Chapter 5

First-Order Logic

5.1 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 any case, notice that this quantified phrase appears inside a larger if-then statement. This is quite normal; quantified statements are themselves propositions and can be combined with and, or, implies, etc., just like any other proposition.

5.1.1 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

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

5.1.2 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:

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

5.1.3 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:

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

For example, it might be that every American shares the dream of owning their own home.

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

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

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 at all about work.

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

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

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

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

5.1.4 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:

$$(\text{NOT } \exists x.\ P(x)) \text{ IFF } \forall x.\ \text{NOT } P(x). \tag{5.1}$$

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

5.1.5 Validity

A propositional formula is called *valid* when it evaluates to **T** no matter what truth values are assigned to the individual propositional variables. For example, the propositional version of the Distributive Law is that P AND $(Q$ OR $R)$ is equivalent to $(P$ AND $Q)$ OR $(P$ AND $R)$. This is the same as saying that

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

is valid.

The same idea 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 (5.1).

Another useful example of a valid assertion is

$$\exists x \forall y.\ P(x,y) \text{ IMPLIES } \forall y \exists x.\ P(x,y). \tag{5.2}$$

Here's an explanation why this is valid:

> Let D be the domain for the variables and P_0 be some binary predicate[1] on D. We need to show that if
>
> $$\exists x \in D\ \forall y \in D.\ P_0(x,y) \tag{5.3}$$
>
> holds under this interpretation, then so does
>
> $$\forall y \in D\ \exists x \in D.\ P_0(x,y). \tag{5.4}$$
>
> So suppose (5.3) 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 (5.4) means, so we've proved that (5.4) holds under this interpretation, as required.

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 (5.2), 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 (5.2) 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 (5.2) means, it becomes obvious.

In contrast to (5.2), the formula

$$\forall y \exists x.\ P(x,y) \text{ IMPLIES } \exists x \forall y.\ P(x,y). \tag{5.5}$$

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

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

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.

5.1.6 Problems

Class Problems

Problem 5.1.
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)

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

(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 5.2.
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 $\mathtt{0}$, $\mathtt{1}$ denoting 0, 1.

A string like $\mathtt{01}x\mathtt{0}y$ 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 $\mathtt{011}$ and the value of y is $\mathtt{1111}$, then the value of $\mathtt{01}x\mathtt{0}y$ is the binary string $\mathtt{0101101111}$.

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	$\text{NOT}(\text{SUBSTRING}(\mathtt{1}, x))$	$\text{NO-1S}(x)$

(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:

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

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

Problem 5.3.
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.

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

Problem 5.4.
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 5.5.
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

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

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 5.6.
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."

5.2 The Logic of Sets

5.2.1 Russell's Paradox

Reasoning naively about sets turns out to be risky. In fact, one of the earliest attempts to come up with precise axioms for sets by a late nineteenth century logican named Gotlob *Frege* was shot down by a three line argument known as *Russell's Paradox*:[2] This was an astonishing blow to efforts to provide an axiomatic foundation for mathematics.

Let S be a variable ranging over all sets, and define

$$W ::= \{S \mid S \notin S\}.$$

So by definition,

$$S \in W \text{ iff } S \notin S,$$

for every set S. In particular, we can let S be W, and obtain the contradictory result that

$$W \in W \text{ iff } W \notin W.$$

A way out of the paradox was clear to Russell and others at the time: *it's unjustified to assume that W is a set*. So the step in the proof where we let S be W has no justification, because S ranges over sets, and W may not be a set. In fact, the paradox implies that W had better not be a set!

But denying that W is a set means we must *reject* the very natural axiom that every mathematically well-defined collection of elements is actually a set. So the problem faced by Frege, Russell and their colleagues was how to specify *which* well-defined collections are sets. Russell and his fellow Cambridge University colleague Whitehead immediately went to work on this problem. They spent a dozen years developing a huge new axiom system in an even huger monograph called *Principia Mathematica*.

5.2.2 The ZFC Axioms for Sets

It's generally agreed that, using some simple logical deduction rules, essentially all of mathematics can be derived from some axioms about sets called the Axioms of Zermelo-Frankel Set Theory with Choice (ZFC).

We're *not* going to be working with these axioms in this course, but we thought

[2]Bertrand *Russell* was a mathematician/logician at Cambridge University at the turn of the Twentieth Century. He reported that when he felt too old to do mathematics, he began to study and write about philosophy, and when he was no longer smart enough to do philosophy, he began writing about politics. He was jailed as a conscientious objector during World War I. For his extensive philosophical and political writing, he won a Nobel Prize for Literature.

you might like to see them –and while you're at it, get some practice reading quantified formulas:

Extensionality. Two sets are equal if they have the same members. In formal logical notation, this would be stated as:

$$(\forall z.\ (z \in x \text{ IFF } z \in y)) \text{ IMPLIES } x = y.$$

Pairing. For any two sets x and y, there is a set, $\{x, y\}$, with x and y as its only elements:

$$\forall x, y.\ \exists u.\ \forall z.\ [z \in u \text{ IFF } (z = x \text{ OR } z = y)]$$

Union. The union, u, of a collection, z, of sets is also a set:

$$\forall z.\ \exists u \forall x.\ (\exists y.\ x \in y \text{ AND } y \in z) \text{ IFF } x \in u.$$

Infinity. There is an infinite set. Specifically, there is a nonempty set, x, such that for any set $y \in x$, the set $\{y\}$ is also a member of x.

Power Set. All the subsets of a set form another set:

$$\forall x.\ \exists p.\ \forall u.\ u \subseteq x \text{ IFF } u \in p.$$

Replacement. Suppose a formula, ϕ, of set theory defines the graph of a function, that is,

$$\forall x, y, z.\ [\phi(x, y) \text{ AND } \phi(x, z)] \text{ IMPLIES } y = z.$$

Then the image of any set, s, under that function is also a set, t. Namely,

$$\forall s\ \exists t\ \forall y.\ [\exists x.\ \phi(x, y) \text{ IFF } y \in t].$$

Foundation. There cannot be an infinite sequence

$$\cdots \in x_n \in \cdots \in x_1 \in x_0$$

of sets each of which is a member of the previous one. This is equivalent to saying every nonempty set has a "member-minimal" element. Namely, define

$$\text{member-minimal}(m, x) ::= [m \in x \text{ AND } \forall y \in x.\ y \notin m].$$

Then the Foundation axiom is

$$\forall x.\ x \neq \emptyset \text{ IMPLIES } \exists m.\ \text{member-minimal}(m, x).$$

Choice. Given a set, s, whose members are nonempty sets no two of which have any element in common, then there is a set, c, consisting of exactly one element from each set in s.

5.2.3 Avoiding Russell's Paradox

These modern ZFC axioms for set theory are much simpler than the system Russell and Whitehead first came up with to avoid paradox. In fact, the ZFC axioms are as simple and intuitive as Frege's original axioms, with one technical addition: the Foundation axiom. Foundation captures the intuitive idea that sets must be built up from "simpler" sets in certain standard ways. And in particular, Foundation implies that no set is ever a member of itself. So the modern resolution of Russell's paradox goes as follows: since $S \notin S$ for all sets S, it follows that W, defined above, contains every set. This means W can't be a set —or it would be a member of itself.

5.2.4 Power sets are strictly bigger

It turns out that the ideas behind Russell's Paradox, which caused so much trouble for the early efforts to formulate Set Theory, lead to a correct and astonishing fact about infinite sets: they are *not all the same size*.

In particular,

Theorem 5.2.1. *For any set, A, the power set, $\mathcal{P}(A)$, is strictly bigger than A.*

Proof. First of all, $\mathcal{P}(A)$ is as big as A: for example, the partial function $f : \mathcal{P}(A) \to A$, where $f(\{a\}) ::= a$ for $a \in A$ and f is only defined on one-element sets, is a surjection.

To show that $\mathcal{P}(A)$ is strictly bigger than A, we have to show that if g is a function from A to $\mathcal{P}(A)$, then g is not a surjection. So, mimicking Russell's Paradox, define

$$A_g ::= \{a \in A \mid a \notin g(a)\}.$$

Now A_g is a well-defined subset of A, which means it is a member of $\mathcal{P}(A)$. But A_g can't be in the range of g, because if it were, we would have

$$A_g = g(a_0)$$

for some $a_0 \in A$, so by definition of A_g,

$$a \in g(a_0) \quad \text{iff} \quad a \in A_g \quad \text{iff} \quad a \notin g(a)$$

for all $a \in A$. Now letting $a = a_0$ yields the contradiction

$$a_0 \in g(a_0) \quad \text{iff} \quad a_0 \notin g(a_0).$$

So g is not a surjection, because there is an element in the power set of A, namely the set A_g, that is not in the range of g. ∎

Larger Infinities

There are lots of different sizes of infinite sets. For example, starting with the infinite set, $\mathbb{N}$, of nonnegative integers, we can build the infinite sequence of sets

$$\mathbb{N},\ \mathcal{P}(\mathbb{N}),\ \mathcal{P}(\mathcal{P}(\mathbb{N})),\ \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathbb{N}))),\ \ldots.$$

By Theorem 5.2.1, each of these sets is strictly bigger than all the preceding ones. But that's not all: the union of all the sets in the sequence is strictly bigger than each set in the sequence (see Problem 5.7). In this way you can keep going, building still bigger infinities.

So there is an endless variety of different size infinities.

5.2.5 Does All This Really Work?

So this is where mainstream mathematics stands today: there is a handful of ZFC axioms from which virtually everything else in mathematics can be logically derived. This sounds like a rosy situation, but there are several dark clouds, suggesting that the essence of truth in mathematics is not completely resolved.

- The ZFC axioms weren't etched in stone by God. Instead, they were mostly made up by some guy named Zermelo. Probably some days he forgot his house keys.

  So maybe Zermelo, just like Frege, didn't get his axioms right and will be shot down by some successor to Russell who will use his axioms to prove a proposition P and its negation NOT P. Then math would be broken. This sounds crazy, but after all, it has happened before.

  In fact, while there is broad agreement that the ZFC axioms are capable of proving all of standard mathematics, the axioms have some further consequences that sound paradoxical. For example, the Banach-Tarski Theorem says that, as a consequence of the Axiom of Choice, a solid ball can be divided into six pieces and then the pieces can be rigidly rearranged to give *two* solid balls, each the same size as the original!

- Georg *Cantor* was a contemporary of Frege and Russell who first developed the theory of infinite sizes (because he thought he needed it in his study of Fourier series). Cantor raised the question whether there is a set whose size is strictly between the "smallest[3]" infinite set, $\mathbb{N}$, and $\mathcal{P}(\mathbb{N})$; he guessed not:

  Cantor's *Continuum Hypothesis*: There is no set, A, such that $\mathcal{P}(\mathbb{N})$ is strictly bigger than A and A is strictly bigger than $\mathbb{N}$.

  The Continuum Hypothesis remains an open problem a century later. Its difficulty arises from one of the deepest results in modern Set Theory — discovered in part by Gödel in the 1930's and Paul Cohen in the 1960's — namely, the ZFC axioms are not sufficient to settle the Continuum Hypothesis: there are two collections of sets, each obeying the laws of ZFC, and in one collection the Continuum Hypothesis is true, and in the other it is false. So settling the Continuum Hypothesis requires a new understanding of what Sets should be to arrive at persuasive new axioms that extend ZFC and are strong enough to determine the truth of the Continuum Hypothesis one way or the other.

[3] See Problem 4.3

- But even if we use more or different axioms about sets, there are some unavoidable problems. In the 1930's, Gödel proved that, assuming that an axiom system like ZFC is consistent —meaning you can't prove both P and NOT P for any proposition, P —then the very proposition that the system is consistent (which is not too hard to express as a logical formula) cannot be proved in the system. In other words, no consistent system is strong enough to verify itself.

5.2.6 Large Infinities in Computer Science

If the romance of different size infinities and continuum hypotheses doesn't appeal to you, not knowing about them is not going to lower your professional abilities as a computer scientist. These abstract issues about infinite sets rarely come up in mainstream mathematics, and they don't come up at all in computer science, where the focus is generally on "countable," and often just finite, sets. In practice, only logicians and set theorists have to worry about collections that are too big to be sets. In fact, at the end of the 19th century, the general mathematical community doubted the relevance of what they called "Cantor's paradise" of unfamiliar sets of arbitrary infinite size.

But the proof that power sets are bigger gives the simplest form of what is known as a "diagonal argument." Diagonal arguments are used to prove many fundamental results about the limitations of computation, such as the undecidability of the Halting Problem for programs (see Problem 5.8) and the inherent, unavoidable, inefficiency (exponential time or worse) of procedures for other computational problems. So computer scientists do need to study diagonal arguments in order to understand the logical limits of computation.

5.2.7 Problems

Class Problems

Problem 5.7.
There are lots of different sizes of infinite sets. For example, starting with the infinite set, $\mathbb{N}$, of nonnegative integers, we can build the infinite sequence of sets

$$\mathbb{N},\ \mathcal{P}(\mathbb{N}),\ \mathcal{P}(\mathcal{P}(\mathbb{N})),\ \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathbb{N}))),\ \ldots.$$

By Theorem 5.2.1 from the Notes, each of these sets is *strictly bigger*[4] than all the preceding ones. But that's not all: if we let U be the union of the sequence of sets above, then U is strictly bigger than every set in the sequence! Prove this:

Lemma. *Let $\mathcal{P}^n(\mathbb{N})$ be the nth set in the sequence, and*

$$U ::= \bigcup_{n=0}^{\infty} \mathcal{P}^n(\mathbb{N}).$$

[4]Reminder: set A is *strictly bigger* than set B just means that A surj B, but NOT(B surj A).

Then

1. U surj $\mathcal{P}^n(\mathbb{N})$ *for every* $n \in \mathbb{N}$, *but*

2. *there is no* $n \in \mathbb{N}$ *for which* $\mathcal{P}^n(\mathbb{N})$ surj U.

Now of course, we could take $U, \mathcal{P}(U), \mathcal{P}(\mathcal{P}(U)), \ldots$ and can keep on indefinitely building still bigger infinities.

Problem 5.8.
Let's refer to a programming procedure (written in your favorite programming language —C++, or Java, or Python, ...) as a *string procedure* when it is applicable to data of type `string` and only returns values of type `boolean`. When a string procedure, P, applied to a `string`, s, returns **True**, we'll say that P *recognizes* s. If $\mathcal{R}$ is the set of strings that P recognizes, we'll call P a *recognizer* for $\mathcal{R}$.

(a) Describe how a recognizer would work for the set of strings containing only lower case Roman letter —`a`, `b`, `...`, `z` —such that each letter occurs twice in a row. For example, `aaccaabbzz`, is such a string, but `abb`, `00bb`, `AAbb`, and `a` are not. (Even better, actually write a recognizer procedure in your favorite programming language).

A set of `string`s is called *recognizable* if there is a recognizer procedure for it.

When you actually program a procedure, you have to type the program text into a computer system. This means that every procedure is described by some `string` of typed characters. If a `string`, s, is actually the typed description of some string procedure, let's refer to that procedure as P_s. You can think of P_s as the result of compiling s.[5]

In fact, it will be helpful to associate every string, s, with a procedure, P_s; we can do this by defining P_s to be some fixed string procedure —it doesn't matter which one —whenever s is not the typed description of an actual procedure that can be applied to `string` s. The result of this is that we have now defined a total function, f, mapping every `string`, s, to the set, $f(s)$, of `string`s recognized by P_s. That is we have a total function,

$$f : \texttt{string} \to \mathcal{P}(\texttt{string}). \tag{5.6}$$

(b) Explain why the actual range of f is the set of all recognizable sets of strings.

This is exactly the set up we need to apply the reasoning behind Russell's Paradox to define a set that is not in the range of f, that is, a set of strings, $\mathcal{N}$, that is *not* recognizable.

[5]The string, s, and the procedure, P_s, have to be distinguished to avoid a type error: you can't apply a string to string. For example, let s be the string that you wrote as your program to answer part (a). Applying s to a string argument, say `oorrmm`, should throw a type exception; what you need to do is apply the procedure P_s to `oorrmm`. This should result in a returned value **True**, since `oorrmm` consists of three pairs of lowercase roman letters

(c) Let

$$\mathcal{N} ::= \{s \in \texttt{string} \mid s \notin f(s)\}\,.$$

Prove that $\mathcal{N}$ is not recognizable.

Hint: Similar to Russell's paradox or the proof of Theorem 5.2.1.

(d) Discuss what the conclusion of part (c) implies about the possibility of writing "program analyzers" that take programs as inputs and analyze their behavior.

Problem 5.9.
Though it was a serious challenge for set theorists to overcome Russells' Paradox, the idea behind the paradox led to some important (and correct :-)) results in Logic and Computer Science.

To show how the idea applies, let's recall the formulas from Problem 5.2 that made assertions about binary strings. For example, one of the formulas in that problem was

$$\text{NOT}[\exists y\, \exists z.s = y1z] \qquad \textbf{(all-0s)}$$

This formula defines a property of a binary string, s, namely that s has no occurrence of a 1. In other words, s is a string of (zero or more) 0's. So we can say that this formula *describes* the set of strings of 0's.

More generally, when G is any formula that defines a string property, let ok-strings(G) be the set of all the strings that have this property. A set of binary strings that equals ok-strings(G) for some G is called a *describable* set of strings. So, for example, the set of all strings of 0's is describable because it equals ok-strings(**all-0s**).

Now let's shift gears for a moment and think about the fact that formula **all-0s** appears above. This happens because instructions for formatting the formula were generated by a computer text processor (in 6.042, we use the LaTeX text processing system), and then an image suitable for printing or display was constructed according to these instructions. Since everybody knows that data is stored in computer memory as binary strings, this means there must have been some binary string in computer memory —call it $t_{\textbf{all-0s}}$ —that enabled a computer to display formula **all-0s** once $t_{\textbf{all-0s}}$ was retrieved from memory.

In fact, it's not hard to find ways to represent *any* formula, G, by a corresponding binary word, t_G, that would allow a computer to reconstruct G from t_G. We needn't be concerned with how this reconstruction process works; all that matters for our purposes is that every formula, G, has a representation as binary string, t_G.

Now let

$$V ::= \{t_G \mid G \text{ defines a property of strings and } t_G \notin \text{ok-strings}(G)\}\,.$$

Use reasoning similar to Russell's paradox to show that V is not describable.

Homework Problems

Problem 5.10.
Let $[\mathbb{N} \to \{1,2,3\}]$ be the set of all sequences containing only the numbers 1, 2, and 3, for example,

$$(1,1,1,1...),$$
$$(2,2,2,2...),$$
$$(3,2,1,3...).$$

For any sequence, s, let $s[m]$ be its mth element.

Prove that $[\mathbb{N} \to \{1,2,3\}]$ is uncountable.

Hint: Suppose there was a list

$$\mathcal{L} = \text{sequence}_0, \text{sequence}_1, \text{sequence}_2, \ldots$$

of sequences in $[\mathbb{N} \to \{1,2,3\}]$ and show that there is a "diagonal" sequence diag $\in [\mathbb{N} \to \{1,2,3\}]$ that does not appear in the list. Namely,

$$\text{diag} ::= r(\text{sequence}_0[0]), r(\text{sequence}_1[1]), r(\text{sequence}_2[2]), \ldots,$$

where $r : \{1,2,3\} \to \{1,2,3\}$ is some function such that $r(i) \neq i$ for $i = 1,2,3$.

Problem 5.11.
For any sets, A, and B, let $[A \to B]$ be the set of total functions from A to B. Prove that if A is not empty and B has more than one element, then NOT$(A \text{ surj } [A \to B])$.

Hint: Suppose there is a function, σ, that maps each element $a \in A$ to a function $\sigma_a : A \to B$. Pick any two elements of B; call them 0 and 1. Then define

$$\text{diag}(a) ::= \begin{cases} 0 \text{ if } \sigma_a(a) = 1, \\ 1 \text{ otherwise.} \end{cases}$$

5.3 Glossary of Symbols

symbol	meaning
$::=$	is defined to be
$\wedge$	and
$\vee$	or
$\longrightarrow$	implies
$\neg$	not
$\neg P$	not P
$\overline{P}$	not P
$\longleftrightarrow$	iff
$\longleftrightarrow$	equivalent
$\oplus$	xor
$\exists$	exists
$\forall$	for all
$\in$	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, $\{\}$

6.042J / 18.062J Mathematics for Computer Science

Spring 2010