Part I

Proofs

Mathematical Proofs

A proof is a method of establishing truth. What constitutes a proof differs among fields.

- *Legal* truth is decided by a jury based on allowable evidence presented at trial.
- *Authoritative* truth is specified by a trusted person or organization.
- *Scientific* truth[1] is confirmed by experiment.
- *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

[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.

am." It comes from the beginning of a 17th century essay by the mathematician/philospher, 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 ~~first axiom~~ idea. However, with just a few more lines of argument in this vein, Descartes goes on to conclude that there is an infinitely beneficent God. Whether or not you believe in a beneficent God, you'll probably agree that any very short proof of God's existence is bound to be far-fetched. So even in masterful hands, this approach is not reliable.

Mathematics ~~also~~ has its own specific notion of "proof."

Definition. A *formal 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. These three ideas are explained ~~in Chapter 1, and Chapter 2~~ in the following chapters, beginning with propositions in chapter 1. We will then ~~describe the most common templates~~ provide lots of examples of proofs

describes some basic ways of organizing proofs.

0.0.1 Problems

Class Problems

and even some examples of "false proofs" (i.e., arguments that look like a proof but that contain mis-steps, or deductions that aren't so logical when examined closely).

Albert - we have lots of "propositions" that are not mathematical in the later sections, so we should remove "mathematical" from the definition.

Chapter 1

Propositions

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

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*!".

Being "mathematical" is a more serious restriction. For example, "Albert's wife's name is 'Irene' " is a true statement, and you could prove it by presenting legal documents and the testimony of their children. But it isn't a proposition because it is not a *mathematical* statement. There is no mathematical definition of Al-

move to position A1

bert or Irene, and statements about them are not part of mathematics. Propositions must be about well-defined mathematical objects like numbers, sets, functions, relations, etc., and they must be stated using mathematically precise language. For example, this with a few examples.

Proposition 1.0.1. $2 + 3 = 5$.

This is a true proposition.

Proposition 1.0.2. *The binary representation of every nonnegative integer starts with a 1.*

This is a false proposition. It could be fixed by ruling out the nonnegative integer zero. So the following proposition is true:

Proposition 1.0.3. *The binary representation of every positive integer starts with a 1.*

unfortunately, it is not always easy to decide if a proposition is true or false, or even what the proposition means. In part, this is because the English language is riddled with ambiguities. For example, here

1.1 Compound Propositions

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

statements

1. "You may have cake, or you may have ice cream."

2. "If pigs can fly, then you can understand the Chebyshev bound."

3. "If you can solve any problem we come up with, then you get an *A* for the course."

4. "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? If the second sentence is true, then is the Chebyshev bound incomprehensible? If you can solve some problems we come up with but not all, then do you get an *A* for the course? And can you still get an *A* even if you can't solve any of the problems? Does the last sentence imply that all Americans have the same dream or might some of them have different dreams?

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 mini-language for talking about logical relationships. This language mostly uses ordinary English words and phrases such as "or", "implies", and "for all". But mathematicians endow these words with definitions more precise than those found in an ordinary dictionary. Without knowing these definitions, you might sometimes get the gist of statements in this language, but you would regularly get misled about what they really meant.

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.

1.2 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 variables, like propositions, can take on only the values $\mathbb{T}$ (true) and $\mathbb{F}$ (false). Such true/false variables are sometimes called *Boolean variables* after their inventor, George —you guessed it —Boole.

these should all update to reflect we are in section 1.1

1.1.1

1.2.1 "Not", "And", and "Or"

We can precisely define these special words using *truth tables*. For example, if P denotes an arbitrary proposition, then the truth of the proposition "NOT P" is defined by 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 setting of the variables. For example, the truth table for the proposition "P AND Q" has four lines, since the two variables can be set in four different ways:

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

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 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 when 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 having both having and eating, you should use "exclusive-or" (XOR):

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

1.1.2

1.2.2 "Implies"

The least intuitive connecting word 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 the Riemann Hypothesis is true, then $x^2 \geq 0$ for every real number x."

The Riemann Hypothesis is a famous unresolved ~~open question~~ conjecture in mathematics (i.e., no one knows if it is true or false).

~~Now, we told you before that no one knows whether Goldbach's Conjecture is true or false~~. But that doesn't prevent you from answering the question! This proposition has the form $P \longrightarrow Q$ where the *hypothesis*, P, is "~~Goldbach's Conjecture~~ the Riemann Hypothesis is true" and the *conclusion*, Q, is "$x^2 \geq 0$ for every 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 implica-

tions.

"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*!

In contrast, here's an example of a false implication:

"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.

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

1.1.3

1.2.3 "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 are logically equivalent; 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

The following if-and-only-if statement is true for every real number x:

$$x^2 - 4 \geq 0 \quad \text{iff} \quad |x| \geq 2$$

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

1.2.4 Problems

Class Problems

Homework Problems

position A2 is here as section 1.1.4

position A3 is here as section 1.1.5

1.2

1.3 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:

Daniel: we need to put "or" etc in CAPS to be consistent.

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

CAPS

The symbol `||` denotes "or", and the symbol `&&` denotes "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 this way:

$$A \text{ or } ((\text{not } A) \text{ and } B) \tag{1.1}$$

A truth table reveals that this complicated expression is logically equivalent to

$$A \text{ or } B. \tag{1.2}$$

A	B	A or ((not A) and B)	A or B
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

This means that we can simplify the code snippet without changing the program's behavior:

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

$\vdots$

(further instructions)

The equivalence of (1.1) and (1.2) can also be confirmed reasoning by cases:

A is T. Then an expression of the form (A or anything) will have truth value T. Since both expressions are of this form, both have the same truth value in this case, namely, T.

A is F. Then (A or P) will have the same truth value as P for any proposition, P. So (1.2) has the same truth value as B. Similarly, (1.1) has the same truth value as ((not F) and B), which also has the same value as B. So in this case, both expressions will have the same truth value, namely, the value of B.

Rewriting a logical expression involving many variables in the simplest form is both difficult and important. Simplifying expressions in software might slightly increase the speed of your program. But, more significantly, chip designers face essentially the same challenge. However, instead of minimizing && and || symbols in a program, their job is to minimize the number of analogous physical devices on a chip. The payoff is potentially enormous: a chip with fewer devices is smaller, consumes less power, has a lower defect rate, and is cheaper to manufacture.

1.3.1 Cryptic Notation

Programming languages use symbols like && and ! in place of words like "and" and "not". Mathematicians have devised their own cryptic symbols to represent

~~these words, which~~ are summarized in the table below.

English	Cryptic 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	$P \longleftrightarrow Q$

For example, using this notation, "If P and not Q, then R" would be written:

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

This symbolic language is helpful for writing complicated logical expressions compactly. But words such as "OR" and "IMPLIES," generally serve just as well as the cryptic symbols $\vee$ and $\longrightarrow$, and their meaning is easy to remember. ~~So we'll use the cryptic notation sparingly, and we advise you to do the same.~~

1.3.2 Logically Equivalent Implications

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.

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

P	Q	P IMPLIES Q	$\overline{Q}$ IMPLIES $\overline{P}$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Sure enough, the columns of truth values under these two statements are the same, which precisely means they are equivalent. In general, "(NOT Q) IMPLIES (NOT P)" is called the *contrapositive* of the implication "P IMPLIES Q." And, as we've just shown, the two 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.

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

Thus, an implication *is* logically equivalent to its contrapositive but is *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 $P)$	Q IFF P
T	T	T	T	T	T
T	F	F	F	T	F
F	T	T	F	F	F
F	F	T	T	T	T

Albert: This was a confusing way to set up the table caused by the notation being too long.

The underlined operators have the same column of truth values, proving that the corresponding formulas are equivalent.

1.3.3 Problems

Class Problems

Homework Problems

move this to location A10 as the last section of chapter 1

1.4 Satisfiability

A proposition is **satisfiable** if some setting of the variables makes the proposition true. For example, P AND $\overline{Q}$ is satisfiable because the expression is true when P is true and Q is false. On the other hand, P AND $\overline{P}$ is not satisfiable because the expression as a whole is false for both settings of P. But determining whether or not a more complicated proposition is satisfiable is not so easy. How about this one?

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

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 $\mathbb{T}$ ever appears. But this approach is not very efficient; a proposition with n variables has a truth table with 2^n lines, so the effort required to decide about 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!

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 satifiable or not? No one knows. And an awful lot hangs on the answer. 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.

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 NOT satisfiable, sat-solvers generally take exponential time to verify that.

So no one has a good idea how to solve SAT in polynomial time or else 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. 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.

1.4.1 Problems

Class Problems

1.5 Predicates and Quantifiers

1.5.1 Some More Propositions

A *prime* is an integer greater than one that is not divisible by any integer greater than 1 besides itself, for example, 2, 3, 5, 7, 11, ...

Proposition 1.5.1. *For every nonnegative integer, n, the value of $n^2 + n + 41$ is prime.*

Let's try some numerical experimentation to check this proposition. Let[1]

$$p(n) ::= n^2 + n + 41. \tag{1.3}$$

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, ..., $p(20) = 461$ which is prime. Hmmm, ... starts to look like a plausible claim. In fact we can keep checking through $n = 39$

[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.

INSERT B1

~~goes in location A~~

Most of the
~~The~~ examples of propositions that we have considered thus far have been nice in the sense that it has been relatively easy to determine if they are true or false. ~~At least once we have been~~ ~~Unfortunately~~ ~~We~~ ~~In some cases (e.g., with~~ ~~as~~ ~~so~~ → At worst, there were only a few cases to check in a truth table. ~~Unfortunately~~ Unfortunately, not all propositions are so ~~Thus~~ easy to check. ~~so some may seem~~ That is because some propositions may involve ~~very large~~ infinite a large or number of possible cases. For example, consider the following proposition involving prime numbers. (A prime is an integer greater than 1 that is divisible only by itself and 1. For example, 2, 3, 5, 7, and 11 are primes, but 4, 6 and 9 are not. A number greater than 1 that is ~~divisible by~~ not prime is said to be composite.)

INSERT B2 ~~go to section 4.5~~

It is not immediately clear whether this proposition is true or false. ~~[illegible]~~ In such circumstances, it is tempting to try to determine its veracity by ~~checking~~ computing the value of $1^{\uparrow}$ (footnote)

INSERT B3 ~~goes to location A6~~

for several values to n and then checking to see if they are ~~indeed~~ prime. If any of the computed values is not prime, then we will know that the proposition is false. If all the computed values are indeed prime, then we might be tempted to conclude that the proposition is true.

and confirm that $p(39) = 1601$ is prime. The proposition certainly does seem to be true!

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, and thus the proposition is false! 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.

By the way, propositions like this about *all* numbers or other things are so common that there is a special notation for it. With this notation, Proposition 1.5.1 would be

$$\forall n \in \mathbb{N}.\ p(n) \text{ is prime.} \tag{1.4}$$

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 of propositions that, at first, seem to be true but which turn out to be false.

Proposition 1.5.2. *$a^4 + b^4 + c^4 = d^4$ has no solution when a, b, c, d are positive integers.*

INSERT B4 goes to location A7

Although surprising, this example is not as ~~rare as~~ contrived or rare as you might suspect.

~~We will~~ As we will soon see, there are many examples of propositions that ~~are~~ seem to be true when you check a few cases, but which turn out to be false. The key to remember is that you

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 1.5.2 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; as follows:

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

Proposition 1.5.3. *$313(x^3 + y^3) = z^3$ 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 1.5.4. *Every map can be colored with 4 colors so that adjacent[2] regions have different colors.*

[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.

Insert BS ~~(goes in to section 4.8)~~

you may be wondering why anyone would care whether or not there is a solution to $313(x^3+y^3)=z^3$ where x, y and z are positive integers.

~~like In fact, solv~~

It turns out that find solutions to such equations is important in the field of elliptic curves, which turns out to be important to the ~~field~~ study of factoring large integers, which turns out (as we will see in chapter 4) to be important ~~to~~ in cracking commonly used cryptosystems, which is why mathematicians went to the effort to find the solution with thousands of digits.

Of course, not all propositions

that have infinitely many cases ~~to~~ to check turn out to be false. The following proposition (known as the "Four-Color Theorem") turns out to be true.

4 The proof of this proposition is difficult and took ~~many etc~~ over a century to perfect. Along the way, many incorrect proofs were proposed,

~~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 extremely 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 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 the colorability of several hundred special maps.[3]

[3] See `http://www.math.gatech.edu/~thomas/FC/fourcolor.html`

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.

~~Location A9~~ & put insert B6 here §

Proposition 1.5.5 (Goldbach). *Every even integer greater than 2 is the sum of two primes.*

while the preceding propositions are important in mathematics, computer scientists are often interested in propositions concerning the

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, to determine 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~~ text.

REMOVE - we may find somewhere else for it.

ISBN 0-691-11533-8.

Insert B6 (put in location A9)

In some cases, we do not know whether or not a proposition is true. For example, the following simple proposition (known as Goldbach's conjecture) has been heavily studied for over since 1742 but we still do not know if it is true. Of course, it has been checked by computer for many values of n, but as we have seen, that is not sufficient to conclude that it is true.

1.3.2

1.5.2 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,

"n 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:

$P(n)$ ::= "n is a perfect square"

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!**

1.3.3

1.5.3 Quantifiers

There are a couple of assertions commonly made about a predicate: that it is *sometimes* true and that it is *always* true. For example, the predicate

$$\text{“}x^2 \geq 0\text{”}$$

is always true when x is a real number. On the other hand, the predicate

$$\text{“}5x^2 - 7 = 0\text{”}$$

is only sometimes true; specifically, when $x = \pm\sqrt{7/5}$.

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

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

Sometimes True

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

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,"

or maybe

"you can solve *at least one* problem we come up with."

In the preceding example, the quantified phrase appears inside a larger if-then state-

ment. This is quite normal; quantified statements are themselves propositions and can be combined with and, or, implies, etc., just like any other proposition.

1.5.4 More Cryptic Notation

There are symbols to represent universal and existential quantification, just as there are symbols for "and" ($\wedge$), "implies" ($\longrightarrow$), and so forth. In particular, to say that a predicate, P, is true for all values of x in some set, D, one writes:

$$\forall x \in D.\ P(x)$$

The symbol $\forall$ is read "for all", so this whole expression is read "for all x in D, $P(x)$ is true". To say that a predicate $P(x)$ is true for at least one value of x in D, one writes:

$$\exists x \in D.\ P(x)$$

The backward-E, $\exists$, is read "there exists". So this expression would be read, "There exists an x in D such that $P(x)$ is true." The symbols $\forall$ and $\exists$ are always followed by a variable —usually with an indication of the set the variable ranges over —and

then a predicate, as in the two examples above.

As an example, 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

> "If you can solve any problem we come up with, then you get an A for the course."

can be written as follows:

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

or maybe

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

1.3.5

1.5.5 Mixing Quantifiers

Many mathematical statements involve several quantifiers. For example, *Goldbach's Conjecture* states:

> "Every even integer greater than 2 is the sum of two primes."

Let's write this more verbosely to make the use of quantification clearer:

> 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:

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

1.5.6 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

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 (1.6), 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 (1.6) into English and then appeal to the meaning, in English, of "for all" and "there exists" as justification. So this wasn't a proof, just an explanation that once you understand what (1.6) means, it becomes obvious.

In contrast to (1.6), the formula

$$\forall y \exists x.\ P(x,y) \text{ IMPLIES } \exists x \forall y.\ P(x,y). \tag{1.9}$$

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.

1.5.9 Problems

Class Problems

Homework Problems

Location A10
(the satisfiability section goes here.)

1.6 Problems

Albert will supply these and they will be organized according to sections.