

INF-253 Lenguajes de Programación

Tarea 4: Scheme

Profesor: José Luis Martí Lara

Ayudante Cátedras: Sebastián Godínez San Martín

Ayudante Tareas: Gabriel Carmona Tabja - Sebastián Campos Muñoz

1. Objetivos

Conocer y aplicar correctamente los conceptos y técnicas del paradigma de programación funcional, utilizando el lenguaje **Scheme**.

2. Reglas

- Se presentarán 4 problemas en scheme. Cada uno posee una función a implementar, con su nombre y parámetros respectivos. Esto no restringe que se puedan crear funciones auxiliares. Cada problema debe ser resuelto por separado, en **archivos distintos** y de nombre tipo `p{número de problema}.rkt`.
- Pueden crear funciones que no estén especificadas para utilizar en los problemas planteados, pero solo se revisará que la función pedida funcione y el problema este resuelto con la característica funcional planteada en el enunciado.
- Para implementar las funciones utilice DrRacket.
 - Descargar DrRacket, [click aquí](#)

3. (((Boss)))

Después de la gran venta de su videojuego MIMS, Bowser y sus secuaces secuestraron a Rosemary, la princesa del mundo trabajador. Entonces Guiseppe se enfrentó a una gran aventura para rescatar a la princesa. Llegando casi al final se debe enfrentar a un jefe que tiene 4 fases, una más difícil que la otra. Ayuda a Guiseppe a derrotar al secuaz de Bowser!

4. Problemas

1. ¿Qué tengo en mi bolsa?

- **Sinopsis:** (tipos 11)
- **Característica Funcional:** Manejo de listas.
- **Descripción:** Se debe implementar una función que reciba una lista que contenga diferentes tipos de números y entregue a que tipo corresponde cada número siguiente la siguiente nomenclatura:
 - C: complejo
 - R: real
 - E: entero
- **Ejemplos:**

```
>(tipos '(4 5.1 3+4i))
(E R C)
>(Tipos '(4 5.0 3+4i))
(E E C)
```

2. ¿Cuánto perímetro tengo?

- **Sinopsis:** (periS lados) (periC lados)
- **Característica Funcional:** Recursión simple y de cola.
- **Descripción:** Se debe implementar una función que reciba una lista con los lados de un polígono, debe calcular el perímetro de este utilizando recursión simple y recursión de cola
- **Ejemplos:**

```
>(periS '(3 4 5)
12
>(periC '(3 4 5)
12
```

3. Fusión HA!

- **Sinopsis:** (merge 11 12)
- **Característica Funcional:** Manejo de listas con do while
- **Descripción:** Se debe implementar una función que reciba dos listas ordenadas y realice un merge entre ellas de forma de que retorne una lista con la unión de ambas de manera ordenada.
- **Ejemplos:**

```
>(merge '(4 6 8 10) '(5 7 9 11))
(4 5 6 7 8 9 10 11)
>(merge '(5 6 7 9) '(1 2 8))
(1 2 5 6 7 8 9)
```

4. Árboles binarios de búsqueda

- **Sinopsis:** (abb string entero)
- **Característica Funcional:** TDA Árboles binarios de búsqueda.

- **Descripción:** Se debe implementar una forma de crear árboles binarios de búsqueda que posean tres métodos: agregar, buscar y mostrar. Agregar debe insertar un elemento en su posición correspondiente en el árbol, buscar debe buscar un elemento dentro del árbol retornando `#f` o `#t` y mostrar debe retornar el árbol creado hasta el momento.

Los arboles binarios en scheme se puede representar como una listas de listas que posee la siguiente estructura :

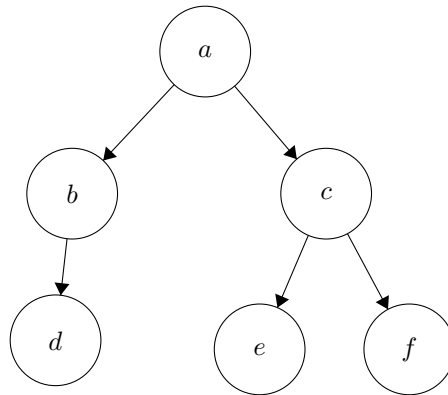
-Un padre y sin hijos se representa como `'(a () ())`

-Un árbol de 3 nodos se representa como `'(a (b () ()) (c () ()))`

Finalmente, un árbol de ejemplo puede ser:

`'(a (b (d () ()) ()) (c (e () ()) (f () ())))`

El cual gráficamente es:



- **Ejemplos:**

```

>(define abb (crear-arbol))
>(abb 'agregar 6)
>(abb 'agregar 7)
>(abb 'agregar 5)
>(abb 'buscar 4)
#f
>(abb 'buscar 5)
#t
>(abb 'mostrar)
(6 (5 () ()) (7 () ()))
  
```

5. Sobre Entrega

- Cada función que **NO** este definida en el enunciado del problema debe llevar una descripción según lo establecido por el siguiente ejemplo:

```
;;(Nombre_función parámetros)  
;;Breve descripción bien explicada.  
;;Que entrega
```
- Se debe trabajar de forma individual.
- Cuidado con el orden y la indentación de su tarea, puede llevar descuentos.
- La entrega debe realizarse en **tar.gz** y debe llevar el nombre:
Tarea4LP_RolIntegrante-1.tar.gz
- El archivo **README.txt** debe contener nombre y rol del alumno e instrucciones para la utilización de su programa en caso de ser necesarias.
- El no cumplir con las reglas de entrega o formato conllevará un máximo de **-30 puntos** en su tarea.
- La entrega será vía aula y el plazo máximo de entrega es hasta el **Día 10 de Julio a las 23:55 hrs.**
- Serán **-10 puntos** por cada hora de atraso.
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.

6. Calificación

- Problema 1: 15 pts
- Problema 2: 25 pts
- Problema 3: 25 pts
- Problema 4: 35 pts
- Código no ordenado (-20 puntos)
- Código no comentado (-5 puntos)
- Reglas de entrega (-30 puntos MAX)