phiC

Phase II - Lexer and Parser

Team Members and Roles

Srinith Dasari - Language Guru

Pranav Varma Pericherla - System Integrator

Akhila Kumbha - System Architect

Santoshi Gayatri Mavuru - Project Manager

OBJECTIVES AND GOALS

- To develop a lexer to tokenize the input source code.
- To create a parser to generate the parse tree from the tokenized code.
- To handle basic syntax and grammar checking.

$$\begin{tikzcd}\text{Input source code file} \arrow[r, bend right] & \boxed{\text{Lexer}}\end{tikzcd}$$

Lexer

It is the first stage of the compiler.
We have used lex tool.

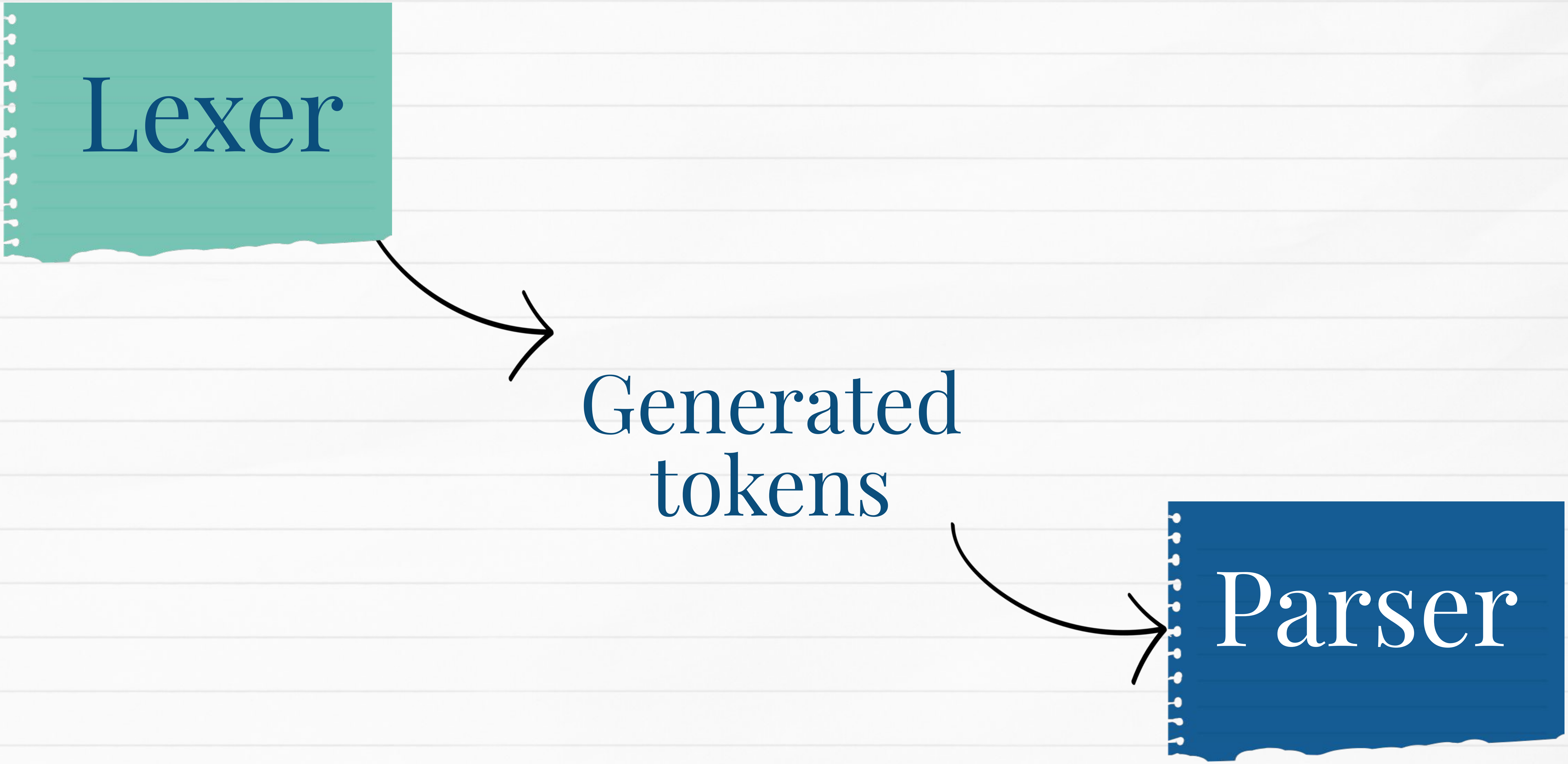
Tokens are distinct code elements like reserved keywords, identifiers, operators, and literals. The lexer processes input character by character, applying defined rules to form tokens.

The lexer defines rules to recognize tokens in the source code, using regular expressions.

The lexer labels tokens with types (e.g., "loop" as a keyword, "x" as an identifier, and "42" as an integer) and keeps track of identifiers in a symbol table.

Input -> testcase.phic
Output -> tokenfile.txt, generated tokens

It ignores white spaces, tabs and newlines. In case of identifying invalid tokens, it reports an error giving the line number where the issue occurred.


Lexer
Generated tokens
Parser

Parser

It is the second stage of the compiler.
It is written using YACC.

The parser uses context-free grammar rules to define valid syntactic structures and relationships between language constructs like expressions, statements, and declarations.

A lexer recognizes individual tokens using regular expressions, while syntax checking requires a parser to validate the code's overall structure.

The parser applies grammar rules to input tokens, selecting rules according to the current token and parsing stack state until the input is fully processed.

Input -> stream of tokens generated by lexer
Output -> parsedfile.txt, parse tree

If the input doesn't conform to the specified grammar, the parser generates error messages along with line number.

$$\begin{tikzcd}\text{Parser} \arrow[r, bend right] & \text{Parse tree} \arrow[r, bend right] & \text{Semantic Analyser}\end{tikzcd}$$

THANK YOU