Date: 25/09/17

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Fourth Year, First Semester, Final Examination, Spring 2017

Course No: **CSE 4129** Course Title: **Formal Languages and Compilers**

Time: 3 hours Full Marks: 70

[There are 7 (seven) questions carrying 14 marks each. Answer any 5 (five) questions]

[*Marks allotted are indicated in the right margin*]

---

1. a) Define Finite Automata and Formal Language with suitable examples. 4

   b) Design Deterministic Finite Automata (DFA) for the following languages: 5

      i. L = {set of all strings ending with abc} where $\sum$ = {a,b,c}

      ii. L = {set of all stings with even number of 0's and even number of 1's} where, $\sum$ = {0,1}

   c) Convert the following Non-Deterministic Finite Automata (NFA) to a Deterministic Finite Automata (DFA) by using subset construction method. 5

	0	1
→P	{Q,S}	{Q}
Q	{R}	{Q,R}
R	{S}	{P}
*S	Φ	{P}

2. a) Briefly explain the concept of Equivalence and Equality of Regular Expressions with the help of following Regular Expressions R and S: 4

   R = a a* (b | c)

   S = a* a (c | b)

   b) Minimize the following Deterministic Finite Automata (DFA) by using Partition Method, where $\sum$ = {a,b} and Q = {0,1,2,3,4,5} 5

	A	B
→0	3	1
*1	4	1
*2	0	2
3	5	2
4	3	1
5	5	5

   c) Convert the following epsilon-Non-Deterministic Finite Automata (ϵ-NFA) to its corresponding Deterministic Finite Automata (DFA). 5

	ϵ	a	b	c
→P	Φ	{P}	{Q}	{R}
Q	{P}	{Q}	{R}	ϕ
*R	{Q}	{R}	ϕ	{P}

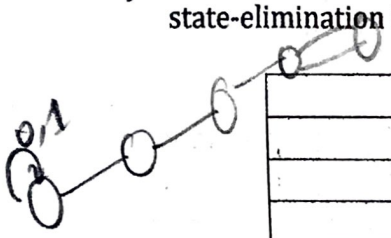
3. a) What do you understand by concatenation and union of languages? Explain with suitable examples. 4

b) Convert the following Regular Expression to Deterministic Finite Automata (DFA). 5

i. (0+1)*010(0+1), where $\sum = \{0,1\}$

ii. (c+a)*abca*, where $\sum = \{a,b,c\}$

c) Convert the following Deterministic Finite Automata (DFA) to Regular Expression using the state-elimination technique (show all intermediate steps): 5



	0	1
→P	P	S
Q	R	P
R	R	Q
*S	R	S

4. a) What are the requirements for compilation? Briefly describe the analysis-synthesis model for compilation. 4

b) Why it is important to separate lexical analyzer and parser? Briefly describe the reasons. 5

c) What do you understand by symbol table management? Write down the attributes for each token for the code statement **E = MC²**. 5

5. a) Check whether the following grammar is ambiguous or not. Justify your answer. 4

$$A \rightarrow A+A|A*A|A|a$$

b) Find out the First and Follow set of all the non-terminals of the following grammar and also generate the LL(1) parsing table for the grammar: 5

A→ TA'
A' →rTA' | ε
T → FT'
T' → m FT' | ε
F → (A)| p

c) Consider the following grammar and the LL(1) parsing tables. Now parse the string "**((id*id)+id)*(id)**" and build the final parse tree. 5

Given Grammar:
E → TE'
E' → +TE' | ε
T → FT'
T' → * FT' | ε
F → (E) | id

Corresponding LL(1) Parser Table:

	id	+	*	(	)	$
E	E → TE'			E→TE'		
E'		E'→+TE'			E'→ ε	E'→ε
T	T→ FT'			T→ FT'		
T'		T'→ ε	T'→* FT'		T'→ ε	T'→ ε
F	F→ id			F→ (E)		

6. a) What are the advantages of using buffer pairs in a scanner? How sentinels can be used to further improve the performance of the buffer pairs? 4

b) Consider the following grammar 5

S→ SS+ | SS* | a

Now perform the recursive descent parsing technique to generate a parse tree for the string **aa+a*** with the help of above grammar.

c) Construct LR(0) parser table for the following grammar: 5

A → TB

B →rTB

T → FC

C → m FT

F → (A) |r

7. a) Define annotated parse tree, synthesized attribute and inherited attribute with examples. 4

b) What do you understand by handle pruning in shift-reduce parser? Perform shift-reduce parser algorithm for the string **aTRe** with the help of following grammar and state the total number of shift and reduce operations. 5

Given grammar:

S → a T R e

T → T b c | b

R → d

c) What is left sentential form and right sentential form? Apply Left Factoring on the following grammar: 5

A → ad | a | ab | abc | b