Práctico 4
Derivación de Programas Imperativos

Algoritmos y Estructuras de Datos I
2do 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

   Const $x, y : Int$;
   Var $q, r : Int$;
   $\{P : \quad x \geq 0 \wedge y > 0\}$ (precondición)
   S
   $\{Q : \quad x = q * y + r \wedge 0 \leq r \wedge r < y\}$ (postcondición)

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

4. (Suma de los elementos de un arreglo) Dado un arreglo de enteros, especificar y derivar un programa que calcule la suma de todos los elementos del arreglo.

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 la suma de los elementos pares de un arreglo de enteros.

7. Especificar y derivar un programa que calcule el factorial de un número.

8. Dado un arreglo $A : array[0, N)\, of\, Num$ con $N \geq 0$, contar cuántas veces coinciden dos elementos:

   Const $N : Int, A : array\ [0, N)\, of\, Int$;
   Var $r : Int$;
   $\{P : N \geq 0\}$
   S
   $\{Q : \ r = \langle N\, i, j \ : 0 \leq i < j < N : \ A.i = A.j \ \rangle\}$

9. Dado un arreglo $A : array[0, N)\ of\ Num$ con $N \geq 0$, determinar si hay dos elementos que suman 8:

   Const $N : Int, A : array\ [0, N)\ of\ Int$;
   Var $r : Bool$;
   $\{P : N \geq 0\}$
   S
   $\{Q :\ r = \langle\, \exists\, i, j\ : 0 \leq i < j < N :\ A.i + A.j = 8\, \rangle\}$

10. Especificar y derivar: 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. Usar fortalecimiento de invariante.

11. Especificar y derivar: Dado un arreglo $a : array[0, N)\ of\ Num$ con $N \geq 0$ determinar si sus elementos son iguales al factorial de la posición. Usar fortalecimiento de invariante.

12. Especificar y derivar un programa imperativo que calcule Fibonacci de un número dado. Usar fortalecimiento de invariante.

13. (Máxima diferencia) Dado un arreglo de enteros, calcular la máxima diferencia entre dos de sus elemento (en orden, el primero menos el segundo).

    La especificación del programa es:

    Const $N : Int$;
    Var $a : array[0, N)\ of\ Int; r : Int$;
    $\{P : N \geq 2\}$
    S
    $\{Q : r = \langle \text{Max}\ \ p, q\ : 0 \leq p < q < N :\ a.p - a.q\, \rangle\}$

14. (Segmento de suma máxima) Dado un arreglo de enteros, calcular la suma del segmento de suma máxima del arreglo.

    La especificación del programa es:

    Const $N : Int$;
    Var $a : array[0, N)\ of\ Int; r : Int$;
    $\{P : N \geq 0\}$
    S
    $\{Q : r = \langle \text{Max}\ \ p, q\ : 0 \leq p \leq q \leq N :\ sum.p.q\, \rangle$
    $\quad [\![sum.p.q = \langle \sum i\ : p \leq i < q :\ a.i\, \rangle]\!]\}$

15. Dada la siguiente especificación:

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

    Decir en palabras qué hace el programa y derivarlo.

Ejercicios extra

16. Derive un programa para calcular el máximo común divisor entre dos enteros positivos. Utilice la siguiente especificación:

Const $X, Y : Int$;
Var $x, y : Int$;
$\{X > 0 \wedge Y > 0 \wedge x = X \wedge y = Y\}$
S
$\{x = mcd.X.Y\}$

Utilice como invariante $\{I : x > 0 \wedge y > 0 \wedge mcd.x.y = mcd.X.Y\}$.

Para la derivación serán de utilidad las siguientes propiedades del mcd:

a) $mcd.x.x = x$

b) $mcd.x.y = mcd.y.x$

c) $x > y \;\Rightarrow\; mcd.x.y = mcd.(x - y).y$

d) $y > x \;\Rightarrow\; mcd.x.y = mcd.x.(y - x)$

17. Considere las siguientes definiciones recursivas de la función de exponenciación $exp.x.y$, especificada como $exp.x.y = x^y$:

a) Definición de complejidad lineal:

$$\begin{array}{lcl} exp.x.y & = & (\;\; y = 0 \to 1 \\ & & \Box\; y \neq 0 \to x * exp.x.(y-1) \\ & &) \end{array}$$

b) Definición de complejidad logarítmica:

$$\begin{array}{lcl} exp.x.y & = & (\;\; y = 0 \to 1 \\ & & \Box\; y \neq 0 \to (\;\; y \bmod 2 = 0 \to exp.(x*x).(y \div 2) \\ & & \qquad\quad \Box\; y \bmod 2 = 1 \to x * exp.x.(y-1) \\ & & \qquad\quad) \\ & &) \end{array}$$

Derive **dos** programas imperativos que calculen la exponenciación, cada uno utilizando una de las definiciones recursivas. Utilice la siguiente especificación:

Const $X, Y : Int$;
Var $x, y, r : Int$;
$\{x = X \wedge y = Y \wedge x \geq 0 \wedge y \geq 0\}$
S
$\{r = X^Y\}$

Utilice como invariante $\{I : y \geq 0 \wedge r * x^y = X^Y\}$.

18. Considere el ejercicio 5.

a) ¿En el programa derivado en el ejercicio 5*a* 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 5*b* y derive si es necesario el programa mejorado.

19. Derivar el siguiente programa

Const $M : Int, A : array\ [0, M)\ of\ Int$;
Var $r : Int$;
$\{P : M \geq 0\}$
S
$\{Q : r = \langle \mathrm{N}\, i \;:\; 0 \leq i < M :\; \langle \sum j \;:\; 0 \leq j < i :\; A.j \,\rangle \leq i * A.i \,\rangle\}$

20. Derivar el siguiente programa

Const $N : Int$;
Var $a : array\ [0, N)\ of\ Int$;
$r : Int$;
$\{P : N > 0\}$
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.

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

Const $N : Int$;
Var $a : array[0, N)\ of\ Int; r : Int$;
$\{P : N \geq 2\}$
S
$\{Q : r = \langle \text{Max } p, q\ : 0 \leq p < q < N :\ (a.p - a.q)^2 \rangle\}$