CST 301	FORMAL LANGUAGES AND AUTOMATA THEORY	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble: This is a core course in theoretical computer science. It covers automata and grammar representations for languages in Chomsky Hierarchy. For regular languages, it also covers representations using regular expression and Myhill-Nerode Relation. The topics covered in this course have applications in various domains including compiler design, decidability and complexity theory, software testing, formal modelling and verification of hardware and software.

Prerequisite: Basic knowledge about the following topic is assumed: sets, relations - equivalence relations, functions, proof by Principle of Mathematical Induction.

Course Outcomes: After the completion of the course the student will be able to

CO1	Classify a given formal language into Regular, Context-Free, Context Sensitive, Recursive or Recursively Enumerable. **[Cognitive knowledge level: Understand]**
CO2	Explain a formal representation of a given regular language as a finite state automaton, regular grammar, regular expression and Myhill-Nerode relation. [**Cognitive knowledge level: Understand**]
CO3	Design a Pushdown Automaton and a Context-Free Grammar for a given context-free language. [**Cognitive knowledge level** : **Apply**]
CO4	Design Turing machines as language acceptors or transducers. [**Cognitive knowledge level: Apply]**
CO5	Explain the notion of decidability. [**Cognitive knowledge level: Understand]**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✔	✔	✔									✔

CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test 1 (Marks)	**Test 2 (Marks)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**
Continuous Assessment - Test : **25 marks**
Continuous Assessment - Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

CST 301 Formal Languages and Automata Theory

Module - 1 (Introduction to Formal Language Theory and Regular Languages)

Introduction to formal language theory– Alphabets, Strings, Concatenation of strings, Languages.

Regular Languages - Deterministic Finite State Automata (DFA) (Proof of correctness of construction not required), Nondeterministic Finite State Automata (NFA), Equivalence of DFA and NFA**,** Regular Grammar (RG), Equivalence of RGs and DFA.

Module - 2 (More on Regular Languages)

Regular Expression (RE), Equivalence of REs and DFA, Homomorphisms, Necessary conditions for regular languages, Closure Properties of Regular Languages, DFA state minimization (No proof required).

Module - 3 (Myhill-Nerode Relations and Context Free Grammars)

Myhill-Nerode Relations (MNR)- MNR for regular languages, Myhill-Nerode Theorem (MNT) (No proof required), Applications of MNT.

Context Free Grammar (CFG)- CFG representation of Context Free Languages (proof of correctness is required), derivation trees and ambiguity, Normal forms for CFGs.

Module - 4 (More on Context-Free Languages)

Nondeterministic Pushdown Automata (PDA), Deterministic Pushdown Automata (DPDA), Equivalence of PDAs and CFGs (Proof not required), Pumping Lemma for Context-Free Languages (Proof not required), Closure Properties of Context Free Languages.

Module - 5 (Context Sensitive Languages, Turing Machines)

Context Sensitive Languages **-** Context Sensitive Grammar (CSG), Linear Bounded Automata.

Turing Machines - Standard Turing Machine, Robustness of Turing Machine, Universal Turing Machine, Halting Problem, Recursive and Recursively Enumerable Languages.

Chomsky classification of formal languages.

Text Book

1. Dexter C. Kozen, Automata and Computability, Springer (1999)

Reference Materials

1. John E Hopcroft, Rajeev Motwani and Jeffrey D Ullman, Introduction to Automata Theory, Languages, and Computation, 3/e, Pearson Education, 2007
2. Michael Sipser, Introduction To Theory of Computation, Cengage Publishers, 2013.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Identify the class of the following languages in Chomsky Hierarchy:

- $L_1 = \{a^p | p \text{ is a prime number}\}$
- $L_2 = \{x\{0,1\}^* | x \text{ is the binary representation of a decimal number which is a multiple of } 5\}$
- $L_3 = \{a^n b^n c^n | n \geq 0\}$
- $L_4 = \{a^m b^n c^{m+n} | m > 0, n \geq 0\}$
- $L_5 = \{M\#x | M \text{ halts on } x\}$. Here, M is a binary encoding of a Turing Machine and x is a binary input to the Turing Machine.

Course Outcome 2 (CO2):

(i) Design a DFA for the language $L = \{axb | x \in \{a,b\}^*\}$

(ii) Write a Regular Expression for the language: $L = \{x \in \{a,b\}^* | \text{third last symbol in } x \text{ is } b\}$

(iii) Write a Regular Grammar for the language: $L = \{x \in \{0,1\}^* | \text{there are no consecutive zeros in } x\}$

(iv) Show the equivalence classes of the canonical Myhill-Nerode relation induced by the language: $L = \{x \in \{a,b\}^* | x \text{ contains even number of } a\text{'s and odd number of } b\text{'s}\}$.

Course Outcome 3 (CO3):

(i) Design a PDA for the language $L = \{ww^R | w \in \{a,b\}^*\}$. Here, the notation w^R represents the reverse of the string w.

(ii) Write a Context-Free Grammar for the language $L = \{a^n b^{2n} | n \geq 0\}$.

Course Outcome 4 (CO4):

(i) Design a Turing Machine for the language $L = \{a^n b^n c^n | n \geq 0\}$

(ii) Design a Turing Machine to compute the square of a natural number. Assume that the input is provided in unary representation.

Course Outcome 5 (CO5): Argue that it is undecidable to check whether a Turing Machine M enters a given state during the computation of a given input x.

Model Question paper

QP CODE: **PAGES:3**

Reg No:______________ **Name :______________**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST301

Course Name: Formal Languages and Automata Theory

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Design a DFA for the language $L = \{x \in \{a, b\}^* | aba\ is\ not\ a\ substring\ in\ x\}$.

2. Write a Regular Grammar for the language: $L = \{axb | x \in \{a, b\}^*\}$

3. Write a Regular Expression for the language: $L = \{x \in \{0,1\}^* | there\ are\ no\ consecutive\ 1's\ in\ x\}$

4. Prove that the language $L_1 = \{a^{n!} | n \in N\}$ is not regular.

5. List out the applications of Myhill-Nerode Theorem.

6. Write a Context-Free Grammar for the language: $L = \{x \in \{a, b\}^* | \#_a(x) = \#_b(x)\}$. Here, the notation $\#_1(w)$ represents the number of occurrences of the symbol 1 in the string w.

7. Design a PDA for the language of odd length binary palindromes (no explanation is required, just list the transitions in the PDA).

8. Prove that Context Free Languages are closed under set union.

9. Write a Context Sensitive Grammar for the language $L = \{a^n b^n c^n | n \geq 0\}$ (no explanation is required, just write the set of productions in the grammar).

10. Differentiate between Recursive and Recursively Enumerable Languages.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Draw the state-transition diagram showing an NFA N for the following language L. Obtain the DFAD equivalent to N by applying the subset construction algorithm. **(7)**

$$L = \{x \in \{a, b\}^* | the\ second\ last\ symbol\ in\ x\ is\ b\}$$

(b) Draw the state-transition diagram showing a DFA for recognizing the following language: **(7)**

$L = \{x \in \{0,1\}^* | x\ is\ a\ binary\ representation\ of\ a\ natural$

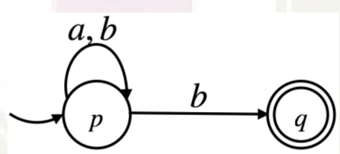
$number\ which\ is\ a$multiple of 5}

OR

12. (a) Write a Regular grammar G for the following language Ldefined as: $L = \{x \in \{a, b\}^* | xdoes\ not\ conatin\ consecutiveb's\}$. **(7)**

(b) Obtain the DFA A_G over the alphabet set $\Sigma = \{a, b\}$, equivalent to the regular grammar G with start symbol S and productions: $S \rightarrow aA$ and $A \rightarrow aA|bA|b$. **(7)**

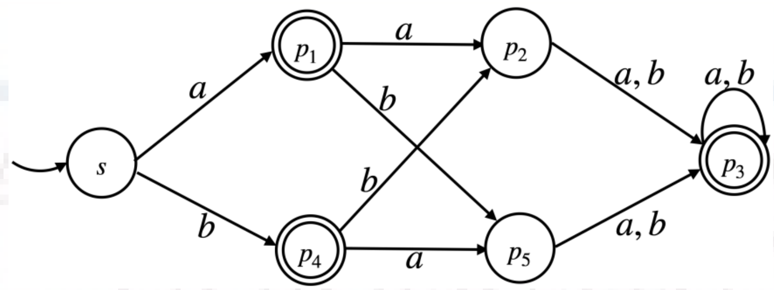
13. (a) Using Kleen's construction, obtain the regular expression for the language represented by the following NFA **(8)**



(b) Using pumping lemma for regular languages, prove that the language $L = \{a^n b^n | n \geq 0\}$ is not regular. **(7)**

OR

14. (a) Obtain the minimum-state DFA from the following DFA. **(8)**



(b) Using ultimate periodicity for regular languages, prove that the language $L = \{a^{n^2} | n \geq 0\}$ is not regular. **(6)**

15. (a) Show the equivalence classes of the canonical Myhill-Nerode relation for the language of binary strings with odd number of 1′s and even number of 0s. **(7)**

(b) With an example, explain ambiguity in Context Free Grammar **(7)**

OR

16. (a) Convert the Context-Free Grammar with productions: $\{S \rightarrow aSb | \epsilon\}$ into Greibach Normal form. **(8)**

(b) Convert the Context-Free Grammar with productions: $\{S \rightarrow aSa | bSb | SS | \epsilon\}$ into Chomsky Normal form. **(6)**

17. (a) Design a PDA for the language $L = \{a^m b^n c^{m+n} | n \geq 0, m \geq 0\}$. Also illustrate the computation of the PDA on a string in the language **(7)**

(b) With an example illustrate how a multi-state PDA can be transformed into an equivalent single-state PDA. **(7)**

OR

18. (a) Using pumping lemma for context-free languages, prove that the language: $L = \{ww|w \in \{a,b\}^*\}$ is not a context-free language. **(6)**

(b) With an example illustrate how a CFG can be converted to a single-state PDA **(8)**

19. (a) Design a Turing machine to obtain the sum of two natural numbers a and b, both represented in unary on the alphabet set $\{1\}$. Assume that initially the tape contains $\vdash 1^a 0 1^b \not b^\omega$. The Turing Machine should halt with $\vdash 1^{a+b} \not b^\omega$ as the tape content. Also, illustrate the computation of your Turing Machine on the input $a = 3$and $b = 2$. **(7)**

(b) With an example illustrate how a CFG can be converted to a single-state PDA. **(7)**

OR

20. (a) Design a Turing machine to obtain the sum of two natural numbers $a$and b, both represented in unary on the alphabet set $\{1\}$. Assume that initially the tape contains $\vdash 1^a 0 1^b \not b^\omega$. The Turing Machine should halt with $\vdash 1^{a+b} \not b^\omega$ as the tape content. Also, illustrate the computation of your Turing Machine on the input $a = 3$and $b = 2$. **(7)**

(b) Write a context sensitive grammar for the language $L = \{a^n b^n c^n | n \geq 0\}$. Also illustrate how the the string $a^2b^2c^2$ can be derived from the start symbol of the proposed grammar. **(7)**

Teaching Plan

Sl. No	Topic	No. of Hours (45 hrs)
Module - 1 (Introduction to Formal Language Theory and Regular Languages)		**9 Hours**
1.1	Introduction to formal language theory – Alphabets, strings, concatenation of strings, Languages	1 Hour
1.2	Deterministic Finite State Automata (DFA) – Example DFA (Proof of correctness of construction not required)	1 Hour
1.3	Formal definition of DFA, Language accepted by the class of DFA	1 Hour
1.4	Nondeterministic Finite State Automata (NFA) – Example NFA	1 Hour
1.5	Formal definition of NFA, NFA with □ transitions - examples, formal definition	1 Hour
1.6	Equivalence of DFA and NFA with and without □ transitions - Subset construction	1 Hour
1.7	Regular Grammar (RG) – Example RGs, derivation of sentences	1 Hour
1.8	Formal definition of RG, Language represented by a RG	1 Hour
1.9	Equivalence of RG and DFA	1 Hour
Module - 2 (More on Regular Languages)		**9 Hours**
2.1	Regular Expression (RE) - Example REs and formal definition	1 Hour
2.2	Conversion of RE to NFA with □ transition	1 Hour
2.3	Conversion of NFA with □ transition to RE (Kleen’s construction)	1 Hour
2.4	Homomorphisms	1 Hour
2.5	Pumping Lemma for regular languages	1 Hour
2.6	Ultimate periodicity	1 Hour
2.7	Closure Properties of Regular Languages (proof not required)	1 Hour

2.8	DFA state minimization - Quotient construction	1 Hour
2.9	State Minimization Algorithm - Example	1 Hour
Module - 3 (Myhill-Nerode Relations and Context Free Grammars)		**10 Hours**
3.1	Myhill-Nerode Relations (MNR) - Example, Properties of MyhillNerode Relation	1 Hour
3.2	Conversion of DFA to MNR (Proof of correctness not required)	1 Hour
3.3	Conversion of MNR to DFA(Proof of correctness not required)	1 Hour
3.4	Myhill-Nerode Theorem (MNT)	1 Hour
3.5	Applications of MNT	1 Hour
3.6	Context Free Grammar (CFG) - Example CFGs and formal definition	1 Hour
3.7	Proving correctness of CFGs	1 Hour
3.8	Derivation Trees and ambiguity	1 Hour
3.9	Chomsky Normal Form	1 Hour
3.10	Greibach Normal Form	1 Hour
Module - 4 (More on Context-Free Languages)		**8 Hours**
4.1	Nondeterministic Pushdown Automata (PDA) – Example PDAs, formal definition	1 Hour
4.2	Acceptance criteria - equivalence	1 Hour
4.3	Deterministic PDA	1 Hour
4.4	Conversion of CFG to PDA (No proof required)	1 Hour
4.5	Conversion of PDA to CGF - Part I (No proof required)	1 Hour
4.6	Conversion of PDA to CGF - Part II (No proof required)	1 Hour
4.7	Pumping Lemma for context-free languages (No proof required)	1 Hour
4.8	Closure Properties of Context Free Languages	1 Hour

Module - 5 (Context Sensitive Languages, Turing Machines)		9 Hours
5.1	Context Sensitive Grammar (CSG) - Examples, formal definition	1 Hour
5.2	Linear Bounded Automata (LBA) - Example LBA, formal definition	1 Hour
5.3	Turing Machine (TM) - TM as language acceptors - examples, formal definition	1 Hour
5.4	TM as transducers - examples	1 Hour
5.5	Robustness of the standard TM model - Multi-tape TMs, Nondeterministic TM	1 Hour
5.6	Universal Turing Machine	1 Hour
5.7	Halting Problem of TM - proof of its undecidability	1 Hour
5.8	Recursive and Recursively Enumerable Languages	1 Hour
5.9	Chomsky classification of formal languages	1 Hour