Marks Scored:

KATHMANDU UNIVERSITY
End Semester Examination
March/ April, 2017

Level : B. E.
Year : IV

Course : COMP 409
Semester : I

Marks Scored:

KATHMANDU UNIVERSITY
End Semester Examination
March/ April, 2017

Level : B. E. | Course : COMP 409
Year : IV | Semester : I
Exam Roll No. : | Time: 30 mins. | F. M. : 10
Registration No.: | Date APR 19 2017

SECTION "A"
[20 Q × 0.5 = 10 marks]

Encircle the most appropriate answer.

1. If a NFA has n states then what is the maximum number of states the equivalent DFA ?
   a. n^2 b. n c. 2^n d. n/2

2. A compiler running on computers with small memory would normally be:
   a. A multi-pass compiler b. Single pass compiler
   c. A compiler with less number of phases d. None of the above

3. Which of the following conflicts cannot arise in LR parsing
   a. Shift-reduce b. Reduce-reduce c. Shift-shift d. Reduce only

4. If a grammar is LALR(1) then it is necessarily
   a. SLR b. CLR c. LL(1) d. Both (a) and (c)

5. An annotated parse tree is:
   a. A parse tree with attribute values shown at the parse tree nodes
   b. A parse tree with values of only some attributes shown at parse tree nodes
   c. A parse tree without attribute values shown at parse tree node
   d. None of the above

6. A synthesized attribute is an attribute whose value at a parse tree node depends on
   a. Attributes at the siblings only b. Attributes at parent node only
   c. Attributes at children nodes only d. Attributes both at parent and children node

7. Consider the statement i = 1: , here the colon is used in place of semicolon. This error is detected by the computer in:
   a. Lexical analysis phase b. Syntax analysis phase
   c. Code optimization phase d. Code generation phase

8. Constant folding means:
   a. Replacing expressions by their values, if the value can be computed at compile time
   b. Replacing an operand by constant
   c. Ignoring the constant
   d. None of the above

9. What is true about ε-closure(q), where q is a state of a finite automata?
   a. It can be empty b. It contains at least q
   c. It is an infinite set d. It contains at most q

10. If G = (V,T,P,S) is a context free grammar, then L(G) will be infinite if and only if:
    a. At least one production in P is recursive
    b. No production in P is recursive
    c. All productions in P are recursive
    d. None of the above

11. A left recursive grammar:
    a. Cannot be LL(1)
    b. Cannot be LR(1)
    c. Is an ambiguous grammar
    d. Will parse with top down parser

12. Which one of the following data structure is used to implement three address code?
    a. Tree
    b. Stack
    c. Quadruple
    d. Queue

Consider the following grammar G:

S'→ S$
S→qABC
A →a | bbD
B → a | ε
C → b | ε
D → C | ε

Question (13-16) depends on the given grammar G.

13. FOLLOW(A) is:
    a. {b, $}
    b. {a,$}
    c. {a,b,$}
    d. {a,b}

14. FOLLOW(B) cannot be
    a. FIRST(C)
    b. FIRST(A)
    c. FIRST(S)
    d. {$}

15. Which of the following is true
    a. FOLLOW(B) = FOLLOW(A)
    b. FOLLOW(B) = FOLLOW(D)
    c. FOLLOW(D) = FOLLOW(A)
    d. FOLLOW(D) = FOLLOW(C)

16. FIRST(D) is :
    a. {a,b}
    b. {b,$}
    c. {a,ε,$}
    d. {b,ε}

Consider the following grammar G':

S→ FR
R →*S | ε
F → id

Question (17-20) depends on given grammar G'

17. Which of the following string is generated by the given grammar?
    a. id*id*id
    b. id*id*
    c. *id
    d. **id

18. For the correct answer in 17, how many steps are required to derive the string from S and how many parse trees are there?
    a. 6 and 1
    b. 6 and 2
    c. 7 and 1
    d. 9 and 1

19. The number of states in SLR automation for G' is:
    a. 5
    b. 6
    c. 7
    d. 8

20. In the predictive parsing table, M, of the grammar, the entries M[S,id] and M[R,$] respectively are:
    a. {S→FR} and {R →□}
    b. {S→ FR} and error
    c. {S → FR} and {R → *S}
    d. {F → id} and {R→ϵ}

KATHMANDU UNIVERSITY
End Semester Examination
March/April, 2017

APR 19 2017

Level	: B.E.	Course	: COMP 409
Year	: IV	Semester	: I
Time	: 2 hrs. 30 mins	F. M.	: 40

SECTION "B"
[6Q. × 4 = 24 marks]

Attempt *ANY SIX* questions.

1. Discuss the advantage of single and multipass compiler. Also explain the action taken by every phase of compiler on the following sentence
if (Expr) then Stmt1 else Stmt2

2. Give the definition of DFA, NFA and ε-NFA. Convert the following NFA into its equivalent DFA.

3. Answer the following questions:
   a. Design a DFA for the following language over $\Sigma = \{a,b\}$
      i. A DFA that accepts only those words that have more than four letters.
      ii. A DFA that accepts only those words that begin or end with a double letters
   b. Construct a regular expression defining each of the following language over the alphabet $\Sigma = \{a,b\}$
      i. All strings in which letter b never triples
      ii. All strings that have an odd number of a's and an odd number of b's

4. Explain the reason for augmenting a regular expression. Convert the following regular expression into its equivalent DFA.
   (aa|bb) (ab|ba)*ab(a|b)

5. Describe the role of lexical analyzer. Identify the lexemes that make up the tokens in the following program segment.

```
void swap(i,j:int);
int temp;
{
        temp p = i;
        i=j;
        j= temp;
}
```

6. Explain the following optimization technique with suitable example.
   a. Common Sub Expression Elimination
   b. Dead Code Elimination
   c. Loop-invariant Code Motion
   d. Loop unrolling

7. Differentiate between s-attributed and l-attributed grammar. Also explain the role of dependency graph in semantic analysis with suitable example.

SECTION "C"

[2 Q × 8 = 16 marks]

Attempt *ANY TWO* questions.

8. Design LL(1) parsing table for the following grammar. [5+3=8]

A →AcB | cC | C
B →bB | id
C →CaB |BbB | B

Also using parsing program test whether the string **w= cid**is accepted or not.

9. Explain the weakness of Simple LR parser. Construct canonical LR(1) parsing table for the following grammar. [2+6=8]

E → T | E – T
T → F | *F
F → id | (E)

10. Consider the following grammar and give the syntax directed definition to construct parse tree. For the input expression 4*7 + 1 * 2, construct an annotated parse tree and dependency graph according to your syntax directed definition [8]

E → TE'
E' → +TE' | ε
T → FT'
T' → *FT' | ε
F → digit