Propositional Logic

Introduction

- ***Propositional Logic***
  - Reasoning about Boolean values.
- ***First-Order Logic***
  - Reasoning about properties of multiple objects.

Outline

- ***Propositional Variables***
  - Booleans, math edition!
- ***Propositional Connectives***
  - Linking things together.
- ***Truth Tables***
  - Rigorously defining connectives.
- ***Simplifying Negations***
  - Mechanically computing negations.

$TakeMath51 \lor TakeCME100$

$\neg FirstSucceed \rightarrow TryAgain$

$IsCardinal \land IsWhite$

$$\text{TakeMath51} \lor \text{TakeCME100}$$

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

$$\text{IsCardinal} \land \text{IsWhite}$$

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

$$\text{TakeMath51} \lor \text{TakeCME100}$$

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

$$\text{IsCardinal} \land \text{IsWhite}$$

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

Propositional Variables

- In propositional logic, individual propositions are represented by ***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!
- 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:

$$\neg \boldsymbol{p}$$

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

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

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

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

- 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

- 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.

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 connectives we've seen so far:

$\neg$ $\wedge$ $\vee$ $\top$ $\bot$

Summary of Important Points

- The $\vee$ 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.
-
-

Quick Question:

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

Quick Question:

- What would I have to show you to convince you that the statement

  $\boldsymbol{p} \vee \boldsymbol{q}$ is false?

de Morgan’s Laws

is equivalent to

$\neg p \lor \neg q$

is equivalent to

$\neg p \land \neg q$

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

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

Mathematical Implication

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?
- Pull out a sheet of paper, make a guess, and talk things over with your neighbors!

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

Important observation:
The statement $p \rightarrow q$ is true whenever $p \wedge \neg q$ is false.

p	q	p → 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.

An Important Equivalence

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

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

- Later on, this equivalence will be incredibly useful:

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

Another Important Equivalence

- Here's a useful equivalence. Start with

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

- By de Morgan's laws:

$$\begin{aligned} p \rightarrow q \quad & \textit{is equivalent to} \quad \neg(p \wedge \neg q) \\ & \textit{is equivalent to} \quad \neg p \vee \neg\neg q \\ & \textit{is equivalent to} \quad \neg p \vee q \end{aligned}$$

- Thus $p \rightarrow q$ is equivalent to $\neg p \vee q$

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

One interpretation of $\leftrightarrow$ 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:

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

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 $\rightarrow$.
- We will commonly write expressions like $p \wedge q \rightarrow r$ without adding parentheses.
- For more complex expressions, we'll try to add parentheses.
-

The Big Table

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

Recap So Far

- A ***propositional variable*** is a variable that is either true or false.
- The ***propositional connectives*** are
  - Negation: $\neg p$
  - Conjunction: $p \wedge q$
  - Disjunction: $p \vee q$
  - Truth: $\top$
  - Falsity: $\bot$
  - Implication: $p \to q$
  - Biconditional: $p \leftrightarrow q$

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.