pay attention to dates!

Spring 2011 Version

Mathematics for Computer Science

revised Sunday 6th February, 2011, 03:08

Prof. Albert R Meyer
Massachusets Institute of Technology

Section by section

Full TOC

Contents

I Proofs

II Structures

III Counting

IV Probability

I Proofs

Reading this after
reading chap 1 + 2
of Fall 2010
book

Introduction

This text explains how to use mathematical models and methods to analyze problems that arise in computer science. Proofs play a central role in this work because the authors share a belief with most mathematicians that proofs are essential for genuine understanding. Proofs also play a growing role in computer science; they are used to certify that software and hardware will *always* behave correctly, something that no amount of testing can do.

Simply put, a proof is a method of establishing truth. Like beauty, "truth" sometimes depends on the eye of the beholder, and it should not be surprising that what constitutes a proof differs among fields. For example, in the judicial system, *legal* truth is decided by a jury based on the allowable evidence presented at trial. In the business world, *authoritative* truth is specified by a trusted person or organization, or maybe just your boss. In fields such as physics or biology, *scientific* truth[1] is confirmed by experiment. In statistics, *probable* truth is established by statistical analysis of sample data.

Philosophical proof involves careful exposition and persuasion typically based on a series of small, plausible arguments. The best example begins with "Cogito ergo sum," a Latin sentence that translates as "I think, therefore I am." It comes from the beginning of a 17th century essay by the mathematician/philosopher, René Descartes, and it is one of the most famous quotes in the world: do a web search on the phrase and you will be flooded with hits.

Deducing your existence from the fact that you're thinking about your existence is a pretty cool and persuasive-sounding idea. However, with just a few more lines

[1] Actually, only scientific *falsehood* can be demonstrated by an experiment—when the experiment fails to behave as predicted. But no amount of experiment can confirm that the *next* experiment won't fail. For this reason, scientists rarely speak of truth, but rather of *theories* that accurately predict past, and anticipated future, experiments.

of argument in this vein, Descartes goes on to conclude that there is an infinitely beneficent God. Whether or not you believe in an infinitely beneficent God, you'll probably agree that any very short "proof" of God's infinite beneficence is bound to be far-fetched. So even in masterful hands, this approach is not reliable.

Mathematics has its own specific notion of "proof."

Definition. A *mathematical proof* of a *proposition* is a chain of *logical deductions* leading to the proposition from a base set of *axioms*.

The three key ideas in this definition are highlighted: *proposition*, *logical deduction*, and *axiom*. In the next Chapter, we'll discuss these three ideas along with some basic ways of organizing proofs.

Problems for Section 0.0

Class Problems

Problem 0.1.
Identify exactly where the bugs are in each of the following bogus proofs.[2]

(a) Bogus Claim: $1/8 > 1/4$.

Bogus proof.

$$\begin{aligned} 3 &> 2 \\ 3\log_{10}(1/2) &> 2\log_{10}(1/2) \\ \log_{10}(1/2)^3 &> \log_{10}(1/2)^2 \\ (1/2)^3 &> (1/2)^2, \end{aligned}$$

and the claim now follows by the rules for multiplying fractions. ■

(b) *Bogus proof*: $1¢ = \$0.01 = (\$0.1)^2 = (10¢)^2 = 100¢ = \$1$. ■

(c) Bogus Claim: If a and b are two equal real numbers, then $a = 0$.

[2] From Stueben, Michael and Diane Sandford. *Twenty Years Before the Blackboard*, Mathematical Association of America, ©1998.

Bogus proof.

$$\begin{aligned} a &= b \\ a^2 &= ab \\ a^2 - b^2 &= ab - b^2 \\ (a-b)(a+b) &= (a-b)b \\ a+b &= b \\ a &= 0. \end{aligned}$$

■

Problem 0.2.
It's a fact that the Arithmetic Mean is at least as large the Geometric Mean, namely,

$$\frac{a+b}{2} \geq \sqrt{ab}$$

for all nonnegative real numbers a and b. But there's something objectionable about the following proof of this fact. What's the objection, and how would you fix it?

Bogus proof.

$$\begin{aligned} \frac{a+b}{2} &\overset{?}{\geq} \sqrt{ab}, && \text{so} \\ a+b &\overset{?}{\geq} 2\sqrt{ab}, && \text{so} \\ a^2 + 2ab + b^2 &\overset{?}{\geq} 4ab, && \text{so} \\ a^2 - 2ab + b^2 &\overset{?}{\geq} 0, && \text{so} \\ (a-b)^2 &\geq 0 && \text{which we know is true.} \end{aligned}$$

The last statement is true because $a - b$ is a real number, and the square of a real number is never negative. This proves the claim. ■

Problem 0.3.
Albert announces to his class that he plans to surprise them with a quiz sometime next week.

His students first wonder if the quiz could be on Friday of next next. They reason that it can't: if Albert didn't give the quiz *before* Friday, then by midnight Thursday, they would know the quiz had to be on Friday, and so the quiz wouldn't be a surprise any more.

Next the students wonder whether Albert could give the surprise quiz Thursday. They observe that if the quiz wasn't given *before* Thursday, it would have to be given *on* the Thursday, since they already know it can't be given on Friday. But having figured that out, it wouldn't be a surprise if the quiz was on Thursday either. Similarly, the students reason that the quiz can't be on Wednesday, Tuesday, or Monday. Namely, it's impossible for Albert to give a surprise quiz next week. All the students now relax, having concluded that Albert must have been bluffing.

And since no one expects the quiz, that's why, when Albert gives it on Tuesday next week, it really is a surprise!

What do you think is wrong with the students' reasoning?

1 What is a Proof?

1.1 Propositions

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

For example, both of the following statements are propositions. The first is true and the second is false.

Proposition 1.1.1. *2 + 3 = 5.*

Proposition 1.1.2. *1 + 1 = 3.*

Being true or false doesn't sound like much of a limitation, but it does exclude statements such as, "Wherefore art thou Romeo?" and "Give me an *A*!". It is also excludes statements whose truth varies with circumstance such as, "It's five o'clock," or "the stock market will rise tomorrow."

Unfortunately it is not always easy to decide if a proposition is true or false:

Proposition 1.1.3. *For every nonnegative integer, n, the value of $n^2 + n + 41$ is prime.*

(A *prime* is an integer greater than one that is not divisible by any other integer greater than 1, for example, 2, 3, 5, 7, 11, ...) Let's try some numerical experimentation to check this proposition. Let [1]

$$p(n) ::= n^2 + n + 41. \tag{1.1}$$

We begin with $p(0) = 41$ which is prime. $p(1) = 43$ which is prime. $p(2) = 47$ which is prime. $p(3) = 53$ which is prime. $\dots p(20) = 461$ which is prime. Hmmm, starts to look like a plausible claim. In fact we can keep checking through $n = 39$ and confirm that $p(39) = 1601$ is prime.

But $p(40) = 40^2 + 40 + 41 = 41 \cdot 41$, which is not prime. So it's not true that the expression is prime *for all* nonnegative integers. In fact, it's not hard to show that *no* polynomial with integer coefficients can map all natural numbers into prime numbers, unless it's a constant (see Problem 1.3). The point is that in general you can't check a claim about an infinite set by checking a finite set of its elements, no matter how large the finite set.

[1] The symbol ::= means "equal by definition." It's always ok to simply write "=" instead of ::=, but reminding the reader that an equality holds by definition can be helpful.

By the way, propositions like this about *all* numbers or all items of some kind are so common that there is a special notation for them. With this notation, Proposition 1.1.3 would be

$$\forall n \in \mathbb{N}.\ p(n) \text{ is prime.} \tag{1.2}$$

Here the symbol $\forall$ is read "for all". The symbol $\mathbb{N}$ stands for the set of *nonnegative integers*, namely, 0, 1, 2, 3, ... (ask your instructor for the complete list). The symbol "$\in$" is read as "is a member of," or "belongs to," or simply as "is in". The period after the $\mathbb{N}$ is just a separator between phrases.

Here are two even more extreme examples:

Proposition 1.1.4. *$a^4 + b^4 + c^4 = d^4$ has no solution when a, b, c, d are positive integers.*

Euler (pronounced "oiler") conjectured this in 1769. But the proposition was proven false 218 years later by Noam Elkies at a liberal arts school up Mass Ave. The solution he found was $a = 95800, b = 217519, c = 414560, d = 422481$.

In logical notation, Proposition 1.1.4 could be written,

$$\forall a \in \mathbb{Z}^+\ \forall b \in \mathbb{Z}^+\ \forall c \in \mathbb{Z}^+\ \forall d \in \mathbb{Z}^+.\ a^4 + b^4 + c^4 \neq d^4.$$

Here, $\mathbb{Z}^+$ is a symbol for the positive integers. Strings of $\forall$'s like this are usually abbreviated for easier reading:

$$\forall a, b, c, d \in \mathbb{Z}^+.\ a^4 + b^4 + c^4 \neq d^4.$$

Proposition 1.1.5. *$313(x^3 + y^3) = z^3$ has no solution when $x, y, z \in \mathbb{Z}^+$.*

This proposition is also false, but the smallest counterexample has more than 1000 digits!

Proposition 1.1.6. *Every map can be colored with 4 colors so that adjacent[2] regions have different colors.*

This proposition is true and is known as the "*Four-Color Theorem*". However, there have been many incorrect proofs, including one that stood for 10 years in the late 19th century before the mistake was found. An laborious proof was finally found in 1976 by mathematicians Appel and Haken, who used a complex computer program to categorize the four-colorable maps; the program left a few thousand maps uncategorized, and these were checked by hand by Haken and his assistants—including his 15-year-old daughter. There was a lot of debate about

[2] Two regions are adjacent only when they share a boundary segment of positive length. They are not considered to be adjacent if their boundaries meet only at a few points.

whether this was a legitimate proof: the proof was too big to be checked without a computer, and no one could guarantee that the computer calculated correctly, nor did anyone have the energy to recheck the four-colorings of thousands of maps that were done by hand. Within the past decade a mostly intelligible proof of the Four-Color Theorem was found, though a computer is still needed to check colorability of several hundred special maps. [3]

Proposition 1.1.7 (Goldbach)**.** *Every even integer greater than 2 is the sum of two primes.*

No one knows whether this proposition is true or false. It is known as *Goldbach's Conjecture*, and dates back to 1742.

For a computer scientist, some of the most important things to prove are the correctness of programs and systems—whether a program or system does what it's supposed to. Programs are notoriously buggy, and there's a growing community of researchers and practitioners trying to find ways to prove program correctness. These efforts have been successful enough in the case of CPU chips that they are now routinely used by leading chip manufacturers to prove chip correctness and avoid mistakes like the notorious Intel division bug in the 1990's.

Developing mathematical methods to verify programs and systems remains an active research area. We'll consider some of these methods later in the course.

1.2 Predicates

A *predicate* is a proposition whose truth depends on the value of one or more variables. Most of the propositions above were defined in terms of predicates. For example,

$$\text{“}n \text{ is a perfect square”}$$

is a predicate whose truth depends on the value of n. The predicate is true for $n = 4$ since four is a perfect square, but false for $n = 5$ since five is not a perfect square.

Like other propositions, predicates are often named with a letter. Furthermore, a function-like notation is used to denote a predicate supplied with specific variable values. For example, we might name our earlier predicate P:

$$P(n) ::= \text{“}n \text{ is a perfect square”}$$

[3] The story of the Four-Color Proof is told in a well-reviewed popular (non-technical) book: "Four Colors Suffice. How the Map Problem was Solved." *Robin Wilson*. Princeton Univ. Press, 2003, 276pp. ISBN 0-691-11533-8.

Now $P(4)$ is true, and $P(5)$ is false.

This notation for predicates is confusingly similar to ordinary function notation. If P is a predicate, then $P(n)$ is either *true* or *false*, depending on the value of n. On the other hand, if p is an ordinary function, like n^2+1, then $p(n)$ is a *numerical quantity*. **Don't confuse these two!**

1.3 The Axiomatic Method

The standard procedure for establishing truth in mathematics was invented by Euclid, a mathematician working in Alexandria, Egypt around 300 BC. His idea was to begin with five *assumptions* about geometry, which seemed undeniable based on direct experience. (For example, "There is a straight line segment between every pair of points.) Propositions like these that are simply accepted as true are called *axioms*.

Starting from these axioms, Euclid established the truth of many additional propositions by providing "proofs". A *proof* is a sequence of logical deductions from axioms and previously-proved statements that concludes with the proposition in question. You probably wrote many proofs in high school geometry class, and you'll see a lot more in this course.

There are several common terms for a proposition that has been proved. The different terms hint at the role of the proposition within a larger body of work.

- Important propositions are called *theorems*.
- A *lemma* is a preliminary proposition useful for proving later propositions.
- A *corollary* is a proposition that follows in just a few logical steps from a theorem.

The definitions are not precise. In fact, sometimes a good lemma turns out to be far more important than the theorem it was originally used to prove.

Euclid's axiom-and-proof approach, now called the *axiomatic method*, is the foundation for mathematics today. In fact, just a handful of axioms, called the axioms Zermelo-Frankel with Choice (ZFC), together with a few logical deduction rules, appear to be sufficient to derive essentially all of mathematics. We'll examine these in Chapter 4.

1.4 Our Axioms

The ZFC axioms are important in studying and justifying the foundations of mathematics, but for practical purposes, they are much too primitive. Proving theorems in ZFC is a little like writing programs in byte code instead of a full-fledged programming language—by one reckoning, a formal proof in ZFC that $2 + 2 = 4$ requires more than 20,000 steps! So instead of starting with ZFC, we're going to take a *huge* set of axioms as our foundation: we'll accept all familiar facts from high school math!

This will give us a quick launch, but you may find this imprecise specification of the axioms troubling at times. For example, in the midst of a proof, you may find yourself wondering, "Must I prove this little fact or can I take it as an axiom?" Feel free to ask for guidance, but really there is no absolute answer. Just be up front about what you're assuming, and don't try to evade homework and exam problems by declaring everything an axiom!

1.4.1 Logical Deductions

Logical deductions or *inference rules* are used to prove new propositions using previously proved ones.

A fundamental inference rule is *modus ponens*. This rule says that a proof of P together with a proof that P IMPLIES Q is a proof of Q.

Inference rules are sometimes written in a funny notation. For example, *modus ponens* is written:

Rule.

$$\frac{P, \quad P \text{ IMPLIES } Q}{Q}$$

When the statements above the line, called the *antecedents*, are proved, then we can consider the statement below the line, called the *conclusion* or *consequent*, to also be proved.

A key requirement of an inference rule is that it must be *sound*: any assignment of truth values that makes all the antecedents true must also make the consequent true. So if we start off with true axioms and apply sound inference rules, everything we prove will also be true.

There are many other natural, sound inference rules, for example:

Rule.

$$\frac{P \text{ IMPLIES } Q, \quad Q \text{ IMPLIES } R}{P \text{ IMPLIES } R}$$

Rule.

$$\frac{\text{NOT}(P) \text{ IMPLIES NOT}(Q)}{Q \text{ IMPLIES } P}$$

On the other hand,

Rule.

$$\frac{\text{NOT}(P) \text{ IMPLIES NOT}(Q)}{P \text{ IMPLIES } Q}$$

is not sound: if P is assigned **T** and Q is assigned **F**, then the antecedent is true and the consequent is not.

Note that a propositional inference rule is sound precisely when the conjunction (AND) of all its antecedents implies its consequent.

As with axioms, we will not be too formal about the set of legal inference rules. Each step in a proof should be clear and "logical"; in particular, you should state what previously proved facts are used to derive each new conclusion.

1.4.2 Patterns of Proof

In principle, a proof can be *any* sequence of logical deductions from axioms and previously proved statements that concludes with the proposition in question. This freedom in constructing a proof can seem overwhelming at first. How do you even *start* a proof?

My problem

Here's the good news: many proofs follow one of a handful of standard templates. Each proof has it own details, of course, but these templates at least provide you with an outline to fill in. We'll go through several of these standard patterns, pointing out the basic idea and common pitfalls and giving some examples. Many of these templates fit together; one may give you a top-level outline while others help you at the next level of detail. And we'll show you other, more sophisticated proof techniques later on.

The recipes below are very specific at times, telling you exactly which words to write down on your piece of paper. You're certainly free to say things your own way instead; we're just giving you something you *could* say so that you're never at a complete loss.

nice

1.5 Proving an Implication

Propositions of the form "If P, then Q" are called *implications*. This implication is often rephrased as "P IMPLIES Q."

Here are some examples:

- (Quadratic Formula) If $ax^2 + bx + c = 0$ and $a \neq 0$, then

$$x = \left(-b \pm \sqrt{b^2 - 4ac}\right)/2a.$$

oh to back out x

- (Goldbach’s Conjecture) If n is an even integer greater than 2, then n is a sum of two primes.

never thought of it that way!

- If $0 \leq x \leq 2$, then $-x^3 + 4x + 1 > 0$.

There are a couple of standard methods for proving an implication.

1.5.1 Method #1

In order to prove that P IMPLIES Q:

1. Write, “Assume P.”
2. Show that Q logically follows.

Example

Theorem 1.5.1. *If* $0 \leq x \leq 2$, *then* $-x^3 + 4x + 1 > 0$.

Before we write a proof of this theorem, we have to do some scratchwork to figure out why it is true.

The inequality certainly holds for $x = 0$; then the left side is equal to 1 and $1 > 0$. As x grows, the $4x$ term (which is positive) initially seems to have greater magnitude than $-x^3$ (which is negative). For example, when $x = 1$, we have $4x = 4$, but $-x^3 = -1$ only. In fact, it looks like $-x^3$ doesn’t begin to dominate until $x > 2$. So it seems the $-x^3 + 4x$ part should be nonnegative for all x between 0 and 2, which would imply that $-x^3 + 4x + 1$ is positive.

So far, so good. But we still have to replace all those “seems like” phrases with solid, logical arguments. We can get a better handle on the critical $-x^3 + 4x$ part by factoring it, which is not too hard:

$$-x^3 + 4x = x(2 - x)(2 + x)$$

Aha! For x between 0 and 2, all of the terms on the right side are nonnegative. And a product of nonnegative terms is also nonnegative. Let’s organize this blizzard of observations into a clean proof.

& so this is key observation

This is key; study

Proof. Assume $0 \leq x \leq 2$. Then x, $2-x$, and $2+x$ are all nonnegative. Therefore, the product of these terms is also nonnegative. Adding 1 to this product gives a positive number, so:

$$x(2 - x)(2 + x) + 1 > 0$$

Multiplying out on the left side proves that

$$-x^3 + 4x + 1 > 0$$

as claimed. ■

There are a couple points here that apply to all proofs:

- You'll often need to do some scratchwork while you're trying to figure out the logical steps of a proof. Your scratchwork can be as disorganized as you like—full of dead-ends, strange diagrams, obscene words, whatever. But keep your scratchwork separate from your final proof, which should be clear and concise.
- Proofs typically begin with the word "Proof" and end with some sort of doohickey like □ or "q.e.d". The only purpose for these conventions is to clarify where proofs begin and end.

1.5.2 Method #2 - Prove the Contrapositive

An implication ("P IMPLIES Q") is logically equivalent to its *contrapositive*

$$\text{NOT}(Q) \text{ IMPLIES } \text{NOT}(P)$$

Proving one is as good as proving the other, and proving the contrapositive is sometimes easier than proving the original statement. If so, then you can proceed as follows:

1. Write, "We prove the contrapositive:" and then state the contrapositive.
2. Proceed as in Method #1.

Example

Theorem 1.5.2. *If $\overset{P}{r}$ is irrational, then $\overset{Q}{\sqrt{r}}$ is also irrational.*

If $\sqrt{r}$ rational than r rational.

Recall that rational numbers are equal to a ratio of integers and irrational numbers are not. So we must show that if r is *not* a ratio of integers, then $\sqrt{r}$ is also *not* a ratio of integers. That's pretty convoluted! We can eliminate both *not*'s and make the proof straightforward by considering the contrapositive instead.

no I was right

Proof. We prove the contrapositive: if $\sqrt{r}$ is rational, then r is rational.

Assume that $\sqrt{r}$ is rational. Then there exist integers a and b such that:

$$\sqrt{r} = \frac{a}{b} \quad \text{def}$$

r rational ~~so $\sqrt{r}$ rational~~

Squaring both sides gives:

$$r = \frac{a^2}{b^2}$$

Since a^2 and b^2 are integers, r is also rational. ■

weird, I would not have thought of that
must really get rules of math

1.6 Proving an "If and Only If"

Many mathematical theorems assert that two statements are logically equivalent; that is, one holds if and only if the other does. Here is an example that has been known for several thousand years:

> Two triangles have the same side lengths if and only if two side lengths and the angle between those sides are the same.

The phrase "if and only if" comes up so often that it is often abbreviated "iff".

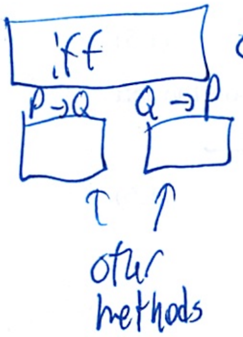
1.6.1 Method #1: Prove Each Statement Implies the Other

The statement "P IFF Q" is equivalent to the two statements "P IMPLIES Q" and "Q IMPLIES P". So you can prove an "iff" by proving *two* implications:

1. Write, "We prove P implies Q and vice-versa."
2. Write, "First, we show P implies Q." Do this by one of the methods in Section 1.5.

parts

3. Write, "Now, we show Q implies P." Again, do this by one of the methods in Section 1.5.



1.6.2 Method #2: Construct a Chain of Iffs

In order to prove that P is true iff Q is true:

1. Write, "We construct a chain of if-and-only-if implications."
2. Prove P is equivalent to a second statement which is equivalent to a third statement and so forth until you reach Q.

This method sometimes requires more ingenuity than the first, but the result can be a short, elegant proof.

$P \leftrightarrow \leftrightarrow \leftrightarrow Q$

Example

The *standard deviation* of a sequence of values $x_1, x_2, \ldots, x_n$ is defined to be:

$$\sqrt{\frac{(x_1-\mu)^2+(x_2-\mu)^2+\cdots+(x_n-\mu)^2}{n}} \tag{1.3}$$

where μ is the *mean* of the values:

$$\mu ::= \frac{x_1+x_2+\cdots+x_n}{n}$$

Theorem 1.6.1. *The standard deviation of a sequence of values $x_1, \ldots, x_n$ is zero iff all the values are equal to the mean.*

For example, the standard deviation of test scores is zero if and only if everyone scored exactly the class average.

Proof. We construct a chain of "iff" implications, starting with the statement that the standard deviation (1.3) is zero:

$$\sqrt{\frac{(x_1-\mu)^2+(x_2-\mu)^2+\cdots+(x_n-\mu)^2}{n}} = 0. \tag{1.4}$$

break it down

Now since zero is the only number whose square root is zero, equation (1.4) holds iff

$$(x_1-\mu)^2+(x_2-\mu)^2+\cdots+(x_n-\mu)^2 = 0. \tag{1.5}$$

Now squares of real numbers are always nonnegative, so every term on the left hand side of equation (1.5) is nonnegative. This means that (1.5) holds iff

$$\text{Every term on the left hand side of (1.5) is zero.} \tag{1.6}$$

But a term $(x_i - \mu)^2$ is zero iff $x_i = \mu$, so (1.6) is true iff

Every x_i equals the mean.

■

Starting to get it

1.7 Proof by Cases

Breaking a complicated proof into cases and proving each case separately is a useful, common proof strategy. Here's an amusing example.

Let's agree that given any two people, either they have met or not. If every pair of people in a group has met, we'll call the group a *club*. If every pair of people in a group has not met, we'll call it a group of *strangers*.

Theorem. *Every collection of 6 people includes a club of 3 people or a group of 3 strangers.*

Proof. The proof is by case analysis[4]. Let x denote one of the six people. There are two cases:

must make sure have all cases

1. Among 5 other people besides x, at least 3 have met x.
2. Among the 5 other people, at least 3 have not met x.

Now we have to be sure that at least one of these two cases must hold,[5] but that's easy: we've split the 5 people into two groups, those who have shaken hands with x and those who have not, so one the groups must have at least half the people.

Case 1: Suppose that at least 3 people did meet x.

This case splits into two subcases:

> **Case 1.1:** No pair among those people met each other. Then these people are a group of at least 3 strangers. So the Theorem holds in this subcase.
>
> **Case 1.2:** Some pair among those people have met each other. Then that pair, together with x, form a club of 3 people. So the Theorem holds in this subcase.

This implies that the Theorem holds in Case 1.

Case 2: Suppose that at least 3 people did not meet x.

This case also splits into two subcases:

[4] Describing your approach at the outset helps orient the reader.

[5] Part of a case analysis argument is showing that you've covered all the cases. Often this is obvious, because the two cases are of the form "P" and "not P". However, the situation above is not stated quite so simply.

Case 2.1: Every pair among those people met each other. Then these people are a club of at least 3 people. So the Theorem holds in this subcase.

Case 2.2: Some pair among those people have not met each other. Then that pair, together with x, form a group of at least 3 strangers. So the Theorem holds in this subcase.

This implies that the Theorem also holds in Case 2, and therefore holds in all cases. ■

1.8 Proof by Contradiction

Starting to get as well

In a *proof by contradiction* or *indirect proof*, you show that if a proposition were false, then some false fact would be true. Since a false fact can't be true, the proposition had better not be false. That is, the proposition really must be true.

Proof by contradiction is *always* a viable approach. However, as the name suggests, indirect proofs can be a little convoluted. So direct proofs are generally preferable as a matter of clarity.

Method: In order to prove a proposition P by contradiction:

1. Write, "We use proof by contradiction."
2. Write, "Suppose P is false."
3. Deduce something known to be false (a logical contradiction).
4. Write, "This is a contradiction. Therefore, P must be true."

Example

Remember that a number is *rational* if it is equal to a ratio of integers. For example, $3.5 = 7/2$ and $0.1111\cdots = 1/9$ are rational numbers. On the other hand, we'll prove by contradiction that $\sqrt{2}$ is irrational.

Theorem 1.8.1. *$\sqrt{2}$ is irrational.*

Proof. We use proof by contradiction. Suppose the claim is false; that is, $\sqrt{2}$ is rational. Then we can write $\sqrt{2}$ as a fraction n/d in *lowest terms*.

Was in class issue

Squaring both sides gives $2 = n^2/d^2$ and so $2d^2 = n^2$. This implies that n is a multiple of 2. Therefore n^2 must be a multiple of 4. But since $2d^2 = n^2$, we know

$2d^2$ is a multiple of 4 and so d^2 is a multiple of 2. This implies that d is a multiple of 2.

So the numerator and denominator have 2 as a common factor, which contradicts the fact that n/d is in lowest terms. So $\sqrt{2}$ must be irrational. ■

1.9 *Good* Proofs in Practice

One purpose of a proof is to establish the truth of an assertion with absolute certainty. Mechanically checkable proofs of enormous length or complexity can accomplish this. But humanly intelligible proofs are the only ones that help someone understand the subject. Mathematicians generally agree that important mathematical results can't be fully understood until their proofs are understood. That is why proofs are an important part of the curriculum.

And learning to write

To be understandable and helpful, more is required of a proof than just logical correctness: a good proof must also be clear. Correctness and clarity usually go together; a well-written proof is more likely to be a correct proof, since mistakes are harder to hide.

was not in other book

In practice, the notion of proof is a moving target. Proofs in a professional research journal are generally unintelligible to all but a few experts who know all the terminology and prior results used in the proof. Conversely, proofs in the first weeks of a beginning course like 6.042 would be regarded as tediously long-winded by a professional mathematician. In fact, what we accept as a good proof later in the term will be different from what we consider good proofs in the first couple of weeks of 6.042. But even so, we can offer some general tips on writing good proofs:

State your game plan. A good proof begins by explaining the general line of reasoning, for example, "We use case analysis" or "We argue by contradiction."

Keep a linear flow. Sometimes proofs are written like mathematical mosaics, with juicy tidbits of independent reasoning sprinkled throughout. This is not good. The steps of an argument should follow one another in an intelligible order.

A proof is an essay, not a calculation. Many students initially write proofs the way they compute integrals. The result is a long sequence of expressions without explanation, making it very hard to follow. This is bad. A good proof usually looks like an essay with some equations thrown in. Use complete sentences.

Avoid excessive symbolism. Your reader is probably good at understanding words, but much less skilled at reading arcane mathematical symbols. So use words where you reasonably can.

Revise and simplify. Your readers will be grateful.

Introduce notation thoughtfully. Sometimes an argument can be greatly simplified by introducing a variable, devising a special notation, or defining a new term. But do this sparingly since you're requiring the reader to remember all that new stuff. And remember to actually *define* the meanings of new variables, terms, or notations; don't just start using them!

Structure long proofs. Long programs are usually broken into a hierarchy of smaller procedures. Long proofs are much the same. Facts needed in your proof that are easily stated, but not readily proved are best pulled out and proved in preliminary lemmas. Also, if you are repeating essentially the same argument over and over, try to capture that argument in a general lemma, which you can cite repeatedly instead.

Be wary of the "obvious". When familiar or truly obvious facts are needed in a proof, it's OK to label them as such and to not prove them. But remember that what's obvious to you, may not be—and typically is not—obvious to your reader.

Most especially, don't use phrases like "clearly" or "obviously" in an attempt to bully the reader into accepting something you're having trouble proving. Also, go on the alert whenever you see one of these phrases in someone else's proof.

Finish. At some point in a proof, you'll have established all the essential facts you need. Resist the temptation to quit and leave the reader to draw the "obvious" conclusion. Instead, tie everything together yourself and explain why the original claim follows.

Creating a good proof is a lot like creating a beautiful work of art. In fact, mathematicians often refer to really good proofs as being "elegant" or "beautiful." It takes a practice and experience to write proofs that merit such praises, but to get you started in the right direction, we will provide templates for the most useful proof techniques.

Throughout the text there are also examples of *bogus proofs*—arguments that look like proofs but aren't. Sometimes a bogus proof can reach false conclusions because of missteps or mistaken assumptions. More subtle bogus proofs reach correct conclusions, but in improper ways, for example by circular reasoning, by learning a lot here

leaping to unjustified conclusions, or by saying that the hard part of "the proof is left to the reader." Learning to spot the flaws in improper proofs will hone your skills at seeing how each proof step follows logically from prior steps. It will also enable you to spot flaws in your own proofs.

The analogy between good proofs and good programs extends beyond structure. The same rigorous thinking needed for proofs is essential in the design of critical computer systems. When algorithms and protocols only "mostly work" due to reliance on hand-waving arguments, the results can range from problematic to catastrophic. An early example was the Therac 25, a machine that provided radiation therapy to cancer victims, but occasionally killed them with massive overdoses due to a software race condition. A more recent (August 2004) example involved a single faulty command to a computer system used by United and American Airlines that grounded the entire fleet of both companies—and all their passengers!

← UI?

It is a certainty that we'll all one day be at the mercy of critical computer systems designed by you and your classmates. So we really hope that you'll develop the ability to formulate rock-solid logical arguments that a system actually does what you think it does!

Problems for Section 1.5

Homework Problems

Problem 1.1.
Show that $\log_7 n$ is either an integer or irrational, where n is a positive integer. Use whatever familiar facts about integers and primes you need, but explicitly state such facts.

Problems for Section 1.7

Class Problems

Problem 1.2.
If we raise an irrational number to an irrational power, can the result be rational? Show that it can by considering $\sqrt{2}^{\sqrt{2}}$ and arguing by cases.

Homework Problems

Problem 1.3.
For $n = 40$, the value of polynomial $p(n) ::= n^2 + n + 41$ is not prime, as noted in Section 1.1. But we could have predicted based on general principles that no nonconstant polynomial can generate only prime numbers.

In particular, let $q(n)$ be a polynomial with integer coefficients, and let $c ::= q(0)$

be the constant term of q.

(a) Verify that $q(cm)$ is a multiple of c for all $m \in \mathbb{Z}$.

(b) Show that if q is nonconstant and $c > 1$, then there are infinitely many $n \in \mathbb{N}$ such that $q(n)$ is not prime.

Hint: You may assume the familiar fact that the magnitude of any nonconstant polynomial, $q(n)$, grows unboundedly as n grows.

(c) Conclude immediately that for every nonconstant polynomial, q, there must be an $n \in \mathbb{N}$ such that $q(n)$ is not prime.

Problems for Section 1.8

Class Problems

Problem 1.4.

Generalize the proof of Theorem 1.8.1 that $\sqrt{2}$ is irrational. For example, how about $\sqrt[3]{2}$?

Problem 1.5.

Here is a different proof that $\sqrt{2}$ is irrational, taken from the American Mathematical Monthly, v.116, #1, Jan. 2009, p.69:

Proof. Suppose for the sake of contradiction that $\sqrt{2}$ is rational, and choose the least integer, $q > 0$, such that $\left(\sqrt{2}-1\right) q$ is a nonnegative integer. Let $q' ::= \left(\sqrt{2}-1\right) q$. Clearly $0 < q' < q$. But an easy computation shows that $\left(\sqrt{2}-1\right) q'$ is a nonnegative integer, contradicting the minimality of q. ■

(a) This proof was written for an audience of college teachers, and at this point it is a little more concise than desirable. Write out a more complete version which includes an explanation of each step.

(b) Now that you have justified the steps in this proof, do you have a preference for one of these proofs over the other? Why? Discuss these questions with your teammates for a few minutes and summarize your team's answers on your whiteboard.

Problem 1.6.

Here is a generalization of Problem 1.4 that you may not have thought of:

Lemma 1.9.1. *Let the coefficients of the polynomial* $a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{m-1} + x^m$ *be integers. Then any real root of the polynomial is either integral or irrational.*

(a) Explain why Lemma 1.9.1 immediately implies that $\sqrt[m]{k}$ is irrational whenever k is not an mth power of some integer.

(b) Collaborate with your tablemates to write a clear, textbook quality proof of Lemma 1.9.1 on your whiteboard. (Besides clarity and correctness, textbook quality requires good English with proper punctuation. When a real textbook writer does this, it usually takes multiple revisions; if you're satisfied with your first draft, you're probably misjudging.) You may find it helpful to appeal to the following:

Lemma 1.9.2. *If a prime, p, is a factor of some power of an integer, then it is a factor of that integer.*

You may assume Lemma 1.9.2 without writing down its proof, but see if you can explain why it is true.

Homework Problems

Problem 1.7.

The fact that that there are irrational numbers a, b such that a^b is rational was proved in Problem 1.2. Unfortunately, that proof was *nonconstructive*: it didn't reveal a specific pair, a, b, with this property. But in fact, it's easy to do this: let $a ::= \sqrt{2}$ and $b ::= 2\log_2 3$.

We know $\sqrt{2}$ is irrational, and obviously $a^b = 3$. Finish the proof that this a, b pair works, by showing that $2\log_2 3$ is irrational.

2 The Well Ordering Principle

Every *nonempty* set of *nonnegative integers* has a *smallest* element.

This statement is known as The *Well Ordering Principle*. Do you believe it? Seems sort of obvious, right? But notice how tight it is: it requires a *nonempty* set —it's false for the empty set which has *no* smallest element because it has no elements at all! And it requires a set of *nonnegative* integers —it's false for the set of *negative* integers and also false for some sets of nonnegative *rationals* —for example, the set of positive rationals. So, the Well Ordering Principle captures something special about the nonnegative integers.

? divisable by integers -why?

2.1 Well Ordering Proofs

While the Well Ordering Principle may seem obvious, it's hard to see offhand why it is useful. But in fact, it provides one of the most important proof rules in discrete mathematics.

Oh the thing I had trouble with

In fact, looking back, we took the Well Ordering Principle for granted in proving that $\sqrt{2}$ is irrational. That proof assumed that for any positive integers m and n, the fraction m/n can be written in *lowest terms*, that is, in the form m'/n' where m' and n' are positive integers with no common factors. How do we know this is always possible?

? what it means to be lowest

Suppose to the contrary that there were $m, n \in \mathbb{Z}^+$ such that the fraction m/n cannot be written in lowest terms. Now let C be the set of positive integers that are numerators of such fractions. Then $m \in C$, so C is nonempty. Therefore, by Well Ordering, there must be a smallest integer, $m_0 \in C$. So by definition of C, there is an integer $n_0 > 0$ such that

$$\text{the fraction } \frac{m_0}{n_0} \text{ cannot be written in lowest terms.}$$

This means that m_0 and n_0 must have a common factor, $p > 1$. But

$$\frac{m_0/p}{n_0/p} = \frac{m_0}{n_0},$$

so any way of expressing the left hand fraction in lowest terms would also work for

m_0/n_0, which implies

$$\text{the fraction } \frac{m_0/p}{n_0/p} \text{ cannot be in written in lowest terms either.}$$

So by definition of C, the numerator, m_0/p, is in C. But $m_0/p < m_0$, which contradicts the fact that m_0 is the smallest element of C.

Since the assumption that C is nonempty leads to a contradiction, it follows that C must be empty. That is, that there are no numerators of fractions that can't be written in lowest terms, and hence there are no such fractions at all.

We've been using the Well Ordering Principle on the sly from early on!

2.2 Template for Well Ordering Proofs

More generally, there is a standard way to use Well Ordering to prove that some property, $P(n)$ holds for every nonnegative integer, n. Here is a standard way to organize such a well ordering proof:

> To prove that "$P(n)$ is true for all $n \in \mathbb{N}$" using the Well Ordering Principle:
>
> - Define the set, C, of *counterexamples* to P being true. Namely, define[1]
>
> $$C ::= \{n \in \mathbb{N} \mid P(n) \text{ is false}\}.$$
>
> - Assume for proof by contradiction that C is nonempty.
> - By the Well Ordering Principle, there will be a smallest element, n, in C.
> - Reach a contradiction (somehow) —often by showing how to use n to find another member of C that is smaller than n. (This is the open-ended part of the proof task.)
> - Conclude that C must be empty, that is, no counterexamples exist. QED

2.3 Summing the Integers

Let's use this this template to prove

I can only read ~15 pg at once

Theorem.

$$1 + 2 + 3 + \cdots + n = n(n + 1)/2 \tag{2.1}$$

for all nonnegative integers, n.

can be 0

First, we better address of a couple of ambiguous special cases before they trip us up:

- If $n = 1$, then there is only one term in the summation, and so $1 + 2 + 3 + \cdots + n$ is just the term 1. Don't be misled by the appearance of 2 and 3 and the suggestion that 1 and n are distinct terms!

- If $n \leq 0$, then there are no terms at all in the summation. By convention, the sum in this case is 0.

So while the dots notation is convenient, you have to watch out for these special cases where the notation is misleading! (In fact, whenever you see the dots, you should be on the lookout to be sure you understand the pattern, watching out for the beginning and the end.)

We could have eliminated the need for guessing by rewriting the left side of (2.1) with *summation notation*:

Clearer

$$\boxed{\sum_{i=1}^{n} i} \quad \text{or} \quad \sum_{1 \leq i \leq n} i.$$

Both of these expressions denote the sum of all values taken by the expression to the right of the sigma as the variable, i, ranges from 1 to n. Both expressions make it clear what (2.1) means when $n = 1$. The second expression makes it clear that when $n = 0$, there are no terms in the sum, though you still have to know the convention that a sum of no numbers equals 0 (the *product* of no numbers is 1, by the way).

OK, back to the proof:

Proof. By contradiction. Assume that the theorem is *false*. Then, some nonnegative integers serve as *counterexamples* to it. Let's collect them in a set:

? but don't have to say what they are —or will later show that there are none

$$C ::= \{n \in \mathbb{N} \mid 1 + 2 + 3 + \cdots + n \neq \frac{n(n + 1)}{2}\}.$$

By our assumption that the theorem admits counterexamples, C is a nonempty set of nonnegative integers. So, by the Well Ordering Principle, C has a minimum element, call it c. That is, c is the *smallest counterexample* to the theorem.

Since c is the smallest counterexample, we know that (2.1) is false for $n = c$ but true for all nonnegative integers $n < c$. But (2.1) is true for $n = 0$, so $c > 0$. This

↑ how do we know ↑ oh here update with what C can actually be

← ugh oh I kinda see

means $c - 1$ is a nonnegative integer, and since it is less than c, equation (2.1) is true for $c - 1$. That is,

weird

$$1 + 2 + 3 + \cdots + (c - 1) = \frac{(c - 1)c}{2}.$$

But then, adding c to both sides we get what does this do?

$$1 + 2 + 3 + \cdots + (c - 1) + c = \frac{(c - 1)c}{2} + c = \frac{c^2 - c + 2c}{2} = \frac{c(c + 1)}{2},$$

which means that (2.1) does hold for c, after all! This is a contradiction, and we are done. ■

2.4 Factoring into Primes

We've previously taken for granted the *Prime Factorization Theorem* that every integer greater than one has a unique[2] expression as a product of prime numbers. This is another of those familiar mathematical facts which are not really obvious. We'll prove the uniqueness of prime factorization in a later chapter, but well ordering gives an easy proof that every integer greater than one can be expressed as *some* product of primes.

Theorem 2.4.1. *Every natural number can be factored as a product of primes.*

Proof. The proof is by Well Ordering.

say something false 1st :)

- this confuses me at first (thought it was actually true!)

Let C be the set of all integers greater than one that cannot be factored as a product of primes. We assume C is not empty and derive a contradiction.

If C is not empty, there is a least element, $n \in C$, by Well Ordering. The n can't be prime, because a prime by itself is considered a (length one) product of primes and no such products are in C.

So n must be a product of two integers a and b where $1 < a, b < n$. Since a and b are smaller than the smallest element in C, we know that $a, b \notin C$. In other words, a can be written as a product of primes $p_1 p_2 \cdots p_k$ and b as a product of primes $q_1 \cdots q_l$. Therefore, $n = p_1 \cdots p_k q_1 \cdots q_l$ can be written as a product of primes, contradicting the claim that $n \in C$. Our assumption that $C \neq \emptyset$ must therefore be false. ■

does not seem very conclusive to me

[2] ...unique up to the order in which the prime factors appear

Problems for Section 2.2

Practice Problems

Problem 2.1.
For practice using the Well Ordering Principle, fill in the template of an easy to prove fact: every amount of postage that can be assembled using only 10 cent and 15 cent stamps is divisible by 5.

In particular, Let $S(n)$ mean that exactly n cents postage can be assembled using only 10 and 15 cent stamps. Then the proof shows that

$$S(n) \text{ IMPLIES } 5 \mid n, \quad \text{for all nonnegative integers } n. \tag{*}$$

Fill in the missing portions (indicated by "...") of the following proof of (*).

> Let C be the set of *counterexamples* to (*), namely
>
> $$C ::= \{n \mid \ldots\}$$
>
> Assume for the purpose of obtaining a contradiction that C is nonempty. Then by the WOP, there is a smallest number, $m \in C$. This m must be positive because ....
>
> But if $S(m)$ holds and m is positive, then $S(m-10)$ or $S(m-15)$ must hold, because ....
>
> So suppose $S(m-10)$ holds. Then $5 \mid (m-10)$, because...
>
> But if $5 \mid (m-10)$, then obviously $5 \mid m$, contradicting the fact that m is a counterexample.
>
> Next, if $S(m-15)$ holds, we arrive at a contradiction in the same way.
>
> Since we get a contradiction in both cases, we conclude that...
>
> which proves that (*) holds.

Class Problems

Problem 2.2.
The proof below uses the Well Ordering Principle to prove that every amount of postage that can be assembled using only 6 cent and 15 cent stamps, is divisible by 3. Let the notation "$j \mid k$" indicate that integer j is a divisor of integer k, and let $S(n)$ mean that exactly n cents postage can be assembled using only 6 and 15 cent stamps. Then the proof shows that

$$S(n) \text{ IMPLIES } 3 \mid n, \quad \text{for all nonnegative integers } n. \tag{*}$$

Fill in the missing portions (indicated by "...") of the following proof of (*).

Let C be the set of *counterexamples* to (*), namely[3]

$$C ::= \{n \mid \ldots\}$$

Assume for the purpose of obtaining a contradiction that C is nonempty. Then by the WOP, there is a smallest number, $m \in C$. This m must be positive because....

But if $S(m)$ holds and m is positive, then $S(m-6)$ or $S(m-15)$ must hold, because....

So suppose $S(m-6)$ holds. Then $3 \mid (m-6)$, because...

But if $3 \mid (m-6)$, then obviously $3 \mid m$, contradicting the fact that m is a counterexample.

Next, if $S(m-15)$ holds, we arrive at a contradiction in the same way. Since we get a contradiction in both cases, we conclude that... which proves that (*) holds.

Problem 2.3.

Euler's Conjecture in 1769 was that there are no positive integer solutions to the equation

$$a^4 + b^4 + c^4 = d^4.$$

Integer values for a, b, c, d that do satisfy this equation, were first discovered in 1986. So Euler guessed wrong, but it took more two hundred years to prove it.

Now let's consider Lehman's equation, similar to Euler's but with some coefficients:

$$8a^4 + 4b^4 + 2c^4 = d^4 \tag{2.2}$$

Prove that Lehman's equation (2.2) really does not have any positive integer solutions.

Hint: Consider the minimum value of a among all possible solutions to (2.2).

Homework Problems

Problem 2.4.

Use the Well Ordering Principle to prove that any integer greater than or equal to 8 can be represented as the sum of integer multiples of 3 and 5.

[3] The notation "$\{n \mid \ldots\}$" means "the set of elements, n, such that ...."

Exam Problems

Problem 2.5.
The proof below uses the Well Ordering Principle to prove that every amount of postage that can be paid exactly, using only 10 cent and 15 cent stamps, is divisible by 5. Let $S(n)$ mean that exactly n cents postage can be paid using only 10 and 15 cent stamps. Then the proof shows that

$$S(n) \text{ IMPLIES } 5 \mid n, \quad \text{for all nonnegative integers } n. \tag{*}$$

Fill in the missing portions (indicated by "...") of the following proof of (*).

> Let C be the set of *counterexamples* to (*), namely
>
> $$C ::= \{n \mid \ldots 4in\}$$
>
> Assume for the purpose of obtaining a contradiction that C is nonempty. Then by the WOP, there is a smallest number, $m \in C$. This m must be positive because...
>
> 6in
>
> But if $S(m)$ holds and m is positive, then $S(m-10)$ or $S(m-15)$ must hold, because....
>
> 6in
>
> So suppose $S(m-10)$ holds. Then $5 \mid (m-10)$, because...
>
> 6in
>
> But if $5 \mid (m-10)$, then obviously $5 \mid m$, contradicting the fact that m is a counterexample.
>
> Next suppose $S(m-15)$ holds. Then the proof for $m-10$ carries over directly for $m-15$ to yield a contradiction in this case as well. Since we get a contradiction in both cases, we conclude that...
>
> 6in
>
> which proves that (*) holds.

Problems for Section 2.3

Practice Problems

Problem 2.6.
The Fibonacci numbers

$$0, 1, 1, 2, 3, 5, 8, 13, \ldots$$

are defined as follows. Let $F(n)$ be the nth Fibonacci number. Then

$$F(0) ::= 0, \tag{2.3}$$

$$F(1) ::= 1, \tag{2.4}$$

$$F(n) ::= F(n-1) + F(n-2) \qquad \text{for } n \geq 2. \tag{2.5}$$

Indicate exactly which sentence(s) in the following bogus proof contain logical errors? Explain.

False Claim. *Every Fibonacci number is even.*

Bogus proof. Let all the variables n, m, k mentioned below be nonnegative integer valued.

1. The proof is by the WOP.
2. Let Even(n) mean that $F(n)$ is even.
3. Let C be the set of counterexamples to the assertion that Even(n) holds for all $n \in \mathbb{N}$, namely,

   $$C ::= \{n \in \mathbb{N} \mid \text{NOT}(\text{Even}(n))\}.$$

4. We prove by contradiction that C is empty. So assume that C is not empty.
5. By WOP, there is a least nonnegative integer, $m \in C$,
6. Then $m > 0$, since $F(0) = 0$ is an even number.
7. Since m is the minimum counterexample, $F(k)$ is even for all $k < m$.
8. In particular, $F(m-1)$ and $F(m-2)$ are both even.
9. But by the defining equation (2.5), $F(m)$ equals the sum $F(m-1)+F(m-2)$ of two even numbers, and so it is also even.
10. That is, Even(m) is true.
11. This contradicts the condition in the definition of m that NOT(Even(m)) holds.
12. This contradition implies that C must be empty. Hence, $F(n)$ is even for all $n \in \mathbb{N}$.

■

Problem 2.7.
In Chapter 2, the Well Ordering was used to show that all positive rational numbers can be written in "lowest terms," that is, as a ratio of positive integers with no common factor prime factor. Below is a different proof which also arrives at this correct conclusion, but this proof is bogus. Identify every step at which the proof makes an unjustified inference.

Bogus proof. Suppose to the contrary that there was positive rational, q, such that q cannot be written in lowest terms. Now let C be the set of such rational numbers that cannot be written in lowest terms. Then $q \in C$, so C is nonempty. So there must be a smallest rational, $q_0 \in C$. So since $q_0/2 < q_0$, it must be possible to express $q_0/2$ in lowest terms, namely,

$$\frac{q_0}{2} = \frac{m}{n} \tag{2.6}$$

for positive integers m, n with no common prime factor. Now we consider two cases:

Case 1: [n is odd]. Then $2m$ and n also have no common prime factor, and therefore

$$q_0 = 2 \cdot \left(\frac{m}{n}\right) = \frac{2m}{n}$$

expresses q_0 in lowest terms, a contradiction.

Case 2: [n is even]. Any common prime factor of m and $n/2$ would also be a common prime factor of m and n. Therefore m and $n/2$ have no common prime factor, and so

$$q_0 = \frac{m}{n/2}$$

expresses q_0 in lowest terms, a contradiction.

Since the assumption that C is nonempty leads to a contradiction, it follows that C is empty—that is, there are no counterexamples. ■

Class Problems

Problem 2.8.
Use the Well Ordering Principle to prove that

$$\sum_{k=0}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}. \tag{2.7}$$

for all nonnegative integers, n.

3 Logical Formulas

It is amazing that people manage to cope with all the ambiguities in the English language. Here are some sentences that illustrate the issue:

From old book part 1

- "You may have cake, or you may have ice cream."
- "If pigs can fly, then you can understand the Chebyshev bound."
- "If you can solve any problem we come up with, then you get an A for the course."
- "Every American has a dream."

What *precisely* do these sentences mean? Can you have both cake and ice cream or must you choose just one dessert? Pigs can't fly, so does the second sentence say anything about your understanding the Chebyshev bound? If you can solve some problems we come up with, can you get an A for the course? And if you can't solve a single one of the problems, does it mean you can't get an A? Finally, does the last sentence imply that all Americans have the same dream—say of owning a house—or might different Americans may have different dreams—say, Eric dreams of designing a "killer" software application, Tom of being a tennis champion, Albert of being able to sing?

Some uncertainty is tolerable in normal conversation. But when we need to formulate ideas precisely—as in mathematics and programming—the ambiguities inherent in everyday language can be a real problem. We can't hope to make an exact argument if we're not sure exactly what the statements mean. So before we start into mathematics, we need to investigate the problem of how to talk about mathematics.

To get around the ambiguity of English, mathematicians have devised a special language for talking about logical relationships. This language mostly uses ordinary English words and phrases such as "or", "implies", and "for all". But mathematicians give these words precise and unambiguous definitions.

Surprisingly, in the midst of learning the language of logic, we'll come across the most important open problem in computer science—a problem whose solution could change the world.

3.1 Propositions from Propositions

In English, we can modify, combine, and relate propositions with words such as "not", "and", "or", "implies", and "if-then". For example, we can combine three propositions into one like this:

> **If** all humans are mortal **and** all Greeks are human, **then** all Greeks are mortal.

For the next while, we won't be much concerned with the internals of propositions—whether they involve mathematics or Greek mortality—but rather with how propositions are combined and related. So we'll frequently use variables such as P and Q in place of specific propositions such as "All humans are mortal" and "$2 + 3 = 5$". The understanding is that these *propositional variables*, like propositions, can take on only the values **T** (true) and **F** (false). Propositional variables are also called *Boolean variables* after their inventor, the nineteenth century mathematician George—you guessed it—Boole.

3.1.1 NOT, AND, and OR

Mathematicians use the words NOT, AND, and OR for operations that change or combine propositions. The precise mathematical meaning of these special words can be specified by *truth tables*. For example, if P is a proposition, then so is "NOT(P)," and the truth value of the proposition "NOT(P)" is determined by the truth value of P according to the following truth table:

P	NOT(P)
T	**F**
F	**T**

The first row of the table indicates that when proposition P is true, the proposition "NOT(P)" is false. The second line indicates that when P is false, "NOT(P)" is true. This is probably what you would expect.

In general, a truth table indicates the true/false value of a proposition for each possible set of truth values for the variables. For example, the truth table for the proposition "P AND Q" has four lines, since there are four settings of truth values for the two variables:

P	Q	P AND Q
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**F**

I ~~am~~ am thinking of it in terms of binary bits

According to this table, the proposition “P AND Q” is true only when P and Q are both true. This is probably the way you ordinarily think about the word “and.”

There is a subtlety in the truth table for “P OR Q”:

P	Q	P OR Q
T	**T**	**T**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

The third row of this table says that “P OR Q” is true even if *both* P and Q are true. This isn’t always the intended meaning of “or” in everyday speech, but this is the standard definition in mathematical writing. So if a mathematician says, “You may have cake, or you may have ice cream,” he means that you *could* have both.

If you want to exclude the possibility of both having and eating, you should combine them with the *exclusive-or* operation, XOR:

P	Q	P XOR Q
T	**T**	**F**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

3.1.2 IMPLIES

The combining operation with the least intuitive technical meaning is “implies.” Here is its truth table, with the lines labeled so we can refer to them later.

P	Q	P IMPLIES Q	
T	**T**	**T**	(tt)
T	**F**	**F**	(tf)
F	**T**	**T**	(ft)
F	**F**	**T**	(ff)

Let’s experiment with this definition. For example, is the following proposition true or false?

“If Goldbach’s Conjecture is true, then $x^2 \geq 0$ for every real number x.”

Now, we already mentioned that no one knows whether Goldbach’s Conjecture, Proposition 1.1.7, is true or false. But that doesn’t prevent you from answering the question! This proposition has the form P IMPLIES Q where the *hypothesis*, P, is “Goldbach’s Conjecture is true” and the *conclusion*, Q, is “$x^2 \geq 0$ for every

If conclusion true, its true

real number x”. Since the conclusion is definitely true, we’re on either line (tt) or line (ft) of the truth table. Either way, the proposition as a whole is *true*!

One of our original examples demonstrates an even stranger side of implications.

> “If pigs fly, then you can understand the Chebyshev bound.”

Don’t take this as an insult; we just need to figure out whether this proposition is true or false. Curiously, the answer has *nothing* to do with whether or not you can understand the Chebyshev bound. Pigs do not fly, so we’re on either line (ft) or line (ff) of the truth table. In both cases, the proposition is *true*! ~~Math~~ Valid

key

In contrast, here’s an example of a false implication: → not makes sense

> “If the moon shines white, then the moon is made of white cheddar.”

Yes, the moon shines white. But, no, the moon is not made of white cheddar cheese. So we’re on line (tf) of the truth table, and the proposition is false.

The truth table for implications can be summarized in words as follows:

> An implication is true exactly when the if-part is false or the then-part is true.

Valid Math or

This sentence is worth remembering; a large fraction of all mathematical statements are of the if-then form!

3.1.3 If and Only If

Mathematicians commonly join propositions in one additional way that doesn’t arise in ordinary speech. The proposition “P if and only if Q” asserts that P and Q have the same truth value, that is, either both are true or both are false.

P	Q	P IFF Q
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**T**

For example, the following if-and-only-if statement is true for every real number x:

$$x^2 - 4 \geq 0 \text{ IFF } |x| \geq 2.$$

For some values of x, *both* inequalities are true. For other values of x, *neither* inequality is true . In every case, however, the IFF proposition as a whole is true.

Valid

3.2 Propositional Logic in Computer Programs

Propositions and logical connectives arise all the time in computer programs. For example, consider the following snippet, which could be either C, C++, or Java:

```
if ( x > 0 || (x <= 0 && y > 100) )
   ⋮
(further instructions)
```

Java uses The symbol `||` for "OR," and the symbol `&&` for "AND." The *further instructions* are carried out only if the proposition following the word `if` is true. On closer inspection, this big expression is built from two simpler propositions. Let A be the proposition that `x > 0`, and let B be the proposition that `y > 100`. Then we can rewrite the condition as

$$A \text{ OR } (\text{NOT}(A) \text{ AND } B). \tag{3.1}$$

3.2.1 Truth Table Calculation

A truth table calculation reveals that the more complicated expression 3.1 always has the same truth value as

$$A \text{ OR } B. \tag{3.2}$$

Namely, we begin with a table with just the truth values of A and B:

A	B	A	OR	(NOT(A)	AND	B)	A OR B
T	**T**						
T	**F**						
F	**T**						
F	**F**						

These values are enough to fill in two more columns:

A	B	A	OR	(NOT(A)	AND	B)	A OR B
T	**T**			F			**T**
T	**F**			F			**T**
F	**T**			T			**T**
F	**F**			T			**F**

? wrong format

Now we have the values needed to fill in the AND column:

More detailed than other book

A	B	A	OR	(NOT(A)	AND	B)	A OR B
T	**T**			**F**	**F**		**T**
T	**F**			**F**	**F**		**T**
F	**T**			**T**	**T**		**T**
F	**F**			**T**	**F**		**F**

and this provides the values needed to fill in the remaining column for the first OR:

A	B	A	OR	(NOT(A)	AND	B)	A OR B
T	**T**		**T**	**F**	**F**		**T**
T	**F**		**T**	**F**	**F**		**T**
F	**T**		**T**	**T**	**T**		**T**
F	**F**		**F**	**T**	**F**		**F**

Matches

Expression whose truth values always match are called *equivalent*. Since the (highlighted) columns of truth values of the two expressions are the same, they are equivalent. So we can simplify the code snippet without changing the program's behavior by replacing the complicated expression with an equivalent simpler one:

```
if ( x > 0 || y > 100 )
        ⋮
```

(further instructions)

The equivalence of (3.1) and (3.2) can also be confirmed reasoning by cases:

A is **T**. An expression of the form (**T** OR anything) is equivalent to **T**. Since A is **T** both (3.1) and (3.2) in this case are of this form, so they have the same truth value, namely, **T**.

A is **F**. Then an expression of the form (A OR *anything*) will have same truth value as *anything*. So (3.2) has the same truth value as B.

Now any expression of the form (**T** AND *anything*) is equivalent to *anything*, as is any expression of the form **F** OR *anything*. So in this case A OR (NOT(A) AND B) is equivalent to (NOT(A) AND B), which in turn is equivalent to B.

Therefore both (3.1) and (3.2) will have the same truth value in this case, namely, the value of B.

Simplifying logical expressions has real practical importance in computer science. Expression simplification in programs like the one above can make a program but obscures your ~~[illegible]~~ intensions

easier to read and understand, and can also make it faster since fewer operations are needed. In hardware, simplifying expressions can decrease the number of logic gates on a chip. That's because digital circuits can be described by logical formulas (see Problems 3.4 and 3.5), and minimizing the logical formulas corresponds to reducing the number of gates in the circuit. The payoff of gate minimization is potentially enormous: a chip with fewer gates is smaller, consumes less power, has a lower defect rate, and is cheaper to manufacture.

3.2.2 Cryptic Notation

Java uses symbols like "&&" and "||" in place of AND and OR. Circuit designers use "·" and "+," and actually refer to AND as a product and OR as a sum. Mathematicians use still other symbols given in the table below.

English	Symbolic Notation
NOT(P)	$\neg P$ (alternatively, $\overline{P}$)
P AND Q	$P \wedge Q$
P OR Q	$P \vee Q$
P IMPLIES Q	$P \longrightarrow Q$
if P then Q	$P \longrightarrow Q$
P IFF Q	$P \longleftrightarrow Q$
P XOR Q	$P \oplus Q$

$\cap$ AND
$\cup$ ~~[illegible]~~ Or

For example, using this notation, "If P AND NOT(Q), then R" would be written:

$$(P \wedge \overline{Q}) \longrightarrow R.$$

The mathematical notation is concise but cryptic. Words such as "AND" and "OR" are easier to remember and won't get confused with operations on numbers. We will often use $\overline{P}$ as an abbreviation ofr NOT(P), but aside from that, we mostly stick to the words—except when formulas would otherwise run off the page.

3.3 Equivalence and Validity

3.3.1 Implications and Contrapositives

Do these two sentences say the same thing?

If I am hungry, then I am grumpy.
If I am not grumpy, then I am not hungry.

Yes
Contrapositive

We can settle the issue by recasting both sentences in terms of propositional logic. Let P be the proposition "I am hungry", and Q be "I am grumpy". The first sentence says "P IMPLIES Q" and the second says "NOT(Q) IMPLIES NOT(P)". Once more, we can compare these two statements in a truth table:

P	Q	(P IMPLIES Q)	(NOT(Q)	IMPLIES	NOT(P))
T	**T**	**T**	**F**	**T**	**F**
T	**F**	**F**	**T**	**F**	**F**
F	**T**	**T**	**F**	**T**	**T**
F	**F**	**T**	**T**	**T**	**T**

if true, must be true
if false does not matter

~~Ale~~ look back

Sure enough, the highlighted columns showing the truth values of these two statements are the same. A statement of the form "(NOT Q) IMPLIES (NOT P)" is called the *contrapositive* of the implication "P IMPLIES Q." The truth table shows that an implication and its contrapositive are equivalent—they are just different ways of saying the same thing.

In contrast, the *converse* of "P IMPLIES Q" is the statement "Q IMPLIES P." In terms of our example, the converse is:

If I am grumpy, then I am hungry. converse

This sounds like a rather different contention, and a truth table confirms this suspicion:

P	Q	P IMPLIES Q	Q IMPLIES P
T	**T**	**T**	**T**
T	**F**	**F**	**T**
F	**T**	**T**	**F**
F	**F**	**T**	**T**

Now the highlighted columns differ in the second and third row, confirming that an implication is generally *not* equivalent to its converse.

One final relationship: an implication and its converse together are equivalent to an iff statement, specifically, to these two statements together. For example,

If I am grumpy then I am hungry, and if I am hungry then I am grumpy.

are equivalent to the single statement:

I am grumpy iff I am hungry.

Once again, we can verify this with a truth table.

P	Q	$(P$ IMPLIES $Q)$	AND	$(Q$ IMPLIES $P)$	P IFF Q
T	**T**	T	**T**	T	**T**
T	**F**	F	**F**	T	**F**
F	**T**	T	**F**	F	**F**
F	**F**	T	**T**	T	**T**

The fourth column giving the truth values of

$$(P \text{ IMPLIES } Q) \text{ AND } (Q \text{ IMPLIES } P)$$

is the same as the sixth column giving the truth values of P IFF Q, which confirms that the AND of the implications is equivalent to the IFF statement.

3.3.2 Validity and Satisfiability

A *valid* formula is one which is always true. The simplest example is

$$P \text{ OR NOT}(P).$$

You can think about valid formulas as capturing fundamental logical truths. For example, a property of implication that we take for granted is the if one statement implies a second one, and the second one imlplies a third, then the first implies the third. The following valid formula comfirms the truth of this property of implication.

$$[(P \text{ IMPLIES } Q) \text{ AND } (Q \text{ IMPLIES } R)] \text{ IMPLIES } (P \text{ IMPLIES } R).$$

Equivalence of formulas is really a special case of validity. Namely, statements F and G and equivalent iff the statement F IFF G is valid. For example, the equivalence of the expressions 3.2 and 3.1 means that

$$(A \text{ OR } B) \text{ IFF } (A \text{ OR } (\text{NOT}(A) \text{ AND } B))$$

is valid. Of course validity can also be viewed as as aspect of equivalence. Namely, a formula is valid iff it is equivalent to **T**.

A *satisfiable* formula is one which can sometimes be true. One way satisfiability comes up is when there are a collection of system specifications. The job of the system designer is to come up with a system that follows all the specs. This means that the AND of all the specs had better be satisfiable or the system will be impossible (see Problem 3.7).

other book; depends on variable value

? error

There is also a close relationship between validity and satisfiability, namely, a statement P is valid iff its negation NOT(P) is *not* satisfiable.

3.4 The Algebra of Propositions

3.4.1 Propositions in Normal Form

Every propositional formula is equivalent to a "sum-of-products" or *disjunctive form*. More precisely, a disjunctive form is simply an OR of AND-terms, where each AND-term is a AND of variables or negations of variables, for example;

$$(A \text{ AND } B) \text{ OR } (A \text{ AND } C). \tag{3.3}$$

You can read a disjunctive form for any propositional formula directly from its truth table. For example, the formula

$$A \text{ AND } (B \text{ OR } C) \tag{3.4}$$

has truth table:

A	B	C	A AND (B OR C)
T	**T**	**T**	**T**
T	**T**	**F**	**T**
T	**F**	**T**	**T**
T	**F**	**F**	**F**
F	**T**	**T**	**F**
F	**T**	**F**	**F**
F	**F**	**T**	**F**
F	**F**	**F**	**F**

The formula (3.4) is true in the first row when A, B, and C are all true, that is, where A AND B AND C is true. It is also true in the second row where A AND B AND $\overline{C}$ ← not C is true, and in the third row when A AND $\overline{B}$ AND C is true, and that's all. So (3.4) is true exactly when

What is this again?
Not

$$(A \text{ AND } B \text{ AND } C) \text{ OR } (A \text{ AND } B \text{ AND } \overline{C}) \text{ OR } (A \text{ AND } \overline{B} \text{ AND } C) \tag{3.5}$$

is true. So (3.4) and (3.5) are equivalent. (more complex

The expression (3.5) is a disjunctive form where each AND-term is an AND of *every one* of the variables or their negations in turn. An expression of this form is called a *disjunctive normal form* (*DNF*). A DNF formula can often be simplified into a smaller disjuctive form. For example, the DNF (3.5) further simplifies to the equivalent disjunctive form (3.3) above.

Incidentally, this equivalence of A AND (B OR C) and (A AND B) OR (A AND C) is called the *distributive law* of AND over OR because of its obvious resemblance to the distributivity of multiplication over addition for numbers.

Applying the same reaasoning to the **F** entries of a truth table yields a *conjunctive form* for any formula, namely a AND of OR-terms, where the OR-terms are OR's only of variables or their negations. For example, formula (3.4) is false in the fourth row of its truth table (3.4.1) where A is **T**, B is **F** and C is **F**. But this is exactly the one row where ($\overline{A}$ OR B OR C) is **F**! Likewise, the (3.4) is false in the fifth row which is exactly where (A OR $\overline{B}$ OR $\overline{C}$) is **F**. This means that (3.4) will be **F** whenever the AND of these two OR-terms is false. Continuing in this way with the OR-terms corresponding to the remaining three rows where (3.4) is false, we get a *conjunctive normal form* (*CNF*) that is equivalent to (3.4), namely,

$$(\overline{A}\text{OR}B\text{OR}C)\text{AND}(A\text{OR}\overline{B}\text{OR}\overline{C})\text{AND}(A\text{OR}\overline{B}\text{OR}\overline{C})\text{AND}(A\text{OR}\overline{B}\text{OR}\overline{C})\text{AND}(A\text{OR}\overline{B}\text{OR}\overline{C}) \tag{3.6}$$

The methods above can obviously be applied to any truth table, which implies

Theorem 3.4.1. *Every propositional formula is equivalent to both a disjunctive normal form and a conjunctive normal form.*

how do the 2 relate more exactly

3.4.2 Proving Equivalences

A check of equivalence or validity by truth table runs out of steam pretty quickly: a proposition with n variables has a truth table with 2^n lines, so the effort required to check a proposition grows exponentially with the number of variables. For a proposition with just 30 variables, that's already over a billion lines to check!

An alternative approach that *sometimes* helps is to use algebra to prove equivalence. A lot of different operators may appear in a propositional formula, so a useful first step is to get rid of all but three: AND, OR, and NOT. This is easy because each of the operators is equivalent to a simple formula using only these three. For example, A IMPLIES B is equivalent to NOT(A) OR B. AND, OR, NOT formulas for the remaining operators are left to Problem 3.8.

must recognize all equivilances

We list below a bunch of equivalence axioms with the symbol "$\Longleftrightarrow$" between equivalent formulas. These axioms are important because they are all that's needed to prove every possible equivalence. We'll start with some equivalences for AND's that look like the familiar ones for multiplication of numbers:

equivilance

$$A \text{ AND } B \Longleftrightarrow B \text{ AND } A \qquad \text{commutativity of AND} \tag{3.7}$$

$$(A \text{ AND } B) \text{ AND } C \Longleftrightarrow A \text{ AND } (B \text{ AND } C) \qquad \text{associativity of AND} \tag{3.8}$$

$$\mathbf{T} \text{ AND } A \Longleftrightarrow A \qquad \text{identity for AND} \tag{3.9}$$

$$\mathbf{F} \text{ AND } A \Longleftrightarrow \mathbf{F} \qquad \text{zero for AND} \tag{3.10}$$

Problems in Class seem very different
Can't just work through

Three axioms that don't directly correspond to number properties are

Some stuff
Very basic

$$A \text{ AND } A \iff A \qquad \text{idempotence for AND} \tag{3.11}$$

$$A \text{ AND } \overline{A} \iff \mathbf{F} \qquad \text{contradiction for AND} \tag{3.12}$$

$$\text{NOT}(\overline{A}) \iff A \qquad \text{double negation} \tag{3.13}$$

$$\tag{3.14}$$

It is associativity (3.8) that justifies writing A AND B AND C without specifying whether it is parenthesized as A AND $(B$ AND $C)$ or $(A$ AND $B)$ AND C. That's because both ways of inserting parentheses yield equivalent formulas.

There are a corresponding set of equivalences for OR which we won't bother to list, except for the OR rule corresponding to contradiction for AND:

$$A \text{ OR } \overline{A} \iff \mathbf{T} \qquad \text{validity for OR} \tag{3.15}$$

$$\tag{3.16}$$

There is also a familiar rule connecting AND and OR:

$$A \text{ AND } (B \text{ OR } C) \iff (A \text{ AND } B) \text{ OR } (A \text{ AND } C) \qquad \text{distributivity of AND over OR} \tag{3.17}$$

Finally, there are *DeMorgan's Laws* which explain how to distribute NOT's over AND's and OR's:

$$\text{NOT}(A \text{ AND } B) \iff \overline{A} \text{ OR } \overline{B} \qquad \text{DeMorgan for AND} \tag{3.18}$$

$$\text{NOT}(A \text{ OR } B) \iff \overline{A} \text{ AND } \overline{B} \qquad \text{DeMorgan for OR} \tag{3.19}$$

All these axioms can be verified easily with truth tables.

Need to
practice|
study

These axioms are all that's needed to convert any formula to a disjunctive normal form. We can illustrate how they work by applying them to turn the negation of formula (3.4), namely,

$$\text{NOT}((A \text{ AND } B) \text{ OR } (A \text{ AND } C)). \tag{3.20}$$

into disjunctive normal form.

We start by applying DeMorgan's Law for OR to (3.20) in order to move the NOT deeper into the formula. This gives

$$\text{NOT}(A \text{ AND } B) \text{ AND NOT}(A \text{ AND } C).$$

Now applying Demorgan's Law for AND to the two innermost AND-terms, gives

$$(\overline{A} \text{ OR } \overline{B}) \text{ AND } (\overline{A} \text{ OR } \overline{C})). \tag{3.21}$$

At this point NOT only applies to variables, and we won't need Demorgan's Laws any further.

Now we will repeatedly apply the distributivity of AND over OR to turn (3.21) into a disjunctive form. To start, we'lll distribute ($\overline{A}$ OR $\overline{B}$) over OR to get

$$((\overline{A} \text{ OR } \overline{B}) \text{ AND } \overline{A}) \text{ OR } ((\overline{A} \text{ OR } \overline{B}) \text{ AND } \overline{C}).$$

Using distributivity over both AND's we get

$$((\overline{A} \text{ AND } \overline{A}) \text{ OR } (\overline{B} \text{ AND } \overline{A})) \text{ OR } ((\overline{A} \text{ AND } \overline{C}) \text{ OR } (\overline{B} \text{ AND } \overline{C})).$$

By the way, we've implicitly used commutativity (3.7) here to justify distributing over a AND from the right. Now applying idempotence to remove the duplicate occurrence of $\overline{A}$ we get

$$(\overline{A} \text{ OR } (\overline{B} \text{ AND } \overline{A})) \text{ OR } ((\overline{A} \text{ AND } \overline{C}) \text{ OR } (\overline{B} \text{ AND } \overline{C})).$$

Associativity now allows dropping the parentheses around the terms being OR'd to yield the following disjunctive form for (3.20):

$$\overline{A} \text{ OR } (\overline{B} \text{ AND } \overline{A}) \text{ OR } (\overline{A} \text{ AND } \overline{C}) \text{ OR } (\overline{B} \text{ AND } \overline{C}). \tag{3.22}$$

The last step is to turn each of these AND-terms into a disjunctive normal form with all three variables A, B, and C. We'll illustrate how to do this for the second AND-term ($\overline{B}$ AND $\overline{A}$). This term needs to mention C to be in normal form. To introduce C, we use validity for ORand identity for ANDto conclude that

$$(\overline{B} \text{ AND } \overline{A}) \iff (\overline{B} \text{ AND } \overline{A}) \text{ AND } (C \text{ OR } \overline{C}).$$

Now distributing ($\overline{B}$ AND $\overline{A}$) over the OR yield the disjunctive normal form

$$(\overline{B} \text{ AND } \overline{A} \text{ AND } C) \text{ OR } (\overline{B} \text{ AND } \overline{A} \text{ AND } \overline{C}).$$

Doing the same thing to the other AND-terms in (3.22) finally gives a disjunctive normal form for (3.4):

$$\begin{aligned}(\overline{A} \text{ AND } B \text{ AND } C) \text{ OR } (\overline{A} \text{ AND } B \text{ AND } \overline{C}) \text{ OR } (\overline{A} \text{ AND } \overline{B} \text{ AND } C) \text{ OR } (\overline{A} \text{ AND } \overline{B} \text{ AND } \overline{C})\text{OR}\\ (\overline{B} \text{ AND } \overline{A} \text{ AND } C) \text{ OR } (\overline{B} \text{ AND } \overline{A} \text{ AND } \overline{C})\text{OR}\\ (\overline{A} \text{ AND } \overline{C} \text{ AND } B) \text{ OR } (\overline{A} \text{ AND } \overline{C} \text{ AND } \overline{B})\text{OR}\\ (\overline{B} \text{ AND } \overline{C} \text{ AND } A) \text{ OR } (\overline{B} \text{ AND } \overline{C} \text{ AND } \overline{A}).\end{aligned}$$

You're probably bored enough by this long series of formulas that you won't be interested in questioning what they illustrate

Theorem 3.4.2. *Any propositional formula can be transformed into disjunctive normal form using the equivalences named above.*

What has this got to do with equivalence? That's easy: to prove that two formulas are equivalent, convert them both to disjunctive normal form over the set of variables that appear in the terms using commutativity to sort the variables and AND-terms where they all appear in some standard order. We claim the formulas are equivalent iff they have the same sorted disjunctive normal form. This is obvious if they do have the same disjunctive normal form. But conversely, the way we read off a disjunctive normal form from a truth table shows that two different sorted DNF's over the same set of variables correspoind to different truth tables and hence to inequivalent formulas. This proves

?? what how is that a theorm ?

Theorem 3.4.3 (Completeness of the propositional equivalence axioms)**.** *Two propositional formula are equivalent iff they can be proved equivalent using the equivalence axioms named above.*

The benefit of the axioms is that they leave room for ingeniously applying them to prove equivalences with less effort than the truth table method. Theorem 3.4.3 then adds the reassurance that the axioms are guaranteed to prove every equivalence, which is a great punchline for this section. But we don't want to mislead you: it's important to realize that using the strategy we gave for applying the axioms involves essentially the same effort it would take to construct truth tables, and there is no guarantee that applyiong the axioms will generally be any easier than using truth tables.

3.5 The SAT Problem

what is this again

valid - allways true

Determining whether or not a more complicated proposition is satisfiable is not so easy. How about this one?

can it be true

$$(P \text{ OR } Q \text{ OR } R) \text{ AND } (\overline{P} \text{ OR } \overline{Q}) \text{ AND } (\overline{P} \text{ OR } \overline{R}) \text{ AND } (\overline{R} \text{ OR } \overline{Q})$$

The general problem of deciding whether a proposition is satisfiable is called *SAT*. One approach to SAT is to construct a truth table and check whether or not a **T** ever appears, but as for validity, this approach quickly bogs down as for formulas with many variables because because truth tables grow exponentially with the number of variables.

Is there a more efficient solution to SAT? In particular, is there some, presumably very ingenious, procedure that determines in a number of steps that (grows *polynomially*—like n^2 or n^{14}—instead of exponentially, whether any given proposition is satisfiable or not? No one knows. And an awful lot hangs on the answer. It turns out that an efficient solution to SAT would immediately imply efficient solutions to many, many other important problems involving packing, scheduling, routing, and circuit verification, among other things. This would be wonderful, but there would also be worldwide chaos. Decrypting coded messages would also become an easy task (for most codes). Online financial transactions would be insecure and secret communications could be read by everyone. In later chapters we'll see some explanations of why this would happen.

Of course, the situation is the same for validity checking, since you can check for validity by checking for satisfiability of negated formula. This also explains why the simplification of formulas mentioned in Section 3.2 would be hard—validity testing is a special case of determining if a formula simplifies to **T**.

Recently there has been exciting progress on *sat-solvers* for practical applications like digital circuit verification. These programs find satisfying assignments with amazing efficiency even for formulas with millions of variables. Unfortunately, it's hard to predict which kind of formulas are amenable to sat-solver methods, and for formulas that are *un*satisfiable, sat-solvers generally get nowhere.

So no one has a good idea how to solve SAT in polynomial time, or how to prove that it can't be done—researchers are completely stuck. The problem of determining whether or not SAT has a polynomial time solution is known as the "**P** vs. **NP**" problem.[1] It is the outstanding unanswered question in theoretical computer science. It is also one of the seven Millenium Problems: the Clay Institute will award you $1,000,000 if you solve the **P** vs. **NP** problem.

3.6 Predicate Formulas

3.6.1 Quantifiers

The "for all" notation, ∀, introduced in Section 1.1 has already made an early appearance. For example, the predicate

$$\text{“}x^2 \geq 0\text{”}$$

is always true when x is a real number. That is,

$$\forall x \in \mathbb{R}.\, x^2 \geq 0$$

[1] **P** stands for problems whose instances can be solved in time that grows polynomially with the size of the instance. **NP** stands for **n**ondeterministic **p**olynomial time, but we'll leave an explanation of what that is to texts on the theory of computational complexity.

is a true statement. On the other hand, the predicate

$$\text{“}5x^2 - 7 = 0\text{”}$$

is only sometimes true; specifically, when $x = \pm\sqrt{7/5}$. There is a "there exists" notation, $\exists$, to indicate that a predicate is true for at least one, but not necessarily all objects. So

$$\exists x \in \mathbb{R}.\, 5x^2 - 7 = 0$$

is true, while

$$\forall x \in \mathbb{R}.\, 5x^2 - 7 = 0$$

is not true.

There are several ways to express the notions of "always true" and "sometimes true" in English. The table below gives some general formats on the left and specific examples using those formats on the right. You can expect to see such phrases hundreds of times in mathematical writing!

Always True

Universal

For all $x \in D$, $P(x)$ is true.	For all $x \in \mathbb{R}$, $x^2 \geq 0$.
$P(x)$ is true for every x in the set, D.	$x^2 \geq 0$ for every $x \in \mathbb{R}$.

Sometimes True

existential

There exists an $x \in D$ such that $P(x)$ is true.	There exists an $x \in \mathbb{R}$ such that $5x^2 - 7 = 0$.
$P(x)$ is true for some x in the set, D.	$5x^2 - 7 = 0$ for some $x \in \mathbb{R}$.
$P(x)$ is true for at least one $x \in D$.	$5x^2 - 7 = 0$ for at least one $x \in \mathbb{R}$.

All these sentences quantify how often the predicate is true. Specifically, an assertion that a predicate is always true is called a *universal* quantification, and an assertion that a predicate is sometimes true is an *existential* quantification. Sometimes the English sentences are unclear with respect to quantification:

If you can solve any problem we come up with,
then you get an A for the course. (3.23)

The phrase "you can solve any problem we can come up with" could reasonably be interpreted as either a universal or existential quantification:

you can solve *every* problem we come up with, (3.24)

or maybe

you can solve *at least one* problem we come up with. (3.25)

To be precise, let Probs be the set of problems we come up with, Solves(x) be the predicate "You can solve problem x", and G be the proposition, "You get an A for the course." Then the two different interpretations of (3.23) can be written as follows:

$$(\forall x \in \text{Probs. Solves}(x)) \text{ IMPLIES } G,$$

for (3.24), and

$$(\exists x \in \text{Probs. Solves}(x)) \text{ IMPLIES } G.$$

for (3.25).

3.6.2 Mixing Quantifiers

Many mathematical statements involve several quantifiers. For example, we already described

> Goldbach's Conjecture 1.1.7: Every even integer greater than 2 is the sum of two primes.

Let's write this out in more detail to be precise about the quantification:

be precise →
in math

> For every even integer n greater than 2, there exist primes p and q such that $n = p + q$.

Let Evens be the set of even integers greater than 2, and let Primes be the set of primes. Then we can write Goldbach's Conjecture in logic notation as follows:

$$\underbrace{\forall n \in \text{Evens.}}_{\substack{\text{for every even}\\ \text{integer } n > 2}} \; \underbrace{\exists p \in \text{Primes. } \exists q \in \text{Primes.}}_{\substack{\text{there exist primes}\\ p \text{ and } q \text{ such that}}} \; n = p + q.$$

3.6.3 Order of Quantifiers

Swapping the order of different kinds of quantifiers (existential or universal) usually changes the meaning of a proposition. For example, let's return to one of our initial, confusing statements:

"Every American has a dream."

This sentence is ambiguous because the order of quantifiers is unclear. Let A be the set of Americans, let D be the set of dreams, and define the predicate $H(a, d)$ to be "American a has dream d.". Now the sentence could mean there is a single dream that every American shares—such as the dream of owning their own home:

$$\exists d \in D \; \forall a \in A. \; H(a, d)$$

↑a dream ↑all Americans share

Or it could mean that every American has a personal dream:

diff order

$$\forall a \in A.\ \exists d \in D.\ H(a,d)$$

For example, some Americans may dream of a peaceful retirement, while others dream of continuing practicing their profession as long as they live, and still others may dream of being so rich they needn't think about work at all.

Order matters

Swapping quantifiers in Goldbach's Conjecture creates a patently false statement that every even number ≥ 2 is the sum of *the same* two primes:

$$\underbrace{\exists p \in \text{Primes}.\ \exists q \in \text{Primes}.}_{\substack{\text{there exist primes}\\ p \text{ and } q \text{ such that}}}\ \underbrace{\forall n \in \text{Evens}.}_{\substack{\text{for every even}\\ \text{integer } n > 2}}\ n = p + q.$$

3.6.4 Variables Over One Domain

When all the variables in a formula are understood to take values from the same nonempty set, D, it's conventional to omit mention of D. For example, instead of $\forall x \in D.\ \exists y \in D.\ Q(x,y)$ we'd write $\forall x \exists y.\ Q(x,y)$. The unnamed nonempty set that x and y range over is called the *domain of discourse*, or just plain *domain*, of the formula.

It's easy to arrange for all the variables to range over one domain. For example, Goldbach's Conjecture could be expressed with all variables ranging over the domain $\mathbb{N}$ as

$$\forall n.\ n \in \text{Evens IMPLIES } (\exists p.\ \exists q.\ p \in \text{Primes AND } q \in \text{Primes AND } n = p + q).$$

3.6.5 Negating Quantifiers

There is a simple relationship between the two kinds of quantifiers. The following two sentences mean the same thing:

It is not the case that everyone likes to snowboard.

There exists someone who does not like to snowboard.

In terms of logic notation, this follows from a general property of predicate formulas:

$$\text{NOT}(\forall x.\ P(x)) \quad \text{is equivalent to} \quad \exists x.\ \text{NOT}(P(x)).$$

Similarly, these sentences mean the same thing:

There does not exist anyone who likes skiing over magma.

Everyone dislikes skiing over magma.

We can express the equivalence in logic notation this way:

$$\text{NOT}(\exists x.\ P(x))\ \text{IFF}\ \forall x.\ \text{NOT}(P(x)). \tag{3.26}$$

The general principle is that *moving a* NOT *across a quantifier changes the kind of quantifier.*

what is predicate again?

3.6.6 Validity for Predicate Formulas

The idea of validity extends to predicate formulas, but to be valid, a formula now must evaluate to true no matter what values its variables may take over any unspecified domain, and no matter what interpretation a predicate variable may be given. For example, we already observed that the rule for negating a quantifier is captured by the valid assertion (3.26).

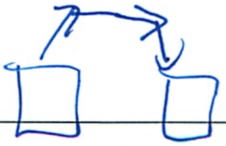
Another useful example of a valid assertion is

$$\exists x \forall y.\ P(x, y)\ \text{IMPLIES}\ \forall y \exists x.\ P(x, y). \tag{3.27}$$

Here's an explanation why this is valid:

> Let D be the domain for the variables and P_0 be some binary predicate[2] on D. We need to show that if
>
> $$\exists x \in D.\ \forall y \in D.\ P_0(x, y) \tag{3.28}$$
>
> holds under this interpretation, then so does
>
> $$\forall y \in D.\ \exists x \in D.\ P_0(x, y). \tag{3.29}$$
>
> So suppose (3.28) is true. Then by definition of $\exists$, this means that some element $d_0 \in D$ has the property that
>
> $$\forall y \in D.\ P_0(d_0, y).$$
>
> By definition of $\forall$, this means that
>
> $$P_0(d_0, d)$$
>
> is true for all $d \in D$. So given any $d \in D$, there is an element in D, namely, d_0, such that $P_0(d_0, d)$ is true. But that's exactly what (3.29) means, so we've proved that (3.29) holds under this interpretation, as required.

nice



← proof diagram hahaha visual thinking

[2] That is, a predicate that depends on two variables.

why not?

We hope this is helpful as an explanation, but we don't really want to call it a "proof." The problem is that with something as basic as (3.27), it's hard to see what more elementary axioms are ok to use in proving it. What the explanation above did was translate the logical formula (3.27) into English and then appeal to the meaning, in English, of "for all" and "there exists" as justification.

In contrast to (3.27), the formula

$$\forall y \exists x.\ P(x,y) \text{ IMPLIES } \exists x \forall y.\ P(x,y). \tag{3.30}$$

is *not* valid. We can prove this just by describing an interpretation where the hypothesis, $\forall y \exists x.\ P(x,y)$, is true but the conclusion, $\exists x \forall y.\ P(x,y)$, is not true. For example, let the domain be the integers and $P(x,y)$ mean $x > y$. Then the hypothesis would be true because, given a value, n, for y we could choose the value of x to be $n+1$, for example. But under this interpretation the conclusion asserts that there is an integer that is bigger than all integers, which is certainly false. An interpretation like this which falsifies an assertion is called a *counter model* to the assertion.

Problems for Section 3.1

Practice Problems

Problem 3.1.
Let the propositional variables P, Q, and R have the following meanings:

P = You get an A on the final exam.

Q = You do every exercise in the book.

R = You get an A in this class.

Write the following propositions using P, Q, and R and logical connectives.

(a) You get an A in this class, but you do not do every exercise in the book.

(b) You get an A on the final, you do every exercise in this book, and you get an A in this class.

(c) To get an A in this class, it is necessary for you to get an A on the final.

(d) You get an A on the final, but you don't do every exercise in this book; nevertheless, you get an A in this class.

Class Problems

Problem 3.2.
When the mathematician says to his student, "If a function is not continuous, then it is not differentiable," then letting D stand for "differentiable" and C for continuous, the only proper translation of the mathematician's statement would be

$$\text{NOT}(C) \text{ IMPLIES } \text{NOT}(D),$$

or equivalently,

$$D \text{ IMPLIES } C.$$

But when a mother says to her son, "If you don't do your homework, then you can't watch TV," then letting T stand for "watch TV" and H for "do your homework," a reasonable translation of the mother's statement would be

$$\text{NOT}(H) \text{ IFF } \text{NOT}(T),$$

or equivalently,

$$H \text{ IFF } T.$$

Explain why it is reasonable to translate these two IF-THEN statements in different ways into propositional formulas.

Homework Problems

Problem 3.3.
Describe a simple recursive procedure which, given a positive integer argument, n, produces a truth table whose rows are all the assignments of truth values to n propositional variables. For example, for $n = 2$, the table might look like:

T	**T**
T	**F**
F	**T**
F	**F**

Your description can be in English, or a simple program in some familiar language (say Scheme or Java), but if you do write a program, be sure to include some sample output.

Problems for Section 3.2

Class Problems

Problem 3.4.
Propositional logic comes up in digital circuit design using the convention that **T**

corresponds to 1 and **F** to 0. A simple example is a 2-bit *half-adder* circuit. This circuit has 3 binary inputs, a_1, a_0 and b, and 3 binary outputs, c, o_1, o_0. The 2-bit word a_1a_0 gives the binary representation of an integer, k, between 0 and 3. The 3-bit word cs_1s_0 gives the binary representation of $k + b$. The third output bit, c, is called the final *carry bit*.

So if k and b were both 1, then the value of a_1a_0 would be `01` and the value of the output cs_1s_0 would `010`, namely, the 3-bit binary representation of $1 + 1$.

In fact, the final carry bit equals 1 only when all three binary inputs are 1, that is, when $k = 3$ and $b = 1$. In that case, the value of cs_1s_0 is `100`, namely, the binary representation of $3 + 1$.

This 2-bit half-adder could be described by the following formulas:

$$
\begin{aligned}
c_0 &= b \\
s_0 &= a_0 \text{ XOR } c_0 \\
c_1 &= a_0 \text{ AND } c_0 && \text{the carry into column 1} \\
s_1 &= a_1 \text{ XOR } c_1 \\
c_2 &= a_1 \text{ AND } c_1 && \text{the carry into column 2} \\
c &= c_2.
\end{aligned}
$$

(a) Generalize the above construction of a 2-bit half-adder to an $n + 1$ bit half-adder with inputs $a_n, \ldots, a_1, a_0$ and b for arbitrary $n \geq 0$. That is, give simple formulas for s_i and c_i for $0 \leq i \leq n + 1$, where c_i is the carry into column i and $c = c_{n+1}$.

(b) Write similar definitions for the digits and carries in the sum of two $n + 1$-bit binary numbers $a_n \ldots a_1a_0$ and $b_n \ldots b_1b_0$.

Visualized as digital circuits, the above adders consist of a sequence of single-digit half-adders or adders strung together in series. These circuits mimic ordinary pencil-and-paper addition, where a carry into a column is calculated directly from the carry into the previous column, and the carries have to ripple across all the columns before the carry into the final column is determined. Circuits with this design are called *ripple-carry* adders. Ripple-carry adders are easy to understand and remember and require a nearly minimal number of operations. But the higher-order output bits and the final carry take time proportional to n to reach their final values.

(c) How many of each of the propositional operations does your adder from part (b) use to calculate the sum?

Homework Problems

Problem 3.5.
There are adder circuits that are much faster than the ripple-carry circuits of Problem 3.4. They work by computing the values in later columns for both a carry of 0 and a carry of 1, *in parallel*. Then, when the carry from the earlier columns finally arrives, the pre-computed answer can be quickly selected. We'll illustrate this idea by working out the equations for an $n + 1$-bit parallel half-adder.

Parallel half-adders are built out of parallel "add1" modules. An $n + 1$-bit add1 module takes as input the $n+1$-bit binary representation, $a_n \dots a_1 a_0$, of an integer, s, and produces as output the binary representation, $c\ p_n \dots p_1\ p_0$, of $s + 1$.

(a) A 1-bit add1 module just has input a_0. Write propositional formulas for its outputs c and p_0.

(b) Explain how to build an $n + 1$-bit parallel half-adder from an $n + 1$-bit add1 module by writing a propositional formula for the half-adder output, o_i, using only the variables a_i, p_i, and b.

We can build a double-size add1 module with $2(n + 1)$ inputs using two single-size add1 modules with $n + 1$ inputs. Suppose the inputs of the double-size module are $a_{2n+1}, \dots, a_1, a_0$ and the outputs are $c, p_{2n+1}, \dots, p_1, p_0$. The setup is illustrated in Figure 3.1.

Namely, the first single size add1 module handles the first $n + 1$ inputs. The inputs to this module are the low-order $n+1$ input bits $a_n, \dots, a_1, a_0$, and its outputs will serve as the first $n + 1$ outputs $p_n, \dots, p_1, p_0$ of the double-size module. Let $c_{(1)}$ be the remaining carry output from this module.

The inputs to the second single-size module are the higher-order $n + 1$ input bits $a_{2n+1}, \dots, a_{n+2}, a_{n+1}$. Call its first $n + 1$ outputs $r_n, \dots, r_1, r_0$ and let $c_{(2)}$ be its carry.

(c) Write a formula for the carry, c, in terms of $c_{(1)}$ and $c_{(2)}$.

(d) Complete the specification of the double-size module by writing propositional formulas for the remaining outputs, p_i, for $n + 1 \le i \le 2n + 1$. The formula for p_i should only involve the variables a_i, $r_{i-(n+1)}$, and $c_{(1)}$.

(e) Parallel half-adders are exponentially faster than ripple-carry half-adders. Confirm this by determining the largest number of propositional operations required to compute any one output bit of an n-bit add module. (You may assume n is a power of 2.)

add1-circuit-diagram.pdf

Redrawn

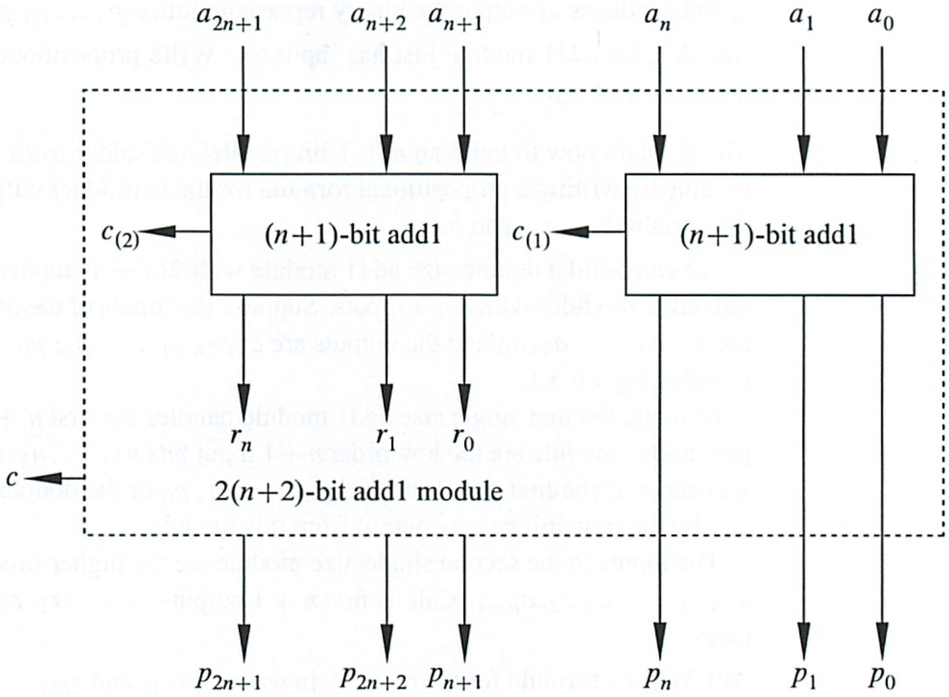


Figure 3.1 Structure of a Double-size Add1 Module.

Problems for Section 3.3

Class Problems

Problem 3.6. **(a)** Verify by truth table that

$$(P \text{ IMPLIES } Q) \text{ OR } (Q \text{ IMPLIES } P)$$

is valid.

(b) Let P and Q be propositional formulas. Describe a single formula, R, using AND's, OR's, and NOT's such that R is valid iff P and Q are equivalent.

Problem 3.7.
This problem[3] examines whether the following specifications are *satisfiable*:

1. If the file system is not locked, then
   (a) new messages will be queued.
   (b) new messages will be sent to the messages buffer.
   (c) the system is functioning normally, and conversely, if the system is functioning normally, then the file system is not locked.
2. If new messages are not queued, then they will be sent to the messages buffer.
3. New messages will not be sent to the message buffer.

(a) Begin by translating the five specifications into propositional formulas using four propositional variables:

$$\begin{aligned} L &::= \text{file system locked,} \\ Q &::= \text{new messages are queued,} \\ B &::= \text{new messages are sent to the message buffer,} \\ N &::= \text{system functioning normally.} \end{aligned}$$

(b) Demonstrate that this set of specifications is satisfiable by describing a single truth assignment for the variables L, Q, B, N and verifying that under this assignment, all the specifications are true.

(c) Argue that the assignment determined in part (b) is the only one that does the job.

[3] From Rosen, 5th edition, Exercise 1.1.36

Problems for Section 3.4

Practice Problems

Problem 3.8.
A half dozen different operators may appear in propositional formulas, but just AND, OR, and NOT are enough to do the job. That is because each of the operators is equivalent to a simple formula using only these three operators. For example, A IMPLIES B is equivalent to NOT(A) OR B. So all occurences of IMPLIES in a formula can be replaced using just NOT and OR.

(a) Write formulas using only AND, OR, NOT that are equivalent to each of A IFF B and A XOR B. Conclude that every propositional formula is equivalent to an AND-OR-NOT formula.

(b) Explain why you don't even need OR.

(c) Explain how to get by with the single operator NAND where A NAND B is equivalent by definition to NOT(A AND B).

Class Problems

Problem 3.9.
Explain how to find a conjunctive form for a propositional formula directly from a disjunctive form for its complement.

Homework Problems

Problem 3.10.
Use the equivalence axioms of Section 3.4.2 to convert the following formula to disjunctive form:

$$A \text{ XOR } B \text{ XOR } C.$$

Problems for Section 3.5

Homework Problems

Problem 3.11.
A 3-conjunctive form (3CF) formula is a conjunctive form formula in which each OR-term is a ORof at most 3 variables or negations of variables. Although it may be hard to tell if a propositional formula, F, is satisfiable, it is always easy to construct a formula, $C(F)$, that is

- in 3-conjunctive form,

- has at most 24 times as many occurrences of variables as F, and
- is satisfiable iff F is satisfiable.

To construct $\mathcal{C}(F)$, introduce a different new variables, one for each operator that occurs in F. For example, if F was

$$((P \text{ XOR } Q) \text{ XOR } R) \text{ OR } (\overline{P} \text{ AND } S) \tag{3.31}$$

we might use new variables X_1, X_2, O, and A corresponding to the the operator occurrences as follows:

$$((P \underbrace{\text{ XOR }}_{X_1} Q) \underbrace{\text{ XOR }}_{X_2} R) \underbrace{\text{ OR }}_{O} (\overline{P} \underbrace{\text{ AND }}_{A} S).$$

Next we write a formula that contrains each new variable to have the same truth value as the subformula determined by its corresponding operator. For the example above, these contraining formulas would be

$$\begin{gathered} X_1 \text{ IFF } (P \text{ XOR } Q), \\ X_2 \text{ IFF } (X_1 \text{ XOR } R), \\ A \text{ IFF } (\overline{P} \text{ AND } S), \\ O \text{ IFF } X_2 \text{ XOR } A. \end{gathered}$$

(a) Explain why the AND of the above four constraining formulas will be satisfiable iff (3.31) is satisfiable.

(b) Explain why any constraining formula will be equivalent to a 3CF formula with at most 24 occurrences of variables.

(c) Using the ideas illustrated in the previous parts, explain how to construct $\mathcal{C}(F)$ for an arbitrary propositional formula, F.

Problems for Section 3.6

Class Problems

Problem 3.12.

A media tycoon has an idea for an all-news television network called LNN: The Logic News Network. Each segment will begin with a definition of the domain of discourse and a few predicates. The day's happenings can then be communicated concisely in logic notation. For example, a broadcast might begin as follows:

> "THIS IS LNN. The domain of discourse is {Albert, Ben, Claire, David, Emily}. Let $D(x)$ be a predicate that is true if x is deceitful. Let $L(x, y)$ be a predicate that is true if x likes y. Let $G(x, y)$ be a predicate that is true if x gave gifts to y."

Translate the following broadcasted logic notation into (English) statements.

(a)

$$(\neg(D(\text{Ben}) \vee D(\text{David}))) \longrightarrow (L(\text{Albert}, \text{Ben}) \wedge L(\text{Ben}, \text{Albert}))$$

(b)

$$\begin{gathered}\forall x\ (x = \text{Claire} \wedge \neg L(x, \text{Emily})) \vee (x \neq \text{Claire} \wedge L(x, \text{Emily})) \wedge \\ \forall x\ (x = \text{David} \wedge L(x, \text{Claire})) \vee (x \neq \text{David} \wedge \neg L(x, \text{Claire}))\end{gathered}$$

(c)

$$\neg D(\text{Claire}) \longrightarrow (G(\text{Albert}, \text{Ben}) \wedge \exists\, x G(\text{Ben}, x))$$

(d)

$$\forall x \exists y \exists z\ (y \neq z) \wedge L(x, y) \wedge \neg L(x, z)$$

(e) How could you express "Everyone except for Claire likes Emily" using just propositional connectives *without* using any quantifiers ($\forall, \exists$)? Can you generalize to explain how *any* logical formula over this domain of discourse can be expressed without quantifiers? How big would the formula in the previous part be if it was expressed this way?

Problem 3.13.

The goal of this problem is to translate some assertions about binary strings into logic notation. The domain of discourse is the set of all finite-length binary strings: λ, 0, 1, 00, 01, 10, 11, 000, 001, ... (Here λ denotes the empty string.) In your translations, you may use all the ordinary logic symbols (including =), variables, and the binary symbols $\texttt{0}$, $\texttt{1}$ denoting 0, 1.

A string like $\texttt{01}x\texttt{0}y$ of binary symbols and variables denotes the *concatenation* of the symbols and the binary strings represented by the variables. For example, if the value of x is $\texttt{011}$ and the value of y is $\texttt{1111}$, then the value of $\texttt{01}x\texttt{0}y$ is the binary string $\texttt{0101101111}$.

Here are some examples of formulas and their English translations. Names for these predicates are listed in the third column so that you can reuse them in your solutions (as we do in the definition of the predicate NO-1S below).

Meaning	Formula	Name
x is a prefix of y	$\exists z\ (xz = y)$	$\text{PREFIX}(x, y)$
x is a substring of y	$\exists u \exists v\ (uxv = y)$	$\text{SUBSTRING}(x, y)$
x is empty or a string of 0's	$\text{NOT}(\text{SUBSTRING}(1, x))$	$\text{NO-1S}(x)$

(a) x consists of three copies of some string.

(b) x is an even-length string of 0's.

(c) x does not contain both a 0 and a 1.

(d) x is the binary representation of $2^k + 1$ for some integer $k \geq 0$.

(e) An elegant, slightly trickier way to define $\text{NO-1S}(x)$ is:

$$\text{PREFIX}(x, 0x). \qquad (*)$$

Explain why (*) is true only when x is a string of 0's.

Problem 3.14.
For each of the logical formulas, indicate whether or not it is true when the domain of discourse is $\mathbb{N}$, (the nonnegative integers 0, 1, 2, ...), $\mathbb{Z}$ (the integers), $\mathbb{Q}$ (the rationals), $\mathbb{R}$ (the real numbers), and $\mathbb{C}$ (the complex numbers). Add a brief explanation to the few cases that merit one.

$$\begin{array}{rlrcl}
 & \exists x & (x^2 & = & 2) \\
\forall x & \exists y & (x^2 & = & y) \\
\forall y & \exists x & (x^2 & = & y) \\
\forall x \neq 0 & \exists y & (xy & = & 1) \\
\exists x & \exists y & (x + 2y & = & 2) \land (2x + 4y = 5)
\end{array}$$

Problem 3.15.
Show that

$$(\forall x \exists y.\ P(x, y)) \longrightarrow \forall z.\ P(z, z)$$

is not valid by describing a counter-model.

Homework Problems

Problem 3.16.
Express each of the following predicates and propositions in formal logic notation.

The domain of discourse is the nonnegative integers, $\mathbb{N}$. Moreover, in addition to the propositional operators, variables and quantifiers, you may define predicates using addition, multiplication, and equality symbols, but no *constants* (like 0, 1,...) and no *exponentiation* (like x^y). For example, the proposition "n is an even number" could be written

$$\exists m. (m + m = n).$$

(a) n is the sum of two fourth-powers (a fourth-power is k^4 for some integer k).

Since the constant 0 is not allowed to appear explicitly, the predicate "$x = 0$" can't be written directly, but note that it could be expressed in a simple way as:

$$x + x = x.$$

Then the predicate $x > y$ could be expressed

$$\exists w. (y + w = x) \land (w \neq 0).$$

Note that we've used "$w \neq 0$" in this formula, even though it's technically not allowed. But since "$w \neq 0$" is equivalent to the allowed formula "$\neg(w + w = w)$," we can use "$w \neq 0$" with the understanding that it abbreviates the real thing. And now that we've shown how to express "$x > y$," it's ok to use it too.

(b) $x = 1$.

(c) m is a divisor of n (notation: $m \mid n$)

(d) n is a prime number (hint: use the predicates from the previous parts)

(e) n is a power of 3.

Problem 3.17.
Translate the following sentence into a predicate formula:

> There is a student who has emailed exactly two other people in the class, besides possibly herself.

The domain of discourse should be the set of students in the class; in addition, the only predicates that you may use are

- equality, and
- $E(x, y)$, meaning that "x has sent e-mail to y."

2/13

4 Mathematical Data Types

4.1 Sets

We've been assuming that the concepts of sets, sequences, and functions are already familiar ones, and we've mentioned them repeatedly. Now we'll do a quick review of the definitions.

Informally, a *set* is a bunch of objects, which are called the *elements* of the set. The elements of a set can be just about anything: numbers, points in space, or even other sets. The conventional way to write down a set is to list the elements inside curly-braces. For example, here are some sets:

? did I read this in previous book?

$$\begin{array}{rcll} A & = & \{\text{Alex}, \text{Tippy}, \text{Shells}, \text{Shadow}\} & \text{dead pets} \\ B & = & \{\text{red}, \text{blue}, \text{yellow}\} & \text{primary colors} \\ C & = & \{\{a, b\}, \{a, c\}, \{b, c\}\} & \text{a set of sets} \end{array}$$

This works fine for small finite sets. Other sets might be defined by indicating how to generate a list of them:

$$D = \{1, 2, 4, 8, 16, \dots\} \qquad \text{the powers of 2}$$

Or did in previous section

The order of elements is not significant, so $\{x, y\}$ and $\{y, x\}$ are the same set written two different ways. Also, any object is, or is not, an element of a given set —there is no notion of an element appearing more than once in a set.[1] So writing $\{x, x\}$ is just indicating the same thing twice, namely, that x is in the set. In particular, $\{x, x\} = \{x\}$.

The expression $e \in S$ asserts that e is an element of set S. For example, $32 \in D$ and blue $\in B$, but Tailspin $\notin A$ —yet.

Sets are simple, flexible, and everywhere. You'll find some set mentioned in nearly every section of this text.

4.1.1 Some Popular Sets

Mathematicians have devised special symbols to represent some common sets.

[1] It's not hard to develop a notion of *multisets* in which elements can occur more than once, but multisets are not ordinary sets.

Memorize

symbol	set	elements
$\emptyset$	the empty set	none
$\mathbb{N}$	nonnegative integers	$\{0, 1, 2, 3, \ldots\}$
$\mathbb{Z}$	integers	$\{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$
$\mathbb{Q}$	rational numbers	$\frac{1}{2},\ -\frac{5}{3},\ 16,$ etc.
$\mathbb{R}$	real numbers	$\pi,\ e,\ -9,\ \sqrt{2},$ etc.
$\mathbb{C}$	complex numbers	$i,\ \frac{19}{2},\ \sqrt{2}-2i,$ etc.

A superscript “+” restricts a set to its positive elements; for example, $\mathbb{R}^+$ denotes the set of positive real numbers. Similarly, $\mathbb{R}^-$ denotes the set of negative reals.

4.1.2 Comparing and Combining Sets

The expression $S \subseteq T$ indicates that set S is a *subset* of set T, which means that every element of S is also an element of T (it could be that $S = T$). For example, $\mathbb{N} \subseteq \mathbb{Z}$ and $\mathbb{Q} \subseteq \mathbb{R}$ (every rational number is a real number), but $\mathbb{C} \not\subseteq \mathbb{Z}$ (not every complex number is an integer).

As a memory trick, notice that the $\subseteq$ points to the smaller set, just like a $\leq$ sign points to the smaller number. Actually, this connection goes a little further: there is a symbol $\subset$ analogous to $<$. Thus, $S \subset T$ means that S is a subset of T, but the two are *not* equal. So $A \subseteq A$, but $A \not\subset A$, for every set A.

Cool

There are several ways to combine sets. Let’s define a couple of sets for use in examples:

$$X ::= \{1, 2, 3\}$$
$$Y ::= \{2, 3, 4\}$$

- The *union* of sets X and Y (denoted $X \cup Y$) contains all elements appearing in X or Y or both. Thus, $X \cup Y = \{1, 2, 3, 4\}$.

~~M~~ OR

- The *intersection* of X and Y (denoted $X \cap Y$) consists of all elements that appear in *both* X and Y. So $X \cap Y = \{2, 3\}$.

AND

- The *set difference* of X and Y (denoted $X - Y$) consists of all elements that are in X, but not in Y. Therefore, $X - Y = \{1\}$ and $Y - X = \{4\}$.

4.1.3 Complement of a Set

Sometimes we are focused on a particular domain, D. Then for any subset, A, of D, we define $\overline{A}$ to be the set of all elements of D *not* in A. That is, $\overline{A} ::= D - A$. The set $\overline{A}$ is called the *complement* of A.

For example, when the domain we're working with is the real numbers, the complement of the positive real numbers is the set of negative real numbers together with zero. That is,

$$\overline{\mathbb{R}^+} = \mathbb{R}^- \cup \{0\}.$$

It can be helpful to rephrase properties of sets using complements. For example, two sets, A and B, are said to be *disjoint* iff they have no elements in common, that is, $A \cap B = \emptyset$. This is the same as saying that A is a subset of the complement of B, that is, $A \subseteq \overline{B}$.

ha!

4.1.4 Power Set

The set of all the subsets of a set, A, is called the *power set*, $\mathcal{P}(A)$, of A. So $B \in \mathcal{P}(A)$ iff $B \subseteq A$. For example, the elements of $\mathcal{P}(\{1, 2\})$ are $\emptyset, \{1\}, \{2\}$ and $\{1, 2\}$. ← also combo

More generally, if A has n elements, then there are 2^n sets in $\mathcal{P}(A)$. For this reason, some authors use the notation 2^A instead of $\mathcal{P}(A)$.

why is this helpful

4.1.5 Set Builder Notation

An important use of predicates is in *set builder notation*. We'll often want to talk about sets that cannot be described very well by listing the elements explicitly or by taking unions, intersections, etc., of easily-described sets. Set builder notation often comes to the rescue. The idea is to define a *set* using a *predicate*; in particular, the set consists of all values that make the predicate true. Here are some examples of set builder notation:

statements that are sometimes true

where

$$\begin{aligned} A &::= \{n \in \mathbb{N} \mid n \text{ is a prime and } n = 4k + 1 \text{ for some integer } k\} \\ B &::= \{x \in \mathbb{R} \mid x^3 - 3x + 1 > 0\} \\ C &::= \{a + bi \in \mathbb{C} \mid a^2 + 2b^2 \le 1\} \end{aligned}$$

The set A consists of all nonnegative integers n for which the predicate

"n is a prime and $n = 4k + 1$ for some integer k"

is true. Thus, the smallest elements of A are:

$$5, 13, 17, 29, 37, 41, 53, 57, 61, 73, \ldots.$$

Trying to indicate the set A by listing these first few elements wouldn't work very well; even after ten terms, the pattern is not obvious! Similarly, the set B consists of all real numbers x for which the predicate

$$x^3 - 3x + 1 > 0$$

is true. In this case, an explicit description of the set B in terms of intervals would require solving a cubic equation. Finally, set C consists of all complex numbers $a + bi$ such that:

$$a^2 + 2b^2 \le 1$$

This is an oval-shaped region around the origin in the complex plane.

4.1.6 Proving Set Equalities

Two sets are defined to be equal if they contain the same elements. That is, $X = Y$ means that $z \in X$ if and only if $z \in Y$, for all elements, z. (This is actually the first of the ZFC axioms.) So set equalities can be formulated and proved as "iff" theorems. For example:

Theorem 4.1.1 (*Distributive Law* for Sets)**.** *Let A, B, and C be sets. Then:*

6.041 →

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \tag{4.1}$$

Proof. The equality (4.1) is equivalent to the assertion that

$$z \in A \cap (B \cup C) \quad \text{iff} \quad z \in (A \cap B) \cup (A \cap C) \tag{4.2}$$

for all z. Now we'll prove (4.2) by a chain of iff's.

First we need a rule for distributing a propositional AND operation over an OR operation. It's easy to verify by truth-table that

Lemma 4.1.2. *The propositional formula*

$$P \text{ AND } (Q \text{ OR } R)$$

and

$$(P \text{ AND } Q) \text{ OR } (P \text{ AND } R)$$

are equivalent.

Now we have

$$\begin{array}{lll}
z \in A \cap (B \cup C) & & \\
\quad \text{iff} & (z \in A) \text{ AND } (z \in B \cup C) & (\text{def of } \cap) \\
\quad \text{iff} & (z \in A) \text{ AND } (z \in B \text{ OR } z \in C) & (\text{def of } \cup) \\
\quad \text{iff} & (z \in A \text{ AND } z \in B) \text{ OR } (z \in A \text{ AND } z \in C) & (\text{Lemma 4.1.2}) \\
\quad \text{iff} & (z \in A \cap B) \text{ OR } (z \in A \cap C) & (\text{def of } \cap) \\
\quad \text{iff} & z \in (A \cap B) \cup (A \cap C) & (\text{def of } \cup)
\end{array}$$

intermediate truth

■

4.2 Sequences

Order matters

Sets provide one way to group a collection of objects. Another way is in a *sequence*, which is a list of objects called *terms* or *components*. Short sequences are commonly described by listing the elements between parentheses; for example, (a, b, c) is a sequence with three terms.

While both sets and sequences perform a gathering role, there are several differences.

- The elements of a set are required to be distinct, but terms in a sequence can be the same. Thus, (a, b, a) is a valid sequence of length three, but $\{a, b, a\}$ is a set with two elements —not three.

- The terms in a sequence have a specified order, but the elements of a set do not. For example, (a, b, c) and (a, c, b) are different sequences, but $\{a, b, c\}$ and $\{a, c, b\}$ are the same set.

- Texts differ on notation for the *empty sequence*; we use λ for the empty sequence.

The product operation is one link between sets and sequences. A *product of sets*, $S_1 \times S_2 \times \cdots \times S_n$, is a new set consisting of all sequences where the first component is drawn from S_1, the second from S_2, and so forth. For example, $\mathbb{N} \times \{a, b\}$ is the set of all pairs whose first element is a nonnegative integer and whose second element is an a or a b:

first ✓ for each

$$\mathbb{N} \times \{a, b\} = \{(0, a), (0, b), (1, a), (1, b), (2, a), (2, b), \ldots\}$$

A product of n copies of a set S is denoted S^n. For example, $\{0, 1\}^3$ is the set of all 3-bit sequences:

$$\{0, 1\}^3 = \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$$

4.3 Functions

A *function* assigns an element of one set, called the *domain*, to elements of another set, called the *codomain*. The notation

$$f : A \to B$$

places/adds to set.

indicates that f is a function with domain, A, and codomain, B. The familiar notation "$f(a) = b$" indicates that f assigns the element $b \in B$ to a. Here b would be called the *value* of f at *argument* a.

Functions are often defined by formulas as in:

$$f_1(x) ::= \frac{1}{x^2}$$

So other way?

explain more

where x is a real-valued variable, or

$$f_2(y, z) ::= y10yz$$

where y and z range over binary strings, or

$$f_3(x, n) ::= \text{ the pair } (n, x)$$

where n ranges over the nonnegative integers.

A function with a finite domain could be specified by a table that shows the value of the function at each element of the domain. For example, a function $f_4(P, Q)$ where P and Q are propositional variables is specified by:

P	Q	$f_4(P, Q)$
T	**T**	**T**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

Notice that f_4 could also have been described by a formula:

$$f_4(P, Q) ::= [P \text{ IMPLIES } Q].$$

assignment

A function might also be defined by a procedure for computing its value at any element of its domain, or by some other kind of specification. For example, define $f_5(y)$ to be the length of a left to right search of the bits in the binary string y until a 1 appears, so

$$\begin{aligned} f_5(0010) &= 3, \\ f_5(100) &= 1, \\ f_5(0000) &\text{ is undefined.} \end{aligned}$$

Notice that f_5 does not assign a value to any string of just 0's. This illustrates an important fact about functions: they need not assign a value to every element in the domain. In fact this came up in our first example $f_1(x) = 1/x^2$, which does not assign a value to 0. So in general, functions may be *partial functions*, meaning that there may be domain elements for which the function is not defined. If a function is defined on every element of its domain, it is called a *total function*.

It's often useful to find the set of values a function takes when applied to the elements in *a set* of arguments. So if $f : A \to B$, and S is a subset of A, we define $f(S)$ to be the set of all the values that f takes when it is applied to elements of S. ??? That is,

$$f(S) ::= \{b \in B \mid f(s) = b \text{ for some } s \in S\}.$$

For example, if we let $[r, s]$ denote the interval from r to s on the real line, then $f_1([1, 2]) = [1/4, 1]$.

For another example, let's take the "search for a `1`" function, f_5. If we let X be the set of binary words which start with an even number of `0`'s followed by a `1`, then $f_5(X)$ would be the odd nonnegative integers.

Applying f to a set, S, of arguments is referred to as "applying f pointwise to S", and the set $f(S)$ is referred to as the *image* of S under f.[2] The set of values that arise from applying f to all possible arguments is called the *range* of f. That is,

$$\text{range}(f) ::= f(\text{domain}(f)).$$

Some authors refer to the codomain as the range of a function, but they shouldn't. The distinction between the range and codomain will be important in Sections 5.1 and 5.2 when we relate sizes of sets to properties of functions between them.

4.3.1 Function Composition

Doing things step by step is a universal idea. Taking a walk is a literal example, but so is cooking from a recipe, executing a computer program, evaluating a formula, and recovering from substance abuse.

Abstractly, taking a step amounts to applying a function, and going step by step corresponds to applying functions one after the other. This is captured by the operation of *composing* functions. Composing the functions f and g means that first f applied is to some argument, x, to produce $f(x)$, and then g is applied to that result to produce $g(f(x))$.

Definition 4.3.1. For functions $f : A \to B$ and $g : B \to C$, the *composition*, $g \circ f$, of g with f is defined to be the function $h : A \to C$ defined by the rule:

$$(g \circ f)(x) = h(x) ::= g(f(x)),$$

[2] There is a picky distinction between the function f which applies to elements of A and the function which applies f pointwise to subsets of A, because the domain of f is A, while the domain of pointwise-f is $\mathcal{P}(A)$. It is usually clear from context whether f or pointwise-f is meant, so there is no harm in overloading the symbol f in this way. ???

for all $x \in A$.

Function composition is familiar as a basic concept from elementary calculus, and it plays an equally basic role in discrete mathematics.

Still don't get — reread

4.4 Binary Relations

Relations are another fundamental mathematical data type. Equality and "less-than" are very familiar examples of mathematical relations. These are called *binary relations* because they apply to a pair (a, b) of objects; the equality relation holds for the pair when $a = b$, and less-than holds when a and b are real numbers and $a < b$.

binary since T or F?

In this chapter we'll define some basic vocabulary and properties of binary relations.

4.5 Binary Relations and Functions

? 4.4.1?

Binary relations are far more general than equality or less-than. Here's the official definition:

Definition 4.5.1. A *binary relation*, R, consists of a set, A, called the *domain* of R, a set, B, called the *codomain* of R, and a subset of $A \times B$ called the *graph of* R.

Notice that Definition 4.5.1 is exactly the same as the definition in Section 4.3 of a *function*, except that it doesn't require the functional condition that, for each domain element, a, there is *at most* one pair in the graph whose first coordinate is a. So a function is a special case of a binary relation.

A relation whose domain is A and codomain is B is said to be "between A and B", or "from A to B." When the domain and codomain are the same set, A, we simply say the relation is "on A." It's common to use infix notation "$a \ R \ b$" to mean that the pair (a, b) is in the graph of R.

Example?

For example, we can define an "in-charge of" relation, T, for MIT in Spring '10 to have domain equal to the set, F, of names of the faculty and codomain equal to all the set, N, of subject numbers in the current catalogue. The graph of T contains precisely the pairs of the form

$$(\langle\text{instructor-name}\rangle, \langle\text{subject-num}\rangle)$$

such that the faculty member named ⟨instructor-name⟩ is in charge of the subject with number ⟨subject-num⟩ in Spring '10. So graph(T) contains pairs like

```
     (A. R. Meyer,   6.042),
     (A. R. Meyer,   18.062),
     (A. R. Meyer,   6.844),
     (T. Leighton,   6.042),
     (T. Leighton,   18.062),
     (G, Freeman,    6.011),
     (G, Freeman,    6.UAT),
     (G. Freeman,    6.881)
     (G. Freeman,    6.882)
  →  (T. Eng,        6.UAT)
     (J. Guttag,     6.00)
           ⋮
```

This is a surprisingly complicated relation: Meyer is in charge of subjects with three numbers. Leighton is also in charge of subjects with two of these three numbers —because the same subject, Mathematics for Computer Science, has two numbers: 6.042 and 18.062, and Meyer and Leighton are co-in-charge of the subject. Freeman is in-charge of even more subjects numbers (around 20), since as Department Education Officer, he is in charge of whole blocks of special subject numbers. Some subjects, like 6.844 and 6.00 have only one person in-charge. Some faculty, like Guttag, are in charge of only one subject number, and no one else is co-in-charge of his subject, 6.00.

Some subjects in the codomain, N, do not appear in the list —that is, they are not an element of any of the pairs in the graph of T; these are the Fall term only subjects. Similarly, there are faculty in the domain, F, who do not appear in the list because all their in-charge subjects are Fall term only.

Or don't teach in

not allowed

4.6 Images and Inverse Images

what is this?

If R is a binary relation from A to B, and X is any set, define the image, $R(X)$, of X under R to be the set elements of B that are related to something in X, namely,

$$R(X) ::= \{b \in B \mid xRb \text{ for some } x \in X\}.$$

In terms of arrows, $R(X)$ is simply the set of endpoints of arrows that start at points in X.

which endpoints?

So, $T(\{\text{A. Meyer}\})$ gives the subject numbers that Meyer is in charge of in Spring '10. In fact, $T(\{\text{A. Meyer}\}) = \{6.042, 18.062, 6.844\}$. Since the domain, F, is the set of all in-charge faculty, $T(F)$ is exactly the set of *all* Spring '10 subjects being taught. Similarly, $T^{-1}(N)$ is the set of people in-charge of a Spring '10 subject.

The faculty in charge of 6.UAT in Spring '10 can be found by taking the pairs of the form

$$(\langle\text{instructor-name}\rangle, 6.UAT)$$

in the graph of the teaching relation, T, and then just listing the left hand sides of these pairs; these turn out to be just Eng and Freeman.

The introductory course 6 subjects have numbers that start with *6.0*. So we can likewise find out all the instructors in-charge of introductory course 6 subjects this term, by taking all the pairs of the form $(\langle\text{instructor-name}\rangle, 6.0\ldots)$ and list the left hand sides of these pairs. For example, from the part of the graph of T shown above, we can see that Meyer, Leighton, Freeman, and Guttag are in-charge of introductory subjects this term.

is this like MySQL?

2/22 Meyer email: yes

These are examples of taking an *inverse image* of a set under a relation. If R is a binary relation from A to B, the inverse, R^{-1}, of R, is the binary relation from B to A defined by reversing relation between elements. Namely,

$$b\ R^{-1}\ a \quad \text{iff} \quad a\ R\ b,$$

for all $a \in A$ and $b \in B$. In terms of arrows, R^{-1} is the relation obtained by reversing the direction of the arrows of R.

The *inverse image* of any set X under R, is simply the image, $R^{-1}(X)$, of X under R^{-1}.

For example, let D be the set of introductory course 6 subject numbers. So $T^{-1}(D)$, the inverse image of the set D under the relation, T, is the set of all faculty members in-charge of introductory course 6 subjects in Spring '10.

It gets interesting when we write composite expressions mixing images, inverse images and set operations. For example, $T(T^{-1}(D))$ is the set of Spring '10 subjects that have people in-charge who also are in-charge of introductory subjects. So $T(T^{-1}(D)) - D$ are the advanced subjects with someone in-charge who is also in-charge of an introductory subject. Similarly, $T^{-1}(D) \cap T^{-1}(N - D)$ is the set of faculty teaching both an introductory *and* an advanced subject in Spring '10.

Gets complex fast

Still somewhat confused want to see lecture slides

2/23 Much clearer

4.7 Glossary of Symbols

symbol	meaning
$\in$	is a member of
$\subseteq$	is a subset of
$\subset$	is a proper subset of
$\cup$	set union
$\cap$	set intersection
$\overline{A}$	complement of a set, A
$\mathcal{P}(A)$	powerset of a set, A
$\emptyset$	the empty set, $\{\}$
$\mathbb{N}$	nonnegative integers
$\mathbb{Z}$	integers
$\mathbb{Z}^+$	positive integers
$\mathbb{Z}^-$	negative integers
$\mathbb{Q}$	rational numbers
$\mathbb{R}$	real numbers
$\mathbb{C}$	complex numbers
λ	the empty string/list Sequence

4.8

Problems for Section 4.1

Homework Problems

Problem 4.1.
Let A, B, and C be sets. Prove that:

$$A \cup B \cup C = (A - B) \cup (B - C) \cup (C - A) \cup (A \cap B \cap C). \qquad (4.3)$$

Hint: P OR Q OR R is equivalent to

$$(P \text{ AND } \overline{Q}) \text{ OR } (Q \text{ AND } \overline{R}) \text{ OR } (R \text{ AND } \overline{P}) \text{ OR } (P \text{ AND } Q \text{ AND } R).$$