Propositional Proofs and Validity

Robert Y. Lewis

CS 0220 2024

February 2, 2024

Overview

The proof game, revisited

Remember our setup from last class:

At any point in a proof, we have some *goals* and their corresponding *contexts*.

- A goal is a proposition that we want to prove.
- A context is a list of *hypotheses*, propositions that we know.

We complete a proof by repeatedly transforming these goals and hypotheses by applying *proof rules*, which are individual reasoning steps.

Introduction rules, revisited

Introduction rules were valid based on the shape of the goal.

- To prove $A \wedge B$, it suffices to prove A, then to prove B.
- To prove $A \vee B$, it suffices to prove A.
- To prove $A \vee B$, it suffices to prove B.
- ...

These proof rules update the *goal* without changing the *context*. Contrast:

- To prove $A \to B$, it suffices to prove B, using the extra hypothesis A.

"And" Elimination

If you know $P \wedge Q$, you know two things:

- P
- Q

Yes, this sounds silly to say out loud. We usually don't think about this.

In terms of proof state: turns one hypothesis into two smaller hypotheses.

"Or" Elimination

This one's more interesting!

If you know $P \vee Q$, and your goal is G, you can *reason by cases*. That is: if you show $P \to G$, and you show $Q \to G$, then you have shown G.

In terms of proof state: creates two goals, each with a new hypothesis.

Implication Elimination: modus ponens

If x is prime, then $x \geq 2$. x is prime. Therefore, $x \geq 2$.

General pattern: if you know $P \to Q$ and you know P, then you know Q.

Adds a hypothesis.

Alternate phrasing: if your goal is to show Q, and you know $P \to Q$, it suffices to show P.

Changes the goal.

(Iff elimination is easy: if you know $P \leftrightarrow Q$, then you know $P \to Q$ and $Q \to P$.)

In Lean

Introduction rules in Lean:

- and elim: `eliminate h with h1 h2`
- or elim: `eliminate h with h1 h2`
- implication elim: `have hb := hab ha`
- iff elim: `eliminate h with h1 h2`

Getting comfortable with contradiction

We live in a world where things make sense. (...)

In our sensible world, some statements are true and some are false. But none are true *and* false.

So if we can prove both a proposition and its negation, we're living in nonsense land. Anything follows.

Negation elimination and introduction

Negation elimination: if you know P and you know $\neg P$, you can prove anything (i.e. close any goal).

Negation introduction: if your goal is to prove $\neg P$, you can assume P, and show "false". "Proof by contradiction!"

Example proof by contradiction

Proposition: $\sqrt{2}$ is not rational.

We prove that $\sqrt{2}$ is not rational by contradiction. Suppose $\sqrt{2}$ is rational. By the definition of "rational", that means $\sqrt{2} = p/q$ where p and q are integers. Furthermore, we can choose p and q to be in lowest terms so they have no factors in common. Squaring both sides, we get $2 = p^2/q^2$ or $2q^2 = p^2$. Since q^2 is an integer, and p^2 is an integer times 2, p^2 is even. By a similar argument to the one for odd squares (from a few lectures ago), that means p must be even. If p is even, p^2 must be divisible by 4. Since $2q^2$ is divisible by 4, q^2 must be divisible by 2 (the other factor of two must be there). That means both p and q are even. But, then p/q is not in lowest terms. Since we already asserted that p/q is in lowest terms when p and q were chosen, we've reached a contradiction. Therefore, $\sqrt{2}$ must be irrational.

A subtlely different proof by contradiction

From the last slide: if your goal is to prove $\neg P$, you can assume P, and show "false".

Compare to:

Proof by contradiction: if your goal is to prove P, you can assume $\neg P$, and show "false".

Validity (3.3.2)

Back to truth for a moment!

DeMorgan's Law

These two statements are equivalent:

- $\neg(P \wedge Q)$
- $\neg P \vee \neg Q$

They are equivalent because they have exactly the same truth table. (Or, because we can *prove* $\neg(P \wedge Q) \leftrightarrow (\neg P \vee \neg Q)$.) You can think of this as negation "distributing" over AND, negating the inputs and switching the AND to OR.

Equivalence and validity: Definitions

A formula can be thought of as a function mapping variable assignments to truth values. Each row of the truth table shows one input and its corresponding output.

Definition: Two formulas over the same set of variables are *equivalent* if they evaluate to the same truth value under every variable assignment.

Definition: A formula is *valid* if it is always true regardless of variable assignment.

Example: $P \vee \neg P$

P	$\neg P$	$P \vee \neg P$
F	**T**	**T**
T	**F**	**T**

Equivalence and validity

A formula is valid iff it is equivalent to **T**.

Two formulas α and β are equivalent iff $\alpha \leftrightarrow \beta$ is valid.

Example: Show "P" is equivalent to "$\neg\neg P$".

P	$\neg P$	$\neg\neg P$	$P \leftrightarrow \neg\neg P$
F			
T			

Equivalence and validity

A formula is valid iff it is equivalent to **T**.

Two formulas α and β are equivalent iff $\alpha \leftrightarrow \beta$ is valid.

Example: Show "P" is equivalent to "$\neg\neg P$".

P	$\neg P$	$\neg\neg P$	$P \leftrightarrow \neg\neg P$
F	**T**		
T	**F**		

Equivalence and validity

A formula is valid iff it is equivalent to **T**.

Two formulas α and β are equivalent iff $\alpha \leftrightarrow \beta$ is valid.

Example: Show “P” is equivalent to “$\neg\neg P$”.

P	$\neg P$	$\neg\neg P$	$P \leftrightarrow \neg\neg P$
F	**T**	**F**	
T	**F**	**T**	

Equivalence and validity

A formula is *valid* iff it is equivalent to **T**.

Two formulas α and β are equivalent iff $\alpha \leftrightarrow \beta$ is valid.

Example: Show "P" is equivalent to "$\neg\neg P$".

P	$\neg P$	$\neg\neg P$	$P \leftrightarrow \neg\neg P$
F	**T**	**F**	**T**
T	**F**	**T**	**T**

Satisfiability

Definition: A formula is *satisfiable* if at least one assignment evaluates to true.

A formula is satisfiable iff its negation is not valid. (DeMorgan's law in another form.)

Validity is kind of like "$\forall$".

Satisfiability is kind of like "$\exists$".

Determining whether a formula is satisfiable, efficiently, is a core problem in computer science. Examples: Solving puzzles, finding successful plans, arranging items in space, factoring, finding paths in graphs...

Checking satisfiability and validity

Easy if few variables. Just write out the truth table!

P	Q	$\neg Q$	$\neg P$	$Q \vee \neg P$	$\neg Q \wedge (Q \vee \neg P)$
F	**F**	**T**	**T**	**T**	**T**
F	**T**	**F**	**T**	**T**	**F**
T	**F**	**T**	**F**	**F**	**F**
T	**T**	**F**	**F**	**T**	**F**

If all rows are T: *valid*. If at least one row is T: *satisfiable*.

Blows up as the number of variables gets large. Need another way.

Theorem: A propositional formula is valid if and only if it can be proved using only the proof rules we have introduced here (including proof by contradiction).