26-28

Begum Rokeya University, Rangpur.

Department of Computer Science and Engineering

(B.Sc. Engg.) 4th Year 1st Semester Final Examination, 2017.(Session:2013-14)

Course Title: **Compiler Design** ; Course Code: **CSE 4101**

Full Marks: 50 Time:3.00 Hours

Answer Any Five from the Given Questions

(Note: Numbers in the right margin indicate marks for each question. Answer questions sequentially)

---

1. (a) What do you mean by front end in compiler design? Show the output produced by it in different stages for ***a:=b*c/36***; where a, b and c are real numbers. 4

   (b) Explain the way in which high level languages are processed by interpreter and compiler. 2

   (c) Explain the role of assembler, compiler, loader and linker in the language processing system. 4

2. (a) Define lexeme, token and pattern. Identify the lexemes that make up the tokens in the following program segment ***E =M*C**2***. Indicate corresponding token and pattern. 4

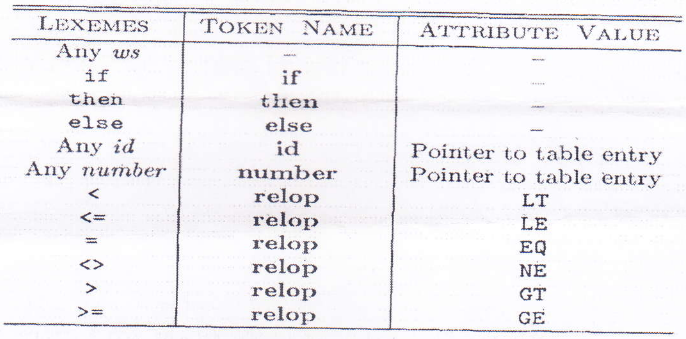
   (b) What is the relationship with lexical analyzer, regular expressions and transition diagram? Give an example. 3

   (c) Write a regular expression for unsigned numbers and design the transition diagram for it. 3

3. (a) You are given a table (Table-1) containing some **tokens**, their tokens and corresponding attribute value. Now write a **lex** program to identify these lexemes. 5

Table-1

2.22F(+1-(t)2



LEXEMES	TOKEN NAME	ATTRIBUTE VALUE
Any *ws*	–	–
if	if	–
then	then	–
else	else	–
Any *id*	id	Pointer to table entry
Any *number*	number	Pointer to table entry
<	relop	LT
<=	relop	LE
=	relop	EQ
<>	relop	NE
>	relop	GT
>=	relop	GE

   (b) Construct an NFA for the regular expression: **(a|b)*abb** and convert this **NFA** to corresponding **DFA** by subset construction of that **NFA**. 5

4. (a) What are the four components of a **CFG**? Define with example. 3

   (b) Give a leftmost and rightmost derivations of the following grammar and hence test for the ambiguity of the grammar: 5

   $E \rightarrow E + T \mid T$ E → T

   $T \rightarrow T * F \mid F$ T → F

   $F \rightarrow (E) \mid a$ F → a

   (c) How can we eliminate ambiguity of a grammar? 2

5. (a) What is an **LL(1)** grammar? When the grammar is said to be **LL(1)** grammar? 2

   (b) Write the comparisons between SLR and LALR parser 2

   (c) Write the conflicts that may occur during shift reduce parsing. Consider the dangling-else grammar. 3

   (d) What do you mean by Handle pruning? Left factor the given grammar: 3

   $L \rightarrow \text{int} \mid \text{int}, L \mid (L)$

6. You are given the following grammar:

$A \rightarrow A * B | B$

$B \rightarrow B + E | E$

$E \rightarrow (K) | \boldsymbol{id}$ $E \rightarrow (A) | id$

Using LL(1) parser-

(i) Check for the recursion and factor of grammar

(ii) Identify the FIRST() and FOLLOW() of above grammar

(iii) Produce corresponding parse table

(iv) Push the operations in a Stack according to Inputs of parse table

(v) Generate the corresponding parse tree.

7. **(a)** What is the purpose of defining augmented grammar before constructing LR(0) items? 3

(b) What are different intermediate code forms? Convert the expression **a = b * - c + b * - c** into three address statements. 4

(c) What is code optimization? Explain machine dependent and independent code optimization. 3

~~$A \rightarrow BA'$~~ $B \rightarrow EB'$

~~$A' \rightarrow *BA' | \epsilon$~~ ~~$B' \rightarrow +EB' | \epsilon$~~

~~$E \rightarrow (A) | id$~~

26-78

Begum Rokeya University, Rangpur.

Department of Computer Science and Engineering

(B.Sc. Engg.) 4th Year 1st Semester Final Examination, 2017.(Session:2013-14)

Course Title: **Compiler Design** ; Course Code: **CSE 4101**

Full Marks: 50 Time:3.00 Hours

Answer Any Five from the Given Questions

(Note: Numbers in the right margin indicate marks for each question. Answer questions sequentially)

1. (a) What do you mean by front end in compiler design? Show the output produced by it in different stages for ***a:=b*c/36***; where a, b and c are real numbers. 4

   (b) Explain the way in which high level languages are processed by interpreter and compiler. 2

   (c) Explain the role of assembler, compiler, loader and linker in the language processing system. 4

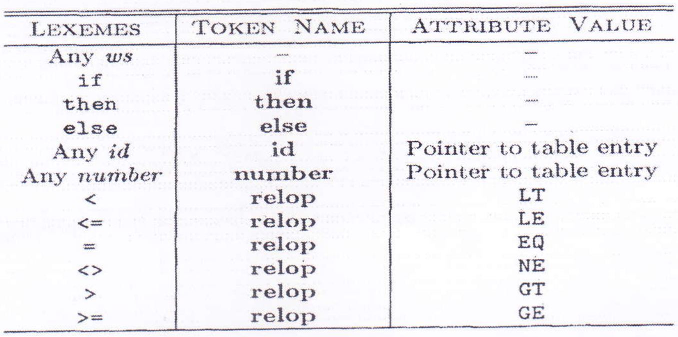
2. (a) Define lexeme, token and pattern. Identify the lexemes that make up the tokens in the following program segment ***E =M*C**2***. Indicate corresponding token and pattern. 4

   (b) What is the relationship with lexical analyzer, regular expressions and transition diagram? Give an example. 3

   (c) Write a regular expression for unsigned numbers and design the transition diagram for it. 3

3. (a) You are given a table (Table-1) containing some **tokens**, their tokens and corresponding attribute value. Now write a **lex** program to identify these lexemes. 5

Table-1



LEXEMES	TOKEN NAME	ATTRIBUTE VALUE
Any *ws*	–	–
if	if	–
then	then	–
else	else	–
Any *id*	id	Pointer to table entry
Any *number*	number	Pointer to table entry
<	relop	LT
<=	relop	LE
=	relop	EQ
<>	relop	NE
>	relop	GT
>=	relop	GE

   (b) Construct an NFA for the regular expression: **(a|b)*abb** and convert this **NFA** to corresponding **DFA** by subset construction of that **NFA**. 5

4. (a) What are the four components of a **CFG**? Define with example. 3

   (b) Give a leftmost and rightmost derivations of the following grammar and hence test for the ambiguity of the grammar: 5

   $E \rightarrow E + T \mid T$

   $T \rightarrow T * F \mid F$

   $F \rightarrow (E) \mid a$

   (c) How can we eliminate ambiguity of a grammar? 2

5. (a) What is an **LL(1)** grammar? When the grammar is said to be **LL(1)** grammar? 2

   (b) Write the comparisons between SLR and LALR parser 2

   (c) Write the conflicts that may occur during shift reduce parsing. Consider the dangling-else grammar. 3

   (d) What do you mean by Handle pruning? Left factor the given grammar: 3

   $L \rightarrow \text{int} \mid \text{int, L} \mid \text{(L)}$

6. You are given the following grammar:

$A \rightarrow A * B|B$

$B \rightarrow B + E|E$

$E \rightarrow (K)|\ \boldsymbol{id}$

Using LL(1) parser-

(i) Check for the recursion and factor of grammar

(ii) Identify the FIRST() and FOLLOW() of above grammar

(iii) Produce corresponding parse table

(iv) Push the operations in a Stack according to Inputs of parse table

(v) Generate the corresponding parse tree.

7. **(a)** What is the purpose of defining augmented grammar before constructing LR(0) items? 3

(b) What are different intermediate code forms? Convert the expression **a = b * - c + b * - c** into three address statements. 4

(c) What is code optimization? Explain machine dependent and independent code optimization. 3

26-28

Begum Rokeya University, Rangpur.

Department of Computer Science and Engineering

(B.Sc. Engg.) 4th Year 1st Semester Final Examination, 2017.(Session:2013-14)

Course Title: **Compiler Design** ; Course Code: **CSE 4101**

Full Marks: 50

Time:3.00 Hours

Answer Any Five from the Given Questions

(Note: Numbers in the right margin indicate marks for each question. Answer questions sequentially)

1. (a) What do you mean by front end in compiler design? Show the output produced by it in different stages for ***a:=b*c/36***; where a, b and c are real numbers. 4

   (b) Explain the way in which high level languages are processed by interpreter and compiler. 2

   (c) Explain the role of assembler, compiler, loader and linker in the language processing system. 4

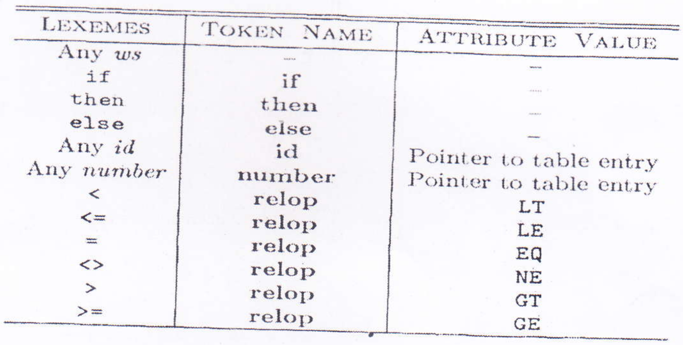
2. (a) Define lexeme, token and pattern. Identify the lexemes that make up the tokens in the following program segment ***E =M*C**2***. Indicate corresponding token and pattern. 4

   (b) What is the relationship with lexical analyzer, regular expressions and transition diagram? Give an example. 3

   (c) Write a regular expression for unsigned numbers and design the transition diagram for it. 3

3. (a) You are given a table (Table-1) containing some **tokens**, their tokens and corresponding attribute value. Now write a **lex** program to identify these lexemes. 5

Table-1



LEXEMES	TOKEN NAME	ATTRIBUTE VALUE
Any *ws*	—	—
if	if	—
then	then	—
else	else	—
Any *id*	id	Pointer to table entry
Any *number*	number	Pointer to table entry
<	relop	LT
<=	relop	LE
=	relop	EQ
<>	relop	NE
>	relop	GT
>=	relop	GE

2.22E(+|-|ε)2

   (b) Construct an NFA for the regular expression: **(a|b)*abb** and convert this **NFA** to corresponding **DFA** by subset construction of that **NFA**. 5

4. (a) What are the four components of a **CFG**? Define with example. 3

   (b) Give a leftmost and rightmost derivations of the following grammar and hence test for the ambiguity of the grammar: 5

   $E \rightarrow E + T \mid T$ $E \rightarrow T$

   $T \rightarrow T * F \mid F$ $T \rightarrow F$

   $F \rightarrow (E) \mid a$ $F \rightarrow a$

   (c) How can we eliminate ambiguity of a grammar? 2

5. (a) What is an **LL(1)** grammar? When the grammar is said to be **LL(1)** grammar? 2

   (b) Write the comparisons between SLR and LALR parser 2

   (c) Write the conflicts that may occur during shift reduce parsing. Consider the dangling-else grammar. 3

   (d) What do you mean by Handle pruning? Left factor the given grammar: 3

   $L \rightarrow$ int | int, L | (L)

6. You are given the following grammar: 10

$A \rightarrow A * B | B$

$B \rightarrow B + E | E$

$E \rightarrow (K) | \boldsymbol{id}$

E → (A)| id

Using LL(1) parser-

(i) Check for the recursion and factor of grammar

(ii) Identify the FIRST() and FOLLOW() of above grammar

(iii) Produce corresponding parse table

(iv) Push the operations in a Stack according to Inputs of parse table

(v) Generate the corresponding parse tree.

7. **(a)** What is the purpose of defining augmented grammar before constructing LR(0) items? 3

(b) What are different intermediate code forms? Convert the expression **a = b * - c + b * - c** into three address statements. 4

(c) What is code optimization? Explain machine dependent and independent code optimization. 3

~~A → BA'~~ B → EB'

~~A' → +BA' | ε~~ ~~B' → +EB' | ε~~

~~E → (A) | id~~