Propositional logic

Readings: Sections 1.1 and 1.2 of Huth and Ryan.

In this module, we will consider propositional logic, which will look familiar to you from Math 135 and CS 251. The difference here is that we first define a formal proof system and practice its use before talking about a semantic interpretation (which will also be familiar) and showing that these two notions coincide.

Declarative sentences (1.1)

A **proposition** or **declarative sentence** is one that can, in principle, be argued as being true or false.

Examples: “My car is green” or “Qiang was born in Canada”.

Many sentences are not declarative, such as “Help!”, “What time is it?”, or “Get me something to eat.”

The declarative sentences above are **atomic**; they cannot be decomposed further. A sentence like “My car is green AND you do not have a car” is a compound sentence or **compositional sentence**.

To clarify the manipulations we perform in logical proofs, we will represent declarative sentences symbolically by **atoms** such as p, q, r. (We avoid t, f, T, F for reasons which will become evident.)

Compositional sentences will be represented by **formulas**, which combine atoms with **connectives**. Formulas are intended to symbolically represent statements in the type of mathematical or logical reasoning we have done in the past.

Our standard set of connectives will be $\neg$, $\wedge$, $\vee$, and $\rightarrow$. (In Math 135, you also used $\leftrightarrow$, which we will not use.) Soon, we will describe the set of formulas as a formal language; for the time being, we use an informal description.

The set of connectives is due to the British mathematician George Boole, who described an algebra using them (now called Boolean algebra) in 1854.

We will introduce the connectives in an intuitive fashion, by describing their effect on declarative sentences. In doing so, we anticipate the semantics which we will use to decide if a sentence is true or false.

However, it’s important to keep in mind that our proof system is not concerned with true or false; it is concerned with what constitutes a legal proof. Each of the rules makes intuitive sense, and this is not surprising in light of our goal to show that provable equals true. But we maintain a distinction between semantics and syntax at this point.

The connectives, by example

Suppose we have the following statements:

p: Ling passed CS 245.

q: Ling fulfilled her breadth requirements.

r: Ling earned her B.CS degree.

The first connective, $\neg$ (pronounced "not"), intuitively expresses negation. $\neg p$ means "Ling did not pass CS 245." This is a **unary** connective, in that it applies to only one formula. The rest of the connectives are **binary**.

The connective $\vee$ (pronounced "or") intuitively expresses disjunction, or the sense that at least one of the two formulas it connects is true. $p \vee q$ means "Ling passed CS 245, or fulfilled her breadth requirements."

Note that the English word "or" sometimes has the sense that only one of the two things it connects can be true, not both. But our logical connective $\vee$ will permit both to be true, so it acts more like the English construct "and/or", as in "Ling passed CS 245 and/or fulfilled her breadth requirements."

There are phrases in English which give this sense without using the English word "or". Translation of an English sentence or paragraph into formal logic is a difficult art.

The connective $\wedge$ (pronounced "and") intuitively expresses conjunction, or the sense that both of the formulas it connects are true. $p \wedge q$ means "Ling passed CS 245 and fulfilled her breadth requirements."

As with $\vee$, there are many phrases in English expressing conjunction. The sentence "Ali passed CS 245, although he failed the midterm" is of the form $p \wedge s$, where s represents "Ali failed the CS 245 midterm".

The final connective, $\rightarrow$ (pronounced "implies") intuitively expresses implication. $r \rightarrow (p \wedge q)$ means "If Ling earned her B.CS degree, then she passed CS 245 and fulfilled her breadth requirements."

Binding priorities

In $r \rightarrow (p \wedge q)$, we used parentheses to make it clear what each connective was connecting.

In algebra, we have conventions to avoid excessive use of parentheses, which are often implemented in programming languages (e.g. Java, C++) as well. We understand $3 + 4 \times 5$ to mean $3 + (4 \times 5)$.

We could say that $\times$ **binds** more tightly than $+$, or that $\times$ has precedence over $+$.

In a similar fashion, we say that $\neg$ binds more tightly than $\vee$ and $\wedge$, which bind more tightly than $\rightarrow$.

These rules allow us to simplify $(\neg p) \rightarrow (q \wedge r)$ to $\neg p \rightarrow q \wedge r$.

The book declares $\rightarrow$ to be right-associative; that is, $p \rightarrow q \rightarrow r$ means $p \rightarrow (q \rightarrow r)$, though it does not use this convention much.

When we use these rules, we should understand that our simplified formulas really represent properly parenthesized formulas (which is what our formal definitions will define).

Other texts might declare further conventions which we will not use, such as $\wedge$ binding more tightly than $\vee$.

Natural deduction (1.2)

A proof system is a mathematical formalization of a notion of proof. There are many proof systems; we will study one called natural deduction, invented by Gerhard Gentzen in the 1930's. This system nicely captures many of the aspects of mathematical proof we're familiar with, as well as having desirable technical qualities beyond the scope of this course.

Proof systems work on a syntactic level; they can be viewed as mechanical manipulations of formulas. Of course, we guide these manipulations towards a desired result.

As an example, let's go back to our earlier set of statements.

p: Ling passed CS 245.

q: Ling fulfilled her breadth requirements.

r: Ling earned her B.CS degree.

Consider the following argument: "If Ling earned her B.CS degree, she passed CS 245 and fulfilled her breadth requirements. Ling did not fulfil her breadth requirements. Therefore, Ling did not earn her B.CS degree."

The three sentences formalize as $r \rightarrow (p \wedge q)$, $\neg q$, and $\neg r$, respectively. But the last one is derived in some fashion from the first two.

In a typical situation, we have a set of formulas $\phi_1, \phi_2, \ldots, \phi_n$, and we wish to apply proof rules to these to derive new formulas, among which is our desired conclusion ψ. We summarize this as:

$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$

This is a **sequent**. The sequent for our example is:

$r \rightarrow (p \wedge q), \neg q \vdash \neg r$

We read the symbol $\vdash$ as "yields" or "proves". A sequent has zero or more formulas on the left of $\vdash$, and one formula on the right. The order of the formulas on the left doesn't matter.

A sequent such as $r \to (p \land q), \neg q \vdash \neg r$ is **valid** if we can find a proof for it in our proof system.

In what follows, we introduce the rules of the proof system of natural deduction one by one, with some intuition as to why they might be considered rules, and examples of their use. Along the way, we will accumulate notation to be used in the proof system.

The rules for conjunction

There is one rule to introduce $\land$ and two rules to eliminate it.

And-introduction ($\land$i) says that if we have ϕ and ψ as formulas, we can add $\phi \land \psi$. We write this:

$$\frac{\phi \quad \psi}{\phi \land \psi} \land \mathsf{i}$$

You can view this as a "before and after" view, with an abbreviated name of the rule on the side. Here are the rules for and-elimination:

$$\frac{\phi \land \psi}{\phi} \land \mathsf{e}_1 \qquad \frac{\phi \land \psi}{\psi} \land \mathsf{e}_2$$

These rules may seem silly, but remember that we are trying to create a set of rules that is comprehensive. A proof, if made up of clear applications of well-defined rules, becomes something that can be checked mechanically.

As we discussed in the introductory module, this was part of the impetus for the formalization of logic, and the idea of a process or procedure for discovering proofs drove part of the early development of computer science.

Our rules should therefore not lead to proofs representing English arguments we consider incorrect. We will use this as a "reality check" on the rules we do introduce.

But we also need to keep the rules separate from their eventual interpretation (semantics). It is for this reason that the text delays the semantic interpretation (except for appeals to intuition) until after we gain a lot of practice in formal manipulations using the rules.

Another thing to note is that in natural deduction, these and all subsequent rules apply to whole formulas. They cannot be used to selectively rewrite subformulas. For instance, we cannot rewrite $(p \land q)$ as p if it appears buried within a larger formula.

A proof

This proof shows the validity of the sequent $(p \land q) \land r \vdash q \land r$.

1	$(p \land q) \land r$	premise
2	$(p \land q)$	$\land \mathrm{e}_1 \quad 1$
3	q	$\land \mathrm{e}_2 \quad 2$
4	r	$\land \mathrm{e}_2 \quad 1$
5	$q \land r$	$\land \mathrm{i} \quad 3, 4$

We have numbered each line on the left, and labelled the rules used on the right, with the corresponding line numbers. We also label as a **premise** anything that is on the left-hand side of the sequent. This is how we will be doing proofs.

This notation is a flattened version of the proof using the rules in the form we gave them, because that more resembles a tree.

$$\dfrac{\dfrac{\dfrac{(p \land q) \land r}{p \land q}\,\land \mathrm{e}_1}{q}\,\land \mathrm{e}_2 \qquad \dfrac{(p \land q) \land r}{r}\,\land \mathrm{e}_2}{q \land r}\,\land \mathrm{i}$$

We can reconstruct such a tree from our version. Since the flattened version is much more convenient, we will not be using tree proofs.

In general, our proofs may not be unique. That is, there may be many different ways to demonstrate the validity of a particular sequent.

Note that although we applied our rules to atoms in the previous proof, the rules are stated in terms of general formulas. In fact, because the rules talk about the "top-level" structure of the formulas to which they are applied, we can get other proofs from this proof almost for free.

We can take our previous proof, mechanically substitute any formulas ϕ, ψ, and χ for p, q, and r, and get a new valid proof.

Thus any proof we write can be used as a template (via consistent substitution of formulas for atoms) for an unbounded number of other proofs.

Let's do this by substituting $(x \land y)$ for p, $(x \lor y)$ for q, and $(\neg x)$ for r, yielding the sequent
$((x \land y) \land (x \lor y)) \land (\neg x)) \vdash (x \lor y) \land (\neg x)$.

1	$((x \land y) \land (x \lor y)) \land (\neg x)$	premise
2	$((x \land y) \land (x \lor y))$	$\land \mathrm{e}_1 \quad 1$
3	$(x \lor y)$	$\land \mathrm{e}_2 \quad 2$
4	$(\neg x)$	$\land \mathrm{e}_2 \quad 1$
5	$(x \lor y) \land (\neg x)$	$\land \mathrm{i} \quad 3, 4$

This may not be useful, but it is correct.

Rules for double negation

The rules for double negation are also fairly straightforward.

$$\frac{\neg\neg\phi}{\phi}\ \neg\neg\mathrm{e} \qquad \frac{\phi}{\neg\neg\phi}\ \neg\neg\mathrm{i}$$

We are maintaining the naming convention that “e” refers to elimination and “i” to introduction.

A double negation in English looks like “It is not true that Ling did not pass CS 245.”

It turns out that the $\neg\neg$i rule can be derived from other rules; the textbook demonstrates this, and then moves it into the category of derived rules.

The rule for eliminating implication

$$\frac{\phi \qquad \phi \rightarrow \psi}{\psi}\ \rightarrow\mathrm{e}$$

This is a famous rule, called *modus ponens* (Latin for “mode that affirms”). In classical deduction, this is the only rule used. As you can imagine, this makes proofs longer, and less related to mathematical proofs we know and love.

Modus ponens captures the argument in the following paragraph.

“Ling earned her B.CS degree. If Ling earned her B.CS degree, Ling passed CS 245. Therefore, Ling passed CS 245.”

In this example, p can stand for “Ling earned her B.CS degree”, q for “Ling passed CS 245”, and the sequent we have shown valid with one application of $\rightarrow$e is $p, p \rightarrow q \vdash q$.

In this simple application of $\rightarrow$e, the formulas ϕ and ψ are atomic. But in general, they may be compound formulas as well.

$$\frac{(p \wedge q) \quad (p \wedge q) \rightarrow (r \vee s)}{(r \vee s)}\ \rightarrow\mathrm{e}$$

The derived rule *modus tollens*

At this point the book cheats a bit and introduces a derived rule. It is a form of implication elimination, which is why it is introduced at this point, but we don’t have enough rules to show its derivation yet. Let’s take it on faith and use it, with the promise that it will be justified later.

$$\frac{\phi \rightarrow \psi \quad \neg\psi}{\neg\phi}\ \mathrm{MT}$$

Modus tollens (Latin for “mode that denies”) captures the argument “If Ling earned her B.CS degree, then Ling passed CS 245. Ling did not pass CS 245. Therefore, Ling did not earn her B.CS degree.”

The validity of the English argument we just made suggests that the following English argument is also valid: “If Ling earned her B.CS degree, then Ling passed CS 245. Therefore, if Ling did not pass CS 245, Ling did not earn her B.CS degree.”

As a sequent, this has the form $p \to q \vdash \neg q \to \neg p$. Recall that in Math 135, $\neg q \to \neg p$ was defined as the “contrapositive” of $p \to q$.

We haven’t enough rules yet to show this sequent valid. We need a rule that introduces implication. This requires a new feature in the way we represent proofs.

A rule introducing implication

We want to prove $p \to q \vdash \neg q \to \neg p$. If we could temporarily make $\neg q$ a premise, we could apply *modus tollens* to prove $\neg p$, and then somehow conclude that $\neg q \to \neg p$. We want something like

this:

1	$p \to q$	premise
2	$\neg q$	assumption
3	$\neg p$	MT 1,2
4	$\neg q \to \neg p$	$\to$i 2–3

What did we do? We introduced a proof box, which was a way of marking the fact that we had made an assumption $\neg q$. This kept the assumption from “leaking” into the rest of the proof. But the premises (and presumably any derived formulas) were valid inside the proof box.

We used the assumption and one premise ($p \to q$) to come to a conclusion $\neg p$ (using MT). Then we introduced an implication connecting the assumption $\neg q$ and the conclusion $\neg p$, and labelled it as having been derived using the rule $\to$i.

We need to describe this rule more generally.

Here is the general form of the rule $\to$i.

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \psi\end{array}}}{\phi \to \psi}\ \to\text{i}$$

The first line in the box is an arbitrary formula ϕ of our choice. In the box, we are allowed to use premises, plus any derived formula that is on a previous line, with the exception of those introduced within boxes that have since been closed. This allows us to nest boxes (and their corresponding assumptions).

This is reminiscent of the scope of local definitions in a programming language. We will see this notion again later in the course.

Here's a proof of $p \vdash (p \to q) \to q$.

1	$(p \to q)$	assumption
2	p	premise
3	q	$\to$e 1,2
4	$(p \to q) \to q$	$\to$i 1–3

Note that here, as in our first example of $\to$i, we annotate its use with a range of lines.

The structure of this proof is more or less forced on us by the rules we know. But other proofs do require some creativity to discover.

We can express the transitivity of implication as $p \to q, q \to r \vdash p \to r$.

1	$p \to q$	premise
2	$q \to r$	premise
3	p	assumption
4	q	$\to$e 1,3
5	r	$\to$e 2,4
6	$p \to r$	$\to$i 3–5

By replicating this proof with the appropriate substitutions, we can show that for any formulas ϕ, ψ, χ, the sequent $\phi \to \psi, \psi \to \chi \vdash \phi \to \chi$ is valid.

The rules for disjunction

Introducing and eliminating conjunction was fairly straightforward. Introducing disjunction is also straightforward.

$$\frac{\phi}{\phi \lor \psi}\lor\mathrm{i}_1 \qquad \frac{\psi}{\phi \lor \psi}\lor\mathrm{i}_2$$

Eliminating disjunction is more complicated. Intuitively, if we have a formula of the form $\phi \lor \psi$, we can't just conclude ϕ or ψ, because we don't know which of them is true. We need a way to get from this to some new conclusion χ.

If we know that $\phi \lor \psi$ is true, we know that at least one of the two is true. So a reasonable rule for concluding χ would require us to be able to prove χ from either one of ϕ or ψ. Since we can't be sure that either one is true, we must assume them in order to prove χ.

$$\frac{\phi \lor \psi \quad \boxed{\begin{array}{c}\phi\\ \vdots\\ \chi\end{array}} \quad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\lor\mathrm{e}$$

The commutativity of $\lor$ is something we take for granted from English and experience, but the sequent $p \lor q \vdash q \lor p$ does require a proof.

1	$p \vee q$	premise
2	p	assumption
3	$q \vee p$	$\vee$i$_2$ 2
4	q	assumption
5	$q \vee p$	$\vee$i$_1$ 4
6	$q \vee p$	$\vee$e 1, 2–3,4–5

The commutativity of $\wedge$ is simpler (exercise).

Theorems

Consider the following proof of $p \vdash q \rightarrow p$.

1	q	assumption
2	p	premise
3	$q \rightarrow p$	$\rightarrow$i 1–2

We could "move p to the right" by wrapping this whole proof in a proof box that starts by assuming p and ends by proving $p \rightarrow (q \rightarrow p)$.

In other words, we can show validity of the sequent $\vdash p \rightarrow (q \rightarrow p)$.

1	p	assumption
2	q	assumption
3	p	copy of 1
4	$q \rightarrow p$	$\rightarrow$i 2–3
5	$p \rightarrow (q \rightarrow p)$	$\rightarrow$i 1–4

We need a "copy rule", which allows us to do what we did in line 3. It lets us copy any assumption whose proof box is still open. Without this, we'd have to complicate our description of the $\rightarrow$i rule.

We call a formula ϕ for which the sequent $\vdash \phi$ is valid a **theorem** within our proof system.

We are still going to use the word "theorem" for things we prove, using "Math 135"-style proofs, about our proof systems. Please make sure you always understand in which sense the word is being used.

By iterating the process we just used, we can convert any proof of validity of a sequent $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ into a proof of validity of $\vdash \phi_1 \rightarrow (\phi_2 \rightarrow \ldots (\phi_n \rightarrow \psi) \ldots)$. In fact, we can put the formulas ϕ_i into such a theorem in any order.

The rules for negation

In order to introduce or eliminate negation, we have to talk about contradictions. A contradiction is any formula of the form $\phi \wedge \neg\phi$ or $\neg\phi \wedge \phi$. We introduce the symbol $\perp$ (pronounced "bottom") into our proofs to represent a contradiction. This naturally leads to the not-elimination or $\neg$e rule:

$$\frac{\phi \quad \neg\phi}{\perp}\ \neg\mathrm{e}$$

We could also call this rule bottom-introduction, since it plays that role. But this is not commonly done.

The rule for bottom elimination may be a little surprising.

$$\frac{\perp}{\phi}\ \perp\mathrm{e}$$

This says that with $\perp$ as a premise, one can conclude anything. In other words, $\perp \rightarrow \phi$ is a theorem for all ϕ.

Is a statement like "If 2+2=5, then I am king of the universe" true or false? It turns out that we need this rule in order to get our notion of proof to coincide with the still-to-be-introduced notion of semantic interpretation of formulas.

Introducing negation is our last rule that requires the use of proof boxes. If we make an assumption ϕ and end up with a contradiction, then ϕ intuitively should not be true, so we must be able to conclude that $\neg\phi$ is true.

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \perp\end{array}}}{\neg\phi}\ \neg\mathrm{i}$$

We motivated formal propositional logic with the following example: "If Ling earned her B.CS degree, she passed CS 245 and fulfilled her breadth requirements. Ling did not fulfil her breadth requirements. Therefore, Ling did not earn her B.CS degree." We can now show the corresponding sequent to be valid.

Here is the proof of validity of $r \rightarrow (p \wedge q), \neg q \vdash \neg r$.

1	r	assumption
2	$r \rightarrow (p \wedge q)$	premise
3	$p \wedge q$	$\rightarrow$e 1–2
4	q	$\wedge\mathrm{e}_2$ 3
5	$\neg q$	premise
6	$\perp$	$\neg$e 4,5
7	$\neg r$	$\neg$i 1–7

Derived rules

We can now prove *modus tollens*, or $\phi \to \psi, \neg\psi \vdash \neg\phi$. But there is a slight problem. This is not really a sequent. It is a template or framework for creating sequents. If we substitute any specific formulas for ϕ and ψ, we will get a sequent.

The textbook does not discuss this, and it is a source of possible confusion. We will call this a **sequent schema**.

In order to prove it, we will create a **proof schema**, which has the same parameters, ϕ and ψ. Any substitution of specific formulas in the proof schema yields a valid proof.

1	$\phi \to \psi$	premise
2	$\neg\psi$	premise
3	ϕ	assumption
4	ψ	$\to$e 1,3
5	$\bot$	$\neg$e 2,4
6	$\neg\phi$	$\neg$i 3–5

Note that everywhere we used MT, we could fill in these lines instead, with the appropriate substitutions. Using MT lets us shorten proofs. We will keep it as a **derived rule** (not a primitive rule).

There are many more useful derived rules; we will present two. *Reductio ab absurdum* (Latin for "reduction to the absurd") is sometimes called proof by contradiction, and that is how we will abbreviate it (PBC). As a rule, it looks like this:

$$\dfrac{\boxed{\begin{array}{c}\neg\phi \\ \vdots \\ \bot\end{array}}}{\phi}\ \text{PBC}$$

This looks like not-introduction turned upside down, so it is not surprising that we can use not-introduction and double-negation-elimination to prove this rule.

The textbook writes PBC as $(\neg\phi \to \bot) \vdash \phi$, and provides a five-line proof schema. However, there is a problem with this: $\bot$ is not part of our language of formulas. We can make a rule that in sequent schema, we can substitute any formula of the form $\psi \wedge \neg\psi$ for $\bot$, but doing this to the five-line proof schema in the book does not yield a valid proof.

Instead, we will reason about why PBC should be valid, by viewing it as shorthand for a longer sequence. Any application of PBC starts with a proof box with $\neg\phi$ at the top and $\bot$ at the bottom, followed by ϕ. If we substitute an application of $\neg$i, we can put $\neg\neg\phi$ after the box. Then it is simple, using $\neg\neg$e, to conclude ϕ.

In other words, one application of PBC can be replaced by one application of $\neg$i and one application of $\neg\neg$e.

Our last derived rule is the law of the excluded middle, or LEM. (For once, we will use this instead of the Latin “tertium non datur”, meaning “a third [thing] is not given”). This simply states that $\phi \vee \neg\phi$ is a theorem for any ϕ. Intuitively, it means that either ϕ is true or $\neg\phi$ is true.

This is a useful rule because its proof is longer than the others, and so it saves much space when it is used. On the other hand, it is less obvious how to use it than some of the other rules we have. It can be a source of disjunctions that can be used in or-elimination (see example 1.24 in the textbook). It plays an important role in a mathematical proof in the next module.

Once again, we use a proof schema.

1	$\neg(\phi \vee \neg\phi)$	assumption
2	ϕ	assumption
3	$\phi \vee \neg\phi$	$\vee i_1$ 2
4	$\bot$	$\neg$e 1,3
5	$\neg\phi$	$\neg$i 2–4
6	$\phi \vee \neg\phi$	$\vee i_2$ 5
7	$\bot$	$\neg$e 1,6
8	$\neg\neg(\phi \vee \neg\phi)$	$\neg$i 1–7
9	$\phi \vee \neg\phi$	$\neg\neg$e 8

Provable equivalence

Two formulas ϕ and ψ are provably equivalent if $\phi \vdash \psi$ and $\psi \vdash \phi$ are valid. We sometimes write this $\phi \dashv\vdash \psi$.

As an example, we proved $p \rightarrow q \vdash \neg q \rightarrow \neg p$ earlier. If we prove $\neg q \rightarrow \neg p \vdash p \rightarrow q$, it will show that the formulas $p \rightarrow q$ and $\neg q \rightarrow \neg p$ are provably equivalent.

1	$\neg q \rightarrow \neg p$	premise
2	p	assumption
3	$\neg\neg p$	$\neg\neg$i 2
4	$\neg\neg q$	MT 1,3
5	q	$\neg\neg$e 4
6	$p \rightarrow q$	$\rightarrow$i 2–5

Just as with mathematical proofs, there is a certain art to constructing proofs in natural deduction. There is no simple recipe or algorithm to create a proof of a sequent. You should study the more complicated examples in the textbook, which also gives some guidelines as to what to try. Do as many proofs as you can manage.

It is traditional at this point to offer practice in converting examples in English to formulas. But these would be of the type we have already seen, involving statements about CS 245 students and their mothers. We will instead defer practice in formalization until we have developed rich enough languages to express what we need in our proofs, our specifications, and in reasoning about our programs.

Intuitionistic logic

There is a philosophy of mathematics called constructivism that rejects proof by contradiction and existence proofs. In effect, to prove the existence of some object or objects (such as two prime numbers adding up to an even natural number greater than 2), one has to actually construct the objects.

The formal logic associated with this philosophy is called intuitionistic. It removes PBC, LEM, and $\neg\neg$e.

You may think that this is rather arbitrary, but intuitionistic logic has applications in computer science, much of which is inherently constructive.

As an example, it turns out that there is a very nice correspondence (the Curry-Howard isomorphism) between proofs of theorems in a form of intuitionistic logic and programs in a simple functional language (the simply-typed lambda calculus, which resembles a stripped-down version of Scheme).

In a deep sense, a proof is a program, and a program is a proof.

Thus extending the capabilities of the functional language may correspond to defining a different form of logic, and vice-versa.

These notions are explored further in CS 442 (Programming Languages), and we will not consider them here. Much later in this course, we will apply logic to reason about the correctness and termination of programs, which is a different topic.

What's coming next?

We defined the system of natural deduction fairly precisely, except for a certain vagueness about what exactly a formula was. We must fix that omission if we are to define a semantic interpretation of formulas, and prove (in a mathematical sense) that this notion coincides with our notion of proof by natural deduction.

Our language of propositional formulas is not adequate to capture the reasoning in the examples we initially considered, such as "Every even number is the sum of two odd numbers whose difference is at most 2." We need to extend it to capture the notion that the word "every" makes this a compositional statement. That will complicate both our proof system and our semantics.

Goals of this module

At the end of each lecture module (which may take up several days of lecture) we will review its goals.

You may not have achieved these goals just by listening to the lecture. Some will require reading in the text, doing assignment questions, and working through examples of your own choosing.

The list of goals is a guide to the level of understanding we expect from you, eventually.

We introduced a fair amount of terminology (formulas, sequents, conjunction, disjunction, etc.) with which you need to be comfortable.

You should have all of the rules of natural deduction in Figure 1.2 in the book (twelve rules, plus the copy rule, plus four derived rules) committed to memory. You should also understand how these rules correspond to elements of mathematical proof with which you are familiar from other math and CS courses.

You should be able to use the system of natural deduction to demonstrate the validity of sequents. In addition to assignment questions, there are many exercises in the book for practice.