

Listas

Cecilia Manzino

7 de mayo de 2024

- ▶ Datos de tamaño fijo:

- ▶ **Atómicos:** números, booleanos, cadenas, imágenes.
- ▶ **Estructuras:** nos permiten unir datos.

¿Podemos representar cualquier información con ellos?

No, porque todos ellos permiten representar una cantidad fija de información

- ▶ Ejemplos: Listas de precios, de contactos, etc.
- ▶ Ejemplos en Racket:

```
'()  
(cons "luis" '())  
(cons 1 (cons 2 (cons 3 '()))))
```

- ▶ Formas abreviada:

```
empty  
(list "luis")  
(list 1 2 3)
```

Definición de listas

Una lista es un tipo de datos **auto-referencial**.

Una Lista-de-números es o bien:

- '() o,
- (cons Number Lista-de-números)

Las definiciones **auto-referenciales** nos permiten construir datos de tamaño arbitrario.

- ▶ Constructores:

- '()' constante que representa la lista vacía
 - cons agrega un elemento a la lista

- ▶ Selectores:

- first devuelve el primer elemento de la lista
 - rest devuelve la lista sin el primer elemento

- ▶ Predicados:

- empty? determina si la lista es vacía
 - cons? determina si la lista es no vacía

Comencemos por un ejemplo que conocemos sobre recursión sobre naturales:

$$\textit{factorial}(0) = 1$$

$$\textit{factorial}(n) = n * \textit{factorial}(n - 1) \quad \text{si } n > 1$$

$$\textit{fib}(0) = 1$$

$$\textit{fib}(1) = 1$$

$$\textit{fib}(n) = \textit{fib}(n - 1) + \textit{fib}(n - 2) \quad \text{si } n > 2$$

En general se definen 2 casos:

```
(define (nombre-función l)
  (cond [(empty? l) ...]
        [(cons? l) ...]))
```

Ejemplo

; sum : Listof Number -> Number

; dada una lista de números devuelve la suma de ellos

```
(check-expect (sum '()) 0)
```

```
(check-expect (sum (list 1 2 3 4)) 10)
```

```
(define (sum l)
  (cond [(empty? l) 0]
        [(cons? l) (+ (first l)
                        (sum (rest l)))]))
```


Diseñar las siguientes funciones recursivas sobre listas:

1. **longitud**, dada una lista devuelve la cantidad de elementos de la misma.
2. **positivos**, dada una lista de números devuelve la cantidad de números positivos de la misma.
3. **coordenadas-x**, dada una lista de posn devuelve una lista con las coordenadas x de los puntos.