Week 7

Proof theory

Katie Casamento
Fall 2018

Portland State University

Motivation

What is a “proof”?

When we say something is “mathematically proven”, what does that mean?

How can we apply this knowledge to write proofs about program behavior?

Classical logic

In the first week of class, we worked with *classical* logic in the `Prop` language.

- Also known as *Aristotelian* or *Boolean* logic
- CS251
- The meaning of a proposition is given by a *truth table*
- Every cell in a truth table is a *Boolean truth value*: `T` or `F`
- Propositions are constructed with *connectives* like $\wedge$ and $\neg$

Classical logic syntax

In general, a system of logic is a formal language - it has a syntax and a semantics.

A *proposition* is an expression that describes something we might want to prove.

The **syntax** of classical logic tells us how to write propositions.

$$p ::= a \mid \mathtt{T} \mid \mathtt{F} \mid p_1 \vee p_2 \mid p_1 \wedge p_2 \mid p_1 \rightarrow p_2 \mid \neg p$$

(Where a is an element of a set of *names*, usually strings.)

The relative precedences of the operators are $\neg \; > \; \wedge \; > \; \vee \; > \rightarrow$.

$\rightarrow$ is right-associative, and the associativity of $\wedge$ and $\vee$ is usually irrelevant.

Boolean semantics

A **semantics** (or *interpretation*) for a logic tells us how to write proofs of propositions.

An *axiom* is one of the rules that defines an interpretation of logic.

In the *Boolean* interpretation of classical logic, the axioms are given as truth tables.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	$\neg p$
T	F
F	T

These tables are the **definitions** of these connectives in the Boolean interpretation.

Boolean semantics

Technically, these definitions are *axiom schemas*: we can substitute any propositions for p and q and get a corresponding truth table.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

$a \wedge b$	$c \wedge b$	$(a \wedge b) \vee (c \wedge b)$
T	T	T
T	F	F
F	T	F
F	F	F

Classical logic semantics

We can apply Boolean semantics to figure out the meaning of any proposition as a truth table.

a	b	c	$a \wedge b$	$c \wedge b$	$(a \wedge b) \vee (c \wedge b)$
T	T	T	T	T	T
T	T	F	T	F	T
T	F	T	F	F	F
T	F	F	F	F	F
F	T	T	F	T	T
F	T	F	F	F	F
F	F	T	F	F	F
F	F	F	F	F	F

Classical implication

The $\rightarrow$ operator can be defined in terms of $\neg$ and $\vee$.

$$(p \rightarrow q) := (\neg p \vee q)$$

p	q	$\neg p$	$p \rightarrow q$
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

Notice a special case of this rule: if $p \rightarrow \mathtt{F}$ is true, then we know p must be false.

In fact, we could have defined $\neg$ in terms of $\rightarrow$ and F.

$$(\neg p) := (p \rightarrow \mathtt{F})$$

(This will come back up later.)

Basis

A set of axioms is a *basis* for Boolean logic if we can get the rest of Boolean logic from just those axioms.

Every basis for Boolean logic gives us the same truth tables for every proposition, so which basis we choose is usually unimportant.

We can actually get away with just a single axiom as a Boolean basis.

The $\overline{\wedge}$ operator is pronounced “nand”, short for “not and”.

p	q	$p \overline{\wedge} q$
T	T	F
T	F	T
F	T	T
F	F	T

$$(\neg p) := (p \overline{\wedge} \mathtt{T}) \qquad (p \wedge q) := \neg(p \overline{\wedge} q)$$

$$(p \vee q) := \neg(\neg p \wedge \neg q) \qquad (p \to q) := (\neg p \vee q)$$

Tautologies

A Boolean *tautology* is a proposition whose column in a truth table is all `T`.

a	$\neg a$	$a \vee a$	$a \to (a \vee a)$
T	F	T	T
T	F	T	T
F	T	F	T
F	T	F	T

This truth table can be thought of as a **proof** that $a \to (a \vee a)$ is a tautology.

Philosophical foundations

So far we've talked about the *formal semantics* of classical logic: we've been treating it as a game we play with symbols, according to rules that we defined in symbols.

Does it have any more significance than that?

A *philosophical foundation* for a system of logic is roughly an argument that the propositions and proofs of that system are meaningful in the "real world" in some way.

Informal propositions

The word *"informal"* is often used to describe a piece of writing in English (or another natural language) about a formal subject.

An *informal proposition* is a sentence that we can ask about the truth of.

Different people may understand the meaning of the same sentence in different ways, so not everybody is in agreement about the truth of every informal proposition.

- Nearly everybody agrees that "One plus one is two." is true.
- Probably nobody believes that "People are always larger than machines." is true.[1]
- Adults and children sometimes disagree about the truth of "Santa Claus is real."

[1] Counterpoint: B. Garfinkel, M. Brundage, D. Filan, C. Flynn, J. Luketina, M. Page, A. Sandberg, A. Snyder-Beattie, M. Tegmark. "On the Impossibility of Supersized Machines." April 1, 2017.

Informal proofs

An *informal proof* is a convincing argument about the truth of a proposition.

Naturally, different people sometimes find different arguments convincing.

(https://xkcd.com/982)

Most mathematical proofs use a combination of formal and informal methods to try to convince the reader that a fully formal proof is **theoretically** possible to write, since fully formal proofs are often too large and complex to **actually** write (or read).

Philosophical foundations

We can give a philosophical foundation for a formal system of logic roughly by giving

- a procedure for translating between formal propositions and informal propositions,
- a procedure for translating between formal proofs and informal proofs,
- and an argument that these translations preserve "truth".

Think about your *intuitive* ideas about what it means for something to be "true" or "false" - the things that **feel** valid about the meaning of those words based on your experiences, as opposed to our formal definitions of the symbols `T` and `F`.

(Again, your intuition might differ from someone else's!)

- Does the use of `T` correspond to your intuitive understanding of the word "true"?
- Does the use of $\wedge$ correspond to your intuitive understanding of the word "and"?
- Does every tautology of the system correspond to something you intuitively understand as a "universal truth"?
- ...

"or", "not"

Let's look more closely at the definitions of "or" and "not".

Formally:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	$\neg p$
T	F
F	T

Informally:

- "p or q" is true exactly when "p" is true or "q" is true
  - "7 $>$ 5 or it is currently raining outside" is true regardless of the weather
- "p and q" is false exactly when "p" and "q" are both false

- "not p" (or "the opposite of p") is true exactly when "p" is false
- "not p" is false exactly when "p" is true

The law of the excluded middle

These definitions lead to an interesting tautology: the *law of the excluded middle*.

a	$\neg a$	$a \vee \neg a$
T	F	T
F	T	T

Informally, this says that every proposition - every sentence that we can ask about the truth of - is either “true” or “false”.

Does that agree with your intuition?

The law of the excluded middle

We can talk informally about the truth of lots of sentences that aren't easy to see as just "true" or just "false", even if we assume that we're all interpreting the sentence the same way.

- "Drinking coffee is healthy."
- "My favorite band is better than your favorite band."
- "No human will ever live to be a thousand years old."

This doesn't mean classical logic is useless in the real world, though - it just means there are **some** informal propositions that classical logic doesn't apply to.

Does it apply to all **mathematical** propositions?

The Collatz conjecture

Consider a recursive function $f(x)$ over natural numbers, defined as follows.

If $x = 0$,	$f(x) = 0$.
If x is even and nonzero,	$f(x) = f(\frac{x}{2})$.
If x is odd,	$f(x) = f(3x + 1)$.

The *Collatz conjecture* is a mathematical proposition about this function:

> "f terminates on all inputs."

Despite more than 80 years of active research, the Collatz conjecture is still unsolved: nobody has provided a counterexample or a proof that the mathematical community recognizes as valid.

Do you believe this proposition?

> "The Collatz conjecture is true or the Collatz conjecture is false."

Arguments for the excluded middle

Do you believe this proposition?

> "The Collatz conjecture is true or the Collatz conjecture is false."

Some arguments in favor:

- If it's not true and it's not false, then **what else** could its truth value be? It's hard to imagine a satisfying answer to this question, since we expect mathematics to be definite.
- There are tons of solved problems in mathematics, and every solved problem has turned out to be either true or false, so we should **expect from experience** that this one is solvable and will turn out to be either true or false.

Mathematical formalism

In technical contexts, we usually implicitly assume a *formalist* view of mathematics.

- Championed by David Hilbert and friends at the start of the 20th century
- Math is a game we play with symbols
- The relevance of math in the real world is outside the study of math

Formalism doesn't try to tell us why math "works" in the real world[2].

From a purely formalist perspective, the law of the excluded middle is simply true in this system of logic and might not be true in other systems; it doesn't necessarily "mean" anything.

Some other philosophies of mathematics attempt to explain why formal math "works".

- *Psychologism* says that math corresponds to patterns in the way our brains work.
- *Platonism* says that mathematical entities "exist" in some abstract realm.
- ...

[2] A good introduction to why this matters is "The Unreasonable Effectiveness of Mathematics in the Natural Sciences" by Eugene Wigner.

Arguments against the excluded middle

Do you believe this proposition?

“The Collatz conjecture is true or the Collatz conjecture is false.”

Some arguments against:

- We **don’t know** the truth value of the Collatz conjecture. We can’t really speak authoritatively about things we don’t know.
- Something or other about “**incompleteness**”. (We’ll come back to this.)

Mathematical intuitionism

The philosophy of *mathematical intuitionism* says that math is an activity that takes place within our minds, and mathematical truth must be **intuitive**.

This means that either

- everyone must **share some common intuition about mathematics**, or
- mathematics must be **inherently subjective**.

Intuitionism was founded by L. E. J. Brouwer as an alternative to formalism; it's generally thought that he believed the first of these, but intuitionists in general sometimes disagree about them.

(Note that the first claim is specifically about **mathematical** intuition - it doesn't claim that everyone shares the same intuition about **everything**.)

“intuition” vs. “intuitionism”

Some confusing vocabulary:

- *intuition:* the capability to know and reason without a conscious effort to think within a specific framework
- *intuitive:* the adjective form of “intuition”

- *intuitionism:* the philosophy about intuition in mathematics
- *intuitionistic:* the adjective form of “intuitionism”

What does an intuitionistic logic look like?

The set of axioms should probably at least be

- **small**: we should be able to hold all of them in our minds at once
- **simple**: we should be able to explain all of them in everyday language
- **sound**: none of the rules should contradict each other

And, of course, the rules should correspond to our **intuitive** ideas about the meanings of the terms involved.

The BHK interpretation

One famous intuitionistic interpretation of logic is the *BHK interpretation*, named after L. E. J. Brouwer, Arend Heyting, and Andrey Kolmogorov.

The BHK interpretation is a *proof theory*: it gives rules for constructing proofs, and says that the only valid proofs are the ones constructed with those rules.

- Anything is a proof of $\mathtt{T}$ (the trivial proposition).
- Nothing is a proof of $\mathtt{F}$ (the false proposition).
- A proof of $P \wedge Q$ is a proof of P and a proof of Q.
- A proof of $P \vee Q$ is a proof of P or a proof of Q.
- A proof of $P \rightarrow Q$ is a procedure to turn a proof of P into a proof of Q.
- A proof of $\neg P$ is a proof of $P \rightarrow \mathtt{F}$.
- We may assume that there are some *atomic* propositions that we already know the proof rules for.

Note that none of these are defined in terms of **truth**, but rather **provability**.

This interpretation is surprisingly relevant to programming languages, as we'll see later.

The law of the excluded middle, intuitionistically

A proposition is intuitionistically

- **true** if and only if we **know a proof** of it
- **false** if and only if we **know a proof that there cannot exist a proof** of it

Consider again:

"The Collatz conjecture is true or the Collatz conjecture is false."

We definitely don't know a proof or counterproof of the Collatz conjecture.

Intuitionism says that **all we can say** about the truth of the Collatz conjecture is that we don't know it yet.

This means the law of the excluded middle is **not an intuitionistic tautology**!

Classical logic vs. intuitionistic logic

A proposition is intuitionistically

- *true* if and only if we **know a proof** of it
- *false* if and only if we **know a proof that there cannot exist a proof** of it

A proposition is an *intuitionistic tautology* if we can prove it **with no assumptions**.

These are some propositions that are classical but not intuitionistic tautologies.

excluded middle:	$p \vee \neg p$
double negation elimination:	$\neg\neg p \to p$
Peirce's law:	$((p \to q) \to p) \to p$

All three turn out to be equivalent: if you assume one, you can prove the others.

Proof theory

If some tautologies of classical logic are not valid in intuitionistic logic, truth tables won't work as formal intuitionistic proofs.

We'll construct a language of *proof trees* that will serve as intuitionistic proofs.

For each connective, we'll give

- *introduction* rules that say how to create proofs, and
- *elimination* rules that say how to use proofs.

The rule for `T` is simple: we can always make a proof of `T`.

$$\dfrac{}{\texttt{T}}\ \texttt{T-Intro}$$

There is no introduction rule for `F`, since there should be no proof of `F`.

"or", "and"

A proof of $P \wedge Q$ is a proof of P and a proof of Q.

$$\frac{P \qquad\qquad Q}{P \wedge Q}\ \wedge\text{-}\mathtt{Intro}$$

A proof of $P \vee Q$ is a proof of P or a proof of Q.

$$\frac{P}{P \vee Q}\ \vee_L\text{-}\mathtt{Intro} \qquad\qquad \frac{Q}{P \vee Q}\ \vee_R\text{-}\mathtt{Intro}$$

The propositions above the line in a rule are called the *premises* of the rule.

The proposition below the line in a rule is called the *conclusion* of the rule.

Proof tree example

Assume we know some propositions A, B, and C.

Assume we know proofs of A and B: $\dfrac{\vdots}{A} \quad \dfrac{\vdots}{B}$

Remember the introduction rules:

$$\frac{P \qquad\qquad Q}{P \wedge Q}\ \wedge\texttt{-Intro}$$

$$\frac{P}{P \vee Q}\ \vee_L\texttt{-Intro} \qquad\qquad\qquad \frac{Q}{P \vee Q}\ \vee_R\texttt{-Intro}$$

Here's a proof of $(A \vee B) \wedge (C \vee B)$.

$$\frac{\dfrac{\dfrac{\vdots}{A}}{A \vee B}\ \vee_L\texttt{-Intro} \qquad\qquad \dfrac{\dfrac{\vdots}{B}}{C \vee B}\ \vee_R\texttt{-Intro}}{(A \vee B) \wedge (C \vee B)}\ \wedge\texttt{-Intro}$$

Assumptions

What does it mean to "assume we know a proof of A"?

We use a *context* to keep track of what we've assumed in each part of a proof.

We'll represent a context as a **list of propositions** that we've assumed.

If an assumption is in the current context, we can prove it "by assumption".

$$\frac{A \in \Gamma}{\Gamma \vdash A}\ \texttt{Ass}$$

The symbol $\vdash$ is usually pronounced "entails", and has the lowest precedence.

The symbol $\in$ is pronounced "in".

"$A \in \Gamma$" means

"A is an element of the list Γ."

"$\Gamma \vdash A$" means

"Assuming we know proofs of everything in Γ, we can construct a proof of A."

Assumption example

We have to modify our axioms a little bit so that they all keep track of the context.

$$\frac{A \in \Gamma}{\Gamma \vdash A}\,\texttt{Ass} \qquad \frac{}{\Gamma \vdash \texttt{T}}\,\texttt{T-Intro}$$

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}\,\wedge\texttt{-Intro}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}\,\vee_L\texttt{-Intro} \qquad \frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}\,\vee_R\texttt{-Intro}$$

Here's the last example again, this time using a context of assumptions.

$$\dfrac{\dfrac{\dfrac{A \in [A, B]}{[A, B] \vdash A}\,\texttt{Ass}}{[A, B] \vdash A \vee B}\,\vee_L\texttt{-Intro} \qquad \dfrac{\dfrac{B \in [A, B]}{[A, B] \vdash B}\,\texttt{Ass}}{[A, B] \vdash C \vee B}\,\vee_R\texttt{-Intro}}{[A, B] \vdash (A \vee B) \wedge (C \vee B)}\,\wedge\texttt{-Intro}$$

List membership

When it's clear that a statement involving $\in$ is true, we usually don't bother writing a proof for it.

$$\frac{B \in [A, B, C]}{[A, B, C] \vdash B}\ \texttt{Ass}$$

If we did want to give explicit proofs for $\in$, we could use a set of rules like these.

$$\frac{}{A \in A :: \Gamma}\ \in\texttt{-Here} \qquad\qquad \frac{A \in \Gamma}{A \in X :: \Gamma}\ \in\texttt{-There}$$

(These lists are in Haskell-like syntax: $[A, B]$ is the same as $A :: B :: []$.)

Here's a full proof using these rules:

$$\cfrac{\cfrac{\cfrac{}{B \in [B, C]}\ \in\texttt{-Here}}{B \in [A, B, C]}\ \in\texttt{-There}}{[A, B, C] \vdash B}\ \texttt{Ass}$$

Structural rules

We can change a context in a couple ways without affecting the validity of a proof.

Specifically, we can

- **add unused** assumptions (weakening),
- **remove duplicate** assumptions (contraction), and
- **rearrange** our context of assumptions (exchange).

$$\frac{\Gamma \vdash P}{X :: \Gamma \vdash P}\ \texttt{Weaken} \qquad \frac{P :: P :: \Gamma \vdash Q}{P :: \Gamma \vdash Q}\ \texttt{Contract}$$

$$\frac{\Gamma{+}{+}\Delta \vdash P}{\Delta{+}{+}\Gamma \vdash P}\ \texttt{Exchange}$$

($\Gamma{+}{+}\Delta$ is the list Γ appended to the list Δ.)

These rules are *admissible* in this proof theory: they follow from the set of axioms.

(Constructing admissibility proofs is beyond the scope of this lecture.)

Structural rules

We can actually summarize all the structural rules with just one rule:

if every proposition in Γ is also in Δ,

then we can turn a proof that assumes Γ into a proof that assumes Δ.

Formally:

$$\frac{\forall X.\ (X \in \Gamma) \to (X \in \Delta) \qquad \Gamma \vdash P}{\Delta \vdash P}\ \texttt{Embed}$$

A logic where this rule **is** admissible is called a *structural* logic.

A logic where this rule **is not** admissible is called a *substructural* logic.

Substructural logics can be used to reason about a collection of "resources".

In this class, we'll only focus on structural logic.

“implies”

A proof of $P \to Q$ is a procedure to turn a proof of P into a proof of Q.

We can model this with contexts:

A proof of $P \to Q$ is a proof of Q in a context extended with P as an assumption.

$$\frac{P :: \Gamma \vdash Q}{\Gamma \vdash P \to Q}\ \to\text{-}\mathtt{Intro}$$

In other words,

if we know Q is true assuming P is true and assuming everything in Γ is true,

then we know P implies Q assuming only that everything in Γ is true.

Implication example

Axioms:

$$\frac{A \in \Gamma}{\Gamma \vdash A}\ \texttt{Ass} \qquad \frac{}{\Gamma \vdash \texttt{T}}\ \texttt{T-Intro} \qquad \frac{P :: \Gamma \vdash Q}{\Gamma \vdash P \to Q}\ \to\texttt{-Intro}$$

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}\ \wedge\texttt{-Intro} \qquad \frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}\ \vee_L\texttt{-Intro} \qquad \frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}\ \vee_R\texttt{-Intro}$$

Now we can express the example from before as a tautology.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \in [A,B]}{[A,B] \vdash A}\ \texttt{Ass}}{[A,B] \vdash A \vee B}\ \vee_L\texttt{-Intro} \qquad \dfrac{\dfrac{B \in [A,B]}{[A,B] \vdash B}\ \texttt{Ass}}{[A,B] \vdash C \vee B}\ \vee_R\texttt{-Intro}}{[A,B] \vdash (A \vee B) \wedge (C \vee B)}\ \wedge\texttt{-Intro}}{[B] \vdash A \to (A \vee B) \wedge (C \vee B)}\ \to\texttt{-Intro}}{[] \vdash B \to A \to (A \vee B) \wedge (C \vee B)}\ \to\texttt{-Intro}$$

Elimination rules

Introduction rules tell us how to **create** a proof of a proposition involving a connective.

If we know P, and we know Q, then we know $P \land Q$.

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash Q}{\Gamma \vdash P \land Q}\ \land\texttt{-Intro}$$

Elimination rules tell us how to **use** a proof of a proposition involving a connective.

If we know $P \land Q$, and we know R assuming P and Q, then we know R.

$$\frac{\Gamma \vdash P \land Q \qquad P :: Q :: \Gamma \vdash R}{\Gamma \vdash R}\ \land\texttt{-Elim}$$

“and” elimination

We can use $\wedge$-`Elim` to build more convenient ways to use proofs involving $\wedge$.

$$\frac{\Gamma \vdash P \wedge Q \qquad P :: Q :: \Gamma \vdash R}{\Gamma \vdash R}\ \wedge\text{-}\mathtt{Elim}$$

Assume we have a proof of $\Gamma \vdash P \wedge Q$: $\dfrac{\vdots}{\Gamma \vdash P \wedge Q}$

Then we can get from $\Gamma \vdash P \wedge Q$ to $\Gamma \vdash P$ and $\Gamma \vdash Q$, as we'd expect to be able to.

$$\frac{\dfrac{\vdots}{\Gamma \vdash P \wedge Q} \qquad \dfrac{P \in P :: Q :: \Gamma}{P :: Q :: \Gamma \vdash P}\ \mathtt{Ass}}{\Gamma \vdash P}\ \wedge\text{-}\mathtt{Elim}$$

$$\frac{\dfrac{\vdots}{\Gamma \vdash P \wedge Q} \qquad \dfrac{Q \in P :: Q :: \Gamma}{P :: Q :: \Gamma \vdash P}\ \mathtt{Ass}}{\Gamma \vdash Q}\ \wedge\text{-}\mathtt{Elim}$$

"or" elimination

If we know P, or we know Q, then we know $P \vee Q$.

$$\frac{P}{P \vee Q} \vee_L\texttt{-Intro} \qquad\qquad \frac{Q}{P \vee Q} \vee_R\texttt{-Intro}$$

If we know $P \vee Q$, and we know R assuming P, and we know R assuming Q, then we know R.

$$\frac{\Gamma \vdash P \vee Q \qquad P :: \Gamma \vdash R \qquad Q :: \Gamma \vdash R}{\Gamma \vdash R} \vee\texttt{-Elim}$$

“implies” elimination

If we know Q assuming P, then we know $P \to Q$.

$$\frac{P :: \Gamma \vdash Q}{\Gamma \vdash P \to Q} \to\text{-}\mathtt{Intro}$$

If we know $P \to Q$, and we know P, then we know Q.

(This rule is sometimes called *modus ponens* if we're being fancy.)

$$\frac{\Gamma \vdash P \to Q \qquad \Gamma \vdash P}{\Gamma \vdash Q} \to\text{-}\mathtt{Elim}$$

“true” and “false” elimination

From the BHK interpretation:

- Anything is a proof of `T` (the trivial proposition).
- Nothing is a proof of `F` (the false proposition).

There is no `T-Elim`: anything is a proof of `T`, so a knowing proof of `T` tells us nothing.

`F-Elim` is a little tricky.

$$\frac{\Gamma \vdash \texttt{F}}{\Gamma \vdash P}\ \texttt{F-Elim}$$

This is called the *principle of explosion:* if we have a proof of `F`, we’ve already reached a contradiction, so we can have a proof of anything we want at that point.

`F-Elim` is interesting, but it’s less relevant to our study of programming languages than the other rules, so we won’t talk about it much.

Intuitionistic propositional proof theory

These are all of the axioms of the proof theory we're working with, in their final forms.

$$\frac{A \in \Gamma}{\Gamma \vdash A}\ \texttt{Ass} \qquad \frac{}{\Gamma \vdash \mathtt{T}}\ \texttt{T-Intro} \qquad \frac{\Gamma \vdash \mathtt{F}}{\Gamma \vdash P}\ \texttt{F-Elim}$$

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}\ \wedge\texttt{-Intro} \qquad \frac{\Gamma \vdash P \wedge Q \qquad P :: Q :: \Gamma \vdash R}{\Gamma \vdash R}\ \wedge\texttt{-Elim}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}\ \vee_L\texttt{-Intro} \qquad \frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}\ \vee_R\texttt{-Intro}$$

$$\frac{\Gamma \vdash P \vee Q \qquad P :: \Gamma \vdash R \qquad Q :: \Gamma \vdash R}{\Gamma \vdash R}\ \vee\texttt{-Elim}$$

$$\frac{P :: \Gamma \vdash Q}{\Gamma \vdash P \to Q}\ \to\texttt{-Intro} \qquad \frac{\Gamma \vdash P \to Q \qquad \Gamma \vdash P}{\Gamma \vdash Q}\ \to\texttt{-Elim}$$

These will all show up again when we talk about *typechecking* a program.

Classical propositional proof theory

We can extend our proof theory to classical logic simply by adding the law of the excluded middle as an axiom.

$$\frac{}{\Gamma \vdash P \vee \neg P}\ \texttt{ExMid}$$

Now we can ask an interesting question:

Does this proof theory agree with the Boolean interpretation of classical logic?

We write $\Gamma \models P$ to mean

the truth table for P says P is true whenever all the propositions in Γ are true.

- Does knowing a proof of $\Gamma \vdash P$ imply $\Gamma \models P$?
- Does knowing $\Gamma \models P$ tell us enough to construct a proof of $\Gamma \vdash P$?

Soundness and completeness

Our proof theory is

- *internally consistent* if it can never prove both P **and** $\neg P$
- *internally complete* if it can always prove either P **or** $\neg P$
- *sound with respect to* the Boolean interpretation if everything provable is true.
- *complete with respect to* the Boolean interpretation if everything true is provable.

Here we use the word "true" in the Boolean sense.

It's possible to show all these properties hold for our classical proof theory.

Incompleteness

You might have heard of *Gödel's (first) incompleteness theorem.*

It's often phrased something like

> There are things that are true but unprovable.

or

> Logic is incomplete.

but those phrasings leave out important details.

There are many different systems of logic, and the incompleteness theorem doesn't apply to all of them.

This is a very deep subject - I just wanted to clear up some common misconceptions.

Incompleteness

What **do** the incompleteness theorems say?

Gödel proved

- any **internally consistent** theory that can talk about **basic arithmetic** must be **internally incomplete**
- no theory can **prove its own completeness**

Second-order predicate logic is very commonly used in mathematics, and is incomplete.

Incompleteness isn't really a problem with a system so much as a tradeoff: we gain expressive power at the cost of completeness.

Incompleteness is intimately related to the idea of *undecidability* in computer science.

- In a sufficiently expressive logic, we can't know a proof or counterproof of every proposition.
- In a sufficiently expressive programming language, we can't decide the halting behavior of every program.
- If there's a decidable proof search procedure for some logic, the logic is complete.

Looking ahead to type theory

We can start moving back to programming language theory with a simple question:

> We have a two-dimensional syntax (proof trees) for our intuitionistic proofs.
>
> Can we come up with a convenient one-dimensional syntax for the same proofs?

It turns out that **lambda calculus** is a good candidate for this syntax!

In the next set of lectures, this idea will lead us to the study of *type theory*.

Lessons learned

So which logic is the right logic?

This is not a settled question - it's still under active debate among experts, and will continue to be under debate for the forseeable future.

My favorite answer is:

> The right logic is the one that's most useful for whatever you're trying to do with it.

> . . . **you may call me a mathematical relativist: there are many worlds of mathematics, and the view of the worlds is relative to which one I am in**. Any attempt to bring mathematics within the scope of a single foundation necessarily limits mathematics in unacceptable ways.
>
> Andrej Bauer, "Am I a constructive mathematician?"

In this class, intuitionistic logic turns out to be pretty useful!