

## Chapter 19

# Sentences That Are Always True (or False)

Two chapters ago, you encountered multiple-valuation truth tables, and one chapter ago, you learned that that's what they're called. You then worked with single-valuation truth tables. In this chapter, we switch our focus back to multiple-valuation truth tables. We do this for one very important reason: Multiple-valuation truth tables are used to identify a very significant group of sentences—sentences that are guaranteed to be true. Most of this chapter will be spent on the mechanics of building multiple-valuation truth tables, but the really important thing to grasp is what we will do with these truth tables—which we'll get to at the end of the chapter.

### Building multiple-valuation truth tables

How do you build a multiple-valuation truth table? First, you build the heading row and the Parsnip tree in exactly the same way as you would for a single-valuation truth table. The question you then have to ask is how many contents rows to leave room for. As it happens, there's a very simple rule for this. Count the number of distinct sentence letters in the truth table (this will be the number of columns in the valuation section). Suppose this number is  $n$ . Then the number of rows in the table will be  $2^n$  (two to the  $n^{\text{th}}$  power).<sup>1</sup> So if you have one sentence letter, you'll need room for only two contents rows; for two sentence letters, four contents rows; for three, eight contents rows; and so on.

Now you have to fill in the contents rows in the valuation section. The objective here is to make sure that every possible combination of truth values—in other words, every valuation—has a row. Here's a method to follow that will work for truth tables of any size:

Start at the rightmost column in the valuation section. Fill it in from top to bottom with one 'T', then one 'F', and then one 'T', and so on, alternating 'T's and 'F's until you've filled in the right number of contents rows.

Now move one column to the left, and fill it in from top to bottom with *two* 'T's followed by two 'F's, and so forth, alternating two 'T's with two 'F's until all your rows have entries in that column.

Now move one more column to the left, and fill this one in with groups of *four* 'T's alternating with groups of four 'F's.

---

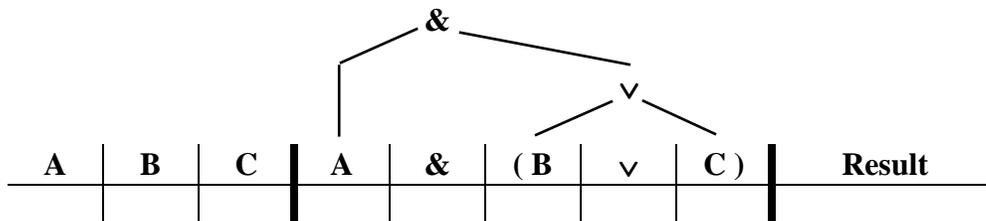
<sup>1</sup> If you've not yet encountered exponents in math class, don't worry— $2^n$  is very easy to compute. All you have to do is multiply 2 together  $n$  times. Thus  $2^1$  is just 2,  $2^2$  is  $2 \times 2$  (i.e., 4),  $2^3$  is  $2 \times 2 \times 2$  (or 8),  $2^4$  is  $2 \times 2 \times 2 \times 2$  (or 16), and so on. So once you've figured out how many distinct sentence letters you have in your truth table, just multiply that many 2's together, and you'll get the number of rows that your truth table will need.

And so it goes. Each time you move left one column, you double the number of ‘T’s and ‘F’s that will alternate in the new column. And you do this until you’ve filled in every column in the valuation section of your truth table.

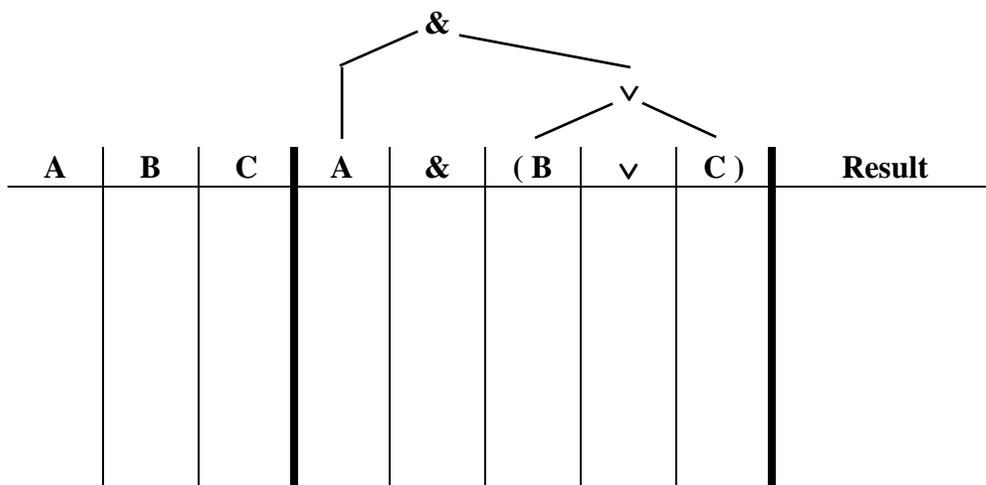
You should follow this method whenever you are building a multiple-valuation truth table, to avoid errors and keep your truth tables consistent with those of your classmates.

Now that the valuation section is filled in, you can fill in the rest of each row in your truth table (that is, the work section and result section). For each row, you follow the same method as was described in Chapter 18 for filling in single-valuation truth tables—though I prefer to carry it out one whole column at a time, rather than a row at a time.

Let’s step through an example. We’ll construct the multiple-valuation truth table for the sentence ‘ $A \& (B \vee C)$ ’. First, we set up the heading row and Parsnip tree, which gives us this:



Of course, we’ll need more than one contents row. The question is, how many? There are three sentence letters, so the number of contents rows required is  $2^3$ , or  $2 \times 2 \times 2$ , which is 8. So our truth table with the right number of contents rows will look like this:



(You can’t see that it has eight rows, but I assure you it does, as you shall see when we fill it in.)

Now comes the part where we fill in the valuation section. We start with the rightmost valuation column and fill it in from top to bottom with alternating ‘T’s and ‘F’s, thus:

A	B	C	A	&	(B	∨	C)	Result
		T						
		F						
		T						
		F						
		T						
		F						
		T						
		F						

Now we move one column to the left, and fill it in with two 'T's followed by two 'F's.

A	B	C	A	&	(B	∨	C)	Result
	T	T						
	T	F						
	F	T						
	F	F						
	T	T						
	T	F						
	F	T						
	F	F						

And we fill in the remaining column, alternating four 'T's with four 'F's.

A	B	C	A	&	(B	∨	C)	Result
T	T	T						
T	T	F						
T	F	T						
T	F	F						
F	T	T						
F	T	F						
F	F	T						
F	F	F						

LOGIC I: TOOLS FOR THINKING

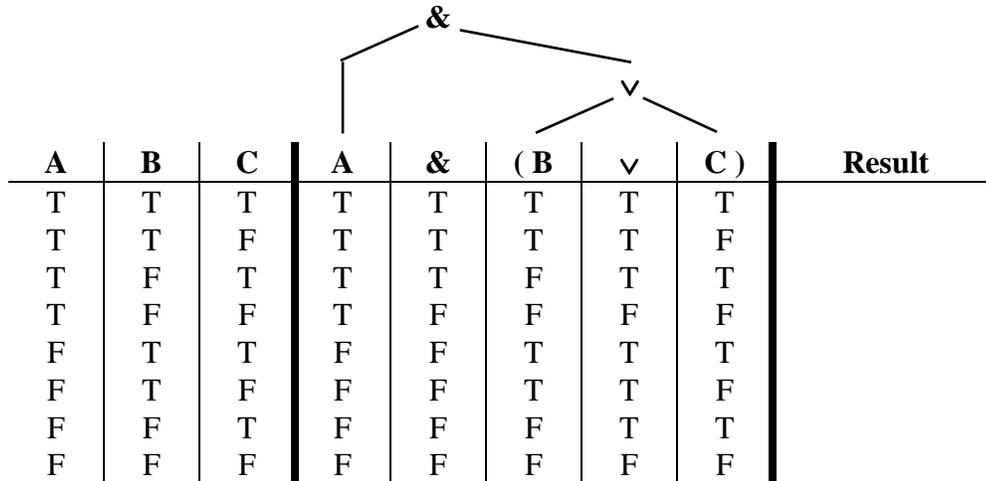
The next step is to copy the valuation section columns over to the columns for the corresponding sentence letters in the work section.

A	B	C	A	&	(B	v	C)	Result
T	T	T	T		T		T	
T	T	F	T		T		F	
T	F	T	T		F		T	
T	F	F	T		F		F	
F	T	T	F		T		T	
F	T	F	F		T		F	
F	F	T	F		F		T	
F	F	F	F		F		F	

Now we can populate the work section. As we work our way up the Parsnip tree, the first operator we come to is the disjunction. For each row, we fill in the disjunction slot with the truth value of the disjunction operator applied to the truth values of its operands in the same row. The result looks like this:

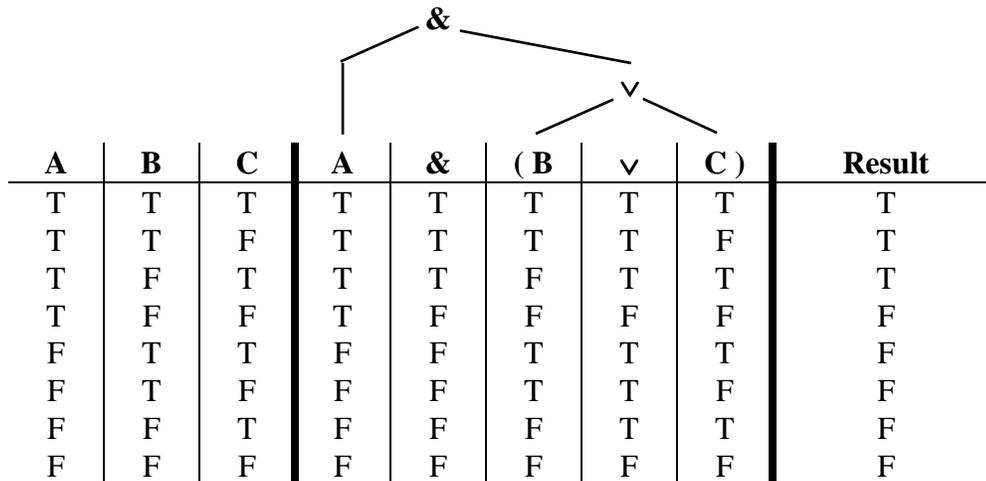
A	B	C	A	&	(B	v	C)	Result
T	T	T	T		T	T	T	
T	T	F	T		T	T	F	
T	F	T	T		F	T	T	
T	F	F	T		F	F	F	
F	T	T	F		T	T	T	
F	T	F	F		T	T	F	
F	F	T	F		F	T	T	
F	F	F	F		F	F	F	

We come next to the conjunction. So we fill in its column, using the truth values of A and the disjunction as inputs (since those are its operands).



A	B	C	A	&	(B	v	C)	Result
T	T	T	T	T	T	T	T	
T	T	F	T	T	T	T	F	
T	F	T	T	T	F	T	T	
T	F	F	T	F	F	F	F	
F	T	T	F	F	T	T	T	
F	T	F	F	F	T	T	F	
F	F	T	F	F	F	T	T	
F	F	F	F	F	F	F	F	

Since that was the last operator, we can now copy its truth values to the results column, and we'll be done.



A	B	C	A	&	(B	v	C)	Result
T	T	T	T	T	T	T	T	T
T	T	F	T	T	T	T	F	T
T	F	T	T	T	F	T	T	T
T	F	F	T	F	F	F	F	F
F	T	T	F	F	T	T	T	F
F	T	F	F	F	T	T	F	F
F	F	T	F	F	F	T	T	F
F	F	F	F	F	F	F	F	F

There we have it: one completed multiple-valuation truth table.

Let's review what the information in the truth table means. Consider the fifth row. From the valuation section, we can see that it's the row for the case in which A is false, B is true, and C is true. In that case, the results column tells us, the sentence 'A & ( B v C )' is false.

Looking at all the rows (i.e., valuations), we see that in three of the eight cases, our sentence is true, and that it is false in the other five cases.

Given any particular valuation of the sentence letters 'A', 'B', and 'C', we can look up the truth value of 'A & ( B v C )' in the result column for that valuation.

### Tautologies

Let's look at a different example, a sentence from an earlier exercise:

$$A \vee (\sim A) \tag{1}$$

You've already done single-valuation truth tables for both possible valuations of this sentence. Let's put them together here into one multiple-valuation truth table:

<b>A</b>	<b>A</b>	<b>v</b>	<b>( ~ A )</b>	<b>Result</b>
T	T	T	F	T
F	F	T	T	T

Notice the results column. It says that the sentence is true under both valuations. That is, no matter what the truth value of A is, sentence (1), 'A v (~ A)', is always true.

If we try some interpretations, we can easily see that this is correct. Let's let 'A' stand for the sentence 'Molly has stripes':

Interpretation:

A: Molly has stripes. (T)

Under that interpretation, (1) translates to (2):

Molly has stripes, or Molly does not have stripes. (2)

In this case, A is true, and so to evaluate (2) we can look at the first contents row of our multiple-valuation truth table for the sentence 'A v (~ A)', and we see that the sentence is true. So if A is true, then (1) is true. Since (2) is just (1) with a true statement assigned to A, it is also true.

What if A is false?

Interpretation:

A: Pluto is a moon of Mars. (F)

Under this interpretation, (1) translates to (3):

Pluto is a moon of Mars, or Pluto is not a moon of Mars. (3)

What's the truth value of that? You can probably figure it out just by looking at it, but to be more official about it you should look at the row in (1)'s multiple-valuation truth table that has a valuation of 'F' for A. There you can see that (3) is true.

It should be apparent to you that we can assign any statement to 'A', and under that interpretation (1) will be true. Even if A's truth value is unknown, we know that (1) is true, because if A is a statement, then it is either true or false (even though we might not know which it is), and either way it gives (1) a truth value of 'T'.

Here's another sentence which is true regardless of the truth values of its sentence letters:

$$\sim (P \& (\sim P)) \tag{4}$$

And here's its truth table:

<b>P</b>	<b>~</b>	<b>(P</b>	<b>&amp;</b>	<b>(~</b>	<b>P))</b>	<b>Result</b>
T	T	T	F	F	T	T
F	T	F	F	T	F	T

Sentences (1) and (4) are called *tautologies*, and since they're Molecular sentences, we can call them *Molecular tautologies*. We can define what a tautology is in three ways.

The first definition is applicable to all tautologies, Molecular or otherwise. It states that a tautology is *any sentence that is true by virtue of its logical structure*. That is, the way it is put together using logical operators guarantees that it will be true regardless of the truth values of the atomic sentences that it contains. (Note that the arrangement of logical operators and parentheses in the statement is what we mean when we refer to its *logical structure*.) By contrast, true sentences which are not tautologies are true by virtue of a combination of their logical structure and the truth values of their atomic sentences. With a tautology, the logical structure alone is sufficient to guarantee the truth of the sentence.

Our second definition is applicable to Molecular tautologies and some others. It defines 'tautology' as *a molecular statement that's true regardless of the truth values of its atomic statements*. This definition relates directly to the examples we've examined, which clearly satisfy its requirement.

To get at the third definition, ask what you would expect to see in the multiple-valuation truth table of a tautology, if a tautology is true no matter what truth values its atomic sentences have? You'd expect to see exactly what we saw in the truth tables of (1) and (4), wouldn't you? Every row (that is, every valuation) would have a result of 'T'. That is to say that the sentence is true regardless of the valuation of its sentence letters.

In fact—and this is our third definition—the Molecular tautologies are precisely *those Molecular statements that have 'T' in every row of the result columns in their multiple-valuation truth tables*. If you start with a Molecular tautology and build its multiple-valuation truth table, you will get a 'T' in every row of the result column. Conversely, if you build the multiple-valuation truth table of any Molecular sentence and find that its result column contains only 'T's, then you know that that sentence is a tautology. So if you want to test a Molecular sentence to see if it's a tautology, all you have to do is build its multiple-valuation truth table and check the result column. If it comes up all 'T's, then you know you've found a tautology.

As I've said, there are tautologies which not Molecular tautologies. You'll learn about some of those later in your logic studies. For now, we're concerned only with Molecular tautologies. And when it comes to Molecular tautologies, our three definitions are equivalent. Make sure you know all of them, since each gives you a different insight into what a tautology is. I'll think you'll find them helpful in a variety of situations when you're thinking about tautologies. (Do keep in mind, however, that two of them aren't applicable to all non-Molecular tautologies.)

Many tautologies have been given names by logicians over the many centuries of logic research. That's because tautologies often embody very important principles of logic. For example, our sentence (1) says something very important about logic, and it is called the **Law of Excluded Middle**. It was first noticed by Aristotle, though in a somewhat different form. It gets its name because it says that a sentence or its negation will be true—there is no middle situation in which the sentence is half true or some such thing (whatever that would mean).

Sentence (4) also has a name. In fact, it has two names! Some logicians call it the **Law of Contradiction**, and others call it the **Law of Non-Contradiction**. I prefer the latter name, though both names are commonly encountered among philosophers and logicians. (It's ironic that its two names sound so contradictory.)

### Antilogies

What would you think if you came across a sentence whose multiple-valuation truth table had 'F's in all the rows of its results column? That would look like the opposite of a tautology, wouldn't it? And that's exactly what it would be.

We will call such sentences **antilogies**. An antilogy is *a sentence that is false by virtue of its logical structure*. Or, to put it a bit differently, a Molecular antilogy is *a molecular sentence which is false no matter what truth values its atomic sentences have*.<sup>2</sup>

You can detect antilogies just as you would expect. You build the multiple-valuation truth table for a Molecular sentence, and if its result column always comes up false, you've found an antilogy.

Any time you find a tautology, you can easily turn it into an antilogy by negating it. Do you see why the negation of a tautology would always be an antilogy? Think about it. Similarly, the negation of an antilogy is always a tautology.

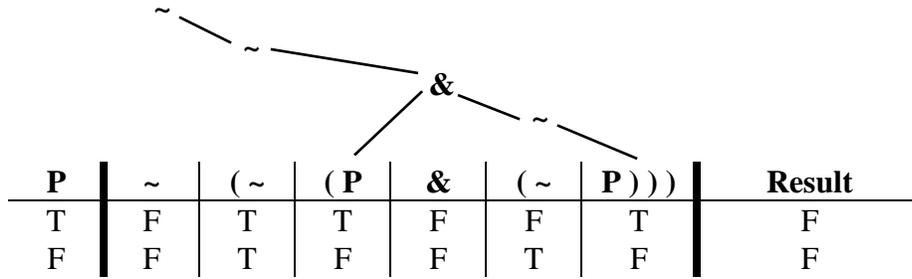
For example, if we negate the Law of Excluded Middle, here's what we get:

A	~	(A	v	(~	A))	Result
T	F	T	T	F	T	F
F	F	F	T	T	F	F

In the same vein, we can produce the truth table for the negation of the Law of Non-Contradiction, and see that it is also an antilogy.

---

<sup>2</sup> The term 'antilogy' is a common one. It is the transliteration of a Greek work meaning 'contradiction'. My use of it here seems reasonable, in light of its dictionary definition ('a contradiction in terms or ideas', according to the *NSOED*), but you shouldn't assume that anyone who hasn't used my book has ever heard of it.



Remember: When all the valuations of a sentence give you a truth value of ‘F’ for the sentence, that sentence is an antilogy. It can’t ever be true, because of its logical structure.

### Terms and concepts discussed in this chapter

- constructing and populating a multiple-valuation truth table
  - computing the number of contents rows
  - populating the valuation section
- tautologies
- how to determine whether a Molecular sentence is a tautology
- Law of Excluded Middle
- Law of Non-Contradiction
- antilogies
- how to tell whether a Molecular sentence is an antilogy

### Exercises

Produce a multiple-valuation truth table for each of the following sentences, and then indicate whether the sentence is a tautology, an antilogy, or neither.

1.  $A \vee (\sim B)$
2.  $P \& (\sim P)$
3.  $(B \& (\sim B)) \rightarrow C$
4.  $C \rightarrow (\sim (D \& E))$
5.  $(P \vee Q) \leftrightarrow ((\sim P) \rightarrow Q)$
6.  $A \rightarrow (P \vee (\sim P))$
7.  $A \rightarrow (P \& (\sim P))$
8.  $((\sim Q) \rightarrow Q) \& (\sim Q)$