DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING AND INFORMATION TECHNOLOGY
YEAR/SEM: III/VI
23CS6401/23IT6401 – COMPILER DESIGN
UNIT II
SYNTAX ANALYSIS

QUESTION BANK

Introduction-Context free grammars –Writing a grammar- Top down parser-Bottom up parser- LR parsers-Constructing an SLR parsing table-LALR parser-CLR parser- Error Handling and Recovery in Syntax Analyzer - YACC

PART - A (2 MARKS)

INTRODUCTION

1.	Define syntax analysis in a compiler.	K1
2.	What is meant by a parser? List the types of parsers.	K1
3.	Mention the basic issues in parsing.	K2
4.	Distinguish between a parse tree and a syntax tree.	K2
5.	What is the need for a parser in compilation?	K2
6.	Why are lexical analysis and syntax analysis separated?	K2

CONTEXT FREE GRAMMAR - WRITING A GRAMMAR

7.	Define context-free grammar.	K1
8.	What is a sentential form and the yield of a parse tree?	K1
9.	What is meant by an ambiguous grammar? Why must an ambiguous grammar be eliminated?	K2
10.	What is left recursion in a grammar? State the rule to eliminate left recursion.	K1
11.	Define left factoring and viable prefixes.	K1
12.	Why does left recursion cause problems in top-down parsing?	K2
13.	What is the dangling else problem?	K1

TOP DOWN PARSER - BOTTOM UP PARSER

14.	Differentiate between a top-down parser and a bottom-up parser.	K2
15.	Why does bottom-up parsing use rightmost derivation in reverse?	K2
16.	What are the difficulties with top-down parsing?	K2
17.	Why is backtracking avoided in LL(1) parsing?	K2
18.	What is meant by a recursive-descent parser?	K1
19.	Define handle and handle pruning.	K1

LR PARSERS- CONSTRUCTING AN SLR PARSING TABLE - LALR PARSER - CLR PARSER

20.	Why are LR parsers more powerful than LL parsers?	K2
21.	Define an LR parser. Why is an LR parser used?	K2
22.	What are the drawbacks of an LR parser?	K2
23.	How do operator precedence and associativity help in resolving ambiguity in expressions?	K2
24.	What are the actions in shift-reduce parsing?	K1

25.	What is shift-reduce parsing? Identify the important rules used in shift-reduce parsing.	K1
26.	Define LALR grammar.	K1
27.	What is meant by operator grammar? Give an example.	K1

ERROR HANDLING AND RECOVERY IN SYNTAX ANALYZER

28.	What are the different levels of a syntax error handler?	K1
29.	List the goals of an error handler used in syntax analysis.	K2

YACC

30.	What are the parts of a YACC program?	K1

PART - B (16 Marks)

1.(i)	Consider the context-free grammar. S → SS + | SS* | a and the string aa + a* a) Give a leftmost derivation for the string. b) Give a rightmost derivation for the string. c) Give a parse tree for the string. d) Is the grammar ambiguous or unambiguous? Justify your answer. (6)	10	K3
1.(ii)	Eliminate the left recursion for the grammar: E→ E + T | T; T → T * F | F; F → (E) | id	6	K3
2.(i)	Consider the following CFG: S → aABe A → Abc | b B → d Terminals: a,b,c,d Non-terminals: S,A,B Starting Symbol: S (a) Parse the sentence "abbcde" using right-most derivation. (b) Parse the sentence "abbcde" using left-most derivation. (c) Draw the parse tree.	8	K4
2.(ii)	Analyze the ambiguity of the grammar E → E + E | E * E | (E) | −E | id with respect to the sentence “id + id * id”.	8	K4
3.(i)	Eliminate left recursion and left factoring for the following grammar. E → E + T | E - T | T T → a | b | (E)	10	K3
3.(ii)	Construct a parse tree for the input string *w*=cad using a top-down parser; S→ cAd A→ ab | a	6	K3
4.	Construct the FIRST and FOLLOW sets and the predictive parse table for the grammar: S → AC$ C → c | Ɛ A → aBCd | BQ | Ɛ	16	K3

	B → bB | d Q → q		
5.(i)	Construct LR (0) items for the following grammar, G : S → S+R | R R → R*T | T T → (S) | i	8	K3
5.(ii)	Find the FIRST and FOLLOW sets for the grammar: E → TE′ E → +TE′ | ε T → FT′ T′ → *FT′ | ε F → (E) | id	8	K3
6.(i)	Show that the following grammar is LL(1): S → AaAb ⎪ BbBa A → Ɛ B → Ɛ	8	K4
6.(ii)	Analyze the given grammar and determine the FIRST and FOLLOW sets for all non-terminals, and justify the impact of nullable non-terminals on predictive parsing decisions. A → BC | EFGH | H B → b C → c | ε E → e | ε F → CE G → g H → h | ε	8	K4
7.(i)	Construct a predictive parsing table for the grammar: S→a| ↑ | (T) T→T, S | S Show how the string "(a, a)" is parsed using the predictive parser.	8	K4
7.(ii)	Analyze the following grammars, find the problems that prevent recursive-descent parsing, modify them if needed, and then write the recursive-descent parser procedures. a) S→+SS ǀ −SS ǀ a b) S→S(S)S ǀ ε	8	K4
8.	Analyze the non-recursive implementation of a predictive parser for the given grammar by examining the role of the parsing stack, input buffer, and parsing table. E → E + E | T T → T * F | F F → (E) | id	16	K4
9.(i)	Write the LR parsing algorithm and check whether the grammar is SLR(1). Justify your answer. S → L=R | R L → *R | id	8	K4

	R → L		
9.(ii)	Consider the Context-Free Grammar (CFG) depicted below, where "begin", "end" and "x" are all terminal symbols of the grammar and Stat is considered the starting symbol for this grammar. Productions are numbered in parentheses, and you can abbreviate "begin" to "b" and "end" to "e" respectively. (1) Stat → Block (2) Block → begin Block end (3) Block → Body (4) Body → x (a) Compute the set of LR(1) items for this grammar and draw the corresponding DFA. Do not forget to augment the grammar with the initial production S → Stat $ as the production (0). (b) Construct the corresponding LR parsing table.	8	K4
10.	Given the grammar S → AS | b A → SA | a Construct an SLR parsing table for the string "baab."	16	K3
11.	Construct an SLR parsing table for the grammar: E → E + T | T T → TF | F F → F* | a | b	16	K3
12.	Consider the grammar: S → L=R S → R L → *R L → id R → L Analyze the LALR parsing method for the above grammar. List the canonical collection of items and construct the parse table.	16	K4
13.	Construct the a) canonical LR, and b) LALR sets of items for the grammar S → SS+ | SS* | a	16	K4
14.	Construct a CLR parsing table for the grammar: S → AA A → Aa | b	16	K3
15.(i)	Explain the error handling and error recovery mechanism of syntax analyser.	10	K2
15.(ii)	Differentiate SLR, LALR, and CLR parsers.	6	K2
16.	Explain the YACC parser generator in detail.	16	K2