Mathematical Logic
Part One
**Question:** How do we formalize the definitions and reasoning we use in our proofs?
Where We're Going

- **Propositional Logic** (Today)
  - Basic logical connectives.
  - Truth tables.
  - Logical equivalences.

- **First-Order Logic** (Wednesday/Friday)
  - Reasoning about properties of multiple objects.
Propositional Logic
A *proposition* is a statement that is, by itself, either 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, so each variable has value true or false.
  - Connectives encode how propositions are related.
Propositional Connectives

• **Logical NOT:** $\neg p$
  - $\neg p$ is true if and only if $p$ is false.
  - Also called *logical negation*.

• **Logical AND:** $p \land q$
  - $p \land q$ is true if and only if both $p$ and $q$ are true.
  - Also called *logical conjunction*.

• **Logical OR:** $p \lor q$
  - $p \lor q$ is true if and only 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:
  • They give a formal definition of what a connective “means.”
  • They give us a mechanical way to evaluate a complex propositional formula.
Truth Table for XOR

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p XOR q</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

• Recall that our OR connective is inclusive.

• The truth table at right defines an exclusive or called XOR.

• We also could have expressed XOR using just the connectives we already had.

Which expresses XOR?

(A) \((p \land q) \lor (p \lor q)\)
(B) \((p \land q) \lor \neg(p \lor q)\)
(C) \((p \lor q) \land \neg(p \land q)\)
(D) \((p \land q) \land (p \lor q)\)
Mathematical Implication
Truth Table for $p \rightarrow q$ (implies)

What is the correct truth table for implication? Enter your guess as a list of four values to fill in the rightmost column of the table.
(ex: F, T, ?, F)

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then your response.
Truth Table for Implication

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
## Truth Table for Implication

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

- • Bad bracket, didn’t get A
- • Bad bracket, got A
- • Perfect bracket, didn’t get A
- • Perfect bracket, got A
Truth Table for Implication

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

The implication is only false if $p$ is true and $q$ isn’t. It’s true otherwise.
### Truth Table for Implication

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

The implication is only false if $p$ is true and $q$ isn’t. It’s true otherwise.

You will need to commit this table to memory. (Consider a tattoo on your forearm.) We’re going to be using it a lot over the rest of the week.
Why This Truth Table?

• The truth values of the $\rightarrow$ are the way they are because they're defined that way.
• Are there other ways we could write a proposition that has the same truth table as $\rightarrow$, using the other connectives? (like what we did for XOR)
  • Yep!
  • Try to think of some
  • What's the truth table for $\neg(p \land \neg q)$?
The Biconditional Connective
The Biconditional Connective

• The biconditional connective ↔ is used to represent a two-directional implication.
• Specifically, $p \leftrightarrow q$ means both that $p \rightarrow q$ and that $q \rightarrow p$.
• Based on that, what should its truth table look like?
• Take a guess, and talk it over with your neighbor!
Biconditionals

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

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \leftrightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Biconditionals

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

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \leftrightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

One interpretation of $\leftrightarrow$ is to think of it as equality: the two propositions must have equal truth values.
True and False

• In addition to variables and connectives, we have constants: true and false.
  • The symbol $\top$ is a value that is always true.
  • The symbol $\bot$ is value that is always false.

• These are often called connectives, though they don't connect anything.
  • (Or rather, they connect zero things.)
Proof by Contradiction

- Suppose you want to prove \( p \) is true using a proof by contradiction.
- The setup looks like this:
  - Assume \( p \) is false.
  - Derive something that we know is false.
  - Conclude that \( p \) is true.
- In propositional logic:
  \[ (\neg p \rightarrow \bot) \rightarrow p \]
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:

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

• 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:

[Diagram showing operator precedence]

• 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?
  \[(\neg x) \to y \lor z \to x \lor y \land z\]
- Operator precedence for propositional logic:
  \[
  \vdash
  \land
  \lor
  \to
  \equiv
  \]
- 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:

\[ \neg \quad \land \quad \lor \quad \rightarrow \quad \leftrightarrow \]

• 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:

$\neg \quad \land \quad \lor \quad \rightarrow \quad \leftrightarrow$

• 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:

\[
\begin{align*}
\neg & \\
\land & \\
\lor & \\
\rightarrow & \\
\leftrightarrow &
\end{align*}
\]

• 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:

\[
\neg \quad \land \quad \lor \quad \rightarrow \quad \leftrightarrow
\]

• 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:

\[
\begin{align*}
\neg & \quad \land & \quad \lor & \quad \to \\
& & & \quad \leftarrow \\
& & & \quad \leftrightarrow
\end{align*}
\]

• 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:

\[
\begin{align*}
\neg & \\
\land & \\
\lor & \\
\rightarrow & \\
\leftrightarrow & 
\end{align*}
\]

• 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.
  • $\land$ and $\lor$ bind more tightly than $\rightarrow$.
• We will commonly write expressions like $p \land q \rightarrow r$ without adding parentheses.
• For more complex expressions,
  • you should add parens (the TAs thank you!)
  • we'll try to add parentheses (if I ever don’t and you’re confused, ok to ask!)
# The Big Table

<table>
<thead>
<tr>
<th>Connective</th>
<th>Read As</th>
<th>C++ Version</th>
<th>Fancy Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>¬</td>
<td>“not”</td>
<td>!</td>
<td>Negation</td>
</tr>
<tr>
<td>∧</td>
<td>“and”</td>
<td>&amp;&amp;</td>
<td>Conjunction</td>
</tr>
<tr>
<td>∨</td>
<td>“or”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>→</td>
<td>“implies”</td>
<td>see PS2!</td>
<td>Implication</td>
</tr>
<tr>
<td>↔</td>
<td>“if and only if”</td>
<td>see PS2!</td>
<td>Biconditional</td>
</tr>
<tr>
<td>⊤</td>
<td>“true”</td>
<td>true</td>
<td>Truth</td>
</tr>
<tr>
<td>⊥</td>
<td>“false”</td>
<td>false</td>
<td>Falsity</td>
</tr>
</tbody>
</table>
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: $p \rightarrow q$
  - Biconditional: $p \leftrightarrow q$
  - True: $\top$
  - False: $\bot$
Translating into Propositional Logic
Some Sample Propositions

\[ a: \text{I will be in the path of totality.} \]
\[ b: \text{I will see a total solar eclipse.} \]
Some Sample Propositions

\[ a: \text{I will be in the path of totality.} \]
\[ b: \text{I will see a total solar eclipse.} \]

“I won't see a total solar eclipse if I'm not in the path of totality.”
Some Sample Propositions

\( a: \) I will be in the path of totality.
\( b: \) I will see a total solar eclipse.

\[ \neg a \rightarrow \neg b \]

“I won't see a total solar eclipse if I'm not in the path of totality.”
“p if q” translates to 

$q \rightarrow p$

It does not translate to 

$p \rightarrow q$
“p, but q” translates to

\[ p \land q \]
Some Sample Propositions

\(a\): I will be in the path of totality.
\(b\): I will see a total solar eclipse.
\(c\): There is a total solar eclipse today.
Some Sample Propositions

\( a \): I will be in the path of totality.

\( b \): I will see a total solar eclipse.

\( c \): There is a total solar eclipse today.

“If I will be in the path of totality, but there's no solar eclipse today, I won't see a total solar eclipse.”

Which is equivalent to the sentence at left?

(A) \( a \land \neg c \land \neg b \)

(B) \( (a \land \neg c) \rightarrow \neg b \)

(C) \( a \rightarrow (\neg c \land \neg b) \)

(D) \( (\neg a \lor c) \lor \neg b \)

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then your response.
Some Sample Propositions

\(a\): I will be in the path of totality.
\(b\): I will see a total solar eclipse.
\(c\): There is a total solar eclipse today.

"If I will be in the path of totality, but there's no solar eclipse today, I won't see a total solar eclipse."

\(a \land \neg c \rightarrow \neg b\)
Propositional Equivalences
Quick Question:

What would I have to show you to convince you that the statement $p \land q$ is false?
Quick Question:

What would I have to show you to convince you that the statement $p \lor q$ 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.
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!)
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

\[
\neg(p \land q) \equiv \neg p \lor \neg q
\]

- The \( \equiv \) 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 “\( \varphi \) and \( \psi \) always have the same truth values, regardless of how the variables are assigned.”
An Important Equivalence

• Earlier, we talked about the truth table for $p \rightarrow q$. We chose it so that
  \[ p \rightarrow q \equiv \neg(p \land \neg q) \]

• Later on, this equivalence will be incredibly useful:
  \[ \neg(p \rightarrow q) \equiv p \land \neg q \]
Another Important Equivalence

• Here's a useful equivalence. Start with

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

• By De Morgan's laws:

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

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

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

• Thus  $$p \rightarrow q \equiv \neg p \lor q$$
Another Important Equivalence

- Here's a useful equivalence. Start with
  \[ p \rightarrow q \equiv \neg(p \land \neg q) \]
- By De Morgan's laws:
  \[ p \rightarrow q \equiv \neg(p \land \neg q) \equiv \neg p \lor \neg q \]
- Thus \[ p \rightarrow q \equiv \neg p \lor q \]

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 contrapositive works:

\[ p \rightarrow q \equiv \neg q \rightarrow \neg p \]
Why All This Matters
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 \( y \geq 8 \)"

\[ x + y = 16 \rightarrow x \geq 8 \lor y \geq 8 \]
Why All This Matters

• Suppose we want to prove the following statement:
  “If $x + y = 16$, then $x \geq 8$ or $y \geq 8$”

  \[ x + y = 16 \rightarrow x \geq 8 \lor y \geq 8 \]
Why All This Matters

- Suppose we want to prove the following statement:
  
  "If $x + y = 16$, then $x \geq 8$ or $y \geq 8$"

  $\neg(x \geq 8 \lor y \geq 8) \rightarrow \neg(x + y = 16)$
Why All This Matters

• Suppose we want to prove the following statement:

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

  $\neg(x \geq 8 \lor y \geq 8) \rightarrow \neg(x + y = 16)$
Why All This Matters

• Suppose we want to prove the following statement:

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

  \[ \neg(x \geq 8 \lor y \geq 8) \rightarrow \neg(x + y = 16) \]
Why All This Matters

• Suppose we want to prove the following statement:

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

\[ \neg(x \geq 8 \lor y \geq 8) \rightarrow x + y \neq 16 \]
Why All This Matters

• Suppose we want to prove the following statement:

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

   \[ \neg(x \geq 8 \lor y \geq 8) \rightarrow x + y \neq 16 \]
Why All This Matters

• Suppose we want to prove the following statement:

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

  $\neg(x \geq 8 \lor y \geq 8) \rightarrow x + y \neq 16$
Suppose we want to prove the following statement:

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

\[\neg(x \geq 8) \land \neg(y \geq 8) \rightarrow x + y \neq 16\]
Why All This Matters

• Suppose we want to prove the following statement:

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

   \[-(x \geq 8) \land -(y \geq 8) \rightarrow x + y \neq 16\]
Why All This Matters

• Suppose we want to prove the following statement:

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

\[
\neg(x \geq 8) \land \neg(y \geq 8) \rightarrow x + y \neq 16
\]
Why All This Matters

• Suppose we want to prove the following statement:

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

$$x < 8 \land \neg(y \geq 8) \implies x + y \neq 16$$
Why All This Matters

• Suppose we want to prove the following statement:

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

   $x < 8 \land \neg(y \geq 8) \rightarrow x + y \neq 16$
Why All This Matters

- Suppose we want to prove the following statement:

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

  \[ x < 8 \land \neg(y \geq 8) \rightarrow x + y \neq 16 \]
Why All This Matters

• Suppose we want to prove the following statement:

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

   \[ x < 8 \land y < 8 \rightarrow x + y \neq 16 \]
Why All This Matters

• Suppose we want to prove the following statement:

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

  $x < 8 \land y < 8 \rightarrow x + y \neq 16$
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 \( x + y \neq 16 \)”
Theorem: If \( x + y = 16 \), then \( x \geq 8 \) or \( y \geq 8 \).

Proof: By contrapositive. We will prove that if \( x < 8 \) and \( y < 8 \), then \( x + y \neq 16 \). Let \( x \) and \( y \) be arbitrary numbers such that \( x < 8 \) and \( y < 8 \).

Note that

\[
\begin{align*}
x + y &< 8 + y \\
&< 8 + 8 \\
&= 16.
\end{align*}
\]

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

• Propositional logic is a tool for reasoning about how various statements affect one another.

• To better understand how to prove a result, it often helps to translate what you're trying to prove into propositional logic first.

• That said, propositional logic isn't expressive enough to capture all statements. For that, we need something more powerful.
Next Time

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