PROPOSITIONAL LOGIC

Now that we've covered sets and how to construct them, let's take a look at one of the many ways we use them in propositional logic. In my opinion, propositional logic is one of the most important topics of CS70, because of how universal this language is throughout all areas of math and theoretical computer science. If you take any kind of theoretical upper division course (like CS170, EE126, EE127, etc.), you'll definitely encounter this language in those classes as well.

2.1 Propositions

At the heart of propositional logic, the basic building block that we're concerned with is what we call a **proposition**. A proposition is basically a sentence that is either true or false; the way you could think about is that it's a statement you can "propose" as a potential truth.

Example 2.1

The following are some examples of valid propositions:

- π is an irrational number.
- There are no real solutions x to the equation $x^2 + x + 1 = 0$.
- $1 + 1 = 5$.
- n is less than 10.

Remark : Notice that a proposition doesn't need to be true at all! The statement $1 + 1 = 5$ is a perfectly valid proposition, depsite it being blatantly false.

More rigorously, we label a proposition as a statement $P(n)$, which has some truth value based on the input to n.[1] In the first three statements above, even though they don't explicitly contain n as an input, they *implicitly* do – in the sense that they don't care about what the value of n is.

Because $P(n)$'s truth value is dependent on the input n, it should also make sense that $P(n)$ could be true for some values of n, and false for some others. Take the last proposition mentioned in the example:

$$P(n) = n \text{ is less than 10}$$

We know that when $n < 10$, then $P(n)$ is true, but when $n \geq 10$, then $P(n)$ is false. This is perfectly allowable in propositional logic. However, notice that there isn't a value of n for which $P(n)$ is true and false at the same time. This is known as the **law of the excluded middle**:

Theorem 2.1: Law of the Excluded Middle

Given a proposition $P(n)$, for every value of n, $P(n)$ is either true or false, but not both.

This principle should make intuitive sense: the whole point of coming up with propositions is so that we can talk about whether it's true or false for values of n, so if $P(n)$ could be true *and* false for some values of n, then how could we ever talk about the truth value of $P(n)$ rigorously?

2.2 Combining Propositions

Now that we know how to make propositions, let's talk about how we can combine propositions together to create new, more complex ones. In this vein, there are only three ways that we really combine propositions:

- **Conjunction:** combining two propositions with an "and", written like $R(n) = P(n) \wedge Q(n)$. Here, $R(n)$ is only true when $P(n)$ is true *and* $Q(n)$ is true.

[1] In this sense, you can almost think of $P(n)$ as a function based on the input n.

- **Disjunction:** combining two propositions with an "or", written like $R(n) = P(n) \lor Q(n)$. Here, $R(n)$ is true when either $P(n)$ is true *or* $Q(n)$ is true, or both.
- **Negation:** flipping the truth value of $P(n)$, written like $R(n) = \neg P(n)$. Here, $R(n)$ is true when $P(n)$ is false.

With these three methods, we can make any proposition we want!

2.2.1 Propositional Sentences

And now we come to the most important concept in propositional logic: the act of generating logical sentences from propositional statements. In a sense, constructing propositions $P(n)$ alone without combining them into a sentence is rather pointless, since we can't really do much with them. However, when we combine them into sentences, that's where we get our system of logic from. What do we mean by combining propositional sentences together? Let's consider the statement:

If n is an integer, then n^2 is an integer.

What do you notice here? This is basically the combination of two propositions! Specifically, if we let $P(n)$ be the statement "n is an integer" and $Q(n)$ be the statement "n^2 is an integer", then this statement basically simplifies to "If $P(n)$ holds, then $Q(n)$ holds". Mathematically, what we've just constructed is what's known as an **implication**, defined below:

> Definition 2.1 (Implication): An *implication* between two propositional statements $P(n)$ and $Q(n)$ is equivalent to the statement "If $P(n)$, then $Q(n)$". We write that as $P(n) \implies Q(n)$.

Before we move on, let's fully understand what an implication is doing: remember that $P(n) \implies Q(n)$ is the same as saying "If $P(n)$ is true, then $Q(n)$ is true". So, another way you can understand that as is that *under the condition* that $P(n)$ is true, then $Q(n)$ is also true. Using the example we had above, we can say that *under the condition* that n is an integer, then n^2 is also an integer.

> Remark : In terms of mathematical theorems that follow the form $P(n) \implies Q(n)$, this is what we mean as well. We first assume that $P(n)$ is true, then the theorem tells us that $Q(n)$ is true as well.

Along with $P \implies Q$, there's also two other sentences with P and Q that are commonly introduced here:

> Definition 2.2 (Contrapositive, Converse): Given an implication $P \implies Q$, the *contrapositive* and *converse* are written as follows:
>
> - **Contrapositive:** $\neg Q \implies \neg P$
> - **Converse:** $Q \implies P$

> Warning : Note that contrapositive and converse are not the same thing! For an implication $P \implies Q$, the contrapositive is $\neg Q \implies \neg P$, while the converse is $Q \implies P$. As we'll see in the next section, the contrapositive is logically equivalent to the original implication, whereas the converse is not.

There's one last detail about implications you need to know: if both $P(n) \implies Q(n)$ and $Q(n) \implies P(n)$ is also true, then this is what we call an **if and only if** statement. In English, we'd say "$P(n)$ if and only if $Q(n)$".

> Notation (If and only if): Given two propositional statements $P(n)$ and $Q(n)$, if $P(n)$ is true if and only if $Q(n)$ is true, then we write $P(n) \iff Q(n)$.

An if and only if (also referred to as iff) statement is powerful because it gives us the ability to say for certain that if *either* $P(n)$ or $Q(n)$ is true, then the other is true automatically. Later when we visit proofs, you'll hopefully appreciate why this is such a powerful condition.

2.3 Logical Equivalence

Now that we can build an infinite number of propositional sentences, how can we tell whether two of these sentences are saying the same thing? As you'll discover, there *are* some propositional forms that initialy look very different, but are in fact are logically equivalent (i.e. mean the same thing). These are particularly useful becuase it can sometimes drastically simplify how we go about proving statements.

How do we determine logical equivalence? Let's look at the theorem below:

Theorem 2.2: Logical Equivalence

If two propositional sentences P and Q have the same truth table, then they are *logically equivalent*.

Now what is a truth table? It's basically a table that summarizes the truth values that the sentence formed by P and Q can take on. Suppose we have the statements P and Q, and we look at the sentence $P \implies Q$:

P	Q	$P \implies Q$
T	T	T
T	F	F
F	T	T
F	F	T

This shows us that given the truth values of P and Q, what the truth value of $P \implies Q$ is. How did I come up with this table? Let's go through each combination of P and Q and see if $P \implies Q$ makes sense:

- $P = \text{T}, Q = \text{T}$. In this case, it is indeed true that a true statement implies a true statement, so this one makes sense.
- $P = \text{T}, Q = \text{F}$. True statements should only imply other true statements, since we want our mathematics to be logically consistent (i.e. truth implies truth). Therefore, a true statement cannot imply a false one, hence the truth value being false.
- $P = \text{F}, Q = \text{T/F}$. The last two cases here both fall under the same category: both of these fall under the category where our initial statement P was false. Imagine this: if you started off a proof with a false premise, then you can ostensibly prove anything you wanted, even false statements. Therefore, both the implications here are assigned true values.

Now that we've looked at $P \implies Q$, let's also take a look at the truth table for $\neg P \lor Q$:

P	Q	$\neg P \lor Q$
T	T	T
T	F	F
F	T	T
F	F	F

Notice that $P \implies Q$ and $\neg P \lor Q$ take on the same truth values given the same truth values to P and Q! Because they have the same truth table, then we say that $P \implies Q$ and $\neg P \lor Q$ are *logically equivalent*.

> Notation (Logical Equivalence): If two propositional sentences A and B are logically equivalent, then we write that as $A \equiv B$.

In our case, we'd write $P \implies Q \equiv \neg P \lor Q$. Another sentence which is logically equivalent to $P \implies Q$ is the contrapositive:

$$\neg Q \implies \neg P$$

This fact is particularly useful since in proofs it's sometimes easier to prove $\neg Q \implies \neg P$ than $P \implies Q$, and because they're logically equivalent statements, proving one proves the other as well!

Exercise 2.1

Verify that $P \implies Q$ and $\neg Q \implies \neg P$ have the same truth tables.

2.4 Quantifiers

We've covered how to make propositional sentences $P(n)$, now it's time to focus on the part with n: how do we specify to the reader what values can n take on? This is the role of quantifiers in propositional phrases.

There are only two quantifiers in propositional logic:

- **Universal:** Denoted by $\forall$, it refers to all the elements of a particular set we define.
- **Existential:** Denoted by $\exists$, it refers to the existence of an object within a set of our choice.

Now we are finally ready to fully create our first propositional logic statement: we're going to transform the sentence:

If n is an integer, then n^2 is an integer

completely into propositional language. To do this, the first thing we'll want to do is to tell the reader what set of numbers n lives in: in this case, it's the integers. Next, the statement makes no reference to the existence of a particular n, so we'll want to use the universal quantifier $\forall$ here. Finally, as before, let $P(n)$ be the statement that n is an integer, and $Q(n)$ be the statement that $Q(n)$ is an integer. Then, we can write:

$$(\forall x \in \mathbb{Z})(P(n) \implies Q(n))$$

And that's the complete transformation of our sentence into mathematical terms! The parentheses here aren't absolutely necessary; you'll see some textbooks that don't use them at all, but I use them here becuase I think they're useful to highlight the different parts of our phrase.

The box below shows a more complex sentence translation, and I encourage you to study it because it involves a technique that is very common in propositional logic.

Example 2.2

Let's try converting the following phrase into propositional logic:

There exists only two distinct real solutions to the equation $x^2 - 1 = 0$.

Where do we even start with this one? First, let's handle the fact that there are two real solutions, and worry about the "only" keyword later. To write the fact that there are two distinct solutions, what we can do is say instead is that there are two numbers x, y such that $x^2 - 1 = 0$ and $y^2 - 1 = 0$, and $x \neq y$. Written in propositional logic, this is what the phrase looks like so far:

$$(\exists x, y \in \mathbb{R})(x^2 - 1 = 0 \land y^2 - 1 = 0 \land x \neq y)$$

Now, how do we say that there are "only" two solutions? The trick is to first *suppose* that there is a third solution z, then say that if z also solves this equation, then z is either x or y[a] Therefore, the full equation is:

$$(\forall z \in \mathbb{R})(\exists x, y \in \mathbb{R})(x^2 - 1 = 0 \land y^2 - 1 = 0 \land x \neq y) \land (z^2 - 1 = 0 \implies z = x \lor z = y)$$

[a]Try convincing yourself as to why this is valid.

And that's all for quantifiers! The final sentence we've made in this example box is a little complex, but spend some time with it, and see if you can identify the purpose of each piece in the sentence and how we joined them together.

2.5 De Morgan's Laws

De Morgan's laws refer to how the negation operator $\neg$ interacts with the other objects we've explored in this chapter. In terms of the conjunction $\land$ and disjunction $\lor$ symbols, De Morgan's laws says that:

$$\neg(P \land Q) \equiv \neg P \lor \neg Q \quad \neg(P \lor Q) \equiv \neg P \land \neg Q$$

You can check using truth tables that these are in fact logically equivalent. The reason these laws are so powerful is because they allow us to potentially discover that two statements are in fact logically equivalent, without having to go through the pain of making a truth table.

What about existential quantifiers? How does negation affect those symbols? Well, let's say you have a statement $\forall x P(x)$, in other words that $P(x)$ is true for all values of x. What would the negation of this statement be? One way to say the opposite of this statement is to say that there *exists* a value of x such that $P(x)$ is true, or equivalently that $\neg P(x)$ is true. And with that, we've discovered how negation works with quanitifers:

$$\neg(\forall x P(x)) \equiv \exists x \neg P(x)$$