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 stone cathedral). 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's in stark contrast to the atomic sentences encountered earlier. "P"and "GA" are by themselves capable of being true or false, just like their English counterparts – say, "Exercise is bad for the soul" or "Neko is a cat".

So – taking **sentences** to be complete-claim makers, capable of truth or falsehood – we don't 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 (complete) 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'll 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	**Quasi-Sentences**
P, GA	Gx

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

Construction Rules (*Final Version*)

Terms

T1. Name letters are terms
T2. Variables are terms

Atomic Formulas:

A1. Sentence letters are atomic formulas
A2. A predicate letter followed by a term 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.

7a. If ★ is a variable and ● is a formula, then ∃★ ● is a formula.
7b. If ★ is a variable and ● is a formula, then ∀★ ● is a formula.

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 Two we called the sentence that the tilde attaches to the "**scope**" of that tilde.[3] Here likewise: the formula which a quantifier attaches to is the **scope** of that quantifier.

So in the formula "∀x Gx", the scope of "∀x" is the formula "Gx".

∀x Gx *(7b)*

|

Gx *(T2, A2, 1)*

Operating on a scope formula will prove central quantifiers semantics.

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 model's domain of discourse; whereas an **existential** sentence will be true if (and only if) what it says holds true of **at least one** object in the domain.

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

To sharpen the claim that a universal sentence is true when "what it says" is true of every object, note that the "what it says" here is just the **scope formula** following the quantifier. Returning to the earlier example "Everything is a material object": we take this sentence to be true just where "It is a material object" is true of each and every object in the domain. Likewise, taking "∀x Gx" as formal translation of "Everything is a material object," "∀x Gx" is true in a model just where its scope formula "Gx" is true of every object in the domain of that model.

But that seems to conflict with what we said earlier about a quasi-sentence such as "Gx": that it's not a candidate for truth or falsehood, since it's not a

[3] In 2.10.

complete-claim-maker. How can this scope formula make true claims about every object in the domain if it isn't capable of truth at all?

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 intimately related to the scope formula of the quantified sentence.

For example, in the following model we expect the sentence "∃x Gx" to be true, since there's at least one object here that's 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.

$\mathbb{D}$: {**2**, **3**, **4**}

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

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

Since every object in the model has a name, each of those observations about truth and falsehood can be restated in terms of sentences.

For example: since every object that's G has a name (**3** is named "B", and **4** is named "C"), the sentences "GB" and "GC" are true. But "GB" and "GC" are each just the **scope formula** of "∃x Gx" – namely, "Gx" – **with** a **name letter in place of the variable** "x".

We'll say that "GB" and "GC" are each an **instance of the scope formula "Gx"**. While more detail is needed before we can apply instances to complex formulas, the following account works for atomic formulas such as "Gx" or "Jy", or their negations ("~Gx" or "~Jx").[4]

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

> **The sentence that results from replacing the variable in the scope formula by a name letter[5]**

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

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

And since "Gx" is the scope formula of the quantified sentences "∃x Gx" and "∀x Gx," we'll say that an instance of "Gx" is (by association) an instance of those quantified sentences as well.

Instance of a Quantified Formula (*First Draft*)

> **The result of removing the quantifier from that formula, and replacing the variable in its scope formula by a name letter.**

So since "Gx" is the scope formula of "∃x Gx" and of "∀x Gx," the above three instances of "Gx" also count as instances of "∃x Gx" and "∀x Gx".

Finally, when we speak of "an instance of a quantified formula **in a model**," we mean: an instance using a name letter which appears in that model. So

[4] The full account of instances is given in 5.8.

[5] 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 like "(Gx → ∃x Jx)". 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's **only one** quantifier in the sentence. Again, the definition will be modified later to scale up to larger sentences.)

the quantified sentences "∃x Gx" and "∀x Gx" have three instances in the above model (since that model features three 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: the sentence "∃x Gx" is **true** in a model if there's at least one object (from that model's domain) 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** otherwise – that is, if there are no objects in the extension of "G" (if **no objects in the model are G**).

But whenever at least one object is in the extension of "G" there's at least one true instance of "∃x Gx" – say, "GB," or "GC". So we can restate those last claims in terms of "instances".

> **"∃x Gx" is true in a model if "∃x Gx" has at least one true instance in that model.**
>
> **"∃x Gx" is false in a model 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 two: "GB" and "GC".) 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," and "JC" – are false.)

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

For universal sentences: the sentence "∀x Gx" should be **true** in a model if every object in the domain is in the extension of "G" – that is, **if every object is G**. And "∀x Gx" should **false** otherwise – that is, if even one object in the domain isn't in the extension of "G" (**if even one object isn't G**). Those points are restated in terms of "instances" as follows.

> **"∀x Gx" is true in a model if every instance of "∀x Gx" in that model is true.**
>
> **"∀x Gx" is false in a model 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," and "HC" – is true. But **"∀x Gx" is false** in this model, since "∀x Hx" has at least one false instance (namely: "GA").

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

3. Quantifier Negation. So far the scope formulas treated semantically have all been atoms – e.g., "Gx" or "Hx" – semantics for complex quantified sentences waiting upon the further technical details of instances. But already we can account for the truth and falsehood of scope formulas which are negations of atoms – for example, "~Gx" or "~Hx".[6]

The semantic rule for negations dictates that if a certain instance of, say, "Gx" is true in a model, then the negation of that instance is false in that model.

[6] Because the negation of such an atom still only uses one variable to be replaced in an instance; so such negations don't need to wait on the details of more complicated instances, given in 5.8.

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".

			Instances of "Gx":	**Instances of "~Gx":**
	𝔻: {**2**, **3**, **4**}			
A: **2**	**G**: {**3**, **4**}	**I**: {**4**}	GA: **0**	GA: **1**
B: **3**	**H**: {**2**, **3**. **4**}	**J**: { }	GB: **1**	GB: **0**
C: **4**			GC: **1**	GC: **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's no coincidence that both "∃x ~Gx" and "~∀x Gx" are true together. In fact, any model making one of these sentences true makes the other true as well. 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 after the tilde (for example, "GA") false. That means "∀x Gx" has at least one false instance, so "∀x Gx" 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. That makes sense intuitively: not everything is G if and only if something is non-G.[7]

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.)

---

[7] [Taking for granted that there's anything at all. Considering a situation where there aren't **any** objects, we might agree that "Not everything is G" ("~∀x Gx") – indeed, that nothing is – but still deny that there's something non-G ("∃x ~Gx"). Faced with such an empty domain, the intuitive equivalence between "~∀x Gx" and "∃x ~Gx" would seem to break down. The earlier semantic stipulation that the domain can't be empty thus satisfies our intuitions here.

(But that our intuitions might balk at equating "Not everything is G" and "Something is non-G" in an empty model is not to say that formal semantics would. So before celebrating too unreservedly here, see X.xx for unintuitive judgments the formal semantics is willing to accept.)]

Together these equivalences make up the law of **Quantifier Negation**.

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 with the double negation "~~∃xGx" 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.[8]

Likewise, by way of quantifier negation and double negation **"~∃x ~Gx" is logically equivalent to "∀xGx"**.[9]

That means any sentence we translate with one quantifier could be translated with the other instead. We can construct a miniature (one-predicate) **Square of Opposition** illustrating these equivalences.[10]

[8] So our formal language could translate all the same English sentences with just the universal quantifier – translating "some" as "~∀x~".
[9] So our formal language could translate all the same English sentences with just the existential quantifier – translating "all" as "~∃x~".
[10] Adapting the more traditional two-predicate Square of Opposition discussed earlier in 4.X.

Mini-Square of Opposition

(A) Everything is G	**(E) Nothing is G**
$\forall x\, Gx$	$\forall x \sim Gx$
$\sim\exists x \sim Gx$	$\sim\exists x\, Gx$
(I) Something is G	**(O) Something is not G**
$\exists x\, Gx$	$\sim\forall x\, Gx$
$\sim\forall x \sim Gx$	$\exists x \sim Gx$

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 item 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 non-G". And "$\sim\forall x\, Gx$" and "$\forall x \sim Gx$" likewise 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 "∃x Gx" has at least one true instance in that model.

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

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

  "∀x Gx" is true in a model if every instance of "∀x Gx" in that model is true.

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

- **Quantifier Negation**:

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