



Guia 2

2do cuatrimestre 2024

Algoritmos y Estructuras de Datos I

Integrante	LU	Correo electrónico
Federico Barberón	112/24	jfedericobarberonj@gmail.com



Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Ciudad Universitaria - Pabellón I

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Argentina

Tel/Fax: (54 11) 4576-3359

<http://exactas.uba.ar>

Índice

1. Guia 2	3
1.1. Ejercicio 1	3
1.2. Ejercicio 2	3
1.3. Ejercicio 3	3
1.4. Ejercicio 4	3

1. Guía 2

1.1. Ejercicio 1

Nombrar los siguientes predicados sobre enteros:

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

1.2. Ejercicio 2

Escriba los siguientes predicados sobre números enteros en lenguaje de especificación:

- a) Que sea verdadero si y sólo si x e y son coprimos.
`pred sonCoprimos` ($x, y: \mathbb{Z}$) {
 $(\forall i: \mathbb{Z}) (i > 1 \rightarrow_L \neg(x \bmod i = 0 \wedge y \bmod i = 0))$
}
- b) Que sea verdadero si y es el mayor primo que divide a x .
`pred mayorPrimoQueDivide` ($x, y: \mathbb{Z}$) {
 $(esPrimo(y) \wedge_L x \bmod y = 0) \wedge \neg(\exists i: \mathbb{Z}) (esPrimo(i) \wedge_L (x \bmod i = 0 \wedge i > y))$
}

1.3. Ejercicio 3

Nombre los siguientes predicados auxiliares sobre secuencias de enteros:

- a) `pred todoPositivos` ($s: seq(\mathbb{Z})$) {
 $(\forall i: \mathbb{Z}) ((0 \leq i < |s|) \rightarrow_L s[i] \geq 0)$
}
- b) `pred todosDistintos` ($s: seq(\mathbb{Z})$) {
 $(\forall i: \mathbb{Z}) ((0 \leq i < |s|) \rightarrow_L (\forall j: \mathbb{Z}) ((0 \leq j < |s| \wedge i \neq j) \rightarrow_L (s[i] \neq s[j])))$
}

1.4. Ejercicio 4

Escriba los siguientes predicados auxiliares sobre secuencias de enteros, aclarando los tipos de los parámetros que recibe:

- a) *esPrefijo*, que determina si una secuencia es prefijo de otra.
`pred esPrefijo` ($s1, s2: seq(\mathbb{Z})$) {
 $(|s1| \leq |s2|) \wedge_L (\forall i: \mathbb{Z}) ((0 \leq i < |s1|) \rightarrow_L (s1[i] = s2[i]))$
}
- b) *estáOrdenada*, que determina si la secuencia está ordenada de menor a mayor.
`pred estáOrdenada` ($s: seq(\mathbb{Z})$) {
 $(\forall i: \mathbb{Z}) ((0 \leq i < |s| - 1) \rightarrow_L (s[i] \leq s[i + 1]))$
}

}

- c) *hayUnoParQueDivideAlResto*, que determina si hay un elemento par en la secuencia que divide a todos los otros elementos de la secuencia.

```

pred divideA (d, n:  $\mathbb{Z}$ ) {
     $(d \neq 0) \wedge_L n \text{ mód } d = 0$ 
}

pred hayUnoParQueDivideAlResto (s:  $\text{seq}\langle\mathbb{Z}\rangle$ ) {
     $(\exists i : \mathbb{Z}) ((0 \leq i < |s|) \wedge_L \text{esPar}(s[i]) \wedge (\forall j : \mathbb{Z}) ((0 \leq j < |s|) \wedge_L \text{divideA}(s[i], s[j])))$ 
}

```

- 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, 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)?

```

pred tieneSoloCeroUnoYDos (s:  $\text{seq}\langle\mathbb{Z}\rangle$ ) {
     $(\forall i : \mathbb{Z}) ((0 \leq i < |s|) \rightarrow_L (s[i] = 0 \vee s[i] = 1 \vee s[i] = 2))$ 
}

pred enTresPartes (s:  $\text{seq}\langle\mathbb{Z}\rangle$ ) {
     $\text{tieneSoloCeroUnoYDos}(s) \wedge \text{estaOrdenada}(s)$ 
}

```