

Introduction to syntax analysis

Study Guide

Asst. Prof. Vaibhavi Parikh
CSE, PIT
Parul University

1. Role Of Parser	3
--------------------------------	----------

2.1 Role Of Parser

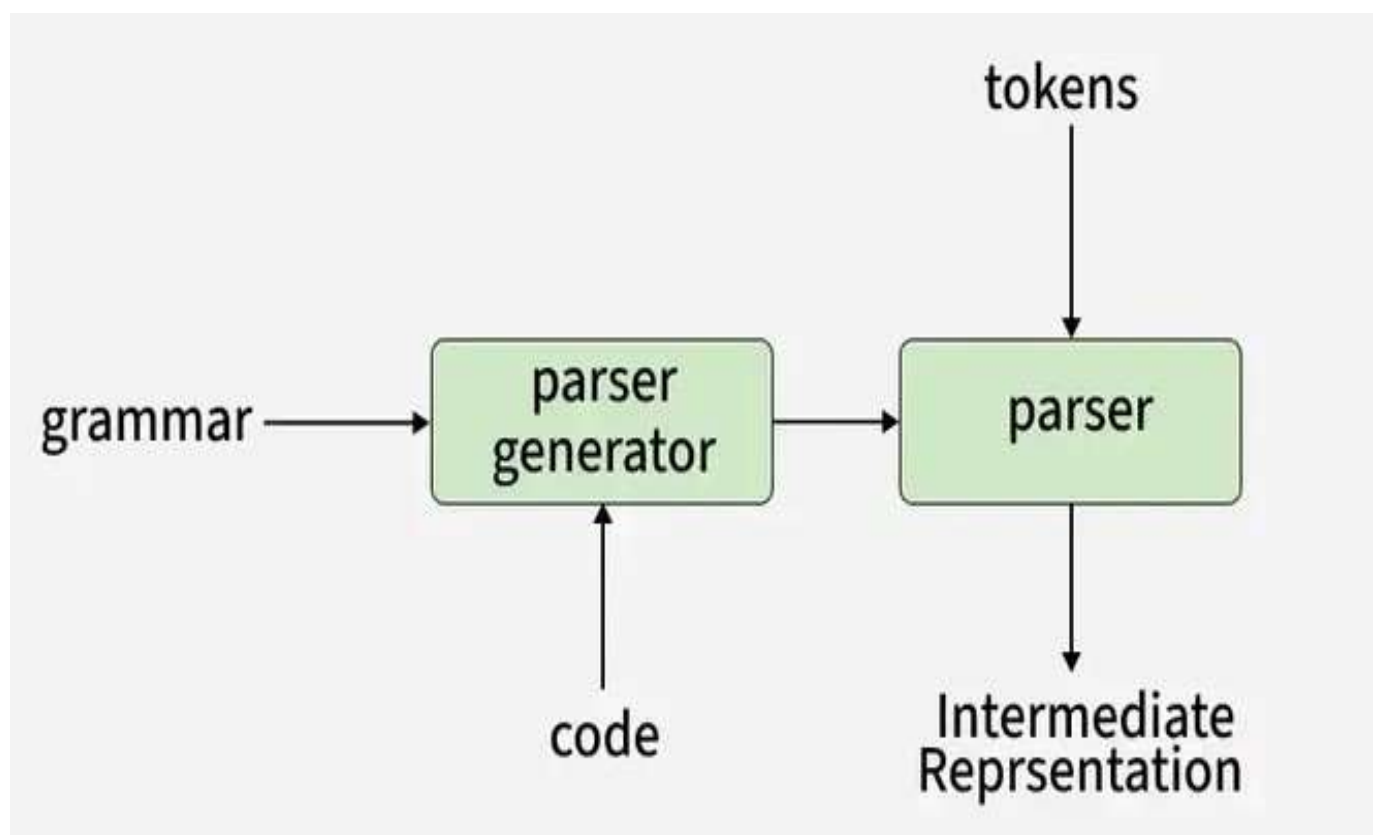
Parsing also known as syntactic analysis, is the process of analyzing a sequence of tokens to determine the grammatical structure of a program.

It takes the stream of tokens, which are generated by a lexical analyzer or tokenizer, and organizes them into a parse tree or syntax tree.

The parse tree visually represents how the tokens fit together according to the rules of the language's syntax.

This tree structure is crucial for understanding the program's structure and helps in the next stages of processing, such as code generation or execution.

Additionally, parsing ensures that the sequence of tokens follows the syntactic rules of the programming language, making the program valid and ready for further analysis or execution.



What is the Role of Parser?

A parser performs syntactic and semantic analysis of source code, converting it into an intermediate representation while detecting and handling errors.

Context-free syntax analysis: The parser checks if the structure of the code follows the basic rules of the programming language (like grammar rules).

It looks at how words and symbols are arranged.

Guides context-sensitive analysis: It helps with deeper checks that depend on the meaning of the code, like making sure variables are used correctly.

For example, it ensures that a variable used in a mathematical operation, like $x + 2$, is a number and not text.

Constructs an intermediate representation: The parser creates a simpler version of your code that's easier for the computer to understand and work with.

Produces meaningful error messages: If there's something wrong in your code, the parser tries to explain the problem clearly so you can fix it.

Attempts error correction: Sometimes, the parser tries to fix small mistakes in your code so it can keep working without breaking completely.

2. Summary of Key Concepts

- The role of a parser in a compiler is to check the source code's syntax, ensuring it follows the rules of the programming language, and to build a structured representation of the code, such as a parse Tree.
- It does this by taking a stream of tokens from the lexical analyzer, verifying the sequence against the language's grammar, and producing an output that can be used for further compilation stages.

Next Steps

- Explore other basic functions of CFG Context Free Grammar and Language.

References:

1. **Book Reference**

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools* (2nd ed.). Pearson Education.

2. **Journal Article**

Muchnick, S. S., & Hecht, M. S. (2018). Advances in compiler optimization: Modern approaches language processing. *Journal of Computer Science and Engineering*, 12(4), 233–245

3. **Website Reference**

GeeksforGeeks. (2024). *Structure of a compiler*. Retrieved November 7, 2025, from <https://www.geeksforgeeks.org/structure-of-compiler/>

4. **Conference Presentation**

Sharma, R., & Patel, D. (2022). *Applications of compiler technology in modern software development*. Paper presented at the International Conference on Advanced Computing and Communication Systems (ICACCS), Chennai, India.

5. **Report**

IEEE Computer Society. (2021). *Trends in programming language translation and compiler design*.

6. **Sources**

TutorialsPoint. (2024). *Lexical Analysis in Compiler Design*.

Parul[®] University | **NAAC** **A++**
Vadodara, Gujarat | **GRADE**



    
<https://paruluniversity.ac.in/>