

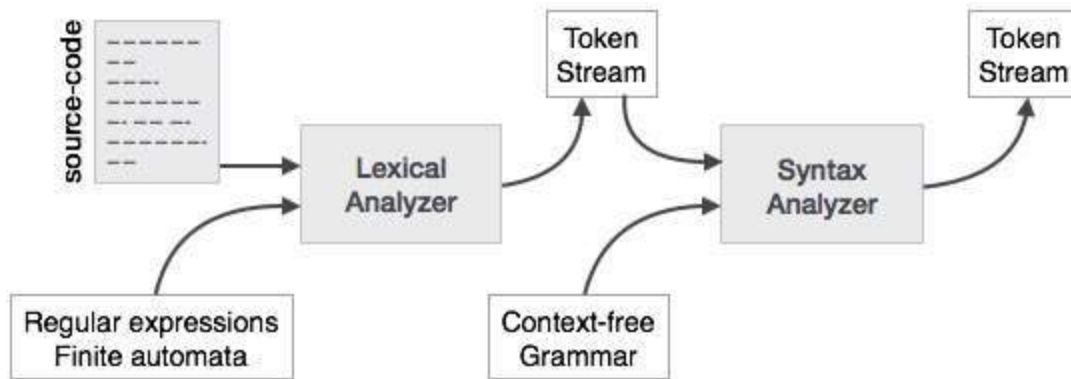
Analizador Sintático (Parser)

Analizador Sintático ou Parsing é a segunda fase, ou seja, após a análise lexical. Ele verifica a estrutura sintática da entrada fornecida, ou seja, se a entrada fornecida está na sintaxe correta (do idioma em que a entrada foi escrita) ou não. Ele faz isso construindo uma estrutura de dados, chamada de árvore de análise ou árvore de sintaxe. A árvore de análise é construída usando a gramática predefinida da linguagem e a string de entrada. Se a string de entrada fornecida pode ser produzida com a ajuda da árvore sintática (no processo de derivação), a string de entrada está na sintaxe correta. caso contrário, o erro é relatado pelo analisador de sintaxe ("Introduction to Syntax Analysis in Compiler Design").

Portanto, esta fase usa gramática livre de contexto (Context-Free Grammar), que é reconhecida por autômatos push-down. O Context-Free Grammar, por outro lado, é um superconjunto da Gramática Regular, conforme ilustrado abaixo:



O Analizador Sintático recebe a entrada de um analisador léxico na forma de fluxos de token. O analisador analisa o código-fonte (fluxo de token) em relação às regras de produção para detectar quaisquer erros no código. A saída desta fase é uma árvore de análise.



Dessa forma, o analisador realiza duas tarefas, ou seja, analisar o código, procurar erros e gerar uma árvore de análise como saída da fase. Espera-se que os analisadores analisem todo o código, mesmo que existam alguns erros no programa. Os analisadores usam estratégias de recuperação de erros (“Compiler Design - Syntax Analysis”).

Referencias bibliográficas

“Compiler Design - Syntax Analysis.” *Tutorialspoint*,

https://www.tutorialspoint.com/compiler_design/compiler_design_syntax_analysis.htm.

Accessed 18 October 2022.

“Introduction to Syntax Analysis in Compiler Design.” *GeeksforGeeks*, 15 June 2022,

<https://www.geeksforgeeks.org/introduction-to-syntax-analysis-in-compiler-design/>.

Accessed 18 October 2022.