5.11. Quantifiers, Variables, and Binding

1. Bound and Free Variables. We stated our updated construction rules in in terms of "formulas" – where "**formula**" is an umbrella term, covering any formal **sentence or quasi-sentence**. So "P," "Ga," and "Gx" are all formulas.

Formulas

Formal Sentences — P, Ga

Quasi-Sentences — Gx

Our construction rules are quite lax about what they attach a quantifier to. For instance, the formal language allows us to attach a universal quantifier to the formula "Ga".

∀x Ga

|

Ga

Featuring as it does a name letter, "Ga" is the formal counterpart to an English sentence such as "Lucretia is an engineering major".

a: Lucretia **G**: __ is an engineering major

Lucretia is an engineering major: **Ga**

So the formula "∀x Ga" translates the following English sentence.

For every object in the universe, the following holds true of it: Lucretia is an engineering major.

Of course that would be a thoroughly strange thing to say, since attaching the long quantifier phrase at the beginning adds nothing to the claim about

Lucretia that follows. To say "For every object in the universe, the following holds true of it: Lucretia is an engineering major" is, meaning- and communication-wise, just to say "Lucretia is an engineering major". The quantifying preamble is **semantically empty**.

And from what we understand of quantifier semantics we can see that the same is true of the formula "∀x Ga". To determine whether "∀x Ga" is true in a model, we construct its **instances** by (i) removing the quantifier "∀x," then (ii) replacing the variable in the remaining scope formula with each name used in that model – counting "∀x Ga" true just in case we get a true sentence for each such instance.

Yet the scope formula of "∀x Ga" is "Ga". Replacing 'the variable' in "Ga" by a name letter involves no change at all – since "Ga" contains no variable to begin with. That means **"∀x Ga" will only ever have one instance: "Ga"**. Now we judge a universal sentence true or false based on the truth or falsehood of its instances. But with "Ga" its only instance, **"∀x Ga" will true in exactly the same models that make "Ga" true**. Adding the "∀x" winds up being semantically null.

In the jargon of formal logic "∀x Ga" is a case of **vacuous quantification**: adding a quantifier that makes no semantic difference.[1]

That's in sharp contrast to earlier examples of quantification such as "∀x Hx," which translates the English sentence "Everything is physical" (or, more technically, "For every object in the universe, the following holds of it: it is physical"). Whether true or false, that sentence has none of the long-winded semantic oddness we find in vacuous quantification. Here the quantifier phrase really seems to be 'quantifying over' things.

Intuitively that's because here the quantifier phrase links up with the English mini-sentence "it is material" which follows – just as in the formal language the quantifier "∀x" links up with the formula "Hx" following it.

[1] While it may seem odd for the construction rules to permit something as semantically pointless as vacuous quantification, this sort of thing isn't new. The construction rules of previous chapters allowed us to build a sentence such as "(P ∧ P)" which is the (needlessly complicated) semantic equivalent of plain old "P". We could think of "(P ∧ P)" as a case of 'vacuous conjunction'.

In both cases the quantifier, and the (quasi-)claim that follows, contain the same variable term: "it" in English, variable "x" in the formal language.

For every object in the universe, the following holds true of **it**:
it is material.

∀**x** H**x**

To this notion – of the two parts 'linking up' thanks to a matching variable – we give the name "**variable binding**". Even before setting out the technical details of variable binding, we spot one crucial component: the quantifier and the formula that follows must **contain the same variable**.

That's what left the quantifier useless in the earlier example "∀x Ga": the "x" in "∀x" found no matching "x" in "Ga" allowing such a connection.

And likewise in English: the variable pronoun "it" in the quantifier phrase "For every object in the universe, the following holds true of it" has no counterpart in the English sentence "Lucretia is an engineering major".

For every object in the universe, the following holds true of **it**:
Lucretia is an engineering major.

∀**x** G**a**

Lack of any opportunity for variable binding is what leaves the quantifier vacuous in the formula "∀x Ga".

The full details of variable binding simply combine two elements encountered above: (1) construction attaches a quantifier to its **scope formula**, and (2) the quantifier binds the **matching variable** in that scope formula.[2]

For a quantifier to **bind** a variable,

(1) the variable being bound must appear in the scope formula of that quantifier,

and

(2) the variable being bound must be the same variable used in the quantifier.

So in the previous example the "x" in "Gx" was bound by "∀x" because it met both these conditions.

∀x Gx

|

Gx

But in "∀x Ga" the quantifier "∀x" finds no matching "x" in its scope to match up with – so "∀x" ends up being extra baggage.

∀x Gx

|

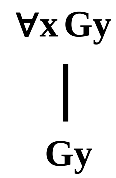
Ga

Indeed, that failure to bind variables is central to our official definition of a "vacuous quantifier".

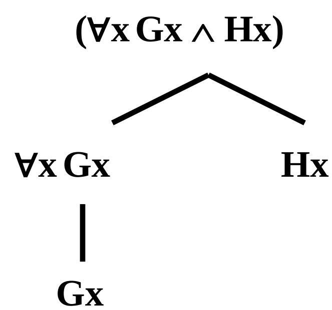
Vacuous quantifier: a quantifier which binds no variables

[2] The variable in the quantifier – for example, the "x" in "∀x" – counts as bound.

“∀x Gy” likewise sins against the matching variable requirement, leaving “∀x” nothing in its scope to bind. So here too “∀x” is a vacuous quantifier.



On the other hand, the formula “(∀x Gx ∧ Hx)” instead violates the scope requirement on binding. Because the “x” in “Hx” isn’t in the scope of quantifier “∀x,” the quantifier can’t reach that variable to bind it (even though the variable matches).



A bit more jargon here helps head off potential confusion over variables.

While the “x” in “Gx” and the one in “Hx” are certainly two different things – they’re in different locations – it seems odd to call them “different variables”. On the contrary, they’re both the **same** variable – namely, “x”. The situation is like one where Rex is reading Elements of Logic in Las Vegas and discussing it on the phone with Suki, who’s reading Elements of Logic in San Diego. Does Elements of Logic then count as **one** book (because Rex and Suki are reading “the same book”) or **two** (because the book Rex is reading is hundreds of miles from the one Suki’s reading)?

With a book we sort out the confusion by speaking of **copies** of the book: Rex and Suki have two **different copies** of one and the **same book**. And in formal logic we speak likewise of different **occurrences** of a variable: in the formula “(∀x Gx ∧ Hx)” the “x” in “Gx” and the one in “Hx” are two different **occurrences** of **the same variable**, “x”.

That allows us to fold in another piece of jargon: a variable-occurrence which isn't bound by any quantifier is said to be **free**. Since in "(∀x Gx ∧ Hx)" the formula "Hx" lay outside the scope of "∀x," the "x" occurrence in "Hx" is free.

> A **free variable-occurrence** is a variable-occurrence not bound by any quantifier; and a **free variable** is one with a free occurrence[3]

We can now make good on an outstanding debt, by drawing the official border between sentences and quasi-sentences.

> A **sentence** is a formula with no free variables.
>
> A **quasi-sentence** is a formula with one or more free variables.[4]

Note that since the formal sentences of previous chapters lacked variables entirely, they had no **free** variables – and so still qualify as sentences.

Vacuous quantification provided one example of something permitted by the construction rules but semantically odd. Free variables provide a second example. Indeed, such is the oddness of free variables, communication-wise, that it's difficult even to give an English example.

---

[3] A variable is free in a formula if even one of its occurrences is free; and a variable is bound in a formula if even one of its occurrences is bound. So – though it might seem odd – **a variable can be both bound and free** in the same formula. But any particular variable-**occurrence** is either bound or free, **not both**. By analogy: if a library with multiple copies of Elements of Logic is taking inventory, and recording whether a book has been written in by readers or is free of notes, Elements of Logic might be recorded as both – because one **copy** has notes in the margins while another doesn't. But any particular copy of Elements of Logic is either free of readers' notes or else written in, **but not both**.

[4] What we call a "sentence" some authors call a "closed sentence," and our "quasi-sentence" is instead an "open sentence" – for example, (Quine 1959: 90, crediting Carnap) and (Kleene 1967: 105). Reaching back even earlier: our "sentence" is what Russell and Whitehead call a "proposition," while our "quasi-sentence" they call a "propositional function" (Russell and Whitehead 1910: 38). The different technical dialects can overlap: (Smullyan 1968/1995: 44) equates "closed formula" with "sentence," while Quine equates "closed sentence" with "statement". Mixing vocabularies, (Gamut 1982/1991 Vol. I: 74) contrast "sentence" and "propositional function," and (Partee, ter Meulen, and Wall 1990: 138) contrast "statement" with "open statement or propositional function".

The following sort of exchange is perhaps the closest we come to illustrating free variables in ordinary language.

> **Kitty**: Something is made of wood, and it's painted red.
> **Trixie**: I assume that the "it" which is painted red is the previously-mentioned wooden object, right?
> **Kitty**: Oh, no. I just mean "it" to point to some object (or objects) – not necessarily the wooden object. I'm not saying *what* object "it" points to in the sentence "It is painted red".
> **Trixie**: That's an odd way of talking. Who goes around using the word "it," but without giving any indication which object they mean?

We're inclined to side with Trixie here: a use of "it" which is explicitly **not** referred to something by the context of utterance (say, a pointing finger) nor bound by a quantifier phrase (such as "something") is communicatively perverse.[5] If there's no saying what "it" points to, its use seems pointless.

It's no coincidence that free variables strike us as semantically odd, just as vacuous quantifiers did earlier. For really they're two sides of the same coin: vacuous quantification involves a quantifier with no variables to bind, while quasi-sentences have variables with no quantifier to bind them. Semantically, quantifiers and variables are made for each other.

And for that reason quasi-sentences – formulas with free variables – will never be of interest to us except as stepping stones toward constructing genuine sentences, free of free variables. That provides a good general policy for translating from English to the formal language: **if the finished translation contains free variables, something has gone wrong**.[6]

2. The Uses of Variable Binding. In expanding the mechanisms of our formal language, we expanded as well the ability to express complex claims. But with greater expressive power comes greater potential for confusion and error. Though it may not be obvious, a proper understanding of variable-

---

[5] As Quine puts it: "The analogue of a free variable in ordinary language is a pronoun for which no grammatical antecedent is expressed or understood, and the analogue of an open sentence [i.e., a quasi-sentence] is a clause containing such a dangling pronoun." (Quine 1982: 134)

[6] As Suppes puts it: "With respect to the problem of correctly symbolizing sentences of everyday language, it should be emphasized that the end result should contain no free variables. In other words, the symbolized expression should also be a sentence" (Suppes 1957: 54).

binding is essential for drawing distinctions between different, though deceptively similar sentences.

So returning to earlier examples, we'd better not confuse the following pair of English sentences – since, despite similar wording, they make very different claims.

(1) Something is a cat and also isn't a cat.
(2) Something is a cat and something isn't a cat.

And the formal language should translate these sentences in a way that preserves that difference. We rephrase (1) in 'technical English'.

(1) Something is a cat and also isn't a cat.

For some object, *x*, the following holds of *x*:
***x* is a cat and *x* also isn't a cat.**

The mini-sentence "x is a cat and also isn't a cat" is a simple conjunction. A translation key yields the following formula.

G: __ is a cat

(Gx ∧ ~Gx)

The first part of the sentence – "For some object, x, the following holds of x" – is translated by an existential quantifier.

∃x (Gx ∧ ~Gx)

Sentence (2), by contrast, is itself a conjunction of two existential claims – as technical rephrasing brings out.

(2) Something is a cat and something isn't a cat.

For some object, *x*, the following holds of *x*: *x* is a cat.
and

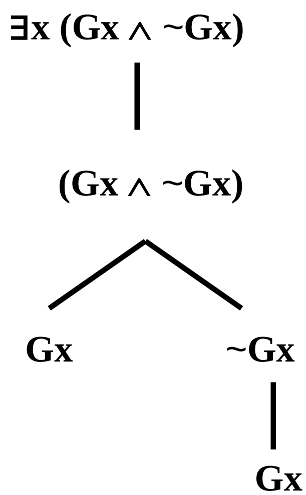
For some object, *x*, the following holds of *x*: *x* is not a cat.

Using the same translation key, the first existential claim translates as “$\exists x\, Gx$,” while the second is “$\exists x \sim Gx$” – yielding this conjunction.

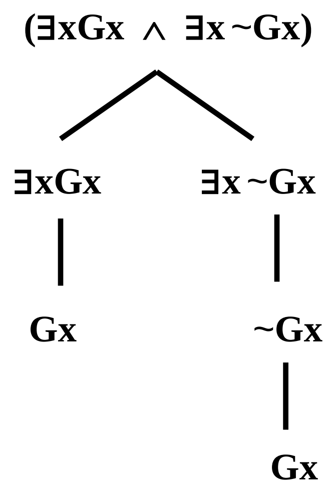
$$(\exists x\ Gx \wedge \exists x \sim Gx)$$

Construction trees show how variable binding draws the necessary distinctions here between the two sentences.

(1) “***Something is a cat and also isn’t a cat***”



(2) “***Something is a cat, and something isn’t a cat***.”



In (1) a single existential quantifier binds both occurrences of “x” that follow; whereas in (2) the “x” occurrences in “Gx” and “~Gx” aren’t bound by the same quantifier. English Sentence (2) shows that there’s nothing absurd about having cathood and lack-of-cathood together in the world (or mentioned in the same sentence).

(1) Something is a cat and also isn’t a cat.	$\exists x\ (Gx \wedge \sim Gx)$
(2) Something is a cat, and something isn’t a cat.	$(\exists x\ Gx \wedge \exists x \sim Gx)$

What’s absurd about English Sentence (1) is its claim that being a cat and not being a cat are found **in the same object**. Sentence (2) escaped absurdity because it **didn’t** claim that the cat object and the non-cat object are **one and the same object**.

Variable binding draws the same distinction in formal terms. In the formal translation of (1), the single quantifier "∃x" binds both the "x" in "Gx" and the one in "~Gx". Binding both variable occurrences by the same quantifier is a formal way of saying that the object that's G and the one that's ~G are the **same object**.

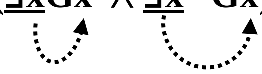
By contrast, in the formal translation of (2) the "x" in "Gx" is bound by one quantifier, while the "x" in "~Gx" by another. Since the different variable occurrences aren't here bound by the same quantifier, Sentence (2) **doesn't** claim that that the G object and the ~G object are the **same** object.

(1) "***Something is a cat and also isn't a cat***"

$\underline{\exists x}$ (Gx ∧ ~Gx)

(2) "***Something is a cat, and something isn't a cat***."

($\underline{\exists x}$Gx ∧ $\underline{\exists x}$~Gx)



We noted earlier that adding a quantifier – and hence binding otherwise free variables – transforms an incomplete quasi-sentence into a full-fledged formal sentence, by forcing those variables to point to **some** object(s). Now we see something more: by having a **single** quantifier bind several variable occurrences, we force all those occurrences to point to the **same** object(s) throughout – and so force the sentence containing those variable-occurrences to **make claims** about the same object(s).

Summary: Sentences, Formulas, and Binding

- For a **quantifier** to **bind** an **occurrence of a variable**:

  (1) the variable-occurrence being bound must be the same variable appearing in that quantifier,

  and

  (2) the variable-occurrence being bound must appear within the *scope* of that quantifier.

- A **free** variable-occurrence is one that's not bound.
- A **variable** is **free** if it has a free occurrence; a **variable** is **bound** if it has a bound occurrence. (So a variable can be both bound and free. Each variable-occurrence, however, is bound or free, but not both.)
- A **quasi-sentence** is a formula with at least one free variable.
- A **sentence** is a formula with no free variables.