Forma

Descripción generada automáticamente con confianza media

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY**

*Implementación de métodos computacionales*

# ***Evidencia 1***

Vladimir Mendoza

Julián Enrique Espinoza Valenzuela A01254679

20, marzo del 2024

**Lexical\_Analyzer\_Documentation**

**Manual de usuario:**

* Paso 1:

Para poder utilizar el analizador léxico llevado a cabo, es necesario que en primer lugar se importen las librerías que vienen adjuntas en el código proporcionado. Las librerías son las siguientes:

import Data.Char (isSpace, isDigit, isLetter, isAlpha, isAlphaNum)

import Data.List.Split (splitOn)

import Data.List.Utils (replace)

import Data.List (stripPrefix, isPrefixOf, isInfixOf,elemIndex)

import Data.Maybe (catMaybes)

* Paso 2:

Ya teniendo instaladas las librerías anteriormente mencionadas es necesario tener en cuenta que debemos de tener presente dos cosas principalmente: la primera es que debemos contar con un archivo .txt que podamos dar como alimento al programa. Cabe destacar este archivo debe de estar en el mismo directorio que se encuentre su código guardado y debe de estar incluido específicamente en la siguiente parte del código para que todo funcione como debe de ser:

Texto

Descripción generada automáticamente

En la parte donde dice “algebraic\_expresions.txt” usted puede poner el archivo .txt que contenga sus expresiones algebraicas o en su defecto puede utilizar el archivo .txt incluido en la entrega. En segundo lugar, se debe resaltar que si se agrega un archivo .txt distinto al que viene con la entrega se debe asegurar de que venga una expresión algebraica por línea para que todo funcione como se espera.

* Paso 3:

Teniendo lo anterior en cuenta, lo único que resta es especificar los comandos necesarios para compilar el programa y correrlo. En este caso y debido al nombre del archivo serían los siguientes:

-Para compilar:

ghc -o lexycal\_analyzer lexycal\_analyzer.hs

-Para correr el código:

./lexical\_analyzer

**Casos de prueba:**

1. Se da como entrada una expresión algebraica con una variable bien definida y un entero:

Input:



Output:

Texto

Descripción generada automáticamente

1. Se da como entrada una variable mal definida y un número real:

Input:



Output:

Texto

Descripción generada automáticamente

1. Se da como entrada una variable bien definida con números reales no válidos, notaciones exponenciales validas y no validas:

Input:



Output:

Imagen que contiene Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

1. Se da como entrado una variable valida con un numero real y un comentario:

Input:



Output:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

1. Se da como entrada un comentario:

Input:



Output:



**NDFA:**

Diagrama

Descripción generada automáticamente

Descripción de conjuntos de estados:

Variables: {q0, q1, q2}

Enteros y reales: {q0, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12}

Comentarios: {q13, q14, q15}

Operador “+”: {q0, q16, q18}

Operador “-”: {q0, q16, q19}

Operador “=”: {q0, q16, q17}

Operador “\*”: {q0, q16, q20}

Operador “^”: {q0, q16, q22}

Signo “(”: {q0, q16, q23}

Signo “)”: {q0, q16, q24}

**DFA:**

Diagrama

Descripción generada automáticamente

Tabla de transición del DFA:

Imagen que contiene Tabla

Descripción generada automáticamente

Descripción de conjuntos de estados:

Estados muertos: {q2, q9, q18}

Variables: {q0, q1}

Enteros y reales: {q0, q3, q4, q5, q6, q7, q8}

Comentarios: {q10, q11}

Operador “+”: {q0, q12}

Operador “-”: {q0, q3}

Operador “=”: {q0, q13}

Operador “\*”: {q0, q14}

Operador “^”: {q0, q15}

Signo “(”: {q0, q16}

Signo “)”: {q0, q17}