CS 373: Theory of Computation

Manoj Prabhakaran Mahesh Viswanathan

Fall 2008

---

1 Staff, and Office Hours

Instructional Staff

- *Instructors:*
  - Manoj Prabhakaran (`mmp`)
  - Mahesh Viswanathan (`vmahesh`)
- *Teaching Assistants:*
  - Micah Hodosh (`mhodosh2`)
  - Pavithra Prabhakar (`pprabha2`)
  - Aparna Sundar (`sundar2`)
- *Office Hours:*
  - Manoj: Tuesday 13:45 – 14:45, and by appointment
  - Mahesh: Thursday 10:00 – 11:00, and by appointment
  - TAs: To be announced; see course webpage

---

2 Resources

Electronic Bulletin Boards

- *Webpage:* `www.cs.uiuc.edu/class/fa08/cs373`
- *Newsgroup:* `uiuc.class.cs373`

Textbooks

- *Prerequisites:* All material in CS 173, and CS 225
- *Text-book:* Introduction to the Theory of Computation by Michael Sipser
- *Lecture Notes:* Available on the web-page after every class
- *Additional References*
  - Introduction to Automata Theory, Languages, and Computation: Hopcroft, Motwani, and Ullman
  - Elements of the Theory of Computation: Lewis, and Papadimitriou
  - Computers and Intractability: Garey and Johnson

---

3 Grading Scheme

Grading Policy: Overview
Total Grade and Weight

- *Homeworks:* 20%
- *Self Assessment Tests:* 10%
- *Midterms:* 40% (2×20)
- *Finals:* 30%

Homeworks

- One homework every week: Assigned on Thursday and due the following Thursday (at 1PM in 3229 Siebel Center). Lowest homework score will be dropped.
- Homeworks "turned in" in groups of size at most 3
- Homeworks will be "turned in" orally every third week; the rest of the times you will turn in a written homework
  - *Oral:* Explain (verbally) to TA the solution to problems that are asked for
  - *Written:* Write solutions to every problem and turn in the written solutions in class
- Read Homework Guidelines on course website

Homework Groups

- Decide the groups in which you will turn in homeworks by Tuesday (September 2)
  - Mail who is in your group to Micah Hodosh (`mhodosh2`)
  - Anyone who does not mail will be assumed to be working alone on the homeworks
- We will allow you to change your groups after the first midterm and second midterm
- We will announce which groups will present orally

Self Assessment Tests

- Roughly every two weeks, containing multiple choice questions
- Will be online on Compass
- *Goal:* To help you ascertain your understanding of the material

- Everyone gets full points for completing the test

Examinations

- First Midterm: October 2, 7PM to 9 PM in room MH 103
- Second Midterm: November 4, 7PM to 9PM in room MSEB 100
- Midterms will only test material since the previous exam
- Final Exam will test *all* the course material

---

4 What is Computation?

Computation

- Solving problems step by step
- No magic oracle to divine the solution
  - Computation: going from a problem instance to its solution
  - By a pre-defined machine (a.k.a program, algorithm) using small steps
- Same machine for all instances

Computation is more than defining the problem

- Through the computational lens, mathematical problems look fundamentally different
- Not all problems are equally easy to solve
  - More *complex* problems take longer, no matter how clever you are!
- Not all problems can be solved!
  - Even when the solution is well-defined mathematically!

Strands of the Tale

- What is the nature of infinity?
- What enables humans to communicate through language?
- How does the human brain work?

---

5 Foundations of Mathematics and Computation

5.1 Hilbert's Program

Infinitum actu non datur

There are no *actual* infinities; only *potential* infinities.

> "I protest against the use of infinite magnitude as something completed, which is never permissible in mathematics. Infinity is merely a way of speaking, the true meaning being a limit which certain ratios approach indefinitely close, while others are permitted to increase without restriction.": Gauss

> "There are more primes than in any given collection of prime numbers": Euclid

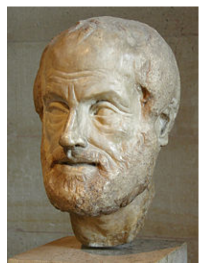


Figure 1: Aristotle

To Infinity and Beyond!

Georg Cantor (1845–1918)

- Laid the foundations of the theory of infinite sets
- Developed the theory of infinite ordinals (numbers)
- Showed how the size of (infinite) sets could be measured
  - Showed there were more real numbers than natural numbers
  - Presaged ideas that would later show that very few problems can actually be solved computationally

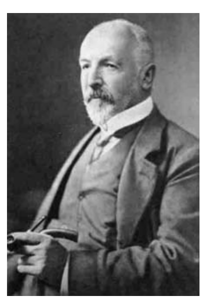


Figure 2: Georg Cantor

Devil or Messiah?

- Cantor's work received widespread opposition during his time
  - Theologians saw Cantor's work as a challenge to the uniqueness of absolute infinity in the nature of God
  - Poincaré called it the "great disease" infecting mathematics
  - Kronecker called Cantor a "charlatan", a "renegade", and a "corrupter of youth"!
- But, it also had some supporters

  "No one shall expel us from the Paradise created by Cantor": Hilbert

Crisis in Set Theory

Bertrand Russell (1872–1970)

- Mathematician, philosopher, writer, and political activist who won the Nobel prize in literature!
- Discovered disturbing paradoxes in Cantor's theory

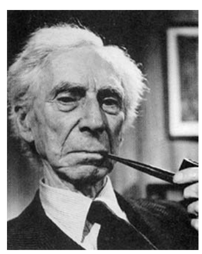


Figure 3: Bertrand Russell

Russell's Paradox

- Riddle: "In a small village, the barber shaves all the men who do not shave themselves (and only them). Does the barber shave himself?"
- Russell: "Consider the set A of all sets that are not members of themselves. Is A a member of itself?"

Axiomatic Method to the Rescue

David Hilbert (1863–1943)

- Solution to the crises: Formalism that avoids the paradoxes
  - Define concepts precisely
  - Define *axioms* and *rules of inference* that can be used to write down formal proofs

Figure 4: David Hilbert

Formal Proofs
Euclid of Alexandria (around 300 BCE)

- Euclid's Elements sets out
  - Axioms (or postulates), which are *self evident truths*, and
  - Proves all results in geometry from these truths formally

Figure 5: Euclid of Alexandria

Euclid's Postulates

A1 A straight line can be drawn from any point to any point.

A2 A finite line segment can be extended to an infinite straight line.

A3 A circle can be drawn with any point as center and any given radius.

A4 All right angles are equal.

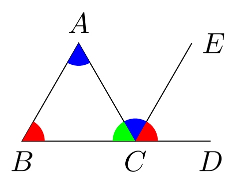
A5 If a straight line falling on two straight lines makes the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, will meet on that side on which the angles are less than two right angles.

Example of a Formal Proof
Elements: Proposition 32

Proposition 1. *The interior angles of a triangle sum to two right angles.*

Proof.

1. Extend one side (say) BC to D [A2]
2. Draw a line parallel to AB through point C; call it CE [P31]
3. Since AB is parallel to CE, BAC = ACE and ABC = ECD [P29]
4. Thus, the sum of the interior angles = ACB + ACE + ECD = 180^o □



Consistency and Completeness

Proof System
Precise definition of what constitutes a proof — each line is an axiom, or is derived from previous lines by rule of inference.

- Correctness of proof reduced to checking if it follows the rules of the system; no ambiguity!

Consistency
A proof system is *consistent* if it only allows true statements to be proved; no false conclusions.

Completeness
A proof system is *complete* if every true statement has a proof that adheres to its rules.

Agenda for the 20th Century
Hilbert's Paris Lecture (1900)

- Suggested 23 open problems to be investigated in the 20th century; some remain open to this day!
- One was to obtain a consistent and complete proof system for mathematics — axioms and rules that will allow all (and only) mathematical truths to be proved

---

5.2 Gödel's Insight

Shocking Discovery
Kurt Gödel (1906–1978)

- In 1930, at the annual meeting of the Society of German Scientists and Physicians, Hilbert said of his program, "We must know. We will know."
- At one of the satellite conferences of the same meeting, Kurt Gödel pronounced that Hilbert's program was fated to fail!

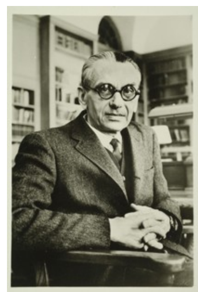


Figure 6: Kurt Gödel

Gödel's Insight

Incompleteness Theorem: Gödel showed that given any consistent proof system for number theory, one can construct a statement about numbers that is true but cannot be proved!

- Relied on the Liar's Paradox which says "This statement is false."
- To get incompleteness, Gödel constructed the statement "This statement is unprovable" inside number theory!
- Proof is a hacker's dream!

5.3 Mechanized Computation

Mechanized Computation

- The heart of the axiomatic method, is that proof correctness reduces to mechanical checking.
- But what is mechanical checking?
- What are limits of mechanical computation?

The Computer Revolution

Alonzo Church, Emil Post, and Alan Turing (1936)

Figure 7: Alonzo Church

Figure 8: Emil Post

Figure 9: Alan Turing

- Church (λ-calculus), Post (Post's machine), Turing (Turing machine) independently come up with formal definition of mechanical computation that are equivalent
- Discovered problems that cannot be solved computationally

Berry's Paradox: Russell

"The smallest natural number that cannot be defined in less than ninety characters"

- The above sentence describes a number as there are only finitely many strings with less than 90 characters
- The sentence itself has only 81 characters, and describes a number that cannot be described by less than 90 characters!

Busy Beaver Function

- *Simple Program:* A program (in say C) that takes no inputs, computes, prints some number and stops.
- *Busy Beaver Function:* $BB(n)$ is the largest number printed by a simple program of less than n bytes

---

An uncomputable function!

Theorem 2. *There is no C program that given number n computes $BB(n)$.*

- Suppose (for contradiction) there is a C function `compBB` to compute $BB(n)$. Let `compBB` be of size `m` bytes.
- Consider the following program:

```
int compBB (int n) {
    ......
}
void main() {
    printf(``%d'',compBB(5000+2*m)+1);
}
```

- Size of above program $\leq \mathtt{m} + 5000 + \log \mathtt{m}$ but prints a number $> BB(5000 + 2 * \mathtt{m})$! Contradiction!

Spectacular Failure of Hilbert's Program

Uncomputable functions provide more examples of incompleteness

- Suppose there is a consistent proof system that for every value of n proves that $BB(n) = k$ for some k.
- The function `compBB(n)` simply goes through all strings one by one, till it comes across a string that is a correct proof of $BB(\mathtt{n}) = \mathtt{k}$.

---

6 Cognitive Revolution and Grammars

6.1 Language Acquisition and Verbal Behavior

The Problem of Language Acquisition

- Language is an important human cognitive process that allows us to share information, thoughts and subjective experiences
- Though language is complex, it is acquired and used skillfully by children
- What is the mechanism behind its acquisition and use?

Behaviorism

- All things that organisms do — actions, thinking, feeling — are behaviors, in response to sensory input
- Behaviors are the only measurable things
- Scientific description of such behavior should not rely on internal physiological events or hypothetical constructs such as the mind

---

Verbal Behavior

Burrhus Skinner (1904–1990)

- The child's mind is a blank slate, and language is learned
- The learning process is a gradual change based on sensory input provided to the organism
- Thinking is a form of "verbal behavior"

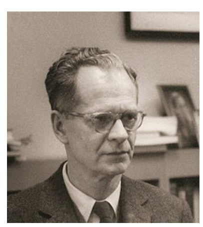


Figure 10: B.F. Skinner

6.2 Chomsky's Critique

Critique of Verbal Behavior
Noam Chomsky (1928–)

Behaviorist account is flawed because the underpinnings of natural language are highly abstract principles, and children acquire language without explicit instruction or environment clues to these principles.

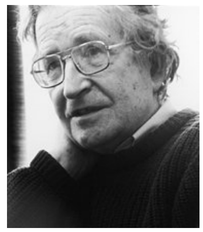


Figure 11: Noam Chomsky

Humboldt: "infinite use of finite media"
Too many sentences to learn

> "A person capable of producing sentences with up to 20 words, can deal with at least 10^{20} sentences. At the rate of 5 seconds per sentence, she would need 100 trillion years (with no time for eating or sleeping) to memorize them.": Pinker

---

Humboldt: "infinite use of finite media"
Recursive build-up

- The longest English sentence in the Guinness Book of World Records is a 1,300 word sentence in Faulkner's Absalom, Absalom! that begins "They both bore it as though . . . "
- Steven Pinker thought of submitting the following record breaker: Faulkner wrote, "They both bore it as though . . . "
- But that could be easily broken by: Pinker wrote that Faulkner wrote, "They both . . . "

Grammatical Correctness independent of Cognition
Ungrammatical Comprehensible Sentences

- The child seems sleeping.
- It's flying finches, they are.
- Sally poured the glass with water.
- Who did a book about impress you?

- This sentence no verb.

Grammatical Correctness independent of Cognition
Grammatical Incomprehensible Sentences

- Chomsky: "Colorless green ideas sleep furiously."
- Edward Lear: "It's a fact the whole word knows, That Pobbles are happier without their toes."

6.3 The Cognitive Revolution

Generative and Universal Grammars
Chomsky's Solution

The key to language acquisition is learning a *Generative Grammar* for the language that describes

- Word categories (like nouns, verbs, etc.)
- Rules determining how categories are put together

Chomsky's theory: A core *Universal Grammar* is innate to all humans.

Chomsky Hierarchy

Chomsky found grammars of different complexity conveniently describe various aspects of languages. Echoed in compiler design

- Lexical tokens described using "regular expressions"
- Language syntax described using "context-free grammars"

---

7 Modeling the Brain and Automata

7.1 Neural Nets

The Human Brain

- Central organ in the body that controls and regulates all human activity
- Believed the seat of "higher mental activity": thought, reason and abstraction
- How does it work?

Neurons

Santiago Ramón y Cajal (1852–1934)

- Neurons are the primary functional unit of the brain and the central nervous system
- Neurons receive information at *dendrites* and transmit via *axons*
- They communicate with each other through junctions called *synapses*

Figure 12: Santiago Ramón y Cajal

Mathematical Model of Neural Nets

McCullough and Pitts (1943)

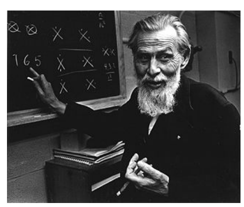


Figure 13: Warren Sturgis McCullough

Figure 14: Walter Pitts

- Came up with a mathematical model of a neuron
- Motivation to compare the computational power of networks of neurons with Turing Machines

Simple Neural Net

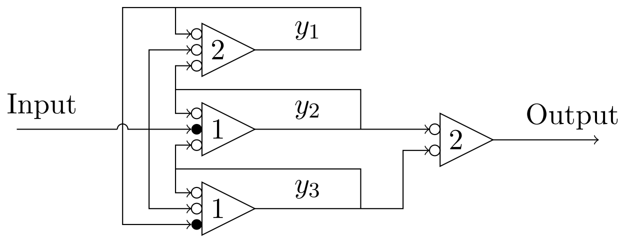


Figure 15: A Simple Neural Net

- Neurons have excitatory (circles) or inhibitory (dots) synapses
- Neuron produces 1 if the number of excitatory synapses with 1-input exceeds the number of inhibitory synapses with 1-input by at least the threshold of the neuron (number inside triangle)

7.2 Automata and Nondeterminism

Finite State Automata

- Finite automata model introduced by Huffman (1954), Moore (1956) and Mealy (1955) to model sequential circuits
- *Kleene, 1956*: Neural Nets of McCullough-Pitts the same as Finite Automata

Nondeterminism
Michael Rabin and Dana Scott (1959)

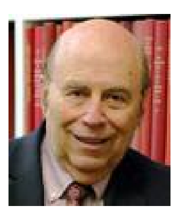


Figure 16: Michael Rabin

Figure 17: Dana Scott

- Introduced the notion of Nondeterministic Finite Automata
- Understanding the power of nondeterministic computation has remained a fundamental problem since then

---

8 Course Overview

Three Tales of Computation

- What is the nature of infinity?
- What enables humans to communicate through language?
- How does the human brain work?

Common to these diverse threads of science is computation.
They laid the foundations of a theory that we will explore.

Course Overview

Goal
Understand problems in terms of how complex it is to computationally solve them (or, if they can be solved at all!)

- A fundamental scientific/mathematical question
- Foundation for the science of computationally solving problems
  - Use simpler computational models for simpler problems
  - Use alternate notions of "solving" if problem is too complex
    - Approximate solutions, probabilistic solutions, partial solutions, ...

Course Overview

The three main computational models/problem classes in the course

---

Computational Model	Applications
Finite State Machines/ Regular Expressions	text processing, lexical analysis, protocol verification
Pushdown Automata/ Context-free Grammars	compiler parsing, software modeling, natural language processing
Turing machines	undecidability, computational complexity, cryptography