Diseño de Compiladores I – Cursada 2023
Trabajo Práctico Nro. 1

La entrega se hará en forma conjunta con el Trabajo Práctico Nro. 2. Fecha de Entrega: A definir

Objetivo

Desarrollar un Analizador Léxico que reconozca los siguientes tokens:

- Identificadores cuyos nombres pueden tener hasta 20 caracteres de longitud. El primer puede ser una letra o '_', y el resto pueden ser letras, dígitos y "_". Los identificadores con longitud mayor serán truncados y esto se informará como Warning. Las letras utilizadas en los nombres de identificador sólo pueden ser minúsculas.
- Constantes correspondientes al tema particular asignado a cada grupo.
  ***Nota*:** Para aquellos tipos de datos que pueden llevar signo, la distinción del uso del símbolo '-' como operador aritmético o signo de una constante, se postergará hasta el trabajo práctico Nro. 2.
- Operadores aritméticos: "**+**", "**-**" , "*****", "**/**" agregando lo que corresponda al tema particular.
- Operador de asignación: "**=**"
- Comparadores: "**>=**", "**<=**", "**>**", "**<**", "**==**", "!!"
- "**{**", "}", "**(**", "**)**","**,**","**.**" y "**;**"
- Cadenas de caracteres correspondientes al tema particular de cada grupo.
- Palabras reservadas (en mayúsculas):
  IF, ELSE, END_IF, PRINT, CLASS, VOID
- y demás símbolos / tokens indicados en los temas particulares asignados a cada grupo.

El Analizador Léxico debe eliminar de la entrada (reconocer, pero no informar como tokens al Analizador Sintáctico), los siguientes elementos.

- Comentarios correspondientes al tema particular de cada grupo.
- Caracteres en blanco, tabulaciones y saltos de línea, que pueden aparecer en cualquier lugar de una sentencia.

Analizador Léxico. Especificaciones

a) El Analizador Léxico deberá leer un código fuente, identificando e informando:
   - Tokens detectados en el código fuente. Por ejemplo:
     Palabra reservada **IF**
     (
     Identificador **var_x**
     +
     Constante entera **25**
     Palabra reservada **ELSE**
     etc.
   - Errores léxicos detectados en el código fuente, indicando: nro. de línea y descripción del error. Por ejemplo:
     Línea 24: Constante entera fuera del rango permitido
   - Contenidos de la Tabla de Símbolos.

Se sugiere la implementación de un consumidor de tokens que invoque al Analizador Léxico solicitándole tokens. En el trabajo práctico 2, esta funcionalidad estará a cargo del Analizador Sintáctico.

b) El código fuente **debe ser leído desde un archivo**, cuyo nombre **debe poder ser elegido** por el usuario del compilador. Se espera que el compilador pueda ejecutarse desde línea de comandos.

c) La numeración de las líneas de código debe comenzar en 1. De este modo, la información de cada error coincidirá con el número de línea en el archivo del código fuente.

d) Para la programación se podrá elegir el lenguaje. Para esta elección, tener en cuenta que el analizador léxico se integrará luego a un Parser (Analizador Sintáctico) generado utilizando una herramienta tipo Yacc. Por lo tanto, es necesario asegurarse la disponibilidad de dicha herramienta para el lenguaje elegido.

e) El Analizador Léxico deberá implementarse mediante una matriz de transición de estados y una matriz de acciones semánticas, de modo que cada cambio de estado y acción semántica asociada, sólo dependa del estado actual y el carácter leído.

f) Implementar una Tabla de Símbolos donde se almacenarán identificadores, constantes, y cadenas de caracteres. Es requisito para la aprobación del trabajo, que la tabla sea implementada con una estructura dinámica.

g) La aplicación deberá mostrar, además de tokens y errores léxicos, los contenidos de La Tabla de Símbolos.

Entrega

La entrega se pactará con el docente asignado al grupo. La asignación se encuentra publicada junto con los temas particulares.
El material entregado debe incluir:

- Ejecutable del compilador
- Código fuente completo del compilador
- Casos de prueba
- Informe

***Informe*:**

Debe incluir:
- **NRO. DE GRUPO** e Integrantes. Incluir **DIRECCIONES DE CORREO** para contacto.
- Temas particulares asignados (esta información deberá repetirse en los informes de los trabajos prácticos subsiguientes).
- Introducción.
- Decisiones de diseño e implementación.
- Diagrama de transición de estados.
- Matriz de transición de estados.
- Descripción del mecanismo empleado para implementar la matriz de transición de estados y la matriz de acciones semánticas.
- Lista de acciones semánticas asociadas a las transiciones del autómata del Analizador Léxico, con una breve descripción de cada una.
- Errores léxicos considerados.

Este informe deberá ser completado con las consignas indicadas en el Trabajo Práctico 2.

Casos de Prueba

Se debe incluir, como mínimo, ejemplos que contemplen las siguientes alternativas:
(Cuando sea posible, agregar un comentario indicando el comportamiento esperado del compilador)

- Constantes con el primer y último valor dentro del rango (Para cada tipo de datos asignado).
- Constantes con el primer y último valor fuera del rango (Para cada tipo de datos asignado).
- Para números de punto flotante: parte entera con y sin parte decimal, parte decimal con y sin parte entera, con y sin exponente, con exponente positivo y negativo.
- Identificadores de menos y más de 25 caracteres.
- Identificadores con letras, dígitos y "_".
- Intento de incluir en el nombre de un identificador un carácter que no sea letra, dígito o “-“.
- Palabras reservadas escritas en minúsculas y mayúsculas.
- Comentarios bien y mal escritos.
- Cadenas bien y mal escritas.

Temas particulares

Cada grupo de trabajo tendrá asignada una combinación de temas particulares.
La información de los temas asignados a cada grupo, estará disponible en el Aula Virtual de la materia.

1. **Enteros cortos (8 bits):** Constantes enteras con valores entre -2^7 y $2^7 - 1$. Estas constantes llevarán el sufijo "**_s**".

   Enteros sin signo (16 bits): Constantes con valores entre 0 y $2^{16} - 1$. Estas constantes llevarán el sufijo "**_ui**".

   Se deben incorporar a la lista de palabras reservadas las palabras **SHORT** y **UINT**.

2. **Enteros cortos (8 bits):** Constantes enteras con valores entre -2^7 y $2^7 - 1$. Estas constantes llevarán el sufijo "**_s**".

   Enteros largos sin signo (32 bits): Constantes enteras con valores entre 0 y $2^{32} - 1$. Estas constantes llevarán el sufijo "**_ul**".

   Se deben incorporar a la lista de palabras reservadas las palabras **SHORT** y **ULONG**

3. **Enteros (16 bits):** Constantes enteras con valores entre -2^{15} y $2^{15} - 1$. Estas constantes llevarán el sufijo "**_i**".

   Enteros cortos sin signo (8 bits): Constantes enteras con valores entre 0 y $2^8 - 1$. Estas constantes llevarán el sufijo "**_us**".

   Se deben incorporar a la lista de palabras reservadas las palabras **INT** y **USHORT**

4. **Enteros (16 bits):** Constantes enteras con valores entre -2^{15} y $2^{15} - 1$. Estas constantes llevarán el sufijo "**_i**".

   Enteros largos sin signo (32 bits): Constantes enteras con valores entre 0 y $2^{32} - 1$. Estas constantes llevarán el sufijo "**_ul**".

   Se deben incorporar a la lista de palabras reservadas las palabras **INT** y **ULONG**

5. **Enteros largos (32 bits):** Constantes enteras con valores entre -2^{31} y $2^{31} - 1$. Estas constantes llevarán el sufijo "**_l**".

   Enteros cortos sin signo (8 bits): Constantes enteras con valores entre 0 y $2^8 - 1$. Estas constantes llevarán el sufijo "**_us**".

   Se deben incorporar a la lista de palabras reservadas las palabras **LONG** y **USHORT**

6. **Enteros largos (32 bits):** Constantes enteras con valores entre -2^{31} y $2^{31} - 1$. Estas constantes llevarán el sufijo "**_l**".

   Enteros sin signo (16 bits): Constantes con valores entre 0 y $2^{16} - 1$. Estas constantes llevarán el sufijo "**_ui**".

   Se deben incorporar a la lista de palabras reservadas las palabras **LONG** y **UINT**.

7. **Punto Flotante de 32 bits:** Números reales con signo y parte exponencial. La parte exponencial puede estar ausente. Si está presente, el exponente comienza con la letra "**e**" (mayúscula o minúscula) y el signo del exponente es obligatorio.

   Puede estar ausente la parte entera o la parte decimal, pero no ambas. El '.' es obligatorio.

   Ejemplos válidos: 1. .6 -1.2 3.e–5 2.E+34 2.5E-1 15. 0. 1.2e+10

   Considerar el rango 1.17549435E-38 < x < 3.40282347E+38 ∪

   -3.40282347E+38 < x < -1.17549435E-38 ∪ 0.0

   Se debe incorporar a la lista de palabras reservadas la palabra **FLOAT**.

8. **Punto flotante de 64 bits:** Números reales con signo y parte exponencial. La parte exponencial puede estar ausente. Si está presente, el exponente comienza con la letra "**D**" (mayúscula o minúscula) y el signo del exponente es obligatorio.

   Puede estar ausente la parte entera o la parte decimal, pero no ambas. El '.' es obligatorio.

   Ejemplos válidos: 1. .6 -1.2 3.d–5 2.D+34 2.5D-1 13. 0. 1.2d+10

   Considerar el rango 2.2250738585072014D-308 < x < 1.7976931348623157D+308 ∪

   -1.7976931348623157D+308 < x < -2.2250738585072014D-308 ∪ 0.0

   Se debe incorporar a la lista de palabras reservadas la palabra **DOUBLE**.

9. Incorporar a la lista de operadores, el operador aritmético ++.
10. Incorporar a la lista de operadores, el operador aritmético --.
11. Incorporar a la lista de operadores, el operador +=.
12. Incorporar a la lista de operadores, el operador -=.
13. Incorporar a la lista de palabras reservadas, las palabras **WHILE** y **DO**.
14. Incorporar a la lista de palabras reservadas, las palabras **DO** y **UNTIL**.
15. Incorporar a la lista de palabras reservadas, las palabras **WHILE** y **DO**.
16. Incorporar a la lista de palabras reservadas, las palabras **FOR**, **IN** y **RANGE**.
17. A definir en Trabajos Prácticos 2/3.
18. A definir en Trabajos Prácticos 2/3.
19. A definir en Trabajos Prácticos 2/3.
20. Incorporar a la lista de palabras reservadas, las palabras **IMPL** y **FOR**.
21. A definir en Trabajos Prácticos 2/3.

22. Incorporar a la lista de palabras reservadas, las palabras **INTERFACE** e **IMPLEMENT**.
23. A definir en Trabajos Prácticos 2/3.
24. A definir en Trabajos Prácticos 2/3.
25. A definir en Trabajos Prácticos 2/3.
26. A definir en Trabajos Prácticos 2/3.
27. A definir en Trabajos Prácticos 2/3.
28. A definir en Trabajos Prácticos 2/3.
29. A definir en Trabajos Prácticos 2/3.
30. A definir en Trabajos Prácticos 2/3.
31. A definir en Trabajos Prácticos 2/3.
32. **Comentarios de 1 línea**: Comentarios que comiencen con “**” y terminen con el fin de línea.
33. **Comentarios multilínea**: Comentarios que comiencen con “*{” y terminen con “}*” (estos comentarios pueden ocupar más de una línea).
34. **Cadenas de 1 línea**: Cadenas de caracteres que comiencen y terminen con “ # ” (estas cadenas no pueden ocupar más de una línea).

    Ejemplo: # ¡Hola mundo ! #
35. **Cadenas multilínea**: Cadenas de caracteres que comiencen y terminen con “ % ” . Estas cadenas pueden ocupar más de una línea. (En la Tabla de símbolos se guardará la cadena sin los saltos de línea).

    Ejemplo: % ¡Hola
    mundo! %