17 - Using Our Own Lemma

Lean: First Steps

Tariq Rashid

December 26, 2024

Using Our Own Lemma

- In the last chapter we created our own lemma.
- Here we'll make use of that lemma to prove a disequality.

Task

- For any natural number n, show that

$$n^2 \neq 7$$

Maths

- Proof strategy - split the natural number sequence into two
  - all the smaller numbers result in $n^2 < 7$
  - all the larger numbers in $n^2 > 7$
- If every possible choice for n leads to either $n^2 < 7$ or $n^2 > 7$, then $n^2 \neq 7$.

Maths

$n^2 \neq 7$	proof objective	
$n \leq m \vee m+1 \leq n$	our lemma, for $m, n \in \mathbb{N}$	(1)
$n \leq 2 \vee 3 \leq n$	lemma (1) with $m = 2$	(2)
case $n \leq 2$	using (2)	
$n^2 \leq 4$		
< 7		
$n^2 \neq 7$	lemma $a < b \implies a \neq b$	(3)
case $n \geq 3$	using (2)	
$n^2 \geq 9$		
> 7		
$n^2 \neq 7$	lemma $a > b \implies a \neq b$	(4)
$n \neq 7$	conclusion of both cases	

Maths

- **Our lemma** (1) gives a disjunction of two cases, $n \leq 2$ and $n \geq 3$.
- The first case $n \leq 2$
  - means $n^2 \leq 4$, which also means $n^2 < 7$
  - Using small lemma $a < b \implies a \neq b$, we conclude $n^2 \neq 7$
- The second case $n \geq 3$
  - means $n^2 \geq 9$, which also means $n^2 > 7$.
  - Using small lemma $a > b \implies a \neq b$, we conclude $n^2 \neq 7$.
- Both cases lead to $n^2 \neq 7$, completing the proof.

Code 1/2

```
-- 17 - Using Our Own Lemma

import Mathlib.Tactic

---

lemma Nat.le_or_succ_le (a b : ℕ): a ≤ b ∨ b + 1 ≤ a := by
  rw [Nat.add_one_le_iff]
  exact le_or_lt a b

---
```

Code 2/2

```
example {n : ℕ} :  n^2 ≠ 7  := by
  have h := Nat.le_or_succ_le n 2
  obtain ha | hb := h
  · apply ne_of_lt
    calc
      n^2 ≤ 2^2 := by rel [ha]
      _ < 7 := by norm_num
  · apply ne_of_gt
    calc
      n^2 ≥ 3^2 := by rel [hb]
      _ > 7 := by norm_num
```

Code

- The code is separated into two sections, separated by comment dashes.
- The first section is a copy of the lemma we developed, `Nat.le_or_succ_le`.
- The second section is the Lean proof that $n^2 \neq 7$.

Code

- **Our lemma** `Nat.le_or_succ_le` with parameters `n` and `2`, gives us `n ≤ 2 ∨ 2 + 1 ≤ n`.

- We capture this as hypothesis `h`. We use `obtain` to split `h` into **two cases**, `n ≤ 2` and `2 + 1 ≤ n`.

- **The first case** starts by changing the goal from `n^2 ≠ 7` to `n^2 < 7` by applying Mathlib lemma `ne_of_lt`. Its definition confirms that if we can prove `n^2 < 7` then we've proven `n^2 ≠ 7`:

```
lemma ne_of_lt (h : a < b) : a ≠ b :=
```

- We use a **calc** section to show `n^2 ≤ 2^2` from the current case `n ≤ 2`, then `2^2 < 7`, all of which resolves the goal `n^2 < 7`.

Code

- **The second case** is similar. It starts by changing the goal from `n^2 ≠ 7` to `n^2 > 7` by applying Mathlib lemma `ne_of_gt`. Its definition confirms that if we can prove `n^2 > 7` then we've proven `n^2 ≠ 7`:

```
lemma ne_of_gt (h : b < a) : a ≠ b :=
```

- Again, we use a **calc** section to show `n^2 ≥ 3^2` from the current case `2 + 1 ≤ n` , then `3^2 > 7`, all of which resolves the goal `n^2 > 7`.

Easy Exercise

- Write a Lean proof to show that $n^3 \neq 10$, for any natural number n.