THE LOGIC BOOK

Sixth Edition

MERRIE BERGMANN *Smith College, Emerita*

JAMES MOOR *Dartmouth College*

JACK NELSON

Mc Graw Hill

Connect Learn Succeed™

Chapter

5

SENTENTIAL LOGIC: DERIVATIONS

In Section 5.1 we introduce the derivation system *SD* and the concept of a derivation. In Section 5.2 we introduce syntactic analogues of core logical concepts: **derivable in** *SD*, **valid in** *SD*, **theorem** in *SD*, **equivalent** in *SD*, and **inconsistent** in *SD*. Section 5.3 is devoted to developing strategies for constructing derivations in *SD*, and Section 5.4 introduces the derivation system *SD*+, which is an expansion of *SD*.

5.1 THE DERIVATION SYSTEM *SD*

In Chapter 3 we presented semantic accounts of consistency, validity, equivalence, entailment, logical truth, and logical falsity. The semantic truth-table and truth-tree tests we developed for these properties in Chapters 3 and 4 show whether there is or is not a truth-value assignment of a particular kind for a particular sentence or group of sentences. These test procedures can hardly be said to reflect the reasoning we do in everyday discourse when we are trying to show, for example, that an argument is valid or that a set of sentences is inconsistent. In this chapter we develop techniques that do, at least in broad outline, parallel the kind of reasoning we do make use of in everyday discourse. These techniques rely on the form or structure of sentences of *SL* and are not intended to reveal whether there is or is not a truth-value assignment of a certain sort. These are therefore syntactic techniques.

Consider the following argument:

> If Marshall survives the current scandal and if her opponent doesn't outspend her then Marshall will be reelected. If it continues to be politics as usual Marshall will survive the latest scandal. The scandal is no longer front page news, so it is going to be politics as usual. Marshall's opponent will not outspend her. So Marshall will be reelected.

How might we, in everyday discourse, convince ourselves that the foregoing argument is valid? We will start by providing an explicit paraphrase of the premises and conclusion of this argument:

<u>If</u> (Marshall will survive the current scandal <u>and</u> <u>it is not the case that</u> Marshall's opponent outspends Marshall) <u>then</u> Marshall will be reelected.

<u>If</u> it continues to be politics as usual <u>then</u> Marshall will survive the current scandal.

<u>It is not the case that</u> the scandal is still front page news <u>and</u> it continues to be politics as usual.

<u>It is not the case that</u> Marshall's opponent outspends Marshall.

Marshall will be reelected.

Note that we paraphrased the third premise as a conjunction. The task before us is to show that starting with the premises as assumptions we can, by a series of obvious inferences, reach the conclusion. We can do this as follows.

1	<u>If</u> (Marshall will survive the current scandal <u>and</u> <u>it is not the case that</u> Marshall's opponent outspends Marshall) <u>then</u> Marshall will be reelected.	Assumption
2	<u>If</u> it continues to be politics as usual <u>then</u> Marshall will survive the current scandal.	Assumption
3	<u>It is not the case that</u> the scandal is still front page news <u>and</u> it continues to be politics as usual.	Assumption
4	<u>It is not the case that</u> Marshall's opponent outspends Marshall.	Assumption
5	It continues to be politics as usual.	From 3
6	Marshall will survive the current scandal.	From 2 and 5

7	Marshall will survive the current scandal and it is not the case that Marshall's opponent outspends Marshall.	From 6 and 4
8	Marshall will be reelected.	From 1 and 7

The structure of our reasoning may be more apparent when we symbolize these paraphrases in *SL* and indicate how each step of our reasoning is justified:

1	$(S \& \sim O) \supset R$	Assumption
2	$C \supset S$	Assumption
3	$\sim F \& C$	Assumption
4	$\sim O$	Assumption
5	C	From 3
6	S	From 2 and 5
7	$S \& \sim O$	From 6 and 4
8	R	From 1 and 7

In this section we develop the derivation system *SD* ('*SD*' for 'Sentential Derivation'), which consists of eleven derivation rules. Each of the inferences represented by lines 5 through 8 above will be justified by a syntactic rule of *SD*. These rules specify that if we have a sentence or sentences of such and such form or forms, then we may infer a sentence of a specified form. The rules are called '**derivation rules**' and the structures we construct using them are called '**derivations**'.

The simplest derivation rule of *SD* is Reiteration:

Reiteration (R)

| **P**

▷ | **P**

Here, and in the rule schema presented below, the '▷' sign indicates the sentence that can be inferred or **derived** using the rule in question. Here is a simple and admittedly uninteresting use of Reiteration:

1	C	Assumption
2	C	1 R

Reiteration is often used in strategies that involve **subderivations**, which we introduce later in this section.

The language *SL* includes five kinds of compound sentences: Negations, Conjunctions, Disjunctions, Material Conditionals, and Material Biconditionals—and there are two derivation rules of *SD* associated with each kind of compound. One rule is for deriving a sentence *from* a compound of the specified sort and the other is for deriving a compound *of* the specified sort. The former are **elimination** rules. A sentence derived by an elimination rule may have a main connective other than that after which the rule is named, or no main connective. The latter are **introduction** rules, so called because they *yield* an *SL* sentence whose main connective is the one after which the rule is named. Some of these ten rules make use of **subderivations**. We first present the rules that *do not* use subderivations.

5.1.1 THE NON-SUBDERIVATION RULES OF SD

The derivation rules that do not make use of subderivations are

Reiteration (R)

$$\begin{array}{l|l} & \mathbf{P} \\ \triangleright & \mathbf{P} \end{array}$$

Conjunction Elimination (&E)

$$\begin{array}{l|l} & \mathbf{P} \,\&\, \mathbf{Q} \\ \triangleright & \mathbf{P} \end{array} \qquad \begin{array}{l|l} & \mathbf{P} \,\&\, \mathbf{Q} \\ \triangleright & \mathbf{Q} \end{array}$$

Conjunction Introduction (&I)

$$\begin{array}{l|l} & \mathbf{P} \\ & \mathbf{Q} \\ \triangleright & \mathbf{P} \,\&\, \mathbf{Q} \end{array}$$

Disjunction Introduction ($\vee$I)

$$\begin{array}{l|l} & \mathbf{P} \\ \triangleright & \mathbf{P} \vee \mathbf{Q} \end{array} \qquad \begin{array}{l|l} & \mathbf{P} \\ \triangleright & \mathbf{Q} \vee \mathbf{P} \end{array}$$

Conditional Elimination ($\supset$E)

$$\begin{array}{l|l} & \mathbf{P} \supset \mathbf{Q} \\ & \mathbf{P} \\ \triangleright & \mathbf{Q} \end{array}$$

Biconditional Elimination ($\equiv$E)

$$\begin{array}{l|l} & \mathbf{P} \equiv \mathbf{Q} \\ & \mathbf{P} \\ \triangleright & \mathbf{Q} \end{array} \qquad \begin{array}{l|l} & \mathbf{P} \equiv \mathbf{Q} \\ & \mathbf{Q} \\ \triangleright & \mathbf{P} \end{array}$$

These rules are all quite straightforward. The abbreviation for each rule is given in parentheses following the rule name. In each case the sentence the '▷' symbol points to can be derived if the one or two sentences occurring above it have already been derived. Some of these rules have two versions.

> **Conjunction Elimination** specifies that if a conjunction occurs on an earlier line of a derivation then we may enter on a subsequent line either the left conjunct or the right conjunct of the conjunction.

Conjunction Introduction specifies that if **P** and **Q** occur on earlier lines of a derivation then we may enter **P** & **Q** on a subsequent line. Here the rule template should *not* be taken as specifying the order in which **P** and **Q** must be derived before **P** & **Q** can be entered.

Disjunction Introduction specifies that if a sentence **P** occurs on an earlier line of a derivation then we may enter on a subsequent line either **P** ∨ **Q** or **Q** ∨ **P**, where **Q** is *any* sentence of *SL*.

Conditional Elimination specifies that if **P** ⊃ **Q** and **P** occur on earlier lines of a derivation, in either order, then we may enter **Q** on a subsequent line.

Biconditional Elimination specifies that if a sentence of the form **P** ≡ **Q** and one of its immediate components (**P** or **Q**) occur on earlier lines of a derivation, in either order, then we may enter on a subsequent line the other immediate component.

Reiteration, which may seem to be a somewhat strange rule, is often used in strategies that involve **subderivations**, which we introduce later in this section.

Here is a derivation that uses both Conjunction Introduction and Conjunction Elimination:

1	B	Assumption
2	C & ~ D	Assumption
3	~ D	2 &E
4	B & ~ D	1, 3 &I

The sentences on lines 1 and 2 are assumptions, as is indicated in the justification column. The sentence on line 3 is obtained from line 2 by Conjunction Elimination. And the sentence on line 4 is obtained from lines 1 and 3 by Conjunction Introduction.

Disjunction Introduction may seem to be an odd rule, for given a sentence **P** why would we want to obtain **P** ∨ **Q** or **Q** ∨ **P**, both of which are clearly weaker than the sentence from which they can be obtained? The following derivation illustrates an application of Disjunction Introduction and why it is useful, as well as an application of Conditional Elimination:

Derive: H

1	F	Assumption
2	(F ∨ G) ⊃ H	Assumption
3	F ∨ G	1 ∨I
4	H	2, 3 ⊃E

In this derivation our goal was to obtain 'H' from our two assumptions. We indicated this by entering the word 'Derive:' followed by the sentence to be derived, in this case 'H', at the top of the derivation. Hereafter we will always so specify the sentence to be derived. The sentence on line 2 is a material conditional whose consequent is ‘H’. We saw that we could derive ‘H’ from this line *if we also had the antecedent,* ‘F ∨ G’. This was not one of our assumptions. We did have ‘F’, at line 1. But from ‘F’ we knew we could derive ‘F ∨ G’ by Disjunction Introduction, and we did so on line 3. ‘H’ then followed by Conditional Elimination on line 4.

The following derivation uses Biconditional Elimination and Conjunction Elimination (each twice) as well as Disjunction Introduction and Conjunction Introduction.

Derive: ~ A & (B ∨ C)

1	B ≡ (D ≡ ~ A)	Assumption
2	B & D	Assumption
3	B	2 &E
4	D ≡ ~ A	1, 3 ≡E
5	D	2 &E
6	~ A	4, 5 ≡E
7	B ∨ C	3 ∨I
8	~ A & (B ∨ C)	6, 7 &I

We will discuss strategies for constructing derivations at length later in this chapter. Here we note that the overall strategy we use in constructing derivations is to try to figure out how the desired sentence might be derived—which sentences we need to derive in order to derive that sentence, and then which sentences we need to derive to obtain those sentences, and so on, until we see a path from the given assumptions to the desired sentence. In the foregoing derivation we noted that the sentence to be derived is a conjunction and that conjunctions can be obtained by Conjunction Introduction. So we set about trying to derive the conjuncts of that conjunction, ‘~ A’ and ‘B ∨ C’. We reasoned that ‘~ A’ could be derived from line 1 by two uses of Biconditional Elimination if we could derive both ‘B’ and ‘D’, and we saw that we could derive both from line 2, by two uses of Conjunction Elimination. And once we had ‘B’ on line 3 it was easy to derive ‘B ∨ C’ on line 7 by Disjunction Introduction.

Our next derivation uses all of the Introduction and Elimination rules of *SD* we have so far introduced:

Derive: ~ C

1	~ A ≡ (B & ~ C)	Assumption
2	B & D	Assumption
3	(D ∨ C) ⊃ ~ A	Assumption
4	D	2 &E
5	D ∨ C	4 ∨I
6	~ A	3, 5 ⊃E
7	B & ~ C	1, 6 ≡E
8	~ C	7 &E

Our goal in this derivation was to derive ‘~ C’, and ‘~ C’ is a component of the sentence on line 1. We realized that ‘~ C’ could be derived from line 1 in two steps if we could first derive ‘~ A’ and that because ‘~ A’ is the consequent of the material conditional on line 3, it could be derived by Conditional Elimination if we could first derive '$D \vee C$'. The latter sentence follows from 'D' by Disjunction Introduction, and 'D' follows from the sentence on line 2, '$B \mathbin{\&} D$', by Conjunction Elimination.

5.1.1E EXERCISES

1. Complete the following derivations by entering justifications for the derived sentences:

a. Derive: $A \mathbin{\&} B$

1	A	Assumption
2	$A \supset B$	Assumption
3	B	
4	$A \mathbin{\&} B$	

*b. Derive: $\sim C$

1	$A \supset (B \mathbin{\&} \sim C)$	Assumption
2	$A \mathbin{\&} B$	Assumption
3	A	
4	$B \mathbin{\&} \sim C$	
5	$\sim C$	

c. Derive: $\sim (A \equiv \sim B)$

1	$\sim (A \equiv \sim B) \equiv (\sim C \vee \sim D)$	Assumption
2	$A \supset (\sim D \mathbin{\&} C)$	Assumption
3	$D \mathbin{\&} A$	Assumption
4	A	
5	$\sim D \mathbin{\&} C$	
6	$\sim D$	
7	$\sim C \vee \sim D$	
8	$\sim (A \equiv \sim B)$	

*d. Derive: $(E \mathbin{\&} D) \mathbin{\&} (\sim B \mathbin{\&} C)$

1	$\sim B \supset (D \mathbin{\&} E)$	Assumption
2	$(A \mathbin{\&} \sim B) \mathbin{\&} C$	Assumption
3	$A \mathbin{\&} \sim B$	
4	$\sim B$	
5	$D \mathbin{\&} E$	
6	D	
7	E	
8	$E \mathbin{\&} D$	
9	C	
10	$\sim B \mathbin{\&} C$	
11	$(E \mathbin{\&} D) \mathbin{\&} (\sim B \mathbin{\&} C)$	

e. Derive: F ⊃ ~ G

1	(E ∨ H) ⊃ (F ⊃ ~ G)	Assumption
2	(C ∨ D) ≡ (E & ~ H)	Assumption
3	C	Assumption
4	C ∨ D	
5	E & ~ H	
6	E	
7	E ∨ H	
8	F ⊃ ~ G	

*f. Derive: ~ G

1	(H & ~ I) ⊃ ~ G	Assumption
2	(F ∨ ~ G) ≡ H	Assumption
3	F & ~ I	Assumption
4	F	
5	F ∨ ~ G	
6	H	
7	~ I	
8	H & ~ I	
9	~ G	

g. Derive: D ≡ ~ B

1	(A & ~ B) ⊃ C	Assumption
2	(C ∨ D) ⊃ (D ≡ ~ B)	Assumption
3	~ B & A	Assumption
4	A	
5	~ B	
6	A & ~ B	
7	C	
8	C ∨ D	
9	D ≡ ~ B	

*h. Derive: M & ~ N

1	(K & ~ L) & (~ I & J)	Assumption
2	~ L ⊃ M	Assumption
3	(K & ~ I) ⊃ ~ N	Assumption
4	K & ~ L	
5	~ L	
6	M	
7	K	
8	~ I & J	
9	~ I	
10	K & ~ I	
11	~ N	
12	M & ~ N	

i. Derive: $\sim D \,\&\, \sim F$

1	$(A \vee \sim B) \equiv (A \,\&\, \sim F)$	Assumption
2	$C \equiv \sim B$	Assumption
3	$C \,\&\, \sim D$	Assumption
4	$\sim D$	
5	C	
6	$\sim B$	
7	$A \vee \sim B$	
8	$A \,\&\, \sim F$	
9	$\sim F$	
10	$\sim D \,\&\, \sim F$	

*j. Derive: $\sim (A \vee B)$

1	$A \supset [\sim B \supset \sim (A \vee B)]$	Assumption
2	$C \equiv (A \,\&\, \sim B)$	Assumption
3	$\sim D \,\&\, C$	Assumption
4	C	
5	$A \,\&\, \sim B$	
6	A	
7	$\sim B \supset \sim (A \vee B)$	
8	$\sim B$	
9	$\sim (A \vee B)$	

2. Complete the following derivations.

a. Derive: $D \,\&\, \sim B$

1	$A \,\&\, \sim B$	Assumption
2	$(A \vee \sim C) \supset D$	Assumption

*b. Derive: $F \,\&\, \sim H$

1	$F \equiv \sim G$	Assumption
2	$D \supset \sim G$	Assumption
3	$\sim H \,\&\, D$	Assumption

c. Derive: $\sim D \vee E$

1	$A \,\&\, \sim B$	Assumption
2	$\sim B \equiv (A \equiv \sim D)$	Assumption

*d. Derive: $\sim E \vee (G \,\&\, \sim F)$

1	$D \equiv (C \,\&\, \sim E)$	Assumption
2	$F \,\&\, (F \equiv D)$	Assumption

e. Derive: $H \,\&\, \sim I$

1	$\sim F \,\&\, \sim G$	Assumption
2	$\sim G \supset H$	Assumption
3	$(H \,\&\, \sim F) \equiv \sim I$	Assumption

f. Derive: $D \,\&\, \sim D$

*1	$(\sim A \,\&\, B) \supset (B \equiv D)$	Assumption
2	$B \supset (C \,\&\, \sim A)$	Assumption
3	$\sim D \,\&\, B$	Assumption

g. Derive: $F \,\&\, \sim G$

1	$(F \vee \sim G) \supset (F \,\&\, \sim H)$	Assumption
2	$\sim H \supset \sim G$	Assumption
3	$(\sim H \supset \sim G) \equiv F$	Assumption

5.1.2 THE SUBDERIVATION RULES OF SD

All five of the derivation rules we are about to introduce make use of subderivations. A subderivation is useful when we want to show that if we add an assumption to those we already made then we can derive a sentence that we may not be able to derive without the additional assumption. But every time we use a subderivation—which adds a new assumption—we must eventually end that subderivation and discontinue reliance on the assumption that starts that subderivation. Each subderivation rule provides a way of ending the subderivation it relies on. Once the way subderivations work is understood, it is fairly easy to master the subderivation rules. Our explication of Conditional Introduction will illustrate how subderivations work.

Conditional Introduction ($\supset$I)

		P
		Q
▷	**P** $\supset$ **Q**

Suppose we are trying to complete the following derivation:

Derive: $A \supset H$

1	$A \supset G$	Assumption
2	$G \supset H$	Assumption
	$A \supset H$	

Here we have entered the sentence we want to derive, ‘$A \supset H$’, some distance below our assumptions—because we want the last line of our derivation to be

'A ⊃ H' but we don't yet know how we can get from our assumptions to 'A ⊃ H'. Intuitively, we might reason as follows. We have 'A ⊃ G' as an assumption. *If* we also had 'A', we could derive 'G' by Conditional Elimination. And once we have 'G', we can derive 'H' by Conditional Elimination. That is, *given* 'A ⊃ G' and 'G ⊃ H' we can derive 'H' *if* we also have 'A'. We can encapsulate this reasoning in a subderivation:

Derive: A ⊃ H

1	A ⊃ G	Assumption
2	G ⊃ H	Assumption
3	A	A / ⊃I
4	G	1, 3 ⊃E
5	H	2,4 ⊃E
6	A ⊃ H	3–5 ⊃I

At line 3 we started a derivation within our existing derivation—hence the name 'subderivation'. In this case our purpose in doing so was to show that once we assume 'A' we can derive 'H' (in two steps), using our original assumptions and the assumption that starts our subderivation. Lines 3–5 show that, given our original assumptions, if we have 'A' we can derive 'H'. Note that this does not show that 'H' is a consequence of our original assumptions. Rather, we have shown that the conditional 'A ⊃ H' is a consequence of the original assumptions because we have shown how to derive 'H' given 'A'. It is the entire subderivation, which occupies lines 3–5, that justifies our entering 'A ⊃ C' on line 6. We indicate this by entering, in the justification column, '3–5 ⊃I', not '3,5 ⊃I'. This notation references the entire subderivation, not just lines 3 and 5.

We often use the reasoning process that is captured by Conditional Introduction in everyday reasoning. For example, suppose we know that if Jean gets an A in Biology 400 her grade point average will be 3.8, and that if her grade point average is 3.8 she will graduate with honors. If we assume she does get an A in Biology 400 it follows that she will have a grade point average of 3.8, and from this and what we know about the requirements for graduating with honors, it follows that Jean will graduate with honors. Of course, we do not conclude that Jean *will* graduate with honors, but rather that *if* she gets an A in Biology 400 *then* she will graduate with honors. If we use 'A' to symbolize 'Jean will get an A in Biology 400', 'G' to symbolize 'Jean will have a grade point average of 3.8', and 'H' to symbolize 'Jean will graduate with honors', the derivation we constructed using Conditional Introduction formalizes this reasoning about Jean and her graduating with honors.

There are several points to note before introducing the remaining subderivation rules. First, the vertical lines in a derivation are called '**scope lines**'. Assumptions with just one scope line to their left are the **primary**

assumptions of a derivation. Primary assumptions hold and are available for the entire derivation, as is indicated by the scope line to their immediate left that continues to the end of the derivation. Each subderivation begins with an **auxiliary assumption**, and the scope line to the immediate left of the auxiliary assumption indicates how far the scope of that assumption extends; the auxiliary assumption may be appealed to only so long as the scope line to its immediate left continues. In the above example there is one subderivation, occupying lines 3 through 5. The assumption of that subderivation is in force only through line 5.

We construct subderivations so that we can use rules that require subderivations. In the above example we constructed the subderivation so that we could use the rule Conditional Introduction. This rule calls for assuming, as an auxiliary assumption, the antecedent of the material conditional we wish to derive, and then deriving the consequent of that material conditional within the subderivation. In the justification column for a sentence entered as an auxiliary assumption, we write 'A' (for 'Assumption') and the abbreviation for the rule that calls for a subderivation of the sort we are constructing (here '⊃I'), separated by a slash ('/').

We end a subderivation by using the rule indicated on the assumption line of the subderivation to derive a sentence *outside* the scope of the subderivation, citing the entire subderivation. It is the entire subderivation that justifies applying a subderivation rule. When a subderivation is ended (by using a rule that cites the entire subderivation) we say that the assumption of that subderivation has been **discharged** and is **closed.** The scope of an assumption includes the assumption itself and all sentences and subderivations that occur subsequent to the assumption but before it is discharged. Once an assumption is discharged, neither it nor any sentence or subderivation lying within its scope can be appealed to in justifying subsequent lines of a derivation. We refer to assumptions that have not been discharged as being **open,** and to those that have been discharged as being **closed.** In our example, the scope of the assumption on line 3 extends only to line 5.

We can now give an informal account of **accessibility:** A sentence or subderivation is accessible at line **n** of a derivation (can be appealed to in justifying line **n**) if and only if every scope line to the left of the sentence or subderivation is also to the left of the sentence on line **n**.

Thus, scope lines, the vertical lines to the left of the sentences of a derivation, provide a visual way of telling when a sentence or subderivation is accessible. The leftmost vertical line is the scope line for the entire derivation. Primary assumptions, if any, appear to the immediate right of this scope line at the top of the derivation. Every auxiliary assumption has its own scope line, a line that continues only so long as that assumption remains open. A sentence is accessible only as long as the scope lines to its left continue. Primary assumptions, of course, are never discharged. If a sentence or subderivation is accessible at a given line of a derivation then it can be appealed to in justifying the sentence entered on that line.

Here is another derivation that uses Conditional Introduction:

Derive: A ⊃ (B ⊃ C)

```
1  | C                      Assumption
   |---
2  |  | A                   A / ⊃I
   |  |---
3  |  |  | B                A / ⊃I
   |  |  |---
4  |  |  | C                1 R
5  |  | B ⊃ C               3–4 ⊃I
6  | A ⊃ (B ⊃ C)            2–5 ⊃I
```

This derivation contains two subderivations, one nested within the other. The innermost subderivation occupies lines 3–4; the outer subderivation lines 2–5. At line 4 we were able to use Reiteration to derive 'C' because every scope line to the left of 'C' on line 1 (there is only one) is also to the left of the sentence we entered on line 4. And on line 5 we were able to enter 'B ⊃ C' by Horseshoe Introduction because every scope line to the left of 'B ⊃ C' at line 5 (there are two) is also to the left of the *subderivation* occupying lines 3–4. Note that while there are three scope lines to the left of the *sentences* on lines 3 and 4, the rightmost of these is *part of* the subderivation occupying lines 3–4. So there are 2, not 3, scope lines to the left of that subderivation. Neither 'A', the auxiliary assumption that begins the subderivation occupying lines 2–5, nor 'B', the auxiliary assumption that begins the subderivation occupying lines 3–4, was appealed to in deriving the sentences 'B ⊃ C' and 'C' within those subderivations. It is often the case that the auxiliary assumption that begins a subderivation is not appealed to until the subderivation is ended (when it is appealed to as part of the entire subderivation).

The following variant of the previous derivation is also constructed in accordance with the rules of *SD*.

Derive: A ⊃ (B ⊃ C)

```
1  | C                      Assumption
   |---
2  |  | A                   A / ⊃I
   |  |---
3  |  |  | B                A / ⊃I
   |  |  |---
4  |  |  | B & C            1, 3 &I
5  |  |  | C                4 &E
6  |  | B ⊃ C               3–5 ⊃I
7  | A ⊃ (B ⊃ C)            2–6 ⊃I
```

At line 4 both 'C' on line 1 and 'B' on line 3 are accessible, so our use of Conjunction Introduction is allowed. However, we did not choose to construct this derivation, because it is one line longer than our earlier derivation of 'A ⊃ (B ⊃ C)' from

'C'. On the other hand, the following variation *is not* constructed in accordance with the rules of *SD*.

Derive: A ⊃ (B ⊃ C)

1	C	Assumption	
2	A	A / ⊃I	
3	B	A / ⊃I	
4	C	1 R	
5	B ⊃ C	3–4 ⊃I	
6	A ⊃ (B ⊃ C)	2–5 ⊃I	
7	B & C	3, 4 &I	**MISTAKE!**

Neither the sentence on line 3 nor that on line 4 is accessible from line 7. There are two scope lines to the left of the sentences on lines 3 and 4 that *do not* extend to the left of the sentence we tried to enter on line 7 (the assumptions on lines 2 and 3 were closed before line 7, so sentences falling within the scope of either assumption cannot be appealed to on line 7).

The remaining four subderivation rules are as follows:

Negation Introduction (~ I)

| **P**
| **Q**
| **~ Q**
▷ **~ P**

Negation Elimination (~ E)

| **~ P**
| **Q**
| **~ Q**
▷ **P**

Disjunction Elimination (∨E)

P ∨ Q
| **P**
| **R**
| **Q**
| **R**
▷ **R**

Biconditional Introduction (≡I)

| **P**
| **Q**
| **Q**
| **P**
▷ **P ≡ Q**

Negation Introduction specifies that if we can derive a sentence and its negation, **Q** and ~ **Q**, within the scope of an auxiliary assumption **P**, then we may end the subderivation and enter ~ **P** on the following line. Here and with the remaining subderivation rules the template should not be taken as specifying the order in which sentences must be derived within the subderivation.

> **Negation Elimination** specifies that if we can derive a sentence and its negation, **Q** and ~ **Q**, within the scope of an auxiliary assumption ~ **P**, then we may end the subderivation and enter **P** on a subsequent line.
>
> **Disjunction Elimination** specifies that if **P** ∨ **Q** occurs on an earlier line of a derivation and subsequent to it there are two subderivations, one of **R** from **P** and the other of **R** from **Q**, then **R** may be entered on a subsequent line.
>
> **Biconditional Introduction** specifies that if the derivation contains two subderivations, one of **Q** from **P** and one of **P** from **Q**, then **P** ≡ **Q** may be entered on a subsequent line.

Negation Introduction and Negation Elimination both parallel a pattern of reasoning we often use in everyday life, *reductio ad absurdum* reasoning. In this reasoning, we make an assumption and then show that an absurd result follows from that assumption and whatever other assumptions we may already have made. To avoid the absurdity we reject one of our assumptions. Here is an example of *reductio ad absurdum* reasoning. Suppose we know the following:

> Billings was shot to death in New York City during the evening hours of October 25. Billings' partner, Jenkins, became sole owner of their company as a result of Billing's death, and Jenkins is in dire financial straits and has always hated his partner.

We want to explore the possibility that Jenkins shot Billings, and we do so by assuming that he did. Further investigation reveals that Jenkins was seen sitting in a *Pizza Uno* restaurant in Chicago the entire evening of the shooting. We now reason as follows:

> Suppose Jenkins shot Billings. Then Jenkins was in New York on the evening of the 25^{th}. But we know he was in Chicago that entire evening, so he was not in New York. Therefore, Jenkins did not shoot Billings.

The assumption that Jenkins shot Billings, along with our knowledge that he was in Chicago the entire evening of October 25, leads to the absurdity the Billings was both in New York City and not in New York City that evening. So we rejected our assumption and concluded that Jenkins did not shoot Billings.

Both Negation Introduction and Negation Elimination mirror this kind of reasoning. In both cases we make an assumption and then derive a sentence and its negation (**Q** and ~ **Q**). It would be absurd to remain committed to both a sentence and its negation. But we are so committed as long as we can derive both. So we reject the assumption that starts the subderivation—we close the subderivation and enter **P** (if our assumption was ~ **P**) or ~ **P** (if our assumption was **P**).

Here are some fairly simple derivations in which Negation Introduction and Negation Elimination are used:

Derive: ~ H

```
 1 | H ⊃ F                 Assumption
 2 | ~ F                   Assumption
   |-----
 3 |   | H                 A / ~ I
   |   |-----
 4 |   | F                 1, 3 ⊃E
 5 |   | ~F                2 R
 6 | ~ H                   3–5 ~ I
```

Derive: N

```
 1 | ~ N ⊃ S               Assumption
 2 | S ⊃ C                 Assumption
 3 | C ⊃ N                 Assumption
   |-----
 4 |   | ~ N               A / ~ E
   |   |-----
 5 |   | S                 1, 4 ⊃E
 6 |   | C                 2, 5 ⊃E
 7 |   | N                 3, 6 ⊃E
 8 |   | ~N                4 R
 9 | N                     4–8 ~ E
```

We noted when we introduced the rule Reiteration that it would often be useful in derivations involving subderivations, and we have used Reiteration in both of these derivations (as we did in an earlier use of Conditional Introduction). Next we present a derivation that uses both Negation Introduction and Negation Elimination:

Derive: ~ A & B

```
 1 | ~ (A ~∨ B)            Assumption
   |-----
 2 |   | A                 A / ~ I
   |   |-----
 3 |   | A ~∨ B            2 ∨I
 4 |   | ~ (A ~∨ B)        1 R
 5 | ~ A                   2–4 ~ I
 6 |   | ~ B               A / ~ E
   |   |-----
 7 |   | A ~∨ B            6 ∨I
 8 |   | ~ (A ~∨ B)        1 R
 9 | B                     6–8 ~ E
10 | ~ A & B               5, 9 &I
```

In this derivation the sentence to be derived is a conjunction, so we opted to derive it by Conjunction Introduction. Having made that decision, we were left with two goals: deriving '~ A' and deriving 'B'. We derived '~ A' by Negation Introduction and 'B' by Negation Elimination. One key to constructing this derivation was recognizing that our primary assumption, '~ (A ∨ ~ B)', was a negation and hence was a candidate for serving as the negation ~ **Q** that would be needed within both our Negation Introduction and our Negation Elimination subderivations. The other key was recognizing that the sentence **Q** in this case, 'A ∨ ~ B', could be derived by Disjunction Introduction from the assumption 'A' in the first subderivation and from '~ B' in the second.

Disjunction Elimination also parallels a pattern of reasoning we use in everyday life. Here is an example:

> The CEO is incompetent and will either resign or be fired. If she resigns she will move to Boston to be near her son. If she is fired she will move to Boston to live with her son. So the CEO will move to Bostson.

In this example we know the CEO will either resign or be fired. If the first happens, she will end up in Boston, and if the second happens, she will also end up in Boston. So whichever happens, the CEO will end up in Boston. The following derivation formalizes this reasoning:

Derive: B

1	I & (R ∨ F)		Assumption
2	R ⊃ (B & N)		Assumption
3	F ⊃ (B & L)		Assumption
4	R ∨ F		1 & E
5		R	A / ∨E
6		B & N	2, 5 ⊃E
7		B	6 &E
8		F	A / ∨E
9		B & L	3, 8 ⊃E
10		B	9 &E
11	B		4, 5–7, 8–10 ∨E

In this derivation we derived a disjunction on line 4 and then constructed two subderivations. The first has 'R', the left disjunct of 'R ∨ F', as its auxiliary assumption. The subderivation shows that given 'R', 'B' can be derived. The second subderivation shows that 'B' can also be derived from the right disjunct, 'F'. Having derived 'B' from each disjunct, we entered 'B' on line 11. The justification for line 11 cites the disjunction on line 4 and the two subderivations, one occupying lines 5–7 and the other lines 8–10.

Material biconditionals of *SL*, sentences of the form **P** ≡ **Q**, have the force of two material conditionals, (**P** ⊃ **Q**) and (**Q** ⊃ **P**). Hence it should not be surprising that Biconditional Introduction requires two subderivations, one in which **Q** is derived from **P** and one in which **P** is derived from **Q**. Here is a simple use of Biconditional Introduction:

Derive: A ≡ B

1	A ⊃ B	Assumption
2	B ⊃ A	Assumption
3	| A	A / ≡I
4	| B	1, 3 ⊃E
5	| B	A / ≡I
6	| A	2, 5 ⊃E
7	A ≡ B	3–4, 5–6 ≡I

The following derivation uses both Biconditional Elimination and Biconditional Introduction:

Derive: A ≡ C

1	A ≡ B	Assumption
2	B ≡ C	Assumption
3	| A	A / ≡I
4	| B	1, 3 ≡E
5	| C	2, 4 ≡E
6	| C	A / ≡I
7	| B	2, 6 ≡E
8	| A	1, 7 ≡E
9	A ≡ C	3–5, 6–8 ≡I

5.1.2E EXERCISES

1. Complete the following derivations by entering the appropriate justifications:

a. Derive: (A ⊃ B) & (A ⊃ ~ D)

1	A ⊃ (B & ~ D)	Assumption
2	| A	
3	| B & ~ D	
4	| B	
5	A ⊃ B	
6	| A	
7	| B & ~ D	
8	| ~ D	
9	A ⊃ ~ D	
10	(A ⊃ B) & (A ⊃ ~ D)	

*b. Derive: A ⊃ [B ⊃ (C ∨ D)]

1	(A & B) ⊃ C	Assumption
2	A	
3	B	
4	A & B	
5	C	
6	C ∨ D	
7	B ⊃ (C ∨ D)	
8	A ⊃ [B ⊃ (C ∨ D)]	

c. Derive: B

1	~ B ⊃ B	Assumption
2	~ B	
3	B	
4	~ B	
5	B	

*d. Derive: A ⊃ ~ B

1	A ⊃ (B ⊃ C)	Assumption
2	~ C	Assumption
3	A	
4	B ⊃ C	
5	B	
6	C	
7	~ C	
8	~ B	
9	A ⊃ ~ B	

e. Derive: E ∨ D

1	A ∨ (B & ~ C)	Assumption
2	A ⊃ D	Assumption
3	~ C ⊃ E	Assumption
4	A	
5	D	
6	E ∨ D	
7	B & ~ C	
8	~ C	
9	E	
10	E ∨ D	
11	E ∨ D	

*f. Derive: E

1	$F \supset (\sim G \vee \sim H)$	Assumption
2	$(\sim G \supset E) \;\&\; (\sim E \supset H)$	Assumption
3	F	Assumption
4	$\sim G \vee \sim H$	
5	$\sim G$	
6	$\sim G \supset E$	
7	E	
8	$\sim H$	
9	$\sim E \supset H$	
10	$\sim E$	
11	H	
12	$\sim H$	
13	E	
14	E	

g. Derive: $F \equiv \sim G$

1	$(F \supset \sim G) \;\&\; (\sim G \supset F)$	Assumption
2	F	
3	$F \supset \sim G$	
4	$\sim G$	
5	$\sim G$	
6	$\sim G \supset F$	
7	F	
8	$F \equiv \sim G$	

*h. Derive: $H \equiv J$

1	$(H \;\&\; I) \equiv J$	Assumption
2	$H \equiv I$	Assumption
3	H	
4	I	
5	$H \;\&\; I$	
6	J	
7	J	
8	$H \;\&\; I$	
9	H	
10	$H \equiv J$	

2. Complete the following derivations.

a. Derive: $A \equiv B$

1	A	Assumption
2	B	Assumption

*b. Derive: $\sim B$

1	$B \supset \sim B$	Assumption

c. Derive: A

1	$\sim \sim A$	Assumption

*d. Derive: $H \; \& \sim I$

1	$I \; \& \sim I$	Assumption

e. Derive: B

1	$\sim B \supset C$	Assumption
2	$\sim C \equiv A$	Assumption
3	A	Assumption

*f. Derive: $A \equiv C$

1	$A \equiv \sim B$	Assumption
2	$\sim B \equiv C$	Assumption

g. Derive: $\sim H$

1	$H \supset I$	Assumption
2	$\sim I$	Assumption

*h. Derive: $\sim G$

1	$\sim F \supset \sim G$	Assumption
2	$\sim F \vee H$	Assumption
3	$H \equiv \sim G$	Assumption

i. Derive: $\sim (F \vee G)$

1	$(F \vee G) \supset (H \; \& \; I)$	Assumption
2	$\sim H$	Assumption

*j. Derive: $\sim (F \; \& \; G)$

1	$F \equiv (\sim G \; \& \; H)$	Assumption

5.1.3 CONCLUDING COMMENTS

All the derivation rules of *SD* have been introduced. We repeat them here, for easy reference. Rather than listing the rules that do not use subderivations separately from those that do use subderivations, we here arrange the derivation rules by the kind of compound that is either appealed to or introduced. The rules can also be found on the inside front cover of this text.

Reiteration (R)

$$\begin{array}{r|l} & \mathbf{P} \\ \triangleright & \mathbf{P} \end{array}$$

Conjunction Introduction (&I)

$$\begin{array}{r|l} & \mathbf{P} \\ & \mathbf{Q} \\ \triangleright & \mathbf{P} \;\&\; \mathbf{Q} \end{array}$$

Conjunction Elimination (&E)

$$\begin{array}{r|l} & \mathbf{P} \;\&\; \mathbf{Q} \\ \triangleright & \mathbf{P} \end{array} \qquad \begin{array}{r|l} & \mathbf{P} \;\&\; \mathbf{Q} \\ \triangleright & \mathbf{Q} \end{array}$$

Conditional Introduction (⊃I)

$$\begin{array}{r|l} & \begin{array}{|l} \mathbf{P} \\ \hline \mathbf{Q} \end{array} \\ \triangleright & \mathbf{P} \supset \mathbf{Q} \end{array}$$

Conditional Elimination (⊃E)

$$\begin{array}{r|l} & \mathbf{P} \supset \mathbf{Q} \\ & \mathbf{P} \\ \triangleright & \mathbf{Q} \end{array}$$

Negation Introduction (~ I)

$$\begin{array}{r|l} & \begin{array}{|l} \mathbf{P} \\ \hline \mathbf{Q} \\ \sim \mathbf{Q} \end{array} \\ \triangleright & \sim \mathbf{P} \end{array}$$

Negation Elimination (~ E)

$$\begin{array}{r|l} & \begin{array}{|l} \sim \mathbf{P} \\ \hline \mathbf{Q} \\ \sim \mathbf{Q} \end{array} \\ \triangleright & \mathbf{P} \end{array}$$

Disjunction Introduction (∨I)

$$\begin{array}{r|l} & \mathbf{P} \\ \triangleright & \mathbf{P} \vee \mathbf{Q} \end{array} \qquad \begin{array}{r|l} & \mathbf{P} \\ \triangleright & \mathbf{Q} \vee \mathbf{P} \end{array}$$

Disjunction Elimination (∨E)

$$\begin{array}{r|l} & \mathbf{P} \vee \mathbf{Q} \\ & \begin{array}{|l} \mathbf{P} \\ \hline \mathbf{R} \end{array} \\ & \begin{array}{|l} \mathbf{Q} \\ \hline \mathbf{R} \end{array} \\ \triangleright & \mathbf{R} \end{array}$$

Biconditional Introduction (≡I)

$$\begin{array}{l|l} & \begin{array}{|l} \mathbf{P} \\ \hline \mathbf{Q} \end{array} \\ & \begin{array}{|l} \mathbf{Q} \\ \hline \mathbf{P} \end{array} \\ \triangleright & \mathbf{P} \equiv \mathbf{Q} \end{array}$$

Biconditional Elimination (≡E)

$$\begin{array}{l|l} & \mathbf{P} \equiv \mathbf{Q} \\ & \mathbf{P} \\ \triangleright & \mathbf{Q} \end{array} \qquad \begin{array}{l|l} & \mathbf{P} \equiv \mathbf{Q} \\ & \mathbf{Q} \\ \triangleright & \mathbf{P} \end{array}$$

We have presented the derivation rules of *SD* and constructed a fair number of derivations. But we haven't actually defined the term 'derivation in *SD*'. We do so now:

> A *derivation in SD* is a series of sentences of *SL,* each of which is either an assumption or is obtained from previous sentences by one of the rules of *SD*.

We will continue to annotate our derivations with line numbers, scope and assumption lines, and line justifications. However, these annotations are not, as the above definition makes clear, officially parts of derivations.

There are many truth-preserving templates we do not include as rules of either *SD* or *SD+*. Why are some included and others not? For *SD* the answer is fairly simple. We want a derivation system to be truth-preserving (include no rule that ever takes us from truths to a falsehood). A system that has this property, never taking us from truths to a falsehood, is said to be **sound.** We also want our derivation systems to be **complete.** A derivation system is complete if and only if every sentence that is truth-functionally entailed by a set of sentences can be derived from that set. *SD* is complete in this sense and it is a fairly minimalist derivation system—it includes only two rules for each connective.[1] *SD+* will also be complete but includes additional derivation rules, some because they mirror reasoning patterns that are common in everyday discourse, some because they have historically been included in derivation systems. We prove that both *SD* and *SD+* are complete in Chapter 6.

Before ending this section we will take time to caution against some mistakes that are commonly made while constructing derivations. First, the derivation rules of *SD* are rules of inference, which is to say that when they appeal to a line earlier in the derivation they appeal to the entire sentence on that line, not to a sentence that is a component of a longer sentence. Here is

[1]Two rules of *SD,* Reiteration and Negation Introduction, could be dropped without making the system incomplete. This is not true of any of the other rules of *SD*.

an attempt at a derivation that misuses Conjunction Elimination by appealing to a component of a longer sentence.

Derive $A \supset C$

1	$A \supset (B \,\&\, C)$	Assumption	
2	$\quad A$	$A / \supset I$	
3	$\quad C$	$1 \,\&E$	**MISTAKE!**
4	$A \supset C$	$2\text{–}3 \supset I$	

The mistake at line 3 results from trying to apply Conjunction Elimination to a component of a longer sentence. The sentence on line 1 is *not* of the form **P** & **Q**, and while a component of that sentence, 'B & C', is of that form, rules of inference work, again, on sentences that are not themselves parts of longer sentences. A correct derivation for this problem is

Derive $A \supset C$

1	$A \supset (B \,\&\, C)$	Assumption
2	$\quad A$	$A / \supset I$
3	$\quad B \,\&\, C$	$1, 2 \supset E$
4	$\quad C$	$3 \,\&E$
5	$A \supset C$	$2\text{–}4 \supset I$

The sentence on line 3 is of the form **P** & **Q**. It is not part of a longer sentence on that line. So we can apply Conjunction Elimination to it and obtain 'C' at line 4.

Here is a similar misuse of a derivation rule.

Derive: C

1	$B \supset (A \supset C)$	Assumption	
2	A	Assumption	
3	C	$1, 2 \supset E$	**MISTAKE!**

Here an attempt has been made to apply Conditional Elimination to a component, '$A \supset C$' of the longer sentence '$B \supset (A \supset C)$' and this cannot be done. In this case there is no correct derivation. 'C' does not follow from the assumptions on lines 1 and 2.

Another common mistake is to appeal to lines or subderivations that are not accessible. In a derivation a sentence or subderivation is *accessible* at line **n** (it can be appealed to when justifying a sentence on line **n**) if and only if that sentence or subderivation does not lie within the scope of a closed assumption, that is, an assumption that has been discharged prior to line **n**. Here is an attempt at a derivation that twice violates the accessibility requirement:

Derive: B

```
1 |  | B                          A / ⊃I
  |  |---
2 |  |   | A                      A / ⊃I
  |  |   |---
3 |  |   | B                      1 R
4 |  | A ⊃ B                      2–3 ⊃I
5 | B ⊃ (A ⊃ B)                   1–4 ⊃I
6 | A ⊃ B                         2–3 ⊃I        MISTAKE!
7 | B                             2, 6 ⊃E       MISTAKE!
```

Line 6 is a mistake because it appeals to a subderivation, that occurring on lines 2 through 3, that is no longer accessible. It is not accessible at line 6, because not every scope line to the left of that subderivation (there are two) continues to line 6. The auxiliary assumption occurring on line 2 was discharged at line 4, when Conditional Introduction was used. (We also cannot use Reiteration to obtain A ⊃ B on line 6, because the sentence on line 4 is inaccessible at that point.) Line 7 is a mistake because it appeals to a line, line 2, which is no longer accessible. Of course, it is also a mistake because it appeals to a line, line 6, which is itself a mistake. In fact, neither line 6 nor line 7 can be derived in a derivation that has no primary assumptions. On the other hand, part of the above attempt, namely the part consisting of lines 1 through 5, is correct, demonstrating that some sentences can be derived starting from no primary assumptions. 'B ⊃ (A ⊃ B)' is one such sentence.

The following derivation is correctly done.

Derive: ~ U ⊃ ~ S

```
1 | ~ U ⊃ ~ W                     Assumption
2 | ~ W ⊃ ~ S                     Assumption
  |----------
3 |  | ~ U                        A / ⊃I
  |  |----
4 |  | ~W                         1, 3 ⊃E
5 |  | ~ S                        2, 4 ⊃E
6 | ~ U ⊃ ~S                      3–5 ⊃I
```

Line 4 cites lines 1 and 3, which are both accessible at line 4. The sentences on lines 1 and 3 do not lie within the scope of an assumption that has been discharged prior to line 4. (Neither the sentence on line 1 nor the sentence on line 3 has a scope line to its left that is not also to the left of the sentence on line 4.) Similarly line 5 cites lines 2 and 4, which are both accessible at line 5. Line 6 cites the subderivation from lines 3–5. This subderivation is accessible at line 6 because the subderivation does not lie within the scope of an assumption that has been closed prior to line 6.

Here is another example in which an inaccessible subderivation is cited:

Derive A ≡ C

1	~ C	Assumption
2	B ⊃ C	Assumption
3	~ A & ~ B	Assumption
4	A	A / ≡I
5	~ B	A / ~ E
6	~ A	3 &E
7	A	4 R
8	B	5–7 ~E
9	C	2, 8 ⊃E
10	C	A / ≡I
11	~ B ⊃ A	5–7 ⊃I **MISTAKE!**
12	~ B	3 &E
13	A	11, 12 ⊃E
14	A ≡ C	4–9, 10–13 ≡I

The mistake at line 11 is that of citing a subderivation that is not accessible from line 11. That it is not accessible is indicated by there being a scope line to the left of the subderivation, the scope line running from line 4 through line 9, that is not to the left of the sentence entered at line 11. More substantively, 'A' was derived at line 7 by Reiteration on line 4. The assumption at line 4 is not accessible at line 11, and neither are results obtained while it was available.

In fact, it is possible to derive 'A ≡ C' from the above primary assumptions. Here is a derivation that does so.

Derive A ≡ C

1	~ C	Assumption
2	B ⊃ C	Assumption
3	~ A & ~ B	Assumption
4	A	A / ≡I
5	~ C	A / ~ E
6	~ A	3 &E
7	A	4 R
8	C	5–7 ~ E
9	C	A / ~ I
10	~ A	A / ~ E
11	C	9 R
12	~ C	1 R
13	A	10–12 ~ E
14	A ≡ C	4–8, 9–13 ≡I

It is possible to use a single auxiliary assumption to generate a subderivation that allows the use of two different subderivation rules. Here is such a case:

Derive: $C \& (A \supset C)$

1	$A \vee B$	Assumption
2	$A \supset D$	Assumption
3	$B \supset D$	Assumption
4	$\sim C \supset \sim D$	Assumption
5	│ A	A / $\vee$E / $\supset$I
6	│ │ $\sim C$	A / $\sim$ E
7	│ │ $\sim D$	4, 6 $\supset$E
8	│ │ D	2, 5 $\supset$E
9	│ C	6–8 $\sim$ E
10	│ B	A / $\vee$E
11	│ │ $\sim C$	A / $\sim$ E
12	│ │ $\sim D$	4, 11 $\supset$E
13	│ │ D	3, 10 $\supset$E
14	│ C	11–13 $\sim$ E
15	C	1, 5–9, 10–14 $\vee$E
16	$A \supset C$	5–9 $\supset$I
17	$C \& (A \supset C)$	15, 16 &I

Notice that the subderivation occupying lines 5 through 9 is cited twice, once as part of an application of the rule Disjunction Elimination (at line 15) and once as the basis for entering a conditional at line 16. In the present case it is unlikely that when the assumption at line 5 is made it was foreseen that the subderivation to be constructed would be used in both of the above indicated ways. So most likely at the time the assumption was made the only notation entered in the justification column was 'A / ∨E'. It is only after reaching 'C' at line 15 and wondering how '$A \supset C$' can be obtained that it became apparent that work already done, the subderivation on lines 5 through 9, could be reused. So the extra notation '/ ⊃I' was added to line 5 when line 16 was entered.

In the above example identical subderivations occur on lines 6 through 8 and lines 11 through 13. We had to do this work twice because when trying to get from 'B' at line 10 to 'C' on a subsequent line the subderivation occupying lines 6 through 8 is no longer accessible.

Finally, it is possible to end a subderivation at any time, without using one of the introduction rules that requires a subderivation. This is likely to occur when one decides the strategy being pursued is unproductive and simply abandons the work done within the subderivation. Here is an example:

```
Derive: A ⊃ (B ⊃ A)

1  │ │ A                        A / ⊃I
   │ ├──
2  │ │ │ ~ (B ⊃ A)              A / ~ E
   │ │ ├──────────
3  │ │ │ A                      1 R
4  │ │ │ B                      A / ⊃I
   │ │ ├──
5  │ │ │ A                      1 R
6  │ │ B ⊃ A                    4–5 ⊃I
7  │ A ⊃ (B ⊃ A)                1–6 ⊃I
```

Here the subderivation on lines 2–3 is in effect wasted work, work we have thrown away. It does no harm, but neither does it do any good.

5.1.3E EXERCISES

1. Complete each of the following derivations by entering the appropriate justifications.

a. Derive: (A & C) ∨ (B & C)

```
1  │ (A ∨ B) & C
   ├────────────
2  │ A ∨ B
3  │ C
4  │ │ A
   │ ├──
5  │ │ A & C
6  │ │ (A & C) ∨ (B & C)
7  │ │ B
   │ ├──
8  │ │ B & C
9  │ │ (A & C) ∨ (B & C)
10 │ (A & C) ∨ (B & C)
```

*b. Derive: A ⊃ (B ⊃ C)

```
1  │ (A & B) ⊃ C
   ├────────────
2  │ │ A
   │ ├──
3  │ │ │ B
   │ │ ├──
4  │ │ │ A & B
5  │ │ │ C
6  │ │ B ⊃ C
7  │ A ⊃ (B ⊃ C)
```

c. Derive: ~ B

```
1  │ B ⊃ (A & ~ B)
   ├──────────────
2  │ │ B
   │ ├──
3  │ │ A & ~ B
4  │ │ ~ B
5  │ │ B
6  │ ~ B
```

*d. Derive: A ⊃ B

```
1  │ (A & ~ B) ⊃ (~ B & C)
2  │ C ⊃ ~ A
   ├──────────────────────
3  │ │ A
   │ ├──
4  │ │ │ ~ B
   │ │ ├────
5  │ │ │ A & ~ B
6  │ │ │ ~ B & C
7  │ │ │ C
8  │ │ │ ~ A
9  │ │ │ A
10 │ │ B
11 │ A ⊃ B
```

e. Derive: C ⊃ (~ A & B)

1 ~ D
2 C ⊃ (A ≡ B)
3 (D ∨ B) ⊃ ~ A
4 (A ≡ B) ⊃ (D & E)
5 ~ B ⊃ D
6 C
7 A ≡ B
8 D & E
9 D
10 D ∨ B
11 ~ A
12 ~ B
13 D
14 ~ D
15 B
16 ~ A & B
17 C ⊃ (~ A & B)

*f. Derive: A ⊃ (B ∨ C)

1 (~ B & ~ C) ⊃ ~ A
2 A
3 ~ (B ∨ C)
4 B
5 B ∨ C
6 ~ (B ∨ C)
7 ~ B
8 C
9 B ∨ C
10 ~ (B ∨ C)
12 ~ C
13 ~ B & ~ C
14 ~ A
15 A
16 B ∨ C
17 A ⊃ (B ∨ C)

g. Derive: A ≡ B

1 ~ A & ~ B
2 A
3 ~ B
4 ~ A
5 A
6 B
7 B
8 ~ A
9 B
10 ~ B
11 A
12 A ≡ B

*h. Derive: A ≡ (B ∨ C)

1 (A ≡ B) & (A ≡ C)
2 A
3 A ≡ B
4 B
5 B ∨ C
6 B ∨ C
7 B
8 A ≡ B
9 A
10 C
11 A ≡ C
12 A
13 A
14 A ≡ (B ∨ C)

5.2 BASIC CONCEPTS OF *SD*

We now define the key concepts of *SD*. These are all syntactical concepts as each is defined by reference to there being a derivation of a certain sort—no reference is made in any of these definitions either to truth-values or to truth-value assignments.

> *Derivability:* A sentence **P** of *SL* is *derivable in SD* from a set Γ of sentences of *SL* if and only if there is a derivation in *SD* in which all the primary assumptions are members of Γ and **P** occurs in the scope of only those assumptions.
>
> *Valid in SD:* An argument of *SL* is *valid in SD* if and only if the conclusion of the argument is derivable in *SD* from the set consisting of the premises. An argument of *SL* is *invalid in SD* if and only if it is not valid in *SD*.
>
> *Theorem in SD:* A sentence **P** of *SL* is a *theorem in SD* if and only if **P** is derivable in *SD* from the empty set.
>
> *Equivalence in SD:* Sentences **P** and **Q** are *equivalent in SD* if and only if **Q** is derivable in *SD* from {**P**} and **P** is derivable in *SD* from {**Q**}.
>
> *Inconsistency in SD:* A set Γ of sentences of *SL* is *inconsistent in SD* if and only if there is a sentence **P** such that both **P** and ~ **P** are derivable in *SD* from Γ. A set Γ is *consistent in SD* if and only if it is not inconsistent in *SD*.

A few additional notational conventions will be useful. We will use the single turnstile, '⊢' to assert derivability, and will read

$\Gamma \vdash \mathbf{P}$

as '**P** is derivable from Γ'. We will read 'Γ ⊬ **P**' as '**P** is not derivable from Γ'. This parallels our use of the double turnstile in previous chapters, where we read

$\Gamma \vDash \mathbf{P}$

as 'Γ truth-functionally entails **P**' and 'Γ ⊭ **P**' as 'Γ does not truth-functionally entail **P**'. The parallelism is for good reason. It will turn out that for any finite set Γ of sentences of *SL* and any sentence **P** of *SL*,

$\Gamma \vdash \mathbf{P}$ in *SD* if and only if $\Gamma \vDash \mathbf{P}$.

This is a key claim of metatheory that we prove in Chapter 6. Finally, we will read

$\vdash \mathbf{P}$

as '**P** is a theorem'. This notation derives from '$\varnothing \vdash$ **P**', which is read '**P** is derivable from the empty set'. And of course a sentence of *SL* is a theorem of *SD* if and only if it is derivable in *SD* from the empty set. We will also refer to a derivation of a sentence of *SL* from no primary assumptions as a **proof** of the theorem that is the last line of that derivation.

The careful reader will recall that there are seven key semantical concepts of *SL:* Truth-functional consistency, truth-functional truth, truth-functional falsity, truth-functional indeterminacy, truth-functional equivalence, truth-functional validity, and truth-functional entailment. We have syntactic parallels for only five of those concepts. These pair up as follows:

Truth-functional consistency	Consistency in *SD*
Truth-functional truth	Theorem in *SD*
Truth-functional equivalence	Equivalence in *SD*
Truth-functional validity	Valid in *SD*
Truth-functional entailment	Derivability in *SD*

There is no syntactic counterpart to either truth-functional falsity or truth-functional indeterminacy. Introducing such counterparts is easy enough—we could define an *anti-theorem* of *SD* as a sentence **P** of *SL* whose negation, ~ **P**, is a theorem of *SD*. And we could take a sentence **P** of *SL* to be *syntactically undetermined* in *SD* if and only if neither it nor its negation is a theorem of *SD*. We would then have syntactic counterparts to all seven central semantic concepts, but historically logicians have never felt the need to add these or equivalent definitions. We will follow their lead.

Below we construct a derivation that establishes that the following simple argument is valid in *SD:*

A ⊃ B
~ B
―――
~ A

Derive: ~ A

1	A ⊃ B	Assumption
2	~ B	Assumption
3	A	A /~ I
4	B	1, 3 ⊃E
5	~ B	2 R
6	~ A	3–5 ~ I

This derivation establishes that the above argument is valid in *SD*. (The conclusion of the argument has been derived from the set consisting of the premises of the argument.)

On the other hand, the following *does not* establish the validity of the above argument:

Derive: ~ A

1	A ⊃ B	Assumption
2	~ B	Assumption
3	~ A	A
4	~ A	3 R

Here '~ A', the conclusion of the argument, has not been derived from the set consisting of the premises of the argument. Rather, it has been derived from those sentences and '~ A'—that is, from the primary assumptions and an auxiliary assumption. We have not shown that '~ A' is derivable from the set consisting of the premises A ⊃ B and ~ A.

Note that no notation has been made on line 3 as to the reason for assuming '~ A'. Someone constructing a derivation such as this may well have reasoned "I want to obtain '~ A'. Since I can assume anything, I will assume what I want, namely '~ A', and then use Reiteration to derive my goal, '~ A'." It is true that any sentence of *SL* can be assumed at any time. But there is no point to assuming a sentence unless one has a rule in mind for discharging that assumption. This is why we require the justification column for auxiliary assumptions to include both the indication that the sentence just entered is an assumption ('A') and an indication of what rule will be used to discharge the assumption. There are only five rules (Conditional Introduction, Disjunction Elimination, Negation Introduction, Negation Elimination, and Biconditional Introduction) that require the construction of a subderivation. These are also the only rules that involve discharging an assumption. Requiring a notation that indicates what rule will be used to discharge an assumption largely prevents the making of assumptions that do not serve a strategic purpose.

A theorem of *SD* is a sentence of *SL* that can be derived from no primary assumptions. A derivation of such a sentence is said to be a **proof** of that sentence. Here is a proof of the theorem '[A ⊃ (B ⊃ C)] ⊃ [(A & B) ⊃ C]':

Derive: [A ⊃ (B ⊃ C)] ⊃ [(A & B) ⊃ C]

1	A ⊃ (B ⊃ C)	A / ⊃I
2	A & B	A / ⊃I
3	A	2 &E
4	B ⊃ C	1, 3 ⊃E
5	B	2 &E
6	C	4, 5 ⊃E
7	(A & B) ⊃ C	2–6 ⊃I
8	[A ⊃ (B ⊃ C)] ⊃ [(A & B) ⊃ C]	1–7 ⊃I

There are no primary assumptions in this derivation, and every auxiliary assumption has been discharged. The sentence '$[A \supset (B \supset C)] \supset [(A \& B) \supset C]$' on the last line does not lie within the scope of any assumption. Hence it has been derived from the empty set and is a theorem of *SD*.

As one would expect, the sentences '$A \equiv B$' and '$B \equiv A$' are equivalent in *SD,* as the following two derivations show. Establishing the equivalence in *SD* of two distinct sentences of *SL* requires *two* derivations because we must establish that each sentence is derivable from the unit set of the other.

Derive: $B \equiv A$

1	$A \equiv B$	Assumption
2	B	A / $\equiv$I
3	A	1, 2 $\equiv$E
4	A	A / $\equiv$I
5	B	1, 4 $\equiv$I
6	$B \equiv A$	2–3, 4–5 $\equiv$I

Having derived '$B \equiv A$' from '$A \equiv B$', we next derive '$A \equiv B$' from '$B \equiv A$'.

Derive: $A \equiv B$

1	$B \equiv A$	Assumption
2	A	A / $\equiv$I
3	B	1, 2 $\equiv$E
4	B	A / $\equiv$I
5	A	1, 4 $\equiv$I
6	$A \equiv B$	2–3, 4–5 $\equiv$I

These two derivations establish that '$A \equiv B$' and '$B \equiv A$' are equivalent in *SD*.

When **P** and **Q** are distinct sentences we need two derivations to show that they are equivalent, one of **Q** from {**P**} and one of **P** from {**Q**}. But when **P** and **Q** are identical, the same sentence, we need only one derivation to show they are equivalent (the one sentence is equivalent to itself) because in this case the derivation of **Q** from {**P**} is also a derivation of **P** from {**Q**}. A sentence can be derived from its own unit set in just one step, using Reiteration:

1	**P**	Assumption
2	**P**	1 Reiteration

5.3 STRATEGIES FOR CONSTRUCTING DERIVATIONS IN *SD*

Derivations are unlike truth-tables and truth-trees in two important respects. First, when one of the syntactic properties we have defined holds (for a sentence, a pair of sentences, an argument, etc.) there is a derivation that demonstrates that this property holds. For example, if an argument is valid in *SD* it is the existence of a derivation of the conclusion of the argument from the set consisting of the argument's premises that makes this so. But if an argument is invalid in *SD* there is no derivation that demonstrates this. Rather, it is the *absence* of a derivation that makes an argument invalid in *SD*. While one can use the derivation system *SD* to show that there is a derivation of a certain sort (by producing such a derivation), one cannot use it to show that there is no derivation of a certain sort. No number of unsuccessful attempts to construct a derivation of a certain sort proves that there is no such derivation. Hence, the system *SD* can be used to establish validity in *SD,* but not invalidity. So too for equivalence in *SD,* inconsistency in *SD,* and theoremhood in *SD*. That is, one cannot use the system *SD* to prove that the members of a pair of sentences are not equivalent in *SD,* that a set is consistent in *SD,* or that a sentence is not a theorem in *SD*. In this way the derivation system is unlike truth-tables and truth-trees, for those procedures are able to establish, for each key semantic concept of *SL,* whether that concept holds or does not hold for a sentence or set of sentences of *SL*.

A second important difference between truth-tables and truth-trees and derivations is that while it is fairly easy to see how an explicit procedure can be developed for constructing truth-tables and truth-trees such that following the procedure does not call for making any choices and always results in a truth-table or truth-tree that yields an answer to the question being asked (*e.g.,* is this set truth-functionally consistent?), it is considerably harder to specify such an explicit procedure for constructing derivations. Procedures that do determine every step of the construction process, whether for truth-tables, trees, or derivations, are said to be **mechanical** procedures. While mechanical procedures for constructing derivations in systems like *SD* (derivation systems for sentential logic)—procedures that will always produce a derivation of a certain sort when one does exist—have been formulated, they are very complex and we will make no attempt to present such a procedure here.[2] There are thus two ways in which one's efforts to construct a derivation of a certain sort might end in frustration—where there is no such derivation and where there is one but all attempts one makes to find it fail. Of course these are very different situations; the first results from trying to do what is impossible, the second from failing to find a solution that does exist.

While we will not present a mechanical procedure for constructing derivations we will provide some useful strategies, strategies that can help avoid

[2]These procedures are generally called theorem provers because what the procedure does, in the first instance, is give mechanical instructions for constructing a proof of a theorem. These procedures are very complicated. It is also important to note that such procedures, when applied to a sentence that is not a theorem of the system, will produce no result that shows the sentence in question is not a theorem.

frustration of the second sort just alluded to. The overarching strategy is that of goal analysis. In every derivation the goal is to derive a sentence, or sentences, from primary assumptions where there are such, otherwise from no assumptions. Goal analysis is the process of determining how a goal sentence can be derived, and involves working backward from the intended last line of the derivation as well as forward from the primary assumptions, if any, of the derivation.

No matter what the goal sentence is, the derivation step that produces that sentence might be the application of any of the elimination rules. To see this one need only remember that the elimination rules tell us nothing about the derived sentence—in each case it might be an atomic sentence, a conjunction, a disjunction, a conditional, a negation, or a biconditional. On the other hand, the introduction rules do tell us a lot about the sentence derived by using one of these rules. First, atomic sentences cannot be derived by using an introduction rule, for all such rules produce truth-functionally compound sentences. Second, we know, for each introduction rule, what the main connective is of a sentence obtained by that rule. Conjunction Introduction produces conjunctions, Disjunction Introduction disjunctions, and so on.

The first step in goal analysis is therefore to determine what kind of a sentence the goal sentence is. If it is an atomic sentence it must be obtained by one of the elimination rules (or by Reiteration). If it is a truth-functional compound sentence it might be obtained by any of the elimination rules or by the appropriate introduction rule, namely the introduction rule that produces sentences whose main connective is the main connective of the goal sentence. The bottom line, of course, is that there will always be multiple ways in which the goal sentence might be derived. But some ways will generally be more plausible than others, as we will soon see.

Having picked one way in which a goal sentence can be obtained, the next step is to determine whether this way of obtaining the goal sentence generates one or more new goal sentences, and then to ask of each of these how they might be obtained. The idea is that eventually the rule picked as a way of obtaining the current goal can be applied directly to currently available sentences, thus completing the derivation. Multiple examples will, we hope, make all of this much clearer.

We here enumerate the strategies we will use throughout the rest of this chapter:

- If the current goal sentence can be obtained by Reiteration, use that rule, otherwise
- If the current goal sentence can be obtained by using a non-subderivation rule, or a series of such rules, do so; otherwise
- Try to obtain the goal sentence by using an appropriate subderivation rule.
- When using a negation rule, try to use an already accessible negation (if there is one) as the ~ Q that the negation rules require be derived.

Most of the derivations we will construct and most of the derivations called for by exercises will involve using multiple strategies, as most will involve deriving one or more subgoals before the final goal can be derived. In practice, this means that most of the time in constructing derivations we work both from the bottom up and from the top down. That is, we work from the bottom up by noting what sentence or sentences we need to obtain before we can obtain the sentence we are trying to derive, and make them subgoals. We work from the top down by deriving from accessible lines and subderivations sentences that will be useful in obtaining our final goal sentence. We will shortly work through the construction of derivations that illustrate this process. Finally, it will often be the case that two or more strategies appear to be viable ways of obtaining a goal sentence. For example, if the current goal sentence is a material conditional and one of the accessible sentences is a disjunction, then both Disjunction Elimination and Conditional Introduction suggest themselves as possible strategies. Rather than puzzling over which strategy is most likely to succeed or which will produce the shortest derivation it is often wise to just pick one and pursue it.

Suppose we are trying to derive '(A & B) ⊃ C' from {A ⊃ C}. The derivation will obviously have just one primary assumption. So we start work as follows:

Derive: (A & B) ⊃ C

1	A ⊃ C	Assumption
G	(A & B) ⊃ C	

Our current goal is the sentence '(A & B) ⊃ C'. We have indicated this by writing 'G' where a line number will eventually be placed. We will follow this convention, of indicating goal sentences by writing 'G' where the number of the line will eventually be, throughout the rest of this section. Readers should follow this convention when constructing their own derivations only if they are working in pencil and can erase these goal sentence markers and replace them with line numbers as appropriate. We write this goal sentence a substantial distance below the primary assumptions, because we do not know, at this stage, how many steps it will take to derive this sentence. At this early stage we know neither the line number nor the justification for the final line of the derivation. We note that the goal sentence is a material conditional. Hence, in principle it could come by any one of the elimination rules, by Reiteration, or by Conditional Introduction. Reiteration is not plausible, as the goal sentence is not among the primary assumptions (there is only one). An elimination rule is not a likely way of generating the goal sentence because the only accessible sentence is the conditional on line 1 and Conditional Elimination requires that we have both a conditional and the antecedent of that conditional. In this case we do not have the antecedent of ‘A ⊃ C’, and even if we did the result of applying Conditional Elimination would be ‘C’, not ‘(A & B) ⊃ C’. So Conditional Introduction seems to be the most likely rule to have produced our goal sentence. We now note that to use Conditional Introduction we need a subderivation whose assumption is the antecedent of our goal sentence, namely ‘A & B’, and we need to derive the consequent of our goal sentence, ‘C’, within the scope of that assumption. That is, we know our derivation will look like this:

```
Derive: (A & B) ⊃ C

1 | A ⊃ C                          Assumption
  |-------
2 | | A & B                        A / ⊃I
  | |-------
  | |
  | |
G | | C
G | (A & B) ⊃ C                    2–__ ⊃I
```

We still do not know the line number of the last line of our derivation, but we do know we will use Conditional Introduction to obtain it and that we will cite a subderivation that begins on line 2. We note this in the justification column for the last line by entering ‘2–__ ⊃I’ where the underscore marks the space where we will subsequently enter the number of the preceding line. We also know that line 2 will be an auxiliary assumption made for the purpose of doing Conditional Introduction. We are now in a position to stop wondering how ‘(A & B) ⊃ C’ will be obtained. We have a strategy for obtaining that sentence, Conditional Introduction. Accordingly we now switch our focus to how we can complete the subderivation we have started. That is, how can we get from our two assumptions, one primary and one auxiliary, to ‘C’? ‘C’ is an atomic sentence, so we know we will not use an introduction rule to obtain this sentence. Nor will Reiteration generate ‘C’. So we are left with the elimination rules. Which elimination rule seems promising? Here it is important to learn to “see” what is available to us at this point in our work. We have two sentences to work from, ‘A ⊃ C’ and ‘A & B’. We want ‘C’. We know that ‘C’ can be obtained from ‘A ⊃ C’ by Conditional Elimination *if* we have ‘A’. We do not currently have ‘A’. But we do have ‘A & B’, and ‘A’ can be obtained from ‘A & B’ by Conjunction Elimination. So we now see a path to the completion of our derivation:

```
Derive: (A & B) ⊃ C

1 | A ⊃ C                          Assumption
  |-------
2 | | A & B                        A / ⊃I
  | |-------
3 | | A                            2 &E
4 | | C                            1, 3 ⊃E
5 | (A & B) ⊃ C                    2–4 ⊃I
```

We will spend the rest of this section illustrating how the strategies we have enumerated can be used to construct derivations. We will first construct derivations that establish validity in *SD*, then ones that establish that a sentence is a theorem in *SD*, then ones that establish the equivalence in *SD* of a pair of sentences, and finally ones that establish inconsistency in *SD*. We again note that while derivations can be used to establish such results, they cannot be used to establish that an argument is invalid in *SD*, that a pair of sentences are not equivalent in *SD*, or that a set of sentences is consistent in *SD*. Nor, except in special cases, can derivations be used to show that a sentence is not a theorem in *SD*.

ARGUMENTS

Consider next the following argument.

$$\begin{array}{l} \sim N \\ \underline{(\sim N \supset L)\ \&\ [D \equiv (\sim N \vee A)]} \\ L\ \&\ D \end{array}$$

To show that this argument is valid in *SD* we need to derive the conclusion from the set consisting of the premises. So we start as follows:

Derive: L & D

1	~ N	Assumption
2	(~ N ⊃ L) & [D ≡ (~ N ∨ A)]	Assumption
G	L & D	—, — &I

Our goal is a conjunction. It seems unlikely that it will be obtained by an elimination rule, in part because ‘L & D’ does not occur as a component of any accessible sentence. An introduction rule seems more promising, and since the main connective of our goal sentence is ‘&’ it is Conjunction Introduction that seems most promising. We have noted this by writing ‘&I’ in the justification column for our goal sentence, and we have indicated with two underscores that two line numbers will need to be supplied later. If we are to use Conjunction Introduction we will need to have the two conjuncts ‘L’ and ‘D’ available on accessible earlier lines. So we now add two **subgoals** to our derivation structure:

Derive: L & D

1	$\sim N$	Assumption
2	$(\sim N \supset L)\ \&\ [D \equiv (\sim N \vee A)]$	Assumption
G	L	
G	D	
G	$L\ \&\ D$	—, — &I

If we can obtain both ‘L’ and ‘D’ we can use Conjunction Introduction to obtain ‘L & D’. Our new goal sentences, ‘L’ and ‘D’ are both atomic sentences, so neither will come by an introduction rule. We note that ‘L’ occurs as the consequent of a conditional embedded in our second primary assumption. If we could get that conditional, ‘$\sim N \supset L$’, out of line 2 we could obtain ‘L’ by Conditional Elimination, as we do have the antecedent of that conditional ‘$\sim N$’ at line 1. Conjunction Elimination does allow us to extract ‘$\sim N \supset L$’ from line 2:

Derive: L & D

1	$\sim N$	Assumption
2	$(\sim N \supset L)\ \&\ [D \equiv (\sim N \vee A)]$	Assumption
3	$\sim N \supset L$	2 &E
4	L	1, 3 $\supset$E
G	D	
G	$L\ \&\ D$	4, — &I

The remaining task, then, is to obtain ‘D’. We note that this sentence occurs in the biconditional embedded in line 2. Since the main connective of the sentence on line 2 is ‘&’, we can obtain the biconditional by Conjunction Elimination. To get ‘D’ from that biconditional we can use Biconditional Elimination, if we have ‘$\sim N \vee A$’. This reasoning allows us to add the following steps to our derivation:

Derive: L & D

1	$\sim N$	Assumption
2	$(\sim N \supset L)\ \&\ [D \equiv (\sim N \vee A)]$	Assumption
3	$\sim N \supset L$	2 &E
4	L	1, 3 $\supset$E
5	$D \equiv (\sim N \vee A)$	2 &E
G	$\sim N \vee A$	
G	D	5, — $\equiv$E
G	$L\ \&\ D$	4, — &I

Note that we have added '~ N ∨ A' as a new goal sentence. The main connective of this sentence is '∨', so if we had either '~ N' or 'A' we could obtain our current goal by Disjunction Introduction. As it happens, we do have '~ N'—it occurs as a primary assumption on line 1. So we can now complete our derivation.

Derive: L & D

1	~ N	Assumption
2	(~ N ⊃ L) & [D ≡ (~ N ∨ A)]	Assumption
3	~ N ⊃ L	2 &E
4	L	1, 3 ⊃E
5	D ≡ (~ N ∨ A)	2 &E
6	~ N ∨ A	1 ∨I
7	D	5, 6 ≡E
8	L & D	4, 7 &I

We will next show that the following argument is valid in *SD* by deriving its conclusion from the set consisting of its premises.

~ A ∨ B

~ A ⊃ B

B ≡ C

C

We begin as always, by taking the premises as primary assumptions and making the conclusion our primary goal.

Derive: C

1	~ A ∨ B	Assumption
2	~ A ⊃ B	Assumption
3	B ≡ C	Assumption
G	C	

After some reflection, two strategies suggest themselves: using Negation Elimination to obtain 'C' and using Disjunction Elimination to obtain 'C'. Both will, in the end, work. We choose to use Disjunction Elimination.

Derive: C

1	~ A ∨ B	Assumption
2	~ A ⊃ B	Assumption
3	B ≡ C	Assumption
4	~ A	A / ∨E
G	C	
	B	A / ∨E
G	C	
G	C	1, 4–__, __–__ ∨E

Our strategy, as the above schema indicates, is to show that the conclusion of the argument, 'C', can be derived from each disjunct of '~ A ∨ B', and hence that 'C' itself can be obtained by Disjunction Elimination. Completing the second subderivation is trivial, for 'C' can be obtained from line 3 and our second auxiliary assumption by Biconditional Elimination.

Derive: C

1	~ A ∨ B	Assumption
2	~ A ⊃ B	Assumption
3	B ≡ C	Assumption
4	~ A	A / ∨E
G	C	
	B	A / ∨E
G	C	3, __ ≡E
G	C	1, 4–__, __–__ ∨E

Completing the first subderivation is only slightly more challenging. From lines 4 and 2 we can obtain 'B' by Conditional Elimination. And we can then use Biconditional Elimination to obtain 'C'.

Derive: C

1	~ A ∨ B	Assumption
2	~ A ⊃ B	Assumption
3	B ≡ C	Assumption
4	~ A	A / ∨E
5	B	2, 4 ⊃E
6	C	3, 5 ≡E
7	B	A / ∨E
8	C	3, 7 ≡E
9	C	1, 4–6, 7–8 ∨E

THEOREMS

Next we will construct proofs of several theorems. We start with a very obvious theorem, 'A ∨ ~ A', whose proof is not obvious. Our task is to derive this sentence using no primary assumptions.

Derive: A ∨ ~ A

1		
G	A ∨ ~ A	

Our goal is 'A ∨ ~ A' and here it should be obvious that though this sentence is a disjunction we will not be able to obtain it by Disjunction Introduction. Neither 'A' nor '~ A' is a theorem, and neither can be derived given no primary assumptions. So the only sensible strategy is to use Negation Elimination.

Derive: A ∨ ~ A

1	~ (A ∨ ~ A)	A / ~ E
G	A ∨ ~ A	1–__ ~ E

Note that the only accessible sentence, the sentence on line 1, is a negation. There is no rule of *SD* that allows us to 'take apart' a negation. In the present context, we can use Reiteration on line 1, but there is little else we can do with it. Fortunately, this will be useful. Our current strategy is to use Negation Elimination and to do so we need to derive a sentence and its negation. So we will use the assumption on line 1 as the negation and make 'A ∨ ~ A' our new goal.

```
Derive: A ∨ ~ A

1  | | ~ (A ∨ ~ A)           A / ~ E
   | |____________
   | |
   | |
G  | | A ∨ ~ A
   | | ~ (A ∨ ~ A)           1 R
G  | A ∨ ~ A                 1–__ ~ E
```

We noted above that obtaining the last line of our derivation by Disjunction Introduction will not work because neither 'A' nor '~ A' is a theorem. But our current goal, which is the same sentence as that occurring on the last line of the derivation, is to be obtained with the help of the auxiliary assumption '~ (A ∨ ~ A)', and here it is reasonable to hope to use Disjunction Introduction. We will make 'A' our new goal and try to derive it by Negation Elimination.

```
Derive: A ∨ ~ A

1  | | ~ (A ∨ ~ A)           A / ~ E
   | |____________
2  | | | ~A                  A / ~ E
   | | |____
   | | |
   | | |
G  | | A                     2–__ ~E
G  | | A ∨ ~ A               — ∨I
G  | | ~ (A ∨ ~ A)           1 R
G  | A ∨ ~ A                 1–__ ~ E
```

One of the points we have emphasized is that when using a Negation Elimination subderivation it is wise to use as the ~ **Q** a negation that is readily available. In the present instance two negations are readily available, '~ A' and '~ (A ∨ ~ A)'. There may be a temptation to select '~ A' as ~ **Q**. But this would be a mistake, for doing so would require that **Q** be 'A' and that sentence is not readily derived from the available assumptions. (We should take a hint from the fact that the point of our current subderivation is to obtain 'A'. If there were an easy way to obtain it we would not be involved in the current Negation Elimination subderivation.) But if we take ~ **Q** to be '~ (A ∨ ~ A)' then our new goal becomes 'A ∨ ~ A' and this sentence is readily derived—by applying Disjunction Introduction to line 2. We are now able to complete the derivation.

Derive: A ∨ ~ A

```
1 │ │ ~ (A ∨ ~ A)               A / ~ E
  │ ├──────────
2 │ │ │ ~ A                     A / ~ E
  │ │ ├──────
3 │ │ │ A ∨ ~ A                 2 ∨I
4 │ │ │ ~ (A ∨ ~ A)             1 R
5 │ │ A                         2–4 ~ E
6 │ │ A ∨ ~ A                   5 ∨I
7 │ │ ~ (A ∨ ~ A)               1 R
8 │ A ∨ ~ A                     1–7 ~ E
```

Next we will prove the theorem '~ (A ∨ B) ≡ (~ A & ~ B)'. This theorem is a biconditional, so it is plausible the last line will come from Biconditional Introduction, and that rule requires two subderivations, one in which we derive '~ A & ~ B' from {~ (A ∨ B)} and the other in which we derive '~ (A ∨ B)' from {~ A & ~ B} .

Derive: ~ (A ∨ B) ≡ (~ A & ~ B)

```
1 │ │ ~ (A ∨ B)                 A / ≡I
  │ ├──────────
  │ │
  │ │
G │ │ ~ A & ~ B
  │ │ ~ A & ~ B                 A / ≡I
  │ ├──────────
  │ │
  │ │
G │ │ ~ (A ∨ B)
G │ ~(A ∨ B) ≡ (~ A & ~ B)      1–__, __–__ ≡I
```

We now have two goals, '~ A & ~ B' in the first subderivation and '~ (A ∨ B)' in the second subderivation. We will work on the upper subderivation first. Since our goal is a conjunction, we will take as new subgoals the two conjuncts of that conjunction, '~ A' and '~ B', and attempt to derive each by Negation Introduction.

Derive: ~ (A ∨ B) ≡ (~ A & ~ B)

```
1   | | ~ (A ∨ B)               A / ≡I
    | |----------
2   | | | A                     A / ~ I
    | | |---
3   | | | A ∨ B                 2 ∨I
4   | | | ~ (A ∨ B)             1 R
5   | | ~ A                     2–4 ~ I
6   | | | B                     A / ~ I
    | | |---
7   | | | A ∨ B                 6 ∨I
8   | | | ~ (A ∨ B)             1 R
9   | | ~ B                     6–8 ~ I
10  | | ~ A & ~ B               5, 9 &I
11  | | ~ A & ~ B               A / ≡I
    | |----------
    | |
    | |
G   | | ~ (A ∨ B)
G   | ~ (A ∨ B) ≡ (~ A & ~ B)   1–10, 11–__ ≡I
```

Note that within the first of our two main subderivations we twice use Negation Introduction, and in each case use 'A ∨ B' and '~ (A ∨ B)' as **Q** and ~ **Q**.

Completing our second main subderivation requires deriving '~ (A ∨ B)', and this invites a Negation Introduction subderivation, giving us a new assumption, 'A ∨ B', which in turn invites a Disjunction Elimination strategy:

```
11  | | ~ A & ~ B               A / ≡I
    | |----------
12  | | | A ∨ B                 A / ~ I
    | | |------
13  | | | | A                   A / ∨E
    | | | |---
    | | | |
    | | | | B
    | | | |---
    | | | |
    | | |
G   | | ~ (A ∨ B)               12–__ ~ I
G   | ~ (A ∨ B) ≡ (~ A & ~ B)   1–10, 11–__ ≡I
```

The question now is what sentence we want to play the role of '**R**' in our Disjunction Elimination subderivation. We need a sentence and its negation to make our Negation Introduction subderivation, begun at line 12, work. Two negations are readily available, '~ A' and '~ B'. So we will arbitrarily select one of these, say '~ B', and then make 'B' the sentence we try to obtain by Disjunction Elimination:

```
11 | | ~ A & ~ B                    A / ≡I
   | |----------
12 | | | A ∨ B                      A / ~ I
   | | |------
13 | | | | A                        A / ∨E
   | | | |---
   | | | |
   | | | |
 G | | | | B
   | | |
   | | | | B                        A / ∨E
   | | | |---
 G | | | | B                        ___ R
 G | | | B                          12, 13–___, ___–___ ∨E
 G | | | ~ B                        11 &E
 G | | ~ (A ∨ B)                    12–___ ~ I
 G | ~ (A ∨ B) ≡ (~ A & ~ B)        1–10, 11–___ ≡I
```

We now have two subderivations to complete. The second is, in fact, already complete, for it involves deriving 'B' from an auxiliary assumption of 'B', so Reiteration will accomplish the task. The first involves deriving 'B' from the assumptions on lines 11 through 13. Fortunately a sentence, 'A', and its negation, '~ A' are both readily available. So Negation Elimination will yield the desired result:

```
11 | | ~ A & ~ B                    A / ≡I
   | |----------
12 | | | A ∨ B                      A / ~ I
   | | |------
13 | | | | A                        A / ∨E
   | | | |---
14 | | | | | ~ B                    A / ~ E
   | | | | |----
15 | | | | | ~ A                    11 &E
16 | | | | | A                      13 R
17 | | | | B                        14–16 ~ E
   | | |
18 | | | | B                        A / ∨E
   | | | |---
19 | | | | B                        18 R
20 | | | B                          12, 13–17, 18–19 ∨E
21 | | | ~ B                        11 &E
22 | | ~ (A ∨ B)                    12–21 ~ I
23 | ~ (A ∨ B) ≡ (~ A & ~ B)        1–10, 11–22 ≡I
```

This completes our proof of the theorem '~ (A ∨ B) ≡ (~ A & ~ B)'.

We will conclude our discussion of theorems by constructing a proof of what has become known as Peirce's Law.[3]

[(A ⊃ B) ⊃ A] ⊃ A

Since the theorem is a conditional it is plausible that we will be using Conditional Introduction as our primary strategy.

```
Derive: [(A ⊃ B) ⊃ A] ⊃ A

1  | | (A ⊃ B) ⊃ A              A / ⊃I
   | |------------
   | |
G  | | A
G  | [(A ⊃ B) ⊃ A] ⊃ A          1–__ ⊃I
```

But how we should proceed next may not be obvious. We could derive our current goal, 'A', from line 1 by Conditional Elimination *if* we also had 'A ⊃ B', but we do not. So perhaps we should take the sentence 'A ⊃ B' as our new goal, and try to obtain it by Conditional Introduction.

```
Derive: [(A ⊃ B) ⊃ A] ⊃ A

1  | | (A ⊃ B) ⊃ A              A / ⊃I
   | |------------
2  | | | A                      A / ⊃I
   | | |---
   | | |
G  | | | B
G  | | A ⊃ B                    2–__ ⊃I
G  | | A                        1, __ ⊃E
G  | [(A ⊃ B) ⊃ A] ⊃ A          1–__ ⊃I
```

So far, one might think, so good. But how are we to obtain 'B' from the sentences on lines 1 and 2? We could assume '~ B' and hope to use Negation Elimination.

```
Derive: [(A ⊃ B) ⊃ A] ⊃ A

1  | | (A ⊃ B) ⊃ A              A / ⊃I
   | |------------
2  | | | A                      A / ⊃I
   | | |---
3  | | | | ~ B                  A / ~ E
   | | | |----
   | | | |
G  | | | B
G  | | A ⊃ B                    2–__ ⊃I
G  | | A                        1, __ ⊃E
G  | [(A ⊃ B) ⊃ A] ⊃ A          1–__ ⊃I
```

[3] The first proof of this theorem was given by Charles Peirce, a nineteenth-century American philosopher.

Unfortunately, the only negation now available is '~ B', so it appears that to make Negation Elimination work we will have to derive '~ B' (by Reiteration) and 'B'. But how do we derive 'B'? We seem to be back where we were before we assumed '~ B'. That is, 'B' is again our goal sentence.

We appear to be on the wrong track. Suppose that when we had 'A' as our goal, instead of planning on deriving 'A' by Conditional Elimination we try to derive it by Negation Elimination.

Derive: [(A ⊃ B) ⊃ A] ⊃ A

1	| | (A ⊃ B) ⊃ A	A / ⊃I
2	| | | ~A	A / ~ E
	| | |	
G	| | A	2–___ ~ E
G	| [(A ⊃ B) ⊃ A] ⊃ A	1–___ ⊃I

Since we have a negation available, '~ A', perhaps we should take 'A' and '~ A' as the sentences **Q** and ~ **Q** we need to use Negation Elimination and accordingly make 'A' our new goal. This may seem no more promising than was the line of reasoning recently abandoned, since deriving 'A' was our goal before assuming '~ A'. But we are, in fact, making progress.

Derive: [(A ⊃ B) ⊃ A] ⊃ A

1	| | (A ⊃ B) ⊃ A	A / ⊃I
2	| | | ~ A	A / ~ E
	| | |	
G	| | | A	
	| | | ~ A	2 R
G	| | A	2–___ ~ E
G	| [(A ⊃ B) ⊃ A] ⊃ A	1–___ ⊃I

We can obtain 'A' from line 1 by Conditional Elimination if we can first obtain 'A ⊃ B'. This is, of course, the position we were in at the start of our work. But now we have an additional assumption available to us, namely '~ A'.

Derive: [(A ⊃ B) ⊃ A] ⊃ A

1	(A ⊃ B) ⊃ A	A / ⊃I
2	~ A	A / ~ E
3	A	A / ⊃I
G	B	
G	A ⊃ B	3–__ ⊃I
G	A	1–__ ⊃E
	~ A	2 R
G	A	2–__ ~ E
G	[(A ⊃ B) ⊃ A] ⊃ A	1–__ ⊃I

And now we can see our way to the end. We need 'B' and we have a sentence and its negation readily available ('A' and '~ A'), so we can assume '~ B' and use Negation Elimination. Here is the completed derivation.

Derive: [(A ⊃ B) ⊃ A] ⊃ A

1	(A ⊃ B) ⊃ A	A / ⊃I
2	~ A	A / ~ E
3	A	A / ⊃I
4	~ B	A / ~ E
5	A	3 R
6	~A	2 R
7	B	4–6 ~ E
8	A ⊃ B	3–7 ⊃I
9	A	1, 8 ⊃E
10	~A	2 R
11	A	2–10 ~ E
12	[(A ⊃ B) ⊃ A] ⊃ A	1–11 ⊃I

It is worth noting that in this example, as is frequently the case, a strategy that at first seems obvious (using Conditional Elimination to obtain 'A' as the penultimate line of the derivation) but proves problematic can successfully be used as a secondary strategy inside an alternative strategy (here Negation Elimination).

EQUIVALENCE

Suppose we want to establish that 'A ≡ ~ B' and '~ A ≡ B' are equivalent in *SD* (they are). Two derivations are required, one deriving '~ A ≡ B' from

{A ≡ ~ B} and one deriving 'A ≡ ~ B' from {~ A ≡ B}. Here is a start for the first of these:

Derive: ~ A ≡ B

1	A ≡ ~ B	Assumption
G	~ A ≡ B	

It should be apparent that our goal, '~ A ≡ B', is not going to be obtained by an elimination rule. We have too little to work with by way of primary assumptions for that to be a viable strategy. Since the main connective of our goal sentence is '≡', Biconditional Introduction may be a viable strategy. So we continue our derivation thus:

Derive: ~ A ≡ B

1	A ≡ ~ B	Assumption
2	~ A	A / ≡I
G	B	
	B	A / ≡I
G	~ A	
G	~ A ≡ B	2–__, __–__ ≡I

We now have two subderivations to complete. The goal of the first is 'B', and it can be obtained by Negation Elimination. The goal of the second, '~ A', can be obtained by Negation Introduction:

```
Derive: ~ A ≡ B

 1  │ A ≡ ~ B                       Assumption
    ├────────
 2  │ │ ~ A                         A / ≡I
    │ ├──────
 3  │ │ │ ~ B                       A / ~ E
    │ │ ├────
 4  │ │ │ A                         1, 3 ≡E
 5  │ │ │ ~ A                       2 R
 6  │ │ B                           3–5 ~ E
 7  │ │ B                           A / ≡I
    │ ├──────
 8  │ │ │ A                         A / ~ I
    │ │ ├────
 9  │ │ │ ~ B                       1, 8 ≡E
10  │ │ │ B                         7 R
11  │ │ ~ A                         8–10 ~ I
12  │ ~ A ≡ B                       2–6, 7–11 ≡I
```

The second half of our current task is to derive 'A ≡ ~ B' from {~A ≡ B}.

```
Derive: A ≡ ~ B

 1  │ ~ A ≡ B                       Assumption
    ├────────
    │
    │
 G  │ A ≡ ~ B
```

Biconditional Introduction is also a good strategy in this case.

```
Derive: A ≡ ~ B

 1  │ ~ A ≡ B                       Assumption
    ├────────
 2  │ │ A                           A / ≡I
    │ ├──────
    │ │
    │ │
 G  │ │ ~ B                         A / ≡I
    │ │ ~ B
    │ ├──────
    │ │
    │ │
 G  │ │ A
 G  │ A ≡ ~ B                       2–__, __–__ ≡I
```

Here, too, negation strategies will yield the desired results:

Derive: A ≡ ~ B

```
 1  │ ~ A ≡ B               Assumption
    ├──────────
 2  │ │ A                   A / ≡I
    │ ├────────
 3  │ │ │ B                 A / ~ I
    │ │ ├──────
 4  │ │ │ ~ A               1, 3 ≡E
 5  │ │ │ A                 2 R
 6  │ │ ~ B                 3–5 ~ I
 7  │ │ ~ B                 A / ≡I
    │ ├────────
 8  │ │ │ ~ A               A / ~ E
    │ │ ├──────
 9  │ │ │ B                 1, 8 ≡E
10  │ │ │ ~ B               7 R
11  │ │ A                   8–10 ~ E
12  │ A ≡ ~ B               2–6, 7–11 ≡I
```

We next show that ‘A ⊃ B’ and ‘~ A ∨ B’ are equivalent in *SD*. To do so will require deriving each sentence from the unit set of the other. So we will be doing two derivations. Both of these derivations are rather difficult but also highly instructive as they will allow us to illustrate strategies associated with a number of introduction and elimination rules. We set up our first derivation as follows:

Derive: ~ A ∨ B

```
1  │ A ⊃ B                 Assumption
   ├────────
   │
   │
G  │ ~ A ∨ B
```

Our goal sentence is ‘~ A ∨ B’, a disjunction. So we might be tempted to try to obtain our goal by Disjunction Introduction. While this strategy will not work, we will explore it anyway to illustrate how one can fall into unproductive strategies. If we are to use Disjunction Introduction we will need to first obtain either ‘~ A’ or ‘B’. We will take ‘B’ as our new goal. (In fact, neither ‘B’ nor ‘~ A’ is obtainable given just ‘A ⊃ B’.)

Derive: ~ A ∨ B

```
1  │ A ⊃ B                 Assumption
   ├────────
   │
   │
G  │ B
G  │ ~ A ∨ B               __ ∨I
```

Since our goal is now 'B', and we have 'A ⊃ B' at line 1, it might seem like a good idea to assume 'A' and then use Conditional Elimination to obtain 'B'.

Derive: ~ A ∨ B

1	A ⊃ B	Assumption	
2	A	A	
3	B	1, 2 ⊃E	
4	B	3 R	**MISTAKE!**
5	~ A ∨ B	4 ∨ I	

Line 4 is a mistake because it appeals to a sentence, 'B', on line 3 that is not accessible at line 4. There is a scope line to the left of 'B' at line 3 that does not continue through line 4. We had two chances to avoid going down this path to a mistake. First, thinking we could get '~ A ∨ B' by first deriving 'B' from the assumption on line 1 was a bad idea. That assumption is 'A ⊃ B'. We are trying to show that 'A ⊃ B' and '~ A ∨ B' are equivalent in *SD,* as indeed they are. Although we are here concerned with syntactic properties of sentences and sets of sentences, it is well to remember that for any set Γ of sentences of *SL* and any sentence **P** of *SL*,

Γ ⊢ **P** in *SD* if and only if Γ ⊨ **P**.

Our ill-advised strategy involved trying to show that

{A ⊃ B} ⊢ B

where in fact 'B' is not derivable from {A ⊃ B}. For if this derivability claim did hold then it would also have to be the case that

{A ⊃ B} ⊨ B

and this claim is false. There are truth-value assignments on which 'A ⊃ B' is true and 'B' false, namely every truth-value assignment on which 'A' and 'B' are both assigned **F**.

We had a second chance to avoid going down an unpromising road when we assumed 'A' at line 2. Note that there is nothing in the justification column for line 2 indicating why we are making this assumption. Had we been paying attention at that time we would have realized that we have no good reason for assuming 'A'. There is no subderivation strategy that will allow us to assume 'A', derive some sentence or sentences, and then end the subderivation and enter 'B' as the next line.

A more promising strategy for completing our first derivation, though one that does not initially come to mind when one is first learning to do derivations, is to use Negation Elimination to obtain '~ A ∨ B'.

Derive: ~ A ∨ B

1	A ⊃ B	A / ≡I
2	~ (~ A ∨ B)	A / ~ E
G	~ A ∨ B	2–__ ~ E

This strategy will seem unpromising if one thinks either that the **Q** and ~ **Q** that need to be derived to use a negation rule must be an atomic sentence and its negation, or that a negation must be among or easily obtained from the sentences that are accessible before one makes the auxiliary assumption that begins a negation subderivation. Neither is the case. The **Q** and ~ **Q** that both negation rules require deriving can be a compound sentence and its negation as well as an atomic sentence and its negation. And the ~ **Q** that is derived can occur as the auxiliary assumption that initiates the negation subderivation. Keeping this in mind we proceed as follows:

Derive: ~ A ∨ B

1	A ⊃ B	Assumption
2	~ (~ A ∨ B)	A / ~ E
G	~ A ∨ B	
	~ (~ A ∨ B)	2 R
G	~ A ∨ B	2–__ ~ E

It certainly might appear that we are making no progress. The goal of this derivation is '~ A ∨ B'. And this same sentence is now our goal within the subderivation begun at line 2. But in fact we are making progress. We noted earlier that we cannot derive '~ A ∨ B' by Disjunction Introduction when the only accessible sentence is 'A ⊃ B'. But we now have two accessible sentences to appeal to, those at lines 1 and 2. If we can use these two assumptions to derive '~ A', we can obtain our current goal, '~ A ∨ B' by Disjunction Introduction. This suggests we try to obtain '~ A' by Negation Introduction.

Derive: ~ A ∨ B

```
1   | A ⊃ B                         Assumption
    |------
2   |  | ~ (~ A ∨ B)                A / ~ E
    |  |-----------
3   |  |  | A                       A / ~ I
    |  |  |---
    |  |  |
    |  |  |
    |  |  |
G   |  | ~ A                        3–__ ~ I
G   |  | ~ A ∨ B
    |  | ~ (~ A ∨ B)                2 R
G   | ~ A ∨ B                       2–__ ~ E
```

We are again at the point where it is essential to be able to 'see' what we can obtain from the sentences that are accessible at the point where we are working (inside the subderivation that we began at line 3). The accessible sentences are those on lines 1–3. At line 3 we have 'A'. At line 1 we have 'A ⊃ B'. From these two sentences we can obtain 'B' by Conditional Elimination. From 'B' we can obtain '~ A ∨ B' by Disjunction Introduction, and we can derive the negation of this sentence, '~ (~ A ∨ B)' by Reiteration on line 2. These steps will complete the first half of our current task, that of showing that 'A ⊃ B' and '~ A ∨ B' are equivalent in *SD*.

Derive: ~ A ∨ B

```
1   | A ⊃ B                         Assumption
    |------
2   |  | ~ (~ A ∨ B)                A / ~ E
    |  |-----------
3   |  |  | A                       A / ~ I
    |  |  |---
4   |  |  | B                       1, 3 ⊃E
5   |  |  | ~ A ∨ B                 4 ∨I
6   |  |  | ~ (~ A ∨ B)             2 R
7   |  | ~ A                        3–6 ~ I
8   |  | ~ A ∨ B                    7 ∨I
9   |  | ~ (~ A ∨ B)                2 R
10  | ~ A ∨ B                       2–9 ~ E
```

This derivation of '~ A ∨ B' from 'A ⊃ B' is instructive in several ways. First, given that a disjunction is derivable, it does not follow that the last step in that derivation is Disjunction Introduction. Second, in picking a goal sentence it is wise to consider whether it is plausible that the selected sentence is derivable from the currently accessible sentences. Third, when using a negation rule the **Q** and ~ **Q** to be derived within the scope of the assumption called for by the rule may well both be compound sentences. Fourth, it does sometimes happen that one sentence is a goal in multiple parts of a derivation. Fifth, in using a negation rule it is advisable to use as ~ **Q** a sentence that is readily available, and it may be available as the assumption of the very subderivation in which we are working. Finally, there is nothing wrong with using two or more instances of negation rules within which the same sentences (on different lines) play the roles of **Q** and ~ **Q**.

The second part of our proof that ‘A ⊃ B’ and ‘~ A ∨ B’ are equivalent in *SD,* a derivation of ‘A ⊃ B’ from {~ A ∨ B}, is also instructive.

Derive: A ⊃ B

1	~ A ∨ B	Assumption
G	A ⊃ B	

We now need a strategy for getting from ‘~ A ∨ B’ to ‘A ⊃ B’. A little reflection suggests two alternative strategies. Since the goal sentence is a material conditional, we could use Conditional Introduction, and accordingly assume ‘A’ at line 2 for the purpose of using Conditional Introduction. Alternatively, since the only accessible sentence, the one at line 1, is a disjunction, we could plan to work to the conditional we want by using Disjunction Elimination. That is, in this case we can either let our goal sentence drive our strategy, working from the bottom up, or we can let our one accessible sentence drive our strategy, working from the top down. Here, as is often the case, both strategies will work. Moreover, whichever strategy we pick as our primary strategy we will end up using the other strategy within the first strategy. This is also often the case. Picking Disjunction Elimination as our primary strategy yields the following:

Derive: A ⊃ B

1	~ A ∨ B	Assumption
2	| ~ A	A / ∨E
	|	
G	| A ⊃ B	
	| B	A / ∨E
	|	
G	| A ⊃ B	
G	A ⊃ B	1, 2–__, __–__ ∨E

Lines 1 and 2, by themselves, don’t suggest a strategy for deriving ‘A ⊃ B’. But ‘A ⊃ B’ is a material conditional and this suggests we use Conditional Introduction to obtain it.

Derive: A ⊃ B

1	~ A ∨ B	Assumption
2	~ A	A / ∨E
3	A	A / ⊃I
G	B	
G	A ⊃ B	3–__ ⊃I
	B	A / ∨E
G	A ⊃ B	
G	A ⊃ B	1, 2–__, __–__ ∨E

Our goal within the subderivation beginning on line 3 is 'B'. We now note that the three accessible sentences include both 'A' and '~ A'. Their availability invites a negation strategy. To obtain 'B' we thus assume '~ B' and derive 'A' and '~ A', both by Reiteration.

Derive: A ⊃ B

1	~ A ∨ B	Assumption
2	~ A	A / ∨E
3	A	A / ⊃I
4	~ B	A / ~ E
5	A	3 R
6	~ A	2 R
7	B	4–6 ~ E
8	A ⊃ B	3–7 ⊃I
9	B	A / ∨E
G	A ⊃ B	
G	A ⊃ B	1, 2–8, 9–__ ∨E

What remains is to derive 'A ⊃ B' from 'B'. This is actually quite easy. We can use Conditional Introduction, assuming 'A' and deriving 'B' by Reiteration on line 9.

Derive: A ⊃ B

```
1  | ~ A ∨ B              Assumption
   |--------
2  | | ~ A                A / ∨E
   | |-----
3  | | | A                A / ⊃I
   | | |---
4  | | | | ~B             A / ~ E
   | | | |----
5  | | | | A              3 R
6  | | | | ~ A            2 R
7  | | | B                4–6 ~ E
8  | | A ⊃ B              3–7 ⊃I
9  | | B                  A / ∨E
   | |---
10 | | | A                A / ⊃I
   | | |---
11 | | | B                9 R
12 | | A ⊃ B              10–11 ⊃I
13 | A ⊃ B                1, 2–8, 9–12 ∨E
```

We have derived '~ A ∨ B' from {A ⊃ B} and 'A ⊃ B' from {~ A ∨ B}, thus demonstrating that these sentences are equivalent in *SD*. Two important lessons about material conditionals are illustrated in our last derivation. The first is that a conditional can be derived from the negation of its antecedent, as we did in lines 2 through 8 above. The second is that a material conditional can be derived from its consequent as we did in lines 9–12 above.

In our last derivation we used Disjunction Elimination as our primary strategy. Using Conditional Introduction as the primary strategy works just as well:

Derive: A ⊃ B

```
1  | ~ A ∨ B              Assumption
   |--------
2  | | A                  A / ⊃I
   | |---
3  | | | ~ A              A / ∨E
   | | |-----
4  | | | | ~ B            A / ~ E
   | | | |-----
5  | | | | A              2 R
6  | | | | ~ A            3 R
7  | | | B                4–6 ~ E
8  | | | B                A / ∨E
   | | |---
9  | | | B                8 R
10 | | B                  1, 3–7, 8–9 ∨E
11 | A ⊃ B                2–10 ⊃I
```

INCONSISTENCY

We will conclude our illustration of strategies for constructing derivations in *SD* by doing several derivations that demonstrate the inconsistency of given sets. Consider first the set {~ (A ⊃ B), B}. To show this set is inconsistent in *SD* we need to derive from it some sentence **Q** and its negation ~ **Q**. In planning a strategy it helps to remember that **Q** need not be an atomic sentence, and that it is often useful to use as ~ **Q** a sentence that is readily available. In the present case the only readily available negation is '~ (A ⊃ B)'. This suggests the following strategy:

Derive: A ⊃ B, ~ (A ⊃ B)

1	~ (A ⊃ B)	Assumption
2	B	Assumption
G	A ⊃ B	
G	~ (A ⊃ B)	1 R

Our goal is now to derive 'A ⊃ B' from our two assumptions. Since this goal sentence is a conditional, we will plan on using Conditional Introduction:

Derive: A ⊃ B, ~ (A ⊃ B)

1	~ (A ⊃ B)	Assumption
2	B	Assumption
3	A	A / ⊃I
G	B	
G	A ⊃ B	3–__ ⊃I
G	~ (A ⊃ B)	1 R

It is now apparent that our derivation is effectively done. Our only remaining goal, 'B', can be obtained by Reiteration on line 2:

Derive: A ⊃ B, ~ (A ⊃ B)

1	~ (A ⊃ B)	Assumption
2	B	Assumption
3	A	A / ⊃I
4	B	2 R
5	A ⊃ B	3–4 ⊃I
6	~ (A ⊃ B)	1 R

Establishing that the following set is inconsistent in *SD* is only modestly more challenging:

{A ≡ ~ B, B ≡ C, A ≡ C}

In this example the only negation that occurs as a component of any of the members of the set is '~ B'. So perhaps our goal should be to derive both 'B' and '~ B', even though neither can be derived by Reiteration or by any other rule in a single step.

Derive: B, ~ B

1	A ≡ ~ B	Assumption
2	B ≡ C	Assumption
3	A ≡ C	Assumption
G	B	
G	~ B	

To obtain our first goal, 'B', we might try using Negation Elimination:

Derive: B, ~ B

1	A ≡ ~ B	Assumption
2	B ≡ C	Assumption
3	A ≡ C	Assumption
4	~ B	A / ~ E
G	B	4–__ ~ E
G	~ B	4–__ ~ I

A cursory inspection of the sentences on lines 1–4 reveals that we can obtain '~ B' by Reiteration and 'B' by repeated uses of Biconditional Elimination:

```
Derive: B, ~ B

1  | A ≡ ~ B                 Assumption
2  | B ≡ C                   Assumption
3  | A ≡ C                   Assumption
   |------
4  |  | ~ B                  A / ~ E
   |  |-----
5  |  | A                    1, 4 ≡E
6  |  | C                    3, 5 ≡E
7  |  | B                    2, 6 ≡E
8  |  | ~ B                  4 R
9  | B                       4–8 ~ E
   |
   |
G  | ~ B
```

The remaining task is to derive '~ B', and this too can be accomplished by repeated applications of Biconditional Elimination:

```
Derive: B, ~ B

1  | A ≡ ~ B                 Assumption
2  | B ≡ C                   Assumption
3  | A ≡ C                   Assumption
   |------
4  |  | ~ B                  A / ~ I
   |  |-----
5  |  | A                    1, 4 ≡E
6  |  | C                    3, 5 ≡E
7  |  | B                    2, 6 ≡E
8  |  | ~ B                  4 R
9  | B                       4–8 ~ I
10 | C                       2, 9 ≡E
11 | A                       3, 10 ≡E
12 | ~ B                     1, 11 ≡E
```

Finally, we will show that the set {~ (A ⊃ B), ~ (B ⊃ C)} is inconsistent in *SD*. This is a challenging exercise. We do have two negations immediately available, so we will probably use one of them as ~ **Q**; which one makes no difference. So we set up our derivation this way:

```
Derive: A ⊃ B, ~ (A ⊃ B)

1  | ~ (A ⊃ B)                    Assumption
2  | ~ (B ⊃ C)                    Assumption
   |----------
   |
   |
G  | A ⊃ B
   | ~ (A ⊃ B)                    1 R
```

We cannot apply any elimination rule to either assumption since they are both negations. So we proceed by asking how our current goal, 'A ⊃ B', could be obtained by an introduction rule, and the answer is of course by Conditional Introduction:

```
Derive: A ⊃ B, ~ (A ⊃ B)

1  | ~ (A ⊃ B)                    Assumption
2  | ~ (B ⊃ C)                    Assumption
   |----------
3  |  | A                          A / ⊃I
   |  |---
   |  |
   |  |
G  |  | B
G  | A ⊃ B                        3–__ ⊃I
   | ~ (A ⊃ B)                    1 R
```

Our new goal is 'B'. The only strategy for obtaining 'B' that seems remotely promising is that of Negation Elimination:

```
Derive: A ⊃ B, ~ (A ⊃ B)

1  | ~ (A ⊃ B)                    Assumption
2  | ~ (B ⊃ C)                    Assumption
   |----------
3  |  | A                          A / ⊃I
   |  |---
4  |  |  | ~ B                    A / ~ E
   |  |  |-----
   |  |  |
   |  |  |
G  |  | B                         4–__ ~ E
G  | A ⊃ B                        3–__ ⊃I
   | ~ (A ⊃ B)                    1 R
```

We need to derive, within the subderivation beginning on line 4, a sentence **Q** and its negation ~ **Q**. Three negations, '~ (A ⊃ B)', '~ (B ⊃ C)', and '~B' are readily available. Since the presumed inconsistency of the set we are testing fairly clearly derives from the interplay of those two assumptions—that is, neither assumption by itself is problematic—we will eventually have to appeal to both assumptions. And we are already using '~ (A ⊃ B)' (as the last line of our derivation), so perhaps it is time to find a role for '~ (B ⊃ C)'. Accordingly we will try to obtain 'B ⊃ C' and '~ (B ⊃ C)'.

```
Derive: A ⊃ B, ~ (A ⊃ B)

1  | ~ (A ⊃ B)                     Assumption
2  | ~ (B ⊃ C)                     Assumption
   |-----------
3  |   | A                         A / ⊃I
   |   |---
4  |   |   | ~ B                   A / ~ E
   |   |   |-----
   |   |   |
   |   |   |
   |   |   | B ⊃ C
   |   |   | ~ (B ⊃ C)             2 R
   |   | B                         4–__ ~ E
   | A ⊃ B                         3–__ ⊃I
   | ~ (A ⊃ B)                     1 R
```

Our new goal, 'B ⊃ C', is a conditional, so Conditional Introduction seems appropriate:

```
Derive: A ⊃ B, ~ (A ⊃ B)

1  | ~ (A ⊃ B)                     Assumption
2  | ~ (B ⊃ C)                     Assumption
   |-----------
3  |   | A                         A / ⊃I
   |   |---
4  |   |   | ~B                    A / ~ E
   |   |   |----
5  |   |   |   | B                 A / ⊃I
   |   |   |   |---
   |   |   |   |
   |   |   |   |
   |   |   |   | C
   |   |   | B ⊃ C                 5–__ ⊃I
   |   |   | ~ (B ⊃ C)             2 R
   |   | B                         4–__ ~ E
   | A ⊃ B                         3–__ ⊃I
   | ~ (A ⊃ B)                     1 R
```

At this point, as is often the case, the 'trick' is to be aware of what sentences are available to us—in this case the sentences on lines 1–5—and what we can do with those sentences. Note that we have both 'B' (at line 5) and '~ B' (at line 4), and we know that whenever we can obtain a sentence and its negation we can obtain any sentence whatsoever by the appropriate negation strategy. We want 'C', so we will obtain it by Negation Elimination.

```
Derive: A ⊃ B, ~ (A ⊃ B)

 1  | ~ (A ⊃ B)                     Assumption
 2  | ~ (B ⊃ C)                     Assumption
    |-----------
 3  | | A                           A / ⊃I
    | |---
 4  | | | ~ B                       A / ~ E
    | | |-----
 5  | | | | B                       A / ⊃I
    | | | |---
 6  | | | | | ~ C                   A / ~ E
    | | | | |-----
 7  | | | | | B                     5 R
 8  | | | | | ~ B                   4 R
 9  | | | | C                       6–8 ~ E
10  | | | B ⊃ C                     5–9 ⊃I
11  | | | ~ (B ⊃ C)                 2 R
12  | | B                           4–11 ~ E
13  | A ⊃ B                         3–12 ⊃I
14  | ~ (A ⊃ B)                     1 R
```

5.3E EXERCISES

1. Construct derivations that establish the following derivability claims. In each case start by setting up the main structure of the derivation—with the primary assumption or assumptions at the top and the sentence to be derived at the bottom, and then identify the initial subgoal or goals. Complete the derivation, remembering to consider both the form of the current goal sentence and the content of the accessible sentences in selecting appropriate subgoals.

a. {A ⊃ B} ⊢ A ⊃ (A & B)
*b. {~ B ≡ A} ⊢ A ⊃ ~ B
c. {(K ⊃ L) & (L ⊃ K)} ⊢ L ≡ K
*d. {M ≡ T, ~ T} ⊢ ~ M
e. {B & ~ B} ⊢ C
*f. {D} ⊢ A ⊃ (B ⊃ D)
g. {A ⊃ C, (~ A ∨ C) ⊃ (D ⊃ B)} ⊢ D ⊃ B
*h. {~ A ⊃ ~ B, A ⊃ C, B ∨ D, D ⊃ E} ⊢ E ∨ C
i. {A ⊃ B, ~ (B & ~ C) ⊃ A} ⊢ B
*j. {~ A ⊃ B, C ⊃ ~ B, ~ (~ C & ~ A)} ⊢ A
k. {A ∨ (B & C), C ⊃ ~ A} ⊢ B ∨ ~ C
*l. {(A ⊃ B) ⊃ ~ B} ⊢ ~ B
m. {(A ∨ B) ⊃ C, (D ∨ E) ⊃ [(F ∨ G) ⊃ A]} ⊢ D ⊃ (F ⊃ C)
*n. {(F ∨ G) ⊃ (H & I)} ⊢ ~ F ∨ H
o. {A ⊃ ~ (B ∨ C), (C ∨ D) ⊃ A, ~ F ⊃ (D & ~ E)} ⊢ B ⊃ F
*p. {(A & B) ≡ (A ∨ B), C & (C ≡ ~ ~ A)} ⊢ B
q. {F ⊃ (G ∨ H), ~ (~ F ∨ H), ~ G} ⊢ H
*r. {~ (A ⊃ B) & (C & ~ D), (B ∨ ~ A) ∨ [(C & E) ⊃ D]} ⊢ ~ E

2. Show that each of the following arguments is valid in *SD*.

a.
$$\begin{array}{l} A \supset \sim B \\ \sim B \supset C \\ \hline A \supset C \end{array}$$

*b.
$$\begin{array}{l} B \supset (A \,\&\, \sim B) \\ \hline \sim B \end{array}$$

c.
$$\begin{array}{l} A \equiv B \\ \sim A \\ \hline \sim B \end{array}$$

*d.
$$\begin{array}{l} A \supset (B \,\&\, C) \\ \sim C \\ \hline \sim A \end{array}$$

e.
$$\begin{array}{l} D \\ \hline A \supset [B \supset (C \supset D)] \end{array}$$

*f.
$$\begin{array}{l} A \equiv B \\ B \equiv C \\ \hline A \equiv C \end{array}$$

g.
$$\begin{array}{l} A \supset (B \supset C) \\ D \supset B \\ \hline A \supset (D \supset C) \end{array}$$

*h.
$$\begin{array}{l} \sim B \supset A \\ C \vee \sim B \\ \sim C \\ \hline A \end{array}$$

i.
$$\begin{array}{l} \sim A \vee B \\ B \supset C \\ \hline A \supset C \end{array}$$

*j.
$$\begin{array}{l} (E \supset T) \,\&\, (T \supset O) \\ O \supset E \\ \hline (E \equiv O) \,\&\, (O \equiv E) \end{array}$$

k.
$$\begin{array}{l} A \supset (C \supset B) \\ \sim C \supset \sim A \\ A \\ \hline B \end{array}$$

*l.
$$\begin{array}{l} \sim F \\ \sim G \\ \hline \sim (F \vee G) \end{array}$$

m.
$$\begin{array}{l} F \equiv G \\ F \vee G \\ \hline F \,\&\, G \end{array}$$

3. Prove that each of the following is a theorem in *SD*.

a. $A \supset (A \vee B)$
*b. $A \supset (B \supset A)$
c. $A \supset [B \supset (A \,\&\, B)]$
*d. $(A \,\&\, B) \supset [(A \vee C) \,\&\, (B \vee C)]$
e. $(A \equiv B) \supset (A \supset B)$
*f. $(A \,\&\, \sim A) \supset (B \,\&\, \sim B)$
g. $(A \supset B) \supset [(C \supset A) \supset (C \supset B)]$
*h. $(A \,\&\, B) \vee (\sim A \vee \sim B)$
i. $[(A \supset B) \,\&\, \sim B] \supset \sim A$
*j. $(A \,\&\, A) \equiv A$
k. $A \supset [B \supset (A \supset B)]$
*l. $\sim A \supset [(B \,\&\, A) \supset C]$
m. $(A \supset B) \supset [\sim B \supset \sim (A \,\&\, D)]$
*n. $[\sim A \supset \sim (A \supset B)] \supset A$

4. Show that the members of each of the following pairs of sentences are equivalent in *SD*.

a.	$A \,\&\, {\sim} A$	$B \,\&\, {\sim} B$
*b.	$A \,\&\, A$	$A \vee A$
c.	$(A \vee B) \supset A$	$B \supset A$
*d.	${\sim} (A \supset B)$	$A \,\&\, {\sim} B$
e.	${\sim} (A \equiv B)$	$(A \,\&\, {\sim} B) \vee (B \,\&\, {\sim} A)$
*f.	$A \equiv {\sim} B$	${\sim} (A \equiv B)$

5. Show that each of the following sets of sentences is inconsistent in *SD*.

a. $\{{\sim} (A \supset A)\}$
*b. $\{A \supset (B \,\&\, {\sim} B), A\}$
c. $\{A \equiv B, B \supset {\sim} A, A\}$
*d. $\{A \equiv {\sim} (A \equiv A), A\}$
e. $\{A \supset {\sim} A, {\sim} A \supset A\}$
*f. $\{A \supset (C \supset B), {\sim} C \supset B, A \,\&\, {\sim} B\}$
g. $\{{\sim} (A \vee B), C \supset A, {\sim} C \supset B\}$
*h. $\{{\sim} (B \equiv A), {\sim} B, {\sim} A\}$
i. $\{{\sim} (F \vee G) \equiv (A \supset A), H \supset F, {\sim} H \supset F\}$

6. Show that the following derivability claims hold in *SD*.

a. $\{A \supset B, {\sim} A \supset {\sim} B\} \vdash A \equiv B$
*b. $\{F \equiv {\sim} (G \equiv {\sim} H), {\sim} (F \vee G)\} \vdash H$
c. $\{A \equiv ({\sim} B \vee C), B \supset C\} \vdash A$
*d. $\{G \vee {\sim} H, {\sim} G \vee {\sim} H\} \vdash {\sim} H$
e. $\{B \vee (C \vee D), C \supset A, A \supset {\sim} C\} \vdash B \vee D$
*f. $\{(A \supset B) \supset C, (A \supset B) \vee {\sim} C \vdash {\sim} C \equiv {\sim} (A \supset B)$
g. $\{A \supset (D \,\&\, B), ({\sim} D \equiv B) \,\&\, (C \supset A)\} \vdash (A \vee B) \supset {\sim} C$
*h. $\{{\sim} (A \equiv B)\} \vdash (A \,\&\, {\sim} B) \vee (B \,\&\, {\sim} A)$

7. Show that each of the following arguments is valid in *SD*.

a.
$$\begin{array}{l} {\sim} (C \vee A) \\ \hline {\sim} (C \equiv {\sim} A) \end{array}$$

*b.
$$\begin{array}{l} C \vee {\sim} D \\ C \supset E \\ D \\ \hline E \end{array}$$

c.
$$\begin{array}{l} {\sim} A \,\&\, {\sim} B \\ \hline A \equiv B \end{array}$$

*d.
$$\begin{array}{l} {\sim} (F \vee {\sim} G) \equiv {\sim} (H \vee I) \\ F \vee I \\ \hline H \vee I \end{array}$$

e.
$$\begin{array}{l} H \equiv {\sim} (I \,\&\, {\sim} J) \\ {\sim} I \equiv {\sim} H \\ J \supset {\sim} I \\ \hline {\sim} H \end{array}$$

*f.
$$\begin{array}{l} {\sim} (F \supset G) \\ {\sim} (G \supset H) \\ \hline I \end{array}$$

g.
$$\begin{array}{l} (F \vee G) \vee (H \vee {\sim} I) \\ F \supset H \\ I \supset {\sim} G \\ \hline H \vee {\sim} I \end{array}$$

*h. ${\sim} D$

$C \supset (A \equiv B)$

$(D \vee B) \supset \sim A$

$(A \equiv B) \supset (D \& E)$

$\sim B \supset D$

$C \supset (\sim A \& B)$

i. $\sim (F \vee \sim G) \equiv \sim (H \vee I)$

$F \vee I$

$F \vee (I \& \sim G)$

*j. $(A \vee \sim B) \supset (C \& D)$

$A \equiv \sim D$

$\sim B \equiv \sim C$

$\sim (A \vee \sim B)$

k. $(\sim A \equiv \sim C) \equiv (B \equiv \sim D)$

$\sim A \supset \sim B$

$C \supset \sim D$

$(\sim A \equiv \sim C) \supset (\sim A \equiv D)$

*l. $F \supset (G \vee H)$

$\sim (\sim F \vee H)$

$\sim G$

H

m. $\sim (A \supset B) \& (C \& \sim D)$

$(B \vee \sim A) \vee [(C \& E) \supset D]$

$\sim E$

8. Prove that each of the following is a theorem in *SD*.

a. $\sim (A \supset B) \supset \sim (A \equiv B)$
*b. $\sim (A \equiv B) \supset \sim (A \& B)$
c. $(A \supset B) \vee (B \supset A)$
*d. $[A \supset (B \supset C)] \equiv [(A \supset B) \supset (A \supset C)]$
e. $[(A \vee B) \supset C] \equiv [(A \supset C) \& (B \supset C)]$
*f. $[A \vee (B \vee C)] \supset [(D \supset A) \vee ((D \supset B) \vee (D \supset C))]$
g. $\sim (A \equiv B) \equiv (A \equiv \sim B)$

9. Show that the members of each of the following pairs of sentences are equivalent in *SD*.

a.	A	$\sim \sim A$	Double Negation
*b.	A	$A \& A$	Idempotence
c.	A	$A \vee A$	Idempotence
*d.	$A \& B$	$B \& A$	Commutation
e.	$A \vee B$	$B \vee A$	Commutation
*f.	$A \& (B \& C)$	$(A \& B) \& C$	Association
g.	$A \vee (B \vee C)$	$(A \vee B) \vee C$	Association
*h.	$A \supset (B \supset C)$	$(A \& B) \supset C$	Exportation
i.	$A \supset B$	$\sim B \supset \sim A$	Transposition
*j.	$A \equiv B$	$(A \supset B) \& (B \supset A)$	Equivalence
k.	$A \equiv B$	$(A \& B) \vee (\sim A \& \sim B)$	Equivalence
*l.	$A \& (B \vee C)$	$(A \& B) \vee (A \& C)$	Distribution
m.	$A \vee (B \& C)$	$(A \vee B) \& (A \vee C)$	Distribution
*n.	$\sim (A \& B)$	$\sim A \vee \sim B$	De Morgan

10. Show that each of the following sets of sentences of *SL* is inconsistent in *SD*.

a. $\{(A \supset B) \& (A \supset \sim B), (C \supset A) \& (\sim C \supset A)\}$
*b. $\{B \equiv (A \& \sim A), \sim B \supset (A \& \sim A)\}$

c. $\{C \equiv \sim A, C \equiv A\}$
*d. $\{\sim (F \vee G) \equiv (\sim F \supset \sim F), \sim G \supset F\}$
e. $\{\sim [A \vee (B \vee C)], A \equiv \sim C\}$
*f. $\{F \vee (G \supset H), \sim H \& \sim (F \vee \sim G)\}$
g. $\{A \& (B \vee C), (\sim C \vee H) \& (H \supset \sim H), \sim B\}$
*h. $\{[(A \equiv B) \equiv (D \& \sim D)] \equiv B, A\}$

11. Symbolize the following arguments in *SL*. Then show that the symbolized arguments are valid in *SD*.

a. Spring has sprung, and the flowers are blooming. If the flowers are blooming, the bees are happy. If the bees are happy but aren't making honey, then spring hasn't sprung. So the bees are making honey.

*b. If Luscious Food Industries goes out of business, then food processing won't be improved. And if they go out of business, canned beans will be available if and only if Brockport Company stays in business. But Brockport Company is going out of business, and canned beans will be available. Hence Luscious Food Industries is staying in business unless food processing is improved.

c. If civil disobedience is moral, then not all resistance to the law is morally prohibited, although our legal code is correct if all resistance to the law is morally prohibited. But civil disobedience is moral if and only if either civil disobedience is moral or our legal code is correct. Our judges have acted well only if all resistance to the law is morally prohibited. So our judges haven't acted well.

*d. If oranges contain citric acid so do lemons, or if lemons don't contain citric acid neither do grapefruit. Thus, if oranges and grapefruit contain citric acid, so do lemons.

e. Neither rubber nor wood is a good conductor of electricity. But either rubber is a good conductor if and only if metal is, or if metal or glass is a good conductor then wood is a good conductor if and only if metal is. So metal isn't a good conductor of electricity.

*f. If the trains stop running then airline prices will increase, and buses will reduce their fares provided that trains don't stop running. If airline prices increase, then buses won't lose their customers. Hence buses will lose their customers only if they reduce their fares.

g. If the house is built and taxes increase, Jones will go bankrupt. If Smith becomes mayor, then the tax director will quit; and Smith will become mayor unless the tax director quits. But taxes won't increase if but only if the tax director doesn't quit and Smith becomes mayor. So if the house is built, Jones will go bankrupt.

*h. Jim is a Democrat only if Howard or Rhoda is. If Howard is a Democrat, so are Barbara and Allen. If Barbara is a Democrat, then Allen is a Democrat only if Freda is. But not both Freda and Jim are Democrats. Therefore Jim is a Democrat only if Rhoda is too.

i. If life is a carnival, then I'm a clown or a trapeze artist. But either life isn't a carnival or there are balloons, and either there aren't any balloons or I'm not a clown. So, if life is a carnival, then I'm a trapeze artist.

12. Symbolize the following passages in *SL* and show that the resulting sets of sentences are inconsistent in *SD*.

a. If motorcycling is dangerous sailboating is also dangerous, and if sailboating is dangerous parachuting is dangerous. Motorcycling is dangerous but parachuting is not.

*b. If the recipe doesn't call for flavoring or it doesn't call for eggs, it's not a recipe for tapioca. If the recipe calls for eggs, then it's a tapioca recipe and it doesn't call for flavoring. But this recipe calls for eggs.

c. Bach is popular only if Beethoven is ignored. If Bach is unpopular and Beethoven isn't ignored, then current musical tastes are hopeless. Current musical tastes aren't hopeless, and Beethoven isn't ignored.

*d. Historians are right just in case theologians are mistaken, if and only if Darwin's theory is correct. And if historians or philosophers are right, then Darwinian theory is correct and theologians are mistaken. Historians are right if and only if philosophers are wrong. But if Darwinian theory is correct, then historians are mistaken.

e. Either Martha was commissioned to write the ballet or, if the fund-raising sale was a failure, Tony was commissioned. Nancy will dance if and only if Tony wasn't commissioned. But the fund-raiser was a failure, Nancy will dance, and Martha wasn't commissioned.

13. Explain:

a. Why we would not want to include the following derivation rule in *SD*.

$$\begin{array}{|l} \mathbf{P} \vee \mathbf{Q} \\ \\ \mathbf{P} \end{array}$$

*b. Why Negation Introduction is a dispensable rule in *SD*. We take a rule to be dispensable in *SD* if and only if the last line of every derivation that makes use of the rule in question can also be derived from the given assumptions without using that rule.

c. Why Reiteration is a dispensable rule in *SD*.

*d. Why deriving a sentence and its negation within the scope of an auxilliary assumption *does not* show that the primary assumptions constitute an inconsistent set but *does* show that the set that consists of the primary assumptions and the assumptions of all open subderivations is inconsistent.

e. Why an argument of *SL* that has as one of its premises the negation of a theorem is valid in *SD*.

14. In Chapter 6 (see Sections 6.3 and 6.4) we prove that, for any sentence **P** and set Γ of sentences of *SL*,

$$\Gamma \vdash \mathbf{P} \text{ in } SD \text{ if and only if } \Gamma \vDash \mathbf{P}.$$

Show that a-c below follow from this result.

a. An argument of *SL* is valid in *SD* if and only if the argument is truth-functionally valid.

*b. A sentence **P** of *SL* is a theorem in *SD* if and only if **P** is truth-functionally true.

c. Sentences **P** and **Q** of *SL* are equivalent in *SD* if and only if **P** and **Q** are truth-functionally equivalent.

5.4 THE DERIVATION SYSTEM *SD+*

In this section we introduce a new natural deduction system, *SD+*, which contains all the derivation rules of *SD* plus some additional rules. However, *SD+* is not a stronger system than *SD* in the sense that more arguments of *SL* can be shown to be valid or that more sentences of *SL* are theorems in *SD* than are in *SD*+. That is

$\Gamma \vdash \mathbf{P}$ in *SD*

if and only if

$\Gamma \vdash \mathbf{P}$ in *SD*+

However, historically a larger set of rules, such as those constituting *SD*+, have been used in many derivation systems. This larger set contains some rules absent from *SD* that do correspond to reasoning patterns commonly used in ordinary discourse, and often derivations in *SD*+ are shorter than corresponding derivations in *SD*.

RULES OF INFERENCE

Suppose that prior to line **n** of a derivation two accessible lines, **i** and **j**, contain $\mathbf{P} \supset \mathbf{Q}$ and $\sim \mathbf{Q}$, respectively. In *SD* we can derive $\sim \mathbf{P}$ as follows:

i	$\mathbf{P} \supset \mathbf{Q}$	
j	$\sim \mathbf{Q}$	
n	$\quad \mathbf{P}$	A / ~ I
n + 1	$\quad \mathbf{Q}$	i, **n** ⊃E
n + 2	$\quad \sim \mathbf{Q}$	j R
n + 3	$\sim \mathbf{P}$	n − (**n** + 2) ~ I

To avoid going through this routine every time such a situation arises, we introduce the rule **Modus Tollens:**

Modus Tollens (MT)

	$\mathbf{P} \supset \mathbf{Q}$
	$\sim \mathbf{P}$
▷	$\sim \mathbf{P}$

Now suppose that prior to line **n** of a derivation two accessible lines, **i** and **j**, contain $\mathbf{P} \supset \mathbf{Q}$ and $\mathbf{Q} \supset \mathbf{R}$. A routine to derive $\mathbf{P} \supset \mathbf{R}$ in *SD* beginning at line **i** is as follows:

i	$\mathbf{P} \supset \mathbf{Q}$	
j	$\mathbf{Q} \supset \mathbf{R}$	
n	$\quad \mathbf{P}$	**A** / ~ I
n + 1	$\quad \mathbf{Q}$	**i, n** ⊃**E**
n + 2	$\quad \mathbf{R}$	**j, n** + **1** ⊃**E**
n + 3	$\mathbf{P} \supset \mathbf{R}$	**n** − (**n** + 2) ⊃**I**

To avoid this routine, we introduce the rule **Hypothetical Syllogism:**

Hypothetical Syllogism (HS)

	$\mathbf{P} \supset \mathbf{Q}$
	$\mathbf{Q} \supset \mathbf{R}$
▷	$\mathbf{P} \supset \mathbf{R}$

Finally suppose that prior to the line **n** of a derivation two accessible lines, **i** and **j**, contain $\mathbf{P} \vee \mathbf{Q}$ and $\sim \mathbf{P}$ and that we wish to derive **Q**. A routine for accomplishing this in *SD* is as follows:

i	$\mathbf{P} \vee \mathbf{Q}$	
j	$\sim \mathbf{P}$	
n	$\quad \mathbf{P}$	A / ∨E
n + 1	$\quad\quad \sim \mathbf{Q}$	A / ~ E
n + 2	$\quad\quad \mathbf{P}$	**n** R
n + 3	$\quad\quad \sim \mathbf{P}$	**j** R
n + 4	$\quad \mathbf{Q}$	**n** + 1 − **n** + 3 ~ E
n + 5	$\quad \mathbf{Q}$	A / ∨E
n + 6	$\quad \mathbf{Q}$	**n** + 5 R
n + 7	$\mathbf{Q}$	**i**, **n** − **n** + 4, **n** + 5 − **n** + 6 ∨E

The rule of *Disjunctive Syllogism* allows us to avoid going through this routine for this and similar cases.

Disjunctive Syllogism (DS)

$\mathbf{P} \vee \mathbf{Q}$		$\mathbf{P} \vee \mathbf{Q}$
$\sim \mathbf{P}$	or	$\sim \mathbf{Q}$
$\mathbf{Q}$		$\mathbf{P}$

The three rules of inference just introduced can be thought of as derived rules. They are added for convenience only; whatever we can derive with them, we can derive without them, using only the rules of *SD*.

RULES OF REPLACEMENT

In addition to rules of inference, there are also derivation rules known as *rules of replacement*. Rules of replacement, as their name suggests, allow us to derive some sentences from other sentences by replacing sentential components. For example, from the sentence

$G \vee (H \,\&\, K)$

we can certainly infer

$G \vee ({\sim}{\sim} H \,\&\, K)$

In this instance the sentential component 'H' has been replaced with '~ ~ H'. Similarly from

$G \vee ({\sim}{\sim} H \,\&\, K)$

we can certainly infer

$G \vee (H \,\&\, K)$

Double Negation is the rule of replacement that licenses such moves within a derivation.

Double Negation (DN)

$\mathbf{P} \lhd \rhd {\sim}{\sim}\, \mathbf{P}$

That is, by using Double Negation, we can derive from a sentence **Q** that contains **P** as a sentential component another sentence that is like **Q**, except that one occurrence of the sentential component **P** has been replaced with ~ ~ **P**. And, by using Double Negation, we can derive from a sentence **Q** that contains ~ ~ **P** as a sentential component another sentence that is like **Q**, except that one occurrence of the sentential component ~ ~ **P** has been replaced with **P**.

Double Negation can be applied to any of the sentential components of a sentence. For instance, from

$G \vee (H \,\&\, K)$

Double Negation permits us to derive

$G \vee {\sim}{\sim} (H \,\&\, K)$

And from

$G \vee {\sim}{\sim} (H \,\&\, K)$

Double Negation allows us to derive

$$G \vee (H \& K)$$

Since every sentence is a sentential component of itself, Double Negation applies to the entire sentence as well. In a derivation Double Negation permits us to go from

$$G \vee (H \& K)$$

to

$$\sim \sim [G \vee (H \& K)]$$

and from

$$\sim \sim [G \vee (H \& K)]$$

to

$$G \vee (H \& K)$$

Here are the rules of replacement for *SD+*:

Commutation (Com)

$\mathbf{P} \& \mathbf{Q} \lhd \rhd \mathbf{Q} \& \mathbf{P}$

$\mathbf{P} \vee \mathbf{Q} \lhd \rhd \mathbf{Q} \vee \mathbf{P}$

Association (Assoc)

$\mathbf{P} \& (\mathbf{Q} \& \mathbf{R}) \lhd \rhd (\mathbf{P} \& \mathbf{Q}) \& \mathbf{R}$

$\mathbf{P} \vee (\mathbf{Q} \vee \mathbf{R}) \lhd \rhd (\mathbf{P} \vee \mathbf{Q}) \vee \mathbf{R}$

Implication (Impl)

$\mathbf{P} \supset \mathbf{Q} \lhd \rhd \sim \mathbf{P} \vee \mathbf{Q}$

Double Negation (DN)

$\mathbf{P} \lhd \rhd \sim \sim \mathbf{P}$

De Morgan (DeM)

$\sim (\mathbf{P} \& \mathbf{Q}) \lhd \rhd \sim \mathbf{P} \vee \sim \mathbf{Q}$

$\sim (\mathbf{P} \vee \mathbf{Q}) \lhd \rhd \sim \mathbf{P} \& \sim \mathbf{Q}$

Idempotence (Idem)

$\mathbf{P} \lhd \rhd \mathbf{P} \& \mathbf{P}$

$\mathbf{P} \lhd \rhd \mathbf{P} \vee \mathbf{P}$

Transposition (Trans)

$\mathbf{P} \supset \mathbf{Q} \lhd \rhd \sim \mathbf{Q} \supset \sim \mathbf{P}$

Exportation (Exp)

$\mathbf{P} \supset (\mathbf{Q} \supset \mathbf{R}) \lhd \rhd (\mathbf{P} \& \mathbf{Q}) \supset \mathbf{R}$

Distribution (Dist)

$\mathbf{P} \& (\mathbf{Q} \vee \mathbf{R}) \lhd \rhd (\mathbf{P} \& \mathbf{Q}) \vee (\mathbf{P} \& \mathbf{R})$

$\mathbf{P} \vee (\mathbf{Q} \& \mathbf{R}) \lhd \rhd (\mathbf{P} \vee \mathbf{Q}) \& (\mathbf{P} \vee \mathbf{R})$

Equivalence (Equiv)

$\mathbf{P} \equiv \mathbf{Q} \lhd \rhd (\mathbf{P} \supset \mathbf{Q}) \& (\mathbf{Q} \supset \mathbf{P})$

$\mathbf{P} \equiv \mathbf{Q} \lhd \rhd (\mathbf{P} \& \mathbf{Q}) \vee (\sim \mathbf{P} \& \sim \mathbf{Q})$

Rules of replacement always allow the replacement of sentential components. In addition, all these rules of replacement are two-way rules; that is, a sentential component that has the form of the sentence on the left of '$\lhd \rhd$' can be replaced with a sentential component that has the form of the sentence on the right of '$\lhd \rhd$', and vice versa.

Consider the following derivation:

Derive: J ⊃ [M ∨ (G ∨ I)]

1	J ⊃ [K ∨ (L ∨ H)]	Assumption
2	[(K ∨ L) ∨ H] ⊃ [(M ∨ G) ∨ I]	Assumption
3	J ⊃ [(K ∨ L) ∨ H]	1 Assoc
4	J ⊃ [(M ∨ G) ∨ I]	2, 3 HS
5	J ⊃ [M ∨ (G ∨ I)]	4 Assoc

Here the replacement rule Association has been used twice—first to replace a sentential component of the form **P** ∨ **(Q** ∨ **R)** with a sentential component of the form **(P** ∨ **Q)** ∨ **R** and then to replace a sentential component of the form **(P** ∨ **Q)** ∨ **R** with a sentential component of the form **P** ∨ **(Q** ∨ **R)**.

Since all the derivation rules of *SD* are derivation rules of *SD+*, the procedures for properly applying the rules of *SD* apply to *SD+* as well. The rules of inference of *SD+*, including Modus Tollens, Hypothetical Syllogism, and Disjunctive Syllogism, must be applied to entire sentences on a line. Rules of replacement, on the other hand, can be applied to all sentential components. The following derivation illustrates the proper use of several of the rules of replacement:

Derive: ~ C ≡ E

1	(D ∨ B) ∨ (E ⊃ ~ C)	Assumption
2	~ B & [~ D & (~ E ⊃ C)]	Assumption
3	(~ B & ~ D) & (~ E ⊃ C)	2 Assoc
4	~ (B ∨ D) & (~ E ⊃ C)	3 DeM
5	~ (B ∨ D)	4 &E
6	~ (D ∨ B)	5 Com
7	E ⊃ ~ C	1, 6 DS
8	~ E ⊃ C	3 &E
9	~ C ⊃ ~ ~ E	8 Trans
10	~ C ⊃ E	9 DN
11	(~ C ⊃ E) & (E ⊃ ~ C)	7, 10 &I
12	~ C ≡ E	11 Equiv

Notice that each application of a derivation rule requires a separate line. Moreover, care must be taken to apply each derivation rule only to sentences that have the proper form (or, in the case of rules of replacement, sentences that have components that have the proper form).

Here is an example in which these points are ignored:

Derive: $\sim A \supset [B \supset (G \vee D)]$

1	$(A \vee \sim B) \vee \sim C$	Assumption	
2	$(D \vee G) \vee C$	Assumption	
3	$\sim (\sim A \;\&\; B) \vee \sim C$	1 DeM	**MISTAKE!**
4	$(\sim A \;\&\; B) \supset \sim C$	3 Impl	
5	$C \vee (G \vee D)$	2 Com	**MISTAKE!**
6	$\sim C \supset (G \vee D)$	5 Impl	**MISTAKE!**
7	$(\sim A \;\&\; B) \supset (G \vee D)$	4, 6 HS	
8	$\sim A \supset [B \supset (G \vee D)]$	7 Exp	

De Morgan does not license entering the sentence on line 3. What De Morgan does allow is the replacement of a sentential component of the form $\sim \mathbf{P} \vee \sim \mathbf{Q}$ with a sentential component of the form $\sim (\mathbf{P} \;\&\; \mathbf{Q})$, but the sentential component '$A \vee \sim B$' does not have the form $\sim \mathbf{P} \vee \sim \mathbf{Q}$. However, by applying Double Negation to the first assumption, we can obtain '$(\sim \sim A \vee \sim B) \vee \sim C$'. And this latter sentence does have a sentential component of the form $\sim \mathbf{P} \vee \sim \mathbf{Q}$, namely, '$\sim \sim A \vee \sim B$'. Here $\mathbf{P}$ is '$\sim A$', and $\mathbf{Q}$ is 'B'. Hence the derivation should begin this way:

Derive: $\sim A \supset [B \supset (G \vee D)]$

1	$(A \vee \sim B) \vee \sim C$	Assumption
2	$(D \vee G) \vee C$	Assumption
3	$(\sim \sim A \vee \sim B) \vee \sim C$	1 DN
4	$\sim (\sim A \;\&\; B) \vee \sim C$	3 DeM

The second mistake in our example, in line 5, is that Commutation is applied twice within the same line. Each application of a rule, even if it is the same rule, requires a separate line. Correctly done, the derivation proceeds:

5	$(\sim A \;\&\; B) \supset \sim C$	4 Impl
6	$C \vee (D \vee G)$	2 Com
7	$C \vee (G \vee D)$	6 Com

The third mistake, in line 6 of the example, also stems from our trying to apply a rule of replacement to a sentential component that does not have the form required by the rule. Implication permits the replacement of a sentential component of the form $\sim \mathbf{P} \vee \mathbf{Q}$ with a sentential component of the form $\mathbf{P} \supset \mathbf{Q}$, but '$C \vee (G \vee D)$' does not have the form $\sim \mathbf{P} \vee \mathbf{Q}$. However, applying Double Negation to 'C', a sentential component of '$C \vee (G \vee D)$', generates '$\sim \sim C \vee (G \vee D)$'. This latter sentence does have the form $\sim \mathbf{P} \vee \mathbf{Q}$, where P is '$\sim C$' and $\mathbf{Q}$ is '$G \vee D$'. Here is the entire derivation done correctly:

Derive: $\sim A \supset [B \supset (G \vee D)]$

1	$(A \vee \sim B) \vee \sim C$	Assumption
2	$(D \vee G) \vee C$	Assumption
3	$(\sim\sim A \vee \sim B) \vee \sim C$	1 DN
4	$\sim(\sim A \;\&\; B) \vee \sim C$	3 DeM
5	$(\sim A \;\&\; B) \supset \sim C$	4 Impl
6	$C \vee (D \vee G)$	2 Com
7	$C \vee (G \vee D)$	6 Com
8	$\sim\sim C \vee (G \vee D)$	7 DN
9	$\sim C \supset (G \vee D)$	8 Impl
10	$(\sim A \;\&\; B) \supset (G \vee D)$	5, 9 HS
11	$\sim A \supset [B \supset (G \vee D)]$	10 Exp

The definitions of the basic concepts of *SD+* parallel the definitions for the basic concepts of *SD,* except that '*SD*' is replaced with '*SD+*'. For example, the concept of derivability is defined as follows:

> A sentence **P** of *SL* is *derivable in SD+* from a set Γ of sentence of *SL* if and only if there is a derivation in *SD+* in which all the primary assumptions are members of Γ and **P** occurs within the scope of only those assumptions.

Consequently tests for the various syntactic properties in *SD+* are analogous to those of *SD*. To show that an argument is valid in *SD+*, we construct a derivation in *SD+* showing that the conclusion of the argument is derivable in *SD+* from the set all of whose members are premises of the argument. To show that a sentence **P** of *SL* is a theorem in *SD+*, we show that **P** is derivable in *SD+* from the empty set. And so on. Remember that, although *SD* and *SD+* are different syntactic systems, whatever can be derived in one can be derived in the other.

The Derivation Rules of SD+

All the Derivation Rules of *SD* and Rules of Inference

Modus Tollens (MT)

	$\mathbf{P} \supset \mathbf{Q}$
	$\sim \mathbf{Q}$
▷	$\sim \mathbf{P}$

Hypothetical Syllogism (HS)

	$\mathbf{P} \supset \mathbf{Q}$
	$\mathbf{Q} \supset \mathbf{R}$
▷	$\mathbf{P} \supset \mathbf{R}$

Disjunctive Syllogism (DS)

	$\mathbf{P} \vee \mathbf{Q}$			$\mathbf{P} \vee \mathbf{Q}$
	$\sim \mathbf{P}$	or		$\sim \mathbf{Q}$
▷	$\mathbf{Q}$		▷	$\mathbf{P}$

Rules of Replacement

Commutation (Com)

$\mathbf{P} \,\&\, \mathbf{Q} \lhd\rhd \mathbf{Q} \,\&\, \mathbf{P}$
$\mathbf{P} \vee \mathbf{Q} \lhd\rhd \mathbf{Q} \vee \mathbf{P}$

Association (Assoc)

$\mathbf{P} \,\&\, (\mathbf{Q} \,\&\, \mathbf{R}) \lhd\rhd (\mathbf{P} \,\&\, \mathbf{Q}) \,\&\, \mathbf{R}$
$\mathbf{P} \vee (\mathbf{Q} \vee \mathbf{R}) \lhd\rhd (\mathbf{P} \vee \mathbf{Q}) \vee \mathbf{R}$

Implication (Impl)

$\mathbf{P} \supset \mathbf{Q} \lhd\rhd \sim \mathbf{P} \vee \mathbf{Q}$

Double Negation (DN)

$\mathbf{P} \lhd\rhd \sim\sim \mathbf{P}$

De Morgan (DeM)

$\sim (\mathbf{P} \,\&\, \mathbf{Q}) \lhd\rhd \sim \mathbf{P} \vee \sim \mathbf{Q}$
$\sim (\mathbf{P} \vee \mathbf{Q}) \lhd\rhd \sim \mathbf{P} \,\&\, \sim \mathbf{Q}$

Idempotence (Idem)

$\mathbf{P} \lhd\rhd \mathbf{P} \,\&\, \mathbf{P}$
$\mathbf{P} \lhd\rhd \mathbf{P} \vee \mathbf{P}$

Transposition (Trans)

$\mathbf{P} \supset \mathbf{Q} \lhd\rhd \sim \mathbf{Q} \supset \sim \mathbf{P}$

Exportation (Exp)

$\mathbf{P} \supset (\mathbf{Q} \supset \mathbf{R}) \lhd\rhd (\mathbf{P} \,\&\, \mathbf{Q}) \supset \mathbf{R}$

Distribution (Dist)

$\mathbf{P} \,\&\, (\mathbf{Q} \vee \mathbf{R}) \lhd\rhd (\mathbf{P} \,\&\, \mathbf{Q}) \vee (\mathbf{P} \,\&\, \mathbf{R})$
$\mathbf{P} \vee (\mathbf{Q} \,\&\, \mathbf{R}) \lhd\rhd (\mathbf{P} \vee \mathbf{Q}) \,\&\, (\mathbf{P} \vee \mathbf{R})$

Equivalence (Equiv)

$\mathbf{P} \equiv \mathbf{Q} \lhd\rhd (\mathbf{P} \supset \mathbf{Q}) \,\&\, (\mathbf{Q} \supset \mathbf{P})$
$\mathbf{P} \equiv \mathbf{Q} \lhd\rhd (\mathbf{P} \,\&\, \mathbf{Q}) \vee (\sim \mathbf{P} \,\&\, \sim \mathbf{Q})$

5.4E EXERCISES

1. Show that the following derivability claims hold in *SD+*.

a. $\{D \supset E, E \supset (Z \,\&\, W), \sim Z \vee \sim W\} \vdash \sim D$
*b. $\{(H \,\&\, G) \supset (L \vee K), G \,\&\, H\} \vdash K \vee L$
c. $\{(W \supset S) \,\&\, \sim M, (\sim W \supset H) \vee M, (\sim S \supset H) \supset K\} \vdash K$
*d. $\{[(K \,\&\, J) \vee I] \vee \sim Y, Y \,\&\, [(I \vee K) \supset F]\} \vdash F \vee N$
e. $\{(M \vee B) \vee (C \vee G), \sim B \,\&\, (\sim G \,\&\, \sim M)\} \vdash C$
*f. $\{\sim L \vee (\sim Z \vee \sim U), (U \,\&\, G) \vee H, Z\} \vdash L \supset H$

2. Show that each of the following is valid in *SD+*.

a.
$\sim Y \supset \sim Z$
$\sim Z \supset \sim X$
$\sim X \supset \sim Y$
———
$Y \equiv Z$

*b.
$(\sim A \,\&\, \sim B) \vee (\sim A \,\&\, \sim C)$
$(E \,\&\, D) \supset A$
———
$\sim E \vee \sim D$

c.
$(F \,\&\, G) \vee (H \,\&\, \sim I)$
$I \supset \sim (F \,\&\, D)$
———
$I \supset \sim D$

*d.
$F \supset (\sim G \vee H)$
$F \supset G$
$\sim (H \vee I)$
———
$F \supset J$

e. $F \supset (G \supset H)$

$\sim I \supset (F \vee H)$

$F \supset G$

$I \vee H$

*f. $G \supset (H \& \sim K)$

$H \equiv (L \& I)$

$\sim I \vee K$

$\sim G$

g. $[(X \& Z) \& Y] \vee (\sim X \supset \sim Y)$

$X \supset Z$

$Z \supset Y$

$X \equiv Y$

3. Show that each of the following is a theorem in *SD+*.

a. $A \vee \sim A$

*b. $\sim \sim \sim \sim \sim (A \& \sim A)$

c. $A \vee [(\sim A \vee B) \& (\sim A \vee C)]$

*d. $[(A \& B) \supset (B \& A)] \& [\sim (A \& B) \supset \sim (B \& A)]$

e. $[A \supset (B \& C)] \equiv [(\sim B \vee \sim C) \supset \sim A]$

*f. $[A \vee (B \vee C)] \equiv [C \vee (B \vee A)]$

g. $[A \supset (B \equiv C)] \equiv (A \supset [(\sim B \vee C) \& (\sim C \vee B)])$

*h. $(A \vee [B \supset (A \supset B)]) \equiv (A \vee [(\sim A \vee \sim B) \vee B])$

i. $[\sim A \supset (\sim B \supset C)] \supset [(A \vee B) \vee (\sim \sim B \vee C)]$

*j. $(\sim A \equiv \sim A) \equiv [\sim (\sim A \supset A) \equiv (A \supset \sim A)]$

4. Show that the members of each of the following pairs of sentences are equivalent in *SD+*.

a. $A \vee B$
$\sim (\sim A \& \sim B)$

*b. $A \& (B \vee C)$
$(B \& A) \vee (C \& A)$

c. $(A \& B) \supset C$
$\sim (A \supset C) \supset \sim B$

*d. $(A \vee B) \vee C$
$\sim A \supset (\sim B \supset C)$

e. $A \vee (B \equiv C)$
$A \vee (\sim B \equiv \sim C)$

*f. $(A \& B) \vee [(C \& D) \vee A]$
$([(C \vee A) \& (C \vee B)] \& [(D \vee A) \& (D \vee B)]) \vee A$

5. Show that the following sets of sentences are inconsistent in *SD+*.

a. $\{[(E \& F) \vee \sim \sim G] \supset M, \sim [[(G \vee E) \& (F \vee G)] \supset (M \& M)]\}$

*b. $\{\sim [(\sim C \vee \sim \sim C) \vee \sim \sim C]\}$

c. $\{M \& L, [L \& (M \& \sim S)] \supset K, \sim K \vee \sim S, \sim (K \equiv \sim S)\}$

*d. $\{B \& (H \vee Z), \sim Z \supset K, (B \equiv Z) \supset \sim Z, \sim K\}$

e. $\{\sim [W \& (Z \vee Y)], (Z \supset Y) \supset Z, (Y \supset Z) \supset W\}$

*f. $\{[(F \supset G) \vee (\sim F \supset G)] \supset H, (A \& H) \supset \sim A, A \vee \sim H\}$

6. Symbolize the following arguments in *SL*, and show that they are valid in *SD+*.

a. If the phone rings Ed is calling, or if the beeper beeps Ed is calling. If not both Ed and Agnes are at home today, then it’s not the case that if the phone rings, Ed is calling. Ed isn’t home today, and he isn’t calling. So the beeper won’t beep.

*b. If Monday is a bad day, then I'll lose my job provided the boss doesn't call in sick. The boss won't call in sick. So I'll lose my job—since either Monday will be a bad day, or the boss won't call in sick only if I lose my job.

c. Army coats are warm only if they're either made of wool or not made of cotton or rayon. If army coats are not made of rayon, then they're made of cotton. Hence, if they're not made of wool, army coats aren't warm.

*d. If either the greenhouse is dry or the greenhouse is sunny if and only if it's not raining, the violets will wither. But if the violets wither the greenhouse is sunny, or if the violets wither the greenhouse isn't dry. It's raining, and the greenhouse isn't sunny. So the greenhouse is dry only if the violets won't wither.

e. It's not the case that John is rich and Hugo isn't. In fact, Hugo isn't rich, unless Moe is. And if Moe just emptied his bank account, then he isn't rich. Thus, if John is rich, then it's not the case that either Moe emptied his bank account or Moe isn't rich,

*f. Neither aspirin nor gin will ease my headache, unless it's psychosomatic. If it's psychosomatic and I'm really not ill, then I'll go out to a party and drink some martinis. So, if I'm not ill and don't drink any martinis, then aspirin won't ease my headache.

g. If I stay on this highway and don't slow down, I'll arrive in Montreal by 5:00. If I don't put my foot on the brake, I won't slow down. Either I won't slow down or I'll stop for a cup of coffee at the next exit. I'll stop for a cup of coffee at the next exit only if I'm falling asleep. So, if I don't arrive in Montreal by 5:00, then I'll stay on this highway only if I'm falling asleep and I put my foot on the brake.

*h. The weather is fine if and only if it's not snowing, and it's not snowing if and only if the sky is clear. So, either the weather is fine, the sky is clear, and it's not snowing; or it's snowing, the sky isn't clear, and the weather is lousy.

7. Symbolize the following passages in *SL,* and show that the resulting sets of sentences of *SL* are inconsistent in *SD+*.

a. Unless Stowe believes that all liberals are atheists, his claims about current politics are unintelligible. But if liberals are atheists only if they're not churchgoers, then Stowe's claims about current politics are nevertheless intelligible. Liberals are, in fact, churchgoers if and only if Stowe doesn't believe that they're all atheists, and if liberals aren't atheists, then Stowe doesn't believe that they are atheists. Liberals aren't atheists.

*b. Either Congress won't cut taxes or the elderly and the poor will riot, if but only if big business prospers. If the elderly don't riot, then Congress won't cut taxes. It won't happen that both the poor will riot and big business will prosper, and it won't happen that the poor don't riot and big business doesn't prosper. But if big business prospers, then Congress will cut taxes.

8. Answer the following.

a. Suppose we can derive **Q** from **P** by using only the rules of replacement. Why can we be sure that we can derive **P** from **Q**?

*b. Why must all arguments that are valid in *SD* be valid in *SD+* as well?

c. Suppose we develop a new natural deduction system *SD**. Let *SD** contain all the derivation rules of *SD* and in addition the derivation rule Absorption.

Absorption

$$\begin{array}{l|l} & \mathbf{P} \supset \mathbf{Q} \\ \triangleright & \mathbf{P} \supset (\mathbf{P} \; \& \; \mathbf{Q}) \end{array}$$

Using only the derivation rules of *SD*, develop a routine showing that any sentence derived by using Absorption could be derived in *SD* without using it.

GLOSSARY[4]

DERIVABILITY IN *SD:* A sentence **P** of *SL* is *derivable in SD* from a set Γ of sentences of *SL* if and only if there is a derivation in *SD* in which all the primary assumptions are members of Γ and **P** occurs in the scope of only those assumptions.

VALIDITY IN *SD:* An argument of *SL* is *valid in SD* if and only if the conclusion of the argument is derivable in *SD* from the set consisting of the premises. An argument of *SL* is *invalid in SD* if and only if it is not valid in *SD*.

THEOREM IN *SD:* A sentence **P** of *SL* is a *theorem in SD* if and only if **P** is derivable in *SD* from the empty set.

EQUIVALENCE IN *SD:* Sentences **P** and **Q** of *SL* are *equivalent in SD* if and only if **Q** is derivable in *SD* from {**P**} and **P** is derivable in *SD* from {**Q**} .

INCONSISTENCY IN *SD:* A set Γ of sentences of *SL* is *inconsistent in SD* if and only if both a sentence **P** of *SL* and its negation ~ **P** are derivable in *SD* from Γ. A set Γ of sentences of *SL* is *consistent in SD* if and only if it is not inconsistent in *SD*.

[4]Similar definitions hold for the derivation system *SD*+.