

Diseño de Compiladores I – Cursada 2025

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 carácter sólo puede ser una letra, 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 **mayúsculas**.
- Constantes correspondientes al tema particular asignado a cada grupo.
Nota: Para aquellas constantes 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: "+", "-", "*", "/".
- Operadores de asignación: "=", ":="
- Comparadores: ">=", "<=", ">", "<", "==", "!="
- Otros símbolos: "(", ")", "{", "}", "_", ";", y ">"
- Cadenas de caracteres correspondientes al tema particular de cada grupo.
- Palabras reservadas que deben escribirse con **minúsculas**:
if, else, endif, print, return
- 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

Implementación:

- a) 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.
- b) Se debe implementar una Tabla de Símbolos donde se almacenarán identificadores, constantes, y cadenas de caracteres. Esta tabla **debe** ser implementada con una estructura dinámica.

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.

Aplicación a desarrollar:

- a) No se requiere interfaz gráfica. Se recomienda la implementación de una aplicación que se ejecute desde consola.
- b) El código fuente a compilar **debe estar en un archivo** cuyo **nombre pueda ser elegido por el usuario**. La ruta del archivo donde está el código a compilar se debe pasar como parámetro al ejecutar la aplicación desde la consola.
- c) El Analizador Léxico deberá leer el código fuente, identificando tokens, y debe generar como salidas.

- 1) Tokens detectados en el código fuente. Por ejemplo:

```
Palabra reservada if
(
Identificador VAR%X
+
Constante entera 251
Palabra reservada else
etc.
```

- 2) 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

- 3) Contenidos de la Tabla de Símbolos.

Estas salidas se pueden imprimir por consola y/o enviarlas a uno o varios archivos de salida que deben quedar en la misma carpeta del archivo compilado.

Lenguaje de desarrollo:

- a) 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) que se podrá generar utilizando:
 - una herramienta tipo Yacc, en caso que el compilador sea desarrollado en Java, C++, u otro lenguaje para el cual se cuente con el generador de parser correspondiente, o
 - SLY, en caso que el compilador sea desarrollado en Python

- b) Si usan GitHub, debe ser privado.

Entrega

La entrega se pactará con el docente asignado al grupo. La asignación se publicará en el Aula Virtual de la materia. 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.
- Errores léxicos considerados, describiendo la forma de tratar cada uno.

Para implementaciones que contemplen el uso de herramienta tipo YACC

- 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.

Para implementaciones con SLY

- Lista de las expresiones regulares utilizadas para el reconocimiento de cada token.
- Lista de acciones semánticas asociadas al reconocimiento de cada token, con una breve descripción de cada una.

[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 20 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 (16 bits):** Constantes enteras con valores entre -2^{15} y $2^{15} - 1$ que se escriben como una secuencia de dígitos seguidos del sufijo **I**.
Se debe incorporar a la lista de palabras reservadas la palabra **int**.

2. **Enteros sin signo (16 bits):** Constantes con valores entre 0 y $2^{16} - 1$ que se escriben como una secuencia de dígitos seguidos del sufijo **UI**.
Se debe incorporar a la lista de palabras reservadas la palabra **uint**.

3. **Enteros largos (32 bits):** Constantes enteras con valores entre -2^{31} y $2^{31} - 1$ que se escriben como una secuencia de dígitos seguidos del sufijo **L**.
Se debe incorporar a la lista de palabras reservadas la palabra **long**.

4. **Enteros largos sin signo (32 bits):** Constantes enteras con valores entre 0 y $2^{32} - 1$ que se escriben como una secuencia de dígitos seguidos del sufijo **UL**.
Se debe incorporar a la lista de palabras reservadas la palabra **ulong**.

5. **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 “**F**” (mayúscula) seguido del signo del exponente que es obligatorio, y luego el valor del exponente.
El ‘.’ es obligatorio. Las partes entera o decimal pueden estar ausentes. Si está ausente la parte entera, debe estar presente la parte decimal. Si está ausente la parte decimal, debe estar presente la parte entera.
Ejemplos válidos: 1. .6 -1.2 3. F-5 2.0F+34 2.5F-1 15.0 0.0 .2F+10
Considerar el rango $1.17549435F-38 < x < 3.40282347F+38 \cup$
 $-3.40282347F+38 < x < -1.17549435F-38 \cup 0.0$
Se debe incorporar a la lista de palabras reservadas la palabra **float**.

6. **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) seguido del signo del exponente que es obligatorio, y luego el valor del exponente.
El ‘.’ es obligatorio. Las partes entera o decimal pueden estar ausentes. Si está ausente la parte entera, debe estar presente la parte decimal. Si está ausente la parte decimal, debe estar presente la parte entera.
Ejemplos válidos: 1. .6 -1.2 3. D-5 2.0D+34 2.5D-1 15.0 0.0 .2D+10
Considerar el rango $2.2250738585072014D-308 < x < 1.7976931348623157D+308 \cup$
 $-1.7976931348623157D+308 < x < -2.2250738585072014D-308 \cup 0.0$
Se debe incorporar a la lista de palabras reservadas la palabra **dfloat**.

7. **Cadenas de 1 línea:** Cadenas de caracteres delimitadas por comillas dobles. Estas cadenas no pueden ocupar más de una línea).
Ejemplo: “¡Hola mundo !”

8. **Cadenas multilínea:** Cadenas de caracteres delimitadas por ‘&’. 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! &

9. Incorporar a la lista de palabras reservadas, la palabra **var**

10. Incorporar a la lista de palabras reservadas, la palabra **var**

11. Incorporar a la lista de palabras reservadas, la palabra **var**

12. Incorporar a la lista de palabras reservadas, las palabras **while** y **do**.

13. Incorporar a la lista de palabras reservadas, las palabras **do** y **until**.

14. Incorporar a la lista de palabras reservadas, las palabras **do** y **while**.

15. Incorporar a la lista de palabras reservadas, las palabras **for**, **from** y **to**.

16. A definir en Trabajos Prácticos 2/3.

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. A definir en Trabajos Prácticos 2/3.

21. A definir en Trabajos Prácticos 2/3.

22. A definir en Trabajos Prácticos 2/3.

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. Incorporar a la lista de palabras reservadas, la palabra **lambda**.
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).