Welcome to CS 4995 : Computability and Logic

Instructor : Toniann Pitassi (Toni)

TA : Oliver Korten

Webpage :

www.cs.columbia.edu/~toni/courses/Logic2021/4995.html

Email : toni@cs.columbia.edu

Contents

Toniann Pitassi
toni at cs dot toronto dot edu

Brief Bio

I received bachelors and masters degrees from Pennsylvania State University and then received a PhD from the University of Toronto in 1992. After that, I spent 2 years as a postdoc at UCSD, and then 2 years as an assistant professor (in mathematics with a joint appointmer in computer science) at the University of Pittsburgh. For the next four years, I was a faculty member of the Computer Science Department the University of Arizona. In the fall of 2001, I moved back to Toronto, as Professor in the Computer Science Department, with a joint appointment in Mathematics. In 2021 I joined the Department of Computer Science at Columbia University.

The above picture was taken in London in front of Bertrand Russell's flat. If you click on the picture to see an enlarged version, and then gc to the upper right quadrant, the blue sign mentioning this landmark will be legible.

Contents

- Research
- Publications
- Talks
- Teaching
- Students and Postdocs

Toniann Pitassi
toni at cs dot toronto dot edu

Teaching

CS4995F	Logic and Computability, 2021
CSC2541F	AI and Ethics: Mathematical Foundations and Algorithms
CSC2429	Proof Complexity, Mathematical Programming and Algorithms, Winter 2018
CSC165	Mathematical Expression and Reasoning for Computer Science, Winter 2018
CS2429	Proof Complexity, 2017
CSC 263	Data Structures and Analysis, Fall 2015
CSC2401	Introduction to Complexity Theory, Fall 2015
CSC 2429	Communication Complexity: Applications and New Directions, Fall 2014
CSC 2429	Approaches to the P versus NP Problem and Related Complexity Questions, Winter 2014
CSC 2429	Communication Complexity, Information Complexity and Applications, Fall 2013
CSC 2429	Foundations of Communication Complexity, Fall 2009
CSC 2402	Methods to Deal with Intractability, Fall 2009
CSC 2429	PCP and Hardness of Approximation, Fall 2007
CSC 448/2405	Formal Languages and Automata, Spring 2006
CSC 448/2405	Formal Languages and Automata, 2005
CSC 448/2405	Formal Languages and Automata, 2003
CSC 2416	Machine Learning Theory, Fall 2005
CSC 364	Computability and Complexity, Fall 2002
CSC 2429	Propositional Proof Complexity, Fall 2002
CSC 2429	Derandomization, Spring 2001

CS 4995: Computability and Logic
Fall, 2021

ANNOUNCEMENTS: (Students, please check for announcements every week.)

Posted Sept 13: Welcome to the class! Stay tunes for more announcements.

COURSE TIMES, CONTACT INFO

Instructor: Toniann Pitassi, email: toni@cs.columbia.edu
Office Hours: Monday 4-5pm
Lectures: MW 2:40-3:55, 415 Shapiro

TA: Oliver Korten

- **Course Information Sheet**

HOMEWORK ASSIGNMENTS:

- **Homework 1, Due Sept 27**

EXAM INFORMATION:

GRADES AND MARKING:

LECTURE NOTES:

- **Week 1**
- **Week 2**

COURSE NOTES:

- **Propositional Calculus**
- **Predicate Calculus**
- **Completeness**
- **Herbrand, Equality, Compactness**
- **Computability**
- **Incompleteness I**

CS 4995 – Fall 2021
Logic and Computability

Lectures: Monday/Wednesday 2:40-3:55, 415 Shapiro
Instructor: Toniann Pitassi, toni@cs.columbia.edu
Office hours: Monday 4-5
TA: Oliver Korten
Web Page: http://www.cs.columbia.edu/~toni/Courses/Logic2021/4995.html

Course Notes: Postscript files for course notes and all course handouts will be available on the web page.

Topics:

Propositional logic: syntax and semantics, Resolution and Propositional Sequent Calculus soundness and completeness. First order logic: syntax and semantics, First Order Sequent Calculus soundness and completeness. Godel's Incompleteness theorems. Computability: Recursive and recursively enumerable functions, Church's thesis, unsolvable problems

Marking Scheme:

3 assignments (each worth 20% of final grade)
First Term test (20% of final grade)
Second Term Test (20% of final grade)

Due Dates:

To be announced

The work you submit must be your own. You may discuss problems with each other; however, you should prepare written solutions alone.

Important

→ All lectures are mandatory.

Check CourseWorks -- some lectures may be held online via zoom.

- Work hard on understanding lecture notes, work hard on assignments
- start <u>early</u> -- cannot cram/solve in a couple of days
- Come to office hrs!
- Homeworks must be written up independently. You may discuss with other students in class but NO outside people/soures allowed.

COURSE INTRO

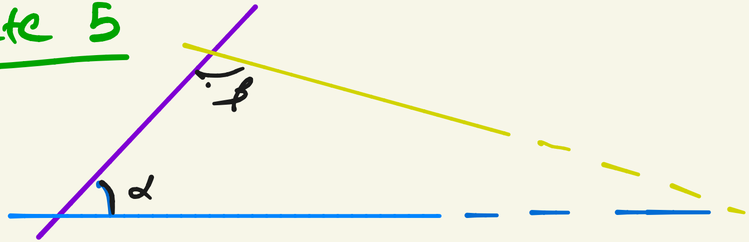
Foundations of mathematics involves the **axiomatic method** – write down axioms (basic truths) and prove theorems from axioms from purely logical reasoning

Example 1 Euclidean geometry (300 BC, "Elements")

The School of Athens, Rafael

Axiomatic system where all theorems are derivable from a small number of simple axioms/postulates

Postulate 5



If sum of $\alpha + \beta$ is < 180 then the 2 lines (blue + yellow) eventually meet (on same side as α, β angles)

Euclid's Postulates

1. A straight line segment can be drawn joining any two points.

2. Any straight line segment can be extended indefinitely in a straight line.

3. Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.

4. All right angles are congruent.

5. If two lines are drawn which intersect a third in such a way that the sum of the inner angles on one side is less than two right angles, then the two lines inevitably must intersect each other on that side if extended far enough. This postulate is equivalent to what is known as the parallel postulate.

Euclid's fifth postulate cannot be proven as a theorem, although this was attempted by many people. Euclid himself used only the first four postulates ("absolute geometry") for the first 28 propositions of the Elements, but was forced to invoke the parallel postulate on the 29th. In 1823, Janos Bolyai and Nicolai Lobachevsky independently realized that entirely self-consistent "non-Euclidean geometries" could be created in which the parallel postulate *did not hold.* (Gauss had also discovered but suppressed the existence of non-Euclidean geometries.)

SEE ALSO:
Absolute Geometry, Circle, Elements, Line Segment, Non-Euclidean Geometry, Parallel Postulate, Pasch's Theorem, Right Angle

REFERENCES:
Hofstadter, D. R. *Gödel, Escher, Bach: An Eternal Golden Braid.* New York: Vintage Books, pp. 88-92, 1989.

Referenced on Wolfram|Alpha: Euclid's Postulates

CITE THIS AS:
Weisstein, Eric W. "Euclid's Postulates." From *MathWorld*--A Wolfram Web Resource. https://mathworld.wolfram.com/EuclidsPostulates.html

Example 2 - Group Theory (Cayley, 1854)

axiom 1 : $\forall xyz\, [x\cdot(y\cdot z) = (x\cdot y)\cdot z]$ (associativity)

axiom 2 : $\exists u$

$[\forall x\, [x\cdot u = u\cdot x = u] \;\wedge$ there exists an identity element

$\forall x \exists y\, [x\cdot y = y\cdot x = u]]$ and every element has an inverse

---

A group is a model for the axioms

$(G, \cdot)$

$\cdot$: a function from $G\times G \to G$

G : a set

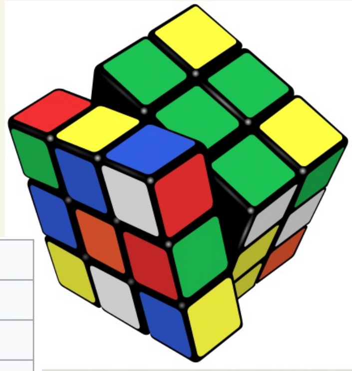
Examples of groups

1. $G = \mathbb{Z}$ (the integers) $\quad \bullet$ = addition

Examples of groups

① $G = \mathbb{Z}$ (the integers) $\quad \bullet =$ addition

② Rubik's cube group



Basic 90°	180°	-90°
F turns the front clockwise	F^2 turns the front clockwise twice	F' turns the front counter-clockwise
B turns the back clockwise	B^2 turns the back clockwise twice	B' turns the back counter-clockwise
U turns the top clockwise	U^2 turns the top clockwise twice	U' turns the top counter-clockwise
D turns the bottom clockwise	D^2 turns the bottom clockwise twice	D' turns the bottom counter-clockwise
L turns the left face clockwise	L^2 turns the left face clockwise twice	L' turns the left face counter-clockwise
R turns the right face clockwise	R^2 turns the right face clockwise twice	R' turns the right face counter-clockwise

basic moves

$G =$ all possible moves

$\bullet =$ composition of moves

Course Outline

We will study FIRST ORDER Logic (PREDICATE Logic)

I. Start with simpler PROPOSITIONAL Logic
(No quantifiers)

- Language of propositional logic ("syntax")
- Meaning ("semantics")
- Two proof systems for prop. logic:
  Resolution, and PK
- We will prove SOUNDNESS + COMPLETENESS
  for both

Course Outline (cont'd)

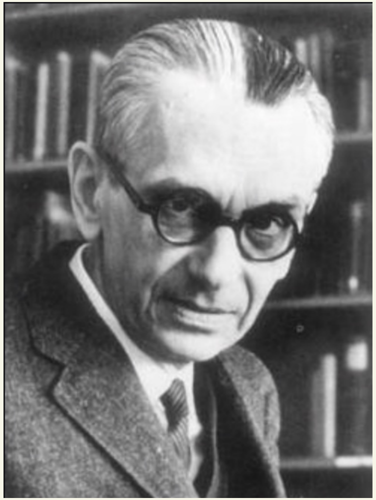
II. FIRST ORDER (PREDICATE) LOGIC

- Language ("syntax")
- Meaning ("semantics")
- Proof system LK (extends PK)

SOUNDNESS

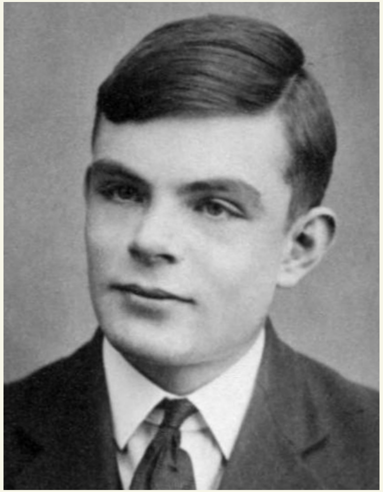
** COMPLETENESS

MAJOR COROLLARIES OF COMPLETENESS

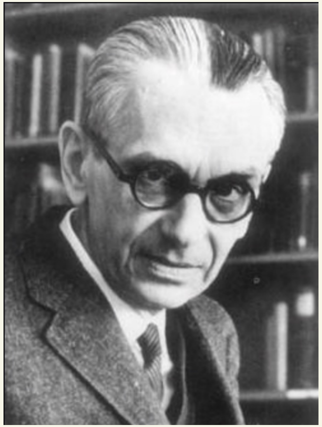


COURSE OUTLINE (cont'd)

III. Computability



IV. Axiomatizable Theories



Incompleteness Theorems

Interplay/connections between computability + logic

PROPOSITIONAL LOGIC

<u>Vocabulary</u>: $P_1, P_2, Q, \ldots$ propositional variables

$\neg, \vee, \wedge, (,)$

<u>Examples</u>: $((P \vee Q) \vee R)$

$(\neg P \vee \neg Q)$

PROPOSITIONAL LOGIC

Inductive Definition of a Propositional Formula

1. Atoms/Propositional variables: $P_1, P_2, \ldots$ are formulas
2. If A is a formula, then so is $\neg A$
3. If A, B are formulas, so is $(A \wedge B)$
4. " " " " " " $(A \vee B)$

$(A \supset B)$ is shorthand for $(\neg A \vee B)$

$(A \Leftrightarrow B)$ is shorthand for $(\neg A \vee B) \wedge (\neg B \vee A)$

A subformula of a formula is any substring of A which itself is a formula

<u>Unique Readability Thm</u> says the grammar for generating formulas is not ambiguous

Semantics

A **truth assignment** $\tau : \{\text{atoms}\} \longrightarrow \overset{\text{true}}{T}, \overset{\text{false}}{F}$

Extending τ to every formula:

(1) $(\neg A)^\tau = T$ iff $A^\tau = F$

(2) $(A \wedge B)^\tau = T$ iff $A^\tau = T \wedge B^\tau = T$

(3) $(A \vee B)^\tau = T$ iff either $A^\tau = T$ or $B^\tau = T$

Example

Definitions

τ **satisfies** A iff $A^\tau = T$

τ **satisfies** a set Φ of formulas iff
τ satisfies A for all $A \in \Phi$

Φ is **satisfiable** iff $\exists \tau$ that satisfies Φ
otherwise Φ is **unsatisfiable**

$\Phi \models A$ (A **is a logical consequence of** Φ) iff
$\forall \tau\, [\,\tau \text{ satisfies } \Phi \Rightarrow \tau \text{ satisfies } A\,]$

$\models A$ (A is **valid** or A is a **tautology**) iff
$\forall \tau\, [\tau \text{ satisfies } A]$

Examples

1. $(A \wedge B) \vDash (A \vee B)$

2. $\vDash (A \vee \neg A)$

3. $\{(A \vee B), (\neg A \vee B)\} \vDash B$

4. $A \vee B \nvDash B$

Some easy facts (check them)

1. If $\Phi \vDash A$ and $\Phi \cup \{A\} \vDash B$ then $\Phi \vDash B$

2. $\Phi \vDash A$ iff $\Phi \cup \{\neg A\}$ is unsatisfiable

3. A is a tautology iff $\neg A$ is unsatisfiable

Equivalence

A and B are equivalent (written $A \Leftrightarrow B$)
iff $A \models B$ and $B \models A$

Examples

1. $(A \wedge B) \overset{?}{\Leftrightarrow} (B \wedge A)$

2. $(\neg A \vee B) \overset{?}{\Leftrightarrow} (\neg B \vee A)$

Resolution : Proof System for Prop Logic

- Resolution is basis for most automated theorem provers
- Proves that formulas are unsatisfiable (recall F is a tautology iff $\neg F$ is valid)
- Formulas have to be in a special form : CNF

$$\underbrace{(x_1 \vee x_2 \vee \bar{x}_3)}_{C_1} \wedge \underbrace{(\bar{x}_2 \vee x_4)}_{C_2} \wedge \underbrace{(\bar{x}_4)}_{C_3} \wedge \underbrace{(x_1 \vee x_3)}_{C_4} \wedge \underbrace{(x_1)}_{C_5}$$

Converting a formula to CNF

- obvious method (deMorgan) could result in an exponential blowup in size

  Example $(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee (x_5 \wedge x_6) \vee \dots (\quad)$

- Better method : SAT THEOREM

  There is an efficient method to transform any propositional formula F into a CNF formula g such that F is satisfiable iff g is satisfiable

SAT THEOREM: proof by example

$$F: \quad \underbrace{\underbrace{(Q \wedge R)}_{P_B} \vee \neg Q}_{P_A}$$

new variables

$$g: \quad (P_B \Leftrightarrow (Q \wedge R)) \wedge (P_A \Leftrightarrow P_B \vee \neg Q) \wedge (P_A)$$

$$(\neg P_B \vee Q)\,(\neg P_B \vee R)\,(\neg Q \vee \neg R \vee P_B)$$