Mathematical Logic

Part One

Question: How do we formalize the logic we've been using in our proofs?

Where We're Going

- **Propositional Logic** (Today)
  - Basic logical connectives.
  - Truth tables.
  - Logical equivalences.
- **First-Order Logic** (Monday/Wednesday)
  - Reasoning about properties of multiple objects.

Propositional Logic

A ***proposition*** is a statement that is, by itself, either true or false.

Some Sample Propositions

- Puppies are cuter than kittens.
- Kittens are cuter than puppies.
- Usain Bolt can outrun everyone in this room.
- CS103 is useful for cocktail parties.
- This is the last entry on this list.

More Propositions

- Got kiss myself
- I'm so pretty
- I'm too hot
- Called a policeman and a fireman
- Made a dragon want to retire, man
- Uptown funk gon' give it to you

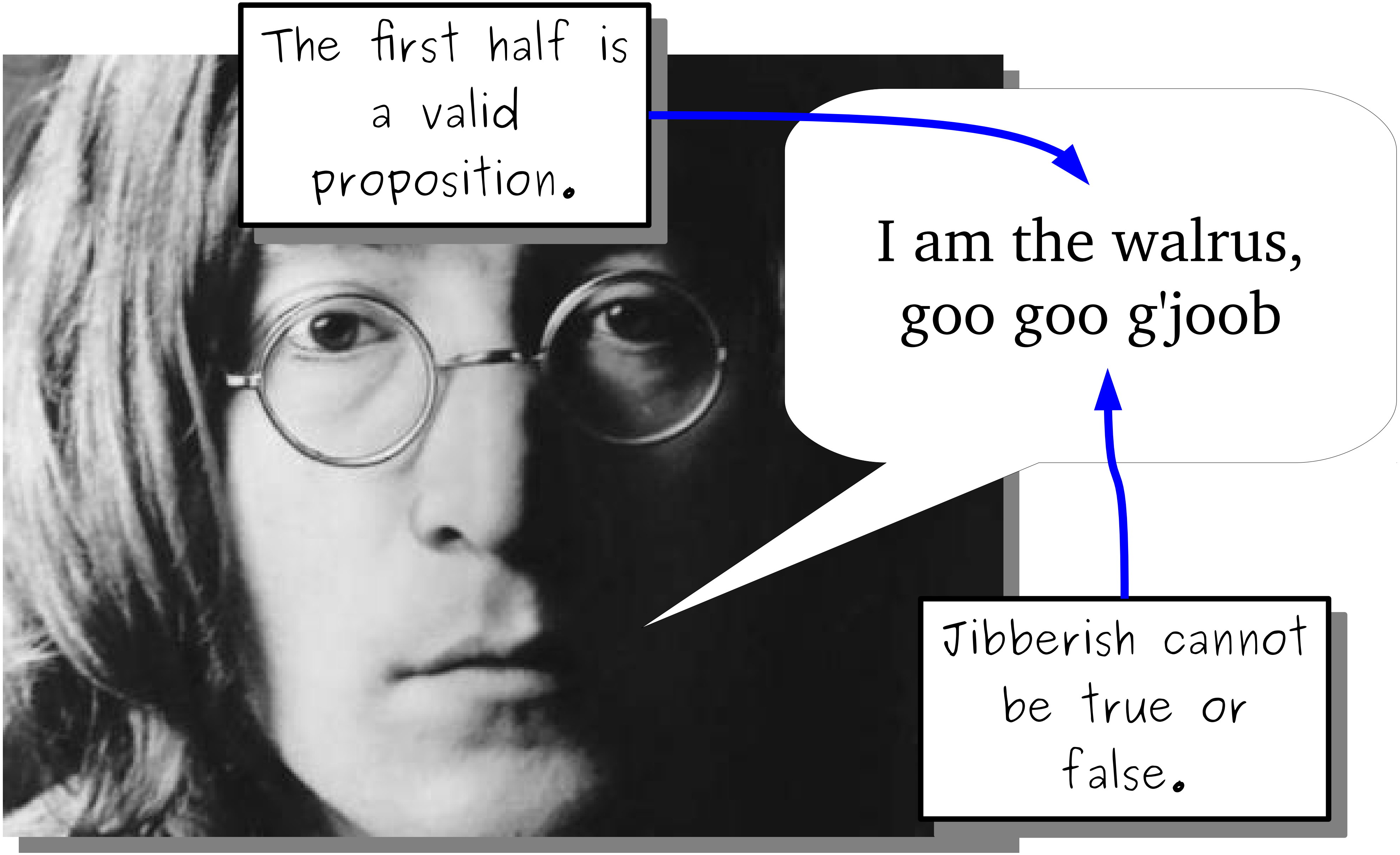
Things That Aren't Propositions

Commands cannot be true or false.

Things That Aren't Propositions

Questions cannot be true or false.

Things That Aren't Propositions



The first half is a valid proposition.

I am the walrus,
goo goo g'joob

Jibberish cannot be true or false.

Propositional Logic

- ***Propositional logic*** is a mathematical system for reasoning about propositions and how they relate to one another.
- Every statement in propositional logic consists of ***propositional variables*** combined via ***propositional connectives***.
  - Each variable represents some proposition, such as "You liked it" or "You should have put a ring on it."
  - Connectives encode how propositions are related, such as "If you liked it, then you should have put a ring on it."

Propositional Variables

- Each proposition will be represented by a ***propositional variable***.
- Propositional variables are usually represented as lower-case letters, such as *p*, *q*, *r*, *s*, etc.
- Each variable can take one one of two values: true or false.

Propositional Connectives

- **Logical NOT: $\neg p$**
  - Read "***not*** *p*"
  - $\neg p$ is true if and only if p is false.
  - Also called **logical negation**.
- **Logical AND: $p \land q$**
  - Read "*p* ***and*** *q*."
  - $p \land q$ is true if both p and q are true.
  - Also called **logical conjunction**.
- **Logical OR: $p \lor q$**
  - Read "*p* ***or*** *q*."
  - $p \lor q$ is true if at least one of *p* or *q* are true (inclusive OR)
  - Also called **logical disjunction**.

Truth Tables

- A ***truth table*** is a table showing the truth value of a propositional logic formula as a function of its inputs.
- Useful for several reasons:
  - Formally defining what a connective "means."
  - Deciphering what a complex propositional formula means.

The Truth Table Tool

Summary of Important Points

- The ∨ operator is an *inclusive* “or.” It's true if at least one of the operands is true.
  - Similar to the || operator in C, C++, Java and the **or** operator in Python.
- If we need an exclusive “or” operator, we can build it out of what we already have.

Mathematical Implication

Implication

- The → connective is used to represent implications.
  - Its technical name is the ***material conditional*** operator.
- What is its truth table?

Why This Truth Table?

- The truth values of the → are the way they are because they're *defined* that way.
- The intuition:
  - We want $p \to q$ to mean "whenever p is true, q is true as well."
  - The only way this *doesn't* happen is if p is true and q is false.
  - In other words, $p \to q$ should be true whenever $\neg(p \land \neg q)$ is true.
  - What's the truth table for $\neg(p \land \neg q)$?

Truth Table for Implication

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

The only way for $p \to q$ to be false is for p to be true and q to be false.

Otherwise, $p \to q$ is by definition true.

The Biconditional Operator

The Biconditional Operator

- The biconditional operator ↔ is used to represent a two-directional implication.
- Specifically, $p \leftrightarrow q$ means that p implies q and q implies p.
- What should its truth table look like?

The Biconditional

- The ***biconditional*** connective $p \leftrightarrow q$ is read "p if and only if q."
- Here's its truth table:

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

One interpretation of ↔ is to think of it as equality: the two propositions must have equal truth values.

True and False

- There are two more “connectives” to speak of: true and false.
  - The symbol ⊤ is a value that is always true.
  - The symbol ⊥ is value that is always false.
- These are often called connectives, though they don't connect anything.
  - (Or rather, they connect zero things.)

Operator Precedence

- How do we parse this statement?

$$\neg x \to y \lor z \to x \lor y \land z$$

- Operator precedence for propositional logic:

$$\neg$$
$$\land$$
$$\lor$$
$$\to$$
$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$\neg x \to y \vee z \to x \vee y \wedge z$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \to y \lor z \to x \lor y \land z$$

- Operator precedence for propositional logic:

$$\neg$$

$$\land$$

$$\lor$$

$$\to$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow y \vee z \rightarrow x \vee y \wedge z$$

- Operator precedence for propositional logic:

$$\begin{gathered}\neg \\ \wedge \\ \vee \\ \rightarrow \\ \leftrightarrow\end{gathered}$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow y \lor z \rightarrow x \lor (y \land z)$$

- Operator precedence for propositional logic:

$$\neg$$

$$\land$$

$$\lor$$

$$\rightarrow$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow y \vee z \rightarrow x \vee (y \wedge z)$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\rightarrow$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow (y \vee z) \rightarrow (x \vee (y \wedge z))$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\rightarrow$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow (y \vee z) \rightarrow (x \vee (y \wedge z))$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\rightarrow$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow ((y \vee z) \rightarrow (x \vee (y \wedge z)))$$

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\rightarrow$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow ((y \lor z) \rightarrow (x \lor (y \land z)))$$

- Operator precedence for propositional logic:

$$\neg$$

$$\land$$

$$\lor$$

$$\rightarrow$$

$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- The main points to remember:
  - ¬ binds to whatever immediately follows it.
  - ∧ and ∨ bind more tightly than →.
- We will commonly write expressions like $p \land q \to r$ without adding parentheses.
- For more complex expressions, we'll try to add parentheses.
- Confused? Just ask!

Time-Out for Announcements!

Problem Set Two

- Problem Set One was due at 12:50PM today.
    - Can use up to two late days on this assignment if you'd like.
- Problem Set Two goes out right now.
    - Checkpoint due Monday, April 13 at 12:50PM.
    - Remaining problems due Friday, April 17 at 12:50PM.
    - Explore induction, puzzles, games, and tournament structures!

WiCS Casual Dinner

- Stanford WiCS is hosting its first of their biquarterly CS Casual Dinners next Wednesday, April 15.
  - 6PM – 8PM in the Gates Fifth Floor lounge.
  - RSVP using **this link**.
- Wonderful event – great way to meet other students, faculty members, industry professionals, and alums. ***Highly recommended!***

Your Questions

“What is your favorite class to teach? CS 106A? CS 103? Or another one?”

They're all my favorite class to teach! ☺

“How do we know that the principle of mathematical induction is valid? Isn't the choice of what makes some logic 'legit' and other logic not pretty arbitrary? What would happen if we chose a different logical system, say, where induction isn't true?”

Induction is usually given as an ***axiom*** – it's legit because we ***define*** it to be legit. I haven't seen any examples of math that excludes induction, but if I find something, I'll be sure to tell you!

“There are many people who come into Stanford with extensive experience in math/cs/etc. How can people who are just finding their passions now catch up to these students who were literally the best in the world of their field during high school?”

I've got a few thoughts on this, in no particular order…

“What was your favorite CS class that you took during your time here at Stanford? What was your favorite non-CS class?”

Oh wow, that's hard.

For CS, probably one of CS154, CS161, or CS227B.

For non-CS, that's a funny story…

Back to CS103!

Recap So Far

- A ***propositional variable*** is a variable that is either true or false.
- The ***propositional connectives*** are
  - Negation: $\neg p$
  - Conjunction: $p \land q$
  - Disjunction: $p \lor q$
  - Implication: $p \to q$
  - Biconditional: $p \leftrightarrow q$
  - True: $\top$
  - False: $\bot$

Translating into Propositional Logic

Some Sample Propositions

a: I will get up early this morning

b: There is a lunar eclipse this morning

c: There are no clouds in the sky this morning

d: I will see the lunar eclipse

Some Sample Propositions

a: I will get up early this morning

b: There is a lunar eclipse this morning

c: There are no clouds in the sky this morning

d: I will see the lunar eclipse

> "I won't see the lunar eclipse if I don't get up early this morning."

Some Sample Propositions

a: I will get up early this morning

b: There is a lunar eclipse this morning

c: There are no clouds in the sky this morning

d: I will see the lunar eclipse

> "I won't see the lunar eclipse if I don't get up early this morning."

$$\neg a \to \neg d$$

“p if q”

translates to

$$q \to p$$

It does *not* translate to

$$p \to q$$

Some Sample Propositions

a: I will get up early this morning

b: There is a lunar eclipse this morning

c: There are no clouds in the sky this morning

d: I will see the lunar eclipse

Some Sample Propositions

a: I will get up early this morning

b: There is a lunar eclipse this morning

c: There are no clouds in the sky this morning

d: I will see the lunar eclipse

> “If I get up early this morning, but it's cloudy outside, I won't see the lunar eclipse.

Some Sample Propositions

a: I will get up early this morning

b: There is a lunar eclipse this morning

c: There are no clouds in the sky this morning

d: I will see the lunar eclipse

"If I get up early this morning, but it's cloudy outside, I won't see the lunar eclipse.

$$a \land \neg c \to \neg d$$

“*p*, but *q*”

translates to

$$p \land q$$

The Takeaway Point

- When translating into or out of propositional logic, be very careful not to get tripped up by nuances of the English language.
  - In fact, this is one of the reasons we have a symbolic notation in the first place!
- Many prepositional phrases lead to counterintuitive translations; make sure to double-check yourself!

Propositional Equivalences

Quick Question

What would I have to show you to convince you that the statement $\boldsymbol{p \land q}$ is false?

Quick Question

What would I have to show you to convince you that the statement $\boldsymbol{p \vee q}$ is false?

De Morgan's Laws

- Using truth tables, we concluded that

  $$\neg(p \land q)$$

  is equivalent to

  $$\neg p \lor \neg q$$

- We also saw that

  $$\neg(p \lor q)$$

  is equivalent to

  $$\neg p \land \neg q$$

- These two equivalences are called ***De Morgan's Laws***.

Logical Equivalence

- Because $\neg(p \land q)$ and $\neg p \lor \neg q$ have the same truth tables, we say that they're ***equivalent*** to one another.
- We denote this by writing

$$\boldsymbol{\neg(p \land q) \equiv \neg p \lor \neg q}$$

- The ≡ symbol is not a connective.
  - The statement $\neg(p \land q) \leftrightarrow (\neg p \lor \neg q)$ is a propositional formula. If you plug in different values of p and q, it will evaluate to a truth value. It just happens to evaluate to true every time.
  - The statement $\neg(p \land q) \equiv \neg p \lor \neg q$ means “these two formulas have exactly the same truth table.”
- In other words, the notation $\varphi \equiv \psi$ means “φ and ψ always have the same truth values, regardless of how the variables are assigned.”

An Important Equivalence

- Earlier, we talked about the truth table for $p \to q$. We chose it so that

$$\boldsymbol{p \to q \quad \equiv \quad \neg(p \land \neg q)}$$

- Later on, this equivalence will be incredibly useful:

$$\boldsymbol{\neg(p \to q) \quad \equiv \quad p \land \neg q}$$

Another Important Equivalence

- Here's a useful equivalence. Start with

$$p \to q \equiv \neg(p \land \neg q)$$

- By De Morgan's laws:

$$\begin{aligned} p \to q &\equiv \neg(p \land \neg q) \\ &\equiv \neg p \lor \neg\neg q \\ &\equiv \neg p \lor q \end{aligned}$$

- Thus $p \to q \equiv \neg p \lor q$

Another Important Equivalence

- Here's a useful equivalence. Start with

$$p \to q \equiv \neg(p \land \neg q)$$

- By De Morgan's laws:

$$\begin{aligned} p \to q &\equiv \neg(p \land \neg q) \\ &\equiv \neg p \lor \neg\neg q \\ &\equiv \neg p \lor \end{aligned}$$

- Thus $p \to q \equiv \neg p \lor q$

If p is false, then $\neg p \lor q$ is true. If p is true, then q has to be true for the whole expression to be true.

One Last Equivalence

The Contrapositive

- The contrapositive of the statement

  $$p \rightarrow q$$

  is the statement

  $$\neg q \rightarrow \neg p$$

- These are logically equivalent, which is why proof by contradiction works:

  $$\boldsymbol{p \rightarrow q \quad \equiv \quad \neg q \rightarrow \neg p}$$