Autómatas y Lenguajes Formales 2019-2
Facultad de Ciencias UNAM
Ejercicio Práctico 2

Favio E. Miranda Perea Javier Enríquez Mendoza

29 de mayo de 2019
Fecha de entrega: 9 de junio de 2019

El ejercicio debe resolverse en los archivos `tm.hs` y `pda.hs` utilizando el lenguaje de programación Haskell. Conservando las firmas de las funciones como se especifican en este PDF.

Todas las funciones y definiciones deben estar debidamente comentadas.

Todas las definiciones formales necesarias para la realización de este ejercicio se encuentran en las notas de clase.

Autómata de Pila

Un autómata de pila se define como PDA $= \langle Q, \Sigma, \Gamma, \delta, q_o, Z_0, F \rangle$ donde:

- Q Conjunto de estados
- Σ Alfabeto de entrada
- Γ Alfabeto de la pila
- δ Función de transición
- q_0 Estado Inicial
- Z_0 Símbolo inicial de la pila
- F Conjunto de estados finales

Ejercicios

1. Implementar un autómata de pila definiendo el tipo de dato algebraico `Automata` con base en su definición formal.
2. Definir el tipo de dato algebraico `Stack` y las funciones necesarias para que su comportamiento sea el de una pila. De tal forma que se pueda definir el sinónimo

```
type Machine = (Automata,Stack)
```

3. Definir la función `copute` que recibe una `Machine`, una cadena e imprime el procesamiento formal de la cadena con configuraciones. Donde una configuración se representa mediante el siguiente sinónimo

```
type Config = (State, String, Stack)


compute :: Machine -> String -> [[Config]]
```

4. Definir la función `acceptByStack` que recibe una `Machine`, una cadena y dice si la cadena es aceptada por el autómata de pila.

```
acceptByStack :: Machine -> String -> Bool
```

5. Definir la función `acceptByEmptyState` que recibe una `Machine`, una cadena y dice si la cadena es aceptada por el autómata de pila.

```
acceptByState :: Machine -> String -> Bool
```

6. Utilizando el tipo de dato algebraico `Machine` definir el autómata de pila que acepte el lenguaje $L = \{a^n b^m c^k \mid m = n \ o \ m = k\}$ y mostrar la formalización de la cadena *aabbcc*.

Máquina de Turing

Una máquina de Turing se define formalmente como TM $= \langle Q, \Sigma, \Gamma, \delta, q_o, , F \rangle$ donde:

- Q Conjunto de estados
- Σ Alfabeto de entrada
- Γ Alfabeto de la cinta
- δ Función de transición
- q_0 Estado Inicial
- ␣Símbolo en blanco
- F Conjunto de estados finales

Ejercicios

1. Implementar una máquina de Turing definiendo el tipo de dato algebraico `MaqT` con base en la definición formal de una máquina de Turing estándar.
2. Definir la función `copute` que recibe una `MaqT`, una cadena e imprime el procesamiento formal de la cadena con configuraciones. Donde una configuración se representa mediante el siguiente sinónimo

```
type Config = (State, String, Int)
```

   En donde el entero representa la posición de la cabeza de lectoescritura en la cadena.

```
compute :: MaqT -> String -> [Config]
```

3. Definir la función `accept` que recibe una `MaqT`, una cadena y dice si la cadena es aceptada por la máquina de Turing.

```
accept :: MaqT -> String -> Bool
```

4. Definir la función `encode` que recibe una `MaqT` y la codifica para pasarla como entrada de la Máquina Universal.

```
encode :: MaqT  -> String
```

5. Utilizando el tipo de dato algebraico `MaqT` definir la máquina de Turing que acepte el lenguaje $L = \{a^n b^n c^n\}$ y mostrar la formalización de la cadena *aabbcc*

Extras

1. Definir la función `decode` que recibe una `String` representando una máquina codificada y regresa la `MaqT` que representa.

```
decode :: String  -> MaqT
```

Entrega.

- Se entrega antes de las 23:59 horas del día fijado como fecha de entrega.
- El ejercicio debe ser entregado de forma individual o en parejas.
- Si se entrega en equipo solo es necesario que uno de los integrantes envíe el ejercicio.
- Debe incluirse un archivo README.txt con los nombre completo del alumnos, así como comentarios, opiniones o ideas sobre el ejercicio.
- Guardar los archivos requeridos en un directorio que tenga como nombre el número de cuenta de uno de los alumnos. Se entregará éste directorio comprimido.
- Entregarse al correo del ayudante con el asunto [AyLF192-EP02].
- **Cualquier copia o plagio será calificado automáticamente con cero.**