5.8. Quantifier Semantics Revisited: Instances

1. Instances. We have so far examined only instances of very simple quantified sentences, such as "∀x Gx" and "∃y ~Gy," where the scope formula is an atom or negation of an atom. But the full treatment of sentence instances, holding even for complex scope formulas, requires more attention to detail than those simple cases did. In order to ward off logical missteps, we here impose general conditions on which sentences qualify as an instance of a quantified sentence, no matter how complex that quantified sentence.

Previously an instance of a quantified sentence was constructed by (i) **removing the quantifier** from the left of that sentence, then (ii) **replacing the variable of quantification** (the variable appearing in the just-removed quantifier) with a name letter in the scope formula that remains. For example, with "∃x Gx" we remove "∃x," leaving just the scope formula "Gx"; and then replace the "x" in "Gx" with a name letter – yielding, say, "GA" or "GB".

Scope Formula:	**Instances of This Formula**
Gx	GA
	GB
	GC
	(etc.)

We add the following three conditions to the second part of that procedure, where name letters replace variables.[1]

(A) Replace **all free** occurrences of the variable with name letters.
(B) Replace **only free** occurrences of that variable with name letters.
(C) Replace all free occurrences of that variable with the **same** name letter.

The reason for imposing condition **(A)** is obvious: in a given model, we want the truth or falsehood of a quantified sentence to depend on the truth or falsehood of its instances. But since only sentences (not quasi-sentences) are capable of being true or false, an instance of a quantified sentence had better contain **no** free variables.

---

[1] A sentence logic version of these conditions was discussed in 3.17.1 Problem E.

Were an instance to retain any free occurrence of a variable, it would fail in its role as a semantic stand-in for the scope formula.

An illustration of condition **(B)** – that only **free** occurrences of a variable be replaced with name letters – comes in the following existential claim.

For some object, x: that object's a cat, but there's something that isn't a cat.

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

Consider the following model, containing just Neko, who's a cat, and Rex, who isn't.

A: Neko **G__**: is a cat
B: Rex

D: {**Neko**, **Rex**}

A: **Neko** G: {**Neko**}
B: **Rex**

Condition (B) has us construct the following instances of sentence (1).

(2) $(GA \wedge \exists x \sim Gx)$
(3) $(GB \wedge \exists x \sim Gx)$

And we expect existential sentence (1) to be true in this model if it has **at least one true instance**.

Now in fact (2) is true in this model. Since Rex isn't a cat here, clearly the right part of (2), "$\exists x \sim Gx$," is true. And since Neko is a cat, the left part of (2), "GA," is true as well. With both parts true, the whole conjunction (2) is true. And since (2) is a true instance of existential sentence (1), **(1) is true** here as well.

That makes intuitive sense: in a situation where Neko's a cat and Rex isn't, it's true to say there's an object (namely, Neko) such that: that object is a cat, but something isn't. **Treating (2) and (3) as instances of (1) yields the right result**.

By contrast, if we disregard Condition (B) (the "only free" restriction) and replace **even bound variable occurrences** in the scope formula of (1), the consequences are less happy. For in that case the instances of (1) would be (4) and (5).

☠ Instances of (1) "∃x (Gx ∧ ∃x ~Gx)" ? ☠

(4) (GA ∧ ∃x ~GA)
(5) (GB ∧ ∃x ~GB)

In order to evaluate the truth of (4) and (5), note first that the right half of each contains a **vacuous quantifier**. The "∃x" in "∃x ~GA" is vacuous because that quantifier has no occurrences of "x" to bind in its scope formula "~GA". Now as noted earlier[2], a vacuous quantifier is semantically empty – making "∃x ~GA" semantically equivalent to "~GA". And the same holds for the right half of (5): "∃x ~GB" is semantically equivalent to just "~GB".

That means that (4) and (5) are semantically equivalent to (6) and (7), respectively.

(6) (GA ∧ ~GA)
(7) (GB ∧ ~GB)

Both these sentences are contradictions, hence **not true in any model**. In that case Sentence (1), "∃x (Gx ∧ ∃x ~Gx)," has no true instances in this (or any) model – the wrong result. So: **violating condition (B)**, and replacing even bound occurrences of a variable, **yields the wrong results**.

The **third** restriction is that we replace the variable of quantification by the **same name letter** throughout.

(C) Replace all free occurrences of that variable with the **same** name letter.

[2] In Section 1 of "5.6. Construction Revisited: Quantifiers, Variables, and Binding".

Whereas (1) didn't look like a contradiction, Sentence (8) does: intuitively, there should be no possible way for (8) to be true.

(8) ∃x (Gx ∧ ~Gx)

To construct an instance of (8), we remove the quantifier "∃x", and replace every free occurrence of "x" in scope formula "(Gx ∧ ~Gx)" with a name letter. Now if, following Condition **(C)**, we use the same name letter throughout, we end up with an instance of the following sort.

(GA ∧ ~GA)
(GB ∧ ~GB)
(GC ∧ ~GC)
(etc.)

Since all of these are contradictions, true in no model, the same holds for (8) – **the correct result**.

But suppose we disregard Condition **(C)**, and replace different occurrences of "x" in "(Gx ∧ ~Gx)" with different name letters. Then Sentence (9) would be treated as an instance of (12).

☠ Instance of "∃x (Gx ∧ ~Gx)" ? ☠

(9) (GA ∧ ~GB)

(9) can certainly be true in a model. In fact the same model as before will suffice: a model with just two objects, cat Neko and non-cat Rex.

(8) ∃x (Gx ∧ ~Gx)

(9) (GA ∧ ~GB)

G__: is a cat

D: {**Neko**, **Rex**}

A: **Neko** **G**: {**Neko**}
B: **Rex**

If (9) counts as an instance of (8), then (8) has at least one true instance in a model, and so isn't a contradiction after all.

(8) $\exists x\,(Gx \wedge \sim Gx)$
(9) $(GA \wedge \sim GB)$

That seems like **the wrong result**. Hence our insistence on Condition (C): when replacing free occurrences of a variable, use the **same name letter** throughout.

2. Quantifier Semantics Revisited. With the official version of "instance" in hand, the semantics for quantified sentences is straightforward.

An **existential** sentence is **true** in a model if (and only if) it has **at least one true instance** in that model.

An **existential** sentence is **false** in a model if (and only if) **every instance** of that sentence is **false** in the model.

A **universal sentence** is **true** in a model if (and only if) **every instance** of that sentence is **true** in the model.

A **universal sentence** is **false** in a model if (and only if) it has **at least one false instance** in that model.

So the following model assigns the truth value listed for each of the quantified sentences below.

D: {**2**, **3, 4**}

A: 2
B: 3
C: 4

G: {4}
H: {3, 4}
I: {2, 3, 4}
J: { }

(10) $\exists x\, Hx$: **1**
(11) $\forall x\, Hx$: **0**
(12) $\exists x\, (Hx \wedge Gx)$: **1**
(13) $(GA \rightarrow \exists x\, Gx)$: **1**
(14) $(GB \rightarrow \exists x\, Gx)$: **1**
(15) $\forall x\, (Ix \rightarrow Hx)$: **0**
(16) $\forall x\, (Hx \rightarrow Ix)$: **1**
(17) $\exists x\, (Jx \rightarrow Gx)$: **1**
(18) $\exists x\, (Jx \leftrightarrow Gx)$: **1**
(19) $\forall x\, (Jx \leftrightarrow Gx)$: **0**
(20) $\forall x\, (Jx \rightarrow Gx)$: **1**
(21) $\forall x\, (Jx \rightarrow {\sim}Gx)$: **1**
(22) $\exists x\, ((Hx \vee Jx) \leftrightarrow Gx)$: **1**
(23) $\forall x\, ((Hx \vee Jx) \leftrightarrow Gx)$: **0**
(24) $\forall x\, ((Hx \vee Ix) \leftrightarrow {\sim}Jx)$: **1**
(25) $\forall x\, ((Hx \vee Ix) \leftrightarrow Ix)$: **1**
(26) $\forall x\, ((Hx \wedge Ix) \leftrightarrow Ix)$: **1**
(27) $\exists x\, ((Hx \wedge Ix) \wedge {\sim}Gx)$: **1**

Summary: Instances and Quantifier Semantics

- **Instance of a Quantified Sentence**: an instance of a quantified sentence is the result of (i) removing the (left-most) quantifier from that sentence and then (ii) in the scope formula that remains, replacing the variable of quantification (the variable appearing in that quantifier) with a name letter, according to the following three constraints.

  (A) Replace **all free** occurrences of the variable.
  (B) Replace **only free** occurrences of the variable.
  (C) Replace occurrences of the variable by the **same** name letter throughout.

- **Existential Semantics**:

  An **existential** sentence is **true** in a model if (and only if) it has **at least one true instance** in that model.

  An **existential** sentence is **false** in a model if (and only if) **every instance** of that sentence is **false** in the model.

- **Universal Semantics**:

  A **universal sentence** is **true** in a model if (and only if) **every instance** of that sentence is **true** in the model.

  A **universal sentence** is **false** in a model if (and only if) it has **at least one false instance** in that model.