CS103
Winter 2026

Lecture 03:

Propositional Logic

Mathematical Proofs

1. **Announcements**
2. PollEV
3. Propositional Logic (Variables, Connectives, and Truth Tables)
4. De Morgan’s Laws
5. Implication, Contrapositive, and Biconditional
6. Operator Precedence
7. Logical Connectives Table
8. Why All This Matters
9. Negation Practice

Submitting Problem Sets

- All assignments will be submitted through GradeScope.
  - The programming portion of each assignment is submitted separately from the written component.
  - The written component ***must*** be typed; handwritten solutions don't scan well and get mangled in GradeScope.
- All assignments are due at 1:00 PM. You have three "late days" you can use throughout the quarter. Each automagically extends assignment deadlines from Friday at 1:00 PM to Saturday at 1:00 PM; at most one late day can be used per assignment.
  - ***Very good idea:*** Leave at least two hours buffer time for your first assignment submission, just in case something goes wrong.
  - ***Very bad idea:*** Wait until the last minute to submit.
- Your score on the problem sets is the square root of your raw score. So an 81% maps to a 90%, a 50% maps to a 71%, etc. This gives a huge boost even if you need to turn something in that isn't done.

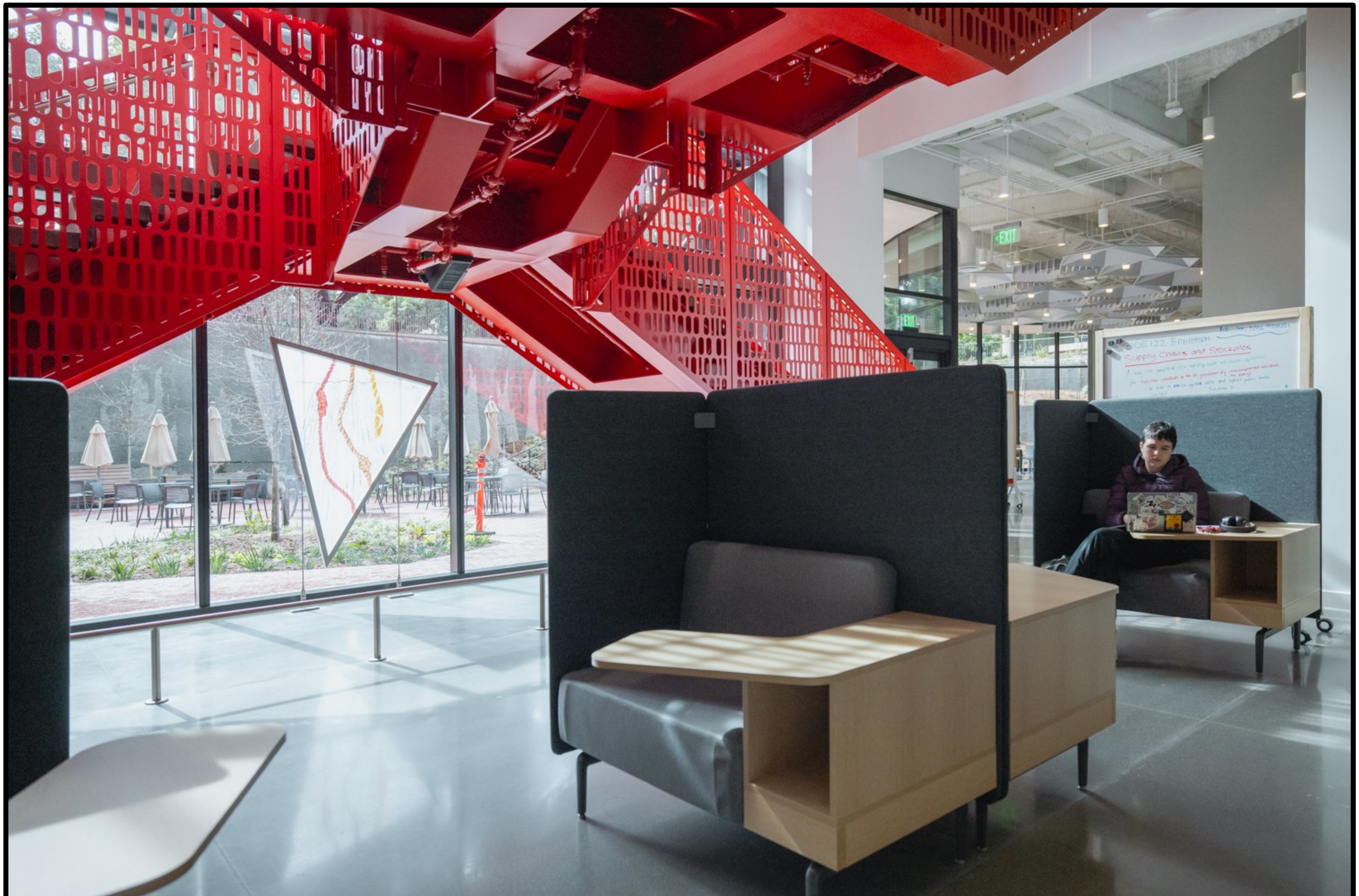
Office Hours

- Office hours have started (as of today)! Think of them as “drop-in help hours” where you can ask questions on problem sets, lecture topics, etc.
    - Check the *Office Hours* link on the course website for the schedule.
- TA office hours are held in person in the CoDa basement (“garden level”). Sean’s are in CoDa E112.
- For TA hours, once you arrive, sign up through the CS Office Hours Queue so that we can help people in the order they arrived:

https://queue.cs.stanford.edu/

- Office hours are *much* less crowded earlier in the week than later. Stop by on Monday and Tuesday!

Office Hours



Study Group Bonus

- I’m offering a small bonus for holding study group sessions this quarter. The amount of bonus is undecided, but the more weeks you’re involved with a study group, the more bonus you receive.
- There will be a weekly assignment bucket in Gradescope for submitting information about your study group for credit. These will be due **every Monday at 11:59 PM**.
- Study groups must have at least 5, and no more than 9, people. They must be current CS103 students. You can switch up groups as often as you want. Finding and forming groups is part of the challenge.
- Your study group should focus on material from the most recent 3-4 lectures each week.
- Study groups must meet **in person**.
- For other requirements, as well as ideas for how to make the most out of your time, see the *Study Groups* page on the course website (under the *Problem Sets* menu).

Mathematical Proofs

1. Announcements
2. **PollEV**
3. Propositional Logic (Variables, Connectives, and Truth Tables)
4. De Morgan’s Laws
5. Implication, Contrapositive, and Biconditional
6. Operator Precedence
7. Logical Connectives Table
8. Why All This Matters
9. Negation Practice

Lecture Participation

- Starting Wednesday, we will be using PollEV to ask questions in lecture for attendance credit.
- If you answer these questions in lecture, you'll get attendance credit for the day.
  - You must be physically present to be eligible for credit.
  - Wrong answers are okay, but sincere effort is required.
- CGOE students: We automatically opt you out of participation, since we assume you aren't physically here.
- If you'd prefer not to attend lectures, that's okay! You can opt to count your final exam in place of participation.
  - We'll send out a form where you can opt-out of participation in Week 4.

Do not miss this deadline!

Lecture Participation

- We'll dry-run PollEV questions today.
- Let's start with the following warm-up:

Make a music recommendation!

Answer at
https://cs103.stanford.edu/pollev

Click "**Register**" and enter your **Stanford e-mail** to get to the SUNet login page.

Also:
pollev.com/**cs103win26**

- Here are a few music recs of our own:
  - Jami Sieber - *Timeless*.
  - Aaron Parks - *Little Big* and *Little Big II*.
  - Arthur Moon - NPR Music Tiny Desk Concert.
  - Shakey Graves – *Roll the Bones* (check out *Audiotree Live* version).

Mathematical Proofs

1. Announcements
2. PollEV
3. **Propositional Logic (Variables, Connectives, and Truth Tables)**
4. De Morgan’s Laws
5. Implication, Contrapositive, and Biconditional
6. Operator Precedence
7. Logical Connectives Table
8. Why All This Matters
9. Negation Practice

Where We're Going

- ***Propositional Logic*** (Today)
  - Reasoning about Boolean values.
- ***First-Order Logic*** (Wednesday / Friday)
  - Reasoning about properties of multiple objects.

Note: Negation!

Goals

Eliminate ambiguity.

How do we formalize the definitions and reasoning we use in our proofs?

Express ideas precisely and concisely.

Systematic approach to manipulating complex statements – less error-prone.

Propositional Logic

$$TakeMath51 \lor TakeCME100$$

$$\neg FirstSucceed \rightarrow TryAgain$$

$$IsCardinal \land IsWhite$$

Propositional Logic

$$\mathit{TakeMath51} \lor \mathit{TakeCME100}$$

$$\neg \mathit{FirstSucceed} \to \mathit{TryAgain}$$

$$\mathit{IsCardinal} \land \mathit{IsWhite}$$

These are ***propositional variables***. Each propositional variable stands for a ***proposition***, something that is either true or false.

Propositional Logic

$TakeMath51 \lor TakeCME100$

$\neg FirstSucceed \to TryAgain$

$IsCardinal \land IsWhite$

These are ***propositional connectives***, which link propositions into larger propositions

Propositional Variables

- In propositional logic, individual propositions are represented by ***propositional variables***.
- Each variable can take one one of two values: true or false. You can think of them as **`bool`** values.

Propositional Connectives

- There are seven propositional connectives, five of which will be familiar from programming.
- First, there's the logical "NOT" operation:

$$\neg p$$

- You'd read this out loud as "not *p*."
- The fancy name for this operation is ***logical negation***.

Truth Tables

- A ***truth table*** is a table showing the truth value of a propositional logic formula as a function of its inputs.
- Let’s examine the truth tables for the connectives we’re exploring today!

Propositional Connectives

"I don't love cupcakes."

LoveCupcakes : I love cupcakes.

¬***LoveCupcakes***

Propositional Variables

- In a move that contravenes programming style conventions, propositional variables are usually represented as lower-case letters, such as *p*, *q*, *r*, *s*, etc.
- That said, there's nothing stopping you from using multiletter names!

Propositional Connectives

"I don't love cupcakes."

LoveCupcakes : I love cupcakes.

$\neg$***LoveCupcakes***

Propositional Connectives

- There are seven propositional connectives, five of which will be familiar from programming.
- Next, there's the logical "AND" operation:

$$\boldsymbol{p \wedge q}$$

- You'd read this out loud as "*p* and *q*."
- The fancy name for this operation is ***logical conjunction***.

Propositional Connectives

“It’s cardinal and white.”

IsCardinal : It’s cardinal.

IsWhite : It’s white.

IsCardinal $\wedge$ ***IsWhite***

Propositional Connectives

- There are seven propositional connectives, five of which will be familiar from programming.
- Then, there's the logical "OR" operation:

$$\boldsymbol{p \lor q}$$

- You'd read this out loud as "*p* or *q*."
- The fancy name for this operation is ***logical disjunction***. This is an *inclusive* or.

Propositional Connectives

"You must take MATH 51 or CME 100."

TakeMath51 : You must take MATH 51.

TakeCME100 : You must take CME 100.

$\textbf{\textit{TakeMath51}} \lor \textbf{\textit{TakeCME100}}$

Propositional Connectives

- There are seven propositional connectives, five of which will be familiar from programming.
- There's also the "truth" connective:

$$\top$$

- You'd read this out loud as "true."
- Although this is technically considered a connective, it "connects" zero things and behaves like a variable that's always true.

Propositional Connectives

- There are seven propositional connectives, five of which will be familiar from programming.
- Finally, there's the "false" connective.

$$\bot$$

- You'd read this out loud as "false."
- Like $\top$, this is technically a connective, but acts like a variable that's always false.

Inclusive and Exclusive OR

- The $\lor$ connective is an *inclusive* "or." It's true if at least one of the operands is true.
  - It's similar to the `||` operator in C, C++, Java, etc. and the `or` operator in Python.
- Sometimes we need an *exclusive* "or," which isn't true if both inputs are true.
- We can build this out of what we already have.

Write a propositional logic formula
for the exclusive OR of p and q.

Answer at
https://cs103.stanford.edu/pollev

Propositional Connectives

Quick Question:

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

Propositional Connectives

Quick Question:

What would I have to show you to convince you that the statement $p \lor q$ is false?

Mathematical Proofs

1. Announcements
2. PollEV
3. Propositional Logic (Variables, Connectives, and Truth Tables)
4. **De Morgan’s Laws**
5. Implication, Contrapositive, and Biconditional
6. Operator Precedence
7. Logical Connectives Table
8. Why All This Matters
9. Negation Practice

De Morgan's Laws

$\neg(p \wedge q)$ ***is equivalent to*** $\neg p \vee \neg q$

$\neg(p \vee q)$ ***is equivalent to*** $\neg p \wedge \neg q$

De Morgan’s Laws in Code

- ***Pro tip:*** Don't write this:

```
if (!(p() && q())) {
    /* … */
}
```

- Write this instead:

```
if (!p() || !q()) {
    /* … */
}
```

- This even short-circuits correctly: if `p()` returns false, `q()` is never evaluated.

Mathematical Proofs

1. Announcements
2. PollEV
3. Propositional Logic (Variables, Connectives, and Truth Tables)
4. De Morgan’s Laws
5. **Implication, Contrapositive, and Biconditional**
6. Operator Precedence
7. Logical Connectives Table
8. Why All This Matters
9. Negation Practice

Mathematical Implication

- We can represent implications using this connective:

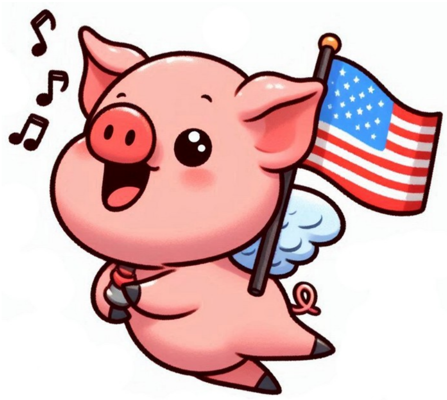
$$\boldsymbol{p \to q}$$

- You'd read this out loud as "p implies q."
  - The fancy name for this is the ***material conditional***.
- ***Question:*** What should the truth table for $p \to q$ look like?

The ***pig*** does the thing.

Sean throws ***qookies***.

Contract upheld?



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

A Contract (from Friday):

If a flying pig bursts into the room and sings a pitch-perfect version of the national anthem, then Sean will throw cookies to the class.

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

An implication is false only when the antecedent is true and the consequent is false.

Every formula is either true or false, so these other entries have to be true.

Important observation:

The statement $p \to q$ is true whenever $p \land \neg q$ is false.

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

An implication with a false antecedent is called ***vacuously true***.

An implication with a true consequent is called ***trivially true***.

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

Please commit this table to memory. We're going to need it, extensively, over the next couple of weeks.

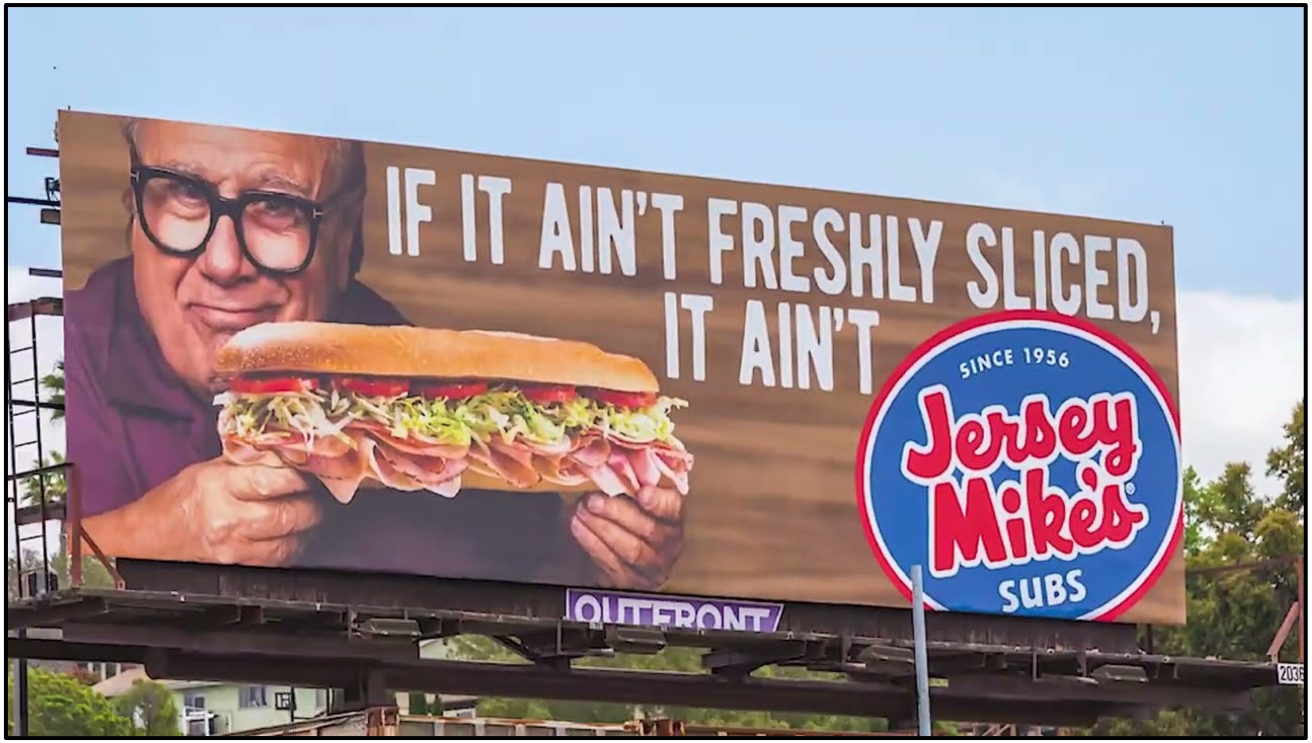
Mathematical Implication

"If at first you don't succeed, try again."

$\boldsymbol{\mathit{FirstSucceed}}$: You succeed at first.

$\boldsymbol{\mathit{TryAgain}}$: You ought to try again.

$$\neg \boldsymbol{\mathit{FirstSucceed}} \rightarrow \boldsymbol{\mathit{TryAgain}}$$

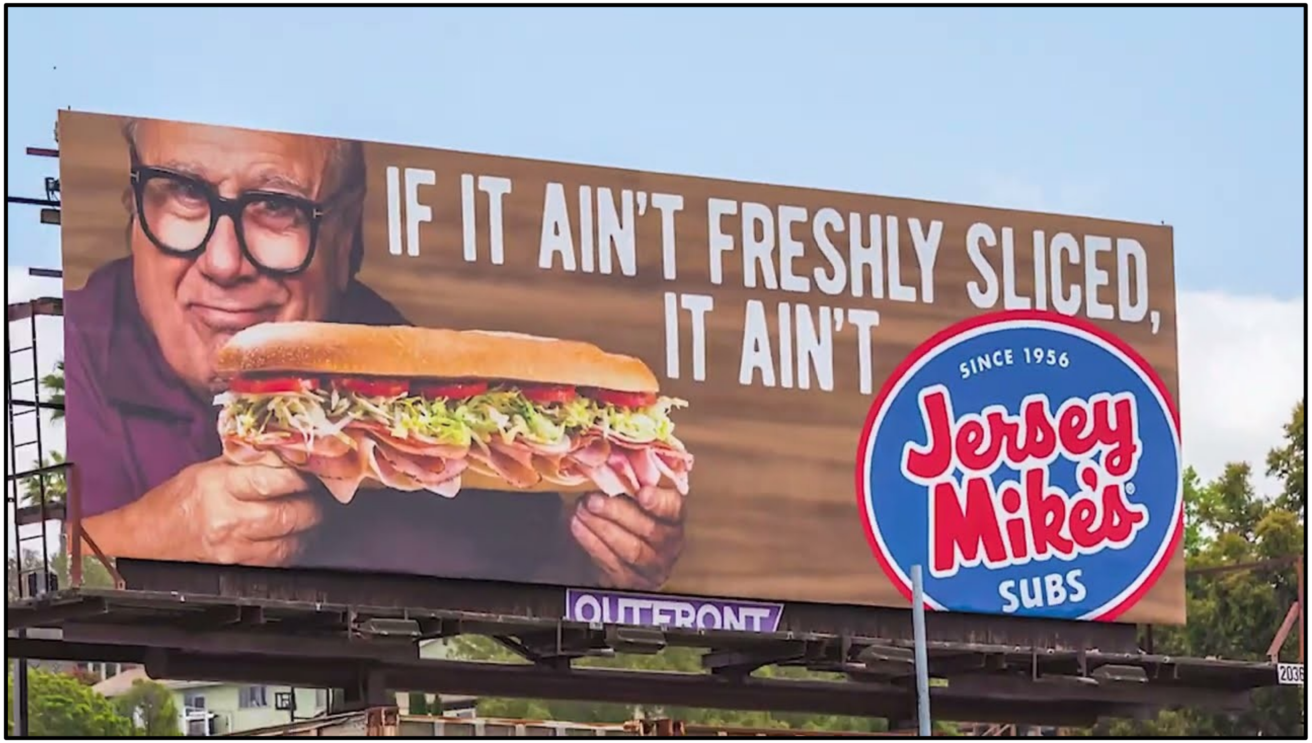


JerseyMikes : It's Jersey Mike's.

FreshlySliced : It's freshly sliced.

¬*FreshlySliced* → ¬*JerseyMikes*

What is the contrapositive here?



JerseyMikes : It's Jersey Mike's.

FreshlySliced : It's freshly sliced.

¬*FreshlySliced* → ¬*JerseyMikes*

JerseyMikes* → *FreshlySliced

An Important Equivalence

- The truth table for for $p \to q$ is chosen so that the following is true:

$$\boldsymbol{p \to q \quad \textit{is equivalent to} \quad \neg(p \land \neg q)}$$

- Later on, this equivalence will be incredibly useful:

$$\boldsymbol{\neg(p \to q) \quad \textit{is equivalent to} \quad p \land \neg q}$$

Contrapositive

We can use truth tables to demonstrate the equivalence of $p \to q$ and $\neg q \to \neg p$.

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

samesies :)

The Biconditional Connective

- In our previous lecture, we saw that the statement "p if and only if q" means both that $p \to q$ and $q \to p$.
- We can write this in propositional logic using the ***biconditional*** connective:

$$\boldsymbol{p \leftrightarrow q}$$

- This connective's truth table has the same meaning as "p implies q and q implies p."
- Based on that, what should its truth table look like?

The Biconditional Connective

- The biconditional connective $p \leftrightarrow q$ has the same truth table as $(p \to q) \land (q \to p)$.
- Here's what that looks like:

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.

Negating a Biconditional

- How do we simplify $\neg(p \leftrightarrow q)$ using the tools we've seen so far?
- There are many options, but here are our two favorites:

$$p \leftrightarrow \neg q \qquad \neg p \leftrightarrow q$$

Question to ponder: what is the truth table for these statements, and where have you seen it before?

Mathematical Proofs

1. Announcements
2. PollEV
3. Propositional Logic (Variables, Connectives, and Truth Tables)
4. De Morgan's Laws
5. Implication, Contrapositive, and Biconditional
6. **Operator Precedence**
7. Logical Connectives Table
8. Why All This Matters
9. Negation Practice

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

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

Mathematical Proofs

1. Announcements
2. PollEV
3. Propositional Logic (Variables, Connectives, and Truth Tables)
4. De Morgan’s Laws
5. Implication, Contrapositive, and Biconditional
6. Operator Precedence
7. **Logical Connectives Table**
8. Why All This Matters
9. Negation Practice

The Big Table

Connective	Read Aloud As	C++ Version	Fancy Name	Negation
$\neg p$	"not"	`!`	Negation	p
$p \wedge q$	"and"	`&&`	Conjunction	$\neg p \vee \neg q$ $p \to \neg q$
$p \vee q$	"or"	`||`	Disjunction	$\neg p \wedge \neg q$
$\top$	"true"	`true`	Truth	$\bot$
$\bot$	"false"	`false`	Falsity	$\top$
$p \to q$	"implies"	*see PS2!*	Implication	$p \wedge \neg q$
$p \leftrightarrow q$	"if and only if"	*see PS2!*	Biconditional	$p \leftrightarrow \neg q$ $\neg p \leftrightarrow q$

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: $p \lor q$
  - Truth: $\top$
  - Falsity: $\bot$
  - Implication: $p \to q$
  - Biconditional: $p \leftrightarrow q$

Mathematical Proofs

1. Announcements
2. PollEV
3. Propositional Logic (Variables, Connectives, and Truth Tables)
4. De Morgan's Laws
5. Implication, Contrapositive, and Biconditional
6. Operator Precedence
7. Logical Connectives Table
8. **Why All This Matters**
9. Negation Practice

Why All This Matters

- Suppose we want to prove the following statement:

“If $x + y = 16$, then $x \geq 8$ or $y \geq 8$”

Why All This Matters

- Suppose we want to prove the following statement:

“If $x + y = 16$, then $x \geq 8$ or $y \geq 8$”

$$x < 8 \land y < 8 \rightarrow x + y \neq 16$$

“If $x < 8$ and $y < 8$, then $x + y \neq 16$”

Most people find this version more intuitive than the theorem in its original form.

Theorem: If $x + y = 16$, then $x \geq 8$ or $y \geq 8$.

Proof: We will prove the contrapositive, namely, that if $x < 8$ and $y < 8$, then $x + y \neq 16$.

Pick x and y where $x < 8$ and $y < 8$. We want to show that $x + y \neq 16$. To see this, note that

$$\begin{aligned} x + y &< 8 + y \\ &< 8 + 8 \\ &= 16. \end{aligned}$$

This means that $x + y < 16$, so $x + y \neq 16$, which is what we needed to show. ■

Why This Matters

- Propositional logic lets us symbolically manipulate statements and theorems.
  - This can help us better understand what a theorem says or what a definition means.
- It's also very useful for proofs by contradiction and contrapositive.
- Being able to negate statements mechanically can reduce the likelihood of taking the negation or contrapositive incorrectly.

Mathematical Proofs

1. Announcements
2. PollEV
3. Propositional Logic (Variables, Connectives, and Truth Tables)
4. De Morgan’s Laws
5. Implication, Contrapositive, and Biconditional
6. Operator Precedence
7. Logical Connectives Table
8. Why All This Matters
9. **Negation Practice**

Negation Practice

- Here's a propositional formula that contains some negations. Simplify it as much as possible:

$$\neg(p \land q \to r \lor s)$$

Negation Practice

- Here's a propositional formula that contains some negations. Simplify it as much as possible:

$$p \wedge q \wedge \neg r \wedge \neg s$$

(*see extended lecture slides for derivation*)

Negation Practice

- Here's a propositional formula that contains some negations. Simplify it as much as possible:

$$\neg((p \lor (q \land r)) \leftrightarrow (a \land b \land c \to d))$$

Negation Practice

- Here's a propositional formula that contains some negations. Simplify it as much as possible:

$$(p \lor (q \land r)) \leftrightarrow (a \land b \land c \land \neg d)$$

(*see extended lecture slides for derivation*)

Next Time

- ***First-Order Logic***
  - Reasoning about groups of objects.
- ***First-Order Translations***
  - Expressing yourself in symbolic math!