Chapter

3

SENTENTIAL LOGIC: SEMANTICS

3.1 TRUTH-VALUE ASSIGNMENTS AND TRUTH-TABLES FOR SENTENCES

In Chapter 1 we introduced logical concepts such as logical truth and deductive validity and used them to evaluate sentences and arguments stated in English. In this chapter we shall develop formal tests for truth-functional versions of the concepts introduced in Section 1.4—specifically truth-functional truth, falsity, and indeterminacy; truth-functional consistency; truth-functional entailment; and truth-functional validity. All these concepts fall within the realm of *semantics:* They concern the truth-values and truth-conditions of sentences. Before defining these truth-functional concepts for sentences and arguments of *SL*, 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 determined completely 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. And, if the immediate sentences of a truth-functionally compound sentence are themselves truth-functionally compound, we can use the information in the characteristic truth-tables to determine how the truth-value of each immediate component depends on the truth-values of *its* immediate components, and so on until we arrive at atomic components.

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

> A truth-value assignment is an assignment of truth-values (Ts or Fs) to the atomic sentences of *SL*.

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 of *SL*. Thus a truth-value assignment gives a *complete* description of a way the world might be. It tells us of each atomic sentence of *SL* whether or not that sentence is true. The truth-values of truth-functionally compound sentences of *SL* are determined uniquely and completely by the truth-values of their atomic components. Because every atomic sentence of *SL* is assigned a truth-value by a truth-value assignment, it follows that every truth-functionally compound sentence also has a truth-value 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 each 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 there are four combinations of truth-values that these components might have, as indicated in the four rows of the table. (Rows in truth-tables go from left to right; columns go from top to bottom.) 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 upon only the truth-values that its atomic components have on that assignment (and not, say, on the truth-value of 'D'), the four combinations that we have displayed will allow us to determine the truth-value of '∼ B ⊃ C' on any truth-value assignment. That is, no matter which of the infinitely many truth-value assignments we might select, that truth-value assignment will assign one of the four pairs of truth-values displayed in the table to 'B' and 'C'.

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 might have. There is a simple way to do this. Consider first the case in which **P** has one atomic component. There are two different combinations of 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 were to add a third sentence letter to the truth-table for '∼ B ⊃ C':

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

What truth-tables 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 we see that 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** atomic components, there are $2^{\mathbf{n}}$ different combinations of truth-values for its atomic components.[1] (If the same sentence letter occurs more than once in **P**, we do not count each occurrence as a different atomic component of **P**. To determine the number of atomic components, we count the number of *different* sentence letters that occur in **P**.)

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, 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. Thus, for a truth-table with **n** 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. (Note that $2^0 = 1$.)

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 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**

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

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

[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.

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	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

The truth-values of the immediate components of 'A ≡ B' for each row have been recorded, so we can now complete 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 for which a truth-table has been constructed. Each row of the truth-table displays, underneath the arrow, the truth-value that the sentence has on every truth-value assignment that assigns to the atomic components of that sentence the truth-values displayed to the left of the vertical line.

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**, and '~ C' has the truth-value **F** on every assignment on which 'C' has the truth-value **T**. The column for '~ C' appears directly underneath the tilde. The immediate components of '(B ≡ A)' are 'B' and 'A'. The characteristic truth-value for '≡' tells us that a material biconditional has the truth-value **T** on all and only those truth-value assignments on which both of its immediate sentential components have the same truth-value (both have the truth-value **T** or both have the truth-value **F**). Thus '(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 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 exactly those truth-value assignments on which at least one disjunct has the truth-value **T**. The disjuncts are '[A ≡ (B ≡ A)]' and '~ C'. So [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**. Where both disjuncts have the truth-value **F**, so does '[A ≡ (B ≡ A)] ∨ ~ C'. 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 represents 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-values of a sentence **P** for every truth-value assignment but are interested only in the truth-value of **P** on a particular truth-value assignment. In that case we may construct a shortened truth-table for **P** that records only the truth-values that its atomic components are assigned by 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, with the atomic components of the sentence to the left of the vertical line and '(A & B) ⊃ B' itself to the right. We list only one combination of truth-values for 'A' and 'B', namely, the truth-values they have on the assignment we are interested in:

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

The truth-values of '(A & B)' and '(A & B) ⊃ B' are determined in accordance with the characteristic truth-tables, as before. Thus '(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 do not display the full truth-value assignment in question. Truth-value assignments assign truth-values to *every* atomic sentence of *SL*. Rather, we display only the combinations of truth-values that the atomic components of the sentence in question have on the assignment. There is no loss here because the truth-value of a sentence on a truth-value assignment depends *only* upon the truth-values of its atomic components on that assignment. Conversely each row of a truth-table for a sentence gives information about infinitely many truth-value assignments. It tells us the truth-value of the sentence on every truth-value assignment that assigns to the atomic components of the sentence the combination of truth-values displayed in that row (there are infinitely many such assignments).

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. This procedure is used in the construction of a truth-table for **P**, where each row displays a different combination of truth-values for the atomic components of **P**. The truth-value of **P** for each such combination is recorded directly underneath the main connective of **P** in the row representing that combination. (If **P** is atomic, the truth-value is recorded under **P**.)

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

3.1E EXERCISES

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

a. A ≡ (~ A ≡ A)
*b. [~ D & (B ∨ G)] ⊃ [~ (H & A) ∨ ~ D]
c. (B & C) ⊃ [B ∨ (C & ~ 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. $\sim (B \supset \sim A) \,\&\, [C \equiv (A \,\&\, B)]$

4. Construct a truth-table for each of the sentences in Exercise 1 in Section 2.2E.

5. Construct a truth-table for each of the sentences in Exercise 3 in Section 2.2E.

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** 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

Γ ⊨ **P**

is read

Γ truth-functionally entails **P**.

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

Γ ⊭ **P**

Thus

{A, B & C} ⊨ 'B'

and

{A, B ∨ C} ⊭ 'B'

mean, respectively,

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

and

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

Henceforth we adopt 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:

$\models$ **P**

The expression '$\models$ **P**' is an abbreviation for '$\varnothing \models$ **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 **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 truth-functionally entails **P**. We can see that {A, B & C} $\models$ B by checking 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 combination of truth-values for the atomic components of all these sentences that will make both 'A' and 'B & C' true and 'B' false. Hence there is no truth-value assignment on which 'A' and 'B & C' are true and 'B' is false: {A, B & C} $\models$ B.

In the same way we can show that {W ∨ J, (W ⊃ Z) ∨ (J ⊃ Z), ~ Z} $\models$ ~ (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; '~ (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 truth-table shows this:

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

The circled row 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 group 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.

Thus an argument of *SL* is truth-functionally valid just in case on every truth-value assignment on which the premises are true the conclusion is true as well. 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.

The argument

F ≡ G
F ∨ G
―――――
F & G

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 lists 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; the conclusion, 'F & G', is true in this row as well. Similarly the argument

(A & G) ∨ (B ⊃ G)
~ G ∨ B
―――――
~ B ∨ G

is truth-functionally valid:

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.

Where 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

For any argument of *SL* that has a finite number of premises, we may form a sentence called the *corresponding material conditional*, and that sentence is truth-functionally true if and only if the argument is truth-functionally valid. First, we may form an *iterated conjunction* $(\ldots(\mathbf{P}_1 \;\&\; \mathbf{P}_2) \;\&\; \ldots \;\&\; \mathbf{P}_n)$ from the sentences $\mathbf{P}_1, \ldots, \mathbf{P}_n$. The iterated conjunction for the sentences '~ (A ⊃ B)', 'D', and 'J ∨ H' is '((~ (A ⊃ B) & D) & (J ∨ H))'. The corresponding material conditional for an argument is then formed by constructing a material conditional with the iterated conjunction of the premises as antecedent and the conclusion of the argument as consequent. The corresponding material conditional for the argument

~ (A ⊃ B)
D
J ∨ H
―――――
~ H ∨ ~ A

is '[[~ (A ⊃ B) & D] & (J ∨ H)] ⊃ (~ H ∨ ~ A)', and the corresponding material conditional for the argument

A
A ⊃ B
―――――
B

is '[A & (A ⊃ B)] ⊃ B'.[5]

An argument with a finite number of premises is truth-functionally valid if and only if its corresponding material conditional is truth-functionally true (see Exercise 5). We can show that the argument

A
A ⊃ B
―――――
B

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

~ A ≡ ~ B
B ∨ A
―――――
~ A

[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.

is truth-functionally invalid by showing that the corresponding material conditional is not truth-functionally true:

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

The first row 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.

3.5E EXERCISES

1. Use truth-tables to determine 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 ∨ B
[~ B ⊃ (~ D ∨ ~ C)] & [(~ D ∨ C) ∨ (~ B ∨ C)]
———
C

g. (G ≡ H) ∨ (~ G ≡ H)
———
(~ G ≡ ~ H) ∨ ~ (G ≡ H)

*h. [(J & T) & Y] ∨ (~ J ⊃ ~ Y)
J ⊃ T
T ⊃ Y
———
Y ≡ T

i. ~ ~ F ⊃ ~ ~ G
~ G ⊃ ~ F
———
G ⊃ F

*j. [A & (B ∨ C)] ≡ (A ∨ B)
B ⊃ ~ B
———
C ∨ 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 ∨ M) ⊃ ~ (J & M)
M ≡ (M ⊃ J)
———
M ⊃ J

*b. B & F
~ (B & G)
———
G

c. A ⊃ ~ A
(B ⊃ A) ⊃ B
———
A ≡ ~ B

*d. J ∨ [M ⊃ (T ≡ J)]
(M ⊃ J) & (T ⊃ M)
———
T & ~ M

e. A & ~ [(B & C) ≡ (C ⊃ A)]
B ⊃ ~ B
———
~ C ⊃ 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. B & C
———
B ∨ C

*b. K ≡ L
L ⊃ J
~ J
———
~ K ∨ L

c. (J ⊃ T) ⊃ J
(T ⊃ J) ⊃ T
———
~ J ∨ ~ T

*d. (A ∨ C) & ~ H
———
~ C

e. B & C
B ∨ D
———
D

*f. ~ [A ∨ ~ (B ∨ ~ C)]
B ⊃ (A ⊃ C)
———
~ A ≡ ~ B

4. Symbolize each of the following arguments and use truth-tables to test for truth-functional validity.

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. Thirty days hath September, April, and November. But February has forty days, since April has thirty days if and only if May doesn't, and May has thirty days if November does.

*d. The town hall is now a grocery store, and, unless I'm mistaken, the little red schoolhouse is a movie theater. No, I'm not mistaken. The old schoolhouse is a boutique, and the old theater is an elementary school if the little red schoolhouse is a movie theater. So the little red schoolhouse is a movie theater.

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.a. Show that $(\ldots(\mathbf{P}_1 \,\&\, \mathbf{P}_2) \,\&\, \ldots \,\&\, \mathbf{P}_n) \supset \mathbf{Q}$ is truth-functionally true if and only if

$\mathbf{P}_1$

$\vdots$

$\mathbf{P}_n$

$\mathbf{Q}$

is truth-functionally valid.

*b. Show that $\{\mathbf{P}\} \models \mathbf{Q}$ and $\{\mathbf{Q}\} \models \mathbf{P}$ if and only if $\mathbf{P}$ and $\mathbf{Q}$ are truth-functionally equivalent.

c. Suppose that $\{\mathbf{P}\} \models \mathbf{Q} \vee \mathbf{R}$. Does it follow that either $\{\mathbf{P}\} \models \mathbf{Q}$ or $\{\mathbf{P}\} \models \mathbf{R}$? Show that you are right.

*d. Show that if $\{\mathbf{P}\} \models \mathbf{Q}$ and $\{\mathbf{Q}\} \models \mathbf{R}$, then $\{\mathbf{P}\} \models \mathbf{R}$.