24.400

Proseminar in philosophy I

Fall 2003

<u>The propositional calculus</u>

The language L_p contains the following symbols:

Sentence letters: p, q, r, ...

Logical connectives: $\supset$, $\leftrightarrow$, &, $\vee$, $\neg$

Brackets: (,)

The sentences of L_p are given by these rules:

(i) Any sentence letter is a sentence.

(ii) If Φ and Ψ are sentences, then so are:

$\ulcorner(\Phi\supset\Psi)\urcorner$, $\ulcorner(\Phi\leftrightarrow\Psi)$, $\ulcorner(\Phi\&\Psi)\urcorner$, $\ulcorner(\Phi\vee\Psi)\urcorner$, $\ulcorner\neg\Phi\urcorner$.

(iii) Nothing else is a sentence.

An *interpretation* of L_p is an assignment of either the value true, or the value false, to every sentence letter of L_p. Such an assignment uniquely determines the truth value of any non-atomic sentence of L_p in accordance with the following truth table.

Φ	Ψ	$\ulcorner(\Phi\supset\Psi)\urcorner$	$\ulcorner(\Phi\leftrightarrow\Psi)\urcorner$	$\ulcorner(\Phi\&\Psi)\urcorner$	$\ulcorner(\Phi\vee\Psi)\urcorner$	$\ulcorner\neg\Phi\urcorner$
T	T	T	T	T	T	F
T	F	F	F	F	T	F
F	T	T	F	F	T	T
F	F	T	T	F	F	T

<u>The first-order predicate calculus (with identity)</u>

The language L_q contains the following symbols:

Logical connectives: $\supset$, $\leftrightarrow$, &, $\vee$, $\neg$

(Individual) variables: x, y, z, x′, y′, z′, x′′, ...

Names: a, b, c, a′, ...

One-place predicate letters: F_1, G_1, H_1,...

Two-place predicate letters: F_2, G_2, H_2,...

Three-place predicate letters, etc.

A special purpose two-place predicate letter: =

Brackets: (,)

(We can also have function symbols, omitted for simplicity.)

The sentences of L_q are given by these rules:

(i) If Π is an n-place predicate letter, and $\alpha_1, \ldots, \alpha_n$ are n names, $\ulcorner \Pi\alpha_1 \ldots \alpha_n \urcorner$ is a sentence. (We make an exception for our special purpose two-place predicate '='. Here we put a name on either side of it to make a sentence.)

(ii) If Φ and Ψ are sentences, then so are:

$\ulcorner(\Phi\supset\Psi)\urcorner$, $\ulcorner(\Phi\leftrightarrow\Psi)$, $\ulcorner(\Phi\&\Psi)\urcorner$, $\ulcorner(\Phi\vee\Psi)\urcorner$, $\ulcorner\neg\Phi\urcorner$.

(iii) If Φ is a sentence, and χ is a variable not occurring in Φ, and Φ∗ is the result of substituting χ for at least one occurrence of a name in Φ, then:

$\ulcorner(\forall\chi)\Phi*\urcorner$ and $\ulcorner(\exists\chi)\Phi*\urcorner$ are both sentences.

(v) Nothing else is a sentence.

An (open) *formula* is just like a sentence except that some names have been replaced by variables.

A *model* M of L_q is an ordered pair <D, I>, where D (the *domain*) is a set and I (the *interpretation*) is a function assigning to each name a member of D, and to each n-place predicate a set of ordered n-tuples of members of D ('=' always gets assigned the same set: all ordered pairs of members of D with identical first and second members).

A model determines the truth values of all the sentences of L_q as follows:

(i) An atomic sentence ⌜$\Pi\alpha_1...\alpha_n$⌝ is true in M iff <I(α_1),...,I(α_n)> is a member of I(Π).
(ii) The connectives are treated just as in the propositional calculus: e.g. ⌜(Φ&Ψ)⌝ is true in M iff Φ is true in M and Ψ is true in M.

Order the names of L_q, and let Φ[α/χ] be the result of replacing every occurrence of the variable χ in Φ by the first name α not occurring in Φ. Then:
(iii) ⌜(∀χ)Φ⌝ is true in M iff Φ[α/χ] is true in M, and also in every model which is just like M apart from what gets assigned to α.
(iv) ⌜(∃χ)Φ⌝ is true in M iff Φ[α/χ] is either true in M, or true in at least one model which is just like M apart from what gets assigned to α.

(Alternatively, Tarski-style, (iii) and (iv) can be replaced by clauses using the notion of a *sequence* satisfying a *formula*.)

A *valid* sentence is one true in all models.

<u>The second-order predicate calculus</u>
The language L_{q^2} contains all the following symbols of L_q plus *predicate* variables:
One-place predicate variables: X, Y, Z, X′, Y′, …
Two-place predicate variables: X_2, Y_2, Z_2, X_2', …
(Three-place predicate variables, etc.)

The formation rules are just the same, except that (iii) is replaced by:
(iii†) If Φ is a sentence, and χ is an individual variable not occurring in Φ, and Φ∗ is the result of substituting χ for at least one occurrence of a name in Φ, then:
⌜(∀χ)Φ∗⌝ and ⌜(∃χ)Φ∗⌝ are both sentences.
If Φ is a sentence, and Ξ is an n-place predicate variable not occurring in Φ, and Φ∗ is the result of substituting Ξ for at least one occurrence of an n-place predicate in Φ, then:
⌜(∀Ξ)Φ∗⌝ and ⌜(∃Ξ)Φ∗⌝ are both sentences.

(We can also have function and sentence variables, omitted for simplicity.)

The definition of truth in a model is just the same, except that we add:
Let $\Phi[\Pi/\Xi]$ be the result of replacing every occurrence of the n-place predicate variable Ξ in F by the first n-place predicate letter Π not occurring in Φ. Then:
(iii) $\ulcorner(\forall\Xi)\Phi\urcorner$ is true in M iff $\Phi[\Pi/\Xi]$ is true in M, and also in every model which is just like M apart from what gets assigned to P.
(iv) $\ulcorner(\exists\Xi)\Phi\urcorner$ is true in M iff $\Phi[\Pi/\Xi]$ is either true in M, or true in at least one model which is just like M apart from what gets assigned to P.

Exercise: show that '$\forall x\exists X(Xx)$' is valid.

Various "logical" notions can be defined in second-order but not first-order logic:
a is identical to b: $\forall X(Xa \leftrightarrow Xb)$ [or $\forall X(Xa \supset Xb)$]
The "ancestral" of a relation R (*Grundlagen*, §79)
There are exactly as many Fs as Gs (§72)
There are (denumerably) infinitely many things

This greater expressive power comes at some cost: unlike first-order logic, there is no mechanical procedure that, when applied to a sentence of second-order logic, will deliver a 'yes' if the sentence is valid.

According to Quine, second-order logic is not logic at all, but rather "set theory in sheep's clothing" (*Philosophy of Logic*, 66). For the opposing view, see Boolos, *Logic, Logic, and Logic*.