Chapter 10: The Logic of Quantifiers

First-order logic

The system of quantificational logic that we are studying is called "first-order logic" because of a restriction in what we can "quantify over." Our language, FOL, contains both individual constants (names) and predicates. The names stand for individuals and the predicates, we might say, stand for properties of those individuals. **In FOL, we quantify over individuals, but not over properties**.

That is, we can take the sentence Cube(b) ∧ Large(b) and obtain a quantified sentence by replacing the **individual constant** with a variable, and attaching a quantifier:

$$\exists x\ (\mathrm{Cube}(x) \land \mathrm{Large}(x))$$

This is a way of saying in FOL that something is both a cube and large. But we cannot similarly replace a **predicate** with a variable and still have an FOL sentence. For example, we cannot start with the sentence Student(max) ∧ Student(claire) and obtain:

$$\exists P\ (P(\mathrm{max}) \land P(\mathrm{claire}))$$

(which seems to say that Max and Claire have something in common), for this is not a sentence of FOL. In **second-order logic**, there are predicate variables as well as individual variables, and second-order quantifiers. But second-order logic is a lot more complicated than FOL, and does not have all of the same features. (For example, our system $\mathcal{F}$ for FOL is complete, but no there is no complete deductive system for second-order logic.) For more on second-order logic, see SecondOrder.pdf

§ 10.1 Tautologies and quantification

Not all cases of logical consequence are cases of tautological consequence. The following argument is valid:

| $\forall x\ \mathrm{Cube}(x)$
| $\underline{\forall x}\ \mathrm{Small}(x)$
| $\forall x\ (\mathrm{Cube}(x) \land \mathrm{Small}(x))$

but the conclusion is not a tautological consequence of the premises. The validity of the argument depends on the meaning of the universal quantifier ∀, and not just on the meaning of the connective ∧.

As *LPL* shows (p. 258), the validity here must depend on more than just the connective ∧, for the following argument is **not** valid:

| $\exists x\ \mathrm{Cube}(x)$
| $\underline{\exists x}\ \mathrm{Small}(x)$
| $\exists x\ (\mathrm{Cube}(x) \land \mathrm{Small}(x))$

Similarly, not all logical truths are tautologies. The following is an example of a logical truth that is not a tautology:

$$\exists x\ \mathrm{Cube}(x) \lor \exists x\ \neg\mathrm{Cube}(x)$$

This is a logical truth because in every world in which we evaluate an FOL sentence, there is at least one object. If a world has a cube in it, the left disjunct is true; otherwise, it contains an object that is not a cube, in which case the right disjunct is true. So the entire sentence is true in every world.

But the sentence is not a tautology, for the similar sentence:

$$\forall x \, \mathrm{Cube}(x) \lor \forall x \, \neg\mathrm{Cube}(x)$$

is clearly not a tautology, or even true in every world. But the two sentences are exactly alike in terms of their connectives.

A sentence containing quantifiers that **is** a tautology is this:

$$\forall x \, \mathrm{Cube}(x) \lor \neg\forall x \, \mathrm{Cube}(x)$$

which is just an instance of the tautologous form $A \lor \neg A$.

Truth-functional form

So we have seen that some logical truths are tautologies, and some are not. To be able to decide whether an FOL sentence that contains quantifiers is a tautology, we need to develop the notion of a sentence's **truth-functional form**.

The truth-functional form of a sentence is basically what Boole sees when it looks at the sentence. It's the structure that the sentence can be seen to have when all of its constituent quantified sentences are treated as if they were atomic. We don't "look inside" the general sentences—we just uniformly replace them with letters. We then replace any remaining atomic sentences with letters.

Example

$$\forall x \, \mathrm{Tet}(x) \rightarrow \neg\exists y \, (\mathrm{Cube}(y) \land \neg \, \mathrm{FrontOf}(b, y) \land \mathrm{Dodec}(b))$$

There are two constituent general sentences here:

$$\forall x \, \mathrm{Tet}(x)$$

$$\exists y \, (\mathrm{Cube}(y) \land \neg \, \mathrm{FrontOf}(b, y) \land \mathrm{Dodec}(b)).$$

So we replace the first general sentence with A and the second with B. The only remaining parts of the sentence are the connectives $\neg$ and $\rightarrow$. So the truth-functional form of the sentence is $A \rightarrow \neg B$.

Another way to put this is to say that from the perspective of truth-functional logic, this sentence is a conditional whose consequent is a negation. This is all Boole sees when it looks at this sentence.

Truth-functional form algorithm

LPL provides a simple mechanical procedure (or "algorithm") for producing the truth-functional form of a sentence. This is described on p. 261; you should study it and be sure you know how to apply it.

Here's a slightly different way of carrying out the procedure: If the sentence contains any quantifiers, start with those of largest scope. For each such quantifier, underline its entire scope (this will include the quantifier itself). Any quantifiers, connectives, or atomic sentences that are included in this scope should be ignored. Once all the quantified sentences have been underlined, underline any remaining atomic sentences, with each atomic sentence being separately underlined. Next, attach a sentence letter (i.e., a capital letter) to each underline, starting from the left and proceeding alphabetically. If any sentence is repeated, it should be given the same sentence letter each time.

Finally, after all the underlines have been assigned sentence letters, replace each underlined sentence with its corresponding letter, and keep any remaining connectives that have not been underlined. The result is the **truth-functional form** of the original sentence.

Example 1

$\forall x\ \text{Tet}(x) \rightarrow \neg\exists y\ (\text{Cube}(y) \land \neg\ \text{FrontOf}(b, y) \land \exists z\ \text{Dodec}(z))$

First, we underline:

$$\underline{\forall x\ \text{Tet}(x)} \rightarrow \neg\underline{\exists y\ (\text{Cube}(y) \land \neg\ \text{FrontOf}(b, y) \land \exists z\ \text{Dodec}(z))}$$

Then we attach sentence letters:

$$\underline{\forall x\ \text{Tet}(x)}_{A} \rightarrow \neg\underline{\exists y\ (\text{Cube}(y) \land \neg\ \text{FrontOf}(b, y) \land \exists z\ \text{Dodec}(z))}_{B}$$

Then we replace the underlined sentences with the letters:

$$A \rightarrow \neg B$$

This sentence is TT-possible, but not a tautology, and therefore so is our original sentence.

Example 2

$\exists x\ \text{Tet}(x) \rightarrow (\neg\exists y\ (\text{Cube}(y) \land \neg\ \text{FrontOf}(y, b)) \rightarrow \exists x\ \text{Tet}(x))$

$$\underline{\exists x\ \text{Tet}(x)} \rightarrow (\neg\underline{\exists y\ (\text{Cube}(y) \land \neg\ \text{FrontOf}(y, b))} \rightarrow \underline{\exists x\ \text{Tet}(x)})$$

$$\underline{\exists x\ \text{Tet}(x)}_{A} \rightarrow (\neg\underline{\exists y\ (\text{Cube}(y) \land \neg\ \text{FrontOf}(y, b))}_{B} \rightarrow \underline{\exists x\ \text{Tet}(x)}_{A})$$

$$A \rightarrow (\neg B \rightarrow A)$$

This sentence is a tautology, and therefore so is our original sentence.

Tautologies of FOL

A quantified sentence of FOL is said to be a *tautology* if and only if its truth-functional form is a tautology.

Note that the same procedure can be applied to arguments as well as to individual sentences. That is, we can apply it to any FOL argument to construct the truth-functional form of the argument, and hence to determine whether its conclusion is a *tautological* consequence of its premises. We'll call such valid arguments "truth-table valid," or TT-valid, for short.

Note that an argument may appear deceptively similar to a TT-valid argument even though it is not TT-valid. For example:

$\exists$x (Cube(x) $\rightarrow$ Small(x))
$\exists$x Cube(x)
$\exists$x Small(x)

This may look like *modus ponens* (**→ Elim**), but it is not. Its truth functional form is actually this:

A
B
C

So our original argument is not TT-valid. Indeed, it is not valid at all. (You can construct a Tarski World counterexample to it. If you're in doubt about what such world would look like, check these sentences in this world.)

§ 10.2 First-order validity and consequence

A logical truth is one that is true in all possible circumstances; a valid argument is one whose conclusion comes out true in every possible circumstance in which its premises all come out true.

In propositional logic, we were able to use truth-tables as a way of expressing more precisely the notion of "possible circumstances"—a possible circumstance was represented as a row on a truth-table.

But since there are valid arguments that are not TT-valid, and logical truths that are not tautologies, we need a way to make the idea of possible circumstances more precise that goes beyond what truth-tables provide.

That is, we need to provide a more precise account of what it is to be a *first-order logical truth*, a *first-order consequence*, or a *first-order equivalence*.

> **Terminological point**: we'll follow *LPL* in calling a first-order logical truth a *first-order validity*, or *FO validity*, for short.

The general idea is this:

> **First-order** validities (or consequences, or equivalences) are truths (or consequences, or equivalences) solely in virtue of the truth-functional connectives, the quantifiers, and the identity symbol.

This means that to determine whether a sentence is an FO validity (or an argument a case of FO consequence, or a pair of sentences FO equivalent) we **ignore** the meanings of the names and predicates they contain.

A convenient way of ignoring the meanings of names and predicates is just to **replace** them with nonsense predicates (e.g., the predicates Tove, Slithy, Outgrabe, Borogove, etc., borrowed from Lewis Carroll's poem *Jabberwocky*[1]).

[1] For the full text of this marvelous poem, see www.jabberwocky.com/carroll/jabber/jabberwocky.html

Thus, we can see that the logical truth ∀x SameSize(x, x) is **not** an FO validity because when we replace the predicate SameSize with the predicate Outgrabe, the resulting sentence, ∀x Outgrabe(x, x), cannot be guaranteed by logic to be true—its truth depends on the "meaning" of Outgrabe.

On the other hand, we can see that ∀x Cube(x) → Cube(b) is an FO validity because the "nonsense" sentence ∀x Tove(x) → Tove(b) is true no matter what Tove means.

Using "nonsense" predicates may be an illuminating device, but we need not resort to this. We can simply replace predicates with predicate letters (and names with individual constants) and consider these letters to be open to **interpretation** in any way we wish. (That is, we can take its individual constants to be names of any objects we like, and its predicate letters to stand for any properties we like.) This leads to the replacement method of pp. 270-71.

Replacement method

1. Replace all names with individual constants and all predicates with predicate letters (maintaining the arity, of course); if a predicate (or a name) is repeated, use the same letter to replace all of its occurrences.
2. To see whether a sentence is an FO validity, try to describe a circumstance, and an interpretation of the predicate letters and individual constants, in which the sentence is false. If there is none, the sentence is an FO validity.
3. To see whether S is an FO consequence of $P_1, \ldots, P_n$, try to describe a circumstance and an interpretation under which S is false and all of $P_1, \ldots, P_n$ are true. If there is none, S is an FO consequence of $P_1, \ldots, P_n$.

This method is used on the example on pp. 269-70. Study it carefully! (**Exercise**: can you provide a Tarski's World counterexample for the argument-form obtained by the replacement method on this example? You should be able to do this.)

Using the notion of *interpretation* that we have just introduced, we can define *FO validity* and *FO consequence* as follows:

- A sentence S is an FO validity iff it comes out true on every interpretation.
- A sentence S is an FO consequence of sentences $P_1, \ldots, P_n$ iff there is no interpretation under which all of $P_1, \ldots, P_n$ come out true and S comes out false.

To show that a sentence is not an FO validity, then, you need to provide an interpretation on which it comes out false. You can often use Tarski's World to do this, but sometimes Tarski's World will not be able to provide the required interpretation. We will be looking at examples of this in subsequent chapters.

Summary

1. If S is a tautology, then S is an FO validity (but not conversely). And if S is an FO validity, then S is a logical truth (but not conversely).
2. If S is a tautological consequence of premises $P_1, \ldots, P_n$, then S is an FO consequence of $P_1, \ldots, P_n$ (but not conversely). And if S is an FO consequence of $P_1, \ldots, P_n$, then S is a logical consequence of $P_1, \ldots, P_n$ (but not conversely).

The Euler diagram on p. 272 depicts these relationships. **Study it carefully.**

§ 10.3 First-order equivalence and DeMorgan's laws

The two sentences:

1. ¬(∃x Cube(x) ∧ ∀y Dodec(y))
2. ¬∃x Cube(x) ∨ ¬∀y Dodec(y))

are tautologically equivalent. Indeed, their equivalence is an instance of DeMorgan's laws.

The two sentences:

3. ∃x ¬(Cube(x) ∧ Large(x))
4. ∃x (¬Cube(x) ∨ ¬Large(x))

are also equivalent, but they are not tautologically equivalent. (Apply the truth-functional form algorithm to this pair if that point is not clear.)

The difference is that in (1) and (2), DeMorgan's Laws are applied to a pair of sentences, whereas in (3) and (4), we appear to be applying DeMorgan's Laws to a pair of wffs that are not sentences.

But how can we say that ¬(Cube(x) ∧ Large(x)) and ¬Cube(x) ∨ ¬Large(x) are equivalent, when they are not even sentences? We need to extend the notion of equivalence to wffs containing free variables.

Logically equivalent wffs

Here is our definition of logically equivalent wffs with free variables:

> A pair of wffs with free variables are logically equivalent if, in any possible circumstance, they are satisfied by the same objects.

And it is easy to see that our two wffs above satisfy this condition. The objects satisfying ¬(Cube(x) ∧ Large(x)) are those that are not large cubes; and the ones satisfying ¬Cube(x) ∨ ¬Large(x) are those that are either not cubes or not large, i.e., those that are not large cubes.

Note that if, within a given sentence, we substitute one logically equivalent wff for another, the resulting sentence will be equivalent to the original. Hence, (3) and (4) are equivalent because (4) can be obtained from (3) by replacing one component wff with another equivalent wff.

> **Caveat**: in the definition above of equivalence for wffs with free variables, we are assuming that the two wffs contain the **same** free variable (e.g., they both have x free, or both have y free, etc.). Otherwise, we would confront the problem that our definition would count Cube(x) as equivalent to ¬¬Cube(y). But it is not so clear that we would want to do this, given that we don't normally require different variables to pick out the same object. And if we allow x and y to pick out different objects, the biconditional Cube(x) ↔ ¬¬Cube(y) might not always come out true. And how can there be an equivalence whose corresponding biconditional can come out false?

DeMorgan laws for quantifiers

Note the connection between ∀ and ∧: in a world of four objects, *a*, *b*, *c*, and *d*, the two sentences

$$\mathrm{Cube}(a) \wedge \mathrm{Cube}(b) \wedge \mathrm{Cube}(c) \wedge \mathrm{Cube}(d)$$

$$\forall x\ \mathrm{Cube}(x)$$

will always agree in truth-value. We have a similar connection between $\exists$ and $\vee$: in a world like the one above, the two sentences

$$\mathrm{Cube}(a) \vee \mathrm{Cube}(b) \vee \mathrm{Cube}(c) \vee \mathrm{Cube}(d)$$

$$\exists x\ \mathrm{Cube}(x)$$

will always agree in truth-value.

So we would expect there to be first-order equivalences for the quantifiers that are counterparts to the DeMorgan equivalences of propositional logic. And indeed there are. Just as these sentences are equivalent:

$$\neg(\mathrm{Cube}(a) \wedge \mathrm{Cube}(b) \wedge \mathrm{Cube}(c) \wedge \mathrm{Cube}(d))$$

$$\neg\mathrm{Cube}(a) \vee \neg\mathrm{Cube}(b) \vee \neg\mathrm{Cube}(c) \vee \neg\mathrm{Cube}(d)$$

So are these:

$$\neg\forall x\ \mathrm{Cube}(x)$$

$$\exists x\ \neg\mathrm{Cube}(x)$$

Hence, we can state the **DeMorgan laws for quantifiers** (also known as the quantifier/negation equivalences):

$$\boxed{\begin{array}{l} \neg\forall x\ P(x) \ \Leftrightarrow\ \exists x\ \neg P(x) \\[1em] \neg\exists x\ P(x) \ \Leftrightarrow\ \forall x\ \neg P(x) \end{array}}$$

Combining laws and equivalences

We can combine the DeMorgan laws for quantifiers and various other equivalent wffs to set up some illuminating **chains of equivalences**.

¬A is equivalent to O

$$\begin{array}{lll} \neg\forall x\ (P(x) \rightarrow Q(x)) & \Leftrightarrow & \neg\forall x\ (\neg P(x) \vee Q(x)) \\ & \Leftrightarrow & \exists x\ \neg(\neg P(x) \vee Q(x)) \\ & \Leftrightarrow & \exists x\ (\neg\neg P(x) \wedge \neg Q(x)) \\ & \Leftrightarrow & \exists x\ (P(x) \wedge \neg Q(x)) \end{array}$$

¬I is equivalent to E

$$\begin{array}{lll} \neg\exists x\ (P(x) \wedge Q(x)) & \Leftrightarrow & \forall x\ \neg(P(x) \wedge Q(x)) \\ & \Leftrightarrow & \forall x\ (\neg P(x) \vee \neg Q(x)) \\ & \Leftrightarrow & \forall x\ (P(x) \rightarrow \neg Q(x)) \end{array}$$

This last chain shows, in effect, that the two FOL forms of **E** are equivalent.

§ 10.4 Other quantifier equivalences and non-equivalences

There are a number of other important quantifier equivalences to be aware of. There are also some important "pseudo-equivalences" to be wary of—non-equivalences that appear deceptively like equivalences. We list both kinds here.

Distributing ∀ through ∧

$$\forall x\,(P(x) \land Q(x)) \quad \Leftrightarrow \quad \forall x\, P(x) \land \forall x\, Q(x)$$

Distributing ∃ through ∨

$$\exists x\,(P(x) \lor Q(x)) \quad \Leftrightarrow \quad \exists x\, P(x) \lor \exists x\, Q(x)$$

Non-equivalences to beware of

Beware of the following non-equivalences:

$$\forall x\,(P(x) \lor Q(x)) \quad \not\Leftrightarrow \quad \forall x\, P(x) \lor \forall x\, Q(x)$$

$$\exists x\,(P(x) \land Q(x)) \quad \not\Leftrightarrow \quad \exists x\, P(x) \land \exists x\, Q(x)$$

Notice that you **can** distribute ∀ through ∧, and you can distribute ∃ through ∨, but you **cannot** distribute ∀ through ∨ or ∃ through ∧. If you are in any doubt about these last two non-equivalences, try problems 10.24 and 10.27. Be sure you understand why the non-equivalent pairs are not equivalent.

Null quantification

In the following examples, P represents any wff in which x does not occur free.

$$\forall x\, P \quad \Leftrightarrow \quad P$$

$$\exists x\, P \quad \Leftrightarrow \quad P$$

$$\forall x\,(P \lor Q(x)) \quad \Leftrightarrow \quad P \lor \forall x\, Q(x)$$

$$\exists x\,(P \land Q(x)) \quad \Leftrightarrow \quad P \land \exists x\, Q(x)$$

The last two might be thought of as providing "limited" distribution of ∀ through ∨ and ∃ through ∧. (For an example of the last one, see problem 10.28) The next four "null quantification over →" equivalences are not discussed in *LPL*, although they are listed in some exercises on p. 315. The third and fourth equivalences are tricky—they appear not to be equivalent—so study them carefully.

Null quantification over $\rightarrow$

$\forall x\ (P \rightarrow Q(x))$	$\Leftrightarrow$	$P \rightarrow \forall x\ Q(x)$
$\exists x\ (P \rightarrow Q(x))$	$\Leftrightarrow$	$P \rightarrow \exists x\ Q(x)$
$\forall x\ (Q(x) \rightarrow P)$	$\Leftrightarrow$	$\exists x\ Q(x) \rightarrow P$
$\exists x\ (Q(x) \rightarrow P)$	$\Leftrightarrow$	$\forall x\ Q(x) \rightarrow P$

More non-equivalences to beware of

$$\forall x\ (Q(x) \rightarrow P) \quad \not\Leftrightarrow \quad \forall x\ Q(x) \rightarrow P$$

$$\exists x\ (Q(x) \rightarrow P) \quad \not\Leftrightarrow \quad \exists x\ Q(x) \rightarrow P$$

These last two "pseudo-equivalences" are easy to miss—the parentheses indicate the crucial differences in the scope of the quantifiers.

Replacing bound variables

In the next examples, $P(x)$ is any wff and y is any variable that does not occur in $P(x)$:

$\forall x\ P(x)$	$\Leftrightarrow$	$\forall y\ P(y)$
$\exists x\ P(x)$	$\Leftrightarrow$	$\exists y\ P(y)$

What these equivalences tell you, in effect, is that it does not matter which variable you use in a universal or existential generalization. Systematically rewriting the bound variables does not change the meaning of the sentence.

Exercises with chains of equivalence

Two of the more puzzling equivalence claims we encountered above were the last two **null quantification over $\rightarrow$** equivalences:

$$\forall x\ (Q(x) \rightarrow P) \quad \Leftrightarrow \quad \exists x\ Q(x) \rightarrow P$$

$$\exists x\ (Q(x) \rightarrow P) \quad \Leftrightarrow \quad \forall x\ Q(x) \rightarrow P$$

To convince yourself that these two equivalence claims are correct, construct for each of them a chain of equivalences that establishes its correctness, making use of **other** equivalence claims that seem more intuitively obvious. Model your chains on those constructed above in §10.3.

In your chains you should make use of the following equivalences (which I hope are familiar by now): null quantification over $\vee$, DeMorgan's laws for quantifiers, definition of $\rightarrow$ in terms of $\vee$ and $\neg$, and equivalence of $P \vee Q$ and $Q \vee P$. Here is what these two equivalence chains will look like; just fill in the missing steps:

$$\begin{aligned} \forall x\ (Q(x) \rightarrow P) \quad &\Leftrightarrow \quad ? \\ &\Leftrightarrow \quad ? \\ &\Leftrightarrow \quad ? \end{aligned}$$

	$\Leftrightarrow$	$\exists x\, Q(x) \rightarrow P$
$\exists x\, (Q(x) \rightarrow P)$	$\Leftrightarrow$	?
	$\Leftrightarrow$	?
	$\Leftrightarrow$	?
	$\Leftrightarrow$	$\forall x\, Q(x) \rightarrow P$