CS474
Logic in Computer Science
Spring 2026

Madhusudan Parthasarathy (Madhu)
TA: none, or perhaps someone

Lectures: Tue, Thu: 3:30pm-4:45pm
Venue: 1304 Siebel Center

Website: https://courses.grainger.illinois.edu/cs474/
Piazza: TBA

Prerequisites:
- Mathematical maturity; discrete math (CS173), especially proofs by induction.
- Theory of computation (as covered in CS374) is preferred, especially decidability, TMs.
- Talk to me if you don't have this background.

What's logic?

- Logic is the study of the principles of valid inference and demonstration. It is the study of correct reasoning.
- Logic is the study of the structure of arguments.
- *Formal* and *Informal* Logic
- Formal logic: study of formal inference; without appeals to human meaning of topics, etc.
- Formal Logic: Syntax + Semantics + Inference
  (form) (meaning) (reasoning)
- *"Contrariwise," … "if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic."*
  - Tweedledum - Through the Looking Glass, Lewis Carroll

Aims of this course

- Primary aims:
  - Mathematical and logical maturity;
    understanding how humans and computers reason formally
  - Formal reasoning and formal arguments
  - How computers can perform automated reasoning
  - Various other ways in which logic is a foundational aspect of computer science

Aims of this course

- More concretely:
  - Fundamentals of mathematical logic
    - Propositional logic and predicate logic
    - Model theory and proof theory
    - Sound and complete proof systems
    - Gödel’s completeness and incompleteness theorems
  - Logic and computability
    - Computability ~ Proofs
    - Finite model theory (computational complexity using logic)
  - Automated reasoning
    - Propositional reasoning using SAT solvers
    - Quantifier elimination
    - Decidability using interpretations
    - Decidable theories supported by SMT solvers
    - Decidability of combined theories
    - Complete reasoning using systematic quantifier instantiation
    - Reasoning using induction

A brief history of logic

- Beginnings
  - Aristotle (~300 B CE)
    - Earliest formal study of logic
    - Rhetoric, syllogism, philosophy, geometry
    - Furthered by Islamic logicians
  - Logic in India
    - Nyaya and Vaisheshika (2 CE)
    - Ontology and epistemiology; obtaining valid knowledge
  - Logic in China
    - Mozi school (~400 B CE)

A brief history of logic

- Syllogism
  - Every man is mortal
  - Socrates is a man
  - Conclude: Socrates is mortal

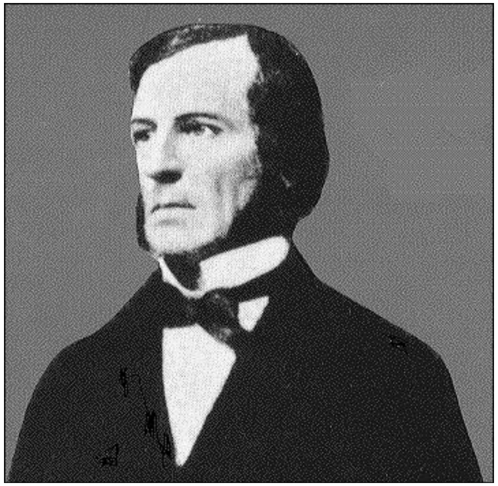
- “man”, “mortal”, “Socrates” – not important
  - Every snark is frubjuous.
  - Betty is a snark
  - Conclude: Betty is frubjous.

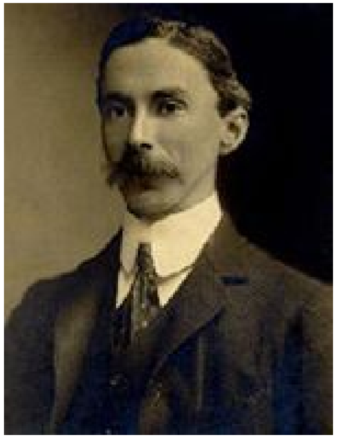
- Or more generally:
  - Every M is D
  - S is an M
  - Conclude: S is D

Forall x. M(x) -> D(x)
M(S)
Conclude: D(S)

A brief history of logic

- The algebraic school
  - George Boole and others (end of 19th century)
  - Algebraic structure of formulas; Boolean algebra





- The logicist school (Russel,Wittgenstein,Frege)
  - Frege developed an elaborate formal language

  Quantifiers, "predicate calculus", number-theory -> logic
  - Russel's paradox
  - Principia Mathematica (Russel-Whitehead) (1910-1913)

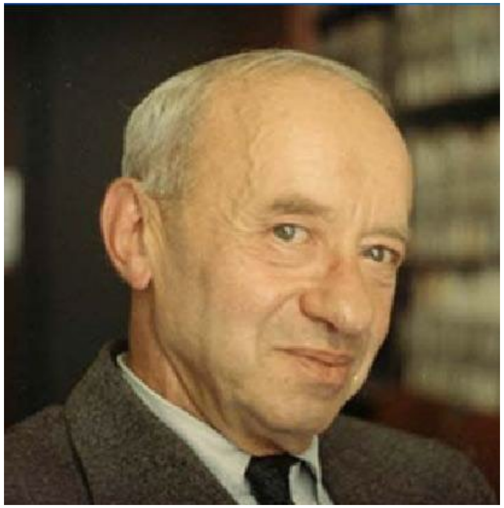
A brief history of logic

- The mathematical school (early 20th century)
  - Dedekind, Peano, Hilbert, Heyton, Zermelo, Tarski
  - Axiomatize branches of math (geometry, arithmetic, set-theory)
  - Zermelo: Axiomatization of set-theory
  - The Hilbert program: formalization of all of mathematics in axiomatic form, with a proof of consistency using "finitary" methods.
  - Godel's theorems:
    - Every FO sentence can be deduced in common deductive systems.

      Godel's completeness theorem
    - Any axiomatization that includes arithmetic must either be unsound or incomplete (i.e. there is a sentence neither provable nor disprovable)

      Godel's incompleteness theorem

A brief history of logic



- The mathematical school (20th century)
  - Tarski: Logician extraoradinaire
  - Completeness, truth, definability, decidability
  - Alonzo Church, and students Kleene and Henkin
    - Church's thesis of computability
    - First-order logic is undecidable

Post WW-II

- model theory
  - Study of mathematical structures (groups, etc.) using logic
- proof theory
  - Study of axiom systems, inference systems, proofs
- computability theory
  - Relationship of logic to computability/complexity
- set theory
  - foundations of mathematics, axiomatic set theory
  - Cohen: independence of continuum hypo and independence of axiom of choice from ZF axioms.
  - constructivism
- applications of logic in computer science
  - applied logic to AI, databases, verification, aided by tools like automated and semi-automated theorem provers, and ML

Logic and computability

- Turing's machines (1941):
  - Formal mechanical notion of computation using "finite" means
  - Church-Turing thesis:

    All computable languages are Turing-machine computable
  - Gödel's incompleteness result is about non-existence of proofs using diagonalization:
    - A proof system is only a particular form of computation!
    - Existence of a sound and complete proof system shows a theory is recursively enumerable (computed by TM that may not halt on all inputs)
    - A deep connection between logic and computability

Logic and computational complexity

- Fundamental idea (Fagin): Descriptive complexity
  - An algorithmic problem can be logically described.
  - The logic used determines the computational complexity of the problem!
  - Eg. Existential SO ~ NP
    Least-fix-point over ordered structures ~ P
  - Field: Finite model theory

Logic in Computer Science

Unusual (and unreasonable!) effectiveness in computer science.

- Computability and proofs
- Computational complexity and descriptive complexity
- Semantics of programming languages (types, type inference, domains and fixpoints, linear logic, etc.)
- Automatic reasoning, theorem provers (CoQ, PVS).
- Specifications for hardware systems (decidable temporal logics LTL/CTL)
- Logics to prove programs correct (Hoare-logic, pre-post conditions)
  - Huge impact in verifying systems
  - Modern software validation tools rely on decidable logics for abstraction and verification
  - Large number of decidable theories; combining decidable theories; automatic theorem provers (eg. Z3 from MSR).
- Proof-carrying code
- Database theory: Queries are simply formulas! SQL ~ FOL
- Logic Programming

Unusual effectiveness of logic in CS

- E.P. Wigner's Nobel prize talk, 1963:

  "On the Unreasonable Effectiveness of Mathematics in the Natural Sciences"

- On the Unusual Effectiveness of Logic in Computer Science

  - Halpern, Harper, Immerman, Kolaitis, Vardi, Vianu

Logic and AI

- AI in the past (till about 2000) was obsessed with formal logic, probably too obsessive.
- AI in 1960s and 1970s:
  Researchers convinced that symbolic approaches will eventually succeed in delivering AI.

  Simon('65): "machines will be capable, within twenty years, of doing any work a man can do"
  Minsky('67): ""within a generation ... the problem of creating 'artificial intelligence' will substantially be solved"

  Didn't happen! ☺ AI Winter.
  Many more remaining tasks… perception, learning, pattern recognition.

  Even playing "chess" doesn't fall into "formal reasoning" (see recent progress in AlphaGo/AlphaGoZero)
  Intelligence is not just formal/symbolic reasoning!
- AI in 1980s: Expert systems… again failed.
- AI in 2012 on: Deep learning. Huge advances in perception and learning/pattern recognition.
  - Vision, NLP, comprehension and question-answering, Reinforcement learning and AlphaGo/AlphaGoZero
  - Generative models. GPT3, ChatGPT, Dall-E
  - Particular problems in science: AlphaFold for protein folding
  - Applications: Self-driving cars

Logic and AI today

Lots of research!

- LLMs and ML models can produce proofs that are vetted
  - Aided by thousands of papers and proofs
  - Proofs in English (vetted by humans)
  - Formal proofs (vetted by theorem provers like Lean/Coq)
  - Combination of proofs and automation:
    Proof strategies by ML followed by automation for
    rest of the proof (e.g., AlphaGeometry for Math Olympiad)
- New theorems in math are now being proved by LLMs/ML models

  E.g., See Terence Tao's blog on AI assisted Erdos problems:

  https://github.com/teorth/erdosproblems/wiki/AI-contributions-to-Erd%C5%91s-problems

Last part of the course will discuss these advancements.

Projects can explore these topics!

Logic in Computer Science: Themes

Theme 1. Classical propositional and predicate/FO logic

(proof systems, axioms, soundness, completeness, compactness, Gödel's theorems)

Theme 2. Decidable logics

(QE for reals, finite-model property, decidability of Presburger arithmetic, interpretations, combining decision procedures, SAT/SMT solvers, tools for logic)

Theme 3. Finite model theory

(descriptive complexity, computational complexity, characterizing complexity classes using logic)

Theme 4. Other logics

Higher-order logics, modal and temporal logics, logics for querying (databases), relational logics, etc.

Administrivia

- What's "trivia"?
  - Greek education system had "trivium" (three ways):
    - Grammar (mechanics of language)
    - Logic (mechanics of thought and analysis)
    - Rhetoric (mechanics of persuasion)

    followed by the "quadrivium":
    - arithmetic, geometry, music, and astronomy

Administrivia

- Textbooks: No particular one (No text covers topics we cover)

  The following can be useful, however:
  - A new textbook we are writing (on website)
    Logic in Computer Science: P. Madhusudan, M. Viswanathan
  - A Mathematical Introduction to Logic: Enderton
  - Logic for Computer Scientists: Uwe Schöning
- More resources (book chapters, papers, notes) will be handed out in class and made available online on the course page.
- Lecture notes (handwritten scribbles) will be posted online

Administrivia

- Homework
  - One homework set every week , with some weeks skipped
- Office hours:
  - To be decided

Projects (for 4 credits)

Starts somewhere mid-term and ends with the semester.

- A project to read a set of papers and present it in class briefly and write a report
- A project that utilizes the course contents and applies to problems that require reasoning or learning
- A project that works on a research-ish problem and tries to build a solution

  (work in groups, work in something that's close to your research, I can work with you, you can even send it for publication if you get good results.)

Exams

- One midterm and a Final Exam
- Written exams

Other tools

- We will probably use Gradescope for homework; will explore peer-grading this semester (your work gets graded by peers; grading your peers is part of the grade).

- And Piazza for discussions.

- Sign up for Piazza! (will send email with code)

- Look out for emails on these topics.

Survey

Email on piazza:

1) What do you expect to learn in this course?

2) What are your general interests?

Grading

Tentative policy: (will get finalized soon)

- Attendance/participation: 10%
- Homework: 40% (for doing homework and peer grading)
- Midterm: 20%
- Final: 30%

- Project: 50% (and normalized with main course)

- Curved grading (for undergrads and grads separately)