IAS DEL CHOICE I		COMPUTABILITY (Stem (CBCS) scheme)	I	
` ·	•	c year 2016 -2017)		
,	<b>SEMESTER</b>			
Subject Code	15CS54	IA Marks	20	
Number of Lecture Hours/Week	4	Exam Marks	80	
Total Number of Lecture Hours	50	Exam Hours	03	
	CREDITS -	04	<u> </u>	
Course objectives: This course wil	l enable students	s to		
• Introduce core concepts in A	Automata and Th	eory of Computation		
• Identify different Formal lar		•		
<ul> <li>Design Grammars and Reco</li> </ul>		_		
• Prove or disprove theorems	_		es	
• Determine the decidability a				
Module – 1	·			Teachin
				Hours
Why study the Theory of Com	putation, Lang	uages and Strings: S	Strings, 1	10 Hou
Languages. A Language Hierard			chines	
(FSM): Deterministic FSM,	•	guages, Designing	FSM,	
Nondeterministic FSMs, From FS				
FSMs, Minimizing FSMs, Canoni		gular languages, Finite	e State	
Transducers, Bidirectional Transducers				
Textbook 1: Ch 1,2, 3,4, 5.1 to 5.1	0			
Module – 2				
D 1 E : (DE) 1 . :	D DO 171	3 .1 A 11 .1		40 TT
• • • • • • • • • • • • • • • • • • • •	·			10 Hou
REs, Manipulating and Simplify	ing REs. Reg	gular Grammars: Defi	inition,	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular la	ing REs. Reg nguages. Regul	gular Grammars: Defi ar Languages (RL) and	inition, d Non-	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular lar regular Languages: How many RLs	ing REs. Reg nguages. Regul s, To show that a	gular Grammars: Defi ar Languages (RL) and a language is regular, C	inition, d Non-	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular lar regular Languages: How many RLs properties of RLs, to show some lar	ing REs. Reguls, To show that anguages are not I	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs.	inition, d Non-	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular lar regular Languages: How many RLs properties of RLs, to show some lar <b>Textbook 1: Ch 6, 7, 8: 6.1 to 6.4,</b>	ing REs. Reguls, To show that anguages are not I	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs.	inition, d Non-	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular lar regular Languages: How many RLs properties of RLs, to show some lar Textbook 1: Ch 6, 7, 8: 6.1 to 6.4, Module – 3	ing REs. Regulages. Regulas, To show that anguages are not F.1, 7.2, 8.1 to 8	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs.	inition, d Non- Closure	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some lare at the second	ing REs. Regulages. Regulas, To show that an anguages are not F. 7.1, 7.2, 8.1 to 8 roduction to Rev	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Gran	inition, d Non- Closure	
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some languages are to 6.4, Section 1: Ch 6, 7, 8: 6.1 to 6.4, Module – 3  Context-Free Grammars (CFG): Integrated and languages, designing	ing REs. Reguls, To show that an aguages are not It 7.1, 7.2, 8.1 to 8 roduction to Reversity CFGs, simplify	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Gran ying CFGs, proving	inition, d Non- Closure mmars, 1 that a	
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some languages are to 6.4, Module – 3  Context-Free Grammars (CFG): Integration of the context of the con	ing REs. Regulates, To show that a reguages are not Formulates and To Service CFGs, simplified and Parse trees,	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Gran ying CFGs, proving Ambiguity, Normal	inition, d Non- Closure  mmars, that a Forms.	
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some lare Textbook 1: Ch 6, 7, 8: 6.1 to 6.4, Module – 3  Context-Free Grammars(CFG): Into CFGs and languages, designing Grammar is correct, Derivation and Pushdown Automata (PDA): Defin	ing REs. Regulages. Regulas, To show that a nguages are not F 7.1, 7.2, 8.1 to 8 roduction to Revertish Parse trees, attion of non-determined Parse trees, attinuous parameters and parse trees, attinuous parameters are parameters and parameters are parameters.	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Gran ying CFGs, proving Ambiguity, Normal	mmars, 1 that a Forms.	
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some languages are some languages. How many RLs properties of RLs, to show some languages are some languages. Characteristic languages and languages, designing Grammar is correct, Derivation and Pushdown Automata (PDA): Definand Non-deterministic PDAs,	ing REs. Regularies, To show that a regularies are not Foundation to Reversition of non-determinism.	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Grar ying CFGs, proving Ambiguity, Normal erministic PDA, Determant	mmars, that a Forms. ninistic rnative	
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some lar Textbook 1: Ch 6, 7, 8: 6.1 to 6.4, Module – 3  Context-Free Grammars(CFG): Interest CFGs and languages, designing Grammar is correct, Derivation a Pushdown Automata (PDA): Definand Non-deterministic PDAs, latequivalent definitions of a PDA, alt	ing REs. Regulages. Regulas, To show that a regulages are not It 7.1, 7.2, 8.1 to 8 reduction to Reversition of non-determinism ernatives that are	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Gran ying CFGs, proving Ambiguity, Normal erministic PDA, Determination and Halting, alternation and equivalent to PDA	mmars, that a Forms. ninistic rnative	
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some languages are some languages. How many RLs properties of RLs, to show some languages are some languages. CFG in the CFGs and languages, designing Grammar is correct, Derivation and Pushdown Automata (PDA): Define and Non-deterministic PDAs, languages, alternative and languages are pulsary and languages. PDAs, landuages are pulsary and languages are pulsary and languages are pulsary and languages. PDAs, languages are pulsary and languages are pulsary and languages are pulsary and languages. PDAs, languages are pulsary and languages are pulsary and languages. PDAs, languages are pulsary and languages are pulsary and languages.	ing REs. Regulages. Regulas, To show that a regulages are not It 7.1, 7.2, 8.1 to 8 reduction to Reversition of non-determinism ernatives that are	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Gran ying CFGs, proving Ambiguity, Normal erministic PDA, Determination and Halting, alternation and equivalent to PDA	mmars, that a Forms. ninistic rnative	
RES, Manipulating and Simplify Regular Grammars and Regular languages: How many RLS properties of RLs, to show some languages: How many RLS properties of RLs, to show some languages and languages and languages, designing Grammar is correct, Derivation and Pushdown Automata (PDA): Definition of Non-deterministic PDAs, languages, designing and Non-deterministic PDAs, languages and languages and languages and languages. Pushdown Automata (PDA): Definition of a PDA, alto and Non-deterministic PDAs, alto and Non-deterministic P	ing REs. Regulages. Regulages. Regulages. Regulas, To show that a regulages are not It 7.1, 7.2, 8.1 to 8 roduction to Revertion of Rev	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Gran ying CFGs, proving Ambiguity, Normal erministic PDA, Determantal Halting, alter the not equivalent to PDA 3.4, 12.5, 12.6	mmars, that a Forms. ministic rnative	10 Hou
RES, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some languages are stook 1: Ch 6, 7, 8: 6.1 to 6.4, Module – 3  Context-Free Grammars(CFG): Interpretation of the CFGs and languages, designing Grammar is correct, Derivation and Non-deterministic PDAs, and Non-deterministic PDAs, alto and Non-deterministic PDAs, alto and Non-deterministic PDA, alto an	ing REs. Regulages. Regulas, To show that a regulages are not F. 7.1, 7.2, 8.1 to 8 reduction to Reversition of non-determinism ernatives that are 8, 12.1, 12.2, 12 ree Languages:	gular Grammars: Defi ar Languages (RL) and a language is regular, C RLs. 3.4 write Systems and Grar ying CFGs, proving Ambiguity, Normal erministic PDA, Determ and Halting, alter enot equivalent to PDA .4, 12.5, 12.6	mmars, that a Forms. ninistic rnative A.	10 Hou
RES, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some languages are some languages. How many RLs properties of RLs, to show some languages are some languages. Characteristic structures and languages, designing Grammar is correct, Derivation and Pushdown Automata (PDA): Defined Non-deterministic PDAs, and Non-deterministic PDAs, alto and Non-Context-Fred Ranguages (CFL) fit, Showing a languages (CFL) fit, Showing a languages and Non-Context-Fred Ranguages (CFL) fit, Showing a languages of Ranguages (CFL) fit, Showing a languages (CFL)	ing REs. Regulages. Regulas, To show that a regulages are not February 7.1, 7.2, 8.1 to 8 roduction to Revertical Parse trees, attitude of non-determinism ernatives that are 8, 12.1, 12.2, 12 ree Languages: nguage is content.	gular Grammars: Defiar Languages (RL) and a language is regular, CRLs.  3.4  Write Systems and Grarying CFGs, proving Ambiguity, Normal Derministic PDA, Determinant and Halting, alternated to equivalent to PDA, 4, 12.5, 12.6  Where do the Contextst-free, Pumping theory	mmars, that a Forms. ninistic rnative A.	10 Hou
RES, Manipulating and Simplify Regular Grammars and Regular languages: How many RLS properties of RLs, to show some languages are some languages. How many RLS properties of RLs, to show some languages are some languages. CFG: Into 6.4, Module – 3  Context-Free Grammars (CFG): Into CFGs and languages, designing Grammar is correct, Derivation and Pushdown Automata (PDA): Define and Non-deterministic PDAs, and Post Post Post Post Post Post Post Post	ing REs. Regulages. Regulas, To show that a regulages are not It 7.1, 7.2, 8.1 to 8 roduction to Rever CFGs, simplified Parse trees, attion of non-determinism ernatives that are 8, 12.1, 12.2, 12 ree Languages: nguage is contempted of CFLs, Determinism of CFLs,	gular Grammars: Defiar Languages (RL) and a language is regular, CRLs.  3.4  Write Systems and Gramying CFGs, proving Ambiguity, Normal Perministic PDA, Determinant and Halting, alternated to equivalent to PDA, 4, 12.5, 12.6  Where do the Context-free, Pumping theoreministic CFLs. Algorithm	mmars, that a Forms. ministic rnative A.	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some languages of RLs, to show some languages of RLs, to show some languages of RLs, to show some language of RLs, interpretation of RLs, and languages, designing of Grammar is correct, Derivation and Pushdown Automata (PDA): Define and Non-deterministic PDAs, and Requivalent definitions of a PDA, alto and Restbook 1: Ch 11, 12: 11.1 to 11.  Module – 4  Context-Free and Non-Context-Free Languages (CFL) fit, Showing a languages (CFL) fit, Showing a languages of RLs. Extended the Rest of RLs. Extended the Regular Indiana.	ing REs. Regularized Regulariz	gular Grammars: Defiar Languages (RL) and a language is regular, CRLs.  3.4  Write Systems and Gramying CFGs, proving Ambiguity, Normal Exaministic PDA, Determinant and Halting, alternated to equivalent to PDA, 4, 12.5, 12.6  Where do the Context of the Context	mmars, that a Forms. ninistic rnative A. xt-Free rem for ms and estions.	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some languages of RLs, to show some languages of RLs, to show some languages of RLs, to show some language of RLs, interpretable of RLs, Important closure properties o	ing REs. Regulages. Regulas, To show that a regulages are not February 7.1, 7.2, 8.1 to 8 roduction to Revertical Parse trees, attion of non-determinism ernatives that are 18, 12.1, 12.2, 12 ree Languages: Inguage is contended of CFLs, Determined Pecidable question odel, Representation of Representa	gular Grammars: Defiar Languages (RL) and a language is regular, CRLs.  3.4  Write Systems and Grarying CFGs, proving Ambiguity, Normal Perministic PDA, Determinant and Halting, alternated to equivalent to PDA (A, 12.5, 12.6)  Where do the Context (A) and (B) and (B) and (B) articles (B) ar	mmars, that a Forms. ninistic rnative A. xt-Free rem for ms and estions.	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some larget Textbook 1: Ch 6, 7, 8: 6.1 to 6.4, Module – 3  Context-Free Grammars(CFG): Interpretation of Languages, designing Grammar is correct, Derivation and Pushdown Automata (PDA): Define and Non-deterministic PDAs, and Pushdown Automata (PDA): Define and Non-deterministic PDAs, and Pextbook 1: Ch 11, 12: 11.1 to 11.  Module – 4  Context-Free and Non-Context-Free Languages(CFL) fit, Showing a languages (CFL) fit, Showing a languages (CFL) fit, Showing a languages (CFL) and Non-Context-Free Languages (CFL) fit, Showing a language	ing REs. Regulages. Regulas, To show that a regulages are not It 7.1, 7.2, 8.1 to 8 roduction to Rever CFGs, simplified Parse trees, attion of non-determinism ernatives that are 8, 12.1, 12.2, 12 ree Languages: nguage is contemporated for CFLs, Determined Proceedings of CFLs, Determined Procedular Representation of TM construction of the second procedular regular regula	gular Grammars: Defiar Languages (RL) and a language is regular, CRLs.  3.4  Write Systems and Gramying CFGs, proving Ambiguity, Normal Perministic PDA, Determine and Halting, altered to equivalent to PDA, 4, 12.5, 12.6  Where do the Context of t	mmars, that a Forms. ninistic rnative A. xt-Free em for ms and estions. tability	10 Hou
REs, Manipulating and Simplify Regular Grammars and Regular languages: How many RLs properties of RLs, to show some larget Textbook 1: Ch 6, 7, 8: 6.1 to 6.4, Module – 3  Context-Free Grammars(CFG): Interest CFGs and languages, designing Grammar is correct, Derivation at Pushdown Automata (PDA): Define and Non-deterministic PDAs, alterest Textbook 1: Ch 11, 12: 11.1 to 11.  Module – 4  Context-Free and Non-Context-Free Languages(CFL) fit, Showing a laterest CFL, Important closure properties of Decision Procedures for CFLs: Decision Procedur	ing REs. Regulages. Regulas, To show that a regulages are not It 7.1, 7.2, 8.1 to 8 roduction to Rever CFGs, simplified Parse trees, attion of non-determinism ernatives that are 8, 12.1, 12.2, 12 ree Languages: nguage is contemporated for CFLs, Determined Proceedings of CFLs, Determined Procedular Representation of TM construction of the second procedular regular regula	gular Grammars: Defiar Languages (RL) and a language is regular, CRLs.  3.4  Write Systems and Gramying CFGs, proving Ambiguity, Normal Perministic PDA, Determine and Halting, altered to equivalent to PDA, 4, 12.5, 12.6  Where do the Context of t	mmars, that a Forms. ninistic rnative A. xt-Free em for ms and estions. tability	10 Hou
Regular Expressions (RE): what i REs, Manipulating and Simplify Regular Grammars and Regular largular Languages: How many RLs properties of RLs, to show some largular Languages: How many RLs properties of RLs, to show some largular Languages: How many RLs properties of RLs, to show some largular Languages of RLs, to 6.4, Module – 3  Context-Free Grammars(CFG): Interpretation of CFGs and languages, designing Grammar is correct, Derivation and Pushdown Automata (PDA): Define and Non-deterministic PDAs, alterated to PDA, alterated to PDA, alterated to English to 11, 12: 11.1 to 11.  Module – 4  Context-Free and Non-Context-Free Languages(CFL) fit, Showing a laterated to CFL, Important closure properties of Decision Procedures for CFLs: Decision Proc	ing REs. Regulars, To show that a regular are not H 7.1, 7.2, 8.1 to 8 roduction to Rever CFGs, simplified Parse trees, attion of non-determinism ernatives that are 8, 12.1, 12.2, 12 ree Language is contended for CFLs, Determined Parse trees of CFLs, Determined CFLs, Determined CFLs, Determined Representation of the construction of the construc	gular Grammars: Defiar Languages (RL) and a language is regular, CRLs.  3.4  Write Systems and Gramying CFGs, proving Ambiguity, Normal Perministic PDA, Determine and Halting, altered and Halting, altered to equivalent to PDA, 4, 12.5, 12.6  Where do the Context at the context of the Context at the CFLs. Algorithm ons, Un-decidable questation, Language acceptation.  2, Textbook 2: Ch 9.1	mmars, that a Forms. ministic rnative A. ext-Free em for ms and estions. tability	10 Hou 10 Hou 10 Hou

Undecidable languages, halting problem of TM, Post correspondence problem. Complexity: Growth rate of functions, the classes of P and NP, Quantum Computation: quantum computers, Church-Turing thesis.

Textbook 2: Ch 9.7 to 9.8, 10.1 to 10.7, 12.1, 12.2, 12.8, 12.8.1, 12.8.2

**Course outcomes:** The students should be able to:

- Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation
- Learn how to translate between different models of Computation (e.g., Deterministic and Non-deterministic and Software models).
- Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers.
- Develop skills in formal reasoning and reduction of a problem to a formal model, with an emphasis on semantic precision and conciseness.
- Classify a problem with respect to different models of Computation.

## Question paper pattern:

The question paper will have TEN questions.

There will be TWO questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer FIVE full questions, selecting ONE full question from each module.

## **Text Books:**

- 1. Elaine Rich, Automata, Computability and Complexity, 1<sup>st</sup> Edition, Pearson Education, 2012/2013
- 2. K L P Mishra, N Chandrasekaran, 3<sup>rd</sup> Edition, Theory of Computer Science, PhI, 2012.

## **Reference Books:**

- 1. John E Hopcroft, Rajeev Motwani, Jeffery D Ullman, Introduction to AutomataTheory, Languages, and Computation, 3rd Edition, Pearson Education, 2013
- 2. Michael Sipser: Introduction to the Theory of Computation, 3rd edition, Cengage learning, 2013
- 3. John C Martin, Introduction to Languages and The Theory of Computation, 3<sup>rd</sup> Edition, Tata McGraw –Hill Publishing Company Limited, 2013
- 4. Peter Linz, "An Introduction to Formal Languages and Automata", 3rd Edition, Narosa Publishers, 1998
- 5. Basavaraj S. Anami, Karibasappa K G, Formal Languages and Automata theory, Wiley India, 2012
- 6. C K Nagpal, Formal Languages and Automata Theory, Oxford University press, 2012.