CS8602 COMPILER DESIGN

IMPORTANT QUESTIONS AND QUESTION BANK

UNIT I -INTRODUCTION TO COMPILERS

2-Marks

1. Define tokens, patterns and lexemes?
2. Apply the regular expression for identifier and white space?
3. Point out why is buffering used in lexical analysis? What are the commonly used buffering methods?
4. Define transition diagram for an identifier?
5. Compare syntax tree and parse tree?
6. Summarize the issues in a lexical analyzer?
7. Define buffer pair?
8. State the interactions between the lexical analyzer and the parser?
9. Explain parse tree and construct a parse tree for –(id + id)?
10. Generalizes the advantage of having sentinels at the end of each buffer halves in buffer pairs?

Part-B

1. Describe the various phases of compiler with suitable example?
2. Give the structure of compiler Analysis the compiler with an assignment statement?
3. Discuss in detail about the role of Lexical Analyzer with the possible error recovery actions? Describe in detail about issues in lexical analysis?
4. Describe the Input buffering techniques in detail?
5. Discuss how a finite automaton is used to represent tokens and perform lexical analysis with examples?
6. Summarize in detail about how the tokens are specified by the compiler with suitable example?
7. Create DFA the following NFA
   M= ({q0, q1} ,{0,1},δ,q0,{q1})
   Where δ(q0,0) ={q0 ,q1}
   δ (q0,1) ={q1}
   δ(q1,0)=ϕ
   δ(q1,1)={q0,q1}
8. Show how the DFA is directly converted from an augmented regular expression?
9. Draw NFA for the regular expression ab*/ab ((ε / a) b *) *?
10. Compare NFA and DFA? And explain about the details?
11. Draw the DFA for the augmented regular expression (a|b) * # directly using syntax tree?
12. Define Lex and Lex specifications. How lexical Analyzer is constructed using lex? Give an example?

13. Explain an algorithm for Lex that recognizes the tokens? Describe the detail in tool for generating lexical Analyzer?
14. Analyzer the algorithm for minimizing the number of states of a DFA?
15. Generalize and give an example one regular expression if we were to revise the definition of a DFA to allow zero or one transition out of each state on each input symbol. Some regular expressions would then have smaller DFA's than they do under the standard definition of a DFA. Give and generalize an example of one such regular expression?

UNIT II SYNTAX ANALYSIS

2-Marks

1. Write the rule to eliminate left recursion in a grammar. Prepare and Eliminate the left recursion for the grammar.?
2. Define handle pruning?
3. Solve FIRST and FOLLOW by use the LL (1) grammar?
4. List the concepts of Predictive parsing and shift reduce parsing?
5. Differentiate TopDown parsing and BottomUp parsing?
6. Define Recursive Descent Parsing?
7. List out the properties of parse tree?
8. Solve the following grammar is ambiguous: S→aSbS / bSaS / €?
9. Difference between ambiguous and unambiguous grammar?
10. Define Phrase level error recovery?

Part -B

1. Explain left recursion and Left Factoring? Eliminate left recursion and left factoring for the following grammar E → E + T | E - T | T
   T → a | b | (E)
2. What is an ambiguous and un ambiguous grammar? Identify the following grammar is ambiguous or not.
   E→E+E | E*E | (E)|-E |id for the sentence id+id*id
3. Describe on detail about the role of parser? Discuss about the context free grammar?
4. Analyze Is it possible, by modifying the grammar in any way to construct predictive parser for the language of
   S→ SS + | SS * | a string "aa+ a *.
5. What are the different kinds of syntax error phased by a program? Explain in details?
6. What are the Error recovery techniques used in Predictive parsing? Explain in detail?
7. Define YAAC parser generator? And list out the error recovery action in YAAC?
8. List out different error recovery strategies. Explain them action in YAAC?

9. Construct a parse tree for the input string w-cad using topdown parser
   S->cAd
   A-> abla?
10. Define SLR (1) parser. Describe the Steps for the SLR parser?
11. Consider the following grammar S → AS|b
    A→SA|a
12. Show the actions of the parser for the input string "abab" Construct the SLR parse table for the grammar?
13. Give the LALR for the given grammar.
    E → E + T | T , T → T * F | F , F → (E) / id and parse the following? (a+b)* c
14. Explain in detail about the various types of Top –down parsing?
15. Analyze the LR parsing algorithm with an example?
16. What is CFG. Explain in detail about the Context-Free Grammar?

UNIT-III INTERMEDIATE CODE GENERATION

2-Marks

1. List out the two rules for type checking?
2. Compare synthesized attributes and inherited attributes?
3. What is Annotated parse tree?
4. Define Type checker.?
5. What is a syntax tree? Draw the syntax tree for the assignment statement a := b * -c + b * -c?
6. Define type systems?
7. Define Syntax directed definition of a simple desk calculator?
8. Summarize about the S-attributed definition?
9. State the type expressions?
10. Illustrate the methods of implementing three-address statements?

Part-B

1. Discuss the following in detail about the Syntax Directed Definitions? Inherited Atrributes and Synthesized attributes?
2. Evaluate the expressions for the SDD annotated parse tree for the following expressions (i)3 * 5 + 4n (ii)1* 2*3*(4+5)
3. Apply the S-attributed definition and constructs syntax trees for a simple expression grammar involving only the binary operators + and -. As usual, these operators are at the same precedence level and are jointly left associative. All nonterminal have one synthesized attribute node, which represents a node of the syntax tree Production:L->E$ E→E1+T,E →T, T->T1*F,T→ (E),T→ digit?
4. Discuss in detail about Dependency graph? Ordering Evaluation of Attributes?

5. Create variants of Syntax tree. Explain in detail about it with suitable example?
6. Analyse the common three address instruction forms?
7. Describe.in detail about Quadruples? Triples?
8. Describe in detail about addressing array Elements? Discuss in detail about Translation of array reference?
9. Describe in detail about types and declaration with suitable examples?
10. Compare three address code for expression with the Incremental translation?
11. State the rules for type checking with example? Give an algorithm for type inference and polymorphic function?
12. Illustrate an algorithm for unification with its operation?
13. Explain the steps for constructing a DAG. Construct the DAG for the following expression ((x+y)-((x+y)*(x-y)))+((x+y)*(x-y))?
14. What are the types of conversion? what are the rules of the conversion?
15. Suppose that we have a production A→BCD. Each of the four non terminal A, B, C and D have two attributes: S is a synthesized attribute and i is an inherited attribute. Analyze For each of the sets of rules below tell whether (i)the rules are consistent with an S attributed definition(ii) the rules are consistent with an L-attributed definition and(iii) whether the rules are consistent with any evaluation order at all? a) A.s = B.i + C.s

b) A.s = B.i + C.s and D.i = A.i + B.s.

UNIT IV- RUN-TIME ENVIRONMENT AND CODE GENERATION

2-Marks

1. List out limitations of the static memory allocation?
2. What is heap allocation?
3. How the activation record is pushed onto the stack?
4. State the principles for designing calling sequences?
5. List out the dynamic storage techniques?
6. Define variable data length on the stack?
7. Differentiate between stack and Heap allocation?
8. Distinguish between static and dynamic storage allocation?
9. Discuss the main idea of Activation tree?
10. How would you solve the issues in the design of code generators?

Part-B

1. Illustrate the storage organization memory in the perspective of compiler writer with neat diagram? Compare static versus dynamic memory allocation?
2. Explain in detail about the various issues in code generation

with examples?

3. Develop a quicksort algorithm for reads nine integers into an array a and sorts them by using the concepts of activation tree? Give the structure of the action record?
4. Analyze the data access without nested procedure and the issues with nested procedure? Give the version of quicksort in ML style using nested procedure?
5. Discuss in detail about heap manager? Describe in detail about the memory hierarchy of a computer?
6. Define fragmentation? Describe in detail about how to reduce the fragment?
7. Write short notes on the following Best fit and next object placement? Managing and coalescing free space?
8. Examine the problems with manual deallocation of memory and explain how the conventional tools are used to cope with the complexity in managing memory?
9. Explain in detail about instruction selection and register allocation of code generation?
10. Illustrate in detail about the code generation algorithm with an example?
11. Describe the usage of stack in the memory allocation and discuss in detail about stack allocation space of memory?
12. Define the heap management of memory and describe in detail about it?
13. Compare the stack and heap allocation memory in detail with suitable example?
14. Generate code for the following sequence assuming that n is in a memory location s=0
15. Create following assignment statement into three address code D:=(a-b)*(a-c)+(a-c) Apply code generation algorithm to generate a code sequence for the three address statement?

UNIT V- CODE OPTIMIZATION

2-Marks

1. Illustrate the concepts of copy propagation?
2. State the use of machine Idioms?
3. Show the flow graph for the quicksort algorithm?
4. Apply the basic block concepts, how would you representing the dummy blocks with no statements indicated in global dataflow analysis?
5. Identify the constructs for optimization in basic block?
6. Define the term data flow analysis?
7. How is liveness of a variable calculated? Identify it.?
8. Give the uses of gen and Kill functions?
9. Discuss the concepts of basic blocks and flow graphs?
10. Give the main idea of dead code elimination and constant folding?

Part-B

1. Explain in detail about optimization of basic blocks?
2. Explain in detail about the principle source of optimization?
3. Discuss the following in detail Semantic preserving transformation? Global Common subexpression?
4. Write about the following in detail copy propagation? Dead code Elimination? code motion?
5. Explain in detail about the data-flow schemas on basic block and the transfer equations for reaching definitions with example?
6. Illustrate the Iterative algorithm for reaching definitions? Discuss the live variable analysis?
7. Analyze Peephole optimization with suitable examples?
8. Demonstrate optimization of Basic Blocks with an example?
9. Discuss in detail about how to find Local Common Sub-expression? (ii)Discuss in detail about the Use of Algebraic Identities?
10. Describe in detail about the flow of control optimization? Identify the methods to eliminate the unreachable code, load and store data?
11. Give an example to identify the dead code in the DAG? Describe the representation of array using DAG with example?
12. Summarize in detail about the dataflow analysis of available expression with suitable example?
13. Formulate steps to identify the loops in the basic block? Describe about induction variable and end reduction in strength?
14. Describe the efficient data flow algorithms in detail?
15. Compute and Kill sets for each Block, In and Out sets for each block, Compute and from the given diagram.?