Chapter 1

Sentential Logic: Semantics

The syntax for a language is concerned with the vocabulary and the rules for what counts as a well-formed expression of the language. The semantics is concerned with the meaning and truth conditions of the expressions of the language. The meanings of sentences in *SL* are always truth values, and every sentence has one, and only one, of two truth values, either 1 (true) or 0 (false).

1.1 Truth Tables for Connectives

Sentences of *SL* can be analyzed using truth-tables. Truth-tables are extremely powerful tools—they provide a foolproof test for all those concepts introduced in chapter **??**, like validity, consistency, and logical truth. We'll start with the most basic truth-tables, which are the *characteristic truth-tables* for the logical connectives of *SL*.

The simplest is the negation. A negation is true whenever the negated sentence is false, and false whenever the negated sentence is true.

α	$\neg\alpha$
1	0
0	1

Conjunctions are true when and only when both conjuncts are true. So, the characteristic truth table for the conjunction looks like this:

α	β	$\alpha \mathbin{\&} \beta$
1	1	1
1	0	0
0	1	0
0	0	0

Disjunctions are true whenever at least one of the disjuncts are true:

α	β	$\alpha \vee \beta$
1	1	1
1	0	1
0	1	1
0	0	0

A conditional is false only when the antecedent is true and the consequent is false.

α	β	$\alpha \supset \beta$
1	1	1
1	0	0
0	1	1
0	0	1

Finally, biconditionals are true whenever the sentences joined by the biconditional operator have the same truth value, and false when they have different truth values.

α	β	$\alpha \equiv \beta$
1	1	1
1	0	0
0	1	0
0	0	1

1.2 Truth Tables for Complex Sentences

The characteristic truth-tables for the logical connectives show us how to determine the truth value for any complex sentence, given the truth values of its constituent simple sentences. The truth values of those simple sentences are provided by an interpretation, also called a valuation or *model*. A model contains all of the information that is required to determine the truth values of every formula in a language. How complex a model needs to be depends on the language; models in *SL* are very simple. Remember that the logical connectives are truth-functional, which means that the truth-values of the complex sentences formed using those connectives are completely determined by the truth-values of the constituent simple sentences. So, to determine the truth-values of all of the formulas, however complex, of *SL*, we only need to assign truth-values to all of the simple sentences of *SL*.[1]

A model is a function that maps atomic formulas of *SL* to truth-values. Thus, a model determines the truth-values of every sentence in *SL*, not just the simple sentences. A partial model assigns truth-values to only some of the sentences in *SL*. Whenever we use the term 'model', we will almost always mean a partial model, since we aren't generally concerned about the truth values of all of the sentences in *SL*, but only those that are contained in the sentence or argument that we are analyzing.

Here are two ways to express models. The first uses standard function notation, using 'v' for 'valuation':

$$v_1(A) = 1, v_1(B) = 0, v_1(C) = 1$$

This says that valuation v_1 assigns 1 to **A** and **C**, and 0 to **B**. The same information can be expressed in a table:

	A	B	C
v_1	1	0	1

A truth-table for a sentence shows its truth-value for every partial model. Let's begin by constructing a very simple truth table for a sentence containing one simple sentence, '$A \supset A$'. Begin simply by writing down the sentence.

[1] Models in *SL* are often called "truth-value assignments."

$$A \quad \supset \quad A$$

Then, to the left of the sentence, write all of the simple sentence letters in alphabetical order, like this:

$$\begin{array}{cccc} A & A & \supset & A \end{array}$$

Now draw a line underneath all of that.

$$\begin{array}{cccc} A & A & \supset & A \\ \hline \end{array}$$

The next thing to do is to determine how many rows are needed. We need one row for each partial model. In this case, since we only have one simple sentence, there are only two models; one in which ‘A’ is true and another in which it is false.

Now that we know we will need two rows, we draw a vertical line between the complex sentence and the simple sentence letters. The line should extend down for however many rows are needed.

$$\begin{array}{c|ccc} A & A & \supset & A \\ \hline & & & \\ & & & \end{array}$$

Now, we list our truth value assignments for the simple sentences. In this case, underneath the ‘A’ that is to the left of the vertical line, we’ll write ‘1’ and ‘0’, like this:

$$\begin{array}{c|ccc} A & A & \supset & A \\ \hline 1 & & & \\ 0 & & & \end{array}$$

Next, copy the column under each sentence letter at the left to every occurrence of that sentence letter in the complex sentence. That's simple in this case, since we only have one simple sentence.

A	A	⊃	A
1	1		1
0	0		0

Now, beginning with the lowest level of parentheses, determine the value of the complex sentences. On the top row, we have a conditional with a true antecedent and a true consequent. So, the conditional is true.

A	A	⊃	A
1	1	1	1
0	0		0

On the second row, we have a conditional with a false antecedent and a false consequent, which still results in a true conditional.

A	A	⊃	A
1	1	1	1
0	0	1	0

When everything is filled in, the truth-table is complete.

Now, let's do one containing two simple sentences, '(P ∨ Q) & ¬P'.

First, write the complex sentence, then, to the left, write the simple simple sentence letters found in that complex sentence in alphabetical order.

P Q (P ∨ Q) & ¬ P

Before we draw the lines, we need to determine how many rows are needed. Now, there are four partial models, one in which 'P' and 'Q' are both true, one in which they are both false, one where 'P' is true and 'Q' is false, and one in which 'Q' is true and 'P' is false. The general rule is that, if there are n simple sentences, the truth-table will have 2^n rows. So, draw the lines, leaving room for four rows.

P	Q	(P	∨	Q)	&	¬	P

Now, we need to fill in the partial models. We’ll do this systematically so that we will be sure to get every one. The best thing to do is to start with the rightmost simple sentence, in this case, ‘Q’. Underneath that, alternate ‘T’ and ‘F’ until you reach the bottom of the truth table.

P	Q	(P	∨	Q)	&	¬	P
	1						
	0						
	1						
	0						

Then, under the next sentence to the left, write ‘1’ twice, then ‘0’ twice. If there were another column, it would have ‘1’ four times, followed by ‘0’ four times, and so on. The top row will always be all 1’s, and the bottom row will always contain all 0’s. The rightmost column will always alternate 1 and 0, and the leftmost column will have all 1’s in the top half and 0’s in the bottome half. Our table should look lie this:

P	Q	(P	∨	Q)	&	¬	P
1	1						
1	0						
0	1						
0	0						

Now, copy those columns to their respective letters in the complex sentence.

P	Q	(P	∨	Q)	&	¬	P
1	1	1		1			1
1	0	1		0			1
0	1	0		1			0
0	0	0		0			0

Next, we have to determine the truth-values of ‘P ∨ Q’ and ‘¬P’ for each model. Let’s start with ‘¬P’. It’s true whenever ‘P’ is false and false whenever ‘P’ is true.

P	Q	(P	∨	Q)	&	¬	P
1	1	1		1		0	1
1	0	1		0		0	1
0	1	0		1		1	0
0	0	0		0		1	0

Now, ‘P ∨ Q’ is false only when both ‘P’ and ‘Q’ are false. So, the column under the disjunction symbol looks like this:

P	Q	(P	∨	Q)	&	¬	P
1	1	1	1	1		0	1
1	0	1	1	0		0	1
0	1	0	1	1		1	0
0	0	0	0	0		1	0

Now, we’re ready to do the column under the main connective of the sentence. This sentence is a conjunction of two sentences, one is a disjunction and the other is a negation. So, we’re only concerned about the columns under the “∨” and the ‘¬’. Remember, conjunctions are true only when both conjuncts are true. On the first two rows, the second conjunct is false, so

P	Q	(P	∨	Q)	&	¬	P
1	1	1	1	1	0	0	1
1	0	1	1	0	0	0	1
0	1	0	1	1		1	0
0	0	0	0	0		1	0

Both conjuncts are true on the third row:

P	Q	(P	∨	Q)	&	¬	P
1	1	1	1	1	0	0	1
1	0	1	1	0	0	0	1
0	1	0	1	1	1	1	0
0	0	0	0	0		1	0

Finally, the first conjunct is false on the fourth row. So, the completed truth table looks like this:

P	Q	(P	∨	Q)	&	¬	P
1	1	1	1	1	0	0	1
1	0	1	1	0	0	0	1
0	1	0	1	1	1	1	0
0	0	0	0	0	0	1	0

Now, let’s try one with three simple sentences. So, we’ll need eight rows.

F	G	H	¬	[(H	&	F)	≡	G]

Filling in the models gets us this:

F	G	H	¬	[(H	&	F)	≡	G]
1	1	1						
1	1	0						
1	0	1						
1	0	0						
0	1	1						
0	1	0						
0	0	1						
0	0	0						

Copying those columns over produces this:

F	G	H	¬	[(H	&	F)	≡	G]
1	1	1		1		1		1
1	1	0		0		1		1
1	0	1		1		1		0
1	0	0		0		1		0
0	1	1		1		0		1
0	1	0		0		0		1
0	0	1		1		0		0
0	0	0		0		0		0

Now, we start with the conjunction. It's true on any TVA in which both 'H' and 'F' are true.

F	G	H	¬	[(H	&	F)	≡	G]
1	1	1		1	1	1		1
1	1	0		0	0	1		1
1	0	1		1	1	1		0
1	0	0		0	0	1		0
0	1	1		1	0	0		1
0	1	0		0	0	0		1
0	0	1		1	0	0		0
0	0	0		0	0	0		0

Now, biconditionals are true whenever the two connected sentences have the same truth value. So, we just need to compare the truth value of the conjunction that we just completed with the truth-value of 'G'

F	G	H	¬	[(H	&	F)	≡	G]
1	1	1		1	1	1	1	1
1	1	0		0	0	1	0	1
1	0	1		1	1	1	0	0
1	0	0		0	0	1	1	1
0	1	1		1	0	0	0	1
0	1	0		0	0	0	0	1
0	0	1		1	0	0	1	0
0	0	0		0	0	0	1	0

Finally, the negation will flip the truth-value of the biconditional.

F	G	H	¬	[(H	&	F)	≡	G]
1	1	1	0	1	1	1	1	1
1	1	0	1	0	0	1	0	1
1	0	1	1	1	1	1	0	0
1	0	0	0	0	0	1	1	1
0	1	1	1	1	0	0	0	1
0	1	0	1	0	0	0	0	1
0	0	1	0	1	0	0	1	0
0	0	0	0	0	0	0	1	0

1.3 Truth Table Shortcuts

If you understand the characteristic truth-tables for the logical connectives, you can complete the most complex truth-table. There is one major problem with truth-tables, however, they get very big very quickly. With every additional simple sentence letter, the truth-table doubles in size. A truth-table for a sentence with seven simple sentences will have 128 rows! Sometimes, though, we can take some shortcuts. Let’s go back and look at some of our earlier examples.

First, we don't need to always copy the columns for the simple sentences. Sometimes it helps to do so for clarity's sake. In this case, there's really no need.

P	Q	(P	∨	Q)	&	¬P
1	1					
1	0					
0	1					
0	0					

First, we know that the column for '¬P' will just be the opposite of 'P'. It's just as easy to switch the truth-values as copy the column in the first place. So, we'll just write this:

P	Q	(P	∨	Q)	&	¬P
1	1					0
1	0					0
0	1					1
0	0					1

Always keep in mind that what we're really interested in is the column under the main connective, which, in this case, is the conjunction symbol. Conjunctions are true only when both conjuncts are true, and since '¬P' is false on the first two lines, the conjunction will be false there also:

P	Q	(P	∨	Q)	&	¬P
1	1				0	0
1	0				0	0
0	1					1
0	0					1

Now, all that we need to do is find the truth-value of the 'P ∨ Q' on the last two lines. We can easily see that it will be false only on the last line, where both disjuncts are false. So, we now have this:

P	Q	(P	∨	Q)	&	¬P
1	1				0	0
1	0				0	0
0	1		1			1
0	0		0			1

Now, we can complete the column for the main connective on the last two rows. Both conjuncts are true on the third row, but the first is false on the fourth row, resulting in this:

P	Q	(P	∨	Q)	&	¬P
1	1				0	0
1	0				0	0
0	1		1		1	1
0	0		0		0	1

Sometimes, we're only asked to determine the truth-value for a sentence on a single model. That is just asking what the truth-value of the sentence would be on one particular row of a truth-table. Here's an example:

> Determine the truth value of '¬(Q & R) ⊃ (S ∨ ¬T)' on this model:
>
> Q = 0
> R = 1
> S = 0
> T = 0

So, we build a single row truth-table.

Q	R	S	T	¬	(Q	&	R)	⊃	(S	∨	¬T)
0	1	0	0								

Since **Q** is false, so is **Q & R**.

Q	R	S	T	¬	(Q	&	R)	⊃	(S	∨	¬T)
0	1	0	0			0					

The negation, then, will be true.

Q	R	S	T	¬	(Q	&	R)	⊃	(S	∨	¬T)
0	1	0	0	1		0					

¬T is true, so the disjunction is also true.

Q	R	S	T	¬	(Q	&	R)	⊃	(S	∨	¬T)
0	1	0	0	1		0				1	1

So, now we have a conditional with a false antecedent and a true consequent, making the conditional true.

Q	R	S	T	¬	(Q	&	R)	⊃	(S	∨	¬T)
0	1	0	0	1		0		1		1	1

Note that we needed to determine the truth-value of only one of the antecedent or the consequent. Either one would have been enough to guarantee that the conditional was true.

1.4 Logical Truth, Falsehood, and Indeterminacy

Now that we know how to complete truth tables, let's see what can be done with them. Truth-tables provide a systematic means for determining those things that were discussed in chapter **??**—validity, consistency, logical truth and falsehood, and logical equivalency.

To check the logical status of a sentence, simply complete the truth table the same way that we have done so far. After completing it, if there are only 1's under the main connective of the sentence, then the sentence is a logical truth. If there are only 0's, then the statement is a logical falsehood. If there is a mixture of 1's and 0's, then it is logically indeterminate. Let's work a few examples.

P ⊃ (Q ⊃ P)

Since there are two simple sentences, the truth-table will have four rows.

P	Q	P	⊃	(Q	⊃	P)
1	1					
1	0					
0	1					
0	0					

Whenever the antecedent of a conditional is false, the conditional must be true. So, the bottom two rows must both be true.

P	Q	P	⊃	(Q	⊃	P)
1	1					
1	0					
0	1	0	1			
0	0	0	1			

Now, the consequent is itself a conditional, ‘Q ⊃ P’. So, when **P** is true, that conditional is true.

P	Q	P	⊃	(Q	⊃	P)
1	1			1		
1	0			1		
0	1	0	1			
0	0	0	1			

Since a true consequent is enough to make a conditional true, the sentence is also true on the top two rows.

P	Q	P	⊃	(Q	⊃	P)
1	1		1	1		
1	0		1	1		
0	1	0	1			
0	0	0	1			

So, since this sentence is true on every model, it is a logical truth, or tautology.

$\neg(A \supset B) \;\&\; (B \vee \neg A)$

This is also a four-row truth table.

A	B	¬	(A	⊃	B)	&	(B	∨	¬A)
1	1								
1	0								
0	1								
0	0								

The left conjunct contains a simple conditional, so let's start there.

a	b	¬	(a	⊃	b)	&	(b	∨	¬a)
1	1			1					
1	0			0					
0	1			1					
0	0			1					

Now, that conditional is negated to form the first conjunct.

A	B	¬	(A	⊃	B)	&	(B	∨	¬A)
1	1	0		1					
1	0	1		0					
0	1	0		1					
0	0	0		1					

That's enough to make the conjunction false on rows 1, 3, and 4.

A	B	¬	(A	⊃	B)	&	(B	∨	¬A)
1	1	0		1		0			
1	0	1		0					
0	1	0		1		0			
0	0	0		1		0			

So, we really just need to find the truth-value of the second conjunct on row 2. Since, on that row, **B** is false and **A** is true, the disjunction $B \vee \neg A$ is false, which makes the conjunction false on the second row also.

A	B	$\neg$	(A	$\supset$	B)	&	(B	$\vee$	$\neg$A)
1	1	0		1		0			
1	0	1		0		0		0	
0	1	0		1		0			
0	0	0		1		0			

So, the truth-table show that the sentence is false on all models, and is therefore a logical falsehood or contradiction.

One more example:

$$(X \equiv Y) \vee (W \vee Z)$$

This truth table, containing four simple sentences, will have sixteen rows. That makes it slightly painful to complete. So, let's think about how we can make things easier. Note that before we can conclude that a sentence is a tautology or contradition, we have to complete all of the rows. On the other hand, to conclude that a sentence is logically indeterminate or contingent, we only need one row that is 1, and one row that is 0. Thinking shrewdly about which rows should have which values may make our job much easier.

W	X	Y	Z	(X	≡	Y)	∨	(W	∨	Z)
1	1	1	1							
1	1	1	0							
1	1	0	1							
1	1	0	0							
1	0	1	1							
1	0	1	0							
1	0	0	1							
1	0	0	0							
0	1	1	1							
0	1	1	0							
0	1	0	1							
0	1	0	0							
0	0	1	1							
0	0	1	0							
0	0	0	1							
0	0	0	0							

X and **Y** are both true on the first row, so the biconditional is true. That’s enough to guarantee that the disjunction is also true.

W	X	Y	Z	(X	$\equiv$	Y)	$\vee$	(W	$\vee$	Z)
1	1	1	1	1	1	1	1			
1	1	1	0							
1	1	0	1							
1	1	0	0							
1	0	1	1							
1	0	1	0							
1	0	0	1							
1	0	0	0							
0	1	1	1							
0	1	1	0							
0	1	0	1							
0	1	0	0							
0	0	1	1							
0	0	1	0							
0	0	0	1							
0	0	0	0							

Now, we just need a row that is false. Since our sentence is a disjunction, we need a row on which both disjuncts are false. Let's start with the second disjunct. For this to be false, both **W** and **Z** must be false. That is the case on rows, 10, 12, 14, and 16. Now, we also need it to be a row on which the first disjunct is false, which will be row on which **X** and **Y** have different truth values. That's true on rows 12 and 14. So, either row 12 or 14 will provide what we need, a row on which the sentence is false.

W	X	Y	Z	(X	$\equiv$	Y)	$\vee$	(W	$\vee$	Z)
1	1	1	1	1	1	1	1			
1	1	1	0							
1	1	0	1							
1	1	0	0							
1	0	1	1							
1	0	1	0							
1	0	0	1							
1	0	0	0							
0	1	1	1							
0	1	1	0							
0	1	0	1							
0	1	0	0							
0	0	1	1							
0	0	1	0	0	0	1	0	0	0	0
0	0	0	1							
0	0	0	0							

We now have at least one row that is true and at least one row that is false, so the sentence is logically indeterminate.

1.5 Logical Equivalence

Two sentences are logically equivalent if and only if they have the same truth-value on every model. To show logical equivalence, put the two sentences on the same truth-table. If they are logically equivalent, the columns under the main connectives of the sentences will be exactly the same.

Two logical truths will always be logically equivalent, as will two logical falsehoods. Logical indeterminacies may or may not be equivalent.

Let's check these two sentences for logical equivalence (We'll separate them with a forward slash for clarity):

$F \supset G \qquad \neg G \supset \neg F$

F	G	F	$\supset$	G	/	$\neg$G	$\supset$	$\neg$F
1	1		1				1	
1	0		0				0	
0	1		1				1	
0	0		1				1	

See that the two sentences have the same truth-value on every model, and are therefore logically equivalent.

Now, consider these two sentences:

$A \vee B \qquad (B \vee C) \vee A$

A	B	C	A	$\vee$	B	/	(B	$\vee$	C)	$\vee$	A
1	1	1		1				1		1	1
1	1	0		1				1		1	1
1	0	1		1				1		1	1
1	0	0		1				0		1	1
0	1	1		1				1		1	0
0	1	0		1				1		1	0
0	0	1		0				1		1	0
0	0	0		0				0		0	0

Note that it is not until line 7 that the two sentences differ in truth-value. Since there is at least one model on which they have different truth-values, the two sentences are not logically equivalent.

1.6 Logical Consistency

A set of sentences is logically consistent if and only if there is at least one model on which all of the members of the set are true. If there is no model on which all of the members are true, then the set is logically inconsistent.

Any set that contains a logical falsehood is inconsistent. Sets that do not contain logical falsehoods *can* be inconsistent. We can check for logical consistency using one truth-table. If the set is consistent, then there will be at least one row on which all of the sentences are true.

$\{\neg A \,\&\, \neg B, A \lor \neg C\}$

A	B	C	¬A	&	¬B	/	A	∨	¬C
1	1	1	0	0	0		1	1	0
1	1	0	0	0	0		1	1	1
1	0	1	0	0	1		1	1	0
1	0	0	0	0	1		1	1	1
0	1	1	1	0	0		0	0	0
0	1	0	1	0	0		0	1	1
0	0	1	1	1	1		0	0	0
0	0	0	1	1	1		0	1	1

This truth table shows that the set of consistent. Not that it is only on the last row on which both sentences are true. We cannot conclude that a set is inconsistent without examining all of the rows of a truth-table. That doesn't mean that we necessarily have to complete each row completely, though. Let's look at this example again to see the shortcuts we could have taken.

A	B	C	¬A	&	¬B	/	A	∨	¬C
1	1	1							
1	1	0							
1	0	1							
1	0	0							
0	1	1							
0	1	0							
0	0	1							
0	0	0							

Keep in mind that we're trying to find a row on which both sentences are true. The first sentence will be true only when A and B are both false, which is only on the last two rows.

A	B	C	¬A	&	¬B	/	A	∨	¬C
1	1	1							
1	1	0							
1	0	1							
1	0	0							
0	1	1							
0	1	0							
0	0	1		1					
0	0	0		1					

The truth-value of the second sentence won't matter on the first six rows since the first sentence is false on those rows. So, we just need to determine the truth value of the second sentence on the last two rows. It is true whenever A is true or C is false. So, it is false on row 7, but true on row 8.

A	B	C	¬A	&	¬B	/	A	∨	¬C
1	1	1							
1	1	0							
1	0	1							
1	0	0							
0	1	1							
0	1	0							
0	0	1		1				0	
0	0	0		1				1	

So, it's only on that last row that both of the sentences are true, but that is enough to make the set consistent.

Here's another example:

{P ∨ ¬P, P ⊃ Q, P & ¬Q}

We know that P ∨ ¬P is true on all rows.

P	Q	P	∨	¬P	/	P	⊃	Q	/	P	&	¬Q
1	1		1									
1	0		1									
0	1		1									
0	0		1									

Unfortunately, we can't draw any conclusions from that alone. The second sentence is true on all rows except for the second.

P	Q	P	∨	¬P	/	P	⊃	Q	/	P	&	¬Q
1	1		1				1					
1	0		1				0					
0	1		1				1					
0	0		1				1					

The third sentence is true only when P is true and Q is false, which is just the second row.

P	Q	P	∨	¬P	/	P	⊃	Q	/	P	&	¬Q
1	1		1				1				0	
1	0		1				0				1	
0	1		1				1				0	
0	0		1				1				0	

So, there is no row on which all of the sentences are true. The set is then inconsistent.

1.7 Logical Entailment and Validity

Entailment is a relation between a set of sentences of a language and a sentence of that language.

A set Γ logically entails a sentence α if and only if there is no model on which every member of Γ is true and α is false.

The symbol for logical entailment is the double turnstile '⊨'

This expression

$$\Gamma \vDash \alpha$$

means that Γ entails α, and the expression

$$\Gamma \nvDash \alpha$$

means that Γ does not entail α.

When we test for logical entailment using a truth table, we'll single slashes between each member of Γ and a double slash between the members of Γ and the purportedly entailed sentence. Here's a simple example of a classic argument form called *Modus Ponens*:

$$\{P, P \supset Q\} \vDash Q$$

P	Q	P	/	P	⊃	Q	//	Q
1	1	1			1			1
1	0	1			0			1
0	1	0			1			0
0	0	0			1			0

We now check to make sure that there are no rows on which all of the premises are true and the conclusion is false. The only row on which all of the premises are true is the first one, but on that row, the conclusion is also true. So, there is no row on which all of the premises are true and the conclusion is false, so the set containing the premises logically entails the conclusion, which is to say that the argument is valid.

Compare that to this:

P	Q	P	⊃	Q	/	Q	//	P
1	1		1			1		1
1	0		0			0		1
0	1		1			1		0
0	0		1			0		0

The third row demonstrates that $\{P \supset Q, Q\}$ does not logically entail P.

Entailment and validity are closely related concepts. An argument is logically valid if and only if there is no model on which all of the premises are true and the conclusion is false. This means that an argument is logically valid just in case the set containing the premises logically entails the conclusion.

Let's check this argument for validity:

1. $H \equiv I$
2. $\underline{\neg H \lor \neg I}$

$\therefore\ \neg H \,\&\, \neg I$

H	I	H	$\equiv$	I	/	$\neg$H	$\lor$	$\neg$I	//	$\neg$H	&	$\neg$I
1	1		1				0				0	
1	0		0				1				0	
0	1		0				1				0	
0	0		1				1				1	

The fourth row is the only row on which both premises are true, but the conclusion is also true on that row. So, the argument is logically valid.

Now, let's look at this argument:

1. $W \supset (X \lor Y)$
2. $X \supset (\neg Y \lor Z)$
3. $\underline{W\qquad\qquad}$

$\therefore\ Z$

With four simple sentences, this is a long truth-table. So, let's take a shortcut or two. Notice first that the third premise is only true on the last eight rows of the truth-table. That means that there won't be any lines in the upper half of the truth-table where the premises are all true and the conclusion false.

W	X	Y	Z	W	⊃	(X	∨	Y)	/	X	⊃	(¬Y	∨	Z)	/	W	//	Z
1	1	1	1															
1	1	1	0															
1	1	0	1															
1	1	0	0															
1	0	1	1															
1	0	1	0															
1	0	0	1															
1	0	0	0															
0	1	1	1													1		
0	1	1	0													1		
0	1	0	1													1		
0	1	0	0													1		
0	0	1	1													1		
0	0	1	0													1		
0	0	0	1													1		
0	0	0	0													1		

Now, the conclusion is false on only half of those rows.

W	X	Y	Z	W	⊃	(X	∨	Y)	/	X	⊃	(¬Y	∨	Z)	/	W	//	Z
1	1	1	1															
1	1	1	0															
1	1	0	1															
1	1	0	0															
1	0	1	1															
1	0	1	0															
1	0	0	1															
1	0	0	0															
0	1	1	1													1		1
0	1	1	0													1		0
0	1	0	1													1		1
0	1	0	0													1		0
0	0	1	1													1		1
0	0	1	0													1		0
0	0	0	1													1		1
0	0	0	0													1		0

That means that we only need to check the truth-value of the other two premises on rows, 10, 12, 14, and 16. The first premise is true on rows 8–12, since W is false on those rows. So, it really depends on the second premise. Since X is false on the bottom four rows, making the second premise true there.

W	X	Y	Z	W	⊃	(X	∨	Y)	/	X	⊃	(¬Y	∨	Z)	/	W	//	Z
1	1	1	1															
1	1	1	0															
1	1	0	1															
1	1	0	0															
1	0	1	1															
1	0	1	0															
1	0	0	1															
1	0	0	0															
0	1	1	1		1											1		1
0	1	1	0		1											1		0
0	1	0	1		1											1		1
0	1	0	0		1											1		0
0	0	1	1		1						1					1		1
0	0	1	0		1						1					1		0
0	0	0	1		1						1					1		1
0	0	0	0		1						1					1		0

We shown that the premises are true and the conclusion is false on at least rows 14 and 16. That’s one more than needed to prove that the argument is invalid.

1.8 Short Truth Tables

If done correctly, truth-tables are a foolproof way of answering questions about truth-functional properties. They do have a significant drawback, though—they can quickly get very large and cumbersome. Although we have seen how we can take some shortcuts, we still often have to complete large portions of a truth table before we can draw a conclusion. It would nice if we were able to go straight the row, if there is one, that proves that an argument is invalid or a set is consistent, or else easily prove that there is no such row.

In this section, I'll introduce a method called the short truth-table that can do that, and that works very well *most* of the time.

Let's try it out on the following argument:

1. $A \vee B$
2. $B \supset C$
3. $\underline{\neg A}$

$\therefore$ C

We start by listing the premises and conclusion at the top of a truth table, but we only need to leave room for one line:

A	B	C	A	∨	B	/	B	⊃	C	/	¬	A	//	C

Now, we will assume that the argument is invalid. That is, we assume that there is at least one row on which the premises are true and the conclusion is false. So, we'll put a 1 under the main connective of each premises and a 0 under the main connective of the conclusion.

A	B	C	A	∨	B	/	B	⊃	C	/	¬	A	//	C
				1				1			1			0

Now, we fill in what we know. The same sentence letter must have the same truth-value no matter where it occurs in the argument. So, 'C' must be false in the second premise.

A	B	C	A	∨	B	/	B	⊃	C	/	¬	A	//	C
				1				1	0		1			0

We can't really do anything with the first premise yet, since there are three different ways that a disjunction can be true. We do know, however, that 'A' must be false in the third premise, which would make it false in the first premise as well.

A	B	C	A	∨	B	/	B	⊃	C	/	¬	A	//	C
			0	1				1	0		1	0		0

Now, since we have a true disjunction with one false disjunct, we know the other disjunct must be true.

A	B	C	A	$\vee$	B	/	B	$\supset$	C	/	$\neg$	A	//	C
			0	1	1			1	0		1	0		0

Finally, we fill in the truth-value for B in the second premise.

A	B	C	A	$\vee$	B	/	B	$\supset$	C	/	$\neg$	A	//	C
			0	1	1		1	1	0		1	0		0

Now, we have a problem. Assuming that the argument was invalid led to a contradiction. In this case, the contradiction is a true conditional with a true antecedent and a false consequent. So, assumption must have been false, and the argument is therefore valid. There are two kinds of contradictions that can occur. The first is having a simple sentence that is true in one place but false in another. The second is having a complex sentence with a truth-value that is incompatible with the truth-values of the component simple sentences.

Let's back up a few steps to see another way that it could have gone, although it won't change anything about the validity of the argument.

A	B	C	A	$\vee$	B	/	B	$\supset$	C	/	$\neg$	A	//	C
			0	1				1	0		1	0		0

Instead of completing the first premise, we could have gone to the second premise. The only way that a conditional can be true if it has a false consequent is that the antecedent is false. So, we'll put 0 under the 'B'.

A	B	C	A	$\vee$	B	/	B	$\supset$	C	/	$\neg$	A	//	C
			0	1			0	1	0		1	0		0

Now, we'll have to put 0 under the 'B' in the first premise.

A	B	C	A	$\vee$	B	/	B	$\supset$	C	/	$\neg$	A	//	C
			0	1	0		0	1	0		1	0		0

But that gives us a true disjunction with two false disjuncts, another contradiction. There is no way that we can construct a line where this valid argument has all true premises and a false conclusion. Let's try another one:

1. A $\equiv$ (B $\vee$ C)

2. $\underline{\neg B \lor \neg D}$
$\therefore$ $\neg A$

We'll start by making the heading row of the truth table.

A	B	C	D	A	$\equiv$	(B	$\lor$	C)	/	$\neg$	B	$\lor$	$\neg$	D	/	$\neg$	A

Fill in the truth values under the main connectives like this:

A	B	C	D	A	$\equiv$	(B	$\lor$	C)	/	$\neg$	B	$\lor$	$\neg$	D	/	$\neg$	A
					1							1				0	

There are too many ways that biconditional and disjunctions can be true, so the only place we can start is the conclusion.

A	B	C	D	A	$\equiv$	(B	$\lor$	C)	/	$\neg$	B	$\lor$	$\neg$	D	/	$\neg$	A
					1							1				0	1

Now, we can fill in something in the first premise.

A	B	C	D	A	$\equiv$	(B	$\lor$	C)	/	$\neg$	B	$\lor$	$\neg$	D	/	$\neg$	A
				1	1							1				0	1

Since the first half of that biconditional is true, we know the second half must be also.

A	B	C	D	A	$\equiv$	(B	$\lor$	C)	/	$\neg$	B	$\lor$	$\neg$	D	/	$\neg$	A
				1	1		1					1				0	1

Now, the only thing we can do is to take a guess. Let's assume that both 'B' and 'C' are true in the first premise. That will make the bicondional true.

A	B	C	D	A	≡	(B	∨	C)	/	¬	B	∨	¬	D	/	¬	A
				1	1	1	1	1				1				0	1

And if ‘B’ is true, then ‘¬B’ must be false in the second premise.

A	B	C	D	A	≡	(B	∨	C)	/	¬	B	∨	¬	D	/	¬	A
				1	1	1	1	1		0	1	1				0	1

So, now we have a true conditional with one false disjunct, so the other disjunct must be true.

A	B	C	D	A	≡	(B	∨	C)	/	¬	B	∨	¬	D	/	¬	A
				1	1	1	1	1		0	1	1	1			0	1

This means that ‘D’ must be false.

A	B	C	D	A	≡	(B	∨	C)	/	¬	B	∨	¬	D	/	¬	A
				1	1	1	1	1		0	1	1	1	0		0	1

We have now filled in the truth-table, and our assumption that the argument was invalid has led to no contradictions. Not only do we know that the argument is invalid, we can state one of the rows on which it would have true premises and a false conclusion. Our last step is to fill in the model on the left by copying the truth-values underneath the simple sentences.

A	B	C	D	A	≡	(B	∨	C)	/	¬	B	∨	¬	D	/	¬	A
1	1	1	0	1	1	1	1	1		0	1	1	1	0		0	1

Remember that I said that this works very well *most* of the time. We start by assuming the argument is invalid. If that’s the only assumption that we have to make, then we should get our answer fairly easily. In the second example, though, I had to make some further assumptions—I assumed that both

'B' and 'C' were true. If, on those assumptions, we do *not* get any contradictions, then we have successfully proved the argument to be invalid. If we do get a contradiction, however, we have only proved that at least one of our assumptions was false. It need not have been the assumption that the argument was invalid. So, if there had been a contradiction, we would have needed to assume that B was true and C was false to see if we still got a contradiction. If we did, then we would have to assume that C was true and B was false and tried again. The bottom line is that the more assumptions we make, the more difficult it will be to fill everything in.

1.9 Semantics for SL

Finally, let's formally state the semantics of *SL*:

- If α is a propositional constant, α is true in v if and only if $v(\alpha) = 1$.
- $\neg\alpha$ is true in v if and only if α is not true in v.
- α & β is true in v if and only if α is true in v and β is true in v.
- $\alpha \vee \beta$ is true in v if and only if either α is true in v or β) is true in v, or both.
- $\alpha \vee \beta$ is true in v if and only if either α is not true in v or β is true in v.
- $\alpha \equiv \beta$ is true in v if and only if either both α and β are true in v or neither α nor β are true in v.

These clauses determine the truth value for any well-formed formula of *SL* that uses any of the five logical connectives defined.

We can also carefully define some important logical concepts using the notion of a model:

- A formula is a logical truth if and only if it is true on every model.
- A formula is a logical false hood if and only if it is false on every model.
- A formula is logically contingent if and only if it is neither a logical truth nor a logical falsehood.
- Two formulas are equivalent if and only if they have the same truth value on every model.
- A set of formulas is consistent if and only if there is at least one model in which they are all true.
- A formula α entails another formula β if and only if there is no model in which α is true and β is false.

- An argument is valid if and only there is no model in which all of its premises are true and its conclusion is false.