KATHMANDU UNIVERSITY
End Semester Examination
August, 2018

Level	: B.Sc.	Course	: COMP 409
Year	: III	Semester	: II
Time	: 2 hrs. 30 mins.	F. M.	: 40

SECTION “B”

[6Q. × 4 = 24 marks]

Attempt *ANY SIX* questions.

1. Draw the block diagram of compiler? Explain different steps in analysis phase.

2. Construct nondeterministic finite automata for the regular expression (a | b)*abb(a | b)* and convert it into DFA. Show the sequence of moves for processing the input string ababbab.

3. Define annotated parse tree. Given a syntax-directed definition of a simple desk calculator.

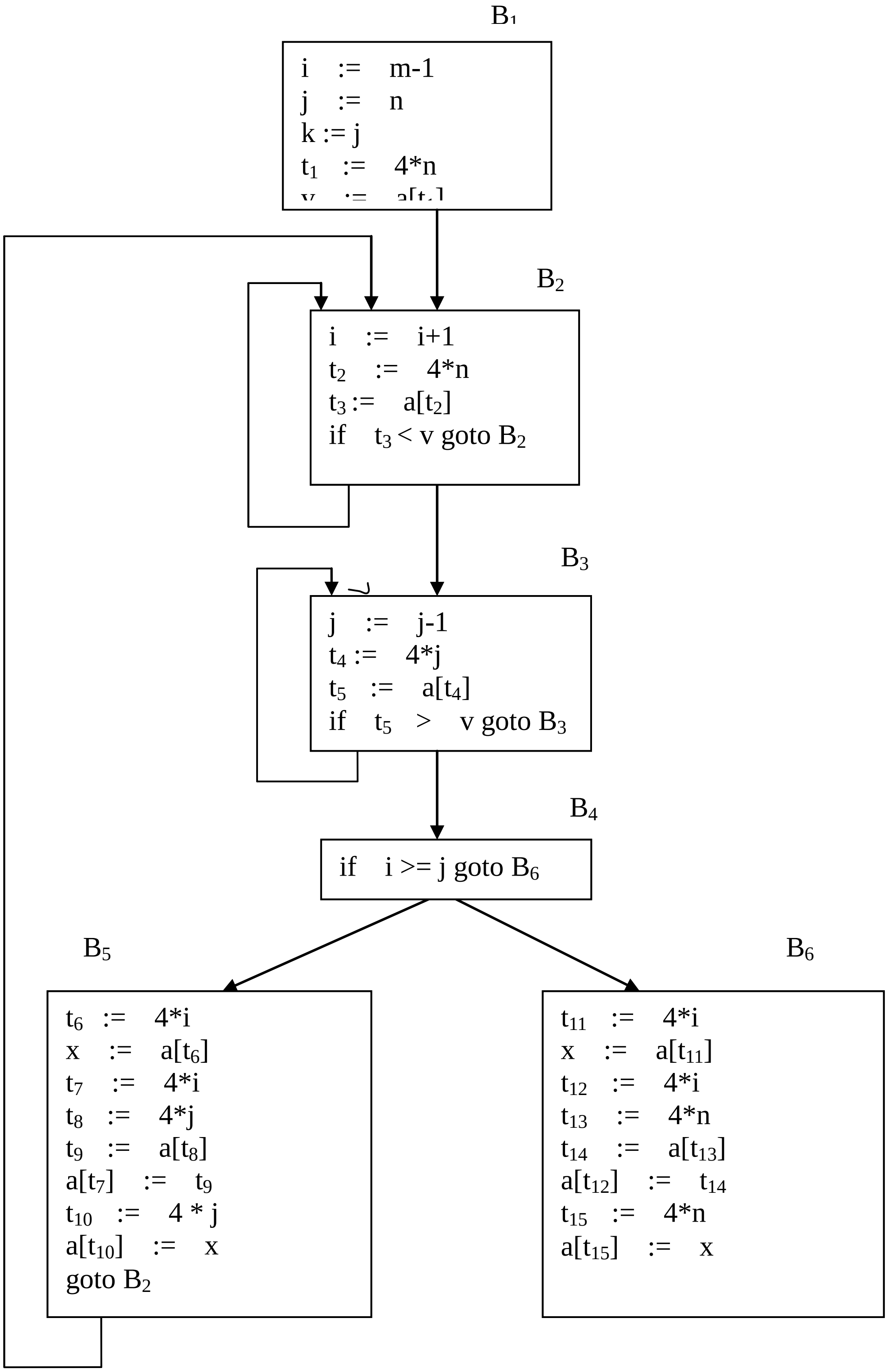
Production	**Semantic Rules**
L → E **n**	Print (E.val)
E →E_1 +T	E.val := E_1.val + T.val
E→ T	E.val := T.val
T →T_1*F	T.val := T_1.val × F.val
T →F	T.val := F.val
F→ (E)	F.val := E.val
F →digit	F.val := digit.lexval

For the input expression (4*7+1)*2, construct an annotated parse tree.

4. Differentiate between recursive decent parsing and non recursive predictive parsing.

5. What is type checking? Write the type expression for the following types:
   a) Function whose domains are functions from two characters and whose range is a pointer of integer.
   b) An array of pointers to real where the array index ranges from 1 to 50.

6. Consider the following flow graph



a) Show the local common sub expression elimination for block B_5

b) Eliminate the common sub expression from block B_5 and B_6

c) Eliminate the dead code from the graph with explanation if any

7. Write short notes on

a. Three-address code b. Equivalence of type expression
c. Activation tree d. Dead code elimination

SECTION “C”

[2 Q × 8 = 16 marks]

Attempt *ANY TWO* questions.

8. Consider the grammar

S → AA
A → bA
B →c

a. Describe the language that the grammar defines.
b. Is the grammar ambiguous? Justify your answer.
c. Construct a Canonical LR(1) and LALR(1) parse table for the grammar.

9. Construct the LL(1) parsing table for the following grammar

E → TE’
E’→ +TE’ | ε
T →FT’
T’→ *FT’ | ε
F→ (E) | id

Also trace for the input string id+id*id showing input and stack content.

10. Consider the following grammar and give the syntax directed definition to construct parse tree. For the input expression 7*2+5*4, construct an annotated parse tree along with dependency graph according to your syntax directed definition.

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

KATHMANDU UNIVERSITY
End Semester Examination
August, 2018

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

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

Registration No.: Date :

SECTION “A”

[20 Q × 0.5 = 10 marks]

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

1. During compilation which of the following phase require the most specialized technique?
[A] Semantic Analysis [B] Analysis
[C] Lexical Analysis [D] Synthesis

2. Which of the following translates a predicate containing relational and Boolean operators into commands to search a database for records?
[A] Query Interpreters [B] Silicon Compiler
[C] Test Formatters [D] Linear Analysis

3. A major advantage of _______________ is that it can utilize the best-known pattern-matching algorithms.
[A] Lexical Analyzer Generator [B] Semantic Analysis
[C] Syntax Analyzer [D] Code Optimizer

4. The number of tokens in the following C statement is
printf(“a=%d, &a=%x”,a,&a);
[A] 3 [B] 26 [C] 21 [D] 10

5. What a compiler can check?
[A] Logical Error [B] Syntax Error
[C] Both Logical and Syntax Error [D] Not Logical and Syntax Error

6. 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

7. The regular expression have all strings of 0’s and 1’s with no two consecutive 0’s is
[A] (0+1) [B] (0+1)* [C] (0+ε) (1+10)* [D] (0+1)* 011

8. Consider the grammas shown below:
S → AA
A→ aA | b
The grammar is
[A] LL(1) [B] SLR(1) but not LL(1)
[C] LR(1) but not LALR(1) [D] LALR(1) but not SLR(1)

9. Consider the grammar

S → (S) | a

Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar be n1, n2, n3 respectively. The following relationship holds true

[A] n1<n2<n3 [B] n1=n3<n2 [C] n1=n2=n3 [D] n1≥n3≥n2

10. Consider the grammar with non-terminals N = { S, C, S1}, terminals T ={a, b, i, t, e}, with S as the start symbol, and the following set of rules:

S→ iCtSS1| a

S1→ eS | ε

C→ b

The grammar is NOT LL(1) because:

[A] It is left recursive [B] It is right recursive

[C] It is ambiguous [D] It is not context-free

11. A bottom up parser generates

[A] Rightmost derivation in reverse [B] Rightmost derivation

[C] Leftmost derivation [C] Leftmost derivation in reverse

12. Which of the following statement is false?

[A] LALR is more powerful than SLR

[B] An ambiguous grammar can never be LR(k) for any k

[C] An ambiguous grammar has same leftmost and rightmost derivation

[D] An LL(1) parser is a top-down parser

Consider the following grammar

S → AS’

S’ → +AS’ | ε

A → BT’

A’ → *BA’ | ε

B → (S) | id

Question (13 and 14) are based on the given grammar

13. FIRST(S) is:

[A] {+, ε} [B] {+,), $} [C] {*, ε } [D] { (, id}

14. FOLLOW(A) is:

[A] {+, ε } [B] {+,), $} [C] {*, ε } [D] { (, id}

15. Consider the translation scheme as shown below

S →TR

R→ +T {print(‘+’);}R | ε

T→ num {print(num.val);}

Here num is a token that represents an integer and num.val represents the corresponding integer value. For an input ‘6+3+2’, this translation scheme will print

[A] 6+3+2 [B] 63+2+ [C] 632++ [D] ++ 632

16. Consider the following translation scheme
S→ ER
R →* E{print{“*”);R | ε
E→ F+ E{print(“+”);}| F
F → (S) | id{print(id.value);
Here id is a token that represents an integer and id.value represents the corresponding integer value. For an input ‘2*3+4’, this translation scheme prints
[A] 2*3+4 [B] 2*+34 [C] 23*4+ [D] 234+*

17. The identification of common sub-expression and replacement of run-time computations by compile-time computations is
[A] Constant Folding [B] Loop Optimization
[C] Local Optimization [D] Data flow analysis

18. The method which merges the bodies of two loops is
[A] Loop Rolling [B] Constant folding
[C] Loop Jamming [D] Algebraic Simplification

19. Optimization of the program that works within a single basic block is called
[A] Global optimization [B] Loop un-controlling
[C] Loop controlling [D] Local optimization

20. The optimization technique which is typically applied on loops is
[A] Removal of invariant computation [B] Peephole optimization
[C] Constant folding [D] All of above

Solution to MCTI problems

MCTI	Answer
1	D
2	A
3	A
4	D
5	B
6	C
7	C
8	A
9	B
10	C
11	A
12	C
13	D
14	B
15	B
16	D
17	A
18	C
19	D
20	D