Práctico 3
Introducción a la programación imperativa

Algoritmos y Estructuras de Datos I

1. Anote en cada programa, los valores que las variables van tomando a medida que este se ejecuta.

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

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

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

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

(h) Var $i : Int$;
$[\![\sigma_0 : (i \mapsto 4)]\!]$
do $i \neq 0 \rightarrow$
  $[\![\sigma^1_1 : \quad , \sigma^2_1 : \quad , \cdots]\!]$
  $i := i - 1$
  $[\![\sigma^1_2 : \quad , \sigma^2_2 : \quad , \cdots]\!]$
od
$[\![\sigma_3 : \quad]\!]$

(i) Var $i : Int$;
$[\![\sigma_0 : (i \mapsto 400)]\!]$
do $i \neq 0 \rightarrow$
  $[\![\sigma^1_1 : \quad , \sigma^2_1 : \quad , \cdots]\!]$
  $i := 0$
  $[\![\sigma^1_2 : \quad , \sigma^2_2 : \quad , \cdots]\!]$
od
$[\![\sigma_3 : \quad]\!]$

(j) Var $i : Int$;
$[\![\sigma_0 : (i \mapsto 4)]\!]$
do $i < 0 \rightarrow$
  $[\![\sigma^1_1 : \quad , \sigma^2_1 : \quad , \cdots]\!]$
  $i := i - 1$
  $[\![\sigma^1_2 : \quad , \sigma^2_2 : \quad , \cdots]\!]$
od
$[\![\sigma_3 : \quad]\!]$

(k) Var $i : Int$;
$[\![\sigma_0 : (i \mapsto 0)]\!]$
do $i \leq 0 \rightarrow$
  $[\![\sigma^1_1 : \quad , \sigma^2_1 : \quad , \cdots]\!]$ ★
  $i := i - 1$
  $[\![\sigma^1_2 : \quad , \sigma^2_2 : \quad , \cdots]\!]$ ★′
od
$[\![\sigma_3 : \quad]\!]$

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

2. Responda las siguientes preguntas con respecto al ejercicio anterior

   (a) ¿Qué se puede concluir de los ejercicios (1b) y (1c)?

   (b) ¿Se puede escribir el programa del ejercicio (1c) con sólo una asignación múltiple? Si sí, escriba el programa. Si no, explique por qué... A todo esto, ¿cuándo dos programas son equivalentes?

   (c) ¿Qué pasa si en el ejercicio (1g) se cambian las guardas del if por "$x > y$" y "$x < y$", respectivamente?

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

3. Para cada programa, anote los valores que las variables van tomando a medida que éste se ejecuta (es decir, describa los estados) utilizando la tabla que se le brinda. Explique qué hace cada programa.

   (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 $i < 4 \rightarrow$
   $[\![\sigma^0_1, \cdots, \sigma^3_1]\!]$
   $s, i := s + A.i, i + 1$
   $[\![\sigma^0_2, \cdots, \sigma^3_2]\!]$
   od
   $[\![\sigma_3]\!]$

Estado	Var. i	Var. s
σ_0		
σ'_0		
σ^0_1		
σ^0_2		
σ^1_1		
σ^1_2		
σ^2_1		
σ^2_2		
σ^3_1		
σ^3_2		
σ_3		

   (b) Const $A : array[0, 4)\ of Int$;
   Var $i, c : Int$;
   $[\![\sigma_0 : (i \mapsto 3, c \mapsto 12, A \mapsto [12, -9, 10, -1])]\!]$
   $i, c := 0, 0$;
   $[\![\sigma'_0]\!]$
   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	Var. i	Val.(or) $A.i$	Var. 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:

   (a) En el ejercicio (3a) ¿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 modificaria el programa del ejercicio (3a) para que calcule el promedio de los valores en el array A?

   (c) En el ejercicio (3a) ¿Qué pasa con el valor $A.i$ en el estado σ_0?¿Por que esto no es un problema?

   (d) En el ejercicio (3b) ¿Qué pasa si la linea de código (★) es movida al principio del loop/ciclo/do?

   (e) ¿Cómo modificaria el programa del ejercicio (3b) para que cuente los valores negativos que hay en el array A?

   (f) ¿Cómo modificaría el programa del ejercicio (3b) para que solo tenga en cuenta las posiciones pares del array A? ¿Y para que tenga en cuenta las posiciones impares?

5. Escriba un programa que calcule $N!$, con N una constante de tipo *Nat*.

6. Escriba dos programas distintos que calculen $\sum_{i=1}^{N} i$, con N una constante de tipo *Nat*. Uno debe usar un ciclo, el otro no (Recordar la identidad: $\sum_{i=1}^{N} i = n * (n + 1)/2$). ¿Cual programa es más eficiente? ¿Cual programa es el que requiere menos conocimientos para entenderlo? *Moraleja: a un mismo resultado se puede llegar de muchas maneras diferentes y cada una de esas maneras puede presentar ventajas y desventajas.*

7. Escriba un programa que cuente la cantidad de positivos por un lado y la cantidad de negativos por otro de un arreglo constante de enteros A de tamaño N.

8. Escriba un programa que sume por un lado los valores positivos y por otro lado los valores negativos de un arreglo constante de enteros A de tamaño N.

9. Modifique el programa anterior para que calcule el promedio de cada suma.

10. Decidir si los siguientes programas anotados 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 $x : Int$;
    $\{True\}$
    $x := x * x$
    $\{x \geq 0\}$

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

    (a) ¿Es útil la anotación del programa (10a)?¿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? (Ver programas (10a) y (10f) como ejemplo de esto).

12. Decida si las siguientes anotaciones son correctas. En caso de que no lo sean, corríjalas. (Notar que el primer programa es exactamente el primer progama que aparece en el ejercicio 3. En cambio el segundo está levemente modificado con respecto al que aparece en el mismo ejercicio.)

    (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 $A.i > 0 \rightarrow$
    $c, i := c + 1, i + 1$
    $\square$ $A.i \leq 0 \rightarrow$
    $i := i + 1$
    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 en función del ejercicio anterior:

    (a) ¿Por qué en las anotaciones no es necesarion mencionar los valores del array A?

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