

## THEORY OF COMPUTATION

<b>IV Semester: CSE / IT</b>																				
Course Code	Category	Hours / Week			Credits	Maximum Marks														
AIT002	Foundation	L	T	P	C	CIA	SEE	Total												
		3	-	-	3	30	70	100												
<b>Contact Classes: 45</b>		<b>Tutorial Classes: Nil</b>		<b>Practical Classes: Nil</b>			<b>Total Classes: 45</b>													
<p><b>I. COURSE OVERVIEW:</b>            This course focuses on infinite languages in finite ways, and classifies machines by their power to recognize. It includes finite automata, regular grammar, push down automata, context free grammars, and Turing machines. It is applicable in designing phrasing and lexical analysis of a compiler, genetic programming and recursively enumerable languages.</p> <p><b>II. OBJECTIVES:</b>  <b>The course should enable the students to:</b></p> <ul style="list-style-type: none"> <li>I The fundamental knowledge of automata theory which is used to solve computational problems</li> <li>II The reorganization of context free language for processing infinite information using push down automata.</li> <li>III The computer based algorithms with the help of an abstract machine to solve recursively Enumerable problems</li> </ul> <p><b>III. COURSE OUTCOMES:</b>  <b>After successful completion of the course, students should be able to:</b></p> <ul style="list-style-type: none"> <li>CO 1 <b>Make use of deterministic finite automata and non deterministic finite automata for modeling lexical analysis and text editors.</b> Apply</li> <li>CO 2 <b>Extend regular expressions and regular grammars for parsing and designing programming languages.</b> Understand</li> <li>CO 3 <b>Illustrate the pumping lemma on regular and context free languages for perform negative test.</b> Understand</li> <li>CO 4 <b>Demonstrate context free grammars, normal forms for generating patterns of strings and minimize the ambiguity in parsing the given strings.</b> Understand</li> <li>CO 5 <b>Construct push down automata for context free languages for developing parsing phase of a compiler.</b> Apply</li> <li>CO 6 <b>Apply Turing machines and Linear bounded automata for recognizing the languages, complex problems.</b> Apply</li> </ul> <p><b>IV. SYLLABUS:</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;"><b>UNIT-I</b></td> <td style="width: 60%; text-align: center;"><b>FINITE AUTOMATA</b></td> <td style="width: 25%; text-align: center;"><b>Classes: 10</b></td> </tr> <tr> <td colspan="3">           Fundamentals: Alphabet, strings, language, operations; Introduction to finite automata: The central concepts of automata theory, deterministic finite automata, nondeterministic finite automata, an application of finite automata, finite automata with epsilon transitions.         </td> </tr> <tr> <td style="text-align: center;"><b>UNIT-II</b></td> <td style="text-align: center;"><b>REGULAR LANGUAGES</b></td> <td style="text-align: center;"><b>Classes: 09</b></td> </tr> <tr> <td colspan="3">           Regular sets, regular expressions, identity rules, constructing finite automata for a given regular expressions, conversion of finite automata to regular expressions, pumping lemma of regular sets, closure properties of regular sets (proofs not required), regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and finite automata, inter conversion.         </td> </tr> </table>									<b>UNIT-I</b>	<b>FINITE AUTOMATA</b>	<b>Classes: 10</b>	Fundamentals: Alphabet, strings, language, operations; Introduction to finite automata: The central concepts of automata theory, deterministic finite automata, nondeterministic finite automata, an application of finite automata, finite automata with epsilon transitions.			<b>UNIT-II</b>	<b>REGULAR LANGUAGES</b>	<b>Classes: 09</b>	Regular sets, regular expressions, identity rules, constructing finite automata for a given regular expressions, conversion of finite automata to regular expressions, pumping lemma of regular sets, closure properties of regular sets (proofs not required), regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and finite automata, inter conversion.		
<b>UNIT-I</b>	<b>FINITE AUTOMATA</b>	<b>Classes: 10</b>																		
Fundamentals: Alphabet, strings, language, operations; Introduction to finite automata: The central concepts of automata theory, deterministic finite automata, nondeterministic finite automata, an application of finite automata, finite automata with epsilon transitions.																				
<b>UNIT-II</b>	<b>REGULAR LANGUAGES</b>	<b>Classes: 09</b>																		
Regular sets, regular expressions, identity rules, constructing finite automata for a given regular expressions, conversion of finite automata to regular expressions, pumping lemma of regular sets, closure properties of regular sets (proofs not required), regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and finite automata, inter conversion.																				

<b>UNIT-III</b>	<b>CONTEXT FREE GRAMMARS</b>	<b>Classes: 08</b>
<p>Context free grammars and languages: Context free grammar, derivation trees, sentential forms, right most and leftmost derivation of strings, applications.</p> <p>Ambiguity in context free grammars, minimization of context free grammars, Chomsky normal form, Greibach normal form, pumping lemma for context free languages, enumeration of properties of context free language (proofs omitted).</p>		
<b>UNIT-IV</b>	<b>PUSHDOWN AUTOMATA</b>	<b>Classes: 09</b>
<p>Pushdown automata, definition, model, acceptance of context free language, acceptance by final state and acceptance by empty stack and its equivalence, equivalence of context free language and pushdown automata, inter conversion;(Proofs not required); Introduction to deterministic context free languages and deterministic pushdown automata.</p>		
<b>UNIT-V</b>	<b>TURING MACHINE</b>	<b>Classes: 10</b>
<p>Turing machine: Turing machine, definition, model, design of Turing machine, computable functions, recursively enumerable languages, Church's hypothesis, counter machine, types of Turing machines (proofs not required), linear bounded automata and context sensitive language, Chomsky hierarchy of languages.</p>		
<b>Text Book:</b>		
<p>John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata, Theory, Languages and Computation", Pearson Education, 3<sup>rd</sup> Edition, 2007.</p>		
<b>Reference Books:</b>		
<ol style="list-style-type: none"> <li>1. John C Martin, "Introduction to Languages and Automata Theory", Tata McGraw-Hill, 3<sup>rd</sup> Edition, 2007.</li> <li>2. Daniel I.A. Cohen, "Introduction to Computer Theory", John Wiley &amp; Sons, 2<sup>nd</sup> Edition, 2004.</li> </ol>		
<b>Web References:</b>		
<ol style="list-style-type: none"> <li>1. <a href="https://www.tutorialspoint.com/automata_theory/index.htm">https://www.tutorialspoint.com/automata_theory/index.htm</a></li> <li>2. <a href="https://www.iitg.ernet.in/dgoswami/Flat-Notes.pdf">https://www.iitg.ernet.in/dgoswami/Flat-Notes.pdf</a></li> </ol>		
<b>E-Text Books:</b>		
<p><a href="https://freefundkenotes.files.wordpress.com/2014/02/toc-klp-mishra.pdf">https://freefundkenotes.files.wordpress.com/2014/02/toc-klp-mishra.pdf</a></p>		
<b>MOOC Courses:</b>		
<ol style="list-style-type: none"> <li>1. <a href="http://nptel.ac.in/courses/111103016/">http://nptel.ac.in/courses/111103016/</a></li> <li>2. <a href="http://nptel.ac.in/courses/106106049/">http://nptel.ac.in/courses/106106049/</a></li> <li>3. <a href="http://onlinevideolecture.com/?course_id=1312">http://onlinevideolecture.com/?course_id=1312</a></li> <li>4. <a href="http://www.nptelvideos.in/2012/11/theory-of-computation.html">http://www.nptelvideos.in/2012/11/theory-of-computation.html</a></li> </ol>		
<b>Course Home Page:</b>		