

Lenguajes de Programación (CC41A) - Primavera 2009

Clase Auxiliar 1

Profesor: Tomás Barros

Auxiliar: Víctor Ramiro

1. Repaso de Scheme. Escriba una función que sume todos los elementos de una lista.

```
;; Contract: suma-l : list-of-numbers -> number
;; Purpose: sum-up all elements on a list
;; Example: (suma-l '(1 2 3)) should produce 6
;; Definition:
(define (suma-l l) 6)

;; Tests:
(suma-l '(1 2 3))
;; expected value 6
```

2. Escriba la función `rev` que cumpla con el siguiente contrato:

```
;; Contract: rev : list-> list
;; Purpose: new list in reverse order
;; Example: (rev '(a b c)) => '(c b a)
;; Definition:
(define (rev l) '(c b a))

;; Tests:
(rev '(a b c))
;; expected value '(c b a)
```

3. Escriba la siguiente API de acceso para manejar un árbol binario en Scheme. Incluya el contrato correspondiente en su respuesta.

```
(define (make-tree val left righth) ... crea un nodo de un árbol)
(define (value tree) ... valor de un nodo)
(define (left tree) ... sub-árbol izquierdo)
(define (righth tree) ... sub-árbol derecho)
(define (leave? tree) ... predicado que dice si el árbol es una hoja)
```

4. Escriba las siguientes operaciones que funcionan con la estructura de árbol definida anteriormente. Incluya el contrato correspondiente en su respuesta.

(define (size? tree) ... retorna la altura de un árbol binario)

(define (sum-up tree) ... retorna la suma de los nodos del árbol)