

11. Kane wants **L**ove if and only if he gets it on his own **T**erms.
12. Either Henry keeps his **M**outh shut and goes to **J**ail for a long time or he **R**ats on his friends and lives the rest of his life like a **S**chnook.
13. Only if Herman builds an **A**quarium will Rosemary **L**ove him.
14. **K**illing Morrie is sufficient for keeping him **Q**uiet.
15. Jeffries will be **V**indicated, provided Thorwald **K**illed his wife and Doyle **A**dmits he was right all along.
16. **C**ollaborating with Cecil B. DeMille is necessary to **R**evive Norma's career, and if she does not **C**ollaborate with DeMille, Norma may go **I**nsane.
17. Either **D**aniel or **E**li will get the oil, but not both.
18. To have a **F**ulfilling life as a toy, it is necessary, but not sufficient, to be **P**layed with by children.
19. The Dude will get **R**ich if Walter's **P**lan works, and if the Dude gets **R**ich, he'll buy a new **B**owling ball and a new **C**arpet.
20. Either the AE-35 **U**nit is really malfunctioning or HAL has gone **C**razy; and if HAL has gone **C**razy, then the **M**ission will be a failure and neither **D**ave nor **F**rank will ever get home.

## V. Testing for Validity in SL

Having dealt with the task of taming natural language, we are finally in a position to complete the second and third steps of building a logic: defining logical form and developing a test for validity. The test will involve applying skills that we've already learned: setting up truth tables and computing the truth-values of compounds. First, we must define logical form in SL.

### Logical Form in SL

This will seem trivial, but it is necessary. We're learning how to evaluate arguments expressed in SL. Like any evaluation of deductive arguments, the outcome hinges on the argument's form. So what is the form of an SL argument? Let's consider an example; here is an argument in SL:

$$\begin{array}{l} A \supset B \\ \underline{\sim B} \\ \therefore \sim A \end{array}$$

‘A’ and ‘B’ stand for simple sentences in English; we don’t care which ones. We’re working within SL: given an argument in this language, how do we determine its form? Quite simply, by systematically replacing capital letters with variables (lower-case letters like ‘p’, ‘q’, and ‘r’). The form of that particular SL argument is this:

$$\begin{array}{l} p \supset q \\ \sim q \\ \hline \therefore \sim p \end{array}$$

The replacement of capital letters with lower-case variables was *systematic* in this sense: each occurrence of the same capital letter (e.g., ‘A’) was replaced with the same variable (e.g., ‘p’).

To generate the logical form of an SL argument, what we do is systematically replace SL *sentences* with what we’ll call *sentence-forms*. An SL sentence is just a well-formed combination of SL symbols—capital letters, operators, and parentheses. A sentence-form is a combination of symbols that *would be* well-formed in SL, except that it has lower-case variables instead of capital letters.

Again, this may seem like a trivial change, but it is necessary. Remember, when we’re testing an argument for validity, we’re checking to see whether its form is such that it’s *possible* for its premises to turn out true and its conclusion false. This means checking various ways of filling in the form with particular sentences. Variables—as the name suggests—can *vary* in the way we need: they are generic and can be replaced with any old particular sentence. Actual SL constructions feature capital letters, which are actual sentences having specific truth-values. It is conceptually incoherent to speak of checking different possibilities for actual sentences. So we must switch to sentence-forms.

## The Truth Table Test for Validity

To test an SL argument for validity, we identify its logical form, then create a truth table with columns for each of the variables and sentence-forms in the argument’s form. Filling in columns of Ts and Fs under each of the operators in those sentence-forms will allow us to check for what we’re looking for: an instance of the argument’s form for which the premises turn out true and the conclusion turns out false. Finding such an instance demonstrates the argument’s invalidity, while failing to find one demonstrates its validity.

To see how this works, it will be useful to work through an example. Consider the following argument in English:

If Democrats take back Congress, lots of new laws will be passed.  
Democrats won’t take back Congress.  
 ∴ Lots of new laws won’t be passed.

We’ll evaluate it by first translating it into SL. Let ‘D’ stand for ‘Democrats take back Congress’ and ‘L’ stand for ‘Lots of new laws will be passed’. This is the argument in SL:

$$\begin{array}{l} D \supset L \\ \sim D \\ \hline \therefore \sim L \end{array}$$

First, the logical form. Replacing ‘D’ with ‘p’ and ‘L’ with ‘q’, we get:

$$\begin{array}{l} p \supset q \\ \sim p \\ \hline \therefore \sim q \end{array}$$

Now we set up a truth table, with columns for each of the variables and columns for each of the sentence-forms. To determine how many rows our table needs, we note the number of different variables that occur in the argument-form (call that number ‘n’); the table will need  $2^n$  rows. In this case, we have two variables—‘p’ and ‘q’—and so we need  $2^2 = 4$  rows. (If there were three variables, we would need  $2^3 = 8$  rows; if there were four,  $2^4 = 16$ ; and so on.) Here is the table we need to fill in for this example:

p	q	$p \supset q$	$\sim p$	$\sim q$

First, we fill in the “base columns”. These are the columns for the variables. We do this systematically. Start with the right-most column (under ‘q’ in this case), and fill in Ts and Fs alternately: T, F, T, F, T, F, ... as many times as you need—until you’ve got a truth-value in every row. That gives us this:

p	q	$p \supset q$	$\sim p$	$\sim q$
	T			
	F			
	T			
	F			

Next, we move to the base column to the left of the one we just filled in (under ‘p’ now), and fill in Ts and Fs by alternating in twos: T, T, F, F, T, T, F, F, ... as many times as we need. The result is this:

p	q	$p \supset q$	$\sim p$	$\sim q$
T	T			
T	F			
F	T			
F	F			

If there were a third base column, we would fill in the Ts and Fs by alternating in fours: T, T, T, T, F, F, F, F.... For a fourth base column, we would alternate every other eight. And so on.

Next, we need to fill in columns of Ts and Fs under each of the operators in the statement-forms' columns. To do this, we apply our knowledge of how to compute the truth-values of compounds in terms of the values of their components, consulting the operators' truth table definitions. We know how to compute the values of  $p \supset q$ : it's false when  $p$  is true and  $q$  false; true otherwise. We know how to compute the values of  $\sim p$  and  $\sim q$ : those are just the opposites of the values of  $p$  and  $q$  in each of the rows. Making these computations, we fill the table in thus:

p	q	$p \supset q$	$\sim p$	$\sim q$
T	T	T	F	F
T	F	F	F	T
F	T	T	T	F
F	F	T	T	T

Once the table is filled in, we check to see if we have a valid or invalid form. The mark of an invalid form is that it's possible for the premises to be true and the conclusion false. Here, the rows of the table are the possibilities—the four possible outcomes of plugging in particular SL sentences for the variables: both true; the first is true, but the second false; the first false but the second true; both false. The reason we systematically fill in the base columns as described above is that the method ensures that our rows will collectively exhaust all these possible combinations.

So, to see if it's possible for the premises to come out true and the conclusion to come out false, we check each of the rows, looking for one in which this happens—one in which there's a T under ' $p \supset q$ ', a T under ' $\sim p$ ', and an F under ' $\sim q$ '. And we have one: in row 3, the premises come out true and the conclusion comes out false. This is enough to show that the argument is invalid:

p	q	$p \supset q$	$\sim p$	$\sim q$
T	T	T	F	F
T	F	F	F	T
F	T	<b>T</b>	<b>T</b>	<b>F</b>
F	F	T	T	T

INVALID

When we're checking for validity, we're looking for one thing, and one thing only: a row (or rows) in which the premises come out true and the conclusion comes out false. If we find one, we have shown that the argument is invalid. If we don't find one, that indicates that it's *impossible* for the premises to be true and the conclusion false—and so the argument is valid. Either way, the only thing we look for is a row with true premises and a false conclusion. Every other kind of row is irrelevant. It's common for beginners to mistakenly think they are. The fourth row in the table above, for example, looks significant. Everything comes out true in that row. Doesn't that mean something—something good, like that the argument's valid? No. Remember, each row represents a *possibility*; what row 4 shows is that it's *possible* for the premises to be true and the conclusion true. But that's not enough for validity. For an argument to be valid, the premises must *guarantee* the conclusion; whenever they're true, the conclusion *must* be true. That it's merely *possible* that they all come out true is not enough.

Let's look at a more involved example, to see how the computation of the truth-values of the statement-forms must sometimes proceed in stages. The skill required here is nothing new—it's just identifying main operators and computing the values of the simplest components first—but it takes careful attention to keep everything straight. Consider this SL argument (never mind what its English counterpart is):

$$\begin{array}{l} (\sim A \bullet B) \vee \sim X \\ \underline{B \supset A} \\ \therefore \sim X \end{array}$$

To get its form, we replace 'A' with 'p', 'B' with 'q', and 'X' with 'r':

$$\begin{array}{l} (\sim p \bullet q) \vee \sim r \\ \underline{q \supset p} \\ \therefore \sim r \end{array}$$

So our truth-table will look like this (eight rows because we have three variables;  $2^3 = 8$ ):

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$

Filling in the base columns as prescribed above—alternating every other one for the column under 'r', every two under 'q', and every four under 'p'—we get:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T			
T	T	F			
T	F	T			
T	F	F			
F	T	T			
F	T	F			
F	F	T			
F	F	F			

Now we turn our attention to the three sentence-forms. We'll start with the first premise, the compound ' $(\sim p \bullet q) \vee \sim r$ '. We need to compute the truth-value of this formula. We know how to do this, provided we have the truth-values of the simplest parts; we've solved problems like that already. The only difference in the case of truth tables is that there are multiple different assignments of truth-values to the simplest parts. In this case, there are eight different ways of assigning truth-values to 'p', 'q', and 'r'; those are represented by the eight different rows of the table. So we're solving a problem we know how to solve; we're just doing it eight times.

We start by identifying the main operator of the compound formula. In this case, it's the wedge: we have a disjunction; the left-hand disjunct is ' $(\sim p \bullet q)$ ', and the right-hand disjunct is ' $\sim r$ '. To figure out what happens under the wedge in our table, we must first figure out the values of these components. Both disjuncts are themselves compound: ' $(\sim p \bullet q)$ ' is a conjunction, and ' $\sim r$ ' is a negation. Let's tackle the conjunction first. To figure out what happens under the dot, we need to know the values of ' $\sim p$ ' and 'q'. We know the values of 'q'; that's one of the base columns. We must compute the value of ' $\sim p$ '. That's easy: in each row, the value of ' $\sim p$ ' will just be the opposite of the value of 'p'. We note the values under the tilde, the operator that generates them:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F		
T	T	F	F		
T	F	T	F		
T	F	F	F		
F	T	T	T		
F	T	F	T		
F	F	T	T		
F	F	F	T		

To compute the value of the conjunction, we consider the result, in each row, of the truth-function for dot, where the inputs are the value under the tilde in ' $\sim p$ ' and the value under 'q' in the base column. In rows 1 and 2, it's  $F \bullet T$ ; in rows 3 and 4,  $F \bullet F$ ; and so on. The results:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F		
T	T	F	F F		
T	F	T	F F		
T	F	F	F F		
F	T	T	T T		
F	T	F	T T		
F	F	T	T F		
F	F	F	T F		

The column we just produced, under the dot, gives us the range of truth-values for the left-hand disjunct in the first premise. We need the values of the right-hand disjunct. That's just ' $\sim r$ ', which is easy to compute: it's just the opposite value of ' $r$ ' in every row:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F F		
T	T	F	F F T		
T	F	T	F F F		
T	F	F	F F T		
F	T	T	T T F		
F	T	F	T T T		
F	F	T	T F F		
F	F	F	T F T		

Now we can finally determine the truth-values for the whole disjunction. We compute the value of the wedge's truth-function, where the inputs are the columns under the dot, on the one hand, and the tilde from ' $\sim r$ ' on the other.  $F \vee F$ ,  $F \vee T$ ,  $F \vee F$ , and so on:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F <b>F</b> F		
T	T	F	F F <b>T</b> T		
T	F	T	F F <b>F</b> F		
T	F	F	F F <b>T</b> T		
F	T	T	T T <b>T</b> F		
F	T	F	T T <b>T</b> T		
F	F	T	T F <b>F</b> F		
F	F	F	T F <b>T</b> T		

Since that column represents the range of possible values for the entire sentence-form, we highlight it. When we test for validity, we're looking for rows where the premises as a whole come out true; we'll be looking for the value under their main operators. To make that easier, just so we don't lose track of things visually because of all those columns, we highlight the one under the main operator.

Next, the second premise, which is thankfully much less complex. It is, however, slightly tricky. We need to compute the value of a conditional here. But notice that things are a bit different than usual: the antecedent, 'q', has its base column to the *right* of the column for the consequent, 'p'. That's a bit awkward. We're used to computing conditionals from left-to-right; we'll have to mentally adjust to the fact that ' $q \supset p$ ' goes from right-to-left. (Alternatively, if it helps, you can simply reproduce the base columns underneath the variables in the ' $q \supset p$ ' column.) So in the first two rows, we compute  $T \supset T$ ; but in rows 3 and 4, it's  $F \supset T$ ; in rows 5 and 6, it's  $T \supset F$  (the only circumstance in which conditionals turn out false); and in rows 7 and 8, it's  $F \supset F$ . Here is the result:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F <b>F</b> F	T	
T	T	F	F F <b>T</b> T	T	
T	F	T	F F <b>F</b> F	T	
T	F	F	F F <b>T</b> T	T	
F	T	T	T T <b>T</b> F	F	
F	T	F	T T <b>T</b> T	F	
F	F	T	T F <b>F</b> F	T	
F	F	F	T F <b>T</b> T	T	

No need to highlight that column, as it's the only one we produced for that premise, so there can be no confusion.

We finish the table by computing the values for the conclusion, which is easy:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F <b>F</b> F	T	F
T	T	F	F F <b>T</b> T	T	T
T	F	T	F F <b>F</b> F	T	F
T	F	F	F F <b>T</b> T	T	T
F	T	T	T T <b>T</b> F	F	F
F	T	F	T T <b>T</b> T	F	T
F	F	T	T F <b>F</b> F	T	F
F	F	F	T F <b>T</b> T	T	T



Is the argument valid? We look for a row with true premises and a false conclusion. There are none. The only two rows in which both premises come out true are the second and the eighth, and in both we also have a true conclusion. It is impossible for the premises to be true and the conclusion false, so the argument is valid.

So that is how we test arguments for validity in SL. It's a straightforward procedure; the main source of error is simple carelessness. Go step by step, keep careful track of what you're doing, and it should be easy. It's worth noting that the truth table test is what logicians call a "decision procedure": it's a rule-governed process (an algorithm) that is guaranteed to answer your question (in this case: valid or invalid?) in a finite number of steps. It is possible to program a computer to run the truth table test on arbitrarily long SL arguments. This is comforting, since once one gets more than four variables or so, the process becomes unwieldy.

## EXERCISES

Test the following arguments for validity. For those that are invalid, specify the row(s) that demonstrate the invalidity.

1.  $\sim A, \therefore \sim A \vee B$
2.  $A \supset B, \sim B, \therefore \sim A$
3.  $A \vee B, \sim A, \therefore \sim B$
4.  $\sim (A \equiv B), A \vee B, \therefore A \bullet B$
5.  $\sim (A \supset B), \sim B \vee \sim A, \therefore \sim (\sim A \equiv B)$
6.  $\sim B \vee A, \sim A, A \equiv B, \therefore \sim B$
7.  $A \supset (B \bullet C), \sim B \vee \sim C, \therefore \sim A$
8.  $\sim A \vee C, \sim B \supset \sim C, \therefore A \equiv C$
9.  $\sim A \vee (\sim B \bullet C), \sim (C \vee B) \supset A, \therefore \sim C \supset \sim B$
10.  $A \vee B, B \supset C, \sim C, \therefore \sim A$