15 - Zero Product

Lean: First Steps

Tariq Rashid

November 12, 2024

Bidirectional Lemmas

- Some lemmas are bidirectional implications $\iff$.
- To apply them, we decide which of the two directions match our task.

Task

- Given the zero product

$$(p-1)\cdot(q-2)=0$$

- show that

$$p=1 \text{ or } q=2$$

- where p and q are rational numbers.

- It is common knowledge that the product of two numbers is zero, if and only if, one or the other is zero.
- We'll think of this as a lemma,

$$a \cdot b = 0 \iff (a = 0) \vee (b = 0)$$

Maths

- $P \iff Q$ is **equivalent** to both of the following being true:
  - $P \implies Q$
  - $Q \implies P$
- We say "P is true **if and only if** Q is true"
  - "P is true **iff** Q is true."

Maths

- So our lemma is equivalent to the following two statements:
  - $a \cdot b = 0 \implies (a = 0) \vee (b = 0)$
  - $(a = 0) \vee (b = 0) \implies a \cdot b = 0$
- The antecedent of the first matches our hypothesis $(p - 1) \cdot (q - 2) = 0$, so applying it gives us a disjunction.

$$(p - 1 = 0) \vee (q - 2 = 0)$$

- We could conclude $p = 1$ or $q = 2$ by inspection, but a lean proof needs more detail.

Maths

$(p-1)\cdot(q-2)=0$	hypothesis	(1)
$a\cdot b=0 \iff (a=0)\vee(b=0)$	existing lemma	(2)
$(p-1=0)\vee(q-2=0)$	apply (2) to (1)	(3)
case $p-1=0$	using (3)	
$p=1$	add 1 to both sides	
case $q-2=0$	using (3)	
$q=2$	add 2 to both sides	
$(p=1)\vee(q=2)$	conclusion	$\square$

- Key step is applying the lemma (2) to the hypothesis (1) to derive a new **intermediate result** (3).
- This intermediate result is a disjunction $(p-1=0) \vee (q-2=0)$, so the proof proceeds as a proof by cases.

Code

```
-- 15 - Using Lemmas: Multiplied Factors Equal Zero

import Mathlib.Tactic

example {p q : ℚ} (h : (p - 1) * (q - 2) = 0) : p = 1 ∨ q = 2 :=
    by
  apply mul_eq_zero.mp at h
  obtain hp | hq := h
  · left
    linarith
  · right
    linarith
```

Code

- Mathlib lemma `mul_eq_zero` says $a \cdot b = 0 \iff (a = 0) \vee (b = 0)$.
- Let's check its definition:

```
theorem mul_eq_zero : a * b = 0 ⟺ a = 0 ∨ b = 0 :=
```

Code

- The first line of the proof applies `mul_eq_zero` to hypothesis `h`, but the lemma name has a `.mp` appended.
  - Adding `.mp` to a bidirectional lemma selects forward direction.
  - Adding `.mpr` to a bidirectional lemma selects reverse direction.
- `mp` is "modus ponens", terminology from logic for the process of reaching a conclusion using an implication.
- Infoview will confirm the hypothesis has been **replaced** with the equivalent disjunction `p - 1 = 0 ∨ q - 2 = 0`.

Code

- The proof then proceeds as a proof by cases.
  - `obtain` splits hypotheses `h` into two cases, `p - 1 = 0` and `q - 2 = 0`.
  - First case uses `left` to prove `p = 1`. The **linarith** tactic can perform "add a number to both sides" to justify the goal `p = 1` from the current hypotheses, one of which is `hp : p - 1 = 0`.
  - The second case is almost the same, but we use `right` to prove `q = 2`.
- Both cases, `p - 1 = 0` and `q - 2 = 0`, confirm the same disjunctive proof goal, so the proof is complete.

Replacing A Hypothesis?

- In the maths proof we derived an **intermediate result** from the hypothesis.
- But in the Lean code the hypothesis was **replaced**.
- That's ok, it was replaced by an **equivalent** statement.

Infoview

- Placing the cursor before `apply mul_eq_zero.mp at h` shows the original hypothesis.

```
h : (p - 1) * (q - 2) = 0
```

- Moving the cursor to the beginning of the next line shows the hypothesis has been replaced.

```
h : p - 1 = 0 ∨ q - 2 = 0
```

Easy Exercise

- Write a Lean program to prove $(p-1)\cdot(q-2) \neq 0$ given $p-1 \neq 0$ and $q-2 \neq 0$, where p and q are rational numbers.
- Use the lemma `mul_ne_zero_iff`.
  - Look up the definition to understand what it says, then decide in which direction to apply it.