

Práctico 4

Derivación de Programas Imperativos

Algoritmos y Estructuras de Datos I
1^{er} cuatrimestre 2023

1. (Algoritmo de la división) Dados dos números, hay que encontrar el cociente y el resto de la división entera entre ellos.

Ayuda: Para enteros $x \geq 0$ e $y > 0$, el cociente q y el resto r de la división entera de x por y están caracterizados por $x = q * y + r \wedge 0 \leq r \wedge r < y$. Por lo tanto, debemos derivar un programa S que satisfaga

Var $x, y, q, r : Int$;
 $\{P : x \geq 0 \wedge y > 0\}$ (precondición)
 S
 $\{Q : x = q * y + r \wedge 0 \leq r \wedge r < y\}$ (postcondición)
donde las variables x y y no cambian.

2. Sea $N \geq 0$, especificar y derivar un programa que calcule el menor entero x que satisface $x^3 + x \geq N$.

Ayuda: Especifique el problema (con pre y poscondición) de forma que en la poscondición queden conjunciones y así poder utilizar la técnica “tomar término de la conjunción”. Para ellos fijarse como se hace esto al especificar el Ejemplo 19.2 del libro.

3. Sea $N \geq 0$, especificar y derivar un programa que calcule el mayor entero x que satisface $x^3 + x \leq N$.

4. (Suma de los elementos de un arreglo) Dado un arreglo de enteros, se debe devolver en una variable la suma de todos los elementos del arreglo.

Ayuda: Debemos derivar un programa que satisfaga la siguiente especificación:

Const $N : Int$;
Var $a : array [0, N) of Int$;
 $s : Int$;
 $\{P : N \geq 0\}$
 S
 $\{Q : s = \langle \sum i : 0 \leq i < N : a.i \rangle\}$
donde el arreglo a no cambia.

5. Sea a un arreglo de enteros.
 - a) Especificar y derivar un programa que determine si todos los elementos de a son mayores a 0.
 - b) Especificar y derivar un programa que determine si algún elemento de a es mayor a 0.
6. Especificar y derivar un programa que calcule el factorial de un número.
7. Dado un arreglo $a : array[0, N) of Num$ con $N \geq 0$ determinar si alguno de sus elementos es igual a la suma de los anteriores.

8. Dado un arreglo $a : \text{array}[0, N) \text{ of } \text{Num}$ con $N \geq 0$ determinar si sus elementos son iguales al factorial de la posición.

Ayuda: Que los elementos sean el factorial de la posición se puede expresar como $\langle \forall i : 0 \leq i < N : a.i = i! \rangle$.

9. Especificar y hacer un programa imperativo que calcule fibonacci de un número dado.
10. (Máxima diferencia) Dado un arreglo de enteros, se debe tomar la máxima diferencia entre dos de sus elemento (en orden, el primero menos el segundo).

La especificación del programa es:

```
Const N : Int;
Var a : array[0, N) of Int; r : Int;
{P : N ≥ 2}
S
{Q : r = ⟨Max p, q : 0 ≤ p < q < N : a.p - a.q⟩}
```

11. (Segmento de suma máxima) Dado un arreglo de enteros, se debe guardar en una variable la suma del segmento de suma máxima del arreglo.

La especificación del programa es:

```
Const N : Int;
Var a : array[0, N) of Int; r : Int;
{P : N ≥ 0}
S
{Q : r = ⟨Max p, q : 0 ≤ p ≤ q ≤ N : sum.p.q⟩
  || sum.p.q = ⟨∑ i : p ≤ i < q : a.i⟩⟩}
```

12. (Máxima diferencia cuadrada) Derivar un programa para la siguiente especificación:

```
Const N : Int;
Var a : array[0, N) of Int; r : Int;
{P : N ≥ 2}
S
{Q : r = ⟨Max p, q : 0 ≤ p < q < N : (a.p - a.q)2⟩}
```

13. Derivar el siguiente programa

```
Const N : Int;
Var a : array [0, N) of Num;
    r : Bool;
{N ≥ 0}
S
{r = ⟨∀ i : 0 ≤ i < N : ⟨∑ j : 0 ≤ j ≤ i : a.j⟩ > 0⟩}
```

14. Derivar un programa para la siguiente especificación:

```
Const N : Int;
Var a : array[0, N) of Int; r : Bool;
{P : N ≥ 2}
S
{Q : r ≡ ⟨∃ p, q : 0 ≤ p < q < N : a.p - a.q ≤ 8⟩}
```

Ayuda: reescribir la postcondición usando el operador de mínimo.

15. Considere el ejercicio 5.

- a) ¿En el programa derivado en el ejercicio 5a se recorre todo el arreglo? Si la respuesta es afirmativa piense si esto es necesario. Si no lo es busque una manera de derivar un programa equivalente que no recorra innecesariamente todo el arreglo.
- b) Repita el mismo razonamiento sobre el ejercicio 5b y derive si es necesario el programa mejorado.

Ejercicios extra

16. Especificar y derivar un programa que calcule la cantidad de elementos pares de un arreglo de enteros.

17. Decir con palabras que pide cada especificación y derivar los programas correspondientes.

- a) $\text{Const } N : \text{Int}, a : \text{array}[0, N) \text{ of Int};$
 $\text{Var } r : \text{Bool};$
 $\{P : N > 1\}$
 S
 $\{Q : r \equiv \langle \exists i : 2 \leq i < N : a.i = a.(i-1) + a.(i-2) \rangle\}$
- b) $\text{Const } N : \text{Int}, a : \text{array}[0, N) \text{ of Int};$
 $\text{Var } r : \text{Num};$
 $\{P : N > 0\}$
 S
 $\{Q : r = \langle \sum i : 0 \leq i < N - 1 : a.i * a.(i+1) \rangle\}$

18. Derivar el siguiente programa

$\text{Const } N : \text{Int};$
 $\text{Var } a : \text{array } [0, N) \text{ of Int};$
 $r : \text{Int};$
 $\{P : N \geq 0\}$
 S
 $\{Q : r = \langle \sum i : 0 \leq i < N \wedge 0 \leq a.i < N : a.(a.i) \rangle\}$
donde el arreglo a no cambia.

19. Derivar el siguiente programa

$\text{Const } M : \text{Int};$
 $\text{Var } a : \text{array } [0, M) \text{ of Int};$
 $r : \text{Int};$
 $\{P : M \geq 0\}$
 S
 $\{Q : r = \langle \sum i : 0 \leq i < M : \langle \sum j : 0 \leq j < i : a.j \rangle \leq i * a.i \rangle\}$
donde el arreglo a no cambia.

20. Especificar y derivar un programa que guarde en una variable el máximo de los primeros n elementos de un arreglo de tamaño N con $0 < n \leq N$. Tener en cuenta que **no se pueden usar** los valores ∞ , $-\infty$.

Una vez finalizado el ejercicio pensar como se podría modificar el programa para detectar el caso en que el arreglo sea vacío.

21. Derivar el siguiente programa

```

Const  $N : Int$ ;
Var  $a : array [0, N) \text{ of } Int$ ;
     $r : Int$ ;
{ $P : N > 0$ }
S
{ $Q : r = \langle \text{Max } i : 0 \leq i < N \wedge \langle \forall j : 0 < j < i : a.(j-1) < a.j \rangle : i \rangle$ }

```

Nota: Antes de derivar decir que debería hacer el programa.

Nota: No se puede usar ∞ en el programa.

22. ¿En el programa derivado en el ejercicio 7 se recorre todo el arreglo? Si la respuesta es afirmativa piense si esto es necesario. Si no lo es busque una manera de derivar un programa equivalente que no recorra innecesariamente todo el arreglo.
23. (Cantidad de productos positivos) Derivar un programa para la siguiente especificación:

```

Const  $M : Int$ ;
Var  $a : array [0, M) \text{ of } Int; r : Int$ ;
{ $P : M \geq 0$ }
S
{ $Q : r = \langle N p, q : 0 \leq p < q < M : a.p * a.q \geq 0 \rangle$ }

```