

Proyecto fase final

Entrega: 25 de abril 2020

Generador de scanner

- La fase final del proyecto tiene como objetivo generar en lenguaje C# un scanner que reconozca los tokens que hemos definido en el archivo de entrada en el formato .
- A partir de calculo de first, last y la tabla del follow, se construyó la tabla de transiciones y Ahora construiremos el **scanner**.
- El generador sigue recibiendo como entrar el archivo con conjuntos, tokens, acciones y error, y la salida debe ser un programa en C# (6 pts) que al ser compilado y ejecutado, permite leer un archivo de entrada y al ser procesado, detecta si los tokens pertenecen o no al lenguaje descrito(4 pts).

Diagrama del proceso

Archivo de entrada
(conjuntos,tokens,acciones)



```
graph TD; A[Archivo de entrada  
(conjuntos,tokens,acciones)] --> B[Generador de scanner]; B --> C[Archivo en C#]; C --> D[Compilar C# y generar .exe]; D --> E[El .exe recibe como entrada un archivo con  
tokens que pertenecen al lenguaje];
```

The diagram illustrates a five-step process. The first two steps are in grey boxes, and the last three are in green boxes. Arrows indicate the flow from top to bottom.

Generador de scanner

Archivo en C#

Compilar C# y generar .exe

El .exe recibe como entrada un archivo con
tokens que pertenecen al lenguaje

Ejemplo (archivo de entrada)

- SETS

```
letra='A'..'Z'+'a'..'z'+'_'  
digito='0'..'9'
```

TOKENS

```
token 1=digito digito*
```

```
token 2('='
```

```
token 3=':'
```

```
token 4=letra(letra | digito)*
```

ACTIONS

```
reservadas()
```

```
{
```

```
5='PROGRAM'
```

```
6='INCLUDE'
```

```
7='CONST'
```

```
8='TYPE'
```

```
}
```

```
ERROR= 9
```

- En base al archivo de entrada usted ya debe de generar sus first, last, follow y finalmente su AUTOMATA (Tabla de transiciones), en base a esta tabla de transiciones debe generar código en C# que permita devolver el número de token según la cadena de entrada, a continuación un ejemplo en pseudocódigo .
- Para el archivo ejemplo (archivo 19 con el que se califico la fase 1) que genera 5 estados.
- Le puede ayudar pasar la siguiente expresión regular a AFD

`((digito digito *) | (=) | (:=) | (letra (letra | digito)*))#`

Con su automata generado podra verificar que son 5 estados generados los cuales deberá programar para que devuelva el token deseado

```

set
    letra='A'..'Z'+'a'..'z'+'_'
    digito='0'..'9'

tokens

    token 1=digito digito*
    token 2='='
    token 3=':'='"='
    token 4=letra(letra|digito)*

Actions
    reservadas()
    {
        5='PROGRAM'
        6='INCLUDE'
        7='CONST'
        8='TYPE'
    }
ERROR= 9

```

***** cadena[i]==10 , donde 10 se esta comparando como ascii.

```

switch (Estado) {
    case 1 :
        if (cadena[i]==9 || cadena[i]==10 || cadena[i]==13 ||
            cadena[i]==26 || cadena[i]==32)
        {
            if (cadena[i]==13) then    i++;
        }
        else
            if ((cadena[i]>=48 && cadena[i]<=57))
                Estado = 2;
            else
                if ((cadena[i]==61))
                    Estado = 3;
                else
                    if ((cadena[i]==58))
                        Estado = 4;
                    else
                        if ((cadena[i]>=97 && cadena[i]<=122)
                            ||(cadena[i]==95)
                            ||(cadena[i]>=65 && cadena[i]<=90))
                            Estado = 5;
                        else
                        {
                            Error = true;
                            Salir = true;
                        };
                    break;
    case 2 :
        if ((cadena[i]>=48 && cadena[i]<=57))
            Estado = 2;
        else
        {
            Retroceso();
            Salir = true;
        };
        break;
};

```

conjuntos

letra='A'..'Z'+ 'a'..'z'+ '_'

digito='0'..'9'

tokens

token 1=digito digito*

token 2='='

token 3=':'='"'

token 4=letra(letra|digito)* {reservadas() }

acciones

reservadas()

{
5='PROGRAM'

6='INCLUDE'

7='CONST'

8='TYPE'

}

ERROR= 9

** se debe validar si es un token aceptado que no este contenido dentro de las reservadas sino el token a devolver es el de las reservadas.

```
case 4 :  
    if ((cadena[i]==61))  
        Estado = 6;  
    else  
    {  
        Error = true;  
        Salir = true;  
    };  
    break;
```

```
case 5 :  
    if ((cadena[i]>=97 && cadena[i]<=122)  
        ||((cadena[i]==95)  
        ||(cadena[i]>=65 && cadena[i]<=90)  
        ||(cadena[i]>=48 && cadena[i]<=57)))  
        Estado = 5;  
    else  
    {  
        Retroceso();  
        Salir = true;  
    };  
    break;
```

/*Case para estado y sus correspondiente no. de token*/

```
switch (Estado) {  
    case 2 : NumToken = 1;break;  
    case 3 : NumToken = 2;break;  
    case 5 : NumToken = 4;break;  
    case 6 : NumToken = 3;break;  
    default : NumToken = 9;// el error  
        Error = true;
```

```
};
```

Ejecución del exe

- Luego al compilar el C# generado , el archivo de entrada podría ser:

Archivo de entrada:

```
Program x a :=b c=d const a
```

- Y la salida debería ser:

Archivo de salida o
mostrar en pantalla:

```
Program = 5  
x=4  
a=4  
:= = 3  
b=4  
c=4  
==2  
const=7  
a=4
```