

## Asignatura Avanzada - Práctica 10 (Cálculo de programas imperativos)

Departamento de Computación-UNRC

# 2017

**Ejercicio 1.** Derivar dos programas que calculen  $\mathbf{r} = X^Y$  a partir de cada una de las siguientes definiciones de la función exponencial:

(a)

$$\text{exp}(x, y) = (y = 0 \rightarrow 1 \\ \quad \boxed{y \neq 0 \rightarrow x * \text{exp}(x, y - 1)})$$

(b)

$$\begin{aligned} \text{exp}(x, y) = & (y = 0 \rightarrow 1 \\ & \quad \boxed{y \neq 0 \rightarrow (y \bmod 2 = 0 \rightarrow \text{exp}(x * x, y \text{div} 2) \\ & \quad \quad \boxed{y \bmod 2 = 1 \rightarrow x * \text{exp}(x, y - 1)} \\ & \quad )} \\ & ) \end{aligned}$$

**Diseñar los dos programas a partir de:**

Precondición R:  $\{\mathbf{x} = \mathbf{X} \wedge \mathbf{y} = \mathbf{Y} \wedge \mathbf{x} \geq 0 \wedge \mathbf{y} \geq 0\}$

Postcondición Q:  $\{r = X^Y\}$

Invariante P:  $\{y \geq 0 \wedge r * x^y = X^Y\}$

Para cada programa usar una de las definiciones. Tener en cuenta las mismas a la hora de decidir la manera de achicar la cota.

**Ejercicio 2.** Dado  $n > 0$ , desarrollar un programa que devuelva en la variable  $k$  la mayor potencia de 2 menor o igual que  $n$ .

Precondición R:  $\{n > 0\}$ 

Postcondición Q:  $\{0 < k \leq n \wedge n < 2 * k \wedge \langle \exists j : 0 < j : k = 2^j \rangle\}$

Invariante P:  $\{0 < k \leq n \wedge \langle \exists j : 0 < j : k = 2^j \rangle\}$

Para cada programa usar una de las definiciones. Tener en cuenta las mismas a la hora de decidir la manera de achicar la cota.

**Ejercicio 3.** Dado el siguiente programa:

```

Array a[0..n) of Int;
Var i,x:Int;
{n>0}
  x,i := a.0,1;
  do i < n ∧ a.i > x → x,i := a.i,i + 1
  [] i < n ∧ a.i ≤ x → i := i + 1
  od
{⟨∃i : 0 ≤ i < n : a.i = x⟩ ∧ ⟨∀i : 0 ≤ i < n : x ≥ a.i⟩}

```

En donde MINInt es el entero más pequeño que puede ser considerado en este lenguaje de programación.

- (a) ¿Qué hace este algoritmo?
- (b) Demostrar la corrección del algoritmo.

**Ejercicio 4.** Sea A un arreglo de enteros.

- Derivar un programa que determine si todos los elementos de A son positivos.
- Derivar un programa que determine si algún elemento de A es positivo.

**Ejercicio 5.** Se quiere obtener un programa que, dado un arreglo de enteros y un par de posiciones i y j del mismo, determine el valor máximo almacenado en el arreglo entre las posiciones i y j.

Suponga que el arreglo tiene N elementos y vale  $0 \leq i < j \leq N$

**Ejercicio 6.** Derivar un programa para la siguiente especificación:

```

M : Int, A : Array[0..M) of Int
var r:Int
{M ≥ 1}
  S
{r = ⟨Np : 0 ≤ p < M : A.p ≥ 0⟩}

```

**Ejercicio 7.** Calcular un programa que, dados dos enteros positivos x e y, devuelva en una variable el mínimo común múltiplo de ambos.

Ayuda: el mínimo común múltiplo de dos enteros positivos se puede especificar por:

$$\text{mcm.x.y} = \langle \text{Min } n : 1 \leq n \wedge n \bmod x = 0 \wedge n \bmod y = 0 : n \rangle$$

**Ejercicio 8.** Derivar un programa para la siguiente especificación:

```

M : Int, A : Array[0..M) of Int
var r:Int
{M ≥ 1}
  S
{r = ⟨Np,q : 0 ≤ p < q < M : A.p * A.q ≥ 0⟩}

```