5.8. Quantifier Semantics for QBSs: Instances

1. Instances. Recall that a **variable basic** is either a **variable atom** – a predicate letter followed by a variable – **or the negation of a variable atom**. So "Gx" and "Hy," are variable atoms; and they and their negations are variable basics: "Gx," "~Gx," "Hy," "~Hy," and so on.

And a **quantified basic** is the result of attaching a quantifier to the left of such a variable basic (using the same variable in both quantifier and variable basic). So each variable basic yields two quantified basics: from "Gx" we get "∃x Gx" and "∀x Gx"; from "~Gx" we get "∃x ~Gx" and "∀x ~Gx"; from "Hy" we get "∃x Hy" and "∀x Hy"; and so on.

While quantified basics qualify as formal sentences in the construction rules, we didn't bother counting variable basics as formal sentences. Rather, variable basics were established by a definition on the side (like the definition of "sentence letter" or "name letter"). And in fact even in later construction rules we'll continue to deny that a string of symbols such as "Gx" or "Hy" is a sentence, for reasons set out earlier.[1]

A variable atom such as "Gx" corresponds to an English mini-sentence such "It's from Pennsylvania". And such a mini-sentence **doesn't make a complete claim** the way a normal sentence would; for I can say truly that "it's from Pennsylvania" while pointing at the Cathedral of Learning, but say something false by uttering "it's from Pennsylvania" while pointing at Jack's new surfboard. A pronoun such as "it" is a varying, recyclable pointer that can point at different things from one moment or context to the next; so without outside help (such as a pointing finger) "it's from Pennsylvania" doesn't make a complete claim. And since "x" is the formal counterpart to pronouns such as "it," "Gx" likewise doesn't make a complete claim, and won't (without outside help) count as true or false in a particular situation (or model).

That fussy reminder is relevant to the semantics of quantified sentences. For intuitively quantifier semantics seems simple enough: a **universal** sentence is true in a certain situation if what the sentence says is true of **every object**

[1] In 5.4 §2.

in that situation; whereas an **existential** sentence is true if what it says holds true of **at least one object** in that situation.

And in each case the "what it says" is just what follows the quantifier. So "Everything is physical" – reworded as "For every object, the following holds true of it: it is physical" – is true just when "It is physical" is true of every object in the situation: true when calling the first object "it," true when calling the second object "it," and so on. Likewise the formal sentence "$\forall x\, Gx$" is true in a model just where "Gx" is true of every object in the domain – calling the first object "x," calling the second object "x," etc.

Yet that conflicts with our earlier claim that "Gx" isn't a candidate for truth or falsehood, since it's not a complete-claim-maker. If what follows the quantifier isn't capable of truth, it can't make true claims about anything.

Here a peculiarity of our semantics comes to the rescue: we required that **every object in the domain has a name**. This guarantees that an object's having or lacking a feature will be reported by a formal sentence: a **name atom**, built from a predicate letter and the object's name letter.[2] And we already know how to evaluate truth in a model for that sort of sentence.

For instance, in the following model we intuitively expect the sentence "$\forall x\, Hx$" to be true, since every object in the domain is H. Whereas we expect the sentence "$\forall x\, Gx$" to be false here, since not every object in the domain is G. (Object **2** isn't G.)

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

a: **2**	**G**: {**3**, **4**}	**I**: {**4**}
b: **3**	**H**: {**2**, **3**, **4**}	**J**: { }
c: **4**		

Now since every object in the model has a name, those intuitive observations can be restated in terms of sentences.

[2] Or one of the object's name letters, if the object has more than one name.

For example: since every object that's H here has a name – object **2** is named "a", **3** is named "b", and **4** is named "c" – the **sentences "Ha", "Hb", and "Hc" are all true** here. Likewise, since not every object in the domain is G (object **2** isn't G), and that object has a name (object **2** is named "a"), we're guaranteed that **"Ga" is false** in this model.

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

a: **2**	**G**: {**3**, **4**}	**I**: {**4**}
b: **3**	**H**: {**2**, **3**, **4**}	**J**: { }
c: **4**		

Moral: **an object's having or failing to have a certain feature is sure to be reflected in the truth or falsehood of a name atom**.

But consider: "Ga," "Gb," and "Gc" are each just what follows the quantifier in "∀x Gx" – namely, "Gx" – but **with a name letter in place of the variable** "x".

Replacing the variable in "Gx" with a name letter yields what we'll call an **instance** of "Gx". So "Ga" and "Gb" are each an instance of "Gx". And by association "Ga" and "Gb" likewise count as instances of any quantified sentence where "Gx" follows the quantifier: "∀x Gx" and "∃x Gx".

Instance of a Quantified Sentence (for QBSs):

> **An instance of a quantified sentence is the sentence resulting from (i) removing the quantifier and (ii) (in the variable basic that remains) replacing the variable with a name letter.**

Moreover, when we speak of "an instance of a quantified sentence **in a model**," we mean: an instance using a name letter which appears in that model. So the quantified sentences "∃x Gx" and "∀x Gx" each have three instances in our model – "Ga," "Gb," and "Gc" – since "a," "b," and "c" are the three name letters that appear in that model.

Variable Basic:	**Instances of This Variable Basic (in this model)**:
Gx	Ga Gb Gc

	$\mathbb{D}$: {**2**, **3**, **4**}		**Instances of "Gx"**:
a: **2** **b**: **3** **c**: **4**	**G**: {**3**, **4**} **H**: {**2**, **3**, **4**}	**I**: {**4**} **J**: { }	Ga: **0** Gb: **1** Gc: **1**

With the notion of "instance" in hand, it's easy to get correct results for the truth and falsehood of quantified sentences.

For universal sentences: intuitively, 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" (**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 model (repeated here), 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: “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**

For existential sentences: the sentence “∃x Gx” is **true** in a model if there’s **at least one** true instance of “∃x Gx” in that model – say, “Gb,” or “Gc”. So we state the semantics of “∃x Gx” in terms of its instances like so.

> **“∃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**.

In our model **“∃x Ix” is true**, since the sentence has at least one true instance in the model: “Ic”. But **“∃x Jx” is false** in this model, since “∃x Jx” has not even one true instance. (Every instance of “∃x Jx” in this model – “Ja,” “Jb,” and “Jc” – is false.)

			Instances of “Ix”:	Instances of “Jx”:
	𝔻: {**2**, **3**, **4**}			
a: **2**	**G**: {**3**, **4**}	**I**: {**4**}	Ia: **0**	Ja: **0**
b: **3**	**H**: {**2**, **3**, **4**}	**J**: { }	Ib: **0**	Jb: **0**
c: **4**			Ic: **1**	Jc: **0**

2. Quantifier Negation. It's simple to extend this account to negations of variable atoms, such as "~Gx" and "~Hy". Here again, an instance is just that variable basic, but with a name letter in place of the variable. For example, "~Ga" and "~Gb" are instances of "~Gx". And then the semantic rule for negations yields the truth value for each such negation: "~Ga" is true when "Ga" is false, and false when "Ga" is true.

So in our earlier model (repeated here): since **"Ga" is false** in that model, **"~Ga" is true** there.

			Instances of "Gx":	**Instances of "~Gx":**
	$\mathbb{D}$: {**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**

That means **"∃x ~Gx" has a true instance** in this model. So **"∃x ~Gx" is true** here.

Note that **"~∀x Gx" is** also **true** here; and it's no coincidence that both "∃x ~Gx" and "~∀x Gx" are true together. For it turns out **any model** making one of these sentences true makes the other true as well. If "∃x ~Gx" is true in a model, that's because there's at least one true instance of "~Gx" – for example, "~Ga". And a model making "~Ga" true makes "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: **something's 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 make up the law of **Quantifier Negation**.

Quantifier Negation

"$\sim\exists x\, Gx$" **is equivalent to** "$\forall x \sim Gx$"

"$\sim\forall x\, Gx$" **is equivalent to** "$\exists x \sim Gx$"

Moreover, since "$\forall x \sim Gx$" and "$\sim\exists x\, Gx$" are logically equivalent, their negations are equivalent as well. That is: "$\sim\forall x \sim Gx$" and "$\sim\sim\exists x\, Gx$" are logically equivalent. And of course the double negation "$\sim\sim\exists xGx$" is equivalent to "$\exists xGx$". So: **"$\sim\forall x \sim Gx$" is logically equivalent to "$\exists x\, Gx$"**. In effect: we can **define** the existential quantifier in terms of the universal and tildes.[3]

Likewise, by way of quantifier negation and double negation **"$\sim\exists x \sim Gx$" is logically equivalent to "$\forall x Gx$"**.[4]

That means any sentence we translate with one quantifier could be translated with the other instead.[5] We can construct a miniature (one-predicate) **Square of Opposition** illustrating these equivalences.[6]

[3] So our formal language could translate all the same English sentences even if we threw out the existential quantifier – translating "some" as "$\sim\forall x\sim$".
[4] So our formal language could translate all the same English sentences even if we threw out the universal quantifier – translating "all" as "$\sim\exists x\sim$".
[5] In the terminology of Chapters 2 and 3: both languages $\{\sim, \wedge, \vee, \rightarrow, \leftrightarrow, \exists\}$ and $\{\sim, \wedge, \vee, \rightarrow, \leftrightarrow, \forall\}$ are **expressively equivalent** to the Chapter Five language $\{\sim, \wedge, \vee, \rightarrow, \leftrightarrow, \exists, \forall\}$.
[6] Adapting the more traditional two-predicate Square of Opposition already encountered 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**
$\exists x\, Gx$	$\sim\forall x\, Gx$
$\sim\forall x \sim Gx$	$\exists x \sim Gx$

We noted earlier of sentences diagonal from one another – for instance, "Nothing is G" and "Something is G" – that whenever one is true the other is false. We can now explain this through simple negation semantics (and bivalence). For instance: "$\sim\exists x\, Gx$" (translating "Nothing is G") is the **negation** of "$\exists x\, Gx$" (translating "Something is G").

The sentences on the right also highlight 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 can't be casual about which formal item comes first – the tilde or the quantifier – since switching the two drastically changes the claim being made.

Second, we can typically follow the order of negation and quantifiers in English to get the scope right 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 (for QBSs)

- **Instance of a Quantified Sentence**:

  An **instance of a quantified sentence** is the result of (1) removing the quantifier and (2) replacing the variable (in the variable basic that remains) by a name letter.

- **Existential Semantics**:

  "∃x Gx" is true in a model if "∃x Gx" has **at least one true instance** in that model (that is: a true instance using a name letter appearing in that model).

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

- **Universal Semantics**:

  "∀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 Gx**" **is equivalent to** "**∀x ~Gx**"

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