7 General First-Order Logic

1. Introduction

In the present chapter, by way of preparing to do general first-order modal logic, we examine ordinary non-modal first-order logic.

We take first-order logic to include the following.

Sentential Logic
Predicate Logic
Function Logic
Identity Logic
Description Logic
Scoped-Terms

Note that the latter is not usually an official part of first-order logic. It represents our technique for reproducing Russell's theory of description within the context of free first-order logic.

We have already treated SL and PL in earlier chapters. In the present chapter, we consider the remaining topics.

As usual, in presenting this material, we will make extensive use of the techniques and notation of categorial grammar.

2. Function Logic; Compound Noun Phrases; Function-Signs

If we symbolize the following sentence

Jay's mother is tall

in predicate logic, we obtain the following formula,

Tm
alt: T[m]

if we use a bracket convention for predicates. The following is our abbreviation scheme.

T[α]	α is tall;
m	Jay's mother.

Similarly, if we symbolize

two plus three is greater than four [2+3 > 4]

in predicate logic, we obtain

Gpf
alt: G[p, f]

where the following is the abbreviation scheme.

G[α,β]	α is greater than β
p	2+3
f	4

These translations are inadequate, because they do not uncover the additional structure contained in the *complex noun phrases* 'Jay's mother' and '2+3'. In order to accomplish this, we need to analyze these noun phrases into the noun phrases of which they are composed. In the case of 'Jay's mother', there is one component noun phrase – 'Jay'. In the case of '2+3', there are two component noun phrases – '2' and '3'.

Both of these expressions are analyzed logically in terms of the notion of *function-sign*, which is a new kind of functor, officially defined as follows.

(d1) A ***function-sign*** is an expression with (zero or more) blanks such that, filling these blanks with noun phrases results in a noun phrase.

As with our earlier functors – connectives and predicates – every function-sign has a degree. This is summarized as followed.

(d2) Where k is any natural number (0, 1, 2, ...), A ***k*-place function-sign** is a function-sign with k many blanks (places).

The following are the various categories of function-signs.

$N^0 \rightarrow N$	0-place function-sign
$N^1 \rightarrow N$	1-place function-sign
$N^2 \rightarrow N$	2-place function-sign
etc.	

Now, let us go back and examine our earlier examples. First, in

Jay's mother is tall

the subject is

Jay's mother.

This is a compound noun phrase, which consists of two grammatical components,

Jay

which is a proper noun, and hence atomic, and

___'s mother,

which is a one-place function-sign. If one fills the blank with a noun phrase, simple or complex, the result is also a noun phrase. It might be an absurd noun phrase, like '33's mother', but nonetheless it is syntactically speaking a noun phrase.

Now, the way we depict function-signs in function logic is very similar to the way we depict predicates. The difference is that we use lower case letters instead of upper case letters, and we use round parentheses instead of square brackets. Thus, for example:

$m(\alpha)$	stands for	α's mother

Therefore, if 'J' stands for the proper noun 'Jay'

$m(j)$	stands for	Jay's mother

So, when we plug back in, we obtain our final formula.

T[m(J)]

In our arithmetic example,

two plus three is greater than four [2+3 > 4]

the two immediate noun phrases are 'two plus three' and 'four'. Whereas the latter is a simple proper noun, the former is a complex noun phrase. In fact, it consists of three grammatical components:

two
three

which are proper nouns, and

___ plus ___

which is a 2-place function-sign, which we abbreviate as follows.

$p(\alpha, \beta)$ α plus β

So, when we plug this back into our formula, we obtain:

G[p(2, 3), 4]

Notice that we use the Arabic numerals in our symbolization, mostly for convenience.

3. Identity Logic

No logical analysis of mathematics, or ordinary language for that matter, would be complete without a logical analysis of such phrases as:

two plus two is four
two plus three is not four
three plus four is seven
etc.

In function logic, we already have the purely grammatical means to symbolize these, as follows.

I[p(2, 2), 4]
~I[p(2, 3), 4]
I[p(3, 4), 7]

Lexicon:

$p(\alpha, \beta)$	α plus β
I[α, β]	α is β

Here, the verb 'is' is represented as a two-place predicate, not unlike 'respects'. The problem, however, is not grammatical, but logical. The following is a valid argument.

2+2 is 4	I[p(2, 2), 4]
4 is even	E[4]
therefore, 2+2 is even	/ E[p(2, 2)]

But its logical form, as depicted to the right, is not a valid form in function logic.

The problem is that 'is', as used in these sentences, is not just any 2-place predicate. It is a very special 2-place predicate, what we would indeed call a *logical predicate*. Our examples are from arithmetic, but 'is' is not peculiar to arithmetic. Arithmetic provides the easiest examples, since everyone is familiar with it. Other examples of this special two-place 'is' include the following.

who is the person standing next to the window?
that is Jay's mother.

who is the president of the U.S.?
George Bush is the President of the U.S.

What these examples have in common is the logical predicate 'is', which is sometimes called:

the 'is' of identity

Notice that in asking someone to *identify* a person, we are asking "who ***is*** that person?" We are using the verb 'is' as the logical notion of identity.

This use of 'is' is in contrast to two other uses.

the 'is' of predication
the 'is' of existence

We are completely familiar with the 'is' of predication. If we say 'Jay's mother *is tall*', the word 'is' is part of the predicate; it is not grammatically autonomous in our analysis. The 'is' of existence is best exemplified by Hamlet's soliloquy "To be, or not to be; that is the question." The 'is' of existence is also apparent in the expression 'there is a thing such that ...' , which we refer to as the *exist*ential quantifier.

Back to the 'is' of identity. Well, we already know how it is symbolized in arithmetic – by the equals sign '='. Logicians have preserved this venerable symbol, and have adopted it to stand for the 'is' of identity. The following is our official syntax.

if σ and τ are singular-terms, then $[\sigma=\tau]$ is a formula.

Notice carefully: Whereas the run-of-the-mill, non-logical, predicates are always written in prefix notation, special, logical, predicates are written in "natural" notation, as in arithmetic. Notice also the appearance of the square brackets; this is a vestige of the bracket notation for predicates. In an important sense, they are completely optional, and are mostly used to help visually parse formulas in which '=' appears. The following are examples.

$\forall x \exists y[x=y]$
$\forall x \exists y \sim[x=y]$
$\forall x \forall y\{x=y \rightarrow (\mathrm{F}x \rightarrow \mathrm{F}y)\}$

Note that the brackets have been dropped in the third example, since they do not visually contribute to parsing.

Also note that the second example can be simplified as follows.

$\forall x \exists y[x \neq y]$

We regard the expression '$x \neq y$' as mere shorthand for '$\sim[x=y]$', for all purposes, including most importantly applying SL-rules like $\vee$O and $\rightarrow$O. For example, the following is a valid application of $\rightarrow$O.

$$\begin{array}{l} \mathrm{F}x \rightarrow x=y \\ x \neq y \\ \hline \sim \mathrm{F}x \end{array}$$

Of course, presenting '=' as a logical predicate is incomplete without also presenting its associated inference rules. In this connection, there are four rules that are usually presented – Reflexivity, Symmetry, Transitivity, and Leibniz's Law.

R=	S=	T=
$\dfrac{}{\sigma = \sigma}$	$\dfrac{\sigma = \tau}{\tau = \sigma}$	$\dfrac{\rho = \sigma \quad \sigma = \tau}{\rho = \tau}$

LL	
$\dfrac{\Phi[\sigma/\nu] \quad \sigma = \tau}{\Phi[\tau/\nu]}$	$\dfrac{\Phi[\sigma/\nu] \quad \tau = \sigma}{\Phi[\tau/\nu]}$

Notice that R= is a zero-place rule. LL is simply the principle of substitution, familiar to anyone who has studied high school algebra. The following are examples.

Jay's mother is tall Jay's mother is the president ——————————— the president is tall	T[m(J)] m(J) = p ——— T[p]
4 is even 2+2 is 4 ——————— 2+2 is even	E[4] 2+2 = 4 ——— E[2+2]

4. Numerical Quantifiers

Not only does the logic of identity help us do arithmetic, which is the science of natural numbers, it also helps us understand the numbers themselves – thought of as *quantities*.

From Intro Logic we already have at least one numerical concept – we can say zero in its most useful form. For example, we can say that there are zero-many unicorns, or that the number of unicorns is, i.e., equals (=), zero, by saying simply that there are no unicorns. The following is the translation from number-talk to quantifier-talk.

the number of unicorns = 0 ⤳ $\sim\exists x\mathrm{U}x$

Similarly, since we can say zero, we can also say not-zero. For example, if the number of unicorns is not zero, then the number of unicorns is 1, or 2, or 3, or 4... (but not 3/4, or –7, or π) In other words, the number of unicorns is *at least one*. Well, of course! We know how to say not-zero, which is the same as at-least-one; we have a quantifier just made for that – the existential quantifier. The following is the translation from number-talk to quantifier-talk.

the number of unicorns ≠ 0
the number of unicorns > 0
the number of unicorns ≥ 1 ⤳ $\exists x\mathrm{U}x$

What about other numerical quantifiers? The following are examples.

there is	***exactly one***	unicorn
there is	***at most one***	unicorn
there are	***at least 2***	unicorns
there are	***at most 2***	unicorns
there are	***exactly 2***	unicorns
etc.		

How do we symbolize these? To make a long story short, the following are the translations of these sentences into the language of identity logic.

#(U)=0	$\sim\exists x\mathrm{U}x$
#(U)=1	$\exists x\forall y(\mathrm{U}y \leftrightarrow y{=}x)$
#(U)=2	$\exists x\exists y(x{\neq}y \;\&\; \forall z\{\mathrm{U}z \leftrightarrow (z{=}x \vee z{=}y)\})$
#(U)=3	$\exists w\exists x\exists y(w{\neq}x \;\&\; w{\neq}y \;\&\; x{\neq}y \;\&\; \forall z\{\mathrm{U}z \leftrightarrow (z{=}w \vee z{=}x \vee z{=}y)\})$
#(U)≥1	$\exists x\mathrm{U}x$
#(U)≥2	$\exists x\exists y(x{\neq}y \;\&\; \mathrm{U}x \;\&\; \mathrm{U}y)$
#(U)≥3	$\exists x\exists y\exists z(x{\neq}y \;\&\; y{\neq}z \;\&\; x{\neq}z \;\&\; \mathrm{U}x \;\&\; \mathrm{U}y \;\&\; \mathrm{U}z)$
#(U)≤1	$\exists x\;\forall y(\mathrm{U}y \rightarrow y{=}x)$
#(U)≤2	$\exists x\exists y\;\forall z\{\mathrm{U}z \rightarrow (z{=}x \vee z{=}y)\})$
#(U)≤3	$\exists x\exists y\exists z\;\forall w\{\;\mathrm{U}w \rightarrow (w{=}x \vee w{=}y \vee w{=}z)\}$

The logical analysis given above was first given by Russell, and provides the inspiration at the basis of Russell and Whitehead's epic *Principia Mathematica* (1910-1913). The analysis of 'exactly

one' is the inspiration at the basis of Russell's outrageously famous theory of definite descriptions.[1] We postpone examining this until we get to descriptions – which we will do shortly.

5. Subnectives

So far, we have talked about three categories of functors:

connectives	$S^k \rightarrow S$
predicates	$N^k \rightarrow S$
function-signs	$N^k \rightarrow N$

Among these simple functor types, there is a notable asymmetry. By way of correcting this theoretical lapse, we introduce the fourth and final simple functor type.

subnectives	$S^k \rightarrow N$

In other words, a subnective takes sentences as input and delivers a noun phrase as output.

Does ordinary language provide any examples of such functors. First, it is plausible to regard each of the following expressions as a one-place subnective.

that ____
"___"
whether ____

Here, we understand the blanks are to filled by sentences. When so filled we have various noun phrases that can be used as subjects or objects of verbs, including the following.

subject	verb (phrase)	direct object
Socrates	said	that all humans are mortal
Socrates	said	"all humans are mortal"
Socrates	did not say	whether all humans are mortal

The first sentence is an example of what is usually called *indirect quotation*, according to which we quote Socrates by *referring to* the proposition he expressed, not by repeating his words. In this example, the direct object

that all humans are mortal

can be grammatically analyzed as follows.

subnective	sentence
that	all humans are mortal

This is in contrast to *direct quotation*, according to which we quote Socrates by repeating his very words. For example, it is highly unlikely that the following is literally true,

Socrates said "all humans are mortal"

[1] Bertrand Russell, 'On denoting', *Mind*, 14 (1905), 479-93.

because it is unlikely Socrates ever uttered this combination of words/sounds. [2]

Nevertheless, direct quotation affords us with another example of a subnective, in accordance with the following grammatical analysis.[3]

subject	verb (phrase)	direct object	
Socrates	said	subnective	sentence
		"__"	all humans are mortal

The word 'whether' is syntactically similar to 'that', and so its analysis is left as an exercise.

6. Definite Descriptions

Whether there are any simple subnectives is perhaps controversial.[4] Description logic, however, does provide an example of a subnective-like functor (just as quantifiers are connective-like functors). The canonical ordinary language expression for the definite description is

the one/unique (person, thing, number, etc.) such that ....

which is symbolized:

ιx

where 'ι' is an upside down iota. Whether 'ιx' (read 'iota x') means 'the person...', or 'the thing...', or 'the number...' will depend upon the domain of discourse.

The critical fact about 'ι' is that, unlike the quantifier functors '$\forall$' and '$\exists$', 'ι' is a subnective-like functor. To be specific, the following is its category.

$(V_0 \times S) \rightarrow N$

In other words, 'ι' takes an individual variable, and a formula, as input, and yields a noun phrase as output. So the expression

$\iota x F x$

is a singular-term, not a formula. We read this expression as:

the (unique) x such that Fx
or: the one who is F
or simply: the F

In order to use this complex noun phrase to produce a formula, we need a predicate. The following are examples.

[2] There are, actually, different ways to repeat someone's "very" words, and so there are different ways to do direct quotation. Socrates may have said words (in Ancient Greek) that translate into English as these very words.
[3] Direct quotation is notably deficient as a subnective, and should rather be called a *pseudo-subnective*. In particular, direct quotation is *completely opaque* semantically.
[4] By the way, how do you analyze this sentence – what is the subject?

T[ι*x*F*x*]	the F is T
R[ι*x*F*x*, K]	the F R's K
ι*x*F*x* = K	the F is K

Notice that a description can appear inside another description. This is the basis of the nursery rhyme "This is the house that Jack built". Consider the following sequence.

t = ι*x*(H*x* & B J*x*)
t = ι*y*{M*y* & L[*y*, ι*x*(H*x* & B J*x*)]}
t = ι*z*{C*z* & C[*z*, ι*y*{M*y* & L[*y*, ι*x*(H*x* & B J*x*)]}]}
etc.

Here, the intended lexicon is as follows.

t	this
J	Jack
H[α]	α is a house
M[α]	α is a mouse
C[α]	α is a cat
L[α, β]	α lived in β
C[α, β]	α chased β
B[α, β]	α built β

7. Russell's Theory of Descriptions

There are three approaches to the logic of descriptions.

Frege's approach (classical first-order logic)
Russell's approach (analyze descriptions so they go away)
Free Logic approach (a compromise between the previous two)

We will discuss just the latter two.

According to Russell's theory of descriptions,[5] any formula with a description, for example,

the F is G

must be *explicated/replaced* by an equivalent formula not explicitly containing the description. The following is Russell's proposed explicating formula.

there is exactly one F, and it is G

This can also be read

there is exactly one F, which is G

which should not be confused with

there is exactly one F that is G.

[5] Russell, "On Denoting", 1905, previously cited.

For the first one to be true, the number of F's must be 1. For the second one to be true, the number of F's may be any number, so long as exactly one of *them* is G.

We ignore the 'that' statement, and concentrate on the 'which' statement.

there is exactly one F, which is G

This can be translated into identity logic as follows.

$\exists x \{ \forall y(Fy \leftrightarrow y=x) \,\&\, Gx \}$

This can be read:

there is an x such that	x is the only F [x is F, and nothing else is]	and x is G

Unfortunately, Russell's analysis faces an immediate problem. What happens when we try to explicate the following formula?

the F is ***not*** G

According to Russell, there are two readings of this, according to how we logically locate the negation; there seem to be two choices as to the logical location of 'not'.

the F is not-G
it is not true that: the F is G

The following are the corresponding Russell formulas.

there is exactly one F, which is not G
it is not true that: there is exactly one F, which is G

And the following are the corresponding formulas.

$\exists x \{ \forall y(Fy \leftrightarrow y=x) \,\&\, \sim Gx \}$
$\sim \exists x \{ \forall y(Fy \leftrightarrow y=x) \,\&\, Gx \}$

The important thing to realize is that these are ***not logically equivalent***. This is regarded as an issue of scope. In one case, the scope of the description is wide; the other case, the scope of the description is narrow (both scopes in relation to 'not'). In other words, the problem is no different from the problem of analyzing the following formula,

every F is not H

which has two readings.

every F is not-H	$\forall x(Fx \rightarrow \sim Hx)$
it is not true that: every F is H	$\sim\forall x(Fx \rightarrow Hx)$

We will get back to the issue of descriptive scope, when we talk about scoped-terms.

8. Free Logic

"Classical" first order logic, whose quantifier principles trace to Frege, is based on the following two presuppositions:

(c1) The domain (universe) of discourse is not empty. Accordingly, the sentence 'there is something' is *logically* true, even though it is not *necessarily* true.

(c2) Every singular-term – no matter how silly – denotes an existing object (i.e., an element of the domain).

Free logic offers an alternative to classical logic that denies both (c1) and (c2).[6] In order to formally implement free logic, we must adjust a number of inference rules. Consider the following dual-pair of rules of classical logic.

∀O	∃I
$\forall\nu\Phi$ / $\Phi[\tau/\nu]$	$\Phi[\tau/\nu]$ / $\exists\nu\mathbb{F}$
Φ is any formula, ν is any variable, τ is any closed singular-term, and $\Phi[\tau/\nu]$ results from Φ when τ replaces every occurrence of ν that is free in Φ.	

The main point is that, according to classical logic, any singular-term τ can be substituted for the variable ν. So, the following are examples.

$\forall x\mathrm{H}x$ / $\mathrm{H}[m(\mathrm{J})]$	$\mathrm{H}[f(\mathrm{K})]$ / $\exists x\mathrm{H}x$

By contrast, free logic requires an additional premise for both ∀O and ∃I, as follows.

∀O(f)	∃I(f)
$\forall\nu\Phi$ $\mathrm{E!}[\tau]$ $\Phi[\tau/\nu]$	$\Phi[\tau/\nu]$ $\mathrm{E!}[\tau]$ $\exists\nu\mathbb{F}$

Here, the predicate 'E!' is the special logical predicate of "existence". If one has the resources of identity logic, then one can simply define existence as follows.

[6] Technically, there are two different versions of free logic. The more radical version (Universally Free Logic) denies both (c1) and (c2). The less radical version of Free Logic rejects (c2), but accepts (c1). We are principally interested in the more radical version.

Def E!
$\mathrm{E}![\tau]$ $\overline{\overline{\exists\nu[\nu = \tau]}}$

Here, τ is any closed singular-term, and ν is any variable.

The easiest way to implement free logic inside our natural deduction system is adjust the $\forall$O and $\exists$I rules as follows.

$\forall$O	$\exists$I
$\forall\nu\Phi$ ——— $\Phi[o/\nu]$	$\Phi[o/\nu]$ ——— $\exists\nu\mathbb{F}$
c must be an ***old constant***, which is to say a constant that appears in a line that is *neither boxed nor cancelled*.	

In order to reconstruct the original free logic quantifier forms, one must use the definition of 'E!', $\exists$O, and Leibniz's Law. The following schematic derivations show that the original free logic quantifier rules are admitted in our system of natural deduction.

(1)	$\forall\nu\Phi$	Pr
(2)	$\mathrm{E}![\tau]$	Pr
(3)	~~SHOW~~: $\Phi[\tau/\nu]$	DD
(4)	$\exists x[x = \tau]$	2, Def E!
(5)	$c = \tau$	4,$\exists$O
(6)	$\Phi[c/\nu]$	1,$\forall$O
(7)	$\Phi[\tau/\nu]$	5,6,LL

(1)	$\Phi[\tau/\nu]$	Pr
(2)	$\mathrm{E}![\tau]$	Pr
(3)	~~SHOW~~: $\exists\nu\Phi$	DD
(4)	$\exists x[x = \tau]$	2, Def E!
(5)	$c = \tau$	4,$\exists$O
(6)	$\Phi[c/\nu]$	1,5,LL
(7)	$\exists\nu\Phi$	6,$\exists$I

9. Proper Nouns in Free Logic

This brings us to a fresh problem. In *elementary* predicate logic, we use constants ('a', 'b', etc.) in two quite different ways. First, they are used to abbreviate certain simple noun phrases (proper nouns). They are also used in derivations, in connection with UD and ∃O. In classical logic, the distinction is unimportant. The reason is that, in classical logic, every singular-term, and hence every proper noun, automatically refers to an object in the domain.

In free logic, we can no longer afford this ambiguity. In free logic, whereas every variable and every constant (unquantified variable) refers to an element of the domain, proper nouns may be *improper*, which is to say they refer to nothing whatsoever.

Accordingly, in doing derivations in free logic, we must carefully distinguish between constants and proper nouns. This can be accomplished in a number of ways.

1. First Way

We can just be careful – specifically, by observing the following simple rule.

(R!) Constants are exclusively derivational constructs, and only come into existence inside of derivations – in particular, in association with UD and ∃O.

This rule has two corollaries – one logical, one practical.

(c1) No lower case letter that is used in (the symbolization of the) argument to be proven by derivation is a constant; every such letter is a proper noun.

(c2) Do not choose as a constant any letter that serves as a proper noun!

2. Second Way

We can adopt alternative notation. For example, we can use bold-face letters or small-caps for proper nouns, and use ordinary lower case letters for constants and variables.

3. Third Way

We can take advantage of our logical machinery, and put those kooky 0-place function-signs to work. In that case, we symbolize 'Jay' by 'j()' and 'Kay' by 'k()'.

4. Fourth Way

We can enlarge our theoretical syntax to include proper nouns as a primitive sort of expression; for pure logic, this is just like the second way. For applied logic, this is perfect for symbolizing theories that include explicit proper nouns; for example, arithmetic has (at least) the proper noun '0'.

We follow a combination of the first, second, and fourth ways. We have already seen symbolizations in which the native symbols are preserved ('0', '1', etc.) We make this official now. We also do not have to rewrite any of our earlier derivations.

10. Descriptions in Free Logic

The treatment of descriptions in free logic is simply a special case of the treatment of all singular-terms [other than variables/constants]. So, for example, the following arguments are *not valid*.

$\forall x\mathrm{G}x$	$\mathrm{G}[\iota x\mathrm{F}x]$
$\mathrm{G}[\iota x\mathrm{F}x]$	$\exists x\mathrm{G}x$

What is missing, in each case, is the premise.

$\mathrm{E}![\iota x\mathrm{F}x]$	$\exists x[x = \iota x\mathrm{F}x]$

Note that according to Frege's general account of singular-terms, both argument forms are valid, but according to Russell's account of descriptions, whereas the first one is invalid, the second one is valid. To see the latter, let us look at the derivation.

(1)	$\mathrm{G}[\iota x\mathrm{F}x]$	Pr
(2)	~~SHOW~~: $\exists x\mathrm{G}x$	DD
(3)	$\exists y\{\forall x(\mathrm{F}x \leftrightarrow x{=}y)\ \&\ \mathrm{G}y\}$	1, Russell's analysis
(4)	$\forall x(\mathrm{F}x \leftrightarrow x{=}a)\ \&\ \mathrm{G}a$	3,∃O
(5)	$\mathrm{G}a$	4,SL
(6)	$\exists x\mathrm{G}x$	5,∃I

According to the free logic account of descriptions, as we present it, we require the additional premise that ιxFx exists [or that the description 'ιxFx' is proper]. Let us look at that derivation.

(1)	$\mathrm{G}[\iota x\mathrm{F}x]$	Pr
(2)	$\mathrm{E}![\iota x\mathrm{F}x]$	Pr
(3)	~~SHOW~~: $\exists x\mathrm{G}x$	DD
(4)	$\exists x[x = \iota x\mathrm{F}x]$	2,Def E!
(5)	$a = \iota x\mathrm{F}x$	4,∃O
(6)	$\mathrm{G}a$	1,5,LL
(7)	$\exists x\mathrm{G}x$	8,∃I

In free logic, we are not entitled to make Russell's move without the additional premise that ιxFx exists. To see how this additional premise also yields the Russell formula, consider the following derivation.

(1)	$\mathrm{G}[\iota x\mathrm{F}x]$	Pr
(2)	$\mathrm{E}![\iota x\mathrm{F}x]$	Pr
(3)	~~SHOW~~: $\exists y\{\forall x(\mathrm{F}x \leftrightarrow x{=}y)\ \&\ \mathrm{G}y\}$	DD
(4)	$\exists x[x = \iota x\mathrm{F}x]$	2, Def E!
(5)	$a = \iota x\mathrm{F}x$	4,∃O
(6)	$\forall x(\mathrm{F}x \leftrightarrow x{=}a)$	5,ιO
(7)	$\mathrm{G}a$	1,5,LL
(8)	$\forall x(\mathrm{F}x \leftrightarrow x{=}a)\ \&\ \mathrm{G}a$	6,7,SL
(9)	$\exists y\{\forall x(\mathrm{F}x \leftrightarrow x{=}y)\ \&\ \mathrm{G}y\}$	8,∃I

The key line is (6), at which point the ιO rule is applied. The two ι rules are given as follows.

ιO	ιI
$c = \iota\nu\Phi$ ——————— $\forall\nu(\Phi \leftrightarrow \nu{=}c)$	$\forall\nu(\Phi \leftrightarrow \nu{=}c)$ ——————— $c = \iota\nu\Phi$
c must be a ***constant***.	

11. Scope

Scope is perhaps the single greatest contribution of modern logic to grammar. We see it in a couple of places in predicate logic. First, the sentence

everyone respects someone

and its passive counterpart

someone is respected by everyone

both have the same grammatical form, given as follows.

R[∀, ∃]

When we go to transform this formula via quantifier-movement, we have a choice concerning which quantifier to do first, and which quantifier to do second. In particular, we obtain the following formulas.

$\forall x\, \exists y\, Rxy$
$\exists y\, \forall x\, Rxy$

The difference between these two formulas concerns which quantifier has wide scope, and which has narrow scope.

But the fact that the two admissible transformations are not equivalent strongly suggests that the original sentence is ambiguous. This can be clarified by saying that the original sentence is ambiguous between the following two readings.

everyone respects someone *or other*
everyone respects someone – *the same someone*

The latter can also be stated as follows.

there is someone ... whom everyone respects

Another place where scope issues arise is in sentences containing the quantifier 'any'. We know that 'any' and 'every' are both universal quantifiers. In fact, we translate '<u>every</u>one is H' as 'for <u>any</u> person x, x is H'. On the other hand, 'any' and 'every' are **not interchangeable**! Consider the following pair of sentences.

if **every**one can fix your car, then I can
if **any**one can fix your car, then I can

Clearly, these are not equivalent; the first one is trivially true; the second one is genuine bragging.

Also, consider the following pair of sentences.

Jay does not respect **every**one
Jay does not respect **any**one

Once again, these are not equivalent; the first one is plausibly true, no matter who Jay is; the second one is only true if Jay is a misanthrope.

Finally, consider the following pair of sentences.

no one respects **every**one
no one respects **any**one

Once again, these are not equivalent. The first one is plausibly true [you cannot find a single person who respects everyone]. The second one is in fact false, since it is easy to find at least one person who respects at least one person.

Now, the analysis of 'any' basically portrays it as a universal quantifier that usually comes attached to a correlative logical expression, with respect to which it ('any') has wide scope. This analysis works fine for 'if any' and 'not any'; it does not work so well for 'no...any', which is best treated *sui generis* – as a whole new class of (double) quantifier.

So, for example, we have the following logical analyses.

if everyone is F, then so is Jay	$\forall x\mathrm{F}x \rightarrow \mathrm{F}\mathrm{J}$	$\rightarrow$ is wide	$\forall$ is narrow
if anyone is F, then so is Jay	$\forall x(\mathrm{F}x \rightarrow \mathrm{F}\mathrm{J})$	$\forall$ is wide	$\rightarrow$ is narrow
Jay does not respect everyone	$\sim\forall x\mathrm{R}\mathrm{J}x$	$\sim$ is wide	$\forall$ is narrow
Jay does not respect anyone	$\forall x\sim\mathrm{R}\mathrm{J}x$	$\forall$ is wide	$\sim$ is narrow

12. Scoped-Terms

Frege taught us that quantifiers have scope. Russell taught us that certain singular-terms – i.e., descriptions – have scope. In order to correlate Russell's discovery with Frege's discovery, we must re-organize Russell's notation so as to parallel Frege's notation.

In the first-order logical analysis of quantified sentences, ordinary language quantifiers are split between two locations – the grammatical location, and the logical location. Whereas the former pertains to number, person, case, etc., the latter pertains exclusively to scope (logic's contribution!) In the logical analysis, whereas the grammatical position is depicted simply by a variable, the scope position is depicted by the official logical quantifier expression itself (e.g., '$\forall x$'). The latter, in turn, may be regarded as a special variable-binding sentential adverb (one-place connective). The transformational process, in its simplest instance, looks like the following.

$\mathrm{H}[\forall]$	everything is H
$\forall x\mathrm{H}[x]$	everything is such that it is H

The transformation does three things.

(1) the quantifier is raised (in a style similar to Chomsky's 'wh'-movement);
(2) the "deep structure" location of the quantifier is marked by a variable;
(3) this same variable is attached to the quantifier, for cross-referencing.

Now, if we wish to apply the transformational strategy to Russell's descriptions, we proceed pretty much the same way. In its simplest instance, it goes as follows.

H[ι*x*F*x*]	the F is H
(ι*x*F*x*/*y*) H[*y*]	the F is such that it is H

Once again, the transformation does three things – (1) description movement, (2) deep-structure marking, (3) cross referencing. Here, the cross-referencing variable is '*y*', but it could also be '*x*', except that '*x*' might be visually confusing (though not to a computer!). Another visual device is utilized, since descriptions are themselves grammatically complex. We enclose the new sentential adverb in parentheses, and we separate the description from the cross-referencing variable. If we wanted our procedures to be consistent, then we might go back and rewrite quantifier expressions as follows.

(∀/*x*), (∀/*y*), (∃/*x*), (∃/*y*), etc.

But this seems unnecessary.

Now, we are in a position to formalize

the F is not G

so that we show the two different scopes that are possible.

(ι*x*F*x*/*y*) ~ G*y*	(ι*x*F*x*/*y*) has wide scope	~ has narrow scope
~ (ι*x*F*x*/*y*) G*y*	~ has wide scope	(ι*x*F*x*/*y*) has narrow scope

Before we continue, it is important to observe that the notion of description scope can be generalized to all singular-terms. This gives us scoped-terms, which are defined as follows.

(d3) Where τ is any singular-term, and where ν is any variable, the expression (τ/ν) is a scoped-term.

Grammatically speaking, scoped-terms are not singular-terms, but sentential adverbs, like quantifiers. This is formulated in the following grammatical rule.

(R) if Φ is a formula, and (τ/ν) is a scoped-term, then (τ/ν)Φ is a formula.

In reading formulas containing scoped-terms, the following is the general rule.

(τ/ν)Φ τ is such that Φ
it is true of τ that Φ

As it turns out, the general concept of scoped-term affords us grammatical generality, but it does not have much payoff in ordinary first-order logic. Note, however, that it has tremendous payoff in modal logic, where it can be used to clarify the so called *de re/de dicto* distinction.

Grammar is important to logic, but logic is more than grammar. To do logic, we need additionally to formulate inference rules for scoped-terms. As it turns out, the rules are actually quite simple; the in-rule and the out-rule can be combined into a single bi-directional rule – Def (τ/ν) ["t for v"].

Def (τ/ν)
$(\tau/\nu)\Phi$ $\exists\nu\{\nu=\tau \;\&\; \Phi)$
Φ is any formula, ν is any variable, and τ is any closed singular-term.

When we apply this new rule to the special case of descriptions, we can deduce the various Russell formulas concerning scope.

(1)	$(\iota x\mathrm{F}x/y)\mathrm{G}y$	Pr	[wide '$\iota x\mathrm{F}x$']
(2)	~~SHOW~~: $\exists y\{\forall x(\mathrm{F}x \leftrightarrow x=y) \;\&\; \mathrm{G}y\}$	DD	[Russell formula]
(3)	$\exists y\{y = \iota x\mathrm{F}x \;\&\; \mathrm{G}y\}$	1,Def (τ/ν)	
(4)	$a = \iota x\mathrm{F}x \;\&\; \mathrm{G}a$	3,∃O	
(5)	$\forall x(\mathrm{F}x \leftrightarrow x=a)$	4a,ιO	
(6)	$\forall x(\mathrm{F}x \leftrightarrow x=a) \;\&\; \mathrm{G}a$	4b,5,SL	
(7)	$\exists y\{\forall x(\mathrm{F}x \leftrightarrow x=y) \;\&\; \mathrm{G}y\}$	6,∃I	

(1)	$\exists y\{\forall x(\mathrm{F}x \leftrightarrow x=y) \;\&\; \mathrm{G}y\}$	Pr	[Russell formula]
(2)	~~SHOW~~: $(\iota x\mathrm{F}x/y)\mathrm{G}y$	3, Def (τ/ν)	[wide '$\iota x\mathrm{F}x$']
(3)	~~SHOW~~: $\exists y\{y = \iota x\mathrm{F}x \;\&\; \mathrm{G}y\}$	DD	
(4)	$\forall x(\mathrm{F}x \leftrightarrow x=a) \;\&\; \mathrm{G}a$	1,∃O	
(5)	$a = \iota x\mathrm{F}x$	4a,ιI	
(6)	$a = \iota x\mathrm{F}x \;\&\; \mathrm{G}a$	4b,5,SL	
(7)	$\exists y\{y = \iota x\mathrm{F}x \;\&\; \mathrm{G}y\}$	6,∃I	

These two derivations give us the following equivalence

$$(\iota x\mathrm{F}x/y)\mathrm{G}y \;\leftrightarrow\; \exists y\{\forall x(\mathrm{F}x \leftrightarrow x=y) \;\&\; \mathrm{G}y\}$$

By very similar reasoning, we can produce the following equivalences.

$$(\iota x\mathrm{F}x/y)\sim\mathrm{G}y \;\leftrightarrow\; \exists y\{\forall x(\mathrm{F}x \leftrightarrow x=y) \;\&\; \sim\mathrm{G}y\}$$
$$\sim(\iota x\mathrm{F}x/y)\mathrm{G}y \;\leftrightarrow\; \sim\exists y\{\forall x(\mathrm{F}x \leftrightarrow x=y) \;\&\; \mathrm{G}y\}$$

These correspond to the different scope readings of 'the F' in relation to the negation operator. In the first case, 'the F' is wide; in the second case, 'the F' is narrow.