

THE MONADIC QUANTIFIER CALCULUS

1. The Monadic Quantifier and The Sentential Calculi	1
Preview – derivations of several valid English arguments that are not valid in The Sentential Calculus	
On what is in a name-letter	
2. The language	4
New signs	
Terms and formulas	
Free and bound occurrences of variables	
Sentences	
3. Translation	7
4. Symbolization	7
5. Derivation	12
Proper substitution	
The three primitive inference rules	
Universal instantiation	
Existential generalization	
Existential instantiation	
Universal derivation – a new form of proof	
Quantifier Negation – four derived inference rules	
Abbreviated derivations	
Strategies, advice, and cautions for derivations	
6. Derivation Illustrations	21
Possible symbolize-to-derive exercise	
7. Invalidation	28
Foundations	
Invalidation methods	
Spelled out interpretation	
Truth-functional expansion	
Finite models	
Open arguments: Arguments with non-sentence premises and/or conclusions	
Which method to use when	
8. Invalidation Illustrations	36
9. Open arguments	40
10. Validity in The Monadic Quantifier Calculus	43

Many valid English arguments are not valid in The Sentential Calculus. Prominent among these are arguments

that essentially involve not only words of connection, but also of quantity. Our quantifier calculi accommodate such arguments. The Monadic Quantifier Calculus of *Logic*, Chapter III deals with one-place predicates, and references by simple names, as well as by complex terms such as ‘Shirlee’s brother’ when these can be replaced without loss of validity by simple names. The Quantifier Calculus of *Logic*, Chapter IV goes further. It deals also with many-place predicates, and some uses of complex names that cannot be replaced by simple names without loss of validity.¹

1. THE MONADIC QUANTIFIER AND SENTENTIAL CALCULI

1.1 The Monadic Quantifier Calculus comes from The Sentential Calculus by several additions and modifications.

(1) Letters are added to the vocabulary of The Sentential Calculus, and assigned to several categories. There are ranges of upper-case *predicate* letters and *name* letters, and there are lower-case individual *variable* letters, in addition to the upper-case sentence letters of The Sentential Calculus.

(2) The 'grammar' of Chapter II, which identifies symbolic sentences, is elaborated in a 'grammar' that identifies not only symbolic sentences, but *formulas* other than sentences, and also *terms*.

(3) Two logical signs are added, *quantifiers*,

$$\wedge, \vee,$$

which are related to the English words ‘*all*’ and ‘*some*’.

(4) To work with these new signs in derivations, three inference rules are added, along with a form of derivation.

1.2 The Monadic Quantifier Calculus - MQC - has greater scope than The Sentential Calculus. There are English arguments that can be validated in The Monadic Quantifier

¹Suppose Howard is Shirlee’s brother. ‘Shirlee’s brother’ *can* be replaced without loss of validity by ‘Howard’ in the argument,
 Shirlee’s brother answered the phone.
 Therefore, someone answered the phone.
 but *not* in the argument,
 Shirlee’s brother answered the phone.
 Therefore, someone’s *brother* answered the phone.

Calculus, but not in The Sentential Calculus.² In this respect MQC relates to SC differently than SC relates to the Conditional-Negation Calculus.

1.3 **Preview.** Now come, to illustrate the greater scope of MQC, derivations of several valid English arguments that are not valid in The Sentential Calculus. These illustrations will display, with little or no present comment, new devices and ideas that will be explained in subsequent sections.

(1) Every Communist is a Marxist. There is a Communist who is a Canadian. ∴ Some Canadian is a Marxist.

Given the following new-style scheme in which English formulas (they are not quite sentences) are associated with predicate letters,

F: *a* is a communist; G: *a* is a Marxist; H: *a* is a Canadian argument (1), can be symbolized thus:

$$\Lambda x(Fx \rightarrow Gx). \quad \exists x(Fx \wedge Hx). \quad \therefore \exists x(Hx \wedge Gx)$$

For the sense of these symbolizations, read ' Λx ' and ' $\exists x$ ' as respectively 'for each *x*' and 'for some *x*'; and read, for example, ' Fx ' by putting '*x*' in '*a* is a communist' which is the formula 'abbreviated' by '*F*'. Here are more or less literal translations of our symbolic sentences.

For each *x*, if *x* is a communist, then *x* is a Marxist. There is an *x* such that *x* is a communist, and *x* is a Canadian. ∴ There is an *x* such that *x* is a Canadian, and *x* is a Marxist.

Our symbolic argument is validated by the following derivation in which all of the new primitive inference rules are used. The new proof procedure is not used in this derivation.

1.	SHOW $\exists x(Hx \wedge Gx)$	(10, DD)
2.	$\Lambda x(Fx \rightarrow Gx)$	premise
3.	$\exists x(Fx \wedge Hx)$	premise
4.	$Fa \wedge Ha$	3, Existential Instantiation (' <i>a</i> ' can be thought of as naming something that is both an F, and an H)
5.	$Fa \rightarrow Ga$	2, Universal Instantiation
6.	Fa	4, S
7.	Ga	5, 6, MP
8.	Ha	4, S
9.	$Ha \wedge Ga$	7, 8, Adj
10.	$\exists x(Hx \wedge Gx)$	9, Existential Generalization

²This is not controversial, though it does not admit of a simple proof. That coming illustrations are not valid in SC, while not controversial, does not admit of simple proofs.

_____3

(2) Men and women are all animals. All animals are mortal. ∴ Every woman is mortal.

Under the new-style scheme,

F: *a* is a man; H: *a* is an animal; G: *a* is a woman;
J: *a* is mortal

this argument has the symbolization,

$$\Lambda x((Fx \vee Gx) \rightarrow Hx). \quad \Lambda x(Hx \rightarrow Jx) \quad \therefore \Lambda x(Gx \rightarrow Jx)$$

Here is a derivation for this symbolic argument. This derivation uses a new inference rule, and the new form of proof according to which, to show ' $\Lambda x(Gx \rightarrow Jx)$ ' – which says that it is true of everything that if it is a G, it is a J it is a G – it is sufficient to get that ' $(Gx \rightarrow Jx)$ ' – which says that it is true of *x* that if it is a G, then it is a J. That is sufficient when, as here, *x* can be anything, when one could have gotten ' $(G \rightarrow J)$ ' with any other term supplanting the dashes.

³It can *seem* that English argument (1) is symbolized by the valid-in-SC

$$(P \rightarrow Q). (P \wedge R) \quad \therefore (Q \wedge R)$$

under the scheme – P: someone is a Communist; Q: this someone is a Marxist; R: this someone is a Canadian.

However, for the symbolization of first premise,

Every Communist is a Marxist.

'someone is a Communist' takes the sense of 'anyone is a Marxist':

If someone (anyone) is a Communist,
then this someone is a Marxist.

whereas for the symbolization of the second premise,

There is a Communist who is a Canadian.

'someone is a Communist' takes the sense of 'at least one person',

Someone (at least one person) is a Communist,
and this someone is a Marxist.

When symbolizing each sentence abbreviated by a letter must be taken in a single sense. English argument (1) is *not* symbolized by

$$(P \rightarrow Q). (P \wedge R) \quad \therefore (Q \wedge R)$$

under the scheme – P: someone is a Communist; Q: this someone is a Marxist; R: this someone is a Canadian.

1.	SHOW $\Lambda x(Gx \rightarrow Jx)$	(4, Universal Derivation: x) ⁴
2.	$\Lambda x((Fx \vee Gx) \rightarrow Hx)$	premise
3.	$\Lambda x(Hx \rightarrow Jx)$	premise
4.	SHOW $Gx \rightarrow Jx$	(10, CD)
5.	Gx	(CD)
6.	$(Fx \vee Gx) \rightarrow Hx$	2, Universal Instantiation
7.	$Hx \rightarrow Jx$	3, Universal Instantiation
8.	$Fx \vee Gx$	3, ADD
9.	Hx	6, 8, MP
10.	Jx	7, 9, MP

(3) All men are mortal. Socrates was a man. \therefore Socrates was mortal.

has, under the new-style scheme,

F: *a* was a man; A: Socrates; G: *a* was mortal

the symbolization,

$$\Lambda x(Fx \rightarrow Gx). FA \therefore GA$$

Here is an appropriately simple derivation for this simple argument.

1.	SHOW GA	(5, DD)
2.	$\Lambda x(Fx \rightarrow Gx)$	premise
3.	FA	premise
4.	$FA \rightarrow GA$	2, Universal Instantiation
5.	GA	3, 4, MP

1.4 **On what is in a name-letter.** English arguments (1) through (3) are not valid in The Sentential Calculus. To reach them predicate letters, name letters, variables and quantifiers are added to the symbolic language, and rules and procedures to the logical system. However, given the rest, I think we could deal with these arguments without

using name letters to abbreviate *English names*⁵: that is, I think that we could still produce adequate symbolizations for using instead predicate letters to abbreviate naming or identifying English formulas, for example, using, instead of the abbreviations of English names,

A: Madonna
and
B: Madonna's husband
the abbreviations,
M: *a* is Madonna
and
W: *a* is Madonna's husband

of the naming or identifying predicates that correspond to those names. I think that is right, but even if it is right, the resultant symbolizations would be more complicated and less natural – they would not ‘mock’ the simple grammar of sentences with name letters being symbolized. And more importantly, derivations for them could not ‘tract’ and illuminate ordinary deductive thinking conducted in English that features names.

To illustrate, using the scheme – F: *a* is a man; H: *a* is Socrates [an identifying or naming predicate]; G: *a* is mortal – argument (3) can be symbolized thus,

$$\Lambda x(Fx \rightarrow Gx). \forall xHx \wedge \Lambda x(Hx \rightarrow Fx) \therefore \forall xHx \wedge \Lambda x(Hx \rightarrow Gx)$$

for which symbolic argument a complete derivation in the new system exists.

This symbolization is adequate. To see this, consider that English sentences symbolized respectively by

'FA'
and
' $(\forall xHx \wedge \Lambda x(Hx \rightarrow Fx))$,
namely,
'Socrates is a man'
and
'There is something that is Socrates, and everything that is Socrates is a man'

⁴Including ‘x’ in this annotation can occasion verifying that it is not free on an ‘antecedent line’, that is, a preceding line that neither is boxed, nor contains an uncanceled ~~SHOW~~.

⁵“An *English name* is a word or group of words that purports to designate (at least within a given context) a single object.” (*Logic*, p. 117.) In fact the text counts as an English name in every context words that in some contexts purport to designate single objects. Thus ‘God’ is counted as an English name in ‘God does not exist’, since it purports to designate a single object in ‘God is great’. Proper names are in the text *Logic* counted as words.

are logically equivalent in the sense that it is necessary that one of them expresses a truth if and only if the other does: going from the first to the second, we have that if Socrates is a man, then something (indeed some man) is Socrates, and everything that is Socrates is a man; going from the second to the first is easy. What has been said of 'FA' and ' $\forall xHx \wedge \wedge x(Hx \rightarrow Fx)$ ', holds as well (and for the same reasons) for 'GA' and ' $\forall xHx \wedge \wedge x(Hx \rightarrow Gx)$ '. They too are logically equivalent.

For our unnatural name-letter-free symbolization of English argument (3), we have the following derivation there in its gyrations is nothing like deductive thinking in English that would validate this argument.

- | | | |
|-----|------------------------------------------------------------------|------------------------------|
| 1. | SHOW $\forall xHx \wedge \wedge x(Hx \rightarrow Gx)$ | (13, DD) |
| 2. | $\wedge x(Fx \rightarrow Gx)$ | premise |
| 3. | $\forall xHx \wedge \wedge x(Hx \rightarrow Fx)$ | premise |
| 4. | SHOW $\wedge x(Hx \rightarrow Gx)$ | (5, Universal Derivation: x) |
| 5. | SHOW $Hx \rightarrow Gx$ | (11, CD) |
| 6. | Hx | (CD) |
| 7. | $\wedge x(Hx \rightarrow Fx)$ | 3, S |
| 8. | $Hx \rightarrow Fx$ | 7, Universal Instantiation |
| 9. | Fx | 6, 8, MP |
| 10. | $Fx \rightarrow Gx$ | 2, Universal Instantiation |
| 11. | Gx | 9, 10, MP |
| 12. | $\forall xHx$ | 3, S |
| 13. | $\forall xHx \wedge \wedge x(Hx \rightarrow Gx)$ | 12, 4, Adj |

I believe that every English argument that features logically relevant references to individual things can be handled in The Monadic Quantifier Calculus *without*, if it can be handled *with*, name letters. Name letters are nice, and we make free use of them when symbolizing. But, in the sense explained, I think they are not needed for validations of English arguments in MQC.

Exactly stated, the proposition concerning the adequacy of MQC without name letters that I believe is correct is that, for any English argument **E** symbolized by the argument

$$A \quad \Psi_1, \dots, \Psi_n \therefore \Psi_{(n+1)}$$

under a scheme that includes abbreviations for English names and for corresponding naming or identifying English

⁶“Black boxes, ■, set off material of theoretical interest that can be skipped over without loss for purposes of most logic courses.” (Preface.)

formulas: the argument (now comes a lengthy identification)

$$A' \quad \Psi_1, \dots, \Psi_n \therefore \Psi_{(n+1)}$$

wherein for each Ψ_i , $1 \leq i \leq (n + 1)$, if no name letter occurs in Ψ_i , then Ψ_i is Φ_i , and if, for $k > 0$, name letters

$$\eta_1, \dots, \eta_k,$$

occur in then Φ_i , then Ψ_i is

$$\forall \alpha_1 \dots \forall \alpha_k [(\theta_1 \alpha_1 \wedge \dots \wedge \theta_k \alpha_k) \wedge \Phi_i'] \wedge \wedge \alpha_1 \dots \wedge \alpha_k [(\theta_1 \alpha_1 \wedge \dots \wedge \theta_k \alpha_k) \rightarrow \Phi_i']$$

wherein, for $1 \leq j \leq k$, θ_j is a predicate letter that abbreviates the naming or identifying formula that corresponds to the name that is abbreviated by η_j , and Φ_i' comes from Φ_i , by, for each name letter η_j , by replacing each occurrence of η_j by an occurrence of the variable α_j (the lengthy identification is now of A' is now complete)⁷ such that, (a), A' symbolizes **E**, and, (b), A is valid in MQC if and only if A' is valid in MQC. ■

⁷To illustrate the intended relation between a Φ and Ψ , when there are name letters in Φ , here are

$$'FA' \text{ and } '[\forall x(Gx \wedge Fx) \wedge \wedge x(Gx \rightarrow Fx)]'$$

and

$$'\sim FA' \text{ and } '[\forall x(Gx \wedge \sim Fx) \wedge \wedge x(Gx \rightarrow \sim Fx)]'.$$

It can be seen that the universal conjuncts of the predicate-letter sentences are important. For these sentences need to make an inconsistent set as the name-letter sentences do.

One consequence of this inconsistency is the validity of,

$$FA, \sim FA \therefore P$$

' $\forall x(Gx \wedge Fx)$ ' and ' $\forall x(Gx \wedge \sim Fx)$ ' make a consistent set, and the argument

$$\forall x(Gx \wedge Fx), \forall x(Gx \wedge \sim Fx) \therefore P$$

is not valid. In welcome contrast, the premises of the argument

$$[\forall x(Gx \wedge Fx) \wedge \wedge x(Gx \rightarrow Fx)].$$

$$[\forall x(Gx \wedge \sim Fx) \wedge \wedge x(Gx \rightarrow \sim Fx)] \therefore P$$

are inconsistent, and it is valid. In a derivation for it 'Fa' and ' $\sim Fa$ ' can be derived for an indirect derivation of 'P'

1. ~~SHOW~~ P (ID)

2. $Ga \wedge Fa$ premise (i), S, EI

The idea of this use of Existential Instantiation is that we are *letting a be a thing that is both 'a G' and 'an F'*: ' $\forall x(Gx \wedge Fx)$ ' says there is such a thing.

3. Fa 2, S
4. $\sim Fa$ premise (ii), S, Universal Instantiation, 2, S, MP

2. THE LANGUAGE

2.1 *New signs.* The Sentential Calculus has only one kind of non-logical sign: it has sentence letters 'P' through 'Z' with or without numerical subscripts. Parentheses are not signs, and the connectives are logical signs. The language of The Monadic Quantifier Calculus has additional sets of *non-logical* signs or letters.

name letters: 'A' through 'E' with or without subscripts

variables: 'a' through 'z' with or without subscripts

predicate letters: 'F' through 'O' with or without subscripts

And for this language, to the *logical signs* of The Sentential Calculus we add the *universal quantifier*, ' \forall ', and the *existential quantifier*, ' \exists '. Typographically these quantifiers are, respectively, big conjunction and disjunction signs. This is suggestive of their senses. As will be evident, they are in their significance 'big conjoiners' and 'big disjoiner,' they are for symbolizing sentences some of which are true if and only if certain repeated conjunctions and disjunctions are true, and other of which are related in this way to what would have to be 'infinitely long' repeated conjunctions and disjunctions.⁸

2.2 *Terms and formulas.* Only one kind of expression is defined for The Sentential Calculus, sentences. Two kinds are defined for The Monadic Quantifier Calculus, terms and formulas.

The *terms* of The Monadic Quantifier Calculus are simply its variables and name letters. Thus the following letters are terms: 'a', 'x', 'A', and 'E'.

The *formulas* of The Monadic Quantifier Calculus are precisely the expressions identified by the following stipulations. First, two kinds of 'atomic formulas' in which no logical signs occur, and then two kinds in which logical signs do occur.

⁸'The countries of North America are democracies' is true if and only if 'Canada is a democracy, the United States of America is a democracy, and Mexico is a democracy'. 'Every odd natural number succeeds and even number' is true if and only if '1 succeeds an even number, 3 succeeds an even number, 5 succeeds an even number, and so on *ad infinitum*'. The sentence 'Every star in the galaxy has a centre of gravity', though addressed to a finite class of things, is not related in the indicated way to a very long repeated conjunction if not every star in the galaxy has a distinguishing description, for not every star in the galaxy has a name.

(I) Sentence letters are formulas.

(ii) If ϕ is a predicate letter, and δ is a term, then $\phi\delta$ is a formula. E.g., 'Fx' and 'FB', *but not* 'Fxy'.

(iii) If ϕ and ψ are formulas, then so are,

$\sim\phi$, $(\phi \rightarrow \psi)$, $(\phi \vee \psi)$, $(\phi \wedge \psi)$, and $(\phi \leftrightarrow \psi)$.

(iv) If ϕ is a formula, and α is a variable, then

$\forall\alpha\phi$ and $\exists\alpha\phi$

are formulas.

Examples

$(Fx \rightarrow Gx)$, $(Ha \wedge P)$, $(FA \wedge FB)$

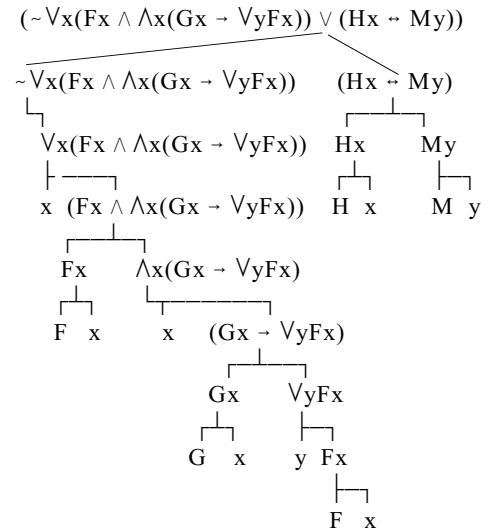
$\forall xFx$, $\forall yGy$

$\forall xFy$, $\forall xP$ [These formulas feature 'vacuous quantifiers,' countenanced to keep the grammar simple.]

$\forall x(Fx \rightarrow Gx)$, $\forall y\sim Ly$ [But neither ' $\forall(Fx \rightarrow Gx)$ ' nor ' $\forall\sim xFx$ ']

$\forall y(Fy \rightarrow \forall xGx)$, $\forall x(Fx \rightarrow \forall xGx)$,
 $\forall x(Fx \wedge Gx) \rightarrow \forall xHx$

It is a consequence of the grammar that every formula decomposes uniquely along lines determined by grammatical rules to immediate components, and ultimately to single letters. Here is an illustration.



[The 'initial nodes' of grammatical trees are here letters, specifically, sentence letters, predicate letters, name letters, and variables. In *Logic* they are 'atomic formulae', specifically, sentence letters and formulae ' $\phi\alpha$ ' wherein ϕ is a predicate letter and α is a name letter or variable.

(Logic, p. 120-1.) In LOGIC 2000, 'initial nodes' are name letters or variables.]

Conventions for *informal* formulas are the same as for informal sentences in The Sentential Calculus. So there are no provisions for *adding* parentheses, and, for example, ' $(\wedge x)Fx$ ' and ' $F(x)$ ', which are not official formulas, are also not unofficial ones according to the conventions of our text, *Logic*.

2.3 Free and bound occurrences of variables

Each occurrence of a quantifier comes with an occurrence of a variable. Occurrences of quantifiers *bind* all occurrences of their variables that 'stand under them.' More exactly, an occurrence of a variable α is bound in a formula ϕ exactly in case it stands in ϕ in an occurrence of a formula

$$\wedge \alpha \psi$$

or

$$\vee \alpha \psi.$$

An occurrence of a variable α is 'bound by' the quantifier that heads the shortest formula in which it is bound.⁹

To help to fix the ideas of freedom and bondage, we are encouraged (see *Logic*, p. 125) to think of occurrences of variables as 'bound once and for all' as a formula is grammatically generated. One can set out a decomposition *sans* branches and single letters. Then, *starting from the bottom of the tree*, when a quantifier is introduced, 'link' the variables it then binds. It holds on to these, and no other quantifiers get at them, as one moves up the tree. Also, as one moves up the tree, that quantifier does not 'pick up and bind' any other occurrences of its variable. So, for the recently displayed tree, starting from the bottom I have emphasized quantifiers and the occurrences of their variables they bind when they 'enter' the sentence as it is composed. It can be seen that these are the occurrences it binds in the fully composed sentence.

$$(\sim \forall x(Fx \wedge \wedge x(Gx \rightarrow \forall yFy)) \vee (Hx \leftrightarrow My))$$



$$\sim \forall x(Fx \wedge \wedge x(Gx \rightarrow \forall yFy)) \quad (Hx \leftrightarrow My)$$



$$\forall x(Fx \wedge \wedge x(Gx \rightarrow \forall yFy)) \quad Hx \quad My$$



$$(Fx \wedge \wedge x(Gx \rightarrow \forall yFy))$$



$$Fx \quad \wedge x(Gx \rightarrow \forall yFy)$$



$$(Gx \rightarrow \forall yFy)$$

$$Gx \quad \forall yFy$$

$$|$$

$$Fx$$

You might indicate with bondage links the pattern of bondage in the following informal sentence. (You should be able to do this *without* constructing a trimmed decomposition tree, and working out the pattern of bondage by the starting as the bottom of the tree and proceeding to the top.)

$$\wedge x(Fx \rightarrow \forall y[Gy \wedge \wedge x(Hx \rightarrow Fy)]) \rightarrow Gx.$$

It is of particular importance in the context of derivations to realize that *bondage is a relative affair*. To explain, the second numbered occurrence of 'x' below,

$$\wedge x(Fx \rightarrow Gy) \wedge Hx$$

$$1 \quad 2 \quad 3$$

is at once an occurrence of 'x' in the whole conjunction formula, and in each of the progressively shorter formulas ' $\wedge x(Fx \rightarrow Gy)$ ', ' $(Fx \rightarrow Gy)$ ', and ' Fx '. This occurrence-(2) of 'x' is *free* in the last two formulas, but *bound* in the first conjunct of the displayed formula, as well as in this formula itself.

2.4 Sentences. A formula ϕ of The Monadic Quantifier Calculus is a *sentence* if and only if no occurrence of a

⁹Let the *scope* of an occurrence of a sentential connective or quantifier phrase in a formula ϕ is the shortest formula in ψ in which this occurrence stands. Then every occurrence of the variable α in a formula that lies in the scope of an occurrence of the quantifier phrase $\wedge \alpha$ is bound, and bound by this occurrence of $\wedge \alpha$, and similarly for an occurrence of the variable α in a formula in the scope of an occurrence of the quantifier phrase $\forall \alpha$.

variable is *free* in ϕ . Thus ' $\forall x(Fx \rightarrow Gx)$ ' is a sentence, but ' $\forall xFx \rightarrow Gx$ ' is *not* a sentence, since the third occurrence of 'x' is free in it.

This idea is important for symbolizations, since a condition for a correct symbolization ϕ of an English sentence S , is that ϕ be it a symbolic sentence. A consequence of this minimal demand is that

If a frog is excited, it croaks.

cannot be symbolized, under the scheme (F: a is a frog; G: a is excited; H: a croaks), by the conditional formula,

$$\forall x(Fx \wedge Gx) \rightarrow Hx.$$

The last occurrence of 'x' is *free* in this formula. So this formula is not a sentence, and therefore cannot symbolize a sentence.

3. TRANSLATION

3.1 *Literal translations.* The universal [ALL/EVERY] quantifier phrase,

$$\forall \alpha \phi,$$

has the literal translation,

for each α .

The existential [SOME/AT LEAST ONE] quantifier phrase,

$$\exists \alpha,$$

has the literal translation,

there is an α such that.

3.2 Translating a symbolic formula under a scheme of abbreviation is, in the first instance, an entirely mechanical process (see *Logic*, p. 129). Here are two very simple but important examples. Under the scheme (F: a is a frog; G: a is green), the universal generalization,

$$\forall x(Fx \rightarrow Gx),$$

has the literal translation,

For each x, if x is a frog, then x is green.

The displayed symbolic universal generalization has the following more idiomatic **All-A'S-are-B'S** free translations.

All frogs are green.

Frogs are green.

Every frog is green.

Each frog is green.

and

Any frog is green.

The existential generalization,

$$\exists x(Fx \wedge Gx),$$

has the literal translation,

There is an x such that, x is a frog and x is green,

and the idiomatic **Some-A-is-B** free translation,

Some frog is green.

as well as free translations such as the following, from which the 'existential' quantifier get its name,

There *is* a green frog.

and

There *exists* a green frog.

4. SYMBOLIZATION

1. *Cows are gentle.* (F: a is a cow; G: a is gentle)

$$\forall x(Fx \rightarrow Gx)$$

English sentence (1) is an All-A'S-are-B generalization. We assign to all such generalizations the form,

$$\forall x(\quad - \quad)$$

The inner antecedent establishes the *subject* of the generalization, what it is about: in this case, cows, that is, x's such that x is a cow, which condition has the symbolization 'Fx'.

$$\forall x(Fx \rightarrow \quad)$$

The inner consequent expresses the *predicate* of the generalization, or what it is saying about its subject: in this case, being gentle – the generalization says of each of the x's of its subject, that x is gentle, which condition has the symbolization 'Gx'.

N.B. English sentence (9) does *not* have as a symbolization,

$$\forall x(Fx \wedge Gx)$$

This symbolic sentence has the literal translation 'for each x (x is a cow and x is gentle)', that is, 'everything is a gentle cow'.

The symbolization we assign to (1) leaves open whether or not there *are* any cows. If a stronger sense is

required, it can be rendered by explicitly adding that there is at least one.

$$\Lambda x(Fx \rightarrow Gx) \wedge \exists xFx$$

2. *Some cows are gentle.*

$$\exists x(Fx \wedge Gx)$$

English sentence (2) is a **Some-A'S-are-B** generalization. We assign to every *such* generalizations the form,

$$\exists x(\quad \wedge \quad)$$

The first inner conjunct establishes the *subject*, the kind of things the generalization is about: here, x's such that x is a cow. The second inner conjunct expresses the *predicate*, what we want to affirm of some such x: here that x is gentle.

It may seem that our symbolization is better suited to the singular generalization 'some cow is gentle' than to the plural generalization 'some cows are gentle', for ' $\exists x$ ' has the translation 'there is a, at least one, x such that'. The language of the present theory is not adequate to this distinction. Our treatment in it of English sentence (2) is not perfect,¹⁰ but it is adequate for many applications. In many arguments the difference between 'some is' and 'some are' is not logically significant.

N.B. The English sentence (2) does *not* have as a symbolization,

$$\exists x(Fx \rightarrow Gx).$$

This symbolic sentence has the literal translation into 'there is an x such that (if x is a cow, then x is gentle)' which could be true even if there were no cows. Indeed, taking 'if ... then' in the truth-table sense of ' \rightarrow ', this English sentence is true not just if there are gentle cows, but is true simply true (with 'no if about it'), since not *everything* is a cow! There *is* an x such that (x is a cow \rightarrow x is gentle). There are many. Take Madonna for example. It is *true* that (Madonna is a cow \rightarrow Madonna is gentle), since it is *false* that Madonna is a cow (she is, of course, a human being). For the same reason it is true that (Madonna is a cow \rightarrow Madonna is *not* gentle).

Some sentences *invite* by their words 'existential conditional symbolizations.' But these temptations are almost always to be resisted. Generally one can see that the sentence *in context* intends something stronger and less

easily true than its existential conditional symbolization. An example is the sentence,

Someone will win \$1,000, if he buys a ticket.

in the context of a winner-take-all-the-proceeds lottery, for which 1,000 tickets are on offer for \$1 each. This sentence says something false if not every ticket is sold, though in that case, if someone does not buy a ticket, the partially symbolic sentence,

$$\exists x(x \text{ will buy a ticket} \rightarrow x \text{ will win } \$1,000)$$

says something true. (So how is that lottery sentence to be symbolized? Either by a single letter, or in another logical language that features a different kind of conditional connective. Supplementary Section 16 explains.)

3. *Only cows are gentle.*

Interpreted weakly or conservatively this sentence has the stylistic variant 'every gentle thing is a cow',

$$\Lambda x(Gx \rightarrow Fx),$$

and also the stylistic variant 'nothing that is not a cow is gentle',

$$\sim \exists x(\sim Fx \wedge Gx).$$

These symbolizations are equivalent, and each is correct. I prefer the first, but tastes can differ here.

We symbolize (3) in a way that leaves open that perhaps there are not cows: to rule this out, one can conjoin $\exists xFx$. Also left open is that perhaps, though there are cows, none are gentle: to rule this out one can $\exists x(Fx \wedge Gx)$.

Sometimes (3) is naturally understood as including that all cows are gentle. Then its symbolization can be, $\Lambda x(Fx \leftrightarrow Gx)$, or, 'with existential import,' $\exists xFx \wedge \Lambda x(Fx \leftrightarrow Gx)$. But, as always, our practice is to symbolize *conservatively*, which in this case means that we stop with $\Lambda x(Gx \rightarrow Fx)$. That, at least, is conveyed by (3) *in every context*, and so, as now, when no particular context is given.

4. *Unicorns do not exist.* (I: a is a unicorn)

$$\sim \exists xIx$$

Variants of (4) include 'there are no unicorns', 'there do not exist any unicorns', 'there does not exist a unicorn', and 'there does not exist even one unicorn'.

5. *No cows are gentle.*

¹⁰We can do better in the language of the Identity Calculus of Chapter V. Sentence (2) has the stylistic variant 'there are at least two cows' and this has, in the language of that calculus, the symbolization ' $\exists x \exists y((Fx \wedge Fy) \wedge (Gx \wedge Gy) \wedge x \neq y)$ '.

$$\sim \forall x(Fx \wedge Gx),$$

or equivalently,

$$\forall x(Fx \rightarrow \sim Gx).$$

Of these I prefer the first. (Again, stronger interpretations can be reached by making explicit that there are some cows, and perhaps also that there are gentle things.)

(5) has the stylistic variant 'cows are not gentle'. Contrast (5) with '*not all cows are gentle*' – $\sim \forall x(Fx \rightarrow Gx)$ – which sentence means something different. Consider the sentence 'all cows are not gentle'. What does it mean? (It is, I think, ambiguous between 'not all' and an 'all not' interpretations.)

6. *None but the gentle are home.* (H: *a* is home)

$$\sim \forall x(Hx \wedge \sim Gx),$$

or equivalently,

$$\forall x(Hx \rightarrow Gx).$$

"But (6) is about persons. What it says is consistent with a vicious dog's being home." This is true. Sentence (6) is in almost every context a stylistic variant of 'It is not the case that *someone* [i.e., some *person*] is home who is not gentle'. In symbolizing it without a letter that abbreviates '*a* is a person' I am engaging in a practice the text *Logic* countenances though it acknowledges that it is 'questionable': "It will sometimes be our (admittedly questionable) practice to treat 'no one', 'someone', 'everyone' as stylistic variants of 'nothing', 'something', 'everything'." (*Logic*, p. 135.)

7. *Brown cows are gentle.* (O: *a* is brown)

Symbolizing step-by-step, (7) is an All-A's-are-B's generalization:

$$\forall x(\quad \rightarrow \quad)$$

It is *about* brown cows,

$$\forall x((Ox \wedge Fx) \rightarrow \quad)$$

And it says each is gentle.

$$\forall x((Ox \wedge Fx) \rightarrow Gx)$$

8. *Members of the physics class are in either S148 or R4242.* (M: *a* is a member of the physics class; I: *a* is in S148; K: *a* is in R4242)

This sentence is ambiguous between,

$$\forall x(Mx \rightarrow Ix \vee Kx),$$

and

$$\forall x(Mx \rightarrow Ix) \vee \forall x(Mx \rightarrow Kx).$$

9. *No cow in the pasture is gentle.* (J: *a* is in the pasture)

$$\sim \forall x[(Fx \wedge Jx) \wedge Gx]$$

or equivalently,

$$\forall x(Fx \wedge Jx \rightarrow \sim Gx)$$

10. *Only gentle cows are in the pasture.*

This sentence is three ways ambiguous between,

$$\forall x(Jx \rightarrow Gx \wedge Fx),$$

$$\forall x(Fx \wedge Jx \rightarrow Gx)]$$

and,

$$\forall x(Gx \wedge Jx \rightarrow Fx).$$

Suppose one is talking about animals in the pasture. Then, with one pattern of emphasis or another, one might use (10) to say that they are all gentle cows, that the *cows* amongst them are all gentle, or that the *gentle* animals amongst them are all cows.

11. *Cows and unicorns are smart.* (H: *a* is smart; I: *a* is a unicorn; F: *a* is a cow)

$$\forall x(Fx \rightarrow x) \wedge \forall x(Ix \rightarrow Hx)$$

and equivalently,

$$\forall x(Fx \vee Ix \rightarrow Hx).$$

But *not*,

$$\forall x(Fx \wedge Ix \rightarrow Hx).$$

12. *If any cows are in the pasture, then so is Annie.* (A: Annie)

'Any' has in this sentence the force of 'some,' and this sentence is symbolized by a conditional.

$$\forall x(Fx \wedge Jx) \rightarrow JA,$$

An equivalent symbolization renders the sentence as a universal generalization, in effect drawing 'any' out to a position before 'if'.

$$\forall x(Fx \wedge Jx \rightarrow JA)$$

The sentence does *not* have the symbolization,

$$\forall x(Fx \rightarrow Jx) \rightarrow JA,$$

which has the translation 'if *every* cow is in the pasture [if *all* cows are in the pasture], then so is Annie'.

13. *If a cow is in the pasture, it is gentle.*

This sentence can be symbolized as a universal generalization,

$$\wedge x(Fx \wedge Jx \rightarrow Gx),$$

but *not* as a conditional. It does not have the symbolization,

$$\vee x(Fx \wedge Jx) \rightarrow Gx.$$

This conditional *reads right* – ‘if, for some x , x is a cow and x is in the pasture, then x is gentle’ – but it *is not a sentence*: The last occurrence of ‘ x ’ is free in the whole conditional formula. And English sentences are *symbolized only by symbolic sentences*.

“We say that ϕ is a *symbolization* of an

English **sentence** ψ on the basis of a given scheme of abbreviation if and only if ϕ is a **symbolic sentence** that has ψ as a translation on the basis of that scheme” (*Logic*, pp. 9-10, bold emphasis added.)

Therefore, translations of

$$[\vee x(Fx \wedge Jx) \rightarrow Gx]$$

under the scheme – F: a is a cow; J: a is in the pasture; G: a is gentle, including its *literal translation*,

If there is an x such that x is a cow and x is in the pasture, then x is gentle.

and this *free translation*

If a cow is in the pasture, it is gentle.

And yet ‘ $[\vee x[(Fx \wedge Jx) \rightarrow Gx]]$ ’ is not a *symbolization* of these English sentences, because it is not a *symbolic sentence*.

.....

But *why?* Why not *liberalize* the definition of a symbolization and say *this?*

ϕ is a *symbolization* of an English **sentence** ψ on

the basis of a given scheme of abbreviation if and only if ϕ is a symbolic formula sentence that has ψ as a translation on the basis of that scheme.

The reason is that then there would be patently invalid English arguments that were valid-in-MQC: that is, then there would be patently invalid English arguments that had symbolized by symbolic arguments there were valid by derivations. A case to this point is the patently invalid English argument,

If anything was found, it was left in H525. My watch was not left in H525. Therefore, *nothing* was found!!

which under this liberalized definition of symbolizations would be symbolized by,

$$(\vee xFx \rightarrow Lx). \sim LA \therefore \sim \vee xFx$$

under the scheme – F: a was found; L: a was left in H525; A: my watch – and this symbolic argument is valid. There is a derivation for it in Section 9.3 below.

The official definition of symbolizations does not have

this awkward consequence. But here is no avoiding that the valid symbolic argument,

$$(\vee xFx \rightarrow Lx). \sim LA \therefore \sim \vee xFx$$

has as a translation the patently invalid English argument,

If anything was found, it was left in H525. My watch was not left in H525. Therefore, *nothing* was found!!

The lesson is that translations, literal and free, of symbolic formulas into English are, to adapt words from our textbook, ‘only approximate guides to the logic of these formulas’. It is said in *Logic* that validity of an English argument in a logical calculus “is only an approximate guide to the logic of ordinary language” (xiii.)

14. *If someone is home, he will answer.* (L: a is a person; H: a is home; N: a will answer)

$$\wedge x(Lx \wedge Hx \rightarrow Nx),$$

and, by the questionable practice mentioned under (6) above,

$$\wedge x(Hx \rightarrow Nx).$$

Again, English sentences can be symbolized only by *symbolic sentences*. For this reason sentence (14) cannot be symbolized by either

$$\vee x(Lx \wedge Hx) \rightarrow Nx$$

or

$$\vee xHx \rightarrow Nx.$$

These two formulas are not sentences. Each contains a free occurrence of ‘ x ’.

15. *All that glitters is not gold.* (L: a glitters; G: a is gold)

This is a proverbial, cautionary saying, as us ‘A stitch in time saves nine’. In its customary use it has the sense of,

$$\sim \wedge x(Lx \rightarrow Gx),$$

though, from the arrangement of its words, one might suppose that it had the sense of,

$$\wedge x(Lx \rightarrow \sim Gx).$$

It may seem that that is what sentence (15) 'ought' to mean, and that its customary use is a misuse. In fact, however, more often than not in ordinary discourse 'all... not...' and 'every...not...' sentences are intended, and are understood in the sense of 'not all...' and 'not every...' sentences. Since established and accepted practice, these are not misuses. Go figure ordinary discourse, and its grammar!

16. *There is hope for the living.* (H: there is hope for *a*; L: *a* is living)

One interpretation reads 'the living' as '*all* the living':

$$\wedge x(Lx \rightarrow Hx),$$

and, making 'existential import' explicit,

$$\wedge x(Lx \rightarrow Hx) \wedge \vee xLx.$$

And there may be an interpretation that backs off, and in effect reads 'the living' as '*some* of the living':

$$\vee x(Lx \wedge Hx).$$

(There is discussion in Supplementary Section 14 of the somewhat similar sentence 'there is resurrection of the dead'. Discussion in that section is relevant also to the next two sentences.)

17. *There is no hope for the living.*

$$\wedge x(Lx \rightarrow \sim Hx).$$

18. *It is not the case that there is hope for the living.*

This sentence seems to have the force of sentence (17). A natural symbolization would, I think, be,

$$\sim \vee x(Lx \wedge Hx),$$

which is equivalent to ' $\wedge x(Lx \rightarrow \sim Hx)$ '.

Sentences (17) and (18) are like the sentence,

There is not hope for the living.

They contrast with the sentence,

'there is not hope for *all* the living',

which sentence has the symbolization,

$$\sim \wedge x(Lx \rightarrow Hx).$$

19. *A Perfect Being has the perfection of existence.* (F: *a* is a Perfect Being; G: *a* has the perfection of existence)

This sentence is ambiguous between

Any perfect being has the perfection of existence.

$$\wedge x(Fx \rightarrow Gx),$$

and

At least one Perfect Being has the perfection of existence.

$$\vee x(Fx \wedge Gx).$$

The first, universal interpretation' can be 'true by definition' somewhat in the manner 'every trillionaire has a trillion dollars': one may rule that by definition a Perfect Being has every perfection, and that existence is a perfection somewhat as red is a colour. However, this interpretation leaves open that perhaps there does not exist a Perfect Being. The second existential interpretation settle that there exists a Perfect Being, but it cannot be claimed to be true by definition, anymore than it is true by definition that there is at least one trillionaire who has a trillion dollars. (In fact, it is still false that there is at least one trillionaire.)

.....

Is every 'a'-sentence universal/existential ambiguous? LOGIC 2000 implies, No. It suggests that, given the scheme

(F: *a* is a chocolate; G: *a* should be enjoyed; H: *a* poisoned my friend)

A chocolate should be enjoyed.

is symbolized by

$$\wedge x(Fx \rightarrow Gx),$$

and never, in any case, by ' $\vee x(Fx \wedge Gx)$ ': I doubt this exclusion. And it suggests that

A chocolate poisoned my friend.

is symbolized by

$$\vee x(Fx \wedge Hx)$$

but not ever, in any case, by ' $\wedge x(Fx \rightarrow Hx)$ ': this seems right.

20. *Sam does not like everything on the menu.*¹¹ (L: Sam likes *a*; M: *a* is on the menu; K: Sam dislikes *a*)

$$\sim \wedge x(Mx \rightarrow Lx)$$

21. *Sam does not like anything on the menu.*

$$\sim \vee x(Mx \wedge Lx),$$

or, equivalently,

$$\wedge x(Mx \rightarrow \sim Lx).$$

¹¹Green eggs and ham are on the menu.

Contrast (21) with (20).

Both contrast with

Sam dislikes everything on the menu.

which cannot be symbolized under our scheme: 'Sam does not like *a*' does not entail 'Sam dislikes *a*' – between liking and disliking lies neutrality.

22. *If a necklace is missing, everything is lost.* (N: *a* is a necklace; M: *a* is missing; L: *a* is lost)

$$\forall x(Nx \wedge Mx) \rightarrow \Lambda xLx$$

or equivalently,

$$\Lambda x(Nx \wedge Mx \rightarrow \Lambda yLy).$$

This sentence is equivalent to,

$$\Lambda x(Nx \wedge Mx \rightarrow \Lambda xLx).$$

Note that these last two symbolic sentences have the same patterns of bondage.

23. *If there is a necklace that is missing, it is lost.*

$$\Lambda x(Nx \wedge Mx \rightarrow Lx),$$

but again, for a now familiar reason, *not*

$$\forall x(Nx \wedge Mx) \rightarrow Lx,$$

which symbolic formula is not a sentence.

24. *If a certain necklace is missing, it is lost.*

The best we can do, under the scheme for sentence (22), for sentence (24) is to treat it as a variant of,

There is a necklace such that if it is missing, it is lost

which has the symbolization,

$$\forall x[Nx \wedge (Mx \rightarrow Lx)].$$

If we add the abbreviation – A: a certain necklace – we have the better symbolization,

$$NA \wedge (MA \rightarrow LA).$$

But is 'a certain necklace' an 'English name,' as it must be to be abbreviated by a name-letter? Yes, this 'indefinite description' is in some context a name. Compare, '*a certain joker* – we all know who I am talking about – short-sheeted my bed.'

25. *The necklaces are all lost, unless they are not all missing.*

$$\Lambda x(Nx \rightarrow Lx) \vee \sim \Lambda x(Nx \rightarrow Mx)$$

26. *Though some are missing, none of the necklaces are lost.*

$$\forall x(Nx \wedge Mx) \wedge \sim \forall x(Nx \wedge Lx).$$

27. *The garage is heated at all times (i.e., it is always heated).* (J: *a* is a time; H: the garage is heat at *a*)

$$\Lambda x(Jx \rightarrow Hx)$$

or equivalently,

$$\sim \forall x(Jx \wedge \sim Hx).$$

28. *At some times the garage is unheated (i.e., it is not always heated).*

$$\forall x(Jx \wedge \sim Hx),$$

or equivalently

$$\sim \Lambda x(Jx \rightarrow Hx).$$

29. *At no time is the garage heated (i.e., it is never heated, i.e., it is at all times unheated).*

$$\sim \forall x(Jx \wedge Hx),$$

or equivalently,

$$\Lambda x(Jx \rightarrow \sim Hx).$$

30. *The garage is not heated at all times.*

This sentence is ambiguous. One of its interpretations is a variant of (29),

The garage is never heated.

Another of its interpretations is a variant of (28),

The garage is not always heated.

31. ¹²It's you or no one.¹² (F: it is *a*; A: you)

$$FA \vee \sim \forall xFx$$

Please see the note under sentence (14): '∀x' has here the translation 'someone x'.

5. DERIVATION

5.1 For deductions in The Monadic Quantifier Calculus, we have *three new primitive inference rules*, UI (universal instantiation), EG (existential generalization), and EI (existential instantiation); and *one new form of*

¹²Best sung by Sarah Vaughan.

proof, universal derivation (UD). Also, nearly from the start, we have a set of *four new derived rules*, quantifier negation rules (QN).

These devices are added to those of The Sentential Calculus, the rules and procedures of which are now extended to formulas in general.. For example, modus ponens is now the rule that, for any *formulas*, ϕ and ψ , from ϕ and $\phi \rightarrow \psi$, one can infer ψ . To illustrate, not only are the following, wherein ϕ and ψ are both sentences, valid by MP-inferences,

$$\frac{\begin{array}{l} \Lambda x Fx \rightarrow \forall x Gx \\ \Lambda x Fx \end{array}}{\forall x Gx} \qquad \frac{\begin{array}{l} \Lambda x (Fx \rightarrow P) \rightarrow (\forall x Fx \rightarrow P) \\ \Lambda x (Fx \rightarrow P) \end{array}}{\forall x Fx \rightarrow P}$$

but so are the following, wherein not both ϕ and ψ are sentences,

$$\frac{\begin{array}{l} Fx \rightarrow Gx \\ Fx \end{array}}{Gx} \qquad \frac{\begin{array}{l} \Lambda x Fx \rightarrow Gx \\ \Lambda x Fx \end{array}}{Gx}$$

5.2 Proper Substitution

5.2.1 The new primitive rules of The Monadic Quantifier Calculus are framed in terms of a certain way in which formulas can be related. Before proceeding to these rules, we take up the idea of this relationship.

Definition:

a formula ϕ' comes from formula ϕ by *proper substitution of a term β for a variable α* if and only if ϕ' is exactly like ϕ except for having free occurrences of β wherever (if anywhere) ϕ has free occurrences of α .

For example, from

$$(Fx \rightarrow \forall y((Gx \wedge Hz) \wedge Jy))$$

come by proper substitution of a variable for 'x' the formulas,

$$(Fa \rightarrow \forall y((Ga \wedge Hz) \wedge Jy))$$

$$(Fz \rightarrow \forall y((Gz \wedge Hz) \wedge Jy))$$

and

$$(Fx \rightarrow \forall y((Gx \wedge Hz) \wedge Jy))$$

None of the following come from

$$(Fx \rightarrow \forall y((Gx \wedge Hz) \wedge Jy))$$

by proper substitution of a variable for 'x':

$$(Fa \wedge \forall y((Gx \wedge Hz) \wedge Jy))$$

$$(Fa \wedge \forall y((Ga \wedge Ha) \wedge Jy))$$

$$(Fy \wedge \forall y((Gy \wedge Hz) \wedge Jy))$$

And neither of the following come from ' $(\Lambda x Fx \rightarrow Gx)$ ' by proper substitution of a variable for 'x':

$$(\Lambda x Fy \rightarrow Gy)$$

$$(\Lambda y Fy \rightarrow Gy)$$

Proper substitution of name letters (the other kind of term) for variables is similar but simpler, since *all* occurrences of name letters are free.

N.B. *Proper substitution is not a form of inference.*

Though we speak of one formula 'coming from another' by proper substitution, this is just a way of say that the formulas in question are related in a certain way. To take a very simple case, 'Fy' does come from 'Fx' by proper substitution of 'y' for 'x', but 'Fy' cannot be *inferred* from 'Fx' – it cannot be inferred by any of our old rules, or by any of the new ones. If it could be, '(Fx \rightarrow Fy)' would be a theorem, which it is not: please see Section 9.2 below.

5.2.2 *The intuitive idea.* Some sense of the significance of this relation of proper substitution can be gained if we allow ourselves to think of formulas as 'saying things.' I shall speak of formulas as 'saying things' of what we can think of some or all free occurrences of a variable's naming.

For example, the formula 'Fx \wedge Gy' can be thought of as saying *of x* that,

it is an F and y is a G,

and thought of as saying *of y* that,

x is an F and it is a G.

For a trickier case, the formula

$$Fx \wedge Gx$$

can be thought of as saying *three* different things *of x*, namely, that

it is an F and x is a G,

x is an F and it is a G,

and

it is an F and it is a G.

Returning to the idea of proper substitution, we can say that if a formula φ' comes from a formula φ by proper substitution of a term β for a variable α , then

φ' says of β everything that φ says of α .

φ' may say *more* things of β , this will happen if β is already free in φ , but it must say *at least* everything of β that φ says of α . Thus,

$$Fy \wedge Gy$$

comes from

$$Fx \wedge Gy$$

by proper substitution of 'y' for 'x' and says *more* things of y than this latter formula says of x (but it does say of y the thing this formula says of x).

5.3 The three primitive inference rules

5.3.1 Universal Instantiation – UI

5.3.1.1 The intuitive idea of UI is that if something is true of *everything*, then it is true of *any particular thing* one wants to mention. Formally, the rule is,

$$\Lambda\alpha\varphi \therefore \varphi',$$

wherein φ' comes from φ by proper substitution of some term β for α .¹³ So φ' 'says of' β , what φ 'says of' α . Since the premise 'says' that that is true of everything, the inference is plainly valid.

Here, for illustration, are several arguments whose conclusions follow by UI-inferences.

$\frac{\Lambda x(Fx \wedge Hx)}{Fx \wedge Hx}$	$\frac{\Lambda x(Fx \wedge Hx)}{Fy \wedge Hy}$	$\frac{\Lambda x(Fx \wedge Hy)}{Fx \wedge Hy}$
$\frac{\Lambda x(Fx \rightarrow GA)}{FA \rightarrow GA}$	$\frac{\Lambda x(Fx \rightarrow \Lambda xGx)}{Fy \rightarrow \Lambda xGx}$	$\frac{\Lambda x\forall y(Fx \rightarrow Gy)}{\forall y(Fx \rightarrow Gy)}$

Problem: It can *seem* that, with the exception of argument (i), these arguments are *not* valid by UI-inferences! Consider for example (ii): 'x' has no free

¹³The text uses Greek subscripts and writes ' $\varphi\alpha'$ ' and ' $\varphi\beta'$ ' where I write ' φ' ' and ' φ'' '. But "there is no requirement that α be free, or indeed even that α occur, in $\varphi\alpha'$ " (*Logic*, p. 189). That is why I have switched to ' φ' ' and ' φ'' '. (This is a switch back to the way of the first edition of *Logic*.)

occurrences in its premise ' $\Lambda x(Fx \wedge Hx)$ '. Proper substitution of 'y' for 'x' involves replacements of only *free* occurrences of 'x' by occurrences of 'y'. Solution: The rule UI, as stated above, would have ' $(Fy \wedge Hy)$ ' coming from ' $(Fx \wedge Hx)$ ' by proper substitution of 'y' for 'x', and this calls for replacements of occurrences of 'x' that are *free in* ' $(Fx \wedge Hx)$ '. Both occurrences of 'x' are free in ' $(Fx \wedge Hx)$ '. That neither is free in the whole premise ' $\Lambda x(Fx \wedge Hx)$ ' is not relevant to the application of UI to this premise.

5.3.1.2 Now come several arguments that are not valid by UI-inferences. They fail to qualify because of details of the 'freedom requirement' which is that β be free in φ' wherever α is free in φ .

$\frac{(ii^*) \quad \Lambda x(Fx \wedge Gx)}{Fy \wedge Gx}$	$\frac{(v^*) \quad \Lambda x(Fx \rightarrow \Lambda xGx)}{Fy \rightarrow \Lambda yGy}$	$\frac{(vi^*) \quad \Lambda x\forall y(Fx \rightarrow Gy)}{\forall y(Fy \rightarrow Gy)}$
-------------------------------------------------------------	----------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------

Argument (ii)* contrasts with (ii): the problem with (ii*) is that ' $(Fy \wedge Gx)$ ' does not come from ' $(Fx \wedge Gx)$ ' by *proper* substitution of 'y' for 'x', since that involves replacing *all* free occurrences of 'x'. *Argument (v*)* contrasts with (v): the problem with (v*) is that occurrences of 'x' that I now emphasize are not free in ' $(Fx \rightarrow \Lambda xGx)$ '; so ' $(Fy \rightarrow \Lambda yGy)$ ' does not come from ' $(Fx \rightarrow \Lambda xGx)$ ' by *proper* substitution of 'y' for 'x', which involves replacing *only free* occurrences of 'x'. *Argument (vi*)* contrasts with (vi): the problem with (vi*) is that the now emphasized occurrence of 'y' is not free in ' $\forall y(Fy \rightarrow Gy)$ '; so ' $\forall y(Fy \rightarrow Gy)$ ' does not come from ' $\forall y(Fx \vee Gy)$ ' by *proper* substitution of 'y' for 'x', since that involves replacing free occurrences of 'x' by *free* occurrences of 'y'.¹⁴

5.3.1.3 One can think of UI as a two-stage affair. Given a universal generalization, to perform a UI-inference: first, remove the leading universal quantifier and take with it each occurrence of its variable that it binds – the result will usually be an 'open formula'; second, insert in each open

¹⁴UI has unintuitive applications of only theoretical interest. Given the exact statement of the rule, and that all that is required is that φ' comes from φ by proper substitution of β for α , the following arguments involving cases of vacuous quantification are valid by UI-inferences:

$$\frac{\Lambda xP}{P} \quad \frac{\Lambda x\forall x(Fx \wedge Gx)}{\forall x(Fx \wedge Gx)}$$

Recall that by definition φ' comes from φ by proper substitution of β for α exactly in case φ' is exactly like φ except for having free occurrences of β *wherever* φ has free occurrences of α : this definition covers the case in which there are no free occurrences of α in φ . In that case φ' must be exactly like φ . (See the previous footnote.)

place some one term, taking care that it is free in each of these places. For example, given

$$\wedge y[Fx \rightarrow (\vee yGy \rightarrow \vee x(Fy \wedge Gx) \wedge Hy)],$$

to UI, first remove ' $\wedge y$ ' and each ' y ' it binds,

$$Fx \rightarrow (\vee yGy \rightarrow \vee x(F \wedge Gx) \wedge H-),$$

then insert *any term* in the underlined open places *except* ' x ', for it would be bound in the first of these place. Almost anything goes when one UI's. *Which* term (other than ' x ') you choose to insert *will depend on the derivation-problem* on which you are working, but almost any term will be 'legal'. The exceptions are terms that when inserted are bound.

5.3.1.4 **N.B.** For final important points, that address common misunderstanding, we have two more arguments that are *not valid by UI-inferences*.

$$\begin{array}{c} \text{(vii*)} \\ \frac{\wedge xFx \rightarrow \wedge xGx}{Fx \rightarrow Gx} \end{array} \qquad \begin{array}{c} \text{(viii*)} \\ \frac{\vee y\wedge x(Fx \rightarrow Gy)}{\vee y(Fx \rightarrow Gy)} \end{array}$$

The rule UI applies to *universal generalizations*: the premise of (vii*) is a conditional, not a universal generalization. The rule UI, like every rule, 'applies to whole sentences' and not, as it would do for argument (viii*), to parts.

5.3.2 Existential generalization – EG

5.3.2.1 The intuitive idea of EG is that if some condition is satisfied by a particular thing, then there is something that satisfies: for example, given that *the moon* is made of green cheese, it follows that *something* is made of green cheese. EG goes from talk about a particular thing, to related less informative general talk. Formally, the rule is,

$$\varphi' \therefore \vee \alpha \varphi,$$

wherein φ' comes from φ by proper substitution of some term β for α . Given this relation of proper substitution, φ' says of β , everything that φ says of α ; which is the 'idea' of UI. Here for illustration are several arguments whose conclusions follow by EG-inferences.

$$\begin{array}{ccc} \text{(i')} & \text{(ii')} & \text{(iii')} \\ \frac{Fx \wedge Hx}{\vee x(Fx \wedge Hx)} & \frac{Fy \wedge Hy}{\vee x(Fx \wedge Hx)} & \frac{Fx \wedge Hy}{\vee x(Fx \wedge Hy)} \\ \text{(iv')} & \text{(v')} & \text{(vi')} \\ \frac{FA \rightarrow GA}{\vee x(Fx \rightarrow GA)} & \frac{Fy \rightarrow \wedge xGx}{\vee x(Fx \rightarrow \wedge xGx)} & \frac{\vee y(Fz \rightarrow Gy)}{\vee x\vee y(Fx \rightarrow Gy)} \end{array}$$

Argument (iv') can seem wrong, but consider that *one* '(FA \rightarrow GA)' says of A is that the open sentence '(F \rightarrow GA)' is true of it; from this it does follow that this open sentence is true of *something*, which is what ' $\vee x(Fx \rightarrow GA)$ ' says. So argument (iv') is intuitively valid by EG. And it is so strictly and formally, since '(FA \rightarrow GA)' (φ' in the rule) does come from '(Fx \rightarrow GA)' (the φ of the rule) by proper substitution of 'A' (the β of the rule) for 'x' (the α of the rule). We come back to argument (iv') shortly for another way of seeing that it is valid by an EG-inference.

5.3.2.2 Comparison of the rules EG and UI reveals a useful connection, specifically, that for any formulas φ and φ' , the argument,

$$\wedge \alpha \varphi \therefore \varphi',$$

is valid by a UI-inference if and only if the argument,

$$\varphi' \therefore \vee \alpha \varphi,$$

is valid by an EG-inference: correct applications of the two rules are *paired*. Going back as promised to argument (iv'), this means that it,

$$FA \rightarrow GA \therefore \vee x(Fx \rightarrow GA),$$

is valid by EG, if and only if argument (iv),

$$\wedge x(Fx \rightarrow GA) \therefore FA \rightarrow GA,$$

is valid by UI. For most people it is easier to see that (iv) is valid by UI, than to see that (iv') is valid by EG; for them, checking out the former is one handy way of confirming the latter.

5.3.2.3 To help fix ideas you might say of the following which are 'valid by either UI- or EG-inferences'. [The question is *not*, Which of the following are valid period?, but, Which of the following are such that there conclusions can be directly inferred by either UI or EG?]

- | | |
|------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| (1) $\frac{Fa \rightarrow VyGy}{\forall x(Fx \rightarrow VyGy)}$ | (5) $\frac{\Lambda x \forall y(Fy \rightarrow Gx)}{\forall y(Fy \rightarrow Gy)}$ |
| (2) $\frac{Fy \rightarrow VyGy}{\forall xFx \rightarrow VyGy}$ | (6) $\frac{Fy \wedge Gy}{\forall x(Fx \wedge Gy)}$ |
| (3) $\frac{\Lambda y(Fy \rightarrow Gy)}{Fx \rightarrow Gy}$ | (7) $\frac{\Lambda x(Fx \rightarrow Vy(Gy \wedge Hx))}{Fz \rightarrow Vy(Gy \wedge Hz)}$ |
| (4) $\frac{\Lambda x(Fy \rightarrow Gy)}{Fx \rightarrow Gx}$ | (8) $\frac{\Lambda x \Lambda y(Fy \rightarrow Gx)}{\Lambda x(FA \rightarrow Gx)}$ |

5.3.3 Existential Instantiation – EI

5.3.3.1 The intuitive idea of EI is that, given that something or other satisfies some condition C, one can when reasoning pick a name for some such thing – a name that is new in the context of one’s reasoning – and say that that thing satisfies the condition. Picking a name new to the context avoids begging questions concerning what else holds of something that satisfies condition C other than this, that it does satisfy this condition.

The rule EI formalizes for use in derivations of MQC this familiar procedure of informal reasoning. Here is this rule:

$$\forall \alpha \phi \quad \therefore \quad \phi'$$

wherein ϕ' comes from ϕ by proper substitution of some *variable* that is *absolutely new to the derivation* for α .

There are two special restrictions on EI. First, in contrast to UI, β must be a variable, not a name letter. And second, β cannot occur on a preceding line of the derivation: this second restriction is incorporated a special clause of the definition of a derivation in MQC (*cf.*, *Logic*, pp. 141 and 144). This second restriction is absolute, and extends to lines that contain uncanceled '*SHOW*' and to lines that are already boxed. The first of these restrictions is required for technical reasons to do with the design of the deductive system. The *second* restriction formalizes the requirement of natural informal reasoning that the name picked for some such thing be new to the context of one’s reasoning.

5.3.3.2 But for the second requirement that β be new to the derivation, it would be possible to construct derivations for obvious invalid arguments. For example, ignoring this restriction, we could validate by derivation the obviously invalid argument,

$$\forall xFx. \quad \forall xGx \quad \therefore \quad \forall x(Fx \wedge Gx)$$

It is clear that this argument must be invalid: that something is an F, and something is a G, does not entail that something is *both* an F and a G. For example, something is an even number, and something is an odd number, but it is false that something is both an even number and an odd number.

Here is a *defective* derivation for this argument, whose only fault is that it ignores the restriction on EI that says that the variable introduced must be new to the proof.

- | | |
|----------------------------------------------|-----------------------------------------------------------------------------|
| 1. SHOW $\forall x(Fx \wedge Gx)$ | (7, DD) |
| 2. $\forall xFx$ | premise |
| 3. $\forall xGx$ | premise |
| 4. Fa | 2, EI |
| 5. Ga | 3, EI [an incorrect application of EI – 'a' is <i>not</i> new to the proof] |
| 6. $Fa \wedge Ga$ | 4, 5, Adj |
| 7. $\forall x(Fx \wedge Gx)$ | 6, EG |

There is not a correct derivation for the argument, which is good, since the argument is invalid.

5.3.3.3 The first restriction that says that β must be a *variable* also serves to block derivations of invalid arguments. But for this restriction one could construct a derivation for the following the argument,

$$FA. \quad \forall xGx \quad \therefore \quad \forall x(Fx \wedge Gx),$$

though it is clear that this argument is not valid: that A is an F, and that something or other is a G, does not entail that something or other is both an F and a G. Consider that 1 is an odd number, and that there is an even number, but that it is false that there is a number that is both odd and even. Here is a derivation for this argument whose only fault is that it ignores the restriction on EI that says that the term introduced must be *variable*.

- | | |
|----------------------------------------------|---------------------------------------------------------------------------------------------------|
| 1. SHOW $\forall x(Fx \wedge Gx)$ | (6, DD) |
| 2. $\forall xGx$ | premise |
| 3. GA | 3, EI [an incorrect application of EI – while 'A' is new to the derivation, it is not a variable] |
| 4. FA | premise |
| 5. $FA \wedge GA$ | 3, 4, Adj |

6. $\boxed{\forall x(Fx \wedge Gx)}$ 5, EG

5.3.3.4 EI,

$\forall \alpha \varphi \therefore \varphi'$,

is like UI,

$\wedge \alpha \varphi \therefore \varphi'$,

but simpler. For UI, φ' can come from φ by proper substitution of any variable or name letter, and so applications of UI present *real choices* concerning what term to use, what term will work best in the derivation. For EI, φ' must come from φ by proper substitution of a variable that is new to the derivation, which means that applications of EI present only *trivial choices*, since it cannot matter which new variable one uses.

5.4 Universal Derivation – UD

5.4.1 The system of *Logic* differs from most other natural deduction systems when it comes to its way of ‘getting’ universal generalizations. Instead of having a specially restricted fourth form of inference which might be called ‘Universalization Generalization – UG’, it has a special form of proof for these generalizations. This makes its procedures *more natural*. The special form of proof explicates a familiar form of informal deductive reasoning.

5.4.2 To prove something about all triangles it is natural to consider a particular triangle and to reason about it taking care not to rely on any feature that is particular to it and not shared by all triangles. One treats it as a *representative* triangle, taking care that whatever is proved about it could have been proved about any other triangle, so that what is established, even though established specifically for it, will hold for all triangles. Universal derivation formalizes, and in MQC makes fool-proof, that familiar procedure for reasoning about representative particulars, taking care not to compromise their representativeness, and then generalizing any conclusions reached about them.

Universal derivation is a provision for cancelling a *SHOW*-line and boxing lines following it, *under certain conditions*. In its simplest form it says that, given the line

SHOW $\wedge \alpha \varphi$

if, (i), there are no subsequent uncanceled *SHOW*-lines and, (ii), φ occupies a subsequent line, then one may box all subsequent lines and cancel that displayed '*SHOW*', if, (iii), *a does not occur free on a line that is 'antecedent' to this SHOW-line*. An 'antecedent' line, recall, is an available

line, one that is not boxed and that does not contain uncanceled '*SHOW*'.¹⁵

To see the reasonableness of this form of proof, consider that only lines on which α was free would 'say anything about α '. *When there are no such lines, it is clear that 'whatever one can prove about α one could prove about anything β '*: more precisely, when there are no such lines, one can derive φ if and only if one can, for any term at all β , derive φ' where φ' comes from φ by proper substitution of β for α .

5.4.3 The restriction on UD is considerably *weaker* than the restriction on EI. For UD, the variable of generalization α cannot occur *free* on a line that is '*antecedent*' to the *SHOW*-line: it can occur bound on lines preceding the *SHOW*-line, and it can occur free on preceding lines that are not 'antecedent' to it (i.e., on preceding lines that are boxed or that contain '*SHOW*' uncanceled). For EI, the variable of instantiation cannot occur (free or bound) even on a preceding boxed or uncanceled *SHOW*-line.

The restriction on UD makes sense. Also, in its absence it would be possible to construct derivations for

¹⁵An equivalent, but more complicated, procedure would replace (ii) and (iii) by,

(ii'), φ' occupies a line under this *SHOW*-line, where φ' comes from φ by proper substitution of β for α ,

and

(iii), the variable β is not free in a formula that occupies an available line above this *SHOW*-line.

Let clauses (i), (ii'), and (iii') define UD*. This is like Daniel Bonevac's adaptation of UD except that he requires that β should be "a *constant* [i.e., a name letter] new to the proof" (Bonevac 1987, p. 176, italics added). In his system a derivation for an argument with premises must begin with lines on which these are entered. Only then can a *SHOW*-line for its conclusion be entered. (Cf., p. 92.) Otherwise there would be a universal derivation in his system for the argument $\neg FA \therefore \wedge xFx$.

The more complicated than UD procedure UD* may be compared with the rule Universal Generalization

$\varphi' \therefore \wedge \alpha \varphi$,

wherein φ' comes from φ by proper substitution of a variable β for α ,

that we do *not* have, and that we could have only if we subjected it to *restrictions that insured that β could be treated as a representative individual*. These restrictions could not be as simple as those of UD. Nor could they be as 'natural' and as simply 'motivated'. For example, C. Stephen Layman uses the rule

$\varphi' \therefore \wedge \alpha \varphi$,

wherein φ' comes from φ by proper substitution of a variable β for α , subject to the restrictions, (i), that β "must not be free in a line arrived at by EI," and, (ii), β does not occur "free in [an antecedent] assumption" (Layman 2000, pp. 336 and 347-8).

patently invalid arguments. For example, a derivation for the argument,

$$\forall xFx \therefore \Lambda xFx,$$

could be constructed.

1.	SHOW ΛxFx	(UD: x)
2.	$\forall xFx$	premise
3.	Fa	2, EI
4.	SHOW ΛaFa	assertion (UD – this is a mistake; 'a' is free on 'antecedent' line (3), so UD box-and-cancel is not licensed)
5.	Fa	3, R
6.	Fx	4, UI

5.4.4 The new form of derivation, UD, is specifically for deriving universal generalizations, but it is not the only way in which such formulas can be derived. The procedures of direct and indirect derivation are available for all formulas, and so they are available for these formulas. In particular, it is permitted to derive them indirectly. To illustrate, the following derivation – which uses a form of the derived rule of Quantifier Negation – for,

$$\Lambda x[Fx \wedge \Lambda xGx \rightarrow \Lambda x(Fx \rightarrow Gx)],$$

is entirely correct.

1.	SHOW $\Lambda x[Fx \wedge \Lambda xGx \rightarrow \Lambda x(Fx \rightarrow Gx)]$	(2, UD: x)
2.	SHOW $Fx \wedge \Lambda xGx \rightarrow \Lambda x(Fx \rightarrow Gx)$	(4, CD)
3.	$Fx \wedge \Lambda xGx$	(CD)
4.	SHOW $\Lambda x(Fx \rightarrow Gx)$	(8, 10, ID)
5.	$\sim \Lambda x(Fx \rightarrow Gx)$	(ID)
6.	$\forall x \sim (Fx \rightarrow Gx)$	5, QN [a derived rule explained in the next section]
7.	$\sim (Fa \rightarrow Ga)$	6, EI
8.	$\sim Ga$	7, C-neg
9.	ΛxGx	3, S
10.	Ga	9, UI

Since 'x' was free on a line 'antecedent' to (4), the 'SHOW' on (4) could not be cancelled by UD even if, as could have been arranged, '(Fx → Gx)' was on a line under that SHOW-line. It has been cancelled, not by UD, but by ID given the contradiction under it on lines (8) and (10). There are no variable restrictions on ID.

5.4.5 Assumptions and universal derivation

“[A]ssumptions [can] be made [only] immediately after the [SHOW] line with which [they are] connected...[But] neglect [of this restriction] will produce no fallacies until our logical apparatus is enlarged by the introduction [of Universal Derivation].” (*Logic*, p. 36.)

But for this restriction on entries of assumptions, the argument,

$$0 \text{ is not an odd number. } \therefore \text{ Every odd number is an even number.}$$

would be valid in the Monadic Quantifier Calculus. The argument has the symbolization,

$$\sim FA \therefore \Lambda x(Fx \rightarrow Gx)$$

under the scheme – A: 0; F: *a* is an odd number; G: *a* is an even number. Now comes a derivation that flaunts the ban on 'delayed assumptions'.

1.	SHOW $\Lambda x(Fx \rightarrow Gx)$	2, UD
2.	SHOW $Fx \rightarrow Gx$	5, 6, ID
3.	SHOW ΛxFx	(4, UD)
4.	Fx	delayed CD-assumption for (2)
5.	$\sim FA$	premise
6.	FA	3, UI

5.4.6 We have been considering universal derivation in its simplest form. For convenience it is given a form that simplifies derivations of multiply general sentences. The rule for UD is that, given

$$SHOW \Lambda a_1 \dots \Lambda a_n \phi,$$

one can box-and-cancel if none of the variables of generalization a_1, \dots, a_n is free on an antecedent line, ϕ is on a subsequent unboxed line, and there are no subsequent uncanceled SHOW-lines. So, to derive ' $\Lambda x \Lambda y (Fx \rightarrow Gy)$ ' one can begin thus,

$$SHOW \Lambda x \Lambda y (Fx \rightarrow Gy) \\ SHOW Fx \rightarrow Gy$$

making sure that neither 'x' nor 'y' is free on an 'antecedent' line, rather than proceed thus,

- (i) *SHOW* $\lambda x \lambda y (Fx \rightarrow Gy)$
- (ii) *SHOW* $\lambda y (Fx \rightarrow Gy)$
- (iii) *SHOW* $Fx \rightarrow Gy$

In LOGIC 2000, however, this sequence is mandatory. In this program to cancel (i) by UD, one needs ' $\lambda y (Fx \rightarrow Gy)$ '. The required sequence of Show-lines can be generated in LOGIC 2000 by repeated uses of the command SHOW INST.

5.5 Quantifier Negation – Four Derived Inference Rules

5.5.1 We have available for use, almost from the start, the following four derived inference rules QN:

$$\begin{array}{ll} \sim \Lambda \alpha \varphi & /: \forall \alpha \sim \varphi & \sim \forall \alpha \varphi & /: \Lambda \alpha \sim \varphi \\ \forall \alpha \sim \varphi & /: \sim \Lambda \alpha \varphi & \Lambda \alpha \sim \varphi & /: \sim \forall \alpha \varphi \end{array}$$

Words can help to make obvious the correctness of these rules. For example, the first one says that from 'not all' one can infer 'some not', and the third one goes the other way; the fourth one says that from 'all not' one can infer 'not some', and the second one reverses this inference.

5.5.2 These rules can be seen to be generalizations of the sentential DeMorgan rules. For example, the QN rule,

$$\sim \forall \alpha \varphi \quad /: \quad \Lambda \alpha \sim \varphi,$$

generalizes the sentential DeMorgan rule,

$$\sim (\varphi \vee \psi) \quad /: \quad (\sim \varphi \wedge \sim \psi).$$

Suppose there are exactly n things; use 'A' with subscripts to name them; and consider that ' $\sim \forall x Fx$ ' is true if and only if

$$\sim (FA_1 \vee \dots \vee FA_n)$$

is true, and that ' $\Lambda x \sim Fx$ ' is true if and only if it is true that,

$$(\sim FA_1 \wedge \dots \wedge \sim FA_n).$$

5.5.3 *À propos* Section 5.4.4 above and Section 5.6.1 below, it can be seen that the procedure of Universal Derivation is needed to derive the QN-rule

$$\Lambda \alpha \varphi \quad /: \quad \forall \alpha \sim \varphi .$$

That is, Universal Derivation is needed for an unabbreviated derivation of the conditional theorem that underlies this rule, which is ' $\sim \Lambda x Fx \rightarrow \forall x \sim Fx$ '. (A derivation of that conditional is subsidiary in the derivation of T203 in *Logic*, p. 148.)

5.6 *Abbreviated derivations.* Provisions for abbreviations are essentially as for derivations in The Sentential Calculus with two major adjustments.

First, an instances of an SC- theorem is, for purposes of MQC, any formula that comes from such a theorem by uniform replacements of formulas of MQC for sentence letters. The license to use theorems remains a license to use instances of *sentential* theorems. The idea of an instance of other theorems, including those of MQC, is not defined until Chapter VII of *Logic*.

Second, when combining lines, EI-inferences can be included only as last steps. (*Cf.*, *Logic*, p. 151.) This insures that new variables introduced by EI-inferences occur on lines of even abbreviated derivations. This is important for subsequent EI-inferences, since it is required "that the variable of instantiation [of an EI-inference] does not occur in any preceding line" (p. 144). Please see comments under illustration 18 in Section 6 for some elaboration of this matter of combining lines, along with comments regarding how the license to queue is different, and more liberal, in LOGIC 2000.

5.7 *Strategies, advice, and cautions for abbreviated derivations.* We add to the strategies assembled for derivations in The Sentential Calculus, strategies for getting generalizations, some 'quantificational advice', and two cautions concerning uses of theorems.

5.7.1 *To derive a universal generalization,*

$$\Lambda \alpha \varphi,$$

enter a *SHOW*-line for this universal generalization and then, without making an assumption, enter a *SHOW*-line for φ for a universal derivation, *unless* α is free on an 'antecedent' line, in which case assume, for an indirect derivation,

$$\sim \Lambda \alpha \varphi,$$

and use QN to infer from this

$$\forall \alpha \sim \varphi.^{16}$$

It is best to use universal derivation for $\ulcorner \forall \alpha \varphi \urcorner$ on a line when possible, that is, when α is not free on an antecedent line. Universal derivation makes simpler and more natural proofs, *even though*, whenever it is possible to use it for $\ulcorner \forall \alpha \varphi \urcorner$ it is, given the QN-rule,

$$\Lambda \alpha \varphi \text{ } /: \text{ } \forall \alpha \sim \varphi ,$$

possible to use indirect derivation instead. Two observations suffice for this point: (i) from $\ulcorner \forall \alpha \sim \varphi \urcorner$ one can derive $\sim \varphi'$ such that φ' comes from φ by proper substitution of β , a variable new to the derivation, from φ ; and, (ii), when α is not free on an antecedent line, if, for a universal derivation, one can derive φ , then, for an indirect derivation, one can derive this φ' . It follows from (i) and (ii) that if $\sim \Lambda \alpha \varphi \text{ } /: \text{ } \forall \alpha \sim \varphi -$ were a primitive rule, universal derivation would not be needed.

5.7.2 To derive an existential generalization,

$$\forall \alpha \varphi,$$

if possible get more or less directly φ' , where φ' comes from φ by proper substitution of some term β for α , and then use EG to infer $\ulcorner \forall \alpha \varphi \urcorner$. Otherwise, for an indirect derivation, assume,

$$\sim \forall \alpha \varphi,$$

and use QN to infer,

$$\Lambda \alpha \sim \varphi.$$

5.7.3 Three principles of 'good practice'.

In so far as possible, begin universal derivations before instantiating universal generalizations. For the reason, please think about the constraint on UD (the variable of generalization must not be free on a line antecedent to the *SHOW*-line) and the unrestricted license to UI to any term (subject to the proper substitution condition).

When instantiating, in so far as possible, use EI before using UI.¹⁷ The reason for this precept is that it is important to minimize the number distinct variables that are free in a derivation.

When using UI, favour instantiating to names that occur in the argument, or to variables that are free on available lines.¹⁸ Otherwise simply instantiate $\ulcorner \Lambda \alpha \varphi \urcorner$ to the variable of generalization α .

5.7.4 Essential cautions regarding uses of theorems

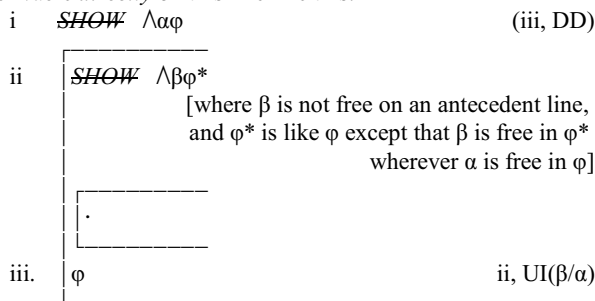
Only instances of already proved **sentential** theorems can be used in MQC and QC abbreviated derivations. (*Logic*, pp. 150 and 220.)

A biconditional theorem $\ulcorner (\varphi \leftrightarrow \psi) \urcorner$ can be used to infer, from an instance of φ that **occupies** a line by, the corresponding instance of ψ . But it cannot be used to infer, from an instance of φ that is merely **part** of a formula that occupies of line, the formula in which that part is replaced by the corresponding instance of ψ .

For example, T90 can be used to infer, from $\ulcorner \sim (Fx \leftrightarrow Gx) \urcorner$ on a line, the formula $\ulcorner (Fx \leftrightarrow \sim Gx) \urcorner$. But it *cannot* be used to infer, from $\ulcorner \Lambda x \sim (Fx \leftrightarrow Gx) \urcorner$, the formula $\ulcorner \Lambda x (Fx \leftrightarrow \sim Gx) \urcorner$.

The license, Interchange of Equivalents, for this latter use, is not introduced until Chapter VII, *Logic*, p. 364.

¹⁶Recourse to *indirect* derivation for $\ulcorner \Lambda \alpha \varphi \urcorner$ on a line when α is free on an antecedent line is not *necessary*. If $\ulcorner \Lambda \alpha \varphi \urcorner$ is derivable on a line though α is free on an antecedent line, then it is derivable *directly* on this line like this:



For if $\ulcorner \Lambda \alpha \varphi \urcorner$ is derivable (somehow) on line i, then $\ulcorner \Lambda \beta \varphi^* \urcorner$ is derivable on line ii. Indeed, given that β is not free on a line antecedent to line ii, $\ulcorner \Lambda \beta \varphi^* \urcorner$ is derivable on line ii by Universal Derivation.

¹⁷Never UI before you EI. That, according to David Kaplan, duly qualified, is the eleventh commandment. According to Warner Wick, however, the eleventh commandment is, *Never spit to the windward*.

¹⁸However, by the rule UI, φ' comes from φ by *proper* substitution of a term β for the variable of generalization α : in short, 'we UI to β '. Suppose the only term that is free on a preceding unboxed line is 'y'. Then if $\ulcorner \Lambda x (Fx \rightarrow Gx) \urcorner$ is to be instantiated, we UI to 'y' for $\ulcorner (Fy \rightarrow Gy) \urcorner$, *no question*. However, if $\ulcorner \Lambda x \forall y (Fx \wedge Gy) \urcorner$ is to be instantiated, *it is not possible to UI to 'y'*, since the 'y' that was substituted for 'x' in $\ulcorner \forall y (Fx \wedge Gy) \urcorner$ would be *bound*, and the substitution *not proper*.

The case is similar to that of inference rules such as DN that 'go both ways'. While one can by DN infer, from 'Fx', the formula ' $\sim\sim Fx$ ', one *cannot* infer from ' $\wedge xFx$ ', the formula ' $\wedge x\sim\sim Fx$ '.

6. DERIVATION ILLUSTRATIONS

(1) $\wedge x(Fx \rightarrow \sim\forall xGx) \therefore \forall xFx \rightarrow \wedge x(Gx \rightarrow Fx)$

1.	SHOW $\forall xFx \rightarrow \wedge x(Gx \rightarrow Fx)$	(4, CD)
2.	$\forall xFx$	(CD)
3.	$\wedge x(Fx \rightarrow \sim\forall xGx)$	premise
4.	SHOW $\wedge x(Gx \rightarrow Fx)$	(5, UD: x) ¹⁹
5.	SHOW $Gx \rightarrow Fx$	(7, CD)
6.	Gx	(CD)
7.	SHOW Fx	(13, 14, ID)
8.	$\sim Fx$	(ID)
9.	Fa	2, EI
10.	$Fa \rightarrow \sim\forall xGx$	3, UI
11.	$\sim\forall xGx$	9, 10, MP
12.	$\wedge x\sim Gx$	11, QN
13.	$\sim Gx$	12, UI
14.	Gx	6, R

Lines (10) through (13) could be combined into,

(10') $\sim Gx$ 3, UI, 9, MP, QN, UI

Line (9) must stand. A sequence of combined lines can end with an EI-inferences, but cannot involve an EI-inference at an earlier point. (See clause (9), *Logic*, p. 151.)

1.	SHOW $\forall xFx \rightarrow \wedge x(Gx \rightarrow Fx)$	(CD)
2.	$\forall xFx$	(CD)

In order not to forget about them, it is best to enter premises on lines as soon as possible, which is *immediately after the initial assumption if any*.

3. $\wedge x(Fx \rightarrow \sim\forall xGx)$ premise

¹⁹Including 'x' in this annotation provides an occasion to confirm that 'x' is not free on an antecedent line.

The sentence ' $\wedge x(Gx \rightarrow Fx)$ ' will complete the derivation. It is a universal generalization, and, since its variable of generalization 'x' is not free on an antecedent or available line, we begin a universal derivation for it. This involves first a *SHOW*-line for the generalization itself, and, *without an intervening assumption*, in this case a *SHOW*-line for its 'interior', ' $(Gx \rightarrow Fx)$ ', since that is what suffices for a universal derivation, and since there are no obvious prospects for simply inferring it from lines (2) and (3).

We want to show that the open formula ' $(G \rightarrow F)$ ' holds for *everything*. To do this we show that it holds for 'x'. Given that 'x' is not free on an antecedent line, what is established for 'x' could be established for 'y', 'z', 'a', 'B', or *any* term: so what is established for 'x' has general significance.

4.	SHOW $\wedge x(Gx \rightarrow Fx)$	(5, UD: x)
5.	SHOW $Gx \rightarrow Fx$	(7, CD)
6.	Gx	(CD)
7.	SHOW Fx	(ID) It turns out that this line and the next one are not needed.
8.	$\sim Fx$	(ID)

It is time to instantiate generalizations and to get down to quantifier-free formulas to which rules and procedures of sentential logic can be applied. Since EI calls for a variable that is new to the derivation, *in so far as possible we use EI before using UI*. This way we can, if we want, 'follow suit' when we UI. We can then UI to the same variable. We want to keep the number of free variables on available lines down to a minimum to increase the chances of lines 'going together' for sentential inferences.

I recommend using 'a' for the first EI in a derivation, 'b' for the second one, etc. This way EI variables stand out, and using a new one for each EI becomes automatic.

9. $||| Fa$ 2, EI

We now apply UI to line 3. I think of a UI-inference as proceeding in two steps. First, to remove the initial universal quantifier phrase, here ' $\wedge x$ ', and with it every occurrence of its variable, here 'x', that it binds. The result will generally be an 'open formula,' here ' $F \rightarrow \sim\forall xGx$ '. Second, to insert a term in the open positions. The only restriction is that this term must be free in these positions. Subject to that restriction, 'any term goes'.

So there is a question every time UI is applied. To which term *shall one* 'go'? If no term is free on a preceding unboxed line, put back the variable occurrences of which have been removed. Otherwise insert a term that is free on a preceding unboxed line. If there are several such terms,

	$\Lambda x \Lambda y (Fx \wedge Gy \rightarrow Hx \wedge Jy)$ (4, CD)	
4.	$\Lambda x (Fx \rightarrow Hx) \wedge \Lambda x (Gx \rightarrow Jx)$ (CD)	
5.	SHOW $\Lambda x \Lambda y (Fx \wedge Gy \rightarrow Hx \wedge Jy)$ (6, UD: x, y)	
6.	SHOW $Fx \wedge Gy \rightarrow Hx \wedge Jy$ (10, CD)	
7.	$Fx \wedge Gy$ (CD)	
8.	Hx 4, S, UI, 7, S, MP	
9.	$Gy \rightarrow Jy$ 4, S, UI	
10.	$Hx \wedge Jy$ 7, S, 9, MP, 8, Adj	
11.	SHOW $\Lambda x \Lambda y (Fx \wedge Gy \rightarrow Hx \wedge Jy) \rightarrow$ $\Lambda x (Fx \rightarrow Hx) \wedge \Lambda x (Gx \rightarrow Jx)$ (27, CD)	
12.	$\Lambda x \Lambda y (Fx \wedge Gy \rightarrow Hx \wedge Jy)$ (CD)	
13.	SHOW $\Lambda x (Fx \rightarrow Hx)$ (14, UD)	
14.	SHOW $Fx \rightarrow Hx$ (19, CD)	
15.	Fx (CD)	
16.	$\forall x Gx$ 2, S	
17.	Ga 16, EI	
18.	$Hx \wedge Ja$ 12, UI, UI, 15, 17, Adj, MP	
19.	Hx 17, S	
20.	SHOW $\Lambda x (Gx \rightarrow Jx)$ (21, UD)	
21.	SHOW $Gx \rightarrow Jx$ (26, CD)	
22.	Gx (CD)	
23.	$\forall x Fx$ 2, S	
24.	Fb 23, EI	
25.	$Fb \wedge Gx \rightarrow Hb \wedge Jx$ 11, UI, UI	
26.	Jx 22, 24, Adj, 25, MP, S	
27.	$\Lambda x (Fx \rightarrow Hx) \wedge \Lambda x (Gx \rightarrow Jx)$ 13, 20, Adj	
28.	$\Lambda x (Fx \rightarrow Hx) \wedge \Lambda x (Gx \rightarrow Jx) \leftrightarrow$ $\Lambda x \Lambda y (Fx \wedge Gy \rightarrow Hx \wedge Jy)$ 3, 11, CB	

For extra practice you might try,

$$\therefore (\forall x Fx \leftrightarrow \forall x Gx) \wedge \Lambda x \Lambda y [Fx \wedge Gy \rightarrow (Hx \leftrightarrow Jy)] \rightarrow [\Lambda x (Fx \rightarrow Hx) \leftrightarrow \Lambda x (Gx \rightarrow Jx)]$$

$$(5) \therefore \sim \forall x \Lambda y (Fx \wedge Gy) \rightarrow (\forall x Fx \rightarrow \sim \Lambda x Gx)$$

$$1. \text{ **SHOW** } \sim \forall x \Lambda y (Fx \wedge Gy) \rightarrow (\forall x Fx \rightarrow \sim \Lambda x Gx) \quad (3, CD)$$

2.	$\sim \forall x \Lambda y (Fx \wedge Gy)$ (CD)	
3.	SHOW $\forall x Fx \rightarrow \sim \Lambda x Gx$ (5, CD)	
4.	$\forall x Fx$ (CD)	
5.	SHOW $\sim \Lambda x Gx$ (11, 12, ID)	
6.	$\Lambda x Gx$ (ID)	
7.	$\Lambda x \sim \Lambda y (Fx \wedge Gy)$ 2, QN	
8.	Fa 4, EI	

To which shall we apply UI first, line 6 or line 7? We should 'work on' line 7, because *eventually* an occasion for EI will obtain, and *in so far as possible* we want to EI before we UI. This is because it is necessary to use a variable for EI that is new to the proof.

9.	$\sim \Lambda y (Fa \wedge Gy)$ 7, UI	
10.	$\forall y \sim (Fa \wedge Gy)$ 9, QN	
11.	$\sim (Fa \wedge Gb)$ 10, EI	
12.	$Fa \wedge Gb$ 6, UI, 8, Adj	

$$(6) \Lambda x (Fx \rightarrow Gx) \therefore \Lambda x [Fx \wedge \sim \forall y (Gx \wedge Hy) \rightarrow \forall x \sim Hx]$$

1.	SHOW $\Lambda x [Fx \wedge \sim \forall y (Gx \wedge Hy) \rightarrow \forall x \sim Hx]$ (3, UD: x)	
2.	$\Lambda x (Fx \rightarrow Gx)$ premise	
3.	SHOW $Fx \wedge \sim \forall y (Gx \wedge Hy) \rightarrow \forall x \sim Hx$ (5, CD)	
4.	$Fx \wedge \sim \forall y (Gx \wedge Hy)$ (CD)	

It is not clear that we can get an ' $\sim H$ ' and infer ' $\forall x \sim Hx$ ' by EG, that is, it is not clear that ' $\forall x \sim Hx$ ' can be derived directly. So we begin an indirect derivation for it.

5.	SHOW $\forall x \sim Hx$ (10, 11, ID)	
6.	$\sim \forall x \sim Hx$ (ID)	
7.	$\Lambda y \sim (Gx \wedge Hy)$ 4, S, QN	
8.	$\Lambda x \sim \sim Hx$ 6, QN	

We can get, from line 7, ' $\sim (Gx \wedge H-)$ ', with *any* variable we want in the hole. An idea is to get ' $(Gx \wedge H-)$ ' with some variable for a contradiction. We use lines 4 and 2 for ' Gx ' and 8 for ' Hx '.

9.	Gx 4, S, 2, UI, MP	
10.	$Gx \wedge Hx$ 8, UI, DN, 9, Adj	
11.	$\sim (Gx \wedge Hx)$ 7, UI	

Instead of the indirect derivation begun on line (5), ' $\forall x \sim Hx$ ' could have been derived directly thus:

5.	$Fx \rightarrow Gx$		3, UI
6.	Gx		4, S, 5, MP
7.	$\wedge y \sim (Gx \wedge Hy)$		4, S, QN
8.	$\sim (Gx \wedge Hx)$		7, UI
9.	$\sim Gx \vee \sim Hx$		8, DeM
10.	$\sim Hx$	6, DN, 9, MTP	
11.	$\forall x \sim Hx$	10, EG	

(7) Using disjunctions

$\wedge x (Fx \rightarrow \wedge x Gx), \wedge x Jx \vee \sim \wedge x (Gx \vee Hx) \therefore \wedge x (Fx \rightarrow Jx)$

1.	<i>SHOW</i> $\wedge x (Fx \rightarrow Jx)$		(4, UD: x)
2.	$\wedge x (Fx \rightarrow \wedge x Gx)$		premise
3.	$\wedge x Jx \vee \sim \wedge x (Gx \vee Hx)$		premise
4.	<i>SHOW</i> $Fx \rightarrow Jx$		(6, CD)
5.	Fx		(CD)
6.	<i>SHOW</i> Jx		(12, 14, ID)
7.	$\sim Jx$		(ID)

Now what? How to use line (3)? One idea is to go after ' $\sim \wedge x Jx$ ' for MTP. It turns out that we can get this negation directly.

8.	$\forall x \sim Jx$		7, EG
9.	$\sim \wedge x Jx$		8, QN
10.	$\sim \wedge x (Gx \vee Hx)$		3, 9, MTP
11.	$\forall x \sim (Gx \vee Hx)$		10, QN
12.	$\sim (Ga \vee Ha)$		11, EI
13.	$\wedge x Gx$		2, UI, 5, MP
14.	$Ga \vee Ha$	13, UI, ADD [This contradicts line 12.]	

Rather than seek a sentence for an MTP-inference, one can let a disjunction and *base a separation of cases* on it for what one is after, which in this derivation 'Fx'. To see what one is after, consult the *SHOW*-line to be cancelled. Suppose one is after χ , and a disjunction ' $(\phi \vee \psi)$ ' is available. Then for a separation of cases argument for χ , seek ' $(\phi \rightarrow \chi)$ ', and then ' $(\psi \rightarrow \chi)$ ', to infer χ by the derived rule of separation of cases (SC),

$$\phi \vee \psi, \phi \rightarrow \chi, \psi \rightarrow \chi \therefore \chi$$

To illustrate, I use this approach after line (7).and replace lines after (7) with the following ones. (It turns out that

line (7) is not used below. In a more difficult problem it could be. Since (7) is not used, (6) is not needed.)

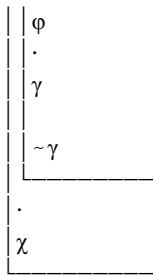
8'.	<i>SHOW</i> $\wedge x Jx \rightarrow Jx$		(10', CD)
9'.	$\wedge x Jx$		(CD)
10'.	Jx		9', UI
11'.	<i>SHOW</i> $\sim \wedge x (Gx \vee Hx) \rightarrow Jx$		(14', 17', ID)
12'.	$\sim \wedge x (Gx \vee Hx)$		(CD)
13'.	$\forall x \sim (Gx \vee Hx)$		12', QN
14'.	$\sim (Ga \vee Ha)$		13', EI
15'.	$Fx \rightarrow \wedge x Gx$		2, UI
16'.	$\wedge x Gx$		5, 15', MP
17'.	$Ga \vee Ha$		16', UI, ADD
18'.	Jx		3, 8', 11', SC

Both ways of using a disjunction work for argument (5). Indeed, each way works whenever the other does. That is, there is, for formulas χ, ϕ, ψ , and γ , a completed derivation of the form,

<i>SHOW</i> χ	$\sim \chi$
	$\phi \vee \psi$
	<i>SHOW</i> $\phi \rightarrow \chi$
	ϕ
	.
	χ
	<i>SHOW</i> $\psi \rightarrow \chi$
	ψ
	.
	χ
	χ

if and only if there is a complete derivation of the form,

<i>SHOW</i> χ	$\sim \chi$
	.
	$\phi \vee \psi$
	<i>SHOW</i> $\sim \phi$



You might, as a theoretical exercise, prove that this is so.

(8) *Using conditionals*

$$\Lambda x(Fx \rightarrow \Lambda xGx). \Lambda x(Gx \vee Hx) \rightarrow \Lambda xJx \therefore \Lambda x(Fx \rightarrow Jx)$$

This argument is like the previous one. Its second premise is logically equivalent to that of the previous argument. Its first premise and conclusion are the same as those of the previous argument.

- | | | |
|----|-------------------------------------------------|--------------|
| 1. | <i>SHOW</i> $\Lambda x(Fx \rightarrow Jx)$ | (4, UD: x) |
| 2. | $\Lambda x(Fx \rightarrow \Lambda xGx)$ | premise |
| 3. | $\Lambda x(Gx \vee Hx) \rightarrow \Lambda xJx$ | premise |
| 4. | <i>SHOW</i> $Fx \rightarrow Jx$ | (6, CD) |
| 5. | Fx | (CD) |
| 6. | <i>SHOW</i> Jx | (11, 12, ID) |
| 7. | ~Jx | (ID) |

The problem is again how best to use line (3). A suggestion from Chapter I is to derive its antecedent ' $\Lambda x(Gx \vee Hx)$ ' for MP. Since 'x' is free on an antecedent line, we cannot use universal derivation for that, but we can use indirect derivation on which there are no restrictions. It works for universal generalizations, given the derived rules of quantifier negation, whenever universal derivation works, and sometimes when that procedure is blocked as it is here.

- | | | |
|-----|-------------------------------------|-----------------------|
| 8. | <i>SHOW</i> $\Lambda x(Gx \vee Hx)$ | (ID) |
| 9. | ~ $\Lambda x(Gx \vee Hx)$ | (ID) |
| 10. | $\forall x \sim(Gx \vee Hx)$ | 9', QN |
| 11. | ~(Ga \vee Ha) | 10', EI |
| 12. | Ga \vee Ha | 2, UI, 5, MP, UI, ADD |
| 13. | Jx | 3, 8, MP, UI |

On combining lines. It is not necessary to isolate EI-inferences and assigned lines of their own. A sequence of premise and inference lines the *last* one of which is an EI-inference in which the variable of instantiation has no occurrences other than those introduced by this inference is allowed. This is by clause (9) of the definition of an abbreviated derivation, *Logic*, pp. 151-2. The restriction is explained on pages 155 and 6.²⁰

Clause (9), p. 151, includes other restrictions on last lines in a sequence that can be combined into one line. Last lines cannot be by either clause (2), which is for premise-lines, or clause (7), which is for theorem-lines. Why not? The text does not say, but one can see that even if earlier lines in a queue are not properly licensed, this cannot lead to an unwanted validation when the formula entered at last since a premise or a theorem could have been entered straight-away.

LOGIC 2000 places no restrictions on queues or on combining lines. The program keeps track of the variables of instantiation introduced by EI-inferences *that occur on lines of the derivation being abbreviated*. If one chooses to 'bury' an EI inference in a queue on an examination, the annotation for the line must indicate the variable of instantiation used so that its novelty-to-the-derivation being abbreviated can be checked.

Also LOGIC 2000 allows assumptions, as well as formulas on which cancellations of *SHOW*-lines are based to be queued. The textbook *Logic*, in contrast, confines queues to premises, inferences, and instances of theorems. It requires *assumptions* to be *displayed* on lines. And it requires cancellations to be based on formulas that are displayed on lines subsequent to the *SHOW*-line being cancelled (pp. 24-5, 143-5). Regarding the appearances of assumptions and bases for derivations (i.e., box-and-

²⁰This restriction on EI in abbreviated derivations makes incorrect the following, *otherwise correct*, abbreviated derivation for the argument

- | | | |
|----|------------------------------------------------------------------------------------|-------------------------------------|
| | $\Lambda x \sim \Lambda y (Fx \wedge Fy) \therefore \forall x (Fx \wedge \sim Fx)$ | |
| 1. | <i>SHOW</i> $\forall x (Fx \wedge \sim Fx)$ | (3, DD) |
| 2. | Fa \wedge ~Fa | premise, UI[to 'a'], QN, EI[to 'a'] |
| 3. | $\forall x (Fx \wedge \sim Fx)$ | 2, EG |

It is necessary that this derivation should somehow be blocked, since the argument it would otherwise validate is not valid. Looking ahead to Section 7.2.2.2, 'the truth-functional expansion' of this argument for one-member domains,

$$\sim(Fa \wedge Fa) \therefore (Fa \wedge \sim Fa)$$

is not tautologically valid, so that the argument is invalidated by 'the method of truth-functional expansion'. The argument is also invalidated by 'the finite arithmetic model' (Section 7.2.3),

$$\begin{array}{l} \underline{U: \{0\}} \\ F: \{ \} \end{array}$$

cancels) abbreviated derivations are, in *Logic*, exactly like unabbreviated derivations. I recommend that the restrictions on queues imposed in *Logic* be self-imposed when assembling derivations in LOGIC 2000

.....

Another way to derive the conclusion of argument (8) would be to use

$$T46 \quad (P \rightarrow Q) \leftrightarrow (\sim P \vee Q)$$

to convert the second premise into the disjunction

$$4'. \quad \sim \wedge x(Gx \vee Hx) \vee \wedge x Jx \quad T46, BC, 3, MP$$

and then proceed, as in the second way for argument (7), by separating cases for '(Fx → Jx)' under this disjunction. You might, for extra practice, construct a derivation for the argument along the

(9) *Using biconditionals*

$$\forall xFx \leftrightarrow \forall xGx \quad \therefore \quad \forall x\forall y(Fx \leftrightarrow Gy)$$

Given, as here, a biconditional,

$$\phi \leftrightarrow \psi$$

if there is a problem what to do with it, there are two possible solutions. One solution is to derive ϕ , and then use BC and MP for ψ . The other is to use a theorem to convert that biconditional into a disjunction, and then proceed under that disjunction by separation of cases for whatever one seeks. The theorem for this conversion is,

$$T83 \quad (P \leftrightarrow Q) \leftrightarrow (P \wedge Q) \vee (\sim P \wedge \sim Q)$$

The BC-MP way – first neat, and then with comments.

- | | | |
|-----|-------------------------------------------------------------|---------------------|
| 1. | SHOW $\forall x\forall y(Fx \leftrightarrow Gy)$ | (16, 17, ID) |
| 2. | $\sim \forall x\forall y(Fx \leftrightarrow Gy)$ | (ID) |
| 3. | $\forall xFx \leftrightarrow \forall xGx$ | premise |
| 4. | SHOW $\forall xFx$ | (5, 13, ID) |
| 5. | $\sim \forall xFx$ | (ID) |
| 6. | $\wedge x \sim Fx$ | 5, QN |
| 7. | $\wedge x \sim \forall y(Fx \leftrightarrow Gy)$ | 2, QN |
| 8. | $\sim Fx$ | 6, UI |
| 9. | $\sim(Fx \leftrightarrow Gy)$ | 7, UI, QN, UI |
| 10. | $Fx \leftrightarrow \sim Gy$ | T90, BC, 9, MP |
| 11. | Gy | 10, BC, 8, MT, DN |
| 12. | $\forall xGx$ | 11, EG |
| 13. | $\forall xFx$ | 3, BC, 12, MP |
| 14. | $\forall xGx$ | 3, BC, 4, MP |

- | | | |
|-----|--------------------------------------------------|-----------------|
| 15. | Fa | 4, EI |
| 16. | Gb | 14, EI |
| 17. | $\wedge x \sim \forall y(Fx \leftrightarrow Gy)$ | 2, QN |
| 18. | $\sim \forall y(Fa \leftrightarrow Gy)$ | 17, UI |
| 19. | $\sim(Fa \leftrightarrow Gb)$ | 18, QN, UI |
| 20. | $Fa \leftrightarrow \sim Gb$ | T90, BC, 19, MP |
| 21. | $\sim Gb$ | 20, BC, 15, MP |

- | | | |
|----|-------------------------------------------------------------|--------------|
| 1. | SHOW $\forall x\forall y(Fx \leftrightarrow Gy)$ | (16, 21, ID) |
| 2. | $\sim \forall x\forall y(Fx \leftrightarrow Gy)$ | (ID) |

Though sometimes an existential generalization can be derived directly by getting its 'interior' and then using EG, this approach – for which we could have a *SHOW*-line for $(Fx \leftrightarrow Gy)$ – does not work here. In this problem it is necessary to use the assumption for indirect on line (2).

- | | | |
|----|-------------------------------------------|---------|
| 3. | $\forall xFx \leftrightarrow \forall xGx$ | premise |
|----|-------------------------------------------|---------|

The problem is what to do with line (3). EI does *not* apply to ' $(\forall xFx \leftrightarrow \forall xGx)$ ', for this sentence is a biconditional, not an existential generalization. EI, like every other inference rule, can be applied only to whole lines. I will go after ' $\forall xFx$ ', and, when I get it, use it on line (3) to get ' $\forall xGx$ ', and then resume work aimed at completing a derivation for line (1).

- | | | |
|----|--------------------------------------------------|---------------|
| 4. | SHOW $\forall xFx$ | (5, 13, ID) |
| 5. | $\sim \forall xFx$ | (ID) |
| 6. | $\wedge x \sim Fx$ | 5, QN |
| 7. | $\wedge x \sim \forall y(Fx \leftrightarrow Gy)$ | 2, QN |
| 8. | $\sim Fx$ | 6, UI |
| 9. | $\sim(Fx \leftrightarrow Gy)$ | 7, UI, QN, UI |

The second UI could have been to 'x' again, thus keeping the free variables down to one. But doing this is not important here.

- | | | |
|-----|------------------------------|-------------------|
| 10. | $Fx \leftrightarrow \sim Gy$ | T90, BC, 9, MP |
| 11. | Gy | 10, BC, 8, MT, DN |
| 12. | $\forall xGx$ | 11, EG |

We are engaged in getting ' $\forall xFx$ ' in order to use it on line (3). It is thus something of a surprise that we should, as we are about to do, use line (3) to get ' $\forall xFx$ '!

- | | | |
|-----|---------------|---------------|
| 13. | $\forall xFx$ | 3, BC, 12, MP |
|-----|---------------|---------------|

Now that we have ' $\forall xFx$ ', we apply it to line (3).

- | | | |
|-----|---------------|--------------|
| 14. | $\forall xGx$ | 3, BC, 4, MP |
|-----|---------------|--------------|

Though $\forall xFx$ was derived to be applied to line (3), it turns out that this sentence has other uses as well served by instantiating it.

15.	Fa	4, EI
16.	Gb	14, EI
17.	$\Lambda x \sim \forall y(Fx \leftrightarrow Gy)$	2, QN
18.	$\sim \forall y(Fa \leftrightarrow Gy)$	17, UI [to 'a' with an eyes to line (15).]
19.	$\sim(Fa \leftrightarrow Gb)$	18, QN, UI
20.	$Fa \leftrightarrow \sim Gb$	
21.	$\sim Gb$	

The Separation-of-Cases way.

1.	SHOW $\forall x \forall y(Fx \leftrightarrow Gy)$	(22, DD)
2.	$\sim \forall x \forall y(Fx \leftrightarrow Gy)$	(ID)
3.	$\forall x Fx \leftrightarrow \forall x Gx$	premise
4.	$(\forall x Fx \wedge \forall x Gx) \vee (\sim \forall x Fx \wedge \sim \forall x Gx)$	T83, BC, 3, MP

For a separation cases argument for $\forall x \forall y(Fx \leftrightarrow Gy)$, which is what we are after on line (1), two conditionals with this sentence as consequent, and one or the other of the disjuncts on line (4) as antecedents, are required.

5.	SHOW $\forall x Fx \wedge \forall x Gx \rightarrow \forall x \forall y(Fx \leftrightarrow Gy)$	(8, 13, ID)
6.	$\forall x Fx \wedge \forall x Gx$	(CD)
7.	Fa	6, S, EI
8.	Gb	6, S, EI
9.	$\Lambda x \sim \forall y(Fx \leftrightarrow Gy)$	2, QN
10.	$\Lambda y \sim(Fa \leftrightarrow Gy)$	9, UI
11.	$\sim(Fa \leftrightarrow Gb)$	10, QN, UI
12.	$Fa \leftrightarrow \sim Gb$	T90, BC, 11, MP
13.	$\sim Gb$	12, BC, 7, MP

Now comes the second conditional for separation of cases.

14.	SHOW $\sim \forall x Fx \wedge \sim \forall x Gx \rightarrow \forall x \forall y(Fx \leftrightarrow Gy)$	(18, 21, ID)
15.	$\sim \forall x Fx \wedge \sim \forall x Gx$	(CD)
16.	$\sim(Fx \leftrightarrow Gy)$	2, QN, UI, QN, UI
17.	$Fx \leftrightarrow \sim Gy$	T90, BC, 16, MP
18.	$\sim Fx$	5, S, QN, UI
19.	$\Lambda x \sim Gx$	15, S, QN
20.	$\sim Gy$	19, UI [to 'y' with an eye to line (17)]
21.	Fx	17, BC, 20, MP
22.	$\forall x \forall y(Fx \leftrightarrow Gy)$	4, 5, 14, SC

(10) *On coping with 'buried' existential quantifiers*

$$\Lambda x \forall y(Fx \rightarrow Gy) \therefore \forall y \Lambda x(Fx \rightarrow Gy)$$

1.	SHOW $\forall y \Lambda x(Fx \rightarrow Gy)$	(11, 14, ID)
2.	$\sim \forall y \Lambda x(Fx \rightarrow Gy)$	(ID)
3.	$\Lambda x \forall y(Fx \rightarrow Gy)$	premise

There is an existential quantifier 'implicitly buried' in line (2): see lines (6) - (8) below, and another plainly buried in line (3). We cannot, in this problem, EI before we UI. *At least not as first.* (This remark is key. Its import will emerge.) It does not matter with which line we begin. I start with the simpler one, and instantiate first the universal quantifier,

4.	$\forall y(Fx \rightarrow Gy)$	3, UI
----	--------------------------------	-------

and now the existential quantifier (this could have been done on one line).

5.	$Fx \rightarrow Ga$	4, EI
----	---------------------	-------

Now for line (2). It is a negation. Only generalizations can be instantiated. First, QN is used to move out the first quantifier, which changes in the process.

6.	$\Lambda y \sim \Lambda x(Fx \rightarrow Gy)$	2, QN
----	-----------------------------------------------	-------

We now UI. To what? Our choice is between variables that are already free in the derivation. These are 'x' and 'a'. We UI to 'a'. Why not to 'x'? Because that would not be legal. The introduced occurrence of 'x' would be bound.

7.	$\sim \Lambda x(Fx \rightarrow Ga)$	6, UI
8.	$\forall x \sim(Fx \rightarrow Ga)$	7, QN
9.	$\sim(Fb \rightarrow Ga)$	8, QN, EI

We must EI to a new variable, and so cannot EI to 'x' for a contradiction with line (5).

What now? Here is an idea. *Go back to (3) and this time UI to 'b'.* This time we UI *after* an EI, and to the variable introduced by that EI.

10.	$Fb \rightarrow Gc$	3, UI, EI
11.	Gc	9, C-neg, 10, MP

Now what? *Go back to (6)*, and UI this time to 'c', the variable introduced by the most recent EI. That will lead to an inference to ' $\sim Gc$ ' for a contradiction.

12.	$\sim \Lambda x(Fx \rightarrow Gc)$	6, UI
13.	$\sim(Fd \rightarrow Gc)$	12, QN, EI

14. $\sim Gc$

13, C neg

A Possible Symbolize-to-Derive Exercise

Let English sentences Σ and Σ' be *logically equivalent in MQC* if and only if they have symbolizations ϕ and ϕ' such that

$$\phi \leftrightarrow \phi'$$

is a theorem of MQC.

Now for the exercise. Confirm that the sentences

No valid argument has a true conclusion unless it has at least one true premise.

and

Every valid argument that has a true conclusion has at least one true premise.

are logically equivalent in MQC by symbolizing them and establishing the equivalence of their symbolizations. Indicate the scheme of abbreviation you use. (A solution to this exercise is linked to the web page for *Words and Symbols*.)

7. INVALIDATION

7.1 *Foundations*

7.1.1 We confine our attention at first to symbolic arguments premises and conclusions of which are all sentences. Methods are then extended to other symbolic arguments whose premises and conclusions include non-sentence formulas.

A symbolic argument is valid in The Monadic Quantifier Calculus – *valid-in-MQC* – if and only if there exists a completed derivation for its conclusion from its premises. Methods for establishing that such arguments are not valid-in-MQC are founded on the following principle.

A symbolic argument whose premises and conclusions are sentences is valid-in-MQC if and only if, *subject to certain conditions*, it is not possible at once to *make* its premises express truths, and its conclusion express a falsehood.

Now comes a more exact statement of this foundational principle.

A symbolic argument whose premises and conclusions are sentences is valid-in-MQC if and only if there does not exist for it a **proper interpretation** such that **translations under this interpretation** of its premises are English

sentences that express truths, and translation under this interpretation of its conclusion is an English sentence that expresses a falsehood.

‘Making’ is here spelled out in terms of translation under an interpretations, and ‘certain conditions’ are here conditions satisfied by proper interpretations.

7.1.2 *Interpretations*

7.1.2.1 An *interpretation for a symbolic argument* includes not only a scheme of abbreviation for exactly its upper-case letters, but also a *universe of discourse* or set of things to which the argument's quantifiers are ‘confined’.²¹ For example, here is an interpretation in which the universe of discourse is specified by a formula: the universe is the set of things such that it is true that they satisfy the condition expressed by this formula. The universe is here the set of positive integers, i.e., the set {1, 2, 3,}.

U: {1, 2, 3,}
F: *a* is odd
G: *a* is even
A: zero

for the argument

$$\wedge x(Fx \vee Gx). \sim (FA \wedge GA) \therefore \sim FA$$

7.1.2.2 *Translation under an interpretation* is like translation under the scheme of abbreviation of this interpretation *except* for literal translations of quantifier phrases. Translations of these under an interpretation ‘restrict’ or ‘confine’ generalizations they prefix to the universe of the translation.

For example, the literal translation of ‘ $\wedge x$ ’ under the interpretation above is *not* simply,

for each *x*,

but rather,

for each x such that x is a positive integer .

Literal translation of the existential quantifier phrase ‘ $\vee y$ ’ under this interpretation would be *not* simply,

there is an *x* such that,

but rather,

²¹An *interpretation for a particular argument* differs from an ‘interpretation for MQC’, in that the latter would assign ‘extensions’ to *all* upper-case letters in the language of MQC, of which there are infinitely many. Unless an argument has infinitely many premises, only a finite number of upper-case letters occur in it.

there is an x such that x is a positive integer and .

Now comes a *free* translation of our argument under this interpretation.

Every positive integer is either odd or even. Zero is not both odd and even. \therefore Zero is not even.

The first sentence of this translation expresses a truth, as does the second. But the third sentence, which translates our symbolic argument's conclusion, expresses a falsehood: a number is even if and only if it equals the product of two and a number $- 0 = 2 \cdot 0$.

So we have an interpretation for our symbolic argument that 'makes' its premises express truths and its conclusion a falsehood. But, as explained in the next section, it is not a **proper** interpretation. As a consequence we cannot conclude on the basis of this translation, and the foundational principle of Section 7.1.1, that our symbolic argument is invalid.

7.1.3 *Proper interpretations.*²² Not every *interpretation*, that is, not every pair consisting of a universe of discourse and a scheme for letters in an argument, is a *proper* interpretation for it. There are several *restrictions* on proper interpretations. Here is the exact idea of such an interpretation with these restrictions stressed. Superscript-letters are to notes of theoretical, but not practical, importance to this exact idea.

A proper interpretation for a symbolic argument of The Monadic Quantifier Calculus consists of a **non-empty universe of discourse U**, and a scheme of abbreviation S for letters in the argument such that, (i) each sentence letter abbreviates a **sentence that expresses a proposition true or false**,^a (ii) each predicate letter abbreviates a **formula that expresses a condition such that, for each thing x in U, it is either true of or false of x that it satisfies this condition**,^b and (iii) each name letter abbreviates a **name of a thing in U**.^c

The interpretation of the previous section is not proper, because 'A' abbreviates '0', and zero is not a positive integer, that is, it is not in the universe {1, 2,...}. So, as

²²The idea of a proper interpretation for an MQC-argument comes from that of a *model* for such arguments explained in *Logic* on pages 177-8. The main differences are that the 'universe' of a proper interpretation can be any non-empty set, whereas that of a model must be a non-empty *finite* set of *natural numbers*, and predicate letters can in a proper interpretation abbreviate any formulas, whereas in a model they abbreviate only 'set-membership' formulas.

said, that interpretation does not invalidate our argument. To invalidate it by application of our foundational principle, we need a proper interpretation that makes its premises true and its conclusion false. One such proper interpretation can be reached by 'tweaking' the interpretation of the previous section. The universe of the coming interpretation is again specified by a formula: the universe is the set of things

U: {0, 1, 2, ...}
F: a is odd
G: a is even
A: zero

The set of *natural numbers* includes zero: it is the set {0, 1, 2, 3, ...}.

■ Notes a, b, and c

^a (i) *each sentence letter abbreviates a sentence that expresses a proposition true or false.* Why is this required for a proper interpretation? Because the sentential connectives are *truth-functional* connectives: they are defined only for sentences that are either true or false. *Propositions* enter here because they are the primary bearers of truth and falsity. A *sentence* is derivatively true (false) if and only if it expresses a proposition that is true (false).

It is a consequence of condition (i) that no sentence letter can abbreviate the sentence 'This sentence expresses a false proposition.' in a proper interpretation, if as I think this sentence does not express a proposition. Discussions in Section 10 of Chapter V below, and in Supplementary Sections 3 above, and 19 and 21 below, are relevant to this 'liar sentence'.

Here for possible interest is another illustration of the bite of the condition (i). Relevant to this illustration is that a sentence S expresses a proposition if and only if each *token* of S expresses this proposition. The illustration is that no abbreviation in a proper interpretation can be of the sentence.

The boxed sentence-token in Section 7.1.3 expresses a proposition that is not true.

The reason is that not every token (indeed, no token) of this sentence expresses a proposition, from which it follows that the sentence of this token does not express a proposition, which in turn entails that this sentence is neither true nor false. Now comes a proof that the boxed sentence token in Section 7.1.3 of this sentence does not express a proposition.

Suppose, for an indirect proof to the contrary, that the boxed sentence-token in Section 7.1.3 expresses a proposition. Then it expresses exactly one proposition and this is *the proposition that the boxed sentence-*

token in Section 7.1.3 expresses a proposition that is not true. Let ¶ be this proposition.

Proposition ¶ is not true. Suppose, (*), for an indirect proof to the contrary, that ¶ is true. Then ¶ is not true. For ¶ says that the token expresses a proposition, and ¶ is the proposition that this token expresses. So ¶ says of itself that it is not true, which is presently, (*), supposed to be true. (The italicized contradiction completes this subsidiary indirect proof.)

Proposition ¶ is true. Suppose, (**), for an indirect proof to the contrary, that it is not true. Then it is true. For again, ¶ says that the token expresses a proposition, and ¶ is the proposition that this token expresses. So ¶ says of itself that it is not true, which agrees with what is presently, (**), supposed. (The italicized contradiction completes this subsidiary indirect proof.)

The boldly emphasized contradiction completes the major indirect proof for the conclusion that the boxed sentence-token in Section 7.1.3 does not express a proposition.²³

^b (ii) *each predicate letter abbreviates a formula that expresses a condition such that, for each thing x in U, it is either true of or false of x that it satisfies this condition.* The object of this restriction is to insure that translation under a proper interpretation operates to assign to each predicate letter an 'extension', where this is a subset of the interpretation's universe, the members of which subset includes exactly the members of this universe 'of which this interpreted predicate letter is true'.

To illustrate the bite of this restriction, in a proper interpretation no predicate letter can abbreviate the formula 'a is so-called for its size'. This is because this formula does not express a condition that is true or false simply of things. That formula expresses a condition that is true or false only of things under names. Consider *The Big Apple*, that is, consider *New York*. Is it true of it – is it true of this city – that it is so-called for its size! "So-called what? So-called 'The Big Apple', so-called 'New York', or so-called 'this city'?" Though it is true that The Big Apple is so-called for its size, it is not true of The Big Apple, simply of that city, that it is so-called for its size. If it were, since the

²³ The proposition that the boxed sentence-token in Section 7.1.3 expresses a proposition that is not true, though not expressed by the sentence-token in the box in Section 7.1.3 of the sentence 'The boxed sentence-token in Section 7.1.3 expresses a proposition that is not true.' is expressed by all other tokens of this sentence.

This proposition (the proposition named in italics at the beginning of this note) is, incidentally, false. This follows from what has been proved. Since (as has been proved) the sentence-token in the box in Section 7.1.3 does not express a proposition, it does not express a proposition that is not true.

city that is The Big Apple is New York, it would be true that New York is so-called for its size. This for the reason that whatever is true of a thing is true of anything identical with that thing! But it is not true of New York that it is so-called for its size: that is, New York is not called 'New York, for its size. Its size has nothing to do with its being called 'New York'. Nor is it called 'that city' for its size. It is not true or false of New York, or of anything, that it is so-called for its size. It is only true of some things under certain names that they are so-called for their sizes. So in a proper interpretation, no predicate letter can abbreviate 'a is so-called for its size'.

For similar reasons, no predicate letter in a proper interpretation can abbreviate 'New York is called 'a' for its size' and 'John said, "a was home."' And no predicate letter can abbreviate 'The Big Apple was called 'a' in the 18th century': consider that The Big Apple was called 'New York' then, but not 'The Big Apple'.

^c (iii) *each name letter abbreviates a name of a thing in U.* For example, if U is the set {0, 1, 2,...}, no letter can abbreviate '-1'.

7.2 Invalidation methods

7.2.1 Spelled out interpretations

7.2.1.1 To invalidate a symbolic argument it is sufficient to spell out an invalidating proper interpretation. Here is another illustration, designed to make clear that we want other more mechanical invalidation methods. The symbolic argument,

$$\wedge x(\wedge yGy \rightarrow Fx) \quad \therefore \quad \forall y(\forall xGx \rightarrow Fy),$$

is invalidated by the proper interpretation,

$$\begin{aligned} U: \{0, 1, 2, \dots\} \\ F: a \text{ is less than } 0 \\ G: a \text{ is greater than } 0 \end{aligned}$$

That symbolic argument has under this interpretation the following translation.

For each number in U, every number in U is greater than 0 only if (in the truth-table, '→'-sense of 'only if') every number in U is less than 0. [This is true (!), since the inner italicized antecedent expresses something is false. 0 is not greater than 0.]

∴ Some number y is such that, if some number is greater than 0, then it (y) is less than 0. [This is false, since some number is greater than 0, but no number in U is less than 0.]

7.2.1.2 Finding invalidating proper interpretations is not always easy even in simple cases such as this one. Even verifying a proposed invalidating proper interpretation is not always easy. Fortunately, there are better ways to invalidate arguments in MQC. There is a completely mechanical method – *the method of truth-functional expansion* – of general application. And there is a non-mechanical constructive method – *the method of finite models* – that is often a good choice.

7.2.2 *The method of truth-functional expansion* (The coming method is equivalent to but somewhat simpler than the method in the textbook *Logic*.)

7.2.2.1 Consider a *finite* proper interpretation for our argument

$$\Lambda x(\Lambda yGy \rightarrow Fx) \quad \therefore \quad \forall y(\forall xGx \rightarrow Fy)$$

Consider, to be more definite, an interpretation in which U contains *only one thing*, *a*, never mind what this thing is, or what 'G' and 'F' abbreviate in this interpretation.

What does ' $\Lambda x(\Lambda yGy \rightarrow Fx)$ ' say of the universe $U = \{a\}$ of this interpretation? 'Expanding' its quantifiers for this U one at a time, it says

$$(\Lambda yGy \rightarrow Fa)$$

and so it says

$$(Ga \rightarrow Fa).$$

That is, if $U = \{a\}$,

$$' \Lambda x(\Lambda yGy \rightarrow Fx) ' \text{ is true of } U$$

if and only if

$$(Ga \rightarrow Fa)$$

is true. I am taking *a* to be some thing, it does not matter what it is, and I am pretending that that the variable 'a' is a symbolic name of this thing, so that 'Ga' and 'Fa' are sentences about this thing.

'Expanding' in turn the two quantifiers in the conclusion, reveals that, if $U = \{a\}$,

$$' \forall y(\forall xGx \rightarrow Fy) ' \text{ is true of } U$$

if and only if

$$'(Ga \rightarrow Fa)' \text{ is true.}$$

From this we see that our argument *cannot* be invalidated by a proper interpretation of *size 1*. No matter what *a* is, and no matter what 'F' and 'G' mean, there is *of course* no way in which what the premise comes to, '(Ga → Fa)' can be made true, while what the conclusion comes to, '(Ga → Fa)' again (!), is made false.

But what about proper interpretations of size 2? Can our argument be invalidated by one of them? We can find out by 'expanding' for an unspecified size-2 universe $U =$

{a, b}, and determining by truth-table methods whether or not the expanded argument is tautologically valid. If it is *not*, then we know that it is *possible* to specify *a* and *b*, and to assign meanings to 'F' and 'G', in ways that makes the premise true of $U = \{a, b\}$. We expand one at a time the quantifiers in the premise,

$$\Lambda x(\Lambda yGy \rightarrow Fx).$$

Since each is a *universal* quantifier, each expands into a *conjunction*, one conjunct for *a* and another conjunct for *b*. [To say 'for each x in {a, b}....' is to say 'for a...., and for b....'] Expanding ' Λx ' leads to,

$$(\Lambda yGy \rightarrow Fa) \wedge (\Lambda yGy \rightarrow Fb).$$

Expanding each ' Λy ' leads to,

$$(Ga \wedge Gb \rightarrow Fa) \wedge (Ga \wedge Gb \rightarrow Fb) .^{24}$$

So we see that the premise ' $\Lambda x(\Lambda yGy \rightarrow Fx)$ ' is true of $U = \{a, b\}$, when *a* and *b* are specified and meanings are assigned to 'F' and 'G', if and only if this just displayed quantifier-free formula is true of {a,b}.

Turning to the conclusion,

$$\forall y(\forall xGx \rightarrow Fy),$$

since its quantifiers are *existential*, each expands into a *disjunction*. [To say 'there is at least one x in {a, b} such that....' is to say 'either *a* is such that...., or *b* is such that....'] Expanding ' $\forall y$ ' leads to,

$$(\forall xGx \rightarrow Fa) \vee (\forall xGx \rightarrow Fb).$$

Expanding each ' $\forall x$ ' leads to,

$$(Ga \vee Gb \rightarrow Fa) \vee (Ga \vee Gb \rightarrow Fb) .$$

²⁴I have written, "We expand one at a time the quantifiers...." This invites the question, "In what order?" The answer is that it does not matter. For example, expanding first ' Λy '

$$\Lambda x(\Lambda yGy \rightarrow Fx),$$

for {a, b} leads to

$$\Lambda x[(Ga \wedge Gb) \rightarrow Fx],$$

and expanding ' Λx ' for {a, b} leads again to

$$(Ga \wedge Gb \rightarrow Fa) \wedge (Ga \wedge Gb \rightarrow Fb) .$$

The general proposition is that if there are several quantifiers in a formula ϕ , and ϕ' and ϕ'' come from ϕ by expanding these quantifiers for finite domains of a size *n*, then ϕ and ϕ' are logically equivalent. This proposition can be established by a strong mathematical induction on the number of occurrences of quantifiers in a formula.

expansion can be accomplished by replacing 'a' and 'b' in the expansion of the argument for {a, b}

$$(Ga \wedge Gb \rightarrow Fa) \wedge (Ga \wedge Gb \rightarrow Fb) \therefore \\ (Ga \vee Gb \rightarrow Fa) \vee (Ga \vee Gb \rightarrow Fb)$$

respectively by '0' and '1'. The result is

$$(G0 \wedge G1 \rightarrow F0) \wedge (G0 \wedge G1 \rightarrow F1) \therefore \\ (G0 \vee G1 \rightarrow F0) \vee (G0 \vee G1 \rightarrow F1)$$

Evaluating this argument in the model, confirms that its premise is true and its conclusion false in this model, so that this model does invalidate it.

$$(G0 \wedge G1 \rightarrow F0) \wedge (G0 \wedge G1 \rightarrow F1) \therefore \\ \begin{array}{cccccc} t & f & f & t & f & f \\ & f & & f & & \\ & & t & & t & \\ & & & t & & \\ & & & & t & \\ & & & & & t \\ & & & & & & t \\ & & & & & & & f \\ & & & & & & & & f \end{array} \\ (G0 \vee G1 \rightarrow F0) \vee (G0 \vee G1 \rightarrow F1) \\ \begin{array}{cccccc} t & f & f & t & f & f \\ & t & & t & & \\ & & f & & t & \\ & & & t & & \\ & & & & f & \\ & & & & & f \end{array}$$

7.2.2.3 Back to the method of truth-functional expansion, to summarize. The method of truth-functional expansion for showing that a symbolic argument of MQC is invalid is, in short,

- (1), to expand this argument for universes of some size n,
- and,
- (2), to establish by truth-assignment techniques that its expansion for such universes is not valid.

In step (2) truth-values are assigned to 'atomic formulas' – to subject-predicate formulas, and to sentence letters if any – in the expansion.

.....

N.B. As said in a footnote, I leave out a step of the truth-functional expansion method explained in *Logic*. I leave out step (4) on page 174 in which formulas such as 'Ga' and 'Fa' are uniformly replaced by distinct sentence letters. 'Ga' is treated as if it were a subscripted sentence letter such as 'P₁'. I in effect 'pretend' that in 'Fa', 'F' is a sentence letter, and 'a' is a numerical subscript.²⁶ Step (4) of *Logic* would have one replace 'Fa', 'Fb', 'Ga', and

²⁶In Chapter IV, sentences are the letters 'F' through 'Z' with numerical superscripts, and with or without numerical subscripts. 'G' is in that language an informal sentence letter, and so is 'G₀', though 'G_a' is not. As said, subscripts are confined to numerals.

'Gb' in our example by distinct sentence letters, for example, respectively by 'P', 'Q', 'R', and 'S', or alternatively, respectively, by 'P₁', 'P₂', 'Q₁', and 'Q₂'.

7.2.2.4 The method is to expand a symbolic argument for universes of some size n. If a symbolic argument of The Monadic Quantifier Calculus is invalid, this can be established by the *truth-functional* expansion method. This method always works. Furthermore, it is possible to say in advance how far one may need to expand an argument before one reaches a tautologically invalid expansion for an argument that is invalid in MQC. The upper limit depends on the number of distinct predicate letters in the argument. If, as in our example, there are 2, then the upper limit is 2² = 4. In general, if there are n then the upper limit is 2ⁿ. The reason for this remarkable property of monadic quantifier arguments is that predicate letters, when interpreted, sort things according as they are true or false of them. For example, 2 predicate letters sort things into 4 kinds depending on whether both are true of a thing, just one or the other is true of it, or neither is true of it. Possibilities for invalidations of a monadic quantifier argument depend only on the number of possible *kinds* of things under the argument's letters, and not at all on how many things can be of a given kind or kinds.

7.2.2.5 The method is not good when there are name letters. There are no *theoretical* limitations to the truth-functional expansion method, but there is an important practical limitation. It is a cumbersome and inefficient method when applied to arguments that contain *name* letters. For before making truth-assignments, *such letters need to be replaced by expansion variables* (Step (3) in *Logic*, p. 174), and when expanding for universes of size greater than 1 there several ways in which they can be replaced, and it can happen that while one way leads to tautological invalidity, none of the other ways do.²⁷ Finding that one way could involve a lot of work. So for practical purposes we need another invalidation method to use when there are name letters in an argument. There is another method which is furthermore more fun!

■ While not practical for evaluating arguments in which name letters occur, the method of truth functional expansion points to a refinement of the opinion floated in a black-boxed passage in Section 1.4 above concerning the

²⁷How many ways will there be? That depends on the size n of domains for which one is expanding, and the number k of name letters in the argument. The number of ways, and thus the number of truth functional expansions of the argument for size n domains if it contains k name letters is n^k: for example, if 3 name letters occur in an argument, it will have 8 truth functional expansions for size 2 domains.

‘adequacy’ of MQC without name letters. It is said there that if an English argument *E* has a symbolization *A* in which name letters occur it has a certain symbolization *A'* in which these letters are in a manner ‘replaced’ by predicate letters such that *A* is valid if and only if *A'*. This implies that

every truth-functional expansion of *A* for size *n* domains, where *n* is the number of predicate letters occurring in *A*, is tautologically valid if and only if *the* truth-functional expansion of *A'* for (*n* + 1) domains is tautologically valid.

This refinement affords a ‘handle’ on the opinion floated in Section 1.4 that may facilitate its assessment and proof or disproof. ■

7.2.3 The method of finite arithmetic models

7.2.3.1 The method of finite arithmetic models consists in constructing step-by-step an invalidating proper interpretation of a certain stylized and compressed form. The attractiveness of the method lies in its *constructive* character. The object is to produce an interpretation whose universe is some small set of numbers, and whose scheme assigns to predicate letters formulas that say that *a* is in some specified sub-set of the designated universe.²⁸ For example, if the universe *U* one arrives at in a construction is the set of numbers {0, 1, 2}, then the formula in an abbreviation specified for 'F' arrived at might be,

F: *a* is in the set {0, 2}.

This formula is true of precisely 0 and 2.

Rather than spell out these stylized formulas it is sufficient simply to specify for each predicate letter the sub-set of numbers of the universe of which it is to be true. In our example, the 'abbreviation' for 'F' is compressed to,

F: {0, 2}

So *predicate letters* are assigned subsets of a finite model's universe. *Name letters* are, as the model is constructed, assigned members of this universe. *Sentence letters* are simply assigned truth-values. The object that drives the construction is, of course, to make the argument's premises true, and its conclusion false of the universe of the model.

²⁸Why numbers? Because there are enough of them for any argument, no matter how many predicate letters occur in it. As long as one is dealing with relatively small arguments, however, other things, for example, cities and countries, serve as well. The only things that matters as far about specified members of a finite domain as far as the invalidation method now being described goes is that they be *distinct one from each other*.

But what about that, what about the universe of the model? How does one know what it should be? One does not know this in advance. Rather, one starts with 0 in the universe, for it cannot be empty, and then adds to it as required in the course of the construction.

7.2.3.2 A pedagogic interlude - evaluating arguments in finite models. Before proceeding to illustrations of *constructions* of finite models, let us practice going in the other direction and deciding whether or not an argument is invalidated by a finite model.

Question: Is the argument

$$\forall x(Fx \wedge \sim Gx). \forall xGx \quad \therefore \quad \wedge x(Fx \vee Gx)$$

invalidated by the following model?

$$\begin{aligned} U: & \{0, 1\} \\ F: & \{0\} \\ G: & \{1\} \end{aligned}$$

To decide we must evaluate the premises and the conclusion of the argument in this model: the model invalidates this argument if and only if it makes its premises true, and its conclusion false. To evaluate these sentences in this model, it is sufficient to *translate* them in this model, and then compare to it what they say.

Premise ‘ $\forall x(Fx \wedge \sim Gx)$ ’ says in the model that

there is an *x* in the set {0, 1} such that
(*x* is in the set {0} and
it is not the case that *x* is in the set {1})

or more succinctly (using predicate letters to name the sets assigned to them in the model) that

there is something in *U* that is in *F* and not in *G*

That is true in the model: in it 0 is in *F*, and not in *G*.

Premise ‘ $\forall xGx$ ’ says in the model that

there is something in *U* that is in *G*

This is true in the model: in it 1 is in *G*.

Conclusion ‘ $\wedge x(Fx \vee Gx)$ ’ says in the model that

for each *x* in the set {0,1}
(*x* is in the set {0} or *x* is in the set {1})

or more succinctly that

everything in *U* is either in *F* or in *G*

This is also true in the model. *Therefore the model does NOT invalidate the argument.* For that we need a model that not only makes the argument's premises true, as this model does, but that makes its conclusion false.

Another question: Is the same argument

$$\forall x(Fx \wedge \sim Gx). \forall xGx \therefore \Lambda x(Fx \vee Gx)$$

invalidated by following model?

$$\begin{array}{l} \underline{U: \{0, 1, 2\}} \\ F: \{0\} \\ G: \{1\} \end{array}$$

It can be seen that the argument's premises are still true in this model, but that now its conclusion, which to repeat, says that

everything that is in U is either in F or in G,

is false: in this model 2 is in U, but not in either F or G.

Therefore the argument IS invalidated by this model.

■ *Truth in a model.* While it is not sentences, but propositions they express, that are true and false, it is convenient to extend these terms to sentences. The 'rule' is that, for a sentence ϕ of an interpreted language (which may be a natural language such as English) that is grammatical correct for saying things that can be believed,

' ϕ ' is here and now true *if and only if* ϕ .

This 'rule' comprises countless biconditionals of this form. Here is one of these:

'It is snowing.' is here and now true *if and only if* it is snowing.

The 'idea' of this biconditional is that, by stipulation, the sentence 'It is snowing.' is here and now true if and only if, *using it here and now to say something*, would result in something *that is a fact*. The words 'is a fact' could be added to the general rule, or stipulation, without changing its import. The 'idea', in other words, is that using that sentence here and now to say something, would result in the assertion a proposition that is true: These are other words for the same 'idea,' since, in general, what one says or asserts is a proposition, and a proposition is true if and only if it is a fact.

For the case of present interest, the rule or stipulation is that, for a sentence ϕ of MQC, a model M in which letters of this sentence are provided interpretations, and a sentence ϕ' that is a translation of ϕ in M,

' ϕ ' is true if M *if and only if* ϕ' .

Again this rule comprises countless biconditionals of this form. Alternative words for

' ϕ ' is true in M

are

' ϕ ' expresses a true proposition in M.

For of the comprised biconditionals we have, for the sentence ' ΛxFx ' and the model M,

$$\begin{array}{l} \underline{U: \{0, 1, 2\}} \\ F^1: \{0, 2\} \end{array}$$

the biconditional,

' ΛxFx ' is true in M *if and only if*,

for each member x of $\{0, 1, 2\}$, x is a member of $\{0, 2\}$.

Given this biconditional-stipulation, ' ΛxFx ' is *not true* in M, since it is *not* the case that, for each member x of $\{0, 1, 2\}$, x is a member of $\{0, 2\}$. When a sentence and model are covered by the general rule of this paragraph, this sentence is *false* in the model if and only if it is *not true* in it: ' ΛxFx ' is thus false in the model M.

I may sometimes write of sentences, for example of premises, and of antecedents, as true or false in models. Usually, however, perhaps always, I will use extra words in order to ascribe truth and falsity in models not to sentences, but to propositions they express in these models.■

7.2.3.3 On constructing finite models. Consider the argument,

$$\Lambda x(Fx \rightarrow Gx). GA. \forall x(Fx \wedge Gx) \therefore FA$$

The object is a finite model in which

$$\begin{array}{l} \Lambda x(Fx \rightarrow Gx). GA. \forall x(Fx \wedge Gx) \therefore FA \\ t \qquad \qquad \qquad t \quad t \qquad \qquad \qquad f \end{array}$$

We work backward from these values to a finite model for which they obtain. This can be accomplished in four steps.

- (i) To begin: the universe which must be non-empty.
- (ii) To make the second premise true. It says that A is G. So 'A' must name something in U, that is in the sub-set assigned to 'G'.

$$\begin{array}{l} \underline{U: \{0\}} \\ F: \{ \\ G: \{ \\ A: \end{array}$$

$$\begin{array}{l} \underline{U: \{0\}} \\ F: \{ \\ G: \{0 \\ A: 0 \end{array}$$

(iii) To make the conclusion, which says that *A is in F*, false, I exclude 0 from F.

(iv) The third premise says that *something in U is in both F and G*. This cannot be 0. I make it 1.

U: {0}	U: {0, 1}
F: {-	F: {-, 1
G: {0	G: {0, 1
A: 0	A: 0

We have made the second premise true, the conclusion false, and the third premise true. What about the first premise? It says that *everything in U that is in F is also in G*. That is true in the model of step (iv). So we are done. Here is our invalidating model.

U: {0, 1}
F: {1}
G: {0, 1}
A: 0

To verify that this model does invalidate our argument, it is sufficient to translate under this model, and then evaluate in it. Here is this translation, though in practice this check is done without writing anything down.

Everything in {0,1} that is in {1} is also in {0,1}. [True.]
 0 is in {0,1}. [True]
 Something in {0,1} is in both {1} and {0,1}. [True.]
 \therefore 0 is in {1}. [False]

7.2.4 Goodbye to spelled out proper interpretations. The invalidity of a symbolic argument of The Monadic Quantifier Calculus can be established by a fully spelled out proper interpretation. But it is always possible to establish that a symbolic argument is not valid in this calculus either by constructing an invalidating finite arithmetic model, the scheme of which is not fully spelled out in words, or by the truth-functional expansion method in which the existence of a finite arithmetic model is established, but no model or interpretation is actually spelled out. These methods are, for arguments in MQC, recommended for use in every case.

7.2.5 Which method to use when – truth-functional expansion or finite models. It is best to use the finite model method for

arguments that are free of 'overlay of 'quantifiers'
 (see *Logic*, p. 224),

and best to use it for

arguments in which name letters occur.

It is best to use *the truth-functional expansion method* when there is overlay, and name letters do not occur.

Buy when there is overlay, if there are also name letters, it is best to use the finite model method, even though it is challenged by overlay.

8. INVALIDATION – ILLUSTRATIONS

Much more is presented in these illustrations than is required for the invalidations they convey. In practice, 'work' can be roughed in. Only the terms of an invalidation need to be exactly stated. These terms are: for the truth-functional expansion method,

a truth-functional expansion of the argument,
 and an invalidating truth-assignment;

and for the finite-model method,

an invalidating finite model in which
 each letter of the argument is assigned an extension.

$$(1) \forall y(Gy \rightarrow \Lambda xFx) \therefore \Lambda x \forall y(Gx \rightarrow Fy)$$

Since this argument features overlay of quantifiers, and does not contain name letters, we use the truth-functional expansion method. For one-member domains that include just some object *a*, the premise expands first to,

$$(Ga \rightarrow \Lambda xFx)$$

And then to,

$$(Ga \rightarrow Fa).$$

The conclusion expands similarly in two steps to the same formula,

$$(Ga \rightarrow Fa).$$

So it is impossible to make this argument's premise true and its conclusion false of any *one*-member domain, and we expand the argument for *two*-member domains that contain just *a* and *b*. Its premise expands in two steps: taking its existential quantifier, \forall , first we get,

$$(Ga \rightarrow \Lambda xFx) \vee (Gb \rightarrow \Lambda xFx);$$

and then, proceeding to these universal quantifiers, Λ , we get

$$[(Ga \rightarrow Fa \wedge Fb) \vee (Gb \rightarrow Fa \wedge Fb)].$$

I expanded the outer quantifier first. The result would have been the same if I had expanded the inner one first.

(You might verify this.) *The order in which quantifiers are expanded never matters to the end result.*

The conclusion can be expanded similarly in two steps: taking its inner \forall -quantifier first, we get,

$$\wedge x[(Gx \rightarrow Fa) \vee (Gx \rightarrow Fb)];$$

and then, proceeding to the \wedge -quantifier, we get

$$[(Ga \rightarrow Fa) \vee (Ga \rightarrow Fb)] \wedge [(Gb \rightarrow Fa) \vee (Gb \rightarrow Fb)].$$

So our argument has, as an expansion for two-member domains,

$$\frac{[(Ga \rightarrow Fa \wedge Fb) \vee (Gb \rightarrow Fa \wedge Fb)]}{(Ga \rightarrow Fa) \vee (Ga \rightarrow Fb) \wedge [(Gb \rightarrow Fa) \vee (Gb \rightarrow Fb)]}$$

The next step is to test this expansion, and find out if it is tautologically invalid. So we look for an assignment of truth-values to the atomic formulas 'Ga', 'Fa', and so on that makes the expansion's premise true, and its conclusion false.

$$\frac{[(Ga \rightarrow Fa \wedge Fb) \vee (Gb \rightarrow Fa \wedge Fb)] \quad \mathbf{t}}{[(Ga \rightarrow Fa) \vee (Ga \rightarrow Fb)] \wedge [(Gb \rightarrow Fa) \vee (Gb \rightarrow Fb)] \quad \mathbf{f}}$$

Nothing is 'forced'. I start with a guess, and, to make the conclusion false, make its first conjunct false.

$$\frac{[(Ga \rightarrow Fa \wedge Fb) \vee (Gb \rightarrow Fa \wedge Fb)] \quad \mathbf{t}}{[(Ga \rightarrow Fa) \vee (Ga \rightarrow Fb)] \wedge [(Gb \rightarrow Fa) \vee (Gb \rightarrow Fb)] \quad \mathbf{t \ f \ f \ f \ t \ f \ f \ f}}$$

This can be seen to make the first disjunct of the expansion of the premise false, so I make the second disjunct true in the only way that it still possible.

$$\frac{[(Ga \rightarrow Fa \wedge Fb) \vee (Gb \rightarrow Fa \wedge Fb)] \quad \mathbf{t \ f \ f \ f \ f \ t \ f \ t \ f \ f \ f}}{[(Ga \rightarrow Fa) \vee (Ga \rightarrow Fb)] \wedge [(Gb \rightarrow Fa) \vee (Gb \rightarrow Fb)] \quad \mathbf{t \ f \ f \ f \ t \ f \ f \ f}}$$

We have found that the following assignment makes the premise of the expansion true and its conclusion false.

$$\begin{array}{cccc} \mathbf{Fa} & \mathbf{Fb} & \mathbf{Ga} & \mathbf{Gb} \\ \hline \mathbf{f} & \mathbf{f} & \mathbf{t} & \mathbf{f} \end{array}$$

If this expansion had been tautologically *valid*, I would have expanded for *three*-member domains containing only *a*, *b*, and *c*, and checked again for tautological validity. A further expansion to four-member domains *could* have been necessary, but *that* expansion would have been *decisive*. As has been said, a symbolic argument in The Monadic Quantifier Calculus is valid if and only if its expansion for 2^n domains is valid where *n* is the number of distinct predicate letters that occur in this argument. Here *n* = 2, and 2^n = 4.

Since an expansion of argument (1) for two-member domains is not tautologically valid, it is possible to make the premise of argument (1) true, and its conclusion false of a two-member domain. That is, we have established that a finite arithmetic model *exists* that invalidates this argument. *It is not necessary actually to spell it out, but for interest, here one is constructed from the just displayed assignment, taking 'a' into '0' and 'b' into '1':*

$$\begin{array}{l} \mathbf{U: \{0,1\}} \\ \mathbf{F: \{ \}} \\ \mathbf{G: \{0\}} \end{array}$$

An 'answer' to the problem of invalidating argument(1) – an answer that uses the truth-functional expansion method – has been highlighted in bold italics: it is an expansion of argument (1), and a truth-assignment that invalidates this expansion. Everything else is 'work,' and commentary.

$$(2) \wedge yGy \rightarrow \wedge xFx \quad \therefore \quad \forall xGx \rightarrow \forall yFy$$

Since this argument is free of over-lay, I use the method of finite arithmetic models. The object is to *construct* such a model that makes the premise of this argument true and its conclusion false.

$$\wedge yGy \rightarrow \wedge xFx \quad \therefore \quad \forall xGx \rightarrow \forall yFy \quad \mathbf{t \quad \quad \quad f}$$

This requirement has useful consequences for the conclusion.

$$\wedge yGy \rightarrow \wedge xFx \quad \therefore \quad \forall xGx \rightarrow \forall yFy \quad \mathbf{t \quad \quad \quad t \quad f \quad f}$$

Now comes construction of a model that yields these truth-values.

Step One	Step Two	Step Three	Step Four
<u>U: {0}</u>	<u>U: {0}</u>	U: {0}	<u>U: {0,1}</u>
F: { }	F: { }	F: { }	F: { }
G: { }	G: {0}	G: {0}	G: {0}

To start: U must be non-empty.
 Towards making conclusion false.
 To make conclusion false.
 To make premise true.*

*As a result of Step Three we have, for the argument,

$$\begin{array}{ccc} \Lambda yGy \rightarrow \Lambda xFx & \therefore & \forall xGx \rightarrow \forall yFy \\ t & f & t & f & f \end{array}$$

To make the premise true, I make 'ΛyGy' false by adding 1 to the universe, and keeping it out of the extension of 'G'. It is false that everything in {0,1} is in {0}.

The 'answer' to (2) by the method of finite arithmetic models is in Step Four: it is the invalidating finite model. To *verify* this answer, one may 'read' the premises in terms of the model.

If every number in {0,1} is in {}, then every number in {0,1} is in {}. [Antecedent false; so conditional true.]

If some number in {0,1} is in {0}, then some number in {0,1} is in {}. [True antecedent and false consequent; so conditional false.]

The model in Step Four abbreviates the following spelled out interpretation.

U: {0,1}
 F: *a* is a member of the set { }
 [i.e., *a* is a member of the *empty* set, the set that has no members!]
 G: *a* is a member of the set {0}
 [i.e., *a* is a member of the set whose sole member is 0]

(3) $\Lambda xFx \rightarrow \Lambda yGy \therefore \forall xFx \rightarrow \Lambda yGy$

The model method is best here. It is more efficient than the truth-functional expansion method. The object is a model that yields these truth-values:

$$\begin{array}{ccc} \Lambda xFx \rightarrow \Lambda yGy & \therefore & \forall xFx \rightarrow \Lambda yGy \\ t & & f \end{array}$$

so

$$\begin{array}{ccc} \Lambda xFx \rightarrow \Lambda yGy & \therefore & \forall xFx \rightarrow \Lambda yGy \\ t & & t & f & f \end{array}$$

thus

$$\begin{array}{ccc} \Lambda xFx \rightarrow \Lambda yGy & \therefore & \forall xFx \rightarrow \Lambda yGy \\ t & f & t & f & f \end{array}$$

and therefore

$$\begin{array}{ccc} \Lambda xFx \rightarrow \Lambda yGy & \therefore & \forall xFx \rightarrow \Lambda yGy \\ f & t & f & t & f & f \end{array}$$

The construction proceeds with the now highlighted truth-values as its complex object.

$$\begin{array}{ccc} \Lambda xFx \rightarrow \Lambda yGy & \therefore & \forall xFx \rightarrow \Lambda yGy \\ f & t & f & t & f & f \end{array}$$

Step One	Step Two	Step Three	Step Four
----------	----------	------------	-----------

<u>U: {0}</u>	<u>U: {0}</u>	<u>U: {0}</u>	<u>U: {0,1}</u>
F: { }	F: {0}	F: {0}	F: {0}
G: { }	G: { }	G: {-}	G: {-}

To start: U must be non-empty.
 Toward making conclusion false, by making 'ΛyGy' false.
 To make conclusion false, by making 'ΛyGy' false.
 To make premise true, by making 'ΛxFx' false.

(4) $\forall x\Lambda y(Fx \rightarrow Gy) \therefore \Lambda x\Lambda y(Fx \rightarrow Gy)$

The truth-functional expansion method is best here because of overlay and absence of name letters. Expansions for one-member domains of this argument are tautologically valid, so I expand it for two-member domains.

$$\begin{array}{c} \Lambda y(Fa \rightarrow Gy) \vee \Lambda y(Fb \rightarrow Gy) \\ (Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb) \vee [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)] \\ \hline \Lambda y(Fa \rightarrow Gy) \wedge \Lambda y(Fb \rightarrow Gy) \\ (Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb) \wedge [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)] \end{array}$$

Now for an invalidating truth-assignment.

$$\begin{array}{c} [(Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb)] \vee [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)] \\ t \\ \hline [(Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb)] \wedge [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)] \\ f \end{array}$$

$$\frac{[(Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb)] \vee [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)]}{t}$$

$$\frac{[(Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb)] \wedge [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)]}{t \quad f \quad f \quad f \quad f}$$

$$\frac{[(Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb)] \vee [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)]}{t \quad f \quad f \quad f \quad t \quad f}$$

$$\frac{[(Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb)] \wedge [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)]}{t \quad f \quad f \quad f \quad f}$$

$$\frac{[(Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb)] \vee [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)]}{t \quad f \quad f \quad f \quad t \quad f \quad t \quad f \quad t \quad f \quad t}$$

$$\frac{[(Fa \rightarrow Ga) \wedge (Fa \rightarrow Gb)] \wedge [(Fb \rightarrow Ga) \wedge (Fb \rightarrow Gb)]}{t \quad f \quad f \quad f \quad f}$$

'Gb' can be either true or false. I pick T.

$$\frac{Fa \quad Fb \quad Ga \quad Gb}{T \quad F \quad F \quad T}$$

(5) $\therefore \sim[\forall x \sim Fx \wedge (FA \wedge FB)]$

The method of finite models is recommended by the presence of name letters. The object is a model such that,

$$\frac{\sim[\forall x \sim Fx \wedge (FA \wedge FB)]}{f}$$

and so such that,

$$\frac{\sim[\forall x \sim Fx \wedge (FA \wedge FB)]}{f \quad t \quad t \quad t \quad t \quad t}$$

An invalidating model is to do just three things that I now highlight.

$$\frac{\sim[\forall x \sim Fx \wedge (FA \wedge FB)]}{f \quad t \quad t \quad t \quad t \quad t}$$

It is to make ' $\forall x \sim Fx$ ' true, 'FA' true, and 'FB' true. The rest 'takes care of itself.'

We assign ELEMENTS of the universe-set U, NOT SUBSETS of it, to name letters. Must different objects be assigned to different name letters? No. As sentence letters can have the same truth values, and as predicate letters can be true of the same objects, so name letters can name the same objects. All that is required is that each name letter name exactly one object in U.

Step One	Step Two	Step Three
<u>U: {0}</u>	<u>U: {0}</u>	<u>U: {0, I}</u>
F: {	F: {-	F: {-, I}
A:	A:	A: I

B:	B:	B: I
To start: U must be non-empty.	To make ' $\forall x \sim Fx$ ' true.	To make both 'FA' and 'FB' true.

For confirmation, here is a translation of the premiseless argument

$$\sim[\forall x \sim Fx \wedge (FA \wedge FB)]$$

under the interpretation reached in Step Three.

It is not the case that both, something in {0,1} is in {1}, and both 1 is in {1} and 1 is in {1}.

And here are an evaluation of the connective-free sentences in this translation, each of which is a simple necessary truth of falsehood

It is not the case that both, something in {0,1} is in {1}, and both 1 is in {1} and 1 is in {1}.

So the whole sentence gets the value False.

(6) $FA \rightarrow P. \sim \wedge x(Fx \rightarrow P) \therefore P$

We seek a model that yields the following truth-values.

$$\frac{FA \rightarrow P. \sim \wedge x(Fx \rightarrow P) \therefore P}{t \quad f \quad f}$$

and thus the values

$$\frac{FA \rightarrow P. \sim \wedge x(Fx \rightarrow P) \therefore P}{f \quad t \quad f \quad f \quad t \quad f}$$

The model must yield exactly the three highlighted values, and so it must include the assignment of false to 'P'. In a model sentence letters are assigned truth values, not, as are predicate letters, subsets of U.

Step One	Step Two	Step Three	Step Four
<u>U: {0}</u>	<u>U: {0}</u>	<u>U: {0}</u>	<u>U: {0, I}</u>
F: {	F: {	F: {-	F: {-, I}
P:	P: F	P: F	P: F
A:	A:	A: 0	A: 0

To start: U must be non-empty.	To make 'P' false.	To make 'FA' false. true.	To make ' $\wedge x(Fx \rightarrow P)$ ' false.*
--------------------------------	--------------------	---------------------------	--------------------------------------------------

*Regarding Step Four. ' $\wedge x(Fx \rightarrow P)$ ' says that everything in {0,1} is such that if it is in {1} then P. So it says

for one thing that

$$(0 \text{ is in } \{1\} \rightarrow P)$$

this is true in the model: in the model:

$$\begin{array}{ccc} (0 \text{ is in } \{1\} \rightarrow P) & & \\ f & & f \\ & & t \end{array}$$

So ' $\forall x(Fx \rightarrow P)$ ' is true *in the model of step three*. But ' $\forall x(Fx \rightarrow P)$ ' says, *in the model of step four*, another thing: it says that

$$(1 \text{ is in } \{1\} \rightarrow P)$$

and this is false in the model of this step, which is the final model constructed.

$$\begin{array}{ccc} (1 \text{ is in } \{1\} \rightarrow P) & & \\ t & & f \\ & & f \end{array}$$

$$(8) \forall xFx. \forall xGx. \Lambda x(Fx \rightarrow \sim Gx) \therefore \Lambda x(Hx \rightarrow Fx \vee Gx)$$

The method of finite models is best here, for there is no overlay, and there are no name letters.

$$\begin{array}{cccc} \forall xFx. & \forall xGx. & \Lambda x(Fx \rightarrow \sim Gx) & \therefore \Lambda x(Hx \rightarrow Fx \vee Gx) \\ t & t & t & f \end{array}$$

Step One Step Two Step Three Step Four

$$\begin{array}{cccc} \underline{U: \{0\}} & \underline{U: \{0\}} & \underline{U: \{0, 1\}} & \underline{U: \{0, 1\}} \\ F: \{ & F: \{0 & F: \{0 & F: \{0, - \\ G: \{ & G: \{- & G: \{-, 1 & G: \{-, 1 \end{array}$$

To start For first For second Toward
 premise, and premise third
 toward third premise

Regarding Step Four, consider the conjunct for 1 of the expansion of the third premise for $\{0,1\}$,

$$F1 \rightarrow \sim G1$$

Since 1 is in G, ' $\sim G1$ ' is false. Were 1 F, ' $F1$ ' would be true, making this conditional false.²⁹

To make an expansion of a universal premise true, each conjunct of it must be true.

²⁹In contexts of finite models, it is convenient to assume that numerals are *constant terms* in MQC that in every model name 'their numbers': '0' names 0, '1' names 1, and so on. And it is convenient to let 'true' and 'false' be short for 'true in U' and 'false in U'.

Step Five

$$\begin{array}{l} \underline{U: \{0, 1, 2\}} \\ F: \{0, \sim, - \\ G: \{-, 1, - \\ H: \{-, \sim, 2 \end{array}$$

Regarding Step Five, the conclusion is true in the model of Step Four: that is, each conjunct of its expansion for $\{0, 1\}$ is true. The conjunct for 2 of its expansion for $\{0, 1, 2\}$

$$H2 \rightarrow F2 \vee G2$$

is false in the model of Step Five.

.....

A Possible Exercise. Invalidate the argument

$$\forall x(Fx \wedge \sim Gx). \forall x(\sim Fx \wedge Gx) \therefore \Lambda x[\forall y(Hx \vee Hy) \rightarrow Fx \vee Gx]$$

either by the truth functional expansion method, or with a finite model. Hint: To decide which way to go, think about the size of universes in which this argument can have true premises and a false conclusion.

9. OPEN ARGUMENTS

Definition: An argument is *open* if and only if it has a premise that is not a sentence (i.e., that is a formula in which some variable is free) or its conclusion is not a sentence.

9.1 Our invalidation methods are addressed to arguments whose premises and conclusions are sentences true or false. To prepare for invalidations of open arguments we need several linked definitions.

Let the *closure* of a formula ϕ that is not a sentence be a universal generalization,

$$\Lambda \alpha_1 \dots \Lambda \alpha_n \phi$$

that is a sentence. Let every sentence be a closure of itself.

For example, ' $\Lambda y \Lambda x(Fx \rightarrow Gy)$ ', ' $\forall xFx$ ' is a closure of ' $(Fx \rightarrow Gy)$ '. And ' $\forall xFx$ ' is a closure of itself

Let an argument A' be a closure of A if and only if the premises of A' are closures of premises of A, and the conclusion of A' is a closure of the conclusion of A.

- | | | |
|----|------------------------------------|-------------------------------------------------------------------------------------------------------|
| 2. | $\Lambda x(Fa \wedge \Lambda yGx)$ | (CD) |
| 3. | $Fa \wedge \Lambda yGa$ | 2, UI |
| 4. | Ga | 3, S, UI [‘Ga’ comes from ‘Ga’ by proper substitution of ‘a’ for ‘y’ since ‘y’ is not free in ‘Ga’!!] |
| 5. | $\forall x(Fx \wedge Gx)$ | 3, S (Fa), 4, EG |

- | | | |
|----|------------------|---------------------------|
| 6. | ΛA | 2, 5, <i>modus ponens</i> |
| 7. | $\sim \Lambda A$ | premise |

10. VALIDITY IN THE MONADIC QUANTIFIER CALCULUS

The trick is somehow to reduce the problem of

$$\therefore \Lambda x(Fy \wedge \Lambda yGx) \rightarrow \forall x(Fx \wedge Gx)$$

to this easy problem. Here is one way of doing this.

- | | | |
|----|-----------------------------------------------------------------------------------------------|------------------|
| 1. | <i>SHOW</i> $\Lambda x(Fy \wedge \Lambda yGx) \rightarrow \forall x(Fx \wedge Gx)$ | (8, DD) |
| 2. | <i>SHOW</i> $\Lambda a[\Lambda x(Fa \wedge \Lambda yGx) \rightarrow \forall x(Fx \wedge Gx)]$ | (3, UD) |
| 3. | <i>SHOW</i> $\Lambda x(Fa \wedge \Lambda yGx) \rightarrow \forall x(Fx \wedge Gx)$ | (7, CD) |
| 4. | $\Lambda x(Fa \wedge \Lambda yGx)$ | (CD) |
| 5. | $Fa \wedge \Lambda yGa$ | 4, UI |
| 6. | Ga | 5, S, UI |
| 7. | $\forall x(Fx \wedge Gx)$ | 5, S (Fa), 6, EG |
| 8. | $\Lambda x(Fy \wedge \Lambda yGx) \rightarrow \forall x(Fx \wedge Gx)$ | 2, UI(y) |

I believe that Donald Kalish would call the strategy implemented on lines (2) and (8) a case of ‘clearing the variables’.

(4) Promised in Section 4 above, under symbolization for exercise (13), was a derivation validating the symbolic argument

$$(\forall xFx \rightarrow Lx). \sim \Lambda A \therefore \sim \forall xFx$$

which has, under the scheme,

F: *a* was found; L: *a* was left in H525; A: my watch

the patently invalid English translation,

If anything was found, it was left in H525. My watch was not left in H525. Therefore, *nothing* was found!!

Here is the promised derivation.

- | | | |
|----|-----------------------------------------------------|------------------------------------------------|
| 1. | <i>SHOW</i> $\sim \forall xFx$ | (6, 7, ID) |
| 2. | $\forall xFx$ | (ID) |
| 3. | <i>SHOW</i> $\Lambda x(\forall xFx \rightarrow Lx)$ | (4, UD: ‘x’ is not free on an antecedent line) |
| 4. | $\forall xFx \rightarrow Lx$ | premise |
| 5. | $\forall xFx \rightarrow \Lambda A$ | 2, UI |

The Monadic Quantifier Calculus is a logic for non-empty domains, and for denoting terms. As a consequence, results, when it is applied incautiously to English arguments, can be misleading. Section 10.1 illustrates these perils with patently invalid English arguments that are valid in The Monadic Quantifier calculus in the sense assumed in *Logic*. Section 10.2 shows how these unwanted validations can be blocked by fixing the definition of *valid-in-MQC* for English arguments. Section 10.3 explains that this fix fails to block unwanted validations of another kind. Two varieties of this kind are illustrated in in Sections 10.4 and 10.5. Section 10.6 describes a variety of English arguments for which there are no unwanted validations.

10.1 *The domains and terms of MQC*

10.1.1 The Monadic Quantifier Calculus is a logic for *non-empty* domains. Consider the argument,

$$(1) \sim \forall xFx. \sim \forall x \sim Fx \therefore \forall xGx$$

This argument is valid-in-MQC, i.e., there is a completed derivation from its premises to its conclusion.³³ Even so it has patently invalid English translations under some interpretations. If there are no things of the kind specified in an interpretation as the range of quantifiers, then translations of the premises of (1) express truths, and the translation of its conclusion expresses a falsehood.

Consider the interpretation,

U: {actually existing unicorns}
F: *a* is furry
G: *a* is green

Translation under an interpretation relativizes phrases of quantity to the domain of the interpretation. (Please see Section 7.1.2 above.) The translation of (1) under this interpretation is invalid.

³³Here is a derivation for this argument.

- | | | |
|----|---------------------------|----------------------|
| 1. | <i>SHOW</i> $\forall xGx$ | (2, 3, ID) |
| 2. | $\sim Fx$ | premise (i), QN, UI |
| 3. | $\sim \sim Fx$ | premise (ii), QN, UI |

(1') It is not the case that there is an actually existing unicorn that is furry. (*True*, for there are no actually existing unicorns.) It is not the case that there is an actually existing unicorn that is not furry. (*True*, for the same reason.) \therefore There is an actually existing unicorn that is green. (*False*, for the same reason.)

10.1.2 The Monadic Quantifier Calculus is a logic for denoting terms. Consider the argument,

$$(2) \ \forall xFx \ \therefore \ FA$$

This argument is valid-in-MQC: its conclusion can be inferred from its premise by UI. Even so it has, under some interpretations, translations that are not valid. Please see Section 7.1.2.2.2 above regarding translation of an English argument *under and interpretation*. It has under the interpretation,

$$\begin{aligned} U: & \{ \text{everything that actually exists} \} \\ F: & a \text{ is identical with itself} \\ A: & \text{The Tooth Fairy} \end{aligned}$$

the invalid translation,

(2') Everything that actually exists is identical with itself. (*True*) \therefore The Tooth Fairy is identical with itself. (*False*)

It is not merely true but necessarily true that everything that actually exists is identical with itself. That The Tooth Fairy is identical with itself, entails that The Tooth Fairy actually exists.* This is false: The Tooth Fairy is a mythical being. So the conclusion is false, since whatever entails something that is false is itself false. (*This claim is somewhat contentious.)

The point of this section is that a symbolic argument that is valid-in-MQC can have translations under interpretations that are not valid. The following argument is a less contentious illustration of this point.

(2'') Every actually existing being is an actually existing being. (*True*) \therefore The Tooth Fairy is an actually existing being. (*False*)

This patently invalid English argument is a translation of symbolic argument (2) under the interpretation,

$$\begin{aligned} U: & \{ \text{actually existing beings} \} \\ F^1: & a \text{ is an actually existing being} \\ A^0: & \text{The Tooth Fairy} \end{aligned}$$

10.2 *Valid-in-MQC English arguments*. According to *Logic*, an English argument is “valid in the monadic quantifier calculus if it has a valid symbolization [i.e., if it has a symbolization that is valid-in-MQC]” (p. 154). This definition has the unfortunate result of making patently invalid English arguments (1'), (2'), and (2'') valid in the monadic quantifier calculus.

These unwanted results can be avoided by defining 'valid-in-MQC' for English arguments not in terms merely of symbolizations under schemes as *Logic* does, and not in terms of symbolizations under interpretations without restriction, but in terms of symbolizations under proper interpretations thus:

Definition. An English argument is valid-in-MQC if and only if has a valid symbolization under a proper interpretation.

This definition has the following 'articulation':

An English argument A is valid-in-MQC if and only if there exists an interpretation $\langle U, S \rangle$ in which U is non-empty and S is a scheme of abbreviation that is proper relative to U, such that a symbolic argument A' that is valid-in-MQC is a symbolization of A under $\langle U, S \rangle$ (or in other words, such that A is translation of under $\langle U, S \rangle$ of a symbolic argument A' that is valid-in-MQC).

It may be recalled from Section 7.1.3 that a scheme S is proper relative to a domain U if and only if every English name abbreviated in S names something in U.

10.3 English arguments (1'), (2'), and (2'') are not valid-in-MQC in the sense just defined. While they are translations under *interpretations* of valid-in-MQC symbolic arguments, they are not translations of valid-in-MQC symbolic arguments under *proper* interpretations.

These English arguments have true premises and false conclusions. Such English arguments are termed *false arguments* in *Logic* (p. 35). No false English argument translates a valid-in-MQC symbolic argument under a *proper* interpretation.³⁴ In other words, no false English

³⁴This is entailed by the 'foundational principle' of Section 7.1.1 above:

A symbolic argument whose premises and conclusions are sentences is valid-in-MQC if and only if there does not exist for it a proper interpretation such that translations under this interpretation of its premises are English sentences that express truths, and translation under this interpretation of its conclusion is an English sentence that expresses a falsehood.

argument is valid-in-MQC in the sense recently defined. *However*, there are *invalid* English arguments that are valid-in-MQC.³⁵ There are two *varieties* of invalid English arguments that are valid-in-MQC. One is served by denoting terms that are not *necessarily* denoting. The other is served by non-empty domains that are not *necessarily* non-empty.

10.4 *Possibly non-denoting terms*.³⁶ English argument,

(2'') Every actually existing tower is identical with itself. ∴ The Eiffel Tower is identical with itself.

is valid-in-MQC, since it is a translation of symbolic argument (2) above under the proper interpretation,

U: {actually existing towers}
 F¹: *a* is identical with itself
 A⁰: The Eiffel Tower

which symbolic argument is valid-in-MQC. And English argument (2'') is not a false argument, for its conclusion is not false. But it is an invalid argument. For it is *not* impossible for it to be true that every actually existing tower is identical with itself, though false that The Eiffel Tower is identical with itself. Why not? Because it is not *necessary* that The Eiffel Tower *exists*. And if The Eiffel Tower did not exist, then it would be false that The Eiffel Tower is identical with itself.* But it would still be true that every actually existing tower is identical with itself.

³⁵For the record, *Logic* does not claim that no *invalid* English argument is symbolized under a scheme of abbreviation by a symbolic argument that is valid in The Monadic Quantifier Calculus. It *says* that this calculus has been designed to avoid validations of *false* English arguments, and implies only that none of these have symbolizations under schemes that are valid in this calculus. *Cf.*:

“[W]e...call an English argument *valid* (in the monadic quantifier calculus) if it has a valid symbolization [under a scheme]. In stating rules of inference and directions for constructing derivations we have...imposed a number of restrictions...needed in order to prevent the validation of *false English arguments* (arguments whose premises are true sentences of English and whose conclusions are false sentences of English).” (P. 155).

³⁶Thanks to Bernie Katz for communicating a challenge to the validity of the English argument,

Everything is identical with itself. Therefore, Socrates is identical with himself.

This argument is a translation of symbolic argument (2) under the scheme of abbreviation,

F¹: *a* is identical with itself; A⁰: Socrates

The present section which concerns invalid translations under proper interpretations elaborates that challenge.

Again for the invalidity of (2''), an argument is valid if and only if its premises *logically entail* its conclusion, and it is not the case that the premise of (2'') logically entails its conclusion. This – that the premise of (2'') does not entail its conclusion – can be seen in two ways. *First*, the conclusion of (2'') logically entails that The Eiffel Tower exists,* and this is not logically entailed by the premise of (2''). So this premise does not logically entail that conclusion, since in general, if p logically entails q, and q logically entails r, then p logically entails r. *Second*, the premise of (2'') is necessarily true: it is necessarily true that, for anything, if it is an actually existing tower, then it is identical with itself. But the conclusion of (2'') is not necessarily true. Why not? Because, (i), this conclusion logically entails that The Eiffel Tower exists,* (ii), it is not necessarily true that The Eiffel Tower exists, and, (iii), only necessary truths are logically entailed by necessary truths. So – since the premise of (2'') and its conclusion is not necessarily true – the conclusion of (2'') is not logically entailed by its premise. (*Somewhat contentious claims have been asterisked. Each comes to the claim that being self-identical is an existence-entailing property, as are most, but not all, properties. Being a mythical character is not existence-entailing.)

The point of this section, that an English argument can be invalid though it is valid-in-MQC, has a simpler and less contentious illustration in this argument:

(2''') Every actually existing tower in France is an actually existing tower in France. ∴ The Eiffel Tower is an actually existing tower in France.

English argument (2''') is valid-in-MQC: it is a translation of the valid-in-MQC symbolic argument (2) under the proper interpretation,

U: {actually existing towers in France}
 F¹: *a* is an actually existing tower in France
 A⁰: The Eiffel Tower

And English argument (2''') is not a false argument, since its conclusion is not false. However, if The Eiffel Tower were not actually to exist, then, though it would still be true that every actually existing tower in France was an actually existing tower in France, it would be false that The Eiffel Tower was an actually existing tower in France. So argument (2''') is not a valid argument. It is *not* impossible that its premise is true and its conclusion false. It is merely *not the case* that its premise is true and its conclusion false. So English argument (2''') is valid-in-MQC. And it is not a false argument. But it is an invalid argument. There is nothing contentious about these points.

10.5 *Possibly empty domains*. The definition of ‘valid-in-MQC’ for English arguments blocks ranges of invalid English arguments from being valid-in-MQC. It deals well

with problems made by interpretations with non-denoting terms, and with interpretations with empty domains. It says that such interpretations do not count! However this definition can have a problem with proper interpretations that feature terms such as 'The Eiffel Tower' that are not necessarily denoting. It can have a similar problem with proper interpretations whose domains of discourse are not necessarily non-empty. To illustrate, the argument,

$$(1) \sim \forall x Fx. \sim \forall x \sim Fx \therefore \forall x Gx$$

which is valid-in-MQC, has, under the proper interpretation,

U: {billiard balls}

F: *a* is red

G: *a* is not identical with itself

the invalid English translation,

(1'') It is not the case that there is a billiard ball that is red. It is not the case that there is a billiard ball that is not red. \therefore There is a billiard ball that is not identical with itself.

For the *invalidity* of this English argument we have that *it would be* that its premises were true and its conclusion false, if there were no billiard balls, and that it is *possible* that there should have been no billiard balls. It is thus *not* impossible that the premises of (1'') should be true while its conclusion is false. This would be impossible, if this English argument were valid. So it is not valid.

Argument (1'') is *not* a *false* English argument. Its premises are false, so it does not have true premises and a false conclusion. But argument (1'') is an *invalid* argument. It is another example of an English argument that is *valid-in-MQC* in the sense that it is a translation under a proper interpretation of a symbolic argument that is valid-in-MQC, though it is *not* valid. For an argument is valid only if it is *impossible* that its premises are true and its conclusion false, and this is *not* impossible in the case of English argument (1'').

10.6 Let an *arithmetic interpretation* be a proper interpretation $\langle U, S \rangle$ in which *U* is either the set of natural numbers or a non-empty subset of the set of natural numbers such as the set of odd natural numbers, or the set that contains a thing *n* if and only if *n* is either 0 or 1. Natural numbers necessarily exist: each exists in every possible world. So if $\langle U, S \rangle$ is an arithmetical interpretation, then *U* is necessarily non-empty, and terms abbreviated in *S* are necessarily denoting. If an English argument is a translation under an arithmetic interpretation of a symbolic argument that is valid-in-MQC, then it is not only not false, but it is also not invalid. Every argument that is a translation under an arithmetic interpretation of a valid-in-MQC symbolic argument is itself valid.