

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE SISTEMAS



Lenguajes Formales y de Programación 1
Escuela de vacaciones de junio 2022

Proyecto #1: Simple C

Una empresa reconocida solicita sus servicios para desarrollar un lenguaje de programación similar al lenguaje de programación C, pero con una sintaxis simplificada con el fin de no utilizar el lenguaje anterior mencionado. Dicho lenguaje se llamará Simple C o SC.

La empresa no requiere de un lenguaje funcional por el momento, únicamente solicita un analizador léxico capaz de reportar si un código fuente pertenece o no a dicho lenguaje y los tokens que el analizador léxico fue capaz de reconocer.

Aplicación

Usted desarrollará una aplicación de línea de comandos en Python. Al ejecutar su aplicación esta solicitará los siguientes parámetros:

- Archivo del código fuente (.sc)
- Nombre de archivo de reporte (.html)

Analizador léxico

La implementación del analizador léxico queda a su discreción, pero se necesita que el reconocimiento de patrones se haga mediante AFD. Los AFD serán creados desde una expresión regular mediante el método del árbol.

Importante: su aplicación será encargada de validar los lexemas del código fuente, en ningún momento tendrá en cuenta el orden ni el resultado del código fuente.

Lenguaje

Resumidamente, las funciones de este lenguaje serán:

- El lenguaje será case insensitive
- Declaración y asignación de variables
- Operaciones aritméticas, lógicas y relacionales
- Estructuras condicionales
- Estructuras iterativas
- Creación de funciones y métodos con y sin parámetros
- Retorno de funciones y métodos
- Llamado de funciones y métodos

La sintaxis detallada se presenta a continuación.

Comentarios

El lenguaje soporta comentarios de una línea y multilínea, de la siguiente manera:

```
// comentario de una línea
```

```
/*
```

```
    Comentario  
    De varias  
    Líneas
```

```
*/
```

Tipos de datos

El lenguaje contará con los siguientes tipos de datos:

Tipo de dato	Ejemplo
Int	67
Double	39.87
String	"Hola mundo"
Char	'a'
Boolean	true

Identificadores

Los identificadores del lenguaje deben tener la siguiente estructura:

```
([UNA LETRA O GUIÓN BAJO])([CERO O MÁS LETRAS, GUIONES BAJOS O NÚMEROS])
```

Operaciones

El lenguaje cuenta con varias operaciones aritméticas, lógicas y relacionales.

Operación	Operador	Ejemplo
Suma	+	7 + 8
Resta	-	5 - 90
Multiplicación	*	17 * 0
División	/	4 / 3
Resto	%	91 % 2
Igualación	==	4 == 5
Diferenciación	!=	7 != 7
Mayor	>	8 > 0
Mayor igual	>=	-9 >= -90
Menor	<	87 < 16
Menor igual	<=	77 <= 4
AND	&&	true && true
OR		true false
NOT	!	! false

Queda fuera del alcance de la aplicación validar los operandos de las operaciones.

Instrucciones

Las instrucciones pueden estar tanto dentro como fuera de funciones y métodos. Las instrucciones del lenguaje son:

- Declaración y asignación de variables
- Estructuras condicionales
- Estructuras iterativas
- Declaración de métodos y funciones
- Retorno de métodos y funciones
- Llamada de métodos y funciones

Declaración y asignación de variables

Podemos asignar datos a variables de la siguiente manera:

```
// declaración  
[TIPO_DATO] [ID] = [DATO] ;  
  
// asignación  
[ID] = [DATO] ;
```

Estructuras condicionales

El lenguaje contará con las siguientes estructuras condicionales:

```
// sentencia if  
if ( [OPERACION] ) {  
    // INSTRUCCIONES  
}  
  
// sentencia if else  
if ( [OPERACION] ) {  
    // INSTRUCCIONES  
} else {  
    // INSTRUCCIONES  
}
```

Estructuras iterativas

El lenguaje contará con las siguientes estructuras iterativas:

```
// sentencia while
while ( [OPERACION] ) {
// INSTRUCCIONES
}

// sentencia do-while
do {
// INSTRUCCIONES
} while ( [OPERACION] );
```

Funciones y métodos

Las funciones y métodos se declararán de la siguiente manera:

```
// método
void [ID] ( [PARAMETROS] ) {
// INSTRUCCIONES
}

// función
[TIPO_DATO] [ID] ( [PARAMETROS] ) {
// INSTRUCCIONES
}

// parámetros
( [PARAMETRO] , ) *
```

```
// parámetro  
[TIPO_DATO] [ID]
```

Retorno de funciones y métodos

Las funciones pueden retornar de la siguiente manera:

```
// método  
return ;  
  
// función  
return [DATO] ;
```

Llamada a funciones o métodos

Las funciones y métodos se llamarán de la siguiente manera:

```
// llamada  
[ID] ( [PARAMETROS] )  
// parámetros  
( [PARAMETRO] , ) *  
  
// parámetro  
[DATO]
```

Reportes

Al finalizar la aplicación se deberá guardar un archivo con el nombre provisto en la línea de comandos. El archivo será un archivo HTML a su creatividad con las siguientes tablas:

Reporte de tokens

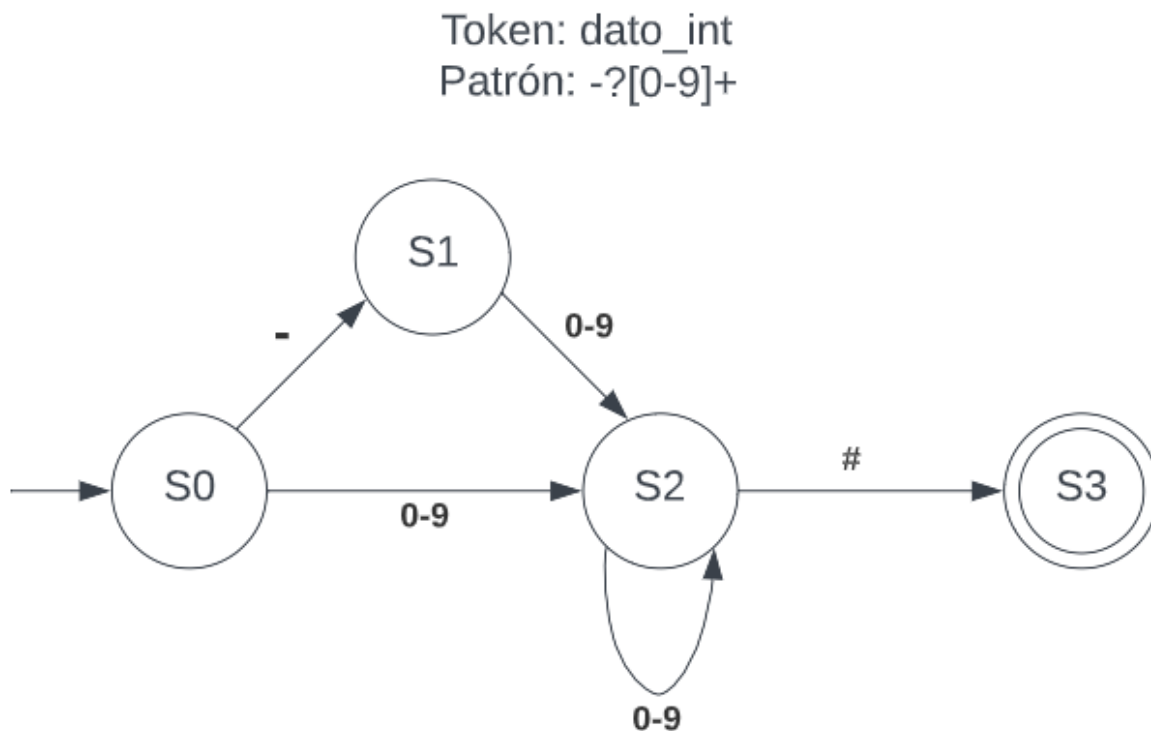
Este reporte contendrá los lexemas con su token correspondiente, así como el patrón de dicho token. Ejemplo:

Línea	Columna	Lexema	Token	Patrón
3	1	55	dato_int	-?[0-9]+

Reporte de AFD

Este reporte contendrá la máquina de estados del AFD por cada lexema reconocido. Ejemplo:

Teniendo en cuenta la siguiente máquina de estados:



Lexema: 55. Token: dato_int

Estado	Caracter	Lexema reconocido	Siguiente estado
S0	5		S2
S2	5	5	S2
S2	#	55	S3 (aceptación)

No es necesario incluir el diagrama en el reporte.

Reporte de errores

Línea	Columna	Lexema
3	1	55

Manuales

Se requiere la creación de un manual de usuario y un manual técnico, ambos en formato Markdown dentro del repositorio.

El manual de usuario deberá guiar a un usuario en el uso de la aplicación de línea de comandos.

El manual técnico deberá contener una explicación técnica del desarrollo de su aplicación, así como el proceso de creación de mínimo dos AFD por el método del árbol.

Consideraciones

- La aplicación deberá ser desarrollada en Python sin ningún framework ni librería de terceros. La librería Regex queda prohibida.
- La aplicación debe ser amigable con el usuario.
- Para la calificación se utilizarán archivos de código fuente escritos por su auxiliar, se recomienda la creación de archivos de código fuente propios para probar todas las funcionalidades.
- Se deberá trabajar en un repositorio privado en GitHub e invitar al auxiliar de colaborador: pabloc54.
- El nombre del repositorio debe ser: LFP-VJ-carnet-P1
- Copias totales o parciales serán detectadas y reportadas con la escuela para su sanción correspondiente.

Entrega

- La fecha de entrega es: Viernes 17 de junio de 2022 a las 23:59.
- La entrega es individual.
- Se deberá entregar el enlace al repositorio en Uedi.
- Se calificará únicamente lo subido al repositorio hasta la fecha de entrega.
- El proyecto será calificado en su computador, tener preparado todo al momento de la calificación pues no se darán nuevos horarios.
- Se tendrá un tiempo de calificación de 15 min por alumno.
- Entregas tarde tendrán una nota de cero puntos.