06 - Itermediate Results

Lean: First Steps

Tariq Rashid

August 25, 2024

Intermediate Results

- Proof with more structure than simple direct calculation.
- Derive an intermediate result, which we make use of later in the full proof.

Task

- Given

$$a = b + 1$$
$$b - 1 = 0$$

- our task is to show

$$a = 2$$

- where $a, b \in \mathbb{Z}$

Maths

- Most mathematicians would immediately find $b = 1$, and then use this derived result in $a = b + 1$ to give $a = 2$.
- Up to this point, learning to think in Lean encouraged us to find a clever substituion trick using only the given facts.

Maths

- Structured proof.

$$
\begin{aligned}
a &= b + 1 && \text{given fact} && (1)\\
b - 1 &= 0 && \text{given fact} && (2)
\end{aligned}
$$

$$
\begin{aligned}
b - 1 &= 0 && \text{using fact (2)} &&\\
b &= 1 && \text{adding 1 to both sides} && (3)
\end{aligned}
$$

$$
\begin{aligned}
a &= b + 1 && \text{using fact (1)} &&\\
&= (1) + 1 && \text{using intermediate result (3)} && (4)\\
a &= 2 && \text{using arithmetic} && \square
\end{aligned}
$$

Maths

- We've derived an **intermediate result** $b = 1$ at line (3), and used it later at line (4).
    - This is new.
- We justify $b = 1$ from $b - 1 = 0$ by adding 1 to both sides.
    - Uncontroversial step, doesn't need deeper justification.

Maths

- Previous thinking in Lean
  - rewrite $a = b + 1$ as $a = b - 1 + 1 + 1$
  - use given fact $b - 1 = 0$ to remove $b - 1$ from the expression
  - to give $a = (0) + 1 + 1 = 2$
- This trickery is not as natural as establishing $b = 1$ as an intermediate result.

Code

```
-- 06 - Intermediate Result

import Mathlib.Tactic

example {a b : ℤ} (h1 : a = b + 1) (h2: b - 1 = 0) : a = 2 := by
  have h3: b = 1 := by linarith [h2]
  calc
    a = b + 1 := by rw [h1]
    _ = 1 + 1 := by rw [h3]
    _ = 2 := by norm_num
```

Code

- The final **calc** section is familiar, but uses a hyopthesis `h3` not given in the proof header.
- **Intermediate result** `b = 1` is given a label `h3`, similar to given facts in the proof header.
- Intermediate result is justified by "adding 1 to both sides" of `h2 : b - 1 = 0` using the tactic **linarith** for linear arithmetic.
- `have` establishes the new hypothesis `h3` alongside `h1` and `h2`.

Infoview

- As we develop and debug proofs, useful to keep track of what Lean thinks are the hypotheses it can use.

- Placing the cursor before the `have` instruction tells us the state of play before the new hypothesis is created.

```
a b : ℤ
h1 : a = b + 1
h2 : b - 1 = 0
⊢ a = 2
```

Infoview

- Moving the cursor to the beginning of the next line, just before `calc`, gives us the following.

```
a b : ℤ
h1 : a = b + 1
h2 : b - 1 = 0
h3 : b = 1
⊢ a = 2
```

- We can see a new hypothesis `h3 : b = 1` has been added to the existing `h1` and `h2`.

Easy Exercise

- Write a Lean program to prove $a = 2$, given $a = b + c$, $b - 1 = 0$, and $c + 1 = 2$ where $a, b, c \in \mathbb{Z}$.
- In your proof create and use two intermediate results, $b = 1$ and $c = 1$.