Tarea 4 ILI-253
Lenguajes de Programación
Primer semestre, 2014

Francisco Bórquez
francisco.borquez@usm.cl

Álvaro Hernández
alvaro.hernandezi@alumnos.usm.cl

Teodoro Saavedra
teodoro.saavedra@alumnos.usm.cl

5 de junio de 2014

1. Objetivos

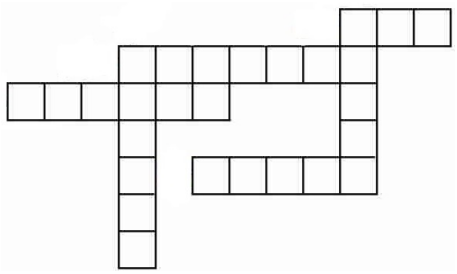
- *Aprender y ejercitar nociones básicas del paradigma funcional.*
- *Comprender el concepto de recursión.*
- *Desarrollar algoritmos de backtracking.*

2. Tarea

Los alumnos deberán programar las distintas funciones en Scheme para la resolución de un crucigrama.

3. Desarrollo

Los alumnos deberán desarrollar un programa que permita resolver un crucigrama, para ello deberán leer el crucigrama vacío desde un archivo llamado `crossword_X.txt`, donde X es un número entero junto con otro archivo llamado `words_X.txt`, el cual contendrá las palabras necesarias para resolverlo. Ejemplo de estos archivos se encuentra en la figura 1 y 2 respectivamente. El archivo `crossword_X` será una lista de listas de caracteres, los cuales son `-` si es un espacio vacio, `*` si es un cuadrado del crucigrama, `r` si es el principio de una palabra que va hacia la derecha, `d` si es el principio de una palabra que va hacia abajo y `b` si son ambos caso. La lista representara un arreglo bidimensional de nxm donde n y m son las dimensiones necesarias para contener el crucigrama y el programa deberá hacer la comprobación de que se cumpla la condición de que las listas sean del mismo largo, además el archivo `words_X` es la lista de palabras que componen la solución del crucigrama.



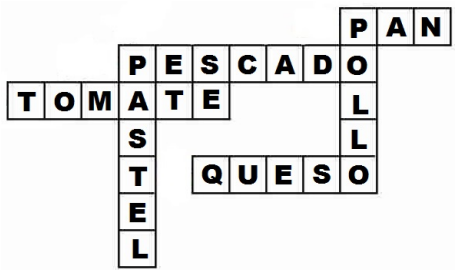
```
'(( - - - - - - - - - b * * )
  ( - - - b * * * * * * - - )
  ( r * * * * * - - - * - - )
  ( - - - * - - - - - * - - )
  ( - - - * - r * * * * - - )
  ( - - - * - - - - - - - - )
  ( - - - * - - - - - - - - ))
```

Figura 1: Ejemplo de un crucigrama pasado a la lista.

```
'( "pescado" "pollo" "pan" "tomate" "queso" "pastel")
```

Figura 2: Ejemplo del archivo con la lista de palabras.

Para resolver el crucigrama deberán aplicar un algoritmo de backtracking. Y llamar a una función llamada `solve(X)` que se encargara de generar la solución y pasarla al archivo de salida `solve_X.txt`, donde X es el mismo número entero que los archivos `crossword_X` y `words_X` como se muestra en la figura 3. La tarea deberá venir con




```
'(( - - - - - - - - - p a n )
  ( - - - p e s c a d o - - )
  ( t o m a t e - - - l - - )
  ( - - - s - - - - - l - - )
  ( - - - t - q u e s o - - )
  ( - - - e - - - - - - - - )
  ( - - - l - - - - - - - - ))
```

Figura 3: Ejemplo de la solución del crucigrama.

al menos un crucigrama, es decir con el archivo `crossword_1.txt` y el archivo `words_1.txt`, además se revisara la tarea con otro crucigrama creado por el ayudante.

4. Requerimientos adicionales

- Para la búsqueda de las soluciones se debe utilizar recursividad.
- La tarea deberá poder ser ejecutada con DrRacket.
- Se debe definir el lenguaje a utilizar en el comienzo de los archivos con la sentencia `"#lang scheme"`.
- Se debe procurar que DrRacket determine el lenguaje desde la fuente para evitar sintaxis y funciones propias de otros dialectos.
- Se acepta el supuesto de que el archivo vendrá con el formato dado, es decir, no es necesario llevar a cabo verificaciones de que el archivo tenga un formato incorrecto distinto al especificado.

5. Archivos a entregar

- crossw_solver.scm
- README.txt
- crossword_1.txt
- words_1.txt

6. Sobre la entrega

- El código debe venir indentado y sin warnings.
- Cada función debe llevar una descripción según lo siguiente:

```
;**** Variables: Nombre_Funcion *****
;Descripcion: : Funcion en cargada de...
;Parametros:
;n1 entero
```

```
;n2 entero
;Retorno: Retorna...
;*********************************/
```

- Se debe trabajar en grupos de dos personas.
- La entrega debe realizarse en un archivo **.tar.gz** y debe llevar el nombre **Tarea4_Rol-1_Rol-2**.
- El archivo README.txt debe contener el nombre y rol de los integrantes del grupo, como también comentarios, si es que lo hay.
- La entrega se debe entregar antes de las **23.55 hrs**. del dia **Jueves 19 de Junio del 2014**.
- Por cada dia de atraso se descontaran 20 puntos.
- Las copias serán evaluadas con nota 0.