

Curso: Compiladores

Semestre: I - 2016

Facilitador: Wider Farid Sánchez Garzón

Instrucciones: A las 7:40 p.m. se deberán tener listos los siguientes entregables, con el fin de tener inmediatamente la revisión y la nota respectiva:

1. Diagrama de transiciones
2. Compilador implementado que acepte:
 - a. Ingreso de programa de entrada por consola o a través de la carga de un archivo.
 - b. Carga del programa de entrada a la caché para disponerlo de ésta manera para el análisis.
 - c. Analizador léxico
 - d. Tabla de símbolos con resultado final del proceso de compilación.
 - e. Tabla de errores con el resultado de analizar el análisis léxico.
 - f. Cuando termine el análisis, el compilador debe mostrar un mensaje indicando si el programa está bien escrito o no.

Compilador extraclase

Construya un compilador que permita reconocer el siguiente lenguaje:

- **Opción 1:**

PARA (<identificador> **DESDE** <limiteInicial> **HASTA** <limiteFinal> **[EN INCREMENTOS DE <valor> EN CADA PASO]** **HACER**
<sentencias>

FIN PARA

Dónde:

- Las sentencias encerradas en corchetes ([...]) son opcionales e implican que pueden o no estar presentes. Tenga en cuenta que los corchetes como tal no aparecen, y sólo indican que el contenido que se encuentra dentro de ellos puede o no aparecer. Si se encuentra el contenido dentro de la sentencia, deben cumplir la regla de formación completa descrita dentro de los mismos.
- <identificador>: Es una variable válida. Ver regla de formación de identificadores válidos.
- <limiteInicial>, <limiteFinal>, <valor>: Pueden ser un número entero o un número decimal válido. Ver regla de formación de números válidos.
- <sentencias>: Pueden presentarse sentencias que cumplan con las siguientes reglas:
 - <identificador> := <expresion>;
 - **ESCRIBIR**(<expresion>;
 - **LEER**(<identificador>;
- <expresion>: Puede ser una expresión de cualquiera de los siguientes tipos:
 - <valor><operador><valor>.
- <operador>: Puede ser:
 - +
 - -

- *
- /
- %
- <
- <=
- >
- >=
- <>
- !=
- =
- Y
- O
- <valor>: Puede ser:
 - Un número entero
 - Un número decimal
 - Una variable

Se tiene un lenguaje donde se tiene definido que el léxico válido debe cumplir las siguientes características:

- Acepta identificadores que cumplan con las siguientes condiciones:
 - No comienza por un número
 - No puede contener caracteres especiales.
 - Contiene letras mayúsculas y minúsculas sin acentuación.

A continuación se muestra ejemplo de identificadores válidos e inválidos:

Identificador	Categoría
casa	IDENTIFICADOR
\$identificador	ERROR
_mi\$variable	ERROR
_123_Variable\$	ERROR
123_variables	ERROR
¿variables?	ERROR
variable#	ERROR
Casa123	IDENTIFICADOR

- Acepta números enteros y decimales, cuyo separador decimal es la coma (,). A continuación se muestra ejemplo de números válidos e inválidos:

Número	Categoría
25	NUMERO ENTERO
26,0	NUMERO DECIMAL
36,2	NUMERO DECIMAL
48,54	NUMERO DECIMAL

48151,	ERROR
.56	ERROR
48.54,56	ERROR

- Acepta operadores matemáticos de suma (+), resta (-), multiplicación (*), división (/), módulo (%). A continuación se muestra ejemplo de operadores matemáticos válidos:

Operador matemático	Categoría
+	SUMA
-	RESTA
*	MULTIPLICACION
/	DIVISIÓN
%	MÓDULO

- Acepta el operador de asignación (:=). A continuación se muestra ejemplo de operador de asignación válido:

Operador matemático	Categoría
:=	ASIGNACIÓN

- Acepta operadores de comparación mayor (>), menor (<), igual (=), mayor o igual (>=), menor o igual (<=), diferente (<>). A continuación se muestra ejemplo de operadores de comparación válidos:

Operador de comparación	Categoría
>	MAYOR QUE
<	MENOR QUE
=	IGUAL QUE
>=	MAYOR O IGUAL QUE
<=	MENOR O IGUAL QUE
<>	DIFERENTE QUE
!=	DIFERENTE QUE

- Acepta operadores de lógicos AND y OR. A continuación se muestra ejemplo de operadores de comparación válidos:

Operador de comparación	Categoría
Y	AND
O	OR

- Acepta como operadores de agrupación los paréntesis "(" y ")". A continuación se muestra ejemplo de operadores de comparación válidos:

Operador de agrupación	Categoría
(PARENTESIS ABRE
)	PARENTESIS CIERRA

- Acepta los siguientes caracteres sobre el archivo leído y los interpreta:

Carácter especial	Categoría
Salto de línea	SALTO LINEA
Fin del archivo	EOF