# Lecture 1: Overview and Administrivia

20 January 2009

## 1 Course overview

The details vary from term to term. This is a rough outline of the course and a motivation for why we study this stuff.

1. Theory of Computation.
   - Build formal mathematical models of computation.
   - Analyze the inherent capabilities and limitations of these models.
2. Course goals:
   - Simple practical tools you can use in later courses, projects, etc. The course will provide you with tools to model complicated systems and analyze them.
   - Inherent limits of computers: problems that no computer can solve.
   - Better fluency with formal mathematics (closely related to skill at debugging programs).
3. What is computable?
   (a) check if a number n is prime
   (b) compute the product of two numbers
   (c) sort a list of numbers
   (d) find the maximum number from a list
4. Computability, complexity, automata.
5. Example:

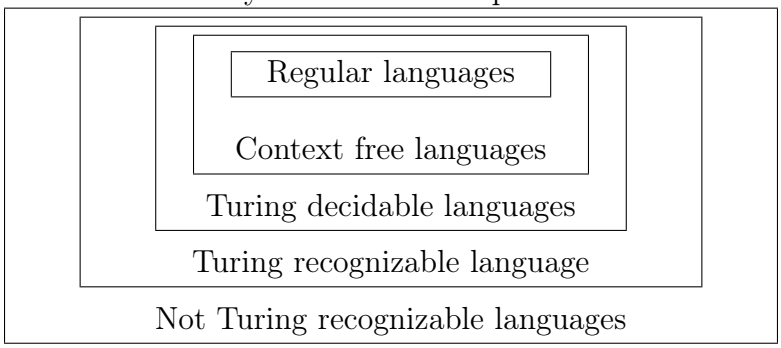
```
input n;
assume n>1;
while (n !=1) {
    if (n is even)
        n := n/2;
    else
        n := 3*n+1;
}
```

Does this program always stop? Not known.

6. Course divides into three sections

   - regular languages ———— practical tools
   - context-free languages
   - Turing machines.

7. Turing machines and decidability → limits of computation.




8. Regular languages and context-free languages:

   - Simple and computationally efficient.
   - heavily used in programming languages and compilers.
   - also in computational linguistics.
   - well-known application: `grep`

9. Difference in scope

   - regular languages describe tokens (e.g. what is a legal variable name?)
   - context-free languages describe syntax (whole program, whole sentence)

   Illustrate with your favorite example from programming languages or natural language.

10. State machines

    - widely used in other areas of CS (e.g. networking)
    - equivalent to regular languages (we will see later in course)
    - used to implement algorithms for regular languages e.g. `grep`.

    Illustrate with your favorite simple state machine, e.g. a vending machine.

11. Decidability:

    - Are there problems that computers can not solve? ⇒ yes!
    - By the end of the course, you will know why.
      Example: the CS 225 grader problem.

- Given a random `C` program (maybe very badly written).
- Will it stop or will it keep running forever?
- Will it return the right answer for all possible inputs?

12. Models of mathematics

    - 19th century - major effort to formalize calculus.
    - Georg Cantor - starts set theory. Proves that the "number" of integers is strictly smaller than the number of integers, using the diagonalization argument.
    - David Hilbert (1920's) tries to formalize all of math and prove it correct
    - Kurt Gödel (1931) shows that one can not prove consistency of a mathematical formalism having non-trivial power.

13. Formal models of computation

    - Alonzo Church: lambda calculus (like LISP).
    - Alan Turing: Turing machines (very simple computers).
    - Church/Turing thesis: these models can do anything that any computer can do.
    - Both showed (1936) that their respective models contain undecidable problems.

14. It is mysterious and "cool" that some simple-looking problems are undecidable.

15. The proofs of undecidability are a bit abstract. Earlier parts of the course will help prepare you, so you can understand the last part.

# 2 Necessary Administrivia

This lecture mentions the highlights. Browse the web page for more details:

`http://www.cs.uiuc.edu/class/sp09/cs373/{}`.

- Prerequisites: CS 125, CS 173, CS 225 (or equivalents). Other experience can sometimes substitute (e.g. advanced math). Speak to us if you are not sure.

- Vital to join the class newsgroup (details on web page). Carries important announcements, e.g. exam times, hints and corrections on homeworks.

  Especially see the Lectures page for schedule of topics, readings, quiz/exam dates.

- Homework 1 should be available on the class website. Due next Thursday. (Normally they will be due Thursdays on 12:30, but next Monday is a holiday.) Browse chapter 0 and read section 1.1. Normally, homeworks and readings will not be announced in class and you must watch the website and newsgroups.

- Read and follow the homework format guidelines on the web page. Especially: each problem on a separate sheet, your name on each problem, your section time (e.g. 10) in upper-right corner. This makes a big difference grading and sorting graded homeworks.

- Course staff.
- Discussion sections. Office hours will be posted in the near future. Email and the newsgroup are always an option. Please do not be shy about contacting us.
- Problem sets, exams, etc are common to all sections. It may be easier to start with your lecture and discussion section instructors, but feel free to also talk to the rest of us.
- Sipser textbook: get a copy. We follow the textbook fairly closely. Our lecture notes only outline what was covered and don't duplicate the text. Used copies, international or first editions, etc are available cheap through Amazon.
- Graded work:
  (a) 30%: Final.
  (b) 20%: First midterm.
  (c) 20%: Second midterm.
  (d) 25%: Homeworks and self-evaluations.
  Worst homework will be dropped.
  Self evaluations would be online quizes on the web.
  (e) 5%: Attending discussion section.
- Late homeworks are not accepted, except in rare cases where you have a major excuse (e.g. serious illness, family emergency, weather unsafe for travel).
- Homeworks can be done in groups of $\leq 3$ students. Write their names under your own on your homework. Also document any other major help you may have gotten. Each person turns in their own write-up **IN THEIR OWN WORDS**.
- Doing homeworks is vital preparation for success on the exams. Getting help from your partners is good, but don't copy their solutions blindly. Make sure you understand the solutions.
- See the web pages for details of our cheating policy. First offense $\rightarrow$ zero on the exam or assignment involved. Second offense or cheating on the final $\Rightarrow$ fail the course. Please do not cheat.
- If you are not sure what is allowed, talk to us and/or document clearly what you did. That is enough to ensure it is not "cheating" (though you might lose points).
- Bugs happen, on homeworks and even in the textbook and on exams. If you think you see a bug, please bring it to our attention.
- Please tell us if you have any disabilities or other special circumstances that we should be aware of.