Models of Computation

CSC 445 – Spring 2014

Introduction

Computation and algorithms seem to occur naturally in our daily lives. Consider counting change or following a recipe to make your favorite dish. The art of programming takes this to the limit by formally encoding computations and algorithms for a machine to follow. What are the limits of this algorithmic approach? Are there mathematical objects that cannot be computed by a machine via an algorithm? If we can express a computation via an algorithm how long would it take to compute? How much space would the computation consume?

In this course we will investigate many of these interesting questions. We start with simple models of computations such as the finite state machine and the push down automaton. As our main tool we will use an idealized general purpose computer invented by Alan Turing: the Turing Machine. This idealized machine allows us to study the limits of computability and the complexity of computations without having to worry too much about actual hardware.

The goals of the course are:

- To be exposed to the terminology of the theory of computing
- Familiarity with some of the major results in computability and complexity theory.
- A basic understanding of the major models of computations.

Required Text

Formal Language: A Practical Introduction, Adam Brook Webber, Franklin, Beedle & Associates, Inc., 2007.

Instructor

Dr. Lutz Hamel
email: hamel@cs.uri.edu
office: Tyler Hall, Rm 251
hours: Tue 12:15-1:15pm, Wed 2-3pm

Schedule

TueTh 11:00am – 12:15pm, Crawford Hall Rm 223
Webpage: http://homepage.cs.uri.edu/faculty/hamel/courses/2014/spring2014/csc445

Prerequisites:

CSC340

Grading:

Homework	50%
Midterm	25%
Final	25%

Policies:

- Check the website (often)! I will try to keep the website as up-to-date as possible.
- Class attendance, promptness, participation, and adequate preparation for each class are expected. If you are absent, it is your responsibility to find out what you missed (e.g. handouts, announcements, assignments, new material, etc.)
- **Late assignments:** Late assignments will **not** be accepted.
- Make-up quizzes and exams will **not** be given without a valid excuse, such as illness. If you are unable to attend a scheduled examination due to valid reasons, please inform myself, or the department office in Tyler Hall, prior to the exam time. Under such circumstances, you are not to discuss the exam with any other class member until after a make-up exam has been completed.
- All work is to be the result of your own individual efforts unless explicitly stated otherwise. Plagiarism, unauthorized cooperation or any form of cheating will be brought to the attention of the Dean for disciplinary action. See the appropriate sections (8.27) of the University Manual.
- Software piracy will be dealt with exactly like stealing of university or departmental property. Any abuse of computer or software equipment will be subject to disciplinary action.

Tentative Course Outline:

Fundamentals
Closure Properties For Regular Languages
Deterministic Finite Automata (DFA)
Nondeterministic Finite Automata (NFA)
Regular Expressions
Grammars
- Non-Regular Languages
- Context-Free Languages
Stack Machines
- The Context-Free Frontier
- Stack Machine Applications
Turing Machines
Computability
Uncomputability
Cost Models
Complexity Classes
- Deterministic Complexity Classes
- Nondeterministic Complexity Classes