19 - Reductio Ad Absurdum

Lean: First Steps

Tariq Rashid

December 17, 2024

More Interesting Proofs

- Over the next few tutorials we'll explore some more interesting kinds of proof.
- Here we'll take a first look at **proof by contradiction**.

Task

- Given these two facts about natural numbers a and b,

$$(a = 5) \implies (b = 6)$$

$$b = 7$$

- show that

$$\neg a = 5$$

- The symbol $\neg$ means **negation**, and reads as "it is not the case that".

- $\neg a = 5$ reads "it is not the case that $a = 5$", or simply, "a is not 5."

Maths

- Intuition
  - $a = 5 \implies b = 6$ tells us that if $a = 5$ then $b = 6$.
  - But b is supposed to be 7.
  - So it can't be the case that $a = 5$.
- This intuition actually matches the more formal approach we'll take.

To prove a statement is false
we show it leads to a **contradiction**.

- In symbols: to show $\neg P$ we need to show the statement P leads to a contradiction.
- For our task, P is $a = 5$. So, to show $\neg a = 5$, we need to show $a = 5$, taken as a hypothesis, leads to a contradiction.

Maths

$$
\begin{aligned}
(a = 5) &\implies (b = 6) && \text{given fact} && (1)\\
b &= 7 && \text{given fact} && (2)\\
\\
\neg a &= 5 && \text{proof objective}\\
\\
\text{assume } a &= 5 && \text{for contradiction} && (3)\\
b &= 6 && \text{using (1)}\\
&\neq 7 && \text{arithmetic} && (4)\\
\\
\text{(4) contradicts (2)} & && \text{(3) must be false} && \square
\end{aligned}
$$

Reductio Ad Absurdum

- Proof by contradiction is sometimes called *reductio ad absurdum*, Latin for "reduction to absurdity".
- In our example, the absurdity is the notion that $b = 6$ and $b = 7$.

Code

```
-- 19 - Proof by Contradiction

import Mathlib.Tactic

example {a b : ℕ} (h1: a = 5 → b = 6) (h2: b = 7) : ¬ a = 5 :=
    by
  intro g
  apply h1 at g
  have h2x : ¬ b = 7 := by linarith
  contradiction
```

Code

- To prove $\neg P$ we assume the statement P is true and try to derive a contradiction. The `intro g` starts this journey. It takes the goal `¬ a = 5`, creates a new hypothesis `g : a = 5`, and then sets the goal to `False`.
- A goal of `False` means we have to show a contradiction.
- We do this by arranging for two hypotheses to exist, one of the form `h1: P` and the other `h2: ¬ P`.
- Once that's done, `contradition` resolves the `False` goal.
  - `contradition` searches the current hypotheses to find any two that are of the form `h1: P` and `h2: ¬ P`.

Code

- Code between `intro g` and `contradiction` arranges for two contradictory hypotheses:
  - The first hypothesis `h1 : a = 5 → b = 6` is applied to the newly created hypothesis `g : a = 5`, which is changed to `b = 6`.
  - `b = 6` contradicts the second hypothesis `h2 : b = 7` but we need to arrange for hypotheses of the form `h1: P` and `h2: ¬ P`.
  - We create a new intermediate result `h2x : ¬b = 7`, justified by surprisingly capable **linarith** tactic.
  - We now have two directly contradictory hypotheses, the given `h2 : b = 7` and the derived `h2x : ¬b = 7`.
  - We're now ready to use `contradiction` to complete the proof.

Infoview

- Placing the cursor before `intro g` shows the original proof goal.

```
⊢ ¬a = 5
```

- Moving the cursor to the start of the next line shows `a = 5` has been added as hypothesis `g`, and the proof goal changed to `False`.

```
g : a = 5
⊢ False
```

Infoview

- Placing the cursor just before `contradiction` shows the two directly contradictory hypotheses `h2` and `h2x`.

```
h2 : b = 7
g : b = 6
h2x : ¬b = 7
```

Easy Exercise

- Write a Lean program to prove $\neg a = 5$, given $a > 5 \iff b = 6$ and $b = 6$.

- Here a and b are natural numbers.