5.5. A First Look at Quantifier Semantics

1. Quasi-Sentences and Formulas. Introducing name and predicate letters was a simple matter of adding a second kind of atomic sentence. But variables bring a new complication to the formal language.

We noted that a little English sentence such as "It is made of wood" doesn't make a complete claim without help from some outside factor – a pointing finger, or prominent bit of background context. For I can utter these words and make a true claim (when pointing at a log cabin), but in the same situation utter these words to make a false claim (while pointing at a brick house). Since variables are the formal counterpart to pronouns like "it," a formal string such as "Gx" suffers the same incompleteness: without some outside factor pinning down what "x" is pointing to, "Gx" **doesn't make a complete claim**.

That is in stark contrast to the atomic sentences encountered earlier. For "P" or "GA" are by themselves capable of being true or false, just like their English counterparts – say, "Learning is a form of remembering," or "Plato is Greek".

So – taking **sentences** to be complete-claim makers, capable of truth or falsehood – we do not count "Gx" as a formal sentence. But we recognize its close resemblance to genuine sentences: it is, construction-wise, built the same way as our new atomic sentences (just a predicate-letter-plus-*variable*, rather than predicate-letter-plus-*name-letter*). Coining a new bit of jargon, we say that such an incomplete-claim-maker is a **quasi-sentence**.[1]

That rough, informal statement leaves undone the task of stating *precisely* what counts as a (genuine) formal sentence, and what is a (mere) quasi-sentence. But even in advance of precise criteria, a further bit of jargon suffices to complete construction rules for the expanded formal language.

We will use "**formula**" as an umbrella term, covering any formal **sentence or quasi-sentence**. So "P," "GA," and "Gx" are all formulas.

[1] Adapting the "quasi-statement" of Lambert and van Fraassen 1972: 79-80.

Formulas

Formal Sentences — P, GA

Quasi-Sentences — Gx

The revised construction rules can then be stated in terms of formulas.[2]

Construction Rules (*Final Version*)

Atomic Formulas:

A1. Sentence letters are atomic formulas

A2. A predicate letter followed by a name letter *or* variable is an atomic formula.

Formulas:

1. Atomic formulas are formulas.
2. If ● is a formula, then ~● is a formula.
3. If ● and ▲ are formulas, then (● ∧ ▲) is a formula.
4. If ● and ▲ are formulas, then (● ∨ ▲) is a formula.
5. If ● and ▲ are formulas, then (● →▲) is a formula.
6. If ● and ▲ are formulas, then (● ↔▲) is a formula.
7. If ★ is a variable and ● is a formula, then

∃★ ●

and

∀★ ●

are both formulas.

Note that since a quantifier attaches to the left of a formula, **construction-wise** quantifiers act just like tildes.

[2] The ★ symbol is pronounced "*star*". It is used here as a generic blank which any *variable* can fill – just as ● is a blank which any *formula* can fill.

In Chapter Three we called the sentence that the tilde attaches to the "**scope**" of that tilde. Here likewise: the formula which a quantifier attaches to is the **scope** of that quantifier.

So in the formula "$\forall x\, Gx$", the scope of "$\forall x$" is the formula "Gx".

$$
\begin{array}{c}
\mathbf{\forall x\, Gx} \quad \textit{(7)} \\
| \\
\mathbf{Gx} \quad \textit{(A2, 1)}
\end{array}
$$

Operating on such a scope formula will prove central to understanding the semantics of quantifiers.

2. Quantifier Semantics: The Elements. Already semantics for name and predicate letters was stated in terms of the model, and its domain of objects. And armed with these, the truth-and-falsehood profiles of universal and existential sentences look straightforward: a universal sentence is true in a given model if (and only if) what it says is true of every object in the domain of that model; whereas an existential sentence will be true just when what it says holds true of at least one object in the model's domain.

Our formal semantics will remain faithful to that intuitive description – but (naturally) in a way that re-states those points with enough formal exactitude to handle even very complex examples.

While we spoke of a universal sentence being true just when "what it says" is true, we can make that rough claim more precise: the "what it says" is just the **scope formula** following the quantifier. So returning to the earlier example "Everything is a material object," we take that sentence to be true just where "It is a material object" is true of each object in the universe. And likewise, taking "$\forall x\, Gx$" as formal translation of "Everything is a material object," "$\forall x\, Gx$" will be true in a model just where its scope formula "Gx" is true of every object in the domain of that model.

But that conflicts with what was said earlier about a quasi-sentence such as "Gx": that it's not a candidate for truth or falsehood since it is not a complete-claim-maker. How can this scope sentence make true claims about every object in the domain, if it isn't capable of truth or falsehood?

Here a peculiarity of our semantics comes to the rescue, allowing full-fledged sentences to stand in place of the scope formula.

Recall that we require every object in the domain to have a name – that is, a name letter. That requirement guarantees that everything true of an object in a model can be expressed in a sentence of the formal language. And those true sentences will be closely related to the scope formula of quantified sentences.

So in the following model with three objects in its domain, we expect that the sentence "∃x Gx" to be true, since there is at least one object here which is G. (In fact, there are two such objects: 3 and 4.) But we expect the sentence "∃x Jx" to be false in this model, since not even one object is J.

D: {2, 3, 4}

A: 2	**G**: {3, 4}
B: 3	**H**: {2, 3, 4}
C: 4	**I**: {4}
D: 3	**J**: { }

Likewise we take the sentence "∀x Hx" to be true in this model, since every object in the domain is H. But we judge "∀x Gx" to be false here, since not every object in this model is G. (2 isn't G.)

Now, since every object in the model must have a name, each of those observations about truth and falsehood will have a sentence counterpart. For example, since every object that's G has a name (3 is named "B" and "D," 4 is named "C"), the sentences "GB," "GC," and "GD" are true. But note: "GB," "GC," and "GD" are each just the scope formula of "∃x Gx" (namely, "Gx") with a name letter in place of the variable "x".

We will say that "GB" and "GC" are each an **instance** of the scope formula "Gx". While more detail must be added to the definition of "instance" in order to apply it to complex formulas, the following preliminary definition works for atomic formulas such as "Gx" or "Jy".

Instance of a Scope Formula (*First Draft*):

> **The result of replacing the variable in the scope formula by a name letter**[3]

So for our model using four name letters – "A," "B," "C," and "D" – there are four instances of the scope formula "Gx".

Scope Formula:	**Instances of This Formula (For This Model)**:
Gx	GA
	GB
	GC
	GD

And where "Gx" is the scope formula of a quantified sentence such as "∃x Gx" or "∀x Gx," we will say that any instance of "Gx" is, by association, an instance of that quantified sentence as well.

Instance of a Quantified Sentence (*First Draft*)

> **For a quantified sentence, an instance of that sentence is the result of removing the quantifier, and replacing the variable in its scope formula by a name letter.**

Since "Gx" is the scope formula of "∃x Gx" and of "∀x Gx," the above four instances of "Gx" also count as instances of "∃x Gx" and "∀x Gx".

---

[3] Note that we speak here of "*the* variable" – thus taking for granted that there will be only one variable in the scope formula. That assumption holds for simple formulas such as "Gx" or "~Jx," but not for more complex formulas. That's why this definition of "instance" is only a first draft. (Likewise the definition of "instance of a quantified sentence" below refers to "*the* quantifier" of the sentence – thereby assuming there is *only one* quantifier in the sentence. Again, the definition will be modified later to scale up to larger sentences.)

And we will speak of "an instance of a formula (or quantified sentence) **in a model**" – meaning, an instance using a name letter that appears in that model. So the quantified sentences "∃x Gx" and "∀x Gx" have four instances in the above model (since that model features four name letters). With the notion of "instance" in hand, it's easy to get correct results for the truth and falsehood of quantified sentences.

For existential sentences: we expect the sentence "∃x Gx" to be **true** in a model if there is at least one object from that model's domain which is in the extension of "G" – i.e., if **at least one object in the model is G**. And we expect "∃x Gx" to be **false** in a model if there are no objects in its domain in the extension of "G" – that is, if **no objects in the model are G**.

But note: because the semantics requires each object in the domain to have a name, whenever at least one object is in the extension of "G" there will be at least one true instance of "∃x Gx" – say, "GA," or "GB". We can thus restate the above points in terms of "instances".

> **"∃x Gx" is true in a model if (and only if) "∃x Gx" has at least one true instance in that model.**
>
> **"∃x Gx" is false in a model if (and only if) "∃x Gx" has no true instances in that model**.

So in our earlier model (repeated below), **"∃x Gx" is true**, since the sentence has at least one true instance in this model. (In fact it has three: "GB," "GC," and "GD".) But **"∃x Jx" is false** in this model, since "∃x Jx" has not even one true instance. (All the instances of "∃x Jx" in this model – "JA," "JB," "JC," and "JD" – are false.)

D: {2, 3, 4}		**Instances of "Gx"**:	**Instances of "Jx"**:
A: 2	**G**: {3, 4}	GA: 0	JA: 0
B: 3	**H**: {2, 3. 4}	GB: 1	JB: 0
C: 4	**I**: {4}	GC: 1	JC: 0
D: 3	**J**: { }	GD: 1	JD: 0

For universal sentences: we expect the sentence "∀x Gx" to be **true** in a model if every object in the model's domain is in the extension of "G" – that is, **if every object is G**. And we expect "∀x Gx" to be **false** in a model if even one object in the domain is not in the extension of "G" – that is, **if even one object is not G**. Those points are restated in terms of "instances" as follows.

> **"∀x Gx" is true in a model if (and only if) every instance of "∀x Gx" in that model is true.**
>
> **"∀x Gx" is false in a model if (and only if) "∀x Gx" has even one false instance in that model**.

So **"∀x Hx" is true** in our earlier model (repeated below), since each instance of "∀x Hx" in this model – "HA," "HB," "HC," and "HD" – is true. But **"∀x Gx" is false** in this model, since "∀x Hx" has at least one false instance in this model (namely: "GA").

D: {2, 3, 4}		**Instances of "Hx"**:	**Instances of "Gx"**:
A: 2	**G**: {3, 4}	HA: 1	GA: 0
B: 3	**H**: {2, 3. 4}	HB: 1	GB: 1
C: 4	**I**: {4}	HC: 1	GC: 1
D: 3	**J**: { }	HD: 1	GD: 1

3. Quantifier Negation. So far the scope formulas treated semantically have all been atomic – e.g., "Gx" or "Hx" – semantics for complex quantified sentences waiting upon further technical details. But already we can account for the truth and falsehood of scope formulas which are negations of atoms – for example, "~Gx" or "~Hx". The semantic rule for negations dictates that if a certain instance of, say, "Gx" is true in a model – for example, "GB" in the previous example – then its negation, "~GB," will be false in that model.

So the quantified sentence **“∃x ~Gx” is true** in our previous model (repeated below). For since “GA” is false in that model, “~GA” is true; and “~GA” is an instance of “∃x ~Gx”.

D: {2, 3, 4}		**Instances of “Gx”:**	**Instances of “~Gx”:**
A: 2	**G**: {3, 4}	GA: 0	~GA: 1
B: 3	**H**: {2, 3. 4}	GB: 1	~GB: 0
C: 4	**I**: {4}	GC: 1	~GC: 0
D: 3	**J**: { }	GD: 1	~GD: 0

Note that **“~∀x Gx” is** also **true** in this model. For “GA” is false here, meaning “∀x Gx” has a false instance – thus making “∀x Gx” false, and “~∀x Gx” true.

Now it is no coincidence that both “∃x ~Gx” and “~∀x Gx” are true together. In fact, *any* model making one of these quantified sentences true will make the other true. For if “∃x ~Gx” is true in a model, that’s because there is at least one true instance of “~Gx” (for example, “~GA”); and a model making that negation true makes the sentence it’s negating (for example, “GA”) false. That means “∀x Gx” has at least one false instance, and so is false – making “~∀x Gx” true.

The same chain of reasoning, started from the other end, ensures that whenever “~∀x Gx” is true, “∃x ~Gx” will be as well. This makes sense intuitively: something is non-G if and only if not everything is G.

Similar semantic reasoning shows that “~∃xGx” and “∀x ~Gx” are likewise semantically equivalent. (Intuitively: if not even one thing is G, then everything is non-G.)

Together, these equivalences makes up the laws of **Quantifier Negation**.[4]

Quantifier Negation

"∃x ~●" **is equivalent to** "~∀x ●"

"∀x ~●" **is equivalent to** "~∃x ●"

Moreover, since "∀x ~Gx" and "~∃x Gx" are logically equivalent, their negations are equivalent as well – that is, "~∀x ~Gx" and "~~∃x Gx" are logically equivalent. And since the double negation "~~∃xGx" is in turn equivalent to "∃xGx," we conclude: **"~∀x ~Gx" is logically equivalent to "∃x Gx"**. In effect: we can **define** the existential quantifier in terms of universal and tildes.[5]

Likewise, by way of quantifier negation and double negation, **"~∃x ~Gx" is logically equivalent to "∀xGx"**.[6]

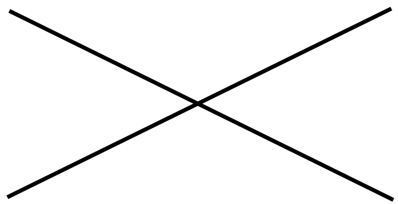
That means any sentence we translate with one quantifier could be translated by the other instead. We can construct a miniature (one-predicate) Square of Opposition illustrating these equivalences.[7]

[4] These are simple forms of Quantifier Negation, because the scope formula following the quantifier here is just an atomic formula such as "Gx," or its negation "~Gx". We will apply Quantifier Negation to universal and existential sentences of any complexity.
[5] So our formal language could translate all the same English sentences with just the universal quantifier – translating "some" as "~∀x~".
[6] So our formal language could translate all the same English sentences with just the existential quantifier – translating "all" as "~∃x~".
[7] Adapting the more traditional two-predicate Square of Opposition tracing back to Aristotle's *On Interpretation* Chapters 6 and 7 and *Prior Analytics* Book I, Chapter 2.

Mini-Square of Opposition

(A) Everything is G
$\forall x\, Gx$
$\sim\exists x \sim Gx$

(E) Nothing is G
$\forall x \sim Gx$
$\sim\exists x\, Gx$



(I) Something is G
$\exists x\, Gx$
$\sim\forall x \sim Gx$

(O) Something is not G
$\sim\forall x\, Gx$
$\exists x \sim Gx$

Note that each sentence is equivalent to the negation of the sentence diagonal from it. So "Nothing is G," translated as "$\sim\exists x\, Gx$," is in effect the negation of "Something is G," translated as "$\exists x\, Gx$".

The sentences on the right also highlight some helpful points about translation. ***First***, the difference of tilde scope in "$\sim\exists x\, Gx$" and "$\exists x \sim Gx$" is a difference that makes a difference. In translating, we cannot be casual about which formal operator comes first – the tilde or the quantifier – since switching the two drastically changes the claim being made. ("Nothing is G" is a claim quite different from "Something is not G". And likewise "$\sim\forall x\, Gx$" and "$\forall x \sim Gx$" make very different claims.)

But ***second***, we can generally follow the order of negation and quantifiers in English in order to get the scope write in formal translation..

"**Not all**" is translated as "$\boldsymbol{\sim\forall x}$"
"**All are non-**" is translated as "$\boldsymbol{\forall x\sim}$"

"**Some are non-**" is translated as "$\boldsymbol{\exists x\sim}$"
"**Not (even) some**" is translated as "$\boldsymbol{\sim\exists x}$"

Summary: Quantifier Semantics (*First Draft*)

- **Instance of a Quantified Sentence** (*First Draft*):

  For a quantified sentence, an instance of that sentence is the result of removing the quantifier, and replacing the variable in its scope formula by a name letter.

- **Existential Semantics** (*Simple Version*):

  "∃x Gx" is true in a model if (and only if) "∃x Gx" has at least one true instance in that model.

  "∃x Gx" is false in a model if (and only if) "∃x Gx" has not even one true instance in that model.

- **Universal Semantics** (*Simple Version*):

  "∀x Gx" is true in a model if (and only if) every instance of "∀x Gx" in that model is true.

  "∀x Gx" is false in a model if (and only if) "∀x Gx" has even one false instance in that model.

- **Quantifier Negation**:

  "∃x~●" **is equivalent to** "~∀x ●"
  "∀x~●" **is equivalent to** "~∃x ●"