

Seminario de Solución de Problemas de Traductores de Lenguaje II

Tarea 2:

Mini Generador Léxico

Nombre del maestro:

López Franco Michel Emanuel

Nombre del alumno:

Ramos Calderón Christian Daniel

Código del alumno:

216577014

Índice

Contents

Índice2

Introducción3

Desarrollo.....3

Capturas de pantalla6

Conclusión8

Bibliografía8

Introducción

Durante esta actividad pasamos a la práctica del analizador léxico generando uno mini, para después pasar a desarrollar uno contemplando lo que el profesor indique.

Desarrollo

Sabemos que el analizador empieza tomando caracteres y analizando si es una letra o un número, para después pasar a analizar la palabra completa y de esta manera saber si es de algún tipo o simplemente es un identificador, para esta actividad solo detectaremos lo que son **identificadores**, **enteros** y **reales**.

Retomando lo mencionado anteriormente va analizando si es numero o letra y por ultimo el espacio que los separa, para así separar la palabra o el número completo esto lo hice mediante un **while** el cual valida que no haya terminado la línea ingresada y que no esté en el estado terminado, después dentro de ese while tengo otro el cual permite evaluar cada carácter de la línea, considerando los espacios, puntos para los reales, enteros y si es un identificador este pueda tener números en su nombre, todo esto está dentro de una función también para poder separar las partes del código de tal manera que sea más fácil de digerir, el cual es la siguiente:

```
void analizar(string palabra)
{
    string frase = palabra;
    int indice = 0;
    int estado = 0;
    int estadoFinal = -1;
    string lexema = "";
    string token;

    while ((indice <= (frase.Length - 1)) && (estadoFinal == -1))
    {
        lexema = " ";
        token = "error";
        while ((indice <= (frase.Length - 1)) && (estadoFinal != 25))
        {
            if (estadoFinal == -1)
            {
                if (char.IsWhiteSpace(frase[indice]))
                {
                    estadoFinal = -1;
                }
                else if (char.IsLetter(frase[indice]) || frase[indice] == '_')
                {
                    estado = 0;
                    estadoFinal = estado;
                    lexema += frase[indice];
                    token = "identificador";
                }
                else if (char.IsDigit(frase[indice]))
                {
                    estado = 1;
                    estadoFinal = estado;
                    lexema += frase[indice];
                    token = "entero";
                }
            }
        }
    }
}
```

4 | Seminario de Solución de Problemas de Traductores de Lenguaje II

```
    }
    else
    {
        estadoFinal = 25;
        lexema = frase[indice].ToString();
        token = "error";
    }
    indice++;
}
else if (estadoFinal == -1)
{
    estadoFinal = 25;
}
else if (estadoFinal == 0)
{
    if (char.IsLetter(frase[indice]) || frase[indice] == '_')
    {
        estado = 0;
        estadoFinal = estado;
        lexema += frase[indice];
        token = "identificador";
        indice++;
    }
    else if (char.IsDigit(frase[indice]))
    {
        estado = 0;
        estadoFinal = estado;
        lexema += frase[indice];
        token = "identificador";
        indice++;
    }
    else
    {
        estadoFinal = 25;
    }
}
else if (estadoFinal == 1)
{
    if (char.IsDigit(frase[indice]))
    {
        estado = 1;
        estadoFinal = estado;
        lexema += frase[indice];
        token = "entero";
        indice++;
    }
    else if (frase[indice] == '.')
    {
        estado = 24;
        estadoFinal = estado;
        lexema += frase[indice];
        token = "punto";
        indice++;
    }
    else
    {
        estadoFinal = 25;
    }
}
else if (estadoFinal == 24)
{
    if (char.IsDigit(frase[indice]))
```

5 | Seminario de Solución de Problemas de Traductores de Lenguaje II

```
        {
            estado = 2;
            estadoFinal = estado;
            lexema += frase[indice];
            token = "Numero real";
            indice++;
        }
        else
        {
            estadoFinal = 25;
        }
    }

    }
    tabla.Rows.Add(estado, lexema, token);
    estadoFinal = -1;
}
```

Las variables son usadas para lo siguiente:

- string frase
La frase como tal que es rescatada del cuadro de texto la cual vamos a analizar
- int indice
indice indica en que carácter estamos analizando para identificar que es
- int estado
estado es para indicar en que estado se encuentra si es letra o numero y de esta manera identificar si es un real o si el identificador tiene números
- int estadoFinal
Con esta variable indicamos que ha llegado al final de la palabra que estamos analizando o en su defecto el número, si es el caso que hemos terminado de leer toda la línea este regresara su valor a -1 y en el if validamos que el indice es menor que el tamaño de la cadena, siendo que es mayor y que el estadoFinal es -1 terminara y saldrá del while
- string lexema
Es donde guardamos la palabra que se analizo completa sin espacios y solo esa palabra
- string token
Sirve para indicar que tipo es, siendo entero, real o un identificador

Capturas de pantalla

Form1

hola

Analizar

	ID	Lexema	Token
▶	0	hola	identificador
*			

Form1

1.1

Analizar

	ID	Lexema	Token
▶	0	hola	identificador
	2	1.1	Numero real
*			

7 | Seminario de Solución de Problemas de Traductores de Lenguaje II

Form1

[

Analizar

	ID	Lexema	Token
▶	0	hola	identificador
	2	1.1	Numero real
	0	[error
*			

Form1

5

Analizar

	ID	Lexema	Token
▶	0	hola	identificador
	2	1.1	Numero real
	0	[error
	1	5	entero
*			

Conclusión

Para finalizar con esta práctica básicamente fue para entrar en calor, recordando como manejar el lenguaje que anteriormente utilizaba aun que es algo básico sirve para volver a reforzar los conocimientos y como funcionan los compiladores mediante la separación del analizador léxico y el sintáctico.

Bibliografía

No valida en este trabajo