

SECUENCIAS Y CUANTIFICADORES

6 de abril de 2018

¿QUÉ VAMOS A VER?

- ▶ Secuencias
- ▶ Ejercicios
- ▶ Cuantificadores
 - Para todo (\forall)
 - Existe (\exists)
- ▶ Ejercicios
- ▶ Sumatorias (\sum)
- ▶ Productorias (\prod)
- ▶ Ejercicios

*Menú
de Hoy*

- ▶ ¿Qué es una secuencia?
 - ▶ Lista o colección de elementos.
 - ▶ Puede tener repetidos.
 - ▶ Los elementos tienen cierto **orden**.
 - ▶ Todos los elementos son del **mismo tipo**.

ALGUNOS EJEMPLOS DE SECUENCIAS

Las siguientes expresiones, ¿son secuencias válidas?

- ▶ $\langle 1, 2, 3 + 4, 5, 7 - 6 \rangle = \langle 1, 2, 7, 5, 1 \rangle$ **SI!**
- ▶ $\langle \langle 1 \rangle, 3, \langle \rangle, \langle 4, 3 + 4 \rangle \rangle$ **NO!**
- ▶ $\langle \langle 2, 3 \rangle, \langle 5 + 1, 2 \rangle, \langle \rangle \rangle$ **SI!**
- ▶ $\langle 1, 2, 3, 'a' \rangle$ **NO!**
- ▶ $\langle '1', '2', '3', 'a' \rangle$ **SI!**

Todo muy lindo... Pero ¿Cómo se si dos secuencias son iguales?

tipos

elementos

orden

¡Más sobre esto luego!

- ▶ $length(a : seq\langle T \rangle) : \mathbb{Z}$ (notación: $|a|$)
- ▶ $a[i]$
- ▶ $head(a : seq\langle T \rangle) : T$
- ▶ $tail(a : seq\langle T \rangle) : seq\langle T \rangle$
- ▶ $addFirst(t : T, a : seq\langle T \rangle) : seq\langle T \rangle$
- ▶ $concat(a : seq\langle T \rangle, b : seq\langle T \rangle) : seq\langle T \rangle$
- ▶ $subseq(a : seq\langle T \rangle, d, h : \mathbb{Z}) : seq\langle T \rangle$
- ▶ $setAt(a : seq\langle T \rangle, i : \mathbb{Z}, val : T) : seq\langle T \rangle$

EJERCITEMOS

- ▶ $|\langle 4, 3, 1 \rangle| = 3$
- ▶ $|\langle 4, 'a', ocho \rangle| = \textcolor{red}{X}$
- ▶ $head(tail(\langle 5, 6, 7, 8 \rangle)) = 6$
- ▶ $\langle 'a', '3', 'q', '2' \rangle[0] = 'a'$
- ▶ $concat(\langle \rangle, \langle 2, 6 \rangle, \langle \langle 1 \rangle \rangle) = \langle \rangle, \langle 2, 6 \rangle, \langle 1 \rangle$
- ▶ $setAt(\langle 5, 4, 3, 2 \rangle, 3, 1) = \langle 5, 4, 3, 1 \rangle$

- ▶ \forall : Para Todo (forma copada de escribir Todos)
 - ▶ Notación: $(\forall x : T)P(x)$
 - ▶ Equivale a: "Para todo x de tipo T se cumple P de x "
- ▶ \exists : Existe (forma copada de escribir Alguno)
 - ▶ Notación: $(\exists x : T)P(x)$
 - ▶ Equivale a: "Hay algún elemento x de tipo T que cumple P de x "

Estos predicados, ¿evalúan a *True* o *False*? ¿Son iguales?

- ▶ $(\exists x : \mathbb{Z})((\forall y : \mathbb{Z})(x + y = 0))$
- ▶ $(\forall x : \mathbb{Z})((\exists y : \mathbb{Z})(x + y = 0))$

Entonces... ¿Cómo definiríamos la igualdad de secuencias?

Dado $S1$ y $S2$

- ▶ $((|S1| = |S2|) \wedge_L$
 $(\forall i : \mathbb{N})((0 \leq i < |S1|) \rightarrow_L (S1[i] = S2[i])))$

EJERCICIOS 1 Y 2

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

- A *TodosPrimos*, que es verdadero sii todos los elementos de una secuencia son primos (se puede usar el predicado *EsPrimo* de la semana pasada).
- B *TodosIguales*, que es verdadero sii todos los elementos de una secuencia son iguales.

EJERCICIOS 3 Y 4

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

- A *Capicúa*, que es verdadero sii una secuencia dada es capicúa.
- B *NingunoEnSuLugar*, que es verdadero sii ninguna posición de la secuencia contiene un valor que sea igual a su índice.

EJERCICIO 5

Sean $P(x)$ y $Q(x)$ predicados que nunca se indefinen y S una secuencia no nula. Escribir el predicado asociado a cada uno de los siguientes enunciados.

- A *“Si un entero en s cumple P , también cumple Q ”*
- B *“Todos los enteros de s que cumplen P y están en posiciones que cumplen Q , son pares”*
- C *“En toda secuencia no vacía de enteros cuyo primer elemento cumple P sucede que su largo cumple Q o su último elemento cumple Q ”*

¿Cómo decimos que un elemento pertenece (\in) a una secuencia?

El término

$$\sum_{i=from}^{to} Expr(i)$$

retorna la suma de todas las expresiones $Expr(i)$ entre $from$ y to .
Es decir,

$$Expr(from) + Expr(from + 1) + \cdots + Expr(to - 1) + Expr(to)$$

El término

$$\prod_{i=from}^{to} Expr(i)$$

retorna el producto de todas las expresiones $Expr(i)$ entre $from$ y to . Es decir,

$$Expr(from) * Expr(from + 1) * \cdots * Expr(to - 1) * Expr(to)$$

- ▶ Dada una secuencia S y un elemento e escribir un predicado que cuente la cantidad de apariciones del elemento e en S
- ▶ Dada una secuencia S escribir una función que sume los elementos en posiciones impares de S
- ▶ Dada una secuencia S escribir una función que devuelva 1 si todos los elementos de S son mayores a 5 y 0 si no

- ▶ Dada una sucesión S y un índice i indique si S es decreciente a partir de la posición i
- ▶ Dada una lista de listas de enteros y un entero N decidir si alguna de las listas tiene longitud N y además contiene el elemento N .

