Chapter

3

SENTENTIAL LOGIC: SEMANTICS

Section 3.1 introduces the foundations of the truth-functional semantics for *SL*: truth-value assignments and the truth-tables that record them. Sections 3.2 through 3.5 present the truth-functional versions of the core logical concepts: truth-functional truth, falsehood, and indeterminacy; truth-functional equivalence; truth-functional consistency; and truth-functional entailment and validity. Section 3.6 explicates all of the truth-functional concepts in terms of truth-functional consistency to provide a framework for truth-trees, which are presented in Chapter 4.

3.1 TRUTH-VALUE ASSIGNMENTS AND TRUTH-TABLES FOR SENTENCES

In Chapter 1, we introduced logical concepts such as logical truth and logical validity. In this chapter we shall develop formal tests for truth-functional versions of the core logical concepts introduced in Chapter 1. Specifically, we will develop tests for truth-functional truth, falsity, and indeterminacy; truth-functional equivalence; truth-functional consistency; and truth-functional entailment and validity. All these concepts fall within the realm of *semantics:* They concern the truth-values and truth-conditions of sentences and sets of sentences of *SL*. Before defining these truth-functional concepts, our first task is to specify how truth-values and truth-conditions for sentences of *SL* are determined.

Every sentence of *SL* can be built up from its atomic components in accordance with the definition of sentences. Similarly the truth-value of a sentence of *SL* is completely determined by the truth-values of its atomic components in accordance with the characteristic truth-tables for the connectives. We repeat the characteristic truth-tables here:

P	**~ P**
T	**F**
F	**T**

P	**Q**	**P & Q**
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**F**

P	**Q**	**P ∨ Q**
T	**T**	**T**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

P	**Q**	**P ⊃ Q**
T	**T**	**T**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

P	**Q**	**P ≡ Q**
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**T**

These tables tell us how to determine the truth-value of a truth-functionally compound sentence given the truth-values of its immediate sentential components.

The truth-values of atomic sentences are fixed by **truth-value assignments:**

> A truth-value assignment is an assignment of truth-values (**T**s and **F**s) to the atomic sentences of *SL*.

The concept of a truth-value assignment is the basic semantic concept of *SL*. Intuitively, each truth-value assignment gives us a description of a way the world *might* be, for in each we consider a combination of truth-values that atomic sentences might have. We assume that the atomic sentences of *SL* are truth-functionally independent—that is, that the truth-value assigned to one does not affect the truth-value assigned to any other. For generality we stipulate that a truth-value assignment must assign a truth-value to every atomic sentence, so that a truth-value assignment gives a *complete* description of a way the world might be. The truth-values of truth-functionally compound sentences of *SL* are uniquely and completely determined by the truth-values of their atomic components, so it follows that every truth-functionally compound sentence also has a truth-value, either **T** or **F**, on each truth-value assignment.

A truth-table for a sentence of *SL* is used to record its truth-value on each truth-value assignment. Because a truth-value assignment assigns truth-values to an infinite number of atomic sentences (*SL* has infinitely many atomic sentences), we cannot list an entire truth-value assignment in a truth-table.

Instead, we list all the possible combinations of truth-values that the sentence's atomic components may have on a truth-value assignment. As an example, here is the beginning of a truth-table for '~ B ⊃ C':

B	C	~ B ⊃ C
T	**T**	
T	**F**	
F	**T**	
F	**F**	

The atomic components of the sentence are 'B' and 'C', and the four rows of the table display the four combinations of truth-values that these components might have. Each row represents an infinite number of truth-value assignments, namely, all the truth-value assignments that assign to 'B' and 'C' the values indicated in that row. Since the truth-value of '~ B ⊃ C' on a truth-value assignment depends only on the truth-values that its atomic components have on that assignment, the four combinations that we have displayed will allow us to determine the truth-value of '~ B ⊃ C' on any truth-value assignment.

The first step in constructing a truth-table for a sentence **P** of *SL* is to determine the number of different combinations of truth-values that its atomic components can have. There is a simple way to do this. Consider first the case in which **P** has one atomic component. There are two different truth-values that the single atomic component may have: **T** and **F**. Now suppose that **P** is a sentence with two atomic components. In this case there are four combinations of truth-values that the atomic components of **P** might have, as we have seen in the case of '~ B ⊃ C' above.

If **P** has three atomic components, there are eight combinations of truth-values that its atomic components might have. To see this, suppose we want to expand this truth-table to record truth-values for a modified sentence that has three atomic components:

A	B	C	(~ B ⊃ C) & (A ≡ B)
	T	**T**	
	T	**F**	
	F	**T**	
	F	**F**	

What truth-values do we enter in the first row under 'A'? The combination of truth-values that would be displayed by entering **T** there is different from the combination that would be displayed by entering **F**. And the same holds for each row. So we need to list each of the four combinations of truth-values that 'B' and 'C' may have *twice* in order to represent all combinations of truth-values for the three atomic components.

A	B	C	(~ B ⊃ C) & (A ≡ B)
T	**T**	**T**	
T	**T**	**F**	
T	**F**	**T**	
T	**F**	**F**	
F	**T**	**T**	
F	**T**	**F**	
F	**F**	**T**	
F	**F**	**F**	

Extending this reasoning, we find that every time we add a new atomic sentence to the list the number of rows in the truth-table doubles. If **P** has **n** distinct atomic components, there are $2^{\mathbf{n}}$ different combinations of truth-values for its atomic components.[1]

In constructing a truth-table, we adopt a systematic method of listing the combinations of truth-values that the atomic components of a sentence **P** might have. We first list the atomic components of **P** to the left of the vertical line at the top of the truth-table, in alphabetical order.[2]

Under the first sentence letter listed, we write a column of $2^{\mathbf{n}}$ entries, the first half of which are **T**s and the second half of which are **F**s. In the second column the number of **T**s and **F**s being alternated is half the number alternated in the first column. In the column under the third sentence letter listed, the number of **T**s and **F**s being alternated will again be half the number in the second column. We repeat this process until a column has been entered under each sentence letter to the left of the vertical line. The column under the last sentence letter in this list will then consist of single **T**s alternating with single **F**s. For a truth-table with **n** distinct sentence letters, the first column consists of $2^{\mathbf{n}-1}$ **T**s alternating with $2^{\mathbf{n}-1}$ **F**s, the second of $2^{\mathbf{n}-2}$ **T**s alternating with $2^{\mathbf{n}-2}$ **F**s, and in general the *i*th column consists of $2^{\mathbf{n}-i}$ **T**s alternating with $2^{\mathbf{n}-i}$ **F**s.

Now we can complete the rest of the truth-table for '(~ B ⊃ C) & (A ≡ B)'. We first repeat under 'A', 'B', and 'C', wherever these occur, the columns we have already entered under these letters to the left of the vertical line:

A	B	C	(~	B	⊃	C)	&	(A	≡	B)
T	**T**	**T**		**T**		**T**		**T**		**T**
T	**T**	**F**		**T**		**F**		**T**		**T**
T	**F**	**T**		**F**		**T**		**T**		**F**
T	**F**	**F**		**F**		**F**		**T**		**F**
F	**T**	**T**		**T**		**T**		**F**		**T**
F	**T**	**F**		**T**		**F**		**F**		**T**
F	**F**	**T**		**F**		**T**		**F**		**F**
F	**F**	**F**		**F**		**F**		**F**		**F**

[1] $2^{\mathbf{n}}$ is 2 if $\mathbf{n} = 1$, 2×2 if $\mathbf{n} = 2$, $2 \times 2 \times 2$ if $\mathbf{n} = 3$, and so on. 2^0 is 1.

[2] This is an extended sense of 'alphabetical order' since some sentence letters have subscripts. In this order all the nonsubscripted letters appear first, then all letters subscripted with '1', then all letters subscripted with '2', and so on.

Next we may enter the column for the component '~ B' under its main connective, the tilde. In each row in which 'B' has the truth-value **T**, '~ B' has the truth-value **F**, and in each row in which 'B' has the truth-value **F**, '~ B' has the truth-value **T**:

A	B	C	(~	B	⊃	C)	&	(A	≡	B)
T	**T**	**T**	**F**	**T**		**T**		**T**		**T**
T	**T**	**F**	**F**	**T**		**F**		**T**		**T**
T	**F**	**T**	**T**	**F**		**T**		**T**		**F**
T	**F**	**F**	**T**	**F**		**F**		**T**		**F**
F	**T**	**T**	**F**	**T**		**T**		**F**		**T**
F	**T**	**F**	**F**	**T**		**F**		**F**		**T**
F	**F**	**T**	**T**	**F**		**T**		**F**		**F**
F	**F**	**F**	**T**	**F**		**F**		**F**		**F**

The column for '~ B ⊃ C' is entered under the horseshoe. A material biconditional has the truth-value **F** when its antecedent has the truth-value **T** and its consequent has the truth-value **F**, and it has the truth-value **T** in all other cases:

A	B	C	(~	B	⊃	C)	&	(A	≡	B)
T	**T**	**T**	**F**	**T**	**T**	**T**		**T**		**T**
T	**T**	**F**	**F**	**T**	**T**	**F**		**T**		**T**
T	**F**	**T**	**T**	**F**	**T**	**T**		**T**		**F**
T	**F**	**F**	**T**	**F**	**F**	**F**		**T**		**F**
F	**T**	**T**	**F**	**T**	**T**	**T**		**F**		**T**
F	**T**	**F**	**F**	**T**	**T**	**F**		**F**		**T**
F	**F**	**T**	**T**	**F**	**T**	**T**		**F**		**F**
F	**F**	**F**	**T**	**F**	**F**	**F**		**F**		**F**

We now enter the column for 'A ≡ B' in accordance with the characteristic truth-table for '≡':

A	B	C	(~	B	⊃	C)	&	(A	≡	B)
T	**T**	**T**	**F**	**T**	**T**	**T**		**T**	**T**	**T**
T	**T**	**F**	**F**	**T**	**T**	**F**		**T**	**T**	**T**
T	**F**	**T**	**T**	**F**	**T**	**T**		**T**	**F**	**F**
T	**F**	**F**	**T**	**F**	**F**	**F**		**T**	**F**	**F**
F	**T**	**T**	**F**	**T**	**T**	**T**		**F**	**F**	**T**
F	**T**	**F**	**F**	**T**	**T**	**F**		**F**	**F**	**T**
F	**F**	**T**	**T**	**F**	**T**	**T**		**F**	**T**	**F**
F	**F**	**F**	**T**	**F**	**F**	**F**		**F**	**T**	**F**

Remember that a material biconditional has the truth-value **T** on all truth-value assignments on which its immediate components have the same truth-value, and the truth-value **F** on all other truth-value assignments. Finally we enter the column for '(~ B ⊃ C) & (A ≡ B)' under its main connective, the ampersand:

							↓			
A	B	C	(~	B	⊃	C)	&	(A	≡	B)
T	**T**	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	**T**	**F**	**F**	**T**	**T**	**F**	**T**	**T**	**T**	**T**
T	**F**	**T**	**T**	**F**	**T**	**T**	**F**	**T**	**F**	**F**
T	**F**	**F**	**T**	**F**	**F**	**F**	**F**	**T**	**F**	**F**
F	**T**	**T**	**F**	**T**	**T**	**T**	**F**	**F**	**F**	**T**
F	**T**	**F**	**F**	**T**	**T**	**F**	**F**	**F**	**F**	**T**
F	**F**	**T**	**T**	**F**	**T**	**T**	**T**	**F**	**T**	**F**
F	**F**	**F**	**T**	**F**	**F**	**F**	**F**	**F**	**T**	**F**

We use arrows to indicate the main connective of the sentence. Each row of the truth-table displays, underneath the arrow, the truth-value that the sentence has on every truth-value assignment that assigns the truth-values displayed to the left of the vertical line to its atomic components.

Here is the truth-table for the sentence '[A ≡ (B ≡ A)] ∨ ~ C':

								↓		
A	B	C	[A	≡	(B	≡	A)]	∨	~	C
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**F**	**T**
T	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**F**
T	**F**	**T**	**T**	**F**	**F**	**F**	**T**	**F**	**F**	**T**
T	**F**	**F**	**T**	**F**	**F**	**F**	**T**	**T**	**T**	**F**
F	**T**	**T**	**F**	**T**	**T**	**F**	**F**	**T**	**F**	**T**
F	**T**	**F**	**F**	**T**	**T**	**F**	**F**	**T**	**T**	**F**
F	**F**	**T**	**F**	**F**	**F**	**T**	**F**	**F**	**F**	**T**
F	**F**	**F**	**F**	**F**	**F**	**T**	**F**	**T**	**T**	**F**

The column for '~ C' is constructed in accordance with the characteristic truth-table for the tilde. '~ C' has the truth-value **T** on all and only those truth-value assignments on which 'C' has the truth-value **F**. The column for '~ C' appears directly underneath the tilde. '(B ≡ A)' has the truth-value **T** for the combinations of truth-values displayed in the first two and last two rows of the truth-table, because 'B' and 'A' have the same truth-value in those rows, and the truth-value **F** for the other combinations.

Similarly '[A ≡ (B ≡ A)]' has the truth-value **T** on exactly those truth-value assignments on which 'A' and '(B ≡ A)' have the same truth-value. The column for '[A ≡ (B ≡ A)]' appears directly underneath its main connective, which is the first occurrence of the triple bar. '[A ≡ (B ≡ A)] ∨ ~ C' has the truth-value **T** on every truth-value assignment on which either '[A ≡ (B ≡ A)]' or '~ C' has the truth-value **T** and the truth-value **F** when both of its immediate components do. The truth-value of the entire sentence for each combination of truth-values assigned to its atomic components is written in the column directly underneath the wedge, the sentence's main connective.

Here is the truth-table for the sentence '~ [(U ∨ (W ⊃ ~ U)) ≡ W]':

U	W	↓ ~	[(U	∨	(W	⊃	~	U))	≡	W]
T	**T**	**F**	**T**	**T**	**T**	**F**	**F**	**T**	**T**	**T**
T	**F**	**T**	**T**	**T**	**F**	**T**	**F**	**T**	**F**	**F**
F	**T**	**F**	**F**	**T**	**T**	**T**	**T**	**F**	**T**	**T**
F	**F**	**T**	**F**	**T**	**F**	**T**	**T**	**F**	**F**	**F**

The column under the first occurrence of the tilde displays the truth-value of the entire sentence '~ [(U ∨ (W ⊃ ~ U)) ≡ W]' for each combination of truth-values that its atomic components might have. The truth-table tells us that '~ [(U ∨ (W ⊃ ~ U)) ≡ W]' has the truth-value **T** on those truth-value assignments on which either 'U' is assigned the truth-value **T** and 'W' is assigned the truth-value **F** or both 'U' and 'W' are assigned the truth-value **F**; the sentence is false on every other truth-value assignment.

Sometimes we are not interested in determining the truth-value of a sentence **P** on every truth-value assignment but are interested only in the truth-value of **P** on a particular truth-value assignment. In this case we may construct a shortened truth-table for **P** that records only the truth-values that its atomic components have on that truth-value assignment. For example, suppose we want to know the truth-value of '(A & B) ⊃ B' on a truth-value assignment that assigns **F** to 'A' and **T** to 'B' and all the other atomic sentences of *SL*. We head the shortened truth-table as before. We list only the combination of truth-values that 'A' and 'B' have on the assignment we are interested in:

A	B	(A	&	B)	↓ ⊃	B
F	**T**	**F**	**F**	**T**	**T**	**T**

Our table shows that '(A & B)' has the truth-value **F** on this truth-value assignment, for 'A' has the truth-value **F**. Since the antecedent of '(A & B) ⊃ B' has the truth-value **F** and the consequent the truth-value **T**, '(A & B) ⊃ B' has the truth-value **T**.

We emphasize that, when we want to determine the truth-value of a sentence on a particular truth-value assignment we display only the truth-values that the assignment assigns to the atomic components of the sentence for which we are constructing a truth-table.

To review: The truth-value of a sentence **P** on a truth-value assignment is determined by starting with the truth-values of the atomic components of **P** on the truth-value assignment and then using the characteristic truth-tables for the connectives of *SL* to compute the truth-values of larger and larger sentential components of **P** on the truth-value assignment. Ultimately we determine the truth-value of the largest sentential component of **P**, namely, **P** itself.

We also define the notions of being **true on a truth-value assignment** and **false on a truth-value assignment:**

> A sentence is *true on a truth-value assignment* if and only if it has the truth-value **T** on that truth-value assignment.
>
> A sentence is *false on a truth-value assignment* if and only if it has the truth-value **F** on that truth-value assignment.

3.1E EXERCISES

1. How many rows will be in the truth-table for each of the following sentences?

a. $A \equiv (\sim A \equiv A)$
*b. $[\sim D \ \& \ (B \vee G)] \supset [\sim (H \ \& \ A) \vee \sim D]$
c. $(B \ \& \ C) \supset [B \vee (C \ \& \sim C)]$

2. Construct truth-tables for the following sentences.

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

3. Construct shortened truth-tables to determine the truth-value of each of the following sentences on the truth-value assignment that assigns **T** to ‘B’ and ‘C’, and **F** to ‘A’ and to every other atomic sentence of *SL*.

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

In Chapter 1 we introduced the concepts of logical truth, logical falsity, and logical indeterminacy. Recall that a logically true sentence of English is one that cannot possibly be false. A sentence that is logically true (or logically false) may be so on purely truth-functional grounds. For example, we may symbolize 'Either Cynthia will get a job or Cynthia will not get a job' as 'C ∨ ~ C', and the truth-table for this sentence shows that it is true on every truth-value assignment:

C	C	↓ ∨	~	C
T	**T**	**T**	**F**	**T**
F	**F**	**T**	**T**	**F**

A sentence that is logically true on truth-functional grounds is a **truth-functionally true** sentence.

> A sentence **P** of *SL* is *truth-functionally true* if and only if **P** is true on every truth-value assignment.[3]

Alternatively, a sentence **P** is truth-functionally true if and only if there is no truth-value assignment on which **P** is false.

Once the truth-table for a sentence has been constructed, it is a simple matter to determine whether that sentence is truth-functionally true: the sentence is truth-functionally true if and only if the column of truth-values under its main connective consists solely of **T**s. Since the rows of the truth-table represent *all* combinations of truth-values that may be assigned to the sentence's atomic components by any truth-value assignment, the absence of **F**s under the main connective shows that there is no truth-value assignment on which the sentence is false.

Here is the truth-table for another truth-functionally true sentence:

X	Z	Z	↓ ⊃	(X	∨	Z)
T	**T**	**T**	**T**	**T**	**T**	**T**
T	**F**	**F**	**T**	**T**	**T**	**F**
F	**T**	**T**	**T**	**F**	**T**	**T**
F	**F**	**F**	**T**	**F**	**F**	**F**

[3]Truth-functionally true sentences are sometimes called *tautologies* or *truth-functionally valid* sentences. Truth-functionally false sentences (introduced shortly) are sometimes called *contradictions,* or *self-contradictory* sentences. Truth-functionally indeterminate sentences (also to be introduced) are sometimes called *contingent* sentences.

The column under the main connective of ‘Z ⊃ (X ∨ Z)’ contains only **T**s. Note that the immediate sentential components of a truth-functionally true sentence need not themselves be truth-functionally true.

Truth-functional falsity is also defined in terms of truth-value assignments.

> A sentence **P** of *SL* is *truth-functionally false* if and only if **P** is false on every truth-value assignment.

It follows that if **P** is truth-functionally false then there is no truth-value assignment on which **P** is true. We can show that a sentence of *SL* is truth-functionally false by constructing a truth-table for the sentence; if the column of truth-values under the sentence's main connective contains only **F**s, then the sentence is truth-functionally false. Here are truth-tables for two truth-functionally false sentences:

A	A	↓ &	~	A
T	**T**	**F**	**F**	**T**
F	**F**	**F**	**T**	**F**

H	K	[(H	∨	K)	⊃	~	(H	∨	K)]	↓ &	H
T	**T**	**T**	**T**	**T**	**F**	**F**	**T**	**T**	**T**	**F**	**T**
T	**F**	**T**	**T**	**F**	**F**	**F**	**T**	**T**	**F**	**F**	**T**
F	**T**	**F**	**T**	**T**	**F**	**F**	**F**	**T**	**T**	**F**	**F**
F	**F**	**F**	**F**	**F**	**T**	**T**	**F**	**F**	**F**	**F**	**F**

Note that the immediate sentential components of a truth-functionally false sentence need not themselves be truth-functionally false. Whenever we negate a truth-functionally true sentence, the result is a truth-functionally false sentence, as the following example shows:

A	↓ ~	(A	∨	~	A)
T	**F**	**T**	**T**	**F**	**T**
F	**F**	**F**	**T**	**T**	**F**

If we add another tilde to obtain ‘~ ~ (A ∨ ~ A)’, we will once again have a truth-functionally true sentence.

Although the two sentences ‘A ⊃ (B ⊃ A)’ and ‘(A ⊃ B) ⊃ A’ look very much alike, one is truth-functionally true and the other is not:

A	B	A	↓ $\supset$	(B	$\supset$	A)
T	**T**	**T**	**T**	**T**	**T**	**T**
T	**F**	**T**	**T**	**F**	**T**	**T**
F	**T**	**F**	**T**	**T**	**F**	**F**
F	**F**	**F**	**T**	**F**	**T**	**F**

A	B	(A	$\supset$	B)	↓ $\supset$	A
T	**T**	**T**	**T**	**T**	**T**	**T**
T	**F**	**T**	**F**	**F**	**T**	**T**
F	**T**	**F**	**T**	**T**	**F**	**F**
F	**F**	**F**	**T**	**F**	**F**	**F**

'A ⊃ (B ⊃ A)' is true on every truth-value assignment, whereas '(A ⊃ B) ⊃ A' is not. The latter sentence is **truth-functionally indeterminate.**

> A sentence **P** of *SL* is *truth-functionally indeterminate* if and only if **P** is neither truth-functionally true nor truth-functionally false.

A truth-functionally indeterminate sentence is true on at least one truth-value assignment and false on at least one truth-value assignment. Every atomic sentence of *SL* is truth-functionally indeterminate. For example, the truth-table for 'H' is

H	↓ H
T	**T**
F	**F**

'H' is true on every truth-value assignment on which it is assigned the truth-value **T**, and false on every other truth-value assignment. Truth-tables for several truth-functionally indeterminate sentences appeared in Section 3.1. Every sentence of *SL* is either truth-functionally true, truth-functionally false, or truth-functionally indeterminate.

Sometimes we can show that a sentence is not truth-functionally true or is not truth-functionally false by constructing a shortened truth-table. Consider the sentence '(A & ~ A) ∨ ~ A'. If this sentence is truth-functionally true, then there is no truth-value assignment on which it is false. So, if we can show that the sentence is false on at least one truth-value assignment, then we can conclude that it is not truth-functionally true. The following shortened truth-table shows this:

A	(A	&	~ A)	↓ ∨	~ A
T	**T**	**F**	**F T**	**F**	**F T**

This shortened truth-table shows that the sentence '(A & ~ A) ∨ ~ A' is false on every truth-value assignment that assigns the truth-value **T** to 'A'. Note that the shortened table shows *only* that '(A & ~ A) ∨ ~ A' is not truth-functionally true. The table does not show whether the sentence is truth-functionally false or truth-functionally indeterminate. However, it is easy to show that it is the latter by constructing a shortened truth-table in which the value under the main connective is **T**.

Similarly we may construct a shortened truth-table in order to show that 'J & (~ K ∨ ~ J)' is not truth-functionally false:

				↓					
J	K		J	&	(~	K	∨	~	J)
T	**F**		**T**	**T**	**T**	**F**	**T**	**F**	**T**

This truth-table shows that the sentence is true on every truth-value assignment that assigns **T** to 'J' and **F** to 'K'. We thus know that the sentence is either truth-functionally indeterminate or truth-functionally true.

There is a systematic way to develop a shortened truth-table that shows that a sentence is true on at least one truth-value assignment or false on at least one truth-value assignment. Let's first consider the previous example, in which we wanted to show that 'J & (~ K ∨ ~ J)' is true on at least one truth-value assignment. We start by placing a **T** under the main connective:

				↓					
J	K		J	&	(~	K	∨	~	J)
				T					

Because the main connective is an ampersand, we know that each conjunct must be true as well:

				↓					
J	K		J	&	(~	K	∨	~	J)
			T	**T**			**T**		

Whenever we place a **T** or **F** under a sentence letter, we repeat it under all occurrences of that sentence letter:

				↓					
J	K		J	&	(~	K	∨	~	J)
T			**T**	**T**			**T**		**T**

Once we have placed a **T** under 'J', we know that we must fill in an **F** under the tilde preceding 'J', since a negation is false if the negated sentence is true:

			↓					
J	K	J	&	(~	K	∨	~	J)
T		**T**	**T**			**T**	**F**	**T**

Now we have a true disjunction with one false disjunct, so we know that the other disjunct must be true (otherwise the disjunction could not be true):

			↓					
J	K	J	&	(~	K	∨	~	J)
T		**T**	**T**	**T**		**T**	**F**	**T**

And if '~ K' is true, then 'K' must be false:

			↓					
J	K	J	&	(~	K	∨	~	J)
T	**F**	**T**	**T**	**T**	**F**	**T**	**F**	**T**

Note that we also placed an **F** under the occurrence of 'K' to the left of the vertical bar. This completes our shortened truth-table, and we have shown that the sentence is not truth-functionally false.

Now consider the earlier example, in which we wanted to show that '(A & ~ A) ∨ ~ A' is false on at least one truth-value assignment (and therefore not truth-functionally true). We begin by placing an **F** under the sentence's main connective:

					↓		
A	(A	&	~	A)	∨	~	A
					F		

If a disjunction is false, both of its disjuncts must be false:

					↓		
A	(A	&	~	A)	∨	~	A
		F			**F**	**F**	

We have just recorded an **F** for '~ A', and since '~ A' occurs elsewhere in the sentence, we repeat the **F** there:

					↓		
A	(A	&	~	A)	∨	~	A
		F	**F**		**F**	**F**	

Note that we have now assigned the value **F** to one of the conjuncts of '(A & ~ A)', thus ensuring that the conjunction is false, so it won't matter if we end up assigning the value **T** to the other conjunct. Next we note that if '~ A' is false then 'A' must be true:

					↓		
A	(A	&	~	A)	∨	~	A
T	**T**	**F**	**F**	**T**	**F**	**F**	**T**

And this completes the shortened truth-table.

In these two examples, every addition to the table was dictated by some previous truth-value that had been entered: If a conjunction is true, both conjuncts must be true; if a disjunction is false, both disjuncts must be false; a negation is true if and only if the negated sentence is false; and a component of a sentence must have the same truth-value for each of its occurrences. But sometimes choices have to be made. For example, suppose we want to show that the sentence '(A ⊃ B) ≡ (B ⊃ A)' is not truth-functionally true. We can begin constructing a shortened truth-table by placing an **F** under the sentence's main connective:

					↓			
A	B	(A	⊃	B)	≡	(B	⊃	A)
					F			

At this point we have to make a choice, because there are two ways that a biconditional can be false. Either the first immediate component is true and the second false, or the first immediate component is false and the second true. There is no simple rule of thumb to follow in this case. So we'll try one of the possibilities and see where it leads:

					↓			
A	B	(A	⊃	B)	≡	(B	⊃	A)
			T		**F**		**F**	

Since '(B ⊃ A)' is false, we know that 'B' must be true and 'A' false. We'll add these values:

					↓			
A	B	(A	⊃	B)	≡	(B	⊃	A)
F	**T**		**T**		**F**	**T**	**F**	**F**

We also need to add the values under the other occurrences of 'A' and 'B'—but in doing so we must make sure that these values are consistent with the assignment of **T** to the conditional ‘(A ⊃ B)’:

					↓			
A	B	(A	⊃	B)	≡	(B	⊃	A)
F	**T**	**F**	**T**	**T**	**F**	**T**	**F**	**F**

Fortunately they are: A conditional with a false antecedent and a true consequent is itself true. So we have successfully completed the shortened table.

It turns out that we could have assigned **F** to the first immediate component of the biconditional and **T** to the second and produced another shortened truth-table representing a different set of truth-value assignments on which the biconditional is false. But sometimes, when we have a choice, one possible way of assigning truth-values won’t work while another one will. Suppose, for example, that we want to show that the sentence ‘(A ⊃ B) ⊃ (B ⊃ ~ A)’ is not truth-functionally false—that is, that there is at least one truth-value assignment on which it is true. We start with

					↓			
A	B	(A	⊃	B)	⊃	(B	⊃	~ A)
					T			

There are three ways in which a conditional can be true: Both the antecedent and consequent are true, or the antecedent is false and the consequent is true, or the antecedent is false and the consequent is false. We might try the first case first:

					↓			
A	B	(A	⊃	B)	⊃	(B	⊃	~ A)
			T		**T**		**T**	

We now have two true conditionals whose immediate components do not have truth-values. We’ll work with the first one, and again, let’s make its antecedent true and its consequent true:

					↓			
A	B	(A	⊃	B)	⊃	(B	⊃	~ A)
		T	**T**	**T**	**T**		**T**	

Filling in **T** under ‘A’ and ‘B’ wherever they occur—because ‘A’ and ‘B’ have each been assigned the truth-value **T**—we get

					↓			
A	B	(A	⊃	B)	⊃	(B	⊃	~ A)
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**

Now we must put **F** under the tilde:

A	B	(A	⊃	B)	↓ ⊃	(B	⊃	~	A)
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**F**	**T**

FAILURE!

The problem is that the conditional '(B ⊃ ~ A)' cannot be true if 'B' is true and '~ A' is false.

But we must not conclude that the sentence *cannot* be true. All we conclude is that we haven't come up with a way of assigning truth-values that will make it true. We can go back to

A	B	(A	⊃	B)	↓ ⊃	(B	⊃	~	A)
			T		**T**		**T**		

and try another way to make the conditional '(A ⊃ B)' true—say, by making 'A' false and 'B' true. This yields

A	B	(A	⊃	B)	↓ ⊃	(B	⊃	~	A)
F	**T**	**F**	**T**	**T**	**T**	**T**	**T**		**F**

and we can fill in a **T** under the tilde:

A	B	(A	⊃	B)	↓ ⊃	(B	⊃	~	A)
F	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**F**

Note that this time the conditional '(B ⊃ ~ A)' will be true since both of its immediate components are, so we have produced a shortened truth-table that shows the sentence is not truth-functionally false. But even if this hadn't worked, there are still other possibilities, including trying to make the entire sentence true by a different assignment of truth-values to its immediate components.[4]

Of course, we may fail even when we try all the possibilities—which means that, although we thought a sentence might be true (or false) on some truth-value assignment, we were incorrect. Here's a simple example: We'll try to

[4]Sometimes we have to try every possibility before coming up with a correct shortened truth-table (or concluding that there is no such table). The problem in constructing a shortened truth-table to show that a sentence can be true or that it can be false is one of a class of problems known to theoreticians as 'NP-complete problems'. These are problems for which the only known solutions guaranteed to produce a correct result are solutions that require us, in the worst case, to try every possibility.

produce a shortened truth-table with an assignment of truth-values that makes the sentence ‘A ⊃ A’ false:

A	A	⊃	A
		↓	
		F	

If the conditional is false, the antecedent must be true and the consequent false:

A	A	⊃	A
		↓	
	T	**F**	**F**

FAILURE!

We failed because ‘A’ cannot have two different truth-values on the same truth-value assignment. Here we have, in fact, tried all the possibilities for making the conditional false (the antecedent must be true and the conclusion must be false)—unsuccessfully. That’s as it should be, since the sentence is truth-functionally true.

3.2E EXERCISES

1. Construct a full truth-table for each of the following sentences of *SL*, and state whether the sentence is truth-functionally true, truth-functionally false, or truth-functionally indeterminate.

a. ~ A ⊃ A
*b. J ⊃ (K ⊃ J)
c. (A ≡ ~ A) ⊃ ~ (A ≡ ~ A)
*d. (E ≡ H) ⊃ (~ E ⊃ ~ H)
e. (~ B & ~ D) ∨ ~ (B ∨ D)
*f. ([(C ⊃ D) & (D ⊃ E)] & C) & ~ E
g. [(A ∨ B) & (A ∨ C)] ⊃ ~ (B & C)
*h. ~ [[(A ∨ B) & (B ∨ B)] & (~ A & ~ B)]
i. (J ∨ ~ K) ≡ ~ ~ (K ⊃ J)
*j. ~ B ⊃ [(B ∨ D) ⊃ D]
k. [(A ∨ ~ D) & ~ (A & D)] ⊃ ~ D
*l. (M ≡ ~ N) & (M ≡ N)

2. For each of the following sentences, either show that the sentence is truth-functionally true by constructing a full truth-table or show that the sentence is not truth-functionally true by constructing an appropriate shortened truth-table.

a. (F ∨ H) ∨ (~ F ≡ H)
*b. (F ∨ H) ∨ ~ (~ F ⊃ H)
c. ~ A ⊃ [(B & A) ⊃ C]
*d. A ≡ (B ≡ A)
e. [(C ∨ ~ C) ⊃ C] ⊃ C
*f. [C ⊃ (C ∨ ~ D)] ⊃ (C ∨ D)

3. Construct truth-tables to show that the following sentences of *SL* are truth-functionally true.

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

4. For each of the following sentences of *SL*, either show that the sentence is truth-functionally false by constructing a full truth-table or show that the sentence is not truth-functionally false by constructing an appropriate shortened truth-table.

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

5. Which of the following claims about sentences of *SL* are true? Explain.

a. A conjunction with one truth-functionally true conjunct must itself be truth-functionally true.
*b. A disjunction with one truth-functionally true disjunct must itself be truth-functionally true.
c. A material conditional with a truth-functionally true consequent must itself be truth-functionally true.
*d. A conjunction with one truth-functionally false conjunct must itself be truth-functionally false.
e. A disjunction with one truth-functionally false disjunct must itself be truth-functionally false.
*f. A material conditional with a truth-functionally false consequent must itself be truth-functionally false.
g. A sentence is truth-functionally true if and only if its negation is truth-functionally false.
*h. A sentence is truth-functionally indeterminate if and only if its negation is truth-functionally indeterminate.
i. A material conditional with a truth-functionally true antecedent must itself be truth-functionally true.
*j. A material conditional with a truth-functionally false antecedent must itself be truth-functionally false.

6. Where **P** and **Q** are sentences of *SL*, answer the following questions; explain your answers.
   a. Suppose that **P** is a truth-functionally true sentence and **Q** is a truth-functionally false sentence. On the basis of this information, can you determine whether **P** ≡ **Q** is truth-functionally true, false, or indeterminate? If so, which is it?
   *b. Suppose that **P** and **Q** are truth-functionally indeterminate sentences. Does it follow that **P** & **Q** is truth-functionally indeterminate?
   c. Suppose that **P** and **Q** are truth-functionally indeterminate. Does it follow that **P** ∨ **Q** is truth-functionally indeterminate?
   *d. Suppose that **P** is a truth-functionally true sentence and that **Q** is truth-functionally indeterminate. On the basis of this information, can you determine whether **P** ⊃ **Q** is truth-functionally true, false, or indeterminate? If so, which is it?

3.3 TRUTH-FUNCTIONAL EQUIVALENCE

We now introduce the concept of **truth-functional equivalence.**

> Sentences **P** and **Q** of *SL* are *truth-functionally equivalent* if and only if there is no truth-value assignment on which **P** and **Q** have different truth-values.

To show that **P** and **Q** are truth-functionally equivalent, we construct a single truth-table for both **P** and **Q** and show that in each row the two sentences have the same truth-value. The columns under the *main* connectives must be identical.

The sentences ‘A & A’ and ‘A ∨ A’ are truth-functionally equivalent, as shown by the following truth-table:

		↓			↓	
A	A	&	A	A	∨	A
T	**T**	**T**	**T**	**T**	**T**	**T**
F	**F**	**F**	**F**	**F**	**F**	**F**

On any truth-value assignment that assigns **T** to ‘A’, both sentences are true. On any truth-value assignment that assigns **F** to ‘A’, both sentences are false. The sentences ‘(W & Y) ⊃ H’ and ‘W ⊃ (Y ⊃ H)’ are also truth-functionally equivalent:

						↓			↓			
H	W	Y	(W	&	Y)	⊃	H	W	⊃	(Y	⊃	H)
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	**T**	**F**	**T**	**F**	**F**	**T**	**T**	**T**	**T**	**F**	**T**	**T**
T	**F**	**T**	**F**	**F**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**T**
T	**F**	**F**	**F**	**F**	**F**	**T**	**T**	**F**	**T**	**F**	**T**	**T**
F	**T**	**T**	**T**	**T**	**T**	**F**	**F**	**T**	**F**	**T**	**F**	**F**
F	**T**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**T**	**F**	**T**	**F**
F	**F**	**T**	**F**	**F**	**T**	**T**	**F**	**F**	**T**	**T**	**F**	**F**
F	**F**	**F**	**F**	**F**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**F**

The columns under the main connectives of ‘(W & Y) ⊃ H’ and ‘W ⊃ (Y⊃ H)’ are identical, which shows that the two sentences have the same truth-value on every truth-value assignment.

Now consider the following truth-table:

				↓					↓	
E	H	J	E	∨	H	(H	∨	J)	∨	E
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**F**	**T**	**T**
T	**F**	**T**	**T**	**T**	**F**	**F**	**T**	**T**	**T**	**T**
T	**F**	**F**	**T**	**T**	**F**	**F**	**F**	**F**	**T**	**T**
F	**T**	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**F**
F	**T**	**F**	**F**	**T**	**T**	**T**	**T**	**F**	**T**	**F**
F	**F**	**T**	**F**	**F**	**F**	**F**	**T**	**T**	**T**	**F**
F	**F**	**F**	**F**	**F**	**F**	**F**	**F**	**F**	**F**	**F**

The table shows that the sentences ‘E ∨ H’ and ‘(H ∨ J) ∨ E’ are not truth-functionally equivalent, for they have different truth-values on any truth-value assignment that assigns **F** to ‘E’ and ‘H’ and **T** to ‘J’. When a truth-table shows that two sentences are not truth-functionally equivalent, we will draw a box around a row of the truth-table in which the sentences do not have the same truth-value.

All truth-functionally true sentences are truth-functionally equivalent. This is because every truth-functionally true sentence has the truth-value **T** on every truth-value assignment. For example, ‘~ (C & ~ C)’ and ‘A ⊃ (B ⊃ A)’ are truth-functionally equivalent:

			↓						↓			
A	B	C	~	(C	&	~	C)	A	⊃	(B	⊃	A)
T	**T**	**T**	**T**	**T**	**F**	**F**	**T**	**T**	**T**	**T**	**T**	**T**
T	**T**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**T**	**T**	**T**	**T**
T	**F**	**T**	**T**	**T**	**F**	**F**	**T**	**T**	**T**	**F**	**T**	**T**
T	**F**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**T**	**F**	**T**	**T**
F	**T**	**T**	**T**	**T**	**F**	**F**	**T**	**F**	**T**	**T**	**F**	**F**
F	**T**	**F**	**T**	**F**	**F**	**T**	**F**	**F**	**T**	**T**	**F**	**F**
F	**F**	**T**	**T**	**T**	**F**	**F**	**T**	**F**	**T**	**F**	**T**	**F**
F	**F**	**F**	**T**	**F**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**F**

The columns under the main connectives are identical. Likewise, all truth-functionally false sentences are truth-functionally equivalent.

But not all truth-functionally indeterminate sentences are truth-functionally equivalent—for example,

			↓				↓	
B	D	B	&	D	~	B	&	D
T	**T**	**T**	**T**	**T**	**F**	**T**	**F**	**T**
T	**F**	**T**	**F**	**F**	**F**	**T**	**F**	**F**
F	**T**	**F**	**F**	**T**	**T**	**F**	**T**	**T**
F	**F**	**F**	**F**	**F**	**T**	**F**	**F**	**F**

On any truth-value assignment on which 'B' and 'D' are both true, or 'B' is false and 'D' is true, the sentences 'B & D' and '~ B & D' have different truth-values. Hence they are not truth-functionally equivalent.

If **P** and **Q** are not truth-functionally equivalent, we can construct a shortened truth-table to show this. The shortened truth-table will display a combination of truth-values for which one sentence is true and the other false. For example, the following shortened truth-table shows that 'A' and 'A ∨ B' are not truth-functionally equivalent:

		↓		↓	
A	B	A	A	∨	B
F	**T**	**F**	**F**	**T**	**T**

The shortened truth-table shows that, on any truth-value assignment that assigns **F** to 'A' and **T** to 'B', 'A' is false and 'A ∨ B' is true. Note that, if we construct a shortened truth-table that includes a row in which both sentences have the same truth-value, this is not sufficient to show that they are truth-functionally equivalent. This is because they are truth-functionally equivalent if and only if they have the same truth-value on *every* truth-value assignment. To show this, we must consider every combination of truth-values that their atomic components might have.

We can construct shortened truth-tables for two (or more) sentences in a systematic way, just as we did for single sentences in Section 3.2. For example, we could begin constructing the previous table by assigning the sentence 'A' the truth-value **F** and 'A ∨ B' the truth-value **T**:

		↓		↓	
A	B	A	A	∨	B
		F		**T**	

(We might first have tried to make 'A' true and 'A ∨ B' false, but this would not lead to a correct truth-table since we would have a false disjunction with a true disjunct.) Filling in **F** under all the other occurrences of 'A' yields

		↓		↓	
A	B	A	A	∨	B
F		**F**	**F**	**T**	

Now we can make 'B' true, which will secure the truth of the disjunction:

		↓		↓	
A	B	A	A	$\vee$	B
F	**T**	**F**	**F**	**T**	**T**

In Chapter 2 we noted that compound sentences whose main connective is 'unless' can be paraphrased either as disjunctions or as material conditionals. That is, English sentences of the form

p unless **q**

can be paraphrased and symbolized in all of the following ways:

<u>Either</u> **p** <u>or</u> **q**	$\mathbf{P} \vee \mathbf{Q}$
<u>If it is not the case that</u> **p** <u>then</u> **q**	$\sim \mathbf{P} \supset \mathbf{Q}$
<u>If it is not the case that</u> **q** <u>then</u> **p**	$\sim \mathbf{Q} \supset \mathbf{P}$

These paraphrases and symbolizations are all correct because, as we can now show, for any sentences **P** and **Q** of *SL*, the sentences $\mathbf{P} \vee \mathbf{Q}$, $\sim \mathbf{P} \supset \mathbf{Q}$, and $\sim \mathbf{Q} \supset \mathbf{P}$ are truth-functionally equivalent:

			↓				↓				↓	
P	**Q**	**P**	$\vee$	**Q**	~	**P**	$\supset$	**Q**	~	**Q**	$\supset$	**P**
T	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**F**	**T**	**T**	**T**
T	**F**	**T**	**T**	**F**	**F**	**T**	**T**	**F**	**T**	**F**	**T**	**T**
F	**T**	**F**	**T**	**T**	**T**	**F**	**T**	**T**	**F**	**T**	**T**	**F**
F	**F**	**F**	**F**	**F**	**T**	**F**	**F**	**F**	**T**	**F**	**F**	**F**

Note that the above table is not a truth-table for specific sentences of *SL*, because '**P**' and '**Q**' are not sentences of *SL* but metavariables ranging over sentences of *SL*.

Similarly, for any sentences **P** and **Q** of *SL*, $\sim (\mathbf{P} \,\&\, \mathbf{Q})$ and $\sim \mathbf{P} \vee \sim \mathbf{Q}$ are also truth-functionally equivalent:

		↓					↓		
P	**Q**	~	**(P**	**&**	**Q)**	~	**P**	$\vee$	~	**Q**
T	**T**	**F**	**T**	**T**	**T**	**F**	**T**	**F**	**F**	**T**
T	**F**	**T**	**T**	**F**	**F**	**F**	**T**	**T**	**T**	**F**
F	**T**	**T**	**F**	**F**	**T**	**T**	**F**	**T**	**F**	**T**
F	**F**	**T**	**F**	**F**	**F**	**T**	**F**	**T**	**T**	**F**

as are $\sim (\mathbf{P} \vee \mathbf{Q})$ and $\sim \mathbf{P} \ \& \sim \mathbf{Q}$:

P	Q	↓ ~	(P	∨	Q)	~	P	↓ &	~	Q
T	T	F	T	T	T	F	T	F	F	T
T	F	F	T	T	F	F	T	F	T	F
F	T	F	F	T	T	T	F	F	F	T
F	F	T	F	F	F	T	F	T	T	F

3.3E EXERCISES

1. Determine, by constructing full truth-tables, which of the following pairs of sentences of *SL* are truth-functionally equivalent.

a.	$\sim (A \ \& \ B)$	$\sim (A \vee B)$
*b.	$A \supset (B \supset A)$	$(C \ \& \sim C) \vee (A \supset A)$
c.	$K \equiv H$	$\sim K \equiv \sim H$
*d.	$C \ \& \ (B \vee A)$	$(C \ \& \ B) \vee A$
e.	$(G \supset F) \supset (F \supset G)$	$(G \equiv F) \vee (\sim F \vee G)$
*f.	$\sim C \supset \sim B$	$B \supset C$
g.	$\sim (H \ \& \ J) \equiv (J \equiv \sim K)$	$(H \ \& \ J) \supset \sim K$
*h.	$\sim (D \vee B) \supset (C \supset B)$	$C \supset (D \ \& \ B)$
i.	$[A \vee \sim (D \ \& \ C)] \supset \sim D$	$[D \vee \sim (A \ \& \ C)] \supset \sim A$
*j.	$A \supset [B \supset (A \supset B)]$	$B \supset [A \supset (B \supset A)]$
k.	$F \vee \sim (G \vee \sim H)$	$(H \equiv \sim F) \vee G$

2. For each of the following pairs of sentences of *SL*, either show that the sentences are truth-functionally equivalent by constructing a full truth-table or show that they are not truth-functionally equivalent by constructing an appropriate shortened truth-table.

a.	$G \vee H$	$\sim G \supset H$
*b.	$\sim (B \ \& \sim A)$	$A \vee B$
c.	$(D \equiv A) \ \& \ D$	$D \ \& \ A$
*d.	$F \ \& \ (J \vee H)$	$(F \ \& \ J) \vee H$
e.	$A \equiv (\sim A \equiv A)$	$\sim (A \supset \sim A)$
*f.	$\sim (\sim B \vee (\sim C \vee \sim D))$	$(D \vee C) \ \& \sim B$

3. Symbolize each of the following pairs of sentences and determine which of the pairs of sentences are truth-functionally equivalent by constructing truth-tables.

a. Unless the sky clouds over, the night will be clear and the moon will shine brightly.
The moon will shine brightly if and only if the night is clear and the sky doesn't cloud over.

*b. Although the new play at the Roxy is a flop, critics won't ignore it unless it is canceled.
The new play at the Roxy is a flop, and if it is canceled critics will ignore it.

c. If the *Daily Herald* reports on our antics, then the antics are effective.
If our antics aren't effective, then the *Daily Herald* won't report on them.

*d. The year 1972 wasn't a good vintage year, 1973 was, and neither 1974 nor 1975 was.
Neither 1974 nor 1972 was a good vintage year, and not both 1973 and 1975 were.

e. If Mary met Tom and she liked him, then Mary didn't ask George to the movies.
If Mary met Tom and she didn't like him, then Mary asked George to the movies.

*f. Either the blue team or the red team will win the tournament, and they won't both win.
The red team will win the tournament if and only if the blue team won't win the tournament.

4. Suppose that sentences **P** and **Q** are truth-functionally equivalent.

a. Are ~ **P** and ~ **Q** truth-functionally equivalent? Explain.

*b. Show that **P** and **P** & **Q** are also truth-functionally equivalent.

c. Show that ~ **P** ∨ **Q** is truth-functionally true.

5. Suppose we construct two truth-tables, one for 'A ∨ B' and another for 'B ∨ C', and that the columns of truth-values under the sentences' main connectives are identical. Does it follow that 'A ∨ B' and 'B ∨ C' are truth-functionally equivalent? Explain.

6. Show that for any sentences **P** and **Q** of *SL*, the following pairs of sentences are truth-functionally equivalent.

a.	**P** ≡ **Q**	(**P** ⊃ **Q**) & (**Q** ⊃ **P**)
*b.	**P** ⊃ **Q**	~ **Q** ⊃ ~ **P**
c.	**P** & (**Q** ∨ **R**)	(**P** & **Q**) ∨ (**P** & **R**)
*d.	**P** ∨ (**Q** & **R**)	(**P** ∨ **Q**) & (**P** ∨ **R**)
e.	~ (**P** ≡ **Q**)	~ **P** ≡ **Q**
*f.	**P** & (**Q** & **R**)	(**P** & **Q**) & **R**

3.4 TRUTH-FUNCTIONAL CONSISTENCY

To define truth-functional consistency, we need the notion of a *set* of sentences, informally introduced in Chapter 1. A set of sentences of *SL* is a collection, possibly empty, of zero or more sentences of *SL*, the members of the set. We can specify a finite set of sentences by listing the names of the sentences, separated by commas, within a pair of curly brackets. Thus {A, B ⊃ H, C ∨ A} is the set of sentences consisting of 'A', 'B ⊃ H', and 'C ∨ A'. We adopt the convention that *SL* sentences occurring between the curly brackets are being mentioned, so that we do not need to enclose them within quotation marks.

All sets of sentences of *SL* that have at least one member are nonempty sets of sentences. The empty set, denoted by '∅', has no members. In what follows we shall use the variable 'Γ' (gamma), with or without a subscript, to range over sets of sentences of *SL*.

We can now introduce **truth-functional consistency:**

A set of sentences of *SL* is *truth-functionally consistent* if and only if there is at least one truth-value assignment on which all the members of the set are true. A set of sentences of *SL* is *truth-functionally inconsistent* if and only if it is not truth-functionally consistent.

The set {A, B ⊃ H, B} is truth-functionally consistent, as is shown by the following truth-table:

A	B	H	↓ A	B	↓ ⊃	H	↓ B
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	**T**	**F**	**T**	**T**	**F**	**F**	**T**
T	**F**	**T**	**T**	**F**	**T**	**T**	**F**
T	**F**	**F**	**T**	**F**	**T**	**F**	**F**
F	**T**	**T**	**F**	**T**	**T**	**T**	**T**
F	**T**	**F**	**F**	**T**	**F**	**F**	**T**
F	**F**	**T**	**F**	**F**	**T**	**T**	**F**
F	**F**	**F**	**F**	**F**	**T**	**F**	**F**

The truth-table shows that, on any truth-value assignment on which 'A', 'B', and 'H' are all true, all three set members are true. So the set is truth-functionally consistent. We have drawn a box around the row of the truth-table that shows this (in this case, there is only one such row).

The set of sentences {L, L ⊃ J, ~ J} is truth-functionally inconsistent:

J	L	↓ L	L	↓ ⊃	J	↓ ~ J
T	**T**	**T**	**T**	**T**	**T**	**F T**
T	**F**	**F**	**F**	**T**	**T**	**F T**
F	**T**	**T**	**T**	**F**	**F**	**T F**
F	**F**	**F**	**F**	**T**	**F**	**T F**

In each row at least one of the three sentences has the truth-value **F** under its main connective. Hence there is no truth-value assignment on which all three set members are true. The following set of sentences is also truth-functionally inconsistent: {C ∨ ~ C, ~ C & D, ~ D}.

C	D	C	↓ ∨	~ C	~ C	↓ &	D	↓ ~ D
T	**T**	**T**	**T**	**F T**	**F T**	**F**	**T**	**F T**
T	**F**	**T**	**T**	**F T**	**F T**	**F**	**F**	**T F**
F	**T**	**F**	**T**	**T F**	**T F**	**T**	**T**	**F T**
F	**F**	**F**	**T**	**T F**	**T F**	**F**	**F**	**T F**

In this case it does not matter that one of the sentences, 'C ∨ ~ C', is true on every truth-value assignment. All that matters for establishing truth-functional inconsistency is that there is no truth-value assignment on which all three members are true.

We can show that a finite set of sentences of *SL* is truth-functionally consistent by constructing a shortened truth-table that displays one row in which all the set members are true. For instance, the following shortened truth-table shows that the set {(E ≡ H) ≡ E, H & ~ E} is truth-functionally consistent:

					↓			↓	
E	H	(E	≡	H)	≡	E	H	&	~ E
F	**T**	**F**	**F**	**T**	**T**	**F**	**T**	**T**	**T F**

Note that if we construct a shortened table that displays a row in which not all the members of the set are true, this is not sufficient to show that the set is truth-functionally inconsistent. This is because a set of sentences is truth-functionally inconsistent if and only if there is *no* truth-value assignment on which every member of the set is true. To show this, we have to consider every combination of truth-values that the atomic components of the set members might have.

3.4E EXERCISES

1. Construct full truth-tables for each of the following sets of sentences and indicate whether they are truth-functionally consistent or truth-functionally inconsistent.

a. {A ⊃ B, B ⊃ C, A ⊃ C}
*b. {B ≡ (J & K), ~ J, ~ B ⊃ B}
c. {~ [J ∨ (H ⊃ L)], L ≡ (~ J ∨ ~ H), H ≡ (J ∨ L)}
*d. {(A & B) & C, C ∨ (B ∨ A), A ≡ (B ⊃ C)}
e. {(J ⊃ J) ⊃ H, ~ J, ~ H}
*f. {U ∨ (W & H), W ≡ (U ∨ H), H ∨ ~ H}
g. {A, B, C}
*h. {~ (A & B), ~ (B & C), ~ (A & C), A ∨ (B & C)}
i. {(A & B) ∨ (C ⊃ B), ~ A, ~ B}
*j. {A ⊃ (B ⊃ (C ⊃ A)), B ⊃ ~ A}

2. For each of the following sets of sentences, either show that the set is truth-functionally consistent by constructing an appropriate shortened truth-table or show that the set is truth-functionally inconsistent by constructing a full truth-table.

a. {B ⊃ (D ⊃ E), ~ D & B}
*b. {H ≡ (~ H ⊃ H)}
c. {F ⊃ (J ∨ K), F ≡ ~ J}
*d. {~ (~ C ∨ ~ B) & A, A ≡ ~ C}
e. {(A ⊃ B) ≡ (~ B ∨ B), A}
*f. {H ⊃ J, J ⊃ K, K ⊃ ~ H}

3. Symbolize each of the following passages in *SL* and determine whether the resulting set of sentences is truth-functionally consistent. If the set is truth-functionally consistent, construct a shortened truth-table that shows this. If it is truth-functionally inconsistent, construct a full truth-table.

a. If space is infinitely divisible, then Zeno's paradoxes are compelling. Zeno's paradoxes are neither convincing nor compelling. Space is infinitely divisible.

*b. Newtonian mechanics can't be right if Einsteinian mechanics is. But Einsteinian mechanics is right if and only if space is non-Euclidean. Space is non-Euclidean, or Newtonian mechanics is correct.

c. Eugene O'Neil was an alcoholic. His plays show it. But *The Iceman Cometh* must have been written by a teetotaler. O'Neill was an alcoholic unless he was a fake.

*d. Neither sugar nor saccharin is desirable if and only if both are lethal. Sugar is lethal if and only if saccharin is desirable. Sugar is undesirable if and only if saccharin isn't lethal.

e. If the Red Sox win next Sunday, then if Joan bet $5 against them she'll buy Ed a hamburger. The Red Sox won't win, and Joan won't buy Ed a hamburger.

*f. Either Johnson or Hartshorne pleaded guilty, or neither did. If Johnson pleaded guilty, then the newspaper story is incorrect. The newspaper story is correct, and Hartshorne pleaded guilty.

4. Where **P** and **Q** are sentences of *SL*,

a. Prove that {**P**} is truth-functionally inconsistent if and only if ~ **P** is truth-functionally true.

*b. If {**P**} is truth-functionally consistent, must {~ **P**} be truth-functionally consistent as well? Show that you are right.

c. If **P** and **Q** are truth-functionally indeterminate, does it follow that {**P**, **Q**} is truth-functionally consistent? Explain your answer.

*d. Prove that if **P** ≡ **Q** is truth-functionally true then {**P**, ~ **Q**} is truth-functionally inconsistent.

3.5 TRUTH-FUNCTIONAL ENTAILMENT AND TRUTH-FUNCTIONAL VALIDITY

Truth-functional entailment is a relation that may hold between a sentence of *SL* and a set of sentences of *SL*.

> A set Γ of sentences of *SL truth-functionally entails* a sentence **P** of *SL* if and only if there is no truth-value assignment on which every member of Γ is true and **P** is false.

In other words, Γ truth-functionally entails **P** just in case **P** is true on every truth-value assignment on which every member of Γ is true. We have a special symbol for truth-functional entailment: the double turnstile '⊨'. The expression

$\Gamma \vDash \mathbf{P}$

is read

Γ truth-functionally entails **P**.

To indicate that Γ does not truth-functionally entail **P**, we write

$\Gamma \nvDash \mathbf{P}$

Thus

$\{A, B \& C\} \vDash B$

and

$\{A, B \vee C\} \nvDash B$

mean, respectively,

{A, B & C} truth-functionally entails 'B'

and

{A, B ∨ C} does not truth-functionally entail 'B'.

Here we have adopted the convention that, when using the turnstile notation, we drop the single quotation marks around the sentence following the turnstile. We also have a special abbreviation to indicate that a sentence is truth-functionally entailed by the empty set of sentences:

$\vDash \mathbf{P}$

The expression '$\vDash$ **P**' is an abbreviation for '$\varnothing \vDash$ **P**'. All and only truth-functionally true sentences are truth-functionally entailed by the empty set of sentences; the proof of this is left as an exercise in Section 3.6.

If Γ is a finite set, we can determine whether Γ truth-functionally entails a sentence **P** by constructing a truth-table for the members of Γ and for **P**. If there is a row in the truth-table in which all the members of Γ have the truth-value **T** and **P** has the truth-value **F**, then Γ does not truth-functionally entail **P**. If there is no such row, then Γ does truth-functionally entail **P**. We can establish that {A, B & C} $\vDash$ B by constructing the following truth-table:

			↓		↓		↓
A	B	C	A	B	&	C	B
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	**T**	**F**	**T**	**T**	**F**	**F**	**T**
T	**F**	**T**	**T**	**F**	**F**	**T**	**F**
T	**F**	**F**	**T**	**F**	**F**	**F**	**F**
F	**T**	**T**	**F**	**T**	**T**	**T**	**T**
F	**T**	**F**	**F**	**T**	**F**	**F**	**T**
F	**F**	**T**	**F**	**F**	**F**	**T**	**F**
F	**F**	**F**	**F**	**F**	**F**	**F**	**F**

There is only one row in which both members of {A, B & C} are true, namely, the row in which 'A', 'B', and 'C' all have the truth-value **T**. But since 'B' is true in this row, it follows that there is no truth-value assignment on which 'A' and 'B & C' are true and 'B' is false. Hence {A, B & C} ⊨ B.

The following truth-table shows that {W ∨ J, (W ⊃ Z) ∨ (J ⊃ Z), ~ Z} ⊨ ~ (W & J):

				↓					↓				↓		↓			
J	W	Z	W	∨	J	(W	⊃	Z)	∨	(J	⊃	Z)	~	Z	~	(W	&	J)
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**F**	**T**	**F**	**T**	**T**	**T**
T	**T**	**F**	**T**	**T**	**T**	**T**	**F**	**F**	**F**	**T**	**F**	**F**	**T**	**F**	**F**	**T**	**T**	**T**
T	**F**	**T**	**F**	**T**	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**F**	**F**	**T**
T	**F**	**F**	**F**	**T**	**T**	**F**	**T**	**F**	**T**	**T**	**F**	**F**	**T**	**F**	**T**	**F**	**F**	**T**
F	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**F**	**T**	**T**	**T**	**F**	**F**
F	**T**	**F**	**T**	**T**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**T**	**F**	**F**
F	**F**	**T**	**F**	**F**	**F**	**F**	**T**	**T**	**T**	**F**	**T**	**T**	**F**	**T**	**T**	**F**	**F**	**F**
F	**F**	**F**	**F**	**F**	**F**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**F**	**F**

The fourth and sixth rows are the only ones in which all the set members are true, and '~ (W & J)' is true in these rows as well. The following truth-table shows that {K ∨ J, ~ (K ∨ J)} ⊨ K:

			↓		↓				↓
J	K	K	∨	J	~	(K	∨	J)	K
T	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**T**
T	**F**	**F**	**T**	**T**	**F**	**F**	**T**	**T**	**F**
F	**T**	**T**	**T**	**F**	**F**	**T**	**T**	**F**	**T**
F	**F**	**F**	**F**	**F**	**T**	**F**	**F**	**F**	**F**

There is no row in which 'K ∨ J' and '~ (K ∨ J)' are both true and hence no truth-value assignment on which the set members are both true. Consequently there is no truth-value assignment on which the members of the set are both true and 'K' is false; so the set truth-functionally entails 'K'.

On the other hand, {A, B ∨ C} does *not* truth-functionally entail 'B'. The following shortened truth-table shows this:

			↓		↓		↓
A	B	C	A	B	∨	C	B
T	**F**	**T**	**T**	**F**	**T**	**T**	**F**

This shortened truth-table shows that 'A' and 'B ∨ C' are both true and 'B' is false on any truth-value assignment that assigns **T** to 'A' and 'C' and **F** to 'B'.

An **argument** of *SL* is a set of two or more sentences of *SL*, one of which is designated as the conclusion and the others as the premises.

An argument of *SL* is *truth-functionally valid* if and only if there is no truth-value assignment on which all the premises are true and the conclusion is false. An argument of *SL* is *truth-functionally invalid* if and only if it is not truth-functionally valid.

Put another way, an argument of *SL* is truth-functionally valid just in case the conclusion is true on every truth-value assignment on which all of the premises are true. This means that an argument is truth-functionally valid if and only if the set consisting of the premises of the argument truth-functionally entails the conclusion.

We can use full truth-tables to determine whether arguments with a finite number of premises are truth-functionally valid, and we can use shortened truth-tables to show truth-functionally invalid arguments with a finite number of premises are truth-functionally invalid. The argument

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

is truth-functionally valid, as the following truth-table shows:

F	G	F	↓ ≡	G	F	↓ ∨	G	F	↓ &	G
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	**F**	**T**	**F**	**F**	**T**	**T**	**F**	**T**	**F**	**F**
F	**T**	**F**	**F**	**T**	**F**	**T**	**T**	**F**	**F**	**T**
F	**F**	**F**	**T**	**F**	**F**	**F**	**F**	**F**	**F**	**F**

The first row displays the only combination of truth-values for the atomic components of these sentences for which the premises, ‘F ≡ G’ and ‘F ∨ G’, are both true, and the conclusion, ‘F & G’, is true in this row as well. Similarly, the argument

$$\begin{array}{l} (A \,\&\, G) \vee (B \supset G) \\ \sim G \vee B \\ \hline \sim B \vee G \end{array}$$

is truth-functionally valid, as the following truth-table establishes.

						↓						↓				↓	
A	B	G	(A	&	G)	∨	(B	⊃	G)	~	G	∨	B	~	B	∨	G
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**F**	**T**	**T**	**T**
T	**T**	**F**	**T**	**F**	**F**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**T**	**F**	**T**	**F**	**F**
T	**F**	**T**	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**T**
T	**F**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**
F	**T**	**T**	**F**	**F**	**T**	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**F**	**T**	**T**	**T**
F	**T**	**F**	**F**	**F**	**F**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**T**	**F**	**T**	**F**	**F**
F	**F**	**T**	**F**	**F**	**T**	**T**	**F**	**T**	**T**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**T**
F	**F**	**F**	**F**	**F**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**

The conclusion, '~ B ∨ G', is true on every truth-value assignment on which the premises are true.

The following argument is truth-functionally invalid:

D ≡ (~ W ∨ G)
G ≡ ~ D
―――――――
~ D

This is shown by the following truth-table:

				↓						↓			↓
D	G	W	D	≡	(~	W	∨	G)	G	≡	~ D		~ D
T	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**T**	**F**	**F**	**T**	**F T**
T	**T**	**F**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**F**	**F**	**T**	**F T**
T	**F**	**T**	**T**	**F**	**F**	**T**	**F**	**F**	**F**	**T**	**F**	**T**	**F T**
T	**F**	**F**	**T**	**T**	**T**	**F**	**T**	**F**	**F**	**T**	**F**	**T**	**F T**
F	**T**	**T**	**F**	**F**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**F**	**T F**
F	**T**	**F**	**F**	**F**	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**F**	**T F**
F	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**F**	**F**	**F**	**T**	**F**	**T F**
F	**F**	**F**	**F**	**F**	**T**	**F**	**T**	**F**	**F**	**F**	**T**	**F**	**T F**

The premises, 'D ≡ (~ W ∨ G)' and 'G ≡ ~ D', are both true on every truth-value assignment that assigns **T** to 'D' and **F** to 'G' and 'W', and the conclusion, '~ D', is false on these truth-value assignments.

If an argument is truth-functionally invalid, we can show this by constructing a shortened truth-table that displays a row in which the premises are true and the conclusion false. The argument

~ (B ∨ D)
~ H
―――――――
B

is truth-functionally invalid, as the following shortened truth-table shows:

B	D	H	~	(B	∨	D)	~	H	B
			↓				↓		↓
F	**F**	**F**	**T**	**F**	**F**	**F**	**T**	**F**	**F**

There is an obvious relationship between validity and entailment: an argument of *SL* that has a finite number of premises is truth-functionally valid if and only if the set consisting of the premises of the argument truth-functionally entails the conclusion of the argument. There is also a relation between an argument of *SL* and a sentence called its *corresponding material conditional,* namely, the argument is truth-functionally valid if and only if its corresponding material conditional is truth-functionally true. To form an argument's corresponding material conditional, we first need the concept of an *iterated conjunction.* The iterated conjunction of a sentence **P** is just **P**, while the iterated conjunction of sentences $\mathbf{P}_1$, $\mathbf{P}_2$, . . . , $\mathbf{P}_n$ is $(\ldots(\mathbf{P}_1 \,\&\, \mathbf{P}_2) \,\&\, \ldots \,\&\, \mathbf{P}_n)$. (We form a conjunction of the first sentence and the second sentence, then a conjunction of that conjunction and the third sentence, if any, and so on.) The *corresponding material conditional* for an argument of *SL* with a finite number of premises is the material conditional whose antecedent is the iterated conjunction of the argument's premises and whose consequent is the conclusion of the argument.[5] So the corresponding material conditional for the argument

$$\begin{array}{l} \mathbf{P}_1 \\ \cdot \\ \cdot \\ \cdot \\ \underline{\mathbf{P}_n} \\ \mathbf{Q} \end{array}$$

is

$$(\ldots(\mathbf{P}_1 \,\&\, \mathbf{P}_2) \,\&\, \ldots \,\&\, \mathbf{P}_n) \supset \mathbf{Q}$$

We will shortly prove that the argument is truth-functionally valid if and only if the corresponding material conditional is truth-functionally true, but first we will consider two examples.

[5]Strictly speaking, an argument with more than one premise will have more than one corresponding material conditional. This is because the premises of an argument can be conjoined in more than one order. But all the corresponding material conditionals for any one argument are truth-functionally equivalent, and so we speak loosely of *the* corresponding material conditional for a given argument.

We can show that the argument

$$\begin{array}{l} A \\ A \supset B \\ \hline B \end{array}$$

is truth-functionally valid by showing that the corresponding material conditional '[A & (A ⊃ B)] ⊃ B' is truth-functionally true:

A	B	[A	&	(A	⊃	B)]	↓ ⊃	B
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	**F**	**T**	**F**	**T**	**F**	**F**	**T**	**F**
F	**T**	**F**	**F**	**F**	**T**	**T**	**T**	**T**
F	**F**	**F**	**F**	**F**	**T**	**F**	**T**	**F**

There is no truth-value assignment on which 'A & (A ⊃ B)' is true and 'B' is false, which means that there is no truth-value assignment on which 'A' and 'A ⊃ B' are both true and 'B' is false. And we can show that the argument

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

is truth-functionally invalid by showing that the corresponding material conditional is not truth-functionally true. The following shortened truth-table shows this:

A	B	((~ A	≡	~ B)	&	(B ∨ A))	↓ ⊃	~ A
T	**T**	**F T**	**T**	**F T**	**T**	**T T T**	**F**	**F T**

The single row of this table represents truth-value assignments on which the antecedent is true and the consequent false. On these truth-value assignments the premises of the argument, '~ A ≡ ~ B' and 'B ∨ A', are both true and the conclusion, '~ A', is false. Hence the argument is truth-functionally invalid.

We now prove that an argument of *SL* with a finite number of premises is truth-functionally valid if and only if its corresponding material conditional is truth-functionally true.

Suppose that

$$\begin{array}{l} \mathbf{P}_1 \\ . \\ . \\ . \\ \mathbf{P_n} \\ \hline \mathbf{Q} \end{array}$$

is a truth-functionally valid argument of *SL*. Then there is no truth-value assignment on which $\mathbf{P}_1, \ldots, \mathbf{P}_n$ are all true and $\mathbf{Q}$ is false. Because the iterated conjunction $(\ldots(\mathbf{P}_1 \mathbin{\&} \mathbf{P}_2) \mathbin{\&} \ldots \mathbf{P}_n)$ has the truth-value **T** on a truth-value assignment if and only if all of $\mathbf{P}_1, \ldots, \mathbf{P}_n$ have the truth-value **T** on that assignment, it follows that there is no truth-value assignment on which the antecedent of the corresponding material conditional, $(\ldots(\mathbf{P}_1 \mathbin{\&} \mathbf{P}_2) \mathbin{\&} \ldots \mathbin{\&} \mathbf{P}_n) \supset \mathbf{Q}$, is true while the consequent is false. Thus, the material conditional is true on every truth-value assignment and is therefore truth-functionally true.

Now assume that $(\ldots(\mathbf{P}_1 \mathbin{\&} \mathbf{P}_2) \mathbin{\&} \ldots \mathbin{\&} \mathbf{P}_n) \supset \mathrm{Q}$ is truth-functionally true. Then there is no truth-value assignment on which the antecedent is true and the consequent false. But the iterated conjunction is true on a truth-value assignment if and only if the sentences $\mathbf{P}_1, \ldots, \mathbf{P}_n$ are all true. So there is no truth-value assignment on which $\mathbf{P}_1, \ldots, \mathbf{P}_n$ are all true and $\mathbf{Q}$ is false; hence the argument is truth-functionally valid.

3.5E EXERCISES

1. Construct truth-tables and state whether the following arguments are truth-functionally valid.

a. A ⊃ (H & J)
J ≡ H
~ J
____
~ A

*b. B ∨ (A & ~ C)
(C ⊃ A) ≡ B
~ B ∨ A
____
~ (A ∨ C)

c. (D ≡ ~ G) & G
(G ∨ [(A ⊃ D) & A]) ⊃ ~ D
____
G ⊃ ~ D

*d. ~ (Y ≡ A)
~ Y
~ A
____
W & ~ W

e. (C ⊃ D) ⊃ (D ⊃ E)
D
____
C ⊃ E

*f. $B \vee B$

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

C

g. $(G \equiv H) \vee (\sim G \equiv H)$

$(\sim G \equiv \sim H) \vee \sim (G \equiv H)$

*h. $[(J \ \& \ T) \ \& \ Y] \vee (\sim J \supset \sim Y)$

$J \supset T$

$T \supset Y$

$Y \equiv T$

i. $\sim \sim F \supset \sim \sim G$

$\sim G \supset \sim F$

$G \supset F$

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

$B \supset \sim B$

$C \vee A$

2. For each of the following arguments, either show that the argument is truth-functionally invalid by constructing an appropriate shortened truth-table or show that the argument is truth-functionally valid by constructing a full truth-table.

a. $(J \vee M) \supset \sim (J \ \& \ M)$

$M \equiv (M \supset J)$

$M \supset J$

*b. $B \ \& \ F$

$\sim (B \ \& \ G)$

G

c. $A \supset \sim A$

$(B \supset A) \supset B$

$A \equiv \sim B$

*d. $J \vee [M \supset (T \equiv J)]$

$(M \supset J) \ \& \ (T \supset M)$

$T \ \& \sim M$

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

$B \supset \sim B$

$\sim C \supset C$

3. Construct the corresponding material conditional for each of the following arguments. For each of the arguments, either show that the argument is truth-functionally invalid by constructing an appropriate shortened truth-table for the corresponding material conditional or show that the argument is truth-functionally valid by constructing a full truth-table for the corresponding material conditional.

a. $\underline{B \,\&\, C}$

$B \vee C$

*b. $K \equiv L$

$L \supset J$

$\underline{\sim J}$

$\sim K \vee L$

c. $(J \supset T) \supset J$

$\underline{(T \supset J) \supset T}$

$\sim J \vee \sim T$

*d. $\underline{(A \vee C) \,\&\, \sim H}$

$\sim C$

e. $B \,\&\, C$

$\underline{B \vee D}$

D

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

$\underline{B \supset (A \supset C)}$

$\sim A \equiv \sim B$

4. Symbolize each of the following arguments and use truth-tables to test for truth-functional validity. Use full truth-tables to establish truth-functional validity and shortened truth-tables to establish truth-functional invalidity.

a. 'Stern' means the same as 'star' if 'Nacht' means the same as 'day'. But 'Nacht' doesn't mean the same as 'day'; therefore 'Stern' means something different from 'star'.

*b. Many people believe that war is inevitable. But war is inevitable if and only if our planet's natural resources are nonrenewable. So many people believe that our natural resources are nonrenewable.

c. If Sophie is in her right mind she doesn't believe in trolls, and she is in her right mind. If Jason is in his right mind he doesn't believe in trolls, but he isn't in his right mind. So Sophie doesn't believe in trolls but Jason does.

d. Sophie doesn't believe in trolls, but she does believe in Bigfoot. Jason believes in both trolls and Bigfoot. If Sophie or Jason both believe in trolls, then neither is a critical thinker. Therefore, Sophie is a critical thinker but Jason isn't.

e. Computers can think if and only if they can have emotions. If computers can have emotions, then they can have desires as well. But computers can't think if they have desires. Therefore computers can't think.

*f. If the butler murdered Devon, then the maid is lying, and if the gardener murdered Devon, then the weapon was a slingshot. The maid is lying if and only if the weapon wasn't a slingshot, and if the weapon wasn't a slingshot, then the butler murdered Devon. Therefore the butler murdered Devon.

5. Where **P**, **Q**, and **R** are sentences of *SL*, prove each of the following.

*a. Show that {**P**} ⊨ **Q** and {**Q**} ⊨ **P** if and only if **P** and **Q** are truth-functionally equivalent.

b. Suppose that {**P**} ⊨ **Q** ∨ **R**. Does it follow that either {**P**} ⊨ **Q** or {**P**} ⊨ **R**? Show that you are right.

*c. Show that if {**P**} ⊨ **Q** and {**Q**} ⊨ **R**, then {**P**} ⊨ **R**.

3.6 TRUTH-FUNCTIONAL PROPERTIES AND TRUTH-FUNCTIONAL CONSISTENCY

In this section we show that the truth-functional concepts of truth-functional truth, truth-functional falsehood, truth-functional indeterminacy, truth-functional equivalence, truth-functional entailment, and truth-functional validity can all be explicated in terms of truth-functional consistency. This is important because in Chapter 4 we shall introduce an alternative test for truth-functional consistency, and the possibility of explicating the other concepts in terms of truth-functional consistency means that we shall be able to use the test we develop in Chapter 4 to determine whether other truth-functional properties of sentences and sets of sentences hold.

We will now state how each truth-functional concept other than consistency can be stated in terms of consistency, and prove each statement.

> A sentence **P** of *SL* is *truth-functionally false* if and only if {**P**} is truth-functionally inconsistent.

Proof: Assume that **P** is truth-functionally false. Then, by definition, there is no truth-value assignment on which **P** is true. Consequently, as **P** is the only member of the unit set {**P**}, there is no truth-value assignment on which every member of that set is true. So {**P**} is truth-functionally inconsistent. Now assume that {**P**} is truth-functionally inconsistent. Then, by definition, there is no truth-value assignment on which every member of {**P**} is true. Since **P** is the only member of its unit set, there is no truth-value assignment on which **P** is true. Hence **P** is truth-functionally false.

> A sentence **P** of *SL* is *truth-functionally true* if and only if {~ **P**} is truth-functionally inconsistent.

Proof: Assume that **P** is truth-functionally true. Then, by definition, **P** is true on every truth-value assignment. We know that a sentence is true on a truth-value assignment if and only if the negation of the sentence is false on that truth-value assignment. So it follows from our assumption that ~ **P** is false on every truth-value assignment; that is, there is no truth-value assignment on which ~ **P** is true. But then there is no truth-value assignment on which every member of {~ **P**} is true, which means that {~ **P**} is truth-functionally inconsistent. The proof of the converse, that if {~ **P**} is truth-functionally inconsistent then **P** is truth-functionally true, is left as an exercise.

> A sentence **P** of *SL* is *truth-functionally indeterminate* if and only if both {~ **P**} and {**P**} are truth-functionally consistent.

Proof: A sentence **P** is truth-functionally indeterminate if and only if **P** is neither truth-functionally true nor truth-functionally false, and hence, by the previous results, if and only if both {~ **P**} and {**P**} are truth-functionally consistent.

> Sentences **P** and **Q** of *SL* are *truth-functionally equivalent* if and only if {~ (**P** ≡ **Q**)} is truth-functionally inconsistent.

Proof: Where **P** and **Q** are sentences of *SL*, **P** ≡ **Q** is their *corresponding material biconditional*. It is straightforward to show that **P** and **Q** are truth-functionally equivalent if and only if their corresponding material biconditional is truth-functionally true. Assume that **P** and **Q** are truth-functionally equivalent. Then, by definition, **P** and **Q** have the same truth-value on every truth-value assignment. We know that a material biconditional has the truth-value **T** on every truth-value assignment on which its immediate sentential components have the same truth-value. It follows that **P** ≡ **Q** is true on every truth-value assignment and hence is truth-functionally true and therefore, by the second result above, {~ (**P** ≡ **Q**)} is truth-functionally inconsistent. The proof of the converse, that if {~ (**P** ≡ **Q**)} is truth-functionally inconsistent, **P** and **Q** are truth-functionally equivalent, is left as an exercise.

To make these results more concrete, we shall consider an example. The set {~ [(A ∨ B) ≡ (~ A ⊃ B)]} is truth-functionally inconsistent, as shown by the following truth-table:

A	B	↓ ~	[(A	∨	B)	≡	(~	A	⊃	B)]
T	**T**	**F**	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**T**
T	**F**	**F**	**T**	**T**	**F**	**T**	**F**	**T**	**T**	**F**
F	**T**	**F**	**F**	**T**	**T**	**T**	**T**	**F**	**T**	**T**
F	**F**	**F**	**F**	**F**	**F**	**T**	**T**	**F**	**F**	**F**

The set is truth-functionally inconsistent because there is no truth-value assignment on which every member of the set (in this case there is just one member) is true. From this we know the following:

1. '~ [(A ∨ B) ≡ (~ A ⊃ B)]' is truth-functionally false. (**P** is truth-functionally false if and only if {**P**} is truth-functionally inconsistent. Here {~ [(A ∨ B) ≡ (~ A ⊃ B)]} is truth-functionally inconsistent. Hence there is no truth-value assignment on which the only member of that set, '~ [(A ∨ B) ≡ (~ A ⊃ B)]', is true. That one member is thus truth-functionally false.)

2. '$(A \vee B) \equiv (\sim A \supset B)$' is truth-functionally true. (**P** is truth-functionally true if and only if {~ **P**} is truth-functionally inconsistent. We have just reasoned that '$\sim [(A \vee B) \equiv (\sim A \supset B)]$' is truth-functionally false. Hence the sentence of which it is the negation, '$(A \vee B) \equiv (\sim A \supset B)$', is true on every truth-value assignment—it is a truth-functionally true sentence.)
3. '$A \vee B$' and '$\sim A \supset B$' are truth-functionally equivalent. (**P** and **Q** are truth-functionally equivalent if and only if {~ (**P** ≡ **Q**)} is truth-functionally inconsistent. Since '$(A \vee B) \equiv (\sim A \supset B)$' is truth-functionally true, '$A \vee B$' and '$\sim A \supset B$' have the same truth-value on every truth-value assignment—they are truth-functionally equivalent.)

Of course, each of these claims can be directly verified by examining the truth-table, but our general proofs show that this is not necessary.

Next we relate the concepts of truth-functional entailment and truth-functional consistency. Where Γ is a set of sentences of *SL* and **P** is any sentence of *SL*, we may form a set that contains **P** and all the members of Γ. This set is represented as

Γ ∪ {**P**}

which is read as

the union of gamma and the unit set of **P**

Thus, if Γ is {A, A ⊃ B} and **P** is 'J', then Γ ∪ {**P**}—that is, {A, A ⊃ B} ∪ {J}—is {A, A ⊃ B, J}. Of course, if **P** is a member of Γ, then Γ ∪ {**P**} is identical with Γ. So {A, A ⊃ B} ∪ {A ⊃ B} is simply {A, A ⊃ B}. In the case where Γ is ∅ (the empty set), Γ ∪ {**P**} is simply {**P**}. This follows because ∅ contains no members.

We can now explicate truth-functional entailment in terms of truth-functional inconsistency:

A set Γ of sentences of *SL* truth-functionally entails a sentence **P** of *SL* if and only if Γ ∪ {~ **P**} is truth-functionally inconsistent.

Proof: Assume that Γ truth-functionally entails **P**. Then, by the definition of truth-functional entailment, there is no truth-value assignment on which all the members of Γ are true and **P** is false. We know that **P** is false on a truth-value assignment if and only if ~ **P** is true on that assignment, so it follows that there is no truth-value assignment on which all the members of Γ are true and ~ **P** is also true. Therefore, Γ ∪ {~ **P**} is truth-functionally inconsistent. The proof of the converse, that if Γ ∪ {~ **P**} is truth-functionally inconsistent then Γ ⊨ **P**, is left as an exercise.

An argument of *SL* is truth-functionally valid if and only if the set consisting of the premises of the argument and the negation of the conclusion of the argument is truth-functionally inconsistent.

Proof: This follows immediately from the previous result.

So the argument

$$\begin{array}{l} (A \supset D)\ \&\ H \\ F \vee H \\ \hline D \end{array}$$

is truth-functionally valid if and only if $\{(A \supset D)\ \&\ H,\ F \vee H,\ {\sim}\ D\}$ is truth-functionally inconsistent.

3.6E EXERCISES

1. Where **P** and **Q** are sentences of *SL* and Γ is a set of sentences of *SL*, prove each of the following:

a. If $\{{\sim}\ \mathbf{P}\}$ is truth-functionally inconsistent, then **P** is truth-functionally true.

*b. If $\mathbf{P} \equiv \mathbf{Q}$ is truth-functionally true, then **P** and **Q** are truth-functionally equivalent.

c. If $\Gamma \cup \{{\sim}\ \mathbf{P}\}$ is truth-functionally inconsistent, then $\Gamma \vDash \mathbf{P}$.

2. Where Γ is a set of sentences of *SL* and **P** and **Q** are sentences of *SL*, prove each of the following:

a. A sentence **P** is truth-functionally true if and only if $\varnothing \vDash \mathbf{P}$.

*b. $\Gamma \vDash \mathbf{P} \supset \mathbf{Q}$ if and only if $\Gamma \cup \{\mathbf{P}\} \vDash \mathbf{Q}$.

c. If Γ is truth-functionally inconsistent, then Γ truth-functionally entails every sentence of *SL*.

*d. For any set Γ of sentences of *SL* and any truth-functionally false sentence **P** of *SL*, $\Gamma \cup \{\mathbf{P}\}$ is truth-functionally inconsistent.

3. Where Γ is a set of sentences of *SL* and **P** and **Q** are sentences of *SL*, prove each of the following:

a. If Γ is truth-functionally consistent and **P** is truth-functionally true, then $\Gamma \cup \{\mathbf{P}\}$ is truth-functionally consistent.

*b. If $\Gamma \vDash \mathbf{P}$ and $\Gamma \vDash {\sim}\ \mathbf{P}$, then Γ is truth-functionally inconsistent.

4. Where Γ and Γ′ are sets of sentences of *SL* and **P**, **Q** , and **R** are sentences of *SL*, prove each of the following:

a. If $\{\mathbf{P}\} \vDash \mathbf{Q}$ and $\{{\sim}\ \mathbf{P}\} \vDash \mathbf{R}$, then $\mathbf{Q} \vee \mathbf{R}$ is truth-functionally true.

*b. If **P** and **Q** are truth-functionally equivalent, then $\{\mathbf{P}\} \vDash \mathbf{R}$ if and only if $\{\mathbf{Q}\} \vDash \mathbf{R}$.

c. If $\Gamma \vDash \mathbf{P}$ and $\Gamma' \vDash \mathbf{Q}$, then $\Gamma \cup \Gamma' \vDash \mathbf{P}\ \&\ \mathbf{Q}$, where $\Gamma \cup \Gamma'$ is the set that contains all the sentences in Γ and all the sentences in Γ′.

GLOSSARY

TRUTH-FUNCTIONAL TRUTH: A sentence **P** of *SL* is *truth-functionally true* if and only if **P** is true on every truth-value assignment.

TRUTH-FUNCTIONAL FALSITY: A sentence **P** of *SL* is *truth-functionally false* if and only if **P** is false on every truth-value assignment.

TRUTH-FUNCTIONAL INDETERMINACY: A sentence **P** of *SL* is *truth-functionally indeterminate* if and only if **P** is neither truth-functionally true nor truth-functionally false.

TRUTH-FUNCTIONAL EQUIVALENCE: Sentences **P** and **Q** of *SL* are *truth-functionally equivalent* if and only if there is no truth-value assignment on which **P** and **Q** have different truth-values.

TRUTH-FUNCTIONAL CONSISTENCY: A set of sentences of *SL* is *truth-functionally consistent* if and only if there is at least one truth-value assignment on which all the members of the set are true. A set of sentences of *SL* is *truth-functionally inconsistent* if and only if the set is not truth-functionally consistent.

TRUTH-FUNCTIONAL ENTAILMENT: A set Γ of sentences of *SL truth-functionally entails* a sentence **P** of *SL* if and only if there is no truth-value assignment on which every member of Γ is true and **P** is false.

TRUTH-FUNCTIONAL VALIDITY: An argument of *SL* is *truth-functionally valid* if and only if there is no truth-value assignment on which all the premises are true and the conclusion is false. An argument of *SL* is *truth-functionally invalid* if and only if it is not truth-functionally valid.