Predicate Logic Review

John MacFarlane

January 24, 2011

1 Grammar

- A *term* is an individual constant or a variable.
- An *individual constant* is a lowercase letter from the beginning of the alphabet, possibly with a numerical subscript.
- A *variable* is a lowercase w, x, y, or z, possibly with a numerical subscript.
- A *predicate* is a capital letter, possibly with a numerical subscript. Predicates can be classified as one-place, two-place, and in general n-place, depending on how many argument places they have. A logically perspicuous language would mark this, say, with a numerical superscript, but we will normally just leave it implicit.
- A *formula* is any of the following:

  1. $\bot$
  2. An *atomic formula*—an n-place predicate followed by n terms. (For example: Fxy, Ga.)
  3. $\forall \alpha \phi$ or $\exists \alpha \phi$, where α is a variable and ϕ is a formula.
  4. $\neg \phi$, where ϕ is a formula.
  5. $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\phi \supset \psi)$, or $(\phi \equiv \psi)$, where ϕ and ψ are formulas.

  Nothing else is a formula.

Note that in some texts (x) is used instead of $\forall x$. Also, some texts put parentheses around $\forall x$ and $\exists x$. We won't do that.

1.1 Scope

The *scope* of a quantifier is the formula directly following the quantifier:

- In $\forall x(Fx \supset Gx)$, the scope of the quantifier is the formula $(Fx \supset Gx)$.
- In $\forall x Fx \supset Gx$, the scope of the quantifier is the formula Fx.
- In $\forall x \neg Fx \vee Ga$, the scope of the quantifier is the formula $\neg Fx$.

A quantifier $\exists\alpha$ or $\forall\alpha$ will *bind* all occurrences of α within their scopes, except those that are already bound by other quantifiers.

A variable that is not *bound* by a quantifier is called *free*. A formula containing free variables is called an *open formula*. A formula without free variables is called a *closed formula* or *sentence*.

Exercises:

1. Translate the following into logical notation. Provide a dictionary that associates individual constants and predicate letters with English names and predicates, and be sure to specify a domain.
    (a) A man who has not bathed repels every woman he meets.
    (b) Every philosopher trusts some lawyer who has sued one of his (the philosopher's) students.
    (c) Not all lawyers and philosophers are rich.
2. Translate the following into English (provide a dictionary—you may make it up):
    (a) $\neg\exists x(Lx \wedge \forall y(Py \supset Sxy))$
    (b) $\forall x((Fx \wedge \forall y(Gy \supset Hxy)) \supset \exists z(Cz \wedge Lxz))$
3. In each of the following sentences, circle the free variables and draw arrows from each of the bound variables to the quantifier that binds it.
    (a) $\forall x(Fy \supset Gxy) \supset Gyx$
    (b) $\forall x \exists y(Gxy \supset \exists x Gyx)$

2 Semantics

A *model* for our language of predicate logic consists in

- a set of objects—the *domain*, and
- an *interpretation function*, which assigns an interpretation to each individual constant and predicate letter. More specifically, it maps
  - each individual constant to an object in the domain
  - each one-place predicate letter to a set of objects in the domain
  - each two-place predicate letter to a set of ordered pairs of objects in the domain
  - each n-place predicate letter to a set of ordered n-tuples of objects in the domain

In specifying a model, we'll generally only write down the interpretations of individual constants and predicate letters that are relevant for our purposes, omitting the "don't cares."

Here are some examples of models (where F is a one-place predicate, G is a two-place predicate, and a is an individual constant):

1. $D = \{1, 2, 3, 8, \text{Moses Hall}\}$. $I(\text{`}F\text{'}) = \{1, 3\}$, $I(\text{`}G\text{'}) = \{\langle 1, 2\rangle, \langle 3, 3\rangle\}$, $I(\text{`}a\text{'}) = \text{Moses Hall}$.
2. D = the set of integers. $I(\text{`}F\text{'}) = \{x : x > 0\}$, $I(\text{`}G\text{'}) = \{\langle x, y\rangle : x > y\}$, $I(\text{`}a\text{'}) = 1$.
3. $D = \{x : x \text{ is a basketball player}\}$. $I(\text{`}F\text{'}) = \{x : x \text{ is Chinese}\}$, $I(\text{`}G\text{'}) = \{\langle x, y\rangle : x \text{ is taller than } y\}$, $I(\text{`}a\text{'}) = \text{Michael Jordan}$.

(You may not be familiar with the set-theoretic notation we use here. It's fairly simple. $\{1, 2\}$ denotes the set containing 1 and 2. $\{x : x \text{ is } F\}$ denotes the set containing every x such that x is F, that is, the set of Fs. Angle brackets indicate ordered sequences. So, $\langle 1, 2\rangle$ is the sequence consisting of 1 and 2 in that order. $\langle 2, 1\rangle$ is a different sequence, because the order is different. By contrast, $\{1, 2\}$ and $\{2, 1\}$ are the same set, because sets are unordered.)

We can also specify models informally using pictures:

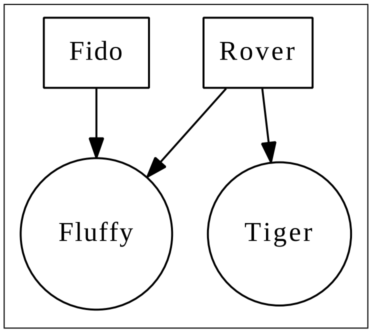

Figure 1: A model of $\forall x(Dx \supset \exists y Cxy)$

We say that a sentence (that is, a closed formula) is *true in a model* just in case it is true when the quantifiers are interpreted as ranging over objects in the domain (and no others) and the individual constants and predicate letters are interpreted as having just the extensions assigned to them by the interpretation function.

To state this condition more precisely, we need the notion of an assignment of values to the variables. An *assignment* is a function that maps each variable to an object in the domain.

We can now specify what it is for an arbitrary formula ϕ (open or closed) to be true in a model $\langle D, I\rangle$ on an assignment a. (Notation: $\vDash^a_{D,I} \phi$, negated: $\nvDash^a_{D,I} \phi$.):

- If ϕ is an atomic formula $F\alpha_1 \ldots \alpha_n$, where F is an n-place predicate and $\alpha_1 \ldots \alpha_n$ are terms, $\vDash^a_{D,I} \phi$ iff $\langle [\![\alpha_1]\!]^a_{D,I}, \ldots, [\![\alpha_n]\!]^a_{D,I}\rangle \in I(F)$, where

$$\begin{array}{rcll} [\![\alpha]\!]^a_{D,I} & = & a(\alpha) & \text{if } \alpha \text{ is a variable} \\ & & I(\alpha) & \text{if } \alpha \text{ is an individual constant.} \end{array}$$

- If ϕ is $\bot$, then $\nvDash^a_{D,I} \phi$.
- If ϕ is $\neg\psi$, for some formula ψ, then $\vDash^a_{D,I} \phi$ iff $\nvDash^a_{D,I} \psi$.
- If ϕ is $(\psi \wedge \chi)$, for some formulas ψ and χ, then $\vDash^a_{D,I} \phi$ iff $\vDash^a_{D,I} \psi$ and $\vDash^a_{D,I} \chi$.
- If ϕ is $(\psi \vee \chi)$, for some formulas ψ and χ, then $\vDash^a_{D,I} \phi$ iff $\vDash^a_{D,I} \psi$ or $\vDash^a_{D,I} \chi$.
- If ϕ is $(\psi \supset \chi)$, for some formulas ψ and χ, then $\vDash^a_{D,I} \phi$ iff $\nvDash^a_{D,I} \psi$ or $\vDash^a_{D,I} \chi$.

- If ϕ is $(\psi \equiv \chi)$, for some formulas ψ and χ, then $\vDash^a_{D,I} \phi$ iff either $\vDash^a_{D,I} \psi$ and $\vDash^a_{D,I} \chi$ or $\nvDash^a_{D,I} \psi$ and $\nvDash^a_{D,I} \chi$.
- If ϕ is $\forall\alpha\psi$, for some variable α and formula ϕ, then $\vDash^a_{D,I} \phi$ iff for *every* assignment a' that agrees with a on the values of every variable except possibly α, $\vDash^{a'}_{D,I} \psi$.
- If ϕ is $\exists\alpha\psi$, for some variable α and formula ϕ, then $\vDash^a_{D,I} \phi$ iff for *some* assignment a' that agrees with a on the values of every variable except possibly α, $\vDash^{a'}_{D,I} \psi$.

Having defined the condition for any open or closed formula ϕ to be true in a model $\langle D, I\rangle$ on an assignment a, we can define truth in a model (not relativized to an assignment) for *closed* formulas as follows:

Truth in a model A closed formula ϕ is true in a model $\langle D, I\rangle$ iff for every assignment a, $\vDash^a_{D,I} \phi$.

(Note: we could have said "some assignment" instead of "every assignment"; it doesn't matter, because if ϕ is a closed sentence, its truth won't vary from one assignment to the next.)

The strategy for defining truth in a model that we have just outlined is due in its essentials to Alfred Tarski. It was one of his great achievements, because it showed logicians how to use *semantic* notions rigorously. Once we've defined truth in a model, of course, we can define logical consequence, logical truth, logical equivalence, logical independence, and so on in the usual way. The definitions are just the same as in propositional logic, only we are now using more complicated models.

Exercises:

1. For each of the three sample models on page 2, above, say which of the following sentences are true in that model:
    (a) $\exists x(Fx \wedge Gxa)$
    (b) $\exists x\exists y(Gxy \wedge Gyx)$
    (c) $\exists x\forall y\neg Gyx$
2. Complete the definitions, using the first line as a paradigm:
    (a) A sentence is *logically true* iff it is true in all models.
    (b) A sentence is *logically false* iff ...
    (c) Two sentences are *logically equivalent* iff ...
    (d) One sentence *logically implies* another iff ...
    (e) A sentence (S) is a *logical consequence* of a set of sentences (Γ) iff ...
    (f) An argument is *logically valid* iff ...
3. Use models to show the following:
    (a) $\exists x\forall yFxy$ and $\forall y\exists xFxy$ are not logically equivalent.
    (b) $(Fa \supset \forall xFx) \supset Fb$ is not a logical truth.
    (c) $Fa \wedge Gb$ does not logically imply $\exists x(Fx \wedge Gx)$.

3 Proofs

3.1 Substitution instances

A **substitution instance** of a quantified formula is the result of deleting the quantifier and its associated variable, then replacing every variable bound by the quantifier with the same individual constant. Thus, for example, $Faab$ is a substitution instance of $\exists xFxxb$ and also of $\forall yFyab$, but *not* of $\forall xFxax$.

Note: An *individual constant* must be substituted for the bound variable. You may not substitute another kind of term, such as a variable or a definite description. (More liberal systems are possible, but the strict rule will simplify things when we get to definite descriptions.)

3.2 Universal instantiation (∀ Elim)

You may write down any substitution instance of any universally quantified formula that is available at your current position in your proof, with the justification "∀ Elim" (citing the line containing the quantified formula). Example:

$$\begin{array}{r|ll} 1 & \forall x\exists yFxay & \text{hyp.} \\ \hline 2 & \exists yFaay & \forall\text{ Elim 1 a/x} \\ 3 & \exists yFbay & \forall\text{ Elim 1 b/x} \end{array} \tag{1}$$

Notes:

1. It is a very good habit to indicate which constant is being substituted for which variable, as in the example.
2. There are no restrictions on which individual constant you use. Just be sure you replace *every* occurrence of the bound variable with the same constant. You can't use ∀ Elim to go from $\forall xFxax$ to $Fbax$, because not every occurrence of the bound variable x was replaced by b.
3. A universally quantified formula is a formula whose *main connective* is ∀. You can't use ∀ Elim to go from $\forall xFx \vee \forall xGx$ to $Fa \vee Ga$, because the former is not a universally quantified formula (the main connective is $\vee$).

3.3 Existential generalization (∃ Intro)

If a substitution instance of an existentially quantified formula is available at your current position in the proof, you may write down the existentially quantified formula of which it is an instance, with the justification "∃ Intro" (citing the line containing the instance). Example:

$$\begin{array}{r|ll} 1 & \forall yFaya & \text{hyp.} \\ \hline 2 & \exists x\forall yFxya & \exists\text{ Intro 1 a/x} \\ 3 & \exists x\forall yFxyx & \exists\text{ Intro 1 a/x} \end{array} \tag{2}$$

Notes:

1. An existentially quantified formula is a formula whose *main connective* is ∃. $\exists xFx \vee Ga$ is not an existentially quantified formula, and it can't be obtained by ∃ Intro from $Fb \vee Ga$.

2. Whereas in ∀ Elim you move from a quantified formula to an instance, in ∃ Intro you move from an instance to a quantified formula.

3. Note that line 1 above is an instance of both 2 and 3.

3.4 Universal generalization (∀ Intro)

You can derive a universally quantified formula $\forall\alpha\phi$ from a subproof whose last step is an instance, with an individual constant in place of α, and whose first step is a *flagging step* containing that individual constant in a box. The justification is "∀ Intro" (citing the lines of the subproof).

A *flagging step* is like a hypothesis, but instead of a formula, it consists of an individual constant in a box:

$$\begin{array}{r|l}
1 & \boxed{\text{a}} \\ \cline{2-2}
2 &
\end{array}$$

There is one important restriction: *the flagged constant may not occur outside of the subproof where it is introduced.* So pick a constant that does not occur in the premises or conclusion or in any previous flagging step.

The flagging step is a formal representation of "Take an arbitrary individual—call it Joe." We then argue that Joe has such and such a property, and since Joe was arbitrary, the same could be shown about any object. The flagging restrictions are there to make sure the individual is really arbitrary, not one that you have already said something about elsewhere in the proof.

Example:

$$\begin{array}{r|l|ll}
1 & \multicolumn{2}{l}{\forall x(Gx \supset Hx)} & \\
2 & \multicolumn{2}{l}{\forall x(Hx \supset Fx)} & \\ \cline{2-2}
3 & & \boxed{\text{a}} & \\ \cline{3-3}
4 & & Ga \supset Ha & \forall\text{ Elim 1 a/x} \\
5 & & Ha \supset Fa & \forall\text{ Elim 2 a/x} \\
6 & & Ga \supset Fa & \text{Taut Con 4, 5} \\
7 & \multicolumn{2}{l}{\forall x(Gx \supset Fx)} & \forall\text{ Intro 3-6 a/x}
\end{array} \tag{3}$$

3.5 Existential instantiation (∃ Elim)

If an existentially quantified formula is available at your current position in the proof, you may start a new subproof with an instance as a hypothesis and the instantial constant "flagged" in a box. You can close the subproof at any point where you have a formula not containing the flagged constant, and reiterate the final formula of the subproof outside of the subproof with justification "∃ Elim", citing the line containing the existentially quantified formula and the whole subproof. As before, the flagged constant may not occur outside of the subproof where it is introduced.

This is easier to see with an example:

$$\begin{array}{lll}
1 & \mid\ \underline{\exists x(Gx \wedge Ha)} & \\
2 & \mid\ \mid\ \underline{Gb \wedge Ha}\ \ \boxed{\text{b}} & \text{b/x} \\
3 & \mid\ \mid\ Gb & \text{Taut Con 2} \\
4 & \mid\ \mid\ \exists x Gx & \exists\ \text{Intro 3} \\
5 & \mid\ \exists x Gx & \exists\ \text{Elim 1, 2–4}
\end{array} \tag{4}$$

Notes:

- We could not have closed off the subproof after line 3, since the flagged constant cannot occur in the last line of the main proof.
- We could not have used a as our flagged term in line 2, since it occurs in line 1.

3.6 Quantifier equivalences (*QE*)

You may use the following substitution rules at any point in a proof, citing "QE" and the line number as justification. Since they are substitution rules, they can operate on subformulas, not just on the main formula. They are also reversible. (See the examples to follow.)

Quantifier equivalences:

- $\neg\forall x\phi \iff \exists x\neg\phi$
- $\neg\exists x\phi \iff \forall x\neg\phi$

Examples:

$$\begin{array}{lll}
1 & \mid\ \underline{\neg\exists x(Gx \wedge Ha)} & \\
2 & \mid\ \forall x\neg(Gx \wedge Ha) & \text{QE 1}
\end{array} \tag{5}$$

$$\begin{array}{lll}
1 & \mid\ \underline{Ha \supset \forall x\neg Gx} & \\
2 & \mid\ Ha \supset \neg\exists x Gx & \text{QE 1}
\end{array} \tag{6}$$

Note that QE is applied to a subformula in example (6). The main connective in (1) is '⊃,' not a quantifier. That's okay, because the QE rules are substitution rules, not rules of inference.

3.7 Tautological Equivalence (Taut Equiv)

What if you wanted to derive $\forall x(Gx \supset \neg Hx)$ from $\neg\exists x(Gx \wedge Hx)$? Given the rules we have so far, you'd have to take a circuitous path:

$$
\begin{array}{lll}
1 & \mid\ \underline{\neg\exists x(Gx \wedge Hx)} & \\
2 & \mid\ \forall x\neg(Gx \wedge Hx) & \text{QE 1} \\
3 & \mid\ \mid\ \underline{\boxed{\text{b}}} & \\
4 & \mid\ \mid\ \neg(Gb \wedge Hb) & \text{$\forall$ Elim, 2, b/x} \\
5 & \mid\ \mid\ Gb \supset \neg Hb & \text{Taut Con, 4} \\
6 & \mid\ \forall x(Gx \supset \neg Hx) & \text{$\forall$ Intro, 3–5, b/x}
\end{array} \tag{7}
$$

To simplify this kind of proof, we introduce a new inference rule, Taut Equiv, that allows you to *substitute* truth-functionally equivalent formulas for each other, even when they occur embedded inside quantifiers or other operators. Then we can do:

$$
\begin{array}{lll}
1 & \mid\ \underline{\neg\exists x(Gx \wedge Hx)} & \\
2 & \mid\ \forall x\neg(Gx \wedge Hx) & \text{QE 1} \\
3 & \mid\ \forall x(Gx \supset \neg Hx) & \text{Taut Equiv 2}
\end{array} \tag{8}
$$

Consider carefully the difference between Taut Con and Taut Equiv. The latter is a substitution rule. It can operate on subformulas, and it is reversible (you can move either way between equivalent formulas). Taut Con, by contrast, operates only on whole formulas, and moves only in one direction.

Exercises:

1. Use Fitch-style natural deductions to prove the following theorems:
    - (a) $\neg\exists x(Fx \wedge Gx) \supset (\forall xFx \supset \neg\exists xGx)$
    - (b) $\exists x\forall y\forall zFxyz \supset \forall y\forall z\exists xFxyz$
2. Use Fitch-style natural deductions to prove $\forall x(Px \supset \exists yLyx)$ from hypotheses $\forall x(Px \supset \exists yFyx)$ and $\forall x\forall y(Fyx \supset Lyx)$.