UNIT 5

1. Pushdown Automata (PDA)

- **Definition**: A PDA is a computational model that extends finite automata with an additional **stack** for memory, allowing it to recognize **context-free languages (CFLs)**.

Components of PDA:

1. **States**: Finite set of states.
2. **Input Alphabet (Σ\SigmaΣ)**: Set of input symbols.
3. **Stack Alphabet (Γ\GammaΓ)**: Set of symbols for the stack.
4. **Transitions**: Defined by a function, where a state, input, and stack top determine the next state, stack operation (push, pop, or no change).
5. **Initial State**: Starting state of the PDA.
6. **Start Stack Symbol**: The initial symbol on the stack.
7. **Final States**: Set of accepting states.

2. Representation of PDA

A PDA can be represented in two ways:

1. **Transition Function Representation**: δ(q,a,X)=(p,γ)\delta(q, a, X) = (p, \gamma)δ(q,a,X)=(p,γ), where:
   - q,pq, pq,p = states
   - aaa = input symbol
   - XXX = stack top
   - γ\gammaγ = new stack contents.
2. **Transition Diagram**: Visual representation with states and labeled transitions showing input, stack operations, and new states.

3. Acceptance by PDA

A PDA accepts a string if:

1. **Final State Acceptance**: PDA reaches a final state after processing the input and stack operations.
2. **Empty Stack Acceptance**: PDA empties the stack after processing the input.

4. Deterministic and Non-Deterministic PDA

1. **Deterministic PDA (DPDA)**:
   - At most one transition for a given state, input, and stack top.
   - Less powerful than NDPDA (not all CFLs are accepted by DPDA).

2. **Non-Deterministic PDA (NDPDA)**:
    - Multiple transitions possible for a given state, input, and stack top.
    - Equivalent in power to context-free languages.

Comparison:

- **NDPDA** recognizes all CFLs; **DPDA** only recognizes a subset of CFLs.
- Deterministic parsing (LL and LR) typically uses DPDAs.

5. Pushdown Automata and Context-Free Languages

- Every CFL can be recognized by a PDA.
- Conversion between CFG and PDA:
    1. From CFG to PDA: For each production A→αA \rightarrow \alphaA→α, modify the PDA to push or pop stack symbols accordingly.
    2. From PDA to CFG: Define rules based on PDA transitions.

6. Closure Properties of CFLs

CFLs are closed under:

- **Union**: If L1L_1L1 and L2L_2L2 are CFLs, L1∪L2L_1 \cup L_2L1∪L2 is also a CFL.
- **Concatenation**: L1L2L_1L_2L1L2 is a CFL.
- **Kleene Star**: L∗L^*L∗ is a CFL.

Not closed under:

- **Intersection**
- **Complementation**

7. LL(k) and LR(k) Grammars

LL(k) Grammars:

- Used for **Top-Down Parsing**.
- Parse input from **Left to Right** and produce **Leftmost Derivation**.
- kkk: Number of lookahead tokens used to decide the next action.
- Properties:
    - Simpler and faster parsing.
    - Cannot handle left recursion or ambiguous grammars.

LR(k) Grammars:

- Used for **Bottom-Up Parsing**.
- Parse input from **Left to Right** and produce **Rightmost Derivation** in reverse.
- kkk: Number of lookahead tokens used.
- Properties:
  - Handles larger class of grammars (including left recursion).
  - More complex but efficient for parsing programming languages.

8. Parsing

Top-Down Parsing:

- Starts from the start symbol and tries to derive the input string.
- Includes:
  - **Recursive Descent Parsing**
  - **Predictive Parsing** (LL(1) parsers)

Bottom-Up Parsing:

- Starts with the input string and reduces it to the start symbol.
- Includes:
  - **Shift-Reduce Parsing**
  - **LR Parsing** (Simple LR, Canonical LR, Lookahead LR).

Important MCQ

Pushdown Automata (PDA)

1. A PDA is a finite automaton with:
   - (A) A queue
   - (B) A stack
   - (C) A tape
   - (D) A counter
     Answer: B
2. The stack in a PDA is used for:
   - (A) Temporary storage of input symbols
   - (B) Keeping track of context
   - (C) Both A and B
   - (D) None of the above
     Answer: C
3. Which language cannot be accepted by a PDA?
   - (A) {anbn∣n≥1}\{a^n b^n \mid n \geq 1\}{anbn∣n≥1}
   - (B) {wwR∣w∈{a,b}∗}\{ww^R \mid w \in \{a, b\}^*\}{wwR∣w∈{a,b}∗}
   - (C) {anbmcn∣n,m≥0}\{a^n b^m c^n \mid n, m \geq 0\}{anbmcn∣n,m≥0}
   - (D) None of the above
     Answer: B
4. A PDA accepts a language if:
   - (A) It reaches a final state
   - (B) Its stack becomes empty
   - (C) Both A and B
   - (D) Either A or B
     Answer: D
5. Which of the following languages is context-free?
   - (A) {anbncn∣n≥1}\{a^n b^n c^n \mid n \geq 1\}{anbncn∣n≥1}
   - (B) {anbmcn+m∣n,m≥1}\{a^n b^m c^{n+m} \mid n, m \geq 1\}{anbmcn+m∣n,m≥1}
   - (C) {aibj∣i≠j}\{a^i b^j \mid i \neq j\}{aibj∣i□=j}
   - (D) {aibj∣i=j}\{a^i b^j \mid i = j\}{aibj∣i=j}
     Answer: D
6. The transition function of a PDA is of the form:
   - (A) δ(q,a)=p\delta(q, a) = pδ(q,a)=p
   - (B) δ(q,a,X)=(p,γ)\delta(q, a, X) = (p, \gamma)δ(q,a,X)=(p,γ)
   - (C) δ(q,a,X)=p\delta(q, a, X) = pδ(q,a,X)=p
   - (D) δ(q,X)=(p,a)\delta(q, X) = (p, a)δ(q,X)=(p,a)
     Answer: B

7. PDAs recognize:
   - (A) Regular languages
   - (B) Context-free languages
   - (C) Context-sensitive languages
   - (D) All of the above
     Answer: B
8. What is the key difference between DPDA and NDPDA?
   - (A) DPDAs can accept more languages
   - (B) NDPDAs can accept more languages
   - (C) Both accept the same languages
   - (D) None of the above
     Answer: B
9. The stack in a PDA can hold symbols from:
   - (A) Input alphabet
   - (B) Stack alphabet
   - (C) Both A and B
   - (D) None of the above
     Answer: B
10. A PDA accepting by empty stack:
    - (A) Always has final states
    - (B) Does not require final states
    - (C) Cannot process infinite strings
    - (D) Accepts only finite languages
      Answer: B

Context-Free Languages and PDA

11. Every PDA recognizes:
    - (A) A regular language
    - (B) A context-free language
    - (C) A context-sensitive language
    - (D) An undecidable language
      Answer: B
12. CFGs are equivalent to:
    - (A) Finite automata
    - (B) PDAs
    - (C) Turing machines
    - (D) None of these
      Answer: B

13. The stack is used in PDA to:
    - (A) Store current input symbols
    - (B) Track context of grammar rules
    - (C) Store output symbols
    - (D) None of the above

    Answer: B
14. A DPDA cannot recognize:
    - (A) Deterministic CFLs
    - (B) Non-deterministic CFLs
    - (C) Regular languages
    - (D) None of these

    Answer: B
15. A language is context-free if and only if:
    - (A) It can be generated by a CFG
    - (B) It can be recognized by a PDA
    - (C) Both A and B
    - (D) None of these

    Answer: C

Closure Properties of CFLs

16. CFLs are closed under:
    - (A) Union
    - (B) Concatenation
    - (C) Kleene star
    - (D) All of the above

    Answer: D
17. CFLs are not closed under:
    - (A) Intersection
    - (B) Complementation
    - (C) Both A and B
    - (D) None of these

    Answer: C
18. Intersection of a CFL and a regular language is:
    - (A) Regular
    - (B) Context-free
    - (C) Context-sensitive
    - (D) None of these

    Answer: B

19. Complement of a CFL is always:
    - (A) Regular
    - (B) CFL
    - (C) Non-CFL
    - (D) None of these
      Answer: C
20. CFLs are closed under:
    - (A) Homomorphism
    - (B) Inverse homomorphism
    - (C) Both A and B
    - (D) None of the above
      Answer: C

LL(k) and LR(k) Grammars

21. LL(k) parsing is a type of:
    - (A) Top-down parsing
    - (B) Bottom-up parsing
    - (C) Shift-reduce parsing
    - (D) None of these
      Answer: A
22. LR(k) parsing is:
    - (A) Predictive parsing
    - (B) Bottom-up parsing
    - (C) Top-down parsing
    - (D) None of these
      Answer: B
23. LL(k) parsers cannot handle:
    - (A) Left recursion
    - (B) Ambiguous grammars
    - (C) Both A and B
    - (D) None of these
      Answer: C
24. LR(k) parsers are:
    - (A) More powerful than LL(k) parsers
    - (B) Faster than LL(k) parsers
    - (C) Suitable only for regular languages
    - (D) None of these
      Answer: A

25. The lookahead kkk in LL(k) and LR(k) parsers represents:
    - (A) Number of tokens used to decide next move
    - (B) Depth of recursion
    - (C) Number of production rules
    - (D) None of the above
      Answer: A

Parsing Techniques

26. Top-down parsing starts with:
    - (A) Start symbol of the grammar
    - (B) Input string
    - (C) Both A and B
    - (D) None of these
      Answer: A
27. Bottom-up parsing starts with:
    - (A) Start symbol of the grammar
    - (B) Input string
    - (C) Both A and B
    - (D) None of these
      Answer: B
28. Predictive parsing is a type of:
    - (A) Top-down parsing
    - (B) Bottom-up parsing
    - (C) Shift-reduce parsing
    - (D) None of these
      Answer: A
29. The main limitation of top-down parsers is:
    - (A) They cannot handle left recursion
    - (B) They cannot handle right recursion
    - (C) They are slow
    - (D) None of the above
      Answer: A
30. LR parsers use:
    - (A) Shift-reduce parsing technique
    - (B) Recursive descent
    - (C) Predictive parsing
    - (D) None of these
      Answer: A

Parsing Techniques

31. In shift-reduce parsing, a shift action means:
    - (A) Adding a production rule
    - (B) Moving input symbol onto the stack
    - (C) Reducing stack contents
    - (D) None of these
      Answer: B
32. A reduce action in shift-reduce parsing involves:
    - (A) Moving input symbol onto the stack
    - (B) Applying a production rule to replace symbols on the stack
    - (C) Removing all stack contents
    - (D) None of these
      Answer: B
33. Which parser uses a parsing table?
    - (A) Recursive descent parser
    - (B) LL parser
    - (C) LR parser
    - (D) Both B and C
      Answer: D
34. Bottom-up parsing can also be called:
    - (A) Predictive parsing
    - (B) Shift-reduce parsing
    - (C) Recursive descent parsing
    - (D) None of these
      Answer: B
35. Which of the following grammars can be parsed by an LR(1) parser but not an LL(1) parser?
    - (A) Left-recursive grammars
    - (B) Ambiguous grammars
    - (C) Right-recursive grammars
    - (D) None of these
      Answer: A

Pushdown Automata Properties

36. The difference between NDPDA and DPDA lies in:
    - (A) The type of stack
    - (B) The number of stacks used
    - (C) The number of transitions allowed for a single input
    - (D) None of these
      Answer: C

37. How does a PDA accept by final state?
    - (A) When the PDA halts
    - (B) When PDA reaches a pre-defined accepting state
    - (C) When PDA empties the stack
    - (D) Both B and C
      Answer: B
38. A deterministic PDA is:
    - (A) More powerful than a non-deterministic PDA
    - (B) Equivalent in power to an NDPDA
    - (C) Less powerful than an NDPDA
    - (D) None of these
      Answer: C
39. A language accepted by an NDPDA is:
    - (A) Always regular
    - (B) Always context-free
    - (C) Always deterministic
    - (D) None of these
      Answer: B
40. Which of the following statements is true for PDAs?
    - (A) They can recognize all context-free languages.
    - (B) They can recognize some context-sensitive languages.
    - (C) Both A and B.
    - (D) None of these.
      Answer: A

LL(k) and LR(k) Parsing Techniques

41. An LL(1) parser cannot handle:
    - (A) Right recursion
    - (B) Left recursion
    - (C) Ambiguity
    - (D) Both B and C
      Answer: D
42. What is a major advantage of LR parsers?
    - (A) Simpler implementation
    - (B) Ability to parse a larger class of grammars
    - (C) Does not require lookahead
    - (D) None of these
      Answer: B

43. LL parsers derive strings by:
    - (A) Rightmost derivation
    - (B) Rightmost derivation in reverse
    - (C) Leftmost derivation
    - (D) None of these
      Answer: C
44. LR parsers derive strings by:
    - (A) Rightmost derivation
    - (B) Rightmost derivation in reverse
    - (C) Leftmost derivation
    - (D) None of these
      Answer: B
45. The main reason for using lookahead in parsers is to:
    - (A) Improve speed
    - (B) Resolve ambiguities
    - (C) Handle left recursion
    - (D) None of these
      Answer: B

Grammar and Parsing Relations

46. A grammar is said to be ambiguous if:
    - (A) It has no derivation for some strings
    - (B) It has two different parse trees for the same string
    - (C) It cannot be converted to a PDA
    - (D) None of the above
      Answer: B
47. Top-down parsing can fail when:
    - (A) The grammar is ambiguous
    - (B) The grammar is left-recursive
    - (C) The grammar is right-recursive
    - (D) Both A and B
      Answer: D
48. A CFG can always be converted to:
    - (A) An ambiguous grammar
    - (B) A PDA
    - (C) A deterministic grammar
    - (D) None of the above
      Answer: B

49. The stack content of a PDA can represent:

   - (A) Leftmost derivations
   - (B) Rightmost derivations
   - (C) Both A and B
   - (D) None of these
     Answer: C

50. Parsing is used in compilers to:

   - (A) Translate source code to machine code
   - (B) Generate an intermediate representation
   - (C) Analyze syntax structure of the source code
   - (D) Both B and C
     Answer: D