

Trabajo Práctico Anual

de Matemática Discreta

Sección 4: Lenguajes

✓ Programar un analizador sintáctico para reconocer un análisis de caso simple y completo  
según las reglas de C++

**Analizador sintáctico**

Es una de las partes de un compilador. El analizador sintáctico convierte el texto de entrada en otras estructuras (comúnmente arboles), que son mas útiles para el posterior análisis y capturan la jerarquía implícita de la entrada. Un analizador léxico crea tokens de una secuencia de caracteres de entrada y son esos tokens los que son procesados por el analizador sintáctico para construir la estructura de datos.

Los lenguajes habitualmente reconocidos por los analizadores sintácticos son los lenguajes libres de contextos. Cabe notar que una justificación formal que establece que los lenguajes libres de contexto son aquellos reconocibles por un autómata de pila , de modo que todo analizador sintáctico que reconozca un lenguaje libre de contexto es equivalente en capacidad computacional a un autómata de pila.

**Proceso**

El siguiente caso demuestra un caso común de análisis de un lenguaje de programación con dos niveles de gramática, léxica y sintáctica.

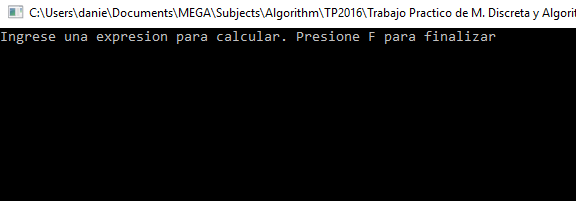
El primer estado es la generación de [tokens](https://es.wikipedia.org/wiki/Token_%28inform%C3%A1tica%29) o análisis léxico, en este proceso la cadena de entrada se parte en símbolos con significado definidos por una gramática de expresiones regulares, por ejemplo un programa [calculadora](https://es.wikipedia.org/wiki/Calculadora) con la siguiente entrada: "12\*(3+4)^2", la dividiría en los siguientes tokens 12, \*, (, 3, +, 4, ), ^ y 2, cada uno de estos símbolos tiene un significado en el contexto de la expresión [aritmética](https://es.wikipedia.org/wiki/Aritm%C3%A9tica). El analizador contendrá reglas para indicar que los símbolos \*, +, ^, ( y ) indican el comienzo de un nuevo token, de modo que otros tokens que no tendrían sentido como 12 o 13 no se generarán.

El siguiente estado es el análisis sintáctico lo que significa comprobar que los tokens forman una expresión válida, esto se hace usualmente usando una gramática libre de contexto que define recursivamente componentes que pueden aparecer en una expresión y el orden en que estos deben aparecer. Las reglas que definen un lenguaje de programación no siempre se pueden expresar usando únicamente una gramática libre de contexto, por ejemplo la validación de tipos y la declaración correcta de identificadores. Estas reglas pueden expresarse formalmente usando gramáticas de atributos.

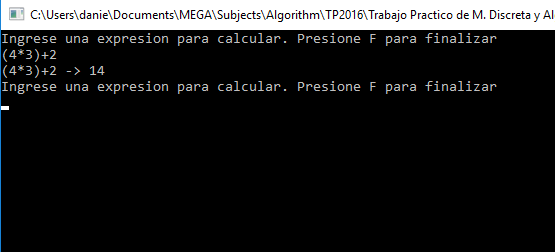
Manual de uso

1. Vamos a implemente el analizador sintáctico usando una calculadora, la cual analiza si la expresión ingresada es válida. Al ingresar la expresión es importante respetar las jerarquías entre las operaciones matemáticas y los paréntesis.

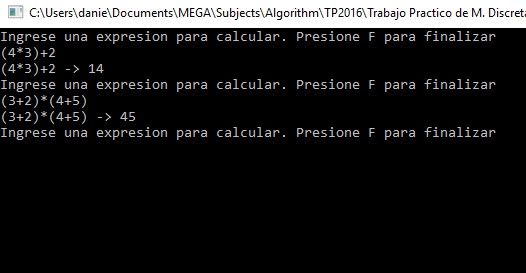
Ingresamos una expresión…..



1. Se resuelve la expresión y a la derecha de la flecha se muestra el resultado



1. Otro ejemplo…



1. Finalmente presionamos F para finalizar

