SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)	**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY** **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

15CS301 THEORY OF COMPUTATION
IMPORTANT QUESTIONS
SIMPLE 4 MARK QUESTION WITH ANSWER

UNIT I

Part A

1. Define the term NFA (Dec 2015)

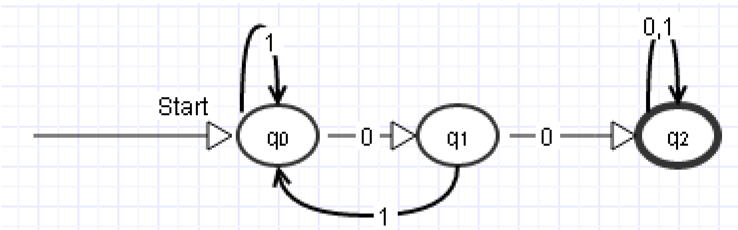
A Non-deterministic Finite automaton is a mathematical model of computation used to design computer program. It is represented by a collection of 5 tuple (Q, Σ,δ,q_0,F) where,

- Q is a finite set of states, which is non empty
- Σ is input alphabet, indicates input set
- q_0 is an initial state, q_0 is in Q
- F is a set of final states
- δ is a transition function

2. Define deductive proof (Nov/Dec 2014)

Deductive proof consists of statement whose truth lead us from initial statement called hypothesis to a conclusion statement. If 'H' is hypothesis and 'C' is conclusion, then the statement would be *"if H then C"*

3. Design DFA to accept strings over $\sum = (0,1)$ with two consecutive 0's (Nov/Dec 2014)



4. Prove or disprove that $(r+s)^* = r^* + s^*$ (Nov/Dec 2014)

LHS	RHS
$(r+s)^*$	$r^* + s^*$
={ε, r,rr,s,ss,rs,sr,…}	={ε, r,rr,s,ss,rrr,sss,…}
= {ε, any combination of r and s}	= {ε, any combination of only r or any combination of only s}

5. State the pumping lemma for regular languages. (Nov/Dec 2014,2013, June 2016)

In the theory of formal languages, the pumping lemma for regular languages describes an essential property of all regular languages. Informally, it says that all sufficiently long words in a regular language may be pumped (i.e), have a middle section of the word repeated an arbitrary number of times to produce a new word that also lies within the same language.

Pumping Lemma : If A is a regular language, then there is a pumping length ***p*** such that:

If ***s*** is any string in A of length at least p,

Then ***s*** may be divided into three pieces, s= x**y**z, satisfying the following condition:

- for each $i \geq 0$, $x\, y^i\, z \in A$
- $|y| > 0$
- $|xy| \leq p$

6. What is a finite automaton (Nov/Dec 2014,Nov/Dec 2015)

Finite state automaton is a mathematical model of computation used to design computer program. Finite automata is represented by a collection of 5 tuple $(Q, \Sigma, \delta, q_0, F)$ where,

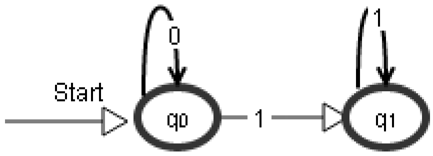
- Q is a finite set of states, which is non empty
- Σ is input alphabet, indicates input set
- q_0 is an initial state, q_0 is in Q
- F is a set of final states
- δ is a transition function

7. Enumerate the difference between DFA and NFA (Nov/Dec 2014)

SN	DFA (May/Jun 2013)	NFA(Nov/Dec 2013)
1	Deterministic Finite state automaton is a mathematical model of computation used to design computer program. Finite automata is represented by a collection of 5 tuple $(Q, \Sigma, \delta, q_0, F)$ where, Q is a finite set of states, which is non empty Σ is input alphabet, indicates input set q_0 is an initial state, q_0 is in Q F is a set of final states δ is a transition function	Like DFA an NFA has finite set of input set, one start state and set of accepting states. The main difference between NFA and DFA is that in NFA, δ function takes a state and input symbol as argument as DFA, but it returns set of zero, one or more states. The NFA is represented by a collection of 5 tuple $(Q, \Sigma, \delta, q_0, F)$ where, Q is a finite set of states, which is non empty Σ is finite set of input alphabet, q_0 is an initial state, q_0 is in Q F is a set of final states (accepting) state δ is a transition function
2	Every input string leads to the unique state FA	For the same input there can be more than one next states.
3	Conversion of regular expression (RE) of DFA is complex	Regular expression can be easily converted to NFA using Thompson's construction
4	The DFA requires more memory for storing the state information	The NFA requires more computations to match RE with input

5	The DFA it is not possible to move to next state without reading any symbol.	In NFA we can move to next state without reading any symbol.
6	Q ×∑ =Q	Q ×∑ = 2^Q

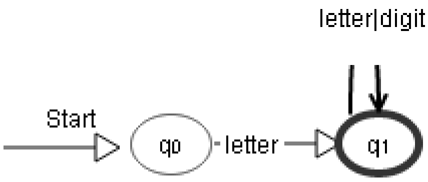
8. Construct a finite automaton for the regular expression 0^*1^* (May/Jun 2014)



9. Mention the closure properties of regular languages. (May/Jun 2014)

If ε-NFA's recognize the languages that are obtained by applying an operation on the ε -NFA recognizable languages then ε-NFA's are said to be closed under the operation. The ε-NFA's are closed under the following operations: Union, Intersection, Concatenation, Negation, Star, and Kleene closure

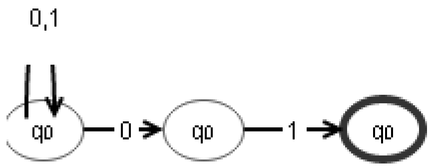
10. Draw the transition diagram (automata) for an identifier (Nov/Dec 2013)



11. (i)Construct a r.e for the language which accepts all strings with atleast two c's over the set Σ={c,b} (ii)Construct a r.e for the language over the set Σ={a,b} in which total number of a's are divisible by 3

Ans. (i) (b+c)* c (b+c)* c (b+c)* (ii) (b* a b* a b* a b*)*

12. Construct NFA equivalent to the regular expression (0+1)01 (Nov/Dec 2013)



13. Differentiate L* and L^+

- L* denotes Kleeneclosure . If L is a set of symbols or characters then L* is the set of all strings over symbols in L, including the empty string εEg. 1* = {ε,1,11,111,1111,……}
- L^+ denotes the positive closure (Kleenpluse). It is the set of all symbols / character in the set L except the empty string Eg. 1^+ = {1,11,111,……}

14. Define the term Epsilon transition (May/Jun 2013, Dec 2015)

In the NFA the transition that does not require input symbols for state transition and is capable of transiting to zero or more states with ε is called epsilon transition.

15. What is a regular expression (May/Jun 2013)

It is a sequence of character that defines a search pattern used for pattern matching. Let Σ be an alphabet used to denote the input set. the regular expression (RE) over Σ can be defined as

- ϕ is a RE which denote the empty set
- ∈ is a RE which denote the null string
- For each 'a' in Σ is denoted as the set {a}
- If 'r' and 's' are RE denoting the language L1 and L2 then
  - ' r + s' is equivalent to L1 ∪ L2
  - 'r s' is equivalent to L1 L2
  - 'r*' is equivalent to L1* (closure)

16. Name any four closure properties of Regular languages. (May/Jun 2013)

a. The union of two regular language is regular
b. The intersection of two regular languages is regular
c. The complement of regular language is regular
d. The closure operation on a regular language is regular

17. State the principle of induction (Nov/Dec 2012)

The proof by Mathematical induction can be carried out using following steps:

1. Basis: In this step we assume the lowest possible value.
2. Induction Hypothesis: In this step we assign value of n to some other value K.
3. Inductive Step: In this step, if n = k is true then we check whether the result is true for n = k + 1 or not. If we get the same result at n = k+1 then we can state that given proof is true by principle of mathematical induction.

18. Give English description of the following language (0 + 10)*1*

The set of strings of 0's and 1's without any pair of consecutive 1's substring except at the end.

19. What is Structural induction (Nov/Dec 2011)

Structural induction is a kind of proof by induction. It is used to prove that there exists a relationship between some function of integers and a given formula. For this proof technique:

- Prove the pattern is true for smallest number
- Assume it holds for an arbitrary number n
- Prove that if it is true for n, then it must be true for n+1 as well

20. What are the application of automata theory

- In compiler construction.
- In switching theory and design of digital circuits.
- To verify the correctness of a program.
- Design and analysis of complex software and hardware systems.
- To design finite state machines such as Moore and mealy machines

21. **Give regular expression for the following: (Nov/Dec 2012)**
    a. **L1 = Set of all strings of 0 and 1 ending in 00**
    b. **L2 = set of all strings of 0 and 1 beginning with 0 and ending with 1**

Ans: L1 = (0+1)*00 L2 = 0 (0+1)* 1

Part B/Part C

THEORETICAL QUESTION

1. **State the Thomson construction algorithm and subset construction algorithm, Construct finite automata for generating any floating point number with an exponential factor for example numeric value of the form 1.23 e^{-10}. Trace for a string. (16) (June 2016)**
2. **Use mathematical induction to solve the problem of Fibonacci series and examine the relationship between recursive definition and proofs by induction. Also state the inductive proofs. (16) (June 2016)**
3. **Prove that "A language L is accepted by some DFA if and only if L is accepted by some NFA" (10) (Dec 2015)**
4. **Discuss on the relation between DFA and Minimal DFA (6) (Dec 2015)**
5. **Discuss on Finite automata with epsion transitions (6) (Dec 2015)**
6. **Let L be a set accepted by a NFA and then prove that there exists a DFA that accept L. (8) (Nov/Dec 2014)**
7. **Prove that a language L is accepted by some NFA if and only if L is accepted by some DFA (8) (Nov/Dec 2014)**
8. **State the pumping lemma for Regular language. Show that the set L =$\{0^{i2}|i \geq 1\}$ not regular (6) (Dec 2015)**
9. **Determine whether the following languages are regular or not with proper justification**
    a. **L1 = $\{a^n bc^{3n} \mid n \geq 0\}$ (8) (June 2016)**
    b. **L2 = $\{a^{5n} \mid n \geq 0\}$**
10. **Prove that $L = \{0^{i^2}$ / i is an integer i ≥ 1} is not regular (8) (May/June 2014)**
11. **Show that the given Language is not regular L = $\{a^n b^m \mid n<m \text{ and } n,m \geq 1\}$ (8 marks)**

PROBLEMATIC QUESTION

1. **Construct a NFA that accepts all strings the end in 01. Give its transition table and the extended transition function for the input string 00101. Also construct a DFA for the above NFA using subset construction method (10) (June 2016)**
2. **Prove the following by principle of induction** $\sum_{x=1}^{n} x^2 = \frac{n(n+1)(2n+1)}{6}$ **(June 2016) (6)**
3. **What is regular expression ? Write expression for set of strings that consists of alternating 0's and 1's (8) (June 2016)**
4. **Design a minimized DFA by converting the following regular expression to NFA, NFA-λ and to DFA over the alphabet Σ = {a,b,c}*. RE = a(a+b+c)* (a+b+c) (16) (June 2016)**
5. **Construct determine finite automata that recognize the regular expression defined over the alphabet Σ={0,1}. RE = (1 +110)*0. Trace for a string acceptance and rejection. (8) (June 2016)**
6. **Construct Finite automata equivalent to the regular expression (ab+a)* (6) (Dec 2015)**
7. **Construct DFA to accept the language L = {w | w is of even length and begins with 10} (6) (Dec 2015)**
8. **Convert the following NFA to a DFA (10) (Dec 2015)**

	0	**1**
p	**{p,q}**	**{p}**
q	**{r,s}**	**{t}**
r	**{p,r}**	**{t}**
***s**	ϕ	ϕ
***t**	ϕ	ϕ

9. **Construct a DFA equivalent to the NFA M = ({a,b,c,d}, {0,1}, δ, a, {b,d}) where δ is a defined as: (8) (Nov/Dec 2014)**

δ	**0**	**1**
a	**{b,d}**	**{b}**
b	**c**	**{b,c}**
c	**d**	**a**
d	**--**	**a**

10. **Prove the following by the principle of induction: (8)**

$$\sum_{k=1}^{n} K^2 = \frac{n(n+1)(2n+1)}{6}$$ **(May/June 2014)**

11. **Construct a DFA that accepts all strings on {0,1} except those containing the substring 101. (8) (May/June 2014)**

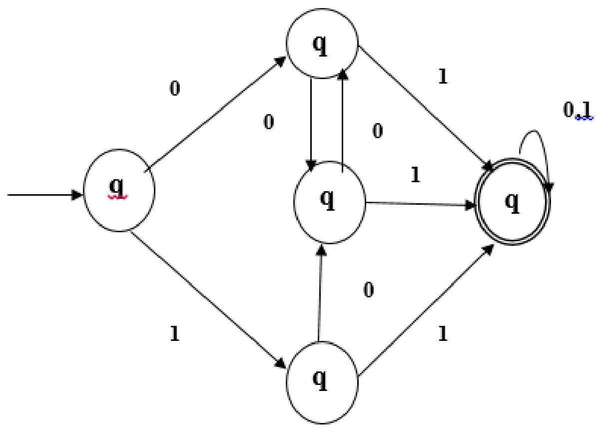
12. **Construct a NFA accepting the set of strings over {a,b} ending in *aba*. Use it to construct a DFA accepting the same set of strings. (8) (May/June 2014)**

13. **Construct a DFA through NFA with ε moves which accepts a language consisting the strings of any number of a's followed by any number of b's followed by any number of c's. (8) (May/June 2014)**

14. **Design a finite automaton for the regular expression $(0+1)^{*}(00+11)(0+1^{*})$ (8) (May/June 2014)**

15. **Find the regular expression of a language that consist of set of string starts with 11 as well as ends with 00 using Rij formula (May 2015)**

16. **Explain the DFA Minimization algorithm and Minimize the finite automaton show in figure below and show both the given and reduced one are equivalent. (May/June 2014)**



UNIT II
SIMPLE 4 MARK QUESTION WITH ANSWER

1. **What do you mean by null production and unit production? Give an example. (May/June 2016)**

   A production which is of the form $A \rightarrow \varepsilon$ is called null production

   A unit production is a production which is of the form $A \rightarrow B$, where A and B are variable

2. **Construct a CFG for the set of strings that contain equal number of a's and b's over ∑={a,b}. (May/June 2016)**

   Consider Grammar G = {V,T,P,S}

   V = (S,a,b) T = (S) P = {$S \rightarrow aSb$, $S \rightarrow bSa$, $S \rightarrow ab/ba$} S = {S}

3. **Define the term parse tree. (Nov/Dec 2015) (May/June 2012)**

   The parse tree for G = (V,T,P,S) are tree with following condition.

   1) Each interior node is labeled by a variable in V
   2) Each leaf is labeled by either a variable, a terminal or $\in$
   3) If interior node is labeled A, and its children are labeled as X1,X2, .. Xk, where $A \rightarrow$ X1 X2..Xk

   Example:

   $E \Rightarrow E * (E)$

4. **What is meant by ambiguity in grammars? (Nov/Dec 2015)(May/June 2013)**

   A grammar is said to be ambiguous if it has more than one derivation trees for a sentence or in other words if it has more than one leftmost derivation or more than one rightmost derivation.

5. **Define the term Chomsky Normal Form. (Nov/Dec 2015), (Nov/Dec 2012)**

   In formal language theory, a context-free grammar is in Chomsky Normal form (CNF) if the right-hand sides of all production rules start with a terminal symbol with following form:
   Eg:

   (1) $S \rightarrow BA$ (2) $S \rightarrow a$

   Here A,B,C are variable and a is a terminal. Further G has no useless symbols.

6. **Construct the context free grammar representing the set of palindromes over (0+1)*. (Nov/Dec 2015)(May/June 2014)**

   Consider Grammar G = {V,T,P,S}

   V = (S,$\in$0,1) T = (S) P = {$S \rightarrow 0 \mid 1 \mid \in$, $S \rightarrow 0S0$, $S \rightarrow 1S1$} S = {S}

7. **Let G be the grammar with**

   $S \rightarrow aB/bA$

   $A \rightarrow a/aS/bAA$

$B \rightarrow b/bS/aBB$

For the string aaabbabbba, find the leftmost derivation.(Nov/Dec 2015)

$S \rightarrow aB$

$S \rightarrow aaBB$

$S \rightarrow aaaBBB$

$S \rightarrow aaabBB$

$S \rightarrow aaabbB$

$S \rightarrow aaabbaBB$

$S \rightarrow aaabbabB$

$S \rightarrow aaabbabbS$

$S \rightarrow aaabbabbbA$

$S \rightarrow aaabbabbba$

8. What is a CFG (May/June 2013)

A Context Free Grammar G = (V,T,P,S) consists of vocabulary V, a subset T of V consisting of terminal elements, a start symbol S from V and a set of production P. Every production in P must contain atleast one non-terminal (N = V-T) on its left side.

9. What is meant by Greibach Normal Form (May/June 2013)

In formal language theory, a context-free grammar is in Greibach normal form (GNF) if the right-hand sides of all production rules start with a terminal symbol, optionally followed by some variables (Terminal or non-terminal).

GNF Form:

$A \rightarrow a\alpha$ where A is variable, a is terminal and α is string of variables

10. Write the CFG for the language $L = \{a^n b^n \mid n \geq 1\}$ (Nov/Dec 2013)

Consider Grammar G = {V,T,P,S}

V = (S,a,b) T = (S) P = {$S \rightarrow aSb$, $S \rightarrow ab$} S = {S}

11. What are useless symbols in a grammar ? (Nov/Dec 2007)

A Symbol X is useful for a grammar G = (V,T,P,S) if there is some derivation of the form

$S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w$, where w is in Terminal. If X is not useful we say it useless.

Ex:

G = (V,T,P,S) where V = (S,T,X) T = (0,1)

$S \rightarrow 0T \mid 1T \mid X \mid 0 \mid 1$ (RULE 1)

$T \rightarrow 00$ (RULE 2)

Here X is an useless symbol has it does not have any useful production

12. List out the Types of Grammar?

According to Noam Chomsky the types of grammar are

Type 0 – Unrestricted grammar

Type 1 – Context Sensitive grammar

Type 2 – Context free grammar

Type 3 – Regular Grammar

13. What is Sentential Forms ?

Derivation from the start symbol produced string are called “Sentential form”
Eg:

If $S \overset{*}{\Rightarrow} \alpha$ is a sentential form

14. What is Recursive inference

It is an approach for inferring whether the given string belong to the given CFG. Here the rules are formed from the body to head. This procedure is called as recursive inferences.

15. What is leftmost and rightmost derivatives

In order to restrict the number of choices we have in deriving a string, it is often useful to require that at each step we replace the leftmost variable by one of its production bodies. Such a derivation is called a leftmost derivation, and we indicate that a derivation is leftmost by using the relation $\underset{lm}{\Rightarrow}$. Similarly it is possible to require that at each step the rightmost variable is replaced by one of its bodies is called as rightmost derivation and it is represented by the symbol $\underset{rm}{\Rightarrow}$.

16. What is recursively enumerated language

The Type 0 unrestricted language is called as recursively enumerated language. It include all formal grammars. They generate exactly all language that can be recognized by Turing Machine.

17. If $S \rightarrow aSb \mid aAb$, $A \rightarrow bAa$, $A \rightarrow ba$ .Find out the CFL

soln.
$S \rightarrow aAb \Rightarrow abab$(sub $A \rightarrow ba$)
$S \rightarrow aSb \Rightarrow a\ aAb\ b \Rightarrow a\ aba\ b\ b$(sub $S \rightarrow aAb$& $A \rightarrow ba$)
$S \rightarrow aSb \Rightarrow a\ aSb\ b \Rightarrow a\ aaAb\ b\ b \Rightarrow a\ aaba\ b\ bb$ ($S \rightarrow aSb$, $S \rightarrow aAb$& $A \rightarrow ba$)
Thus $L=\{a^n b^m a^m b^n$, where $n,m \geq 1\}$

18. What are the applications of Context free languages?

Context free languages are used in :

- Defining programming languages.
- Formalizing the notion of parsing.
- Translation of programming languages.
- String processing applications.

PART B/C
<u>Theory questions</u>

1. Discuss the following:
(i)CFG and Parse trees

(ii) Ambiguity in Context Free Grammar with example (16)(Nov/Dec 2015)

2. **Let G =(V,T,P,S) be a context free grammar then prove that if the recursive inference procedure tells us that terminal string W is in the language of variable A, then there is a parse tree with root A and yield w. (10) (Nov/Dec 2015)**

3. **Explain the Type of Grammar in detail with proper examples (6)(Nov/Dec 2015)**
4. **What is an ambiguous grammar? Explain with an example. (6) (Nov/Dec 2015), (May/June 2016)**

Normal Form question

5. **Define the two normal forms that are to be converted from a context free grammar(CFG). Convert the following CFG to Chomsky normal form:**
   S→A/B/C
   A→aAa/B
   B→bB/bb
   C→baD/abD/aa
   D→aCaa/D (10)(May/June 2016)

6. **Construct the following grammar in CNF:**
   S→ ABC/BaB
   A→aA/BaC/aaa
   B →bBb/a/D
   C →CA/AC
   D→ Є (8)(Nov/Dec 2015)

7. **Construct a equivalent grammar G in CNF for the grammar G1 where**
   G1=({S,A,B},{a,b},{S →ASB/Є , A→aAS/a,B→SbS/A/bb},S) (10) (Nov/Dec 2015)

8. **Convert the following CFG G to Greibach normal form generating the same language**
   S→ABA
   A→aA/λ
   B→bB/λ (6) (May/June 2016)

9. **What is the purpose of normalization? Construct the CNF and GNF for the following grammar and explain the steps.**
   S→aAa/bBb/Є
   A→C/a
   B→C/b
   C→CDE/Є
   D→A/B/ab (10)(May/June 2016)
10. **Construct a reduced grammar equivalent to the grammar G=(N,T,S,P) where**
    N={S,A,C,D,E}
    T={a,b}
    P={ S→aAa,A→Sb/bCC/DaA,C→abb/DD,D→aDA,E→aC } (6) (May/June 2016)

11. **Convert the following grammar into CNF**

S → cBA
S → A
A → cB | AbbS
B → aaa (8)

12. Convert the following grammar into GNF
S → XY1 | 0
X → 00X | Y
Y → 1X1 (8)

13. Construct a equivalent grammar G in CNF for the grammar G1 where
G1 =({S,A,B},{a,b},{S->bA/aB,A->bAA/aS/a, B->aBB/bS/b},S) (8)

Ambiguity and Parse tree

14. Given the grammar G=(V,∑,R,E), where V={E,D,1,2,3,4,5,6,7,8,9,0,+,-,*,/,(,)},
∑={1,2,3,4,5,6,7,8,9,0,+,-,*,/,(,)}, and R contains the following rules:
E D/(E)/E+E/E-E/E*E/E/E
D 0/1/2/3/4/5/6/7/8/9
Find a parse tree for the string 1+2*3 (6) (Nov/Dec 2015)

15. Consider the following grammar:
E→ E+T/T
T →T*F/F
F →(E)/ id
(i) Give a rightmost derivation and leftmost derivation for the sentence w=id*(id+id)*id
(ii) Is the above grammar ambiguous? Justify.
(iii) Construct the parse tree for the sentence w=id*(id+id)*id(16)(Nov/Dec 2015)

16. Show the derivation steps and construct derivation tree for the string ababbb by using leftmost derivation with the grammar
S→AB/€
A→aB
B→Sb (5) (May/June 2016)

17. Construct a CFG for the regular expression (011+1)(01) (6) (May/June 2016)

18. Consider the following grammar for list structures:
S→ a | ^ | (T)
T → T,S | S
Find the left most derivation, right most derivation and parse tree for (((a,a),^(a)),a) (8)

19. Explain about Parse trees. For the following grammar
S →aB| bA
A→a|aS|bAA
B →b|bS|aBB
For the string aaabbabbba. Find left most derivation, right most derivation and parse tree (8)

20. If G is the grammar S →SbS | a show that G is ambiguous (5)