

Lenguajes de programación 2016-2

Ejercicio Semanal 1

Noé Salomón Hernández Sánchez
Albert M. Orozco Camacho
C. Moisés Vázquez Reyes

Facultad de Ciencias UNAM

En clase se vieron las siguientes gramáticas:

$$\mathcal{M} := \varepsilon \mid (\mathcal{M}) \mid \mathcal{M}\mathcal{M}$$

$$\mathcal{L} := \varepsilon \mid (\mathcal{L})\mathcal{L}$$

sus respectivas definiciones en Haskell son las siguientes:

```
data M = Em | Par M | ConctM M M
data L = El | ConctP L L
```

1. Ejercicios:

- Investiga qué son las *clases* en Haskell y haz que los tipos `M` y `L` formen parte de la clase `Show`.
- `conctL :: L->L->L`
Toma dos objetos de tipo `L` y los concatena.
- `mToL :: M->L`
Convierte un objeto de tipo `M` en su equivalente de tipo `L`.
- `lToM :: L->M`
Convierte un objeto de tipo `L` en su equivalente de tipo `M`.

Considera el tipo:

```
data Tokens = ParA | ParC | Desc deriving (Show, Eq)
```

el cual servirá para el proceso de análisis sintáctico.

- **lexer :: String->[Tokens]**

Toma una cadena de texto y la descompone en tokens; nota que sólo nos interesa capturar los paréntesis, cualquier otro símbolo deberá ser capturado con el tipo **Desc**.

Considera el tipo:

```
data Pila = Pila Int [Tokens]
```

el cual servirá para representar el juicio *pila*.

- **analiSintc :: Pila->Bool**

Realiza el proceso de análisis sintáctico utilizando los juicios *pila* vistos en clase.

- **esBalanceada :: String->Bool**

Determina si una cadena de texto está balanceada.

- **parserM :: String->M**

Transforma una cadena en su equivalente de tipo M; esta función debe lanzar un mensaje de error cuando la cadena no esté balanceada.

- **parserL :: String->M**

Transforma una cadena en su equivalente de tipo L; esta función debe lanzar un mensaje de error cuando la cadena no esté balanceada.