Turbomachinery Institute of Technology and Sciences, Hyderabad-319

(Approved by AICTE. & Govt. of Andhra Pradesh, Affiliated to JNTU., Hyderabad)

QUESTION BANK

Year: III-II Branch: CSE Subject: CD Name of the Faculty: Mrs.Vidha Sharma

COMPILER DESIGN
UNIT – 1

SYLLABUS:

Overview of Compilation: Phases of Compilation – Lexical Analysis, Regular Grammar and regular expression for common programming language features, pass and Phases of translation, interpretation, bootstrapping, data structures in compilation – LEX lexical analyzer generator.

Questions:

1.Design a DFA that accepts the language over the alphabet, _ = {0, 1, 2} where the decimal equivalent of the language is divisible by 3. [Sup.Feb 2008, Set1]
2.Compare compiler and an interpreter with the help of suitable examples. [Sup.Feb 2008, Set1]
3. Obtain the Kleen Closure and Positive Closure of the language {ba, bb}, where the alphabet _ = {a, b}.[Sup.Feb 2008, Set2]
4. Give a finite state diagram that accepts all the floating-point numbers. [Sup.Feb 2008, Set2]
5. Explain Regular Expressions with suitable examples. [Sup.Feb 2008, Set3]
6. Design a DFA that accepts the language over _ = {a, b} of all strings that contain the sub-string either aa or bb. [Sup.Feb 2008, Set4]
7. Explain the input buffer scheme for scanning the source program. How the use of sentinels can improve its performance? Describe in detail.[Apr/May 2008,Set1]
8. Explain the different phases of a compiler, showing the output of each phase, using the example of the following statement:
position : = initial + rate * 60[Apr/May 2008,Set2]
9. Compare compiler and interpreter with suitable diagrams. [Apr/May 2008,Set2]
10. Explain, in detail, lexical analyzer generator. [Apr/May 2008,Set3]
11. Describe the lexical errors and various error recovery strategies with suitable examples. [Apr/May 2008,Set3]
12. Consider the following fragment of ‘C’ code:
float i, j;
i = i * 70 + j + 2;
Write the output at all phases of the compiler for the above ‘C’ code. [Apr/May 2008,Set4]
13. Write short notes on: input buffering. [Apr/May 2008,Set4]
14. Write a Lex Specification to read a C program and calculate number of new line characters, tabs and white spaces in the program.
15. Whether lexical analysis detects any errors? Explain with example. [Feb 2003]
16. Explain with example
various Compiler Construction tools. [9]
17.Why compilation phases are divided into front-end and back-end? What are the advantages?
18 Explain the following :
i) token.
ii)pattern.
iii) lexeme.

UNIT-2

SYLLABUS:

Top down Parsing : Context free grammars, Top down parsing – Backtracking, LL (1), recursive descent parsing, Predictive parsing, Preprocessing steps required for predictive parsing.

Questions:

1.Test whether the following grammar is LL(1) or not.
S ! AaAb |BbBa
A ! 2
B !2 [Sup.Feb 2008, Set1]
2. Construct the predictive parse table for the following grammar:
S ! A
A ! aB|Ad
B ! bBC|f
C ! g. [Sup.Feb 2008, Set1]
3. What is the time complexity of a parser to parse a string of ‘n’ tokens? [Sup.Feb 2008, Set2]
4. Consider the Grammar: G = ({S, A}, {a, b}, {S ! aAa |bAb| |A, A ! SS}, S)
Find the leftmost derivation, rightmost derivation, and parse tree for the string: baabbb. [Sup.Feb 2008, Set2]
5. Write a procedure to combine two NFA?s into a single NFA. The operations to be performed are those of concatenation, union and closure. [Sup.Feb 2008, Set3]
6. Obtain the Non-deterministic Finite Automaton (NFA) corresponds to the Grammar,
G = ({S, X, Y}, {a, b}, P, S), where P is defined as follows:
P ! aS |bS| bX
X ! bY |b
Y ! aY |bY| a |b. [Sup.Feb 2008, Set3]
7. Write a Context Free Grammar(CFG) for the while statement in ‘C’ language. [Sup.Feb 2008, Set4]
8. Construct predictive parsing table for the following grammar.
E ! T E′
E′ ! +T E′|ε
T ! F T′
T′ ! ∗FT′|ε
F ! (E)|id[Apr/May 2008,Set1]
9. What is recursive descent parser? Construct recursive descent parser for the following grammar.
E ! E + T|T
T ! TF|F
F ! F_|a|b[Apr/May 2008,Set2]
10. What is ambiguous grammar? Eliminate ambiguities for the grammar:
E ! E + E|E_E|(E)|id. [Apr/May 2008,Set2]
11. Consider the following grammar.
S ! 0A|1B|0| 1
A ! 0S|1B| 1
B ! 0A|1 S
Construct leftmost derivations and parse trees for the following sentences
i. 0101
ii. 1100101[Apr/May 2008,Set3]
12. Consider the following grammar
E ! T + E|T
T ! V_T|V
V ! id
Write down the procedures for the nonterminals of the grammar to make a recursive descent parser. [Apr/May 2008,Set3]
13. Show that the following grammar is LR(1) but not LALR(1). [Feb 2003]
S -> Aa | bAc | Bc | bBa
A -> d
B -> d
14. Exlain Recursive Descent parser with an example.
15. Show that the following grammar is LL(1) but not SLR(1). [Feb 2003]
S -> AaAb | BbBa
A -> e
B -> e
16. What is Shift-Reduce and Reduce-Reduce conflict? How these can be resolved? With examples explain in which condition S-R and R-R conflict can occur in SLR, canonical LR and LALR parsers. (Make use of LR(0), LR(1) items).

UNIT-3

SYLLABUS:

Bottom up parsing : Shift Reduce parsing, LR and LALR parsing, Error recovery in parsing , handling ambiguous grammar, YACC – automatic parser generator.

Questions:

1.What is LR parser? Compare and contrast the different types of LR parsers. [Sup.Feb 2008, Set1]
2. Construct the CLR parse table for the following augmented grammar:
A′ ! A
A ! (A) |a[Sup.Feb 2008, Set1]
3. Construct the collection of non-empty sets of LR(0) items for the following augmented grammar:
S ! E1
E1 ! T3E1 |T1
E2 ! T3E2 |T2
T1 ! a$ |(E2$
T2 ! a) |(E2)
T3 ! a+|(E2+.   [Sup.Feb 2008, Set2]
4. What is meant by a parser generator? Illustrate with examples using YACC. [Sup.Feb 2008, Set3]
5. How are ambiguities resolved in YACC? [Sup.Feb 2008, Set4]
6. What is an operator grammar? Give an example. [Apr/May 2008,Set1]
7. Write an operator precedence parsing algorithm. [Apr/May 2008,Set1]
8. Construct SLR parsing table for the following grammar.
S ! AS|b
A ! SA|a[Apr/May 2008,Set2]
9. Define LR(k) parser. Draw and explain model of LR parser. [Apr/May 2008,Set3]
10. Write LR parsing algorithm. [Apr/May 2008,Set3]
11. Show that the following grammar is LL(1) but not SLR(1). [Feb 2003]
S -> AaAb | BbBa
A -> e
B -> e
12. What is Shift-Reduce and Reduce-Reduce conflict? How these can be resolved? With examples explain in which condition S-R and R-R conflict can occur in SLR, canonical LR and LALR parsers. (Make use of LR(0), LR(1) items).[Feb 2003]
13. Write a translation scheme to generate three address code for assignment sentences with array and pointer references. [Feb 2003]
14. Explain concept of back-patching with example.
15. Translate executable sentences of the following C program. [Feb 2003]

```
Main()
{
Int i = 1;
Int a[10];
While (i <= 10)
{
a[i] = 0;
i = i + 1;
}
}
```

into
a) Syntax tree
b) Postfix notation
c) Three-address code.
b) What are synthesized and inherited attributes? What are Marker Non-terminal symbols?

Give example.

UNIT-4

SYLLABUS:

Semantic analysis : Intermediate forms of source Programs – abstract syntax tree, polish notation and three address codes. Attributed grammars, Syntax directed translation, Conversion of popular Programming languages language Constructs into Intermediate code forms, Type checker.

Questions:

1.Compare Inherited attributes and Synthesized attributes with an example. [Sup.Feb 2008, Set1]
2. Construct triples of an expression: a * - (b + c). [Sup.Feb 2008, Set1]
3. Let synthesized attribute, Val give the value of the binary number generated by S
in the following grammar. For example, on input 101.101, S.Val = 5.625.
S ! L • L |L
L ! LB|B
B ! 0 |1
Write synthesized attribute values corresponding to each of the productions to
determine the S.Val. [Sup.Feb 2008, Set2]
4. Generate the three-address code for the following ?C? program fragment: [16]
while(a > b)
{
if (c < d) x = y + z;
else x = y - z;
}[Sup.Feb 2008, Set3]
5. What are L-attributed definitions? Explain with an example. [Sup.Feb 2008, Set4]
6. Draw the syntax tree for the following Boolean expression:
(P < Q AND R < S) OR (T < U AND R <Q). [Sup.Feb 2008, Set4]
7. Write a note on the specification of a simple type checker. [Apr/May 2008,Set1]
8. What is a type expression? Explain the equivalence of type expressions with
an appropriate examples. [Apr/May 2008,Set1]
9. Write the quadruple, triple, indirect triple for the statement a := b_ − c + b_ − c. [Apr/May 2008,Set2]
10. Explain the role of intermediate code generator in compilation process. [Apr/May 2008,Set2]
11. Write short notes on the following:
(a) S-attributed definitions.
(b) L-attributed definitions.
(c) Dependency graph. [Apr/May 2008,Set4]

UNIT-5

SYLLABUS:

Symbol Tables : Symbol table format, organization for block structures languages, hashing, tree structures representation of scope information. Block structures and non block structure storage allocation: static, Runtime stack and heap storage allocation, storage allocation for arrays, strings and records.

Questions:

1.List out various typical semantic errors .Explain the procedure to rectify them? [Sup.Feb 2008, Set1]
2. What is Static Checking? List out some examples of static checks? [Sup.Feb 2008, Set1]
3. Explain the following:
(a) Type checking of Expressions
(b) Translation scheme for checking the type of statements. [Sup.Feb 2008, Set2]
4. What is Type Expression? Write type Expressions for the following type
i. A Two dimensional array integers (i.e. an array of arrays) whose rows are
indexed from 0 to 9 and whose columns are indexed from -10 to 10. [Sup.Feb 2008, Set2]
5. What is Type System? Discuss static and dynamic Checking of types? [Sup.Feb 2008, Set3]
6.Distinguish static and dynamic Type checking ? [Sup.Feb 2008, Set4]
7. Discuss in detail about semantic analysis phase? [Sup.Feb 2008, Set4]
8. Compare three different storage allocation strategies. [Apr/May 2008,Set1]
9. Consider the following array declaration in ‘c’;

float a[100][100];
Assume that the main memory in byte addressable and that the array is stored starting from the memory address 100. What is the address of a[40][50]? [Apr/May 2008,Set1]
10. Write an algorithm to perform the table lookup and insertion operation for hashed symbol table. [Apr/May 2008,Set2]
11. What is an ordered and unordered symbol table? What is the function of symbol table in the compliation process? Explain. [Apr/May 2008,Set3]
12. What are the various attributes of a Symbol Table? [Apr/May 2008,Set3]

UNIT-6

SYLLABUS:

Code optimization : Consideration for Optimization, Scope of Optimization, local optimization, loop optimization, frequency reduction, folding, DAG representation.

Questions:

1.Write a notes on the static storage allocation strategy with example and discuss its limitations? [Sup.Feb 2008, Set1]
2. Discuss about the stack allocation strategy of runtime environment with an example? [Sup.Feb 2008, Set1]
3. Explain the concept of implicit deallocation of memory. [Sup.Feb 2008, Set2]
4. Give an example of creating dangling references and explain how garbage is created[Sup.Feb 2008, Set2]
5. Write a notes on the static storage allocation strategy with example and discuss its limitations? [Sup.Feb 2008, Set3]
6. Explain how scope information is represented in the symbol table for block structured language? [Sup.Feb 2008, Set4]
7. Write and explain about activation record? [Sup.Feb 2008, Set4]
8. Explain different principal sources of optimization technique with suitable examples. [Apr/May 2008,Set1]
10. What is code optimization? What are its advantages? [Apr/May 2008,Set2]
11. Explain briefly about folding. [Apr/May 2008,Set2]
12. What are the problems in optimizing compiler design? [Apr/May 2008,Set2]

UNIT-7

SYLLABUS:

Data flow analysis : Flow graph, data flow equation, global optimization, redundant sub expression elimination, Induction variable elements, Live variable analysis, Copy propagation.

Questions:

1.Write about the following Algorithms
(a) Detection of Loop Invariant Computation
(b) Code Motion. [Sup.Feb 2008, Set1]
2. Explain about Data-Flow analysis of structured flow graphs[Sup.Feb 2008, Set2].
3. Explain the following
(a) Copy Propagation
(b) Dead-Code Elimination[Sup.Feb 2008, Set3].
4. Explain in detail the procedure that eliminating global common sub expression? [Sup.Feb 2008, Set4].
5. Write and explain live variable analysis algorithm. [Apr/May 2008,Set1]
6. Explain the use of algebraic transformations with an example[Apr/May 2008,Set1]
7. Explain reducible and non-reducible flow graphs with an example. [Apr/May 2008,Set2]
8. Explain natural loops and inner loops of a flow graph with an example. [Apr/May 2008,Set2]

9. Explain about data flow analysis of structured programs. [Apr/May 2008,Set3]
10.Explain the following:
(c) Code Motion
(d) Reduction in Strength. [Sup.Feb 2008, Set3].
11. Explain Runtime support and Storage organization.[Feb 2003]

12 With example explain what is Global Common Sub-expression? Write an algorithm for
Global Common Sub-expression Elimination.[Feb 2003]

UNIT-8

SYLLABUS:

Object code generation : Object code forms, machine dependent code optimization, register allocation and assignment generic code generation algorithms, DAG for register allocation.

Questions:

1.Explain about Generic code generation algorithm? [Sup.Feb 2008, Set1]
2. What are legal evolution orders and names for the values at the nodes for the DAG
for following?
d := b + c
e := a + b
b := b * c
a := e - d. [Sup.Feb 2008, Set2]
3. Construct DAG for the following basic block:
d: = b+c
e: = a+b
b: =b*c
a: = e-d. [Sup.Feb 2008, Set3].
4. Write and explain about object code forms? [Sup.Feb 2008, Set4].
5. Explain the different issues in the design of a code generator. [Apr/May 2008,Set1]
6. Generate code for the following C statements:
i. x= f(a) + f(a) + f(a)
ii. x= f(a) /g(b,c)
iii. x= f(f(a))
iv. x= ++f(a) [Apr/May 2008,Set1]
7. Explain the concept of object code forms. [Apr/May 2008,Set2]
8. Generate optimal machine code for the following C program.

```
main()
{
int i, a[10];
while (i<=10) a[i] =0
}           [Apr/May 2008,Set2]
```

9. What is a basic block and flow graph? Generate three address code for the following
program. Find the basic blocks in it and write flow graph for the same.

```
begin
prod := 0;
i := 1;
do
begin
prod := prod + a[i] * b[i];
i := i + 1;
end
while i <= 20
end        [Feb2003]
```

10. What is a DAG? Explain role of DAG in code generation phase. [Feb 2008]
11. Enlist and explain with example various transformations on basic blocks. [Feb 2003]
12. Explain peephole optimization in detail. [Feb 2003]