Course Outcomes	**CSC 4890**

CSC 4890: Introduction to Theory of Computation

Credit Hours: 3 hours

Prerequisites:

CSC 2259

Prerequisites by Topics:

Set algebra, elementary formal logic, constructing proofs, recurrence relations.

Catalog Course Description:

Introduction to finite automata, regular expressions and languages; push-down automata and context-free languages; selected advanced language theoretical topics; emphasis on technique.

Course Outcomes:

1. Master regular languages and finite automata.
2. Master context-free languages, push-down automata, and Turing recognizable languages.
3. Be exposed to a broad overview of the theoretical foundations of computer science.
4. Be familiar with thinking analytically and intuitively for problem-solving situations in related areas of theory in computer science.

Texts and Other Course Materials

Introduction to the Theory of Computation, by Michael Sipser, latest edition, PWS Publishing Company.

Major Topics

- Review definitions and notations for sets, relations and functions.
- Introduction to formal languages and Kleene’s Theorem.
- Mathematical formal proofs including proof by induction and by contradiction.
- The recursive definitions of regular languages, regular expressions and the use of regular expressions to represent regular languages.
- Detailed knowledge and the relationship between regular expressions and finite automata.
- Nondeterminism.
- Minimal finite automata in terms of equivalence classes of strings and associated algorithm for finding minimal DFA.

- Pumping lemma for proving that languages are not regular.
- Context-free grammars and how to prove properties of context-free grammars.
- Pushdown automata.
- Application of the pumping lemma for CFL to demonstrate that a language is not context-free.
- Turing machines (deterministic and non-deterministic) and Church-Turing Thesis.
- Brief introduction to ecursively enumerable languages.
- Brief introduction to computability including the halting problem and related problems.
- Brief introduction to complexity theory including the classes P and NP, and the NP-complete problems.

Assignments/Projects/Laboratory Projects/Homework

- Consider the PDA shown in Figure 2.26 (on page 118). Use the methods described in Lemma 2.27 to convert it to a CFG. Next, prove that the derived CFG is equivalent to the CFG shown at the beginning of Example 2.25, which, in turn, is equivalent to the PDA you used at the beginning of this problem.

- Find the minimal DFA using the algorithm discussed in class.

- Define the difference between regular language and a context free language. Are all context free languages regular languages?

Curriculum Category Content (estimated in semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	0	29 hrs	Data Structures	4.5	0
Software Design	0	0	Prog. Languages	0	13.5
Computer Arch.	0	2 hrs			

Relationship to Criterion 3 Outcomes

A	B	C	D	E	F	G	H	I	J	K
*	*	*		*		*		*		*

Theoretical Content:
The entire course is devoted to the study of concepts, proofs, and algorithms related to the patterns of strings in formal and programming languages.

Math and Fundamentals:

Data Structures:

Stacks, tapes.

Algorithms and Software:

Analysis – 50% (22.5 hrs).

Design – 15% (6.75 hrs) evaluation of efficiency and suitability of the design to the problem.

Computer Organization and Architecture:

Concept of the model machine 5% (2.25 hrs).

Concepts of Programming Languages:

Abstract programming languages for an abstract computing device 30% (13.5 hrs).

Social and Ethical Issues:

Oral Communication (presentations)

Class discussion is an essential part of the teaching experience.

Written Communication:

In homeworks and exams students must articulate thoughts and arguments using good grammar and composition skills.

Course Coordinator: Dr. Evangelos Triantaphyllou
Last Modified: June 13, 2007