DMP Notes K. Sutner

Formal Proofs

2023/06/09 v0.6

1 Formal Proofs

So far, we have discussed the syntax and semantics of propositional logic. There is one other important aspect of logic that we have so far ignored: proof theory. Proof theory is based on the fundamental insight that proofs, the hallmark of mathematical reasoning, can themselves be analyzed using mathematical tools. We can give rigorous definitions of what constitutes a proof, we can study the properties of proofs and, most amazingly, we can establish theorems about proofs. Just as we can develop a theory of numbers, we can also develop a theory of proofs.

This may sound a little over the top, after all, novices have a hard time finding proofs at all, even when the claim in question is fairly modest. True enough, and it is important to develop one's proof-building skills first, concerns about the theory come later. For a great many practicing mathematicians, later means never. However, in the presence of digital computers, the ground has started to shift, now there are powerful programs that can help to check, manipulate and even find proofs: proof checkers, proof assistants and theorem provers.

Proof checkers have a relatively simple task: there are given an alleged proof as input and are supposed to verify that it is in fact a correct proof. For this to work, the proof in question cannot be copied out of a mathematics or computer science journal, it needs to be formalized first. Formal proofs follow a very rigid and precise syntax, they tend to be quite long, and, worst of all, they are not very easy for humans to understand. Still, a formal proof that has gone through a checker can safely be assumed to be correct. The same cannot be said about all proofs published in journals, even respectable ones. A theorem prover goes one step further in that it tries to find the proof in the first place. For serious theorems this usually involves working in wildly exponential search spaces and is quite often computationally difficult to impossible. However, it is perfectly feasible for a theorem prover to fill in parts of an argument, to establish relatively simple propositions, stepping stones that ultimately lead to the otherwise elusive goal. Finding the right propositions and organizing them in a way that ultimately congeals into the "big proof" currently needs to handled by human domain experts (and it is not clear that this will change, formalizing human intuition sounds like a contradiction in terms).

Provers can be very powerful when it comes to formal verification: one is given a complex system and one has to demonstrate that it has certain properties. Typical examples are software and hardware systems as well as protocols. Reasoning about systems by hand is often utterly hopeless, there are too many interrelated parts that are mostly combinatorial in nature. If you want to get an idea what a state-of-the-art prover looks like, consider LEAN. This system has been developed at MS Research and was originally intended for program verification, but has become quite popular in the mathematical community. Proving results in, say, graph theory, requires a huge library of formalized mathematics before one even can get started; building such a library is a sysiphean undertaking and can only be handled by international cooperation. See mathlib for the current state of affairs.

Let's return to the logic puzzle from section 2. We gave a fairly rigorous proof that the list of assertions given there has the consequence that $E \Rightarrow \neg G$. The argument is rigorous, but falls short of being a formal proof. For example, to make the first step in the argument more precise, we would need to show that

$$\text{MP-axioms}, E, G \quad \text{implies} \quad \neg F$$

What would be a solid framework for this kind of argument? Apart from the given assumptions we also want access to some basic properties of propositional logic, and we want a mechanism that allows to draw further conclusions from already established facts.

A little experimentation shows that we need two critical ingredients:

- Axioms: a collection of formulae that we think of as given, as unchallenged. Axioms can be used in an argument without further justification.
- Rules of Inference: methods that make it possible to derive new formulae from given ones. These rules are syntactical in nature and do not require any reference to the meaning of the formulae in question.

Axioms and rules of inference together form a formal system, a framework in which we can study proofs. Needless to say, we will here only consider a formal system built around propositional logic. This is nowhere near enough to model arguments that one encounters in realistic mathematical contexts; one needs more powerful systems of logic; take a look at LEAN to see how this might work. Still, many of the main ideas are quite similar and one can get at least some impression of how things would work in more advanced systems.

As a general rule, axioms are chosen to be as simple as ever possible. In the case of propositional logic, we might consider axioms of the form $\varphi \Rightarrow \varphi$. At first glance, these axioms might seem utterly useless, but we will see in a moment that, together with the right rules of inference, they are entirely sufficient. Most rules of inference require one or two hypotheses to produce a conclusion. A famous example is the modus ponens (also known as detachment): from φ and $\varphi \Rightarrow \psi$ we can conclude ψ. This is usually written

$$\frac{\varphi \qquad \varphi \Rightarrow \psi}{\psi}$$

The hypotheses are written above the horizontal line, and the conclusion is written underneath. This notation comes in handy when one constructs proof trees, see below. As one might suspect, there is a trade-off between the axioms and the rules of inference: one can have powerful axioms and weak rules, or the other way around. Either approach has advantages and disadvantages.

At any rate, for propositional logic all the axioms all have to be tautologies. Moreover, the rules of inference must be sound: when applied to tautologies, they must produce new tautologies.

For example, modus ponens is clearly sound. So in a sound system we can only derive tautologies from the axioms.

Unfortunately, this is the easy part. We also would like the system to be complete, we want it to be capable of producing all tautologies. It might be tempting to accomplish this by simply declaring all tautologies to be axioms, but that completely defeats the purpose of the whole exercise: we want a nice, purely syntactical way of generating all tautologies from just a few, very simple ones. So we want to be able to derive every tautology given our axioms and rules of inference. In our toy formal system we can precisely define what we mean by a (formal) proof or a derivation. A proof of a particular formula ψ is a sequence of formulae

$$\varphi_0, \varphi_1, \varphi_2, \ldots, \varphi_{n-1}, \varphi_n = \psi$$

where each term is either an axiom, or can be derived from prior terms in the sequence by applying some rule of inference. Since we insist that axioms are easy to recognize, and that the rules are based on straightforward pattern matching, it is in a sense trivial to check whether a given sequence of formulae is indeed a proof: we check that φ_0 is an axiom, that φ_1 is either an axiom or derived from φ_0, that φ_2 is either an axiom or derived from φ_0 and φ_1, and so on and so forth, until we get to the conclusion ψ.

Just to be clear, checking a proof is one thing, finding it in the first place is quite another. We are given the target ψ and are supposed to somehow concoct the φ_k—assuming that ψ really is a tautology, otherwise our attempts are doomed from the start. This intuitive understanding that proof checking is easier than proof search can be made precise in complexity theory, it is not just a hunch. It certainly matches up perfectly with experience.

We now have a fairly precise definition of provability or derivability; in symbols

$$\vdash \ \psi$$

This means that there is a proof of ψ that uses only the given axioms and rules of inference. More useful in practice is a slight generalization of this concept: we allow some set Γ of hypotheses or premises and think of them as being temporarily added to the pure axioms. This is written

$$\Gamma \ \vdash \ \psi$$

Plain provability relies only on the permanent axioms and thus corresponds to the special case $\Gamma = \emptyset$. On the other hand, since every proof is finite, $\Gamma \ \vdash \ \psi$ means that $\gamma_1 \land \gamma_2 \land \ldots \land \gamma_k \Rightarrow \psi$ for appropriate premises $\gamma_i \in \Gamma$.

As an example, in our original puzzle, we can think of Γ as the list of given MP-axioms. The goal of the puzzle is to show that $\Gamma \ \vdash \ \neg(E \land G)$. We will see in a moment how this task can be handled in a particular formal system of propositional logic, see section 2.

2 A Hilbert System

We need to specify a formal system with specific axioms and rules of inference. There are many ways to do this, we have chosen a particularly simple system $\mathcal{H}$, a so-called Hilbert style system for propositional logic, where we could construct a proof for the math profs puzzle explicitly.

Note the hedge: “we could,” but we actually will only give a sketch of the full proof, simply because the full argument would be far too long and boring. We are just trying to get an idea here how the argument could work. In section 3 we will produce some full proofs, albeit in a different system that is more suitable for this purpose.

At any rate, in system $\mathcal{H}$ there is only one type of axiom, and four rules of inference; of those, three are entirely trivial. It is hard to beat this system in its simplicity[*]. This simplicity does have a huge drawback, though: the system is basically impossible to use in any practical sense. It is a proof of concept, not a system anyone would use in a proof assistant, or in any actual mathematical argument. For that purpose, there are *much* better solutions as in section 3, but they are substantially more complicated to describe and not really suitable as an introduction.

One feature that helps cutting down on complexity in $\mathcal{H}$ is that it uses only two connectives: negation and disjunction. If we want to use other connectives, we have to add them as abbreviations. The great advantage of this minimalist approach is that a complete description of the system is quite small:

Logical Axioms:

tertium non datur $\qquad \neg\varphi \lor \varphi$

Rules of Inference:

expansion $\qquad \dfrac{\varphi}{\psi \lor \varphi}$

contraction $\qquad \dfrac{\varphi \lor \varphi}{\varphi}$

associativity $\qquad \dfrac{\varphi \lor (\psi \lor \chi)}{(\varphi \lor \psi) \lor \chi}$

cut rule $\qquad \dfrac{\varphi \lor \psi \qquad \neg\varphi \lor \chi}{\psi \lor \chi}$

Again, these axioms and rules are purely syntactic, it is emphatically not allowed to replace a formula by an equivalent one. For example, $\varphi \lor \neg\varphi$ is not an axiom, no matter how much you would like it to be one. Similarly, you cannot conclude $\varphi \lor \psi$ from φ using expansion, the ψ must be on the left. Still, the first three rules are blindingly obvious. To make the cut rule more intelligible, let us introduce a few abbreviations that will come in handy in several other places.

$$\begin{aligned} \bot &:= \neg(\neg p \lor p) \\ \varphi \land \psi &:= \neg(\neg\varphi \lor \neg\psi) \\ \varphi \Rightarrow \psi &:= \neg\varphi \lor \psi \end{aligned}$$

Beware, though, there are no rules of inference for these new symbols, we have to unfold the definitions, apply the original rules, and then, perhaps, reintroduce some abbreviations. The choice of p in the definition of $\bot$ is arbitrary, any variable will do. We can now write the cut

[*] Full disclosure: Hilbert is my academic great-grandfather, so I’m not entirely impartial in this matter.

rule like so:

$$\frac{\varphi \lor \psi \qquad \varphi \Rightarrow \chi}{\psi \lor \chi}$$

This makes a bit more sense; clearly the conclusion is a weaker statement than the first hypothesis. If you expected the conclusion to be $\chi \lor \psi$ you are absolutely right, however, this form is required since we have no built-in commutativity.

The axioms in $\mathcal{H}$ are trivially tautologies and it is easy to check that the rules, when applied to tautologies, will only produce more tautologies. Hence, the system $\mathcal{H}$ is sound, it can only prove tautologies. What would simple proofs in the system look like? Since the axioms and rules are quite so primitive, it is not even clear that we can handle really straightforward assertions. For example, suppose we want to show that $\vdash p \lor \neg p$. A proof according to our original definition is the sequence of formulae

$$\neg p \lor p \quad \neg\neg p \lor \neg p \quad p \lor \neg p$$

but this representation is not particularly useful. Who wants to search for the necessary justifications that demonstrate that this is not some nonsense sequence, but a perfectly valid proof. It is better to write a numbered table where the sequence appears as a column, together with annotations that explain why each step is admissible. Here is the proof converted into such a table.

1	$\neg p \lor p$	axiom
2	$\neg\neg p \lor \neg p$	axiom
3	$p \lor \neg p$	cut, $1, 2$

To find proofs such as the preceding one, it is often helpful to do a bit of reverse engineering: it seems like a good guess that the last rule used in a putative proof must be a cut. We can figure out what the various formulae φ, ψ and χ involved in the cut are and then try to prove these. With luck, they might be premises or axioms. In a similar way we can display a proof for $p \vdash p \lor q$.

1	p	premise
2	$q \lor p$	expansion, 1
3	$\neg q \lor q$	axiom
3	$p \lor q$	cut, $2, 3$

Lemma 2.1 *Commutativity disjunctions: $p \lor q \vdash q \lor p$.*

Proof.

1	$p \lor q$	premise
3	$\neg p \lor p$	axiom
3	$q \lor p$	cut, $2, 3$

□

It is clear from these examples that cut is the power rule in $\mathcal{H}$, the others are auxiliary. To be clear, we cannot drop any of them without ruining completeness. Try.

Lemma 2.2 *Double Negation:* $\neg\neg p \vdash p$.

1	$\neg\neg p$	premise
2	$p \lor \neg\neg p$	expansion, 1
3	$\neg\neg p \lor p$	commutativity, 2
4	$\neg p \lor p$	axiom
5	$p \lor p$	cut, $4, 3$
6	p	contraction, 5

□

Note that step 3 in the last proof was justified not by a rule of inference, but by the previous lemma 2.1. In fact, we can think of these lemmata as additional rules of inference, adding them to the core rules will not affect the derivable formulae overall, but it will shorten some proofs.

$$\frac{q \lor p}{p \lor q}\ (\mathsf{com}) \qquad\qquad \frac{\neg\neg p}{p}\ (\mathsf{dneg})$$

So far so good, but how about something a little more complicated? Here is an example of a proof that essentially shows that *modus ponens* works in our system (except that we have to rewrite the implication as a disjunction for the proof).

Lemma 2.3 *modus ponens:* $p, p \Rightarrow q \ \vdash \ q$.

Proof.

1	p	premise
2	$q \lor p$	expansion, 1
3	$p \lor q$	commutativity, 2
4	$\neg p \lor q$	premise
5	$q \lor q$	cut, $3, 4$
6	q	contraction, 5

We could have used lemma 2.1 to get from line 2 to line 4 directly. □

By induction, we can generalize the lemma to an arbitrary number of hypotheses:

$$p_1, p_2, \ldots, p_{n-1}, p_1 \Rightarrow p_2 \Rightarrow \ldots \Rightarrow p_{n-1} \Rightarrow q \vdash q$$

Here we treat implication as right-associative, $p \Rightarrow q \Rightarrow r$ is short for $p \Rightarrow (q \Rightarrow r)$.

Even annotated proof tables are not particularly easy to read for humans, it is often preferable to organize a derivation in tree form using the fractional notation from above:

$$\cfrac{\cfrac{\cfrac{\cfrac{p}{q \lor p}\ \textsf{(exp)} \qquad \neg q \lor q}{p \lor q}\ \textsf{(cut)} \qquad \neg p \lor q}{q \lor q}\ \textsf{(cut)}}{q}\ \textsf{(cont)}$$

The root is the proven formula, and the hypotheses and axioms are at the leaves. The labels indicate the type of rule used, making it easy to check correctness. Here is the tiny proof tree for commutativity.

$$\frac{p \lor q \qquad \neg p \lor p}{q \lor p}\ \textsf{(cut)}$$

As promised, with annotated tables or proof trees make it relatively easy to check correctness, at least once one has memorized the rules. Still, the proofs in $\mathcal{H}$ are not well suited for humans, the steps typically do not correspond to anything we would think of as natural reasoning, and, quite often, there just are too many steps. Everything works out in the end, but why should the argument be organized this way? You might have a sense of déjà vu all over again, we have been in this situation before in the simplification of formulae. There we had to deal with equivalence transformations instead of inference rules, but similar difficulties arise. It turns out that a standard way to come up with somewhat complicated proofs in $\mathcal{H}$ is to first construct a proof in a different, better behaved system (natural deduction), and then to translate the proof back to $\mathcal{H}$. Again, the only reason for using $\mathcal{H}$ is that it is so very primitive.

The example show that we can prove at least a few tautological statements in $\mathcal{H}$, but we really need to establish completeness.

Theorem 2.1 (Completeness Theorem)
The deduction system $\mathcal{H}$ is complete: all tautologies can be derived in it.

Just as an indication of what needed to be done if we want to show that the system is indeed complete, here is a little lemma. To preserve the gentle reader's sanity, we write $\varphi_1 \lor \varphi_2 \lor \ldots \lor \varphi_n$ as shorthand for $\varphi_1 \lor (\varphi_2 \lor (\ldots(\varphi_{n-1} \lor \varphi_n)\ldots)$, the properly right-associative version of a long disjunction.

Lemma 2.4 (Expansion Lemma)
Let $1 \leq i_1, i_2, \ldots, i_m \leq n$. Then $\quad \varphi_{i_1} \lor \varphi_{i_2} \lor \ldots \lor \varphi_{i_m} \vdash \varphi_1 \lor \varphi_2 \lor \ldots \lor \varphi_n$.

Semantically, this is mind-numbingly obvious: if we add a few terms to a disjunction, the bigger disjunction follows from the shorter one. Duh. Alas, we need a proof in $\mathcal{H}$, and that turns out to require induction on m and a few major convulsions. Here are the details.

Proof. The argument is by induction on m, the number of disjuncts on the left.

For $m = 1$ let $i = i_1$. The proof uses expansion, the commutativity lemma 2.1, and another chain of expansion steps. We refer to the lemma by label (com).

$$\begin{array}{cl}
\varphi_i & \\ \hline
(\varphi_{i+1} \lor \ldots \lor \varphi_n) \lor \varphi_i & (\mathsf{exp}) \\ \hline
\varphi_i \lor \varphi_{i+1} \lor \ldots \lor \varphi_n & (\mathsf{com}) \\ \hline
\varphi_{i-1} \lor \varphi_i \lor \ldots \lor \varphi_n & (\mathsf{exp}) \\ \hline
\vdots & (\mathsf{exp}) \\ \hline
\varphi_1 \lor \ldots \lor \varphi_n & (\mathsf{exp})
\end{array}$$

Next consider $m = 2$ and write $i = i_1$, $j = i_2$. If $i = j$ we can use contraction in the first step and the proceed as in the case $m = 1$. By commutativity we may safely assume $i < j$. We use induction on $n \geq 2$. The case $n = 2$ is trivial, so assume $n > 2$ and let $\psi = \varphi_3 \lor \ldots \lor \varphi_n$. We consider 3 case depending on i and j. If $i \geq 2$ we have

$$\begin{array}{cl}
\varphi_i \lor \varphi_j & \\ \hline
\varphi_2 \lor \psi & (\mathsf{IH}) \\ \hline
\varphi_1 \lor \varphi_2 \lor \psi & (\mathsf{com})
\end{array}$$

If $i = 1$ and $j \geq 3$ we have

$$\begin{array}{cl}
\varphi_1 \lor \varphi_j & \\ \hline
\varphi_i \lor \psi & (\mathsf{IH}) \\ \hline
\psi \lor \varphi_i & (\mathsf{com}) \\ \hline
\varphi_2 \lor (\psi \lor \varphi_1) & (\mathsf{exp}) \\ \hline
(\varphi_2 \lor \psi) \lor \varphi_1 & (\mathsf{assc}) \\ \hline
\varphi_1 \lor \varphi_2 \lor \psi & (\mathsf{com})
\end{array}$$

Lastly, we consider $i = 1$ and $j = 2$.

$$\begin{array}{cl}
\varphi_1 \lor \varphi_2 & \\ \hline
\psi \lor (\varphi_1 \lor \varphi_2) & (\mathsf{exp}) \\ \hline
(\psi \lor \varphi_1) \lor \varphi_2 & (\mathsf{assc}) \\ \hline
\varphi_2 \lor (\psi \lor \varphi_1) & (\mathsf{com}) \\ \hline
(\varphi_2 \lor \psi) \lor \varphi_1 & (\mathsf{assc}) \\ \hline
\varphi_1 \lor \varphi_2 \lor \psi & (\mathsf{com})
\end{array}$$

Now for the final argument: $m > 2$. Write $\psi = \varphi_1 \lor \varphi_2 \ldots \lor \varphi_n$.

$$\begin{array}{cl}
\varphi_{i_1} \lor (\varphi_{i_2} \ldots \varphi_{i_m}) & \\ \hline
(\varphi_{i_1} \lor \varphi_{i_2}) \lor (\varphi_{i_3} \ldots \varphi_{i_m}) & (\mathsf{assc}) \\ \hline
(\varphi_{i_1} \lor \varphi_{i_2}) \lor \psi & (\mathsf{IH}) \\ \hline
\psi \lor (\varphi_{i_1} \lor \varphi_{i_2}) & (\mathsf{com}) \\ \hline
(\psi \lor \varphi_{i_1}) \lor \varphi_{i_2} & (\mathsf{assc}) \\ \hline
(\psi \lor \varphi_{i_1}) \lor \psi & (\mathsf{IH}) \\ \hline
\psi \lor (\psi \lor \varphi_{i_1}) & (\mathsf{com}) \\ \hline
(\psi \lor \psi) \lor \varphi_{i_1} & (\mathsf{assc}) \\ \hline
(\psi \lor \psi) \lor (\psi \lor \psi) & (\mathsf{IH}) \\ \hline
\psi \lor \psi & (\mathsf{cont}) \\ \hline
\psi & (\mathsf{cont})
\end{array}$$

□

Since Hilbert systems are quite so spartan, one needs results like the expansion lemma to work with them at all. Another very helpful result that holds in $\mathcal{H}$ is the following.

Theorem 2.2 (Deduction Theorem)
$\Gamma, \varphi \vdash \psi$ implies $\Gamma \vdash \varphi \Rightarrow \psi$

So the theorem says that if we can prove ψ from assumptions Γ and the additional assumption φ, then we can already prove $\varphi \Rightarrow \psi$ directly from Γ. Well, we can prove $\neg\varphi \vee \psi$ and use our abbreviations. This is a very natural feature that holds in all civilized systems and corresponds exactly to the way one usually reasons in mathematics to establish an implication: assume the hypothesis of the implication, then argue for a while, using whatever other knowledge is available, and finally produce the conclusion.

Just to be clear, there are many alternatives to the system $\mathcal{H}$. For example, here is a well-studied system based on implication $\Rightarrow$ and the constant $\bot$. The axioms all have their own names, and the whole system is known as Tarski-Bernays-Waijsberg:

$$\begin{array}{rl} \text{simplification} & p \Rightarrow (q \Rightarrow p) \\ \text{Peirce's law} & ((p \Rightarrow q) \Rightarrow p) \Rightarrow p \\ \text{hypothetical syllogism} & (p \Rightarrow q) \Rightarrow (q \Rightarrow r) \Rightarrow (p \Rightarrow r) \\ \text{ex falso quodlibet} & \bot \Rightarrow p \end{array}$$

The first three axioms all make sense, but the last one is a bit peculiar. To be sure, it is a tautology, but it is unclear how one would use this axiom in any sort of proof. Modus ponens is the only rule of inference (that's a white lie, we also need substitution, we can replace propositional variables by formulae). Again one can show that this system is complete, it derives exactly all tautologies, just like $\mathcal{H}$. The structure of the proofs would change completely, but exactly the same formulae are provable, after we add missing connectives by definition to both systems. Incidentally, finding proofs in this system is a real challenge.

The Math Profs

If we want to give a proof for the math prof puzzle in $\mathcal{H}$, we have to rewrite the given hypotheses in terms of just disjunctions and negation. The conversion algorithm I used decided to sort the variables, but that is not a problem: the expansion lemma 2.4 allows us to move things around if need be. For example, we have $x \vee y \vee z \vdash y \vee z \vee x$. Recall that we generally assume that binary operators associate to the right. The math prof hypotheses now look like this:

$$\begin{array}{ll}
\neg x_1 \lor \neg x_2 \lor \neg x_3 \lor x_4 & \neg x_5 \lor x_{11} \lor x_{12} \\
x_1 \lor \neg x_5 \lor x_6 & \neg x_5 \lor x_{10} \lor x_{13} \\
\neg x_4 \lor \neg x_7 \lor \neg x_8 & x_6 \lor x_8 \lor \neg x_{12} \lor \neg x_{13} \\
x_3 \lor \neg x_5 \lor \neg x_9 & x_6 \lor \neg x_7 \lor \neg x_{10} \\
x_2 \lor \neg x_7 \lor x_{10} & \neg x_5 \lor \neg x_6 \lor \neg x_7 \\
\neg x_2 \lor x_4 \lor \neg x_8 \lor x_9 & \neg x_1 \lor x_8 \lor \neg x_{11} \lor \neg x_{13}
\end{array}$$

Let us call this set of formulae Γ. We are supposed to show that $\Gamma \vdash \neg x_5 \lor \neg x_7$. We could invoke the completeness theorem 2.1, but then we would need to show that $\Gamma \Rightarrow (\neg x_5 \lor \neg x_7)$ is a tautology. We could do this by building a truth table of size $2^{13} = 8192$, but the whole point of our little exercise is to show that we can get around such brute-force computations. By the deduction theorem 2.2, we have to show $\Gamma, x_5 \vdash \neg x_7$. One can show in $\mathcal{H}$ that $\neg x_7$ is equivalent to $x_7 \Rightarrow \bot$, see the exercises, so after another application of the deduction theorem we have to show that $\Gamma, x_5, x_7 \vdash \bot$.

The first step in constructing such a proof is to show that $\Gamma_{11}, x_5, x_7 \vdash \neg x_6$. This comes down to applying modus ponens twice.

$$\dfrac{x_7 \qquad \dfrac{\dfrac{x_5 \quad \neg x_5 \lor \neg x_6 \lor \neg x_7}{\neg x_6 \lor \neg x_7}\,(\mathsf{mop})}{\neg x_7 \lor \neg x_6}\,(\mathsf{com})}{\neg x_6}\,(\mathsf{mop})$$

At this point, we can add $\neg x_6$ to our list of assumptions. The next goal in the proof is to show that $\Gamma_2, x_5, x_7, \neg x_6 \vdash x_1$.

$$\dfrac{\neg x_6 \qquad \dfrac{\dfrac{\dfrac{x_5 \quad \dfrac{x_1 \lor \neg x_5 \lor x_6}{\neg x_5 \lor x_1 \lor x_6}\,(\mathsf{exp})}{x_1 \lor x_6}\,(\mathsf{mop})}{x_6 \lor x_1}\,(\mathsf{com})}{\neg\neg x_6 \lor x_1}\,(\mathsf{dneg})}{x_1}\,(\mathsf{mop})$$

The rule labeled (dneg) replaces φ by $\neg\neg\varphi$ and is clearly valid, but we have not shown that yet (we did show the opposite direction, $\neg\neg\varphi \vdash \varphi$).

At any rate, you get the idea. We are reformatting our perfectly rigorous but informal proof in the online course into a proof in $\mathcal{H}$. The argument becomes annoyingly long and tedious, writing it out in full would require a lot of stamina. Still, the critical advantage of this presentation is that it can be checked for correctness in a purely mechanical fashion. Again, the final proof is very easy to check for correctness, it all comes down to simple pattern matching. Finding the proof is a totally different matter, recall that we used a few common-sense heuristics to figure out the informal argument in the first place. In other words, we already knew a perfectly good proof sketch, all that was left to do is to fill in many pesky details. Without the benefit of our informal reasoning, finding a proof would be a nightmare.

For the math prof puzzle there is no pressing need to establish a perfectly reliable solution, but imagine the issue at hand were to avoid collisions between cars or airplanes. Most people would feel that the effort is well justified in these cases. Of course, for that sort of challenge we need much bigger guns, but the basic ideas are very similar.

3 Sequents

The sketch of a proof of the math prof puzzle in the system $\mathcal{H}$ is unsatisfactory in two ways: first off, it is just a sketch, there are lots of details missing; second, it is a translation of an entirely human made, heuristics based argument. It would be nice to see a proof that is complete and machine generated to emphasize the mechanical aspects of proving. To have any chance for this to happen, one has to move to a different formal system, the Hilbert style approach is just not suitable.

The question then becomes how one should organize a formal system for propositional logic that makes it computationally more accessible. The key problem with $\mathcal{H}$ is that the inference rules are not particularly intuitive. For example, to prove an implication $\varphi \Rightarrow \psi$ one would like to make the temporary assumption φ, then go through some proof steps, and conclude first ψ and ultimately $\varphi \Rightarrow \psi$. One says that the assumption φ is discharged, it is no longer an assumption. In $\mathcal{H}$ we have invoke the deduction theorem 2.2 to do this. Moreover, in longer arguments, it becomes difficult to keep track of dangling assumptions. We also would like to get rid of the nuisance of having to deal with commutativity and associativity of connectives directly, our proof mechanism should handle those automatically, just like a human prover would.

After some experimentation one is lead to the following key idea.

Definition 3.1 *A sequent consists of two finite sets of formulae Γ and Δ. Γ is the antecedent and Δ is the consequent of the sequent. Notation: $\Gamma \supset \Delta$.*

The intended meaning of a sequent is that the conjunction of the formulae in Γ (all the assumptions) implies the disjunction of the formulae in Δ. Loosely speaking, we have a list of hypotheses $\gamma_1, \dots, \gamma_n$ and a list of goals $\delta_1, \dots, \delta_m$ and we would like to establish the implication

$$\gamma_1 \wedge \ldots \wedge \gamma_n \;\Rightarrow\; \delta_1 \vee \ldots \vee \delta_m \tag{$*$}$$

Both lists may be empty, if the antecedent is empty, the consequent is a tautology; if the consequent is empty, the antecedent is a contradiction. To express the symmetry even more directly, we can also think of a sequent as representing the formula

$$\neg\gamma_1 \vee \ldots \vee \neg\gamma_n \vee \delta_1 \vee \ldots \vee \delta_m$$

It may sound tempting to replace the list Δ by a single formula δ, but that leads to a slightly different calculus that is less appropriate for out purposes.

Keeping the interpretation of "conjunction of antecedent implies disjunction of consequent" in mind, all sequents where the antecedent and consequent contain the same formula are trivially valid. These are called basic sequents and are adopted as axioms:

$$\frac{-}{\Gamma, \varphi \supset \varphi, \Delta.}$$

Note that we write sets as comma separated lists here and we have placed the "interesting" part of the antecedent and consequent near the sequent symbol $\supset$, just to improve legibility (admittedly, this is a matter of taste). As to the rules of inference, there will be two rules associated with each connective, depending on whether the connective appears in the antecedent or the consequent.

$$\begin{array}{lllll}
\text{Negation} & \dfrac{\mathit{\Gamma} \supset \varphi, \mathit{\Delta}}{\mathit{\Gamma}, \neg\varphi \supset \mathit{\Delta}} & \textsf{(not-L)} & \dfrac{\varphi, \mathit{\Gamma} \supset \mathit{\Delta}}{\mathit{\Gamma} \supset \mathit{\Delta}, \neg\varphi} & \textsf{(not-R)} \\
\text{And} & \dfrac{\varphi, \psi, \mathit{\Gamma} \supset \mathit{\Delta}}{\mathit{\Gamma}, \varphi \wedge \psi \supset \mathit{\Delta}} & \textsf{(and-L)} & \dfrac{\mathit{\Gamma} \supset \mathit{\Delta}, \varphi \quad \mathit{\Gamma} \supset \psi, \mathit{\Delta}}{\mathit{\Gamma} \supset \varphi \wedge \psi, \mathit{\Delta}} & \textsf{(and-R)} \\
\text{Or} & \dfrac{\mathit{\Gamma}, \varphi \supset \mathit{\Delta} \quad \psi, \mathit{\Gamma} \supset \mathit{\Delta}}{\mathit{\Gamma} \supset \varphi \vee \psi, \mathit{\Delta}} & \textsf{(or-L)} & \dfrac{\mathit{\Gamma} \supset \varphi, \psi, \mathit{\Delta}}{\mathit{\Gamma} \supset \varphi \vee \psi, \mathit{\Delta}} & \textsf{(or-R)} \\
\text{Implication} & \dfrac{\mathit{\Gamma} \supset \varphi, \mathit{\Delta} \quad \mathit{\Gamma}, \psi \supset \mathit{\Delta}}{\mathit{\Gamma}, \varphi \Rightarrow \psi \supset \mathit{\Delta}} & \textsf{(imp-L)} & \dfrac{\mathit{\Gamma}, \varphi \supset \psi, \mathit{\Delta}}{\mathit{\Gamma} \supset \varphi \Rightarrow \psi, \mathit{\Delta}} & \textsf{(imp-R)}
\end{array}$$

There is one more rule that allows us to chain together two arguments, removing an intermediate conclusion (the φ below).

Cut

$$\frac{\mathit{\Gamma} \supset \varphi, \mathit{\Delta} \quad \mathit{\Gamma}, \varphi \supset \mathit{\Delta}}{\mathit{\Gamma} \supset \mathit{\Delta}} \; \mathit{cut}$$

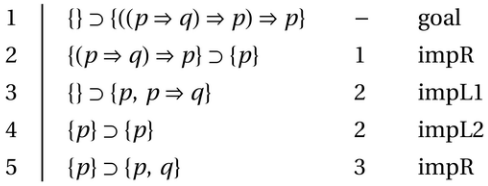
One can show that this system is sound and complete in the sense that exactly all the sequents are derivable for which the formula in $(*)$ are tautologies. There is a hugely important theorem by G. Gentzen from 1934 that shows that the cut rule is superfluous in the sense that one can still prove all tautologies without it, though the proofs may become longer. This so-called cut-elimination result has major implications for proof-search: a closer look at all the other rules reveals that every formula in a premise also occurs as a subformula in the conclusion—which fact greatly limits possible choices for the premises. In the cut rule, on the other hand, the cut formula φ simply disappears and any attempt to construct a proof by reverse engineering becomes substantially harder.

As a first example, let's try to find a proof for modus ponens, $p \wedge (p \Rightarrow q) \Rightarrow q$. Translated into the world of sequents, we need to show that the sequent $\{\} \supset p \wedge (p \Rightarrow q) \Rightarrow q$ is derivable; alternatively we could work with $p, p \Rightarrow q \supset q$. Here is the full argument:

1	$p, q \supset q$	-	axiom
2	$p \supset p, q$	-	axiom
3	$p, p \Rightarrow q \supset q$	1,2	(imp-L)
4	$p \wedge p \Rightarrow q \supset q$	3	(and-L)
5	$\{\} \supset p \wedge (p \Rightarrow q) \Rightarrow q$	4	(imp-R)

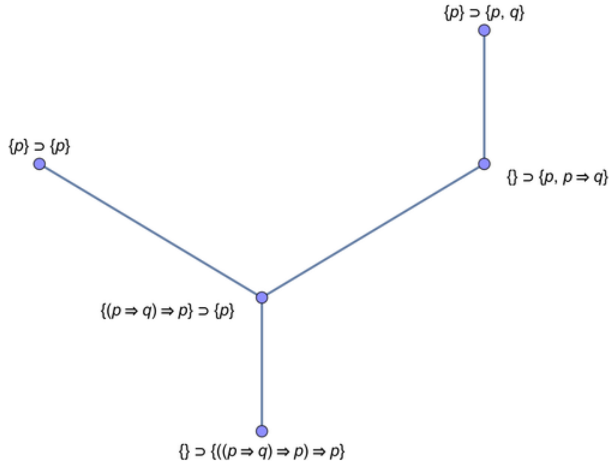
This particular proof is so short that it is fairly easy to find by hand. For more complicated sequents it is preferable to have a little help from an algorithm. We can exploit the civilized behavior of the rules other than cut as follows: we are going to reverse all the rules and work backwards from the goal sequent to the axioms. Each backwards step will remove a connective and, after a while, we will wind up with sequents that contain only propositional variables. But it is trivial to check whether these are axioms or not, we just have to test for a common variable. This observation is the basis of an algorithm due to Hao Wang in 1960. Note that the argument branches when we get to a conjunction on the right, or a disjunction/implication on the left.

Here is an example: Peirce's law, as a sequent $\{\} \supset ((p \Rightarrow q) \Rightarrow p) \Rightarrow p$. The table constructed by Wang's algorithm is a proof run backwards:



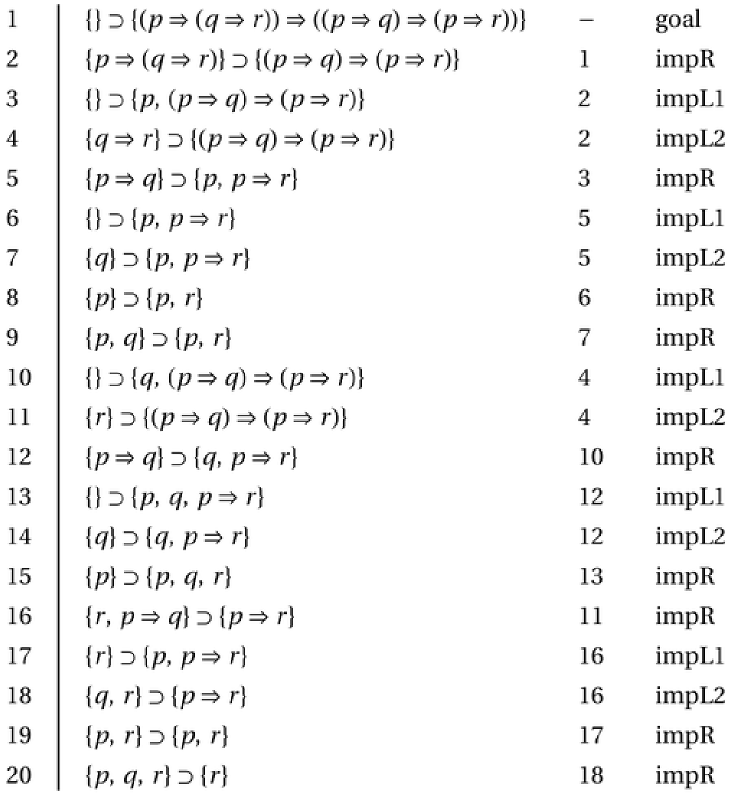
1	$\{\} \supset \{((p \Rightarrow q) \Rightarrow p) \Rightarrow p\}$	–	goal
2	$\{(p \Rightarrow q) \Rightarrow p\} \supset \{p\}$	1	impR
3	$\{\} \supset \{p, p \Rightarrow q\}$	2	impL1
4	$\{p\} \supset \{p\}$	2	impL2
5	$\{p\} \supset \{p, q\}$	3	impR

We could turn the table around, but it is more useful to plot a picture of a proof tree.

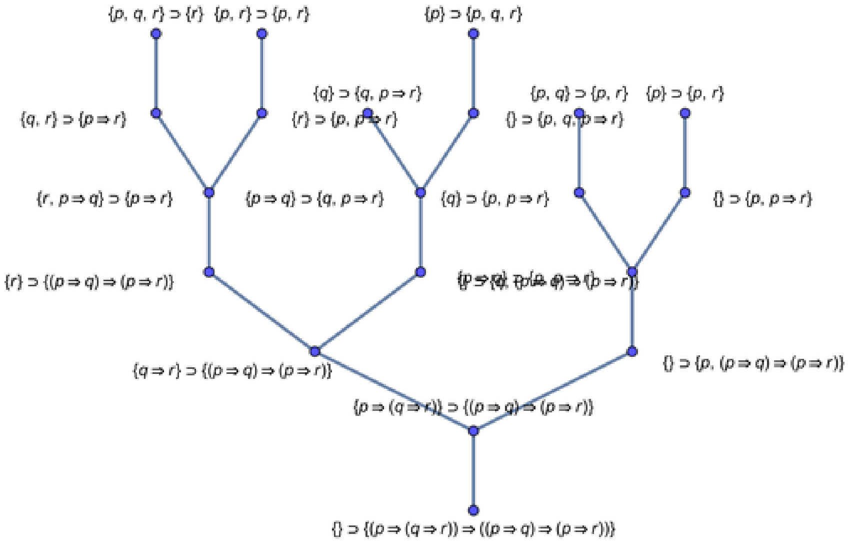


The top leaf on the right is a basic sequent and transformed into the node below by (imp-R). Together with the other leaf on the left we apply (imp-L) and then (imp-R) to get the root.

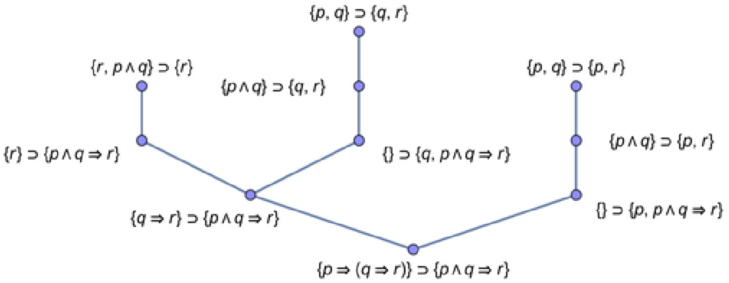
A similar construction for the hypothetical syllogism $(p \Rightarrow (q \Rightarrow r)) \Rightarrow (p \Rightarrow q) \Rightarrow (p \Rightarrow r)$ produces a slightly bigger table and tree.

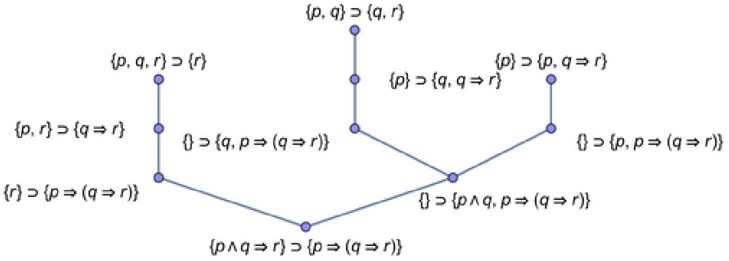


1	$\{\} \supset \{(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))\}$	–	goal
2	$\{p \Rightarrow (q \Rightarrow r)\} \supset \{(p \Rightarrow q) \Rightarrow (p \Rightarrow r)\}$	1	impR
3	$\{\} \supset \{p, (p \Rightarrow q) \Rightarrow (p \Rightarrow r)\}$	2	impL1
4	$\{q \Rightarrow r\} \supset \{(p \Rightarrow q) \Rightarrow (p \Rightarrow r)\}$	2	impL2
5	$\{p \Rightarrow q\} \supset \{p, p \Rightarrow r\}$	3	impR
6	$\{\} \supset \{p, p \Rightarrow r\}$	5	impL1
7	$\{q\} \supset \{p, p \Rightarrow r\}$	5	impL2
8	$\{p\} \supset \{p, r\}$	6	impR
9	$\{p, q\} \supset \{p, r\}$	7	impR
10	$\{\} \supset \{q, (p \Rightarrow q) \Rightarrow (p \Rightarrow r)\}$	4	impL1
11	$\{r\} \supset \{(p \Rightarrow q) \Rightarrow (p \Rightarrow r)\}$	4	impL2
12	$\{p \Rightarrow q\} \supset \{q, p \Rightarrow r\}$	10	impR
13	$\{\} \supset \{p, q, p \Rightarrow r\}$	12	impL1
14	$\{q\} \supset \{q, p \Rightarrow r\}$	12	impL2
15	$\{p\} \supset \{p, q, r\}$	13	impR
16	$\{r, p \Rightarrow q\} \supset \{p \Rightarrow r\}$	11	impR
17	$\{r\} \supset \{p, p \Rightarrow r\}$	16	impL1
18	$\{q, r\} \supset \{p \Rightarrow r\}$	16	impL2
19	$\{p, r\} \supset \{p, r\}$	17	impR
20	$\{p, q, r\} \supset \{r\}$	18	impR



With some amount of effort, a human can still check both table and tree for accuracy. As another example, it is intuitively clear that $p \Rightarrow q \Rightarrow r$ is equivalent with $p \wedge q \Rightarrow r$. The proof trees for the two implications look like so:





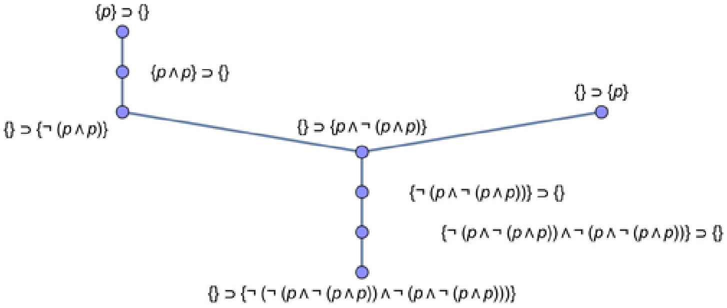
Our sequent calculus does not have rules for other operators such as nand, but we can still work with them by unfolding definitions as in $p \uparrow q = \neg(p \wedge q)$. To show that, say, $(p \uparrow (p \uparrow p) \uparrow (p \uparrow (p \uparrow p)$ is always false we can apply the algorithm to the sequent

$$\neg((\neg(p \wedge \neg(p \wedge p))) \wedge (\neg(p \wedge \neg(p \wedge p)))) \supset \{\}$$

which directly expresses the assertion that the formula on the left is false. Somewhat more interesting is to have the algorithm attempt to show that the formula is a tautology:

$$\{\} \supset \neg((\neg(p \wedge \neg(p \wedge p))) \wedge (\neg(p \wedge \neg(p \wedge p))))$$

The result is the following proof tree:


{p} ⊃ {}
{p ∧ p} ⊃ {}
{} ⊃ {p}
{} ⊃ {¬ (p ∧ p)}
{} ⊃ {p ∧ ¬ (p ∧ p)}
{¬ (p ∧ ¬ (p ∧ p))} ⊃ {}
{¬ (p ∧ ¬ (p ∧ p)) ∧ ¬ (p ∧ ¬ (p ∧ p))} ⊃ {}
{} ⊃ {¬ (¬ (p ∧ ¬ (p ∧ p)) ∧ ¬ (p ∧ ¬ (p ∧ p)))}

Note the two leaves $p \supset \{\}$ and $\{\} \supset p$. The first one translates into $\neg p$ and the second into p, a contradiction. Since our sequent calculus does not have a constant $\perp$, this is the best we can do.

We can also abuse Wang's algorithm to convert a formula in DNF into a formula in CNF. As a cheap instance, consider a formula

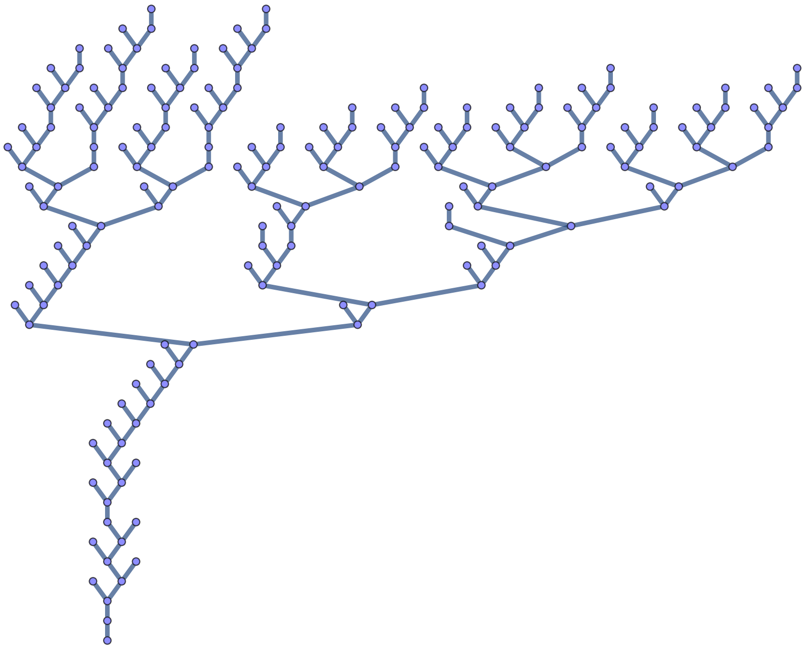
$$(\neg p_1 \wedge q_1) \vee (\neg p_2 \wedge q_2) \vee (\neg p_3 \wedge q_3)$$

The algorithm produces a tree with 8 atomic, non-basic leaves that can be rewritten as disjunctions as follows:

$$\begin{array}{rr}
q_1 \vee q_2 \vee q_3 & \neg p_1 \vee \neg p_2 \vee q_3 \\
\neg p_1 \vee q_2 \vee q_3 & \neg p_1 \vee \neg p_3 \vee q_2 \\
\neg p_2 \vee q_1 \vee q_3 & \neg p_2 \vee \neg p_3 \vee q_1 \\
\neg p_3 \vee q_1 \vee q_2 & \neg p_1 \vee \neg p_2 \vee \neg p_3
\end{array}$$

The conjunction of these 8 terms is equivalent to the original formula.

As a last example, we show the proof tree Wang's algorithm found for the math prof puzzle. Recall the collection Γ of given assertions in section 2. The goal of the puzzle was to show that $\Gamma \Rightarrow (\neg x_5 \vee \neg x_7)$ is a tautology, we can try prove the sequent $\Gamma \supset \neg x_5, \neg x_7$. Γ is a bit larger than other antecedents we have seen, as a result, the table has length 179 and is too large for humans to survey. Attaching labels to the proof tree similarly creates visual chaos, but at least we can plot the plain tree to bring out the overall structure of the argument by just plotting the underlying tree.



If you want to see the whole proof, it is available at Wang Proof. Actually, do take a look, you will become convinced that a proof checker is needed to verify the accuracy of the argument; humans cannot handle this sort of combinatorial complexity.

4 Equational Proofs

As a small step in the direction of formalizing slightly more complicated types of reasoning, let us switch to a somewhat unusual algebraic system. We have some carrier set A, two binary operations that we will call addition and multiplication and a unary operation complement. In analogy to ordinary arithmetic we write $x + y$, $x \cdot y$ and $\bar{x}$. Lastly, there are two constants 0 and 1, but don't think of these as integers. We insist that the following axioms all hold:

$$\begin{array}{cc}
x + (y + z) = (x + y) + z & x \cdot (y \cdot z) = (x \cdot y) \cdot z \\
x + y = y + x & x \cdot y = y \cdot x \\
x + 0 = x & x \cdot 1 = x \\
x + (y \cdot z) = (x + y) \cdot (x + z) & x \cdot (y + z) = (x \cdot y) + (x \cdot z) \\
x + \bar{x} = 1 & x \cdot \bar{x} = 0
\end{array}$$

The first three pairs are the same as if we were to describe addition and multiplication on, say, the integers or the rationals: associativity, commutativity and neutral elements. The next pair asserts distributivity, but beware: addition distributes over multiplication, not just the other way around. This clashes with our intuition about the laws of arithmetic. Lastly, the complement axioms have no counterpart in arithmetic at all, there simply is no corresponding operation. This is good for our purposes: since our intuition fails, we need to rely on formal reasoning—we have to make sure that some claim holds even though we don't see exactly why.

Before we come to proofs, though, there is a burning question: do these axioms make any sense at all? Is there any example for a structure that has these properties? If not, we are wasting our time. Not to worry, it turns out that here are quite a few structures that model our axioms.

Example 1: Let $A = \{\mathsf{ff}, \mathsf{tt}\}$ be the set of truth values "false" and "true." Interpret the functions and constants as follows, using the logical connectives from propositional logic:

$$\begin{aligned} x + y &= x \lor y \\ x \cdot y &= x \land y \\ \bar{x} &= \neg x \\ 0 &= \mathsf{ff} \\ 1 &= \mathsf{tt} \end{aligned}$$

One can check that the all the axioms are satisfied.

Example 2: Let A be the set of divisors of some natural number $n > 1$ and set

$$\begin{aligned} x + y &= \mathrm{lcm}(x, y) \\ x \cdot y &= \mathrm{gcd}(x, y) \\ \bar{x} &= n/x \\ 0 &= 1_{\mathbb{N}} \\ 1 &= n \end{aligned}$$

This is a bit more exotic, but all the axioms are satisfied.

Example 3: Let A be the powerset of some set B and define

$$\begin{aligned} x + y &= x \cup y \\ x \cdot y &= x \cap y \\ \bar{x} &= B - x \\ 0 &= \emptyset \\ 1 &= B \end{aligned}$$

Again, all the axioms are satisfied. Note that we get back the first example when B is a singleton set $\{\bullet\}$: $A = \{\emptyset, \{\bullet\}\}$ and we can think of $\emptyset$ as ff and of $\{\bullet\}$ as tt.

Example 4: We keep the operations from the last example, but we change A to be the set of all subsets of $\mathbb{N}$ that are either finite or coinfinite. So $\mathbb{N}$ is in A, as is $\{\, x \in \mathbb{N} \mid x \leq 10^{100} \,\}$, but the even numbers are not. Not that there are far fewer elements than in the powerset of $\mathbb{N}$.

Definition 4.1 *Any structure that satisfies the given axioms is called a Boolean algebra.*

The name is derived from the first example, George Boole investigated this sort of algebra in the middle of the 19th century to develop a version of propositional logic based on algebra. The axioms above define the basic laws of Boolean algebra. As usual, we would expect that more laws follow from the basic ones. Here as some examples.

Lemma 4.1 *The following equations hold in every Boolean algebra.*

$$\begin{aligned} x + x &= x & x \cdot x &= x \\ x + x \cdot y &= x & x \cdot (x + y) &= x \\ x + 1 &= 1 & x \cdot 0 &= 0 \\ \overline{x + y} &= \bar{x} \cdot \bar{y} & \overline{x \cdot y} &= \bar{x} + \bar{y} \\ \bar{\bar{x}} &= x \end{aligned}$$

One can check that these equations hold in all our examples, but perhaps they could fail in other Boolean algebras? No, because we can derive them all from just the axioms.

Claim 4.1 $x + x = x$ holds in any Boolean algebra.

Proof. To annotate the table, we refer to our axioms as (assc), (com), (iden), (dist) and (cmp).

$$\begin{aligned} x + x &= (x + x) \cdot 1 && \text{(iden)} \\ &= (x + x) \cdot (x + \bar{x}) && \text{(cmp)} \\ &= x + x \cdot \bar{x} && \text{(dist)} \\ &= x + 0 && \text{(cmp)} \\ &= x && \text{(iden)} \end{aligned}$$

The third step uses distributivity of plus over times in the “opposite” direction. □

It is also true that $x \cdot x = x$, but we won’t give a proof, try to find one.

Claim 4.2 $1 + x = 1$ holds in any Boolean algebra.

Proof. Here is a partial argument, with several steps elided.

$$1 + x = (x + \bar{x}) + x = (x + x) + \bar{x} = x + \bar{x} = 1$$

□

Claim 4.3 $x + xy = x$ holds in any Boolean algebra.

Proof. Again, some steps are elided.

$$x + xy = x \cdot (1 + y) = x \cdot 1 = x$$

□

We are not going to worry about technical details here, but note how equational proofs are very similar to proofs in propositional logic. We start with the left-hand side, perform a number of manipulations that are justified by the given axioms and common-sense substitutions, and ultimately wind up with the right-hand side. Checking an argument for correctness is fairly easy, but to find the right sequence of steps can be quite challenging.

Exercises

Exercise 4.1 Find a proof in $\mathcal{H}$ for $\psi \vee \varphi \vdash \neg\neg\psi \vee \varphi$. Conclude from this that $\psi \vdash \neg\neg\psi$.

Exercise 4.2 Find a proof in $\mathcal{H}$ for $\neg\psi_1 \vee \varphi, \neg\psi_2 \vee \varphi \vdash \neg(\psi_1 \vee \psi_2) \vee \varphi$. To see why this makes sense, look at the version using implications: $\psi_1 \Rightarrow \varphi, \psi_2 \Rightarrow \varphi \vdash (\psi_1 \vee \psi_2) \Rightarrow \varphi$.

Exercise 4.3 Go through a few more step of the Math Prof solution from the $\mathcal{H}$ perspective. How would we finally get to a contradiction?

Exercise 4.4 Verify that our four example Boolean algebras all satisfy the axioms as well as the additional properties in lemma 4.1.

Exercise 4.5 Suppose set B in the powerset example is finite. Can you find a natural number n such that the divisibility example produces the same structure?

Exercise 4.6 Expand the arguments in claims 4.2 and 4.3 to a full proof, and annotate the steps by the corresponding axioms (or previously established claims).

Exercise 4.7 Proof some of the other equations in lemma 4.1.

Exercise 4.8 Suppose we wanted to implement a theorem prover based on $\mathcal{H}$. The program would take as input a list Γ of premises and a conclusion ψ where ψ is a tautological consequence of Γ so that $\Gamma \vdash \psi$. It then constructs a proof, say, in the form of an annotated table. How hard would it be to tackle this problem? Which rules are difficult to handle?