

Marco teórico

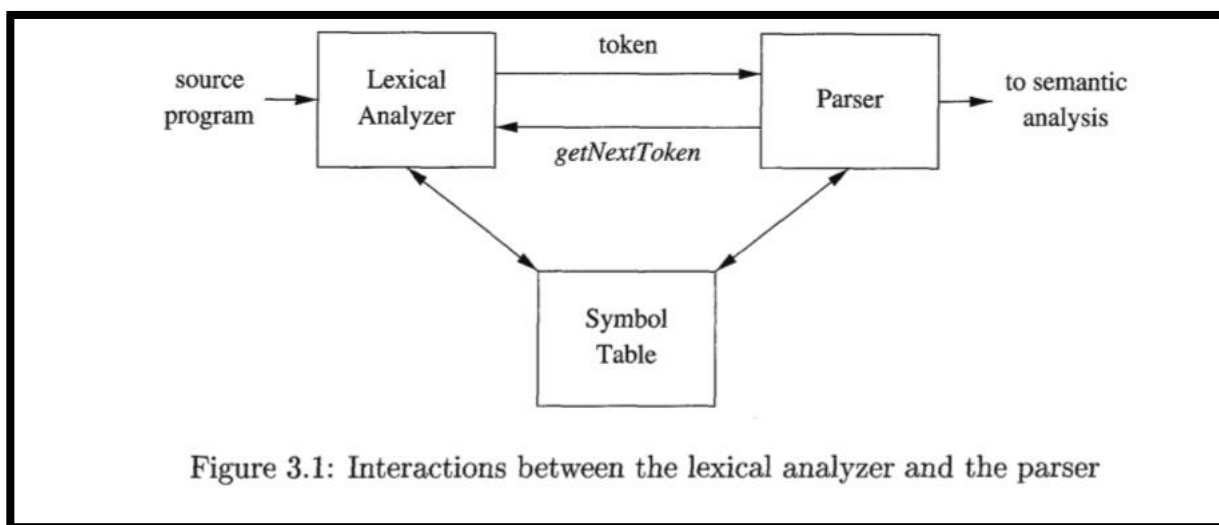
Analizador léxico: Constituye la primera fase de un compilador. Su tarea principal es leer el código fuente del programa, identificar los lexemas y producir como salida tokens correspondientes a esos lexemas. Esta secuencia de tokens será a su vez transferida al analizador sintáctico posteriormente.

Es posible que esta primera fase de compilación incluya también ignorar el texto que corresponde a comentarios, saltos de línea y espacios, dado que es este analizador el que recibe el programa fuente inicialmente. Esta sección de compilación puede tener dos tipos de tareas: escáner y analizador. En esta última es en la que se divide el programa en tokens.

Token: Es el elemento que identifica un tipo de unidad léxica. Se compone de un nombre y de un atributo de valor opcional

Lexema: Es la secuencia de caracteres de un programa que coincide con el patrón de un determinado token y es identificado por el analizador como una instancia de ese token.

Patrones: Describen la forma en que un token puede tomar forma, pueden ser descritos a través de lo que hemos definido como expresiones regulares.



En la figura 3.1 se ve el flujo del analizador léxico, analizador sintáctico y la tabla de símbolos. Esta última también puede comunicarse directamente con el analizador léxico para suplir información de un identificador, o de manera inversa, el analizador puede leer información en ella para ayudarle a determinar el token apropiado que le va a transferir al analizador sintáctico.

Lógica de primer orden

La lógica de primer orden es un conjunto de axiomas y reglas de inferencia desarrolladas por Gotlob Frege. Está basada en la lógica proposicional establecida por Boole, pero incluye otros conceptos importantes que no están en esta última, como cuantificadores y predicados. Las reglas de inferencia permiten demostrar formulas verdaderas a partir de otras fórmulas verdaderas, lo cual ha permitido formar una especie de 'cálculo' en este ámbito.

La lógica de primer puede usar elementos de lógica proposicional y términos que representan objetos. Para construir dichos términos se usan simbolos constantes, variables y funciones. La sintaxis usada para la lógica de primer orden se verá de la siguiente manera:

$$\begin{aligned}p &: 2 \times (7 - 9) < 12; \\x &:= 45; \\s &: \left[(3x1 + 4x23) \right] z89 < x;\end{aligned}$$

En estos ejemplos, p es una proposición, s un predicado y x es una instancia de una variable de un objeto numérico.

BIBLIOGRAFÍA

- Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. (2006). *Compilers Principles, Techniques & tools (Second Edition)* [archivo PDF]. Edited by Addison Wesley. Recuperado de <https://drive.google.com/file/d/1CqUoG-eqYDRZxEHXCsbF9yC-bZShe60V/view>
- Ricardo Peña Marí. (2006). *De Euclides a Java* [archivo PDF]. S.L. NIVOLA LIBROS Y EDICIONES. Recuperado de <https://drive.google.com/drive/folders/1RqzslL7kyvBpbdFsBVcwRhfcf6azqlTe>