Logic and Computation – CS 2800, Fall 2023

Formal proofs (by hand)

Stavros Tripakis

Northeastern University
Khoury College of Computer Sciences

September 28, 2023

PROOFS

Proofs

Suppose we want to prove that a given formula is valid.

How to do it?

- We can try to reason in natural language (e.g., in English), as in

  $(p \wedge p \to q) \to q$ *is valid, because assuming both* p *and* $p \to q$ *to be true, since* p *is true, and* p *implies* q *by* $p \to q$*, we can conclude that* q *must also be true.*

- This is an **informal** proof.

  Not very satisfactory ...

Brute-force method: generate truth table

If the formula is a propositional logic formula, we can build its truth table and check the last column: if it's all 1s (trues) then the formula is valid, otherwise it's not.

Brute-force method: generate truth table

If the formula is a propositional logic formula, we can build its truth table and check the last column: if it's all 1s (trues) then the formula is valid, otherwise it's not.

Problems:

- Only works for propositional logic! What about $\forall n : \texttt{nat}, ...$?
  Cannot build an infinite truth table!

Brute-force method: generate truth table

If the formula is a propositional logic formula, we can build its truth table and check the last column: if it's all 1s (trues) then the formula is valid, otherwise it's not.

Problems:

- Only works for propositional logic! What about $\forall n : \texttt{nat}, ...$?
  Cannot build an infinite truth table!
- Even for propositional logic, it's often **intractable**: what if my formula has 1000 variables? How big is the truth table?

Brute-force method: generate truth table

If the formula is a propositional logic formula, we can build its truth table and check the last column: if it's all 1s (trues) then the formula is valid, otherwise it's not.

Problems:

- Only works for propositional logic! What about $\forall n : \texttt{nat}, \ldots$? **Cannot build an infinite truth table!**
- Even for propositional logic, it's often **intractable**: what if my formula has 1000 variables? How big is the truth table?
  - However, **SAT solvers** today do a great job at proving propositional logic formulas with even millions of variables! We will discuss SAT solvers later in this course.

Formal proofs

- Systematic and rigorous.
- Syntax-based (we'll see what this means).
- Pros: general = work not just for propositional logic, but for infinite domains as well!
- Cons: need human guidance $\Rightarrow$ non-automated or partially automated.
  - But lots of people are working towards automation, e.g., with **SMT solvers**. We'll discuss those later in the course.

Formal proofs

- Systematic and rigorous.
- Syntax-based (we'll see what this means).
- Pros: general = work not just for propositional logic, but for infinite domains as well!
- Cons: need human guidance $\Rightarrow$ non-automated or partially automated.
  - But lots of people are working towards automation, e.g., with **SMT solvers**. We'll discuss those later in the course.

- We will first learn how to do formal proofs "by hand" ("manually", i.e., without LEAN)!
- Then we will learn to do them in LEAN.

FORMAL PROOFS

Formal proofs

- A **proof state**: can be either of the form $\mathcal{H} \vdash G$, where
  - $\mathcal{H}$ is a set of propositions that we call **hypotheses** or **assumptions**; ($\mathcal{H}$ might be empty)
  - G is a proposition that we call the **goal**

  or the special proof state "`goals accomplished`" which we will also abbreviate as `DONE`.
- The intuitive meaning of $\mathcal{H} \vdash G$ is: *prove that G is true, assuming that all things in $\mathcal{H}$ are true*.

Formal proofs

- A **proof state**: can be either of the form $\mathcal{H} \vdash G$, where
  - $\mathcal{H}$ is a set of propositions that we call **hypotheses** or **assumptions**; ($\mathcal{H}$ might be empty)
  - G is a proposition that we call the **goal**

  or the special proof state "`goals accomplished`" which we will also abbreviate as `DONE`.
- The intuitive meaning of $\mathcal{H} \vdash G$ is: *prove that G is true, assuming that all things in $\mathcal{H}$ are true*.
- A **formal proof** is a tree whose nodes are proof states.
- In that tree, every link between a parent proof state and its children is labeled by a **proof rule**, and these states must **obey** that proof rule. (We will see what that means.)
- If all the leaves of the tree are `DONE` then the proof is **complete**, otherwise it's **incomplete**.
- Leaves of the form $\mathcal{H} \vdash G$ are also called **proof obligations**.

Formal proof: an example

(Just like in a real tree, in our proof trees the root is at the bottom and we go upwards!)

$$\dfrac{\dfrac{\texttt{DONE}}{p \vdash p}\ \text{Assumption}}{\vdash p \to p}\ \text{ImpIntro}$$

This proof is complete.

Formal proof: another example

$$\dfrac{\dfrac{\texttt{DONE}}{p \vdash p}\,\text{Assumption} \qquad \dfrac{\texttt{DONE}}{p \vdash p}\,\text{Assumption}}{\dfrac{p \vdash p \wedge p}{\vdash p \rightarrow (p \wedge p)}\,\text{ImpIntro}}\,\text{And}$$

This proof is complete.

Formal proof: another example

$$\dfrac{\dfrac{\dfrac{\texttt{DONE}}{p \vdash p}\ \text{Assumption}}{p \vdash p \vee q}\ \text{OrLeft}}{\vdash p \rightarrow (p \vee q)}\ \text{ImpIntro}$$

This proof is complete.

Formal proof: another example

$$\dfrac{\dfrac{p \vdash q}{p \vdash p \vee q}\;\text{OrRight}}{\vdash p \to (p \vee q)}\;\text{ImpIntro}$$

This proof is incomplete.

How do we build a formal proof?

- Let's say we want to prove some proposition G (the goal).
- Typically we start with the proof state $\vdash G$ (no hypotheses). This will be the root of the proof tree.
- Then, for every leaf s of the tree which is not `DONE`:
  - We pick a proof rule R that **applies** to s (we will see what applies means).
  - We apply R to s: this generates one or more children of s.
  - We add the children of s to the tree and continue.

How do we build a formal proof?

- Let's say we want to prove some proposition G (the goal).
- Typically we start with the proof state $\vdash G$ (no hypotheses). This will be the root of the proof tree.
- Then, for every leaf s of the tree which is not `DONE`:
  - We pick a proof rule R that **applies** to s (we will see what applies means).
  - We apply R to s: this generates one or more children of s.
  - We add the children of s to the tree and continue.
- When all leaves are `DONE` the proof is complete! ☺

How do we build a formal proof?

- Let's say we want to prove some proposition G (the goal).
- Typically we start with the proof state $\vdash G$ (no hypotheses). This will be the root of the proof tree.
- Then, for every leaf s of the tree which is not `DONE`:
  - We pick a proof rule R that **applies** to s (we will see what applies means).
  - We apply R to s: this generates one or more children of s.
  - We add the children of s to the tree and continue.
- When all leaves are `DONE` the proof is complete! ☺
- If we still have proof obligations but we cannot find any rule that applies to them, we are stuck, and the proof is left incomplete … ☹

How do we build a formal proof?

- Let's say we want to prove some proposition G (the goal).

- Typically we start with the proof state $\vdash G$ (no hypotheses).
  This will be the root of the proof tree.

- Then, for every leaf s of the tree which is not `DONE`:
  - We pick a proof rule R that **applies** to s (we will see what applies means).
  - We apply R to s: this generates one or more children of s.
  - We add the children of s to the tree and continue.

- When all leaves are `DONE` the proof is complete! ☺

- If we still have proof obligations but we cannot find any rule that applies to them, we are stuck, and the proof is left incomplete ... ☹

- Or we might keep adding more and more nodes to the tree with no end in sight ... ☹

SOME PROOF RULES

Closing a brach when the goal is already in the hypotheses:

Assumption

$$\frac{\texttt{DONE}}{\mathcal{H}, G \vdash G}\ \text{Assumption}$$

- This rule applies to proof states where the goal G is already in the set of hypotheses.
- Intuition: G is true by assumption (G is in my set of hypotheses).
- This rule generates only one child.

Eliminating an implication in the goal: ImpIntro

$$\frac{\mathcal{H}, A \vdash B}{\mathcal{H} \vdash A \to B} \text{ ImpIntro}$$

- This rule applies to proof states where the goal is an implication, i.e., where the goal is of the form $A \to B$.
- Intuition: to prove $A \to B$ assuming that all things in $\mathcal{H}$ are true, it suffices to prove B assuming that all things in $\mathcal{H}$ are true and also that A is true.
- This rule generates only one child.

Eliminating a conjunction in the goal: And

$$\frac{\mathcal{H} \vdash A \qquad\qquad \mathcal{H} \vdash B}{\mathcal{H} \vdash A \wedge B}\ \text{And}$$

- This rule applies to proof states where the goal is a conjunction, i.e., where the goal is of the form $A \wedge B$.
- Intuition: to prove $A \wedge B$ assuming that all things in $\mathcal{H}$ are true, it suffices to do two separate proofs: first, prove A assuming that all things in $\mathcal{H}$ are true; and second, prove B assuming that all things in $\mathcal{H}$ are true.
- This rule generates two children.

Eliminating a disjunction in the goal by choosing to prove the left part: OrLeft

$$\frac{\mathcal{H} \vdash A}{\mathcal{H} \vdash A \lor B} \text{ OrLeft}$$

- This rule applies to proof states where the goal is an implication, i.e., where the goal is of the form $A \lor B$.
- Intuition: to prove $A \lor B$ assuming that all things in $\mathcal{H}$ are true, it suffices to prove A assuming that all things in $\mathcal{H}$ are true.
- This rule generates only one child.

Eliminating a disjunction in the goal by choosing to prove the right part: OrRight

$$\frac{\mathcal{H} \vdash B}{\mathcal{H} \vdash A \vee B} \text{ OrRight}$$

- This rule applies to proof states where the goal is an implication, i.e., where the goal is of the form $A \vee B$.
- Intuition: to prove $A \vee B$ assuming that all things in $\mathcal{H}$ are true, it suffices to prove B assuming that all things in $\mathcal{H}$ are true.
- This rule generates only one child.

If the goal is `true` we are done: True

$$\frac{\texttt{DONE}}{\mathcal{H} \vdash \texttt{true}} \ \text{True}$$

- This rule applies to proof states where the goal is the proposition `true`.
- Intuition: `true` holds by definition.
- This rule generates only one child.

If a hypothesis is `false` we are done: False

$$\frac{\texttt{DONE}}{\mathcal{H}, \texttt{false} \vdash G}\ \textsf{False}$$

- This rule applies to proof states where the set of hypotheses contains the proposition `false`.
- Intuition: if I assume `false` then I can prove anything I want.
- This rule generates only one child.