# Welcome to CSC 438/2404 !

Instructor : Toniann Pitassi (Toni)

TA : Noah Fleming

Webpage : www.cs.toronto.edu/~toni/courses/438-2019/438.html

## Contents




*Toniann Pitassi*
*toni at cs dot toronto dot edu*


# Toniann Pitassi's Webpage



## Toniann Pitassi

Professor,
Bell Canada Chair in
Information Systems
Department of Computer
Science
University of Toronto
**Email:** toni at cs dot toronto dot
edu
**Phone:** (416) 978-3695
**Fax:** (416) 978-1931
**Address:**
Sandford Fleming Building
10 King's College Road, Room
2305A
Toronto, Ontario
CANADA
M5S 3G4


### 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 appointment in computer science) at the University of Pittsburgh. For the next four years, I was a faculty member of the Computer Science Department at the University of Arizona. In the fall of 2001, I moved back to Toronto, where I am currently a professor in the Computer Science Department, with a joint appointment in Mathematics.

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 go to the upper right quadrant, the blue sign mentioning this landmark will be legible.

## Contents




*Toniann Pitassi*
*toni at cs dot toronto dot edu*


## Teaching

| | |
|---|---|
| CSC438F/2404F | Logic and Computability, 2019 |
| 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 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 |

# CSC 438F/2404F: Computability and Logic
# Fall, 2019

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

**Posted on Aug 24: The first class is Monday Sept 9, 2019.**

**COURSE TIMES, CONTACT INFO**

**Instructor: Toniann Pitassi, email: toni@cs**
**Office Hours: Monday 5:15-6pm, Sandford Fleming 2305A**
**Lectures: Monday 3-5 BA 1200**
**Tutorial: Friday 12-1 BA 1200**

**Tutor: Noah Fleming, noahfleming@cs**
**Noah's Office Hours: to be announced soon**

- **Course Information Sheet**

**HOMEWORK ASSIGNMENTS:**

- **Homework 1, Coming Soon**

**GRADES AND MARKING:**

- **Coming Soon**

**COURSE NOTES:**

- **Propositional Calculus**
- **Predicate Calculus**
- **Completeness**
- **Herbrand, Equality, Compactness**

# CSC 438F/2404F – Fall 2019
# Computability and Logic

**Exclusions:** MAT 309H1, PHL348H1
**Prerequisites (ugrads):** (CSC363H1/CSC463H1)/CSC365H1/CSC373H1/CSC375H1/MAT247H1
**Lectures:** Monday 3-5, BA 1200
**Tutorial:** Friday 12-1, BA 1200
**Instructor:** Toniann Pitassi, toni@cs.toronto.edu
**Office hours:** Monday 5:10-6, SF2305A
**Tutor:** Noah Fleming, SF 4306, noahfleming@cs.toronto.edu
**Web Page:** http://www.cs.toronto.edu/ toni/Courses/438-2019/438.html

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

**Topics:**

Syntax and semantics of the propositional and predicate calculus, completeness of Gentzen proof systems, formal theories, nonstandard models, and the Godel Incompleteness Theorems. Recursive and primitive recursive functions, Church's thesis, unsolvable problems, recursively enumerable sets.

**Marking Scheme:**

Class attendance/participation (2% of final grade)
4 assignments (each worth 12% of final grade)
First Term test (25% of final grade)
Second Term Test (25% of final grade)

**Due Dates:**

First Term Test: Monday Oct 21, 3-5pm BA 1200
Second Term Test: Thursday Dec 5, 3-5pm BA 1200
Assignment 1 due date: Friday Sept 27 12pm, before tutorial
Assignment 2 due date: Friday Oct 18 12pm, before tutorial
Assignment 3 due date: Friday Nov 1 12pm, before tutorial
Assignment 4 due date: Friday Nov 29 12pm, before tutorial

Assignments are due at the *beginning* of class, since solutions will be discussed during the beginning of class/tutorial.

*The work you submit must be your own.* You may discuss problems with each other; however, you should prepare written solutions alone. Copying assignments is a serious academic offence and will be dealt with accordingly.

**Supplementary References:**

S Buss: Chapter I: An introduction to proof theory, in **Handbook of Proof Theory**, S Buss Ed., Elsevier, 1998, pp1-78. (grad)
J Bell and M Machover: **A Course in Mathematical Logic**. North-Holland, 1977. (grad)
H.B. Enderton, **A Mathematical Introduction to Logic** (undergrad)
G Boolos and R.C. Jeffrey, **Computability and Logic** (undergrad)
E. Mendelson, **Introduction to Mathematical Logic**, 3rd edition (undergrad/ grad)
J.N. Crossley and others, **What is Mathematical Logic?** (informal, readable)
A.J.Kfoury, R.Moll, and M. Arbib, **A Programming Approach to Computability** (undergrad)
M.Davis, R. Sigal, and E. Weyuker, **Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science** (undergrad/grad)

# Important

- All lectures and tutorials are manditory.
  Sometimes Friday 12-1 will be a lecture, other times a tutorial
- all assignments due at start of lecture/tutorial
  Late assignments Not accepted
- You may discuss your solutions with other students in the current course.
  Discussing with anyone outside course or consulting web is prohibited

- Work hard on understanding lecture notes, work hard on assignments
- start early -- cannot cram / solve in a couple of days
- Come to office hrs!
- Writeups must be completed independently.

# COURSE INTRO

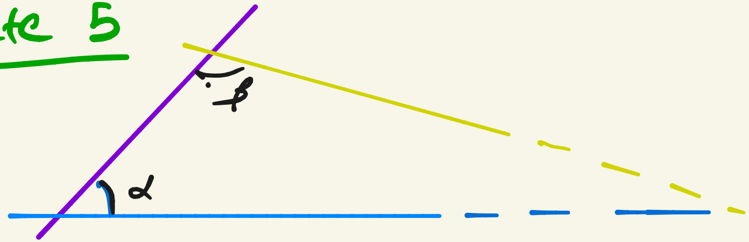
Foundations of mathematics involves the axiomatic method – write down axioms (basic truths) and prove theorems from axioms from purely logical reasoning

# 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 $\alpha, \beta$ angles)

# 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

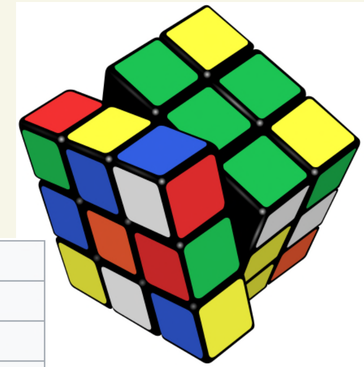
# Examples of groups

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)

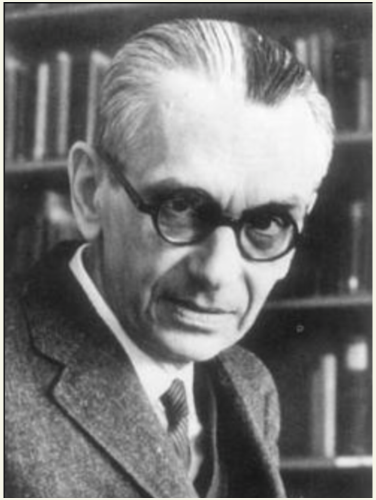
## II. FIRST ORDER (PREDICATE) LOGIC

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

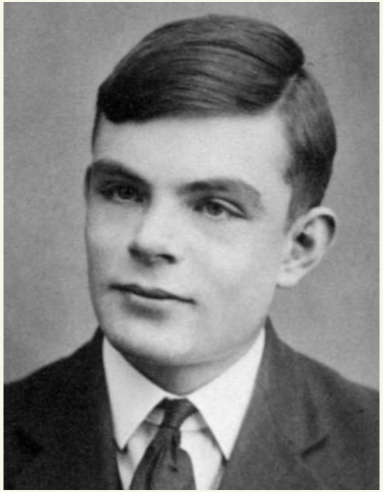
SOUNDNESS

** COMPLETENESS

MAJOR COROLLARIES OF COMPLETENESS

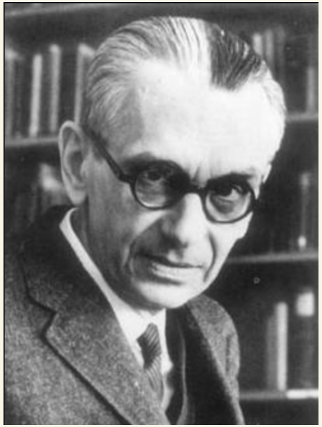


# COURSE OUTLINE (cont'd)



III. Computability

IV. Axiomatizable Theories



Incompleteness Theorems

Interplay/connections between computability & logic

# PROPOSITIONAL LOGIC

<u>Vocabulary</u>: $P_1$, $P_2$, $Q$, .. 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 $\tau : \{\text{atoms}\} \to T, F$ (true, false)

Extending $\tau$ 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

$\tau$ **satisfies** $A$ iff $A^{\tau} = T$

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

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

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

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

# Examples

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

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

## 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

$$(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_4) \wedge (\bar{x}_4) \wedge (x_1 \vee x_3) \wedge (x_1)$$

# 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

$Q=1 \quad R=1$

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

new variables

$$g: P_B \Leftrightarrow (Q \wedge R) \wedge \underline{P_A \Leftrightarrow P_B \vee \neg Q} \wedge P_A$$

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

# RESOLUTION

Start with CNF formula $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$

view $F$ as a set of clauses $\{C_1, C_2, \dots, C_m\}$

**Resolution Rule** :

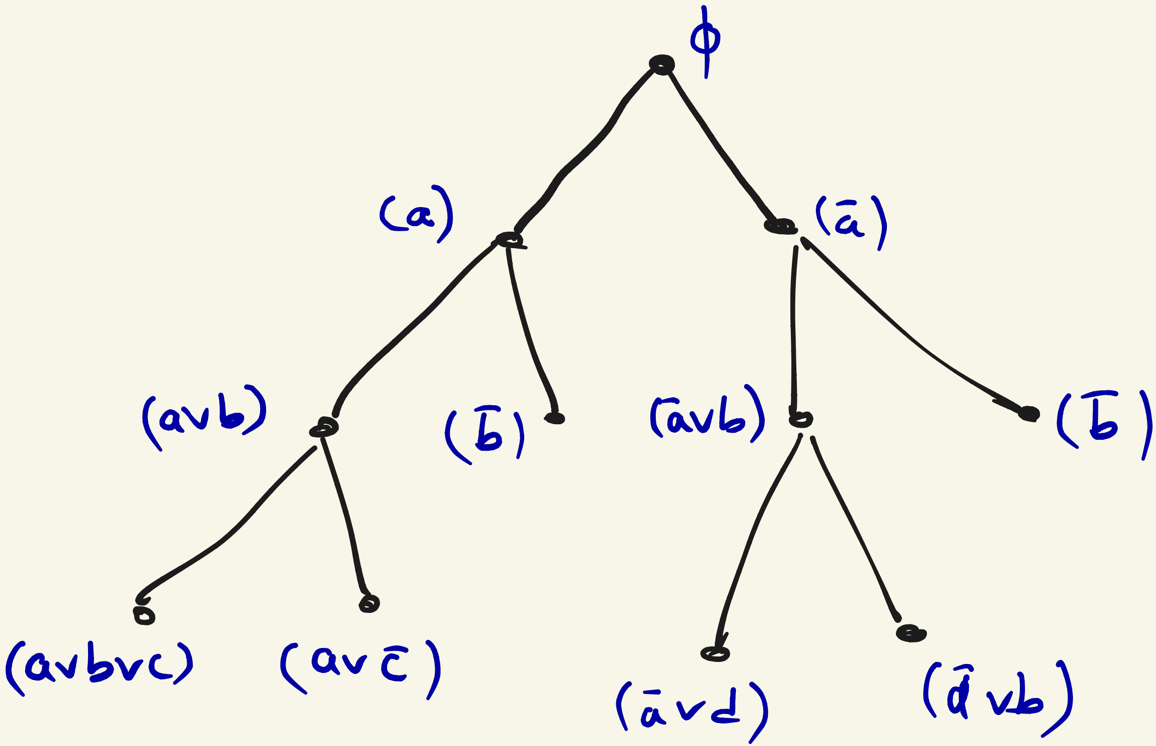
$(A \vee x)$, $(B \vee \bar{x})$ derive $(A \vee B)$

A Resolution Refutation of $F$ is a sequence of clauses $D_1, D_2, \dots, D_q$ such that:

each $D_i$ is either a clause from $F$, or follows from 2 previous clauses by Resolution rule, and final clause $D_q = \phi$ (the empty clause)

# Resolution Refutation

$F = (a \vee b \vee c)(a \vee \bar{c})(\bar{b})(\bar{a} \vee d)(\bar{d} \vee b)$



# Resolution Soundness

**Fact:** If $C_1, C_2$ derive $C_3$ by Resolution rule, then $C_1, C_2 \models C_3$

From above Fact we can prove:

## RESOLUTION SOUNDNESS THEOREM

If a CNF formula $F$ has a RES refutation, then $F$ is unsatisfiable

# RESOLUTION COMPLETENESS THM

Every unsatisfiable CNF formula F has a RESOLUTION Refutation

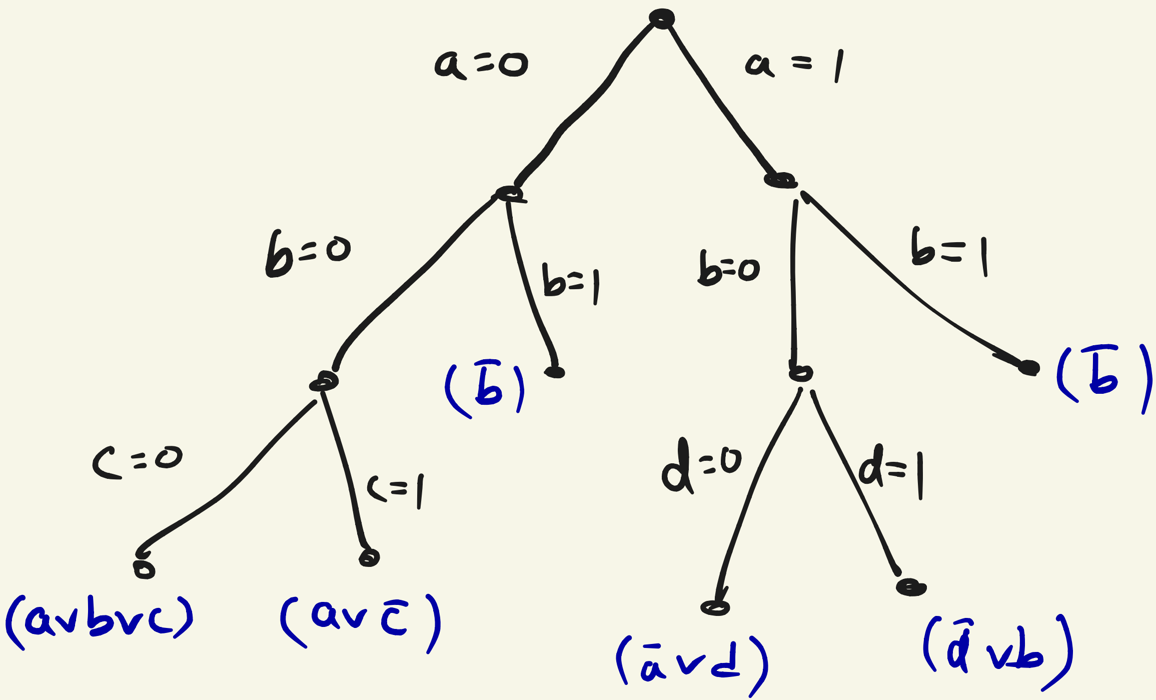
## Proof idea

We describe a canonical procedure for obtaining a RES refutation for F

The procedure exhaustively tries all truth ass's – via a decision tree
then we show that any such decision tree can be viewed as a RES refutation

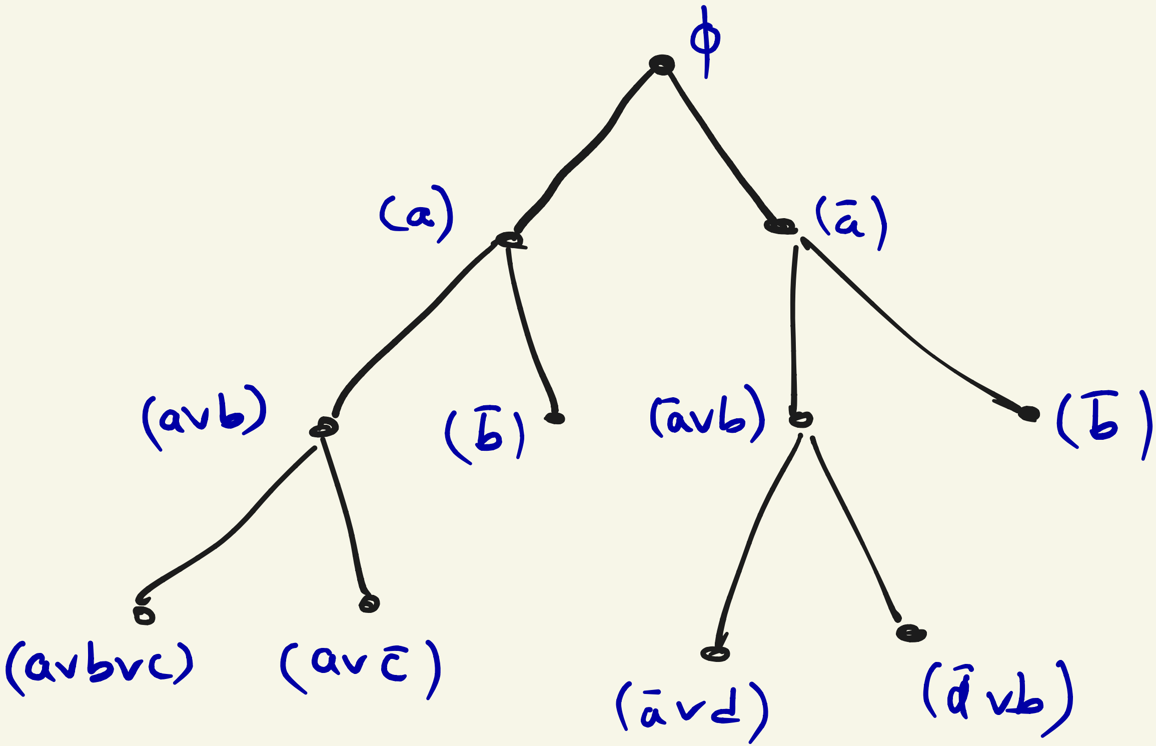
# DECISION TREES

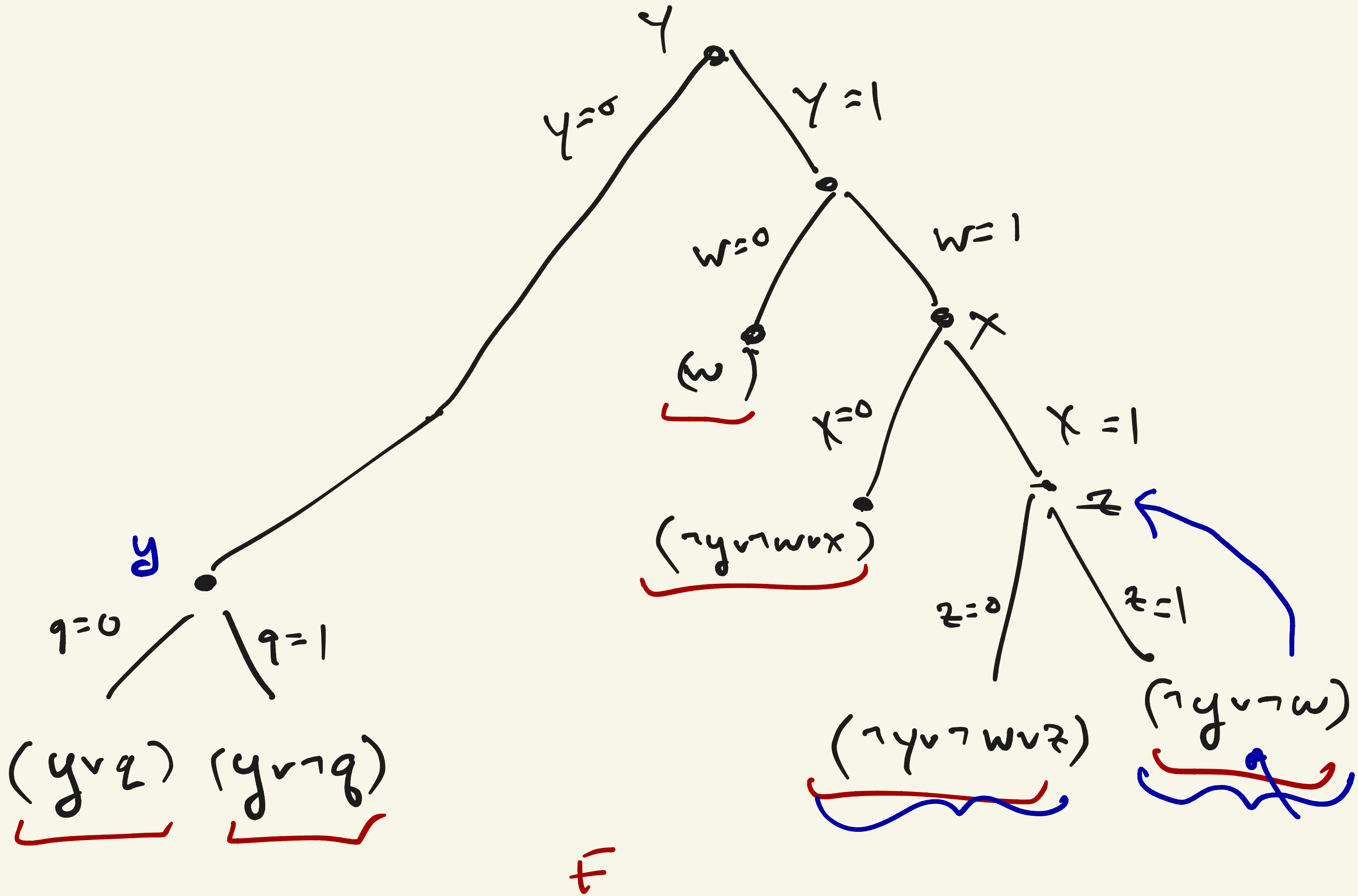
$F = (a \vee b \vee c)(a \vee \bar{c})(\bar{b})(\bar{a} \vee d)(\bar{d} \vee b)$



$\exists$ Resolution Refutation

$F = (a \vee b \vee c)(a \vee \bar{c})(\bar{b})(\bar{a} \vee d)(\bar{d} \vee b)$





y
y=0
y=1
w=0
w=1
x
(w)
x=0
x=1
z
(¬y∨¬w∨x)
y
q=0
q=1
z=0
z=1
(¬y∨¬w)
(¬y∨¬w∨z)
(y∨q)
(y∨¬q)
F