Mindluster Platform

About the course Theory of computation fundamentals

Theory of computation fundamentals course, in this course we will learn about the Theory of Computation fundamentals, exploring the mathematical and conceptual foundations that underpin computer science. We will begin with computational models such as Turing Machines, Finite Automata, and Pushdown Automata, which help us understand what can be computed and how. We will delve into formal languages, examining Regular Languages, Context-Free Languages, Context-Sensitive Languages, and Recursively Enumerable Languages, and their respective grammars and automata. The course will cover Computational Complexity Theory, focusing on classifying problems based on their difficulty and resources required, exploring complexity classes like P, NP, and NP-Complete, and understanding lower bounds. Decidability will also be a key topic, distinguishing between decidable and undecidable problems. We will learn about reductions and completeness, transforming problems to prove their complexity and understanding NP-Completeness. By the end of this course, students will grasp the essential principles of computation, enabling them to analyze and design efficient algorithms and computational systems. This knowledge is crucial for anyone looking to deepen their understanding of the theoretical aspects of computer science and its practical applications.

Computer Science Category's Courses

Course Lesson(119)

Lesson 1 :
Basic Concepts in Automata Theory Mathematical Notations TOC FLAT Theory of Computation

Lesson 2 : **Sets Set Representation Set Operations power Set finite infinite subset proper**

Lesson 3 : **Finite Automata Model Formal Definition TOC FLAT Theory of Computation**

Lesson 4 : **Representation of Finite Automata Transition Diagram Transition Table TOC FLAT**

Lesson 5 : **What is DFA Deterministic Finite Automata FLAT TOC Theory of Computation**

Lesson 6 : **DFA Examples 1 2 Set of all strings Begins with a Starts with ab FLAT TOC**

Lesson 7 : **DFA Examples 3 4 Set of all strings Containing substring ab substring abb FLAT**

Lesson 8 :
DFA Examples 5 6 Set of all strings ends with ab ends with 00 FLAT Theory of computation

Lesson 9 :
DFA Examples 7 Set of all strings Containing 1100 as substring FLAT Theory of computation

Lesson 10 : **DFA Examples 8 9 Accepts the String 1100 Only Accepts the String 101 Only FLAT**

Lesson 11 : **DFA Examples 10 11 Set of all strings with Exactly One a with Exactly Two a s**

Lesson 12 : **DFA Examples 12 13 Set of all strings with At Least One a with At Least Two a s**

Lesson 13 :
DFA Examples 14 Set of all strings not Containing the sub string aab Except substring aab

Lesson 14 : **DFA Examples 15 Set of all strings with Even no of a s and Even no of b s ODD NUMBER**

Lesson 15 :
DFA Examples 16 Set of all strings Containing the sub string ab or ba sub string 01or 10

Lesson 16 :
DFA Examples 17 18 Set of all strings Starts with 1 and Ends with 0 Starts with a Ends with b

Lesson 17 : **Design a DFA for Strings with 3 Consecutive 0 s Without 3 Consecutive 0 s a s b s FLAT**

Lesson 18 :
Design DFA to accept all Binary Strings which are divisible by 3 Three Theory of computation

Lesson 19 :
Design DFA to accept all Binary Strings which are divisible by 4 Four Theory of computation

Lesson 20 :
Design DFA to accept all Binary Strings which are divisible by 5 Five Theory of computation

Lesson 21 : **What is NFA Non Deterministic Finite Automata FLAT TOC Theory of Computation**

Lesson 22 :
Differences between DFA and NFA Deterministic Finite Automata Non Deterministic Types of

Lesson 23 : **NFA Examples Non Deterministic Finite Automata Theory of Computation TOC FLAT**

Lesson 24 : **Converting NFA to DFA Equivalence of DFA and NFA Theory of Computation TOC FLAT**

Lesson 25 :
converting nfa with epsilon to nfa without epsilon with example FLAT TOC Theory of Computation

Lesson 26 : **Converting NFA with epsilon transitions to DFA Theory of Computation FLAT TOC**

Lesson 27 :
Equivalence of 2 Finite State Machines Equivalence between Two Finite Automata DFA NFA

Lesson 28 :
Minimization of Finite Automata Equivalence Partition Table Filling Myhill Nerode DFA NFA

Lesson 29 :
Finite Automata With Output Moore Machine Mealy Machine Theory of Computation TOC

Lesson 30 :
Design a Moore Machine to find 1 s Complement of a given Binary number Theory of Computation

Lesson 31 :
Moore Machine Examples Count number of occurrences of abb Print a when the sequence is 01

Lesson 32 :
Moore Machine produces A as output if the input ends with 10 or B as output if input ends with 11

Lesson 33 : **Design a Moore Machine to Determine residue modulo 3 of a given binary number**

Lesson 34 :
Design a Moore Machine to determine residue modulo 4 of a given Binary number Theory of Computation

Lesson 35 : **Design a Moore Machine Construct a Examples of Theory of Computation TOC FLAT**

Lesson 36 :
Construct Mealy Machine that prints a whenever the sequence 01 is encountered in any input string

Lesson 37 :
Design a Mealy Machine for finding 1 s complement 2 s complement Increment Binary number by 1

Lesson 38 : **Conversion of Mealy Machine to Moore Machine**

Lesson 39 : **Conversion of Moore Machine to Mealy Machine**

Lesson 40 : **What is Regular Expression in Theory of Computation Automata Theory FLAT Define**

Lesson 41 : **Regular Expression Examples Theory of Computation Automata Theory FLAT Design a**

Lesson 42 : **Algebraic Laws of Regular Expression Identity Rules Theory of Computation TOC FLAT**

Lesson 43 :
Simplification of Regular Expressions in TOC An Example Proof using Algebraic laws Identities

Lesson 44 : **Conversion of Finite Automata to Regular Expression using Arden s Method TOC FLAT**

Lesson 45 :
Conversion of DFA to Regular Expression Equivalence of DFA and Regular Expression TOC FLAT

Lesson 46 :
Converting Regular Expression to Finite Automata Example 1 Theory of Computation TOC FLAT

Lesson 47 :
Converting Regular Expression to Finite Automata Example 2 Theory of Computation TOC FLAT

Lesson 48 :
Conversion of Regular expression to Finite Automata using Direct Method Theory of computation

Lesson 49 : **Regular Grammar Left Linear Grammar Right Linear Theory of Computation TOC FLAT**

Lesson 50 :
Converting Finite Automata to Left Linear Grammar Right Linear Grammar Regular Grammar TOC FLAT

Lesson 51 :
Construction of left linear grammar convert right linear grammar to left linear grammar TOC

Lesson 52 :
Construction of Regular Grammar from Finite Automata Theory of Computation TOC FLAT

Lesson 53 :
Pumping Lemma for Regular Languages with an example Theory of Computation TOC FLAT

Lesson 54 :
Prove that L { a p p is prime} is not regular Pumping Lemma for Regular Languages TOC FLAT

Lesson 55 :
Show that L {0 n 2} is not regular perfect square Pumping Lemma for Regular Languages TOC

Lesson 56 :
Show that a language of balanced parenthesis is not regular pumping Lemma for Regular Languages

Lesson 57 :
Pumping Lemma for regular languages with examples prime perfect square balanced parenthesis

Lesson 58 :
Closure properties of Regular Languages Regular Sets TOC FLAT Theory of Computation

Lesson 59 :
Definition of CFG with an Example Context Free Grammar TOC Theory of Computation FLAT CD

Lesson 60 :
Derivation Derivation Tree Left Most Derivation Right Most Derivation TOC Theory of Computation

Lesson 61 :
Construction of CFG for the given language Examples part 1 TOC Theory of Computation FLAT

Lesson 62 :
Construction of CFG for the given language Examples part 2 TOC Theory of Computation FLAT

Lesson 63 :
Construction of CFG for the given language Examples part 3 TOC Theory of Computation FLAT

Lesson 64 :
Construction of CFG for the given language Examples part 4 TOC Theory of Computation FLAT

Lesson 65 : **Construction of CFG for a given language Regular Expression Examples TOC FLAT**

Lesson 66 : **Removal of Unit Productions Simplification of CFG TOC Theory of Computation**

Lesson 67 :
Removal of NULL Epsilon Productions Simplification of CFG TOC Theory of Computation

Lesson 68 : **Removal of Useless Symbols Simplification of CFG TOC Theory of Computation**

Lesson 69 :
Simplification of CFG Reduction of CFG Minimization of CFG Theory of Computation TOC

Lesson 70 : **Ambiguous Grammar Theory of Computation TOC FLAT CD Compiler Design**

Lesson 71 :
Elimination of left recursion Removal of Ambiguity in Grammar CFG TOC Compiler Design

Lesson 72 :
Elimination of left factoring Removal of Ambiguity in Grammar CFG TOC Compiler Design

Lesson 73 : **Chomsky Normal Form Converting CFG to CNF TOC FLAT Theory of Computation**

Lesson 74 : **Greibach Normal Form Converting CFG to GNF TOC FLAT Theory of Computation**

Lesson 75 :
Pumping Lemma for Context Free Language with example a n b n c n TOC FLAT Theory of Com

Lesson 76 :
Prove that L { a p p is prime} is not Context Free Language Pumping Lemma for CFL TOC FLAT

Lesson 77 :
Show that L { a n 2 } is not Context Free Language square Pumping Lemma for CFL TOC FLAT

Lesson 78 :
Show that L { a i b j j i 2 } is not Context Free Language Pumping Lemma for CFL TOC FLAT

Lesson 79 :
Differences between Regular Language and Context Free Language by Theory of Computation FLAT

Lesson 80 : **Closure properties of Context Free Languages CFG TOC FLAT Theory of Computation**

Lesson 81 :
Introduction to Pushdown Automata What Definition Model FLAT TOC Theory of Computation

Lesson 82 :
Construct PDA for the language L {a n b n} Pushdown Automata TOC FLAT Theory of Comp

Lesson 83 : **Graphical Notation for PDA Theory of Computation TOC FLAT**

Lesson 84 :
Instantaneous Description of PDA ID of PDA Push down Automata Theory of computation TOC

Lesson 85 : **Acceptance of PDA Language accepted by pda Theory of computation TOC**

Lesson 86 :
Construct PDA for the language L {a n b 2n} Pushdown Automata TOC FLAT Theory of Comp

Lesson 87 :
Construct PDA for the language L {WcW r} Pushdown Automata TOC FLAT Theory of Computation

Lesson 88 :
Construct PDA for the language L {WW r} What is NPDA Non Deterministic Push down Automata

Lesson 89 : **DPDA in TOC with Examples Deterministic Push down Automata Types of PDA**

Lesson 90 : **Construct PDA for the language L { w na w nb w } PDA for equal number of a s and b s**

Lesson 91 :
Design PDA for Balanced Parentheses Push down Automata Theory of computation TOC FLAT

Lesson 92 :

Design PDA for number of a s greater than number of b s Push down Automata no TOC FLAT

Lesson 93 :
Design PDA for number of a s less than number of b s Push down Automata no TOC FLAT

Lesson 94 : **Two Stack PDA 2 Stack PDA for a n b n c n Theory of computation TOC FLAT**

Lesson 95 :
CFG to PDA Conversion Construction of PDA from CFG Equivalence of CFG and PDA TOC FLAT

Lesson 96 : **PDA to CFG Conversion TOC FLAT Theory of Computation**

Lesson 97 :
Introduction to Turing Machine Formal Definition Model FLAT TOC Theory of Computation

Lesson 98 : **Turing Machine for a n b n Design Construct TOC FLAT Theory of Computation**

Lesson 99 : **Turing Machine for a n b n c n Design Construct TOC FLAT Theory of Computation**

Lesson 100 : **Turing Machine for a 2n b n Design Construct TOC FLAT Theory of Computation**

Lesson 101 : **Turing Machine for a n b 2n Design Construct TOC FLAT Theory of Computation**

Lesson 102 :
Turning Machine for Even Palindrome ww r Length TOC FLAT Theory of Computation Design

Lesson 103 :
Turning Machine for Odd Palindrome waw r wbw r Length TOC FLAT Theory of Computation

Lesson 104 :
Turning Machine for all Palindromes Even Odd Length TOC FLAT Theory of Computation

Lesson 105 :
Design Turing Machine for 1 s Complement and 2 s Complement Theory of computation

Lesson 106 : **Design a Turing Machine for L { wcw w belongs to a s and b s } Theory of computation**

Lesson 107 : **Design a Turing Machine for equal number of a s and b s Theory of computation TOC**

Lesson 108 :
Turing Machine for Addition of 2 numbers Unary integers TOC FLAT Theory of Computation

Lesson 109 :
Turing Machine for Subtraction of 2 numbers Unary integers TOC FLAT Theory of Computation

Lesson 110 :
Design a Turing Machine for Multiplication of 2 unary numbers TOC FLAT Theory of Computation

Lesson 111 :
Design a Turing Machine which accepts the substring aab Contains Theory of computation

Lesson 112 :
Context Sensitive Language Context Sensitive Grammar TOC FLAT Theory of Computation

Lesson 113 : **Linear Bounded Automata TOC FLAT Theory of Computation**

Lesson 114 : **Design Linear Bounded Automata for a n b n c n LBA Theory of computation TOC**

Lesson 115 : **Chomsky hierarchy of languages Types of languages TOC FLAT Theory of Computation**

Lesson 116 : **Recursive and Recursive Enumerable language TOC FLAT Theory of Computation**

Lesson 117 : **Post Correspondence Problem with 2 examples PCP FLAT TOC Theory of Computation**

Lesson 118 :
Types of Turing Machines Variants of Turing Machine Modifications of Turing Machine TOC FLAT

Related courses

Linear Algebra for Computer Scientists

Insertion Sort

Bubble Sort

GCSE Computer Science

Random Access Memory

Binary Trees

for Business Contact
business@mindluster.com