

Algoritmos y Estructuras de Datos II_(AED-II/AED)

Guía Práctica II

Estudiante	L.U.	E-mail (principal)
Bayón, Nicolás	160/20	nico.bayon@hotmail.com



Figure 1: **Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires**

Ciudad Universitaria
Intendente Güiraldes 2160
Ciudad Autonoma de Buenos Aires- Rep. Argentina
Tel. (++54 +11) 5285-7400
<http://www.exactas.uba.ar>

Ejercicios

2.1 Funciones auxiliares

Ejercicio 1.

- a) **pred** es_Raiz_Cuadrada $(x : \mathbb{Z}) \{$
 $(\exists c : \mathbb{Z})(c > 0 \wedge (c * c = x))$
 $\}$
b) **pred** no_Existen_Divisor_De $(x : \mathbb{Z}) \{$
 $(x > 1) \wedge (\forall n : \mathbb{Z})(1 < n < x \rightarrow_L (x \bmod n \neq 0))$
 $\}$

Ejercicio 2.

- a) "pred sonCoprimos $(x, y : \mathbb{Z})$ que sea verdadero si y sólo si x e y son coprimos":

pred sonCoprimos $(x, y : \mathbb{Z}) \{$
 $(\forall g : \mathbb{Z})(((\underline{g \neq 0}) \wedge_L ((x \bmod g = 0) \wedge (y \bmod g = 0))) \leftrightarrow (g = \pm 1))$
 $\}$

- b) "pred mayorPrimoQueDivide $(x : \mathbb{Z}, y : \mathbb{Z})$ que sea verdadero si y es el mayor primo que divide a x":

pred mayorPrimoQueDivide $(x : \mathbb{Z}, y : \mathbb{Z}) \{$
 $((esPrimo(y)) \wedge (x \bmod y = 0)) \rightarrow ((\forall u : \mathbb{Z})(((esPrimo(u)) \wedge (x \bmod u = 0)) \rightarrow (u < y)))$
 $\}$

predicado auxiliar:

pred esPrimo $(t : \mathbb{Z}) \{$
 $(\forall p : \mathbb{Z})(((p \neq 0) \wedge_L (t \bmod p = 0)) \rightarrow ((p = \pm 1) \vee (p = \pm t)))$
 /*O equivalentemente: */ \uparrow
 $(\forall a, b : \mathbb{Z})(((t \neq 0) \wedge_L ((a * b) \bmod t = 0)) \xrightarrow{\text{rojo}}_L ((a \bmod t = 0) \vee (b \bmod t = 0)))$
 /*la flecha roja vale por transitividad de la division*/
 $\}$

Ejercicio 3.

- a) **pred** noHayMenorACero $(s : \text{seq}(\mathbb{Z})) \{$
 $(\forall i : \mathbb{Z})((0 \leq i < |s|) \rightarrow_L s[i] \geq 0)$
 $\}$
b) **pred** sonTodosElementosDistintos $(s : \langle \mathbb{Z} \rangle) \{$
 $(\forall i : \mathbb{Z})((0 \leq i < |s|) \rightarrow_L (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L (s[i] \neq s[j])))$
 $\}$

Ejercicio 4.

- a) esPrefijo, que determina si una secuencia es prefijo de otra.
- b) estáOrdenada, que determina si la secuencia está ordenada de menor a mayor.
- c) hayUnoParQueDivideAlResto, que determina si hay un elemento par en la secuencia que divide a todos los otros elementos de la secuencia.
- d) enTresPartes, que determina si en la secuencia aparecen (de izquierda a derecha) primero 0s, después 1s y por último 2s. Por ejemplo $\langle 0, 0, 1, 1, 1, 1, 2 \rangle$ cumple con enTresPartes, pero $\langle 0, 1, 3, 0 \rangle$ o $\langle 0, 0, 0, 1, 1 \rangle$ no. ¿Cómo modificaría la expresión para que se admitan cero apariciones de 0s, 1s y 2s

(es decir, para que por ejemplo $\langle 0, 0, 0, 1, 1 \rangle$ o $\langle \rangle$ sí cumplan enTresPartes)?

a)

pred esPrefijo ($s1 : seq\langle \mathbb{Z} \rangle, s2 : seq\langle \mathbb{Z} \rangle$) {
 $(|s1| \leq |s2|) \wedge_L ((\forall i : \mathbb{Z})((0 \leq i < |s1|) \rightarrow_L (s1[i] = s2[i])))$
}

b)

pred estáOrdenada ($l : seq\langle \mathbb{Z} \rangle$) {
 $(\forall k : \mathbb{Z})((0 \leq k < |l|) \rightarrow_L ((\forall j : \mathbb{Z})((0 \leq k \leq j < |l|) \rightarrow_L (l[k] \leq l[j])))$
}

c)

pred hayUnoParQueDivideAlResto ($l : seq\langle \mathbb{Z} \rangle$) {
 $(\exists ls : \mathbb{Z})(((0 \leq ls < |l|) \wedge (esPar(l[ls]))) \rightarrow_L (\neg((\exists n : \mathbb{Z})((0 \leq n < |l|) \wedge (l[n] \bmod l[ls] \neq 0))))$

predicado auxiliar:

pred esPar ($m : \mathbb{Z}$) {
 $m \bmod 2 = 0$
}

d)

pred enTresPartes ($l : seq\langle \mathbb{Z} \rangle$) {
 $(estáOrdenada(l)) \wedge ((\forall i, j, k : \mathbb{Z})((0 \leq i < j < k < |l|) \rightarrow_L ((l[i] = 0)(l[j] = 1)(l[k] = 2))))$

Ejercicio 5.

a) Cuento la cantidad de veces que aparece el elemento e de tipo Z en la secuencia s.

b) Sumo los elementos en las posiciones impares de la secuencia s.

c) Sumo los elementos mayores a 0 contenidos en la secuencia s.

d) Sumo los inversos multiplicativos ($\frac{1}{x}$) de los elementos contenidos en la secuencia s distintos a 0.

a)

aux cantidadDeApariciones($e : \mathbb{Z}, l : seq\langle \mathbb{Z} \rangle$): $\mathbb{Z} =$
 $\sum_{n=0}^{|l|-1} (\text{if } l[n] = e \text{ then } 1 \text{ else } 0 \text{ fi})$

b)

aux sumaEnPosicionesImpares($l : seq\langle \mathbb{Z} \rangle$): $\mathbb{Z} =$
 $\sum_{n=0}^{|l|-1} (\text{if } \neg(esPar(l[n])) \text{ then } l[n] \text{ else } 0 \text{ fi})$

c)

aux sumaPositivosEstrictos($l : seq\langle \mathbb{Z} \rangle$): $\mathbb{Z} =$
 $\sum_{n=0}^{|l|-1} (\text{if } l[n] > 0 \text{ then } l[n] \text{ else } 0 \text{ fi})$

d)

aux sumaEnPosicionesImpares($l : seq\langle \mathbb{Z} \rangle$): $\mathbb{Z} =$
 $\sum_{n=0}^{|l|-1} (\text{if } l[n] \neq 0 \text{ then } \frac{1}{l[n]} \text{ else } 0 \text{ fi})$

2.2 Análisis de especificación

Ejercicio 6.

a) **progresionGeometricaFactor2:** Indica si la secuencia l representa una progresión geométrica factor 2. Es decir, si cada elemento de la secuencia es el doble del elemento anterior.

```
proc progresionGeometricaFactor2 (in l: seq⟨ℤ⟩) : Bool {  
  requiere : { True }  
  asegura : {  $res = True \leftrightarrow_L ((\forall i : \mathbb{Z})(0 \leq i < |l| \rightarrow_L l[i] = 2 * l[i - 1]))$  }  
}
```

/*El problema que pude ver es que cuando $i=0$ el antecedente se vuelve verdadero así que el entonces luego evalúa el consecuente y en el mismo $l[0-1]$ se indefiniría así que quedaría indefinido pero no se si hay algo más*/

b) **minimo:** Devuelve en res el menor elemento de l . **proc minimo** (in l: seq⟨ℤ⟩) : ℤ {

```
  requiere : { True }  
  asegura : {  $(\forall y : \mathbb{Z})(y \in l \wedge y \neq x \rightarrow y > res)$  }  
}
```

/*El problema que pude ver es que aparece un x que no sale de ningún lado y no se sabe que es y además lo usa para comparar con y en el antecedente de la implicación y no lo usa después en el consecuente ni en ningún otro lado y esa condición tampoco me parece necesaria y aparte incluso sacándola no se si está contemplando el caso de $res = y$ que debería ya que el mínimo de una lista es $j =$ a todos los demás elementos no simplemente j a ellos pero no se si hay algo más*/