COURSE DESCRIPTION

Course No. **Course Type**	CSC 553 Required	**Course Title**	Theory of Computation
Sem. Hours	3	**Coordinator**	Francine Blanchet-Sadri

Current Catalog Description:

Finite state automata and regular expressions, context-free grammars, push-down automata and their use in parsing, overview of language translation systems, models for programming language semantics, computability and undecidability.

Textbook:

<u>Automata and Computability.</u> Dexter C. Kozen, Springer-Verlag, 1997.

References:

None

Course Outcomes:

Upon successful completion of this course, a student should be able to:

1. *Understand* the basic theoretical models of computability: deterministic and nondeterministic finite automata, pushdown automata, and variants of Turing machines
2. *Design* finite automata corresponding to given regular sets, and *describe* the regular set associated to a given finite automaton. Do the same with pushdown automata and context-free languages, and with Turing machines and recursively enumerable sets
3. *Comprehend* and *apply* a number of algorithms such as: the subset construction to transform a nondeterministic finite automaton into a deterministic one; the DFA state minimization to minimize the number of states in a deterministic finite automaton; and the conversion algorithms from regular expressions to finite automata and vice versa
4. *Understand* limitations of finite automata (respectively, pushdown automata) and *prove* that some sets are not regular (respectively, context-free) by using the pumping lemma for regular languages (respectively, context-free languages)
5. *Simulate* CFGs by NPDAs and vice versa, that is, convert a given context-free grammar to an equivalent nondeterministic pushdown automaton, and convert a nondeterministic pushdown automaton to an equivalent context-free grammar
6. *Apply* algorithms to transform context-free grammars into normal forms such as the Chomsky and Greibach normal forms

7. *Prove* that some problems are decidable or undecidable using techniques such as diagonalization and reduction

Activities Enabling Program Outcomes (POx refers to program student outcome x)

Instruction: The course teaches students the basic theoretical models of computability, finite state automata, pushdown automata, and Turing machines, to enable the students to design them and to analyze their respective languages, the regular, context-free, and recursively enumerable languages (POa,b,c,i,j). The course trains students to apply a number of algorithms related to these models such as an algorithm on the equivalence of finite state automata and regular expressions, and one on the equivalence of context-free grammars and pushdown automata (POa,c,j). The course imparts students the necessary skills and knowledge to be competent in solving problems related to these models, and more specifically, in proving that some problems are decidable or undecidable (POa,b,i,j).

Student Activities and Assessment: Every offering of this course will include (details of assessment criteria and expectations are in outcome rubrics):

- One or more test questions or assignments in which students choose computing and mathematical principles to formulate models (POa)
- One or more test questions or assignments in which students translate computing and mathematical theory into application (POa)
- One or more test questions or assignments in which students produce written proofs of fundamental computer science properties (e.g., using the Pumping Lemma or proofs of undecidability) (POfw)

Prerequisites by Topic:

Students must have
- a grade of at least C (2.0) in CSC 350 (Foundations of Computer Science II), or
- permission of instructor.

Major Topics Covered in the Course:

- Finite Automata
- Pushdown Automata and Context-Free Languages
- Turing Machines and Effective Computability

Estimated Curriculum Category Content (Semester hours):

Area	***Core***	***Advanced***	***Area***	***Core***	***Advanced***
Algorithms	.5	0	**Software design**	0	0
Data structures	0	0	**Prog. Languages**	2.5	0
Comp Org & Arch	0	0			