Unit–II

1. Syntax analysis is also known as:

   A) Scanning

   B) Parsing

   C) Tokenization

   D) Optimization

   Answer: B

2. The primary function of syntax analysis is to:

   A) Generate machine code

   B) Check the lexical structure of a program

   C) Determine whether a token sequence conforms to the grammar

   D) Allocate memory for variables

   Answer: C

3. The input to a parser is:

   A) Source code

   B) Intermediate code

   C) Sequence of tokens generated by the lexical analyzer

   D) Object code

   Answer: C

4. The output of a parser is usually:

   A) Symbol table

   B) Parse tree or syntax tree

   C) Object code

   D) Machine instructions

   Answer: B

5. Context-Free Grammars (CFGs) are used to describe:

A) Lexical structure of a language

B) Syntax of a programming language

C) Memory allocation

D) Code optimization

Answer: B

6. A Context-Free Grammar consists of:

A) Tokens only

B) States and transitions

C) Terminals, non-terminals, productions, and a start symbol

D) Variables and constants

Answer: C

7. Symbols that can be replaced during derivation are called:

A) Terminals

B) Non-terminals

C) Tokens

D) Operators

Answer: B

8. Symbols that cannot be replaced further are called:

A) Non-terminals

B) Productions

C) Terminals

D) Variables

Answer: C

9. The start symbol of a grammar is:

A) The first terminal in the grammar

B) The symbol from which derivation begins

C) The last production rule

D) The first token of the input

Answer: B

10. A production rule in a grammar is represented using:

A) =

B) →

C) <>

D) :

Answer: B

11. A sequence of production rule applications is called:

A) Transition

B) Derivation

C) Reduction

D) Recognition

Answer: B

12. In a leftmost derivation:

A) The rightmost non-terminal is expanded first

B) Any non-terminal can be expanded

C) The leftmost non-terminal is expanded first

D) Only terminals are expanded

Answer: C

13. In a rightmost derivation:

A) The leftmost non-terminal is expanded first

B) The rightmost non-terminal is expanded first

C) Only terminals are expanded

D) Productions are not applied

Answer: B

14. A parse tree represents:

    A) The execution sequence of a program

    B) The hierarchical structure of derivations

    C) The memory organization

    D) The generated object code

Answer: B

15. A grammar that generates more than one parse tree for the same string is called:

    A) Deterministic grammar

    B) Recursive grammar

    C) Ambiguous grammar

    D) Reduced grammar

Answer: C

16. Which of the following is a top-down parsing technique?

    A) Shift-reduce parsing

    B) LR parsing

    C) Recursive descent parsing

    D) Operator precedence parsing

Answer: C

17. Top-down parsing constructs the parse tree from:

    A) Leaves to root

    B) Root to leaves

    C) Left to right only

    D) Right to left only

Answer: B

18. Recursive descent parsing is based on:

A) Bottom-up parsing

B) Top-down parsing

C) Operator precedence

D) LR parsing

Answer: B

19. A grammar suitable for predictive parsing must be:

A) Ambiguous

B) Left recursive

C) Free from left recursion

D) Context sensitive

Answer: C

20. Left factoring is used to:

A) Eliminate ambiguity

B) Remove left recursion conflicts

C) Make grammars suitable for predictive parsing

D) Generate machine code

Answer: C

21. Bottom-up parsing constructs the parse tree from:

A) Root to leaves

B) Leaves to root

C) Left to right

D) Right to left

Answer: B

22. Shift-reduce parsing is an example of:

A) Top-down parsing

B) Bottom-up parsing

C) Recursive parsing

D) Predictive parsing

Answer: B

23. In shift-reduce parsing, the "shift" action means:

A) Remove symbols from the stack

B) Move the next input symbol onto the stack

C) Generate intermediate code

D) Expand a non-terminal

Answer: B

24. In shift-reduce parsing, the "reduce" action means:

A) Replace a handle with its corresponding non-terminal

B) Read the next input symbol

C) Ignore the current token

D) Eliminate recursion

Answer: A

25. LR parsing scans the input from:

A) Left to right and produces a rightmost derivation in reverse

B) Right to left and produces a leftmost derivation

C) Left to right and produces a leftmost derivation

D) Right to left and produces a rightmost derivation

Answer: A

26. SLR stands for:

A) Simple Left-to-right Reduction

B) Simple LR

C) Syntax Level Reduction

D) Shift Left Reduce

Answer: B

27. Which parser is more powerful than an SLR parser?

    A) Recursive descent parser

    B) Predictive parser

    C) Canonical LR parser

    D) LL(1) parser

Answer: C

28. LALR parsers are preferred in practice because they:

    A) Require no parsing tables

    B) Have smaller parsing tables than canonical LR parsers

    C) Eliminate lexical analysis

    D) Accept only regular languages

Answer: B

29. Parser generators automatically generate parsers from:

    A) Source programs

    B) Machine code

    C) Grammar specifications

    D) Symbol tables

Answer: C

30. Which of the following is a widely used parser generator?

    A) Lex

    B) Yacc

    C) GCC

    D) Java

Answer: B