Listas

Cecilia Manzino

2 de mayo de 2023

Tipos de datos

- 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

Listas

- 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.

Operaciones sobre listas

- 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

Recursión sobre listas

Comencemos por un ejemplo que conocemos sobre recursión sobre naturales:

$$
\begin{aligned}
\mathit{factorial}(0) &= 1 \\
\mathit{factorial}(n) &= n * \mathit{factorial}(n-1) \quad \text{si } n > 1
\end{aligned}
$$

$$
\begin{aligned}
\mathit{fib}(0) &= 1 \\
\mathit{fib}(1) &= 1 \\
\mathit{fib}(n) &= \mathit{fib}(n-1) + \mathit{fib}(n-2) \quad \text{si } n > 2
\end{aligned}
$$

Recursión sobre listas

En general se definen 2 casos:

```
define  (f l)
        (cond  [(empty? l)  ...]
               [(cons? l)   ...])
```

Ejemplo

```
; sum : List (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)))
```