ST.MARY'S GROUP,HYDERABAD

Unit--1

| | |
|---|---|
| 1 | **Define** Complier briefly? |
| 2 | **Explain** the cousins of compiler? |
| 3 | **Define** the two main parts of compilation? What they perform? |
| 4 | **How** many phases does analysis consists? |
| 5 | **Define** and explain the Loader? |
| 6 | **Explain** about preprocessor? |
| 7 | **State** the general phases of a compiler? |
| | **State** the rules, and define regular expression? |
| 9 | **Explain** a lexeme and Define regular sets? |
| 10 | **Explain** the issues of lexical analyzer? |
| 11 | **State** some compiler construction tools? |
| 12 | **Define** the term Symbol table? |
| 13 | **Define** the term Interpreter? |

| **S. No.** | **Questions** |
|---|---|
| 14 | **Define** the term Tokens in lexical analysis phase? |
| 15 | **Explain** about error Handler? |
| 16 | **Define** a translator and types of translator? |
| 17 | **Explain** about parser and its types? |
| 18 | **Construct** NFA for (a/b)* and convert into DFA? |
| 19 | **Define** bootstrap and cross compiler? |
| 20 | **Define** pass and phase? |
| 21 | **Analyze** the output of syntax analysis phase? What are the three general types of parsers for grammars? |
| 22 | **List** the different strategies that a parser can employ to recover from a syntactic error? |
| 23 | **Explain** the goals of error handler in a parser? |
| 24 | **How** will you define a context free grammar? |
| 25 | **Define** context free language. When will you say that two CFGs are equal? |
| 26 | **Give** the definition for leftmost and canonical derivations? |
| 27 | **Define** a parse tree? |
| 28 | **Explain** an ambiguous grammar with an example? |
| 29 | **When** will you call a grammar as the left recursive one? |
| 30 | **List** different types of compiler? |

## (LONG ANSWER QUESTIONS)

| | |
|---|---|
| 1 | **Define** compiler? State various phases of a compiler and explain them in detail. |
| 2 | **Explain** the various phases of a compiler in detail. Also Write down the output for the following expression after each phase  a: =b*c-d. |
| 3 | **Explain** the cousins of a Compiler? Explain them in detail. |
| 4 | **Describe** how various phases could be combined as a pass in a compiler? Also briefly explain Compiler construction tools. |
| 5 | **For** the following expression<br>Position:=initial+ rate*60<br>Write down the output after each phase |
| 6 | **Explain** the role Lexical Analyzer and issues of Lexical Analyzer. |
| 7 | **Differentiate** the pass and phase in compiler construction? |
| 8 | **Explain** single pass and multi pass compiler? with example? |
| 9 | **Define** bootstrapping concept in brief? |
| 10 | **Explain** the general format of a LEX program with example? |
| 11 | **Construct** the predictive parser the following grammar:<br>S->(L)\|a<br>L->L,S\|S.<br>Construct the behavior of the parser on the sentence (a, a) using the grammar specified above |
| 12 | **Explain** the algorithm for finding the FIRST and FOLLOW positions for a given non-terminal.<br>ii) Consider the grammar,<br>E ->TE‟<br>E->+TE‟\|@<br>T ->FT‟<br>T->*FT‟\|@<br>F->(E)\|id.<br>**Construct** a predictive parsing table for the grammar given above. Verify whether the input string id + id * id is accepted by the grammar or not. |
| 13 | **Prepare** the predictive parser for the following grammar:<br>S->a\|b\|(T)<br>T ->T, S\|S<br>Write down the necessary algorithms and define FIRST and FOLLOW. Show the behavior of the parser in the sentences,<br>i.<br>(a,(a,a))<br>ii.<br>(((a,a),a,(a),a) |
| 14 | **Explain** operator grammar? Draw the precedence function graph for the following table. |

| | |
|---|---|
| 15 | **Analyze** whether the following grammar is LR(1) or not. Explain your answer with reasons.<br>S → L,R<br>S → R<br>L → * R<br>L → id<br>R → L. |

## UNIT – II

| | |
|---|---|
| 1 | **Define** the term handle used in operator precedence? |
| 2 | **Define** LR(0) items in bottom up parsing? |
| 3 | **State** the disadvantages of operator precedence parsing? |
| 4 | LR(k) parsing stands for what? |
| 5 | **Why** LR parsing is attractive one and explain? |
| 6 | **Define** goto function in LR parser with an example? |
| 7 | **Why** SLR and LALR are more economical to construct Cananonical LR? |
| 8 | **Explain** about handle pruning? |
| 9 | **Explain** types of LR parsers? |
| 10 | **List** down the conflicts during shift-reduce parsing. |
| 1 | **Consider** the grammar E  E + E\|E *E\|(E)\| id<br>Show the sequence of moves made by the shift-reduce parser on the input id1+id2*id3 and determine whether the given string is accepted<br>by the parser or not. |
| 2 | i) **State** shift-reduce parsing? Explain in detail the conflicts that may occur during shift-reduce parsing.<br>ii)For the grammar given below, calculate the operator precedence relation and the precedence functions<br>E → E + E\|E- E\|E * E\|E / E\|E  E\|(E)\|-E\|id |

## (LONG ANSWER QUESTIONS)

| | |
|---|---|
| 1 | **Consider** the grammar E  E + E\|E *E\|(E)\| id<br>Show the sequence of moves made by the shift-reduce parser on the input id1+id2*id3 and determine whether the given string is accepted<br>by the parser or not. |
| 2 | ) **State** shift-reduce parsing? Explain in detail the conflicts that may occur during shift-reduce parsing.<br>ii)For the grammar given below, calculate the operator precedence relation and the precedence functions<br>E → E + E\|E- E\|E * E\|E / E\|E  E\|(E)\|-E\|id |
| 3 | **repare a canonical parsing table for the grammar given below**<br>**S→CC**<br>**C→Cc/Dc** |
| 4 | **nalyze whether the following grammar is SLR(1) or not. Explain your answer with reasons.**<br>**S→L,R**<br>**S→R**<br>**L→*R**<br>**L→id** |
| 5 | **onsider the grammar given below.**<br>**E→E+T**<br>**E→T**<br>**T→T*F**<br>**T→F**<br>**F→(E)/id**<br>**repare LR parsing table for the above grammar .Give the moves of LR parser on id * id + id**<br>**i) Briefly explain error recovery in LR parsing.** |