

CC4101- Lenguajes de Programación

Auxiliar 1

Richard Ibarra Ramírez

28 de marzo de 2010

1. Defina la función `contains?` en Scheme. La función `contains?` verifica si un elemento se encuentra en una lista.

```
;; contains? : a list -> boolean
> (contains? 'b '(a b c))
#t
> (contains? 'd '(a b c))
#f
```

2. Defina la función `concat` en Scheme. La función `concat` retorna una lista resultantes de “pegar” dos listas.

```
;; concat : list list -> list
> (concat '(a b c) '(d e f))
(a b c d e f)
```

3. Defina la función `fold` en Scheme. La función `fold` reciba una función `f`, un valor inicial y una lista; un ejemplo de ejecución de `fold` se muestra a continuación.

```
;; (fold cons '() '(1 2 3))
(cons 1 '())
(cons 2 '(1))
(cons 3 '(2 1))
=> (3 2 1)
```

4. Usando la función `fold`, defina la función `sum-list` que suma los elementos de una lista.

```
;; (sum-list '(1 2 3))
> 6
```

