Mark Scored:

KATHMANDU UNIVERSITY
End Semester Examination
August/September, 2017

Level : B. Sc.
Year : III
Course : COMP 409
Semester : I

Exam Roll No. :
Time: 30 min
F. M. : 10

Registration No.:
Date :

SECTION "A"
[20 Q × 0.5 = 10 marks]

Encircle the alphabetical letter of the most appropriate answer.

1. In a compiler, keyword of a language are recognized during
[A] Parsing of a program
[B] The code generation phase
[C] Lexical analysis of the program
[D] Data follow analysis

2. The best data structure to check whether an arithmetic expression has balanced parenthesis is:
[A] List [B] Queue [C] Stack [D] Tree

3. Assume that the operators +, - , x are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, x, +, - . The postfix expression corresponding to the infix expression a + b x c – d ^ e ^ f is
[A] - + a x b c ^ ^ d e f
[B] a b c x + d e ^ f ^ -
[C]a b + c x d – e ^ f ^
[D] a b c x + d e f ^ ^ -

4. The analysis phase of compilation is to:
[A] Break up source program into pieces and impose a grammatical structures
[B] Creates IR of source program
[C] Determines the operations and records them in a structure
[D] All of the above

5. Which of the following is not performed by a preprocessor?
[A] File Inclusion
[B] Macro processing
[C] Error Handling
[D] Language extension

6. Which phase of compilation can find out that the upcoming sequence of numeric characters represents no number in the source language?
[A] Code optimization
[B] Lexical Analysis
[C] Semantic Analysis
[D] Syntax Analysis

7. For a NFA of n states, corresponding DFA can have
[A] n^2 states [B] 2^n states [C] 2^{n-1} states [D] n/n^2 states

8. For the given ε-NFA, the equivalent DFA will have which of the following optimal states?

δ	ε	0	1
→q0	Φ	q1	Φ
q1	q0	Φ	q2
*q2	Φ	Φ	q0

[A] [q0], [q0,q1], [q2]
[B] [q0,q1], [q1], [q1,q2]
[C] [q0], [q1], [q1,q2]
[D] [q0], [q1], [q2]

9. A set of strings in the input for which the same token is produced as output is called?
[A] token [B] pattern [C] lexeme [D] parse tree

10. Which of the following is not a role of a parser?
[A] It obtains a string of tokens for a lexical analyzer
[B] It should report any syntax error in the program
[C] It should keep tokens in the symbol table
[D] It should recover from the errors so that it can continue to process the rest of the input

11. Drawback of recursive descent parser is:
[A] Suffers with left recursion
[B] Parse with top down method
[C] Recursion is not an issue
[D] Each non terminal is implemented as a procedure of recursion

12. If n is a cat-node with left child C1 and right child C2 and i is a position in lastpos(C1) then,
[A] all position in lastpos(C2) are in followpos(i)
[B] all position in firstpos(C1) are in followpos(i)
[C] all position in firstpos(C2) are in followpos(i)
[D] all position in lastpos(C1) are in followpos(i)

Consider the following grammar
S→aBDh
B→cC
C→bC | ε
D→ EF
E→ g | ε
F→f | ε
Question (13 to 14) are based on the given Grammar:

13. FOLLOW(B) is:
[A] {g,f,h} [B] {g,h} [C] {f,h} [D] {a,g,h}

14. FOLLOW(F) is equivalent to:
[A] FOLLOW(B) [B] FOLLOW(C) [C] FOLLOW(E) [D] FOLLOW(D)

15. Which of the following is not true for SLR?
[A] Parser is largest in size
[B] It requires less time and space
[C] It exposes less syntactic features
[D] Immediate error detection is not done

16. Consider the grammar G: { S →iEtS | iEtSeS | a ; E→b }
The Left factored grammar is:
[A]{ S→iEtSS' | a ; S' →eS | ε ; E → b} [B] { S→iEtS' | a ; S' →SeS | ε ; E → b}
[C]{ S→iS' | a ; S' →EtSeS | ε ; E →b} [D] { S →iES' | a ; S' →tSeS | ε ; E→ b}

17. For the grammar G: { E → E + T ; E →T; T→ T*F; T→F; F→ id} the number of states for SLR automation is
[A] 7 [B] 8 [C] 9 [D] 10

18. For grammar in Q17, Goto [I0, +] is :

[A] {T→T*.F; F→id}
[B] {E→ E+.T; T→.T*F; T→.F; F→.id}
[C] {T→T*F.}
[D] {E→ E.+T; E→ .T}

19. if L<R gotoYesTarget
gotoNoTarget
Which of the following can use the given code?

[A] While condition
[B] Procedure call
[C] Function call
[D] Pointer assignment

20. How many states are required to validate keyword *printf*

[A] 6
[B] 7
[C] 8
[D] 5

KATHMANDU UNIVERSITY
End Semester Examination
August/September, 2017

Level : B. Sc.
Year : III
Time : 2 hrs. 30 mins.

Course : COMP 409
Semester : I
F. M. : 40

SECTION "B"
[6Q. ×4 = 24 marks]

Attempt *ANY SIX* questions.

1. Explain with specific example how tokens are specified. Also explain with subroutine how recognition of token is carried out during lexical analysis phase.

2. Construct DFA from the given regular expression over the alphabet {a.b}
R.E :**(aa|bb) (a|b)*aba (a|bb)**

3. Consider the following piece of code:

```
c = 10;
a = c;
i = j = 0;
while(i != (a*3))
{
        if (j) print("Hello");
        i++;
        d[i] = 10*i;
        b = 5 * i;
}
```

Identify different kinds of optimizations possible. Rewrite the code after making optimized.

4. Write a C-program that identifies the greatest number from n natural numbers. Convert your code into three address code.

5. Explain different error handling technique used in analysis phase of compilation.

6. Explain the significance of Intermediate Representation in compilation process.

7. Write Short notes on with example
   a. Local and Loop optimization
   b. Induction variable elimination
   c. Dead code elimination
   d. Constant folding and combining

SECTION "C"

[2 Q × 8 = 16 marks]

Attempt *ANY TWO* questions.

8. Consider the following grammar:

   S → aAC | bB
   A → Abc | Abd | e
   B → f | g
   C → h | i

   a. Design LL(1) parsing table for the grammar.
   b. Check whether the string w = **aebch** can be parsed or not.

9. Construct the following grammar:

   E → E+T | T
   T → TF | F
   F → F* | a | b

   a. Construct LR(0) item sets
   b. Construct LR(0) parsing table
   c. Check whether the string w = **a+a+ba*** can be parsed or not.

10. Consider the following grammar and give the syntax directed definition to construct parse tree. For the input expression **5*3+4*9**, construct an annotated parse tree along with dependency graph according to your syntax directed definition. (4+4)

    E → TE′
    E′ → +TE′
    E′ → ε
    T → FT′
    T′ → *FT′
    T′ → ε
    F → digit

Marks Scored:

KATHMANDU UNIVERSITY
End Semester Examination
March/ April, 2017

Level : B. E.
Year : IV
Course : COMP 409
Semester : I
Exam Roll No. :
Time: 30 mins.
F. M. : 10
Registration No.:
Date APR 1 3 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'→ SS
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.
Year : IV
Time : 2 hrs. 30 mins

Course : COMP 409
Semester : I
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= cidis** 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

KATHMANDU UNIVERSITY
End Semester Examination
February/March, 2016

Marks Scored :

Level : B.E. / B.Sc.
Year : IV

06 MAR 2016

Course : COMP 409
Semester : I

Exam Roll No. :

Time: 30 mins.

F.M. : 10

Registration No.:

Date :

SECTION "A"
[20Q. × 0.5 = 10 marks]

Encircle the alphabet of correct answer.

1. In a compiler, keyword of a language are recognized during
   a. Parsing of a program
   b. The code generation phase
   c. Data follow analysis
   d. Lexical analysis of the program

2. The best data structure to check whether an arithmetic expression has balanced parenthesis is:
   a. Queue b. Tree c. Stack d. List

3. Assume that the operators +, - , x are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, x, +, - . What is the postfix expression corresponding to the infix expression a + b x c – d ^ e ^ f?
   a. a b c x + d e f ^ ^ -
   b. a b + c x d – e ^ f ^
   c. a b c x + d e ^ f ^ -
   d. - + a x b c ^ ^ d e f

4. The regular expression 0*(10*)* denotes the same set as:
   a. (0+1)*10(0+1)* b. (1*0)*1* c. 0+(0+10)* d. (10*+01*)

5. Which data structure in compiler is used for managing information about variables and their attributes?
   a. Abstract Systax Tree
   b. Semantic Stack
   c. Symbol Table
   d. Parse Table

6. If a computation produces the same value in every loop iteration, move it out of the loop. This method is called
   a. Loop unrolling
   b. Loop fusion
   c. Loop invariant code motion
   d. Constant folding and combining

7. Consider the grammar: E→E + n | E x n | n. For a sentence n + n x n, the handles in the right sentential form of the reduction are:
   a. n, E + n and E + n x n
   b. n, n + n and n + n x n
   c. n, E + n and E + E + n
   d. n, E + n and E x n

8. If L<R goto Yes Target
   gotoNoTarget
   Which of the following can use the given code?
   a. While condition b. Function call c. Procedure call d. Pointer Assignment

Consider the following Context Free grammar G:

G: { S→ i S e S
S → i S
S→a }

Following four questions (9-12) are based upon the given grammar G.

9. Which one of the following string is generated by the above grammar?
a. iieae b. iiaea c. iiaae d. iieea

10. For the correct answer in 9, how many steps are required to derive the string from S and how many parse trees are there?
a. 5 and 2 b. 4 and 1 c. 4 and 2 d. 6 and 2

11. In the predictive parsing table, M, of the grammar, the entries M[S, i] and M[S, $] respectively are:
a. {S →iSeS} and {S → a} b. {S →iSeS ; S → iS} and Error()
c. {S → iS} and {S → a} d. {S → iSeS ; S→a; S→iS} and Error()

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

13. Which of the following common programming errors occurs during syntactic level?
a. Operator applied to incompatible operand
b. Misspelling an identifier or keyword or operator
c. Arithmetic expression with unbalanced parenthesis
d. None of the above

14. An error recovery technique in which the system discards the input until a valid token is found is called?
a. Global Correction b. Error production
c. Phrase Level recovery d. Panic Mode recovery

15. Consider the grammar G: { S →iEtS | iEtSeS | a ; E→b }
The Left factored grammar is:
a. { S→iEtSS' | a ; S' →eS | ϵ ; E → b} b. { S→iEtS' | a ; S' →SeS | ϵ ; E → b}
c. { S →iES' | a ; S' →tSeS | ϵ ; E→ b} d. { S →iS' | a ; S' →EtSeS | ϵ ; E →b}

Consider the following grammar G:

G: { E→TE'
E'→+E | ε
T→FT'
T'→T | ε
F →PF'
F'→*F' | ε
P → (E) | a | b| ε }

Following 5 questions (16-20) are based upon the given grammar G:

16. FIRST(T') is
a. {a,b} b. {+, ϵ} c. {(, a, b,ϵ} d. {(, a, b, ϵ}

17. FIRST(E') is equal to:
a. FIRST(E) b. FIRST(F) c. FIRST(F') d. None of the above

18. FOLLOW(F) is:
a. {$, a, b, (,)} b. {$, +, *, a, b, (} c. ($, +, a, b, (,)} d. ($, +, *, a, b,)}

19. FOLLOW(P) is :
a. {+, a, b, (,)} b. {$, +, *, a, b, (,)} c. ($, +, *, a, b,)} d. ($, +, a, b, (,)}

20. Which of the following answer is incorrect?
a. FOLLOW(P) = FOLLOW(F)
b. FIRST(T) = FIRST(P)
c. FOLLOW sets does not contain ε
d. FOLLOW(E) = FOLLOW(E')

KATHMANDU UNIVERSITY
End Semester Examination
February/March, 2016

Level : B.E./B. Sc.
Year : IV
Time : 2 hrs. 30 mins.

06 MAR 2016

Course : COMP 409
Semester: I
F. M. : 40

SECTION "B"

[6Q. × 4 = 24 marks]

Attempt *ANY SIX* questions of the following.

1. Explain with specific example how tokens are specified. Also explain with subroutine how recognition of token is carried out during lexical analysis phase. [2+2]

2. Write the step involved to convert NFA into its equivalent DFA. Convert the given NFA into its equivalent DFA. [2+2]

δ	ε	0	1
→Q0	Q1	Φ	Φ
Q1	Φ	Q2	Q3
Q2	Φ	Q1	{Q1,Q4}
Q3	Q2	Q4	Φ
*Q4	Φ	Φ	Φ

3. Write the steps involved in shift-reduce parsing. Design Shift-reduce parser for the following grammar G using the input string w = **10201** [2+2]

G: S →0S0 | 1S1 | 2

4. Write a C-program function using recursion that computes factorial of n. Convert your code into three address code. [2 + 2]

5. Write a regular expression over an alphabet {a,b} such that it generate a string in which the first symbol is either **a** or **b**, second symbol is **b**, third symbol is **a** and it might end with **aa** or **bb** or both. Augment the regular expression and finally convert it to equivalent DFA using syntax tree. [4]

6. Explain the need of optimization. Explain any three loop optimization technique with suitable example. [1+3]

7. Write Short notes (ANY TWO) with suitable example:
   a. Recovery Mode
   b. Attribute Grammar
   c. Indirect Triple
   d. Symbol Table

SECTION "C"

[2Q × 8 = 16 marks]

Attempt *ANY TWO* questions of the following.

[*While answering long questions, your answer must demonstrate all the steps involved in detail*]

8. 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 along with dependency graph according to your syntax directed definition. [4+4]

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

9. Consider the given grammar G:

   G: { E → T

   E → E + T

   T → int

   T → (E) }

   Construct LR(1) parsing table. Test whether the string w = int + (int + int) can be parsed or not.

10. Check whether the given grammar G is LL(1) by testing with the string w = cidbidcid

    G : { A → A c B | c C | C

    B → b B | id

    C → C a B | B b B | B }

KATHMANDU UNIVERSITY
End Semester Examination [C]
June/July, 2016

Marks Scored :

Level : B.E.
Year : IV

26 JUN 2016

Course : COMP 409
Semester : I

Exam Roll No. :

Time: 30 mins.

F.M. : 10

Registration No.:

Date :

SECTION "A"

[20Q. × 0.5 = 10 marks]

Encircle the alphabetical letter of the most appropriate answer.

1. A compiler for a high level language that runs on one machine and produce code for different machine is called
   a. Optimizing compiler
   b. One Pass compiler
   c. Cross compiler
   d. Multipass compiler

2. Synthesized attribute can be easily simulated by a
   a. LL grammar
   b. Ambiguous grammar
   c. LR grammar
   d. Recursive grammar

3. Reduction in strength means
   a. Replacing run time computation by compile time computation
   b. Removing loop invariant computation
   c. Removing common sub expression
   d. Replacing a costly operation by a relatively cheaper one

4. The optimization which avoids test at every iteration is
   a. Loop unrolling
   b. Loop jamming
   c. Constant folding
   d. None of these

5. Which of the following is not an intermediate code form?
   a. Postfix notation
   b. Syntax trees
   c. Three address codes
   d. Quadruples

6. Which of the following pairs of regular expression are equivalent?
   a. 1(01)* and 1(10)*
   b. x(xx)* and (xx)*x
   c. x^+ and x^+x^{*+}
   d. All of the above

7. Given the following attribute grammar,

   $E \rightarrow TE' \{ E'.y = T.x\}$
   $\{E.x = E'.x\}$

   $E'_1 \rightarrow + TE'_2 \{E'2.y = t(E'1.y + T.x)$
   $\{E'1.x = E'2.x\}$

   $E' \rightarrow \varepsilon \{E'.x = E'.y\}$

   a. Both x and y are inherited attributes
   b. Both x and y are synthetic attributes
   c. y is inherited and x is synthetic attribute
   d. x is inherited and y is synthetic attribute

Question number (8-11) is based on the following grammar G:

$S \rightarrow aS \mid A$

$A \rightarrow aAb \mid bAa \mid \varepsilon$

8. Which of the following string is generated by the above grammar?
   a. aabbaba b. aabaaba c. abababb d. aabbaab

9. For the correct answer in 8, how many steps are required to derive the string S and how many parse tree are there?
   a. 6 and 1 b. 6 and 2 c. 7 and 1 d. 7 and 2

10. The number of states in SLR automation is:
   a. 10 b. 11 c. 12 d. 13

11. In the predictive parsing table M, of the grammar the entries M[S,b] and M[A,b] respectively are:
   a. {S→A} and {A→bAa}
   b. {S→A; S→aS} and {A→ bAa ; A→ ε}
   c. {S→A } and {A→ bAa ; A→ ε}
   d. {S→A } and {A→ ε}

12. Any string of terminals that can be generated by the following CFG is :
   S → XY
   X →aX | bX | a
   Y →Ya | Yb | a
   a. has at least one 'b' b. should end in a 'a'
   c. has at least two a's d. has no consecutive a's or b's

13. The language of all words with at least 2 a's can be described by the regular expression
   a. (ab)*a and a(ba)* b. (a+b)* ab* a(a+b)*
   c. b* ab* a (a+b)* d. all of these

14. L = {w Є{0,1}* | w has at least one pair of consecutive zeros}
   a. (0+1) 00 (0+1)* b. (0+1)*00 (0+1)
   c. (0+1)* 00 (0+1)* d. (0+1)* (00)* (0+1)*

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

16. 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. None of the above

17. Postfix equivalent of (a+b)*c is
   a. ab+c* b. +ab*c c. +abc* d. a+bc*

18. A set of strings in the input for which the same token is produced as output is called
   a. Token b. Lexeme c. Pattern d. None of the above

19. How many states are required to validate a keyword "BEGIN"
   a. 3 b. 4 c. 5 d. 6

20. Consider the grammar with the following translation rules and E as the start symbol.

E -> E1 #T {E.value = E1.value * T.value}
E -> T {E.value = T.value}
T -> T1 & F {T.value = T1.value + F.value}
T ->F {T.value= F.value}
F -> num {F.value = num.value}

Compute E.value for the root of the parse tree for the expression: 5 # 4 & 7 # 8 &3

a. 79 b. 648 c. 605 d. 97

KATHMANDU UNIVERSITY
End Semester Examination [C]
June/July, 2016

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

26 JUN 2016

SECTION "B"

[6Q. × 4 = 24 marks]

Attempt *ANY SIX* questions.

1. What is bootstrapping? Explain with suitable example how bootstrapping is done on more than one machine.

2. What is the use of deterministic finite automata in lexical analysis? Also explain input buffering scheme.

3. Define CFG. Show that the following grammar is ambiguous.
   S → a
   S → abSb
   S → aAb
   A → bS
   A → aAAb

4. Write and explain rules with suitable example the FIRST and FOLLOW sets of any grammar of your choice.

5. Translate the expression **A = - B * (C + D) / E** into Quadruples, Triples and Indirect Triples representation.

6. Consider the following code fragment:

```
sum = 0;
for ( i=0; i<=20; i++)
        sum = sum + a[i];
```

   Design its three address representation.

7. Write Short notes on with example (*ANY TWO*) :
   a. Local and Loop optimization
   b. Induction variable elimination
   c. Dead code elimination
   d. Constant folding and combining

SECTION "C"

[2 Q × 8 = 16 marks]

Attempt *ANY TWO* questions.

8. Consider the following grammar:

$S' \rightarrow S$
$S \rightarrow ABC$
$A \rightarrow a \mid bbD$
$B \rightarrow a \mid \varepsilon$
$C \rightarrow b \mid \varepsilon$
$D \rightarrow c \mid \varepsilon$

a. Construct the FIRST and FOLLOW sets for the grammar
b. Design LL(1) parsing table for the grammar.
c. Check whether the string w = **aab** can be parsed or not.

9. Construct the following grammar:

$E \rightarrow E+T \mid T$
$T \rightarrow TF \mid F$
$F \rightarrow F^* \mid a \mid b$

a. Construct SLR item sets
b. Construct SLR parsing table
c. Check whether the string w = **a+a+ba*** can be parsed or not.

10. Show that the following grammar is LR(1) but not LALR(1).

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$
$A \rightarrow d$
$B \rightarrow d$

KATHMANDU UNIVERSITY
End Semester Examination
February/March, 2019

Marks Scored:

Level : B. E./B. Sc.
Year : IV/III

Course : COMP 409
Semester : I/II

Exam Roll No. :

Time: 30 mins.

F. M. : 10

Registration No. :

Date 10 MAR 2019

SECTION "A"
[20Q × 0.5 = 10 marks]

Tick (√) the best answer(s) or fill in the blanks with most appropriate word/phrase.

1. Programming that uses a form of symbolic logic as a programming language is often called.....
[A] Imperative Language
[B] Declarative Language
[C] Functional Language
[D] Object-oriented Language

2. ..... not only performs the text-creation and modification functions of an ordinary text editor, but it also analyzes the program text.
[A] Structure Editors [B] Pretty Printers [C] Static Checkers [D] Interpreters

3. During..... the stream of characters making up the source program is read from left-to-right and grouped into tokens.
[A] Linear Analysis
[B] Hierarchical Analysis
[C] Semantic Analysis
[D] Syntax Analysis

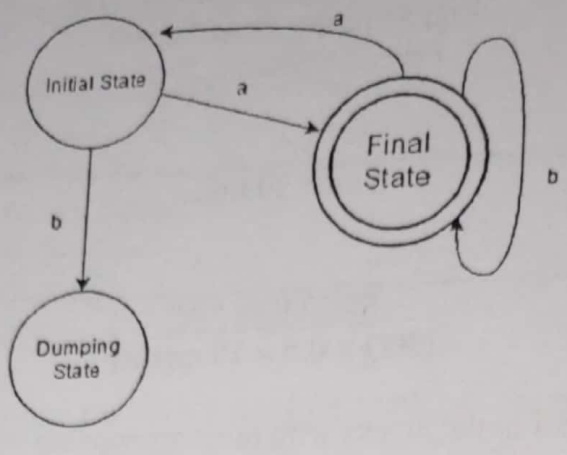
4. Positive closure $\sum^+$ is
[A] $\sum$ [B] $\sum^*$ [C] $\sum^* - \sum^0$ [D] $\sum^* + \sum^0$

5. The language which is generated by the grammar $S \rightarrow aSa \mid bSb \mid a \mid b$ over the alphabet {a,b} is the set of
[A] All odd and even length palindromes
[B] All even length palindrome
[C] String that begins and end with the different symbol
[D] All odd length palindromes

6. For predictive parsing the grammar $A \rightarrow AA \mid (A) \mid \varepsilon$ is not suitable because
[A] The grammar is right recursive
[B] The grammar is ambiguous
[C] The grammar is left recursive
[D] The grammar is operator grammar

7. The regular expression for the set of strings of 0's and 1's whose third symbol from the right end is 1.
[A] (0+1)*11(0+1)
[B] (0+1)*1(0+1)(0+1)*
[C] (0+1)(0+1)1(0+1)*(0+1)
[D] (0+1)*1(0+1)(0+1)

8. Which of the following will not be accepted by the following finite automata?

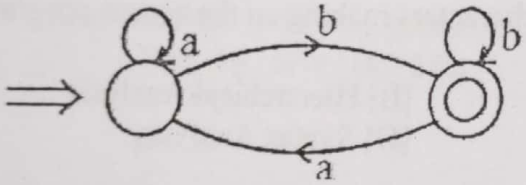


[A] abaabbbbaabb [B] abaabbbbaabbaa [C] abaabaabbaab [D] abaabbbbab

9. Given the language L = {ab, aa, baa}, which of the following strings are in L*?
1) abaabaaabaa
2) aaaabaaaa
3) baaaaabaaaab
4) baaaaabaa

[A] 1, 2 and 3 [B] 2, 3 and 4 [C] 1, 2 and 4 [D] 1, 3 and 4

10. The regular expression for the following DFA is



[A] ab*(b + aa*b)* [B] a*b(b + aa*b)* [C] a*b(b* + aa*b) [D] a*b(b* + aa*b)*

11. Shift reduce parser announces successful completion of parsing if action is
[A] Shift [B] Accept [C] Reduce [D] Error

Consider the following grammar

$S \rightarrow ACB \mid CbB \mid Ba$
$A \rightarrow da \mid BC$
$B \rightarrow g \mid \varepsilon$
$C \rightarrow h \mid \varepsilon$

Question (12to 13) are based on the given grammar

12. FIRST(S)
[A] {d,g,h,b,a} [B] {d,g,h,ε,b,a} [C] { d,g,h, b}{ [D] {d,g,h,ε, b}

13. FOLLOW(B)
[A] {g,$,a,b,h} [B] {g,$,a,b} [C] {g,$,b,h} [D] {g,h,a, $}

14. Consider the following grammar

S →Aa | bAC | dC | bda

A→ d

[A] The grammar is not LL(1)
[B] The grammar is LR(0)
[C] The grammar is not LL(1) but SLR(1)
[D] The grammar is SLR(1) only

15. Consider the following grammar

S→ AaAb BbBa

A → ε

B→ ε

[A] The grammar is not LL(1)
[B] The grammar is LR(0)
[C] The grammar is LL(1) but not SLR(1)
[D] The grammar is not LALR(1) but CLR(1)

16. A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar

S → xxW (PRINT "1")
S → y (PRINT "2")
W →Sz (PRINT "3")

What is the translation of xxxxyzz using the syntax directed translation scheme described by the above rules?

[A] 23131 [B] 23132 [C] 11233 [D] 33211

17. When is the type checking usually done?

[A] During syntax analysis
[B] During lexical analysis
[C] During code optimization
[D] During syntax directed translation

18. Optimization of program that works within a single block is called

[A] Global optimization
[B] Local optimization
[C] Loop controlling
[D] Loop un-controlling

19. Which of the following is not a loop optimization?

[A] Copy propagation
[B] Loop fusion
[C] Loop unrolling
[D] None of these

20. Consider the following codes, on the left is original and on the right is after optimization.

Before	After
r7 = 4 + 1	r7 = 5
r5 = 2 * r4	r5 = 2 * r4
r6 = r5 * 2	r6 = r4 * 4

This is an example of

[A] constant folding
[B] copy propagation
[C] constant combining
[D] both a and c

KATHMANDU UNIVERSITY
End Semester Examination
February/March, 2019

10 MAR 2019

Level : B. E./B. Sc.
Year : IV/III
Time : 2 hrs. 30 mins.

Course : COMP 409
Semester : I/II
F. M. : 40

SECTION "B"

[6Q. × 4 = 24 marks]

Attempt *ANY SIX* questions.

1. A compiler uses a two-pass scheme of translation, performing analysis of the source program in the first pass, and synthesis of the target program in the second pass. Clearly explain what use is made of the symbol table in the two passes of the compiler.

2. Construct a NFA for the regular expression (a | b)*ab. Convert this NFA into an equivalent DFA.

3. Give the DFA for languages of strings over {0,1} in which
   a. Each string ends with 00.
   b. The set of all string whose third last symbol is 1

4. How will you determine, with the help of parse tree, that the given grammar is ambiguous? Explain with example.

5. What is type checking? Explain about static and dynamic type checking.

6. Explain the importance of transformations on basic blocks in code generation and optimization. Also explain each phase of local transformation with suitable example.

7. Write short notes on
   a. Regular expression
   b. Union operation on items
   c. Inherited attributes
   d. Type systems

SECTION "C"

[2Q. × 8 = 16 marks]

Attempt *ANY TWO* questions.

8. Consider the grammar

lexp → atom | list
atom → **number** | **identifier**
list → (lexp-seq)
lexp-seq → lexp-seq lexp | lexp

[2+2+4]

   a. Remove the left recursion
   b. Construct First and Follow sets for the resulting grammar
   c. Construct the LL(1) parsing table for the resulting grammar.

9. Discuss the algorithm for computation of the sets of LR(1) items. Also show that the following grammar is LR(1) but not LALR(1) [2+3+3]

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

10. Explain about syntax-directed definition and dependency graph. Consider the following syntax-directed definition and construct an annotated parse tree for input expression (4*7+1)*2, along with dependency graph. [2+2+2+2]

Production	Semantic Rule
$L \rightarrow En$	print(E.val)
$E \rightarrow E_1 + T$	E . val := E_1 . val + T . val
$E \rightarrow T$	E . val := T . val
$T \rightarrow T_1 * F$	T . val := T_1 . val × F . val
$T \rightarrow F$	T . val := F . val
$F \rightarrow (E)$	F . val := E . val
$F \rightarrow digit$	F . val := digit . lexval