CS 373: Theory of Computation

Mahesh Viswanathan
vmahesh@cs.uiuc.edu
3232 Siebel Center

University of Illinois, Urbana-Champaign

Fall 2010

Part I

Administrivia

Instructional Staff

- Instructor:
  - Mahesh Viswanathan (`vmahesh`)
- Teaching Assistants:
  - Joe Di Febo (`difebo1`)
  - Sridhar Duggirala (`duggira3`)
  - Stephen Graessle (`sdgraes2`)
- Office Hours: See course webpage

Electronic Bulletin Boards

- Webpage: `courses.engr.illinois.edu/cs373`
- Newsgroup: We will use Piazza. Sign up at `piazza.com/illinois/fall2012/cs373`. Piazza discussion page is `piazza.com/illinois/fall2012/cs373/home`

Resources for class material

- Prerequisites: All material in CS 173, and CS 225
- Lecture Notes: Available on the web-page
- Additional References
  - Introduction to the Theory of Computation: Michael Sipser
  - Introduction to Automata Theory, Languages, and Computation: Hopcroft, and Ullman
  - Introduction to Automata Theory, Languages, and Computation: Hopcroft, Motwani, and Ullman
  - Elements of the Theory of Computation: Lewis, and Papadimitriou

Grading Policy: Overview

Total Grade and Weight

- Homeworks: 20%
- Quizzes: 10%
- Midterms: 40% (2×20)
- Finals: 30%

Homeworks

- One homework every week: Assigned on Thursday and due the following Thursday (midnight in homework drop boxes)
- No late homeworks. Lowest two homework scores will be dropped.
- Homeworks may be solved in groups of size at most 3.

Homeworks

- One homework every week: Assigned on Thursday and due the following Thursday (midnight in homework drop boxes)
- No late homeworks. Lowest two homework scores will be dropped.
- Homeworks may be solved in groups of size at most 3.
- Read Homework Guidelines on course website.

Quizzes

- The day before every class on Moodle.
- About 25 to 26 in total.
- We will drop the 5 lowest scores.

Examinations

- First Midterm: October 4, 7pm to 9pm
- Second Midterm: November 8, 7pm to 9pm
- Final Exam: December 18, 8am to 11am

Examinations

- First Midterm: October 4, 7pm to 9pm
- Second Midterm: November 8, 7pm to 9pm
- Final Exam: December 18, 8am to 11am
- Midterms will only test material since the previous exam
- Final Exam will test all the course material

Part II

Three Tales of Computation . . .

The three Tales

- What is the nature of infinity?
- How do we learn language?
- How does the human brain work?

The three Tales

- What is the nature of infinity?
- How do we learn language?
- How does the human brain work?

These diverse threads of scientific enquiry led to an understanding of computation.

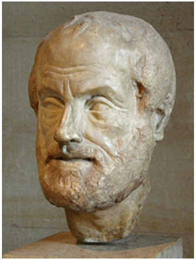
The first story

Understanding Infinity

One machine to solve them all . . .

Infinitum actu non datur

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

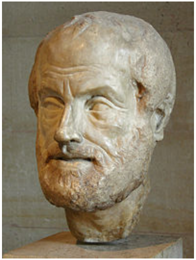


Aristotle

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*



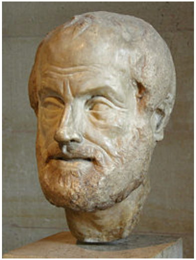
Aristotle

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*



Aristotle

To Infinity and Beyond!

Georg Cantor (1845–1918)

- Laid the foundations of the theory of infinite sets

Georg Cantor

To Infinity and Beyond!

Georg Cantor (1845–1918)

- Laid the foundations of the theory of infinite sets
- Developed the theory of infinite ordinals (numbers)

Georg Cantor

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

Georg Cantor

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

Georg Cantor

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

Georg Cantor

Devil

- 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"!

Devil or Messiah?

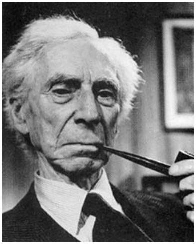
- Cantor's work received widespread opposition during his time
  - Theologians saw Cantor's work as a challenge to the uniqueness of absolute infinity in the nature of God
  - Poincaré called it the "great disease" infecting mathematics
  - Kronecker called Cantor a "charlatan", a "renegade", and a "corrupter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



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'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

David Hilbert

Axiomatic Method to the Rescue

David Hilbert (1863–1943)

- Solution to the crises: Formalism that avoids the paradoxes
  - Define concepts precisely

David Hilbert

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

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

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

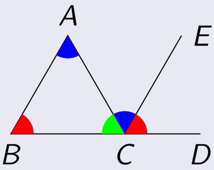
Elements: Proposition 32

Proposition

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 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.

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

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!

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

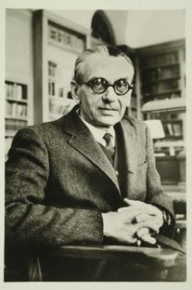
Shocking Discovery

- 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."

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!



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!

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."

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!

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!

Mechanized Computation

- The heart of the axiomatic method, is that proof correctness reduces to mechanical checking.

Mechanized Computation

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

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)

Alonzo Church

Emil Post

Alan Turing

- Church (λ-calculus), Post (Post's machine), Turing (Turing machine) independently come up with formal definition of mechanical computation that are equivalent

The Computer Revolution

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


Alonzo Church


Emil Post


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"

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

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

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

An uncomputable function!

Theorem

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.

An uncomputable function!

Theorem

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) {
    ......
}
```

An uncomputable function!

Theorem

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);
}
```

An uncomputable function!

Theorem

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

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.

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 $BB(\mathtt{n}) = \mathtt{k}$.

The second story

Understanding Language

One grammar to generate it all . . .

The Problem of Language Acquisition

- Language is an important human cognitive process that allows us to share information, thoughts and subjective experiences

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

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

Behaviorism

- All things that organisms do — actions, thinking, feeling — are behaviors, in response to sensory input
- Behaviors are the only measurable things

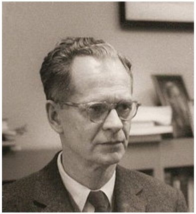
Behaviorism

- 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"

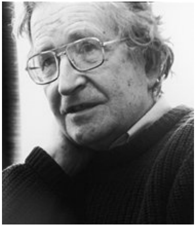


B.F. Skinner

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.



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 . . ."

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:

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 . . . "

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."

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.

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"

The third story

Understanding the Brain

One machine to think it all . . .

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

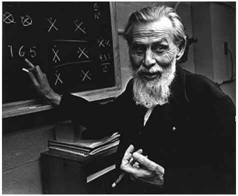
Santiago Ramón y Cajal (1852–1934)

- 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

Santiago Ramón y Cajal

Mathematical Model of Neural Nets

McCullough and Pitts (1943)

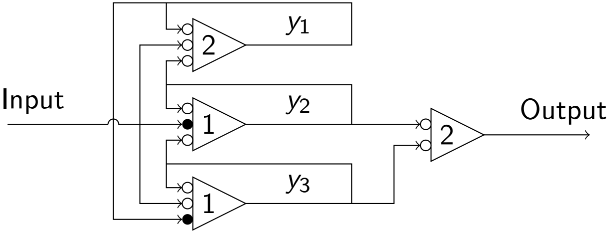


Warren Sturgis McCullough

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



A Simple Neural Net

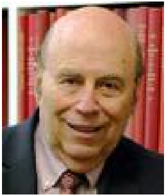
- Neurons have excitatory (circles) or inhibitory (dots) synapses
- Neuron produces 1 if the number of excitatory synapses with 1-input exceeds the number of inhibitory synapses with 1-input by at least the threshold of the neuron (number inside triangle)

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)



Michael Rabin

Dana Scott

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

Part III

Course Overview

Three Tales of Computation

- What is the nature of infinity?
- How do we learn language?
- How does the human brain work?

Computation underlies all these diverse scientific enquiries and they laid the foundations of a theory that we will explore.

Objectives

Understand the nature of computation

Objectives

Understand the nature of computation in a manner that is independent of our understanding of physical laws (or of the laws themselves)

Objectives

Understand the nature of computation in a manner that is independent of our understanding of physical laws (or of the laws themselves)

- Its a fundamental scientific question

Objectives

Understand the nature of computation in a manner that is independent of our understanding of physical laws (or of the laws themselves)

- Its a fundamental scientific question
- Provides the foundation for the science of computationally solving problems

Problems through the Computational Lens

Mathematical problems look fundamentally different when viewed through the computational lens

Problems through the Computational Lens

Mathematical problems look fundamentally different when viewed through the computational lens

- Not all problems equally easy to solve — some will take longer or use more memory, no matter how clever you are

Problems through the Computational Lens

Mathematical problems look fundamentally different when viewed through the computational lens

- Not all problems equally easy to solve — some will take longer or use more memory, no matter how clever you are
- Not all problems can be solved!

Problems through the Computational Lens

Mathematical problems look fundamentally different when viewed through the computational lens

- Not all problems equally easy to solve — some will take longer or use more memory, no matter how clever you are
- Not all problems can be solved!
- The "complexity" of the problem influences the nature of the solution

Problems through the Computational Lens

Mathematical problems look fundamentally different when viewed through the computational lens

- Not all problems equally easy to solve — some will take longer or use more memory, no matter how clever you are
- Not all problems can be solved!
- The "complexity" of the problem influences the nature of the solution
    - May explore alternate notions of "solving" like approximate solutions, "probabilistically correct" solutions, partial solutions, etc.

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

Skills

- Comprehend mathematical definitions
- Write mathematical definitions
- Comprehend mathematical proofs
- Write mathematical proofs