Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Organización de Lenguajes y Compiladores 1
Segundo Semestre 2020
Ing. Manuel Castillo
Aux Huriel Gomez

Proyecto 1 – ML WEB
Manual Técnico

Horacio Ciraiz Orellana
201513758

15 de septiembre del 2020

Índice

Objetivos:

General:

Introducir al técnico al diseño, estructuración, funcionalidades y metodologías de la aplicación

Especifico:

Introducir al técnico a la estructuración de los distintos componentes desarrollados en esta herramienta, así como las distintas funciones, análisis y fases necesarias para entender el desarrollo de esta herramienta

Proyecto ML WEB

Resumen:

La aplicación ML Web es una aplicación desarrollada en lenguaje Python que sirve como un analizador de distintos lenguajes que tiene como fin la detección y eliminación de errores lexicográficos para que el usuario pueda hacer uso de esta y sirva como herramienta para los programas que un usuario pueda desarrollar. Dicha eliminación de errores se hará por medio de la implementación de un Analizador Léxico de estados el cual se encargará de recolectar todos los tokens correctos y crear un archivo limpio de errores el cual se le especificará la ubicación de guardado.

ML Web cuenta con una interfaz minimalista e intuitiva para facilitar el uso a los usuarios, todos y cada uno de los analizadores, así como la interfaz están separados por medio de distintos módulos y clases, por lo que es fácil analizar el código fuente del programa y aislar cualquier sección que sea necesaria para el técnico.

Para el área de Reporte se implementó varias herramientas, Graphviz 2.38 para la realización del reporte JS y la implementación de Paginas HTML para los distintos reportes de errores.

Analizadores

Analizador Léxico:

El analizador Léxico cuenta con 4 módulos de Análisis Léxico con el nombre de LexicoJS.py, LexicoCSS.py, LexicoHTML.py y LexicoExpresion.py en los cuales se manejó dicho análisis por medio de estados los cuales se encargan de reconocer carácter a carácter , dichos estados fueron implementados basándose en ER (Expresiones Regulares) mediante un análisis de los requerimientos del lenguajes y la creación de AFD (Autómatas Finitos Deterministas) para dicha implementación.

Expresiones Regulares:

Las expresiones regulares son patrones utilizados para encontrar determinadas combinaciones de caracteres dentro de una cadena de texto. Proporcionan de una manera flexible la capacidad de buscar y reconocer cadenas de texto

Identificador:

La expresión regular utilizada para reconocer el indentificador es [Letra]+([Letra]|[Numero]|_) * con la cual se procedió a realizar los distintos estados dentro del analizador

Numero:

La expresión regular utilizada para reconocer Numero es [Numero]+(.[Numero]+)| [Numero]+ con la cual se procedió a realizar los distintos estados dentro del analizador

Símbolo:

La expresión regular de los símbolos dado que no reconocen grandes cadenas como tal , si no un carácter en específico su expresión regular es (%|=|;|:| | > | <| (|) | } | { |+| -| .| ,| !| &| | con la cual se procedió a realizar los distintos estados dentro del analizador

Cadena

La expresión regular utilizada para reconocer la Cadena es “(Todo)” |’ (Todo)’ teniendo dos variantes ya que puede ser con comillas dobles o con comillas simples con la cual se procedió a realizar los distintos estados dentro del analizador

Comentarios

La expresión regular utilizada para reconocer Comentario es /*(Todo menos “*”)*/ | // (Todo menos\n) \n con la cual se procedió a realizar los distintos estados dentro del analizador

Analizador Sintáctico:

El analizador sintáctico cuenta con 1 modulo llamado SintacticoExpresion.py el cual se baso en una gramática y se implemento por medio de un método de Pila para hacer las verificaciones correspondientes para la validación de la cadena en un orden correcto.

Gramática:

Una gramática puede describirse como un conjunto limitado de reglas que sirven para describir una secuencia de símbolos que pertenecen a un lenguaje

E := T + E
| T – E
| T
T:= F* T
| F / T
| F
F:=(E)
|Numero
|ID

Gramática Reconocimiento de Operaciones Aritméticas

Reportes

Los distintos reportes con los que cuenta la herramienta se crean ya se de forma automática en algunos casos o por medio de algún evento en la interfaz gráfica

Reporte HTML

Los reportes en formato HTML son reportes que se crean por medio de la acción en el menú de Reportes

1. Reporte Errores JavaScript
2. Reporte Errores CSS
3. Reporte Errores HTML
4. Reporte Resultado RMT

Reporte Imagen Autómata

Este reporte crea de forma automática luego del análisis del JS una imagen con el Grafo del recorrido de ciertas cadenas a través del analizador

Reporte Bitácora

Este reporte crea de forma automática luego del análisis del CSS una bitácora de transiciones que sufrió el analizador CSS.

Autómatas:

AFD JavaScript

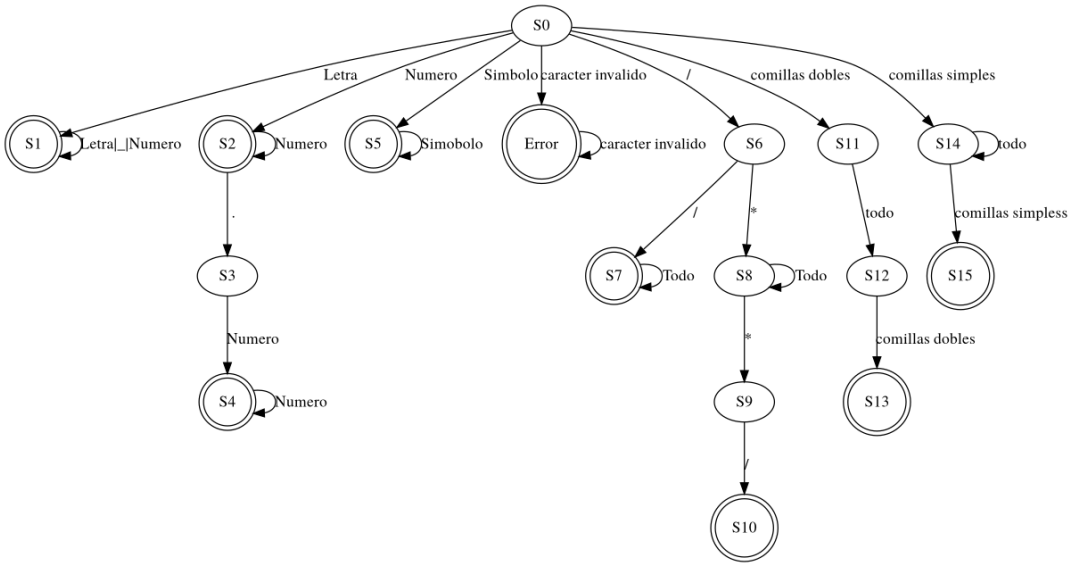

Imagen Autómata JavaScript

AFD HTML

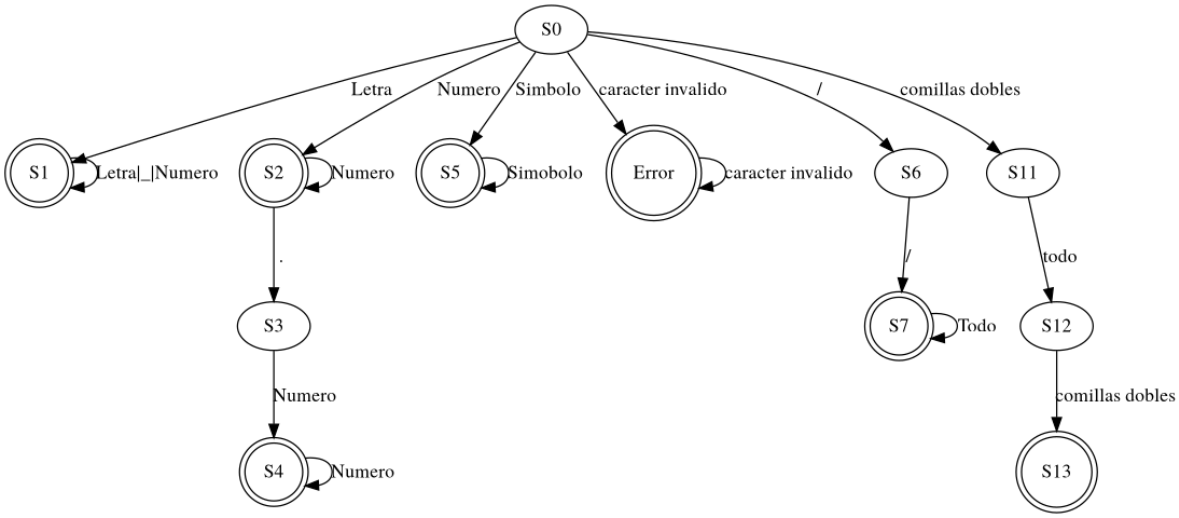

Imagen Autómata HTML

AFD CSS

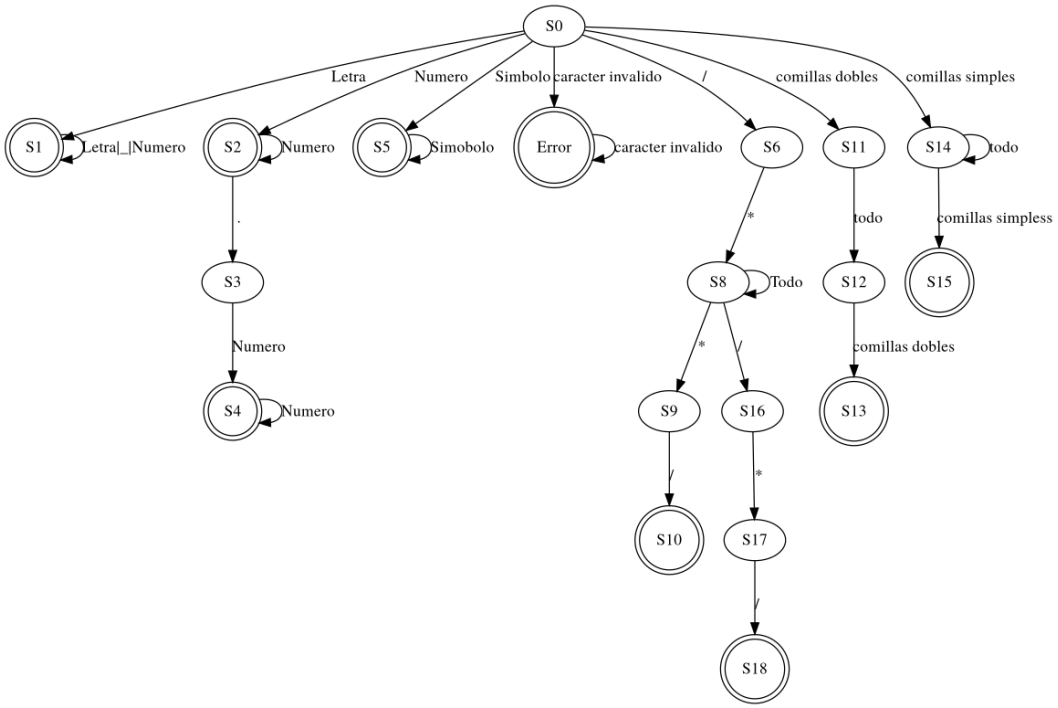

Imagen Autómata CSS

AFD Expresión

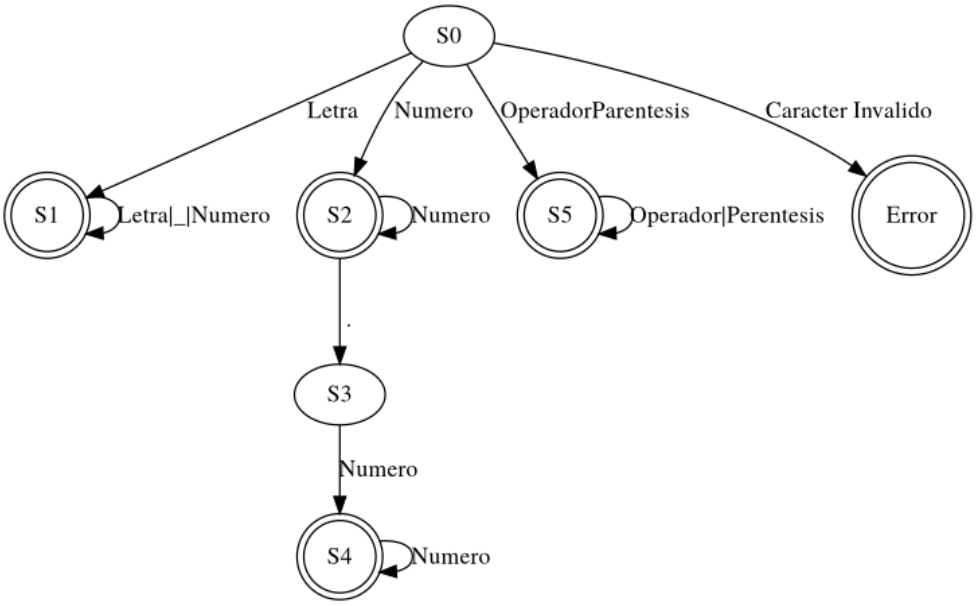

Imagen Autómata Expresión