LÓGICA PARA COMPUTACIÓN

Año 2026

PRÁCTICO 8: VERIFICACIÓN DE PROGRAMAS SECUENCIALES

Ejercicio 1:

¿En qué circunstancias $if(B)\{C_1\}else\{C_2\}$ fallaría para terminar?

Ejercicio 2:

Un comando familiar que falta en nuestro lenguaje es la estructura de control **for**. El mismo puede ser utilizado para sumar los elementos de un arreglo, por ejemplo, programando de la siguiente manera:

$s = 0;$
for $i = 0; i \leq max; i = i + 1$ **do**
  $s = s + a[i];$

Despues de ejecutar la asignación inicial $s = 0$, ejecuta primero $i = 0$, luego ejecuta el cuerpo $s = s + a[i]$ y seguidamente incrementa $i = i + 1$ continuamente hasta que $i \leq max$ se vuelva falso. Explicar cómo $for(C_1; B; C_2)\{C_3\}$ puede ser definido como un programa derivado en nuestro lenguaje central.

Ejercicio 3:

Utilizar las reglas de prueba para la asignación y la implicación lógica según corresponda para mostrar la validez de:

1. $\vdash_{par} \{x > 0\}(\mathrm{y = x + 1})\{y > 1\}$
2. $\vdash_{par} \{\top\}(\mathrm{y = x;\ y = x + x + y})\{y = 3 * x\}$
3. $\vdash_{par} \{x > 1\}(\mathrm{a = 1;\ y = x;\ y = y - a})\{y > 0 \wedge x > y\}$

Ejercicio 4:

Escribir un programa P tal que cumpla con la corrección parcial, luego probar que esto es así.

1. $\{\top\}P\{y = x + 2\}$
2. $\{\top\}P\{z > x + y + 4\}$

Ejercicio 5:

Probar la validez de la secuencia $\vdash_{par} \{\top\}P\{z = min(x, y)\}$, donde $min(x, y)$ es el menor de los números entre x e y (por ejemplo, $min(7, 3) = 3$) y el código de P es el que se muestra a continuación:

if $(x > y)$ **then**
  $z = y;$
else
  $z = x;$

Ejercicio 6:

Para cada una de las especificaciones siguientes, escribir el psudocódigo para P y demuestrar la corrección parcial del comportamiento de entrada/salida especificado:

1. $\{\top\}P\{z = max(w, x, y)\}$, donde $max(w, x, y)$ denota el mayor entre w, x e y.
2. $\{\top\}P\{((x = 5) \rightarrow (y = 3)) \wedge ((x = 3) \rightarrow (y = -1))\}$

Ejercicio 7:

Mostrar que $\vdash_{par} \{x \geq 0\}Copy1\{x = y\}$ es valido, donde *Copy1* corresponde al siguiente código:

$a = x;$
$y = 0;$
while $(a \neq 0)$ **do**
    $y = y + 1;$
    $a = a - 1;$

Ejercicio 8:

Mostrar que $\vdash_{par} \{y \geq 0\}Multi1\{z = x \cdot y\}$ es valido, donde *Multi1* corresponde al siguiente código:

$a = 0;$
$z = 0;$
while $(a \neq y)$ **do**
    $z = z + x;$
    $a = a + 1;$

Ejercicio 9:

Mostrar que $\vdash_{par} \{x \geq 0\}Copy2\{x = y\}$ es valido, donde *Copy2* corresponde al siguiente código:

$y = 0;$
while $(y \neq x)$ **do**
    $y = y + 1;$

Ejercicio 10:

Probar la correctitud parcial del siguiente programa:

$\{a \geq 0\}$
$x = 0;$
$y = 1;$
while $(y \leq a)$ **do**
    $x = x + 1;$
    $y = y + 2 * x + 1;$
$\{0 \leq x^2 \leq a < (x + 1)^2\}$

Ejercicio 11:

Probar la correctitud parcial del siguiente programa, donde la función $mcd(\cdot,\cdot)$ devuelve el máximo común divisor de los dos parámetros que recibe.

$\{a > 0 \land b > 0\}$

$x = a;$
$y = b;$
while $(x \neq y)$ **do**
  if $(x > y)$ **then**
    $x = x - y;$
  else
    $y = y - x;$

$\{x = mcd(a, b)\}$

Ejercicio 12:

Probar la correctitud parcial del siguiente programa, donde la función $mcd(\cdot,\cdot)$ devuelve el máximo común divisor de los dos parámetros que recibe.

$\{a > 0 \land b > 0\}$

$x = a;$
$y = b;$
while $(x \neq y)$ **do**
  while $(x > y)$ **do**
    $x = x - y;$
  while $(y > x)$ **do**
    $y = y - x;$

$\{x = mcd(a, b)\}$

Ejercicio 13:

Probar la correctitud parcial del siguiente programa:

$\{a \geq 0 \land b \geq 0\}$

$x = a;$
$y = b;$
$z = 1;$
while $(y \neq 0)$ **do** ▷ *y es impar*
  if $(y\,\%2 == 1)$ **then**
    $y = y - 1;$
    $z = x * z;$
  else
    $x = x * x;$
    $y = y/2;$

$\{z = a^b\}$

Ejercicio 14:

Probar la correctitud total de los siguientes programas:

1. $\vdash_{tot} \{x \geq 0\} Copy1 \{x = y\}$
2. $\vdash_{tot} \{y \geq 0\} Multi1 \{z = x \cdot y\}$
3. $\vdash_{tot} \{x \geq 0\} Copy2 \{x = y\}$
   ¿Tiene su invariante un papel activo para asegurar la corrección?