OBJECTIVE TYPE QUESTIONS

Ⓡ 1. Can context-sensitive grammar be used for representing the programming language?

(a) Yes (b) No

Ⓡ 2. The output of the parser is

(a) Tokens (c) Syntax tree

(b) Parse tree (d) Non-terminals

Ⓤ 3. Production rules can be written

(a) Only in recursive manner

(b) Only in non-recursive manner

(c) Both in recursive and non-recursive manners

(d) None of these

Ⓡ 4. In the production rule $A \rightarrow \alpha$, the head and definition of the rule are

(a) Terminal, non-terminal

(b) Non-terminal, terminal

(c) Non-terminal, grammar symbol

(d) Non-terminal, string of grammar symbols

Ⓐ 5. Consider the following grammar:

$S \rightarrow aAbS$ $A \rightarrow bSaA$

$S \rightarrow e$ $A \rightarrow c \mid d$

The language produced by the grammar is

(a) *abecbe* (c) *abebedbe*

(b) *abebecbe* (d) None of these

Ⓤ 6. Which one of the following grammar is more powerful?

(a) Regular grammar

(b) Context-sensitive grammar

(c) Context-free grammar

(d) Unrestricted grammar

Ⓡ 7. Parsing is done using derivations in

(a) top-down parser (c) All of these

(b) bottom-up parser (d) None of these

Ⓡ 8. Ambiguities in the grammar yields

(a) more than two parse trees

(b) more than one parse tree

(c) only one parse tree

(d) no parse tree

Ⓡ 9. Handle is

(a) head of the grammar rule

(b) definition of the grammar rule

(c) set of non-terminals

(d) set of terminals

Ⓡ10. Recursive descent parser is

(a) bottom-up parser (c) All of these

(b) top-down parser (d) None of these

Ⓡ11. Predictive parser is

(a) LL(1) parser (c) LR(1) parser

(b) LR(0) parser (d) LL(0) parser

Ⓡ12. Backtracking in the top-down parser

(a) goes to the previous expansion of rule

(b) goes to the start symbol of the grammar where there are alternative definitions

(c) goes to the scanner and tries with other grammar

(d) goes to the previous expansion of rule where there are alternative definitions

Ⓡ13. After elimination of the left recursion from the grammar $A \rightarrow Ab \mid c$, it gives

(a) $A \rightarrow cA^1, A^1 \rightarrow bA^1 \mid \varepsilon$

(b) $A \rightarrow bA^1, A^1 \rightarrow cA^1 \mid \varepsilon$

(c) $A \rightarrow cA^1, A^1 \rightarrow cA^1 \mid \varepsilon$

(d) $A \rightarrow bA^1, A^1 \rightarrow bA^1 \mid \varepsilon$

Ⓐ14. The grammar *exp* → *exp-exp* | *exp/exp* | ***id*** can be rewritten as

(a) *exp* → *exp A* | *id, A* → *-exp* | */exp*

(b) *exp* → *exp B* | *id, B* → *-exp* | */exp*

(c) All of these

(d) None of these

Ⓐ15. Consider a grammar $A \rightarrow X_1X_2X_3X_4$. If, X_1 is a non-terminal and X_2 produces ε, then follow of X_1 is

(a) *first*(X_1) (c) *first*(X_3)

(b) *first*(X_2) (d) *first*(X_4)

Ⓤ16. Is the grammar $A \rightarrow X_1X_2X_3X_4$ an operator grammar when X_1 through X_4 are non-terminals?

(a) Yes (b) No

Ⓡ17. The basic actions in the parsing process of any bottom-up parser are

(a) shift, produce, accept, reject

(b) shift, produce, accept, reject

(c) rotate, produce, accept, reject

(d) shift, reduce, accept, reject

Ⓤ18. LR stands for

(a) left to right

(b) left to right, rightmost derivation in reverse

(c) left to right, rightmost derivation

(d) left to tight, right to left

Ⓤ19. The LR(0) item $B \rightarrow a.\gamma$ refers to

(a) the parser has completed the parsing

(b) the parser is about to start parsing the derivatives of γ

(c) the parser has parsed a and is about to start parsing the derivatives of γ

(d) the parser is about to start parsing the derivatives of γ

Ⓤ20. The LR item $\{B \rightarrow a.\gamma, b\}$ is

(a) LR(0) item (c) LR(2) item

(b) LR(1) item (d) None of these

Ⓤ21. The grammar $S \rightarrow L = R$ leads to the presentation of

(a) arithmetic expression

(b) left to right expression

(c) shift from left to right expression

(d) an assignment statement

Ⓤ22. In the LR(1) item $[A \rightarrow \alpha.\beta, b]$, which one of the following is the core item?

(a) $A \rightarrow \alpha.$ (c) $A \rightarrow \alpha.\beta, b$

(b) $A \rightarrow \alpha.\beta$ (d) $A \rightarrow .\beta$

Ⓡ23. Relatively, the number of states is high in

(a) SLR parser (c) LALR parser

(b) CLR parser

Ⓡ24. Yacc is

(a) yet another compiler-computer

(b) yet another computer-compiler

(c) yet another computer-computer

(d) yet another compiler-compiler

Ⓡ25. In using Yacc tool, which one of the following sections is mandatory?

(a) Definition section

(b) User subroutine section

(c) Rules section

(d) None of these