Chapter 9: Introduction to Quantification

§ 9.1 Variables and atomic wffs

Variables behave **syntactically** like names—they appear in sentences in the same places that names appear. So all of the following count as correct atomic expressions of FOL:

Cube(d)	FrontOf(a, b)	Adjoins(c, e)	
Cube(x)	FrontOf(x, y)	Adjoins(c, x)	Adjoins(y, e)

These are all **well formed formulas (wffs)** of FOL. In fact, they are all atomic wffs. But the ones with variables in them (in these examples, the ones in the second row, containing x and y) are **semantically** different. For unlike the ones in the first row (whose individual symbols are restricted to names), the ones with variables in them do not make determinate statements, and hence do not have truth-values.

All of the expressions above are **wffs**; but only those in the top row are **sentences**.

FOL contains an infinite supply of variables: t, u, v, w, x, y, z, t_1, u_1, etc. Fitch understands all of these, but Tarski's World is restricted to these six: u, v, w, x, y, z.

§ 9.2 The quantifier symbols: ∀, ∃

The quantifier symbols, ∀ and ∃, are used with variables and wffs to create FOL sentences.

Universal quantifier (∀)

∀x is read "for every object x." Thus, "Every object is a cube" would be expressed in FOL as ∀x Cube(x). Some other obvious translations:

English	FOL
Everything is either a cube or a tetrahedron.	∀x (Cube(x) ∨ Tet(x))
Every tetrahedron is small.	∀x (Tet(x) → Small(x))

Existential quantifier (∃)

∃x is read "for at least one object x." Thus, "At least object is a tetrahedron" would be expressed in FOL as ∃x Tet(x). Some other obvious translations:

English	FOL
Some tetrahedron is small.	∃x (Tet(x) ∧ Small(x))
There is at least one cube in front of *b*.	∃x (Cube(x) ∧ FrontOf(x, b))

Pay particular attention to the two "small tetrahedron" sentences:

Every tetrahedron is small.	∀x (Tet(x) → Small(x))
Some tetrahedron is small.	∃x (Tet(x) ∧ Small(x))

In English, the only difference between them is that one contains *every* where the other contains *some*. So one might suppose that in FOL, the only difference between them would be that one contains ∀ where the other contains ∃. But this is not the case, as you can see. The universally quantified sentence contains a → where the existentially quantified sentence contains a ∧. We will spend some time later getting clear exactly why this is so.

§ 9.3 Wffs and sentences

In the portion of FOL we have studied up until now (the logic of sentences, or "propositional logic"), all sentences are built up out of atomic sentences, truth-functional connectives, and parentheses. In quantificational logic, we still have all of these sentences, but we have a lot more. For we can now form sentences out of parts that are neither sentences nor connectives, namely, out of wffs that are not sentences. That is, the parts will include wffs that contain variables.

What we need to do is to give the rules of the **syntax** of FOL. We will approach this in two stages. First, we'll describe the rules for constructing the wffs; then we will state the rules for determining which of the wffs are sentences.

Wffs

We begin with the notion of an **atomic wff**: any *n*-ary predicate followed by *n* individual symbols. (An individual symbol is either an individual constant or a variable.) Atomic wffs are the "building blocks" of FOL.

The examples we looked at earlier are all atomic wffs:

Cube(d)	FrontOf(a, b)	Adjoins(c, e)	
Cube(x)	FrontOf(x, y)	Adjoins(c, x)	Adjoins(y, e)

Any variable occurring in an atomic wff is *free* (*unbound*). Thus, there are free variables (x and y) in the atomic wffs in the second row, and no variables in the atomic wffs in the first row.

We can now give the rules for constructing more complex wffs out of atomic wffs, connectives, parentheses, and quantifiers:

1. If P is a wff, so is ¬P.
2. If $P_1, \ldots, P_n$ are wffs, so is $(P_1 \land \ldots \land P_n)$.
3. If $P_1, \ldots, P_n$ are wffs, so is $(P_1 \lor \ldots \lor P_n)$.
4. If P and Q are wffs, so is (P → Q).
5. If P and Q are wffs, so is (P ↔ Q).
6. If P is a wff and ν is a variable, then ∀ν P is a wff, and any occurrence of ν in ∀ν P is said to be bound.
7. If P is a wff and ν is a variable, then ∃ν P is a wff, and any occurrence of ν in ∃ν P is said to be bound.

Examples

Cube(x) and Dodec(y) are both atomic wffs, so (Cube(x) ∧Dodec(y)) is a wff (by clause 2).

Since (Cube(x) ∧Dodec(y)) is a wff, so is ¬(Cube(x) ∧Dodec(y)) (by clause 1).

Since Adjoins(x, y) is a wff, so is ∃x Adjoins(x, y) (by clause 7).

Since ¬(Cube(x) ∧Dodec(y)) and ∃x Adjoins(x, y) are both wffs, so is (¬(Cube(x) ∧ Dodec(y)) → ∃x Adjoins(x, y)) (by clause 4).

And so on. Note that in our last wff above, the occurrence of x in the antecedent is free, while both occurrences of x in the consequent are bound. All occurrences of y are free.

Sentences

A sentence is a wff in which no variable has a free occurrence. So, to convert our wff above into a sentence, we will have to do something to its free variables.

Bind with a quantifier

One way to convert our wff to a sentence is to attach quantifiers to bind the variables. We would do this in two stages:

∀y(¬(Cube(x) ∧ Dodec(y)) → ∃x Adjoins(x, y))

This takes care of y, as all three of its occurrences are now bound. But the leftmost occurrence of x is still free. So we can attach another quantifier, this one containing an x. Note that it will bind only the leftmost x; the ones in the consequent are already bound, and so are not bindable by the new quantifier.

∀x∀y(¬(Cube(x) ∧Dodec(y)) → ∃x Adjoins(x, y))

There are no free variables in this wff, and so it is a sentence. (We are not worried right now about what this sentence **means**. We are only trying to see what makes it a sentence.)

Substitution

Another way to convert a wff to a sentence is to replace the free variables it contains with constants. Starting with:

(¬(Cube(x) ∧ Dodec(y)) → ∃x Adjoins(x, y))

we replace **both** occurrences of y with the same constant (in this case a)—replacement must be uniform. As for x, we do not replace its occurrences in the consequent, because they are not free; we replace only the occurrence in the antecedent. We can replace that occurrence of x with any constant we like (including a). Or, we can use a different constant:

(¬(Cube(b) ∧ Dodec(a)) → ∃x Adjoins(x, a))

There are no free variables in this wff, and so it is a sentence.

You can confirm that these are sentences in Tarski's World. Open the file Ch9Ex1.sen from the Supplementary Exercises page of the course web site:

1. ∀x∀y(¬(Cube(x) ∧Dodec(y)) → ∃x Adjoins(x, y))
2. (¬(Cube(b) ∧Dodec(a)) → ∃x Adjoins(x, a))

Then try to verify the two sentences (above) that it contains. You will find that both are sentences, but neither is evaluable. (1) is not evaluable because the world is empty, and no sentence is evaluable in an empty world. As soon as you put a block into the world, (1) will be evaluable (it will come out false if all you do is to put in one block. Even when you add a second block, (1) will remain false unless your two blocks adjoin one another).

Note that (2) remains unevaluable. It cannot be evaluated until the names it contains are assigned to objects in the world. (Note this disparity in FOL between names and predicates: the predicates can be "empty", but the names cannot.) As soon as you assign the names a and b to blocks, (2) will evaluate as true or false.

Notice, also, that, for convenience, I have omitted the outermost pair of parentheses on (2). It is **always** permissible to omit the outermost pair of parentheses. Just don't forget to put them back on if you are embedding the sentence in a larger context (e.g., negating it, or making it a component of a compound sentence, or attaching a quantifier to it). We will turn next to an example of what can happen if you are not careful about this.

Scope of quantifier

Pay careful attention to the example discussed on p. 233. Parentheses are important in indicating the **scope** of a quantifier, that is, which part of the sentence contains occurrences of variables bindable by that quantifier.

So we must distinguish between these two wffs:

∃x (Doctor(x) ∧ Smart(x))

and

∃x Doctor(x) ∧ Smart(x)

The first is a sentence (it says, roughly, "some doctor is smart"); the second is not a sentence, since the x in Smart(x) is free. (This wff says, roughly, "There are doctors, and *x* is smart.")

It's easy to make the mistake of writing the second wff when you intend the first sentence. Here's how it might happen:

> You start with the atomic sentences Doctor(x) and Smart(x). You then conjoin them and get (Doctor(x) ∧ Smart(x)). You decide to drop the outer parentheses for convenience, and get the perfectly acceptable Doctor(x) ∧ Smart(x). Then, when you attach the quantifier, you forget to put the missing parentheses back. So instead of the intended sentence ∃x (Doctor(x) ∧ Smart(x)) you get the mistaken wff ∃x Doctor(x) ∧ Smart(x). **Be careful!**

§ 9.4 Semantics for the quantifiers

Satisfaction

Wffs containing free variables don't have truth-values—they are not true or false. Consequently, a quantified sentence that is built of such wffs, such as ∃x Cube(x), cannot have its truth-value defined in terms of the truth-value of its component wff, Cube(x), since that atomic wff does not have a truth-value.

Wffs containing free variables, although not true or false *simpliciter*, nevertheless can be said to be true or false **of** things. The wff Cube(x) is true of each cube, and false of every other thing. The wff Tet(x) ∧ Small(x) is true of each small tetrahedron, and false of every other thing. Another way to put this is to say that each cube *satisfies* Cube(x) and each small tetrahedron *satisfies* Tet(x) ∧ Small(x).

Satisfaction, then, is a relation between an object and a wff with a free variable.

[We are simplifying for ease of comprehension. Strictly speaking, we should say that satisfaction is a relation between an ordered *n-tuple* of objects and a wff with *n* free variables. For example, consider a wff with two free variables, such as Larger(x, y). Which objects stand in the **satisfaction-relation** to this wff? No object taken by itself does so; rather, it is **pairs** of objects that satisfy this wff. Thus, if *a* is a small cube and *b* is a large tetrahedron, then the pair of objects *b* and *a*, taken in that order—<*b*, *a*> is how we write this—satisfies the wff Larger(x, y). Note that <*a*, *b*> does not satisfy this wff, since *a* is not larger than *b*.]

We can state what it is for an object to satisfy a wff in terms of the truth of a certain sentence. For example, if S(x) is a wff containing one free variable, then a given object satisfies S(x) iff we get a true sentence when we replace every free occurrence of x in S(x) with the name of that object.

For example, an object named b satisfies Cube(x) ∧ Adjoins(x, a) iff the sentence Cube(b) ∧ Adjoins(b, a) is true.

But not every object has a name. (In many of the worlds in Tarski's World, lots of objects are nameless.) How do we explain what it is for a *nameless* object to satisfy a wff? We assign the object a *temporary* name and proceed as we did above for named objects.

Tarski's World reserves a number of individual constants, n_1, n_2, n_3, …etc. , for just this purpose. If we want to know whether a given nameless object satisfies a wff, we temporarily give it a name, choosing as its name the first of these constants not already in use. Suppose n_2 is the first such constant. Then, using n_2 as a name for our nameless object, that object satisfies S(x) iff we get a true sentence when we replace every free occurrence of x in S(x) with n_2.

For example, a nameless object satisfies Cube(x) ∧ Adjoins(x, a) iff, treating n_2 as the name of that object, the following sentence is true: Cube(n_2) ∧ Adjoins(n_2, a)

Semantics of ∃

A sentence of the form ∃x S(x) is true iff there is at least one object satisfying S(x).

Example: ∃x (Cube(x) ∧ Small(x)) is true iff there is at least one object satisfying Cube(x) ∧ Small(x), i.e., iff there is at least one small cube.

Semantics of ∀

A sentence of the form ∀x S(x) is true iff every object satisfies S(x).

Example: ∀x (Cube(x) → Small(x)) is true iff every object satisfies Cube(x) → Small(x), i.e., iff every object satisfying Cube(x) also satisfies Small(x), i.e., iff every cube is small.

Domain of discourse

The domain of discourse is the entire collection of things that we take our FOL sentences to be "about"—the things we allow our quantifiers to "range over" or pick out. Sometimes, the domain is unrestricted, in which case we are talking about everything, and our quantifiers range over all objects. More often, the domain is restricted in some way (restricted to a smaller collection of objects—people, numbers, politicians, elementary particles, etc.). The choice of domains affects how we read the quantifiers and quantified sentences. But in any case, the domain **must be non-empty**.

Examples

In the domain of *persons*, we read $\forall x$ as 'for every person, *x* …'.

In the domain of *numbers*, we read $\exists x$ as 'there is at least one number *x* such that …'.

In the domain of *politicians*, we read $\forall y$ as 'for every politician, *y* …'.

In Tarski's World, the domain is restricted to *blocks*. Hence, in sentences about a Tarski World, we read $\forall x$ as 'for every block, *x* …' and $\exists x$ as 'for at least one block, *x* …'.

If the domain is unrestricted, then $\forall x$ is read as 'for everything, *x* …' and $\exists x$ is read as 'there is at least one thing *x* such that …'. When a domain has not been specified, it will be assumed to be unrestricted.

A difference in domain is reflected in a difference in the way we translate sentences from English to FOL, and vice versa:

In the domain of *numbers*, we could translate *Some numbers are even* as $\exists x\ \text{Even}(x)$. But in an unrestricted domain, we'd have to write $\exists x\ (\text{Number}(x) \land \text{Even}(x))$.

Similarly, in the domain of *politicians*, we could translate *All politicians are crooks* as $\forall x\ \text{Crook}(x)$. But in an unrestricted domain, we'd have to write $\forall x\ (\text{Politician}(x) \rightarrow \text{Crook}(x))$.

Obviously, the advantage of a restricted domain is that it makes translation easier. The drawback is that once the domain has been restricted, your sentences cannot talk about anything outside of the restricted domain.

Hence, a sentence like *Every person owns a pet* cannot be translated adequately into an FOL whose domain has been restricted to *persons*, since this sentence requires us to quantify over *pets*, and pets are not persons (at least, many pets are not persons!).

A notational convention

In stating the semantics of the quantifiers, and in stating the game rules for Tarski's World, we talk about "sentences of the **form** $\exists x\ S(x)$," for example. $S(x)$ here can be any wff that contains at least one free occurrence of x. So, for example, the following FOL sentence is of the form $\exists x\ S(x)$:

$$\exists x\ (\text{Cube}(x) \land \forall y\ (\text{Tet}(y) \rightarrow \text{Larger}(x, y)))$$

If we then want to talk about a given substitution instance of this existential generalization, we would use the notation $S(b)$, for example. Here, $S(b)$ means "the result of replacing every free occurrence of x in $S(x)$ with an occurrence of b." Hence, where $\exists x\ S(x)$ is the sentence above, $S(b)$ is:

$$\text{Cube}(b) \land \forall y\ (\text{Tet}(y) \rightarrow \text{Larger}(b, y))$$

Game rules for the quantifiers

The game rules are summarized on p. 237. The only rules that are new are the ones for the quantifiers, that is, for sentences of the form $\exists x\ P(x)$ and $\forall x\ P(x)$. Study these rules carefully. Here's a handy way of remembering how they work.

Existential quantifier

$\exists x\ P(x)$ is true iff at least one object satisfies $P(x)$. Call any object that does this a "**witness**." Then the game rule for $\exists x\ P(x)$ can be stated as follows: **whoever is committed to TRUE must try to find a witness**. If you are committed to TRUE, Tarski's World will ask you to choose a witness; if you are committed to FALSE, Tarski's World will try to choose a witness.

Universal quantifier

$\forall x\ P(x)$ is true iff every object satisfies $P(x)$. Call any object that does **not** satisfy $P(x)$ a "**counterexample**." Then the game rule for $\forall x\ P(x)$ can be stated as follows: **whoever is committed to FALSE must try to find a counterexample**. If you are committed to TRUE, Tarski's World will try to find a counterexample; if you are committed to FALSE, Tarski's World will ask you to find a counterexample.

In both cases, remember that if it is Tarski's World's move (that is, you have committed to TRUE for $\forall x\ P(x)$ or to FALSE for $\exists x\ P(x)$), and your commitment is **correct**, there will be no counterexample to $\forall x\ P(x)$ and no witness for $\exists x\ P(x)$. But Tarski's World will not give up—it will choose an object anyway, and try to trick you into thinking that it is a witness (or a counterexample). So don't be intimidated just because Tarski's World has made a choice. It may be bluffing!

§ 9.5 The four Aristotelian forms

Aristotle (384-322 BCE) invented the first system of formal logic. He focused on four forms of sentences—universal affirmative, universal negative, particular affirmative, and particular negative:

A *All P's are Q's.*

I *Some P's are Q's.*

E *No P's are Q's.*

O *Some P's are not Q's.*

The labels (**A**, **I**, **E**, **O**) were not due to Aristotle. They were a medieval mnemonic device, from the Latin words *affirmo* (meaning "I affirm") and *nego* (meaning "I deny"). **A** and **I** (from ***a****ff****i****rmo*) are the positive, or affirmative, ones; **E** and **O** (from *n****e****g****o***) are the negative ones.

It is important to learn these forms well, as many very complicated sentences can be shown to be based on these simple forms.

A vs. I

The most important point to be clear on at the start is the difference between **A** and **I** sentences when they are translated into FOL.

English	**FOL**
All P's are Q's	$\forall x\ (P(x) \rightarrow Q(x))$
Some P's are Q's	$\exists x\ (P(x) \land Q(x))$

Why do these FOL sentences have different connectives, as well as different quantifiers? It's pretty easy to see that ∀x (P(x) ∧ Q(x)) could not be right for *All P's are Q's*. For this FOL sentence says *everything is both P and Q*, and this is obviously too strong. *All humans are mortal* is true, but it is not true that everything is both human and mortal.

Seeing why ∃x (P(x) → Q(x)) is wrong for *Some P's are Q's* is harder. The **You try it** on p. 240 will help you see this.

An easy way to see what's wrong with this translation into FOL is to remember that P → Q is equivalent to ¬P ∨Q. This means that ∃x (P(x) → Q(x)) is equivalent to ∃x (¬P(x) ∨ Q(x)). Now compare these two sentences:

1. Some cubes are large.
2. Something is either not a cube or large.

Clearly, these are not equivalent. (1) cannot be true unless there is a large cube; but (2) does not require this—it comes out true if there is a non-cube. It also comes out true if there is a large thing—whether or not it's a cube!

In fact, the only way (2) comes out false is if **everything** is a cube and **nothing** is large. Here's a world in which ∃x (Cube(x) → Large(x)) is false. Open the files Ch9Ex2.wld and Ch9Ex2.sen. Notice that in this world of small and medium cubes, our sentence comes out false. But almost any change we make to the world makes our sentence true. Change any of the cubes into a non-cube, and the sentence becomes true; or, add any large object (of any shape) to the world, and the sentence becomes true. Notice that the correct translation of *some cubes are large*, ∃x (Cube(x) ∧ Large(x)), remains false when these changes are made.

Now let's take what we've learned from this example and apply it to any FOL sentence of the form ∃x (P(x) → Q(x)). It almost always comes out true. The only way it comes out false is if **everything** satisfies P(x) and **nothing** satisfies Q(x). Hence, it makes a statement so weak (it almost always comes out true) that it is seldom worth asserting.

Two ways of writing E

There are two ways of thinking about *No P's are Q's*. You might think of it as (a) a universal generalization or (b) a negative sentence.

(a) Universal generalization

(a) encourages this reading: for any object, if it's a *P*, then it's not a *Q*. That is, in FOL: ∀x (P(x) → ¬Q(x)).

(b) Negation

(b) encourages this reading: it is false that even one *P* is a *Q*.
That is, in FOL: ¬∃x (P(x) ∧ Q(x)).

These are both correct and perfectly acceptable ways of translating **E** sentences into FOL.

All vs. *Only*

Notice that just as *all* can be a quantifier in English (as in the phrase *all freshmen*), so too *only* can be used as a quantifier (as in *only freshmen*). Compare the following two sentences:

1. All freshmen are eligible for the Kershner prize.
2. Only freshmen are eligible for the Kershner prize.

Clearly, (1) and (2) are not equivalent. What is the difference between them? (1) tells us that being a freshman is a *sufficient* condition for eligibility—if you're a freshman, then you're eligible. But (2) tells us that being a freshman is a *necessary* condition for eligibility—you're eligible **only if** you're a freshman (but perhaps there are other necessary conditions as well). Hence, our two sentences go into FOL as follows:

1. $\forall$x (Freshman(x) $\rightarrow$ Eligible(x))
2. $\forall$x (Eligible(x) $\rightarrow$ Freshman(x))

Notice that just as, in propositional logic, *only if* indicates that the sentence that follows is the **consequent** of a conditional, so too in quantificational logic *only* indicates that the noun phrase that follows should be translated by a wff that is the **consequent** of a conditional embedded in the scope of a universal quantifier.

For practice, open Tarski's World and construct a world in which there is a small tetrahedron, a medium dodecahedron, a small cube, and a large cube. Notice that although not all the cubes are large, the only large block is a cube. Now write two FOL sentences that correspond to the English sentences (1) *All cubes are large* and (2) *Only cubes are large*. Then click Verify All. (1) should come out false and (2) should come out true. Now make the small cube large and click Verify All again. This time they should both be true. Now make the tetrahedron or the dodecahedron large (but leave the cubes both large) and re-verify. This time (1) should come out true and (2) should come out false.

For a handy chart of FOL translations of some common English quantificational sentences, download Common Quantificational Forms on the Supplementary Exercises page for this chapter.

§ 9.6 Translating complex noun phrases

It is now time to investigate sentences that are more complex than the ones we've seen so far, but that still have the basic structure of one of the four Aristotelian forms. Our first look will be at sentences that involve **complex noun phrases**, such as the following:

small happy dog

large cube in front of b

an apple or an orange

freshman or sophomore who has studied logic

In all of these cases, we could treat the complex noun phrase as a single predicate, and then use these predicates to construct atomic sentences, such as:

SmallHappyDog(pris)

But such translations are undesirable, in that they make some important logical relationships less perspicuous than they should be. We'd like to translate *Pris is a small happy dog* into FOL in a way that makes clear that this sentence has *Pris is a dog* as a consequence. And our proposed "atomic" translation above does not do this.

A better way is to use truth-functional connectives and more familiar (and less complicated) predicates. So *Pris is a small happy dog* will become:

Small(pris) $\wedge$ Happy(pris) $\wedge$ Dog(pris)

Translating the complex noun phrase, then, means finding an appropriate truth-functional compound of wffs that are not sentences (i.e., wffs containing a free variable). Our remaining examples look like this:

Large(x) ∧ Cube(x) ∧ FrontOf(x, b)

Apple(y) ∨ Orange(y)

(Frosh(x) ∨ Soph(x)) ∧ StudiedLogic(x)

We can then embed these wffs in sentences, either by replacing the free variables with a name, or prefixing the appropriate quantifier. We'll do that with these sentences:

There is no large cube in front of b.

¬∃x (Large(x) ∧ Cube(x) ∧ FrontOf(x, b))

If Bob eats anything, it will be an apple or an orange.

∀y (Eats(bob, y) → (Apple(y) ∨ Orange(y))

Any freshman or sophomore who has studied logic will succeed.

∀x ((Frosh(x) ∨ Soph(x)) ∧ StudiedLogic(x)) → Succeed(x))

[There is a possible ambiguity in this last sentence: in which of these two ways do we read the noun phrase?

(*freshman or sophomore*) *who has studied logic*

freshman or (*sophomore who has studied logic*)

The first is more natural (it's the one we used above), but the second is still possible.]

Sometimes, the correct rendition of a complex noun phrase in FOL is surprising. Take, for example, the phrase *apples and oranges*. We might expect this to go into FOL as Apple(x) ∧ Orange(x). But study this wff carefully. Which objects satisfy it? It takes only a little thought to realize that **nothing** satisfies it, for in order to satisfy this wff, an object would have to satisfy **both** of the wffs Apple(x) and Orange(x). But no object does this, since no object is both an apple and an orange.

The correct rendition of *apples and oranges* is more likely to be Apple(x) ∨ Orange(x). For when you consider such sentences as:

Apples and oranges are fruits.

Bob eats only apples and oranges.

it is clear that the FOL sentences that capture their meanings are:

∀x ((Apple(x) ∨ Orange(x)) → Fruit(x))

∀x (Eats(bob, x) → (Apple(x) ∨ Orange(x))

Conversational implicature and quantification

When we use such English quantificational phrases as *every applicant*, *all my grandchildren*, etc., there is an apparent implication that there **are** some applicants, that I **have** some grandchildren, etc.

But in FOL, such sentences as:

$$\forall x\ (\text{Applicant}(x) \rightarrow \text{Hired}(\text{bill}, x))$$

$$\forall y\ (\text{Grandchild}(y, \text{marc}) \rightarrow \text{Brilliant}(y))$$

come out true when nothing satisfies the wff in the antecedent. So if there were no applicants, the FOL translation of *Bill hired every applicant* comes out true; and if I have no grandchildren, the FOL translation of *All my grandchildren are brilliant* comes out true.

What are we to say of these *vacuous* generalizations? They come out true in FOL, but when we assert their English translations, something seems wrong with them. But what is wrong with them? Is it that they are **false**?

The most widely accepted answer to this question makes use of Grice's notion of conversational implicature. Grice's answer is not that vacuous generalizations are false, but that they are **misleading**.

What is misleading about them is that the speaker has not been fully forthcoming with all the information at his or her disposal. If the speaker knows that there are no applicants, or that Marc has no grandchildren, the most fully informative statements he or she could make about the applicants, or about Marc's grandchildren, are:

There were no applicants.

Marc has no grandchildren.

The statements we are considering:

Bill hired every applicant.

All of Marc's grandchildren are brilliant.

make weaker claims—each is a logical consequence of its counterpart "negative existential," but does not logically imply it. So the vacuous generalization makes a weaker claim.

The relation is just the same as that between a disjunction and one of its disjuncts—the disjunction makes a weaker claim than the disjunct standing alone. But clearly the weaker claim is not false—it is just a weaker version of the truth. For example, if I tell my wife "Your keys are either in the kitchen or by the front door," and I know that they are in the kitchen, I have not lied—I have not said something false. I have misled her, of course, by withholding relevant information that I possessed, namely, that they are not by the front door. I should have said simply "They are in the kitchen." But my mistake was not in saying something false; rather, it was in not telling all of the truth I was in possession of. I may have conversationally implicated that I did not know the exact location of her keys, but I did not assert that.

That this is exactly what is going on in the cases we are considering becomes apparent when we consider an equivalent FOL sentence:

$$\forall y\ (\neg\text{Grandchild}(y, \text{marc}) \lor \text{Brilliant}(y))$$

This is equivalent to the standard FOL version of *All of Marc's grandchildren are brilliant* and it stands in just the same relation to its counterpart negative existential:

$$\forall y\ \neg\text{Grandchild}(y, \text{marc})$$

Here, too, the weaker statement is a disjunction and the stronger statement is one of its disjuncts.

So when I, who have no grandchildren, say that all of my grandchildren are brilliant, I say something that is misleading, but not false. For I have not asserted, falsely, that I have grandchildren, although I have **implicated** this. The implicature can be cancelled, for I can say, without contradicting myself (barely!) *All of my grandchildren are brilliant—unfortunately, I don't have any grandchildren*.

