R. Mohammed Basil
192210640

Assignment - 3 FOC

Compiler

* lexical analyser
* Syntactic analyser
* Semantic analyser

lexical Analyser

→ also called as scanner.

→ Reads the successive line

→ It breaks into terms like identefier, operator, delimiter.

→ Analyser constructs symbol table.

→ The symbol table allocates memory.

How it works ?

* Input Processing : This stage involves cleaning up input text and preparing it for lexical analyser this may include for lexical analyser removing contents, whitespace and other non-essential character from input text.

* Token classification: The lexer checks that each token is valid according to roles of programming language.

* Token validation: lexer determines type of each token. Might classify keywords, identifiers, operations.

* Output generation: lexer generates output of the lexical analysis process which is typically a list of tokens.

Syntactic Analyser

⇒ The syntactic analyser refers to the expression, statement, declaration identified.

⇒ It is aided by formal grammer by programming language.

⇒ also called as parsing.

* Purpose: Is to exact meaning

⇒ checks text for meaningfulness comparing to rules of grammer.

Features : Syntax tree * context free grammer

* Top- down & bottom up * Error detection

* Optimisitation

Advantages :

1. Structural validation : Syntax analysis allows compiler to check if source code follows grammatical rules of programming language, which helps to detect & report error in source code.

2. Improved code generation : Syntax analysis can generate in a parse tree or abstract syntax tree of source code.

Semantic Analyser

→ It is also called as phase bridge.

→ Analysis phase of syntax

→ last phase of translation is code generation.

Errors recognised are

* Type mismatch * Undeclared variables

Function:

1. Type checking - Ensures that data types are used in a way consistent with their definition.

2. Label checking - A program should contain labels references.

3. Flow control check - keeps a check that control structures are used in a proper manner.

Types : 1. Static semantic - these are checked at compile time.

* Make sure that declarations & statements of program are semantically correct collection of produces which is called by passes as & when required by grammer.