

All Lean code below assumes `import Mathlib`.

The following questions are *samples* of the sort of problem phrasing I may use on the midterm exam. There are more problems here than I would give on the actual exam. The exam will probably also include several short-to-medium length proofs in Lean, based on the techniques shown in class and used on homework assignments.

I may continue to add to this list in the next several days.

1. Translating statements into Lean

Convert each mathematical statement from English into a Lean statement. (You do not need to prove it.)

- (a) A product of two integers, each of which is equal to a sum of two integer squares, is again a sum of two integer squares.

Solution.

```

∀ m n : ℤ, (∃ a b : ℤ, m = a^2 + b^2) → (∃ c d : ℤ, n = c^2 + d^2) →
  ∃ e f : ℤ, m * n = e^2 + f^2

```

- (b) Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a function. For every real number $\varepsilon > 0$, there exists a real number $\delta > 0$ such that $|f(x) - f(y)| < \varepsilon$ whenever $|x - y| < \delta$.

Note: in Lean you can write absolute values using the `|` symbol, e.g. `|x + 1|`.

Solution. With $(f : \mathbb{R} \rightarrow \mathbb{R})$ as a hypothesis:

```

∀ ε : ℝ, ε > 0 → ∃ δ : ℝ, δ > 0 ∧ ∀ x y : ℝ, |x - y| < δ → |f x - f y| < ε

```

Lean also accepts the shorthand $\forall \varepsilon > 0, \exists \delta > 0, \forall x y : \mathbb{R}, |x - y| < \delta \rightarrow |f x - f y| < \varepsilon$.

- (c) For any odd integer n , $n^2 - 1$ is divisible by 4.

Solution.

```

∀ n : ℤ, Odd n → 4 ∣ n^2 - 1

```

- (d) The product of any two real numbers is at most the square of their average.

Solution.

```

∀ x y : ℝ, x * y ≤ ((x + y) / 2)^2

```

- (e) There is no nonzero integer b for which $2b^2$ is a perfect square.

Solution.

```

¬ ∃ b : ℤ, b ≠ 0 ∧ ∃ c : ℤ, 2 * b^2 = c^2

```

- (f) Every natural number $n > 1$ that is not prime has a divisor m with $1 < m < n$.

Solution.

$$\forall n : \mathbb{N}, n > 1 \rightarrow \neg \text{Prime } n \rightarrow \exists m : \mathbb{N}, 1 < m \wedge m < n \wedge m \mid n$$

- (g) There is no largest integer.

Solution.

$$\neg \exists M : \mathbb{Z}, \forall n : \mathbb{Z}, n \leq M$$

(Equivalently, $\forall M : \mathbb{Z}, \exists n : \mathbb{Z}, n > M$.)

2. Tactic states

In each problem, give the tactic state (that is, what is shown in the infoview) when the cursor is on the indicated line.

- (a)

```
example :  $\forall x : \mathbb{R}, x^2 \leq 2x \rightarrow \neg x > 2 :=$  by
  intro x hx hxpos
  -- What is the tactic state here?
```

Solution.

```
x :  $\mathbb{R}$ 
hx :  $x^2 \leq 2 * x$ 
hxpos :  $x > 0$ 
⊢ False
```

- (b) Give the tactic state at each of the lines (A) and (B).

```
example (n :  $\mathbb{Z}$ ) : Odd n  $\leftrightarrow$  4  $\mid$   $n^2 - 1 :=$  by
  constructor
  · intro ⟨k, hk⟩
    -- (A)
  · intro hn
    rw [← Int.not_even_iff_odd]
    intro h_even
    -- (B)
```

Solution. At (A):

```
case mp
n k :  $\mathbb{Z}$ 
hk :  $n = 2 * k + 1$ 
⊢ 4  $\mid$   $n^2 - 1$ 
```

At (B):

```
case mpr
  n : ℤ
  hn : 4 ∣ n ^ 2 - 1
h_even : Even n
  ⊢ False
```

(c)

```
example : ∀ y : ℝ, (∀ x : ℝ, x^2 - 4*x > y) → y < -4 := by
  intro y hx
  specialize hx 2
  -- What is the tactic state here?
```

Solution.

```
y : ℝ
hx : 2 ^ 2 - 4 * 2 > y
  ⊢ y < -4
```

(d)

```
example (n : ℤ) : ¬ Even (3*n) → ¬ Even n := by
  intro h1
  contrapose! h1 with h2
  -- What is the tactic state here?
```

Solution.

```
n : ℤ
h2 : Even n
  ⊢ Even (3 * n)
```

(e) The proof below has two `sorry`s. Give the tactic state at each of them.

```
example (x : ℝ) (h : x^2 - 1 = 0) : x = 1 ∨ x = -1 := by
  by_cases hx : x ≥ 0
  · left
    sorry
  · right
    sorry
```

Solution. At the first `sorry`:

```
case pos
  x : ℝ
  h : x ^ 2 - 1 = 0
hx : x ≥ 0
  ⊢ x = 1
```

At the second `sorry`:

```
case neg
  x : ℝ
  h : x ^ 2 - 1 = 0
hx : ¬x ≥ 0
⊢ x = -1
```

(f)

```
example : ∀ x : ℝ, x^2 + 9 ≥ 6*x := by
  intro x
  -- What is the tactic state here?
```

Solution.

```
x : ℝ
⊢ x ^ 2 + 9 ≥ 6 * x
```

(g)

```
example : ∃ x : ℝ, x^2 + 1 < 3*x := by
  use 1
  -- What is the tactic state here?
```

Solution.

```
case h
⊢ 1 ^ 2 + 1 < 3 * 1
```

Note that x has been replaced by 1 , and no longer appears.

(h)

```
example (n : ℕ) : 3^n ≥ 2*n + 1 := by
  induction' n with n ih
  · norm_num
  · -- What is the tactic state here?
```

Solution.

```
case succ
  n : ℕ
  ih : 3 ^ n ≥ 2 * n + 1
  ⊢ 3 ^ (n + 1) ≥ 2 * (n + 1) + 1
```

(i)

```
example (m n : ℤ) (h : m^2 = 2 * n^2) : n = 0 := by
  wlog hn : n ≥ 0
  · -- What is the tactic state here?
```

Solution.

```

case isFalse
  m n : ℤ
    h : m ^ 2 = 2 * n ^ 2
  this : ∀ (m n : ℤ), m ^ 2 = 2 * n ^ 2 → n ≥ 0 → n = 0
    hn : ¬n ≥ 0
    ⊢ n = 0

```

The hypothesis `this` is the whole lemma, for arbitrary `m` and `n`, but with the extra hypothesis `n ≥ 0`. To finish this case, apply it to `m` and `-n`.

3. Defining functions

- (a) Let $S(n) = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \cdots + \frac{1}{n(n+1)}$, so that $S(0) = 0$. Write a Lean definition of S as a function $S : \mathbb{N} \rightarrow \mathbb{Q}$.

Solution.

```

def S : ℕ → ℚ
| 0 => 0
| n + 1 => S n + 1 / ((n + 1) * (n + 2))

```

- (b) Prove in Lean that $S(n) = 1 - \frac{1}{n+1}$ for all n :

```

lemma S_formula (n : ℕ) : S n = 1 - 1 / (n + 1) := by

```

Solution.

```

induction' n with n ih
· rw [S]
  norm_num
· rw [S, ih]
  norm_num
  field_simp
  ring

```

- (c) Define a sequence by $a_0 = 0$, $a_1 = 3$, and $a_{n+2} = 7a_{n+1} - 10a_n$. Give a Lean definition of this sequence, as a function $a : \mathbb{N} \rightarrow \mathbb{Z}$.

Solution.

```

def a : ℕ → ℤ
| 0 => 0
| 1 => 3
| n + 2 => 7 * a (n + 1) - 10 * a n

```

- (d) Prove in Lean that $a_n = 5^n - 2^n$ for all n :

```

lemma a_formula (n : ℕ) : a n = 5^n - 2^n := by

```

Solution.

```

match n with
| 0 =>
  rw [a]
  norm_num
| 1 =>
  rw [a]
  norm_num
| n + 2 =>
  rw [a, a_formula n, a_formula (n + 1)]
  ring

```

4. One-line tasks

Each of these can be done with a single line of Lean.

- (a) Suppose the tactic state contains the hypothesis

```
h : ∃ u v : ℤ, u ^ 2 + v ^ 2 = 29
```

How do you get the following into the tactic state?

```

u v : ℤ
huv : u ^ 2 + v ^ 2 = 29

```

Answer. `obtain ⟨u, v, huv⟩ := h`

- (b) Suppose instead that the goal is $\vdash \exists u v : \mathbb{Z}, u^2 + v^2 = 29$. How do you change the goal to $\vdash 5^2 + 2^2 = 29$?

Answer. `use 5, 2`

- (c) Suppose the tactic state contains $h : \forall x : \mathbb{R}, x^2 - 4 * x + 3 \geq y$. How do you turn this into $h : 2^2 - 4 * 2 + 3 \geq y$?

Answer. `specialize h 2`

- (d) Suppose instead that the goal is $\vdash \exists x : \mathbb{R}, x^2 - 4 * x + 3 \geq y$. How do you change the goal to $\vdash 2^2 - 4 * 2 + 3 \geq y$?

Answer. `use 2`

- (e) Suppose the tactic state contains $h : x = 2 \vee x = -2$. How do you split the proof into two cases, one with $hx : x = 2$ and one with $hx : x = -2$?

Answer. `obtain hx | hx := h`

- (f) Suppose the goal is $\vdash \neg \forall x : \mathbb{R}, x^2 > x$. How do you change the goal to $\vdash \exists x, x^2 \leq x$?

Answer. `push Not`

- (g) Suppose the tactic state contains $h : \neg \text{Even } n$, where $n : \mathbb{Z}$. How do you change this to $h : \text{Odd } n$?

Answer. `rw [Int.not_even_iff_odd] at h`

- (h) Suppose the goal is $\vdash n = 0$, where $n : \mathbb{Z}$. How do you begin a proof by contradiction, so that the goal becomes $\vdash \text{False}$ and the tactic state contains $hn : n \neq 0$?

Answer. `by_contra! hn`

5. Short proofs

Write a Lean proof of each lemma.

- (a) Complete the following proof by filling in each blank. Use the `calc` block as given.

```
example {a b : ℝ} (h1 : a - b = 4) (h2 : a * b = 1) : (a + b)^2 = 20 := by
  calc
    (a + b)^2 = (a - b)^2 + 4 * (a * b) := by ____
    _ = 4^2 + 4 * 1 := by ____
    _ = 20 := by ____
```

Solution. The three blanks are `ring`, `rw [h1, h2]`, and `norm_num`.

- (b) Complete the following proof by filling in each blank. Use the `calc` block as given.

```
example {x : ℝ} (hx : x ≥ 3) : x^2 - 2*x ≥ 3 := by
  calc
    x^2 - 2*x = ____ := by ____
    _ ≥ ____ := by rel [hx]
    _ = 3 := by norm_num
```

Solution. Complete the square:

```
calc
  x^2 - 2*x = (x - 1)^2 - 1 := by ring
  _ ≥ (3 - 1)^2 - 1 := by rel [hx]
  _ = 3 := by norm_num
```

- (c) You may use the library theorem `mul_eq_zero : a * b = 0 ↔ a = 0 ∨ b = 0`.

```
example {x : ℝ} (h : x^2 = 4*x) : x = 0 ∨ x = 4 := by
```

Solution.

```
have h' : x * (x - 4) = 0 := by
  calc
    x * (x - 4) = x^2 - 4*x := by ring
    _ = 0 := by rw [h]; ring
obtain h0 | h4 := mul_eq_zero.mp h'
· left
  exact h0
· right
  linarith
```

(d)

```
example {n : ℤ} (hn : Odd n) : Even (n^2 + 3) := by
```

Solution. Recall that `Even m` means $\exists r, m = r + r$.

```
obtain ⟨k, hk⟩ := hn
use 2*k^2 + 2*k + 2
rw [hk]
ring
```

(e)

```
example {a b : ℤ} (ha : 3 ∣ a) (hb : 3 ∣ b) : 3 ∣ a^2 + 2*b := by
```

Solution.

```
obtain ⟨k, hk⟩ := ha
obtain ⟨l, hl⟩ := hb
use 3*k^2 + 2*1
rw [hk, hl]
ring
```

(f)

```
example : ¬ ∀ x : ℝ, x^2 ≥ 2*x := by
```

Solution.

```
intro h
specialize h 1
norm_num at h
```

Alternatively: push `Not`, then use `1`, then `norm_num`.

(g) Complete the following proof by induction by filling in each blank.

```
example (n : ℕ) : 4^n ≥ 3*n + 1 := by
  induction' n with n ih
  · ----
  · calc
    4^(n+1) = 4 * 4^n := by ----
    _ ≥ 4 * (3*n + 1) := by ----
    _ ≥ 3*(n+1) + 1 := by ----
```

Solution. The four blanks are `norm_num`, `ring`, `rel [ih]`, and `linarith`.