Draft, Sketch, and Prove

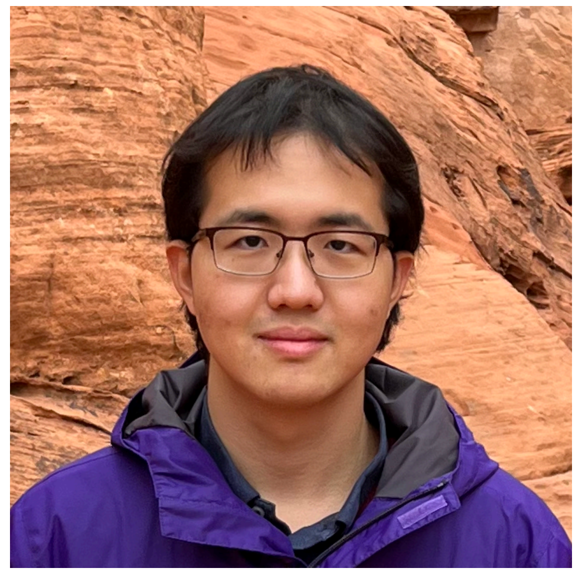
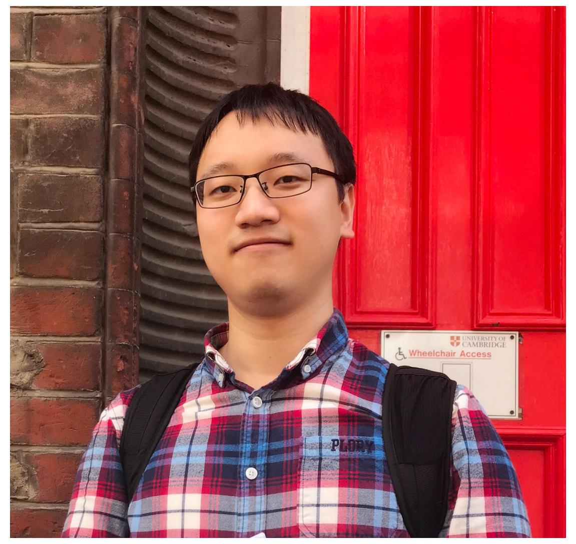
Sean Welleck

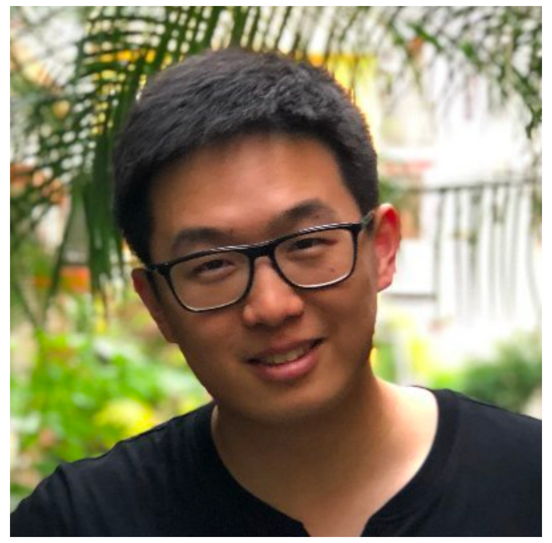
Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs.

Jiang*, Welleck*, Zhou*, Lacroix, Liu, Li, Jamnik, Lample+, Wu+

ICLR 2023



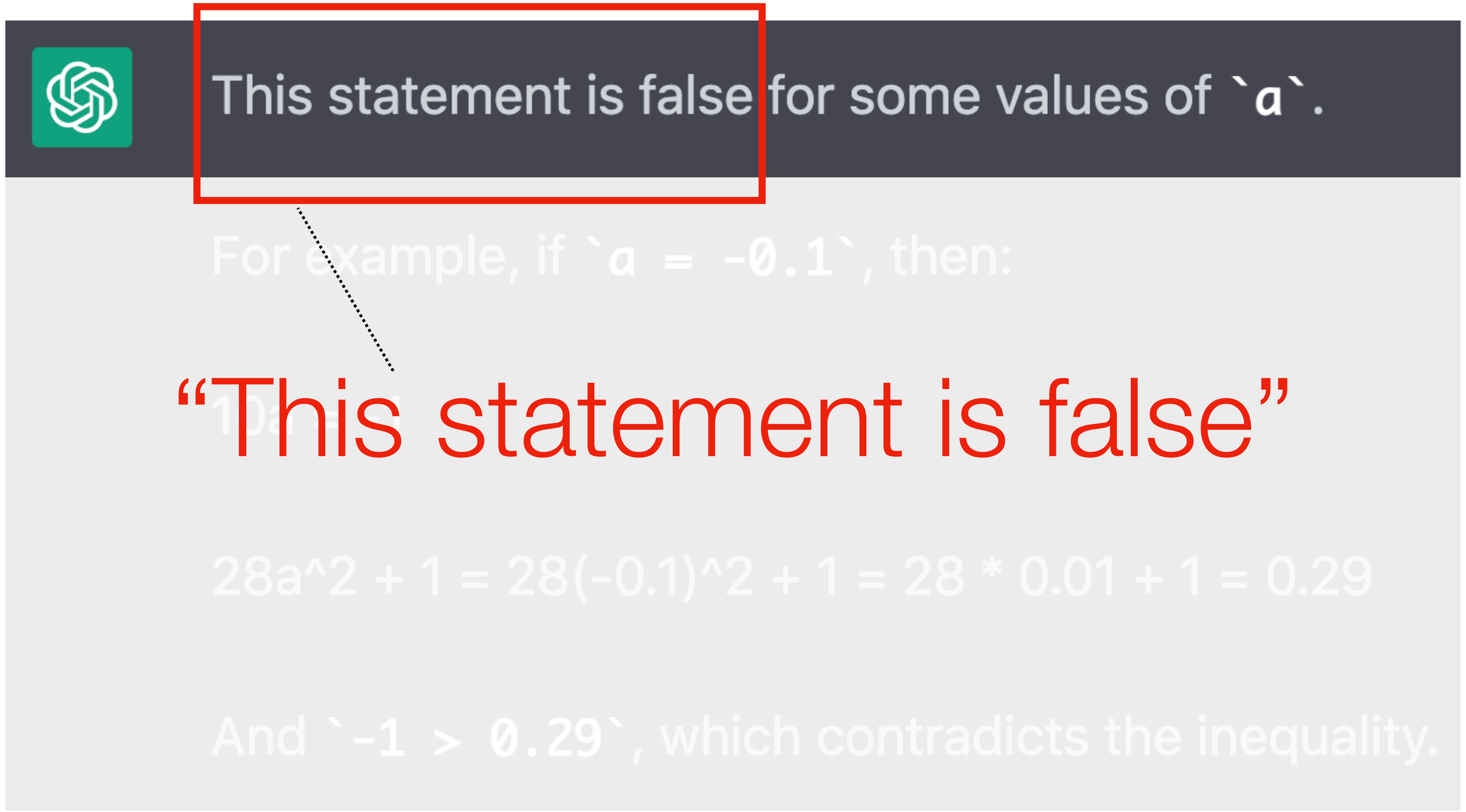




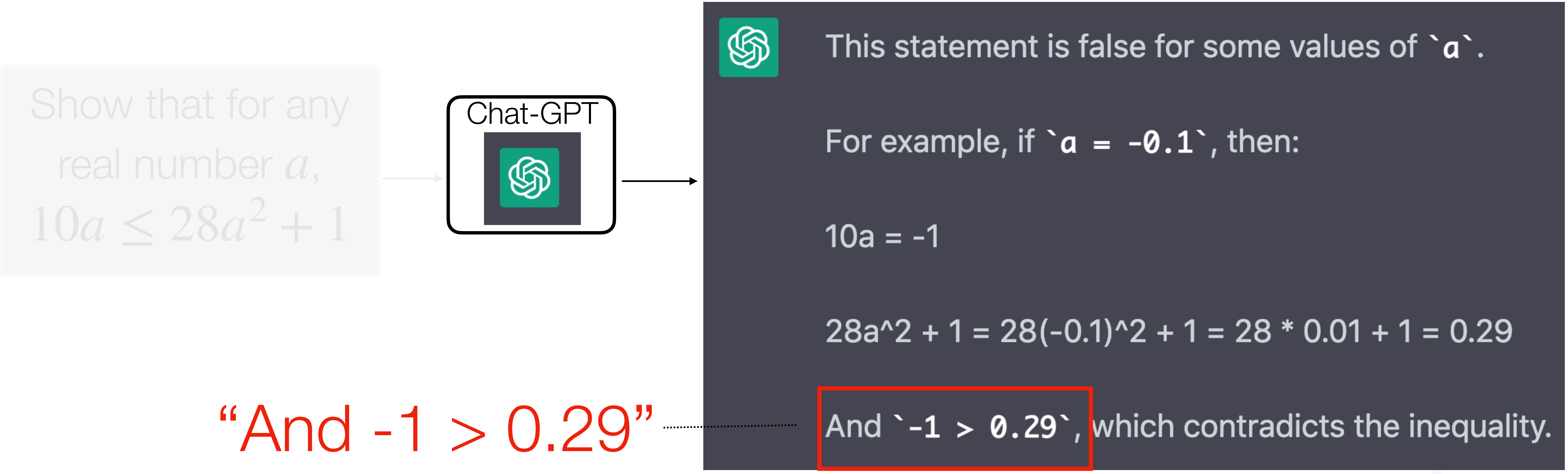
Informal mathematics

Show that for any real number a,
$$10a \leq 28a^2 + 1$$

Chat-GPT



Informal mathematics



Formalized mathematics

- Translate mathematics into “code”, grounded in logic
  - Verified correctness
  - New ways of collaborating, teaching, thinking

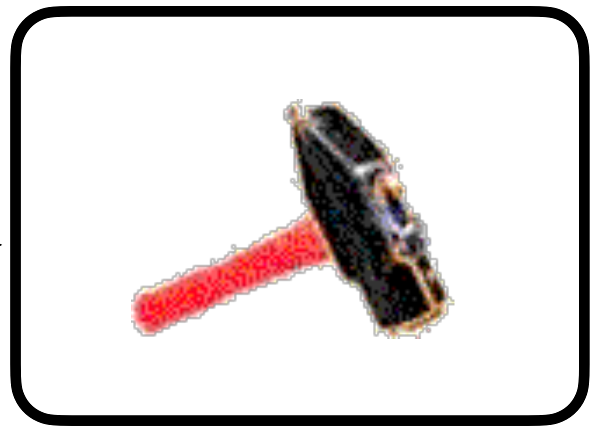
Declarative proof

```
have c1: “10*280 = n*40”
using assms
    by (smt (z3) prod_gcd_lcm_nat)
then have c2: “n = 10*280/40”
    by auto
then show ?thesis
    by auto
```

Isabelle

Proof automation

```
have c1: “10*280 = n*40”
using assms
```

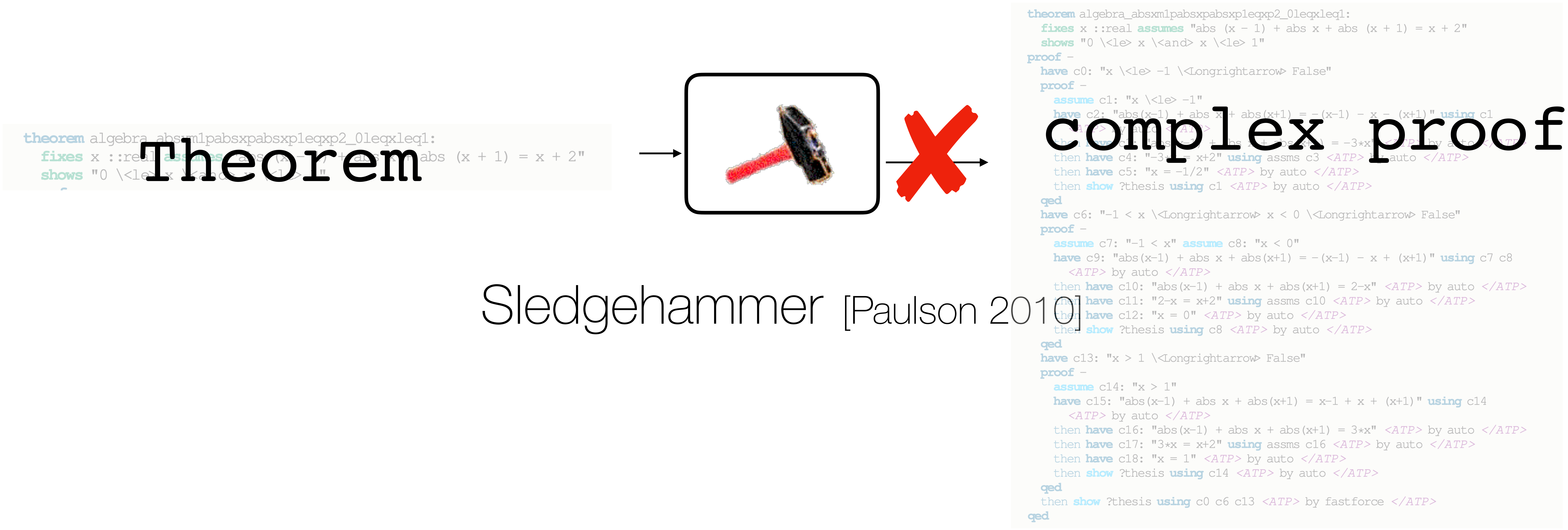


```
have c1: “10*280 = n*40”
using assms
by (smt (z3) prod_gcd_lcm_nat)
```

Sledgehammer [Paulson 2010]

- calls out to external provers

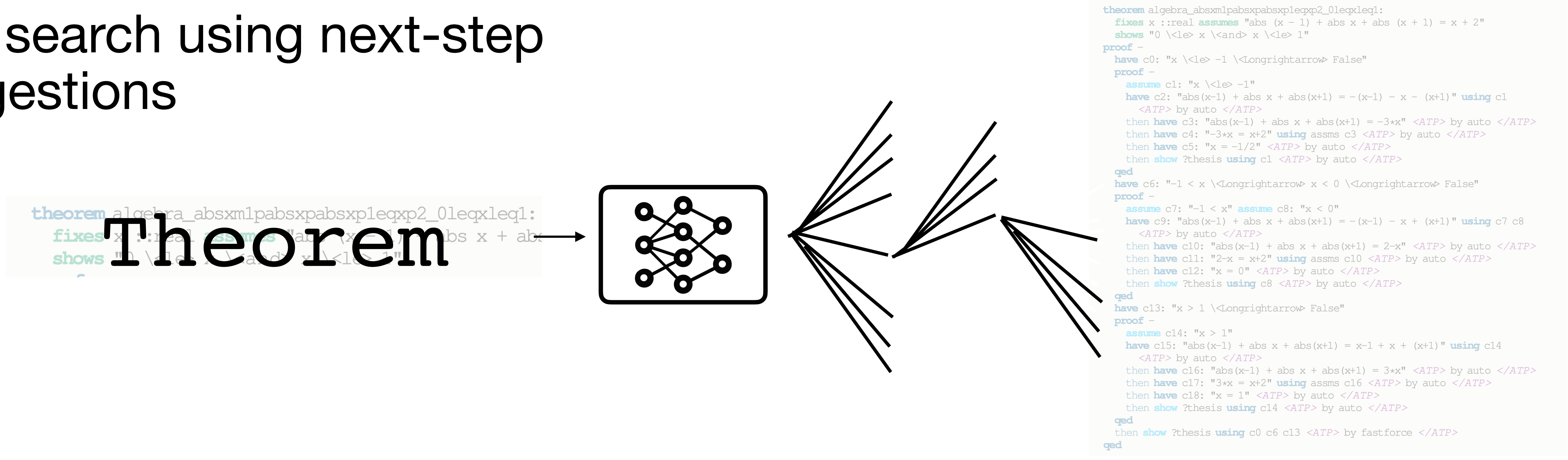
Proof automation



Sledgehammer [Paulson 2010]

Neural proof automation

- Train neural network on (context, next-step) pairs
- Tree search using next-step suggestions



E.g. [Yang & Deng 2019, Polu & Sutskever 2020, Han et al 2022]

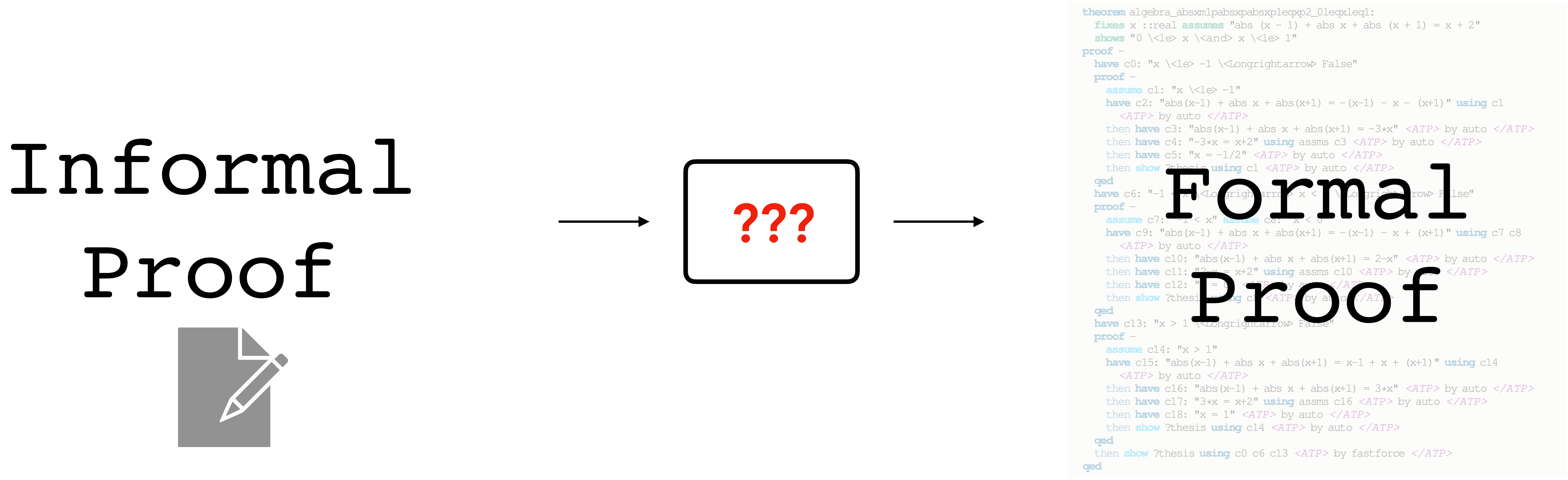
Neural proof automation

- Train neural network on (context, next-step) pairs
- Tree search using next-step candidates

- Limited amount of formal data
- Large search space -> expensive
  - Smaller network
  - Smaller # of candidates

E.g. [Yang & Deng 2019, Polu & Sutskever 2020, Han et al 2022]

Informal → formal proof automation



- Leverages informal data
- Cuts down the search space

Challenge 1 : different levels of abstraction

- **Solution**: translate into *proof sketches*

We know that $\gcd(a, b) \cdot \mathrm{lcm}(a, b) = ab$, hence $10 \cdot 280 = n \cdot 40$.

Then $n = 10 \cdot 280/40 = 70$

completing the proof. ■

→

```
have c1: "10*280 = n*40"
using assms
    <proof>
then have c2: "n = 10*280/40"
    <proof>
then show ?thesis
    <proof>
```

Informal Proof

Formal Proof *Sketch*

Challenge 1 : different levels of abstraction

- **Solution**: translate into *proof sketches*
- In-line comments show the alignment between the informal and formal proof.

Informal Statement: Show that for any real number a, $10a \leq 28a^2 + 1$.

Informal Proof:
It suffices to show $0 \leq 28a^2 - 10a + 1$. First, consider completing the square for $28a^2 - 10a$ and observe that $\left(a - \frac{5}{28}\right)^2 = a^2 - \frac{10}{28}a + (5/28)^2$. Since $0 \leq \left(a - \frac{5}{28}\right)^2$, we get $0 \leq a^2 - \frac{10}{28}a + (5/28)^2$. Multiplying by 28 and simplifying gives $0 \leq 28a^2 - 10a + (25/28)$. Since $25/28 < 1$, the result follows.

Formal Statement:

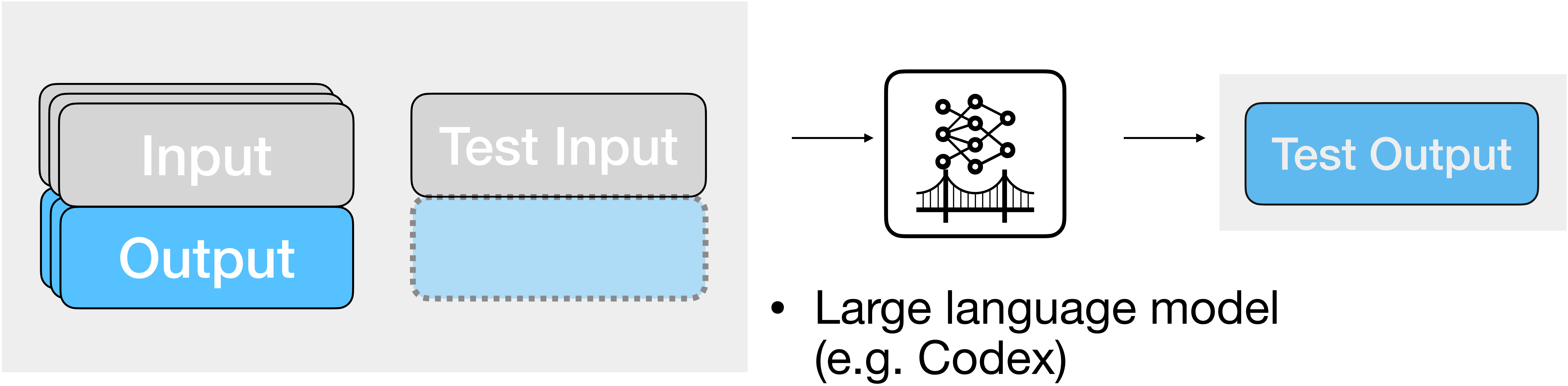
```
theorem algebra_binomnegdiscrineq_10alt28asqp1:
  fixes a :: real
  shows "10 * a ≤ 28 * a^2 + 1"
```

Formal Proof Sketch:

```
proof - (* it suffices to show 0 <= 28a^2 - 10a + 1 *)
  have c0: "0 ≤ 28a^2 - 10a + 1"
  proof - (* observe that (a - (5/28))^2 = a^2 - (10/28)a + (5/28)^2 *)
    have c1: "(a - (5/28))^2 = a^2 - 10/28a + (5/28)^2" <···>
    (* we get 0 <= a^2 - (10/28)a + (5/28)^2 *)
    have c2: "0 ≤ a^2 - 10/28a + (5/28)^2" using c1 <···>
    (* Multiplying by 28 and simplifying gives 0 <= 28a^2 - 10a + (25/28) *)
    have c3: "0 ≤ 28a^2 - 10a + 28((5/28)^2)" using c2 <···>
    have c4: "0 ≤ 28a^2 - 10a + 28((5/28)*(5/28))" using c3 <···>
    have c5: "0 ≤ 28a^2 - 10a + (25/28)" using c4 <···>
    (* Since 25/28 < 1, the result follows. *)
    show ?thesis using c5 <···>
  qed
  show ?thesis <···>
qed
```

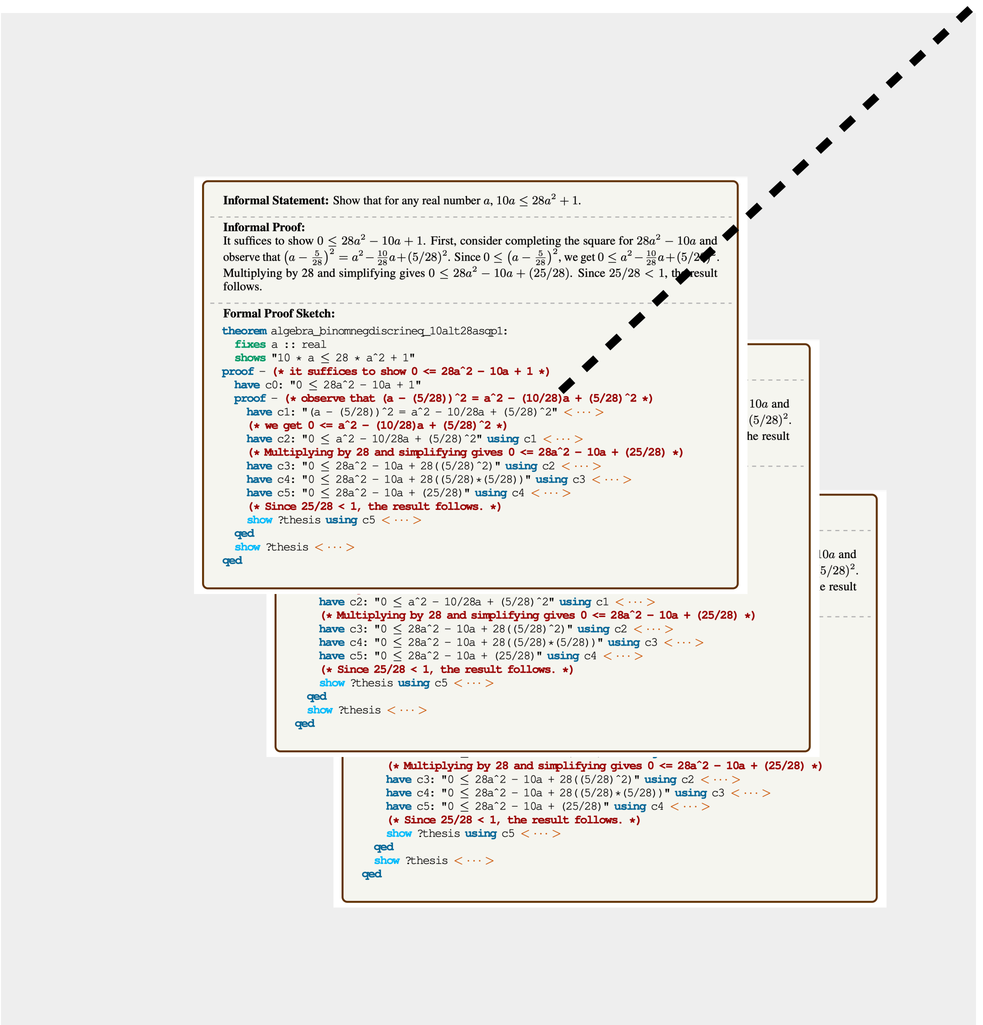
Challenge 2 : no parallel data

- **Solution**: few-shot in-context learning



- Large language model (e.g. Codex)

~~Challenge 2~~ : few-shot sketching



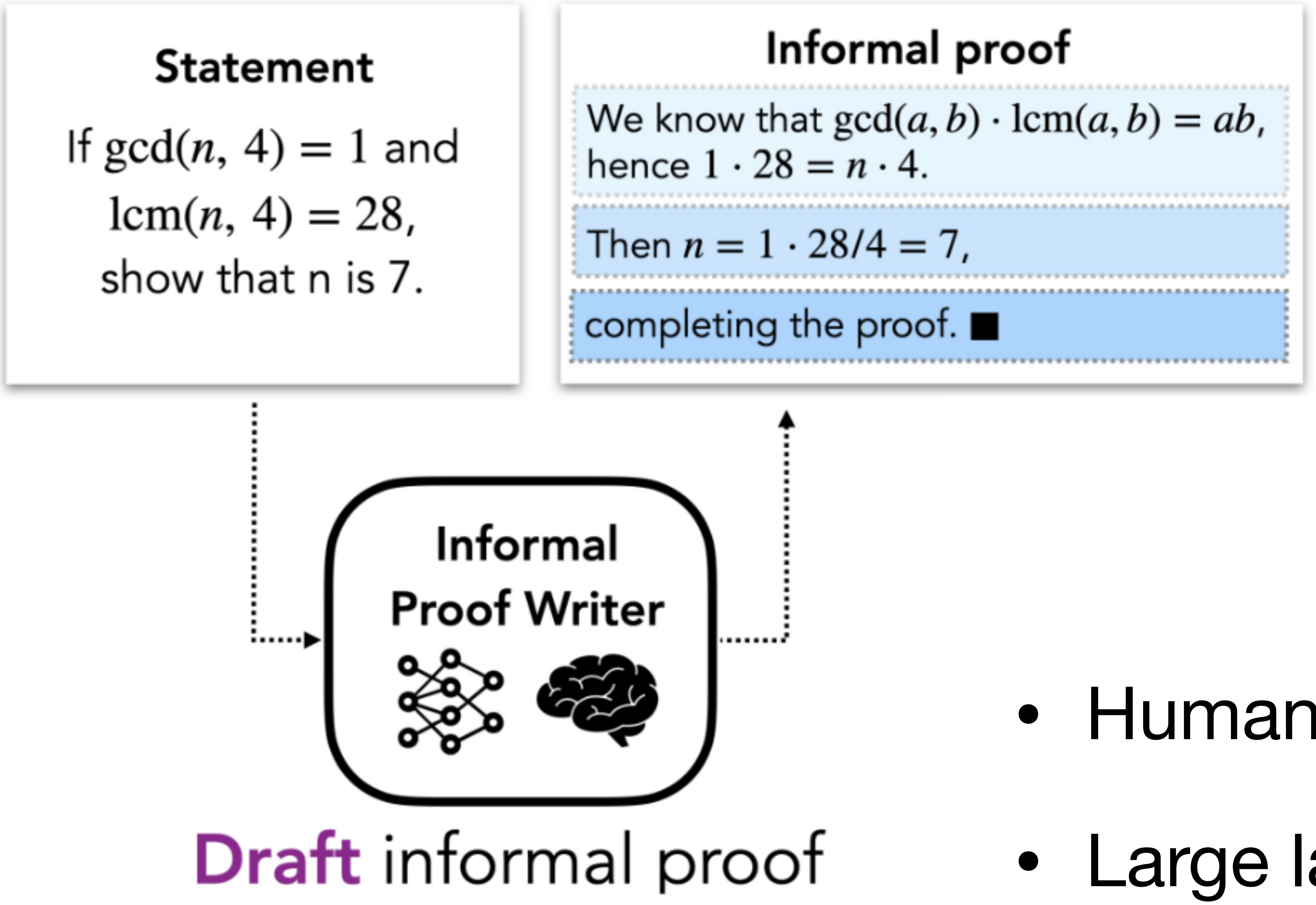
Informal Statement: Show that for any real number a, $10a \leq 28a^2 + 1$.

Informal Proof:
It suffices to show $0 \leq 28a^2 - 10a + 1$. First, consider completing the square for $28a^2 - 10a$ and observe that $\left(a - \frac{5}{28}\right)^2 = a^2 - \frac{10}{28}a + (5/28)^2$. Since $0 \leq \left(a - \frac{5}{28}\right)^2$, we get $0 \leq a^2 - \frac{10}{28}a + (5/28)^2$. Multiplying by 28 and simplifying gives $0 \leq 28a^2 - 10a + (25/28)$. Since $25/28 < 1$, the result follows.

Formal Proof Sketch:

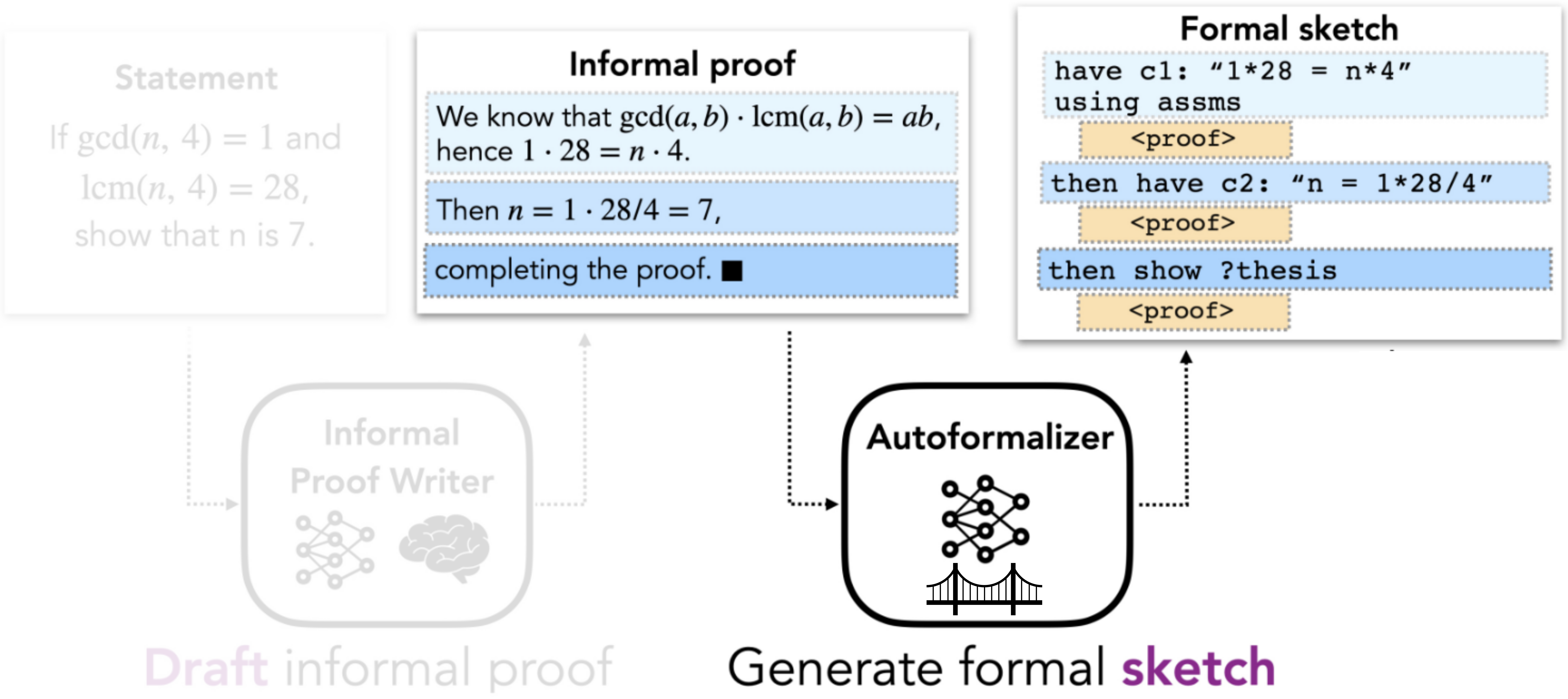
```
theorem algebra_binomnegdiscrineq_10alt28asqp1:
  fixes a :: real
  shows "10 * a ≤ 28 * a^2 + 1"
proof - (* it suffices to show 0 <= 28a^2 - 10a + 1 *)
  have c0: "0 ≤ 28a^2 - 10a + 1"
  proof - (* observe that (a - (5/28))^2 = a^2 - (10/28)a + (5/28)^2 *)
    have c1: "(a - (5/28))^2 = a^2 - 10/28a + (5/28)^2" <···>
    (* we get 0 <= a^2 - (10/28)a + (5/28)^2 *)
    have c2: "0 ≤ a^2 - 10/28a + (5/28)^2" using c1 <···>
    (* Multiplying by 28 and simplifying gives 0 <= 28a^2 - 10a + (25/28) *)
    have c3: "0 ≤ 28a^2 - 10a + 28((5/28)^2)" using c2 <···>
    have c4: "0 ≤ 28a^2 - 10a + 28((5/28)*(5/28))" using c3 <···>
    have c5: "0 ≤ 28a^2 - 10a + (25/28)" using c4 <···>
    (* Since 25/28 < 1, the result follows. *)
    show ?thesis using c5 <···>
  qed
  show ?thesis <···>
qed
```

Draft, sketch, prove



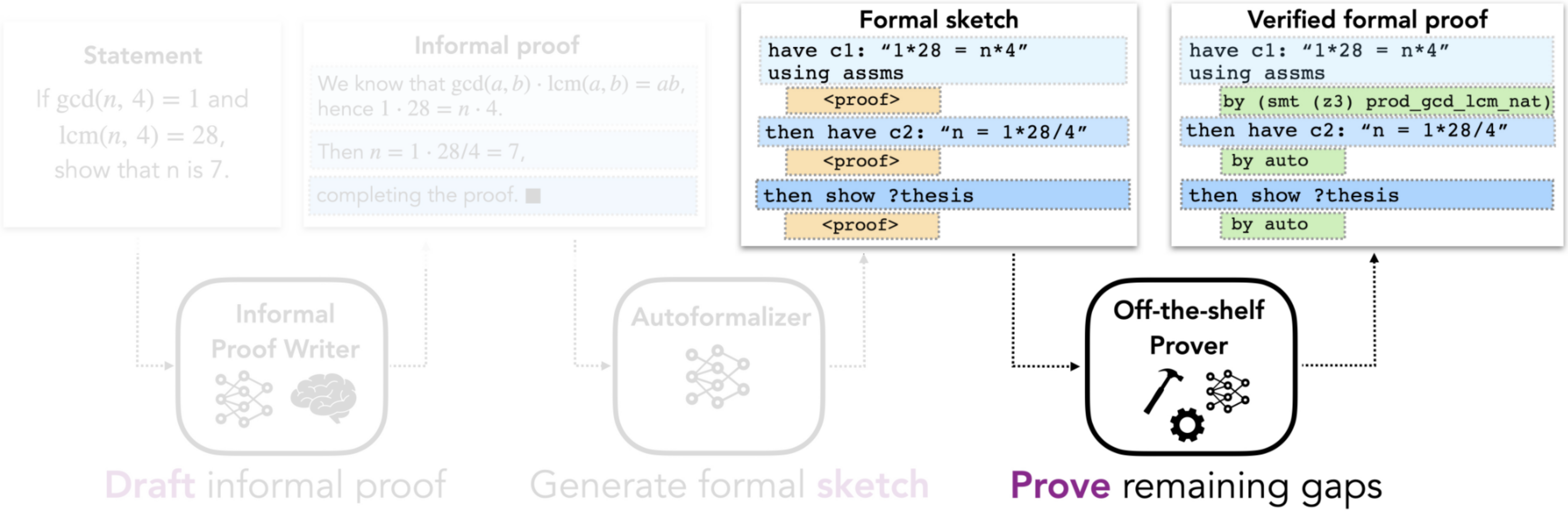
- Human-written proof
- Large language model (Minerva)

Draft, **sketch**, prove



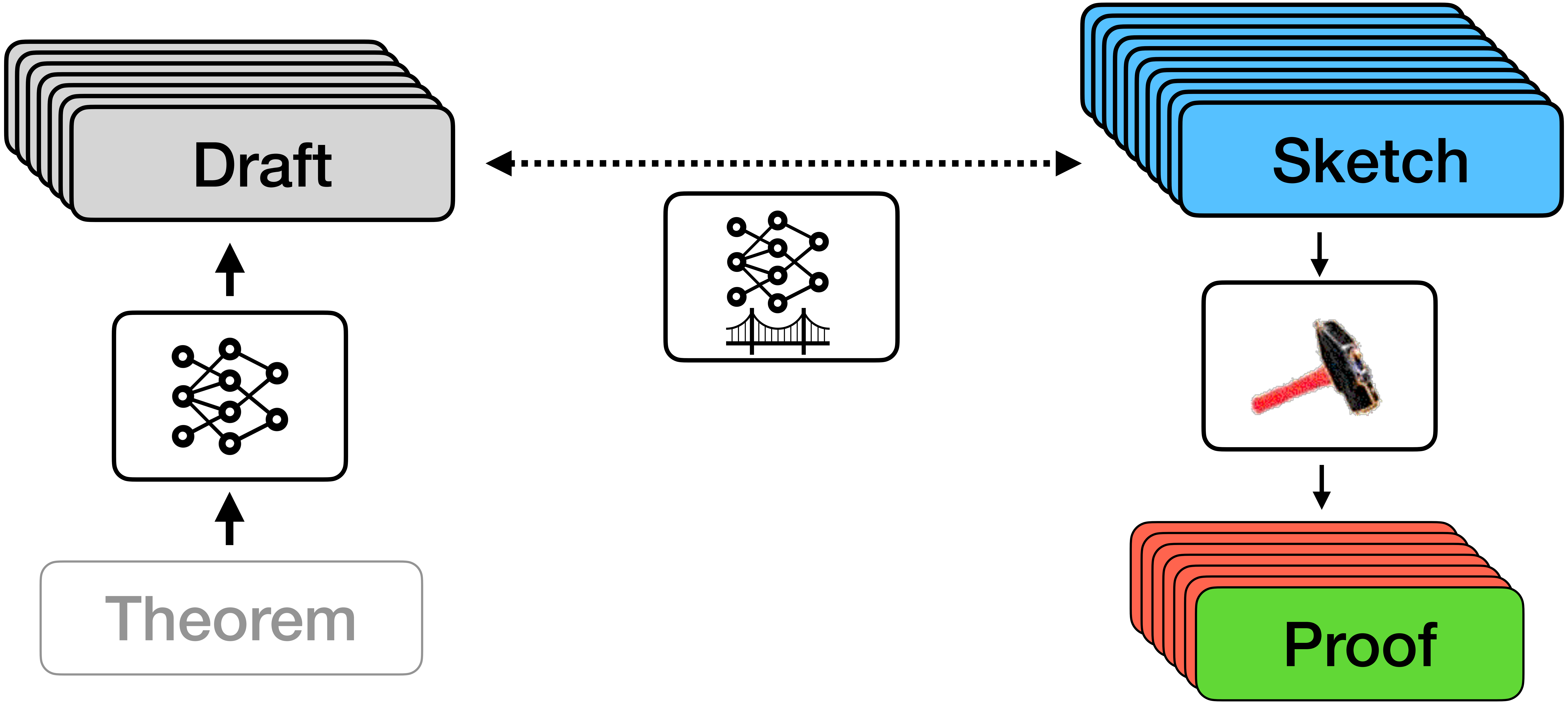
- Large language model (Codex)

Draft, sketch, **prove**



- Sledgehammer + heuristics

Proof search with draft, sketch, prove



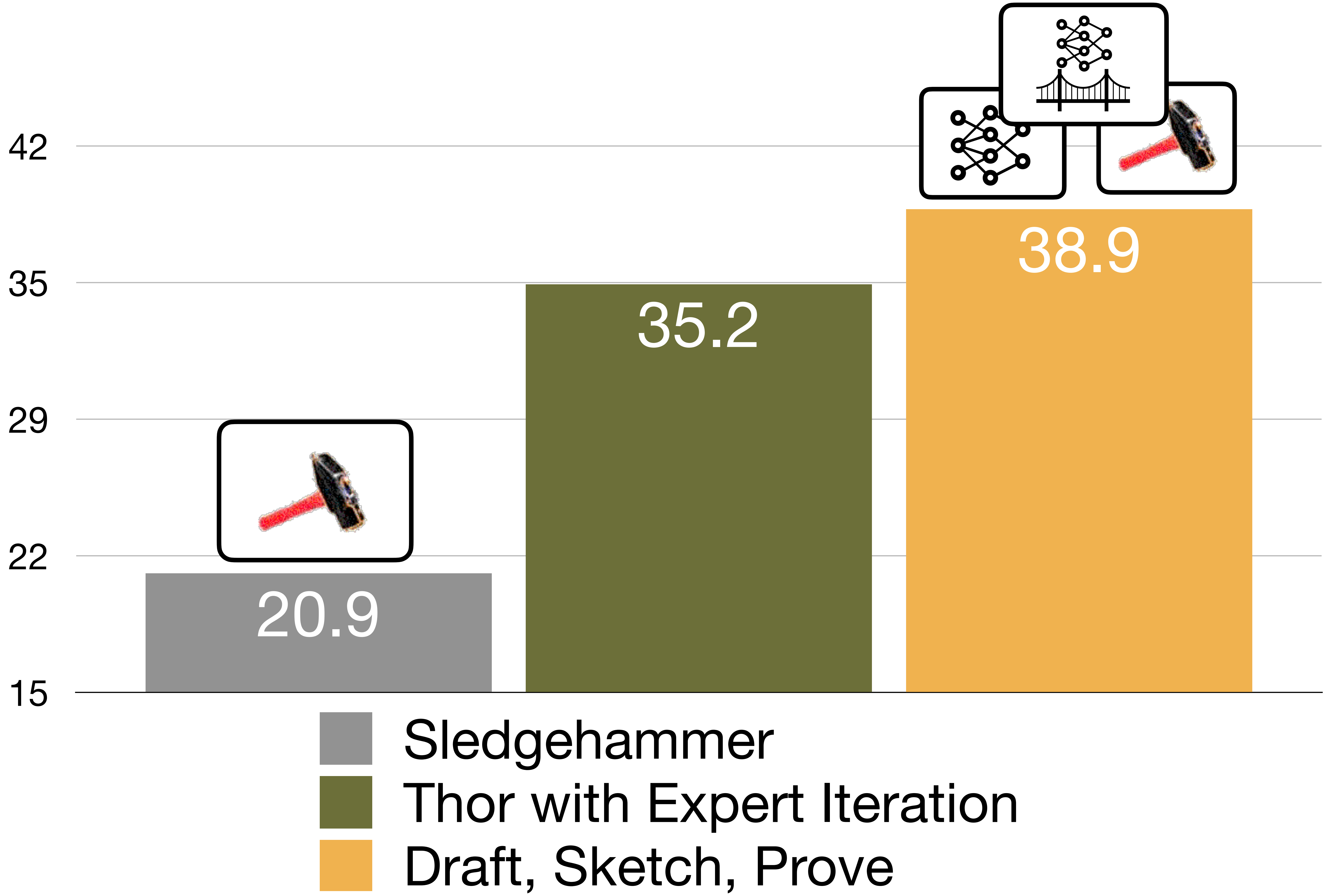
Experiments | miniF2F [Zheng et al 2022]

- 488 high-school competition problems (AMC, AIME, IMO, …)
  - 244 validation, 244 test
- Isabelle proof assistant
- Baselines:
  - Sledgehammer
  - THOR with Expert Iteration [Wu et al 2022]

Experiments | miniF2F

- Human informal draft: 1 human draft x 100 sketches/draft
- LLM informal drafts: 100 drafts x 1 sketch / draft
  - Prompting:
    - Write 20 sketch examples from miniF2F
    - Sample 3 examples and put in prompt

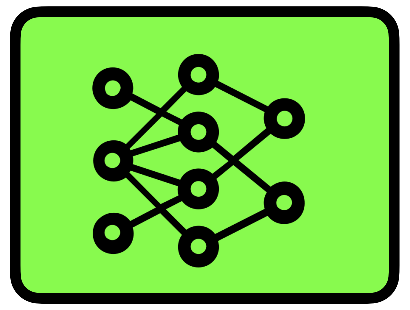
Top-level results



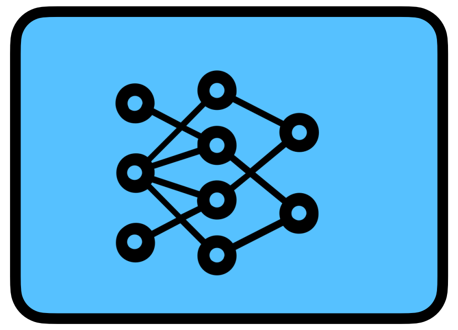
Minerva 540B drafts, 100 sketches

International Math Olympiad
1959 P.1

Draft
(Minerva)



Sketch
(Codex)



Formal Prover
(Sledgehammer)

Statement: Prove that the fraction $\frac{21n+4}{14n+3}$ is irreducible for every natural number n.

Informal Proof (Minerva 62B):
We must show that $\gcd(21n+4, 14n+3) = 1$.
The Euclidean algorithm gives

$$\begin{aligned} 21n+4 &= 1 \cdot (14n+3) + 7n + 1 \\ 14n+3 &= 2 \cdot (7n+1) + 1. \end{aligned}$$

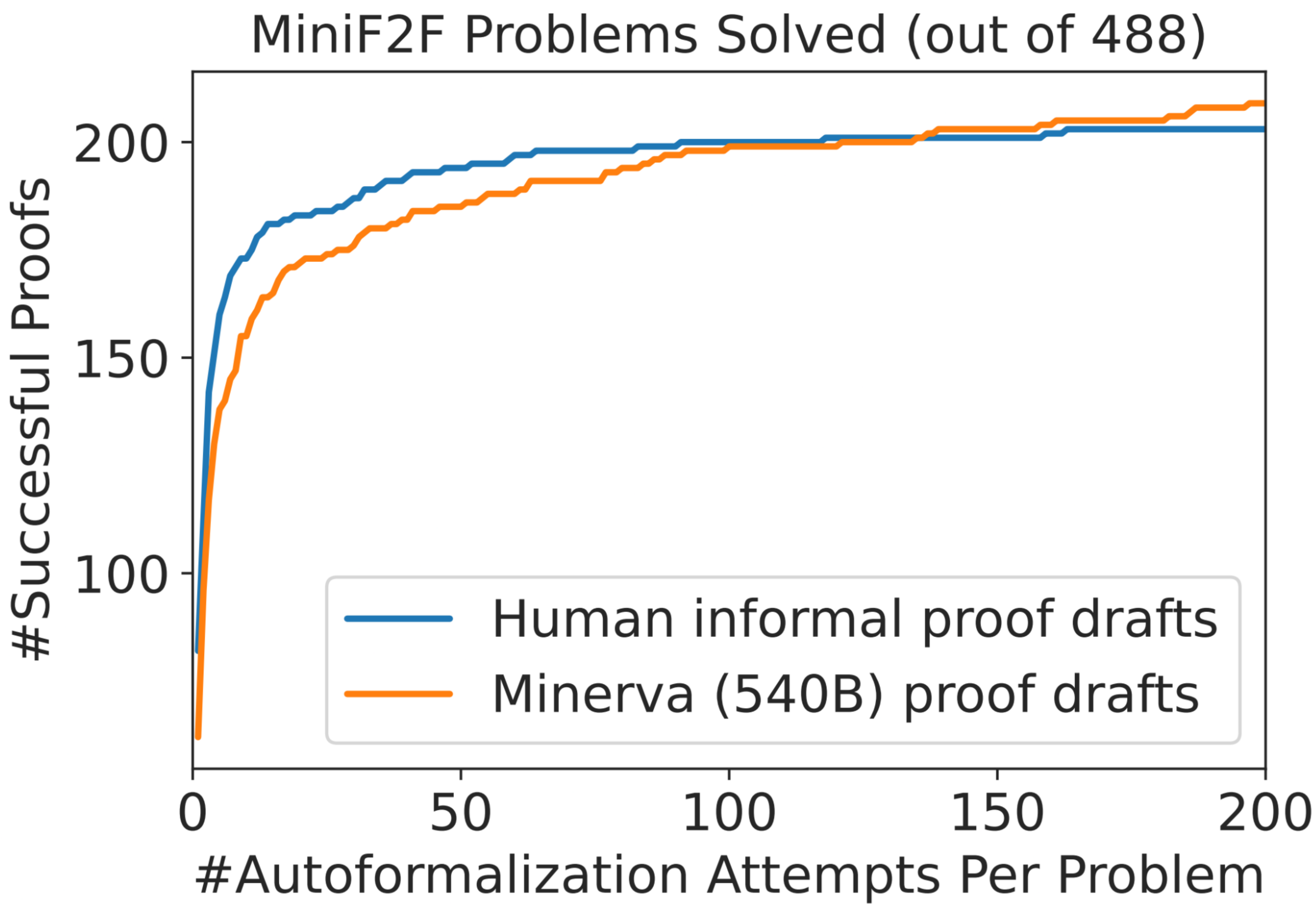
Since $\gcd(7n+1, 1) = 1$, we have $\gcd(21n+4, 14n+3) = 1$.

Formal Proof:

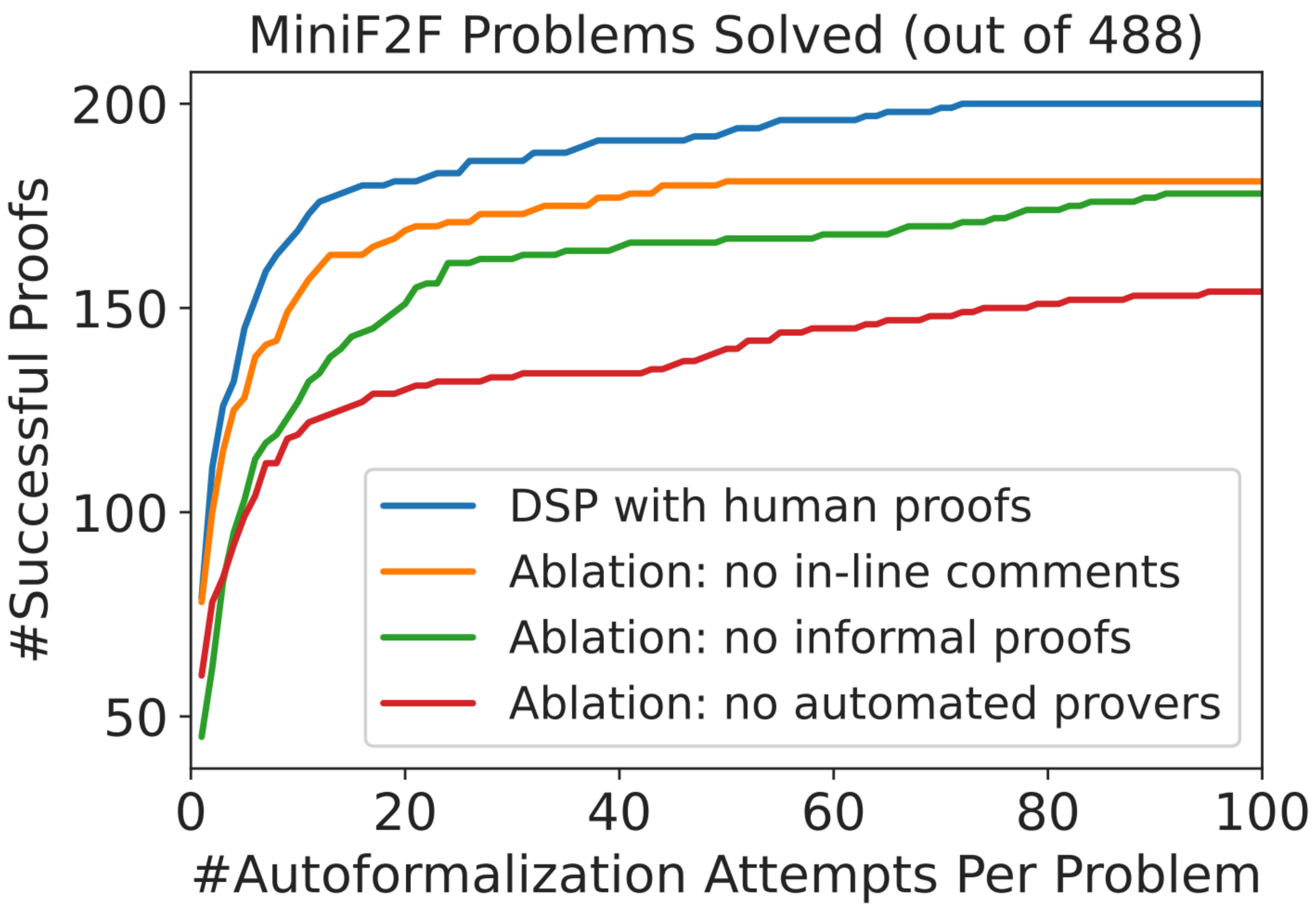
```
theorem imo_1959_p1:
  fixes n :: nat
  shows "gcd (21*n + 4) (14*n + 3) = 1"
proof -
  (* The Euclidean algorithm gives
  21n+4=1\cdot(14n+3)+7n+1
  14n+3=2\cdot(7n+1)+1. *)
  have c0: "21*n + 4 = 1*(14*n + 3) + 7*n + 1"
    by auto   [ATP]
  have c1: "14*n + 3 = 2*(7*n + 1) + 1" using c0
    by auto   [ATP]

  (* Since \gcd(7n+1,1)=1, we have \gcd(21n+4,14n+3)=1. *)
  then have "gcd (7*n + 1) 1 = 1"
    using c1
    by auto   [ATP]
  then have "gcd (21*n + 4) (14*n + 3) = 1"
    using c1
    by (smt (z3) BitM_plus_one ab_semigroup_add_class.add_ac(1)
    add.assoc c0 gcd.commute gcd_add2 gcd_add_mult mult_numeral_1
    numeral_One numeral_eq_Suc numerals(1) semiring_norm(3))   [ATP]
  then show ?thesis
    using c1
    by blast   [ATP]
qed
```

Increasing search budget



Ablations



MiniF2F Problems Solved (out of 488)
200
150
100
50
#Successful Proofs
0
20
40
60
80
100
#Autoformalization Attempts Per Problem
DSP with human proofs
Ablation: no in-line comments
Ablation: no informal proofs
Ablation: no automated provers

Model vs. human informal draft

Statement: For any integer a, show that $a^2 \equiv 0 \pmod 4$ or $a^2 \equiv 1 \pmod 4$.

Informal Proof (Minerva 62B):
Let $a = 2k$ for some integer k.
Then $a^2 = 4k^2$.
Since $4k^2$ is divisible by 4, we have $a^2 \equiv 0 \pmod 4$.
Now, let $a = 2k + 1$ for some integer k.
Then $a^2 = 4k^2 + 4k + 1 = 4(k^2 + k) + 1$.
Since $k^2 + k$ is an integer, $4(k^2 + k) + 1$ is not divisible by 4.
Thus, $a^2 \equiv 1 \pmod 4$.

Statement: For any integer a, show that $a^2 \equiv 0 \pmod 4$ or $a^2 \equiv 1 \pmod 4$.

Informal Proof (Human):
$a \pmod 4 \in \{0, 1, 2, 3\}$.
Using that for any natural number k, $a \equiv b \pmod 4$ implies $a^k \equiv b^k \pmod 4$, we have $a^2 \pmod 4 \in \{0, 1, 4, 9\}$.
Since $4 \equiv 0 \pmod 4$ and $9 \equiv 1 \pmod 4$, the result follows.

Model vs. human informal draft

Formal Proof:

```
theorem numbertheory_sqmod4in01d:
  fixes a :: int
  shows "(a^2 mod 4 = 0)
  \<or> (a^2 mod 4 = 1)"
proof (cases "even a")
case True
 (* Let a=2k for some integer k.
    Then a^2=4k^2.
    Since 4k^2 is divisible by 4,
    we have a^2 \equiv 0 \pmod{4}.*)
 then obtain k where "a=2*k"
   using evenE
   <ATP> by auto </ATP>
 then have "a^2 = 4*k^2"
   unfolding power2_eq_square
   <ATP> by auto </ATP>
 then have "a^2 mod 4 = 0"
   <ATP> by auto </ATP>
 then show ?thesis
   <ATP> by auto </ATP>
next
case False
 (* Now, let a=2k+1 for some integer k.
    Then a^2=4k^2+4k+1=4(k^2+k)+1.
    Since k^2+k is an integer,
    4(k^2+k)+1 is not divisible by 4.
    Thus, a^2 \equiv 1 \pmod{4}.*)
 then obtain k where "a=2*k+1"
  using oddE <ATP> by auto </ATP>
 then have "a^2 = 4*k^2+4*k+1"
  unfolding power2_eq_square <ATP>
    by (auto simp: field_simps) </ATP>
 then have "a^2 mod 4 = 1"
  <ATP> by presburger </ATP>
 then show ?thesis
  <ATP> by auto </ATP>
qed
```

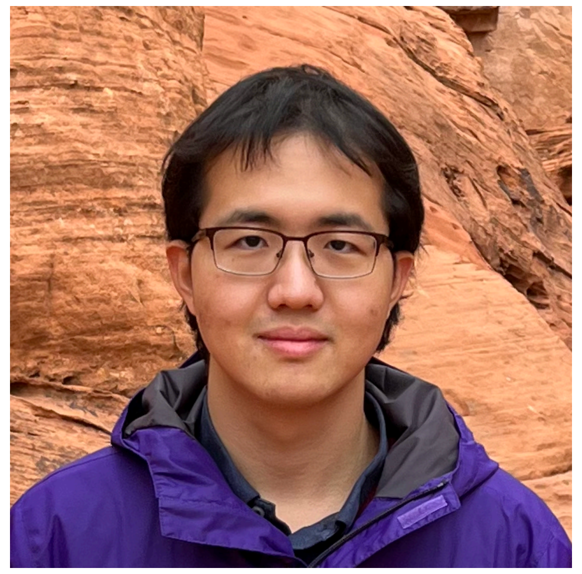
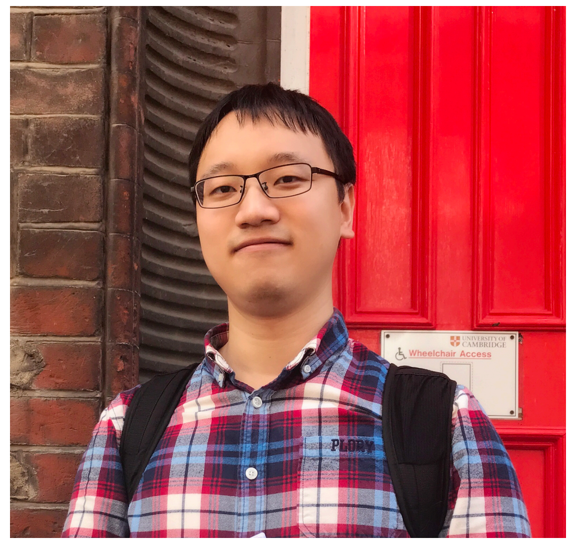
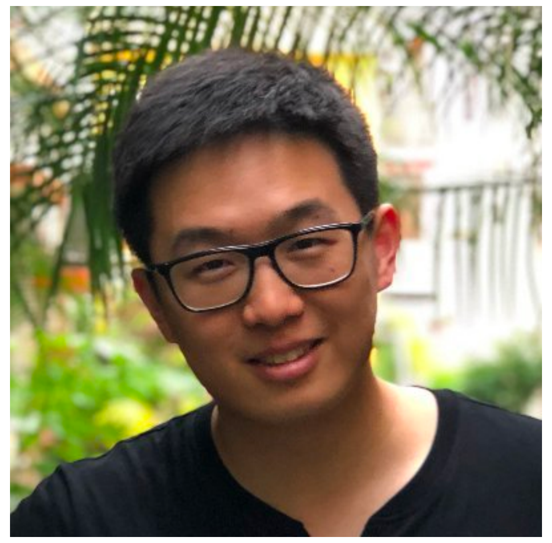
Formal Proof:

```
theorem numbertheory_sqmod4in01d:
  fixes a :: int
  shows "(a^2 mod 4 = 0)
  \<or> (a^2 mod 4 = 1)"
proof -
(* a \pmod 4 \in {0, 1, 2, 3}. *)
have c0: "a mod 4 \<in> {0, 1, 2, 3}"
  <ATP> by auto </ATP>
(* Using that for any natural number k,
   a \equiv b \pmod 4 implies
   a^k \equiv b^k \pmod 4,*)
have "a^2 mod 4 = (a mod 4)\<^sup>
      2 mod 4" <ATP> by (smt (z3)
      numeral_eq_Suc power_mod) </ATP>
(* we have
   a^2 \pmod 4 \in {0, 1, 4, 9}. *)
also have "... \<in> {0, 1, 4, 9}"
  using c0
  <ATP> by auto </ATP>
(* Since 4 \equiv 0 \pmod 4 and
   9 \equiv 1 \pmod 4,
   the result follows. *)
finally show ?thesis
  <ATP> by auto </ATP>
qed
```

Conclusion

- Draft informal proofs and sketch formal proofs
  - Combines flexibility of language models with formal systems
- Opens many possibilities and questions
  - Memorization
  - Alternative ways of integrating language models
  - Exploration and reinforcement learning

Thank you

Appendix: full results

Success rate	miniF2F-valid	miniF2F-test
Baselines		
Sledgehammer	9.9%	10.4%
Sledgehammer + heuristics	18.0%	20.9%
Thor (Jiang et al., 2022)	28.3%	29.9%
Thor + expert iteration (Wu et al., 2022)	37.3%	35.2%
Draft, Sketch, and Prove		
Human informal proof	42.6%	**39.3**%
Codex informal proof	40.6%	35.3%
8B Minerva informal proof	40.6%	35.3%
62B Minerva informal proof	**43.9**%	37.7%
540B Minerva informal proof	42.6%	38.9%
Ablations (with human informal statements and proofs)		
– In-line comments	37.7% (−4.9%)	36.5% (−2.8%)
– Informal proofs	38.9% (−3.7%)	34.0% (−5.3%)
– Automated provers	32.8% (−9.8%)	30.3% (−9.0%)