Compiler Design Test I

Number of Questions: 25 **Section Marks: 30**

Directions for questions 1 to 25: Select the correct alternative from the given choices.

1. Which of the following implementation of Lexical analyzer is easy to modify and faster?
(i) Lexical-analyzer by hand
(ii) Lexical-analyzer generator
(A) (i) only (B) (ii) only
(C) Both (i) and (ii) (D) Neither (i) nor (ii)

2. Which of the following tasks are performed by a Lexical analyzer?
(i) Identification of Lexemes
(ii) Stripping-out Comments
(iii) Stripping-out white spaces
(iv) Correlating error messages generated by the compiler with the source program.
(A) (i), (ii) (B) (i), (ii), (iii)
(C) (iii), (iv) (D) (i), (ii), (iii), (iv)

3. Which of the following statement is TRUE?
(i) Every LL(k) language is deterministic context-free language.
(ii) Every deterministic context free language is LL(k) language.
(A) (i) only (B) (ii) only
(C) Both (i) and (ii) (D) Neither (i) nor (ii)

4. Which of the following grammar is LL(3) grammar but not LL(2) and not LL(1)?
(A) $S \to aS|b$ (B) $S \to aaS \mid ab \mid b$
(C) $S \to aaaS \mid aab \mid ab \mid b$ (D) None of the above

5. On errorneous input,
(A) LALR parser makes more moves than LR parser.
(B) LR parser makes more moves than LALR parser.
(C) LALR parser makes equal number of moves as an LR parser.
(D) Unpredictable

6. What kind of conflict does the following grammar leads to?
$S \to A$
$A \to xA \mid yA \mid y$
(A) Shift-reduce conflict
(B) Reduce-reduce conflict
(C) No conflict exist
(D) Both (A) and (B)

7. Which of the following statement(s) is/are TRUE?
(i) Synthesized attributes may be calculated only from attributes of children.
(ii) Inherited attributes may be calculated only from attributes of parents.
(A) (i) only (B) (ii) only
(C) Both (i) and (ii) (D) Neither (i) nor (ii)

8. Which of the following action is not performed by an LL parser?
(i) Matching top of the parser stack with next input token.
(ii) Predicting a production and apply it in a derivation step.
(iii) Accept and terminate the parsing of a sequence of tokens.
(iv) Reporting an error message.
(A) (i), (iv) (B) (iii), (iv)
(C) (ii), (iii) (D) None of the above

9. Consider the following grammar:
$S \to AcB$
$A \to aA \mid \varepsilon$
$B \to bBS \mid \varepsilon$
For this grammar, FOLLOW(B) is
(A) $\{a, \$\}$ (B) $\{a, c\}$
(C) $\{a, c, \$\}$ (D) $\{c\}$

10. Which of the following statements is FALSE?
(A) Every LR(0) grammar is in SLR (1).
(B) Every SLR (1) grammar is in LR (0).
(C) Every SLR(1) grammar is a canonical LR(1) grammar.
(D) Every LR(1) grammar is not necessarily SLR(1).

11. Which of the following grammars for $\{a^n b^{n+k} | n, k \in N\}$ are in LL(1)?
(i) $S \to PQ$
$P \to aPb \mid \varepsilon$
$Q \to bQ \mid \varepsilon$
(ii) $S \to aSb| T$
$T \to bT| \varepsilon$
(A) (i) only (B) (ii) only
(C) Both (i) and (ii) (D) Neither (i) nor (ii)

12. Based on which of the following parameters does a LR(k) parser performs reduction?
(i) The complete left context
(ii) The reducible phrase itself
(iii) The k-terminal symbols to its right
(A) (i), (iii) only (B) (ii), (iii) only
(C) (i), (ii),(iii) (D) (i) only

13. After removing left-recursion from the following grammar the resultant grammar will be:
$P \to P + Q \mid Q$
$Q \to \text{int} \mid (P)$
(A) $P \to Q + Q$
$Q \to \text{int} \mid (P)$
(B) $P \to QP^1$
$P^1 \to +QP^1 \mid \varepsilon$
$Q \to \text{int} \mid (P)$

(C) $P \to QP^1$
$P^1 \to +QP^1$
$Q \to$ int | (P)

(D) $P \to Q + P^1$
$P^1 \to Q + P^1 \mid \varepsilon$
$Q \to$ int | (P)

14. Consider the following grammar:
$S \to PQRt$
$P \to a \mid b \mid \varepsilon$
$Q \to c \mid d \mid \varepsilon$
$R \to e \mid f$
For this grammar, FIRST(S) is
(A) $\{a, b, c, d, e, f, \varepsilon\}$
(B) $\{a, b, c, d, e, f\}$
(C) $\{a, b, t\}$
(D) $\{a, b, c, d\}$

15. Consider the following grammar, G:
$S \to (L) \mid a$
$L \to L, S \mid S$
Which of the following is TRUE?
(i) G is not suitable to be parsed using recursive descent parsing.
(ii) FIRST (S) = FIRST (L) = $\{(, a\}$
(A) (i) only
(B) (ii) only
(C) Both (i) and (ii)
(D) Neither (i) nor (ii)

16. Consider the following configuration:
$A \to X_1 X_2 \ldots X_j, a$
Then which of the following is TRUE?
(i) With SLR(1) parsing we would reduce if the next token was any of those in FOLLOW (A).
(ii) With LR(1) parsing, we reduce only if the next token is exactly 'a'.
(A) (i) only
(B) (ii) only
(C) Both (i) and (ii)
(D) Neither (i) nor (ii)

17. Which of the following is FALSE?
(A) An LALR(1) grammar is one which allows a LALR(1) parse table to be constructed with no more than one action per cell.
(B) LALR(1) grammar allow a wider set of structures than SLR(1) grammar.
(C) LALR(1) grammars do not handle some cases allowed in LR(1) grammars.
(D) None of these

18. Identify the correct statements from the following:
(i) LALR(1) is a subset of LR(1).
(ii) LALR(1) is a super set of SLR(1).
(iii) A grammar that is not LR(1) may LALR(1).
(A) (i), (ii) only
(B) (ii), (iii) only
(C) (i), (iii) only
(D) (i), (ii), (iii)

19. Based on the following grammar,
$E \to E + T \mid T$
$T \to T * F \mid F$
$F \to (E) \mid id$
What is the closure $(\{T \to T \bullet * F\}$?
(A) $\{T \to T \bullet * F\}$
(B) $\{T \to T \bullet * F \; T \to \bullet F\}$
(C) $\{T \to T * \bullet F \; F \to \bullet (E) \; F \to \bullet id\}$
(D) $\{T \to \bullet T * F\}$

20. Consider the following grammar:
Stmts' → Stmts
Stmts → Stmt | Stmts; Stmt
Stmt → $Id = E$ | Print E
$E \to Id \mid E + E \mid (E)$
What does the closure set of the following rule will include?
Stmts' → • Stmts
(A) Stmts' → • Stmts
(B) Stmt's → • Stmts
Stmts → • Stmt
Stmts → • Stmts ; Stmt
Stmt → • $id = E$
Stmt → • Print E
(C) Stmts' → • Stmts
Stmts → • Stmt
Stmts → • Stmts ; Stmt
Stmt → • $id = E$
Stmt → • Print E
$E \to \bullet\, id$
$E \to \bullet\, E + E$
$E \to \bullet\, (E)$
(D) Stmts' → • Stmts
Stmts → • Stmt
Stmts → • Stmts; Stmt

21. For the following grammar:
$A \to BCD$
$B \to Bb \mid \varepsilon$
$C \to cC \mid \varepsilon$
$D \to d$
What is FOLLOW (B)?
(A) $\{b, \$\}$
(B) $\{b, c\}$
(C) $\{b, d\}$
(D) $\{b, c, d\}$

22. Consider the following grammar, G:
$S \to AB$
$A \to aAa \mid \varepsilon$
$B \to bBb \mid \varepsilon$
Which of the following statements is TRUE?
(i) G is ambiguous.
(ii) G is not in LL(1).
(A) (i) only
(B) (ii) only
(C) Both (i) and (ii)
(D) Neither (i) nor (ii)

23. Consider the following grammar:
$S \to a \mid AbC$
$A \to a$
$C \to A \mid c$
Which of the following is FALSE?
(A) FOLLOW(S) = $\{\$\}$
(B) FOLLOW(A) = $\{b, \$\}$
(C) FOLLOW(C) = $\{\$\}$
(D) The grammar is in SLR(1).

24. Which of the following is TRUE about LR-attributed grammar?
 (i) Attributes can be evaluated in LR parsing
 (ii) LR-Attribute grammars are a subset of L-attributed grammars
 (iii) LR-attribute grammars are a subset of S-attributed grammars.
 (A) (i), (ii) only (B) (ii), (iii) only
 (C) (i), (iii) only (D) (i), (ii), (iii) only

25. Which of the following conditions must hold to call a grammar is in SLR(1)?
 (i) For any production $A \to x.yz$ in the set, with terminal y, there is no complete item $B \to W$. In that set with 'y' in FOLLOW (B).
 (ii) For any two complete productions $A \to X$ and $B \to Y$. In the set, the follow sets must be disjoint.
 (A) (i) only (B) (ii) only
 (C) Both (i) and (ii) (D) Neither (i) nor (ii)

Answer Keys

1. B	**2.** D	**3.** A	**4.** C	**5.** A	**6.** A	**7.** A	**8.** D	**9.** C	**10.** B
11. A	**12.** C	**13.** B	**14.** B	**15.** C	**16.** C	**17.** D	**18.** A	**19.** A	**20.** B
21. D	**22.** B	**23.** D	**24.** A	**25.** C					

Hints and Explanations

1. Lexical-analyzer generator makes the code easier to modify a lexical analyzer (Since we have to rewrite only the affected patterns).
It also speeds up the process of implementing the analyzer (Since the programmer specifies the software at the very high level of patterns and relies on generator to produce code). Choice (B)

3. LL(k) languages are Deterministic CFLs but there are Deterministic CFL's that are not LL(k). Choice (A)

4. $S \to aS/b$ is in LL(1)
($\because$ By looking single look-ahead, we can decide whether to shift or reduce).
$S \to aaS|ab|b$ is in LL(2).
$S \to aaaS|\ aab|\ ab|\ b$ is in LL(3). Choice (C)

5. On errorneous input, LALR parser makes more moves (i.e., reductions) than the LR parser. Choice (A)

6. Given grammar,
$S \to A$
$A \to xA \mid yA \mid y$
This grammar has shift-reduce conflict.
$A \to y\ .\ A$ || leads to shift
$A \to y.$ || leads of reduce
These productions leads to shift-reduce conflict.
Choice (A)

7. Synthesized attributes are calculated from attributes of children.
Inherited attributes are calculated from attributes of parents or siblings. Choice (A)

8. An LL parser perform the following actions:
 (i) Match (ii) Predict
 (iii) Accept (iv) Error Choice (D)

9. Given grammar,
$S \to AcB$
$A \to a\,A \mid \varepsilon$
$B \to bBS|\ \varepsilon$
FOLLOW (B) = FIRST (S) U FOLLOW (S)
FIRST (S) = {FIRST (A) – ε} ∪ {c} = {a, c}
Follow (S) = {\$}
$\therefore$ Follow (B) = {a, c, \$} Choice (C)

10. LR(0) ⊂ SLR (1) ⊂ CLR (1) Choice (B)

11. Given language:
$\{a^n\, b^{n+k} \mid n, k \in N\}$
 (i) $S \to PQ$
 $P \to aPb|\ \varepsilon$
 $Q \to bQ|\ \varepsilon$
 This is LL(1) (No ambiguity, no left recursion).
 To derive 'ab':
 $S \to PQ$
 $\to aPbQ$
 $\to ab$.
 (ii) $S \to aSb \mid T$
 $T \to bT \mid \varepsilon$
 To derive 'ab' :
 $S \to aSb$
 $\to aTb$
 $\to abTb$ ($T \to bT$)
 (Back track)
 (or)
 $S \to aSb$
 $\to aTb$
 $\to ab(T \to \varepsilon)$
$\therefore$ (ii) not in LL(1) (as which production to use in some derivation is unknown). Choice (A)

12. All the three parameters are required during reduction in LR(k) parser. Choice (C)

13. Given grammar,
$P \to P + Q \mid Q$

$Q \rightarrow$ int | (P)

To avoid left-recursion, replace recursive term with other terms and introduce a new variable as shown below.

$P \rightarrow P + Q \mid Q$ can be written as

$P \rightarrow QP^1$

$P^1 \rightarrow + QP^1 \mid \varepsilon$ Choice (B)

14. FIRST (S) = {FIRST (P) – ε} U {FIRST (Q) – ε} U {FIRST (R)}

= $\{a, b, c, d, e, f\}$. Choice (B)

15. Given grammar is not parsed using recursive descent parsing as the grammar is left recursive.

FIRST (S) = {(, a}

FIRST (L) = FIRST (S) = {(, a} Choice (C)

16. For the configuration,

$A \rightarrow X_1, X_2, \ldots X_j, a$

If 'a' is in FOLLOW(A) then reduce in SLR(1).

If the next token is exactly 'a' then reduce in LR(1).

Choice (C)

18. SLR(1) ⊂ LALR (1) ⊂ LR(1)

If a grammar is not in LR(1) then it is not in LALR(1).

(∵ Conflict exist in LALR also) Choice (A)

19. Given grammar

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

Closure ({$T \rightarrow T \bullet * F$}) will be {$T \rightarrow T \bullet * F$} only.

To calculate closure we add new productions to the set if the • is before a non-terminal.

(Here • is before '∗' which is a terminal)

Choice (A)

20. Given grammar,

Stmts' → Stmts

Stmts → Stmt | Stmts; Stmt

Stmt → id = E| Print E

$E \rightarrow id \mid E + E \mid (E)$

Closure ({Stmts' → • Stmts}),

will include the productions of Stmts (∵ • is before Stmts)

Stmts → • Stmt

Stmts → •Stmts; Stmt

Now production of 'Stmt' will be included (∵ • is before Stmt)

Stmt → •id = E

Stmt → • Print E Choice (B)

21. Follow (B) = {b} ∪ {FIRST (C) – ∈} ∪ {FIRST (D)}

= {b} ∪ {c} ∪ {d}

= {b, c, d} Choice (D)

22. Given grammar G:

$S \rightarrow AB$

$A \rightarrow aAa \mid \varepsilon$

$B \rightarrow bBb \mid \varepsilon$

G is not ambiguous.

(∵ To derive any string only one possible method exists)

G is not in LL(1).

To derive aa.

$S \rightarrow AB$

$\rightarrow aA\ aB$

$\rightarrow aa\ A\ aaB$ (or) $S \rightarrow AB$

(Back track)	$\rightarrow aA\ aB$
$\rightarrow aAa\ B$	$\rightarrow aaB$
$\rightarrow aa$	$\rightarrow aa$

∴ G is not in LL(1).

[OR]

To check whether G is in LL(1) or not:

Check whether for all $A \in N$ with $A \rightarrow \alpha_1 \mid \alpha_2 \mid \ldots \alpha_n$ being all A-Productions in G, the following holds:

(i) FIRST(α_1),………FIRST(α_n) are pairwise disjoint.

(ii) If ∈ ∈ FIRST(α_j) for some $j \in [1, n]$ then FOLLOW (A) η FIRST (α_i) = ϕ for all $1 \le i \le n, j \ne i$.

(i) is satisfied.

FIRST (aAa) = {a}

FIRST (ε) = {ε}

(ii) is failed.

FIRST ($aA\ a$) = {a}

FOLLOW (A) = {a, b, $\$$}

These are not disjoint.

∴ G is not in LL(1). Choice (B)

23. Given grammar,

$S \rightarrow a \mid AbC$

$A \rightarrow a$

$C \rightarrow A \mid c$

FOLLOW (S) = {\$}

FOLLOW (A) = {b} ∪ follow (C) = {b, \$}

FOLLOW (C) = FOLLOW (S) = {\$}

The grammar is not in SLR(1) , since there is reduce/reduce conflict with the productions.

$S \rightarrow a$.

$A \rightarrow a$. Choice (D)

24. LR-attribute grammar is a special type of attribute grammar. It allows the attributes to be evaluated on LR parsing.

LR-attribute grammar ⊂ L-attributed grammar.

S-attributed grammar ⊂ LR-attributed grammar.

Choice (A)

25. (i) avoids shift-reduce conflict.

(ii) avoids reduce-reduce conflict. Choice (C)