CS 3613: Theoretical Foundations of Computing

Required Course: Required
Course Number: CS 3613
Course Name: Theoretical Foundations of Computing
Credit Hours: 3
Lecture Hours: 3
Lab Hours: 0
Instructors: Dr. H. K. Dai

Book Title(s): Introduction to the Theory of Computation, 3rd edition
Book Author(s): M. Sipser
Book Year(s): 2012

Book Title(s) / Reference: Introduction to Languages and the Theory of Computation
Book Author(s): J. A. Martin
Book Year(s): 2010

Course Description: Introduction to the classical theory of computer science. Sequential machines and their applications to devices, processes, and programming. Models of computation: finite-state automata, push-down automata, Turing machines. The role of non-determinism. Limits of digital computation. Computability and unsolvability. The Church-Turing Thesis.

Course Prerequisites: CS 2133 (Computer Science II) and CS 3653 (Discrete Mathematics for Computer Science)

Course Goals: The goal of this course is to give students an ability to develop and rigorously reason about abstract formal models of computations and learn the powers and limitations of such formalism. “Classical” models, such as finite automata/regular expressions, pushdown automata/context-free grammars, and Turing machines will be studied in depth.

Student Outcomes:

Student Outcome	Course Outcome
1	Analyze language-recognition and generation-problems through the powers and limitations of abstract formal models of computation (regularity, context-freedom, recursive enumerability), and identify possible machine and/or grammar constructions for the languages.
2	Identify and implement possible machine and/or grammar constructions for language-recognition and generation-problems with resource-restrictions: determinism/nondeterminism, "normal-form" machines/grammars, variants of machines/computation models (Church-Turing Thesis).
6	Constructions of machines and/or grammars for language-recognition and generation-problems follow general programming paradigms such as semantics of states/variables, recursion/induction/iteration-loops, and divide/conquer. Limitations of the abstract formal models of computation are studied through contradictory arguments (pumping lemmas), closure properties, and diagonalization argument.

Course Topics:

Knowledge Area	Total Hours of Coverage
Algorithms and Complexity (AL)	38

Body of Knowledge Coverage

Knowledge Area	Knowledge Unit	Topics Covered	Hours

AL	Basic Automata Computability and Complexity	All core-tier 1: finite-state machines, regular expressions, and the halting problem Core-tier 2: context-free grammars, and introduction to the P and NP classes and the P versus NP problem	3 3
AL	Advanced Automata Theory and Computability	Elective: sets and languages (regular languages, review of deterministic finite automata (DFAs), nondeterministic finite automata (NFAs), equivalence of DFAs and NFAs, review of regular expressions and their equivalence to finite automata, closure properties and proving languages non-regular, via the pumping lemma or alternative means)	12
AL	Advanced Automata Theory and Computability	Elective: context free languages (push-down automata (PDAs), relationship of PDAs and context free grammars, properties of context-free languages)	11
AL	Advanced Automata Theory and Computability	Elective: Turing machines or an equivalent formal model of universal computation, nondeterministic Turing machines, and the Church-Turing thesis	9