Práctico 3: Introducción a la Programación Imperativa

Algoritmos y Estructuras de Datos I
1er cuatrimestre 2017

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

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

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

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

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

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

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

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
$[\![\sigma_3 : \quad]\!]$

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
$[\![\sigma_3 : \quad]\!]$

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? Si se puede, escriba el programa. Si no, explique por qué. A todo esto, ¿cuándo dos programas son equivalentes?

   c) 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 $A.i > 0 \rightarrow$
$c := c + 1$
▯ $A.i \leq 0 \rightarrow$
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) En el ítem 3*a* ¿Qué pasa si la linea de código (★) es eliminada? Esa asignación tiene una función específica, por lo cual recibe un nombre particular ¿Cuál es ese nombre?

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

   c) En el ítem 3*b* ¿Qué pasa si la linea de código (★) es movida al principio del cuerpo del ciclo?

   d) ¿Cómo modificaría el programa del ítem 3*b* para que cuente los valores negativos que hay en el arreglo A?

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

5. *a*) Escriba dos programas distintos que calculen

$$\langle \sum i : 1 \leq i \leq N : i \rangle,$$

   a donde $N \geq 0$ es una constante de tipo *Int*. Uno debe usar un ciclo, y el otro debe basarse en la propiedad matemática $\langle \sum i : 1 \leq i \leq N : i \rangle = N * (N + 1)/2$.

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

6. *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 operaciones en un orden diferente.

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

7. *a*) Escriba 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 $N \geq 0$.

   b) Escriba otro programa que haga lo mismo pero recorriendo el arreglo de una manera diferente.

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

8. *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.

9. *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.

10. Decida cuáles de las siguientes anotaciones son correctas y cuáles incorrectas:

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

11. Responda las siguientes preguntas sobre el ejercicio anterior:

a) ¿Cuál es la utilidad de las anotaciones del programa 10*a*?

b) De entre las anotaciones correctas ¿cuál tiene la precondición más fuerte? ¿cuál tiene la precondición más débil? ¿se te ocurre alguna otra precondición aún más débil?

c) En general, ¿qué implica tener "$\{True\}$" como precondición? ¿y como postcondición?

12. Decida si las siguientes anotaciones son correctas. En caso de que no lo sean, corríjalas. (Notar que los programas son los del ejercicio 3, pero el segundo está levemente modificado.)

a) Const $A : array[0, 4)\ of Int$;
Var $i, s : Int$;
$\{i = -3 \wedge s = 5\}$
$i, s := 0, 0;$
$\{i = -3 \wedge s = 5\}$
do $i < 4 \rightarrow$
$\{0 \leq i < 4 \wedge s = \langle \sum j \ : \ 0 \leq j < i \ : \ A.j \rangle\}$
$s, i := s + A.i, i + 1$
$\{0 \leq i < 4 \wedge s = \langle \sum j \ : \ 0 \leq j < i \ : \ A.j \rangle\}$
od
$\{i = 3 \wedge s = \langle \sum j \ : \ 0 \leq j < i \ : \ A.j \rangle\}$

b) Const $A : array[0, 4)\ of Int$;
Var $i, s : Int$;
$\{i = 3 \wedge c = 12\}$
$i, c := 0, 0;$
$\{i = 0 \wedge c = 0\}$
do $i < 4 \rightarrow$
$\{0 \leq i < 4 \wedge c = 0\}$
if $A.i > 0 \rightarrow$
$c, i := c + 1, i + 1$
▯ $A.i \leq 0 \rightarrow$
$i := i + 1$
fi
$\{0 \leq i < 4 \wedge c = \langle \mathrm{N}\ j \ : \ 0 \leq j < i \ : \ A.j > 0 \rangle\}$
od
$\{i = 4 \wedge c = \langle \mathrm{N}\ j \ : \ 0 \leq j < i \ : \ A.j > 0 \rangle\}$

13. Responda las siguientes preguntas sobre el ejercicio anterior:

a) ¿Por qué en las anotaciones no es necesario mencionar los valores del arreglo A?

b) Compare las anotaciones finales de cada programa con la explicación informal que realizó en el Ejercicio 3. ¿Las mismas se relacionan?