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

<u>Guía Práctica II</u>

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_Existe_Divisor_De $(x:\mathbb{Z})\{$
$(x>1) \wedge (\forall n:\mathbb{Z})((1<n<x) \rightarrow_L (x \, mod \, n \neq 0))$
$\}$

Ejercicio 2.
a) "pred sonCoprimos $(x,y:\mathbb{Z})$ que sea verdadero si y sólo si x e y son coprimos":

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

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

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

predicado auxiliar:
pred esPrimo $(t:\mathbb{Z})$ {
$(\forall p:\mathbb{Z})(((p \neq 0) \wedge_L (t \, mod \, p=0)) \rightarrow ((p=\pm 1) \vee (p=\pm t)))$
/*O equivalentemente: */ $\updownarrow$
$(\forall a,b:\mathbb{Z})(((t \neq 0) \wedge_L ((a*b) \, mod \, t=0)) \leftarrow \rightarrow_L ((a \, mod \, t=0) \vee (b \, mod \, t=0)))$
/*la flecha roja vale por transitividad de la division*/
}

Ejercicio 3.
a) **pred** noHayMenorACero (s: seq$\langle\mathbb{Z}\rangle$) {
$(\forall i:\mathbb{Z})((0 \leq i < |s|) \rightarrow_L s[i] \geq 0)$
}

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

<u>**predicado auxiliar:**</u>
pred esPar ($m : \mathbb{Z}$) {
m mod 2 = 0
}

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

<u>Ejercicio 5.</u>
a) Cuente la cantidad de veces que aparece el elemento e de tipo Z en la secuencia s.
b) Sume los elementos en las posiciones impares de la secuencia s.
c) Sume los elementos mayores a 0 contenidos en la secuencia s.
d) Sume 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$\langle\mathbb{Z}\rangle$): $\mathbb{Z}$ =
$\sum_{n=0}^{|l|-1}$ (if $l[n] = e$ then 1 else 0 fi)

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

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

d)
aux sumaEnPosicionesImpares(l : seq$\langle\mathbb{Z}\rangle$): $\mathbb{Z}$ =
$\sum_{n=0}^{|l|-1}$ (if $l[n] \neq 0$ then $\frac{1}{l[n]}$ else 0 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$\langle\mathbb{Z}\rangle$) : **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 asi que el entonces luego evalua el consecuente y en el mismo l[0-1] se indefine así que quedaria indefinido pero no se si hay algo más*/

b) **minimo:** Devuelve en res el menor elemento de l. **proc minimo** (in l: seq$\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ {
requiere :{ True }
asegura : { $(\forall y : \mathbb{Z})((y \in l \land y \neq x) \rightarrow y > res)$ }
}

/*El problema que pude ver es que aparece un x que no sale de ningun lado y no se sabe que es y ademas lo usa para comparar con y en el antecedente de la implicacion y no lo usa después en el consecuente ni en ningun otro lado y esa condicion tampoco me parece necesaria y aparte incluso sacandola no se si esta contemplando el caso de res = y que deberia ya que el minimo de una lista es ¡= a todos los demás elementos no simplemente ¡ a ellos pero no se si hay algo más*/