The Search-for-Counterexample Test for Validity

The method of truth tables is a perfectly serviceable way of testing whether an SC sentence is valid, but it tends to be quite lengthy, particularly for sentences that contain a lot of atomic sentences. The search-for-counterexample method is a technique for testing validity that, in most cases, proceeds much faster. The idea behind the method is to try to find a N.T.A. under which the given sentence is false. Our method of search will be so thorough and systematic that, if there is a N.T.A. under which the sentence is false, we're sure to find it. If we succeed, this will show that the given sentence is invalid. If we fail, because our method has been thorough and systematic, it can only be because the sentence was valid.

Before giving detailed rules for applying the method, let's illustrate the method with some examples. Consider the sentence "$(((P \rightarrow Q) \wedge (\neg P \rightarrow \neg Q)) \rightarrow (P \vee Q))$." We are trying to find a line of the truth table where the sentence is false. We indicate this aim by putting a "0" under the main connective:

$$\underline{(((P \rightarrow Q) \wedge (\neg P \rightarrow \neg Q)) \rightarrow (P \vee Q))}$$
$$0$$

The only way a conditional can be true will be to have a true antecedent and a false consequent. We indicate this by putting a "1" under the antecedent and a "0" under the consequent:

$$\underline{(((P \rightarrow Q) \wedge (\neg P \rightarrow \neg Q)) \rightarrow (P \vee Q))}$$
$$1 \qquad\qquad 0 \quad 0$$

In order for "$(P \vee Q)$" to be false, "P" and "Q" must both be false; so we put a "0" under each occurrence of "P" or of "Q":

$$\underline{(((P \rightarrow Q) \wedge (\neg P \rightarrow \neg Q)) \rightarrow (P \vee Q))}$$
$$0 \quad 0 \; 1 \; 0 \qquad 0 \qquad 0\,0\;0\,0$$

If "P" and "Q" are both false, "¬P" and "¬Q" are both true; so we put a "1" under each "¬":

$$\underline{(((P \rightarrow Q) \wedge (\neg P \rightarrow \neg Q)) \rightarrow (P \vee Q))}$$
$$0 \quad 0 \; 1 \; 10 \quad 10 \qquad 0\,0\;0\,0$$

If "P" is false, "$(P \rightarrow Q)$" will be true, and if "¬Q" is true, "$(\neg P \rightarrow \neg Q)$" will be true; hence:

$$\underline{(((P \rightarrow Q) \wedge (\neg P \rightarrow \neg Q)) \rightarrow (P \vee Q))}$$
$$0\;1\;0 \; 1 \;\; 10\;1 \; 10 \qquad 0\,0 \;\; 0\,0$$

Voila, we have produced a line of the truth table under which the given sentence is false. So the sentence is invalid.

As a second example, consider the sentence "$(((P \land Q) \to R) \to (P \to (R \lor \neg Q)))$." We be gin by putting a "0" under the main connective:

$$(((P \land Q) \to R) \to (P \to (R \lor \neg Q)))$$
$$0$$

A false conditional has a true antecedent and a false consequent; hence:

$$(((P \land Q) \to R) \to (P \to (R \lor \neg Q)))$$
$$1 \quad\quad 0 \quad 0$$

Again, a false conditional has a true antecedent and a false consequent. So we put "1"s under all occurrences of "P" and "0" under the main connective of "$(R \lor \neg Q)$":

$$(((P \land Q) \to R) \to (P \to (R \lor \neg Q)))$$
$$1 \quad\quad 1 \quad 0\ 1\ 0 \quad 0$$

For a disjunct to be false, both disjuncts have to be false. So we put a "0" under all occurrences of "R" and under the main connective of "$\neg Q$":

$$(((P \land Q) \to R) \to (P \to (R \lor \neg Q)))$$
$$1 \quad\quad 1\ 0\ \ 0\ 1\ 0\ 0\ 0\ 0$$

For "$\neg Q$" to be false, "Q" must be true. So we put "1"s under all the "Q"s:

$$(((P \land Q) \to R) \to (P \to (R \lor \neg Q)))$$
$$1 \quad 1\ \ 1\ 0 \quad 0\ 1\ 0 \quad 0\ 0\ 0\ 1$$

Uh-oh. If "P" and "Q" are both true, "$(P \land Q)$" has to be true, whereas in order for "$((P \land Q) \to R)$" to be true in "R" is false, "$(P \land Q)$" has to be false. So our attempt to find a line of the truth table where the given sentence is false have failed. We indicate our failure by putting an "X" at the problem place, and drawing a line through the whole row of "0"s and "1"s, to indicate that it is not a line of a truth table:

$$(((P \land Q) \to R) \to (P \to (R \lor \neg Q)))$$
$$\overline{1\ X\ 1 \quad 1\ 0\ \ 0\ 1\ 0\ \ 0\ 0\ 1\ 0}$$

As our next example, let's take the sentence "$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$." We put a "0" under the main connective:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$
$$0$$

Now we're stuck. We know that for a biconditional to be false, either the first component has to be true and the second component false or else the first component has to be false and the second component true. But we have no way of telling, at this stage, which, if either of these alternatives will give us a line of the truth table. We could guess, and if our guess gave us a line of the truth table, we would know that the sentence was invalid. But if our guess failed to give us a line of the truth table, we wouldn't know whether that was because the given sentence was valid or because we have made the wrong guess. In order for our failure to find a line of the truth table in which the given sentence is false to give us warrant to conclude that the sentence was valid, our method has to be sure of finding a line of the truth table in which the sentence is false if there is one. For this me must avoid guesswork. Instead, we make two rows under the sentence, both with "0" under "↔." In the top row, we investigate the consequences of declaring the first component true, while in the second row we investigation of calling the first component false.

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$
$$ 1 \quad 0$$
$$ 0 \quad 0$$

We work on the first row first. Since the first component is true, the second must be false:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$
$$ 1 \quad 0 \qquad 0$$
$$ 0 \quad 0$$

Since a false disjunction has both disjuncts false, we continue:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$
$$ 1 \quad 0 \quad 0 \quad 0 \quad 0$$
$$ 0 \quad 0$$

Since a false conditional has to have a true antecedent and a false consequent, we put "1"s under all the "P"s and "Q"s, and we put "0"s under all the "R"s:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$
$$ 1 \quad 1 \quad 1 \, 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0$$
$$ 0 \quad 0$$

Now we have a conflict. We've marked "((P ∨ Q) → R)" true and its consequent false, which means that we ought to mark "(P ∨ Q)" false. But we've also marked "P" and "Q" both true, which means we need to mark "(P ∨ Q)" true. What does this show us? It doesn't yet show us whether the given sentence is valid. It shows that we can't show the sentence invalid by giving a N.T.A. under which the first component is true and the second component false, but we still haven't examined the possibility of a N.T.A. under which the first component is false and the

second true. That is the possibility we now want to investigate. So we strike out the first line, and turn our attention to the second line:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$

~~1   1   1 0   0   1   0   0   1   0~~
               0   0

Since the first component is false, the second component must be true:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$

~~1   1   1 0   0   1   0   0   1   0~~
               0   0   1

Also, since the first component is a false conditional, it must have a true antecedent and a false consequent. So we put "1" under "(P ∨ Q)" and "0" beneath every occurrence of "R":

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$

~~1   1   1 0   0   1   0   0   1   0~~
        1   0 0 0   0 1   0

Now we're stuck again. There is no nonguessing way to decide what to fill in next. So we start a new line, copying over what we've done so far:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$

~~1   1   1 0   0   1   0   0   1   0~~
        1   0 0 0   0 1   0
        1   0 0 0   0 1   0

In the second line, we'll investigate the consequences of declaring "P" true. If that doesn't pan out, in the new line we'll investigate what happens if we declare "P" false:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$

~~1   1   1 0   0   1   0   0   1   0~~
     1 1   0 0 0   0 1   0
     0 1   0 0 0   0 1   0

Making "P" true means that "(P → R)" is going to be false, which, in turn, means that "(Q → R)" has to be true:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$

~~1   1   1 0   0   1   0   0   1   0~~
     1 1   0 0 0 1 0 0 1   1 0
     0 1   0 0 0   0 1   0

Making "(Q → R)" true after having declared "R" false means that "Q" has to be false:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \lor (Q \to R)))$$

~~1 1 1 0 0 1 0 0 1 0~~
1 1 0 0 0 0 1 0 0 1 0 1 0
0 1   0 0 0   0 1   0

We can easily check that we now have a line of the truth table in which the given sentence is false. So the sentence is invalid. There is no need to examine the third line.

Now let's try testing "(((P ∨ Q) → R) ↔ ((P → R) ∧ (Q → R)))":

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$
0

We immediately have to start a second line. In the first line, we'll try making the first component true, while in the second we'll try making it false:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$
          1   0   0
          1   0

Now we're stuck again, so we start a third line, copying over the first line. In first line, we'll try making "R" true, while in the new line we'll make "R" true. (The reason for working with "R" rather than "P" or "Q" is that "R" appears oftener in the sentence, so we get to fill out more of the line:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$
          1 1 1   0
          0   0
          1 0 0

Making "R" true will make "(P → R)" and "(Q → R)" both true, which contradicts the assignment of 0 to their conjunction. So we can strike our the first line and pay our attention to the second:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$
          ~~1 1 0   1 1 0   X 1~~
          0   0
          1 0 0   0

Having the first component of the biconditional false will make the second component true, which will, in turn, make both conjuncts of the second component true:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$

~~11 0   11 0   X 1~~
0   0   1   1   1
10 0   0

Also, since the first component is a biconditional, having it false will make the antecedent true and the consequent false:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$

~~11 0   11 0   X 1~~
1   00 0   10 1   10
10 0   0

Since "(P → R)" and "(Q → R)" are both true conditionals with a false consequents, they must have false antecedents. But making "P" and "Q" both false conflicts with trying to make "(P ∨ Q)" true. So we may strike out the second line, and go to work on the third:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$

~~11 0   11 0   X 1~~
~~0 1 X 0 0   0 0 1 0 1 0 1 0~~
10 0   0

We first mark all the "R"s false:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$

~~11 0   11 0   X 1~~
~~0 1 X 0 0   0 0 1 0 1 0 1 0~~
10 0   0 0   0

Having "((P ∨ Q) → R)" be true while "R" is false makes "(P ∨ Q)" false, which, in turn, makes "P" and "Q" both false:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$

~~11 0   11 0   X 1~~
~~0 1 X 0 0   0 0 1 0 1 0 1 0~~
0 00 1 0   0 0   0 0 0   0

Having "P" and "Q" false makes "(P → R)" and "(Q → R)" both true. But this contradicts the supposition that their conjunction is false:

$$(((P \lor Q) \to R) \leftrightarrow ((P \to R) \land (Q \to R)))$$

~~11 0   11 0   X 1~~
~~0 1 X 0 0   0 0 1 0 1 0 1 0~~

~~0　0 0 1 0　0　0 1 0　0　0 X 0~~

We've run out of options, so we conclude that the given sentence is valid.

Next, consider the sentence "$((P \leftrightarrow Q) \vee ((P \leftrightarrow R) \vee (Q \leftrightarrow R)))$." It's easy to see that the sentence is valid by employing the pigeonhole principle, which says that, if you have *n* holes and more than *n* pigeons, then two or more pigeons have to occupy the same hole. In this case, we have three atomic sentences and only two truth values, so two or more of the atomic sentences have to have the same truth value. Let's see how we can show the sentence valid by employing our algorithm:

$((P \leftrightarrow Q) \vee ((P \leftrightarrow R) \vee (Q \leftrightarrow R)))$
　　　0　　0　　　0　0　　0

Now we're stuck, so we begin a new line. In the top line, we set "P" true, and we set it false in the second:

$((P \leftrightarrow Q) \vee ((P \leftrightarrow R) \vee (Q \leftrightarrow R)))$
　1 0　　0　1 0　0　　0
　0 0　　0　　0　0　　0

Having "P" true and "$(P \leftrightarrow Q)$" false means that "Q" should be false:

$((P \leftrightarrow Q) \vee ((P \leftrightarrow R) \vee (Q \leftrightarrow R)))$
　1 0 0 0　1 0　　0　0 0
　0 0　　0　　0　0　　0

Now we have conflicting instructions about what to do with "R." The truth of "P" and falsity of "$(P \leftrightarrow R)$" tells us that "R" should be true, but the falsity of "Q" and falsity of "$(Q \leftrightarrow R)$" tells us that "R" should be false. So we cross out the first line, and move on the second:

$((P \leftrightarrow Q) \vee ((P \leftrightarrow R) \vee (Q \leftrightarrow R)))$
　~~1 0 0　0　1 0 0　0　0 0 0 X~~
　0 0　　0　　0　0　　0

Because "P" and "$(P \leftrightarrow Q)$" are both false, "Q" should be true:

$((P \leftrightarrow Q) \vee ((P \leftrightarrow R) \vee (Q \leftrightarrow R)))$
　~~1 0 0　0　1 0 0　0　0 0 0 X~~
　0 0 1　0　0 0　0 1 0

But this gives us conflicting advice about "R":

$$((P \leftrightarrow Q) \vee ((P \leftrightarrow R) \vee (Q \leftrightarrow R)))$$
$$\overline{1\ \ 0\,0\ \ 0\ \ 1\ \ 0\,0\ \ 0\ \ 0\ 0\,X}$$
$$\overline{0\ 0\,1\ \ 0\ \ 0\ 0\,1\ 0\ \ 1\ 0\,X}$$

As expected, the sentence is valid.

So far, our examples have been reasonably friendly. The next one turns nasty:

$$(((P \leftrightarrow Q) \leftrightarrow R) \leftrightarrow (P \leftrightarrow (Q \leftrightarrow R)))$$
$$\overline{1\,X\,1\ \ 1\,0\ \ 0\ \ 1\,0\ \ \ 1\,0\,0}$$
$$\overline{1\,X\,1\ \ 0\,1\ \ \ 0\ 1\,1\ \ 1\ 1\,1}$$
$$\overline{0\,X\,1\ \ 1\,1\ \ 0\ \ 0\,0\ \ \ 1\ 1\,1}$$
$$\overline{1\,X\,0\ \ 1\,1\ \ 0\ \ 1\,0\ \ 0\ 0\,1}$$
$$\overline{0\,X\,1\ 0\,0\ \ \ 0\ 0\,1\ 1\ 0\,0}$$
$$\overline{1\,X\,0\ \ 0\,0\ \ \ 0\ 1\,1\ 0\ 1\,0}$$
$$\overline{0\,X\,0\ 1\,0\ \ \ 0\ \ 0\,0\ 0\ 1\,0}$$
$$\overline{0\,X\,0\ 0\,1\ \ \ 0\ 0\,1\ 0\ 0\,1}$$

We have showed the sentence valid, but we had to write out the whole truth table in order to do it. We would have been better off just doing the truth table in the first place.

There are a lot of known methods for testing for validity in the predicate calculus, but they all have this defect: Although they may decide a lot of cases efficiently, in the worst cases they aren't significantly quicker than the method of truth tables. Indeed, the leading open problem in theoretical computer science is this: Either to find a method for testing SC validity that is uniformly significantly more efficient than the method of truth tables; or else to show that no such method is possible.

The search-for-counterexample technique also lets us test for consistency of a sentence: This time, put a "1" under the main connective, and try to fill out the line. Similarly, we have a test whether an argument is valid: Put a "1" under the main connective of each premiss, and a "0" under the main connective of the conclusion. If the argument is valid, you won't be able to fill out the line.

START

Write the given sentence.

Write "0" under the main connective.

Fill out the topmost unstricken line as fully as
you can, using the rules on the next page.

Do
conflicts
arise in trying to
YES   fill out the line? Do you   NO
need to write a "0" at a place where
you've already put a "1" or a
"1" where you now
have a
"0"?

Strike out the line you're working on   YES   Is the   NO
line completely
filled?

Is   The given sentence is invalid   Copy over the line you are working
there   on as a new line. Put a "1" in some
another   blank place in the line you're
YES   unstricken   NO   STOP   currently working on and a "0"
line?   the same place on the new line.

The given sentence is valid

STOP

Rules for filling out a line

Atomic sentences.
If a "1" appears under one occurrence of an atomic sentence, put it under the others.
If a "0" appears under one occurrence of an atomic sentence, put it under the others.

Conjunctions.
If you've written "1" under "∧," put "1" under both conjuncts.
If you've written "1" under both conjuncts, put "1" under "∧."
If you've written "0" under "∧" and you've written "1" under one conjunct, put "0" under the other.
If you've written "0" under either conjunct, write "0" under "∧."

Disjunctions.
If you've written "1" under "∨" and you've written "0" under one disjunct, put "1" under the other.
If you've written "1" under either disjunct, put "1" under "∨."
If you've written "0" under "∨," put "0" under both disjuncts.
If you've written "0" under both disjuncts, put "0" under "∨."

Conditionals.
If you've written "1" under "→" and you've put "1" under the antecedent, put "1" under the consequent.
If you've written "1" under "→ and you've put "0" under the consequent, put "0" under the antecedent.
If you've written "1" under the consequent, put "1" under "→."
If you've written "0" under "→," put "1" under the antecedent and "0" under the consequent.
If you've written "0" under the antecedent, put "1" under "→.
If you've written "1" under the antecedent and "0" under the consequent, put "0" under "→."

Biconditionals.
If you've written "1" under "↔" and you've written "1" under one component, put "1" under the other.
If you've written "1" under "↔" and you've written "0" under one component, put "0" under the other.
If you've written "1" under both components, put "1" under "↔."
If you've written "0" under both components, put "1" under "↔."
If you've written "0" under "↔" and you've put "1" under one component, put "0" under the other.
If you've written "0" under "↔" and you've written "0" under one component, put "1" under the other.
If you've written "1" under one component and "0" under the other, put "0" under "↔."

Negations.
If you've written "1" under "¬," put "0" under the negatum.
If you've written "1" under the negatum, put "0" under "¬."
If you've written "0" under "¬," put "1" under the negatum.
If you've written "0" under the negatum, put "1" under "¬."