

Multiple Quantifiers: Syntax and Semantics

G. J. Matthey

Winter, 2009 / Philosophy 112

Governance

- We have stated that a quantifier governs the shortest full sentence following it.
- Thus far, we have restricted our attention to sentences all of whose components are governed by a single quantifier, if any.
- In the rest of the course, we will consider quantified sentences some of whose quantifiers govern sentences which themselves contain quantifiers.
 - Consider the sentence $(\forall x)(\exists y)Lxy.$
 - $(\exists y)$ governs $Lxy.$
 - $(\forall x)$ governs $(\exists y)Lxy.$
- The use of multiple quantifiers greatly increases the expressive power of Predicate Logic.

Scope and Bound Variables

- We have stated that a quantifier binds all occurrences of its variable in the sentence it governs.
 - $(\exists y)$ governs Lxy and so it binds y in $(\exists y)Lxy.$
- We now state that the **scope** of a quantifier is the sentence governed by the quantifier.
- We state in a preliminary way that a variable **u** is **bound** if and only if it occurs in the scope of a quantifier $(\forall \mathbf{u})$ or $(\exists \mathbf{u})$ containing it.
 - y is bound by $(\exists y)$ in $(\exists y)Lxy.$

Free Variables and Binding

- A variable is **free** if and only if it is not bound.
 - ‘y’ is free in ‘ $(\exists x)Lxy$.’
- An ambiguity might arise when a variable occurs within the scope of two quantifiers containing it.
 - $(\exists x)[(\forall x)Lxa \supset Lxb]$
- Is the ‘x’ in ‘Lxa’ bound by ‘ $(\exists x)$,’ by ‘ $(\forall x)$,’ or both?
- We resolve the ambiguity by saying that a quantifier binds only the free variables in the shortest full sentence following it.
 - ‘x’ in ‘Lxa’ is bound by ‘ $(\forall x)$ ’ only, since it is not free in ‘ $(\forall x)Lxa \supset Lxb$.’
- It may be easiest to think of binding as occurring at the initial application of the quantifier as the sentence is being built up.

Substitution Instances with Multiple Quantification

- The original definition of a substitution instance is inadequate when a variable occurs within the scope of two quantifiers containing it.
- Using the name ‘c,’ how do we instantiate ‘ $(\exists x)[(\forall x)Lxa \supset Lxb]$ ’?
 - $(\forall x)Lca \supset Lcb$?
 - $(\forall x)Lxa \supset Lcb$?
- The answer lies in the fact that the initial ‘x’ in ‘ $(\forall x)Lxa \supset Lxb$ ’ is bound to ‘ $(\forall x)$ ’ and thus should not be replaced by a name.
- The occurrence of ‘x’ in ‘Lxa’ is not free, but is bound to another quantifier, and thus should not be instantiated.
- A substitution instance of a quantified sentence is formed when all free occurrences of the variable in the quantifier which occur in the sentence governed by the quantifier are replaced by names (or more generally, by constant terms).

Semantics for Sentences with Multiple Quantifiers

- The interpretation of sentences with multiple quantifiers is based on an extension of the semantics for sentences with a single quantifier.
- A variable assignment \mathbf{d} satisfies ‘ $(\forall x)(\forall y)Lxy$ ’ just in case for all \mathbf{o}_i in the domain, the x-variants $\mathbf{d}[\mathbf{o}_i/x]$ satisfy ‘ $(\forall y)Lxy$.’
- The x-variant $\mathbf{d}[\mathbf{o}_i/x]$ satisfies ‘ $(\forall y)Lxy$ ’ just in case for all \mathbf{o}_j in the domain, the y-variants $\mathbf{d}[\mathbf{o}_i/x, \mathbf{o}_j/y]$ satisfy ‘Lxy.’
- Thus, we evaluate components of sentences using variants of variants (of variants . . .) until we arrive at atomic sentences.

An Example of Interpretation of a Multiply-Quantified Sentence

- $I = \{D, v\}$, $D = \{\text{Adam}, \text{Eve}\}$, $v(L) = \{\langle \text{Adam}, \text{Adam} \rangle, \langle \text{Adam}, \text{Eve} \rangle\}$
- The sentence whose truth or falsehood in I is to be evaluated is $(\exists x)(\forall y)Lxy$.
- The sentence is true in I , since there is one member of the domain, Adam, who bears the relation indicated by 'L' to every member of the domain.
- That is, $d[\text{Adam}/x, \text{Adam}/y]$ satisfies 'Lxy,' as does $d[\text{Adam}/x, \text{Eve}/y]$, where Adam and Eve comprise all the members of the domain.
- So, $d[\text{Adam}/x]$ satisfies $(\forall y)Lxy$, in which case there is an x-variant of d which satisfies the open sentence $(\forall y)Lxy$, and d satisfies $(\exists x)(\forall y)Lxy$.
- Because the choice of d was arbitrary, the sentence is satisfied by all variable assignments and hence is true.

Another Example of Interpretation of a Multiply-Quantified Sentence

- $I = \{D, v\}$, $D = \{\text{Adam}, \text{Eve}\}$, $v(L) = \{\langle \text{Adam}, \text{Adam} \rangle, \langle \text{Adam}, \text{Eve} \rangle\}$
- The target sentence is $(\forall x)(\exists y)Lxy$.
- We can see by inspection that the sentence is false in I , since there is one member of the domain, Eve, who does not bear the relation indicated by 'L' to at least one member of the domain.
- That is, $d[\text{Eve}/x, \text{Adam}/y]$ does not satisfy 'Lxy,' nor does $d[\text{Eve}/x, \text{Eve}/y]$.
- So, $d[\text{Eve}/x]$ does not satisfy $(\exists y)Lxy$, in which case there is an x-variant of d which does not satisfy the open sentence, and d does not satisfy $(\forall x)(\exists y)Lxy$.
- Because the choice of d was arbitrary, the sentence is satisfied by no variable assignments and hence is false.

Multiple Quantifiers in Substitutional Semantics

- Recall that a universally quantified sentence is true in the substitutional semantics just in case all its substitution instances are true, and an existentially quantified sentence is true just in case at least one of its substitution instances is true.
- For multiple quantification, these truth-definitions are iterated and involve substitution instances of substitution instances.
- For example $(\forall x)(\exists y)Lxy$ is true in an interpretation I if and only if all substitution instances $(\exists y)Lay$, $(\exists y)Lby$, etc. are true in the interpretation.
- This holds just in case at least one substitution instance of $(\exists y)Lay$, i.e., 'Laa,' 'Lab,' etc. is true in the interpretation, and at least one of the substitution instances of $(\exists y)Lby$, i.e., 'Lba,' 'Lbb,' is true in the interpretation, etc.

Logical Equivalence

- Two closed sentences (i.e., sentences containing no free variables) of Predicate Logic are logically equivalent if and only if they have the same truth value in all interpretations.
- For example, $(\forall x)\sim Fx$ is logically equivalent to $\sim(\exists x)Fx$.
- An interpretation **I** makes $(\forall x)\sim Fx$ true if and only if for all **d**, **d** satisfies $(\forall x)\sim Fx$.
- This holds just in case for all **o** in the domain of **I**, the x-variants **d[o/x]** satisfy $\sim Fx$, and thus **d[o/x]** does not satisfy Fx for all **o** in the domain.
- But this holds just in case **d** does not satisfy $(\exists x)Fx$, and hence **d** satisfies $\sim(\exists x)Fx$, which holds for all **d**, QED.

Logical Equivalence for Open Sentences

- We should like to be able to say that $(\forall x)\sim Lxy$ is logically equivalent to $\sim(\exists x)Lxy$.
- However, our definition of logical equivalence covers only closed sentences.
- It is easy to extend the notion of logical equivalence to open sentences.
- Two open sentences are logically equivalent if and only if they are satisfied by exactly the same variable assignments in any interpretation.

An Example of Logically Equivalent Open Sentences

- $(\forall x)\sim Lxy$ is logically equivalent to $\sim(\exists x)Lxy$.
- A variable assignment **d[o_i/y]** satisfies $(\forall x)\sim Lxy$ if and only if for all **o** in the domain, the x-variants **d[o_i/y, o/x]** satisfy $\sim Lxy$.
- This holds if and only if for all **o** in the domain, **d[o_i/y, o/x]** does not satisfy Lxy .
- This holds if and only if **d[o_i/y]** does not satisfy $(\exists x)Lxy$.
- This holds if and only if **d[o_i/x]** satisfies $\sim(\exists x)Lxy$, QED.

A Law of Substitution

- If X is the result of substituting a logically equivalent sentence for a sentence occurring in Y , then X and Y are logically equivalent.
- Here are some examples, with the substituted equivalents underlined:
 - ‘ $(\exists x)(\forall y)\sim Lxy$ ’ is equivalent to ‘ $(\exists x)\sim(\exists y)Lxy$.’
 - ‘ $(\forall x)(\forall y)Fxy$ ’ is equivalent to ‘ $(\forall x)(\forall z)Fxz$.’
- These results are due to the fact that the substituting and substituted sentences are satisfied by exactly the same variable assignments, so the truth-values of X and Y are unaffected by the substitution.

Logical Equivalence and Substitutional Semantics

- Logical equivalence is not straightforwardly defined in substitutional semantics.
- The problem is that open sentences such as ‘ $(\exists x)\sim Lxy$ ’ have no truth-value and are not substitution instances of any sentence.
- One could define a “name-replacement” of an open sentence as a sentence resulting from replacing the occurrences of a free variable with an name, as ‘ $(\exists x)\sim Lxa$.’
- Then two open sentences with one free variable are logically equivalent just in case all name-replacements have the same truth-value.
 - ‘ $(\exists x)\sim Lxa$ ’ is true if and only if ‘ $\sim(\forall x)Lxa$ ’ is true, etc.
- Alternatively, one could hold that two open sentences are equivalent if and only if the universal generalization of their biconditional is a logical truth (true in all interpretations).
 - $(\forall y)[(\exists x)\sim Lxy \equiv \sim(\forall x)Lxy]$.