1. Overview

Having seen how modal sentential logic works, we now turn to modal predicate logic. This follows the same progression as introductory symbolic logic; one does sentential logic, followed by predicate logic.

This chapter is divided into three parts. Part A reviews the general ideas of ordinary predicate logic. Part B develops Simple Modal Predicate Logic (MPL), which is obtained by simply combining the apparatus of modal sentential logic (including indexing) with the apparatus of ordinary predicate logic. The system so obtained is fairly simple-minded; in particular, MPL ignores proper nouns, function signs, identity, and descriptions.\(^1\) It also ignores the issue of quantificational domain, which is taken up in Part C, which presents actualist quantificational principles.

A. Ordinary Predicate Logic

2. Introduction

Ordinary predicate logic is the logical system most students encounter after doing sentential logic. The move from sentential logic to predicate logic involves three major new concepts.

(1) noun phrases
(2) predicates
(3) quantifiers

3. Noun Phrases

Noun phrases (NPs) constitute a new underlying grammatical category. Syntactically speaking, an NP is an expression that can serve as an argument/complement of a relational expression. For example, an NP can serve as a subject or object of a verb. Among noun phrases are a special sub-category of definite-noun-phrases (DNPs). These are not defined syntactically, but rather semantically. Specifically, a DNP is a phrase that purports to denote an entity, where an entity is an element of the underlying domain of discourse, which includes all particulars – persons, places, and things. For example, if we are doing Arithmetic, then the domain of discourse consists of all (natural) numbers, and if we are doing Person Theory, then the domain of discourse consists of all persons.

<table>
<thead>
<tr>
<th>category</th>
<th>examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>proper nouns</td>
<td>Mozart</td>
</tr>
<tr>
<td>pronouns</td>
<td>I</td>
</tr>
<tr>
<td>compound DNPs</td>
<td>Jay's mother</td>
</tr>
<tr>
<td>descriptive DNPs</td>
<td>the man standing next to Bill</td>
</tr>
</tbody>
</table>

4. Two Key Simplifications in (Elementary) Predicate Logic

1. Plural NPs are Ignored

Noun phrases have a grammatical feature called number, which is either singular or plural.\(^2\) In the case of pronouns, this is drilled into us in middle school, if not before. For example, we know that the pronoun ‘we’ is first person plural, and we know that the pronoun ‘she’ is third person singular. In the case of other such expressions, perhaps the easiest way to decide number is to ask what the appropriate form of the verb ‘to be’ is — if the appropriate verb is ‘is’, then the noun phrase is singular; if the appropriate verb is ‘are’, then the noun phrase is plural; simple as that! And pronouns mostly work the same way; I leave the reader to think about what the exceptions are.

---

\(^1\) These notions are added in a later chapter on full quantified modal logic.

\(^2\) Other number-values occur in natural languages, but we consider just singular and plural here.
In elementary predicate logic, only singular noun phrases – officially called singular terms – are considered, whereas plural noun phrases are either ignored or finessed.

2. Complex DNPs are Ignored

Also, a (definite) noun phrase can be grammatically simple or complex, but predicate logic completely ignores their internal structure. For example, from the viewpoint of predicate logic, all of the following are treated as simple (atomic).

Jay's mother
5 + 7
the woman standing near the window

5. Predicates

In addition to definite-noun-phrases, predicate logic concerns predicates. Whereas the category $S$ of sentences, and the category $D$ of definite-noun-phrases, are primitive types, the category of predicates is a derivative type, just like connectives. The notion of predicate is defined as follows, which is followed by the subordinate definition of $k$-place predicate.

(d1) A predicate is an expression with zero or more blanks such that, filling these blanks with DNPs results in a sentence.

(d2) Where $k$ is any natural number (0, 1, 2, ...),

a $k$-place predicate is a predicate with $k$ places (blanks).

We propose to depict the various types of predicates as follows.

\[
\begin{align*}
&D^0 \rightarrow S & \text{0-place predicates} \\
&D^1 \rightarrow S & \text{1-place predicates} \\
&D^2 \rightarrow S & \text{2-place predicates} \\
&\text{etc.}
\end{align*}
\]

Compare this with the corresponding account of connectives.

\[
\begin{align*}
&S^0 \rightarrow S & \text{0-place connectives} \\
&S^1 \rightarrow S & \text{1-place connectives} \\
&S^2 \rightarrow S & \text{2-place connectives} \\
&\text{etc.}
\end{align*}
\]

Notice that we allow $k$ to be 0. Grammatically, a 0-place predicate is a predicate that takes no grammatical subject. The best examples of subject-less declarative sentences occur in weather reports – for example, ‘it is raining’, ‘it is snowing’, etc. Even though these sentences have an official grammatical subject ‘it’, it is obvious that ‘it’ does not refer to anything. What exactly is "it" in ‘it is raining’? Since there is no "real" subject in such sentences, when we symbolize them in predicate logic (see below), the word ‘it’ simply disappears.

The standard symbolization technique for predicates follows a general pattern, given as follows.

---

3 By ‘elementary predicate logic’ we mean predicate logic as it is customarily taught in elementary logic classes. Elementary predicate logic ignores plural noun phrases such as in ‘the crows mobbed the hawk’ for purely pedagogical reasons, not theoretical reasons.

4 Syntactic complexity is related to, but not identical to, semantic complexity. For example, ‘the Eiffel Tower’ consists of three words, but it is regarded as semantically simple.
Predicates are symbolized by upper case Roman letters.

Singular terms are symbolized by lower case Roman letters.\(^5\)

The predicate is written first, followed by its arguments, surrounded by square-brackets, and separated by commas.

**Examples**

<table>
<thead>
<tr>
<th>subject</th>
<th>(\emptyset)</th>
<th>it</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicate</td>
<td>R</td>
<td>is raining</td>
</tr>
<tr>
<td>sentence</td>
<td>R[ ]</td>
<td>it is raining</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>subject</th>
<th>J</th>
<th>Jay</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicate</td>
<td>T</td>
<td>is tall</td>
</tr>
<tr>
<td>sentence</td>
<td>T[J]</td>
<td>Jay is tall</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>subject</th>
<th>J</th>
<th>Jay</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicate</td>
<td>R</td>
<td>respects</td>
</tr>
<tr>
<td>object</td>
<td>K</td>
<td>Kay</td>
</tr>
<tr>
<td>sentence</td>
<td>R[J,K]</td>
<td>Jay respects Kay</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>subject</th>
<th>J</th>
<th>Jay</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicate</td>
<td>R</td>
<td>recommends</td>
</tr>
<tr>
<td>direct object</td>
<td>K</td>
<td>Kay</td>
</tr>
<tr>
<td>indirect object</td>
<td>L</td>
<td>(to) Elle</td>
</tr>
<tr>
<td>sentence</td>
<td>R[J,K,L]</td>
<td>Jay recommends Kay to Elle</td>
</tr>
</tbody>
</table>

Note that it is customary to drop brackets and commas in writing atomic formulas of these forms.

\(^5\) More specifically, we propose to use ordinary lower case letters for constants and variables, and to use small caps for proper nouns, so for example, ‘K’ abbreviates ‘Kay’. See Chapter “General First-Order Logic” for the logical difference between constants and proper nouns.
6. Quantifiers and Quantifier Phrases

Predicate Logic involves predicates, but more importantly it involves quantification, which plays a central role in predicate logic inferences. For this reason, predicate logic is sometimes called *quantifier logic*. A quantifier is a special kind of modifier that conveys quantity, as illustrated in the following examples.

<table>
<thead>
<tr>
<th>quantifier phrase</th>
<th>quantifier</th>
<th>common noun phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>every</td>
<td>dog</td>
<td></td>
</tr>
<tr>
<td>some</td>
<td>brown dog(s)</td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>dog(s) in the yard</td>
<td></td>
</tr>
<tr>
<td>any</td>
<td>dog(s) I own</td>
<td></td>
</tr>
<tr>
<td>all</td>
<td>cats</td>
<td></td>
</tr>
<tr>
<td>few</td>
<td>cats that hunt mice</td>
<td></td>
</tr>
<tr>
<td>several</td>
<td>cats from Egypt</td>
<td></td>
</tr>
<tr>
<td>most</td>
<td>cats with extra toes</td>
<td></td>
</tr>
</tbody>
</table>

The basic syntactic principle is:

```
| a quantifier followed by a common-noun-phrase |
| is a quantifier-phrase,                     |
| which is a species of noun-phrase.          |
```

For example, the sentence

```
every well-fed dog is happy
```

may be parsed as follows.

```
S
   NP VP
     QP is happy
       Q every CNP well-fed dog
       △
```

Note that VP (verb phrase) and CNP (common noun phrase) are syntactic categories that correspond to 1-place predicate (D→S).

Although a QP is an NP, it is not a DNP, which is *semantically* defined to be a phrase that denotes an entity – i.e., a particular in the domain of discourse. Does ‘every dog’ denote a dog, or any particular entity? No!

The moral we draw is that, even though the following sentences all have the same *syntactic structure*,

---

6 Quantifier logic also includes the logic of function-signs, which we examine in a later chapter.

7 More generally, *quantifier* is a sub-species of *determiner*, which includes non-quantifier words like ‘the’, ‘this’, and ‘my’. Accordingly, *quantifier-phrase* is a sub-species of *determiner-phrase*, which is a sub-species of *noun-phrase*.

8 The appearance of the triangle ‘△’ indicates that the phrase has further structure that we ignore, or take for granted.
Kay is virtuous
Elle is virtuous
every woman is virtuous
some woman is virtuous
no woman is virtuous

they do not have the same semantic structure. The first two have the following semantic structure.

Kay is virtuous
Elle is virtuous

What about the quantifier-phrase sentences?

every woman is virtuous
some woman is virtuous
no woman is virtuous

The contemporary approach to this question, due to Montague, is that whereas ‘is virtuous’ denotes a property of entities (presumably people), ‘every woman’ denotes a property of properties (a second-order property) So ‘every woman is virtuous’ does not say that an entity “every woman” has the first-order property of being virtuous. Rather, it says that the property “is virtuous” has the second-order property of applying to all women. This can be categorially rendered as follows.

The difference between

Kay is virtuous
every woman is virtuous

goes to whether ‘is virtuous’ is the function or the argument.

function argument form
is virtuous Kay [is virtuous](Kay)
every woman is virtuous [every woman](is virtuous)

Montague (1973), “The Proper Treatment of Quantification in Ordinary English”.

---

<table>
<thead>
<tr>
<th>NP (subject)</th>
<th>VP (predicate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kay</td>
<td>is virtuous</td>
</tr>
<tr>
<td>Elle</td>
<td>is virtuous</td>
</tr>
<tr>
<td>every woman</td>
<td>is virtuous</td>
</tr>
<tr>
<td>some woman</td>
<td>is virtuous</td>
</tr>
<tr>
<td>no woman</td>
<td>is virtuous</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>D → S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kay</td>
<td>is virtuous</td>
</tr>
<tr>
<td>Elle</td>
<td>is virtuous</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>?</th>
<th>D → S</th>
</tr>
</thead>
<tbody>
<tr>
<td>every woman</td>
<td>is virtuous</td>
</tr>
<tr>
<td>some woman</td>
<td>is virtuous</td>
</tr>
<tr>
<td>no woman</td>
<td>is virtuous</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(D → S) → S</th>
<th>D → S</th>
</tr>
</thead>
<tbody>
<tr>
<td>every woman</td>
<td>is virtuous</td>
</tr>
<tr>
<td>some woman</td>
<td>is virtuous</td>
</tr>
<tr>
<td>no woman</td>
<td>is virtuous</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Kay</th>
<th>is virtuous</th>
</tr>
</thead>
</table>
7. Quantifier Phrases as Logical-Operators

Our task then is to develop a logical system that formally implements the above insight. We pursue this in stages. First, we show how to transform the usual QP-VP sentences so as to reveal their underlying logical form. Then we show how this connects to Montague’s proposal.

Consider a well-known QP-VP sentence.

every human is mortal

We take this sentence and transform it,\(^{10}\) by moving the subject phrase from its original grammatical position forward in the sentence to produce the following sentence,

every human is such that he/she is mortal

which we propose to analyze as follows.

<table>
<thead>
<tr>
<th>every human is such that</th>
<th>he/she is mortal</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\forall H/x)</td>
<td>(M[x])</td>
</tr>
</tbody>
</table>

Note the new item – the variable ‘\(x\)’ – which performs a number of inter-related grammatical functions.

1. it marks the original grammatical location of the quantifier phrase;
2. it symbolizes the third person singular pronoun ‘he/she’;
3. it marks the quantifier phrase ‘every human’ \(\forall H\) as the antecedent of the pronoun ‘he/she’.

 Whereas linguists prefer to stop the analysis with the formula

\[\forall H/x \ M[x]\]  
every human \(x\) is such that \(x\) is mortal

logicians propose a further transformation, which reduces specific quantifier phrases (for example)

every human, every apple, every electron, …

to a single generic quantifier phrase:

every thing

This is accomplished by the following paraphrase.

every \(A\) is such that it is \(B\)

\[\iff\]

every thing is such that, IF it is \(A\), THEN it is \(B\)

Accordingly, our earlier sentence

every human is such that he/she is mortal

is paraphrased as:

every thing is such that, IF it is human, THEN it is mortal

which we propose to analyze as follows.

<table>
<thead>
<tr>
<th>every thing is such that</th>
<th>if it is a human, then it is mortal</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\forall x)</td>
<td>((Hx \rightarrow Mx))</td>
</tr>
<tr>
<td>(\forall x(Hx \rightarrow Mx))</td>
<td></td>
</tr>
</tbody>
</table>

\(^{10}\) This transformation, known as quantifier-movement, is on a par with the transformation called wh-movement (‘who’ ‘what’ ‘how’ etc.), which is probably the most famous grammatical transformation, originally proposed by Noam Chomsky [+++reference+++].
8. The Corresponding Analysis of ‘Some’

A similar story can be told about sentences involving ‘some’. We start with the sentence

some human is mortal

and we transform it as follows.

some human is such that he/she is mortal

which we propose to analyze as follows.

\[
\begin{array}{|c|c|}
\hline
\text{some human is such that} & \text{he/she is mortal} \\
\exists H/x & M[x] \\
\hline
\end{array}
\]

Next, we perform a further transformation, which reduces specific quantifier phrases

some human, some apple, some electron, …

to a single generic quantifier phrase:

some thing

using the following paraphrase.

some A is such that it is B ≡ some thing is such that, it is A, AND it is B

Accordingly, our earlier sentence

some human is such that he/she is mortal

is paraphrased as:

some thing is such that, it is human, AND it is mortal

which we propose to analyze as follows.

\[
\begin{array}{|c|c|}
\hline
\text{some thing is such that} & \text{it is human, and it is mortal} \\
\exists x & (Hx \land Mx) \\
\hline
\end{array}
\]

\[
\exists x(Hx \land Mx)
\]

9. Connection to Montague

The connection to Montague involves two steps. First, one reparses the formula

\[
\forall x(Hx \to Mx)
\]

as

\[
\forall [ x(Hx \to Mx) ]
\]

which we analyze as follows.

\[
\begin{array}{|c|c|}
\hline
\text{function} & x(Hx \to Mx) \\
\text{is universal} & \text{being an } x \text{ such that if } x \text{ is human then } x \text{ is mortal} \\
\hline
\end{array}
\]

Similarly, one reparses the formula

\[
\exists x(Hx \land Mx)
\]

as

\[
\exists [ x(Hx \land Mx) ]
\]

\[\text{This is usually rewritten with a variable-binding operator ‘λ’ to produce ‘λx(Hx \to Mx)’, which is a one-place predicate. The resulting formal system is known as the Lambda Calculus.}\]
which we analyze as follows.

| ∃         | x(Hx & Mx)       |
| function  | argument         |
| is instantiated | being an x such that x is Human and x is Mortal |

This treats quantifiers as monadic predicates, which most linguists reject (for various reasons). It is grammatically, if not logically, more perspicuous to treat quantifiers as dyadic predicates, more in accordance with the following analyses.

| ∀         | xHx         | xMx          |
| function  | argument 1  | argument 2   |
| is included in | being an x such that x is Human | being an x such that x is Mortal |

| ∃         | xHx         | xMx          |
| function  | argument 1  | argument 2   |
| overlaps  | being an x such that x is Human | being an x such that x is Mortal |

Whereas many language theorists are content treating ∀ and ∃ as two-place operators, which involves ternary trees as follows.

![Ternary Tree Diagram](image)

others prefer to convert ∀ and ∃ into one-place operators, in which case we have binary trees, as follows.

![Binary Tree Diagram](image)

The latter can be paraphrased as follows.

| ∀         | xHx         | xMx          |
| QP        | ∀(xHx)      | ∀(xMx)       |
| quantifier | CNP         | VP           |
| ∀         | CNP         | xMx          |

| ∃         | xHx         | xMx          |
| QP        | ∃(xHx)      | ∃(xMx)       |
| quantifier | CNP         | VP           |
| ∃         | CNP         | xMx          |

being an x such that x is mortal has the property of applying to all humans |

being an x such that x is mortal has the property of applying to at least one human
10. Scope

Treating quantifiers as logical-operators allows us to construct formulas that are considerably more complex than any formula that is written in subject-verb-object form. It also allows us to construct a powerful, but simple, set of inference rules, as we see later. Equally importantly from a grammatical viewpoint, treating quantifiers as logical-operators endows them with a new logico-grammatical feature – *scope*.

Scope, which is an algebraic notion, is a prominent feature of both symbolic logic and symbolic arithmetic. For example, the following arithmetical expressions,

(e1) two plus three times four
(e2) the square root of four plus five
(e3) minus two squared

are all ambiguous. For example, (e1) is ambiguous between the following.

(i1) two plus three … times four
(i2) two plus … three times four

The logical pause represented by ‘…’ tells us which operator is the major operator – i.e., which operator has wide scope.

Now, as everyone knows, the official algebraic method of indicating scope is to use parentheses, which are prominent in both sentential logic and arithmetic. For example, the two informal expressions (i1) and (i2) can be replaced by the following algebraic expressions, respectively.

(a1) \((2 + 3) \times 4\) \[= 20]\n(a2) \(2 + (3 \times 4)\) \[= 14]\n
Similarly, (e2) is ambiguous between the following.

the square root of four … plus five \(sr(4) + 5\) \[= 7]\nthe square root of … four plus five \(sr(4 + 5)\) \[= 3]\n
Finally, (e3) is ambiguous between the following.

minus two … squared \((-2)^2\) \[= 4]\nminus … two squared \(-(2^2)\) \[= -4]\n
Sentential examples follow a similar pattern. For example, the following expressions are ambiguous.

\(\neg A\) and \(B\)
\(\neg A\) … and \(B\) \(\neg A \& B\)
\(\neg \) … \(A\) and \(B\) \(\neg (A \& B)\)

\(A\) and \(B\) or \(C\)
\(A\) and \(B\) … or \(C\) \((A \& B) \lor C\)
\(A\) and … \(B\) or \(C\) \(A \& (B \lor C)\)

---

12 These are examples of harmful ambiguity, which are distinguished from expressions like ‘two plus three plus four’, which are ambiguous, but not harmfully so. Also note that numerous conventions can be adopted according to which expressions without needed parentheses have *preferred* parses. For example, some calculators will parse the input string ‘2 + 3 \times 4’ by adding 2 and 3, and then multiplying the result by 4. Other calculators treat \(\times\) as out-scoping + by *default*, and accordingly add 2 to the result of multiplying 3 by 4.
11. **Quantifier Scope**

Let us next consider the quasi-formula

\[ \sim M[\forall H] \]

which corresponds roughly to the sentence

*every human is not mortal*

When we transform the quantifier phrase ‘every human’ into an operator, we notice immediately that there are two equally plausible ways it can be moved forward in the formula. It can be moved in front of the ‘M’, or it can be moved in front of the ‘\(\sim\)’, which results in the following two formulas, respectively.

\[
\begin{align*}
(m1) & \sim [\forall H/x] M[x] \\
(m2) & [\forall H/x] \sim M[x]
\end{align*}
\]

In the first case, the quantifier is given *narrow scope* relative to the negation operator; in the second case, the quantifier is given *wide scope relative* to the negation operator.

The difference between these two becomes more obvious perhaps when we transform the specific quantifier ‘every H’ into a generic quantifier, as follows.

| \(\sim [\forall H/x] M[x]\) | \(\sim \forall x(Hx \rightarrow Mx)\) | *not every human is mortal* |
| \(\forall H/x \sim M[x]\) | \(\forall x(Hx \rightarrow \sim Mx)\) | *every human is immortal* |
Appendix – The Official Syntax of Ordinary Predicate Logic

1. Vocabulary

1. Predicate letters
   0. 0-place: A-Z, A₁-Z₁, A₂-Z₂, etc.
   1. 1-place: ¹A-¹Z, ¹A₁-¹Z₁, ¹A₂-¹Z₂, etc.
   2. 2-place: ²A-²Z, ²A₁-²Z₁, ²A₂-²Z₂, etc.

2. Individual Variables
t₁-z₁, t₁-z₁, etc.

3. Individual Constants
   a-s, a₁-s₁, etc.

4. SL Connective Symbols
   &, ∨, →, ↔, ~

5. Quantifiers
   ∀, ∃

6. Punctuation symbols
   (, ), [ , ], etc.

2. Rules of Formation

1. Singular-Terms
   1. Every individual variable is a singular-term.
   2. Every individual constant is a singular-term.
   3. Nothing else is a singular-term.

2. Atomic Formulas
   1. If P is an n-place predicate, and τ₁,...,τₙ are singular-terms, then P[τ₁,...,τ₂] is an atomic formula.
   2. Nothing else is an atomic formula.

3. Formulas
   1. Every atomic formula is a formula.
   2. If Φ is a formula, then so is: ¬Φ
   3. If Φ₁ and Φ₂ are formulas, then so are:
      (a) (Φ₁ & Φ₂)
      (b) (Φ₁ ∨ Φ₂)
      (c) (Φ₁ → Φ₂)
      (d) (Φ₁ ↔ Φ₂)
   4. If Φ is a formula and v is a variable, then the following are formulas:
      (a) ∀vΦ
      (b) ∃vΦ
   5. Nothing else is a formula.

Notational Conventions

- $\Phi$ is any formula,
- $\nu$ is any variable.
- For any expression $\varepsilon$,
- $\Phi[\varepsilon/\nu]$ is the formula that results when every free occurrence of $\nu$ in $\Phi$ is replaced by $\varepsilon$.

<table>
<thead>
<tr>
<th>Universal-Out ($\forall O$)</th>
<th>Existential-In ($\exists I$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall \nu \Phi$</td>
<td>$\Phi[o/\nu]$</td>
</tr>
<tr>
<td>$\Phi[a/\nu]$</td>
<td>$\exists \nu \Phi$</td>
</tr>
<tr>
<td>$a$ is any constant</td>
<td>$a$ is any constant</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Universal-Derivation (UD)</th>
<th>Existential-Out ($\exists O$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHOW: $\forall \nu \Phi$</td>
<td>$\exists \nu \Phi$</td>
</tr>
<tr>
<td>SHOW: $\Phi[n/\nu]$</td>
<td>$\Phi[n/\nu]$</td>
</tr>
<tr>
<td>$n$ is any new constant</td>
<td>$n$ is any new constant</td>
</tr>
</tbody>
</table>

Quantifier Negation

<table>
<thead>
<tr>
<th>$\sim \forall \nu \Phi$</th>
<th>$\sim \exists \nu \Phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\exists \nu \sim \Phi$</td>
<td>$\forall \nu \sim \Phi$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tilde-Universal-Out ($\sim \forall O$)</th>
<th>Tilde-Existential-Out ($\sim \exists O$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sim \forall \nu \Phi$</td>
<td>$\sim \exists \nu \Phi$</td>
</tr>
<tr>
<td>$\sim \Phi[n/\nu]$</td>
<td>$\sim \Phi[a/\nu]$</td>
</tr>
<tr>
<td>$n$ is any new constant.</td>
<td>$a$ is any constant.</td>
</tr>
</tbody>
</table>

Definition of ‘old’ and ‘new’

A constant counts as **old** precisely when it occurs in a line that is neither boxed nor cancelled; otherwise, it counts as **new**.\(^\text{13}\)

\(^\text{13}\) The notion of **old** becomes a key player later in free logic, which we will encounter in “General First-Order Logic”. 

B. Simple Modal Predicate Logic

1. Introduction

Simple Modal Predicate Logic is obtained by simply combining the principles of predicate logic with the principles of sentential modal logic. Syntactically, this is very easy to accomplish; we simply add the following clause to the official syntax of predicate logic.

\[(c1) \text{the symbols ‘□’ and ‘♦’ are one-place connectives; in other words, if } \Phi \text{ is a formula, then so are } \Box \Phi \text{ and } \Diamond \Phi.\]

2. Basic Rules for Simple Modal Predicate Logic

The derivation rules for Simple Modal Predicate Logic are also very easy to write down.

1. Sentential Rules – Ordinary SL Rules plus Indexing

As before, the SL rules are the same as ordinary SL, except that they are indexed.

2. Modal Operator Rules (Round up the Usual Suspects!)

For every sentential modal system \(\sigma\), there is a corresponding modal predicate logic, denoted \(\text{MPL}(\sigma)\). For example, \(\text{MPL}(K)\) admits all K-rules, \(\text{MPL}(K4)\) admits all K4-rules, etc.

3. Quantifier Rules – Ordinary Rules plus Indexing

The quantifier rules are the same as ordinary predicate logic, except that they are indexed. For example, the universal-out rule is officially written as follows.

\[
\begin{array}{c|c|c}
\text{Universal-Out (\(\forall\O\))} & & \\
\hline \\
\forall \nu \Phi & /i & \\
\hline \\
\Phi[a/\nu] & /i & \\
\hline \\
a \text{ is any constant} & & \\
\end{array}
\]

3. Examples of Derivations in Modal Predicate Logic

In order to illustrate the derivation rules, we look at a few examples. We begin with an equivalence that is well-known in the history of modal logic – the Barcan formula.

\[(1) \Box \forall x Fx \leftrightarrow \forall x \Box Fx\]

In order to show this formula is logically true in \(\text{MPL}(k)\), it is sufficient to derive each constituent from the other.

\[
\begin{array}{l|l|l}
(1) & \Box \forall x Fx & /0 \text{ As} \\
(2) & \text{SHOW: } \forall x \Box Fx & /0 \text{ UD} \\
(3) & \text{SHOW: } \Box Fa & /0 \text{ } \Box D \\
(4) & \text{SHOW: } Fa & /01 \text{ DD} \\
(5) & \forall xFx & /01 1,\Box O(k) \\
(6) & Fa & /01 5,\forall O \\
\end{array}
\]

---

14 Also, we can optionally add other modal operators, including ‘≤’, and ‘≡’.
15 Recall that the contradiction symbol ‘\(\spadesuit\)’ is either not indexed at all, or indexed by the wild card symbol ‘\(\star\)’.
This shows how $\Box$ and $\forall$ interact – they commute. By duality, the same thing can be said about the relation between $\Diamond$ and $\exists$.

(2) $\Diamond \exists x Fx \leftrightarrow \exists x \Diamond Fx$

Once again, in order to show this formula is logically true in MPL($k$), it is sufficient to derive each constituent from the other.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>$\Diamond \exists x Fx$</td>
</tr>
<tr>
<td>(2)</td>
<td>SHOW: $\exists x \Diamond Fx$</td>
</tr>
<tr>
<td>(3)</td>
<td>$\exists x Fx$</td>
</tr>
<tr>
<td>(4)</td>
<td>$Fa$</td>
</tr>
<tr>
<td>(5)</td>
<td>$\Diamond Fa$</td>
</tr>
<tr>
<td>(6)</td>
<td>$\exists x \Diamond Fx$</td>
</tr>
</tbody>
</table>

$\Box$ interacts nicely with $\forall$, and $\Diamond$ interacts nicely with $\exists$. What about the cross-relations. We start with the relation between $\exists$ and $\Box$. Consider the following exchange.

Q: why are you cleaning up the litter on the street?
A: well, someone's got to!

The following is a natural paraphrase of the answer.

someone must clean up the litter

Now, the sense of ‘must’ may not be obvious; is it legal, ethical, metaphysical, logical? What sort of world is being ruled out? For the sake of concreteness, let us suppose that ‘must’ here means ‘it is legally required that’.

Still there is another question worth asking – what are the relative scopes of ‘someone’ and ‘must’. Depending upon which of the two operators gets wide scope, we have two different readings of ‘someone must clean up the litter’, given as follows.  

(17) 

(t1) $\exists x \Box Cx$

(t2) $\Box \exists x Cx$

In the first reading, $\exists$ has wide scope; this means that there is a particular individual who must be $C$. For example, there is a particular individual who is legally required to clean up the litter. For example, suppose that a person, Jones, has been convicted of littering, and the judge has sentenced Jones to perform community service – in particular, to clean up litter on the street. Therefore, since Jones is legally required to clean up litter, by existential-introduction ($\exists I$), someone is required to clean up litter.

On the second reading, $\Box$ has wide scope. In this case, although it is required that someone clean up the street, no particular person is so required. So long as someone does it, it doesn't matter who, the law is satisfied.

\[17\] There are actually two more readings, which we discuss in the chapter on actuality.
Another example of the difference between wide and narrow scope occurs in the following example. Imagine a TV ad that tells you that you can order anytime by phone; the ad might even say:

someone is always on duty here to take your call

Now, ‘always’ is an example of a box-modality, so the two readings of ‘someone is always on duty here to take your call’ are:

(r1)  \( \exists x \Box D x \)

(r2)  \( \Box \exists x D x \)

The first one is read more or less literally:

there is someone such that, it is always true that he/she is on duty

In other words, there is some poor guy who is always on duty; he is waiting at the phone 24 hours a day!

The more plausible reading is the second one, which says more or less literally:

it is always true that there is someone who is on duty

In other words, no matter when you call, there will be someone on duty to answer your call, although it need not be the same same person every time.

Now, we consider the logical relation between the two readings; this is summarized by the following.

\[
(3) \quad \exists x \Box F x \rightarrow \Box \exists x F x
\]

As one might expect, there is a dual problem involving \( \Diamond \) and \( \forall \). Consider the sentence.

everyone can be a winner!!

Here, let us suppose there is a contest, and ‘everyone’ means “every contestant”. Nevertheless, the sentence is ambiguous. Does it mean that it is possible that every single contestant gets a prize? That is a lot of prizes! Or, does it mean that the contest is fair; no matter who you are, you have a non-zero chance of winning the contest. These two readings may be formulated as follows.

(r1)  \( \Diamond \forall x W x \)  there is a chance that everyone wins

(r2)  \( \forall x \Diamond W x \)  everyone has a chance of winning

As it turns out, the logical relation is given as follows.

\[
(4) \quad \Diamond \forall x F x \rightarrow \forall x \Diamond F x
\]

Once again, the converse is not logically true. The fact that everyone has a chance to win does not logically imply that there is a chance that everyone will win.
C. Actualist Modal Predicate Logic

1. Introduction

In Part B, we examined the logical system that results when one simply combines the principles of modal sentential logic with the principles of non-modal quantifier logic. This process was done in a largely innocent and un-self-conscious manner.

Now, we wish to reconsider what we have created. In particular, we wish to examine more carefully the quantifier principles implicit in Simple Modal Predicate Logic. In particular – what exactly do, or should, ‘∃x’ and ‘∀x’ mean in the context of modal logic? For example, what is the domain of quantification?

Consider the following expressions.

\[ \exists x Fx \ / i \]
\[ \forall x Fx \ / i \]

These read roughly:

- there is something which is F ... at \( i \)
- everything is F... at \( i \)

The issue is how do we understand the adverbial modifier ‘at \( i \)’. We know how the modifier ‘at \( i \)’ interacts with the standard SL connectives, which is summarized in the following principles of World Theory.

\[ \text{wt}(\sim) \quad [\sim P \ / i] \leftrightarrow \sim [P / i] \]
\[ \text{wt}(\&) \quad [(P\&Q) \ / i] \leftrightarrow [P / i] \& [Q / i] \]
\[ \text{wt}(\lor) \quad [(P\lor Q) \ / i] \leftrightarrow [P / i] \lor [Q / i] \]
\[ \text{wt}(\rightarrow) \quad [(P\rightarrow Q) \ / i] \leftrightarrow [P/i] \rightarrow [Q/i] \]
\[ \text{wt}(\leftrightarrow) \quad [(P\leftrightarrow Q) \ / i] \leftrightarrow [P/i] \leftrightarrow [Q/i] \]

What we need are the corresponding principles for the quantifiers.

\[ (\exists ?) \quad [\exists \nu \Phi \ / i] \leftrightarrow ?? \]
\[ (\forall ?) \quad [\forall \nu \Phi \ / i] \leftrightarrow ?? \]

Let us concentrate on the first one. There seem to be at least two readings of the ‘at \( i \)’ modifier. In the following, the original sentence is given, followed by two parsings.

(0) there is something which is F... at \( i \)
\[ [\exists x Fx/i] \]

(1) there is some (generic) thing ...... which is F  ... at \( i \)
\[ \exists x [Fx/i] \]

(2) there is some thing at \( i \)...... which is F ... at \( i \)
\[ \exists [x/i] [Fx/i] \]

Also, the clause

(c) which is F ... at \( i \),

must be parsed. The most plausible reading is:

(r) which is F-at-\( i \),

where ‘F-at-\( i \)’ is a relativized predicate. Specifically, for any given world \( i \), and predicate F, there are the things that are F at \( i \).

Now, back to the original readings. Reading (1) is plausible; indeed, it is the parsing that is implicit in Simple Modal Predicate Logic, as presented in Part A. The second parsing also seems plausible, so long as we can clarify what it means to be a "thing-at-\( i \)".
This is probably best understood by reference to tense (time) logic, in which case the indices are temporal. Temporal indices may be moments, or instants, or intervals, or whatever, depending on your semantic preferences. For example, in theoretical physics, the temporal index is instantaneous time (the "time" of mathematical analysis); yet, instantaneous functions are all defined in terms of limits of interval ("average") functions.\(^{18}\)

In any case, in tense logic it is pretty clear what it means to be a thing-at-\(i\). To be a thing-at-\(i\) is to be a thing that exists at \(i\) (or during \(i\), if \(i\) is an interval). This is understood in terms of birth, life, and death, broadly understood. Specifically, the underlying intuition is that an entity (e.g., the earth) is "born" at some point in time, after which it "lives" for awhile, until it "dies" at some later point in time.\(^{19}\) During (inside) this life-interval, the object exists; outside this life-interval, the object does not exist. Where \(i\) is a temporal index (a time), to be a thing-at-\(i\) is to be a thing whose life-interval includes \(i\).

### 2. Actualism versus Possibilism

This leads us to an important metaphysical question: is there a difference between existence simpliciter and existence-at-a-time. Concerning this question, there are two prominent viewpoints, called actualism and possibilism.

According to actualism, to exist is to be actual. Or, in the specific case of time, temporal actualism claims that to exist is to be present (i.e., existing now).\(^{20}\) Accordingly, actualist quantifiers range over actual/present objects; in particular, ‘\(\exists x Fx\)’ means “some actual/present thing is \(F\)”.

The actualist viewpoint divides the world simply into the actual and the non-actual. The non-actual includes Aristotle (and all other dead persons), Santa Claus (and all other fictional objects), and the round square (and all other impossible objects).

Possibilism divides the world differently. First, it divides the world into the possible and the non-possible. Second, it divides the world into the actual and the non-actual. Note, of course, that the latter division is indexical; what is actual depends upon when/where you are speaking. Technically speaking, for any given index \(i\), there are the things that exist-at-\(i\). These are overlapping categories: every actual thing is possible.

Next, the possibilist viewpoint maintains that quantifiers range over possible objects; ‘\(\exists x Fx\)’ means “some possible thing is \(F\)”. The temporal version of possibilism claims that ‘\(\exists x Fx\)’ means ‘some past, present, or future thing is \(F\)’.

Since the possibilist viewpoint distinguishes between (possible) existence and actuality, it must accordingly distinguish between the sentences ‘some (possible) thing is \(F\)’ and ‘some actual (present) thing is \(F\)’. This is accomplished by introducing an additional logical predicate, called actual existence, which we propose to symbolize by the ligature ‘\(\mathcal{E}\)’. This allows us the following translations.

\[
\exists x Fx \quad \text{some possible thing is } F \\
\exists x (\mathcal{E} x \land Fx) \quad \text{some actual thing is } F
\]

The indexical (world-theoretic) renderings of these formulas are as follows.

\[
[\exists x Fx/i] \\
\equiv \exists x [Fx/i] \\
\equiv \text{there is a possible thing, which is } F\text{-at-}i \\
[\exists x (\mathcal{E} x \land Fx)/i] \\
\equiv \exists x \{[\mathcal{E} x/i] \land [Fx/i]\} \\
\equiv \text{there is a possible thing, which is actual-at-}i, \text{and which is } F\text{-at-}i
\]

---

\(^{18}\) For example, the derivative function is defined as a limit of interval-change functions.

\(^{19}\) Let us ignore the very tricky idea of re-birth.

\(^{20}\) Note: the French cognate ‘actuel’ translates as ‘present’.
3. Reducing Actualist Modal Logic to Possibilist Modal Logic

The logical principles of actualist modal logic can be simulated, or mimicked, within possibilist modal logic by introducing the following definitions, which are the possibilist versions of actualist quantifiers.

\[(d1) \quad \forall' \forall \Phi = \forall \forall(\mathcal{E}[\nu] \to \Phi)\]
\[(d2) \quad \exists' \forall \Phi = \exists \forall(\mathcal{E}[\nu] & \Phi)\]

In the following, the rules of Actualist Quantified Modal Logic are listed in one column, and in the other column are listed the corresponding argument forms as rendered in Possibilist Quantified Modal Logic. Note, that the indices are the same for every line, and are accordingly dropped.

<table>
<thead>
<tr>
<th>Actualist Rule</th>
<th>Possibilist Counterpart</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\forall' \mathcal{O})</td>
<td>(\forall\forall(\mathcal{E}[\nu] \to \Phi))</td>
</tr>
<tr>
<td>(\forall' \nu \Phi)</td>
<td>(\forall\forall(\mathcal{E}[\nu] \to \Phi))</td>
</tr>
<tr>
<td>(\mathcal{E}[c])</td>
<td>(\mathcal{E}[c])</td>
</tr>
<tr>
<td>(\Phi[c/\nu])</td>
<td>(\Phi[c/\nu])</td>
</tr>
<tr>
<td>(\exists' \mathcal{O})</td>
<td>(\exists\forall(\mathcal{E}[\nu] &amp; \Phi))</td>
</tr>
<tr>
<td>(\exists' \nu \Phi)</td>
<td>(\exists\forall(\mathcal{E}[\nu] &amp; \Phi))</td>
</tr>
<tr>
<td>(\mathcal{E}[n] &amp; \Phi[n/\nu])</td>
<td>(\mathcal{E}[n] &amp; \Phi[n/\nu])</td>
</tr>
<tr>
<td>(\exists' \mathcal{I})</td>
<td>(\exists\forall(\mathcal{E}[\nu] &amp; \Phi))</td>
</tr>
<tr>
<td>(\exists' \nu \Phi)</td>
<td>(\exists\forall(\mathcal{E}[\nu] &amp; \Phi))</td>
</tr>
<tr>
<td>(\mathcal{E}[n])</td>
<td>(\mathcal{E}[n])</td>
</tr>
<tr>
<td>(\Phi[c/\nu])</td>
<td>(\Phi[c/\nu])</td>
</tr>
<tr>
<td>(\exists' \nu \Phi)</td>
<td>(\exists\forall(\mathcal{E}[\nu] &amp; \Phi))</td>
</tr>
<tr>
<td>(\mathcal{E}[n] &amp; \Phi[n/\nu])</td>
<td>(\mathcal{E}[n] &amp; \Phi[n/\nu])</td>
</tr>
</tbody>
</table>

Here, \(\Phi\) is any formula, \(\nu\) is any variable, \(c\) is any constant, and \(n\) is any new constant. Also, where \(\varepsilon\) is any expression, \(\Phi[\varepsilon/\nu]\) is the formula that results when \(\varepsilon\) replaces every occurrence of \(\nu\) that is free in \(\Phi\).

Notice that \(\exists' \mathcal{O}\) is simply a special case of \(\exists \mathcal{O}\), and \(\exists' \mathcal{I}\) is a special case of \(\exists \mathcal{I}\). They are nevertheless derivable rules, the proofs of which are left as an exercise.

4. Rules for Modal Predicate Logic with Actual Existence

To obtain system MPLA(\(\sigma\)), one takes the rules of MPL(\(\sigma\)) and adds the following.

<table>
<thead>
<tr>
<th>Def (\forall')</th>
<th>Def (\exists')</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\forall' \forall \Phi)</td>
<td>(\exists' \forall \Phi)</td>
</tr>
<tr>
<td>(\forall \forall(\mathcal{E}[\nu] \to \Phi))</td>
<td>(\exists \forall(\mathcal{E}[\nu] &amp; \Phi))</td>
</tr>
</tbody>
</table>

Note: As usual, \(\text{‘} = \text{’}\) indicates a dual-direction rule, which can be used as an in-rule, an out-rule, and a show-rule.\(^{21}\)

\(^{21}\) Later, when we consider general first-order modal logic, we will add two more rules pertaining to the \(\mathcal{E}\) predicate.
D. Exercises

1. Simple Modal Predicate Logic

Directions: for each of the following argument forms, construct a formal derivation of the conclusion from the premises (if any) in MPL(\(K\)). In cases in which two formulas are separated by ‘//’, derive each formula from the other.

1. \(\forall x \Box Fx \leftrightarrow \Box \forall x Fx\)
2. \(\exists x \Diamond Fx \leftrightarrow \Diamond \exists x Fx\)
3. \(\exists x \Box Fx \rightarrow \Box \exists x Fx\)
4. \(\Diamond \forall x Fx \rightarrow \forall x \Diamond Fx\)
5. \(\forall x \Box(Fx \rightarrow Gx) ; \Box \forall x Fx / \Box \forall x Gx\)
6. \(\Box \forall x(Fx \rightarrow Gx) ; \forall x \Box Fx / \forall x \Box Gx\)
7. \(\forall x \Box(Fx \rightarrow Gx) ; \Diamond \forall x Fx / \Diamond \forall x Gx\)
8. \(\Box \forall x(Fx \rightarrow Gx) ; \exists x \Box Fx / \exists x \Box Gx\)
9. \(\forall x \Box(Fx \rightarrow Gx) ; \Box \exists x Fx / \Box \exists x Gx\)
10. \(\Diamond \forall x(Fx \rightarrow Gx) ; \exists x \Box Fx / \Diamond \exists x Gx\)
11. \(\Diamond \forall x(Fx \rightarrow Gx) ; \Box \exists x Fx / \Diamond \exists x Gx\)
12. \(\Box \forall x(Fx \rightarrow P) ; \Box \exists x Fx / \Box P\)
13. \(\exists x \Box(Fx \rightarrow P) ; \exists x \Box Fx / \Box P\)
14. \(\Diamond Q ; \Box \exists x(Fx \rightarrow P) ; \Box \forall x Fx / \Box P\)
15. \(\forall x \Box(Fx \rightarrow Gx) / \Box(\forall x Fx \rightarrow \forall x Gx)\)
16. \(\forall x \Box(Fx \rightarrow Gx) / \Box(\exists x Fx \rightarrow \exists x Gx)\)
17. \(\forall x(\Box Fx \lor \Box Gx) / \Box \forall x(Fx \lor Gx)\)
18. \(\Diamond \forall x Fx ; \Diamond \forall x Gx / \forall x(\Diamond Fx & \Diamond Gx)\)
19. \(\exists x(\Box Fx \lor \Box Gx) / \Box \exists x Fx \lor \Box \exists x Gx\)
20. \(\Diamond(\exists x Fx \& \exists x Gx) / \exists x \Diamond Fx \& \exists x \Diamond Gx\)
21. \(\exists x \Diamond(\exists x Fx \& \exists x Gx) / \Diamond(\exists x Fx \& \exists x Gx)\)
22. \(\Diamond \forall x(Fx \rightarrow Gx) ; \Box \exists x Fx / \Diamond \exists x(Fx \& Gx)\)
23. \(\Diamond \exists x Fx ; \Box \forall x(Fx \rightarrow Gx) / \Diamond \exists x(Fx \& Gx)\)
24. \(\Diamond \forall x(Fx \rightarrow Gx) ; \exists x \Box(Fx \& \Diamond Hx) / \Diamond \exists x \Diamond(Gx \& Hx)\)
25. \(\exists x \Box(Fx \rightarrow Gx) ; \Diamond \forall x Fx / \exists x \Diamond Gx\)
26. \(\forall x(\Diamond Fx \rightarrow \Box Gx) ; \Diamond \sim \exists x Gx / \Box \sim \exists x Fx\)
27. \(\forall x \forall y \Box Rxy \leftrightarrow \Box \forall x \forall y Rxy\)
28. \(\exists x \exists y \Diamond Rxy \leftrightarrow \Diamond \exists x \exists y Rxy\)
29. \(\Box \exists x \forall y Rxy / \forall x \Box \exists y Rxy\)
30. \(\exists x \Box \forall y Rxy / \forall x \exists y \Box Rxy\)
31. \(\exists x \exists y \Box Rxy / \Box \exists x \exists y Rxy\)
32. \(\forall x \exists y \Box Rxy / \Box \forall x \exists y Rxy\)
33. \(\exists x(\Box Fx \& \Diamond \forall y(Gy \rightarrow Rxy)) / \forall x(\Box Gx \rightarrow \exists y(\Box Fy \& \Diamond Ryx))\)
2. **Actualist Modal Predicate Logic**

1. **Comparing Actualist MPL and Simple (Possibilist) MPL**

Consider the derivation problems in Section 1. Consider replacing every quantifier expression by its actualist counterpart [replace \( \forall \) by \( \forall' \) and \( \exists \) by \( \exists' \)].

(1) Using System L as the SL underpinning, which of the arguments remain valid? Construct derivations or counterexamples as is appropriate.

(2) Using System K as the SL underpinning, which of the arguments remain valid? Construct derivations or counterexamples as is appropriate.

---

**Directions for Parts 2–4:** for each of the following argument forms, construct a formal derivation of the conclusion from the premises (if any) in the above system, using the sentential rules listed in parentheses.

2. **Pure (Actualist) Quantifiers (system K)**

   1. \( \forall' x (Fx \rightarrow \Box Ex) ; \exists' x \Diamond Fx / \Diamond \exists' x Fx \)
   2. \( \forall' x \Box Ex ; \exists' x \Diamond Fx / \Diamond \exists' x Fx \)
   3. \( \forall' x (Fx \rightarrow \Box Ex) ; \exists' x \Box Fx / \Box \exists' x Fx \)
   4. \( \forall' x \Box Ex ; \exists' x Fx / \Box \exists' x Fx \)
   5. \( \forall' x \Box Ex ; \Box \forall' x Fx / \forall' x \Box Fx \)

3. **Pure (Actualist) Quantifiers (system KB)**

   6. \( \Box \forall' x \Box Ex ; \Diamond \exists' x Fx / \exists' x \Diamond Fx \)
   7. \( \Box \forall' x (Fx \rightarrow \Box Ex) ; \Diamond \exists' x Fx / \exists' x \Diamond Fx \)
   8. \( \Box \forall' x \Box Ex ; \forall' x Fx / \Box \forall' x Fx \)
   9. \( \Box \forall' x \Box Ex ; \exists' x Fx / \exists' x \Diamond Fx \)
   10. \( \Box \forall' x \Box Ex ; \forall' x Fx / \Box \forall' x Fx \)

4. **Mixed (Possibilist and Actualist) Quantifiers (system K)**

   11. \( \Box \forall x (Fx \rightarrow \Box Ex) ; \exists' x \Diamond Fx / \Diamond \exists' x Fx \)
   12. \( \Box \forall x (Fx \rightarrow \Box Ex) ; \exists' x \Box Fx / \Box \exists' x Fx \)
   13. \( \forall x (\Diamond Fx \rightarrow \Box Ex) ; \exists' x Fx / \exists' x \Diamond Fx \)
   14. \( \forall x (\Diamond Ex \rightarrow \Box Ex) ; \exists' x Fx / \exists' x \Diamond Fx \)
   15. \( \forall x (\Diamond Ex \rightarrow \Box Ex) ; \forall' x \Box Fx / \Box \forall' x Fx \)
   16. \( \forall x (\Diamond Ex \rightarrow \Box Fx) / \Box \forall' x Fx \)
3. Answers to Selected Exercises

1. Simple Modal Predicate Logic

#5

(1) \( \forall x \Box (Fx \rightarrow Gx) \) /0 Pr
(2) \( \Box \forall x Fx \) /0 Pr
(3) SHOW: \( \Box \forall x Gx \) /0 ND
(4) SHOW: \( \forall x Gx \) /01 UD
(5) SHOW: \( Ga \) /01 7,9,SL
(6) \( \Box (Fa \rightarrow Ga) \) /0 1,\( \forall O \)
(7) \( Fa \rightarrow Ga \) /01 6,\( \Box O \)
(8) \( \forall x Fx \) /01 2,\( \Box O \)
(9) \( Fa \) /01 8,\( \forall O \)

#8

(1) \( \Box \forall x (Fx \rightarrow Gx) \) /0 Pr
(2) \( \exists x \Box Fx \) /0 Pr
(3) SHOW: \( \exists x \Box Gx \) /0 5,\( \exists I \)
(4) \( \Box Fa \) /0 2,\( \exists O \)
(5) SHOW: \( \Box Ga \) /0 ND
(6) SHOW: \( Ga \) /01 DD
(7) \( \forall x (Fx \rightarrow Gx) \) /01 1,\( \Box O(k) \)
(8) \( Fa \) /01 4,\( \Box O(k) \)
(9) \( Ga \) /01 7,8,QL

#14

(1) \( \Diamond Q \) /0 Pr
(2) \( \Box \exists x (Fx \rightarrow P) \) /0 Pr
(3) \( \Box \forall x Fx \) /0 Pr
(4) SHOW: \( \Diamond P \) /0 DD
(5) \( Q \) /01 1,\( \Diamond O \)
(6) \( \exists x (Fx \rightarrow P) \) /01 2,\( \Box O(k) \)
(7) \( \forall x Fx \) /01 3,\( \Box O(k) \)
(8) \( Fa \rightarrow P \) /01 6,\( \exists O \)
(9) \( Fa \) /01 7,\( \forall O \)
(10) \( P \) /01 8,9,SL
(11) \( \Diamond P \) /0 10,\( \Diamond I(k) \)

#19

(1) \( \exists x (\Box Fx \vee \Box Gx) \) /0 Pr
(2) SHOW: \( \Box \exists x Fx \vee \Box \exists x Gx \) /0 \( \lor ID \)
(3) \( \sim \exists x Fx \vee \sim \exists x Gx \) /0 As
(4) SHOW: \( \neg \) /\neg 7,8-15, SC
(5) \( \sim \exists x Fx \) /01 3a,\( \sim \Box O \)
(6) \( \sim \exists Gx \) /02 3b,\( \sim \Box O \)
(7) \( \Box Fa \lor \Box Ga \) /0 1,\( \exists O \)
(8) c1: \( \Box Fa \) /0 As
(9) \( Fa \) /01 8,\( \Box O(k) \)
(10) \( \sim Fa \) /01 5,\( \sim \exists O \)
(11) \( \neg \) /\neg 9,10
(12) c2: \( \Box Ga \) /0 As
(13) \( Ga \) /02 12,\( \Box O(k) \)
(14) \( \sim Ga \) /02 6,\( \sim \exists O \)
(15) \( \neg \) /\neg 13,14,SL
#22

1. $\Diamond \forall x (F_x \rightarrow G_x)$ /0 Pr
2. $\Box \exists x F_x$ /0 Pr
3. SHOW: $\Diamond \exists x (F_x \& G_x)$ /0 DD
4. $\forall x (F_x \rightarrow G_x)$ /01 1,1O
5. $\exists x F_x$ /01 2,2O(k)
6. $Fa$ /01 5,3O
7. $Ga$ /01 4,6,QL
8. $Fa \& Ga$ /01 6,7,SL
9. $\exists x (F_x \& G_x)$ /01 8,1I
10. $\Diamond \exists x (F_x \& G_x)$ /0 9,1I(k)

#24

1. $\Diamond \forall x (F_x \rightarrow \Box G_x)$ /0 Pr
2. $\exists x \Box (F_x \& \Diamond H_x)$ /0 Pr
3. SHOW: $\Diamond \exists x \Diamond (G_x \& H_x)$ /0 ID
4. $\sim \Diamond \exists x \Diamond (G_x \& H_x)$ /0 As
5. SHOW: $\bot$ /1 12-13,SL
6. $\forall x (F_x \rightarrow \Box G_x)$ /01 1,1O
7. $\Box (Fa \& \Diamond Ha)$ /0 2,3O
8. $\sim \exists x \Diamond (G_x \& H_x)$ /01 4,\sim \Diamond O(k)
9. $\sim \Diamond (Ga \& Ha)$ /01 8,QL
10. $Fa \& \Diamond Ha$ /01 7,\Box O(k)
11. $\Diamond Ga$ /01 4,10a,QL
12. $Ha$ /01 10b,3O
13. $Ga$ /01 11,\Box O(k)
14. $\sim (Ga \& Ha)$ /01 9,\sim \Diamond O(k)

#26

1. $\forall x (\Diamond F_x \rightarrow \Box G_x)$ /0 Pr
2. $\Diamond \sim \exists x G_x$ /0 Pr
3. SHOW: $\Box \sim \exists x F_x$ /0 ND
4. SHOW: $\sim \exists x F_x$ /01 ID
5. $\exists x F_x$ /01 As
6. SHOW: $\bot$ /1 11,12,SL
7. $Fa$ /01 5,3O
8. $\Diamond Fa$ /0 7,\Diamond I(k)
9. $\Box Ga$ /0 1,8,QL
10. $\sim \exists x G_x$ /02 2,\Box O
11. $\sim Ga$ /02 10,QL
12. $Ga$ /02 9,\sim \Diamond O(k)

#30

1. $\exists x \Box \forall y R_{xy}$ /0 Pr
2. SHOW: $\forall x \exists y \Box R_{xy}$ /0 UD
3. SHOW: $\exists y \Box R_{ya}$ /0 8,QL
4. $\Box \forall y R_{by}$ /0 1,3O
5. SHOW: $\Box R_{ba}$ /0 ND
6. SHOW: $R_{ba}$ /01 DD
7. $\forall y R_{by}$ /01 4,\Box O(k)
8. $R_{ba}$ /01 7,\forall O
2. Actualist Modal Predicate Logic

(From Part 3.1)

#1a

(1) \( \forall x \Box Fx \) /0 Pr
(2) SHOW: \( \Box \forall x Fx \) /0 ND
(3) SHOW: \( \forall x Fx \) /0 Def \( \forall' \)
(4) SHOW: \( \forall x (\forall x Fx \rightarrow Fx) \) /0 UCD
(5) \( \forall x (\forall x Fx \rightarrow Fx) \) /0 As
(6) SHOW: \( \forall x Fx \) /0 ID
(7) \( \neg Fa \) /0 As
(8) SHOW: \( \forall x (\forall x Fx \rightarrow Fx) \) /0 ?
(9) \( \forall x (\forall x Fx \rightarrow Fx) \) /0 1,\( \forall' \)
(10) \( \forall x (\forall x Fx \rightarrow Fx) \) /0 7,\( \forall O \)
(11) \( \neg Fa \) /0 7,\( \forall O(\neg)(k) \)
(12) \( \neg Fa \) /0 10,11,SL
(13) (5) and (12) are not contradictory!

#1b

(1) \( \Box \forall x Fx \) /0 Pr
(2) SHOW: \( \forall x \Box Fx \) /0 Def \( \forall' \)
(3) SHOW: \( \forall x (\forall x Fx \rightarrow Fx) \) /0 UCD
(4) \( \forall x (\forall x Fx \rightarrow Fx) \) /0 As
(5) SHOW: \( \Box Fa \) /0 ND
(6) SHOW: \( Fa \) /0 ID
(7) \( \sim Fa \) /0 As
(8) SHOW: \( \forall x (\forall x Fx \rightarrow Fx) \) /0 ?
(9) \( \forall x (\forall x Fx \rightarrow Fx) \) /0 1,\( \Box O(k) \)
(10) \( \forall x (\forall x Fx \rightarrow Fx) \) /0 9,\( \Box O(\neg)(k) \)
(11) \( \forall x (\forall x Fx \rightarrow Fx) \) /0 10,\( \forall O \)
(12) \( \sim \forall x (\forall x Fx \rightarrow Fx) \) /0 7,11,SL
(13) (4) and (12) are not contradictory!
#2a

(1) $\exists' x \Diamond Fx$ /0 Pr
(2) SHOW: $\Diamond \exists' x Fx$ /0 ID
(3) $\sim \Diamond \exists' x Fx$ /0 As
(4) SHOW: $\star$ /star $?$
(5) $\exists x (\neg Fx \& \Diamond Fx)$ /0 Def $\exists'$
(6) $\neg Fx \& \Diamond Fx$ /0 $5, \exists O$
(7) $Fa$ /0 6b, $\Diamond O$
(8) $\exists' x Fx$ /0 3, $\sim \Diamond O(k)$
(9) $\exists x Fx \& \neg Fx$ /0 8, Def $\exists'(-)$
(10) $\neg (Fa \& \neg Fx)$ /0 9, $\sim \exists O$
(11) $\neg \exists Fx$ /0 7, 9, 10, SL
(12) (6a) and (11) are not contradictory!

#2b

(1) $\Diamond \exists' x Fx$ /0 Pr
(2) SHOW: $\exists' x \Diamond Fx$ /0 Def $\exists'$
(3) SHOW: $\exists x (\neg Fx \& \Diamond Fx)$ /0 ID
(4) $\neg \exists x (\neg Fx \& \Diamond Fx)$ /0 As
(5) SHOW: $\star$ /star $?$
(6) $\exists' x Fx$ /0 1, $\Diamond O$
(7) $\exists x (\neg Fx \& \Diamond Fx)$ /0 6, Def $\exists'$
(8) $\neg Fx \& \Diamond Fx$ /0 7, $\exists O$
(9) $\neg (Fa \& \Diamond Fx)$ /0 4, $\sim \exists O$
(10) $\Diamond Fx$ /0 8b, $\Diamond I(k)$
(11) $\neg \exists Fx$ /0 9, 10, SL
(12) (8a) and (11) are not contradictory!

#3

(1) $\exists' x \Box Fx$ /0 Pr
(2) SHOW: $\Box \exists' x Fx$ /0 ND
(3) SHOW: $\exists' x Fx$ /0 Def $\exists'$
(4) SHOW: $\exists x (\neg Fx \& \Box Fx)$ /0 ID
(5) $\neg \exists x (\neg Fx \& \Box Fx)$ /0 As
(6) SHOW: $\star$ /star $?$
(7) $\exists x (\neg Fx \& \Box Fx)$ /0 1, Def $\exists'$
(8) $\neg Fx \& \Box Fx$ /0 7, $\exists O$
(9) $Fa$ /0 8b, $\Box O(k)$
(10) $\neg (Fa \& \Box Fx)$ /0 5, $\sim \exists O$
(11) $\neg \exists Fx$ /0 9, 10, SL
(12) (8a) and (11) are not contradictory!

#4

(1) $\Diamond \neg' x Fx$ /0 As
(2) SHOW: $\neg' x \Diamond Fx$ /0 Def $\forall'$
(3) SHOW: $\forall x (\neg Fx \rightarrow \Diamond Fx)$ /0 UCD
(4) $\exists Fx$ /0 As
(5) SHOW: $\Diamond Fx$ /0 ID
(6) $\sim \Diamond Fx$ /0 As
(7) SHOW: $\star$ /star $?$
(8) $\forall' x Fx$ /0 1, $\Diamond O$
(9) $\forall x (\neg Fx \rightarrow Fx)$ /0 8, Def $\forall'$
(10) $\sim Fx$ /0 6, $\sim \Diamond O(k)$
(11) $\exists Fx \rightarrow Fx$ /0 9, $\forall O$
(12) $\sim \exists Fx$ /0 10, 11, SL
(13) (4) and (12) are not contradictory!
(From Parts 3.2–3.4)

#1

1. \( \forall x \Box (Fx \to \forall x) \) /0 Pr
2. \( \exists x \Diamond Fx \) /0 Pr
3. SHOW: \( \Diamond \exists x Fx \) /0 DD
4. \( \forall x (\forall x \to \Box (Fx \to \forall x)) \) /0 1, Def '∀'
5. \( \exists x (\forall x \& \Diamond Fx) \) /0 2, Def '∃'
6. \( \forall F \& \Diamond Fa \) /0 5, ∃O
7. \( \Box (Fa \to \forall F) \) /0 4, 6a, QL
8. \( Fa \) /0 1 6b, ∆O
9. \( Fa \to \forall F \) /0 7, ∆O(k)
10. \( \forall F \& Fa \) /0 1 8, 9, SL
11. \( \exists x (\forall x \& Fx) \) /0 10, ∃I
12. \( \exists x Fx \) /0 11, Def '∃'
13. \( \Box \exists x Fx \) /0 12, ∆I(k)

#15

1. \( \forall x (\Diamond \forall x \to \forall x) \) /0 Pr
2. \( \forall x \Box Fx \) /0 Pr
3. SHOW: \( \Box \forall x Fx \) /0 ND
4. SHOW: \( \forall x Fx \) /0 1 Def '∀'
5. SHOW: \( \forall x (\forall x \to Fx) \) /0 1 UCD
6. \( \forall F \) /0 1 As
7. SHOW: \( Fa \) /0 1 DD
8. \( \Diamond \forall F \to \forall F \) /0 1, ∆O
9. \( \Diamond \forall F \) /0 6, ∆I(k)
10. \( \forall F \) /0 8, 9, SL
11. \( \forall x (\forall F \to \Box Fx) \) /0 2, Def '∀'
12. \( \Box Fa \) /0 10, 11, QL
13. \( Fa \) /0 12, ∆O(k)