

Diseño de Compiladores I – Cursada 2022

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 25 caracteres de longitud. El primer debe 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 pueden ser minúsculas y/o mayúsculas.
- 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 minúsculas):
if, then, else, end-if, out, fun, return, break
- y demás símbolos / tokens indicados en los temas particulares asignados al 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.
- c) La numeración de las líneas de código debe comenzar en 1. Si se implementa una interfaz que permite mostrar o editar el código fuente, incluir alguna manera de identificar el número de cada línea del código.
- 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.
- Constantes con el primer y último valor fuera del rango.
- 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.

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.