

Symbolic Logic 3

Testing deductive validity with truth tables

For an inference to be deductively valid it is *impossible for the conclusion to be false if the premises are true*.

So, given that truth tables tell us about every possible combination of truth values for the component sentences, we can look at them to discover if what the deductive guarantee rules out is ever in fact possible. So we look at the truth tables for all of the premises to see in which situations (rows of the table) they are true, and check what happens to the conclusion in those rows. If the conclusion is also true in all those cases, the argument is deductively valid. If it isn't, then we can say that arguments of that form may well include invalid cases where the premises are all true while the conclusion is false.

Let us look at a couple of simple inferences: If P then Q , P , therefore Q . If P then Q , Q , therefore P . Using our standard translation manual, we can evaluate them both with the following truth table:

P	Q	$P \rightarrow Q$	Q	$P \rightarrow Q$	P
T	T	T	T	T	T
T	F	F	F	F	T
F	T	T	T	T	F
F	F	T	F	T	F

For the first argument, the premises are all true only in row 1, and the conclusion, Q , is also true in that row; so the inference is valid. But the second inference has its premises true in rows 1 and 3 while the conclusion, P , is false in row 3, so its form is an invalid form.

Notice that we have never mentioned rows 2 and 4. What happens in those cases doesn't impinge on the question of validity for these two arguments.

Let us look at another example: Either P or not- Q , Q , so either P or R .

P	Q	R	$P \vee \neg Q$	Q	$P \vee R$
T	T	T	T	T	T
T	T	F	T	T	T
T	F	T	T	F	T
T	F	F	T	F	T
F	T	T	F	T	T
F	T	F	F	T	F
F	F	T	T	F	T
F	F	F	T	F	F

The two premises are both true only in rows 1 and 2; the conclusion is true in those rows so the argument is deductively valid.

The conditional corresponding to an argument

So far we have looked at arguments as they are presented as a sequence of individual statements, with the premises joined to the conclusion by the word ‘therefore’ or ‘so’. To test their validity we have to look at the tables for each premise, check when they are all true together, and then check the value of the conclusion. There is room for careless error in such a procedure. An alternative approach makes use of the fact that there is a very close relationship between an argument and what we call its ‘corresponding conditional’, *i.e.* the conditional sentence constructed by conjoining all the premises, making this conjunction the antecedent, and putting the conclusion as consequent. So you have in effect ‘If the premises then the conclusion’.

Invalidity is a matter of there being a possible situation in which the premises are all true and the conclusion false. When we move to the corresponding conditional such a case would be one where the conditional is false; in all other situations the conditional will be true. A statement that is true in every row of the truth table is a *tautology*. The conditional corresponding to a valid argument will be a tautology. If the conditional fails to be a tautology it has at least one row, there is at least one possible structure, in which its antecedent is true and its consequent false. We can call this possibility a *counterexample* to the inference.

Let us see that for our first two inferences: If P then Q , P , therefore Q . If P then Q , Q , therefore P .

P	Q	$((P \rightarrow Q) \wedge P)$	\rightarrow	Q
T	T	T	T	T
T	F	F	T	F
F	T	T	T	T
F	F	T	T	F
		2	3	1 =

P	Q	$((P \rightarrow Q) \wedge Q)$	\rightarrow	P
T	T	T	T	T
T	F	F	T	T
F	T	T	F	F
F	F	T	T	F
		2	3	1 =

With our valid argument the final column for the corresponding conditional reveals a tautology, whereas we have a contingent¹ conditional corresponding to the invalid form.

¹ Where a compound statement has both T and F in its table, we say it is *contingent*; where it has all Fs it is

One can cut the amount of work involved in constructing the truth table by exploiting simple properties of the truth table for the conditional (and other connectives). The key issue in testing for validity is whether the corresponding conditional is a tautology or not, *i.e.* will we ever get the antecedent true and the consequent false? Since the antecedent is a conjunction, and since a conjunction is true only when every conjunct is true, once we find a conjunct in the antecedent that is false we can stop work on that row. Similarly since a conditional is true when its consequent is true, once we find that the consequent is true we don't need to work out the rest of the row for the antecedent. So here is the second table with unnecessary working omitted (I have put 1-5 under the main column since you start filling it in from the start):

P	Q	$((P \rightarrow Q) \wedge Q)$	\rightarrow	P		
T	T		T	T		
T	F		T	T		
F	T	T	F	F		
F	F		T	F		
		3	4	2	1-5	1

What does the truth table test tell you?

For any argument that we regiment in our mini-language, if its corresponding conditional is a tautology we can say that the argument is deductively valid. NB we are not saying that it has true premises; only that if its premises are true then so necessarily would be its conclusion.

I wasn't so definite when I said what the conditional's not being a tautology told us. One thing it does tell us is that *that structure* is invalid: it is a structure in which it is possible for the premises all to be true and the conclusion false. But we cannot condemn the particular argument we started from just because it embodies an invalid structure. All arguments embody the structure '*P* therefore *Q*', which is obviously invalid. The archetypal valid argument exemplar, *All men are mortal, Socrates is a man, so Socrates is mortal*, translates into our present mini-language simply as *P, Q, so R*.

So the fact that we start with an argument, translate it into our mini-language, and find that it has an invalid structure, is not a death sentence. We have to be sure that our mini-language has captured everything that is relevant to its validity or invalidity. If it has, and if the resulting structure is invalid, then the original argument is invalid. But we have to be watchful in actual arguments in case there is more going on than our mini-language can express.

Related to this point is the idea that truth tables not only embody the standard belief that sentences have one and only one truth value but also assume that the atomic sentences are logically independent of each other.² Take for instance our the invalid structure we have been looking at, in this exemplar: *if 4 is divisible by 2, then 8 is divisible by 2, 8 is divisible by 2, so 4 is divisible by 2*. We said that row 3 shows the structure to be invalid; but row 3 for our

contradictory; all Ts it is a *tautology*.

2 Truth tables were invented by Wittgenstein in the course of a work that assumed, I believe, that the basic elements were thus logically encapsulated.

example would be a world in which 4 is not divisible by 2 while 8 is divisible by 2, and there is no such world.

Farewell to truth tables

Truth tables work for our mini-language, but they quickly get enormous. And we cannot extend them to work with the more complicated mini-language we need to deal with Socrates being mortal, so we shall now leave them aside.

Truth trees

When you have a valid deductive argument, if the premises are true the conclusion will be true too (must be, in fact). The premises are such that their truth rules out the falsehood of the conclusion. So the premises plus the negation of the conclusion will be an inconsistent set.

Conversely, if the premises plus the negation of the conclusion is a consistent set of sentences then you can in fact have all the premises true and the conclusion false, and so you have a formally invalid structure.

The idea behind truth trees (often called *tableaux*) as a means of testing for validity exploits the preceding ideas by using a method of decomposing compound sentences into those parts which must be true if the compound is true, thereby extracting the atomic truths required to make the premises and the negation of the conclusion all true together. If the premises and the negation of the conclusion can indeed be all true together we have found a counterexample to the inference; if it turns out to be impossible for the premises and the negation of the conclusion to be true together in any circumstances then we have shown that the original argument was valid (you couldn't have the premises true and the conclusion false).

So, given a set of sentences, we need a way of decomposing them till they are simply atomic sentences or negations of atomic sentences. We then need a procedure for telling us what that decomposition shows.

In our mini-language, any sentence, however complex, will be of one of only six types:

1. an atomic sentence
2. a negation of a sentence
3. a conjunction of two sentences
4. a disjunction of two sentences
5. a conditional composed of two sentences
6. a biconditional composed of two sentences

Since we can negate any type of sentence, we need to know how to decompose both positive and negative instances of our six types.

1. An atomic sentence is an atomic sentence; it can't be further decomposed so we leave it as it is.
2. A negation of a positive atomic sentence cannot be further decomposed so again we leave it as it is; a negation of a negation of a sentence can be decomposed to yield the sentence (this move is often known as the law of double negation):

$$\neg \neg \phi$$

$$\quad |$$

$$\phi$$

3. A conjunction is true only when both sentences are true, so we decompose a positive conjunction into its conjuncts; the negation of a conjunction is true if either or both of its conjuncts is false, so we decompose it into two branches, each the negation of one conjunct:

$$\phi \wedge \psi$$

$$\quad |$$

$$\phi$$

$$\quad \psi$$

$$\neg(\phi \wedge \psi)$$

$$\quad |$$

$$\text{-----}$$

$$\quad | \qquad \quad |$$

$$\neg \phi \qquad \quad \neg \psi$$

4. A disjunction of two sentences is true when one or both disjuncts are true, so we decompose into two branches, one for each disjunct; the negation of a disjunction is true only when both disjuncts are false, so we replace that with the negations of the two disjuncts:

$$\phi \vee \psi$$

$$\quad |$$

$$\text{-----}$$

$$\quad | \qquad \quad |$$

$$\phi \qquad \quad \psi$$

$$\neg(\phi \vee \psi)$$

$$\quad |$$

$$\neg \phi$$

$$\quad \neg \psi$$

5. A conditional is true when its consequent is true or its antecedent is false, so it is decomposed into two branches for those possibilities; the negation of a conditional is true if the antecedent is true and the consequent is false, so it decomposes into those two:

$$\phi \rightarrow \psi$$

$$\quad |$$

$$\text{-----}$$

$$\quad | \qquad \quad |$$

$$\neg \phi \qquad \quad \psi$$

$$\neg(\phi \rightarrow \psi)$$

$$\quad |$$

$$\phi$$

$$\quad \neg \psi$$

Let us work out the branching rules for the biconditional, given that it is defined as the conjunction of two conditionals: $\phi \leftrightarrow \psi =_{\text{def}} (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$. Starting with

$$(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$$

$$\quad |$$

$$(\phi \rightarrow \psi)$$

$$(\psi \rightarrow \phi)$$

then using the rules for conditionals we get

$$\quad |$$

$$\text{-----}$$

$$\quad | \qquad \quad |$$

$$\neg \phi \qquad \quad \psi$$

$$\text{-----}$$

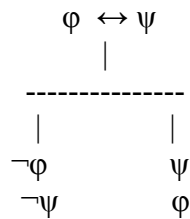
$$\quad | \qquad \quad |$$

$$\text{-----}$$

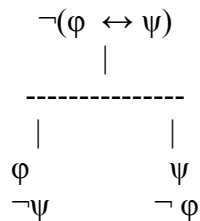
$$\quad | \qquad \quad | \qquad \quad | \qquad \quad |$$

$$\neg \psi \qquad \quad \phi \qquad \quad \neg \psi \qquad \quad \phi$$

Note there are two branches here where we have both a sentence and its negation on the same branch - such branches are closed and for present purposes we can cut them out, so we are left with these branches for the positive biconditional:



The negative biconditional could be worked out in a similar way, or by reference to its truth table. It is true only when one component is true and the other false, so we have two branches:



Notice that when we decomposed $(\psi \rightarrow \varphi)$ we put its products on both branches we had at the time - when you use up a sentence you have to decompose it on every open branch that exists.

Open and closed branches

I have mentioned already that when a branch has a sentence and its negation somewhere along its path, it is closed. Typically we draw a line under it. Such a branch is not worth decomposing further since it is an impossible branch, having a contradiction in its path.

If all the branches of a decompositional tree close then the starting set of sentences is inconsistent. If there is at least one open branch when the decompositional process has terminated, then that branch presents a set of atomic sentences that can all be true together, so the original set is consistent.

Remembering what we said about the status of the set consisting of the premises plus the negation of the conclusion, you can see that our procedure is to decompose that set and see if all branches close. If they do, the argument we derived the set from is valid; if they don't it exhibits an invalid formal structure.

That's all there is to testing for validity in our mini-language:

- translate the argument from English to the truth-functional language;
- form the set of the argument's premises and the negation of its conclusion;
- decompose the set until either a sentence and its negation are on the same branch (in which case, close the branch) or there are only atomic sentences or negations of atomic sentences on a branch;
- if all branches have closed, the original argument is valid;
- if there is at least one open branch the original argument is truth-functionally invalid.

There are some tricks for making life easier. It doesn't matter in which order you deal with the compound sentences in a tree, but you can see that some compounds only require one branch while others require two for their decomposition. Since generally the fewer branches the easier things are to keep track of, decompose compounds that need only one branch before the others. Another thing to look out for is whether dealing with a particular compound will lead immediately to a closed branch. If it will, then that will mean less work in general.

People often tick compounds that have been decomposed, to keep track of what they are doing.