Module 4
Basic Derivation Rules for Modal Logic

G. J. Mattey
July 8, 2010

Contents

1 Derivation Rules for the ‘□’ Operator 2
   1.1 Strict Reiteration for □ ................................. 2
   1.2 □ Introduction ............................................. 4

2 Derivation Rules for ‘♦’ Operator 7
   2.1 Impossibility Rules .......................................... 8
      2.1.1 Strict Reiteration for ∼♦ ............................... 8
      2.1.2 ∼♦ Introduction ......................................... 9
   2.2 Possibility Rules ........................................... 11
      2.2.1 Strict Reiteration for ♦ ................................. 12
      2.2.2 ♦ Elimination .......................................... 13

3 Derivation Rules for the ‘⇒’ Operator 16
   3.1 Strict Reiteration for ⇒ ..................................... 17
   3.2 ⇒ Introduction ............................................. 17

4 Interaction of the Rules 18

5 Conclusion 19
The first derivation system for Modal Sentential Logic in the Fitch style was given by Fitch himself. The central idea was to allow a kind of construction in a derivation, the *restricted sub-derivation*, which is not a feature of his derivation system for non-modal sentential logic. The restriction is on the use of Reiteration. Only certain sentences (or their components) are allowed to be reiterated into a restricted sub-derivation. This restrictive form of Reiteration is embodied in new modal reiteration rules. In this module, we will focus on several versions of such a rule. We will also look at some rules for introducing and eliminating modal operators—rules which work in tandem with the restricted reiteration rules. The goal of formulating the derivation systems employing these rules is to mirror the semantical results which were the subject of the last module.

1 Derivation Rules for the ‘□’ Operator

We shall begin our treatment of the derivation rules with the ‘□’ operator. The semantical rule governing this operator **SR-□**, implies two conditionals:

- If \( v_I(□\alpha, w) = T \), then \( v_I(\alpha, w_i) = T \) at all worlds \( w_i \) in \( I \) such that \( Rww_i \).
- If \( v_I(\alpha, w_i) = T \) at all worlds \( w_i \) in \( I \) such that \( Rww_i \), then \( v_I(□\alpha, w) = T \).

Each of these consequences of **SR-□** is the basis for a derivational rule in modal logic.

1.1 Strict Reiteration for □

When we use the first rule to reason about what follows from the truth of □\( \alpha \) at a world \( w \), we assume that \( w_1 \) is an arbitrary world accessible to \( w \) and note that \( \alpha \) is assigned true there. Since this holds for all arbitrary accessible worlds, an asterisk is placed above the arrow.

\[
\begin{array}{c}
\text{w} \\
□\alpha \\
T \\
\end{array} \rightarrow^* \begin{array}{c}
w_1 \\
\alpha \\
T \\
\end{array}
\]

Our modal derivations will employ *restricted scope lines* to represent arbitrary accessible possible worlds. A restricted scope line is a vertical line flanked at the top left by a ‘□,’ which graphically represents the relation of accessibility from a given world. We take the (implicitly assumed) truth of sentences outside the restricted scope line to be truth at a world, and truth inside the scope line to be truth at a world accessible to that world.
Where □α occurs in a derivation, we will allow the importation of α across a restricted scope line which is in the immediate scope of the scope line in which □α occurs.¹ This rule is known as ‘Strict Reiteration for ‘□’ or ‘SR-□’.² (This rule should not be confused with the semantical rule SR-□, whose name uses bold-face letters.)

**Strict Reiteration for ‘□’**

<table>
<thead>
<tr>
<th>□α</th>
<th>Already Derived</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>α</td>
<td>SR-□</td>
</tr>
</tbody>
</table>

Provided that α is strictly reiterated across exactly one restricted scope line.

The rules for the non-modal operators work in the usual way within a restricted scope line. This reflects the fact that the semantical rules for those operators function in the normal way at any possible world.

It is crucial that importation of earlier steps across restricted scope lines be limited to what is permitted by the rule of Strict Reiteration, and that the non-modal rule of Reiteration cannot be used to bring a sentence across a restricted scope line. What is outside the restricted scope line reflects a truth-value at a world, and that value may be different from the one represented inside the restricted scope line. Suppose a sentence α has the value T at a world w, and that Rww₁. We cannot in general expect that the value of α at w will be preserved at w₁. If α is a sentence letter such as ‘A,’ it may well be true at w but false at w₁.

\[
\begin{array}{c|c|c|c|c}
A & T & F \\
\hline
w & \rightarrow & w₁ & A & A
\end{array}
\]

Since each step in a derivation is supposed to represent the assignment of truth to a sentence, we might get the truth-value of ‘A’ wrong by writing ‘A’ to the right of a restricted scope line when it occurs to the left. Previously, we defined allowed Reiteration only across non-modal assumptions. Since restricted scope lines differ from scope lines indicating non-modal assumptions, there is no provision for Reiteration to be used across them, and therefore it may not be so used.

Just as in non-modal SD, a derivation may not end until all assumptions have been discharged, no modal derivation may end until the restricted scope line has been terminated. A restricted scope line is intended to represent what is true at an accessible possible world. But in our semantical reasoning, the goal of having information about the truth-value at the accessible world is to discover the value of a sentence at the “home” world—the world to which the accessible world is accessible. So just as we do not terminate our semantical reasoning when we establish a truth-value at an accessible world, we do not end our derivations with continuing restricted scope lines.

¹Note that only one restricted scope line may be crossed. The semantical rule allows only truth at all worlds accessible to the given world.
²The rule might more intuitively be thought of as a □ Elimination rule, but in keeping with common practice, we will reserve that denomination for another rule to be introduced in a subsequent module.
1.2 □ Introduction

At this point, we will model a second derivational rule after the second of the two consequences of SR-□ noted above:

If \( v_I(\alpha, w_i) = T \) at all worlds \( w_i \) in \( I \) such that \( Rww_i \), then \( v_I(\square \alpha, w) = T \).

This rule allows us to reason from the value \( T \) for \( \alpha \) in an arbitrary accessible world \( w_i \) to the value \( T \) of \( \alpha \) in a world \( w \) to which \( w_i \) is accessible.

We will begin our discussion with an example. Suppose we wish to prove that if \( \square (A \land B) \) is true at an arbitrary world \( w \), then \( \square A \) is true at \( w \). We would begin our reasoning using either a truth-table or the semantical rule SR-□ and consider what would have to be the case in an arbitrary world \( w_1 \) accessible to \( w \).

\[
\begin{array}{ccc}
 w & \rightarrow^* & w_1 \\
 \square (A \land B) & \quad & A \land B \\
 T & \quad & T \\
 & \quad & A \\
 & \quad & T \\
\end{array}
\]

1 \( v_I(\square (A \land B), w) = T \) Assumption
2 \( Rww_1 \) Assumption
3 \( v_I(A \land B, w_1) = T \) 1 2 SR-□
4 \( v_I(A, w_1) = T \) 3 SR-∧

This start to our reasoning would be reflected in a derivation using SR-□:

**Partial derivation of \( \square A \) from \( \square (A \land B) \)**

\[
\begin{array}{ccc}
 1 & \square (A \land B) & \text{Assumption} \\
 2 & \square A & 1 \text{ SR-□} \\
 3 & A & 2 \land E \\
\end{array}
\]

Now we continue our semantical reasoning. Since the choice of \( w_i \) is arbitrary, \( A \) is true at all accessible worlds, in which case \( \square A \) is true in the home world.
To return to our incomplete derivation, given the intended interpretation of the restricted scope line, what we are representing is the truth of ‘A’ at an arbitrary accessible world. If ‘A’ is true at an arbitrary accessible world, it is true at all accessible worlds. This means that ‘□A’ should be true at the home world, as SR-□ requires.

**Rule needed for the derivation of □A from □(A ∧ B)**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>□(A ∧ B)</td>
<td>Assumption</td>
</tr>
<tr>
<td>2</td>
<td>□A</td>
<td>1 SR-□</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>2 ∧ E</td>
</tr>
<tr>
<td>4</td>
<td>□A</td>
<td>Rule Needed Here</td>
</tr>
</tbody>
</table>

We shall generalize this reasoning and state a rule which allows us to end a restricted scope line and prefix a ‘□’ to the last step not in the scope of any assumption within the restricted scope line. This is the rule ‘□ Introduction,’ or ‘□ I.’ It holds for all MSL derivation systems which conform to the semantics given in the last section.4

---

4We will not here give a □ Elimination rule. One such rule would allow the removal of the ‘□’ operator from □α, so that α can be written down. But nothing about the semantical rule for the ‘□’ requires that if \( v_1(□, w) = T \), then \( v_1(α, w) = T \). Such a rule will be forthcoming when we consider semantical systems strong enough to support it.

4These systems are called “normal” systems of Modal Sentential Logic. The way in which derivational systems conform to semantical systems will be explained in the context of system \( K \), which we will examine in the next module.
□ Introduction

\[
\begin{array}{c}
\vdash \\
\vdash \\
\vdash \\
\vdash \\
\alpha \\
\hline
\square \alpha & \Box I
\end{array}
\]

Provided that \(\alpha\) is not to the right of any scope line.

In annotating the use of the rule, we will indicate the whole series of steps within the restricted scope line.

**Derivation of □A from □(A ∧ B)**

1. □(A ∧ B) Assumption
2. □ A ∧ B 1 SR-□
3. A 2 ∧ E
4. □ A 1 2-3 □ I

Strict Reiteration is not required for the use of □ introduction, since we may establish results entirely within the restricted scope line. For example, we can derive ‘□ ~(A ∧ ~A)’ using some rules of SD and the rule of □ Introduction.

**Derivation of: □ ~(A ∧ ~A)**

1. □ A ∧ ~A Assumption
2. A 1 ∧ E
3. ~A 1 ∧ E
4. ~(A ∧ ~A) 1-3 ~ I
5. □ ~(A ∧ ~A) 1-4 □ I

Note that □~ \bot can easily be derived in a similar way.

Here is another example of how the reasoning in a modal derivation parallels semantical reasoning.

**Derivation of □B from □A and □(A ⊃ B)**

1. □A Assumption
2. □(A ⊃ B) Assumption
3. A 1 SR-□
4. A ⊃ B 2 SR-□
5. B 3 4 ⊃ E
6. □B 3-5 □ I
Due to the close resemblance in this case between the structure of the meta-logical proof of semantical entailment and the derivation, we can think of the latter as an abbreviation of the former. The reader can verify that other derivations can be re-cast in a similar way, so that the derivation system as a whole can be thought of as providing an abbreviated version of semantical proofs.

The derivation also looks much like truth-tabular reasoning.

\[
\begin{array}{c|c|c}
\hline
w & w_1 \\
\hline
\square A & (A \supset B) \\
\hline
T & T \\
\hline
A & A \supset B \\
\hline
T & T \\
\hline
B &  \\
\hline
T &  \\
\hline
\square B &  \\
\hline
T & \\
\hline
\end{array}
\]

2 Derivation Rules for ‘◊’ Operator

We will consider two distinct sets of derivation rules for the ‘◊’ operator. The first set consists of two derived rules based on the rules for the ‘□’ operator. The second set consists of rules specifically designed for the ‘◊’ operator. The first set has the advantage of producing derivations that are easily converted into derivations in the system for the ‘□’ operator. Using the first set will allow the derivation of whatever can be derived in the ‘□’-based system. The second set appears to be incomplete relative to the ‘□’-based system, in the sense that it seems that it will not allow all the derivations possible in it.
2.1 Impossibility Rules

One way to generate a set of rules for the ‘◊’ operator is to proceed in the manner of Lewis and base the rules on the combination ‘∼◊,’ which is intended to represent impossibility. The two rules we will give will yield results equivalent to the rules for necessity.

2.1.1 Strict Reiteration for ∼◊

From the basic semantics for the ‘□’ and the ‘◊,’ we have the following result.

□∼α always has the same truth-value as ∼◊α.

Proof.

\[ v_I(□\neg α, w) = T \text{ iff} \]
for all \( w_i \) such that \( Rw_{w_i}, v_I(\neg α, w_i) = T \) (by SR-□) iff
for all \( w_i \) such that \( Rw_{w_i}, v_I(α, w_i) = F \) (by SR-∼) iff
\[ v_I(\neg α, w) = F \] (by SR-∼)
\[ v_I(∼α, w) = T \] (by SR-∼).

With this result in hand, we can re-write the rules of Strict Reiteration and □ Introduction. First, we present Strict Reiteration, with ‘□∼.’

**Strict Reiteration for a Negated Sentence**

<table>
<thead>
<tr>
<th>□∼α Already Derived</th>
<th>□</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>∼α SR-□</th>
</tr>
</thead>
</table>

Then we substitute ‘∼◊α’ for ‘□∼α.’ As with Strict Reiteration for ‘□,’ α may only be strictly reiterated across a restricted scope line which is in the immediate scope of the line where ∼α occurs.

**Strict Reiteration for ∼◊**

<table>
<thead>
<tr>
<th>∼◊α Already Derived</th>
<th>□</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>∼α</th>
</tr>
</thead>
</table>

Provided that ∼α crosses no more than one restricted scope line.

The basis of the rule can be seen from this semantical argument. If \( v_I(∼◊α, w) = T \), then \( v_I(◊α, w) = T \), so there is no accessible world \( w_i \) at which \( v_I(α, w_i) = F \). So given an accessible world \( w_i \), we must assign \( α \) the value \( F \) there, in which case the value of ∼α must be \( T \) at \( w_i \), which was to be shown.
\( v_I(\neg \alpha, w) = T \)  
Given

\( v_I(\alpha, w) = F \)  
SR \( \neg \)

\((\Pi w_i)(R w w_i \rightarrow v_I(\alpha, w_i) = F)\)  
SR \( \neg \alpha \)

\[
\begin{array}{c}
R w w_i \\
\text{Assumption}
\end{array}
\]

\[
\begin{array}{c}
R w w_i \rightarrow v_I(\alpha, w_i) = F \\
\Pi E
\end{array}
\]

\[
\begin{array}{c}
v_I(\alpha, w_i) = F \\
\rightarrow E
\end{array}
\]

\[
\begin{array}{c}
v_I(\neg \alpha, w_i) = T \\
SR \neg
\end{array}
\]

\[
\begin{array}{c}
w \rightarrow w_i \\
\sim \neg \alpha
\end{array}
\]

\[
\begin{array}{c}
T
\end{array}
\]

\[
\begin{array}{c}
\neg \alpha
\end{array}
\]

\[
\begin{array}{c}
F
\end{array}
\]

\[
\begin{array}{c}
\alpha
\end{array}
\]

\[
\begin{array}{c}
F
\end{array}
\]

\[
\begin{array}{c}
\sim \alpha
\end{array}
\]

\[
\begin{array}{c}
T
\end{array}
\]

2.1.2 \( \sim \Box \) Introduction

The same kind of reasoning can be used to produce a derived rule for the introduction of \( \sim \Box \).

\( \square \) Introduction for a Negated Sentence

\[
\begin{array}{c}
\square
\end{array}
\]

\[
\begin{array}{c}
\sim \alpha
\end{array}
\]

\[
\begin{array}{c}
\square \sim \alpha \quad \square I
\end{array}
\]

Given the semantic duality of \( \sim \Box \) and \( \square \sim \alpha \), we get:
\[ \sim \Diamond \text{ Introduction} \]

\[
\begin{array}{l}
\Box \\
\cdot \\
\cdot \\
\sim \alpha \\
\sim \Diamond \alpha \quad \sim \Diamond \Box I
\end{array}
\]

Provided that \( \sim \alpha \) is strictly reiterated across exactly one restricted scope line.

This rule is also motivated by the basic semantics. Suppose an arbitrary world \( w_1 \) is accessible to a world \( w \). If a sentence \( \sim \alpha \) is assigned \( T \) at \( w_1 \), then \( \alpha \) has the value \( F \) at that world. Since \( w_i \) is arbitrary, \( \alpha \) has the value \( F \) at all worlds accessible to \( w \), in which case \( \Diamond \alpha \) has the value \( F \) at \( w \). Thus \( \sim \Diamond \alpha \) has the value \( T \) at \( w \), which was to be shown.

\[
\begin{array}{l}
R_{w_1} \\
v_I(\sim \alpha, w_1) = T \\
v_I(\alpha, w_1) = F \\
R_{w_1} \rightarrow v_I(\alpha, w_1) = F \\
(\Pi w_i)(R_{w_1} \rightarrow v_I(\alpha, w_1) = F) \\
v_I(\Diamond \alpha, w) = F \\
v_I(\sim \Diamond \alpha, w) = T
\end{array}
\]

\[
\begin{array}{c}
w \xrightarrow{\sim \alpha} w_1 \\
\sim \alpha \\
T
\end{array}
\]

\[
\begin{array}{c}
\alpha \\
F
\end{array}
\]

\[
\begin{array}{c}
\Diamond \alpha \\
F
\end{array}
\]

\[
\begin{array}{c}
\sim \Diamond \alpha \\
T
\end{array}
\]

With these two rules we can derive the semantical duals of any necessity-sentence that can be derived using the necessity rules. For example, we have seen that from \( \Box A \) and \( \Box (A \supset B) \) we can derive \( \Box B \). This gives the same semantical results as deriving \( \sim \Diamond A \) from \( \sim \Diamond B \) and \( \sim \Diamond (A \land \sim B) \). (The second premise can also be read as \( A \rightarrow B \).)
Exercise. Explain why these two derivations give the same semantic results.

**Derivation of** $\neg \Box A$ **from** $\neg \Box B$ **and** $\neg \Box (A \land \neg B)$

1. $\neg \Box B$ Assumption
2. $\neg \Box (A \land \neg B)$ Assumption
3. $\Box \neg B$ 1 SR-$\neg \Box$
4. $\neg (A \land \neg B)$ 2 SR-$\neg \Box$
5. $A$ Assumption
6. $\neg B$ 3 Reiteration
7. $A \land \neg B$ 5 6 $\land$ I
8. $\neg (A \land \neg B)$ 4 Reiteration
9. $\neg A$ 5-8 $\neg$ I
10. $\neg \Box A$ 3-9 $\neg \Box$ I

Another derivation shows the $\neg \Box$ Introduction rule working by itself. We can derive $\neg \Box (A \land \neg A)$ from no undischarged assumptions, showing that it is a theorem of given the basic derivation system of Modal Sentential Logic.

**Derivation of** $\neg \Box (A \land \neg A)$

1. $\Box A \land \neg A$ Assumption
2. $A$ 1 $\land$ E
3. $\neg A$ 1 $\land$ E
4. $\neg (A \land \neg A)$ 1-3 $\neg$ I
5. $\neg \Box (A \land \neg A)$ 1-4 $\neg \Box$ I

It can easily be shown in similar fashion that $\neg \Box \bot$ is derivable within the basic system.

2.2 Possibility Rules

Despite their completeness relative to the necessity rules, the two impossibility rules seem less than satisfactory because they involve a non-modal operator. They are not pure possibility rules. We shall give a derived pure possibility rule shortly. This rule was given by Fitch in the original adaptation of derivability rules to modal logic.\(^5\)

\(^5\)It is the opinion of the author that no combination of pure possibility rules will be complete relative to the basic modal semantics. In particular, the theorem just demonstrated seems not to be derivable without impossibility rules of some kind.
2.2.1 Strict Reiteration for ♦

As with the ‘□’ operator, we take our cue from the semantical rule for the modal operator, in this case, SR-♦, which has two consequences.

If $v_I(\alpha, w_i) = T$ at some world $w_i$ in $I$ such that $Rw_w$, then $v_I(\Diamond \alpha, w) = T$.

If $v_I(\Diamond \alpha, w) = T$, then $v_I(\alpha, w_i) = T$ at some world $w_i$ in $I$ such that $Rww_i$.

The rules for ‘♦’ work somewhat differently from the Strict Reiteration and the Introduction rules we have provided. The clue for their structure is found in the fact that the second consequence of SR-♦ involves an existential rather than a universal quantifier in the meta-language. When we reason from the truth of $\Diamond \alpha$ at a world $w$, we can only infer that there is at least one accessible world $w_i$ at which $\alpha$ is true. Because a restricted scope line is supposed to represent an arbitrary world, it might not represent a world at which $\alpha$ is true. The only thing we can do is to assume that $\alpha$ is true at an arbitrary world and consider what would happen if this were so.

\[
\begin{array}{c}
v_I(\Diamond \alpha, w) = T & \text{Given} \\
(\sum w_i)(Rw_w_i \land v(\alpha, w_i) = T) & \text{SR-♦} \\
Rw_w_i \land v(\alpha, w_i) = T & \text{Assumption}
\end{array}
\]

To represent this kind of reasoning, we will use a new kind of restricted scope line, which contains a strict assumption. It is written like an assumption line, but with a ‘□’ to the left of it. Then the reiterated sentence $\alpha$ is written as an assumption. For the purposes of Strict Reiteration for the other modal operators, a strict scope line with a strict assumption is treated the same way as a strict scope line without one. Only the newly-minted restricted scope line may be crossed in the process of strictly reiterating $\alpha$.

**Strict Reiteration for ‘♦’**

\[
\begin{array}{c}
\Diamond \alpha & \text{Already Derived} \\
\Box \alpha & \text{SR-♢} \\
\end{array}
\]

Provided that $\alpha$ is strictly reiterated across exactly one restricted scope line.

The situation may also be represented with truth-tables. Given the truth of $\Diamond \alpha$ at a world $w$, it follows that there is a world $w_1$ accessible to $w$ where $\alpha$ is true, though it is not specified which world this is. This indeterminacy is represented by the asterisk below the arrow. Representing a world to the right of the arrow is tantamount to assuming that $w_1$ is such a world.
2.2.2 ♦ Elimination

As with the necessity operator, we would like to be able to use the information within the restricted scope line to establish something outside it. When we have the situation where $\alpha$ is assumed to be true at some world $w_i$ accessible to $w$, and some sentence $\beta$ is found to be true there as well, we can say by SR-♦ that $\Diamond \beta$ is true at $w$.

\[
\begin{array}{c}
v_I(\Diamond \alpha, w) = T \\
(\Sigma w_i)(Rww_i \land v(\alpha, w_i) = T) \\
Rww_1 \land v(\alpha, w_1) = T \\
Rww_1 \\
v(\alpha, w_1) = T \\
\vdots \\
v(\beta, w_1) = T \\
Rww_1 \land v(\beta, w_1) = T \\
(\Sigma w_i)(Rww_i \land v(\beta, w_i) = T) \\
(\Sigma w_i)(Rww_i \land v(\beta, w_i) = T) \\
v_I(\Diamond \beta, w) = T \\
\end{array}
\]

Given
SR-♦
Assumption
$\land$ E
$\land$ E
Can be proved
$\land$ I
$\Sigma$ I
$\Sigma$ E
SR-♦

To reflect this condition, we lay down a rule of ♦ Elimination.\(^6\) When the strict assumption is discharged, the last step $\beta$ (not within any other scope lines) is brought out as $\Diamond \beta$.

\(^6\)Calling this an “elimination” rule sounds odd, but it will turn out that we must reserve the Introduction rule for another purpose. You can think of the rule as eliminating the operator in the process of Strict Reiteration to an assumption. This rule is parallel to the Predicate Logic rule of $\exists$ Elimination.
Diamond Elimination

\[ \Diamond \alpha \quad \text{Already Derived} \]
\[ \square \alpha \quad \text{SR-} \Diamond \]
\[ . \]
\[ . \]
\[ . \]
\[ \beta \]
\[ \Diamond \beta \quad \Diamond \text{E} \]

**Provided** that \( \beta \) is not in the scope of any assumption within the restricted scope line.

Note that an application of SR-\( \Diamond \) always is accompanied by an application of \( \Diamond \) Elimination, and *vice-versa*. The only way to discharge a strict assumption is to use \( \Diamond \) Elimination. And the only way to use \( \Diamond \) Elimination is on the basis of a strict assumption.

The situation is represented as follows with truth-tables.

\[
\begin{array}{c|c}
\text{w} & \text{w}_1 \\
\hline
\Diamond \alpha & \text{T} \\
\hline
\alpha & \text{T} \\
\hline
\beta & \text{T} \\
\hline
\Diamond \beta & \text{T} \\
\end{array}
\]

It is always open for one to make a regular assumption, rather than an SR-\( \Diamond \) assumption inside a restricted scope line. We did so in deriving \( \square \neg (A \land \neg A) \). But we could not move from step 4 to step 5 as follows:

**Incorrect Derivation**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \square )</td>
<td>( A \land \neg A )</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>( A )</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>( \neg A )</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>( \neg (A \land \neg A) )</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>( \Diamond \neg (A \land \neg A) )</td>
</tr>
</tbody>
</table>
Even if a theorem is derived inside a restricted scope line, it could not be brought out unless SR-♦ is used as well. This is because in the basic semantics some frames will contain worlds to which there is no world accessible. We do not want to be able to derive even the possibility of α, where α is a theorem, unless we are working with the supposition that there are accessible worlds. The presence of the possibility sentence ♦α taken to be true indicates that there is an accessible world at which α is true.

To illustrate the use of the possibility rules, we will derive ‘♦B’ from ‘♦A’ and ‘□(A ⊃ B).’

**Derivation of ♦B from ♦A and □(A ⊃ B)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>♦A</td>
</tr>
<tr>
<td>2</td>
<td>□(A ⊃ B)</td>
</tr>
<tr>
<td>3</td>
<td>□A</td>
</tr>
<tr>
<td>4</td>
<td>A ⊃ B</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>♦B</td>
</tr>
</tbody>
</table>

Note that in annotating this rule, we make reference to the possibility-sentence that was strictly reiterated.

Reasoning from truth-tables would proceed in the following way. Given the truth of ♦A at w, there is at least one accessible world w₁ at which A is true. But we cannot generalize the truth-values there to all worlds accessible to w, as we could when we reasoned about an arbitrary accessible world wᵢ.

\[
\begin{array}{ccc}
\text{w} & \rightarrow & w_1 \\
♦A & □(A ⊃ B) & \\
T & T & \\
A & A ⊃ B & \\
T & T & \\
B & \\
T & \\
♦B & \\
T & \\
\end{array}
\]

The reasoning can be done using a meta-logical derivation, which illustrates why it is sound. The metalogical expression ‘w₁’ is used here as a constant term for purposes of instantiation.
Semantical proof that if ♦A and □(A ⊃ B) are true, then ♦B is true

1. v(♦A, w) = T  
   Assumption
2. v(□(A ⊃ B), w) = T  
   Assumption
3. (∑ w_i)(Rww_i \wedge v(A, w_i) = T)  
   1 SR-♦
4. Rww_1 \wedge v(A, w_1) = T  
   Assumption
5. Rww_1  
   4 \wedge E
6. v(A, w_1) = T  
   4 \wedge E
7. v(A ⊃ B, w_1) = T  
   2 SR-□
8. v(B, w_1) = T  
   6 7 SR-♦
9. Rww_1 \wedge v(B, w_1) = T  
   5 8 \wedge I
10. (∑ w_i)(Rww_i \wedge v(B, w_i) = T)  
    9 ∑ I
11. (∑ w_i)(Rww_i \wedge v(B, w_i) = T)  
    3 4-10 ∑ E
12. v(♦B, w) = T  
    11 SR-♦

The tandem-rules SR-♦ and □ Elimination can be derived from the two rules for impossibility stated in the last sub-section. That is, if we assume that we have derived ♦α and that we can derive β from α within a restricted scope line with ‘α’ as an assumption, we can derive ♦β.

Exercise: Show how the derivation works.

3 Derivation Rules for the ‘−’ Operator

As with the one-place operators, we will motivate the derivation rules for the two-place ‘−’ operator on the basis of its semantical rule. We will not do so directly, however. Because α − β is semantically equivalent to □(α ⊃ β), the derivation rules for the ‘−’ can be developed with this relation in mind.

Thus the following consequences of the rule SR-⇒ are relevant:

If v_I(α − β, w) = T, then at all worlds w_i in I such that Rww_i either v_I(α, w_i) = F or v_I(β, w_i) = T.

If at all worlds w_i in I such that Rww_i, either v_I(α, w_i) = F or v_I(β, w_i) = T, then v_I(α − β, w) = T.

It follows from the first of these consequences and the rule SR-⇒ that:

(A) If v_I(α − β, w) = T, then v_I(α ⊃ β, w_i) = T at all worlds w_i in I such that Rww_i.

From the second consequence, we have it that:

(B) If at all worlds w_i in I such that Rww_i, if v_I(α, w_i) = T, then v_I(β, w_i) = T, then v_I(α − β, w) = T.  

Because of the close relation between the derivation rules and the semantical consequences (A) and (B), no meta-logical derivations or truth-tables will be used to illustrate them.

Here the “if … then” is a meta-logical material conditional.
3.1 Strict Reiteration for \( \rightarrow \)

Given (A), we can state a rule for Strict Reiteration. If a strict conditional occurs on a line, we may open up a new restricted scope line and “strictly reiterate” the corresponding material conditional, not crossing any more scope lines that are not in the immediate scope of the line in which \( \alpha \rightarrow \beta \) occurs.

**Strict Reiteration for ‘\( \rightarrow \)’**

\[
\begin{array}{c}
\alpha \rightarrow \beta & \text{Already Derived} \\
\vdots \\
\alpha \supset \beta & \text{SR-\( \rightarrow \)}
\end{array}
\]

**Provided** that \( \alpha \supset \beta \) is strictly reiterated across exactly one restricted scope line.

3.2 \( \supset \) Introduction

The rule for \( \supset \) Introduction follows (B) and resembles the rule for \( \supset \) Introduction. If within a restricted scope line, the assumption of \( \alpha \) is made and \( \beta \) is derived in the scope of that assumption (and not in the scope of any other assumption), then the scope line of the assumption as well as the restricted scope line may be terminated, and \( \alpha \supset \beta \) written.

**\( \supset \) Introduction**

\[
\begin{array}{c}
\alpha \\
\vdots \\
\beta \\
\alpha \supset \beta & \supset \text{I}
\end{array}
\]

**Provided** that \( \beta \) is not in the scope of any assumption other than \( \alpha \) within the restricted scope line.

The motivation for the rule is as follows. Suppose an arbitrary world \( w_i \) is accessible to a world \( w \). If a sentence \( \alpha \) is assumed to have the value \( T \) at \( w_i \), and it can be shown that \( \beta \) is assigned \( T \) as well, then by (B), it follows that \( \alpha \supset \beta \) is true at \( w \), which was to be shown.

It is easy to see that the following meta-theorem holds:

If \( \{ \alpha \} \vdash_{SD} \beta \), then \( \alpha \supset \beta \) is derivable in the basic derivational system from no undischarged assumptions.

If the antecedent holds, then there is a derivation of \( \beta \) using only SD rules and with \( \alpha \) as a strict assumption. By the rule \( \supset \) Introduction, one may then write down \( \alpha \supset \beta \) outside the scope of any assumption. Here is an example.

**Derivation of:** \( (A \land B) \supset A \)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>□</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(( A \land B ) \supset A) \hspace{1cm} 1-2 \supset \text{I}</td>
</tr>
</tbody>
</table>
Another example of the use of the ∃ rules is the derivation of one of the so-called “paradoxes of strict implication.” One of the “paradoxes of material implication” is the fact that ⊢ S D α ⊃ (β ⊃ α). On Russell’s intended interpretation, this means that a true sentence α is “materially implied” by any sentence β. In Lewis’s systems, it turns out that a necessarily true sentence α is “strictly implied” by any sentence β, i.e., □α → (β → α), which can be derived using our rules.

**Derivation of:** □A → (B → A)

1. □A Assumption
2. □B Assumption
3. □A 1 SR-□
4. B → A 2-3 → I
5. □A → (B → A) 1-4 → I

**Exercise:** Prove the other “paradox of strict implication,” that a necessarily false sentence strictly implies any sentence, i.e., □¬α → (α → β) is derivable.

A parallel semantical proof illustrates the motivation for the rules we have chosen.

**Semantical proof of:** $\vDash □A \rightarrow (B \rightarrow A)$

1. Rww Assumption
2. $\nu(□A, w_i) = T$ Assumption
3. Rww Assumption
4. $\nu(B, w_j) = T$ Assumption
5. $\nu(A, w_j) = T$ 2 SR-□
6. $\nu(B, w_j) = T \rightarrow \nu(A, w_j) = T$ 4-5 → I
7. Rww Assumption
8. $\nu(B \rightarrow A), w_i = T$ 7 SR-¬¬ (B)
9. $\nu(□A, w_i) = T \rightarrow \nu(B \rightarrow A), w_i = T$ 2-8 → I
10. Rww Assumption
11. $\nu(□A \rightarrow (B \rightarrow A), w) = T$ 10 SR-¬¬ (B)

**4 Interaction of the Rules**

Four pairs of Strict Reiteration and modal operator Introduction rules have been set out above. The rules may be used in combination with one another. In this section, the permissible and impermissible uses of the rules will be discussed.

The Strict Reiteration rules for ‘□,’ ‘¬¬,’ and ‘¬’ all allow the removal of the modal operator when any single restricted scope line is crossed. Moreover, the Introduction rules for necessity and impossibility may
be employed on any restricted scope line, so long as all assumptions are discharged. The following example shows how SR-□ and SR-→ can both be used within the same restricted scope line, the derivation ending with an application of ∼◊ Introduction.

**Derivation of ∼◊B from □A, and A → B from**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>□A</td>
</tr>
<tr>
<td>2</td>
<td>A → B</td>
</tr>
<tr>
<td>3</td>
<td>□A</td>
</tr>
<tr>
<td>4</td>
<td>A ∪ B</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>∼B</td>
</tr>
<tr>
<td>7</td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>∼B</td>
</tr>
<tr>
<td>9</td>
<td>∼◊B</td>
</tr>
</tbody>
</table>

The rule of → Introduction requires that an assumption be made within a restricted scope line. Strict Reiteration for ‘□’, ‘∼’, and ‘→’ may be made in the scope of the requisite assumption. The proof of the theoremhood of □A → (B → A) above made use of this permitted move.

The rules of Strict Reiteration for ‘◊’ and of ◊ Introduction are not so free. Strict Reiteration for ‘□’, ‘∼’, and ‘→’ may be made across the scope line of a Strict Assumption, as in the above derivation of ‘◊B’ from ‘◊A’ and ‘□(A ∪ B).’ But a sentence ◊α may be strictly reiterated only in the guise of a Strict Assumption. Thus, if we had the following beginning of a derivation, we would not be able to move ‘B’ across the restricted scope line, since the rule of Strict Reiteration for ‘◊’ allows movement across only the restricted scope line drawn explicitly for its use.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>◊A</td>
</tr>
<tr>
<td></td>
<td>◊B</td>
</tr>
<tr>
<td></td>
<td>□A</td>
</tr>
</tbody>
</table>

Finally, the rule ◊ Introduction may be used only when the restricted scope line contains a Strict Assumption.

**5 Conclusion**

This completes the exposition of basic semantical and derivational rules for our study of modal sentential logic. We shall now look at the system $K$ which results from the use of the basic rules examined here. Later, we will turn to a number of other systems that result from strengthening the rules in various ways.