Propositional Logic

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

Where We're Going

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

Propositional Logic

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

In other words, English sentences can be propositions, but not all are (for example, commands and questions can't be propositions).

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 Logic as a Boolean Algebra

- In elementary school arithmetic, we learn that two expressions are equivalent, *for specific numbers*:

$$(9 + 5) / 7 = (1/7)(9 + 5)$$

$$(14)/7 = (1/7)(14)$$

$$2 = 2$$

- In high school, we learn algebra, which lets us study the structural patterns of equivalence, *regardless of the specific numbers involved*:

$$(a + b) / c = (1/c)(a + b)$$

- Algebra replaces the numbers with variables so we can focus on analyzing and manipulating the structure.

Propositional Logic as a Boolean Algebra

- Philosophers, mathematicians, and logicians wanted to do the same thing that algebra does for arithmetic, but for the analysis of the structure of arguments not analysis of the structure of numeric calculations.
- We replace individual English sentences that state facts with propositional variables, and replace the “if…then,” “and,” “or,” etc. with operator symbols.
- So we can focus on analyzing and manipulating the structure.

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

- There are seven propositional connectives, many 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***.

Propositional Connectives

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

$$\boldsymbol{p \land q}$$

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

Propositional Connectives

- There are seven propositional connectives, many 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.

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 go look at the truth tables for the three connectives we've seen so far:

$$\neg \qquad \wedge \qquad \vee$$

Summary of Important Points

- The ∨ connective is an *inclusive* “or.” It's true if at least one of the operands is true.
  - Similar to the **||** operator in C, C++, Java, etc. and the **or** operator in Python.
- If we need an exclusive “or” operator, we can build it out of what we already have.
- Try this yourself! Take a minute to combine these operators together to form an expression that represents the exclusive or of p and q (something that’s true if and only if exactly one of p and q are true.)

Mathematical Implication

Implication

- We can represent implications using this connective:

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

- You'd read this out loud as "p implies q."

- ***Question:*** What should the truth table for $p \to q$ look like?
- Pull out a sheet of paper, make a guess, and talk things over with your neighbors!

Implication

Dr. Lee: "**If** you pick a perfect March Madness bracket this year, **then** I'll give you an A+ in CS103."

What if…

- …you pick a perfect bracket and get an A+?
- …you pick a bad bracket and get an A+?
- …you pick a perfect bracket and get a C?
- …you pick a bad bracket and get a C?

Implication

- ...you pick a perfect bracket and get an A+?
- ...you pick a bad bracket and get an A+?
- ...you pick a perfect bracket and get a C?
- ...you pick a bad bracket and get a C?

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

Implication

- ...you pick a perfect bracket and get an A+?
- ...you pick a bad bracket and get an A+?
- ...you pick a perfect bracket and get a C?
- ...you pick a bad bracket and get a C?

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

Implication

- …you pick a perfect bracket and get an A+?
- …you pick a bad bracket and get an A+?
- …you pick a perfect bracket and get a C?
- …you pick a bad bracket and get a C?

p	q	$p \rightarrow q$
F	F	
F	T	
T	F	
T	T	T

Implication

- …you pick a perfect bracket and get an A+?
- …you pick a bad bracket and get an A+?
- …you pick a perfect bracket and get a C?
- …you pick a bad bracket and get a C?

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

Implication

- ...you pick a perfect bracket and get an A+?
- ...you pick a bad bracket and get an A+?
- ...you pick a perfect bracket and get a C?
- ...you pick a bad bracket and get a C?

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

Implication

- ...you pick a perfect bracket and get an A+?
- ...you pick a bad bracket and get an A+?
- ...you pick a perfect bracket and get a C?
- ...you pick a bad bracket and get a C?

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

Implication

- …you pick a perfect bracket and get an A+?
- …you pick a bad bracket and get an A+?
- …you pick a perfect bracket and get a C?
- …you pick a bad bracket and get a C?

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

Implication

- ...you pick a perfect bracket and get an A+?
- ...you pick a bad bracket and get an A+?
- ...you pick a perfect bracket and get a C?
- ...you pick a bad bracket and get a C?

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

Implication

- ...you pick a perfect bracket and get an A+?
- ...you pick a bad bracket and get an A+?
- ...you pick a perfect bracket and get a C?
- ...you pick a bad bracket and get a C?

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

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

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.

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

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***.

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.

Fun Fact: The Contrapositive Revisited

The Biconditional Connective

The Biconditional Connective

- On Friday, we saw that "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?
- 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:

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

Biconditionals

- 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 $\top$ 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:

$$(\neg p \to \bot) \to p$$

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?

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

- 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:

$$\neg$$
$$\boldsymbol{\wedge}$$
$$\vee$$
$$\rightarrow$$
$$\leftrightarrow$$

- 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:

$$\neg$$

$$\boldsymbol{\wedge}$$

$$\vee$$

$$\rightarrow$$

$$\leftrightarrow$$

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

Operator Precedence

- How do we parse this statement?

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

- 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?

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

- 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?

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

- 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?

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

- Operator precedence for propositional logic:

$$\neg$$

$$\wedge$$

$$\vee$$

$$\to$$

$$\leftrightarrow$$

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

Operator Precedence

- The main points to remember:
  - $\neg$ binds to whatever immediately follows it.
  - $\wedge$ and $\vee$ bind more tightly than $\rightarrow$.
  - We will commonly write expressions like $p \wedge q \rightarrow r$ without adding parentheses.
- *For more complex expressions, let's agree to use parentheses!*

The Big Table

<table>
<tr><th>Connective</th><th>Read Aloud As</th><th>C++ Version</th><th>Fancy Name</th></tr>
<tr><td>$\neg$</td><td>“not”</td><td><code>!</code></td><td>Negation</td></tr>
<tr><td>$\wedge$</td><td>“and”</td><td><code>&&</code></td><td>Conjunction</td></tr>
<tr><td>$\vee$</td><td>“or”</td><td><code>||</code></td><td>Disjunction</td></tr>
<tr><td>$\to$</td><td>“implies”</td><td>see PS2!</td><td>Implication</td></tr>
<tr><td>$\leftrightarrow$</td><td>“if and only if”</td><td>see PS2!</td><td>Biconditional</td></tr>
<tr><td>$\top$</td><td>“true”</td><td><code>true</code></td><td>Truth</td></tr>
<tr><td>$\bot$</td><td>“false”</td><td><code>false</code></td><td>Falsity</td></tr>
</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: $p \to q$
  - Biconditional: $p \leftrightarrow q$
  - True: $\top$
  - False: $\bot$

Translating into Propositional Logic

Some Sample Propositions

a: I will be in the path of totality.

b: I will see a total solar eclipse.

Some Sample Propositions

a: I will be in the path of totality.

b: 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.

"I won't see a total solar eclipse if I'm not in the path of totality."

$$\neg a \to \neg b$$

“$\boldsymbol{p}$ **if** $\boldsymbol{q}$”

translates to

$$q \to p$$

It does *not* translate to

⚠ $p \to q$ ⚠

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."

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."

$$a \land \neg c \to \neg b$$

“***p*, but *q***”

translates to

$\boldsymbol{p \wedge q}$

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

$$\neg(p \wedge q)$$

  is equivalent to

$$\neg p \vee \neg q$$

- We also saw that

$$\neg(p \vee q)$$

  is equivalent to

$$\neg p \wedge \neg q$$

- 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!)

An Important Equivalence

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

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

- Later on, this equivalence will be incredibly useful:

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

Another Important Equivalence

- Here's a useful equivalence. Start with

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

- By de Morgan's laws:

$$\begin{aligned} p \to q \quad & \textit{is equivalent to} \quad \neg(p \land \neg q) \\ & \textit{is equivalent to} \quad \neg p \lor \neg\neg q \\ & \textit{is equivalent to} \quad \neg p \lor q \end{aligned}$$

- Thus $p \to q$ is equivalent to $\neg p \lor q$

Another Important Equivalence

- Here's a useful equivalence. Start with

$p \to q$ *is equivalent to* $\neg(p \land \neg q)$

- By de Morgan's laws:

$p \to q$ *is equivalent to*

is equivalent to

is equivalent to

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.

- Thus $p \to q$ is equivalent to $\neg p \lor q$

Next Time

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