Propositional Argument Order

by Sven Nilsen, 2020

In this paper I present a technique to assign an order to arguments of propositional logic.

Proofs in propositional logic are normally used to check that they return `true` for all possible inputs. However, there is another use case where one wants to find a sort given by some map `f`:

$$f : nat \to nat$$

Such that (interpreted informally):

∀ A, B { imply(imply(A, B), le(f(“A”), f(“B”))) }

Where `imply` is material implication and `le` is the `<=` operator (less or equal).
Here I use quotes to refer to the argument `A` (this is usually impossible because it is a proposition).

I suggest counting the number of cases when an argument is `true` and the output is `true`.

For example, for material implication itself:

A	**B**	**A => B**
0	0	1
0	1	1
1	0	0
1	1	1
1	2	

Here, `A` is `true` 1 time when `A => B` is `true`.
`B` is `true` 2 times when `A => B` is true.

Another example:

A	**B**	**C**	**(A => B) ∧ (B => C)**
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1
1	2	3	