12 - Odd Numbers
Lean: First Steps

Tariq Rashid

November 11, 2024

Using Definitions: Odd Numbers

- There is a huge body of commonly agreed knowledge that mathematicians refer to in their own proofs.

- Lots of these **lemmas**, **theorems**, and **definitions** are in Mathlib .

- We'll start by using the **definition** of an **odd number**.

- Show the integer 13 is odd.

Task

- To show 13 is odd, we need to show it meets the **definition** of odd.

    - An **odd** integer is of the form $2k + 1$, where k is an integer.

    - If **there exists** an integer k **such that** $n = 2k + 1$, then n is odd.

- The task has become an **existence** proof.

    - If we can find an integer k such that $13 = 2k + 1$, then we have shown 13 is odd.

13 is odd   proof objective

$\exists k \in \mathbb{Z}[n = 2k + 1] \implies n$ is odd   definition of odd   (1)

$\exists k \in \mathbb{Z}[13 = 2k + 1]$ sufficient goal, using (1)

use $k = 6$ chosen example   (2)
$13 = 2(6) + 1$ using (2)

$13 = 2(6) + 1 \implies 13$ is odd   by definition (1) $\square$

Maths

- We start with the proof objective, to show 13 is odd.

- We then state the **definition**, that n is odd if it can be written in the form $2k + 1$, where k is an integer.

- To show 13 is odd, it is **sufficient** to show it can be written in the form $2k + 1$.

  - New goal, to show there exists an integer k such that $13 = 2k + 1$.

  - We choose $k = 6$, and confirm that $13 = 2(6) + 1$.

- We have shown 13 can be written in the form $2k + 1$.

- So, by the definition of odd, we have shown 13 is odd.

Code

```
-- 12 - Definition: Odd Number

import Mathlib.Tactic

example : Odd (13: ℤ) := by
  dsimp [Odd]
  use 6
  norm_num
```

Code

- The proof objective states that `13` is `Odd`.

  - `13` interpreted as a natural number.

  - `(13: ℤ)` specifies `13` as an integer.

- `Odd` is defined in Mathlib.

- `dsimp [Odd]` expands that definition in the Infoview.

  - From being displayed as `Odd 13`, to `∃ k, 13 = 2 * k + 1`.

Code

- After this point, the proof proceeds as a simple existence proof.

  - The instruction `use 6` tells Lean we want to try `6` for `k`.

  - This changes the goal to `13 = 2 * 6 + 1`.

  - We resolve this goal by arithmetic, using **norm_num**.

- For this simple goal, there is no need for a multi-line **calc** section.

What is `Odd`?

- `Odd` is a **definition** in Mathlib, and happens to be a function.

- That **function** `Odd` takes a number, and outputs a **proposition** involving that number.

  - `Odd` applied to `13`, output is a (true) proposition; $\exists k$ such that `13 = 2*k + 1`.

  - `Odd` applied to `14`, output is a (false) proposition; $\exists k$ such that `14 = 2*k + 1`.

- Interesting to see the definition of `Odd` inside Mathlib:

```
def Odd (a : α) : Prop := ∃ k, a = 2 * k + 1
```

Infoview

- Infoview is very useful when working with definitions and existence.

- Placing the cursor before `dsimp [Odd]` shows the original proof goal.

  ⊢ Odd 13

- Moving the cursor to the end of the line after `dsimp [Odd]` shows the goal is now displayed using the definition of `Odd`.

  ⊢ ∃ k, 13 = 2 * k + 1

- Placing the cursor after `use 6` shows the goal is now specific to `k = 6`.

  ⊢ ∃ k, 13 = 2 * 6 + 1

Easy Exercise

- Write a Lean program to prove the integer 14 is even.

- Your proof should use Mathlib's definition `Even` for even numbers.

- Use **dsimp** to see how the definition is applied to 14.