SIMONS FOUNDATION

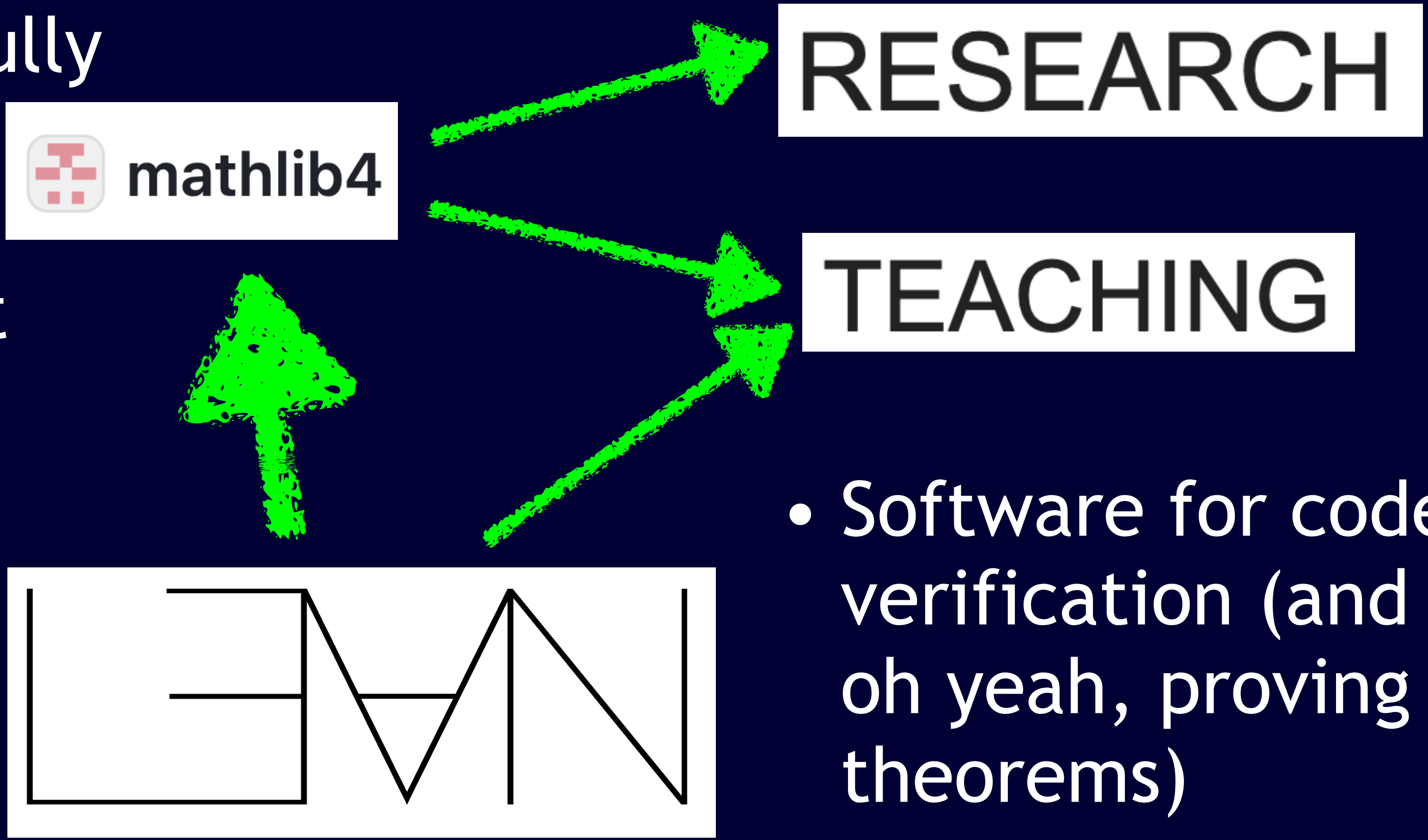
Workshop on LEAN

Alex Kontorovich

RUTGERS UNIVERSITY / IAS

SIMONS FOUNDATION Workshop on LEAN

- “Lean” means different things in contexts, so let’s clarify:
- Massive, beautifully synthesized, interconnected library, making it possible to do research/teach formally



- Software for code verification (and oh yeah, proving theorems)

Analogy: Chess

TEACHING

Imagine a world where we discuss chess games like this:

“Here goes the game: 1. Nf3 Nf6 2. c4 g6 3. Nc3 Bg7 4. d4 0-0 5. Bf4 d5 (this is a transposed Grünfeld Defence) 6. Qb3 dxc4 7. Qxc4 c6 8. e4 Nbd7 9. Rd1 Nb6 10. Qc5 Bg4 11. Bg5 Na4!! Holy cow, what a move!! Can you believe he did that?…”

If you’re a chess aficionado, you have no trouble reading this and converting algebraic notation for moves into an actual game board in your mind’s eye.

For the rest of us, this is difficult and painful to do (until sufficiently practiced).

But this is exactly how we currently teach math!

TEACHING

But this is exactly how we currently teach math!

Example: Theorem: $\sqrt{2} \notin \mathbb{Q}$.

Proof: If not, then it's equal to a fraction in lowest terms, and we can square both sides and cross multiply to get $p^2 = 2q^2$.
Then p must be even, …

At every move, the "mathematical game board" (what are the assumptions and what is to be proved) is changing!

This is effortless (System I) for you all to track,
but very difficult (System II) for beginners.

Let's look at the game boards:

TEACHING

But this is exactly how we currently teach math!

Example: Theorem: $\sqrt{2} \notin \mathbb{Q}$.

Proof: If not, then it's equal to a fraction in lowest terms, and we can square both sides and cross multiply to get $p^2 = 2q^2$.

Then p must be even, …

Let's look at the game boards:

```
⊢ √2 ∉ Set.range Rat.cast
```

TEACHING

But this is exactly how we currently teach math!

Example: Theorem: $\sqrt{2} \notin \mathbb{Q}$.

```
⊢ √2 ∉ Set.range Rat.cast
```

Proof: If not, then it's equal to a fraction in lowest terms, and we can square both sides and cross multiply to get $p^2 = 2q^2$.

Then p must be even, …

Let's look at the game boards:

```
h : √2 ∈ Set.range Rat.cast
⊢ False
```

TEACHING

But this is exactly how we currently teach math!

Example: Theorem: $\sqrt{2} \notin \mathbb{Q}$.

```
⊢ √2 ∉ Set.range Rat.cast
```

```
h : √2 ∈ Set.range Rat.cast
⊢ False
```

Proof: If not, then it's equal to a fraction in lowest terms, and we can square both sides and cross multiply to get $p^2 = 2q^2$.

Then p must be even, …

Let's look at the game boards:

```
h1 : ∃ p q, q ≠ 0 ∧ p.gcd q = 1 ∧
√2 = ↑p / ↑q
⊢ False
```

TEACHING

But this is exactly how we currently teach math!

Example: Theorem: $\sqrt{2} \notin \mathbb{Q}$.

```
⊢ √2 ∉ Set.range Rat.cast
```

```
h : √2 ∈ Set.range Rat.cast
⊢ False
```

```
h1 : ∃ p q, q ≠ 0 ∧ p.gcd q = 1 ∧
√2 = ↑p / ↑q
⊢ False
```

Proof: If not, then it's equal to a fraction in lowest terms, and we can square both sides and cross multiply to get $p^2 = 2q^2$.

Then p must be even, …

Let's look at the game boards:

```
h2 : 2 = ↑p ^ 2 / ↑q ^ 2
⊢ False
```

TEACHING

But this is exactly how we currently teach math!

Example: Theorem: $\sqrt{2} \notin \mathbb{Q}$.

```
⊢ √2 ∉ Set.range Rat.cast
```

```
h : √2 ∈ Set.range Rat.cast
⊢ False
```

```
h1 : ∃ p q, q ≠ 0 ∧ p.gcd q = 1 ∧
√2 = ↑p / ↑q
⊢ False
```

Proof: If not, then it's equal to a fraction in lowest terms, and we can square both sides and cross multiply to get $p^2 = 2q^2$.

```
h2 : 2 = ↑p ^ 2 / ↑q ^ 2
⊢ False
```

Then p must be even, …

Let's look at the game boards:

```
p q : ℤ
hq_ne_zero : q ≠ 0
hgcd : p.gcd q = 1
h3 : p ^ 2 = 2 * q ^ 2
⊢ False
```

TEACHING

But this is exactly how we currently teach math!

Example: Theorem: $\sqrt{2} \notin \mathbb{Q}$.

```
⊢ √2 ∉ Set.range Rat.cast
```

```
h : √2 ∈ Set.range Rat.cast
⊢ False
```

```
h1 : ∃ p q, q ≠ 0 ∧ p.gcd q = 1 ∧
√2 = ↑p / ↑q
⊢ False
```

Proof: If not, then it's equal to a fraction in lowest terms, and we can square both sides and cross multiply to get $p^2 = 2q^2$.

```
h2 : 2 = ↑p ^ 2 / ↑q ^ 2
⊢ False
```

```
p q : ℤ
hq_ne_zero : q ≠ 0
hgcd : p.gcd q = 1
h3 : p ^ 2 = 2 * q ^ 2
⊢ False
```

Then p must be even, …

Let's look at the game boards:

We would never write this all out when teaching

- Take way too long (cumbersome)
- Unnecessary; we all learned without it.
- Eventually: learning to make those mental images is vital to being able to do math at a high level. But not from the start!

Q: How to force people to use Lean?

TEACHING

Q: How to force people to use Lean? A: Don't!

- 1978: Knuth releases TeX
- Every mathematician handwrites papers, gives to secretary to typeset, waits a few weeks/months, hopes it's faithful
- 1980's: Spivak pushes for AMSTeX, still few people use it
- 1985-1990's: LaTeX comes out, lots of macros, by 2000 *very* few mathematicians (Sarnak, Bourgain, Iwaniec…) still handwrite own papers. (Now AI can do it for them…)
- Also: overleaf (free web-app).
- The "Knuth constant" (= time to typeset ($ \ { …) / time to handwrite) went below 1. Everybody switched voluntarily!

- Same can happen with Lean!

TEACHING

- Same can happen with Lean!
- de Bruijin constant (lines of formal code / lines of natural proof) is wrong metric! (LLMs can produce lots of lines of code very quickly; no longer a proxy for Time!)
- Instead, measure: Time to formalize paper in Lean, letting kernel check correctness / Time to typeset paper in LaTeX, rechecking each lemma again and again for corre
- The instant that ratio gets below 1 through great, efficient libraries, automation/tactics/LLMs, Lean as a free web-app (GitHub codespaces, Gitpod, live.lean-lang.org), etc, etc…, everyone will start working formally, voluntarily!

TEACHING

everyone will start working formally, voluntarily!

- We’re now at the stage (thanks to Mathlib, LLMs) where people with a modicum of understanding of how Lean works can already meaningfully play with formalization.
    - Key idea:

(Quasi)-Autoformalization

(Quasi)-Autoformalization

IDEA

INFORMAL STATEMENT

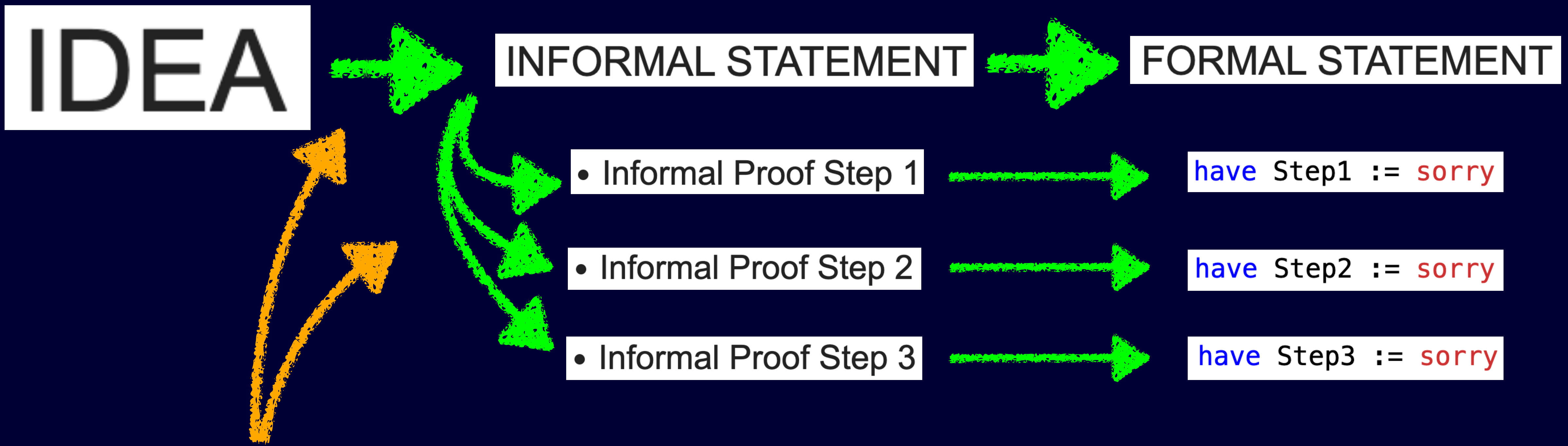
FORMAL STATEMENT

- Informal Proof Step 1
- Informal Proof Step 2
- Informal Proof Step 3

```
have Step1 := sorry
have Step2 := sorry
have Step3 := sorry
```

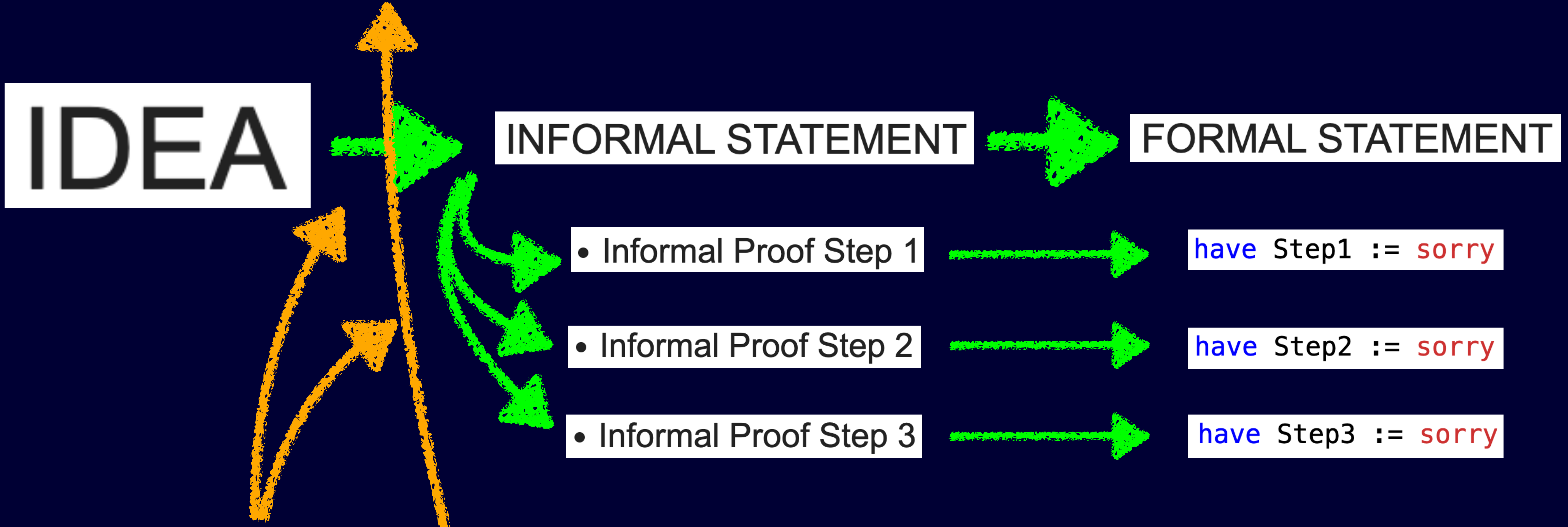
- This is where the dialogue with the computer can take place, at the level of "ideas"!

(Quasi)-Autoformalization



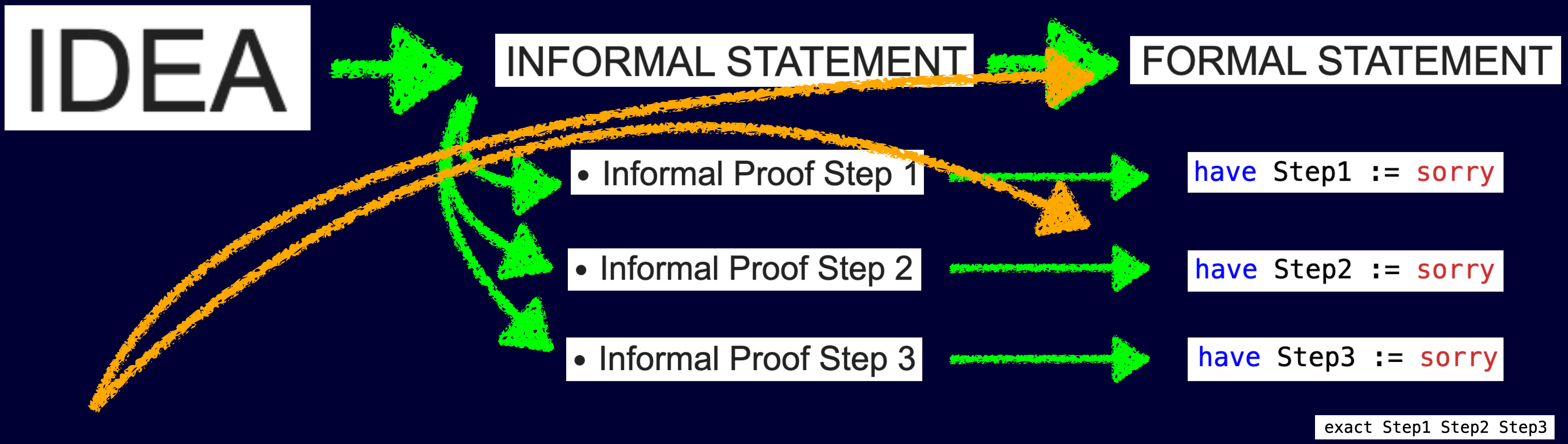
- Then an LLM can try to convert ideas into natural language statements, and can propose natural language proof sketch

(Quasi)-Autoformalization



- Then an LLM can try to convert ideas into natural language statements, and can propose natural language proof sketch
- At every stage, human can intervene!

(Quasi)-Autoformalization



- Then an LLM formalizes informal statement *and* uses informal proof as scaffold for proposed formal proof!

(Quasi)-Autoformalization

- Then an LLM formalizes informal statement *and* uses informal proof as scaffold for proposed formal proof!
- Let’s try it out!
- Suggestion: put your computer (iPad/phone) away and just follow along on paper.
- After “lecture” comes time to try it yourself.