

Theory of Computation - Video course

COURSE OUTLINE

The objective of the course is to provide an exposition first to the notion of computability, then to the notion of computational feasibility or tractability.

We first convince ourselves that for our purpose it suffices to consider only language recognition problems instead of general computational problems.

We then provide a thorough account of finite state automata and regular languages, not only because these capture the simplest language class of interest and are useful in many diverse domains.

But also because many fundamental notions like nondeterminism, proofs of impossibility, etc. get discussed at a conceptually very simple level. We then consider context grammars and languages, and their properties.

Next, we consider Turing machines (TMs), show that as a model it is very robust, and the reasonableness of the Church-Turing hypothesis. After we realize TMs can work with (codes of) TMs as inputs, we obtain a universal TM.

We then obtain the separation of the classes r.e., and recursive. A number of TM related problems are shown to be undecidable. Next, Post's correspondence problem (PCP) is shown undecidable.

Finally, we introduce the notion of feasible or tractable computation. Classes NP, co-NP are defined and we discuss why these are important. We discuss the extended Church-Turing hypothesis.

After we discuss polynomial time many-one reducibility and prove Cook-Levin theorem, a number of natural problems from different domains are shown NP-complete.

The treatment is informal but rigorous. Emphasis is on appreciating that the naturalness and the connectedness of all the different notions and the results that we see in the course.

COURSE DETAIL

Module	Topics	No.of Hours
Regular languages	<ul style="list-style-type: none"> Introduction: Scope of study as limits to computability and tractability. Why it suffices to consider only decision problems, equivalently, set membership problems. Notion of a formal language. 	1
	<ul style="list-style-type: none"> DFAs and notion for their acceptance, informal and then formal definitions. Class of regular languages. Closure of the class under complementation, union and intersection. Strategy for designing DFAs. 	3



NP-TEL

NPTEL

<http://nptel.iitm.ac.in>

Computer Science and Engineering

Pre-requisites:

1. Discrete Mathematics.
2. Introductory Algorithms course.

Coordinators:

Prof. Somenath Biswas
Department of Computer Science and
Engineering IIT Kanpur

	<ul style="list-style-type: none"> • Pumping lemma for regular languages. Its use as an adversarial game. • Generalized version. Converses of lemmas do not hold. 	2
	<ul style="list-style-type: none"> • NFAs. Notion of computation trees. Definition of languages accepted. Construction of equivalent DFAs of NFAs. NFAs with epsilon transitions. Guess and check paradigm for design of NFAs. 	4
	<ul style="list-style-type: none"> • Regular expressions. Proof that they capture precisely class of regular languages. Closure properties of and decision problems for regular languages. 	3
	<ul style="list-style-type: none"> • Myhill-Nerode theorem as characterization of regular languages. States minimization of DFAs. 	2
Context free languages	<ul style="list-style-type: none"> • Notion of grammars and languages generated by grammars. Equivalence of regular grammars and finite automata. Context free grammars and their parse trees. Context free languages. Ambiguity. 	2
	<ul style="list-style-type: none"> • Pushdown automata (PDAs): deterministic and nondeterministic. Instantaneous descriptions of PDAs. Language acceptance by final states and by empty stack. Equivalence of these two. 	2
	<ul style="list-style-type: none"> • PDAs and CFGs capture precisely the same language class. 	1
	<ul style="list-style-type: none"> • Elimination of useless symbols, epsilon productions, unit productions from CFGs. Chomsky normal form. 	2
	<ul style="list-style-type: none"> • Pumping lemma for CFLs and its use. Closure properties of CFLs. Decision problems for CFLs. 	3
Turing machines, r.e. languages, undecidability	<ul style="list-style-type: none"> • Informal proofs that some computational problems cannot be solved. 	1
	<ul style="list-style-type: none"> • Turing machines (TMs), their instantaneous descriptions. Language acceptance by TMs. 	5

	Hennie convention for TM transition diagrams. Robustness of the model - equivalence of natural generalizations as well as restrictions equivalent to basic model. Church-Turing hypothesis and its foundational implications.	
	<ul style="list-style-type: none"> Codes for TMs. Recursively enumerable (r.e.) and recursive languages. Existence of non-r.e. languages. Notion of undecidable problems. Universal language and universal TM. Separation of recursive and r.e. classes. Notion of reduction. Some undecidable problems of TMs. Rice's theorem. 	5
	<ul style="list-style-type: none"> Undecidability of Post's correspondence problem (PCP), some simple applications of undecidability of PCP. 	3
Intractability	<ul style="list-style-type: none"> Notion of tractability/feasibility. The classes NP and co-NP, their importance. Polynomial time many-one reduction. Completeness under this reduction. Cook-Levin theorem: NP-completeness of propositional satisfiability, other variants of satisfiability. NP-complete problems from other domains: graphs (clique, vertex cover, independent sets, Hamiltonian cycle), number problem (partition), set cover. 	6

References:

1. John E. Hopcroft, Rajeev Motwani and Jeffery D. Ullman, Automata Theory, Languages, and Computation (3rd. Edition), Pearson Education, 2008.
2. Michael Sipser, Introduction to the Theory of Computation, Books/Cole Thomson Learning, 2001.
3. JE Hopcroft and JD Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.