2 Parsing

1. The output of scanner is fed to:
(A) Lexical analyzer
(B) Syntax analyzer
(C) Semantic analyzer
(D) Code optimizer

2. Which of the following can define the fundamental nature of a Parser?
(A) Linear
(B) Hierarchical
(C) Random access based
(D) Iterative

3. The output of a syntax analyzer is:
(A) Intermediate code
(B) Parse tree
(C) Object code
(D) Three address code

4. Consider the following phases of a compiler:
(i) Syntax Analysis Phase
(ii) Intermediate Code generation Phase
(iii) Lexical Analysis Phase
(iv) code optimization Phase
Which of the below statements shows a set of Machine Independent phases of compiler?
(A) (i) only
(B) (ii) only
(C) (ii),(iv) only
(D) (iv) only
(E) (ii), (iii) and (iv) only
(F) (i), (ii) and (iii) only
(G) (i), (ii), (iii) and (iv) only

5. An operator precedence parser is a:
(A) Bottom-up parser.
(B) Top-down parser.
(C) Back tracking parser.
(D) None of the above.

6. Type checking is normally done during:
(A) lexical analysis
(B) syntax analysis

(C) syntax directed translation
(D) code optimization

7. Which of the following statements is not true?
(A) LALR(1) parser is more powerful than SLR(1)
(B) LALR(1) parser is more powerful than Canonical LR(1) parser
(C) Canonical LR(1) parser is more powerful than LALR(1) parser
(D) None of the above

8. Which of the following is the most powerful parsing method?
(A) LL(1)
(B) CLR(1)
(C) SLR(1)
(D) LALR(1)

9. Which of the following derivations does a bottom-up parser use while parsing an input string? The input is assumed to be scanned in left to right order.
(A) Leftmost derivation
(B) Leftmost derivation traced out in reverse
(C) Rightmost derivation
(D) Rightmost derivation traced out in reverse

10. Consider following statements about SLR(1) and LALR(1) parsing tables for a context free grammar. (i) The goto part of both tables are always identical.
(ii) The shift entries are identical in both the tables.
(iii) The reduce entries in the tables are always identical.
(iv) The error entries in tables may be different
Which of the above statements is/are not necessarily true?:
(A) (i) only
(B) (ii) only
(C) (iii) only
(D) (ii) and (iv) only
(E) (iv) only
(F) (i), (ii) and (iv) only

11. Write a grammar for the following languages. Are the grammars ambiguous?
(i) All strings of 0's and 1's that have the same number of 0's and 1's.
(ii) All strings of 0's and 1's that have more 0's than 1's.
(iii) All balanced pairs of left and right parentheses (e.g., "()", "()(())").

12. Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?
(A) Removing left recursion alone
(B) Factoring the grammar alone
(C) Removing left recursion and left factoring the grammar
(D) None of the above

13. Which one of the following is a top-down parser?
(A) Recursive descent parser.
(B) Operator precedence parser.
(C) An LR(k) parser.
(D) An LALR(k) parser.

14. Among simple LR (SLR), canonical LR, and look-ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is the most powerful, in that order?
(A) SLR, LALR
(B) Canonical LR, LALR
(C) SLR, canonical LR
(D) LALR, canonical LR

15. Assume that the SLR parser for a grammar G has n_1 states and the LALR parser for G has n_2 states. The relationship between and is:
(A) n_1 is necessarily less than n_2
(B) n_1 is necessarily equal to n_2
(C) n_1 is necessarily greater than n_2
(D) None of the above

16. An LALR(1) parser for a grammar G can have shift-reduce (S-R) conflicts if and only if
(A) The SLR(1) parser for G has S-R conflicts
(B) The LR(1) parser for G has S-R conflicts
(C) The LR(0) parser for G has S-R conflicts
(D) The LALR(1) parser for G has reduce-reduce conflicts

17. Consider the grammar shwn below:
$A \rightarrow BA|(A)|\epsilon$
$B \rightarrow AB|b|\epsilon$
The above grammar is not suitable for predictive-parsing because the grammar is:
(A) ambiguous
(B) directly left-recursive
(C) right-recursive

(D) indirectly left recursive

18. Consider the grammar shown below:
$S \rightarrow (L)|a$
$L \rightarrow L, S|S$
For each of the two strings: "(a,a)" and "(a,(a,a))""
(i) Find the parse tree
(ii) Find the leftmost derivation
(iii) Find the rightmost derivation

19. Which of the following describes a handle (as applicable to LR-parsing) appropriately?
(A) It is the position in a sentential form where the next shift or reduce operation will occur.
(B) It is non-terminal whose production will be used for reduction in the next step.
(C) It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur.
(D) It is the production that will be used for reduction in the next step along with a position in the sentential form where the right hand side of the production may be found.

20. Which of the following statements about parser is/are not CORRECT?
I. Canonical LR is more powerful than SLR
II. SLR is more powerful than LALR
III. SLR is more powerful than Canonical LR
(A) I only
(B) II only
(C) III only
(D) II and III only

21. Which of the following statements are not always TRUE?
I. There exist parsing algorithms for some programming languages whose complexities are less than $\theta(n^3)$.
II. A programming language which allows recursion can be implemented with static storage allocation.
III. No L-attributed definition can be evaluated in the framework of bottom-up parsing.
IV. Code improving transformations can be performed at both source language and intermediate code level.
(A) I and II
(B) I and IV

(C) II and III
(D) I, III and IV

22. Which of the following correctly describes parsing?
(A) The input string is alternately scanned left to right and right to left with reversals.
(B) Input string is scanned once left to right with rightmost derivation and symbol look-ahead.
(C) $LR(k)$ grammars are expressively as powerful as context-free grammars.
(D) Parser makes left-to-right passes over input string.
(E) Input string is scanned from left to right once with symbol to the right as look-ahead to give left-most derivation.

23. For a context free grammar, FOLLOW(A) is the set of terminals that can appear immediately to the right of non-terminal in some "sentential" form. We define two sets LFOLLOW(A) and RFOLLOW(A) by replacing the word "sentential" by "left sentential" and "right most sentential" respectively in the definition of FOLLOW (A).
(A) FOLLOW(A) and LFOLLOW(A) are always the same.
(B) FOLLOW(A) and RFOLLOW(A) are always the same.
(C) All the three sets are identical.
(D) All the three sets are different.

24. Consider the grammar shown below.
$S \rightarrow CC$
$C \rightarrow cC|d$
This grammar is
(A) LL(1)
(B) SLR(1) but not LL(1)
(C) LALR(1) but not SLR(1)
(D) LR(1) but not LALR(1)
(E) None of the above

25. Consider the grammar:
$S \rightarrow aSbS|bSaS|\epsilon$
Which of the following parser is able to parse this grammar unambiguously?
(A) LL(1)
(B) SLR(1) but not LL(1)
(C) LALR(1) but not SLR(1)
(D) LR(1) but not LALR(1)
(E) None of the above

26. Attributes of following arithmetic operators in some programming language are given below. Precedence levels are shown in the range [1,5] with 1 being lowest and 5 being highest precedence. Unary operators are used if it is required to represent some output in signed form.

Operator	**Precedence**	**Associativity**	**Arity**
$+$	3	Left	Binary
$-$	4	Right	Binary
$*$	1	Left	Binary
$/$	4	Right	Binary
$-$	4	Right	Unary

The value of the expression $2-5+7*3-10/2-2-1+1*10$ in this language is:
(A) -96
(B) ∞
(C) -220
(D) -156
(E) -240

27. Consider the following expression grammar :
$E \to E - T|T$
$T \to T + F|F$
$F \to (E)|id$
Which of the following grammars is not left recursive, but is equivalent to ?
(A)
$E \to E - T|T$
$T \to T + F|F$
$F \to (E)|id$

(B)
$E \to TE'$
$E' \to -TE'|\epsilon$
$T \to T + F|F$
$F \to (E)|id$

(C)
$E \to TX$
$X \to -TX|\epsilon$
$T \to FY$
$Y \to +FY|\epsilon$

$F \rightarrow (E)|id$

(D)
$E \rightarrow TX|(TX)$
$X \rightarrow -TX|+TX|\epsilon$
$T \rightarrow id$

28. Consider the production rules of a grammar given below:

$E \rightarrow aA|(E)$
$A \rightarrow +E|*E|\epsilon$

Which of the following statements are true about First and Follow notations for Nonterminals of this grammar?

(A)First (E) = { a,(,+ }, First (A) = { +,*,ϵ},
Follow (E) = Follow (A) = { \$,) }

(B) First (E) = { a,* }, First (A) = { +,(,ϵ },
Follow (E) = Follow (A) = { \$,) }

(C) First (E) = { a,(,* }, First (A) = { +,ϵ },
Follow (E) = Follow (A) = { \$,) }

(D) First (E) = { a,(}, First (A) = { +,*,ϵ },
Follow (E) = Follow (A) = { \$,) }

29. Consider the following grammar designed to represent Boolean operations.
$S \rightarrow S \wedge S|S \vee S|(S)|true|false$
Answer following questions.
(i) Compute First sets for each nonterminal.
(ii) Can this grammar be parsed by a LL(1) parser? Provide answer with proper reason.

30. Consider the following grammar:
$S \rightarrow abS|acS|c$
Answer following questions.
(i) Compute First sets for each nonterminal.
(ii) Can this grammar be parsed by a LL(1) parser.
(iii) If answer of (ii) question is "Yes", rewrite the grammar so it can be parsed by a LL(1) parser.

31. What is the maximum number of reduce moves that can be taken by a bottom-up parser for a grammar with no epsilon and unit-production (i.e., of type $A \to \epsilon$ and $A \to a$) to parse a string with n tokens is?
(A) $\frac{n}{2}$
(B) $n-1$
(C) $2n-1$
(D) 2^n
(E) 2^{n-1}

32. Consider the following grammar:
$Goal \to Expr$
$Expr \to Term + Expr | Term – Expr | Term$
$Term \to Factor * Term | Factor / Term | Factor$
$Factor \to num | id$
(i) Apply left factoring to create an LL(1) grammar.
(ii) Compute FIRST and FOLLOW for the modified grammar.

33. Consider the production rules of a grammar given below:

$E \to aA | (E)$
$A \to +E | *E | \epsilon$
Which of the following represents number of invalid entries in its LL(1) parsing table?
(A) 4
(B) 6
(C) 8
(D) 10

34. Consider the following grammar:
$S \to abS | acS | c$
The weakest parser which can parse the above grammar is:
(A) SLR(1) parser
(B) LALR(1) parser
(C) CLR(1) parser
(D) None of the above

35. Consider the grammar shown below:
$E \to (aEb) | a | b | \epsilon$
Let the number of states in SLR (1), LR(1) and LALR(1) parsers for the grammar be n_1, n_2 and n_3 respectively. Which of the following relationships is always true for the above grammar?
(A) $n_1 = n_2 = n_3$
(B) $n_1 \leq n_2 \geq n_3$
(C) $n_1 \leq n_2 \leq n_3$

(D) $n_1 = n_2 \leq n_3$

36. Consider the context-free grammar:
$E \rightarrow E + E$
$E \rightarrow (E * E)$
$E \rightarrow id$
where E is the starting symbol.
Which of the following terminal strings has more than one parse tree when parsed according to the above grammar?
(A) id + id + id + id
(B) id + (id* (id * id))
(C) (id* (id * id)) + id
(D) ((id * id + id) * id)

37. Consider the following grammar: $S \rightarrow AB$
$A \rightarrow id$
$B \rightarrow *S|\epsilon$
In the predictive parser table of the grammar (denoted by M) the entries M[S,id] and M[B,$] respectively are:
(A) $A \rightarrow id$ and $B \rightarrow *S$
(B) $S \rightarrow AB$ and $B \rightarrow \epsilon$
(C) $S \rightarrow AB$ and $B \rightarrow *S$
(D) $A \rightarrow id$ and $B \rightarrow \epsilon$

38. How many states are there in the SLR(1) parsing table for the grammar shown below?
$S \rightarrow L = R$
$S \rightarrow R$
$L \rightarrow *R$
$L \rightarrow id$
$R \rightarrow L$
(A) 10
(B) 11
(C) 12
(D) 13
(E) 14

39. Consider the following grammar:
$S \rightarrow Sa|Sc|c$
(i) Compute the canonical set of LR(1) items for the grammar.
(ii) Bui;d the Parsing table for this grammar and explain why the grammar is or is not LR(1).
(iii) Using the parsing table, show all steps of parsing of string *caca*. The

explanation should be in the form of a tuple: $< stack, input, action >$

40. Given the following CFG grammar G = ({S,A,B}, S, {a, b, x}, P) with P:
(1)$S \rightarrow A$
(2)$S \rightarrow xb$
(3)$A \rightarrow aAb$
(4)$A \rightarrow B$
(5)$B \rightarrow x$

For this grammar answer the following questions:
(i) Compute the set of LR(1) items for this grammar and the corresponding DFA.
(ii) Construct the corresponding LR parsing table.
(iii) Would this grammar be LR(0)? Why or why not?
(iv) Show the stack contents, the input and the rules used during parsing for the input string $w = axb$
(v) Would this grammar be suitable to be parsed using a top-down LL parsing method? Why?

41. Consider the grammar:
$S \rightarrow aSbS|bSaS|\epsilon$
If number of states in the SLR(1) parsing table is S and total number of conflicts are denoted by C Find the value of $S + C$?
(A) 10
(B) 15
(C) 20
(D) 24
(E) 28

42. Consider the following grammar G:
$S \rightarrow F|H$
$F \rightarrow p|c$
$H \rightarrow d|c$
Where S, F, and H are non-terminal symbols, p, d, and c are terminal symbols. Which of the following statement(s) is/are correct?
S1: LL(1) can parse all strings that are generated using grammar G.
S2: LR(1) can parse all strings that are generated using grammar G.
(A) Only S1
(B) Only S2
(C) Both S1 and S2
(D) Neither S1 and S2

43. A CFG G is given with the following productions where S is the start symbol, A is a non-terminal and a and b are terminals.

$S \rightarrow aS|bA$
$A \rightarrow aAb|bAa|\epsilon$

For the string "aabaabbaabb" how many steps are required to derive the string and how many parse trees are there?
(A) 6 and 1
(B) 6 and 2
(C) 8 and 1
(D) 7 and 2

44. Consider the following grammar:
$stmt \rightarrow ifexprthenexprelseexpr; stmt|0$
$expr \rightarrow termrelopterm|term$
$term \rightarrow id|number$
$id \rightarrow a|b|c$
$number \rightarrow [0-9]$
where relop is a relational operator (e.g.. ¡ . ¿ ...).
0 refers to the empty statement, and if, then, else are terminals. Consider a program P following the above grammar containing n if terminals. The number of control flow paths in P is:
(A) $n-1$
(B) $2*(n-2)$
(C) n^2
(D) 2^n

45. Consider the following grammar:
$E \rightarrow E+E|E*E|id$
Consider this input string which is to be parsed with the rules of this grammar:
$a+b*c*d$
Which of the following shows the number of Parse trees possible for this input string?
(A) 1
(B) 2
(C) 3
(D) 4
(E) 5

46. Consider the following grammar:
$S \rightarrow S*E$
$S \rightarrow E$
$E \rightarrow F+E$
$E \rightarrow F$
$F \rightarrow id$

Consider the following LR(0) items corresponding to the grammar above.
(i) $S \to S * .E$
(ii) $E \to F. + E$
(iii) $E \to F + .E$
Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar? (A) (i) and (ii)
(B) (ii) and (iii)
(C) (i) and (iii)
(D) None of the above

47. Consider the following grammar:
$E \to E + E|E * E|id$
Consider this input string which is to be parsed with the rules of this grammar:
$a + b * c * d$
For this grammar and input string what is the probability of "*" being right associative?
(A) 0.33
(B) 0.40
(C) 0.50
(D) 0.67

48. Consider the following grammar:
$E \to E + E|E * E|id$
Consider this input string which is to be parsed with the rules of this grammar:
$a + b * c * d + e$
For given grammar and given string there are more than one parse trees exist. If we want to reduce the tree-count to 1 we need to introduce P new productions and Q new symbols within these new productions. What is the value of the cardinality of the power set of a set whose cardinality is denoted by $|P * Q|$?
(A) 2
(B) 4
(C) 16
(D) 64
(E) 256

49. Consider the following sets of LR(1) items in the states of a LR(1) parser.

State 0:
[A → • a , b]
[A → a • , c]
[B → a • , b]

State 1:
[A → • a , a]
[A → • a , b]
[B → a • , b]

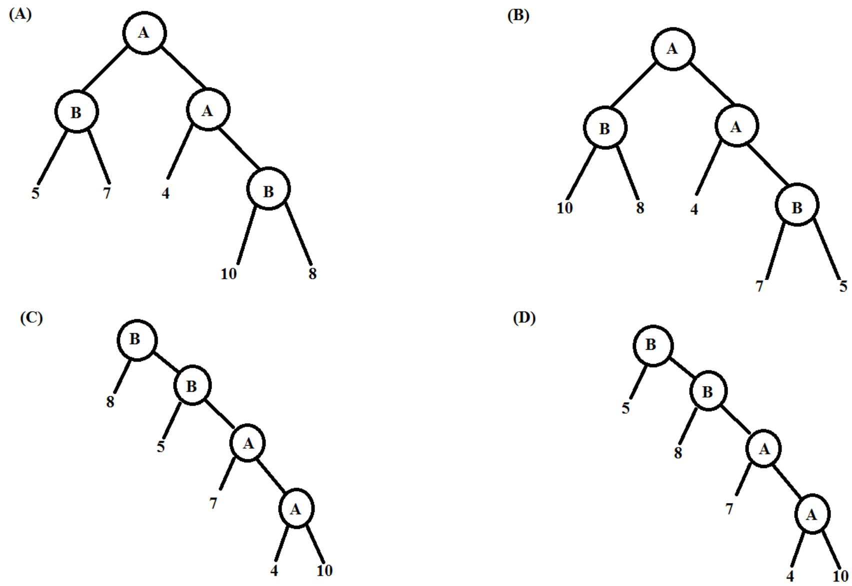
State 2:
[A → • a , c]
[A → a • , b]
[B → a • , a]

State 3:
[A → • a , b]
[B → • a , b]

(i) Find all shift/reduce and reduce/reduce conflicts. List the LR(1) items and lookaheads causing conflicts.
(ii) What states would be merged in a LALR(1) parser?
(iii) Are any new reduce/reduce conflicts introduced in the LALR(1) parser?
(iv) Explain why LALR(1) parsers will not introduce new shift/reduce conflicts.

50. Consider the following grammar:
$E \rightarrow E + E|(E * E)|id$
Consider this input string which is to be parsed with the rules of this grammar:
$a + (b + c + d * f)$
Which of the following shows the number of Parse trees possible for this input string?
(A) 1
(B) 2
(C) 3
(D) 4
(E) 5

51. Consider two binary operators A and B with the precedence of operator A being lower than that of the operator B. Operator A is right associative while operator B is left associative. Which one of the following represents the parse tree for the expression: $5\ B\ 7\ A\ 4\ A\ 10\ B\ 8$



52. Consider the following grammar:
$E \rightarrow E + E | E * E | (E) | id$
Consider this input string which is to be parsed with the rules of this grammar:
$a + (b + (c * d) + f)$
Which of the following shows the number of Parse trees possible for this input string?
(A) 1
(B) 2
(C) 3
(D) 4
(E) 5

53. In the above question, the sum of the heights of all the Parse trees generated by the parser for the given input string is:?
(A) 10
(B) 12
(C) 14
(D) 16
(E) 20

54. Consider the grammar shown below:
$S \rightarrow Aa | bAc | Bc | bBa$
$A \rightarrow d$

$B \to d$
The grammar is:
(A) LL(1)
(B) SLR(1)
(C) LALR(1)
(D) CLR(1)
(E) None of the above

55. Consider the following grammar, which generates expressions formed by applying "+" to integer and floating point constants. When two integers are added, the result is integer, otherwise, it is a float.
$E \to E + T|T$
$T \to num.num|num$
How many parse trees are constructed for the expression: $5 + 4 + 3.5 + 2 + 1.3$. Draw all parse trees.

56. Consider the following grammar:
$E \to E + E|E * E|id$
Consider this input string which is to be parsed with the rules of this grammar:
$a + b * c * d + e$
The above grammar is fed to "yacc", an LALR(1) parser generator, for parsing and evaluating arithmetic expressions.
Which one of the following is true about the action of "yaac" for the given grammar? (A) It detects recursion and eliminates recursion
(B) It detects reduce-reduce conflict, and resolves
(C) It detects shift-reduce conflict, and resolves the conflict in favor of a shift over a reduce action
(D) It detects shift-reduce conflict, and resolves the conflict in favor of a reduce over a shift action

57. Consider the following grammar:
$E \to E + E|E * E|id$
The above grammar is fed to "yacc", an LALR(1) parser generator, for parsing and evaluating arithmetic expressions. Assume the conflicts of this question are resolved using "yacc" tool and an LALR(1) parser is generated for parsing arithmetic expressions as per the given grammar. Consider an expression $a * b + c + d * e * f$.
What precedence and associativity properties does the generated parser realize?
(A) Equal precedence and left associativity
(B) Equal precedence and right associativity
(C) Precedence of $*$ is higher than that of $+$, and both operators are left associative
(D) Precedence of $+$ is higher than that of $*$, and both operators are left

associative

58. Consider the grammar shown below:
$S \to A|xb$
$A \to aAb|B|x$
How many states are there in the LALR(1) Parsing table for the above grammar and by how many states it differs from the number of states in CLR(1)?
(A) 8,6
(B) 11,4
(C) 13,5
(D) 15,5

59. Consider the following grammar:
$S \to S + S|S * S|(S)|num$
(i) Compute the canonical set of LR(1) items for the grammar.
(ii) How many sets are there in the transition diagram for this parser.
(iii) Is there any conflict in the parsing table?

60. Consider the following grammar:
$S \to AS|bA \to SA|a$
(i) Compute the canonical set of LR(0) items for the grammar (ii) Build the ACTION/GOTO table for the grammar (iii) Compute the canonical set of LR(1) items for the grammar (iv) Build the ACTION/GOTO table for the grammar

61. Consider the following grammar:
$S \to aAd|bBd|aBe|bAe$
$A \to c$
$B \to c$
The grammar is:
(A) LL(1)
(B) LALR(1)
(C) SLR(1)
(D) CLR(1)

62. Consider the following grammar:
$S \to Aa|bAc|Bc|bBa$
$A \to d$
$B \to d$
The grammar is:
(A) LL(1)
(B) LALR(1)
(C) SLR(1)

(D) CLR(1)