**UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA**

**DEPARTMENT OF COMPUTER SCIENCE**

**Formal Language & Automata Theory**

**Code: PCCCS402**

**Contacts: 3L**

**Credits: 3**

**Module-1:**

Introduction to concepts of alphabet, language, production rules, grammar and automaton, finite state model, concept of DFA and its problems, concept of NFA and its problems. NFA to DFA conversion, Construction of DFA & NFA for any given string and vice versa, Minimization of FA and equivalence of two FA, Mealy & moore machine and their problems. Limitations of FSM.

**Module-2:**

Introduction to the concept of Chomosky Classification of Grammar, language generation from production rules and vice- versa. regular language and regular expressions, identity rules. Arden’s theorem state and prove, Construction of NFA from regular expression, Conversion of NFA with null moves to without null moves, closure properties, pumping lemma and its applications.

**Module-3:**

Introduction to Context Free Grammer, Derivation trees, sentential forms. Right most and leftmost derivation of strings, concepts of ambiguity. Minimization of CFG, Chomsky normal form, Greibach normal form, Pumping Lemma for Context Free Languages, Enumeration of properties of CFL (proofs omitted). Closure property of CFL, Ogden’s lemma & its applications, Push Down Automata: Push down automata, definition and description, Acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence, Equivalence of CFL and PDA, interconversion, DCFL and DPDA.

**Module-4:**

Turing Machine : Turing Machine, definition, model, Design of TM, Computable functions, Church’s hypothesis, counter machine, Types of Turing machines (proofs not required), Universal Turing Machine, Halting problem, P, NP.

**UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA**

**DEPARTMENT OF COMPUTER SCIENCE**

Definitions, capability & state equivalent, kth- equivalent concept, Merger graph, Merger table, Compatibility graph, Finite memory definiteness, testing table & testing graph.

**TEXT BOOKS:**

1. “Theory of Computer Science “, Automata Languages and computation”, Mishra and Chandrashekaran, 2nd edition, PHI.
2. "An Introduction to Formal Languages and Automata" , Peter Linz.
3. “Formal Languages and Automata Theory”, C.K.Nagpal, Oxford.

**REFERENCES:**

1. “Switching & Finite Automata”, ZVI Kohavi, 2nd Edn., Tata McGraw Hill.
2. “Introduction to Automata Theory Language and Computation”, Hopcroft H.E. and Ullman J. D., Pearson Education.



**UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Paper Name:** |  | **Formal Language & Automata Theory** |  |  |
| **Paper Code:** |  | **PCCCS402** |  |  |
| **Contact Hours:** |  | **3** |  |  |
| **Credit Point:** |  | **3** |  |  |
|  |  |  |  |  |
| **Course** |  | Understanding of theory of computation, grammars and basics |  |  |
| **Educational** |  |  |  |
|  | of compiler design. |  |  |
| **Objective:** |  |  |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  | Elementary discrete mathematics including the notion of |  |  |
|  |  | set,function,relation,product,partial order,equivalence |  |  |
|  |  | relation,graph& tree. They should have a thorough |  |  |
| **Prerequisites:** |  | understanding of the principle of mathematical induction. |  |  |
|  |  |  |  |  |
|  |  | After studying Finite Automata The student will be able to |  |  |
|  | **CO1** | define a system and recognize the behavior of a system. They |  |  |
| **Course** | will be able to minimize a system and compare different |  |  |
|  |  |  |
| **Outcome:** |  | systems. |  |  |
|  |  | After studying regular language and grammer Student will |  |  |
|  | **CO2** | convert Finite Automata to regular expression. Students will |  |  |
|  | be able to check equivalence between regular linear grammar |  |  |
|  |  |  |  |
|  |  | and FA. |  |  |
|  |  | After studying CFG and PDA Students will be able to minimize |  |  |
|  | **CO3** | context free grammar. Student will be able to check |  |  |
|  | equivalence of CFL and PDA. They will be able to design |  |  |
|  |  |  |  |
|  |  | Turing Machine. |  |  |
|  | **CO4** | After studying turing machine Students will be able to design |  |  |
|  | Turing machine. |  |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  | **Detailed Planning:** |  |  |
|  |  |  |  |  |
| **Module** | **Class** | **Topic** | **CO Mapping** |  |
| **Sequence** |  |
|  |  |  |  |
|  | 1st Class | Introduction to concepts of alphabet,language,production |  |  |
| **1** | rules,grammer and automaton. | CO1 |  |
|  |  |
|  | 2nd Class | Introduction to finite state model, concept of DFA and its |  |  |
| **1** | problems. | CO1 |  |
|  |  |
|  | 3rd Class | Revision of DFA and introduction of NFA and its problems. |  |  |
| **1** | NFA to DFA conversion | CO1 |  |
|  |  |
|  | 4th Class | Construction of DFA & NFA for any given string and vice |  |  |
| **1** | versa. | CO1 |  |
|  |  |
| **1** | 5th Class | Minimization of FA and equivalence of two FA. | CO1 |  |
| **1** | 6th Class | Introduction to mealy & moore machine and their problems. | CO1 |  |
|  |  |
|  | 7th Class | Introduction to the concept of Chomosky Classification of |  |  |
| **2** | Grammer. | CO2 |  |
|  |  |
|  | 8th Class | Introduction to language generation from production rules and |  |  |
| **2** | vice versa. | CO2 |  |
|  |  |
| **2** | 9th Class | Introduction to regular language and regular expressions. | CO2 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2** | 10th Class | Discussion of identity rules and corresponding problems. | CO2 |  |
|  | 11th Class | Discussion of Arden's theorem and proof of arden's theorem |  |  |
| **2** | and its applications. | CO2 |  |
|  |  |
| **2** | 12th Class | Construction of NFA from regular expression. | CO2 |  |
| **2** | 13th Class | Conversion of NFA with null moves to without null moves. | CO2 |  |
| **2** | 14th Class | Discussion of closure properties. | CO2 |  |
| **2** | 15th Class | Discussion of pumping lemma and its applications. | CO2 |  |
|  |  | Introduction to Context Free Grammer, Derivation trees, |  |  |
|  | 16th Class | sentential forms. Right most and leftmost derivation of strings, |  |  |
| **3** |  | concepts of ambiguity. | CO3 |  |
| **3** | 17th Class | Minimization of CFG. | CO3 |  |
| **3** | 18th Class | Discussion of Chomsky normal form and its problems. | CO3 |  |
| **3** | 19th Class | Discussion of Greibach normal form and its problems. | CO3 |  |
| **3** | 20th Class | Pumping Lemma for Context Free Languages. | CO3 |  |
|  | 21st Class | Enumeration of properties of CFL (proofs omitted). Closure |  |  |
| **3** |  | property of CFL, Ogden’s lemma & its applications | CO3 |  |
|  | 22nd Class | Push Down Automata: Push down automata, definition and |  |  |
| **3** | description. | CO3 |  |
|  | 23rd Class | Acceptance of CFL, Acceptance by final state and acceptance |  |  |
| **3** | by empty state and its equivalence. | CO3 |  |
|  |  |
| **3** | 24th Class | Equivalence of CFL and PDA, interconversion. | CO3 |  |
| **3** | 25th Class | Introduction to DCFL and DPDA. | CO3 |  |
|  | 26th Class | Turing Machine : Turing Machine, definition, model, Design of |  |  |
| **4** | TM, Computable functions. | CO4 |  |
|  |  |
|  | 27th Class | Church’s hypothesis, counter machine, Types of Turing |  |  |
| **4** | machines . | CO4 |  |
|  |  |
|  | 28th Class | Universal Turing Machine, Halting problem. Concept of P, NP |  |  |
| **4** | class. | CO4 |  |
|  |  |
|  |  |  |  |  |

**Text Books:**

**1**

**2**

**3**

"An Introduction to Formal Languages and Automata" , Peter Linz

“Formal Languages and Automata Theory”, C.K.Nagpal, Oxford

“Theory of Computer Science “, Automata Languages and computation”, Mishra and Chandrashekaran, 2nd edition, PHI.

**Reference**

**Books and** **1**

**Journals:**

**2**

“Switching & Finite Automata”, ZVI Kohavi, 2nd Edn., Tata McGraw Hill

“Introduction to Automata Theory Language and Computation”, Hopcroft H.E. and Ullman J. D., Pearson Education.



**University of Engineering & Management, Kolkata**

**Workbook, Even Semester, 2020**

**Course: B.Tech (CSE)** **Semester: 4th**

**Paper Name: Formal Language and Automata Theory**

**Paper Code: PCCCS402**

**Finite Automata**

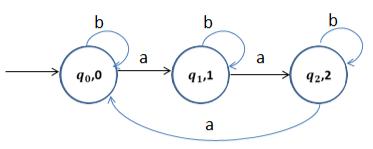
1. Construct a DFA that accept all strings over the alphabets {a, b} where the string length is divisible by two.
2. Construct a DFA that accept all strings over the alphabets {a, b} where the string length is not divisible by two.
3. Construct a DFA where || ≅ 1 3
4. Construct a DFA that accept all strings over the alphabets {a, b} where number of ‘a’ in the string is two.
5. Construct a DFA that accept all strings over the alphabets {a, b} where number of ‘a’ in the string at most two.
6. Construct a DFA that accept all strings over the alphabets {a, b} where number of ‘a’ in the string is even.
7. Construct a DFA that accept all strings over the alphabets {a, b} where number of ‘a’ in the string is divisible by 3.
8. Construct a DFA that accept all strings over the alphabets {a, b} where number of ‘b’ in the string is divisible by 5.

9. Construct a DFA that accept all strings over the alphabets {a, b} such that ≅ 1 3

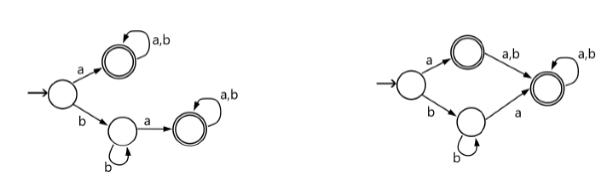
1. Construct a DFA that accept all strings over the alphabets {a, b} where number of ‘a’ and number of ‘b’ in the string is even.
2. Construct a DFA that accept all strings over the alphabets {a, b} where number of ‘a’ is divisible by 3 and number of ‘b’ in the string is divisible by 2.
3. Construct a minimal DFA which accepts all strings over {0,1}, which when interpreted as binary number is ≅ 2 3.
4. Construct a minimal DFA which accepts all strings over {0,1,2}, which when interpreted as ternary number is divisible by 4.
5. Construct a minimal DFA which accepts set of all strings over {a, b} where each string starts with an ‘a’.

1. Construct a minimal DFA which accepts set of all strings over {a, b} where each string contains an ‘a’.
2. Construct a minimal DFA which accepts set of all strings over {a, b} where each string ends with an ‘a’.
3. Show that =|is not regular.
4. Show that =| ≥ 1is not regular.
5. Construct Finite Automata equivalent the Regular Expression+ + ∗.
6. Construct a grammar for the language =, , ≥ 0.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 21. | Construct a grammar for the language = | 0$1,where | ≠ ,, | ≥ 1 . |
| 22. | Construct a grammar for the language = | 0$1,where | ≠ ,, | ≥ 1 . |



23. Check two given DFA are equivalent or not:



**Push Down Automata**

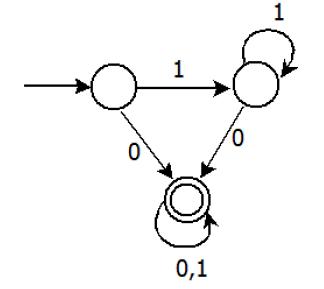
1. Write down the procedure or rule to construct the grammar G from PDA.
2. Construct the grammar G from the PDA A that accepting the language =$| , ≥ 1 by null store.
3. Construct the grammar G from the PDA A that accepting the language =&| ≥ 1 by null store.
4. Construct the PDA A equivalent to the following grammar CFG: ' → '⁄.
5. Construct the PDA A equivalent to the following grammar CFG: ' → \*++⁄\*\* ; \* → ++ ∕ ;+ →++∕\*
6. Show that = .&| ≥ 1 /is not CFL.
7. Design the PDA for =| ≥ 1 that accepting by final state.
8. Design the PDA for = 001|0 ∈ ,∗ that accepting by final state.

1. What do you mean by useless symbols in CFG? Eliminate the useless symbols from the following grammar ' → \*+ ⁄ ; \* → .
2. Given the grammar ' → \*+; \* → ; + → 3⁄ ; 3 → 4; 5 → 5; 5 → , find an equivalent grammar which is reduced and has no unit production.
3. Reduce the following grammar G into Chomsky Normal Form (CNF). ' → \*4; \* → +⁄\*+;+ → ;4 → .
4. Convert the following grammar G into its equivalent GNF. ' → '⁄.
5. Convert the following grammar G into its equivalent GNF. ' → \*+⁄ ; \* → \*⁄+ ; + → \*.

**Turing Machine**

1. a) Write an unrestricted grammar to accept the language =678| = 9 : = ; . Mention the start symbol of your grammar. Use upper case Roman letters for non-terminal symbols.b) Show a derivation of the string &<&< according to your grammar.
2. Design a Turing Machine to find 2’s complement of a given binary number.

**Gate questions for your Practice**



1.

Consider the DFA given. gatecs201313 Which of the following are FALSE?

1. Complement of L(A) is context-free.
2. L(A) = L((11\*0+0)(0 + 1)\*0\*1\*)
3. For the language accepted by A, A is the minimal DFA.
4. A accepts all strings over {0, 1} of length at least 2.

A 1 and 3 only

B 2 and 4 only

C 2 and 3 only

D 3 and 4 only

1. 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

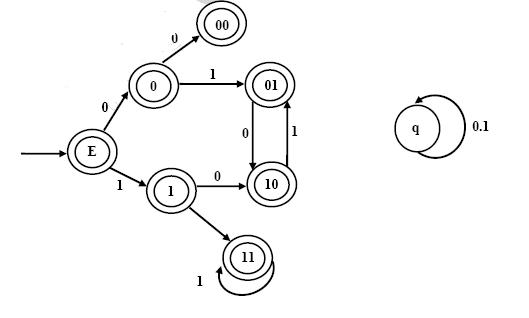
C.1, 2 and 4

D.1, 3 and 4

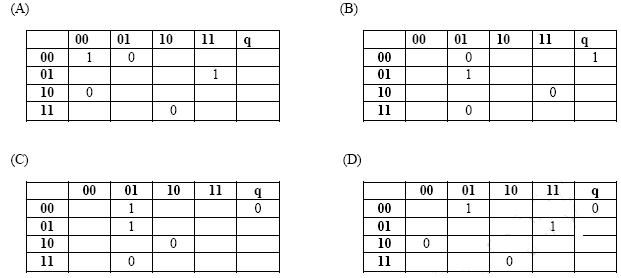
1. Given the language L = {ab, aa, baa}, which of the following strings are in L\*?
   1. abaabaaabaa
   2. aaaabaaaa
   3. baaaaabaaaab

4) baaaaabaa A.1, 2 and 3 B.2, 3 and 4 C.1, 2 and 4 D.1, 3 and 4

1. Consider the set of strings on {0,1} in which, every substring of 3 symbols has at most two zeros. For example, 001110 and 011001 are in the language, but 100010 is not. All strings of length less than 3 are also in the language. A partially completed DFA that accepts this language is shown below.



The missing arcs in the DFA are:



5. Definition of a language L with alphabet {a} is given as following.

L={a^(nk)| k>0, and n is a positive integer constant}

What is the minimum number of states needed in DFA to recognize L?

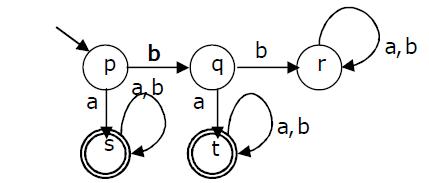
A.k+1

B.n+1

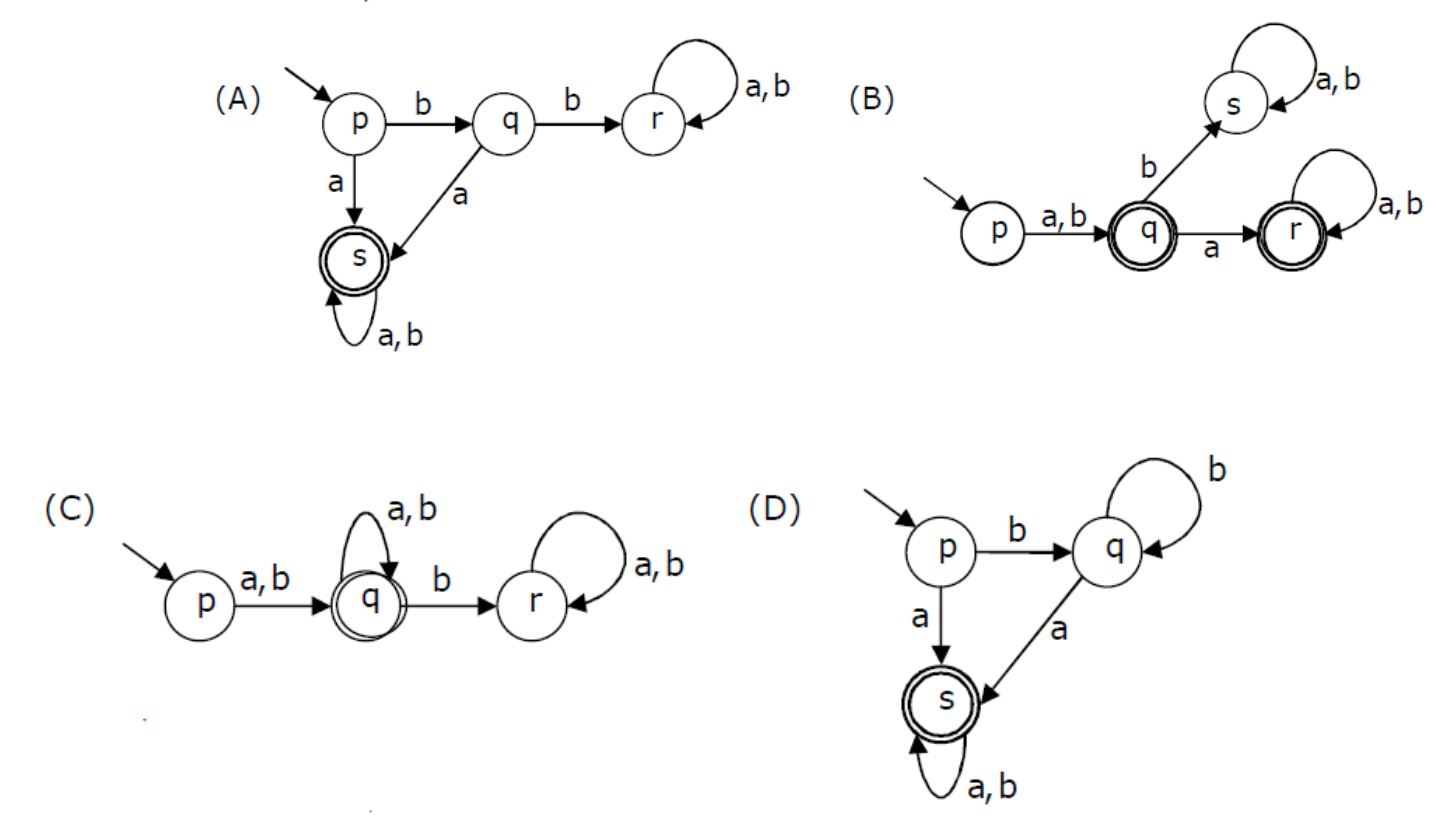
C.2^(n+1)

D.2^(k+1)

6. A deterministic finite automation (DFA)D with alphabet {a,b} is given below



Which of the following finite state machines is a valid minimal DFA which accepts the same language as D?



1. Let w be any string of length n is {0,1}\*. Let L be the set of all substrings of w. What is the minimum number of states in a non-deterministic finite automaton that accepts L?

A.n-1

B.n

C.n+1

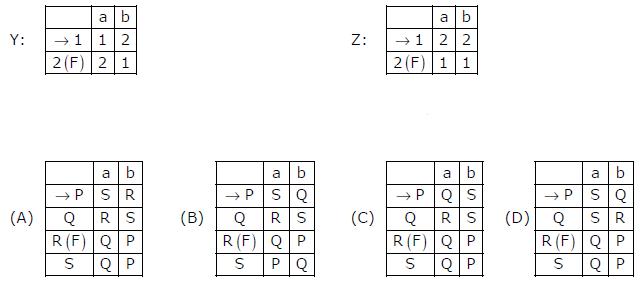
D.2n-1

1. Which one of the following languages over the alphabet {0,1} is described by the regular expression: (0+1)\*0(0+1)\*0(0+1)\*?

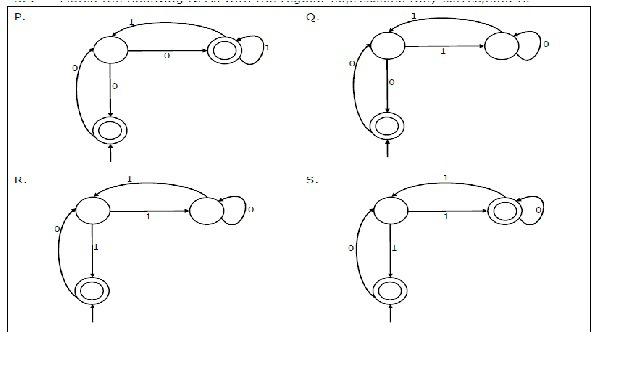
A.The set of all strings containing the substring 00. B.The set of all strings containing at most two 0’s. C.The set of all strings containing at least two 0’s.

D.The set of all strings that begin and end with either 0 or 1.

1. Given below are two finite state automata (→ indicates the start state and F indicates a final state)Which of the following represents the product automaton Z×Y?



1. Match the following NFAs with the regular expressions they correspond to 1. ϵ + 0(01\*1 + 00) \* 01\*
2. ϵ+0(10\*1+00)\*0
3. ϵ + 0(10 \*1 + 10) \*1
4. ϵ + 0(10 \*1 + 10) \*10 \*



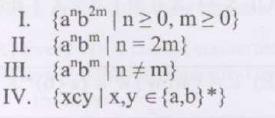
A.P-2,Q-1,R-3,S-4

B.P-1,Q-3,R-2,S-4

C.P-1,Q-2,R-3,S-4

D.P-3,Q-2,R-1,S–4

11. Which of the following are regular sets?



A.I and IV only

B.I and III only

C.I only

D.IV only

1. A minimum state deterministic finite automaton accepting the language L={w | w ε {0,1} \*, number of 0s and 1s in w are divisible by 3 and 5, respectively} has

A.15 states

B.11 states

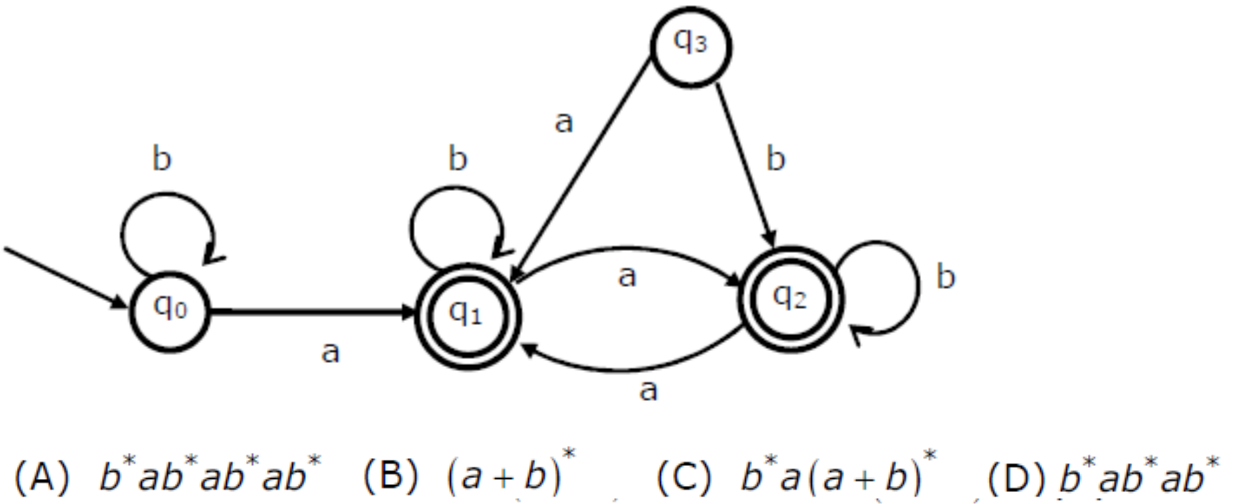
C.10 states

D.9 states

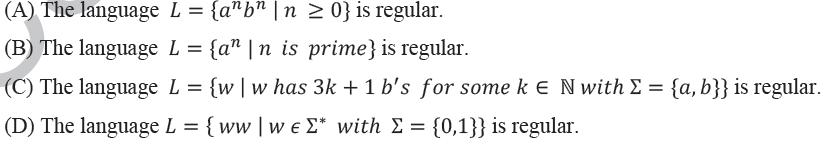
13. Which of the following languages is regular?



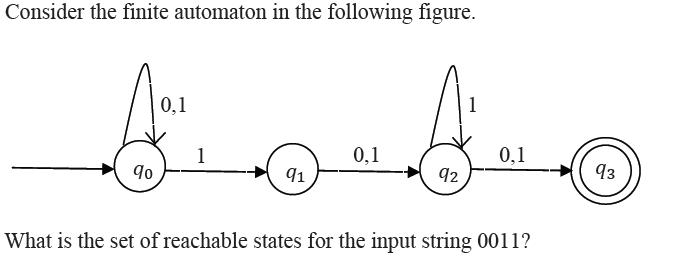
1. Consider the following Finite State Automaton. The language accepted by this automaton is given by the regular expression



15. Which one of the following is TRUE?



**16.**



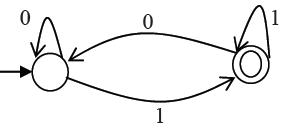
A.{q0, q1, q2}

B.{q0, q1}

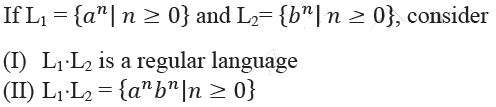
C.{q0, q1, q2, q3}

D.{q3}

17. Which of the regular expressions given below represent the following DFA?



1. 0\*1(1+00\*1)\*
2. 0\*1\*1+11\*0\*1
3. (0+1)\*1 A.I and II only B.I and III only C.II and III only D.I, II, and III



18.

Which one of the following is CORRECT?

A.Only (I)

B.Only (II)

C.Both (I) and (II)

D.Neither (I) nor (II)

1. Let L1 = {w ∈ {0,1}∗ | w has at least as many occurrences of (110)’s as (011)’s}.

Let L2 = { ∈ {0,1}∗ | w has at least as many occurrences of (000)’s as (111)’s}.

Which one of the following is TRUE?

A.L1 is regular but not L2

B.L2 is regular but not L!

C.Both L2 and L1 are regular

D.Neither L1 nor L2 are regular

1. The length of the shortest string NOT in the language (over Σ = {a, b}) of the following regular expression is \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

a\*b\*(ba)\*a\*

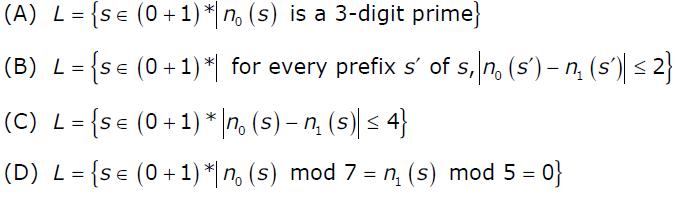
A.2

B.3

C.4

D.5

1. If s is a string over (0 + 1)\* then let n0(s) denote the number of 0’s in s and n1(s) the number of 1’s in s. Which one of the following languages is not regular?



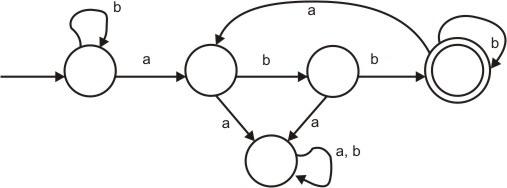
1. Consider the regular language L = (111 + 11111)\*. The minimum number of states in any DFA accepting this languages is:

A.3

B.5

C.8

D.9



23.

Consider the machine M: GATECS2005Q53 The language recognized by M is :

A.{w ∈ {a, b}\* / every a in w is followed by ex-actly two b's}

B.{w ∈ {a, b}\* every a in w is followed by at least two b’}

C.{w ∈ {a, b}\* w contains the substring 'abb'}

D.{w ∈ {a, b}\* w does not contain 'aa' as a substring}

1. Let Nf and Np denote the classes of languages accepted by non-deterministic finite automata and n on-deterministic push-down automata, respectiv ely. Let Df and Dp denote the classes of languages accepted by deterministic finite a utomata and deterministic push-do wn automata, respectively. Which one of the following is TRUE?

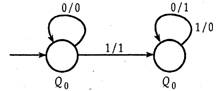
A.Df ⊂ Nf and Dp ⊂ Np

B.Df ⊂ Nf and Dp = Np

C.Df = Nf and Dp = Np

D.Df = Nf and Dp ⊂ Np

25. The following dia gram represents a finite state machine which ta kes as input a



binary number from the least significant bit. Which one of the following is TRUE?

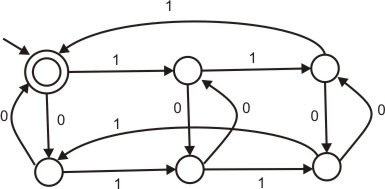
A.It computes 1's commplement of the input number

B.It computes 2's com plement of the input number

C.It increments the in put number

D.It decrements the i nput number

1. The following finite state machine accepts all those binary strings in which the number of 1's and 0's are respectively.



A.divisible by 3 and 2

B.odd and even

C.even and odd

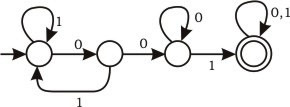
D.divisible by 2 and 3

1. The regular expression 0\*(10\*)\* denotes the same set as A.(1\*0)\*1\*

B.0 + (0 + 10)\*

C.(0 + 1)\* 10(0 + 1)\* D.none of these

1. Consider the follo wing deterministic finite state automaton M.



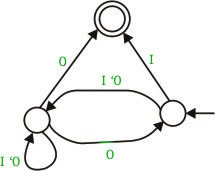
Let S denote the set o f seven bit binary strings in which the first, the fourth, and the last bits are 1. The number of strings in S that are accepted by M is

A.1

B.5

C.7

D.8



29.

Consider the NFA M shown below. GATECS2003Q55 Let the language accepted by M be L. Let L1 be th e language accepted by the NFA M1, obtained b y changing the

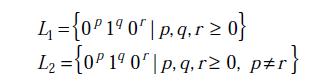
accepting state of M to a non-accepting state and by changing the non-accepting state of M to accepting states. Which of the following statements is true ?

A.L1 = {0, 1}\* - L

B.L1 = {0, 1}\*

C.L1 ⊆ L

D.L1 = L



30.

Consider the following languages. Which one of the following statements is FALSE?

A.L2 is context-free.

B.L1 intersection L2 is context-free.

C.Complement of L2 is recursive.

D.Complement of L1 is context-free but not regular

1. Which of the following pairs have DIFFERENT expressive power? A.Deterministic finite automata(DFA) and Non-deterministic finite automata(NFA)

B.Deterministic push down automata(DPDA)and Non-deterministic push down automata(NPDA)

C.Deterministic single-tape Turing machine and Non-deterministic single-tape Turing machine

D.Single-tape Turing machine and multi-tape Turing machine

1. Consider the language L1,L2,L3 as given below. L1={0^{p}1^{q} | p,q € N} L2={0^{p}1^{q} | p,q € N and p=q} L3={0^{p}1^{q}0^{r} | p,q,r € N and p=q=r} Which of the following statements is NOT TRUE?

A.Push Down Automata (PDA) can be used to recognize L1 and L2

B.L1 is a regular language

C.All the three languages are context free

D.Turing machine can be used to recognize all the three languages

1. Consider the languages L1 = {0^i1^j | i != j}. L2 = {0^i1^j | i = j}. L3 = {0^i1^j | i = 2j+1}. L4 = {0i1j | i != 2j}.
2. S -> aSa|bSb|a|b; The language generated by the above grammar over the alphabet {a,b} is the set of
3. Let L = L1∩L2, where L1 and L2 are languages as defined below:

L1 = {a^mb^mca^nb^n | m, n >= 0 } L2 = {a^ib^jc^k | i, j, k >= 0 } Then L is

A.Not recursive B.Regular

C.Context free but not regular

D.Recursively enumerable but not context free.

1. The language L= {0^i21^i | i≥0 } over the alphabet {0,1, 2} is: A.not recursive

B.is recursive and is a deterministic CFL.

C.is a regular language.

D.is not a deterministic CFL but a CFL.

1. Consider the CFG with {S,A,B) as the non-terminal alphabet, {a,b) as the terminal alphabet, S as the start symbol and the following set of production rules

|  |  |  |  |
| --- | --- | --- | --- |
| S | --> aB | S -- | > bA |
| B | --> b | A -- | > a |
| B | --> bS | A -- | > aS |
| B | --> aBB | A | --> bAA |

Which of the following strings is generated by the grammar?

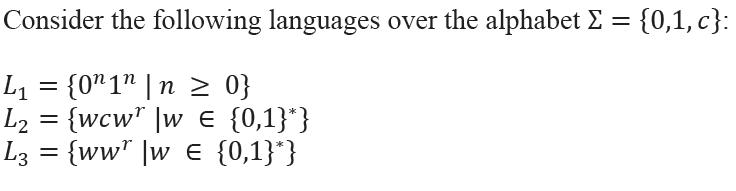
A.aaaabb

B.aabbbb

C.aabbab

D.abbbba

38.



Here, w^r is the reverse of the string w. Which of these languages are deterministic Context-free languages?

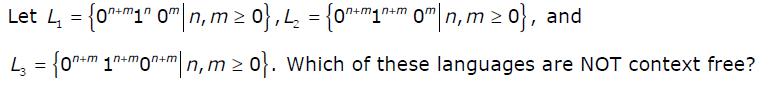
A.None of the languages

B.Only L1

C.Only L1 and L2

D.All the three languages

39.



A.L1 only

B.L3 Only

C.L1 and L2

D.L2 and L3

* 1. Consider the following statements about the context free grammar G = {S → SS, S → ab, S → ba, S → Ε}

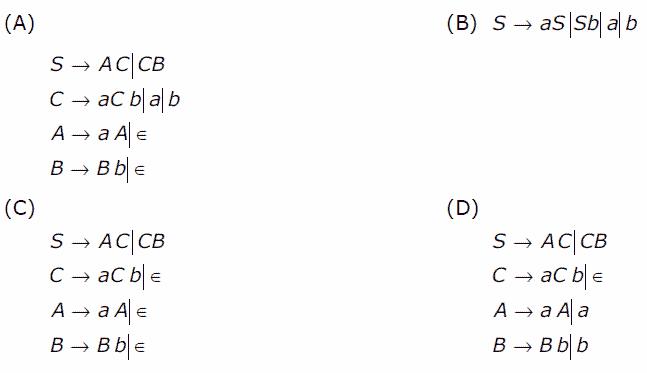
I. G is ambiguous

II. G produces all strings with equal number of a’s and b’s III. G can be accepted by a deterministic PDA.

Which combination below expresses all the true statements about G? A.I only

B.I and III only C.II and III only D.I, II and III

1. Which one of the following grammars generates the language L = {aibj | i ≠ j}



1. In the correct grammar of above question, what is the length of the derivation (number of steps starring from S) to generate the string a^lb^m with l ≠ m?
2. Consider the languages:

L1 = {a^nb^nc^m | n, m > 0}

L2 = {a^nb^mc^m | n, m > 0}

Which one of the following statements is FALSE? A.L1 ∩ L2 is a context-free language

B.L1 U L2 is a context-free language C.L1 and L2 are context-free language D.L1 ∩ L2 is a context sensitive language

1. Consider the languages:

L1 = {ww^R |w ∈ {0, 1}\*}

L2 = {w#w^R | w ∈ {0, 1}\*}, where # is a special symbol

L3 = {ww | w ∈ (0, 1}\*)

Which one of the following is TRUE?

A.L1 is a deterministic CFL

B.L2 is a deterministic CFL

C.L3 is a CFL, but not a deterministic CFL

D.L3 is a deterministic CFL

1. The language {a^m b^n C^(m+n) | m, n ≥ 1} is A.regular

B.context-free but not regular C.context sensitive but not context free D.type-0 but not context sensitive

46**.** If the strings of a language L can be effectively enumerated in lexicographic (i.e., alphabetic) order, which of the following statements is true ?

A.L is necessarily finite

B.L is regular but not necessarily finite

C.L is context free but not necessarily regular

D.L is recursive but not necessarily context free

1. Let G = ({S}, {a, b} R, S) be a context free grammar where the rule set R is S → a S b | SS | ε Which of the following statements is true?

A.G is not ambiguous

B.There exist x, y, ∈ L (G) such that xy ∉ L(G)

C.There is a deterministic pushdown automaton that accepts L(G)

D.We can find a deterministic finite state automaton that accepts L(G)

1. Which of the following statements is/are FALSE?
2. For every non-deterministic Turing machine,

there exists an equivalent deterministic Turing machine.

1. Turing recognizable languages are closed under union and complementation.
2. Turing decidable languages are closed under intersection and complementation.
3. Turing recognizable languages are closed under union and intersection.

A.1 and 4 only

B.1 and 3 only

C.2 only

D.3 only

1. Let L1 be a recursive language. Let L2 and L3 be languages that are recursively enumerable but not recursive. Which of the following statements is not necessarily true?
2. L2 – L1 is recursively enumerable.
   1. L1 – L3 is recursively enumerable
   2. L2 ∩ L1 is recursively enumerable
   3. L2 ∪ L1 is recursively enumerable

50. 

Which of the following is true for the language 16

A.It is not accepted by a Turing Machine

B.It is regular but not context-free

C.It is context-free but not regular

D.It is neither regular nor context-free, but accepted by a Turing machine

1. If L and L' are recursively enumerable, then L is A.regular

B.context-free C.context-sensitive D.recursive

1. Let L be a language and L' be its complement. Which one of the following is NOT a viable possibility?

A.Neither L nor L' is recursively enumerable (r.e.).

B.One of L and L' is r.e. but not recursive; the other is not r.e.

C.Both L and L' are r.e. but not recursive.

D.Both L and L' are recursive

1. Let A ≤m B denotes that language A is mapping reducible (also known as many-to-one reducible) to language B. Which one of the following is FALSE? a) If A ≤m B and B is recursive then A is recursive. b) If A ≤m B and A is undecidable then B is undecidable. c) If A ≤m B and B is recursively enumerable then A is recursively enumerable. d) If A ≤m B and B is not recursively enumerable then A is not recursively enumerable.

A.a

B.b

C.c

D.d

1. For S ∈ (0 + 1) \* let d(s) denote the decimal value of s (e.g. d(101) = 5). Let L = {s ∈ (0 + 1)\* d(s)mod5 = 2 and d(s)mod7 != 4}. Which one of the following statements is true?

A.L is recursively enumerable, but not recursive

B.L is recursive, but not context-free

C.L is context-free, but not regular

D.L is regular

1. Let L1 be a recursive language, and let L2 be a recursively enumerable but not a recursive language. Which one of the following is TRUE?

L1' --> Complement of L1

L2' --> Complement of L2

A.L1' is recursive and L2' is recursively enumer-able

B.L1' is recursive and L2' is not recursively enumerable

C.L1' and L2' are recursively enumerable

D.L1' is recursively enumerable and L2' is recursive

1. L1 is a recursively enumerable language over Σ. An algorithm A effectively enumerates its words as w1, w2, w3, ... Define another language L2 over Σ Union {#} as {w^i # w^j : wi, wj ∈ L1, i < j}. Here # is a new symbol. Consider the following assertions.

S1 : L1 is recursive implies L2 is recursive

S2 : L2 is recursive implies L1 is recursive

Which of the following statements is true ?

A.Both S1 and S2 are true

B.S1 is true but S2 is not necessarily true

C.S2 is true but S1 is not necessarily true

D.Neither is necessarily true

1. Nobody knows yet if P = NP. Consider the language L defined as follows : GATECS2003Q13 Which of the following statements is true ?

A.L is recursive

B.L is recursively enumerable but not recursive

C.L is not recursively enumerable

D.Whether L is recursive or not will be known after we find out if P = NP

1. A single tape Turing Machine M has two states q0 and q1, of which q0 is the starting state. The tape alphabet of M is {0, 1, B} and its input alphabet is {0, 1}. The symbol B is the blank

symbol used to indicate end of an input string. The transition function of M is described in the following table

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | B |
| q0 | q1, 1, R | q1, 1, R | Halt |
| q1 | q1, 1, R | q0, 1, L | q0, B, L |

The table is interpreted as illustrated below. The entry (q1, 1, R) in row q0 and column 1 signifies that if M is in state q0 and reads 1 on the current tape square, then it writes 1 on the same tape square, moves its tape head one position to the right and transitions to state q1. Which of the following statements is true about M ?

A.M does not halt on any string in (0 + 1)+

B.M does not halt on any string in (00 + 1)\*

C.M halts on all string ending in a 0

D.M halts on all string ending in a 1

1. Define languages L0 and L1 as follows :

L0 = {< M, w, 0 > | M halts on w}

L1 = {< M, w, 1 > | M does not halts on w}

Here < M, w, i > is a triplet, whose first component. M is an encoding of a Turing Machine, second component, w, is a string, and third component, i, is a bit. Let L = L0 ∪ L1. Which of the following is true ?

A.L is recursively enumerable, but L' is not

B.L' is recursively enumerable, but L is not

C.Both L and L' are recursive

D.Neither L nor L' is recursively enumerable

1. For any two languages L1 and L2 such that L1 is context free and L2 is recursively enumerable but not recursive, which of the following is/are necessarily true?
2. L1' (complement of L1) is recursive
3. L2' (complement of L2) is recursive
4. L1' is context-free
5. L1' ∪ L2 is recursively enumerable A.1 only

B.3 only

C.3 and 4 only

D.1 and 4 only