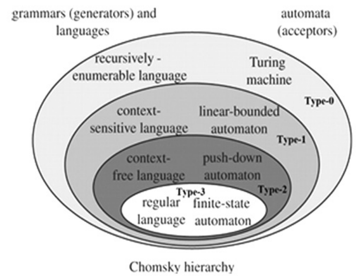
22CS208 THEORY OF COMPUTATION



Source: https://sameer9247.wordpress.com/2016/11/15/theory-of-computation/

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Knowledge of graphs, trees and logic.

COURSE DESCRIPTION AND OBJECTIVES:

This course aims to teach the student to identify different formal language classes and their relationships, strong theoretical foundation for designing compilers. In addition to this the student will be able to learn the techniques for information processing, design different grammars, automata and recognizers for different formal languages.

MODULE-1

UNIT-1 **12L+8T+0P=20 Hours**

INTRODUCTION

Alphabets, Strings and languages, Automata and Grammars, Regular languages, Chomsky hierarchy of languages, Deterministic finite automata (DFA)-Formal definition, Simplified notation, State transition graph, Transition table, Language of DFA; Nondeterministic finite automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of finite automata, FA with output - Moore and Mealy machine, Equivalence of Moore and Mealy machine, Applications and Limitation of FA.

UNIT-2 **12L+8T+0P=20 Hours**

REGULAR EXPRESSIONS

Regular Expression (RE): Definition, Operators of regular expression and their precedence, Algebraic laws for Regular Expressions, Kleen's Theorem, Regular Expression to FA, DFA to regular expression, Arden theorem, Non regular languages, pumping lemma for regular languages (proofs not Required), Application of pumping lemma, Closure properties of regular languages, Decision properties of regular languages.

Grammar Formalism: Regular Grammars-Right linear and left linear grammars, Equivalence between regular linear grammar and FA;

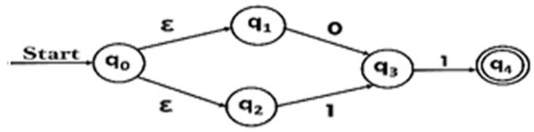
PRACTICES:

- Design DFA and NFA which accepts the following languages over the alphabet {0,1}. And also covert NFA to DFA. Give separate Automata for each and also write RE for the obtained automata.
  a) The set of all strings ends with 00.
  b) With three consecutive 0's.
  c) With 011 as a substring.
  d) Either begin or ends with 01.
  e) Strings whose fourth symbol from the right end is 1.
  f) Even number of 0's.
  g) number of 1 's are divisible by three.
- Design NFA to recognize the following set of strings.
  a) abc, abd, and aacd: Assume the alphabet is {a,b,c,d}.
  b) 0101,101 and 011: Assume the alphabet is {0,1}.
  c) ab,bc and ca: Assume the alphabet is {a,b,c}.

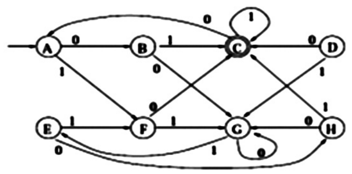
SKILLS:

- ✓ *Investigate syntax and semantics of a regular and context free languages.*
- ✓ *Develop the problem understanding solving ability.*
- ✓ *Design optimized solutions for a language.*

- Convert epsilon NFA to DFA.



- Minimize the following DFA.



- Construct Mealy and Moore Machines and equivalent them for the residue (remainder) mod 3 of binary input.
- Construct Finite Automata for the following Regular Expressions.
  a) RE=ab(a+b)*
  b) RE=(a+ab)(ab+ab)*
- Prove that the following languages are nor Regular.
  a) L={a^p | p is a prime number}
  b) L={b^n | n=i^2 and i>1}
  c) L={ WW^R | W is (a,b)*}
  d) L={a^nb^n+1 |n≥1}

MODULE-2

UNIT-1 **12L+8T+0P=20 Hours**

CONTEXT FREE GRAMMAR

Definition, Examples, Derivation, Derivation trees, Ambiguity in grammar, Inherent ambiguity, Ambiguous to unambiguous CFG, Useless symbols, Simplification of CFGs; Normal forms for CFGs - CNF and GNF, CFLs; Closure properties of Decision properties of CFLs-Emptiness, Finiteness and membership, pumping lemma for CFLs (proofs not Required), Application of pumping lemma.

UNIT-2 **12L+8T+0P=20 Hours**

PDA AND TM

Push Down PDA AND TM Automata (PDA): Description and definition, Instantaneous description, Language of PDA, Acceptance by final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA.

Turing Machines (TM): Basic model, Definition and representation, Instantaneous Description, Language acceptance by TM, Computable functions, Types of Turing Machines, Universal TM, Recursive and Recursively Enumerable Languages, undecidability, Church Turing Thesis, Universal Turing Machine, The universal and diagonalization languages, Reduction between languages and Rice's Theorem.

PRACTICES:

- Construct CFG for the following:
  a) L={a^nb^n|n>1}
  b) L={WW^R | W is (a,b)*}
  c) L={a^p | p is a prime}

- Derive the strings 10001 using left most derivation and right most derivation and parse tree by using the following grammar. And show that grammar is ambiguous.
  S-> T000T
  T->0T|1T|$\in$
- Convert the following CFG to CNF.
  S->ABC|Aa
  A->a
  B->d|$\in$
  C-> Aab|a
- Convert the following CFG to GNF.
  S->AA | 0
  A->SS | 1
- Prove that the following are not CFL.
  e) L={a^p | p is a prime number}
  f) L={b^n | n=i^2 and i>1}
  g) L={ WW^R | W is (a,b)*}
  h) L={a^nb^{n+1} |n≥1}
- Convert the following language or PDA to CFG.
  a) L={a^nb^n|n≥1} and
  b) δ(q,0,z)={(q,xz)}
  c) δ(q,0,x)={(q,xx)}
  d) δ(q,1,x)={(q,x)}
  e) δ(q, $\in$, x)={(p, $\in$)}
  f) δ(p,1,x)={(p,xx)}
  g) δ(p,$\in$, x)={(p, $\in$)}
  h) δ(p, 1,z)={(p, $\in$)}
- Construct PDA for the following Languages.
  a) L={0^n1m | n≥m}
  b) L={a^nb^n|n≥1}
  c) L= L={w\w $\in$ {a, b}*
  d) L={w| $n_a(w) > n_b(w)$}
  e) L={0^n1^{2n} | n>0}
  f) L=Where $w^R$is reverse of w
  g) L= Where $w^R$is reverse of w
  h) L= $\{wcw^R \mid w \in \{a,b\}^*\}$ Where $w^R$is reverse of w
- Construct PDA for the following Languages.
  a) L= {$a^nb^nc^n$ | n >1}
  b) L={$a^nb^ma^mb^n$ | n,m≥1}
- Construct Turing Machine for the following Languages.
  a) L={a^nb^n | n ≥ 1}
  b) L={$0^{2n}1^n$ |n >0 }
  c) L={ww^r | w is (0+1)*}
  d) L={$a^nb^nc^n$|n≥1}.
  e) Well balanced Parenthesis for example: ()()

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Design abstract models of computing, including Deterministic Finite Automata (DFA), non-deterministic Finite. Automata (NFA), Push Down Automata (PDA) and Turing Machine (TM) models and their power to recognize the languages.	Analyze	1,2	1,2,3
2	Design different finite state machines to perform various operations.	Apply	1,2	1,2,3
3	Analyze the given language is regular or not regular, CFL or not, Ambiguous unambiguous, Recursive and Recursive Enumerable.	Analyze	1,2	2
4	Design Regular grammar and context free grammars for a language.	Apply	1,2	1,3

TEXT BOOK:

1. Hopcroft and Ullman, “Introduction to Automata Theory, Languages and Computation”, 2nd Edition, Pearson/ Prentice Hall India, 2007.

REFERENCE BOOKS:

1. Zed A Shaw, Learn C the Hard Way: Practical Exercises on the Computational Subjects You Keep Avoiding (Like C), Addison Wesley, 2015.
2. Christoph Dürr, Sorbonne University, Jill-Jênn Vie, Inria, Competitive programming in Python, Cambridge University Press, 2020.
3. Michael Sipser, “Introduction to Theory of Computation”, 3rd Edition, Course Technology, 2012.