## ❖ *Formal Semantics: Truth Trees* ❖

# 3.21. The Semantic Test of Validity: An Indirect Approach

The truth table test of validity provides what we might call a '**direct'** method for testing an argument's validity: constructing truth tables for each premise and the conclusion, we exhaustively scour the possibilities, showing either that true premises are always accompanied by true conclusion or that there is a validity counterexample. And this sweep is guided by the semantic rules for whatever sort of sentences (negations, conjunctions, disjunctions) appear in the argument.

But those same semantic rules can also settle issues of validity in a more roundabout way: not exhaustively examining every possibility, as truth tables do, but adopting a 'guilty until proven innocent' attitude toward arguments, and weighing the results. Following this more **indirect** approach, we ask: what happens if we assume that the argument is *invalid*?



The answer: if the argument really is invalid, assuming it invalid poses no problem; whereas if the argument is valid, the assumption of invalidity blows up in our face.

An example illustrates this.

$$\begin{array}{c} (P \lor Q) \\ \sim P \\ \hline Q \end{array}$$

This familiar argument was an early example of a **valid** logical form. Truth tables bear out its validity.

| P | Q | **(P ∨ Q)** | **~P** | **∴ Q** |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | **1** | **1** | **1** |
| 0 | 0 | 0 | 1 | 0 |

Knowing in advance that this argument is valid, what havoc is wrought by assuming it to be *invalid*?

Assuming an argument invalid is, in effect, assuming there is a **validity counterexample** for that argument: a possible situation where the premises are all true, while the conclusion is false. In broadest outline, such a validity counterexample would look like this.

**Assuming the Argument is Invalid:**

1 (P ∨ Q)
1 ~P
———
0 Q

We now trace out all the consequences of this assumption. First, since the conclusion is a sentence letter, we know already that this situation is one where “Q” is false.

1 (P ∨ Q)
1 ~P
———
⇨ **0** Q

**Q: False**

That information spells consequences for the first premise, “(P ∨ Q),” since “Q” is the right part of that disjunction. We’ve now staked two claims concerning the first premise: that it's a **true disjunction**, with a **false right part**.

$$\begin{array}{ll} & \qquad\;\;\;\mathbf{0} \\ \mathbf{1} & (P \vee Q) \\ 1 & \sim P \\ \hline 0 & Q \end{array}$$

**Q: False**

The semantic rule for disjunctions makes clear there's only *one* way a disjunction could be true with a false right part.

| ● | ▲ | (● ∨ ▲) |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

That's when its **left part is true**.

| ● | ▲ | (● ∨ ▲) |
|---|---|---|
| 1 | 1 | 1 |
| **1** | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

So the left part of that first premise must be true.

$$\begin{array}{ll} & \;\;\mathbf{1} \quad\;\; 0 \\ 1 & (\mathbf{P} \vee Q) \end{array}$$

And now we’ve chased down another consequence of our original assumption: this validity counterexample is a situation where “P” is true.

$$\begin{array}{l} \quad\ \ \mathbf{1}\quad\ 0 \\ 1\ (\mathbf{P} \lor Q) \end{array}$$

**Q: False**
**P: True**

But having “P” true sits ill with the second premise – which, we’ve assumed, is true.

$$\begin{array}{l} \quad\ \ 1\quad\ 0 \\ 1\ (P \lor Q) \\ \mathbf{1}\ \ {\sim}P \\ \hline 0\ \ Q \end{array}$$

**Q: False**
**P: True**

The semantic rule for negations dictates that if “~P” is true,

| ● | ~● |
|---|---|
| 1 | 0 |
| 0 | **1** |

then “P” itself is **false**.

| ● | ~● |
|---|---|
| 1 | 0 |
| **0** | 1 |

We’ve chase down one last piece of information about this validity counterexample: it’s a situation where **“P” is false**.

$$
\begin{array}{l}
\quad\;\; 1 \quad\; 0 \\
1 \; (P \vee Q) \\
\qquad\;\; \mathbf{0} \\
1 \quad \sim P \\
\hline
0 \quad Q
\end{array}
$$

**Q: False**
**P: True**
**P: False**

Earlier we asked: were we to assume the argument invalid (assume that it has a validity counterexample), where would this assumption lead?

Now we have our answer: a situation where the premises are true, but the conclusion false, is a situation where **“P” is both true and false**. That is: assuming the argument invalid leads to a violation of the **Principle of Bivalence** – since Bivalence says there’s no possible way to have “P” both true and false.

So it’s **logically impossible** for the argument to be invalid. The argument must be **valid** instead.

Such is the **indirect** approach to validity: assume the argument invalid; use the semantic rules to trace out the consequences of this assumption; and, finding a violation of Bivalence among these consequences, conclude that the argument must be valid after all.

This indirect approach is admittedly less intuitive than the truth table test of validity, where we mechanically sift through each possibility in search of a validity counterexample. But the indirect approach holds an important advantage over the truth table test: it **reduces our workload** dramatically.

Compare: let’s say that every “1” or “0” in a truth table counts as one step in the truth table test; and likewise count each sentence, across the top of the truth table, as a step. Then the truth table test of validity for our argument takes 4 x 5 = **20 steps**.

| | **1** | **2** | **3** | **4** | |
|---|---|---|---|---|---|
| **1** | **P** | **Q** | **(P ∨ Q)** | **~P** | **∴Q** |
| **2** | 1 | 1 | 1 | 0 | 1 |
| **3** | 1 | 0 | 1 | 0 | 0 |
| **4** | 0 | 1 | 1 | 1 | 1 |
| **5** | 0 | 0 | 0 | 1 | 0 |

(Since “Q” was copied to the end just for ease of reading, it isn’t fair to count it a second time.)

By the same standard – counting each “1,” “0,” and sentence as a step – the indirect approach established the argument’s validity in only **9 steps**.

$$\begin{array}{ll} & \;\;1\;\;\;\;\;\;0 \\ 1 & (P \vee Q) \\ & \;\;\;\;\;0 \\ 1 & \;\;\sim P \\ \hline 0 & \;\;Q \end{array}$$

9 steps instead of 20: a reduction in labor of **more than half**.

The same holds when applying the indirect test to a genuinely invalid argument, such as the following.

$$\begin{array}{c} (P \vee Q) \\ P \\ \hline Q \end{array}$$

Again we begin by assuming the argument invalid – assuming, that is, that there's a validity counterexample.

1 $(P \vee Q)$
1 $P$
———
0 $Q$

In light of the conclusion being false, "Q" must be false in this situation. And since "Q" is the right part of the first premise, the first premise is a true disjunction with a false right part.

**0**
1 $(P \vee Q)$
1 $P$
———
**0** $Q$

**Q: False**

The semantic rule for disjunctions again dictates that a disjunction can be true despite a false right part only if its left part is true.

| ● | ▲ | (● ∨ ▲) |
|---|---|---|
| 1 | 1 | 1 |
| **1** | **0** | **1** |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

So "P," the left part of the first premise, must be true.

**1** 0
1 $(P \vee Q)$
1 $P$
———
0 $Q$

**Q: False**
**P: True**

Turning finally to the second premise, we find that “P” is consistently true throughout this situation. In this example tracing through the consequences of our original assumption of invalidity turned up **no violation of Bivalence**. Here there *really is* a possible way of having premises true and conclusion false: when “P” is true, and “Q” false. The argument really is **invalid**.

That’s just what the truth table test reports: the argument has a validity counterexample in the second valuation, where “P” is true, and “Q” is false.

| **P** | Q | **(P ∨ Q)** | **∴ Q** |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| **1** | 0 | **1** | **0** |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |

Using the truth table test, that result cost us **15** steps.

| | **1** | **2** | **3** | |
|---|---|---|---|---|
| **1** | **P** | **Q** | **(P ∨ Q)** | **∴Q** |
| **2** | 1 | 1 | 1 | 1 |
| **3** | 1 | 0 | 1 | 0 |
| **4** | 0 | 1 | 1 | 1 |
| **5** | 0 | 0 | 0 | 0 |

We achieve the same result *indirectly* in only **8** steps – about half the labor.

$$
\begin{array}{l}
\quad\;\; 1 \quad\; 0 \\
1 \; (P \lor Q) \\
\\
1 \;\; P \\
\hline
0 \;\; Q
\end{array}
$$

This savings in labor is our main motivation for replacing the truth table test with a more indirect approach. But we might instead describe this move as simply retooling and streamlining the same old test, since the indirect approach uses the very same semantic rules the truth tables do – just in a more efficient fashion.

And as long as we're retooling the semantic test of validity, it will be worth our while to revise as well the 'bookkeeping' methods used in truth tables – that is, the notation used for depicting different possible situations. It will turn out that a simple upgrade in this notation cuts our workload in half again. We set out those bookkeeping changes in the next section.

**Summary**

**Indirect Test of Validity**

(i) **Assume the argument is invalid** by picturing a validity counterexample: a situation where all the premises are true, but the conclusion is false.

(ii) Use the **semantic rules** to follow through all the consequences of this assumption.

(iii) If the assumption of invalidity leads to a **violation of Bivalence** – some sentence being both true and false – then the argument is **valid**.

If the assumption of invalidity leads to **no violation of Bivalence** – if each sentence has only one truth value – then the argument really is **invalid**.