VISTOOMA, VIsualisation TOOl for MAth. Module 6: The Logical Calculus

Signe Hermann

June 11, 2010

Contents

1 Module 6: The Logical Calculus

The logical calculus is a symbolic representation of the way we argue mathematically. It is only concerned with the *shape* of arguments, but it is, never the less, the basis for constructing mathematical proofs and the fundament on which all mathematical argumentation rests.

It is based on a *list of tautologies*, and in a manner similar to solving a crosswords puzzle, this list provides us with the rules for interchanging one proposition with another. Once this technique is mastered, it is easy to see the same argumentational patterns repeated every time you read a mathematical proof.

We do logical calculus by starting with a set of propositions called the hypotheses, reading through the list of tautologies, adding the propositions that this list tells us are logically equivalent to the ones we already have, and repeating this process with the new set of propositions until we arrive at a conclusion, creating a calculus such as the following:

Given the propositions q and $p \leftrightarrow q$, we get the following calculus:

1) q	(hypothesis)
2) $p \leftrightarrow q$	(hypothesis)
3) $(p \rightarrow q) \wedge (q \rightarrow p)$	(equivalence of 2))
4) $q \rightarrow p$	(logical consequence of 3))
5) p	(modus ponens of 1) and 4))

The challenge for math students is partly that the symbols p, $\neg q$, r and so on are merely placeholders in the same way as x is it in an equation and may, confusingly, be represented by each other in the list of tautologies, and partly that propositions may very well be made up of other propositions, and thus play the role of both atoms and propositions to be evaluated.

Note that there are usually many ways of arriving at a conclusion given the same set of propositions to begin with, depending on the student's preferences and creativity.

1.1 Software Requirements for Module 6: The Logical Calculus

1.1.1 Software Requirements for Mathematical Propositions

This is a recycling of the software from module 5.

1.1.2 Software Requirements for The Logical Calculus

The most important feature here is the routines that will be able to translate propositions, no matter whether they're composite or atomic, into a form that just represents the shape of the argument so that it can be compared with the list of tautologies, represented in a similar manner.

A recursive function is needed to scan the input list of propositions in the logical calculus, translate them into the form that just represents the shape of the argument, compare them with the list of tautologies and generate a list of possible new propositions and compare the student's input wiht that list, at the same time dynamically expanding the logical calculus shown on the screen, which is some sort of table with 2 columns, one for propositions and one for justifications or explanations of the new propositions.

Vistooma needs to have some sort of table look-up function for its representation of the list of tautologies so that the justification of the arguments can be added in the column next to the new propositions each time one is added.

1.1.3 Worked examples

This requires a simple database of worked examples of proofs by induction, along with a random generator.

1.2 "User Manual": Mathematical Propositions

See description in Module 5.

1.3 "User Manual": Doing Logical Calculus with Vistooma

Once you open Module 6, you're provided with a logical calculus dialogue box and the list of tautologies provided here below:

Number	Tautology	Name
1.	$p \vee \neg p$	
2.	$\neg(p \wedge \neg p)$	
3.	$p \to p$	
4a.	$p \leftrightarrow (p \vee p)$	
4b.	$p \leftrightarrow (p \wedge p)$	idempotent laws
5.	$\neg\neg p \leftrightarrow p$	double negation
6a.	$(p \vee q) \leftrightarrow (q \vee p)$	
6b.	$(p \wedge q) \leftrightarrow (q \wedge p)$	
6c.	$(p \leftrightarrow q) \leftrightarrow (q \leftrightarrow p)$	commutative laws
7a.	$(p \vee (q \vee r)) \leftrightarrow ((p \vee q) \vee r)$	
7b.	$(p \wedge (q \wedge r)) \leftrightarrow ((p \wedge q) \wedge r)$	associative laws
8a.	$(p \wedge (q \vee r)) \leftrightarrow ((p \wedge q) \vee (p \wedge r))$	
8b.	$(p \vee (q \wedge r)) \leftrightarrow ((p \vee q) \wedge (p \vee r))$	distributive laws
9a.	$(p \vee \mathbf{c}) \leftrightarrow p$	
9b.	$(p \wedge \mathbf{c}) \leftrightarrow \mathbf{c}$	
9c.	$(p \vee \mathbf{t}) \leftrightarrow \mathbf{t}$	
9d.	$(p \wedge \mathbf{t}) \leftrightarrow p$	identity laws
10a.	$\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$	
10b.	$\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$	De Morgan's Laws
11a.	$(p \leftrightarrow q) \leftrightarrow ((p \to q) \wedge (q \to p))$	
11b.	$(p \leftrightarrow q) \leftrightarrow ((p \wedge q) \vee (\neg p \wedge \neg q))$	
11c.	$(p \leftrightarrow q) \leftrightarrow (\neg p \leftrightarrow \neg q)$	equivalence
12a.	$(p \to q) \leftrightarrow (\neg p \vee q)$	
12b.	$\neg(p \to q) \leftrightarrow (p \wedge \neg q)$	implication
13.	$(p \to q) \leftrightarrow (\neg q \to \neg p)$	contrapositive
14.	$(p \to q) \leftrightarrow ((p \wedge \neg q) \to \mathbf{c})$	reductio ad absurdum
15a.	$((p \to r) \wedge (q \to r)) \leftrightarrow ((p \vee q) \to r)$	
15b.	$((p \to q) \wedge (p \to r)) \leftrightarrow (p \to (q \wedge r))$	
16.	$((p \wedge q) \to r) \leftrightarrow (p \to (q \to r))$	exportation law
17.	$p \to (p \vee q)$	addition
18.	$(p \wedge q) \to p$	simplification
19.	$(p \wedge (p \to q)) \to q$	modus ponens
20.	$((p \to q) \wedge \neg q) \to \neg p$	modus tollens
21.	$((p \to q) \wedge (q \to r)) \to (p \to r)$	hypothetical syllogism
22.	$((p \vee q) \wedge \neg p) \to q$	disjunctive syllogism
23.	$(p \to \mathbf{c}) \to \neg p$	absurdity
24.	$((p \to q) \wedge (r \to s)) \to ((p \vee r) \to (q \vee s))$	
25.	$(p \to q) \to ((p \vee r) \to (q \vee r))$	

At first, you're asked to type in your hypotheses, e.g.

$\neg p \leftrightarrow q$
q

Vistooma will then generate a culculus table such as the following:

1) $\neg p \leftrightarrow q$	Hypothesis
2) q	Hypothesis
3)	

You can add more rows by clicking on the calculus and select "Add Row". Vistooma will internally evaluate the shape of the proposals and look them up in the list of tautologies, generating a list of possible logical equivalences and consequences. When you type in a new proposition in the calculus, Vistooma will compare it with the list it has generated and return an error message if your new proposition cannot be arrived at as a logical consequence of equivalence of the propositions given.

If your new proposition can be arrived at, Vistooma will write the justification next to it, such as

1) $\neg p \leftrightarrow q$	Hypothesis
2) q	Hypothesis
3) $\neg p \rightarrow q \wedge q \rightarrow \neg p$	Equivalence of 1)
4)	

The process just described will be repeated with the new set of propositions, leading to a calculus such as the following:

1) $\neg p \leftrightarrow q$	Hypothesis
2) q	Hypothesis
3) $\neg p \rightarrow q \wedge q \rightarrow \neg p$	Equivalence of 1)
4) $\neg p \rightarrow q$	Logical consequence of 3)
5) $q \rightarrow \neg p$	Logical consequence of 3)
6)	

And again:

1) $\neg p \leftrightarrow q$	Hypothesis
2) q	Hypothesis
3) $\neg p \rightarrow q \wedge q \rightarrow \neg p$	Equivalence of 1)
4) $\neg p \rightarrow q$	Logical consequence of 3)
5) $q \rightarrow \neg p$	Logical consequence of 3)
6) $\neg p$	Modus ponens of 2) and 5)

If you think that $\neg p$ is a suitable conslucion, you can click on the calculus and choose "Conclusion Reached". Vistooma will then add a facit line so as to look like this:

1) $\neg p \leftrightarrow q$	Hypothesis
2) q	Hypothesis
3) $\neg p \rightarrow q \wedge q \rightarrow \neg p$	Equivalence of 1)
4) $\neg p \rightarrow q$	Logical consequence of 3)
5) $q \rightarrow \neg p$	Logical consequence of 3)
6) $\neg p$	Modus ponens of 2) and 5)

1.4 "User Manual": Worked Examples

Vistooma provides a selection of worked examples for each module. Here, you can randomly generate a worked example to get a feel for the functionalities or to practise your understanding by looking at examples.