\!]$
od
$[\![\sigma_3 :\quad\quad]\!]$

j) Var $i : Int$;
$[\![\sigma_0 : (i \mapsto 4)]\!]$
do $i < 0 \to$
$[\![\sigma_1^1 : \quad , \sigma_1^2 : \quad , \cdots]\!]$
$i := i - 1$
$[\![\sigma_2^1 : \quad , \sigma_2^2 : \quad , \cdots]\!]$
od
$[\![\sigma_3 : \quad]\!]$

k) Var $i : Int$;
$[\![\sigma_0 : (i \mapsto 0)]\!]$
do $i \leq 0 \to$
$[\![\sigma_1^1 : \quad , \sigma_1^2 : \quad , \cdots]\!]$ ★
$i := i - 1$
$[\![\sigma_2^1 : \quad , \sigma_2^2 : \quad , \cdots]\!]$ ★′
od
$[\![\sigma_3 : \quad]\!]$

l) Var $r : Int$;
$[\![\sigma_0 : (r \mapsto 3)]\!]$
do $r \neq 0 \to$
$[\![\quad]\!]$
if $r < 0 \to$
$[\![\quad]\!]$
$r := r + 1$
$[\![\quad]\!]$
▯ $r > 0 \to$
$[\![\quad]\!]$
$r := r - 1$
$[\![\quad]\!]$
fi
$[\![\quad]\!]$
od
$[\![\quad]\!]$

2. Responda las siguientes preguntas respecto al ejercicio anterior:

   a) ¿Qué se puede concluir de los ítems 1*b*, 1*c* y 1*d*? ¿Qué tienen en común? ¿Cuáles son las diferencias en la ejecución de cada uno de ellos?

   b) ¿Se puede escribir un programa equivalente al del ítem 1*c* con sólo una asignación múltiple? Justifique.

   c) ¿Qué sucedería si en el ítem 1*g* se utilizaran las guardas "$x > y$" y "$x < y$"?

   d) Con respecto al programa del ítem 1*k*: ¿Existen predicados que caractericen exactamente todos los valores que puede tomar la variable i cuando el mismo se encuentra en ★ y en ★′? Si la respuesta es sí, escríbalos. Es más, si existen, ¿cuál es la ventaja con respecto a enumerar todos los estados? ¿La desventaja?

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

   a) Const $A : array[0, 4)\ of Int$;
   Var $i, s : Int$;
   $[\![\sigma_0 : (i \mapsto -3, s \mapsto 5, A \mapsto$ | 2 | 10 | 10 | −1 | $)]\!]$
   $i, s := 0, 0$;
   $[\![\sigma_0']\!]$
   do $i < 4 \to$
   $[\![\sigma_1^0, \cdots, \sigma_1^3]\!]$
   $s, i := s + A.i, i + 1$
   $[\![\sigma_2^0, \cdots, \sigma_2^3]\!]$
   od
   $[\![\sigma_3]\!]$

Estado	i	s
σ_0		
σ_0'		
σ_1^0		
σ_2^0		
σ_1^1		
σ_2^1		
σ_1^2		
σ_2^2		
σ_1^3		
σ_2^3		
σ_3		

b) Const $A : array[0, 4)\ of Int$;
Var $i, c : Int$;
$[\![\sigma_0 : (i \mapsto 3, c \mapsto 12, A \mapsto \boxed{12}\boxed{-9}\boxed{10}\boxed{-1})]\!]$
$i, c := 0, 0$;
$[\![\sigma'_0]\!]$
do $i < 4 \rightarrow$
$[\![\sigma^0_1, \cdots, \sigma^3_1]\!]$
if $A.i > 0 \rightarrow$
$c := c + 1$
$\square$ $A.i \leq 0 \rightarrow$
skip
fi
$[\![\sigma^0_2, \cdots, \sigma^3_2]\!]$
$i := i + 1$
$[\![\sigma^0_3, \cdots, \sigma^3_3]\!]$
od
$[\![\sigma_4]\!]$

Estado	i	$A.i$	c
σ_0			
σ'_0			
σ^0_1			
σ^0_2			
σ^0_3			
σ^1_1			
σ^1_2			
σ^1_3			
σ^2_1			
σ^2_2			
σ^2_3			
σ^3_1			
σ^3_2			
σ^3_3			
σ_4			

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

   a) ¿Cómo modificaría el programa del ítem 3*a* para que calcule el promedio de los valores en el arreglo A?

   b) ¿Cómo modificaría el programa del ítem 3*b* para que solo tenga en cuenta las posiciones pares del arreglo A?

5. *a*) Escriba un programa que calcule $N!$, a donde $N \geq 0$ es una constante de tipo *Int*.

   b) Escriba otro programa que calcule $N!$ efectuando las multiplicaciones en un orden diferente.

   c) Para $N = 5$, ejecute manualmente ambos programas y escriba las tablas de estados.

6. *a*) Escriba un programa que sume, por un lado, los valores positivos y, por otro, los valores múltiplos de 3 de un arreglo A de tamaño N.

   b) Para $N = 5, A = \boxed{12}\boxed{-9}\boxed{10}\boxed{0}\boxed{-1}$, ejecute manualmente el programa y escriba la tabla de estados.

7. *a*) Escriba un programa que, dados dos arreglos A y B de tamaño N y M respectivamente, calcule cuántas veces coinciden dos elementos entre ellos.

   b) Para $A = \boxed{8}\boxed{0}\boxed{8}$, $B = \boxed{0}\boxed{8}$, ejecute manualmente el programa y escriba la tabla de estados.

Ternas de Hoare y Weakest Precondition

8. Decidir si los siguientes programas anotados como ternas de Hoare son correctos (equivalentes a *True*) o incorrectos (equivalentes a *False*).

   a) Var $x : Int$;
   $\{x > 0\}$
   $x := x * x$
   $\{True\}$

   b) Var $x : Int$;
   $\{x \neq 100\}$
   $x := x * x$
   $\{x \geq 0\}$

   c) Var $x : Int$;
   $\{x > 0 \wedge x < 100\}$
   $x := x * x$
   $\{x \geq 0\}$

   d) Var $x : Int$;
   $\{x > 0\}$
   $x := x * x$
   $\{x < 0\}$

   e) Var $x : Int$;
   $\{x < 0\}$
   $x := x * x$
   $\{x < 0\}$

   f) Var $x : Int$;
   $\{True\}$
   $x := x * x$
   $\{x \geq 0\}$

9. Responda las siguientes preguntas en función del ejercicio anterior:

   a) ¿Es útil la anotación del programa (8*a*)?¿Por qué?

b) Descartando los programas incorrectos, ¿Cual programa tiene la precondición más débil? ¿Cual tiene la precondición más fuerte? Explíquelo con sus propias palabras.

c) Con respecto a la última pregunta, ¿se puede definir alguna otra precondición tal que esta sea aún más débil?

d) En general, ¿qué implica tener un "$\{True\}$" al principio de un programa? ¿Al final?

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

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

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

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

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

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

f) Var $x, y : Num$;
$\{\quad\}$
if $x \geq y \rightarrow$
$\{\quad\}$
$x := 0$
$\{\quad\}$
▯ $x \leq y \rightarrow$
$\{\quad\}$
$x := 2$
$\{\quad\}$
fi
$\{(x = 0 \lor x = 2) \land y = 1\}$

11. 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 $x \geq 1 \rightarrow x := x + 1$
▯ $x \leq 1 \rightarrow x := x - 1$
fi
$\{x \neq 1\}$

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

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

d) $\{True\}$
if $\neg a \lor b \rightarrow a := \neg a$
▯ $a \lor \neg b \rightarrow b := \neg b$
fi
$\{a \lor b\}$

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 $r < 0 \rightarrow r := r + 1$
▯ $r > 0 \rightarrow r := r - 1$
fi
od
$\{r = 0\}$

12. Analice los siguientes programas anotados. En cada caso, describa en lenguaje natural la postcondición, y decida si el programa efectivamente valida las anotaciones. No es necesario hacer las demostraciones.

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

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

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

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 $i \neq N \wedge \neg r \rightarrow$
$\quad$ **if** $A.i = E \rightarrow r := True$
$\quad\ \square\ A.i \neq E \rightarrow$ **skip**
$\quad$ **fi ;**
$\quad i := i + 1$
od
$\{\langle \exists k\ : 0 \leq k < N :\ A.k = E \rangle \Rightarrow A.i = E\}$

Ejercicios extra

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

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

a) $\{P\}$	*b*) $\{P\}$
if $B_0 \rightarrow S_0$	**if** $B_0 \rightarrow S_0$
$\square\ B_1 \rightarrow S_1$	$\square\ \neg B_0 \rightarrow S_1$
fi	**fi**
$\{Q\}$	$\{Q\}$

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

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

$x, y := y, x$	$z := x;$ $x := y;$ $y := z$	$x := x - y;$ $y := x + y;$ $x := y - x$

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

16. 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** $B_0 \to S$ $\square$ $B_1 \to S$ **fi** — *ii*) S

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