Laboratorio Lenguajes Formales y de Programación

TRADUCTOR DE HTML

Primer Semestre de 2024

Universidad San Carlos de Guatemala

Programador: Angel Guillermo de Jesús Pérez Jiménez

Carne: 202100215

Correo: 3870961320101@ingenieria.usac.edu.gt

Descripción del Proyecto

El proyecto tiene como objetivo principal desarrollar un sistema de análisis léxico basado en un Autómata Finito Determinista (AFD), que permita procesar archivos de texto con un formato especializado. Este formato incluye instrucciones para la generación de contenido HTML, las cuales serán reconocidas y clasificadas por el AFD durante el proceso de análisis léxico.

El sistema se encargará de leer el texto de entrada, aplicar las reglas definidas en el AFD para reconocer tokens y clasificarlos en diferentes categorías, como palabras clave, números, caracteres especiales y errores. Una vez completado el análisis léxico, el sistema generará un archivo HTML que resuma la información procesada de manera organizada.

El proceso de análisis léxico se llevará a cabo siguiendo las transiciones definidas en el AFD, donde cada estado representa una etapa del análisis y cada transición representa la aceptación de ciertos tokens. Se prestará especial atención a la detección de palabras reservadas, instrucciones específicas para la generación de contenido HTML y la identificación de posibles errores sintácticos o de formato en el texto.

La salida del sistema será un archivo HTML que contendrá tablas con la información detallada sobre las palabras procesadas, los caracteres especiales encontrados y los errores identificados durante el análisis léxico. Además.

Para facilitar el uso del sistema, se proporcionará una interfaz gráfica de usuario (GUI) que permitirá a los usuarios cargar archivos de texto, iniciar el proceso de análisis léxico y visualizar el resultado en forma de archivo HTML generado. La interfaz gráfica será intuitiva y fácil de usar, con opciones para personalizar la apariencia y el formato del archivo de salida.

El proyecto se desarrollará utilizando el lenguaje de programación Python, haciendo uso de sus librerías estándar y herramientas disponibles para el análisis de texto y la generación de contenido HTML. Se seguirán las mejores prácticas de programación y se aplicarán técnicas de diseño modular para garantizar la mantenibilidad, la escalabilidad y la eficiencia del sistema.

Objetivos

Objetivo General:

- Desarrollar un sistema de análisis léxico basado en un Autómata Finito Determinista (AFD) que procese archivos de texto con formato especializado y genere un archivo HTML con información detallada sobre las palabras procesadas, caracteres especiales y errores identificados durante el análisis.

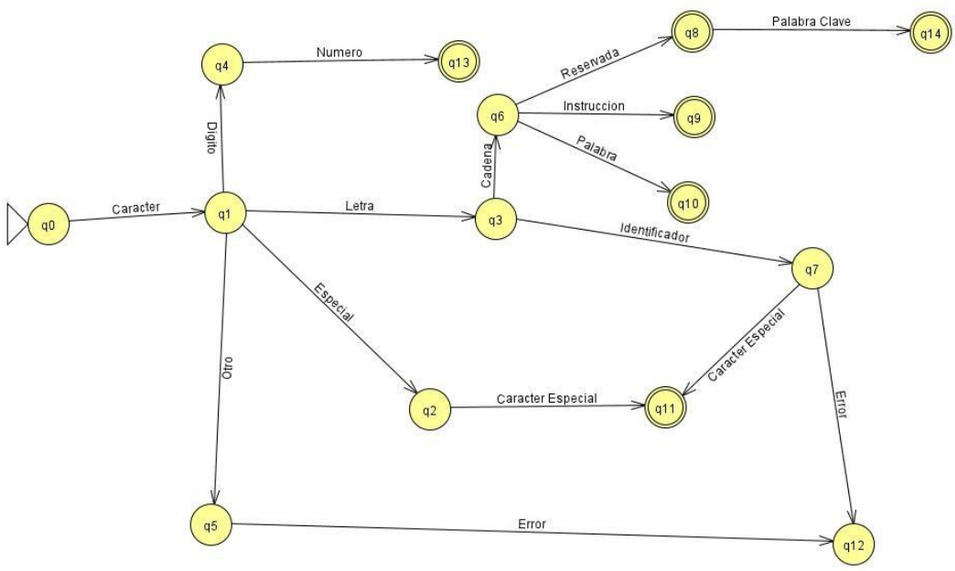
Objetivos Específicos:

- Implementar un analizador léxico que reconozca y clasifique las palabras clave, números, caracteres especiales y errores en el texto de entrada utilizando las reglas definidas en el AFD.
- Diseñar un algoritmo eficiente para la generación de tablas HTML que resuman la información procesada por el analizador léxico.
- Desarrollar una interfaz gráfica de usuario intuitiva que permita a los usuarios cargar archivos de texto, iniciar el análisis léxico y visualizar el resultado en forma de archivo HTML.
- Documentar adecuadamente el código fuente, el proceso de desarrollo y las instrucciones de uso del sistema para facilitar su mantenimiento y su adopción por parte de otros usuarios o desarrolladores.

Estructura del proyecto

```
LFP_S1_2024_PROYECTO1_202100215
    AFD PROYECTO 1 LFP.jpg
    Documentos Prueba
        hola.JSON
        prueba.html
        prueba.txt
    Interfaz
        interfaz.py
    Logica
        analizador_lexico.py
        document_handler.py
        Estructuras.py
        lector.py
    main.py
    README.md
```

Estructura del AFD



Estructura Tipos de Token

```python
class Token:
    def __init__(self, valor, tipo, linea, columna):
        self.valor = valor
        self.tipo = tipo
        self.linea = linea
        self.columna = columna

class Reservada(Token):
    def __init__(self, valor, linea, columna):
        super().__init__(valor, 'RESERVADA', linea, columna)

class Instruccion(Token):
    def __init__(self, valor, linea, columna):
        super().__init__(valor, 'INSTRUCCION', linea, columna)

class Numero(Token):
    def __init__(self, valor, linea, columna):
        super().__init__(valor, 'NUMERO', linea, columna)

class Palabra(Token):
    def __init__(self, valor, linea, columna):
        super().__init__(valor, 'PALABRA', linea, columna)

class CaracterEspecial(Token):
    def __init__(self, valor, linea, columna):
        super().__init__(valor, 'CARACTER_ESPECIAL', linea, columna)

class Error:
    def __init__(self, valor, tipo, linea, columna):
        self.valor = valor
        self.tipo = tipo
        self.linea = linea
        self.columna = columna
```

Estructura de las instrucciones básica

```
Inicio:{
    Encabezado:{
            TituloPagina:"Ejemplo titulo";
        },
    Cuerpo:[
        Titulo:{
            texto:"Este es un titulo";
            posicion:"izquierda";
            tamaño:"t1";
            color:"rojo";
        },
        Fondo:{
            color:"cyan";
        },
        Parrafo:{
            texto:"Este es un parrafo de ejemplo.";
            posicion:"izquierda";
        },
        Texto:{
            fuente="Arial";
            color="azul";
            tamaño="11";
        },
        Codigo:{
            texto:"Muestra el texto con fuente de codigo de ordenador.";
            posicion:"centro";
        },
        Negrita:{
            texto:"Este texto aparecerá en negrita.";
        },
        Subrayado:{
            texto:"Este texto aparecerá Subrayado.";
        },
        Tachado:{
            texto:"Este texto aparecerá tachado.";
        },
        Cursiva:{
            texto:"Este texto aparecerá en cursiva.";
        },
        Salto:{
            cantidad:"5";
        }
    ]
}
```