INDIAN INSTITUTE OF
TECHNOLOGY
HYDERABAD

LEXICAL ANALYZER LiTeC

TEAM MEMBERS

BATHINI ASHWITHA

KETHAVATH PRANEETH NAYAK

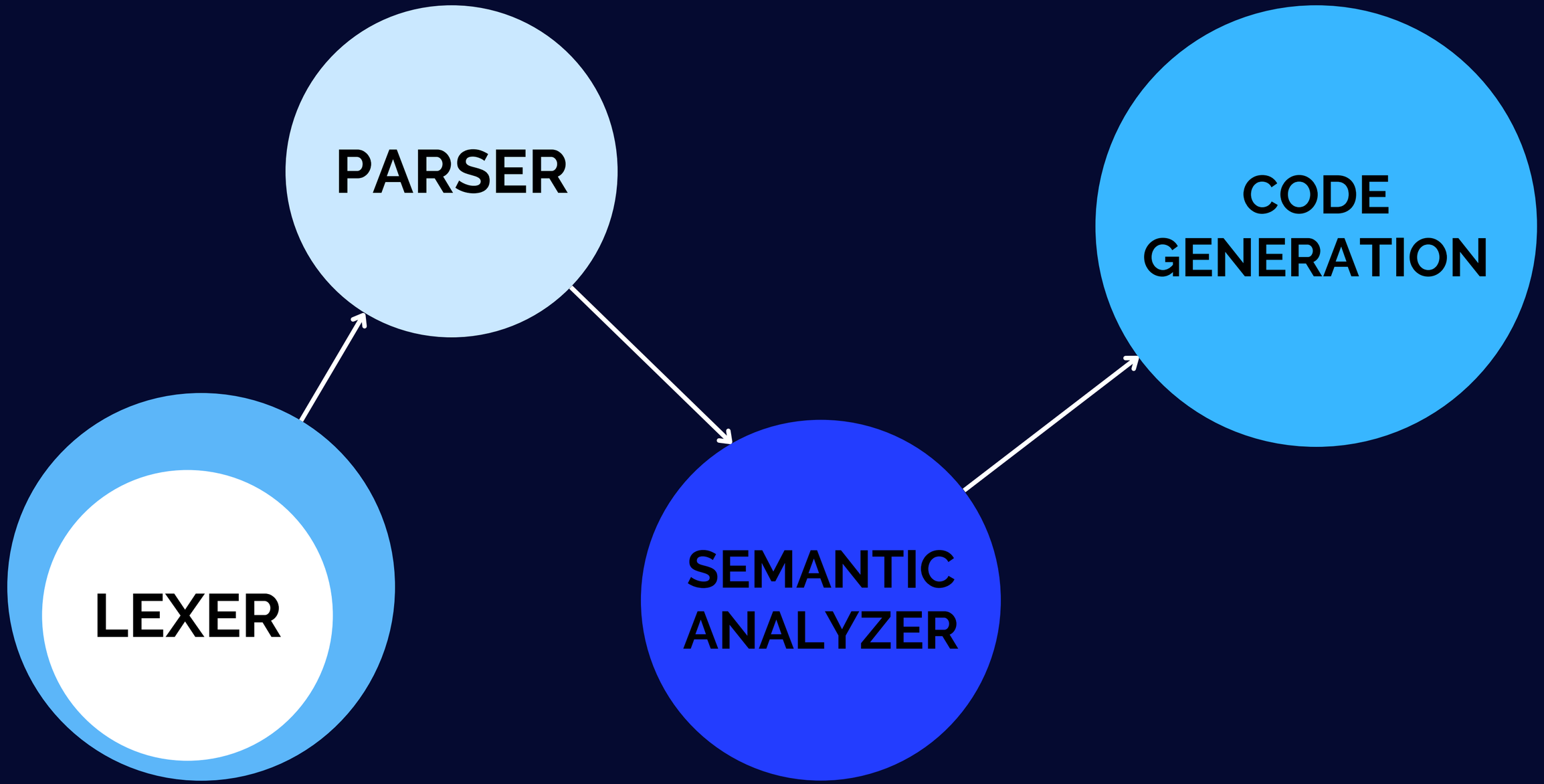
LANKA PRASANNA

MEGH SHAH

SANDEEP L

SIDDA BOENA JEEVAN SAMMESWAR

SAI SIVA ROHITH TIRUMALASETTI


PARSER
CODE GENERATION
LEXER
SEMANTIC ANALYZER

LEXICAL ANALYSIS

The first phase in the compiler's front end is known as lexical analysis. It is also known as a scanner.
The smallest individual and meaningful elements for compiler are known as tokens.
The output of this phase would be a series of tokens while the input is a high level program. This output is sent to the parser for syntax analysis.
Comments and whitespaces are also removed in the lexical phase.

LEXER & LEX

A program performing lexical analysis is known as a lexer. Lex is a computer program that generates lexical analyzers. It is a tool for automatically generating a lexer. It is used with the YACC parser generator. Lex compiler .l file and creates lex.yy.c program. C compiler runs this and outputs lexical analyzer.

Structure of lex program:

{ declaration }

%%

{ translation rules }

%%

{ auxiliary functions }

IMPLEMENTATION

We started with writing LiTeC grammar (handwritten) with lex specification in lex.l file.
We referred our language specification document for tokens and patterns of our language.
We generated lexical analyzer using lex tool.
Our output file shows tokens of our language taking sample test LiTeC program as input.

Sample input and ouput

```
1 int_64 main ()
2 {
3         declare int_64 x = 10 ;
4         TeX Latex programming
5         return 0;
6 }
```

```
jeevan@jeevan-HP-Laptop-14s-dr2xxx:~/Desktop/test3$ ./test.out
opened
1INT_64
2ID
3(
4)
5{
6DECLARE
7INT_64
8ID
9=
10Number
11;
12TEX
13RETURN
14Number
15;
16}
```

TOOLS

We used the following tools for lexical analysis of LiTeC:

1. Git
2. Lex/Flex
3. YACC
4. VS Code
5. GCC

THANK YOU

Final Defense