Universdad Nacional Autónoma de México

Lenguajes de Programacion
Examen Parcial IV

- Edgar Montiel Ledesma
  317317794
- Carlos Daniel Cortes Jimenez
  420004846

---

1. Considera el siguiente programa en el lenguaje While :

   new z = 0;
   while (y < x + 1) do
   (z := z + 1;
   x := x − y)
   end

   a) Ejecuta el programa en el estado en el que σ(x) = 17 y σ(y) = 5 ¿Cual es el estado resultante de la evaluacion?
   Inicialmente, tenemos σ(x) = 17 y σ(y) = 5. Vamos a ejecutar el programa paso a paso:

   Iteración 1:
   - z := 0
   - y < x + 1 es verdadero ya que 5 < 17 + 1
   - z := z + 1; z=1
   - x := x - y; x=12

   Iteración 2:
   - y < x + 1 es verdadero ya que 5 < 12 + 1
   - z := z + 1; z=2
   - x := x - y; x=7

   Iteración 3:
   - y ¡ x + 1 es verdadero ya que 5 ¡ 7 + 1
   - z := z + 1; z=3
   - x := x - y; x=2

   Iteración 4:
   - y ¡ x + 1 es verdadero ya que 5 ¡ 2 + 1
   - z := z + 1; z=4
   - x := x - y; x=-3

Iteración 5:

- y ¡ x + 1 es falso ya que 5 ¿= -3 + 1

El estado resultante es σ(x) = -3 y σ(y) = 5.

b) Da un estado σ tal que si se evalua el programa anterior con dicho estado la evaluacion se ciclaria infinitamente.

Para que el programa entre en un bucle infinito, necesitamos un estado en el que la condición del bucle siempre sea verdadera. La condición del bucle es $y < x + 1$. Por lo tanto, si tenemos y igual a cualquier número mayor que x+1, el bucle se ejecutará infinitamente.

Un ejemplo de un estado que causaría una ejecución infinita podría ser σ(x) = 5 y σ(y) = 10. En este caso, la condición $y < x+1$ siempre será verdadera ($10 < 5+1$), y el bucle se ejecutará sin llegar a una condición de salida.

2. Extiende el lenguaje While con el operador:

$$\text{for x := a1 to a2 do c}$$

esto es:

a) Modifica la estructura de la maquina W (agregando marcos, estados o transiciones) para evaluar la expresion for.

b) Da las reglas de semantica estatica para verificacion de tipos para el nuevo operador for.

c) ¿Es posible definir el operador for como azucar sintactica dentro del lenguaje While? justifica tu respuesta.

3. Decimos que dos programas en el lenguaje While son equivalentes (c1 ≡w c2) si y solo sı la ejecucion de ambos programas resulta en el mismo estado, es decir, si para todo estado de las variables σ, $\blacklozenge \succ \langle$c1, $\sigma\rangle \rightarrow *$ W $\blacklozenge \prec \sigma'$ y $\blacklozenge \succ \langle$c2, $\sigma\rangle \longrightarrow *W \blacklozenge \prec \sigma'$ entonces c1 ≡w c2.
Con la definicion de equivalencia anterior, demuestra o da un contraejemplo de lo siguiente:

a) ≡w realmente es una relacion de equivalencia. Esto es, demuestra que la relacion ≡w es transitiva, reflexiva y simetrica.

b) c;skip ≡w c

c) c1; c2 ≡w c2; c1

d) c1; (c2; c3) ≡w (c1; c2); c3