Theory of Computation and Compiler Design Test 4

Number of Questions: 35 **Section Marks: 30**

Directions for questions 1 to 35: Select the correct alternative from the given choices.

1. Which of the following is FALSE for regular expressions L, M and N?
(A) $L + M = M + L$
(B) $(L + M) + N = L + (M + N)$
(C) $(LM) N = (LM) N$
(D) $LM = ML$

2. Which of the following does not hold for two regular expressions R and S?
(A) $(R^*)^* = R^*$
(B) $(\in + R)^* = R^*$
(C) $(R^*S^*)^* = (R + S)^*$
(D) None of the above

3. Which of the following language is not regular?
(i) $(0 + 1)^*$
(ii) Palindromes over $\{0, 1\}$
(iii) $\{0^n10^n/n \geq 1\}$
(A) (i), (ii), (iii) (B) (i), (iii)
(C) (ii), (iii) (D) (iii) only

4. Identify the regular languages from the following:
(i) Set of strings of balanced parentheses
(ii) $\{0^n/n$ is a perfect square$\}$
(iii) $\{ww/w \in \{0, 1\}^*\}$
(A) (i), (ii) (B) (ii), (iii)
(C) (i), (iii) (D) None of these

5. Which of the following properties hold for regular languages?
(A) Complement (B) Difference
(C) Reversal (D) All the above

6. Which of the following is TRUE?
(A) Pushdown automata has same expressing power as Deterministic finite automata.
(B) NFA is an extension of pushdown automata.
(C) Pushdown automata is an extension of $\in$-NFA.
(D) PDA does not have any storage component.

7. Which type of language is accepted by DPDA?
(A) Regular languages only
(B) Context free languages only
(C) The class of languages between regular languages and context free languages.
(D) Recursively enumerable languages only

8. Which of the following statement is TRUE about Top down parsing and bottom up parsing?
(A) Bottom up parser is more powerful.
(B) Error detection in Top down parser is easy.
(C) Table size in Bottom up parser and Top down parser is approximately equal.
(D) Design complexity of Top down parser and Bottom up parser are same.

9. Limitation of L-Attributed Grammar is:
(A) uses only inherited Attributes.
(B) each inherited attribute inherits either from parent or Left side sibling
(C) each inherit attribute inherits either from parent or right side sibling
(D) each inherit attribute inherits either from parent or Left side sub tree.

10. Which of the following statement is false?
(A) It is always possible to rewrite a SDD to use only synthesized attributes.
(B) Evolution of L-Attributed grammar is BUP (Bottom up parsing)
(C) Evolution of S-Attribute grammar is BUP (Bottom up parsing)
(D) Evolution of L-Attributed grammar is Depth first. Traversal, left to right.

11. A grammar is said to be 'attribute grammar' if and only if it has
(A) L-Attribute definition
(B) S-Attribute definition
(C) SDT which does not have any side effects
(D) SDT which must have side effects

12. Which of the following statement is FALSE about L-Attributed definition?
(A) Traverse the parse tree in Breadth first manner from Left to Right.
(B) Evaluate inherited attributes if the node is visited for first time.
(C) Evaluate synthesized attributes if the node is visited for Last time.
(D) None of the above.

13. Which of the following is not an advantage of reverse polish notation?
(A) no rules of precedences
(B) no associativity
(C) no parenthesis
(D) not low level representation

14. A single tape Turing machine M has two states q_0 and q_1, of which q_0 is the starting state. The tape alphabet of M is $\{0, 1, B\}$ and its input alphabet is $\{0, 1\}$, the symbol B is the blank symbol used to indicate end of an input string. The transition function of M is described in the following table.

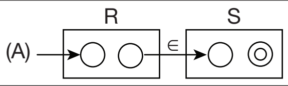
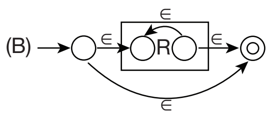
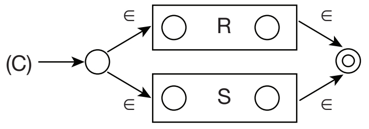
	0	1	B
q_0	$q_1, 1, R$	$q_1, 0, R$	q_1, B, L
q_1	$q_0, 1, R$	$q_0, 0, R$	Halt

Which of the following statements is TRUE about M?
(A) M accepts all strings that end with 0 only
(B) M accepts all strings that end with 1 only

(C) M accepts all strings of $(0 + 1)^+$ only.
(D) M accepts all strings of $(0 + 1)^*\ 01\ (0 + 1)^* + 1^*0^*$

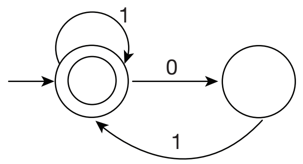
15. Which phase is most important for complex parts of any modern compiler?
(A) Syntax Analysis (B) Semantic Analysis
(C) Target code generator (D) Code optimizer

16. Match the following:

Regular expression	Finite Automata
(i) $R + S$	(A) 
(ii) RS	(B) 
(iii) R^*	(C) 

(A) (i)–(C), (ii)–(B), (iii)–(A)
(B) (i)–(A), (ii)–(B), (iii)–(C)
(C) (i)–(C), (ii)–(A), (iii)–(B)
(D) (i)–(B), (ii)–(A), (iii)–(C)

17. Consider below DFA:



What is the language accepted by this DFA?
(A) $1^* + (01)^*$ (B) $[1^*\ (01)^*]^*$
(C) $(1 + 01)^*$ (D) Both (B) and (C)

18. Consider the grammar:
$S \rightarrow AS/\in$
$A \rightarrow aa/ab/ba/bb$
What is the language generated by this grammar?
(A) Generates all the strings of the form $\{a^nb^n/n \geq 0\}$
(B) Generates all the strings of even length.
(C) Generates all the strings of equal number of a's and b's.
(D) Generates all the strings possible with $\{a, b\}^*$.

19. Which of the following is not Context-free language?
(A) $\{a^nb^nc^n/n \geq 1\}$ (B) $\{ww^R/w \in (0 + 1)^*\}$
(C) $\{wcw^R/w \in (0 + 1)^*\}$ (D) $\{a^nb^n/n \geq 1\}$

20. Which of the following are Deterministic context free language(s)?
(i) $\{0^n1^{m+n}\ 1^m/m, n \geq 0\}$
(ii) $\{0^n1^m0^n/n$ and m are arbitrary$\}$
(A) (i) only (B) (ii) only
(C) Both (i) and (ii) (D) Neither (i) nor (ii)

21. Let L and M be two context free languages then which of the following is not context free?
(A) $L \cup M$ (B) $L\,M$
(C) L^* (D) $L \cap M$

22. Which of the following are undecidable?
(A) Is a given CFG, G ambiguous?
(B) Is the intersection of two CFL's empty?
(C) Are two CFL's the same.
(D) All the above

23. Which languages are accepted by Turing machines only?
(A) Regular languages
(B) Context free languages
(C) Recursively enumerable languages
(D) All the above

24. Which of the following languages are accepted by Turing machines?
(i) set of all strings of balanced parentheses
(ii) $\{0^n1^m0^n1^m/m, n \geq 1\}$
(iii) $\{ww/w \in \{0, 1\}^*\}$
(A) (i), (ii) (B) (ii), (iii)
(C) (i), (iii) (D) (i), (ii), (iii)

25. Which of the following is FALSE?
(A) A problem L is decidable if it is a recursive language.
(B) A problem L is undecidable if it is not a recursive language.
(C) If L is a recursive language then $\bar{L}$ is also recursive.
(D) If L is recursively enumerable language then it is also recursive language.

26. Consider the following grammar:
$S \rightarrow Sd/aA/aB$
$A \rightarrow ab/d$
$B \rightarrow acd/dde$
Parse a string $w = adde$ using Brute force technique, How many back tracks are required?
(A) 6 (B) 4
(C) 0 (D) None of the above

27. Consider the following Grammar:
S → Aa/bAc/Bc/bBa
A → d/a/aB
B → d/e
Parse a string w = "bea" using Brute Force Technique, What is the maximum height of parse tree and how many back tracks are required?
(A) 2, 13 (B) 3, 16
(C) 3, 4 (D) 2, 4

28. In LL(1) parsing, while parsing a string 'w', the Top of the stack contains non terminal 'X' and input string look ahead symbol 'a' and LL(1) parsing Table contains $M[X, a]X \rightarrow aBc$

After performing appropriate action, how many symbols are there in stack and what is the Top of the stack?
(A) 4, a (B) 4, c
(C) 3, a (D) 3, c

29. Consider the following grammar:
$A \to X_1|X_2|\varepsilon$
here X_1 and X_2 are the productions of 'A'.
The grammar should be LL(1) grammar if and only if
(A) first(X_1) $\cap$ first (x_2) = ϕ
(B) first(X_1) = first(x_2)
(C) follow(A) $\cap$ first(x_1) $\neq \phi$
(D) first(X_1) $\cap$ first (x_2) $\neq \phi$

30. A non-terminal 'A' is deriving one production
$$A \to X_1 X_2 X_3,$$
here X_1, X_2, X_3 are non terminals, which have some productions. Which of the following statement always holds to find first(A)?
(A) first(A) = first(X_1) if ($X_1 \overset{*}{\Rightarrow} \in$)
(B) first(A) = first(X_1)$\cup$ first(X_2) $\cup$ first(X_3)
(C) first(A) = first(X_1)$\cup$ first(X_2) $\cup$ first(X_3) – {$\in$} if ($X_1 \overset{*}{\Rightarrow} \in$)
(D) first(A) = first(X_1) – {$\in$} $\cup$ first(X_2 X_3) if ($X_1 \overset{*}{\Rightarrow} \in$)

31. Consider the following Grammar:
S $\to$ AB/aAbAd
B $\to$ a/b/c/$\in$, find follow of the Non-terminal A?
(A) follow(A) = first(B) – {$\in$}
(B) follow(A) = first(B) $\cup$ {d}
(C) follow(A) = first(B) $\cup$ {\$} – {$\in$}
(D) follow(A) = first(B) $\cup$ {d, \$} – {$\in$}

32. While constructing LL(1) parsing Table, place production '$A \to \alpha$' in the following cells of $M[X, a]$, where 'a' is [first (α) contains '$\in$' and terminals]:
(A) only first (α) – {$\in$}
(B) first (α) $\cup$ follow(A)
(C) only in follow(A) – {$\in$}
(D) first (α) $\cup$ follow(A) – {$\in$}

33. In LL(1) parsing Table, $M[X, \$]$ contains production if and only if [here $X \to ABC$]
(A) first (X) contains {$\in$}
(B) first (A) contains {$\in$}
(C) first (AB) contains {$\in$}
(D) data is insufficient

34. Consider the following operator precedence Relational Table. For the string $W = id + id, \times id$, how many handles are possible to accept?

	id	+	*	$
id		•>	•>	•>
+	<•	•>	<•	•>
*	<•	•>	•>	•>
$	<•	<•	<•	

(A) 4 (B) 5
(C) 6 (D) Data insufficient

35. In the above precedence table, what is the associativity of '+' and '*'?
(A) + is right associative and * is left associative
(B) + and * Left associative
(C) + and * Right associative
(D) Data insufficient

Answer Keys

1. D	**2.** D	**3.** C	**4.** D	**5.** D	**6.** C	**7.** C	**8.** A	**9.** B	**10.** B
11. C	**12.** A	**13.** D	**14.** C	**15.** D	**16.** C	**17.** D	**18.** B	**19.** C	**20.** C
21. D	**22.** D	**23.** C	**24.** D	**25.** D	**26.** D	**27.** B	**28.** C	**29.** A	**30.** D
31. D	**32.** D	**33.** A	**34.** B	**35.** B					

Hints and Explanations

1. Commutative law for union, associative law for union and concatenation holds for regular expression. But commutative law for concatenation doesn't hold. Choice (D)

2. Choice (D)

3. Regular languages are accepted by FA. It has only limited storage, so it can't compare the front of a string with its back. Similarly it can't check the equality of 0's. Choice (C)

4. No language is regular (Not recognized by FA). Choice (D)

5. Choice (D)

6. PDA is an $\in$-NFA with the addition of a stack. Choice (C)

7. Choice (C)

8. Error detection in Top down parser is difficult compared to Bottom up parser. Choice (A)

9. S-Attribute grammar uses synthesized attributes. L-Attributed grammar uses both synthesized and inherited attributes. As it scans Left to right inherited attribute, it must inherit from parent or left hand side siblings. Choice (B)

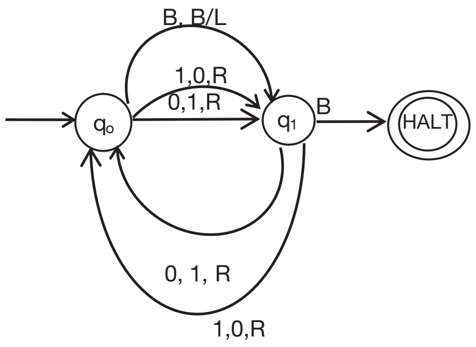
10. L-attributed grammar uses both synthesized and inherited attributes, it scans from Root node to leaf nodes in Depth first search, left to right manner. Choice (B)

11. A grammar is said to be attribute grammar if and only if it does not contain side effects like any action rule print something. Choice (C)

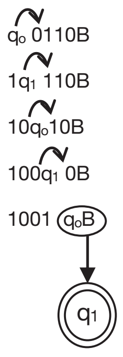
12. L-Attributed grammar Traverses parse tree in Depth first search manner, left to right. Choice (A)

13. Reverse polish notation, Abstract syntax tree, Direct acyclic graph are high level intermediate code forms, three address code is low level intermediate code form. Choice (D)

14.



For example, take string 0110 and check whether it is accepted
by TM or not



Final accepting state
∴ Any string of length 1 or more of {0, 1} is accepted by given TM. Choice (C)

15. In early age of compilers, developers focused on syntax analysis and recently, on optimization, focused mainly on code optimization and risc. Choice (D)

16. Choice (C)

17. Given DFA accepts any number of 1's and (01'*s*) repeatedly. The regular expression is (1 + 01)* = [1* (01)*]*. Choice (D)

18. $S \to AS/\in$
$A \to aa/ab/ba/bb$
It generates $\{a^n\ b^n\}$ but it generates $\{aa, aaaa, ...\}$ also. It also won't generate equal number of *a*'s and *b*'s. It won't generate single '*a*'. It generates even length strings. Choice (B)

19. The CFL's are accepted by PDA's. The PDA has a stack storage it moves only towards right side of string. It can't check equality of *a*'s *b*'s and *c*'s. But it can check $\{ww^R\}$, $\{wcw^R\}$, $\{a^n\ b^n\}$ patterns. Choice (C)

20. $\{0^n\ 1^{m+n}\ 0^m/m, n \in 0\}$
Push 0's whenever 1's are in input, pop 0's i.e., matching between 0's and 1's. After that push remaining 1's and pop them whenever 0's occur.
There is no non determinism. So it is DCFL.
Similarly we can check the equality of 0's in (ii) by pushing first set of 0's & pop them with last set of 0's. Choice (C)

21. CFL's are not closed under intersection. Choice (D)

22. Choice (D)

23. Recursively enumerable languages are only accepted by TM. Regular, CFL's are accepted by TM but they are also accepted by FA, PDA respectively. Choice (C)

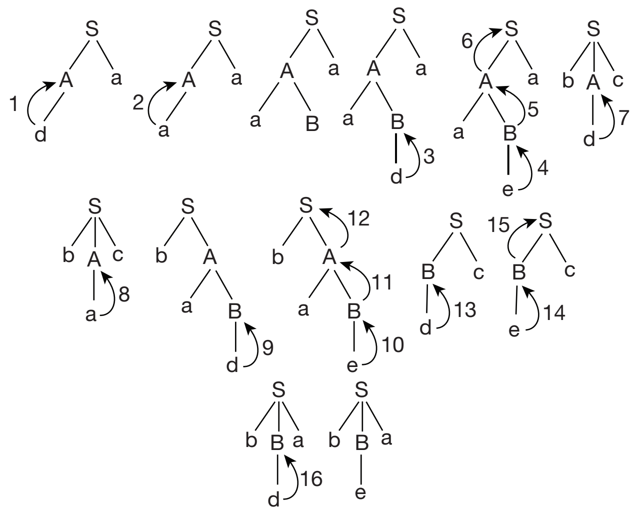
24. A TM has infinite storage buffer, which can move in two directions (forward, backward) and also it has both read and write capabilities.
All the given three languages are accepted by TM's. Choice (D)

25. A language L is recursive if both L and $\overline{L}$ are recursively enumerable. Choice (D)

26. Fall into infinite Loop in Brute Force Technique, Blindly substitute the first productions.
$S \Rightarrow Sd$
$\Rightarrow Sdd$
$\Rightarrow Sddd$
$\Rightarrow Sdddd$
:
:
Infinite Loop

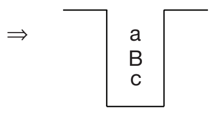
Choice (D)

27.



Choice (B)

28.



$M[X, a]$: $X \to aBc$
Replace X by aBc such that a is Top of the stack.

Choice (C)

29. In LL(1) parsing table construction, for a production '$A \to \alpha$'
Place the production in first (α) column and in 'A' row.
$A \to X_1|X_{21}$ $A \to X_1$, in first (x_1), and $A \to x_2$ in first (x_2).
So if we have common symbol then it is a conflict in LL(1) parser. Choice (A)

30. first(A) = first (X_1 X_2 X_3) = first(X_1), if first (X_1) contains '$\in$' then we have to take first (X_2). If first (x_2) also contains '$\in$' then go for first (x_3) also. If first (x_3) contains '$\in$' then it will be added to 'A's first.

Choice (D)

31. follow(A) = $\{a, b, c, d, \$\}$
follow(A) = first (B) $\cup$ first(b) $\cup$ first (d)
first(B) contains '$\in$' then 'A' is followed by '$\in$'. If a non terminal followed by '$\in$' then take follow of Left hand side non terminal i.e., follow(A) = follow(S)

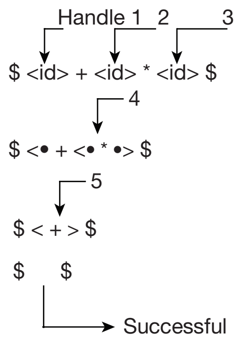
Choice (D)

32. In LL(1) parsing, place '$A \to \alpha$' in first (α) columns. If first (α) contains '$\in$' also place '$A \to \alpha$' in follow (A).

Choice (D)

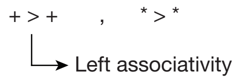
33. \$ is in follow set only. A production '$A \to \alpha$' placed in every symbol follow set if and only if first(α) contains '$\in$'. So '$X \to ABC$' placed in '\$' column if and only if first (ABC) contains '$\in$'. Choice (A)

34.



Choice (B)

35. In operator precedence parsing table θ_1 and θ_2 are two Operators and have equal precedence then
$\theta_1 < \theta_2$ if Right associative
$\theta_1 > \theta_2$ if Left associative
So



Choice (B)