1 Quantifiers

To formulate more complex mathematical statements, we use the quantifiers *there exists*, written $\exists$, and *for all*, written $\forall$. If $P(x)$ is a predicate, then

- $\exists x : P(x)$ means, "There exists an x such that $P(x)$ holds."
- $\forall x : P(x)$ means, "For all x, it is the case that $P(x)$ holds."

So for example, if x denotes a real number, then

- $\exists x : x^2 = 4$ is true, since 2 is an x for which $x^2 = 4$. On the other hand, $\forall x : x^2 = 4$ is clearly false; not all numbers, when squared, are equal to 4.
- $\forall x : x^2 + 1 > 0$ is true, but $\forall x : x^2 > 2$ is false, since for example $x = 1$ doesn't satisfy the predicate. On the other hand, $\exists x : x^2 > 2$ is true, since $x = 2$ is an example that satisfies it.

Whenever we see a variable in a quantified expression, there's an underlying assumption that the variable comes from some base set. So to go back to $\forall x : x^2 + 1 > 0$, this is true because we specified that x was a real number. But it would be false if we specified that x was drawn from the complex numbers, since then $x = i$ would not satisfy the predicate. Often we'll keep the underlying set implicit, but it is important to be careful about this.

Negating quantified statements. Earlier we said that $\forall x : x^2 > 2$ is false, because we were able to think of an x ($x = 1$) that fails to satisfy the predicate. This suggests how to negate a $\forall$ statement: we flip $\forall$ to $\exists$, and then negate the predicate inside. That is,

- the negation of $\forall x : P(x)$ is $\exists x : \overline{P(x)}$.

This, incidentally, is where the term "counterexample" comes from. If $\forall x : P(x)$ is false, then $\exists x : \overline{P(x)}$ — and the x that exists to satisfy $\overline{P(x)}$ is the *counterexample* to the claim $\forall x : P(x)$.

On the other hand, to negate $\exists x : P(x)$, we must claim that $P(x)$ fails to hold for any possible x. So again we flip the quantifier and then negate the predicate:

- the negation of $\exists x : P(x)$ is $\forall x : \overline{P(x)}$.

Quantifiers in standard English usage. If we think about it, this is all familiar from standard English usage. For example, if someone says, "Everyone at Cornell is at least 18 years old," you might reply, "No, I know someone at Cornell who's under 18." What are you doing when you say this?

- At least subconsciously, you're interpreting this statement as " $\forall x$ at Cornell, x is at least 18 years old."
- To disagree with this, you're negating the statement by flipping the $\forall$ to $\exists$ and then negating the the predicate: " $\exists x$ at Cornell such that x is not at least 18 years old."
- Note also that we're careful about the set over which x is being quantified: the set is all people at Cornell.

The same thing happens in the reverse direction, from $\exists$ to $\forall$: if someone says, "There's an NBA player who makes over ten million dollars a year," you might disagree by saying, "No, every NBA player makes under ten million dollars a year."

1.1 Nested Quantifiers

Most serious mathematical statements use *nested* quantifiers. For example,

- Suppose we claimed, "For every real number, there's a real number larger than it." We'd write this as $\forall x\ \exists y : y > x$.
- Or if we wanted to claim, "There exists a Boolean formula such that every truth assignment to its variables satisfies it," we could write this as $\exists$ formula$F$ $\forall$ assignments$A$: A satisfies F.

The difference between a statement that says $\forall x\ \exists y$ and a statement that says $\exists x\ \forall y$ is something to watch out for. For example, if we're talking about real numbers, then our earlier example $\forall x\ \exists y : y > x$ is true. But writing it with the quantifiers in the other order, it would become false: $\exists y\ \forall x : y > x$. This version would require there to be a single number that's greater than every number.

Negating Nested Quantifiers. To negate a sequence of nested quantifiers, you flip each quantifier in the sequence and then negate the predicate. So the negation of $\forall x\ \exists y : P(x, y)$ is $\exists x\ \forall y : \overline{P(x, y)}$ and So the negation of $\exists x\ \forall y : P(x, y)$ and $\forall x\ \exists y : \overline{P(x, y)}$.

Again, after some thought, this make sense intuitively. For example, let's take the definition of an unbounded sequence from calculus. If we have an infinite sequence of real numbers $a_1 \le a_2 \le a_3 \le \cdots$, then we say it's *unbounded* if for every number x, it eventually grows larger than x. You can already see the quantifiers lurking in here: $\forall x\ \exists n : a_n > x$.

Now, some sequences, like $1, 4, 9, 16, 25, 36, \ldots$, are unbounded, and some, like $\frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \ldots$, are not. What does it mean for a sequence not to be unbounded: there is an upper bound x such that every number in the sequence is at most x.

In fact, we could have derived this mechanically by negating the definition of unboundedness. If "unbounded" means $\forall x\ \exists n : a_n > x$, then "not unbounded" must mean (flipping quantifiers) $\exists x\ \forall n : a_n \leq x$. Notice that this is what just said, but here we worked it out without even thinking about what the symbols mean.

Nested Quantifiers in Standard English Usage. People manipulate sequences of two nested quantifiers in conversation all the time. For example, if someone says, "Everyone experiences moments of doubt," you might reply, "No, I know someone who seems never to have experienced any moments of doubt in their whole life." What are you doing here?

- You're interpreting the statement as " $\forall$ people p $\exists$ time t : p experiences doubt at time t."

- To disagree with this, you're negating the statement by flipping each quantifier and then negating the the predicate: " $\exists$ person p $\forall$ times t : p did not experience doubt at time t." (It appears that you added "in their whole life" just for effect ...)

More than two nested quantifiers. There's no problem writing longer sequences of nested quantifiers, but it's a general rule of thumb that people really have to work, cognitively, to handle more than two. This is why most of us have such a hard time digesting the definition of a limit when we first learn it in calculus: $\lim_{x \to a} f(x) = b$ means

$$\forall \epsilon\ \exists \delta\ \forall x : (0 < |x - a| < \delta) \longrightarrow |f(x) - b| < \epsilon.$$

Given their complexity, it's interesting to ask how often sentences with three nested quantifiers come up in standard conversation.

There are certainly examples. Here's one that requires a very mild knowledge of baseball, in which all three quantifiers are hidden.

- In 1941, Joe Dimaggio had a 56-game hitting streak.

(A record, by the way, that many people think will never be broken ...)

Even if you don't know baseball, just think of the rules this way: over the course of one game, each player comes up to bat several times; in each at-bat, they either get a hit (a good thing) or an out (a bad thing). So here's what the sentence above is really saying:

- In 1941, there existed a sequence of 56 consecutive games, such that for all games in this sequence, there was at least one at-bat in which Joe Dimaggio got a hit.

So sports fans, just like mathematicians and software designers, know how to make up definitions (like "hitting streak") that hide a lot of complexity.

Notice by the way that "streaks" in sports don't generally involve three nested quantifiers; in fact, they usually involve just two. For example, the Kansas City Royals' 19-game losing streak this baseball season could be written

- This past summer, there was a sequence of 15 consecutive Royals games such that for all games in this sequence, the Royals lost.

1.2 Connections to Game-Playing

We used Boolean formulas to model two-player games in the previous lecture; we can also use quantifiers for this purpose, in a very similar way. Suppose we have a game in which players alternate moves, and we want to know whether Player 1 can force a win after at most k moves by each side. If we simply try stating what this means, we notice the quantifier structure very quickly: we want to know if there's a move for Player 1, so that for any possible move by Player 2, there's a move for Player 1, so that for any possible move by Player 2, and so forth, leading to a position that is a win for Player 1.

In symbols, let $x_1, x_2, \ldots, x_k$ be variables corresponding to the moves by Player 1, and let $y_1, y_2, \ldots, y_k$ be variables corresponding to the moves by Player 2. Let the predicate $P(x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$ be T when the position after moves $x_1, y_1, \ldots, x_k, y_k$ in order is a win for Player 1, and let is be F otherwise. Then the question of a forced win is determined simply by the truth or falsehood of a single quantified statement:

$$\exists x_1 \; \forall y_1 \; \exists x_2 \; \forall y_2 \cdots \; \exists x_k \; \forall y_k : P(x_1, y_1, \ldots, x_k, y_k.$$

And again, the strategy for Player 1 shows up here as well: every time a quantifier of the form $\exists x_i$ is satisfied, it says that there's a way to set the value of x_i (in other words, how to choose the right move) to continue forcing the win.

The relationship to quantifiers is so close that people often find it useful to think about the relationship as going in both directions: reasoning about alternating, nested quantifiers is like thinking about a strategy in a two-player game. And this also hints at why it's so hard to play games like chess well. Recall that even three nested quantifiers tends to place a large cognitive load on people; to look k moves out in chess requires manipulating $2k$ nested quantifiers! In later courses, like CS 482, you'll see actual formal evidence for why manipulating long strings of quantifiers is a computationally intractable problem.

1.3 Connections to Natural Language Processing

The analysis of quantifiers also shows up in the field of *natural language processing*, which the area of computer science concerned with getting computers to understand and communicate in human languages like English, Chinese, Arabic, and so forth. This is a very challenging problem that is far from being solved any time soon: the complexities of any given human language are much more subtle than its native speakers tend to realize. Or to put it another way, our brains our doing all sorts of things to process language, and it's hard to pin down precisely what these are. These issues are also core problems in the field of linguistics, of course, and there are many people who work on the boundary of natural language processing and linguistics.

Quantifiers in sentences are one of the linguistic constructs that are hard for computers to handle in general. Some simple examples illustrate why. To begin, there's ambiguity; to paraphrase an old Groucho Marx joke,

> In America, someone steals a car every fifteen seconds. We have to find that person and stop them.

There's no better way to ruin a joke than to try writing it out in logical notation, but for our particular purpose here it's useful, since it's entirely about quantifiers. Clearly the first sentence was meant to say: " $\forall$ blocks of fifteen seconds, $\exists$ a person who is stealing a car right then." The second sentence then willfully misinterprets this by swapping the quantifiers: "$\exists$ a person P such that $\forall$ blocks of fifteen seconds, P is stealing a car right then."

But there are much bigger problems than this, if we want to get computers to understand the quantifiers embedded in sentences A real difficulty is that we humans resolve the same ambiguities differently *depending on what the sentence is talking about.* Here's a nice pair of example dialogues from Lillian Lee that illustrate this.

1. "How was the party after I left?"

   "It was fun. Everyone had a drink."

2. "How was the party after I left?"

   "It was fun. Everyone saw a movie."

Did everyone have their own drink, or did they all share the same drink? Clearly everyone had their own drink. But did everyone watch a different movie (on little personal video players), or did they all sit around watching the same movie. Clearly everyone watched the same movie.

So the first is, " $\forall$ people P at the party $\exists$ a drink that P had," while the second is " $\exists$ a movie M such that $\forall$ people P, P saw the movie M." In other words, we interpret the order of quantifiers differently, even though the sentences are written exactly the same way. The differences in interpretation come about simply because we know about the differences between drinks and movies. This is the kind of thing that makes life very difficult for natural language processing software.

1.4 Quantifiers and Proofs

As we discussed earlier, our main interest in quantifiers for the purposes of this course is to manipulate mathematical statements in a careful way.

When faced with a mathematical claim, understanding its quantifier is often a very good strategy for thinking about how to work out a proof. For example:

- If the statement has the form $\forall x : P(x)$, then the global outline is likely have the form: Consider any possible x, and show that it satisfies the property $P(x)$.

- If the statement has the form $\exists x : P(x)$, then the global outline is different: One needs to specify a *particular* x, and then show it satisfies $P(x)$.

It's particularly useful to watch for the role that $\exists$ plays in proofs. When you come to a $\exists x$, it generally means: At this point, you have to describe an x that does what you want.

Quantifiers and Proofs by Contradiction. Of course, one can try looking for alternate strategies, and proof by contradiction is one useful example of this. If you're trying to prove $\exists x : P(x)$, and you don't see how to describe the x you need, you can try supposing it's false and looking for a contradiction. Supposing it's false, concretely, means assuming $\forall x : \overline{P(x)}$. You're now in a situation where you can assume that $\overline{P(x)}$ holds for all x, and start looking for a contradiction.

Analogously, if you don't see how to get started proving a statement like $\forall x : P(x)$, you can similarly negate it and start searching for a contradiction. This means you can start by assuming $\exists x : \overline{P(x)}$, which lets you assume the existence of a counterexample x for which $\overline{P(x)}$ holds.