Quantifiers

What Are They

You are already familiar with predicates. A predicate is a judgment about an element of a collection; whether its value is true or false (or something else), depends on the element. Quantifiers are used to express judgments about the whole collection of elements.

Example 1. “All men are mortal”. Here we say nothing about any particular man (and we say nothing about women), but what we say applies to any man, due to the word “all”. So from here we normally deduce that Justin Bieber, whoever this person may be, is mortal. I could also name another mortal guy whose name starts with ‘B’, but then I’d have to go into hiding, I guess.

Example 2. “Every ten minutes a man is mugged in the streets of New York”. Here we are talking about men in general again; but we cannot say for sure whether this or that man is or ever will be mugged in the streets of New York; this definitely does not apply to those who are not in New York. What’s more important, while this statement may be generally correct, we may never manage to find such a man for an interview.

These two examples mention the two quantifiers we are going to talk about, existential and universal.

Universal Quantifier

Universal quantifier, denoted as $\forall$, is used for judgments that state something about all elements of a collection. An extreme case would be when the statement is about the whole world (e.g. “nobody’s perfect”, or “every statement is a lie”).

We write it as $\forall x\ P(x)$; it is pronounced as “for all x, P(x)”.

If we look at things from computing point of view, we are either in a world with types or in a world without types. In a typeless world, we can only delimit the scope of a quantifier by specifying a collection. In a world with types we can delimit the scope by either specifying a collection, or specifying a type for each variable used in the quantifier.

In Java/Scala

In Scala, having a collection c (actually, TraversableLike) of values of type A, and a predicate p: A=>Boolean (that is, a function from A to Boolean), we can val itsPerfect = c forall p, which is true if and only if p(x) is true for all x in c.

In Java before version 8 there was no such thing; but you can always start using Bolts library, FunctionalJ, or Google Guava library; each of these can handle quantifiers.

Existential Quantifier

Similar to universal quantifier, *existential quantifier*, denoted as $\exists$, is used for judgments that state existence of an element in collection that satisfies a predicate. Again, an extreme case would be when the range of the statement is unreasonably wide (e.g. "there is somewhere a perfect female" (our cat knows such a female, it's herself), or "there exists a set that contains every other set as an element").

We write it as $\exists x\ P(x)$; it is pronounced as "there exists an x, such that P(x)"

From computing point of view, we again may have an option of limiting the range by either specifying a collection or a type.

In Java/Scala

In Scala, if we have a collection c (actually, TraversableLike) of values of type A, and a predicate p: A=>Boolean, we can val haveSome = c exists p, which is true if and only if p(x) is true for some x in c.

In Java you can use Bolts, FunctionalJ, or Google Guava to for this functionality.

Quantifiers and Logical Connectives

The predicate you have inside a quantifier may be a conjunction, a negation, a disjunction, or an implication; on the other hand, we can use a statement with quantifiers with an expression with such connectives; it must be interesting, are they related in any way? E.g. $\exists x(P(x) \wedge Q)$ - how is it related to $(\exists x\ P(x)) \wedge Q$?

To figure out each combination, we can delimit the range of the quantifier; instead of looking

through maybe a continuum of values and trying to figure out whether we can ever prove anything, let's take just two values in the range. If a statement does not hold for a choice of two, it won't hold for a bigger collection, in general settings.
Also, to be on the safe side, we have to check if our statement holds on an empty collection; we will see further why so.

Conjunction and Existential Quantifier

Can we check how $\exists x(P(x) \land Q(x))$ and $(\exists x\, P(x)) \land (\exists x\, Q(x))$ are related?

Suppose we have just $x0$ and $x1$.

The first statement says: $(P(x0) \land Q(x0)) \lor (P(x1) \land Q(x1))$

The second statement says: $(P(x0) \lor P(x1)) \land (Q(x0) \lor Q(x1))$

If we convert the second statement to disjunctive normal form, we will have

$(P(x0) \land Q(x0)) \lor (P(x0) \land Q(x1)) \lor (P(x1) \land Q(x0)) \lor (P(x1) \land Q(x1))$

You see that the second statement follows from the first, but not the first from the second. So now we have a good reason to suspect that

$\exists x(P(x) \land Q(x)) \vdash (\exists x\, P(x)) \land ((\exists x\, Q(x))$

But wait, what if our domain is empty? Then, obviously, both sides are false – there's no x to satisfy P or Q.

If the first one holds, we found an x such that both $P(x)$ and $Q(x)$ hold; abandoning the information about Q, we produce $(\exists x\, P(x))$; similarly, we have $(\exists x\, Q(x))$.

Conjunction and Universal Quantifier

This will be an exercise for the reader to enjoy.

Disjunction and Universal Quantifier

Compare two statements, $\forall x(P(x) \lor Q(x))$ and $(\forall x\, P(x)) \lor (\forall x\, Q(x))$.

If, again, we have just a range of two elements, x0 and x1, we can rewrite the statements as

$(P(x0) \lor Q(x0)) \land (P(x1) \lor Q(x1))$

and

$(P(x0) \land P(x1)) \lor (Q(x0) \land Q(x1))$

Converting the first one into disjunctive normal form, we get (reordered)

$(P(x0) \land P(x1)) \lor (Q(x0) \land Q(x1)) \lor (P(x0) \land Q(x1)) \lor (P(x1) \land Q(x0))$

As you see, the first one is part of the second, so the second follows from the first; so

$(\forall x\, P(x)) \lor (\forall x\, Q(x)) \vdash \forall x (P(x) \lor Q(x))$

We could argue differently, considering two cases, either $(\forall x\, P(x))$ or $(\forall x\, Q(x))$, and come to the same conclusion.

Negation and Universal Quantifier

What happens if we write $\forall x\, (\neg P(x))$? Is it related to $\forall x\, (P(x))$?

Let's start with the case of x0 and x1. The statement is equivalent to $\neg P(x0) \land \neg P(x1)$, which is the same as $\neg(P(x0) \lor P(x1))$, that is, $\neg(\exists x\, P(x))$.

The same holds for an arbitrary amount of values. In plain English it can be rephrased as "The statement <<for all x $\neg P(x)$>> holds if and only if there is no such x that P(x)".

Proving this fact in a general setting would require formal definitions for quantifiers; such definitions vary; and besides, we are not into proofs in this book.

Negation and Existential Quantifier

Now what about $\exists x\, (\neg P(x))$?

Again, take the case of just two elements, x0 and x1. In this case, the statement is equivalent to $\neg P(x0) \lor \neg P(x1)$, which is, in our Boolean logic, the same as $\neg(P(x0) \land P(x1))$, that is, $\neg(\forall x\, P(x))$.

This is interesting. If we say that there is an x that does not satisfy P, then we know that not every x satisfies P. But what if we go in the opposite direction, and from a general fact that not every x satisfies P try to deduce existence of such an x? (like in the example in the beginning, interview the man that is being mugged every 10 minutes)?

Here's an example. Let's start enumerating sets of integers. Of course some of them can be easily enumerated. But definitely not all of them, right? So there must exist a set of integers that cannot be enumerated. Which one is it?

The problem is that Booleanness and the implicitly used Axiom of Choice promise us that any question has an answer, without saying how to find the answer.

Non-Boolean logic usually does not make any such assumptions.

Combining Quantifiers

Quantifiers of the same kind commute: $\forall x\ \forall y\ P(x,y)$ is the same as $\forall y\ \forall x\ P(x,y)$, and $\exists x\ \exists y\ P(x,y)$) is the same as $\exists y\ \exists x\ P(x,y)$.

But how about $\forall x\ \exists y\ P(x,y)$ vs $\exists y\ \forall x\ P(x,y)$? Let's show an informal example. Say, $P(x, y)$ stands for "x is a Facebook friend of y". The first one, $\forall x\ \exists y\ P(x,y)$, says that everybody has a friend on Facebook; the second one, $\exists y\ \forall x\ P(x,y)$?, says that somebody is everybody's friend on Facebook. This is probably the invisible friend, or rather a brother, that Edward Snowden was talking about.