13 - Disequality

Lean: First Steps

Tariq Rashid

October 27, 2024

Using Lemmas

- In addition to **definitions**, like `Odd` and `Even`, Mathlib also contains **lemmas** and **theorems** we can use.

- Here we'll use a lemma to support a simple disequality proof.

Task

- Given a natural number $n < 5$, show that

$$n \neq 5$$

- $n \neq 5$ is a **disequality**
- $n < 5$ is an **inequality**

Task

- Common knowledge : given $a, b \in \mathbb{N}$, if $a < b$ is true, then $a \neq b$.
- That common knowledge might seem trivial, but we'll think of it as a small **lemma**.
- If we can show $n < 5$, then we can conclude $n \neq 5$, using that lemma.

Maths

$$n < 5 \qquad \text{hypothesis} \qquad (1)$$

$$n \neq 5 \qquad \text{proof objective} \qquad (2)$$

$$a < b \implies a \neq b \qquad \text{existing lemma} \qquad (3)$$

$$n < 5 \qquad \text{sufficient goal, by lemma (3)} \qquad (4)$$

$$n < 5 \qquad \text{using (1)} \qquad (5)$$

$$n < 5 \implies n \neq 5 \qquad \text{by lemma (3)} \qquad \square$$

Maths

- We start with the hypothesis $n < 5$, and our proof objective $n \neq 5$.
- We know about a lemma (3) applicable to natural numbers, that if $a < b$ then $a \neq b$.
    - If we can prove $n < 5$, then we can conclude $n \neq 5$.
    - This changes our proof goal from $n \neq 5$ to $n < 5$.
- $n < 5$ is given by hypothesis (1).
- So $n < 5$, and by lemma (3) we finally conclude $n \neq 5$.

Code

```
-- 13 - Lemma: Not Equal from Less Than

import Mathlib.Tactic

example {n : ℕ} (h: n < 5): n ≠ 5 := by
  apply ne_of_lt
  exact h
```

Code

- Proof header declares `n` as a natural number, establishes hypothesis `h: n < 5`, specifies proof objective `n ≠ 5`.
- `apply` **applies** a lemma or theorem to the current goal, usually resulting in a **change in goal**.
- Here, it applies a lemma named `ne_of_lt`, which means "not equal from less than".
  - we can prove the "not equal" goal by proving a "less than" goal.
- The Infoview will show that `apply ne_of_lt` does indeed change the current proof goal from `n ≠ 5` to `n < 5`.

Code

- The current goal is now `n < 5`.
- We could use `apply h` to resolve the goal.
- Since the goal matches **exactly** hypothesis `h`, we can use `exact h`.
- It may be helpful to correlate this new code back to the maths proof:
  - `apply ne_of_lt` corresponds to line (4) of the maths proof
  - `exact h` corresponds to line (5).

`apply` & `exact`

- We can use `apply` wherever we use `exact`.
- The benefit of `exact` is that it is stricter than `apply`.
  - The hypothesis or lemma must exactly match the current goal, and if a misunderstanding has led to that not being true, it will be exposed immediately.

Infoview

- Placing the cursor before `apply ne_of_lt` shows the original proof goal.

```
n : ℕ
h : n < 5
⊢ n ≠ 5
```

- Moving the cursor to the beginning of the next line after `apply ne_of_lt` shows the goal has changed.

```
n : ℕ
h : n < 5
⊢ n < 5
```

Lemmas & Theorems

- The distinction between what is called a **lemma** or a **theorem** in Mathlib is not precise.
- Ultimately it doesn't matter as both are used in the same way.
- Searching for suitable lemmas and theorems in Mathlib is currently not ideal. Many do conform to a **naming convention**, which helps.

Easy Exercise

- Write a Lean program to prove $n \neq 5$, given $n > 5$, where n is a natural number.
- The proof will be almost exactly the same as this chapter's example, except the lemma will be "not equal from greater than".
- Work out the required lemma's Mathlib name fom the naming convention, or search the online Lean documentation to find it.