CS 70 Discrete Mathematics and Probability Theory
Spring 2019 Course Notes

Note 1

A Brief Introduction

Welcome to Discrete Math and Probability Theory! You might be wondering what you've gotten yourself into — we're delighted to tell you that the answer is something quite remarkable. For as you will find in this course, computer science is a unique field which straddles the fine line between a wealth of research areas: Natural sciences such as physics and chemistry, applied fields such as engineering, and abstract areas such as mathematics. It is precisely this pervasive diversity that not only allows computer science to have something to offer for everyone, but also makes it a driving force of life as we know it today.

So let us begin by dispelling a common myth: What exactly *is* computer science? Is it programming in Java or C++? Networking via the Internet? Or perhaps videogames and troubleshooting personal computers? The answer is that *all* of these are indeed aspects of computer science; they form part of the "engineering" or applied sides of the field, which works to create useful, reliable, working systems. Yet, this is just part of the answer. For at the heart of computer science lies a fundamental core of highly theoretical areas such as artificial intelligence, algorithms, cryptography, and even quantum computing. These areas investigate questions such as: How can we make a robot smart? Find the shortest path between cities on the map? Or send secretly coded messages to one another? It is this theoretical foundation which makes computer science fundamentally possible.

Sound interesting so far? Good. In this course, our goal is to equip you with the basic tools and language you need to move forward in your study of computer science. Which language are we talking about? Why, mathematics, of course! It turns out that the underlying principles of computer science can be described very elegantly using mathematics, particularly in theoretical branches.

With that said, it is quite understandable if aspects of this material seem at first abstract and difficult to master. Yet, do not be discouraged. Any beginning guitarist, for example, dreams of shredding on stage like, say, Jimi Hendrix (or whoever their favorite musician happens to be). To attain this level of mastery, however, first requires the development of basic terminology and skills, such as the ability to read music and to play scales. In this sense, computer science is no different; there are basic mathematical tools which first need to be internalized. Thus, if at any time it should strike you as difficult to appreciate the material in these notes, fret not, and remind yourself that exciting ideas in computer science lie just ahead!

High-level overview of the course. This course can be thought of as consisting of two halves. The first begins with the basic language of mathematics: Logic and proofs. At first, these two topics may seem completely disconnected from anything related to computers; however, it is precisely their study which led to the invention of computers to begin with! Moreover, logic is directly related to digital circuit design, and proofs by induction play a central role in analyzing programs. Indeed, induction is deeply connected to recursion. In this section, developing your ability to write coherent, rigorous proofs will be an important focus. Next, we move on to the study of a special type of arithmetic known as modular arithmetic. The beautiful properties of such numbers leads directly to schemes for computer security, and for storing and transmitting data so that it is immune to noise.

The second half of the course will introduce you to the basic principles of probability theory. This subject plays a crucial role in every aspect of computer science today, from machine learning and artificial intelligence, to dealing with massive data sets, to computer security.

Finally, let us close this introduction by highlighting one of the primary skills this course will allow you to develop: Problem solving. Indeed, one of the hallmarks of discrete mathematics is its wealth of interesting and mind-stretching puzzles and problems. As you work through the homework questions, you will learn how to attack questions, even when at first you may have no idea what the answer might look like. This kind of problem solving ability will greatly help you in the rest of your computer science career, in addition to being a critical skill for many job interviews and an invaluable asset throughout life.

1 Propositional Logic

In order to be fluent in working with mathematical statements, you need to understand the basic framework of the language of mathematics. This first lecture, we will start by learning about what logical forms mathematical theorems may take, and how to manipulate those forms to make them easier to prove. In the next few lectures, we will learn several different methods of proving things.

Our first building block is the notion of a **proposition**, which is simply a statement which is either true or false.

These statements are all propositions:

(1) $\sqrt{3}$ is irrational.
(2) $1+1=5$.
(3) Julius Caesar had 2 eggs for breakfast on his 10^{th} birthday.

These statements are clearly not propositions:

(4) $2+2$.
(5) $x^2+3x=5$.

These statements aren't propositions either (although some books say they are). Propositions should not include fuzzy terms.

(6) Arnold Schwarzenegger often eats broccoli. (What is "often?")
(7) Henry VIII was unpopular. (What is "unpopular?")

Propositions may be joined together to form more complex statements. Let P, Q, and R be variables representing propositions (for example, P could stand for "3 is odd"). The simplest way of joining these propositions together is to use the connectives "and", "or" and "not."

(1) **Conjunction**: $P \wedge Q$ ("P and Q"). True only when both P and Q are true.

(2) **Disjunction**: $P \vee Q$ ("P or Q"). True when at least one of P and Q is true.

(3) **Negation**: $\neg P$ ("not P"). True when P is false.

Statements like these, with variables, are called *propositional forms*. Note that $P \vee \neg P$ is always true, regardless of the truth value of P. A propositional form that is always true regardless of the truth values of its variables is called a *tautology*. Conversely, a statement such as $P \wedge \neg P$, which is always false, is called a *contradiction*.

Sanity check! If we let P stand for the proposition "3 is odd," Q stand for "4 is odd," and R for "4+5=49," what are the values of $P \wedge R$, $P \vee R$ and $\neg Q$?

A useful tool for describing the possible values of a propositional form is a **truth table**. Truth tables are the same as function tables: you list all possible input values for the variables, and then list the outputs given those inputs. (The order does not matter.)

Here are truth tables for conjunction and negation:

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

P	$\neg P$
T	F
F	T

Sanity check! Write down the truth table for disjunction (OR).

The most important and subtle propositional form is an **implication**:

(4) **Implication**: $P \Longrightarrow Q$ ("P implies Q"). This is the same as "If P, then Q."

Here, P is called the *hypothesis* of the implication, and Q is the *conclusion*. [1]

We encounter implications frequently in everyday life; here are a couple of examples:

If you stand in the rain, then you'll get wet.
If you passed the class, you received a certificate.

An implication $P \Longrightarrow Q$ is false only when P is true and Q is false. For example, the first statement would be false only if you stood in the rain but didn't get wet. The second statement above would be false only if you passed the class yet you didn't receive a certificate.

[1] P is sometimes also called the *antecedent* and Q the *consequent*.

Here is the truth table for $P \implies Q$ (along with an extra column that we'll explain in a moment):

P	Q	$P \implies Q$	$\neg P \vee Q$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Note that $P \implies Q$ is always true when P is false. This means that many statements that sound nonsensical in English are true, mathematically speaking. Examples are statements like: "If pigs can fly, then horses can read" or "If 14 is odd then $1+2=18$." (These are statements that we never make in everyday life, but are perfectly natural in mathematics.) When an implication is stupidly true because the hypothesis is false, we say that it is *vacuously true*.

Note also that $P \implies Q$ is **logically equivalent** to $\neg P \vee Q$, as can be seen by comparing the last two columns of the above truth table: for all possible truth values of P and Q, $P \implies Q$ and $\neg P \vee Q$ take the same values (i.e., they have the same truth table). We write this as $(P \implies Q) \equiv (\neg P \vee Q)$.

$P \implies Q$ is the most common form mathematical theorems take. Here are some of the different ways of saying it:

(1) if P, then Q;
(2) Q if P;
(3) P only if Q;
(4) P is sufficient for Q;
(5) Q is necessary for P;
(6) Q unless not P.

If both $P \implies Q$ and $Q \implies P$ are true, then we say "P if and only if Q" (abbreviated "P iff Q"). Formally, we write $P \iff Q$. Note that $P \iff Q$ is true only when P and Q have the same truth values (both true or both false).

For example, if we let P be "3 is odd," Q be "4 is odd," and R be "6 is even," then the following are all true: $P \implies R$, $Q \implies P$ (vacuously), and $R \implies P$. Because $P \implies R$ and $R \implies P$, we also see that $P \iff R$ is true.

Given an implication $P \implies Q$, we can also define its

(a) **Contrapositive**: $\neg Q \implies \neg P$
(b) **Converse**: $Q \implies P$

The contrapositive of "If you passed the class, you received a certificate" is "If you did not get a certificate, you did not pass the class." The converse is "If you got a certificate, you passed the class." Does the contrapositive say the same thing as the original statement? Does the converse?

Let's look at the truth tables:

P	Q	$\neg P$	$\neg Q$	$P \implies Q$	$Q \implies P$	$\neg Q \implies \neg P$	$P \iff Q$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	F	F
F	T	T	F	T	F	T	F
F	F	T	T	T	T	T	T

Note that $P \implies Q$ and its contrapositive have the *same* truth values everywhere in their truth tables, so they are logically equivalent: $(P \implies Q) \equiv (\neg Q \implies \neg P)$. Many students confuse the contrapositive with the converse: note that $P \implies Q$ and $\neg Q \implies \neg P$ are logically equivalent, but $P \implies Q$ and $Q \implies P$ are not!

When two propositional forms are logically equivalent, we can think of them as "meaning the same thing." We will see next time how useful this can be for proving theorems.

2 Quantifiers

The mathematical statements you'll see in practice will not be made up of simple propositions like "3 is odd." Instead you'll see statements like:

(1) For all natural numbers n, n^2+n+41 is prime.
(2) If n is an odd integer, so is n^3.
(3) There is an integer k that is both even and odd.

In essence, these statements assert something about lots of simple propositions (even infinitely many!) all at once. For instance, the first statement is asserting that 0^2+0+41 is prime, 1^2+1+41 is prime, and so on. The last statement is saying that, as k ranges over every possible integer, we will find at least one value for which the statement is satisfied.

Why are the above three examples considered to be propositions, while earlier we claimed that "$x^2+3x=5$" was not? The reason is that in these three examples, there is an underlying "universe" that we are working in. The statements are then *quantified* over that universe. To express these statements mathematically we need two **quantifiers**: The *universal quantifier* $\forall$ ("for all") and the *existential quantifer* $\exists$ ("there exists"). Note that in a *finite* universe, we can express existentially and universally quantified propositions without quantifiers, using disjunctions and conjunctions respectively. For example, if our universe U is $\{1, 2, 3, 4\}$, then $\exists x P(x)$ is logically equivalent to $P(1) \vee P(2) \vee P(3) \vee P(4)$, and $\forall x P(x)$ is logically equivalent to $P(1) \wedge P(2) \wedge P(3) \wedge P(4)$. However, in an infinite universe, such as the natural numbers, this is not possible.

Examples:

(1) "Some mammals lay eggs." Mathematically, "some" means "at least one," so the statement is saying "There exists a mammal x such that x lays eggs." If we let our universe U be the set of mammals, then we can write: $(\exists x \in U)(x \text{ lays eggs})$. (Sometimes, when the universe is clear, we omit U and simply write $\exists x(x \text{ lays eggs})$.)

(2) "For all natural numbers n, n^2+n+41 is prime," can be expressed by taking our universe to be the set of natural numbers, denoted as $\mathbb{N}$: $(\forall n \in \mathbb{N})(n^2+n+41 \text{ is prime})$.

Sanity check! Use quantifiers to express the following two statements: "For all integers x, $2x+1$ is odd", and "There exists an integer between 2 and 4".

Some statements can have multiple quantifiers. As we will see, however, quantifiers do not commute. You can see this just by thinking about English statements. Consider the following (rather gory) example:

Example:

"Every time I ride the subway in New York, somebody gets stabbed."
"There is someone, such that every time I ride the subway in New York, that someone gets stabbed."

The first statement is saying that every time I ride the subway someone gets stabbed, but it could be a different person each time. The second statement is saying something truly horrible: that there is some poor guy Joe with the misfortune that every time I get on the New York subway, there is Joe, getting stabbed again. (Poor Joe will run for his life the second he sees me.)

Mathematically, we are quantifying over two universes: $T = \{\text{times when I ride on the subway}\}$ and $P = \{\text{people}\}$. The first statement can be written: $(\forall t \in T)(\exists p \in P)(p \text{ gets stabbed at time } t)$. The second statement says: $(\exists p \in P)(\forall t \in T)(p \text{ gets stabbed at time } t)$.

Let's look at a more mathematical example:

Consider

1. $(\forall x \in \mathbb{Z})(\exists y \in \mathbb{Z})(x < y)$
2. $(\exists y \in \mathbb{Z})(\forall x \in \mathbb{Z})(x < y)$

The first statement says that, given an integer, I can find a larger one. The second statement says something very different: that there is a largest integer! The first statement is true, the second is not.

3 Much Ado About Negation

What does it mean for a proposition P to be false? It means that its negation $\neg P$ is true. It's helpful to have some rules for working with negation, as will become more obvious next lecture when we look at proofs.

First, let's look at how to negate conjunctions and disjunctions:

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

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

These two equivalences are known as *De Morgan's Laws*, and they are quite intuitive: for example, if it is not the case that $P \land Q$ is true, then either P or Q must be false.

Sanity check! Verify both of De Morgan's Laws by writing down the appropriate truth tables.

Negating propositions involving quantifiers actually follows analogous laws. Let's start with a simple example. Assume that the universe is $\{1,2,3,4\}$ and let $P(x)$ denote the propositional formula "$x^2 > 10$." Check that $\exists x P(x)$ is true but $\forall x P(x)$ is false. Observe that both $\neg(\forall x P(x))$ and $\exists x \neg P(x)$ are true because $P(1)$ is false. Also note that both $\forall x \neg P(x)$ and $\neg(\exists x P(x))$ are false, since $P(4)$ is true. The fact that each pair of statements had the same truth value is no accident, as the equivalences

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

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

are laws that hold for any proposition P quantified over any universe (including infinite ones).

It is helpful to think of English sentences to convince yourself (informally) that these laws are true. For example, assume that we are working within the universe $\mathbb{Z}$ (the set of all integers), and that $P(x)$ is the proposition "x is odd." We know that the statement $(\forall x P(x))$ is false, since not every integer is odd. Therefore, we expect its negation, $\neg(\forall x P(x))$, to be true. But how would you say the negation in English? Well, if it is not true that every integer is odd, then that must mean there is some integer which is not odd (i.e., even). How would this be written in propositional form? That's easy, it's just: $(\exists x \neg P(x))$.

To see a more complex example, fix some universe and propositional formula $P(x,y)$. Assume we have the proposition $\neg(\forall x \exists y P(x,y))$ and we want to push the negation operator inside the quantifiers. By the above laws, we can do it as follows:

$$\neg(\forall x \exists y P(x,y)) \equiv \exists x \neg(\exists y P(x,y)) \equiv \exists x \forall y \neg P(x,y).$$

Notice that we broke the complex negation into a smaller, easier problem as the negation propagated itself through the quantifiers. Note also that the quantifiers "flip" as we go.

Let's look at a trickier example:

Write the sentence "there are at least three distinct integers x that satisfy $P(x)$" as a proposition using quantifiers! One way to do it is

$$\exists x \exists y \exists z (x \neq y \wedge y \neq z \wedge z \neq x \wedge P(x) \wedge P(y) \wedge P(z)).$$

(Here all quantifiers are over the universe $\mathbb{Z}$ of integers.) Now write the sentence "there are **at most** three distinct integers x that satisfy $P(x)$" as a proposition using quantifiers. One way to do it is

$$\exists x \exists y \exists z \forall d (P(d) \implies d = x \vee d = y \vee d = z).$$

Here is an equivalent way to do it:

$$\forall x \forall y \forall v \forall z ((x \neq y \wedge y \neq v \wedge v \neq x \wedge x \neq z \wedge y \neq z \wedge v \neq z) \implies \neg(P(x) \wedge P(y) \wedge P(v) \wedge P(z))).$$

[Check that you understand both of the above alternatives.] Finally, what if we want to express the sentence "there are **exactly** three distinct integers x that satisfy $P(x)$"? This is now easy: we can just use the *conjunction* of the two propositions above.