reason and argument

John Kulvicki

Schematization

Schematization can help one determine whether an argument is valid, and it can also help us keep track of the many parts of an argument and how they relate to one another. As a first pass, schematization (sometimes called 'formalization') is a way of understanding the forms (hence "formal" logic) of the claims we make, irrespective of their meanings. Sound obvious? Of course not. We'll learn this as we work through it.

In sentential logic, we use individual letters to stand for statements, or the kinds of things that can be true or false. Statements in English are interesting because sometimes they can be combined with other statements to form a complex whole.

> It rained today and class met today.

That's one statement. It's the kind of thing that can be true or false. It's not a question or an exclamation, for example. But that statement has parts that can be true or false on their own: specifically "it rained today" and "class met today". These two statements are connected to one another by "and". Sounds basic, but it's a very important thing to notice. Here is another statement:

> It rained today or class met today.

Same parts as the first statement, but it's a different statement overall, because these individual statements are connected with an "or". Similarly,

> If it rained today, then class met today.

This connective, in English, uses two words, but it is just a way of connecting two claims. That one is a conditional. We saw in the previous set of notes that conditionals are strange, and we will see just how strange they can be as we move on.

These connectives can be confusing, but they are united by the fact they express relations between statements, and they are different in expressing different relations between them.

Sentential logic uses letters to stand for statements, and it also has a vocabulary for representing the relations expressed by these connectives. For example, if we were to make p stand for “it rained today” and q stand for “class met today”, we could write

p & q

to stand for “It rained today and class met today”. And we could use other symbols for the other connectives:

“p v q” stands for “It rained today or class met today”,

and

“p → q” stands for “If it rained today then class met today”.

We also want to say that something is *not* the case. So to say it did *not* rain today, we could write:

~ p.

So far, this might seem like a useless exercise in simplifying our language, but that is not quite what we are doing. First, notice that these schematizations of English sentences work very well, no matter what sentences you have in mind. You could also schematize “I read a book last week and the Giants will win the Super Bowl” using “p & q”. What this schematization captures is that when you use conjunction you are relating two statements, each of which could be true or false, in a certain way. In a sense, it doesn’t matter which statements you relate: if you have two things that can be true or false, you can combine them into a new statement using “and”, or, schematically, “&”.

How does the truth, or falsity, of the compound statement depend on the truth or falsity of the ones that make it up? One of the reasons we can productively schematize statements in this manner is that the truth of many conjunctions, and the truth of many of the other compound statements mentioned above, *relies exclusively on the truth or falsehood of their components.* Consider this table:

p	q	-p	-q	p & q	p v q	p → q
T	T	F	F	T	T	T
T	F	F	T	F	T	F
F	T	T	F	F	T	T
F	F	T	T	F	F	T

The two leftmost columns list all the combinations of truth values for p and q. Given those, the other columns show the truth values for the negations, conjunctions, disjunctions and conditionals involving p and q. The truth of the conditional, for example, depends just upon whether p is true or false and whether q is true or false. It does not depend on which statement p stands for, or which statement q stands for. The same holds true for all of these connectives and negation.

The results in this table are largely what you would expect. If p is true, then you expect not-p to be false, and vice versa. The only way for p & q to be true is for both p and q to be true. The only way for p v q to be false is for both p and q to be false.

Four important points:

First, if the waiter asks whether you want the soup or the salad it is inappropriate to answer "Yes!", just as it is somewhat inappropriate to answer "Both!". In such cases, "or" is being used in a way that suggests you can have one, or the other, but not both. In other cases, we use "or" in a non-exclusive sense. For example: "I don't remember the car's color: it was blue or grey." If, later on, we find out that the car was blue and grey, we would think you were right on. You didn't say something false just because the car was not only blue or only grey. For the sake of convenience, when we use the connective "v" to stand for "or", we mean the non-exclusive sense of "or", sometimes called the "logician's 'or'".

The logician's "or" comes out true when either just one or both of its constituents are true.

Second, the conditional is true unless its antecedent is true and its consequent false. Because there are many kinds of conditionals at work in English or any other language, it's important to keep our eyes on the one intended here. The conditional schematized using the arrow, $\rightarrow$, is truth-functional. That is, its truth depends only on the truth or falsity of its parts. If both are true, the conditional is true, and if the antecedent is false, the conditional is true as well. Consider:

> If the moon is made of cheese,
> then cucumbers taste better pickled.

Everyone knows cucumbers taste better pickled. Surprisingly, this conditional is true, if we read it truth-functionally. It would certainly be odd to say something like the above conditional in conversation because there could be little point in doing so. You can't use such a conditional to help establish the truth of the consequent, because the antecedent is false. The antecedent thus cannot appear as a true premise in an argument for the consequent. So, these are odd beasts, these conditionals that are true if their antecedents are false, but they are not so odd as to be counterintuitive. After all, did you have any intuitions at all about conditionals with false antecedents and true consequents? We sometimes use conditionals with false antecedents and consequents, often to make points in a rather wise-assed fashion. The conditional

> If that's a Rolex, then I am the Pope.

is a fun way of saying: That's not a Rolex. The reasoning goes: since it's clear that I am not the Pope, and intend to assert something true (the conditional is taken to be true), it must also be that the antecedent is false. The only way a conditional with an obviously false consequent can be true is if its antecedent is also false. You know this. You got the point of this objection to the argument immediately. You just knew this implicitly, and this class is trying to make these things explicit for you.

The third strange thing is quite important. There is often some distance between what someone says and what she intends to express by saying it. This is true for many uses of connectives in English, and any other natural language. Sometimes, 'and' is used to express something much closer to 'if…then'. Here, for example, are two perfectly good translations of a cheery quote from Soren Kierkegaard's *Either/Or*:

> If you marry, then you will regret it. If you do not marry, then you will regret it. (Eremita, trans., Penguin 2004)
> Marry, and you will regret it. Do not marry, and you will regret it. (Hong and Hong, trans., Princeton UP 1987)

In the Hongs' translation, the word 'and' carries the same logical force as the expression 'if…then' in Eremita's translation. Similar remarks hold for many connectives. We have to interpret what someone is saying in order to find the logical structure of their remarks. There is no obvious mechanical process for doing so.

In a related vein, sometimes the use of a connecting word like 'and' or 'or' does not express a truth-functional connection at all. Your friend tells you: "I got up and went to breakfast." The truth of this claim does *not* just depend on the truth of its parts, because in this case she suggests that she first got up, and then went to breakfast. If she had breakfast after an all-nighter and then went to bed, you would justifiably fault her for claiming that she got up and went to breakfast, even though she really did both things. In this case, she expresses something more like: I first got up and, later, went to breakfast. The 'and' in the last claim is now truth-functional, even though the one that she used initially was not.

Fourth, there are many words in English, and any other language, that can be used to express the same logical connection between statements. Compare:

> The concert will be outside unless it rains.
> Either it rains or the concert will be outside.
> If it doesn't rain, the concert will be outside.

The connectors 'or' and 'unless' typically express the same logical connection between statements. The 'if…then' locution expresses a similar one, but notice in that case that one of the original claims, the one appearing as the antecedent of the conditional, has been negated. Can you think of others that work this way?

Logical equivalence

Two expressions are *logically equivalent* when they are true and false under exactly the same circumstances. For example, let's schematize "If that's a Rolex, then it's stolen" as: r→s. This schema is true just in s is true or r is false. Now consider the contrapositive formulation "If that's not stolen, then that's not a Rolex": -s → -r. And, "Either that's stolen or it's not a Rolex": -r v s. (Note: this is not the same claim as -(r v s)!)

Now let's make a truth table for each of these expressions:

r	s	r → s	-s → -r	-r v s
T	T	T	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Each of those three expressions comes out true or false under exactly the same circumstances. So, our intuitions that each of those ways of saying things amounts to saying the same thing finds its roots here. Those three ways of saying something are logically equivalent.

For a more complex example, let's consider a conditional related to our argument about the Rolex.

> If you get a 90% discount on something desirable, then you should buy it, unless there are better deals around.

There are four statements make up this one complex one:

> p: You get a 90% discount on something.
> q: That thing is desirable.
> r: You should buy it.
> s: There are better deals around.

There are a few possibilities for schematizing this conditional:

(p & q)→(r v s)

(p & q & -s) → r

[(p & q) → r] v s

Let's make a truth table including truth values for all of the components, p, q, r, and s, and add columns for the three different formulations and see how they compare:

p	q	r	s	(p&q)→(rvs)	(p&q&-s)→r	[(p&q)→r]vs
T	T	T	T	T	T	T
T	F	T	T	T	T	T
T	T	F	T	T	T	T
T	T	T	F	T	T	T
F	T	T	T	T	T	T
T	T	F	F	F	F	F
T	F	T	F	T	T	T
T	F	F	T	T	T	T
F	T	F	T	T	T	T
F	F	T	T	T	T	T
F	T	T	F	T	T	T
T	F	F	F	T	T	T
F	F	F	T	T	T	T
F	F	T	F	T	T	T
F	T	F	F	T	T	T
F	F	F	F	T	T	T

I strongly recommend that you try to work out a table like this, or at a minimum check my results. What they show is that all three candidates are, despite surface appearances, logically identical. These three messy schemata are logically equivalent: true and false under exactly the same circumstances. This, by the way, is just the conditional that ends the argument we have been working with. In order to see how this conditional figures in the argument as a whole, we need to schematize the argument as a whole. Since p, q, and not-s are premises in the original argument, the middle column makes it clear that r, our desired conclusion, follows from them. The other schemata are harder to read, perhaps, but they are no different from the one in the middle, logically speaking.

How do you check whether you are putting in the right truth value for a complex expression like "[(p & q) → r] v s"? Some cases are easy to check. For example, any case in which s is true should render the whole thing true because it has the form "something or s" and no matter how complicated that "something" is, you know the whole is true as long as s is true. Similarly, you know the whole thing is true if (p & q) is false, because that renders the other half of the conditional true (any conditional of this sort with a false antecedent is true. That just leaves a few cases in which you have to be careful, namely those cases in which s is false and (p & q) is true.

A more tedious way to do this is to construct a truth table, as above, for each of the parts of the complex expression. Such a truth table would be just like the one above, but it would have columns for (p & q), (p & q) → r, r v s, and (p & q & -s). That's one huge, ugly truth table, but if you're careful about it, that technique will not lead you astray. Here are some equivalences worth noting. You can work them out using truth tables like the one above.

> p & q = -(-p v -q) ; p v q = -(-p & -q) (DeMorgan's laws)
> p → q = -p v q = -q → -p = -(p & -q)
> p → (r v q) = (p & -r) → q

Formal validity & ways of schematizing

The big monster truth table above is meant to show that the three candidate formulations of a claim are logically equivalent. Let's consider an argument in which such a premise figures.

Rolexes cost $5000 in the store.	E
This is a Rolex.	G
I will sell it to you for $500.	H
Therefore, you get a 90% discount.	p
Rolexes are desirable.	J
Thus, this watch is desirable.	q
This is a rare deal.	L
So, there are no better deals around.	-s
If you get a 90% discount on something that is desirable, you should buy it, unless there are better deals around.	(p&q&-s) → r
————	
Therefore, you should buy it.	r

Here I keep the original letters p, q, r, and s and use capitals for the other claims within the argument. If we restrict our attention to the lower case stuff we see what looks like a valid argument:

p
q
-s
(p&q&-s)→r
———
r

In what sense does this look valid? Well, try to interpret the premises as true while avoiding rendering the conclusion true. You cannot do it. If all four of those premise schemata are true, you must take r to be true too. Here we see the value of including a conditional in our argument. When we formalize an argument, as above, we cannot understand it as being valid unless we have a line that ties the premises to the conclusion. For example, consider the argument without the conditional premise:

p
q
-s
______
r

With this schematization of the argument, interpreting the premises as true does not *require* you to interpret the conclusion as true too. So schematized, the premises have nothing to do with the conclusion because they do not contain the sentence letter r. You see this point when you look back at the schematization of the whole argument. The capital letters play no obvious role in the argument as a whole. Why are they there? Once the argument has been schematized, we have no idea what role the sentences corresponding to the capital letters play. In this case, we might want to add explicit conditional premises so as to make the structure of the argument clear:

E
G
H
(E&G&H)→p
Therefore, p
J
(G&J)→q
Therefore, q
L
L→-s
Therefore, -s
(p&q&-s)→r
______
Therefore, r

This argument, so schematized, wears its validity on its sleeve. This is a *formally valid* argument.

It's important to note that pretty much any valid argument in ordinary language, no matter how pedantic and explicit, can be schematized accurately, but in a way that is *not* formally valid. To see this, let's say that you had explicitly, pedantically, stated each of those conditional sentences, like:

> Rolexes are desirable.
> This is a Rolex.
> If this is a Rolex and Rolexes are desirable, then Rolexes are desirable.
> Therefore, this is desirable.

You could accurately, unhelpfully schematize this argument as:

> a
> b
> c
> Therefore, d.

This schema is not formally valid, but it is not inaccurate, either. It is *useless*, but not inaccurate. We value schematization because it can show arguments to be valid when they are, even though it need not do so. The good way to schematize that subargument above is:

> a
> b
> (a & b) → c
> ____
> c

Your goal when schematizing arguments is first to be accurate and then to be useful. What would an *inaccurate* schematization be? There are a couple of varieties. First

> a
> b
> (a v b) → c
> ____
> c

This fails to capture the structure of the argument, because it casts the antecedent of the conditional as being a disjunction, not a conjunction.

This schema is formally valid, but it does not accurately reflect the argument in English from which it is derived.

Another fail:

$$\begin{array}{l} a \\ a \\ (a \mathbin{\&} b) \rightarrow c \\ \hline c \end{array}$$

This schematization uses the same sentence letter "a" to stand for two distinct sentences. That is a no-no. Also, this argument schema is not valid. Can you show why? Try it out.

More generally, how do you show that an argument is formally valid? One excellent way is to make a single schema out of the argument. This schema will be a *conditional* whose *antecedent* is a conjunction of the premises and whose *consequent* is the argument's conclusion. For example, the argument

$$\begin{array}{l} a \\ b \\ (a \mathbin{\&} b) \rightarrow c \\ \hline c \end{array}$$

Becomes:

$$\{a \mathbin{\&} b \mathbin{\&} [(a \mathbin{\&} b) \rightarrow c]\} \rightarrow c$$

You then check whether that schema can be interpreted as being false. It cannot, as the following truth table shows:

a	b	c	[a&b&(a&b)→c]→c
T	T	T	T
T	T	F	T
T	F	T	T
F	T	T	T
T	F	F	T
F	T	F	T
F	F	T	T
F	F	F	T

That messy schema "{a&b&[(a&b)→c]}→c" is valid, in that there is no way to assign truth values to its parts that end up resulting in the whole thing being false. Any way you assign truth values to a, b, and c, the whole schema comes out true. That's the sign of a formally valid argument: you can make a formally valid schema out of it by conjoining its premises in the antecedent of a conditional and using its conclusion as the conditional's consequent (you can check for validity by using a truth table, as above).

It can often be difficult to know how one ought to schematize a given English sentence. It's always easy to schematize a sentence accurately: just use a single sentence letter like "p". But when a sentence is complex, and you know what parts of it are themselves statements that can be true or false, you want to schematize it by using separate sentence letters for each of its parts. And doing *that* can be quite tricky. It takes practice. Please practice!

*