

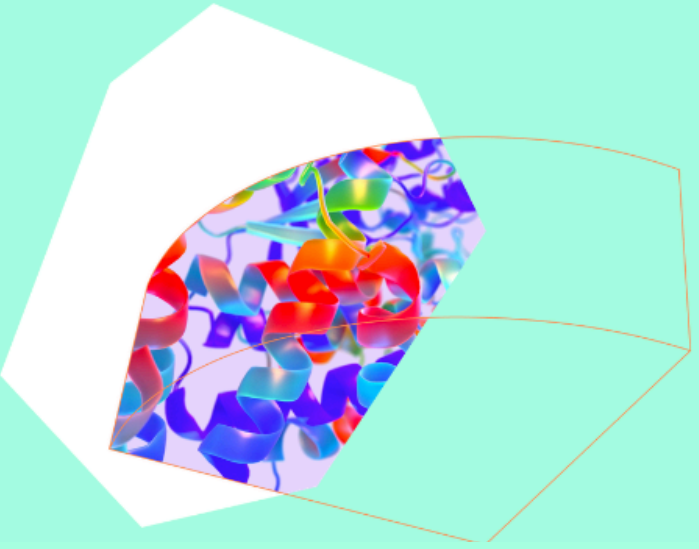
Bridging Informal & Formal Mathematical Reasoning

(with neural language models)

Science

Engineering

Education



AlphaFold

Accelerating scientific discovery

AlphaFold can accurately predict 3D models of protein structures and is accelerating research in nearly every field of biology.

Science

Engineering

Education

Science

Engineering

Education

Your AI pair programmer

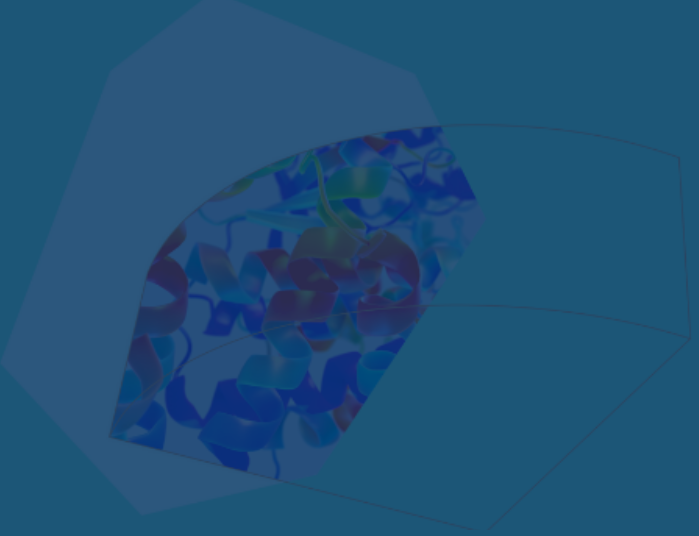
GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.

[Start my free trial >](#)

[Explore docs](#)

iments.ts write_sql.go parse_expenses.py addresses.rb

```
usr/bin/env ts-node  
ort { fetch } from "fetch-h2";
```

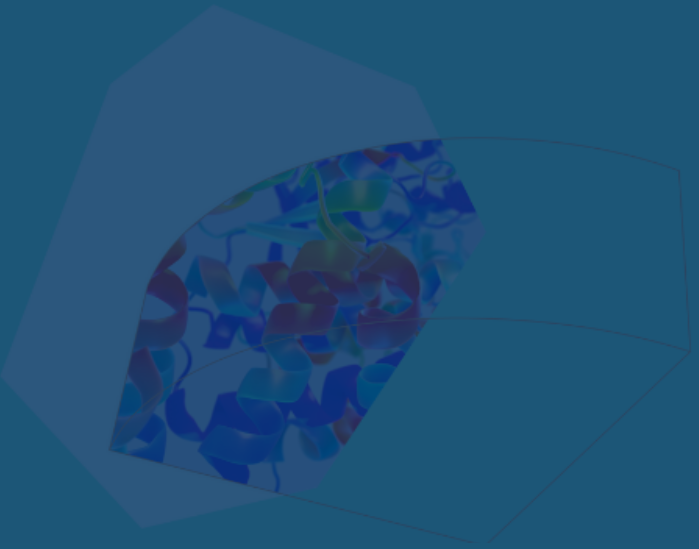


AlphaFold

Accelerating scientific discovery

AlphaFold can accurately predict 3D models of protein structures and is accelerating research in nearly every field of biology.

Science



AlphaFold

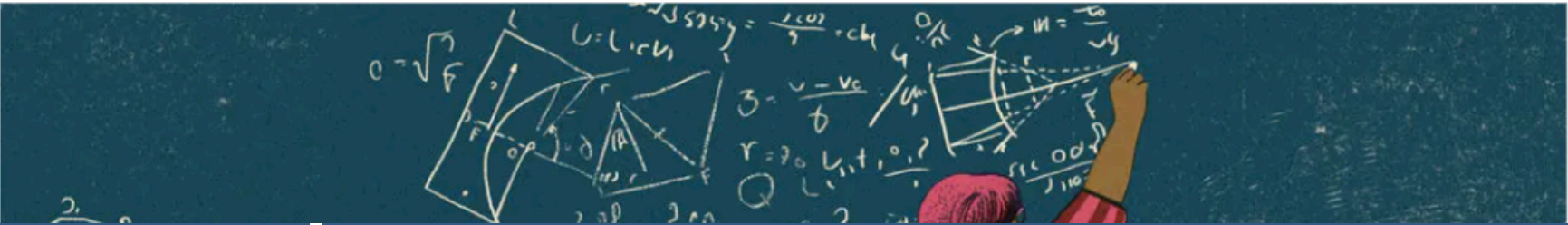
Accelerating scientific discovery

AlphaFold can accurately predict 3D models of protein structures and is accelerating research in nearly every field of biology.

Can A.I. Grade Your Next Test?

Neural networks could give online education a boost by providing automated feedback to students.

Give this article



Engineering

Education

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.

Start my free trial >

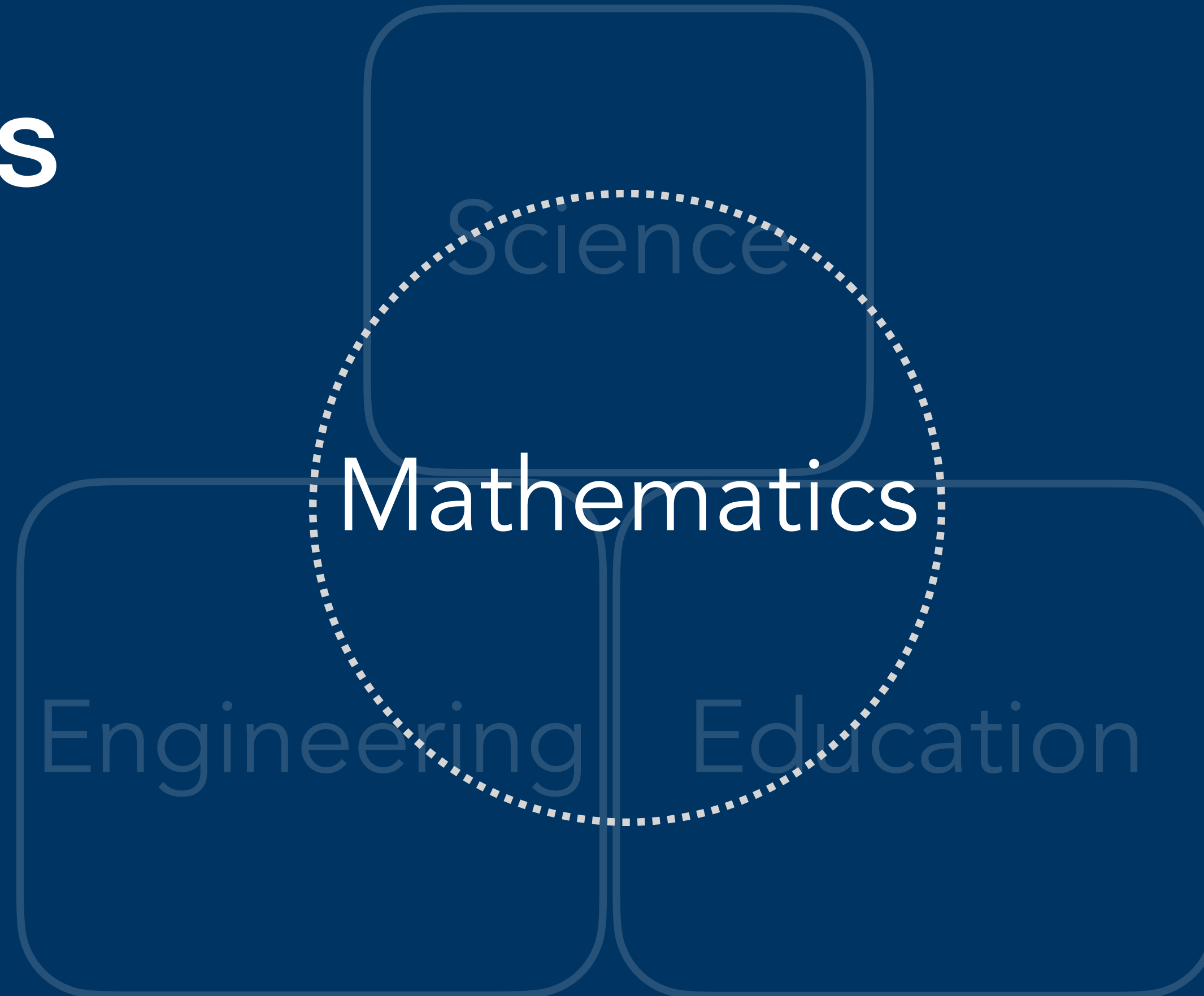
Explore docs

iments.ts write_sql.go parse_expenses.py addresses.rb

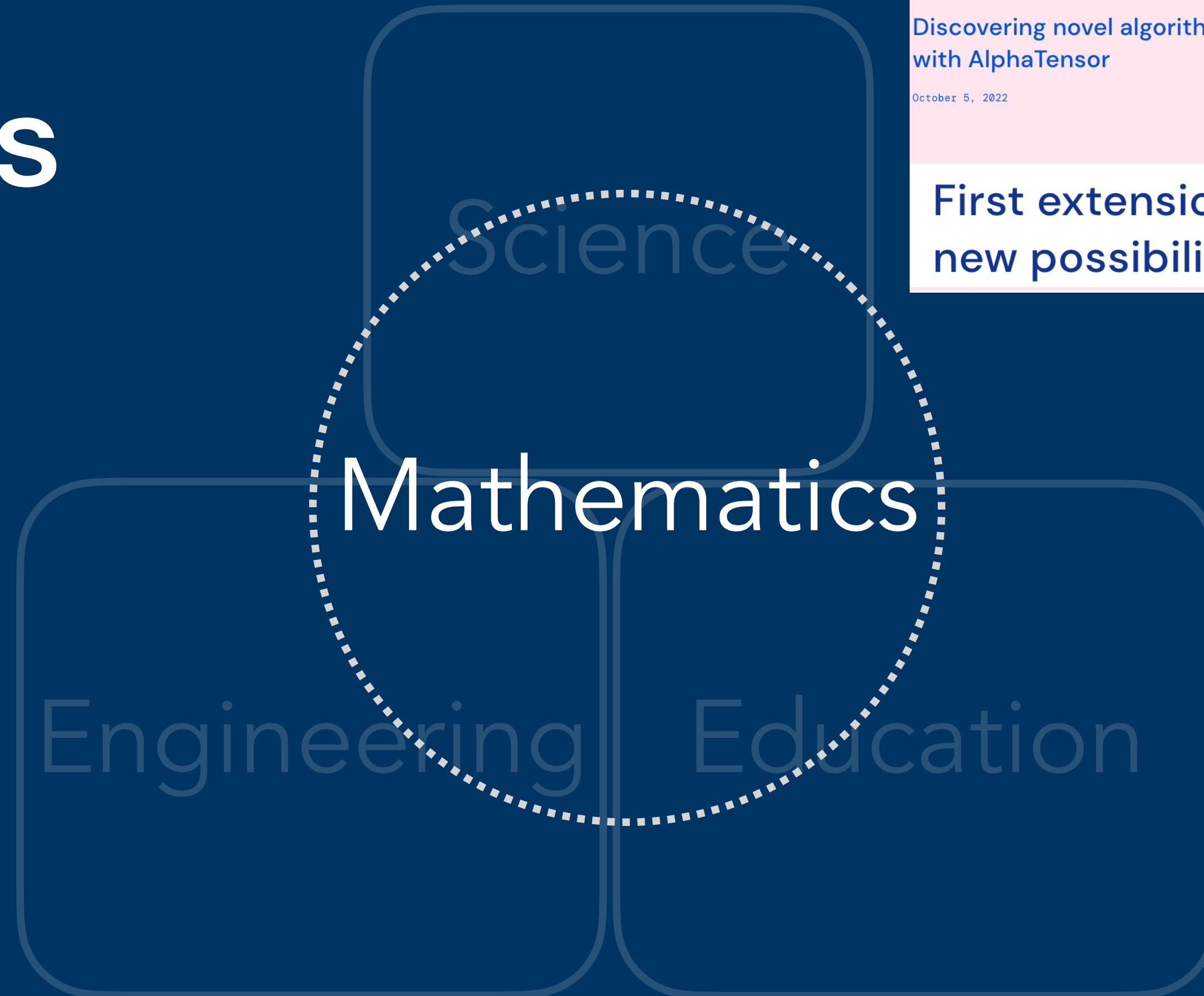
usr/bin/env ts-node

ort { fetch } from "fetch-h2";

AI for mathematics



AI for mathematics



Discovering novel algorithms
with AlphaTensor

October 5, 2022



First extension of AlphaZero to mathematics unlocks
new possibilities for research

AI for mathematics

Discovering novel algorithms with AlphaTensor

October 5, 2022



First extension of AlphaZero to mathematics unlocks new possibilities for research

Mathematics

MACHINE LEARNING

In New Math Proofs, Artificial Intelligence Plays to Win

3 |

A new computer program fashioned after artificial intelligence systems like AlphaGo has solved several open problems in combinatorics and graph theory.

AI for mathematics

Mathematics

Discovering novel algorithms with AlphaTensor

October 5, 2022

First extension of AlphaZero to mathematics unlocks new possibilities for research



MACHINE LEARNING

In New Math Proofs, Artificial Intelligence Plays to Win

3 |

A new computer program fashioned after artificial intelligence systems like AlphaGo has solved several open problems in combinatorics and graph theory.

Microsoft / The AI Blog



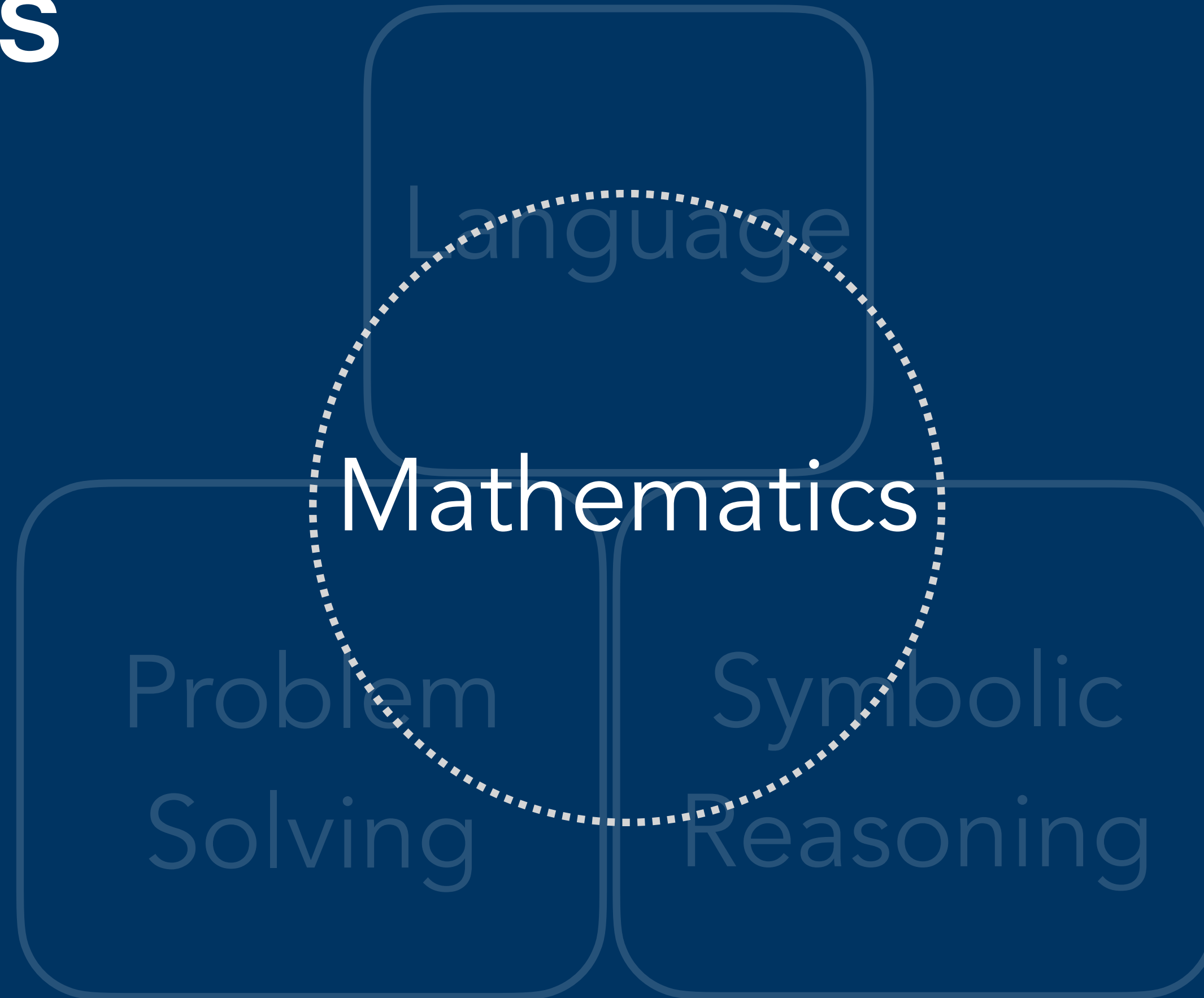
Online math tutoring service uses AI to help boost students' skills and confidence



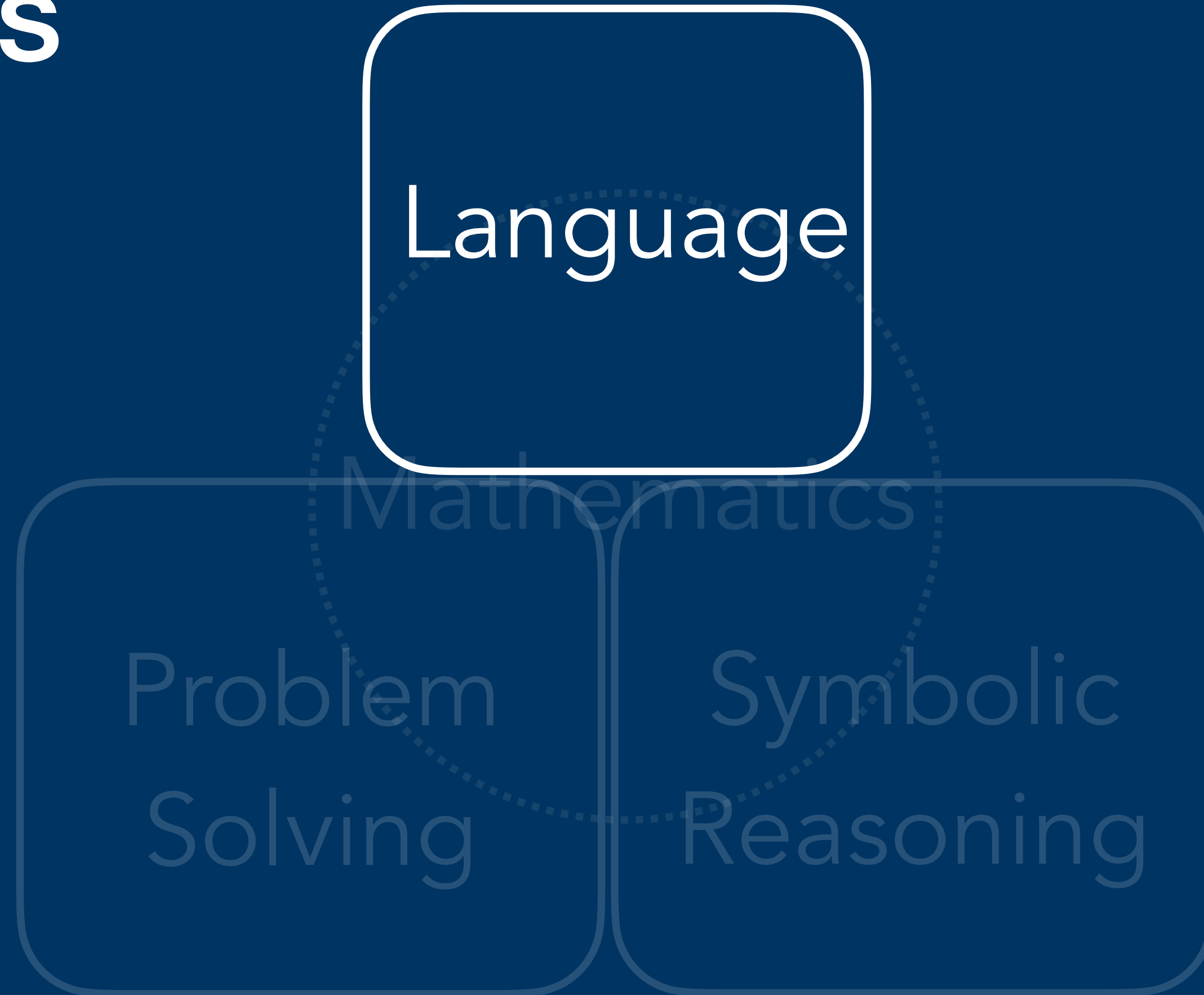
John Roach
Jul 13, 2022



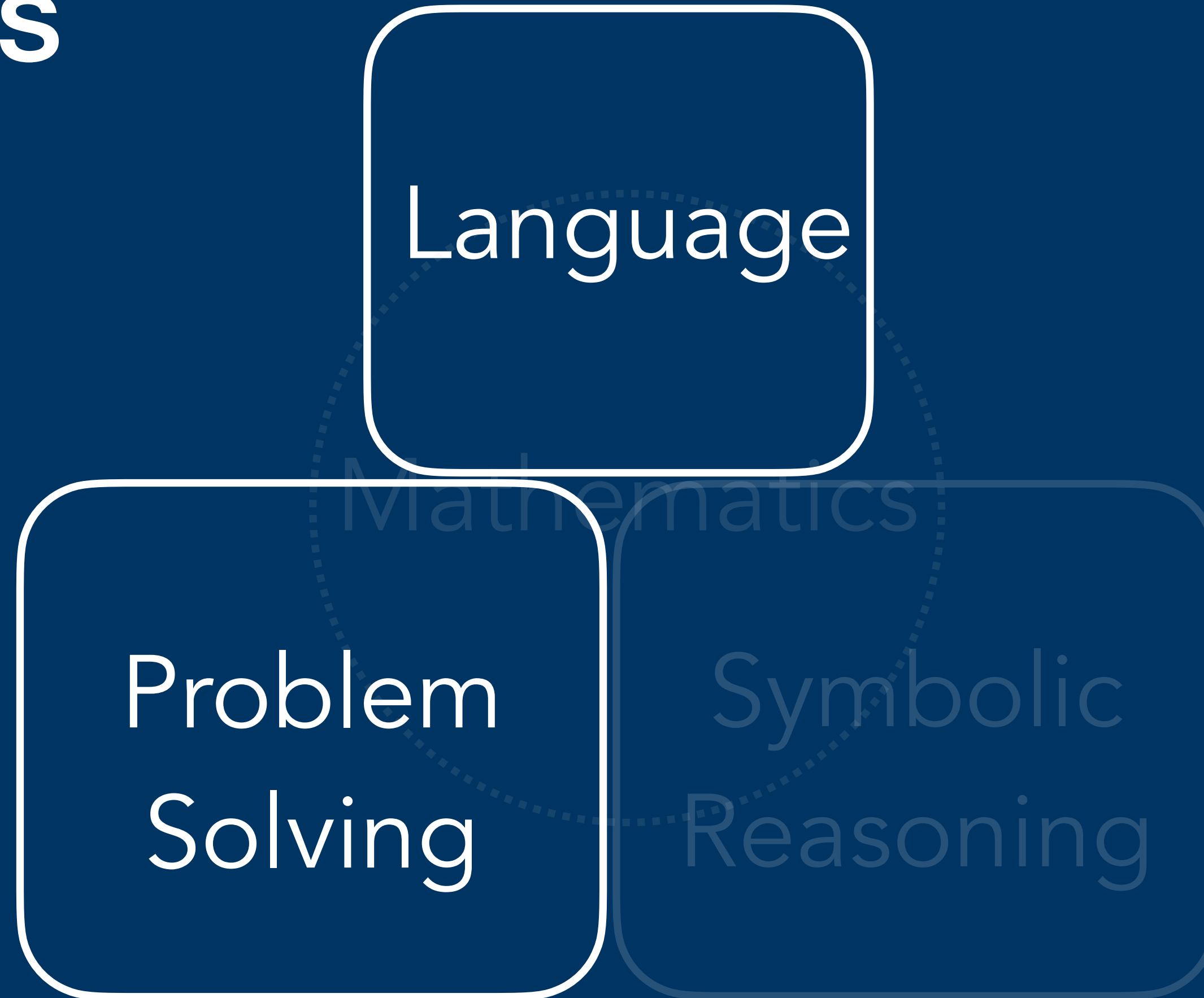
Mathematics for AI



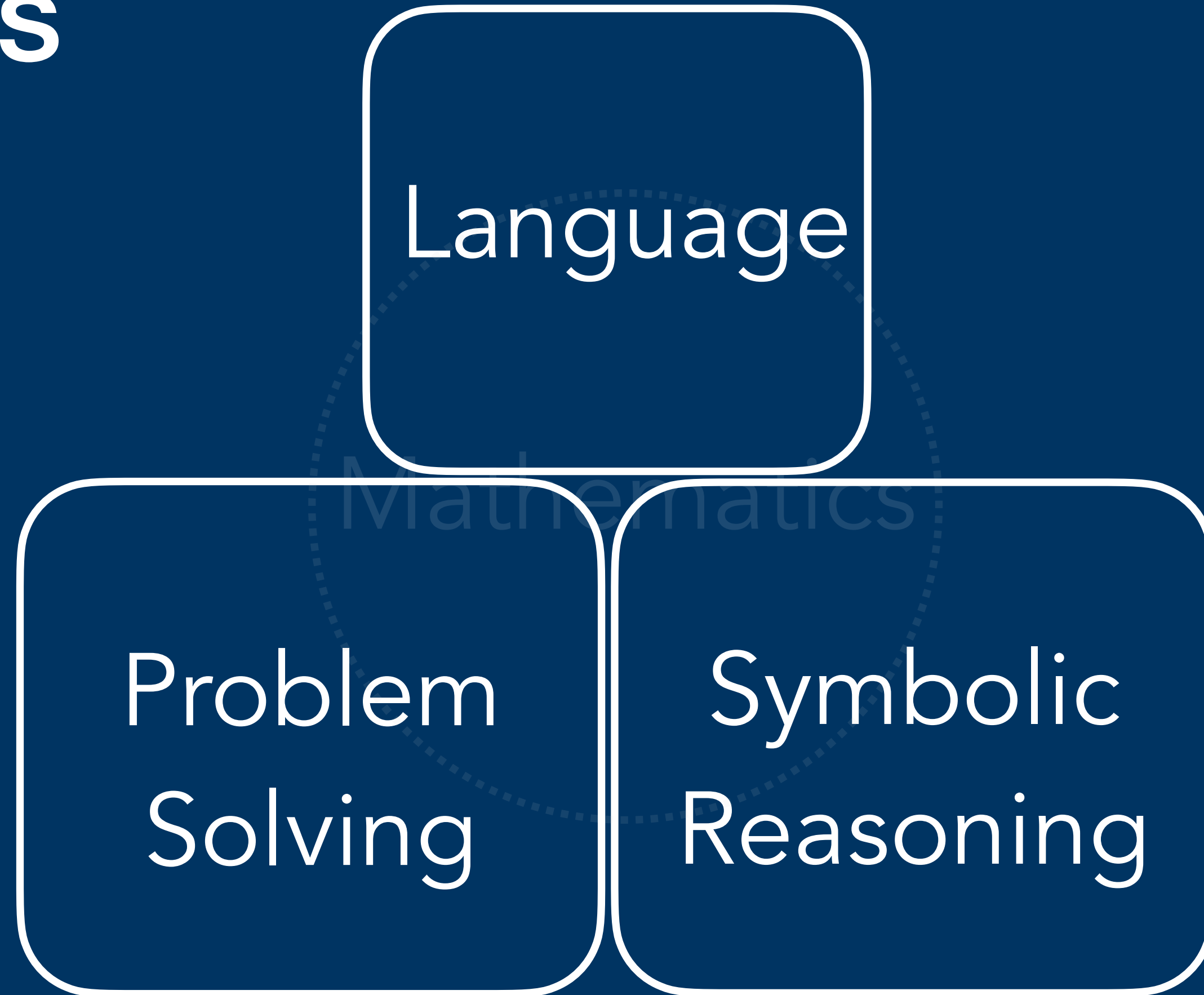
Mathematics for AI



Mathematics for AI



Mathematics for AI



Claim:

$$1 + 1 = 2$$

Proof:



Claim:

$$1 + 1 = 2$$

Proof:

It's obvious: if you pick up one apple, and then pick up another, you've picked up two apples.

Claim:

$$1 + 1 = 2$$

Proof:

*54·43. $\vdash :: \alpha, \beta \in 1 . \supset : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54·26 . \supset \vdash :: \alpha = \iota'x . \beta = \iota'y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

[*51·231] $\equiv . \iota'x \cap \iota'y = \Lambda .$

[*13·12] $\equiv . \alpha \cap \beta = \Lambda \quad (1)$

$\vdash . (1) . *11·11·35 . \supset$

$\vdash :: (\exists x, y) . \alpha = \iota'x . \beta = \iota'y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda \quad (2)$

$\vdash . (2) . *11·54 . *52·1 . \supset \vdash . \text{Prop}$

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

Russell 1910, *Principia Mathematica*

$$1 + 1 = 2$$

It's obvious: if you pick up one apple, and then pick up another, you've picked up two apples.

Intuitive
Ambiguous

*54.43. $\vdash :: \alpha, \beta \in 1. \supset : \alpha \cap \beta = \Lambda. \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54.26. \supset \vdash :: \alpha = \iota'x. \beta = \iota'y. \supset : \alpha \cup \beta \in 2. \equiv . x \neq y.$

[*51.231]

$\equiv . \iota'x \cap \iota'y = \Lambda.$

[*13.12]

$\equiv . \alpha \cap \beta = \Lambda \quad (1)$

$\vdash . (1). *11.11.35. \supset$

$\vdash :: (\forall x, y). \alpha = \iota'x. \beta = \iota'y. \supset : \alpha \cup \beta \in 2. \equiv . \alpha \cap \beta = \Lambda \quad (2)$

$\vdash . (2). *11.54. *52.1. \supset \vdash . \text{Prop}$

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

Precise
Explicit

Informal

Formal

It's obvious: if you pick up one apple, and then pick up another, you've picked up two apples.



Theorem

Let $x \in \mathbb{Z}$ be an even integer.
 Then $x + 5$ is odd.

Proof

Let x be an even integer.
 Let $y = 2n + 5$.

Assume $y = x + 5$ is not an odd integer.
 Then:

$$y = x + 5 = 2n$$
 where $n \in \mathbb{Z}$.

Then:

$$x = 2n - 5$$

$= 2$

Using the definition of the successor of 1, the inductive definition of $S(1) = 2$.

*54.43. $\vdash : \alpha, \beta \in 1. \supset : \alpha \cap \beta = \Lambda. \equiv . \alpha \cup \beta \in 2$
Dem.
 $\vdash . *54.26. \supset \vdash : \alpha = t'x. \beta = t'y. \supset : \alpha \cup \beta \in 2. \equiv . x \neq y.$
 [*51.231] $\equiv . t'x \cap t'y = \Lambda.$
 [*13.12] $\equiv . \alpha \cap \beta = \Lambda$ (1)
 $\vdash . (1). *11.11.35. \supset$
 $\vdash : (\forall x, y). \alpha = t'x. \beta = t'y. \supset : \alpha \cup \beta \in 2. \equiv . \alpha \cap \beta = \Lambda$ (2)
 $\vdash . (2). *11.54. *52.1. \supset \vdash . \text{Prop}$
 From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

Precision
 Explicit Logic

Informal

Formal



$$1 + 1 = 2$$

It's obvious: if you pick up one apple, and then pick up another, you've picked up two apples.



Ambiguity Language

Problem

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

Let $\text{gcd}(21n + 4, 14n + 3) = a$. So for some co-prime positive integers x, y we have

$$21n + 4 = ax \quad (1)$$

$$14n + 3 = ay \quad (2)$$

Multiplying (1) by 2 and (2) by 3 and then subtracting (1) from (2) we get

$$42n + 9 - (42n + 8) = 3ay - 2ax$$

$$\implies a(3y - 2x) = 1.$$

We must have $a = 1$, since a is a positive integer. Thus, $\text{gcd}(21n + 4, 14n + 3) = 1$ which means the fraction is irreducible, as needed.

Precision
Explicit Logic

Informal

Formal

1 + 1 = 2

It's obvious: if you pick up one apple, and then pick up another, you've picked up two apples.



Ambiguity Language

Problem

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

We prove that $\gcd(21n + 4, 14n + 3) = 1$ by recalling the definition of 1 as the successor of 0, 2 as the successor of 1, and then invoking twice the recursive definition of addition:

$$1 + 1 = 1 + S(0) = S(1 + 0) = S(1) = 2$$

Multiplying (1) by 2 and (2) by 3 and then subtracting (1) from (2) we get

$$42n + 9 - (42n + 8) = 3ay - 2ax$$

$$\implies a(3y - 2x) = 1.$$

We must have $a = 1$, since a is a positive integer. Thus, $\gcd(21n + 4, 14n + 3) = 1$ which means the fraction is irreducible, as needed.

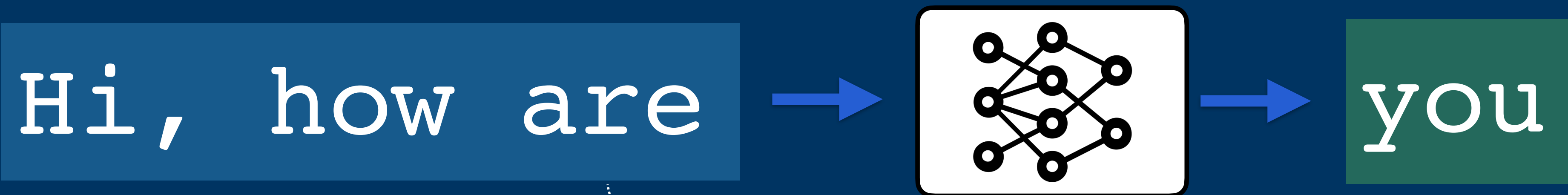
Precision
Explicit Logic

Informal

Formal

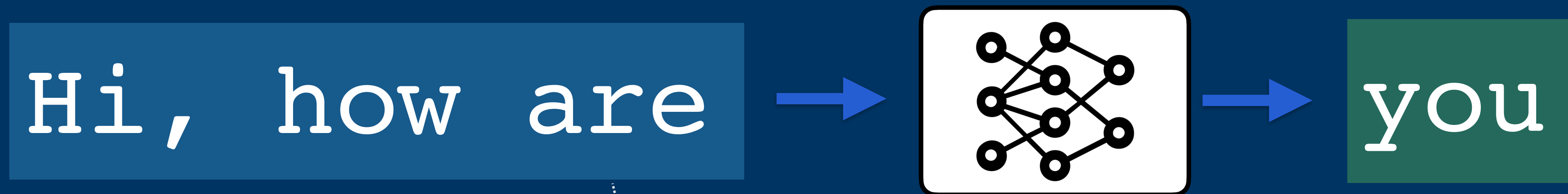


Large Language Models



Informal

Large Language Models



Implicit mixture of tasks



Informal

Large Language Models

Addition is written using the **plus sign** "+" between the terms;^[3]

1 + 1 = 2 ("one plus one equals two")



Implicit mixture of tasks

← Informal →

Large Language Models

Addition is written using the **plus sign** "+" between the terms;^[3]

1 + 1 = 2 ("one plus one equals two")



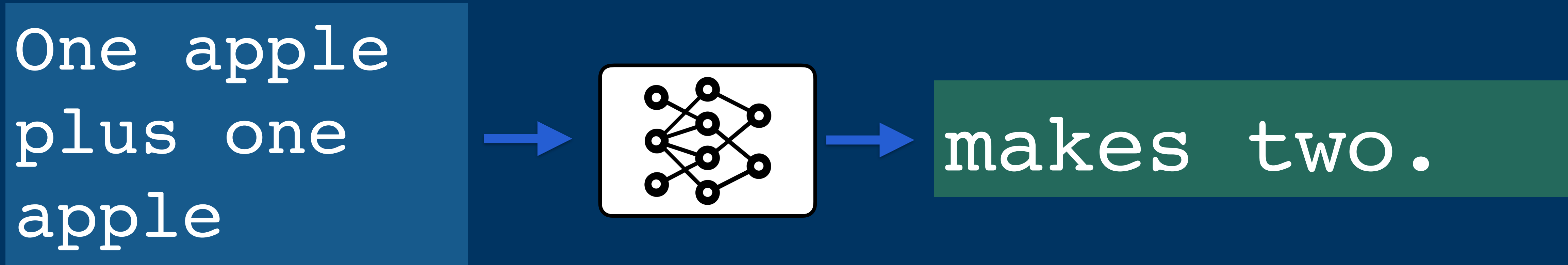
Implicit mixture of tasks

Data rich



Informal

Large Language Models



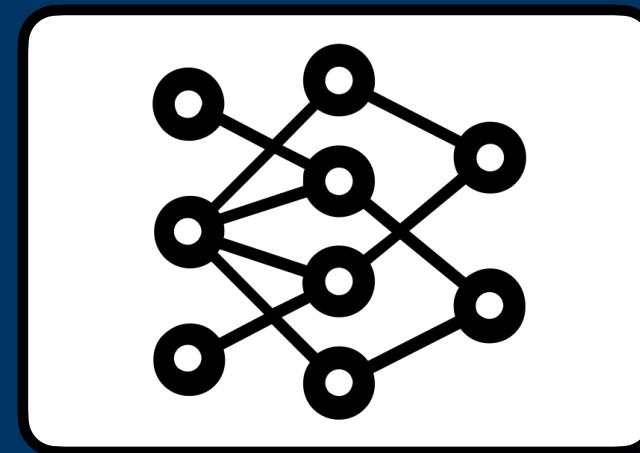
Data rich



Informal

Large Language Models

One apple
plus one
apple



makes two.

Flexible
Data rich



Informal

Large Language Models

Solving Math Word Problems

WE'VE TRAINED a system that solves grade school math problems with nearly twice the accuracy of a fine-tuned GPT-3 model. It solves about 90% as many problems as real kids: a small sample of 9-12 year olds

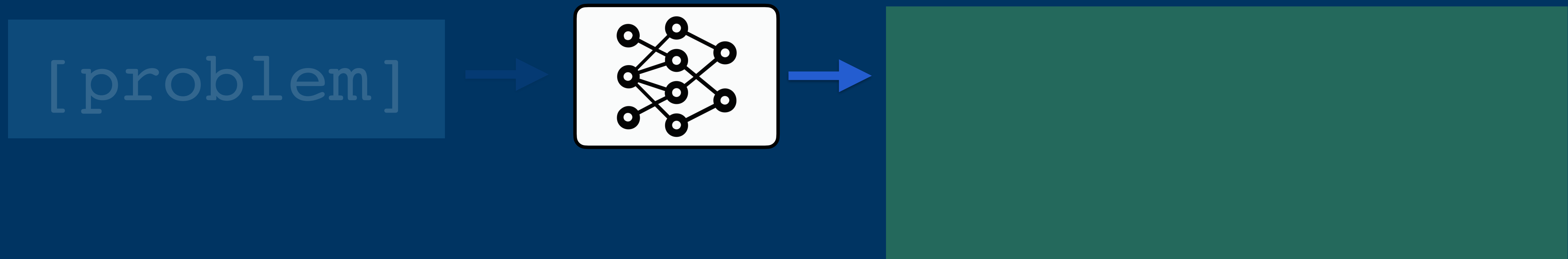
Flexible

Data rich



Informal

Large Language Models

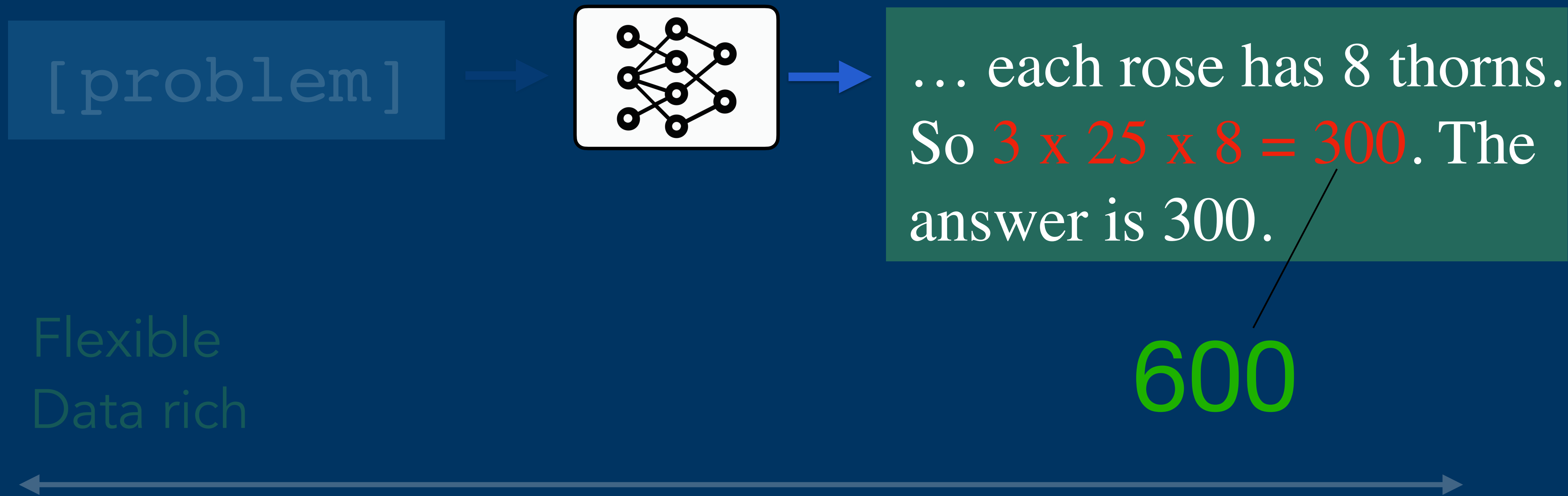


Flexible
Data rich



Informal

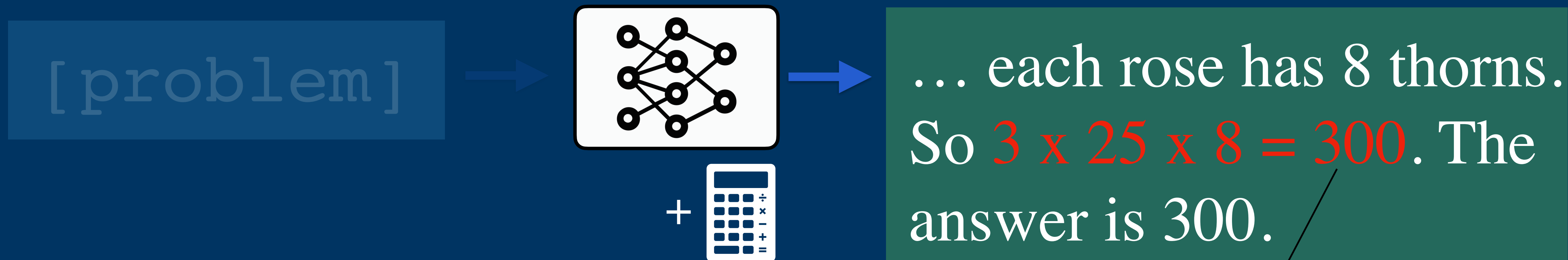
Large Language Models



Flexible
Data rich

Informal

Large Language Models



Flexible
Data rich

600

Informal

Large Language

Direct Proof

Let x be an even integer.

Then by definition:

$$x = 2n$$

for some integer n .

Let $y = 2n + 5$.

Then:

$$\begin{aligned} y &= 2n + 5 \\ &= 2n + 2 \times 2 + 1 \\ &= 2(n + 2) + 1 \\ &= 2r + 1 \end{aligned}$$

[problem]

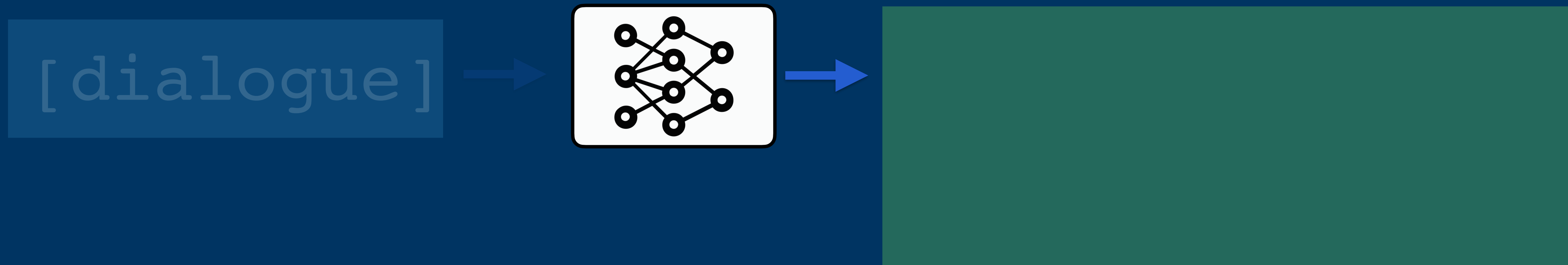
Flexible
Data rich

Informal

... each rose has 8 thorns.
... $3 \times 25 \times 8 = 300$. The
answer is 300.

600

Large Language Models



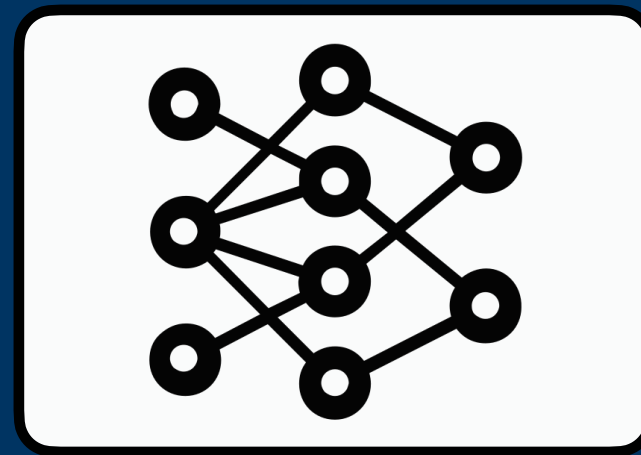
Flexible
Data rich



Informal

Large Language Models

[dialogue]



My favorite Elvis Presley song is “Love Me Do” which came out in 1999.

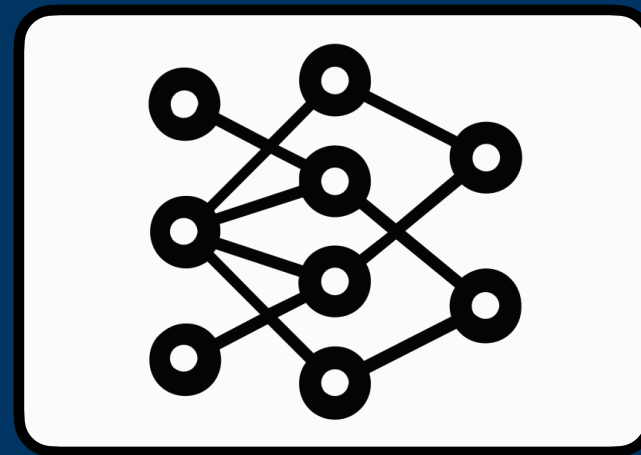
Flexible
Data rich



Informal

Large Language Models

[dialogue]



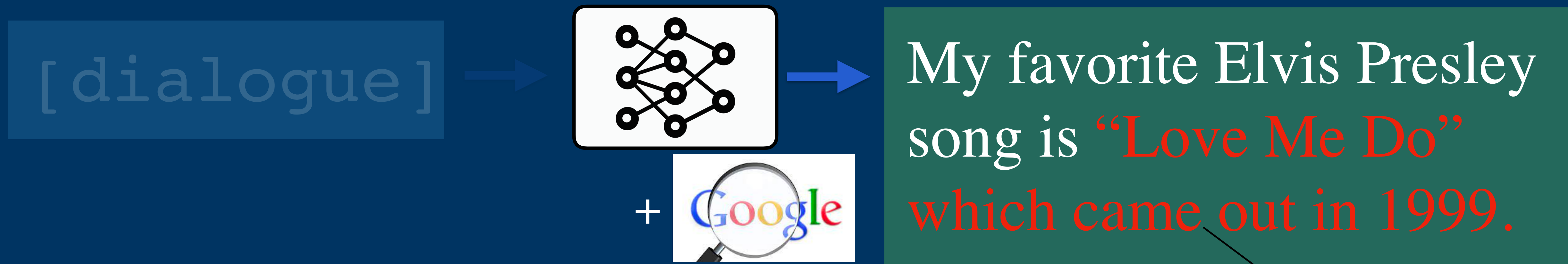
My favorite Elvis Presley song is “Love Me Do” which came out in 1999.

Released by the Beatles in 1962

Flexible
Data rich

Informal

Large Language Models



Flexible
Data rich

Released by the
Beatles in 1962

Informal

Large Language Models

Proof

If $p \nmid n$ then from Fermat's Little Theorem:

$$n^{p-1} \equiv 1 \pmod{p}$$

If $p \mid n$ then:

$$p \mid n^{p-1}$$

and $n^{p-1} \equiv 0 \pmod{p}$ by definition.

Hence the result by definition of Iverson's convention.



Theorem

Let p be a prime number.

Let $n \in \mathbb{Z}_{>0}$ be a positive integer such that p is not a divisor of n .

Then:

$$n^{p-1} \equiv 1 \pmod{p}$$

[di

Flexible
Data rich

Released by the
Beatles in 1962

Informal

Example based on [Shuster et al 2021]

Large Language Models

Proof

If $p \nmid n$ then from Fermat's Little Theorem:

$$n^{p-1} \equiv 1 \pmod{p}$$

If $p \mid n$ then:

$$p \mid n^{p-1}$$

and $n^{p-1} \equiv 0 \pmod{p}$ by definition.

Hence the result by definition of Iverson's convention.



Theorem

Let p be a prime number.

Let $n \in \mathbb{Z}_{>0}$ be a positive integer such that p is not a divisor of n .

Then:

$$n^{p-1} \equiv 1 \pmod{p}$$

[di

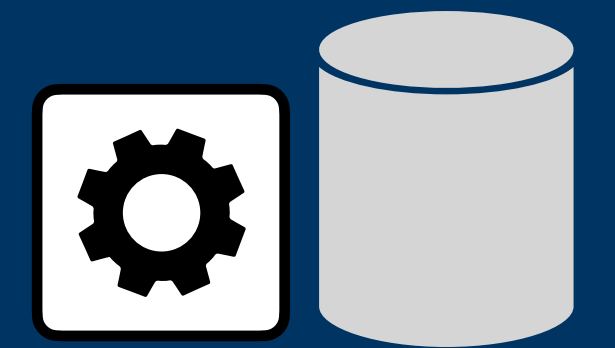
Flexible

Data rich

Grounding

Released by the
Beatles in 1962

Informal

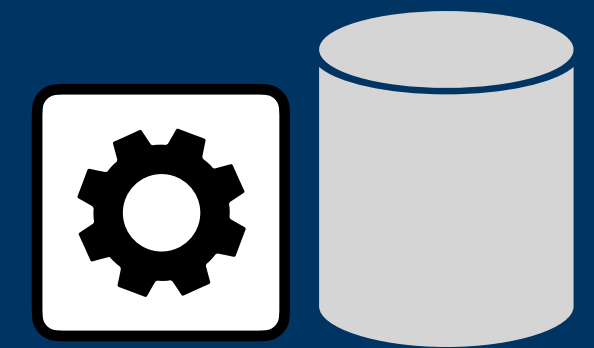


Informal

Formal

NaturalProver

Document grounding

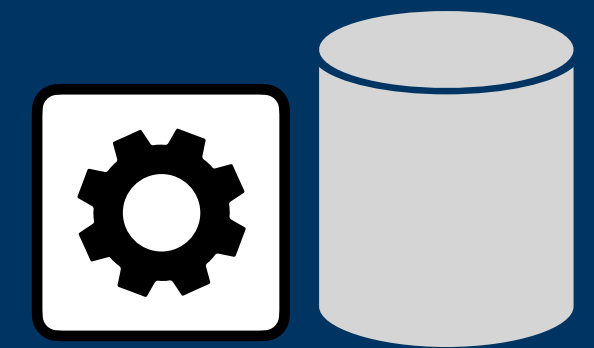


Informal

Formal

Draft, Sketch, Prove

Logical grounding



Informal

Formal

Theorem

The sum of any finite number of even integers is even.

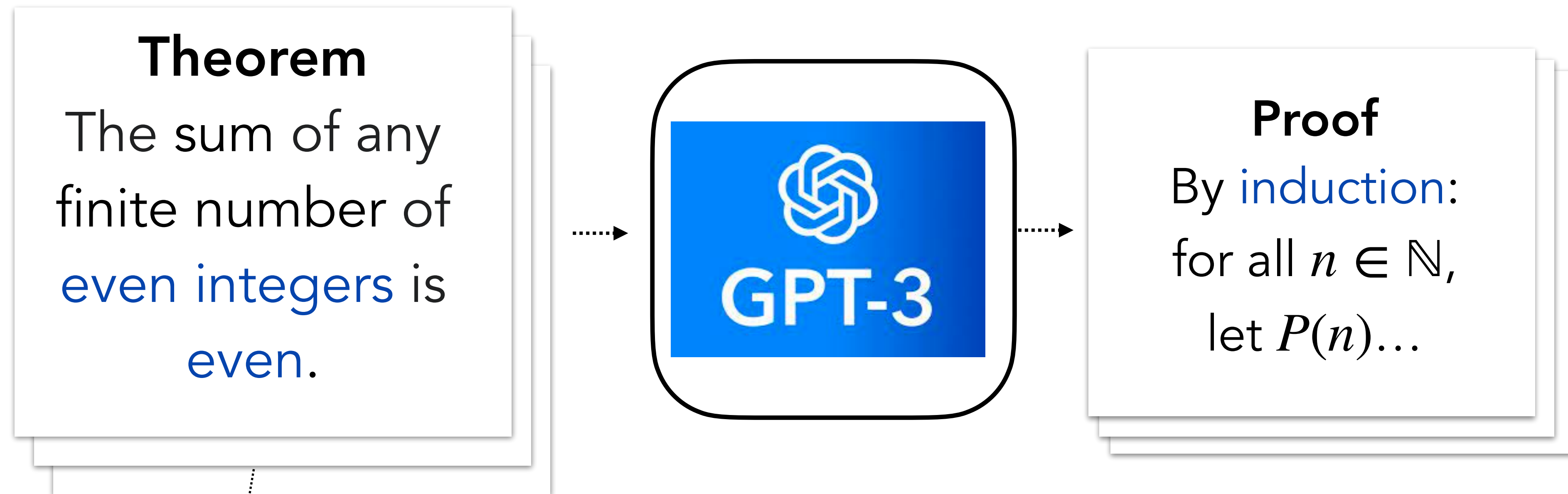


Proof

By induction:
for all $n \in \mathbb{N}$,
let $P(n)$...

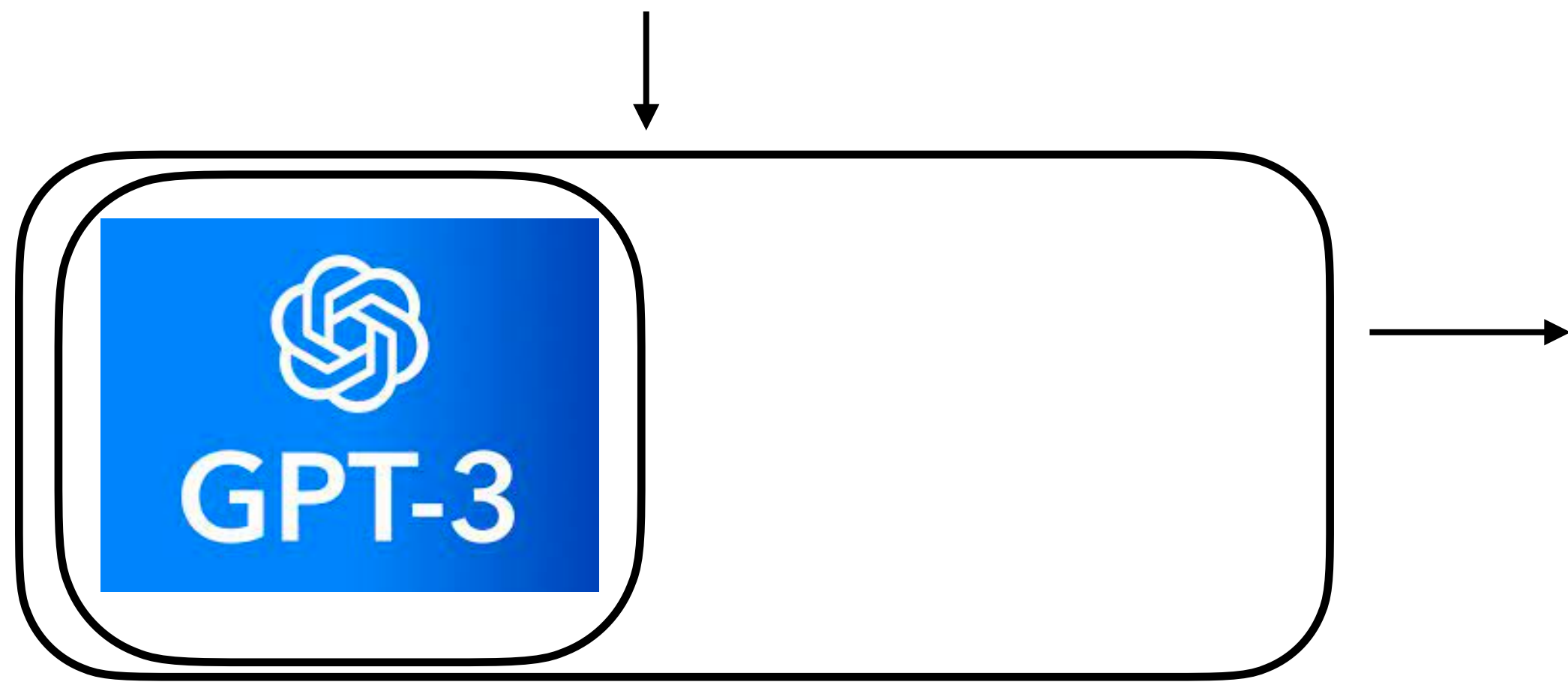


Informal



- ▶ **NaturalProofs** [Welleck et al 2021]:
20k theorems and proofs from ProofWiki

Informal

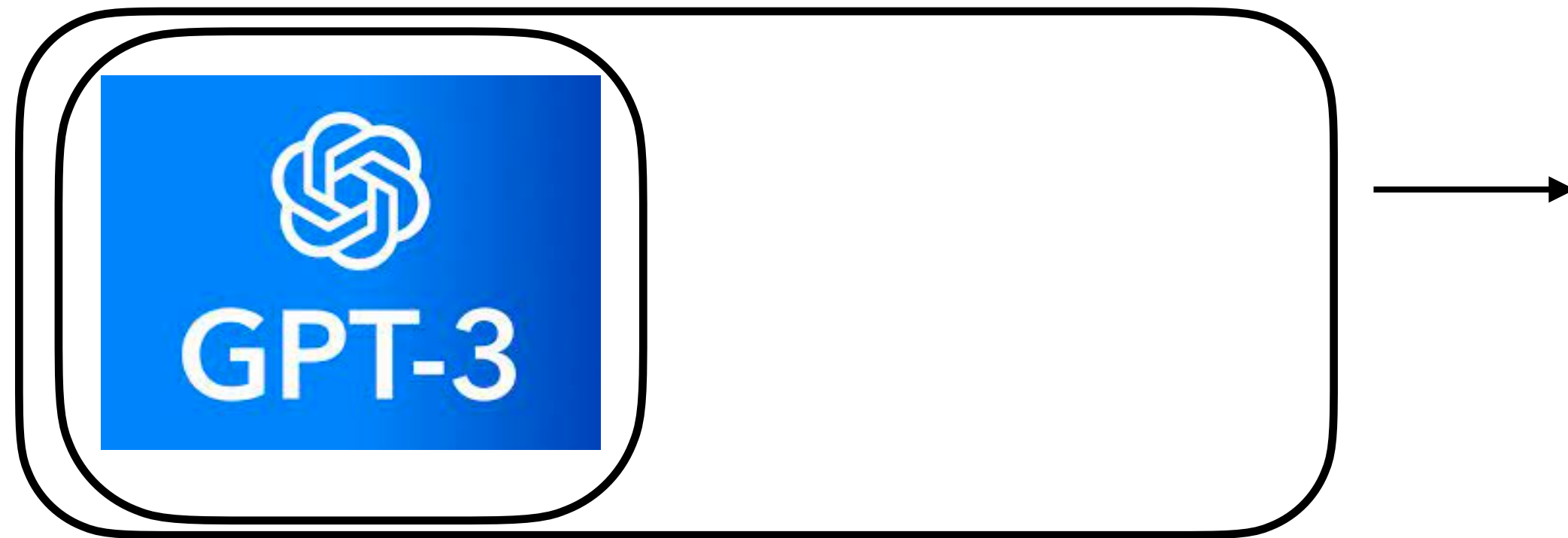


Informal

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Informal

Theorem

Let x be an even integer.

Then $x + 5$ is odd.

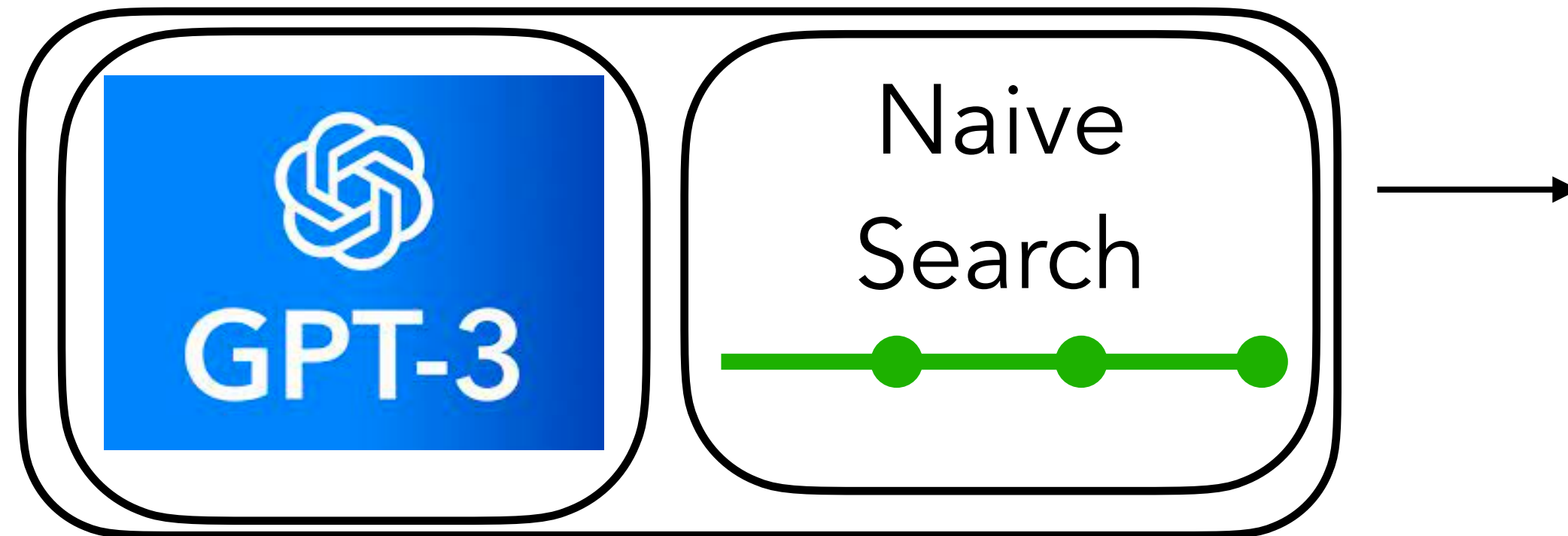


Informal

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

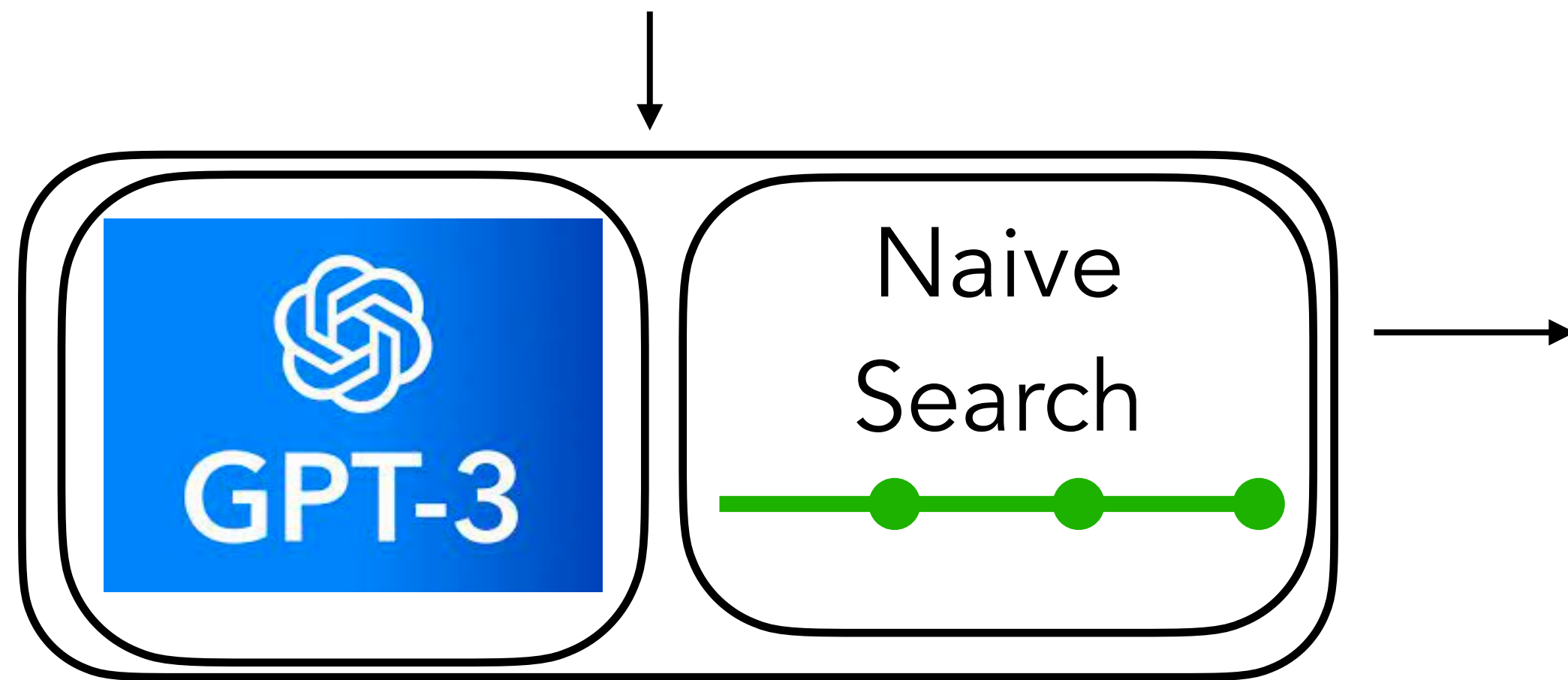


Informal

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



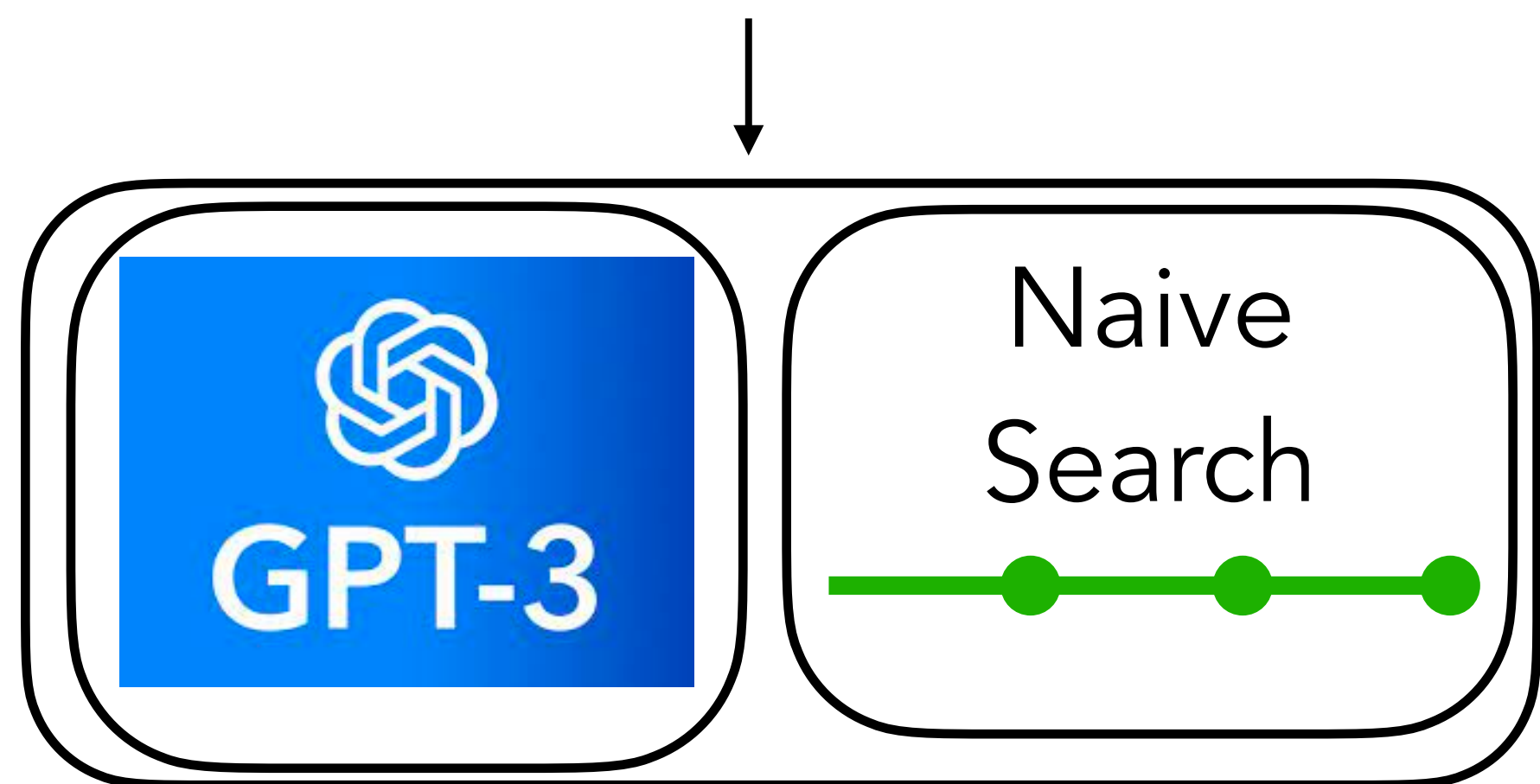
Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even. Then there exists $r \in \mathbb{Z}$ such that $x + 5 = 2r$.

Informal

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

Then there exists $r \in \mathbb{Z}$ such that $x + 5 = 2r$.

But from **Odd Integer Plus Odd Integer is Odd**, $2r + 1$ is odd.

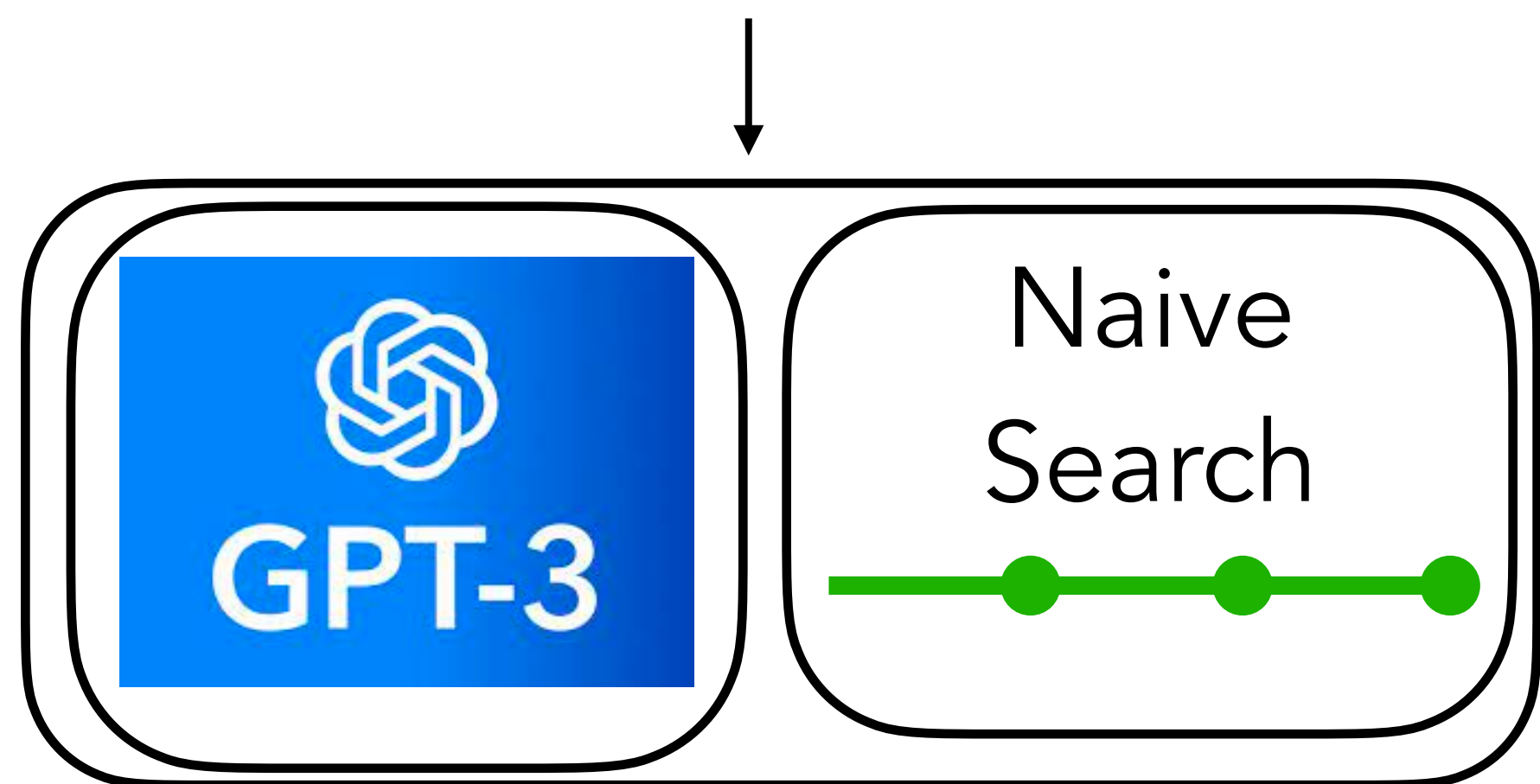
Hallucinated theorem

Informal

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

Then there exists $r \in \mathbb{Z}$ such that $x + 5 = 2r$.

But from **Odd Integer Plus Odd Integer is Odd**, $2r + 1$ is odd.

The result follows by **Proof by Contradiction**.

Hallucinated theorem

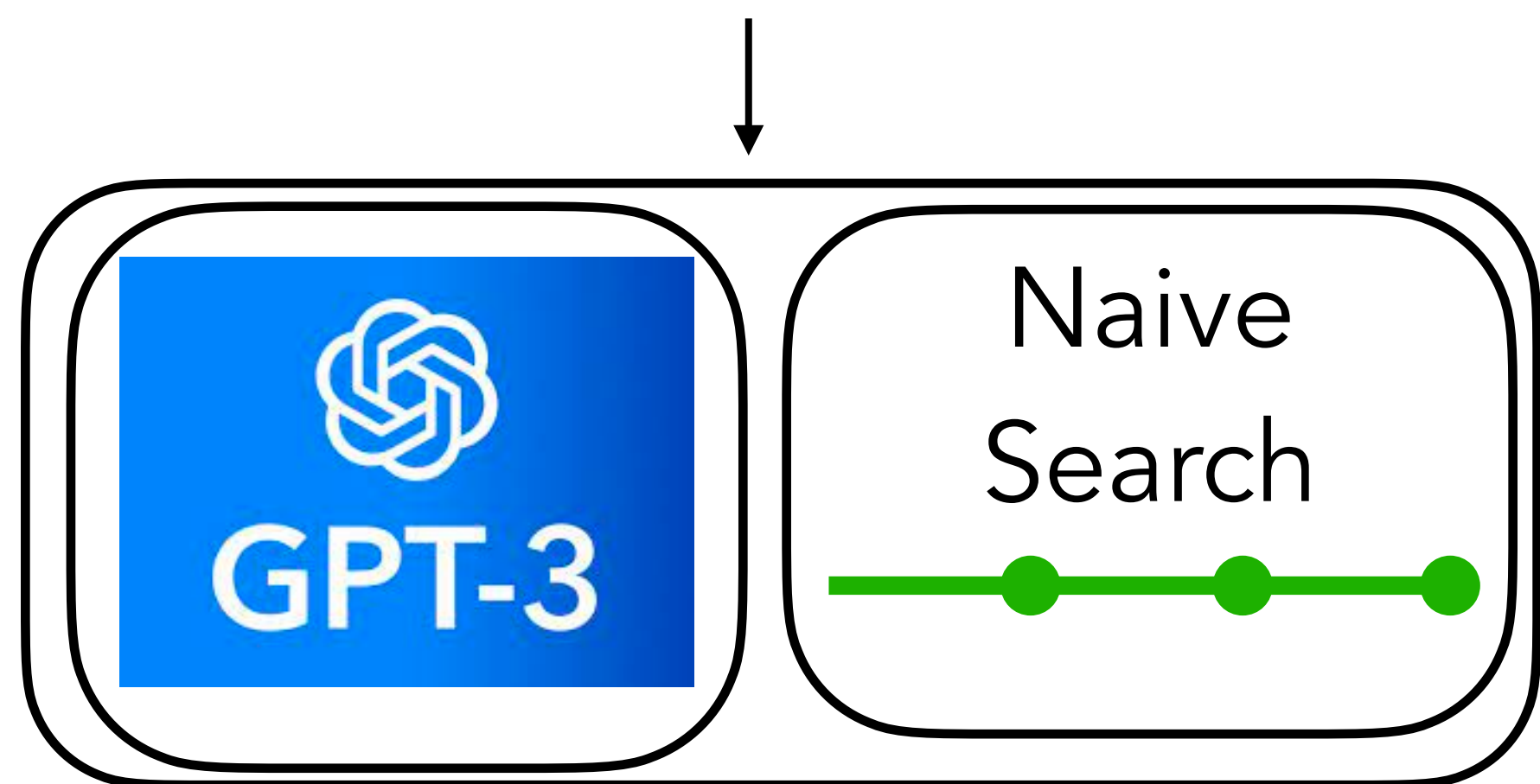
Invalid reasoning

Informal

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

Then there exists $r \in \mathbb{Z}$ such that $x + 5 = 2r$.

But from **Odd Integer Plus Odd Integer is Odd**, $2r + 1$ is odd.

The result follows by Proof by Contradiction.

Hallucinated theorem

Invalid reasoning

knowledge + search \implies grounding

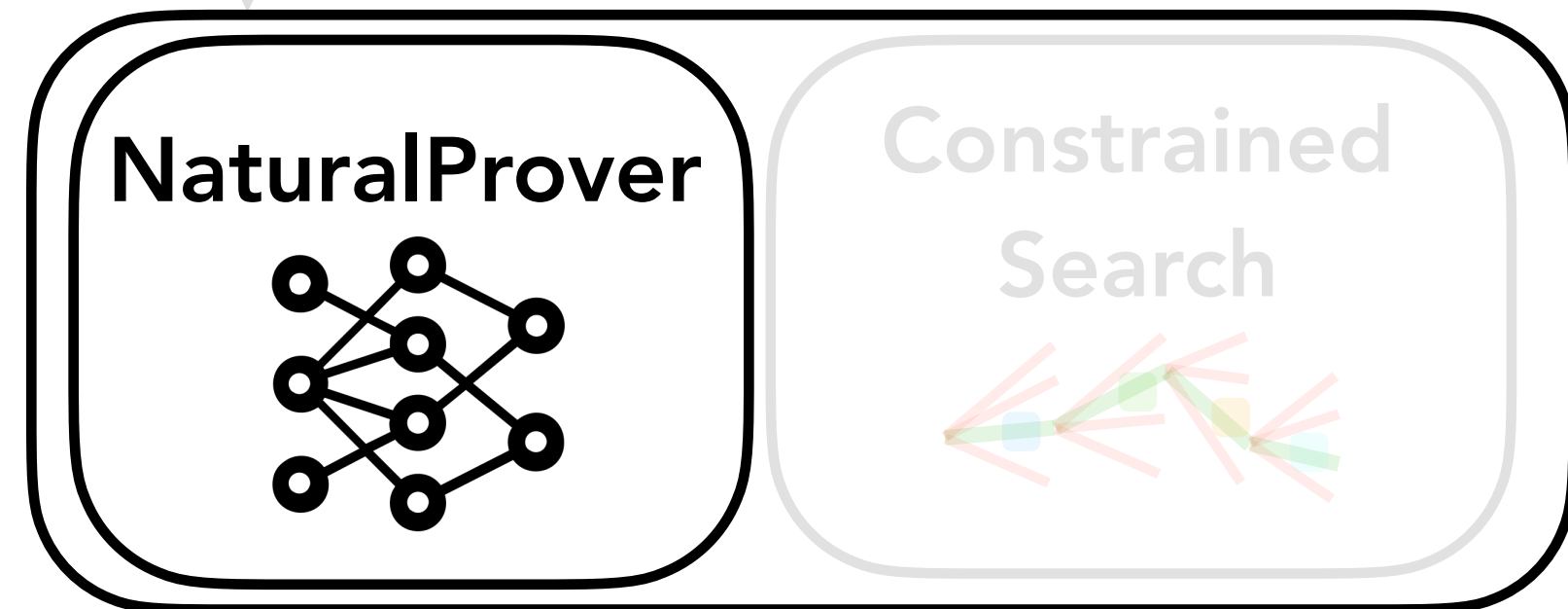
Informal

NaturalProver : knowledge

Theorem

Let x be an even integer.

Then $x + 5$ is odd.

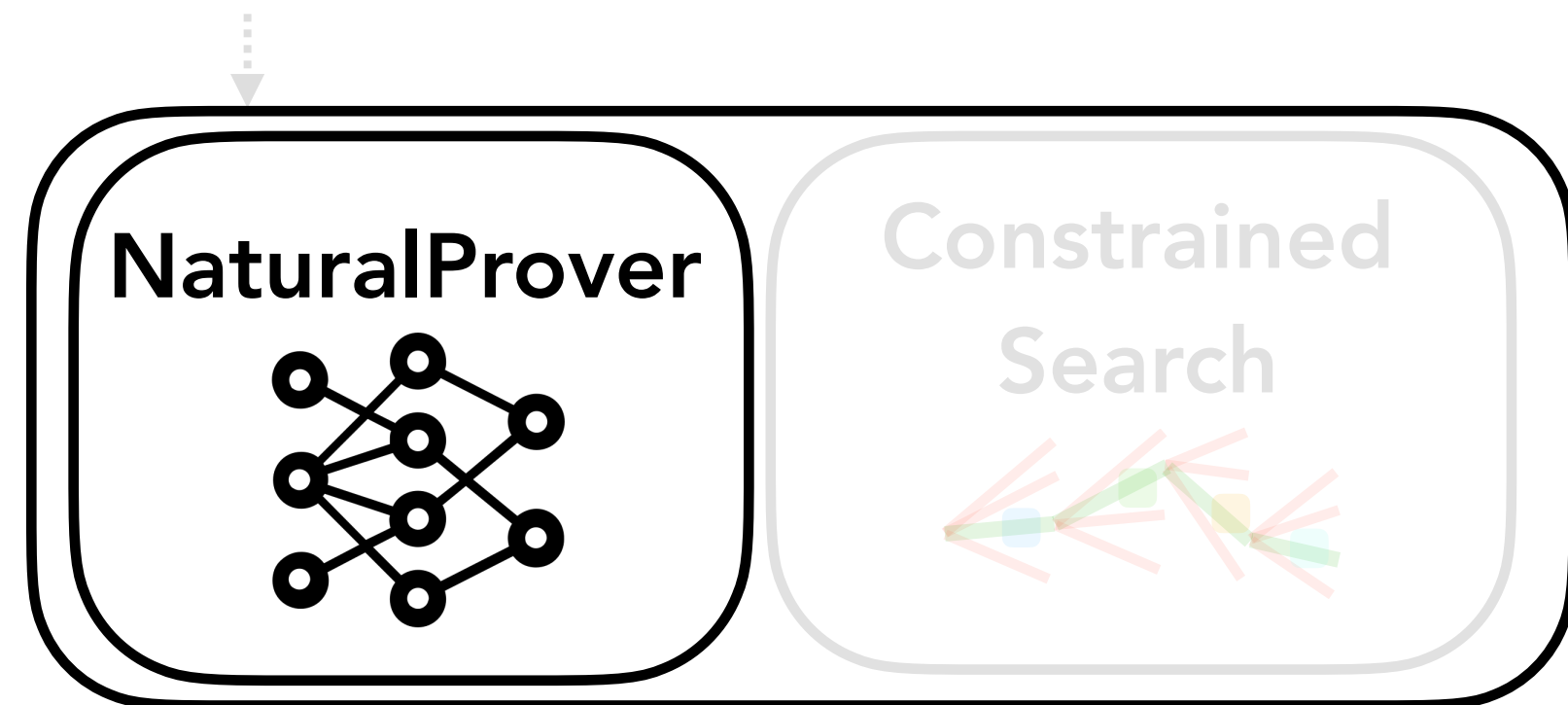


NaturalProver : knowledge

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Even Integer

Odd Integer

Integer

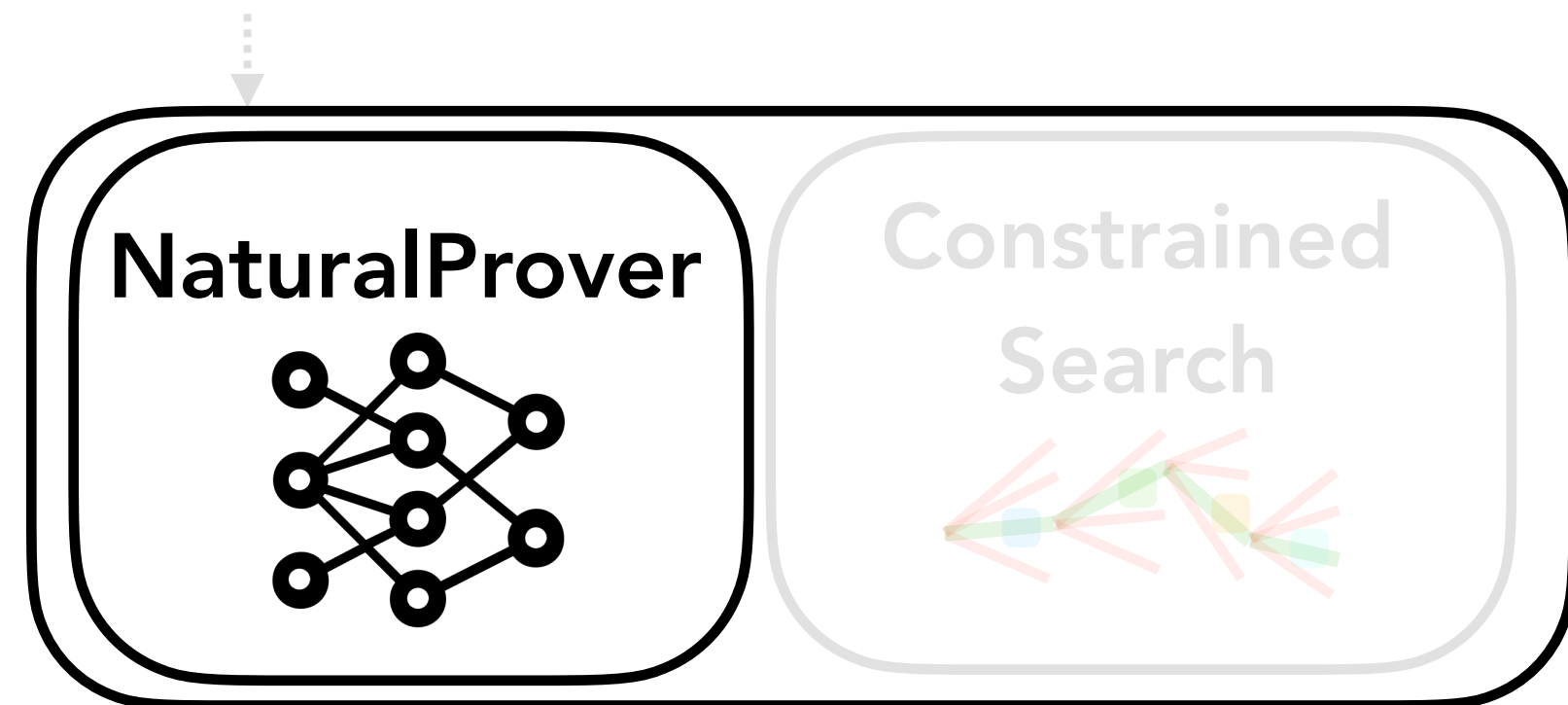
Proof by Contradiction

NaturalProver : knowledge

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Even Integer

Odd Integer

Integer

Proof by Contradiction

Definition

An integer $n \in \mathbb{Z}$ is **odd** if and only if:

$$x \equiv 1 \pmod{2}$$

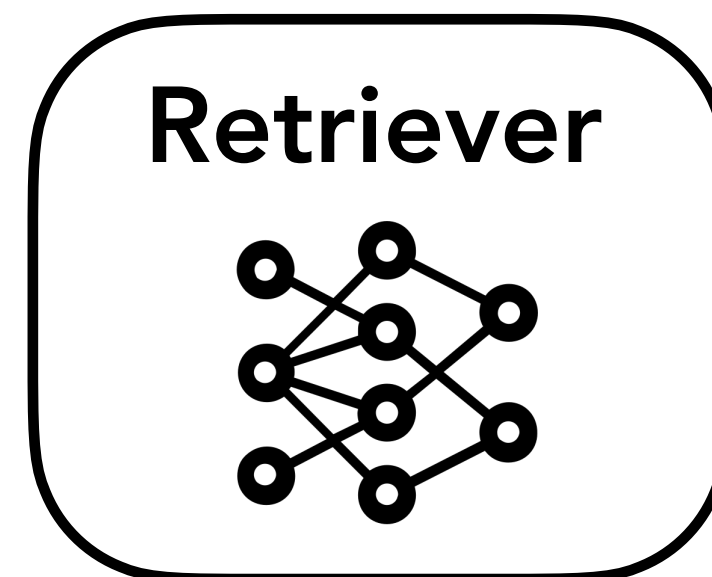
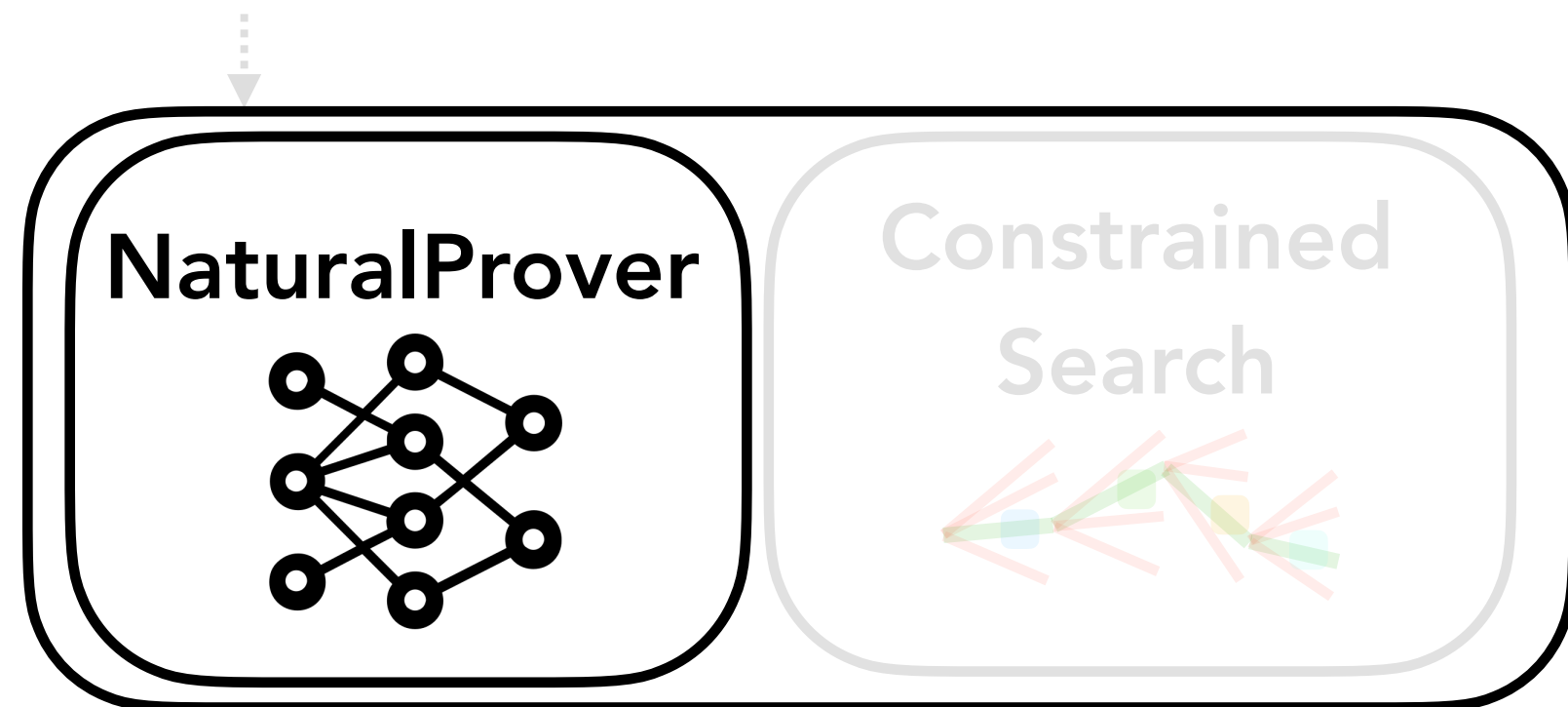
where the notation denotes congruence modulo 2.

NaturalProver : knowledge

Theorem

Let x be an even integer.

Then $x + 5$ is odd.

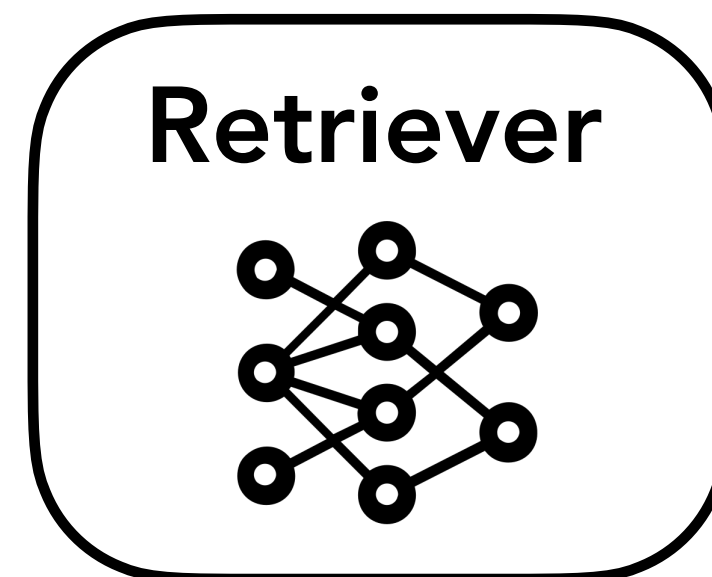
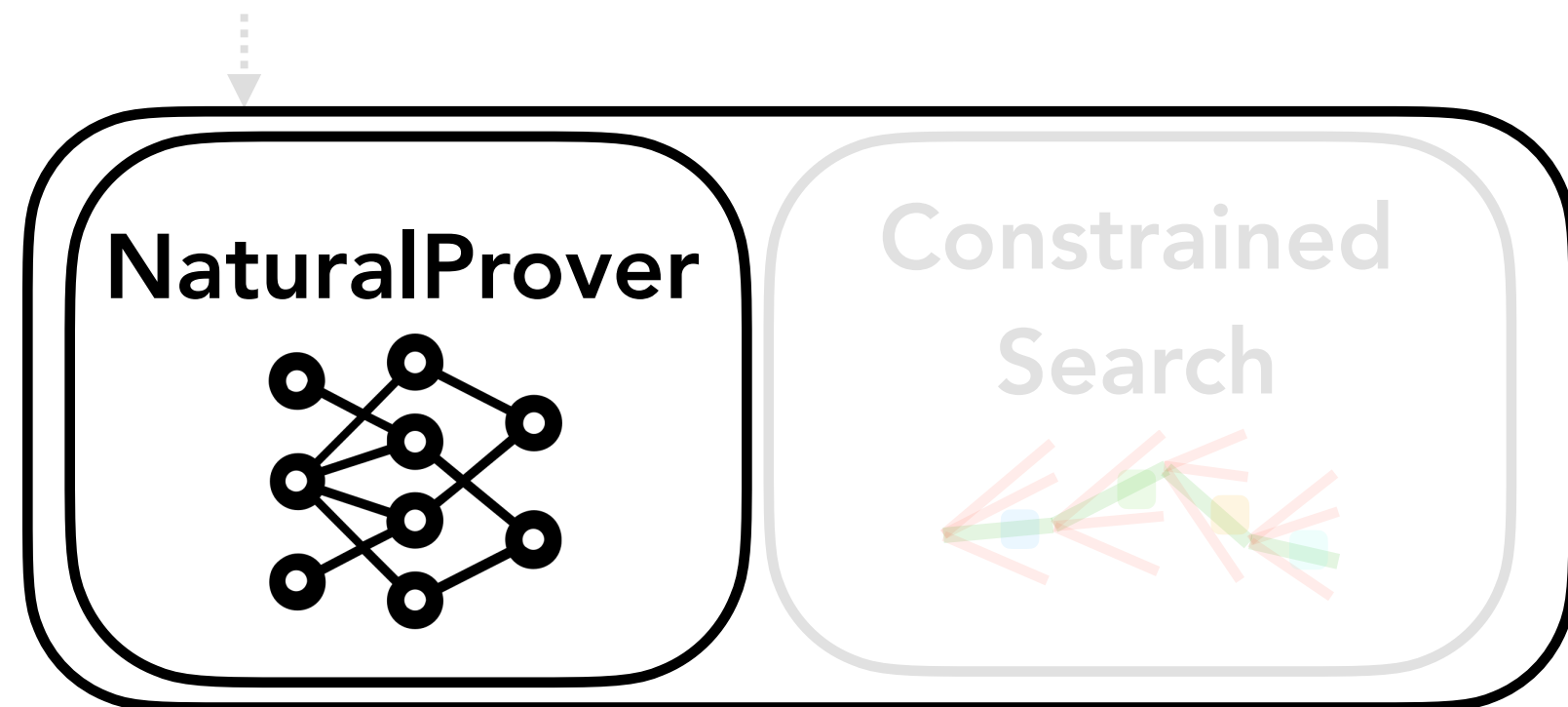


NaturalProver : knowledge

Theorem

Let x be an even integer.

Then $x + 5$ is odd.

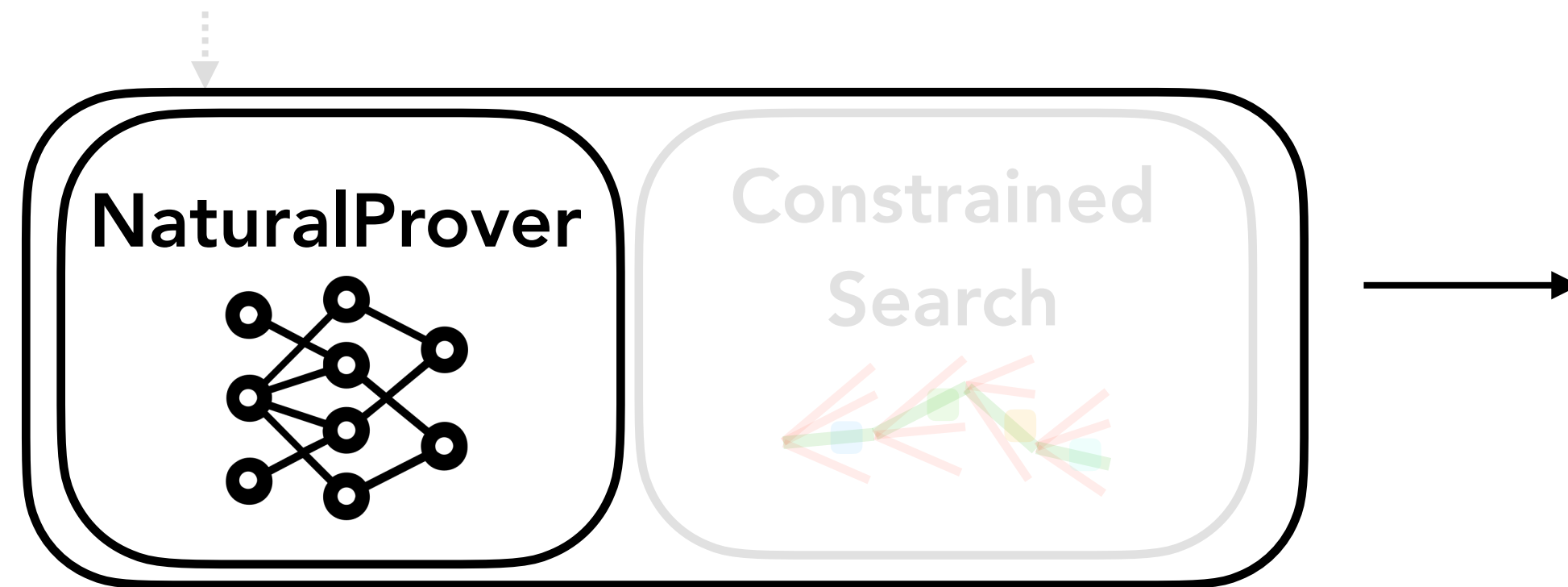


NaturalProver : constrained search

Theorem

Let x be an even integer.

Then $x + 5$ is odd.

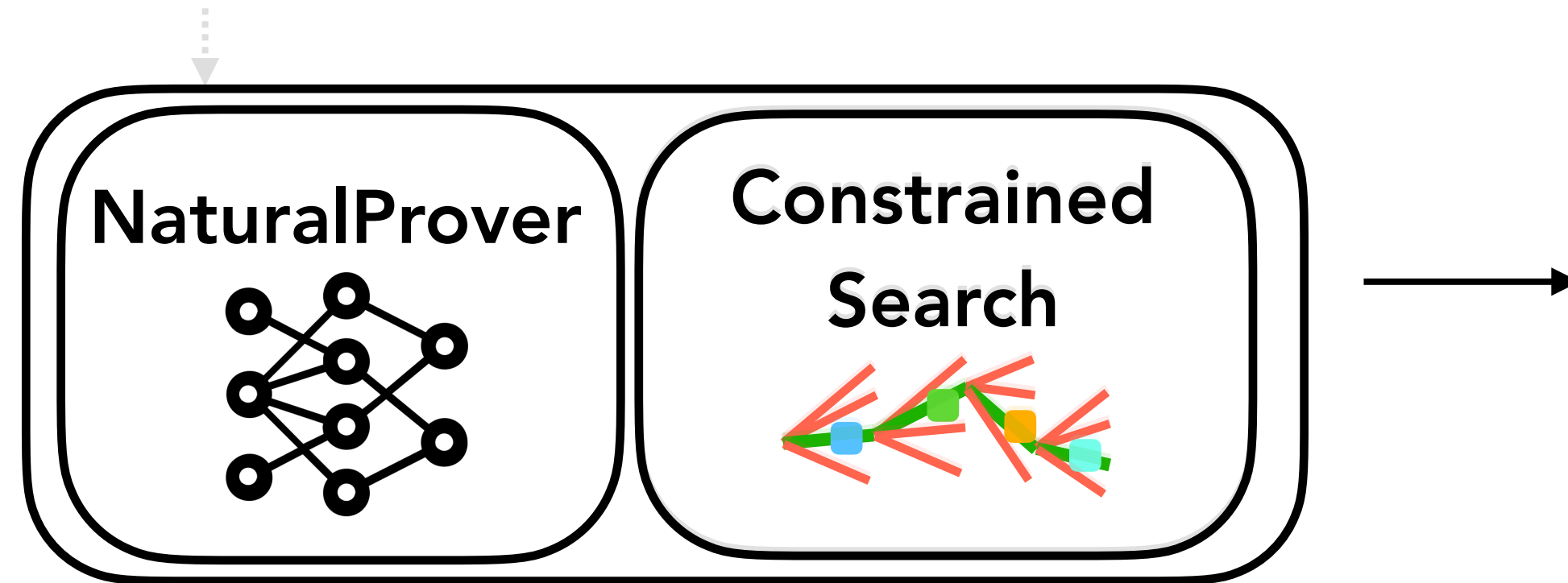


NaturalProver : constrained search

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



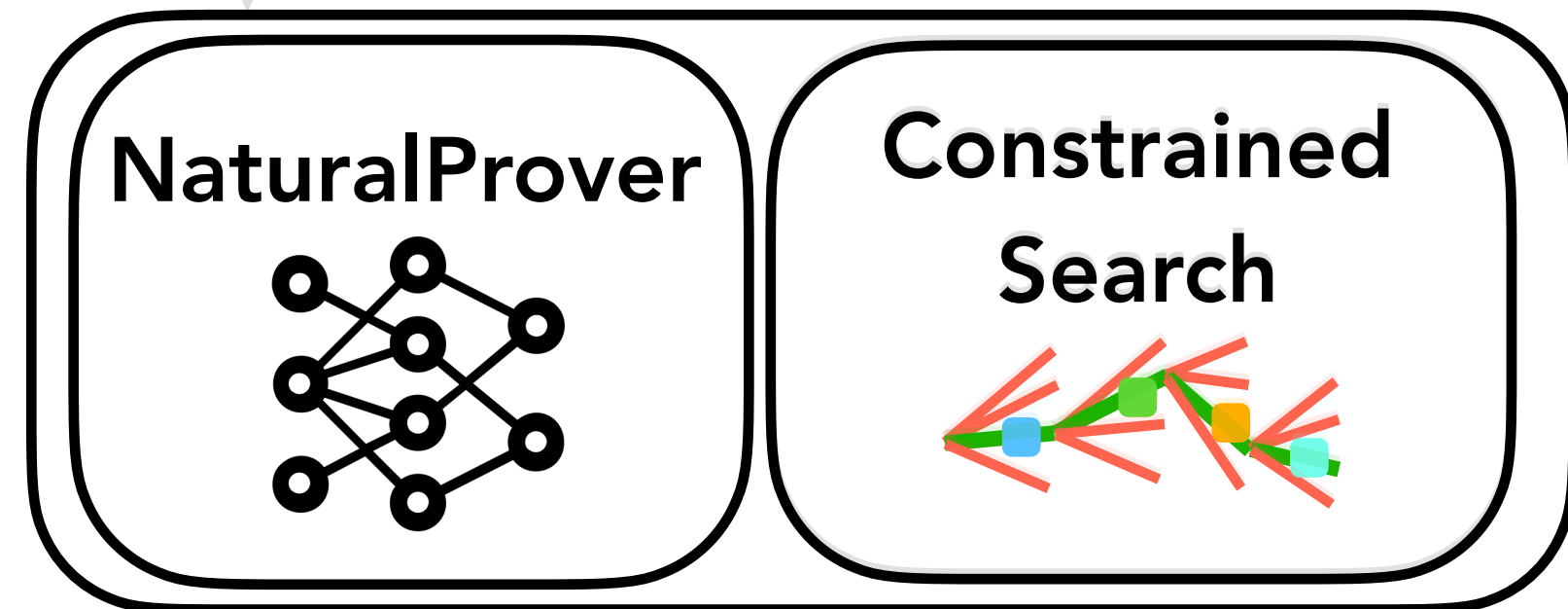
Even Integer
Odd Integer
Integer
Proof by Contradiction

NaturalProver : constrained search

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

Even Integer

Odd Integer

Integer

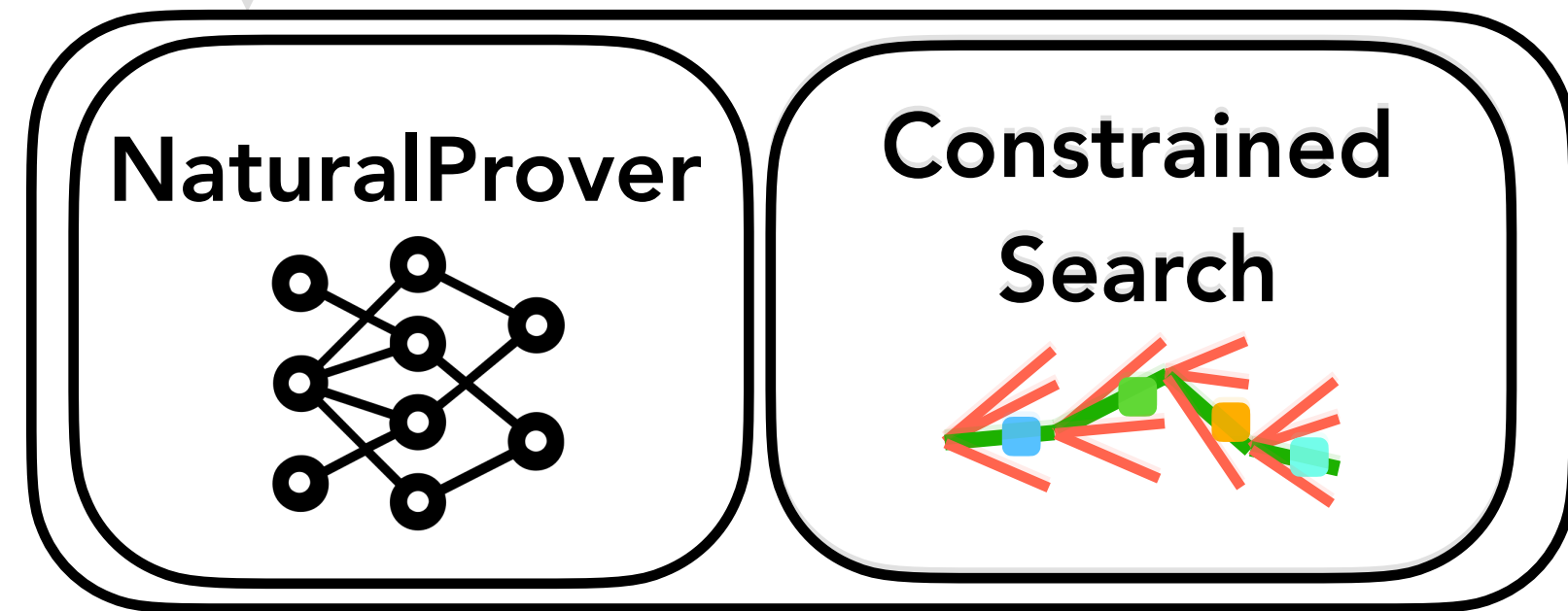
Proof by Contradiction

NaturalProver : constrained search

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

Then there exists an integer k such that $x + 5 = 2k$.

Even Integer

Odd Integer

Integer

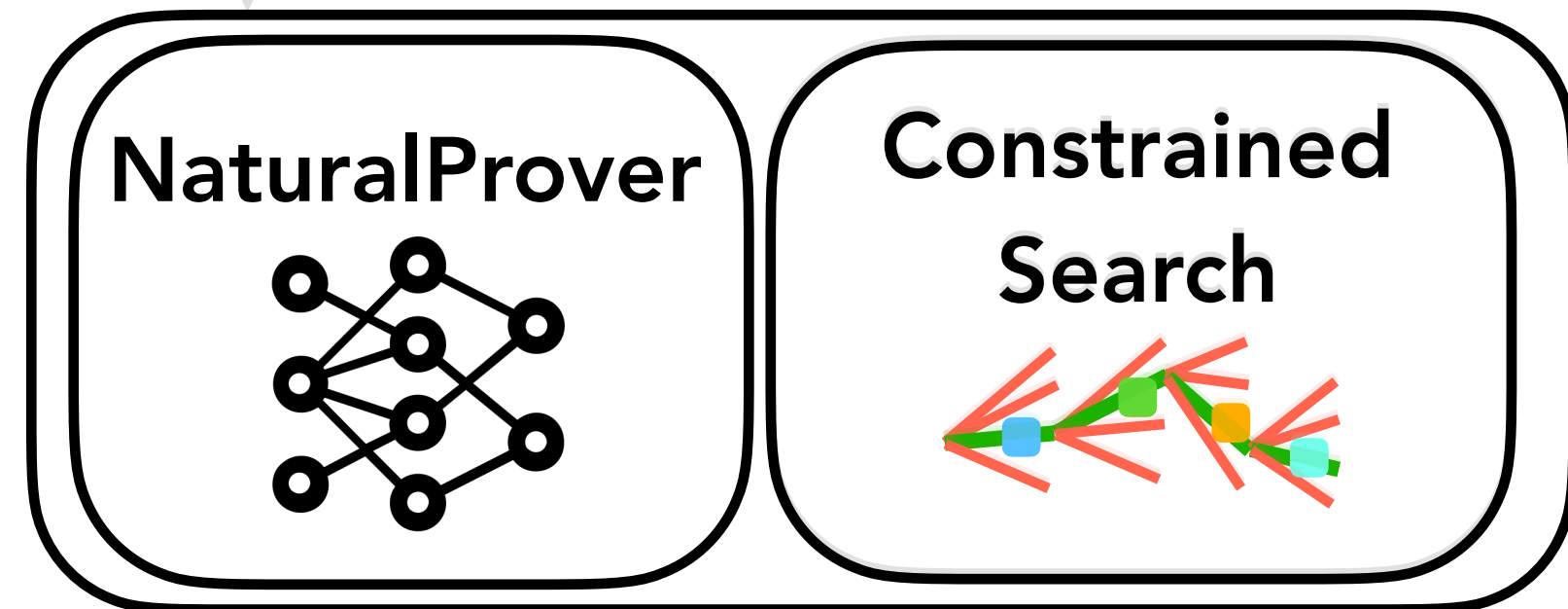
Proof by Contradiction

NaturalProver : constrained search

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

Then there exists an integer k such that $x + 5 = 2k$.

This contradicts the premise that x is even.

Even Integer

Odd Integer

Integer

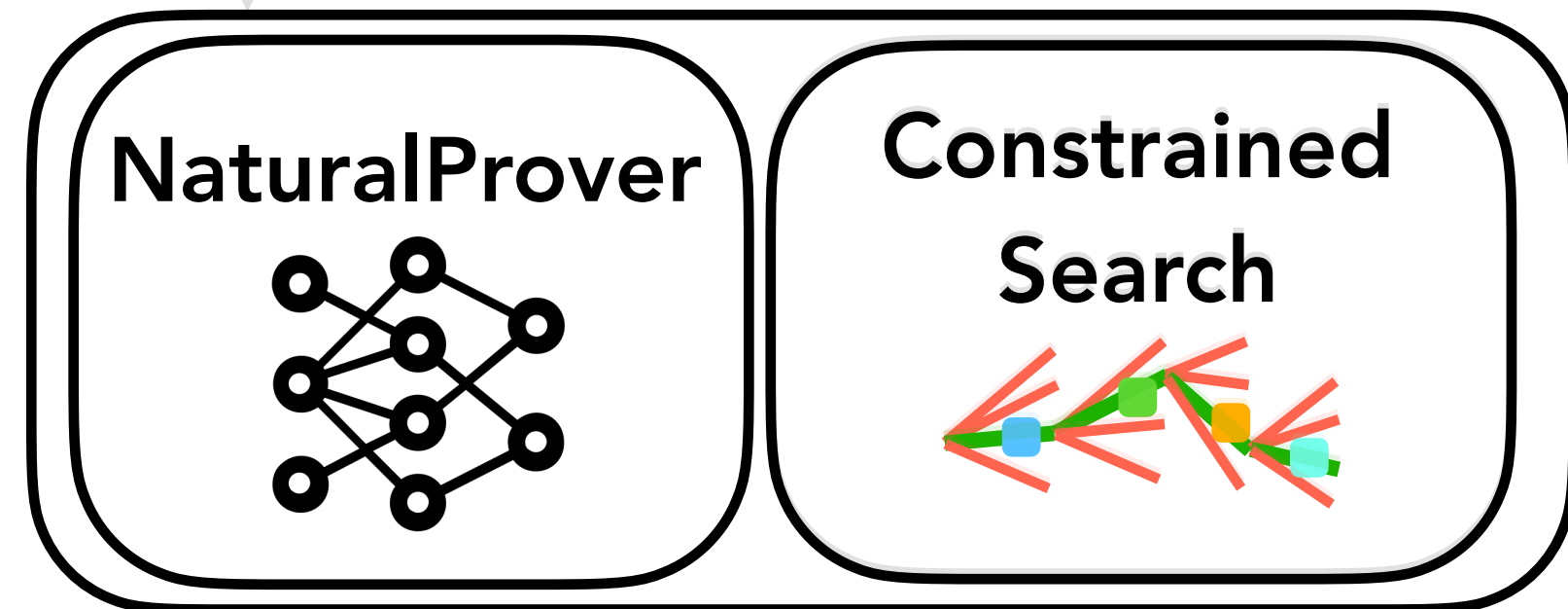
Proof by Contradiction

NaturalProver : constrained search

Theorem

Let x be an even integer.

Then $x + 5$ is odd.



Even Integer
Odd Integer
Integer
Proof by Contradiction

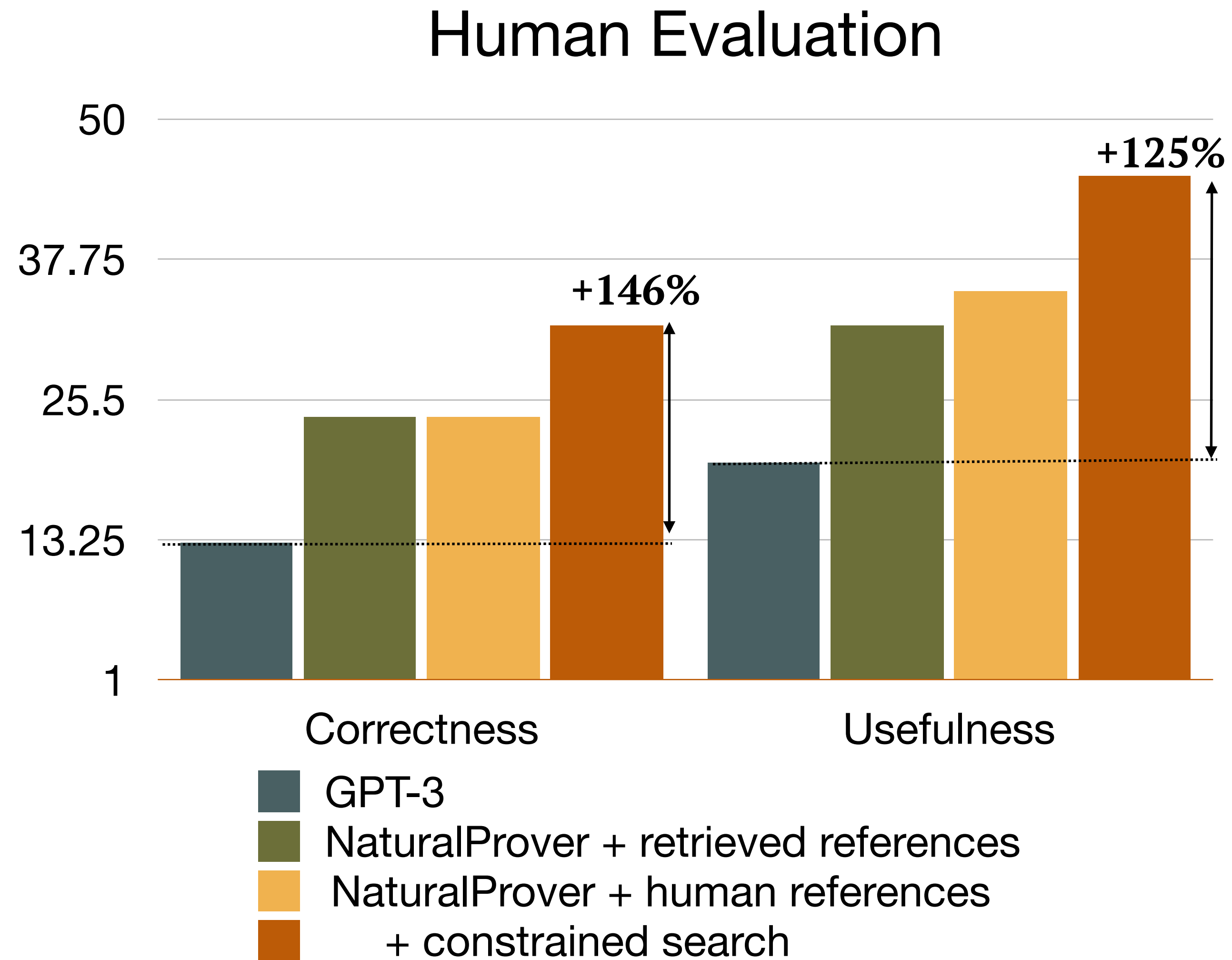
Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

Then there exists an integer k such that $x + 5 = 2k$.



This contradicts the premise that x is even.

Hence, by Proof by Contradiction, $x + 5$ is odd.

Correct and useful?



Human-machine collaboration

 — Generated suggestion
 — Human selection

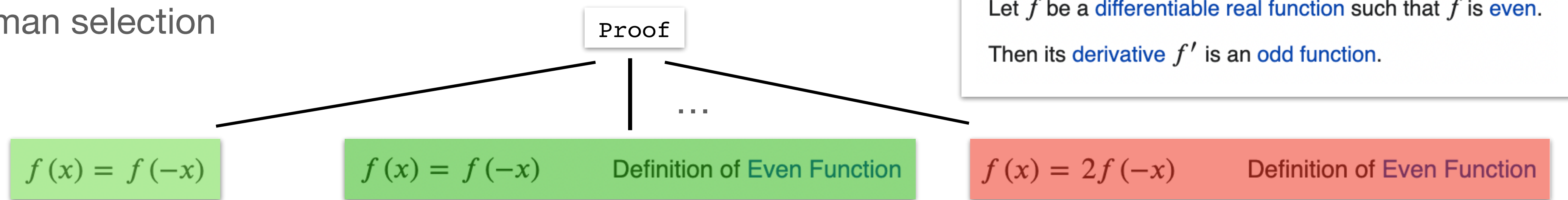
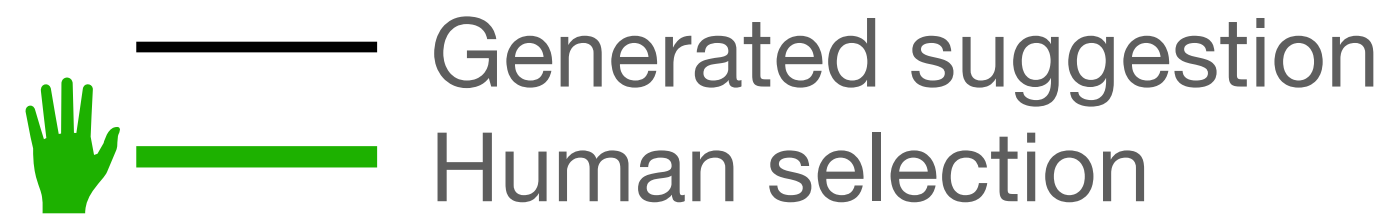
Proof

Derivative of Even Function is Odd



Let f be a [differentiable real function](#) such that f is [even](#).

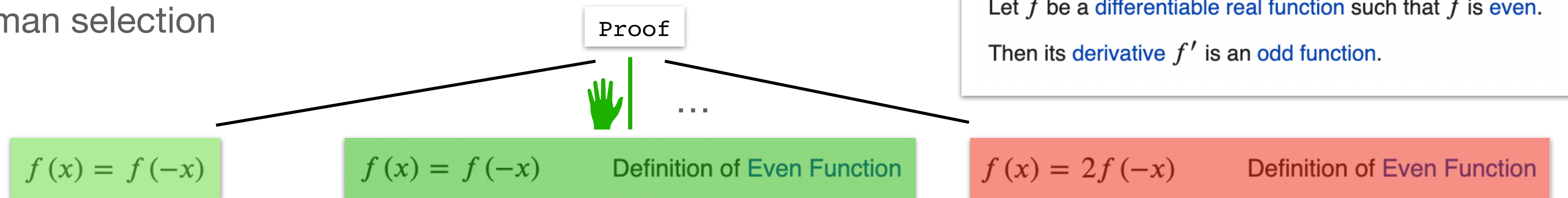
Then its [derivative](#) f' is an [odd function](#).

Human-machine collaboration





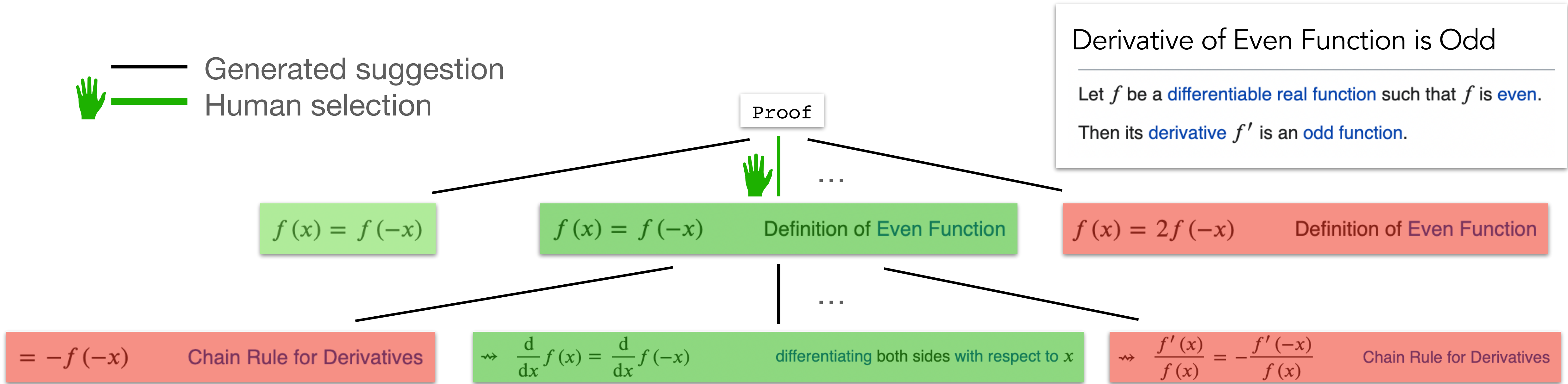
Human-machine collaboration

 Generated suggestion
 Human selection



Human-machine collaboration



 Generated suggestion
 Human selection

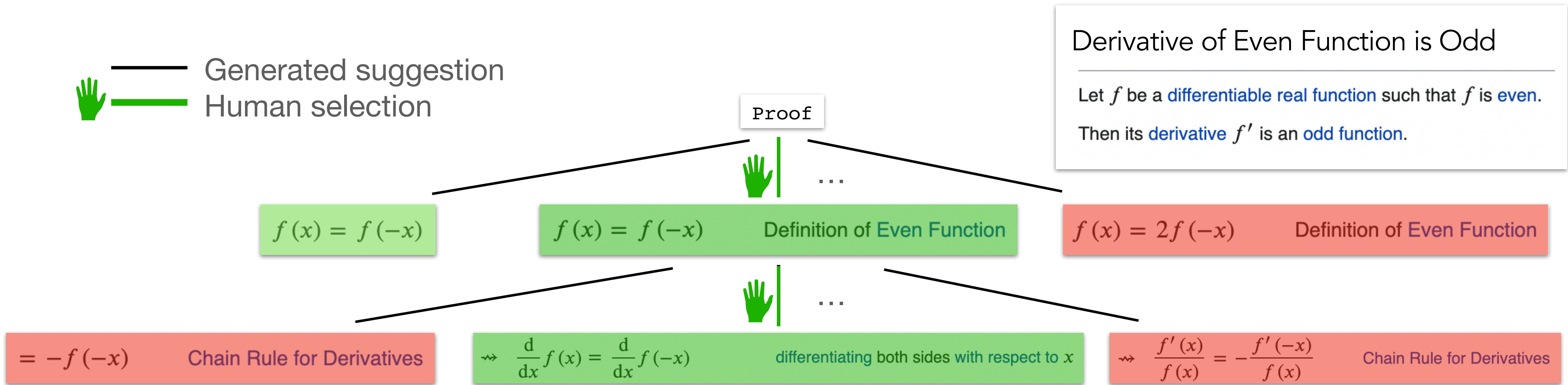


Derivative of Even Function is Odd



Let f be a differentiable real function such that f is even.
Then its derivative f' is an odd function.

Human-machine collaboration

 — Generated suggestion
 — Human selection

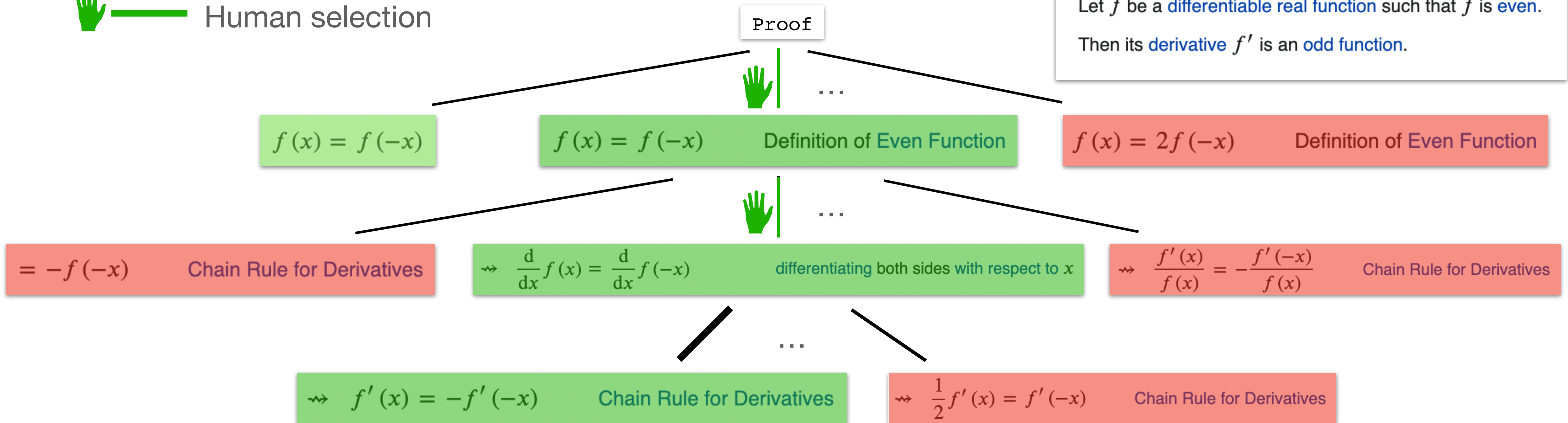


Human-machine collaboration



 Generated suggestion
 Human selection

Derivative of Even Function is Odd

Let f be a differentiable real function such that f is even.
Then its derivative f' is an odd function.

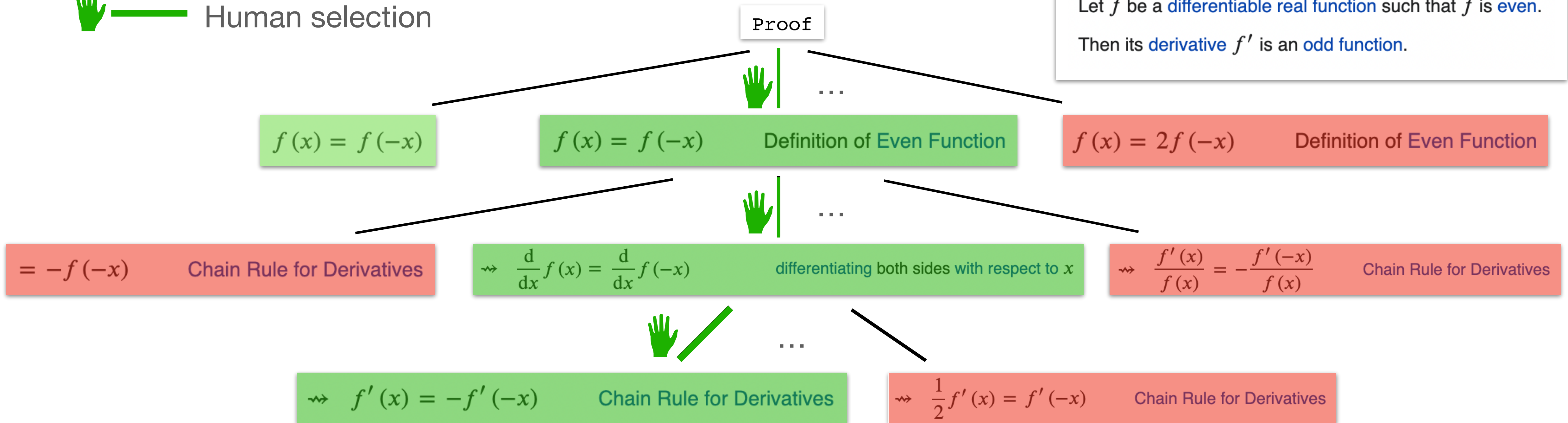


Human-machine collaboration



 Generated suggestion
 Human selection

Derivative of Even Function is Odd

Let f be a differentiable real function such that f is even.
Then its derivative f' is an odd function.

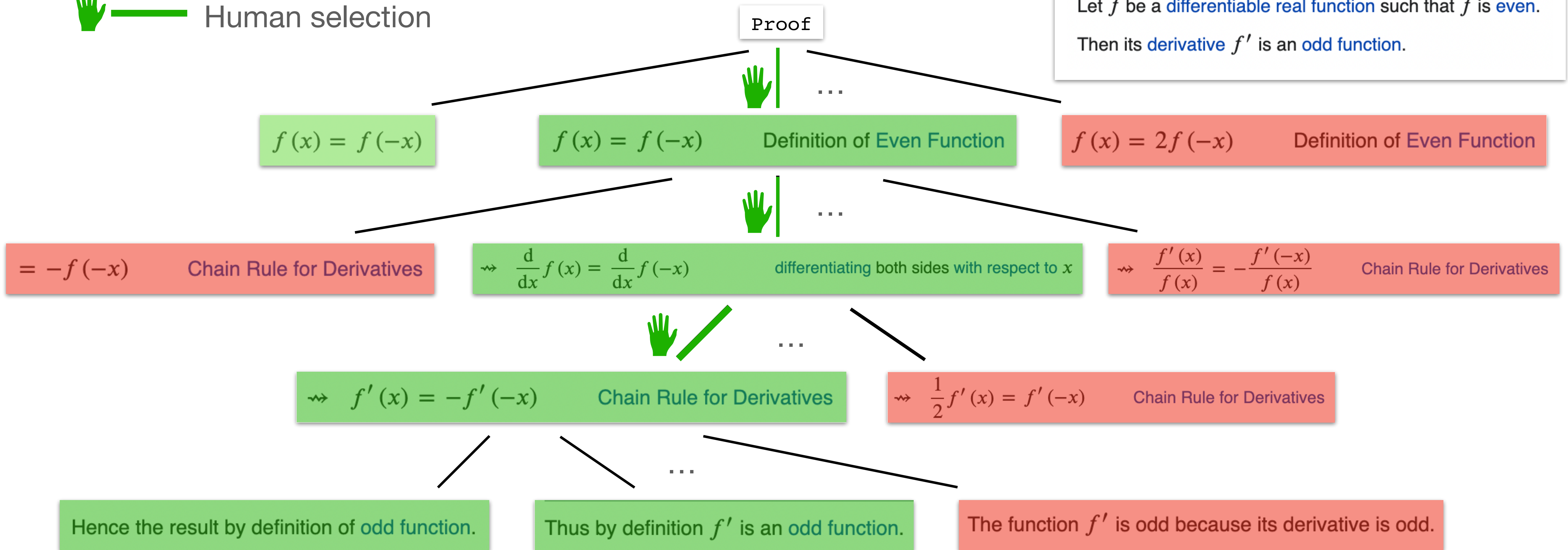


Human-machine collaboration


 Generated suggestion
 Human selection

Derivative of Even Function is Odd

Let f be a differentiable real function such that f is even.
Then its derivative f' is an odd function.

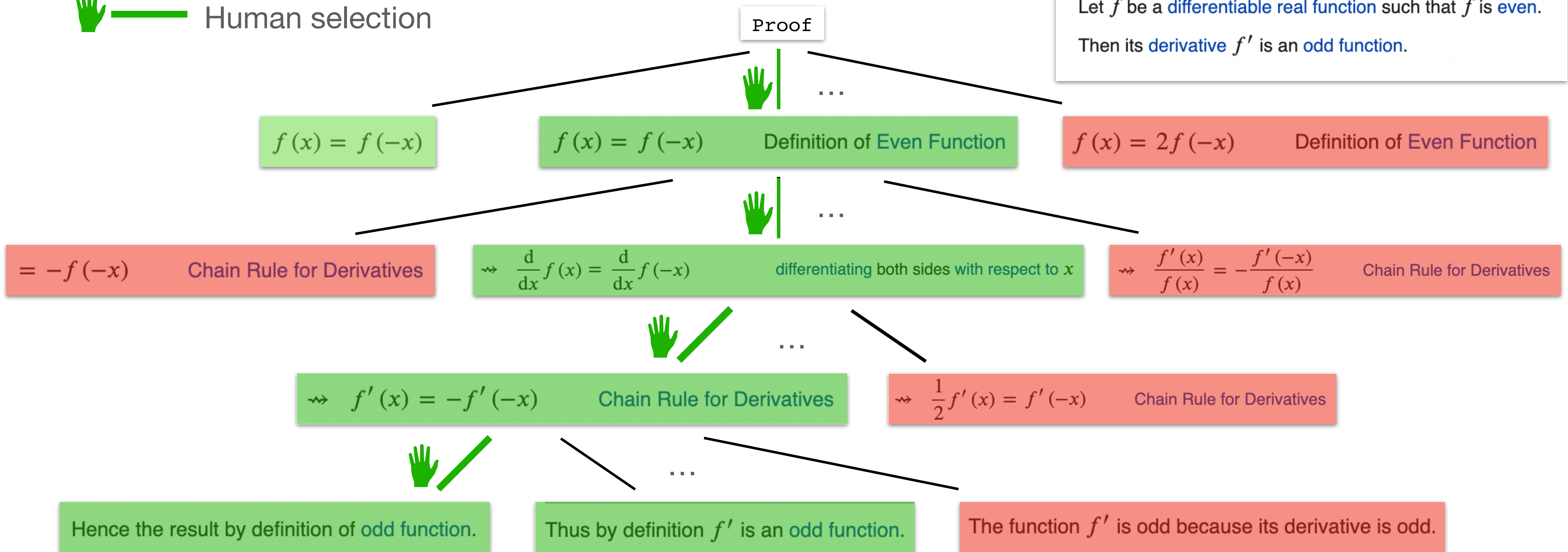


Human-machine collaboration



 Generated suggestion
 Human selection

Derivative of Even Function is Odd

Let f be a differentiable real function such that f is even.
Then its derivative f' is an odd function.

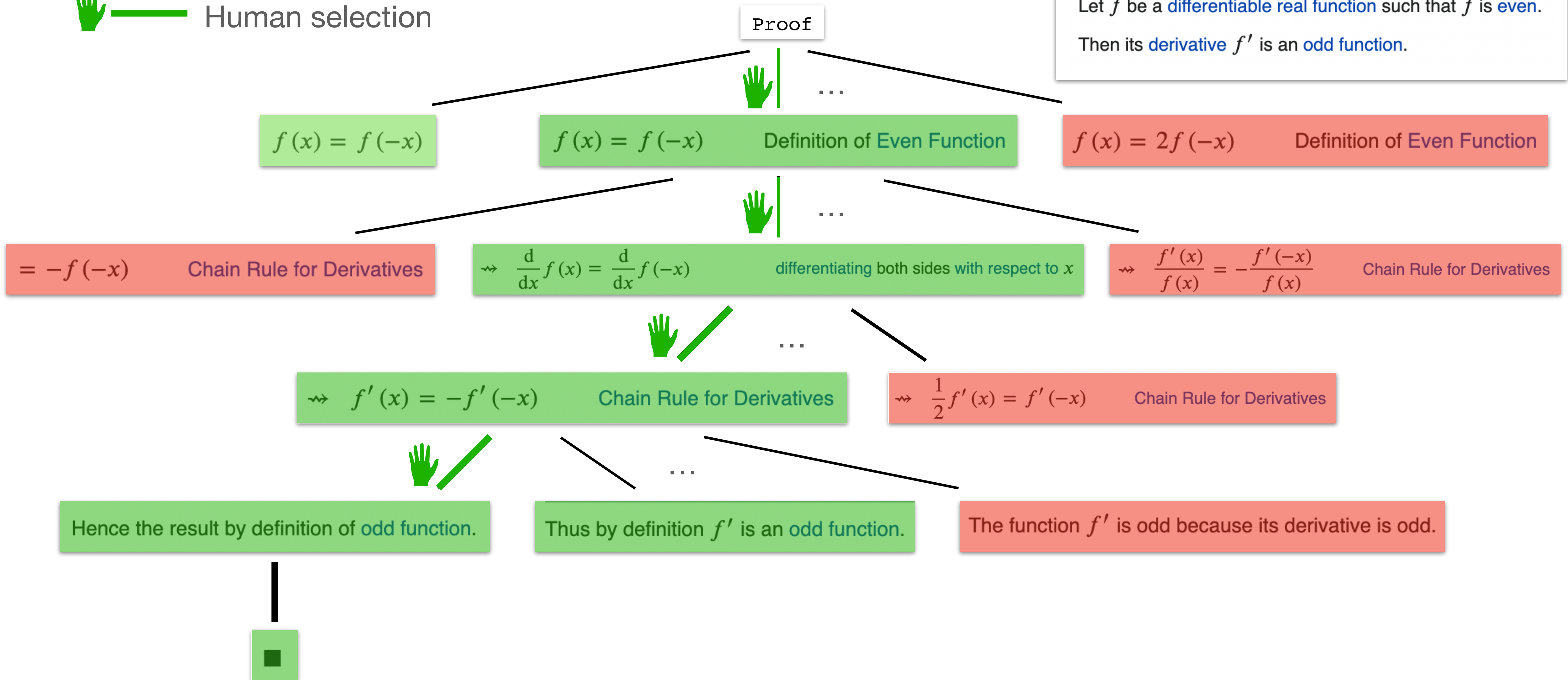


Human-machine collaboration



 Generated suggestion
 Human selection

Derivative of Even Function is Odd

Let f be a differentiable real function such that f is even.
Then its derivative f' is an odd function.

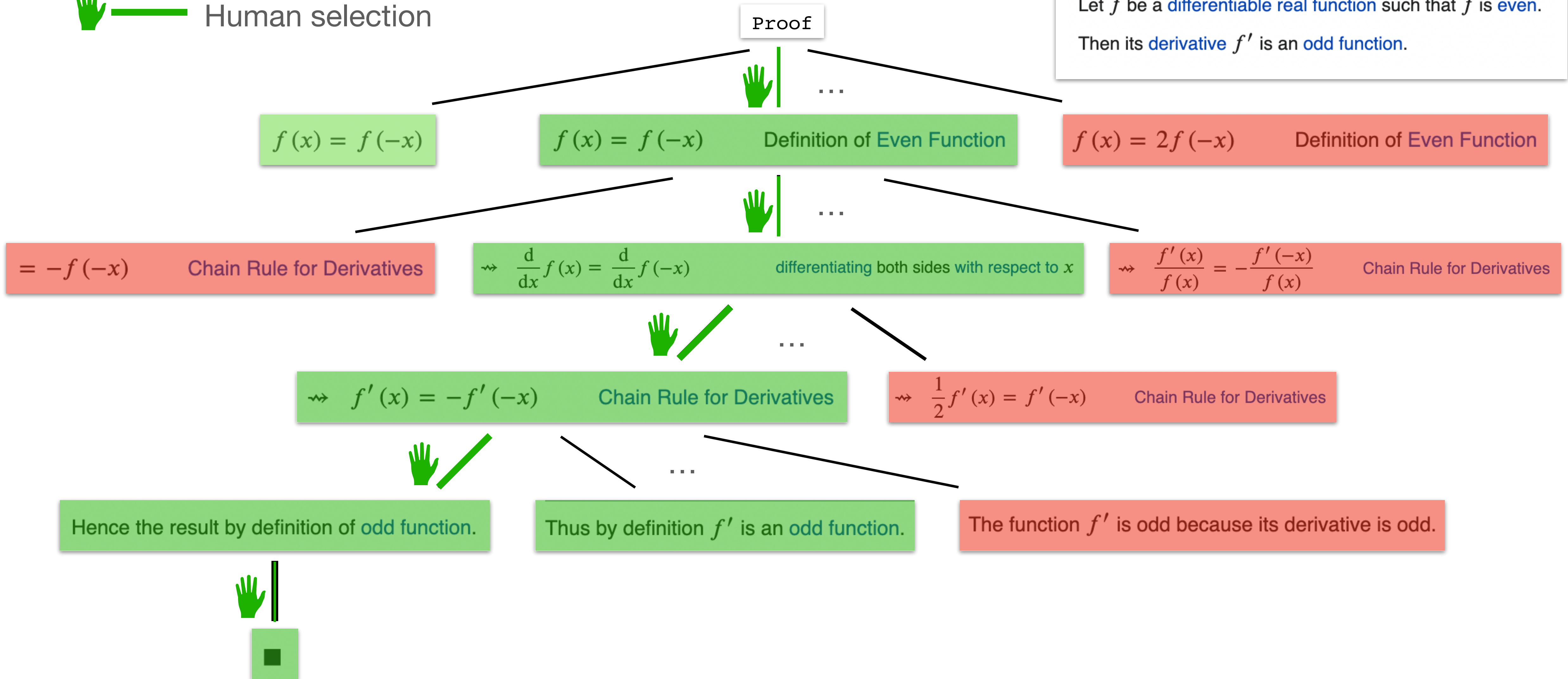


Human-machine collaboration

 Generated suggestion
 Human selection

Derivative of Even Function is Odd

Let f be a differentiable real function such that f is even.
Then its derivative f' is an odd function.



Human-machine collaboration

Derivative of Even Function is Odd

Let f be a differentiable real function such that f is even.

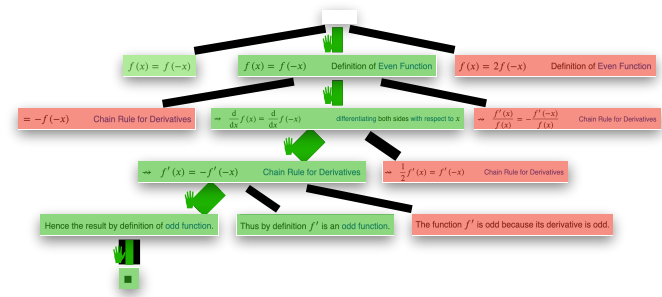
Then its derivative f' is an odd function.

$$f(x) = f(-x) \quad \text{Definition of Even Function}$$

$$\rightsquigarrow \frac{d}{dx} f(x) = \frac{d}{dx} f(-x) \quad \text{differentiating both sides with respect to } x$$

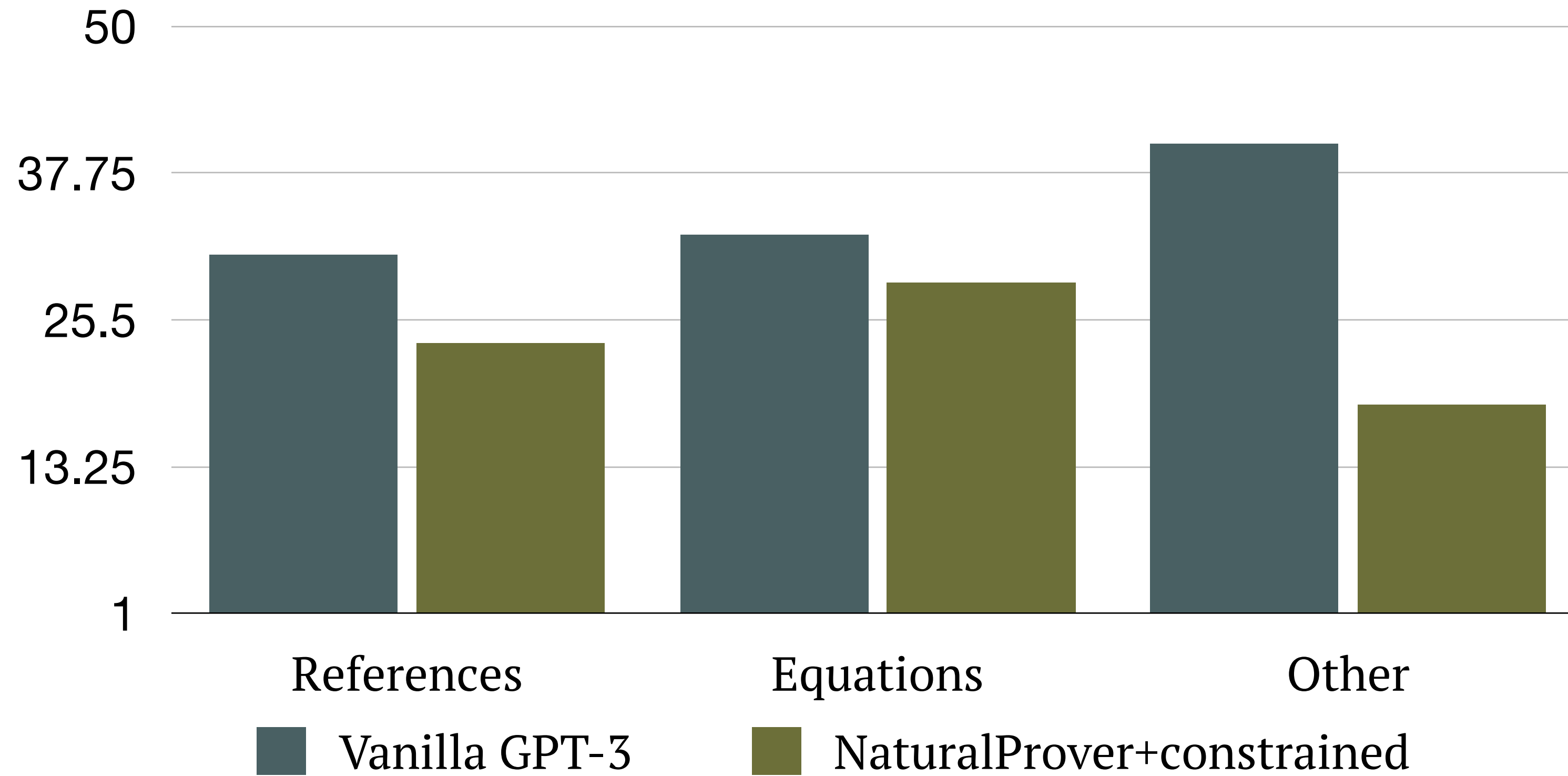
$$\rightsquigarrow f'(x) = -f'(-x) \quad \text{Chain Rule for Derivatives}$$

Hence the result by definition of odd function.



Capable vs. trustworthy

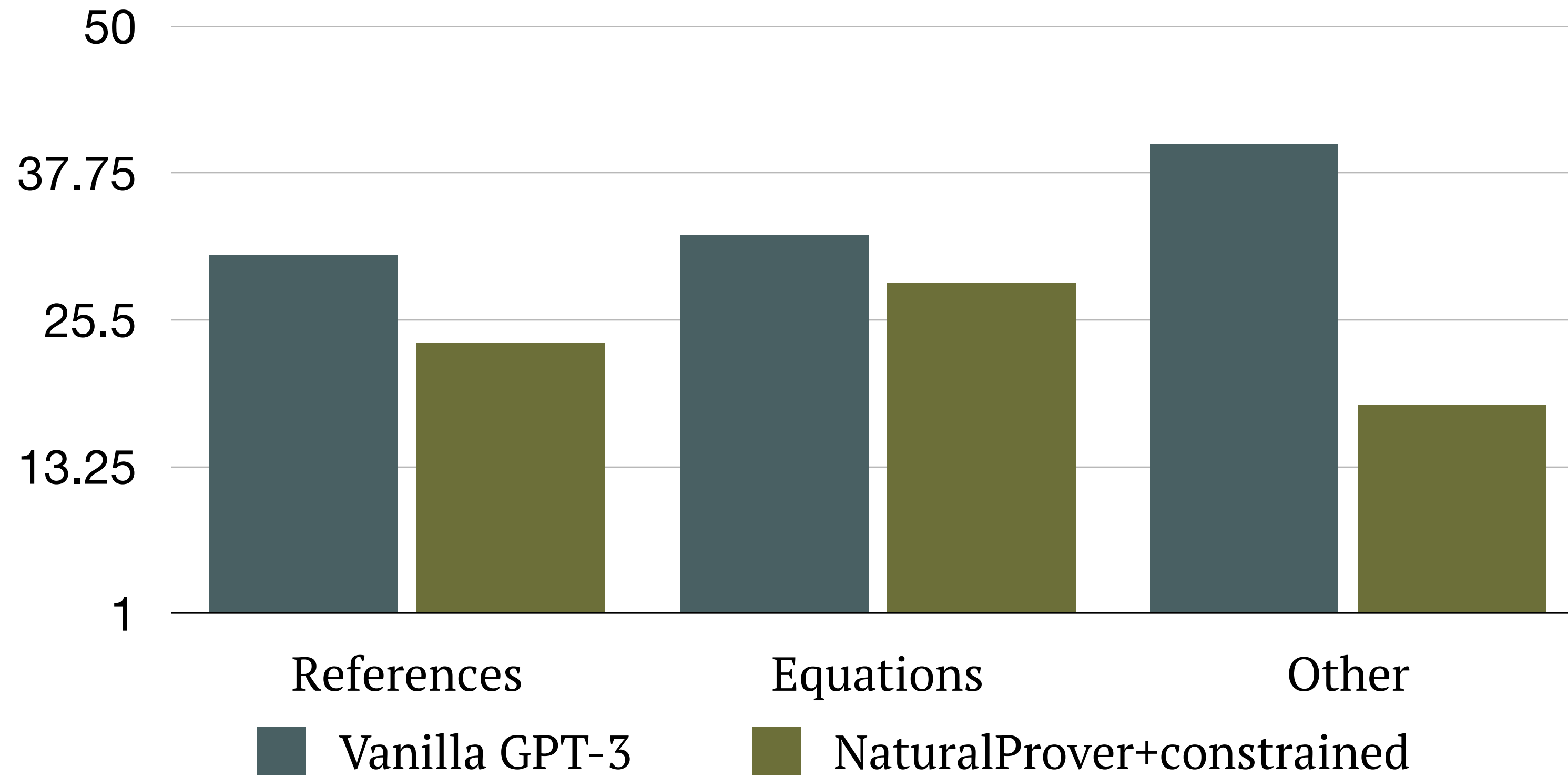
Reasoning Errors (↓)



← Informal →

Capable vs. trustworthy

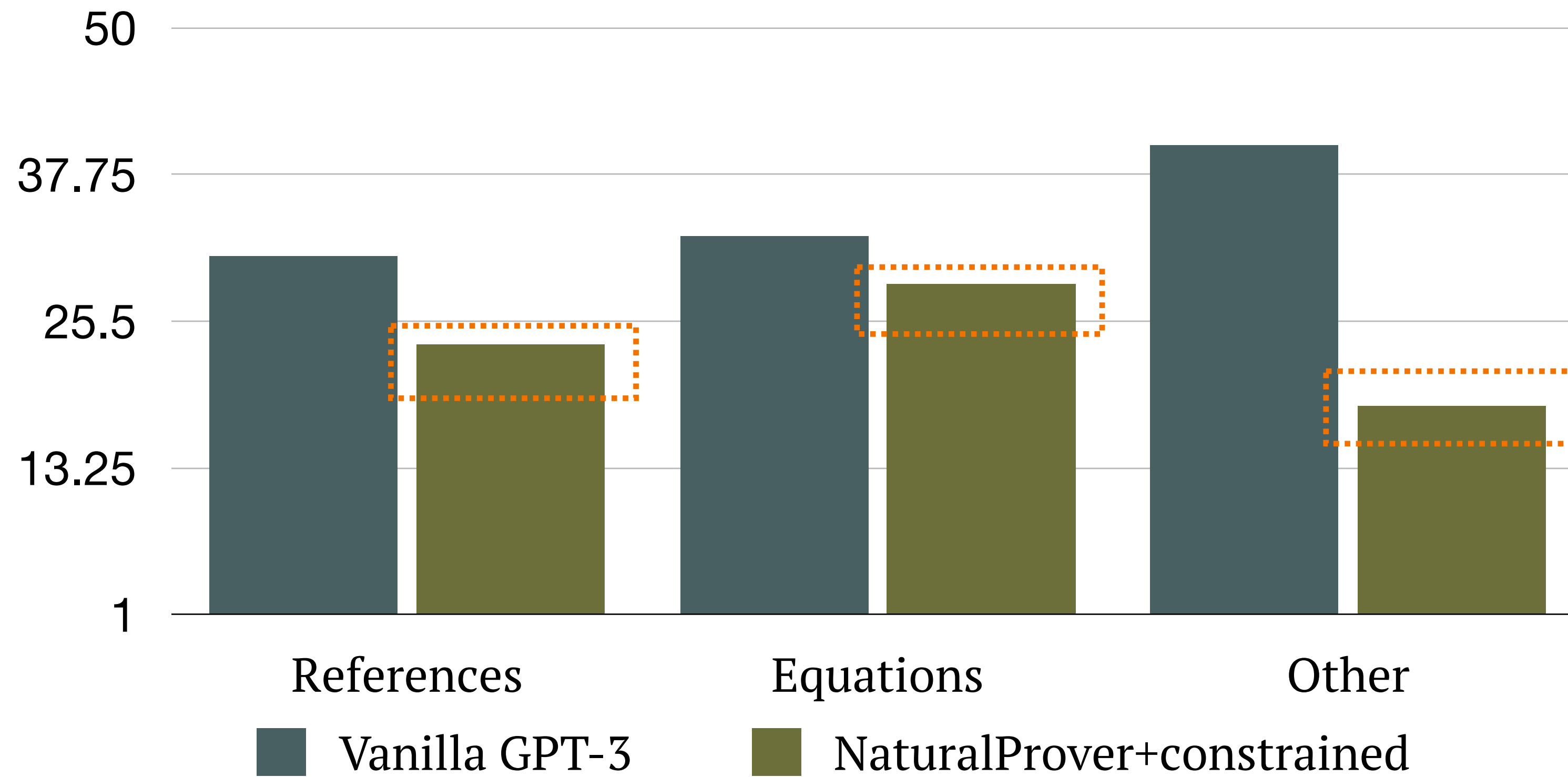
Reasoning Errors (↓)



← Informal →

Capable vs. trustworthy

Reasoning Errors (↓)



← Informal →

Capable vs. trustworthy



Informal

Capable vs. trustworthy

Proof:

$$0 < 1$$

$$\rightsquigarrow -1 < 0$$

Real Zero is Less than Real One

Order of Real Numbers is Dual of Order of their Negatives

Informal



Capable vs. trustworthy

Proof:

$$0 < 1$$

$$\rightsquigarrow -1 < 0$$

Proof:

$$0 < 1$$

$$\rightsquigarrow -0 > -1$$

Real Zero is Less than Real One

Order of Real Numbers is Dual of Order of their Negatives

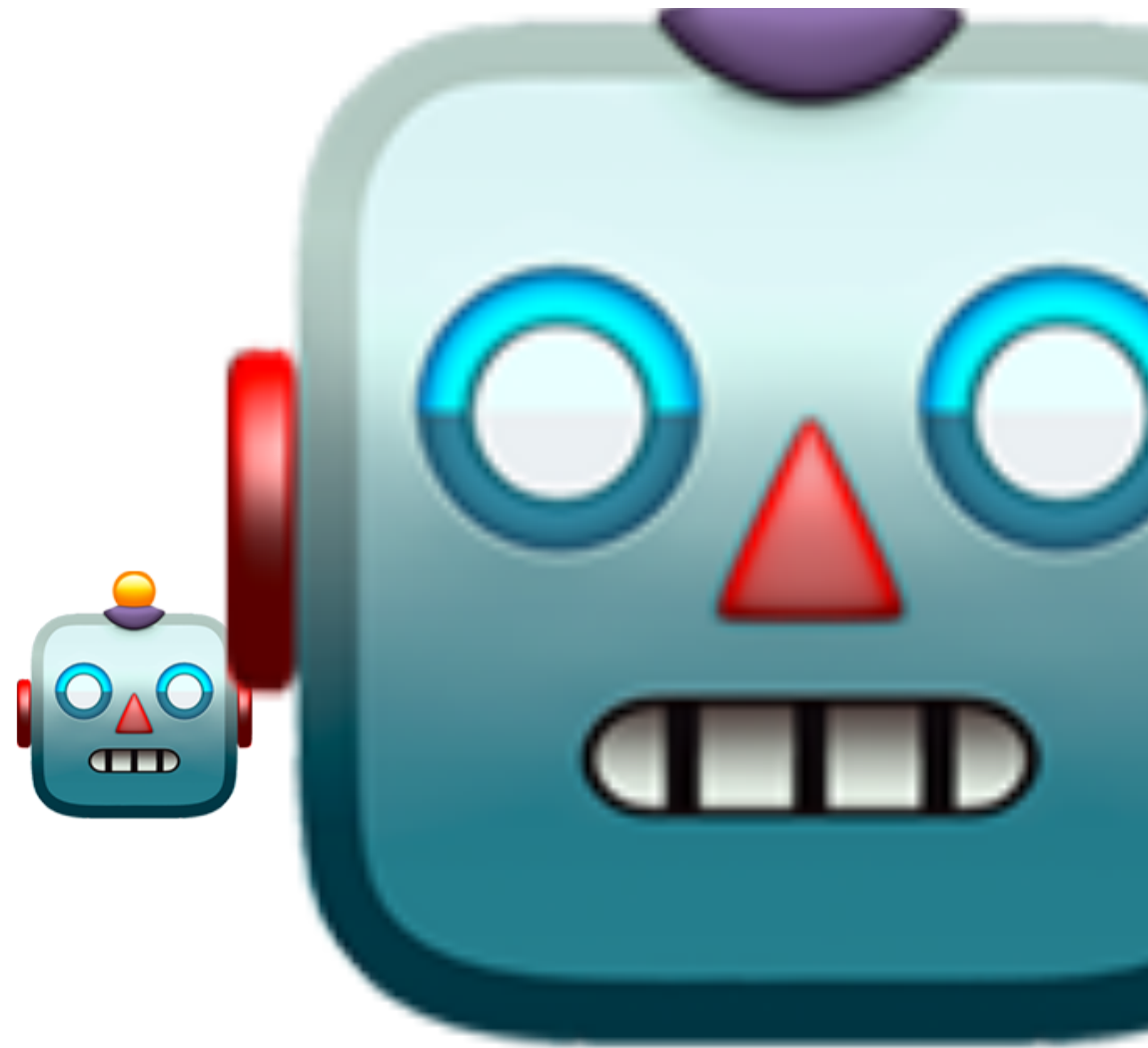
Real Zero is Less than Real One

Order of Real Numbers is Dual of Order of their Negatives



Informal

Capable vs. trustworthy



Minerva

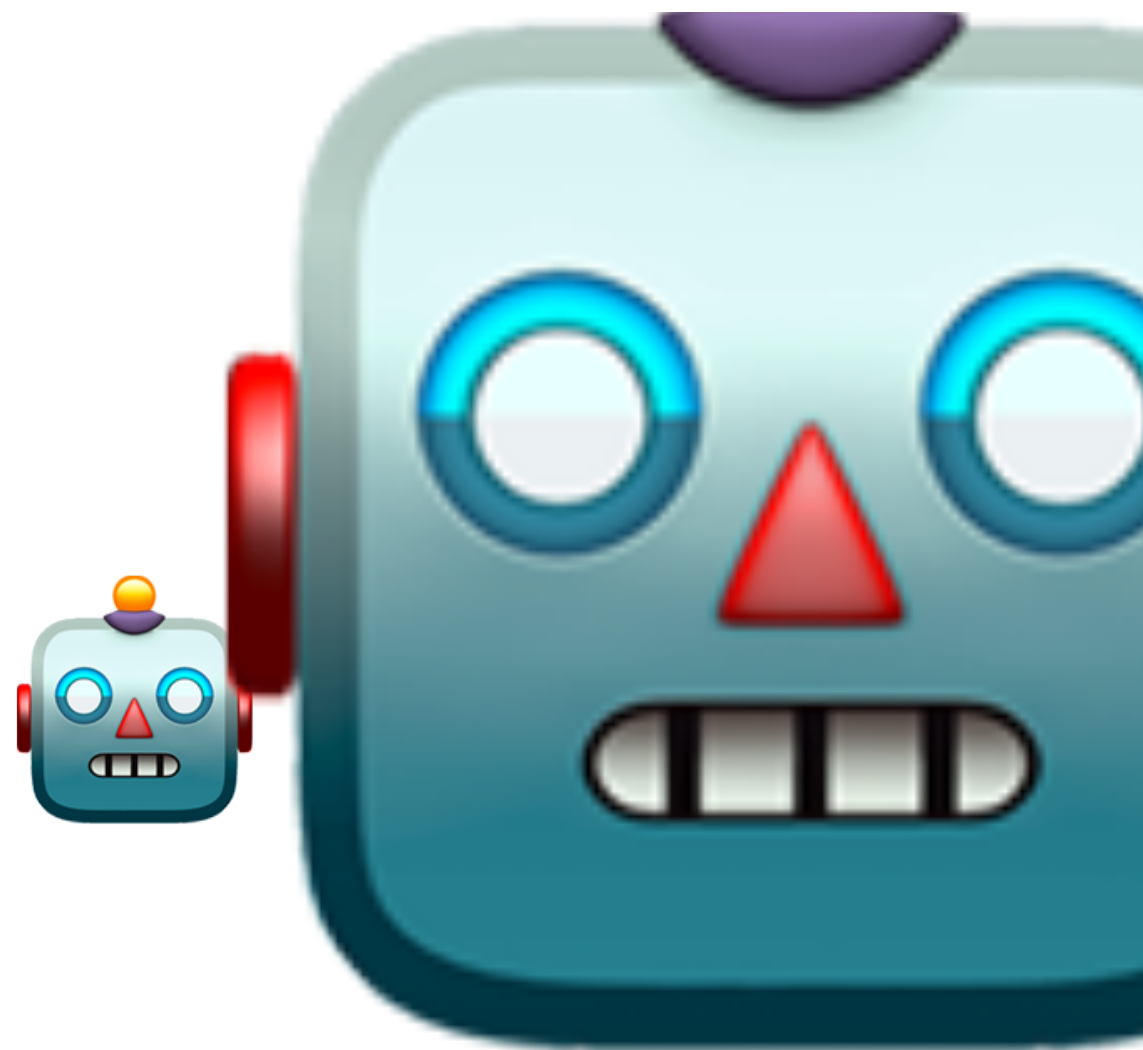
540 billion params



Informal

Capable vs. trustworthy

Find the greatest common divisor of 3339, 2961, 1491.



Minerva

540 billion params

QUESTION Find the greatest common divisor of 3339, 2961, and 1491.

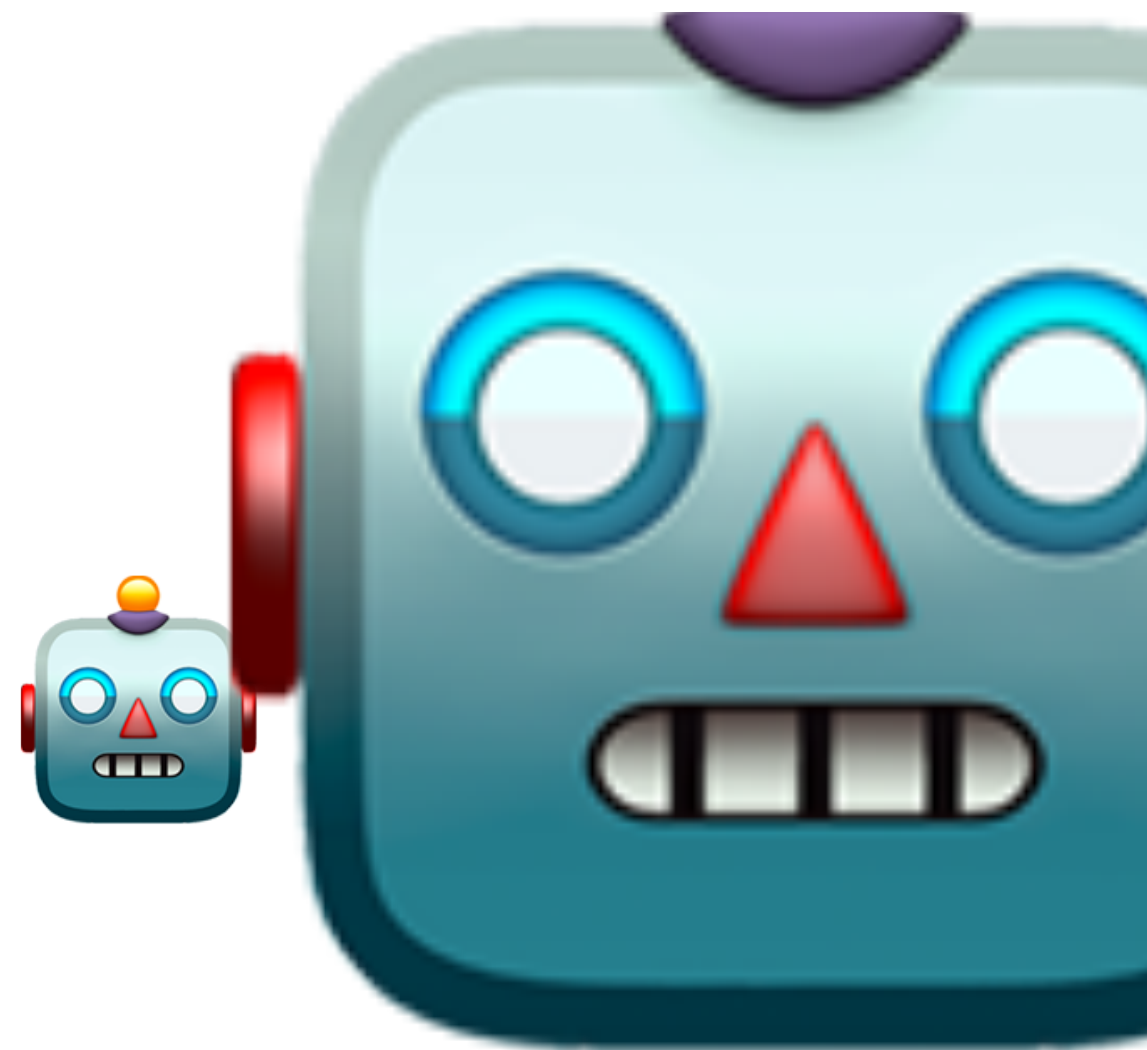
MODEL ANSWER The prime factorizations of 3339, 2961, and 1491 are

$$3339 = 3 \cdot 7 \cdot 13 \cdot 19,$$
$$2961 = 3 \cdot 7 \cdot 11 \cdot 17,$$
$$1491 = 3 \cdot 7 \cdot 19 \cdot 7.$$

The greatest common divisor of 3339, 2961, and 1491 is the product of the common prime factors, which is $3 \cdot 7 = 21$. Final Answer: The final answer is 21.

Informal

Capable vs. trustworthy



Minerva

540 billion params



QUESTION Find the greatest common divisor of 3339, 2961, and 1491.

MODEL ANSWER The prime factorizations of 3339, 2961, and 1491 are

$$3339 = 3 \cdot 7 \cdot 13 \cdot 19,$$
$$2961 = 3 \cdot 7 \cdot 11 \cdot 17,$$
$$1491 = 3 \cdot 7 \cdot 19 \cdot 7.$$

The greatest common divisor of 3339, 2961, and 1491 is the product of the common prime factors, which is $3 \cdot 7 = 21$. Final Answer: The final answer is 21.

Find the greatest common divisor of 3339, 2961, 1491.

Final answer: 21.

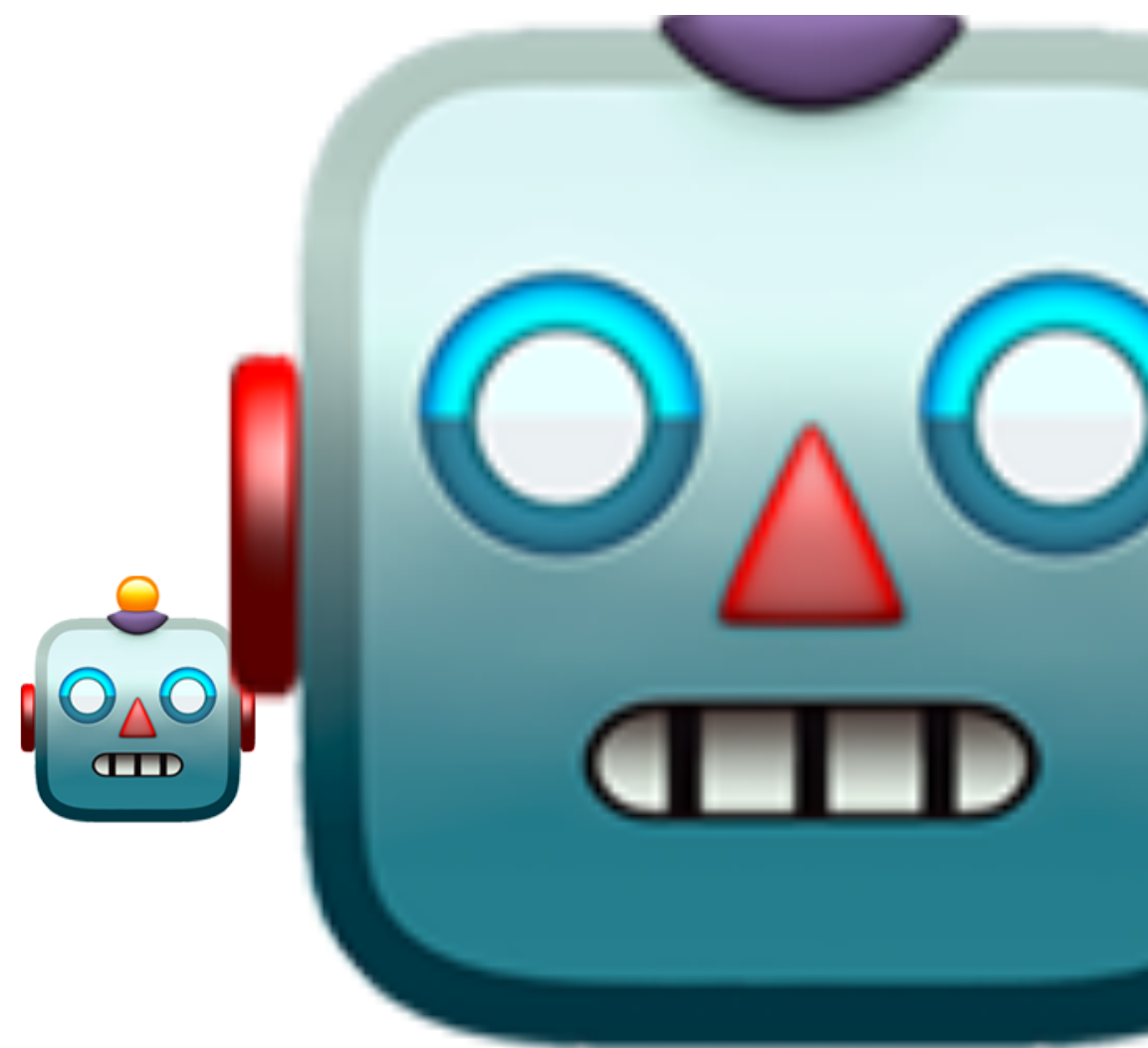
Correct answer



Informal



Capable vs. trustworthy



Minerva
540 billion params



QUESTION Find the greatest common divisor of 3339, 2961, and 1491.

MODEL ANSWER The prime factorizations of 3339, 2961, and 1491 are

$$\begin{aligned} 3339 &= 3 \cdot 7 \cdot 13 \cdot 19, \\ 2961 &= 3 \cdot 7 \cdot 11 \cdot 17, \\ 1491 &= 3 \cdot 7 \cdot 19 \cdot 7. \end{aligned}$$

The greatest common divisor of 3339, 2961, and 1491 is the product of the common prime factors, which is $3 \cdot 7 = 21$. Final Answer: The final answer is 21.

Find the greatest common divisor of 3339, 2961, 1491.

$$3339 = 3 * 7 * 13 * 19$$

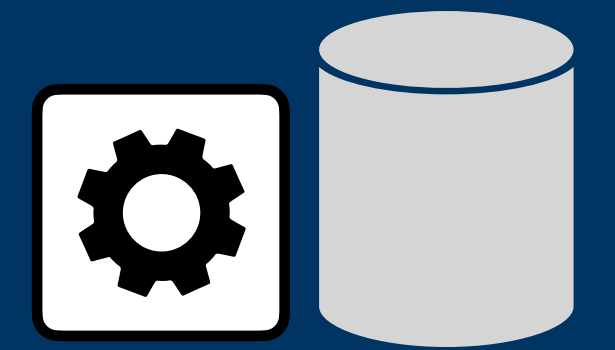
Invalid reasoning ❌

Final answer: 21.

Correct answer ✅

Informal



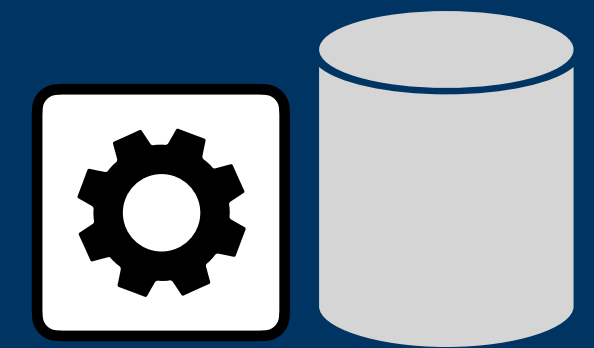


Informal

Formal

NaturalProver

Document grounding

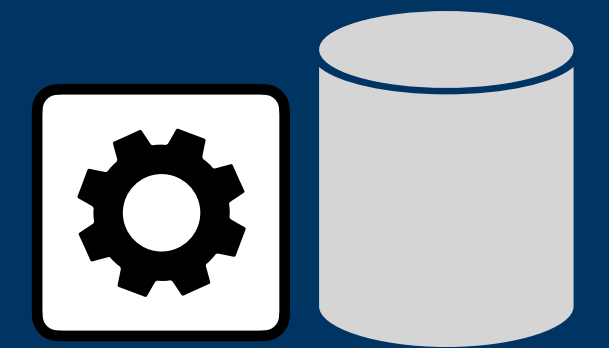


Informal

Formal

Draft, Sketch, Prove

Logical grounding



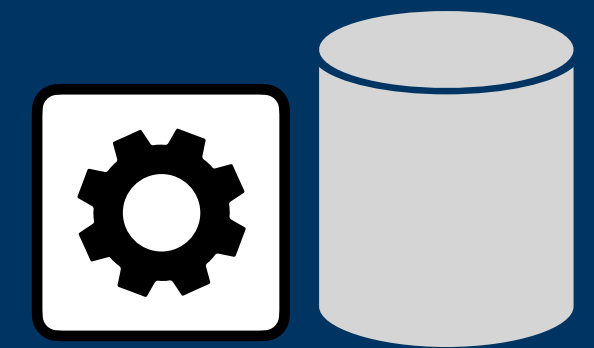
Informal

Formal

Draft, Sketch, Prove

Logical grounding

Flexibility
Data



Informal

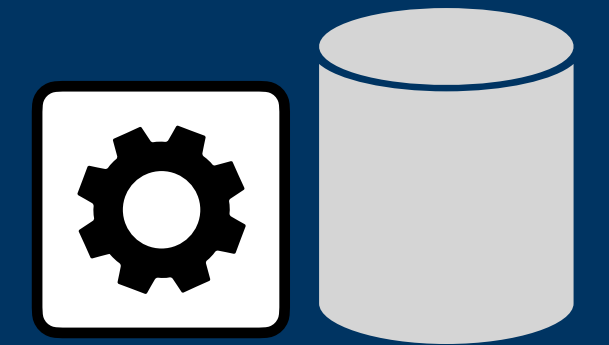
Formal

Draft, Sketch, Prove

Logical grounding

Flexibility
Data

Grounding

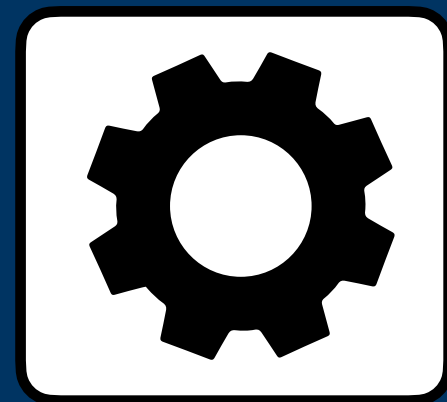


Informal

Formal

Formal proof assistants

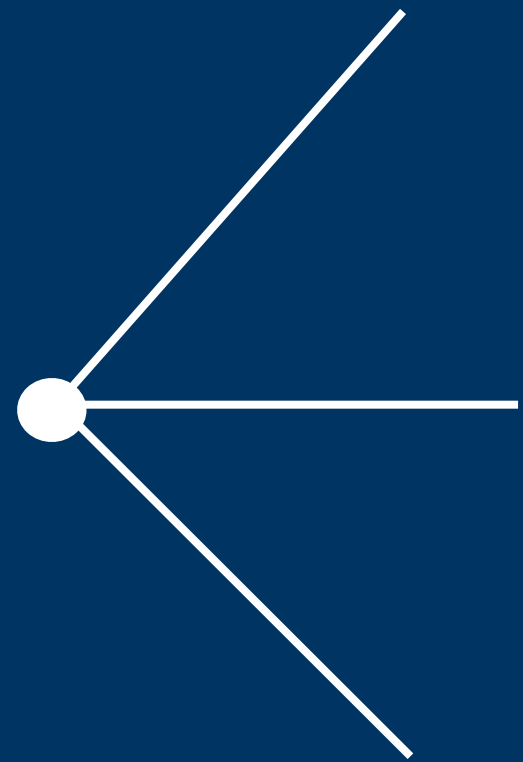
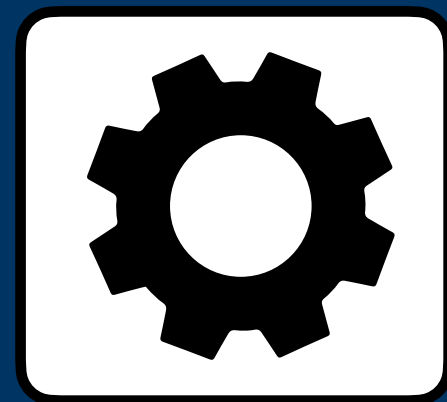
$$x+y = y+x$$



Formal

Formal proof assistants

$$x+y = y+x$$



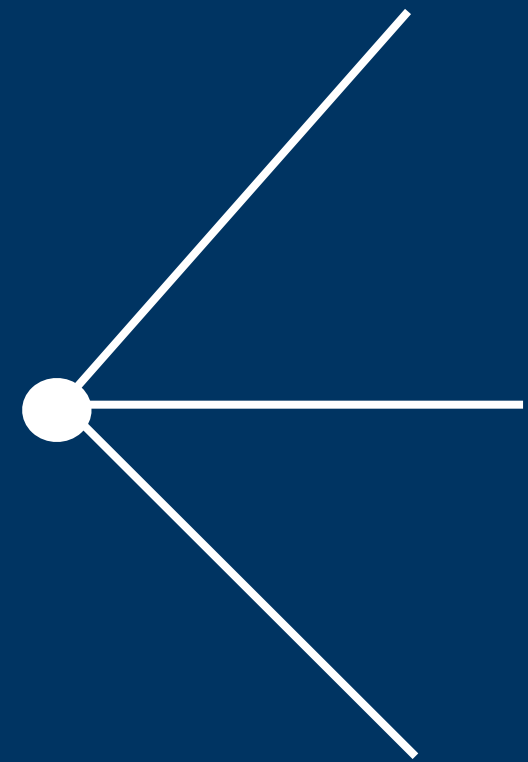
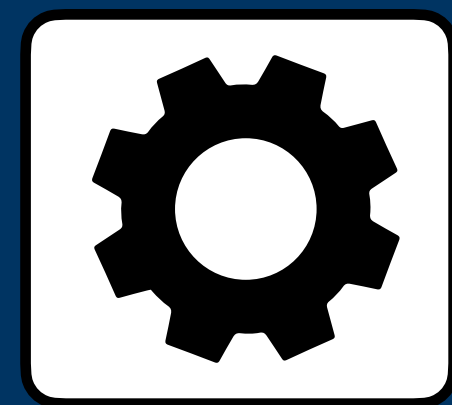
$$x+y = x+y$$



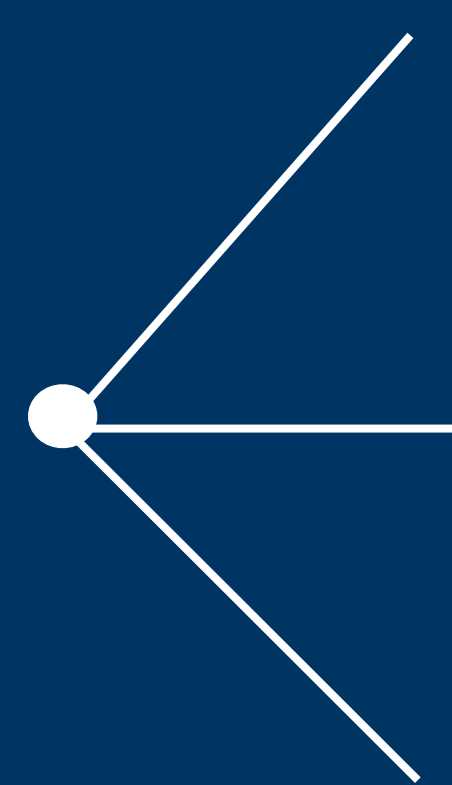
Formal

Formal proof assistants

$$x+y = y+x$$



$$x+y = x+y$$



$$0 = 0$$



Formal

Formal proof assistants

$$x+y = y+x$$



$$x+y = x+y$$



$$0 = 0$$



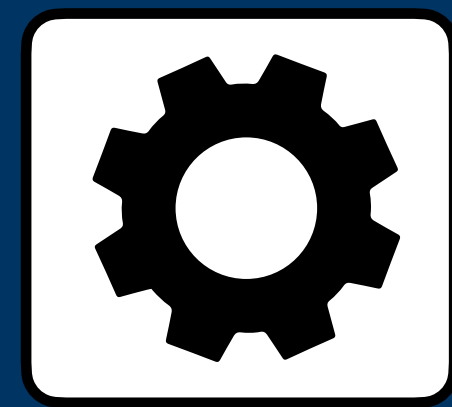
Grounding



Formal

Proof automation

$$x+y = y+x$$



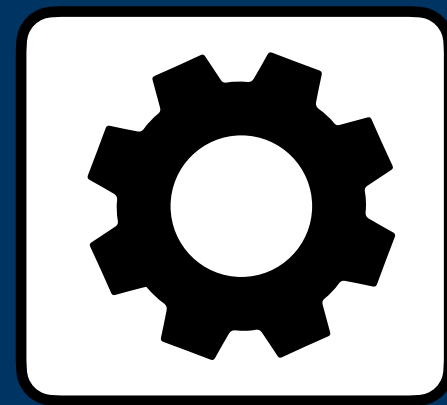
Grounding



Formal

Proof automation

$$x+y = y+x$$



Sledgehammer [Paulson 2010]

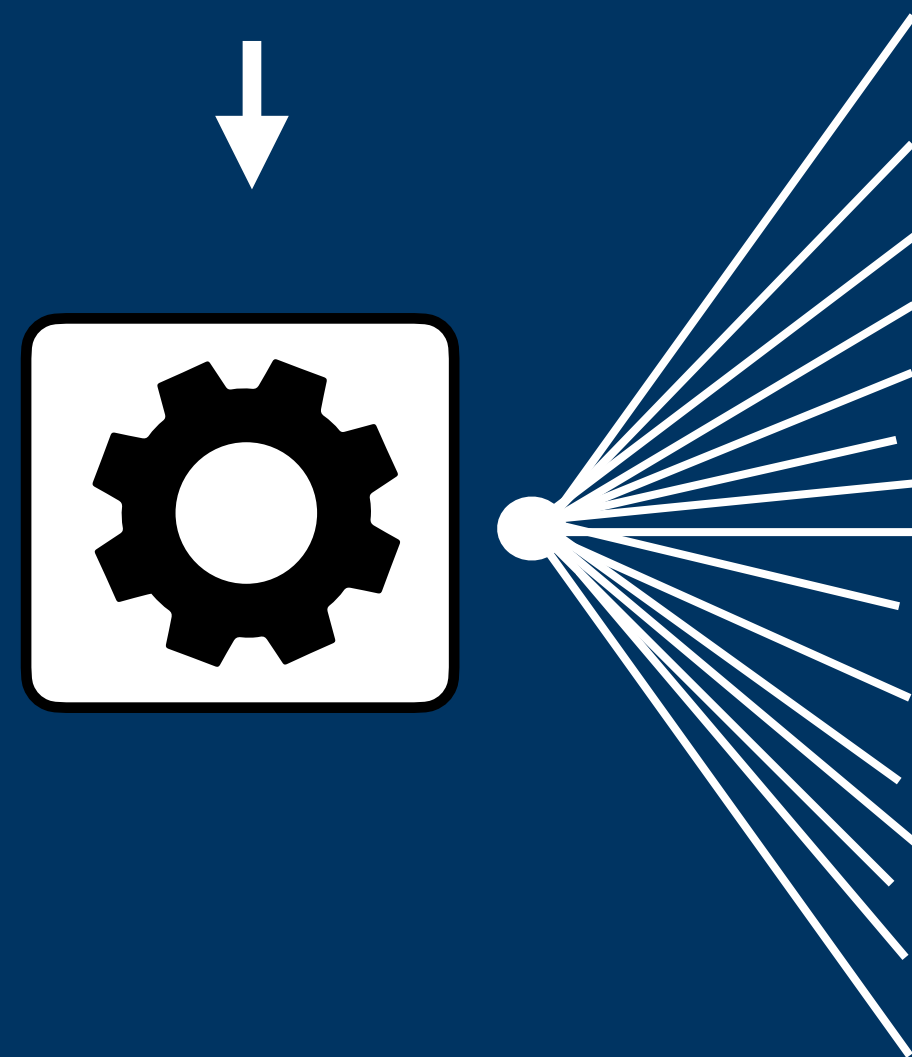
- calls out to external provers

Grounding



Formal

Proof automation

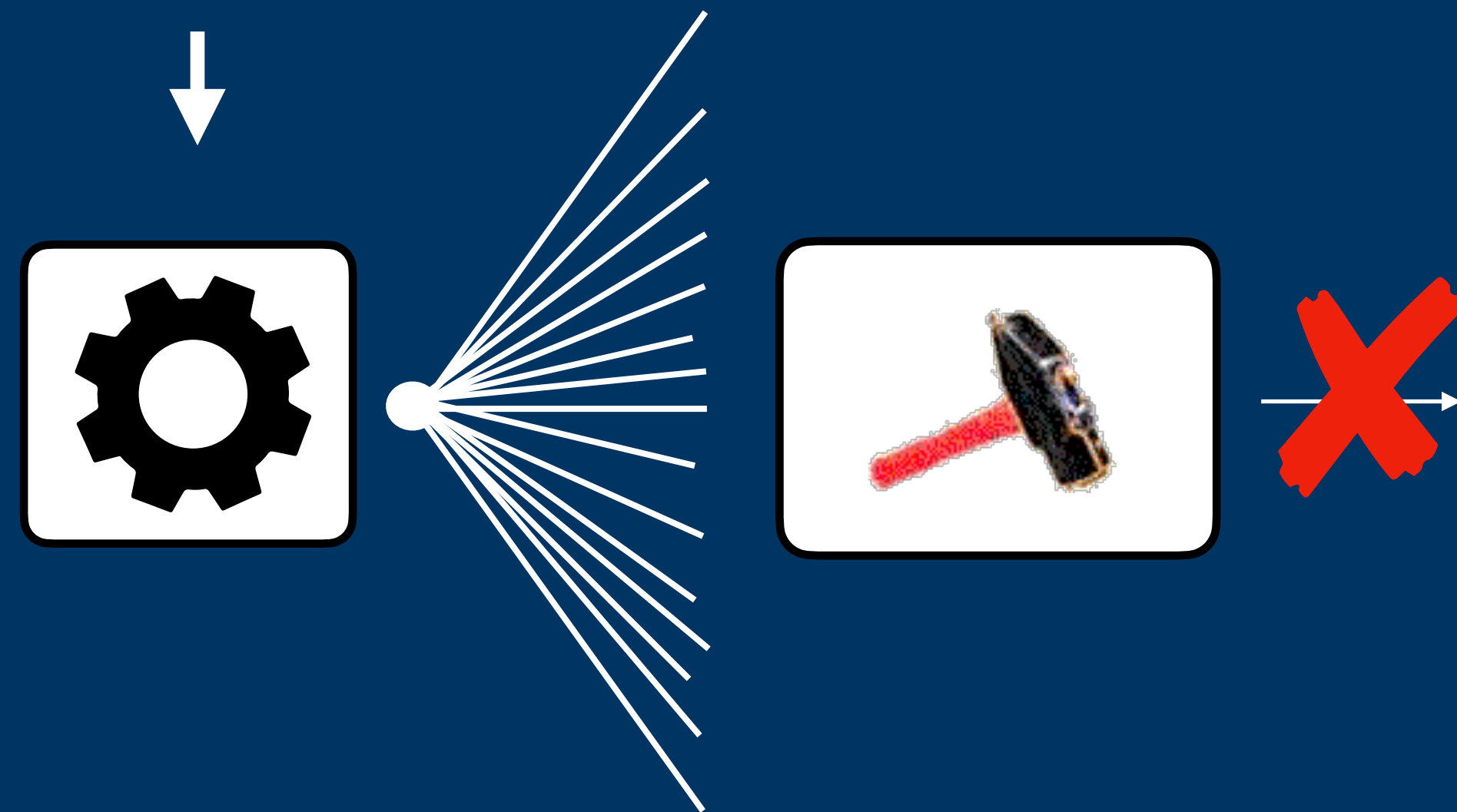


```
theorem algebra_absx1pabsxpabsxp1eqxp2_0leqxleq1:
  fixes x ::real assumes "abs (x - 1) + abs x + abs (x + 1) = x + 2"
  shows "0 <le> x <and> x <le> 1"
proof -
  have c0: "x <le> -1 <Longrightarrow> False"
  proof -
    assume c1: "x <le> -1"
    have c2: "abs(x-1) + abs x + abs(x+1) = -(x-1) - x - (x+1)" using c1
      <ATP> by auto </ATP>
    then have c3: "abs(x-1) + abs x + abs(x+1) = -3*x" <ATP> by auto </ATP>
    then have c4: "-3*x = x+2" using assms c3 <ATP> by auto </ATP>
    then have c5: "x = -1/2" <ATP> by auto </ATP>
    then show ?thesis using c1 <ATP> by auto </ATP>
  qed
  have c6: "-1 < x <Longrightarrow> x < 0 <Longrightarrow> False"
  proof -
    assume c7: "-1 < x" assume c8: "x < 0"
    have c9: "abs(x-1) + abs x + abs(x+1) = -(x-1) - x + (x+1)" using c7 c8
      <ATP> by auto </ATP>
    then have c10: "abs(x-1) + abs x + abs(x+1) = 2-x" <ATP> by auto </ATP>
    then have c11: "2-x = x+2" using assms c10 <ATP> by auto </ATP>
    then have c12: "x = 0" <ATP> by auto </ATP>
    then show ?thesis using c8 <ATP> by auto </ATP>
  qed
  have c13: "x > 1 <Longrightarrow> False"
  proof -
    assume c14: "x > 1"
    have c15: "abs(x-1) + abs x + abs(x+1) = x-1 + x + (x+1)" using c14
      <ATP> by auto </ATP>
    then have c16: "abs(x-1) + abs x + abs(x+1) = 3*x" <ATP> by auto </ATP>
    then have c17: "3*x = x+2" using assms c16 <ATP> by auto </ATP>
    then have c18: "x = 1" <ATP> by auto </ATP>
    then show ?thesis using c14 <ATP> by auto </ATP>
  qed
  then show ?thesis using c0 c6 c13 <ATP> by fastforce </ATP>
qed
```

Formal

Proof automation

Real theorem



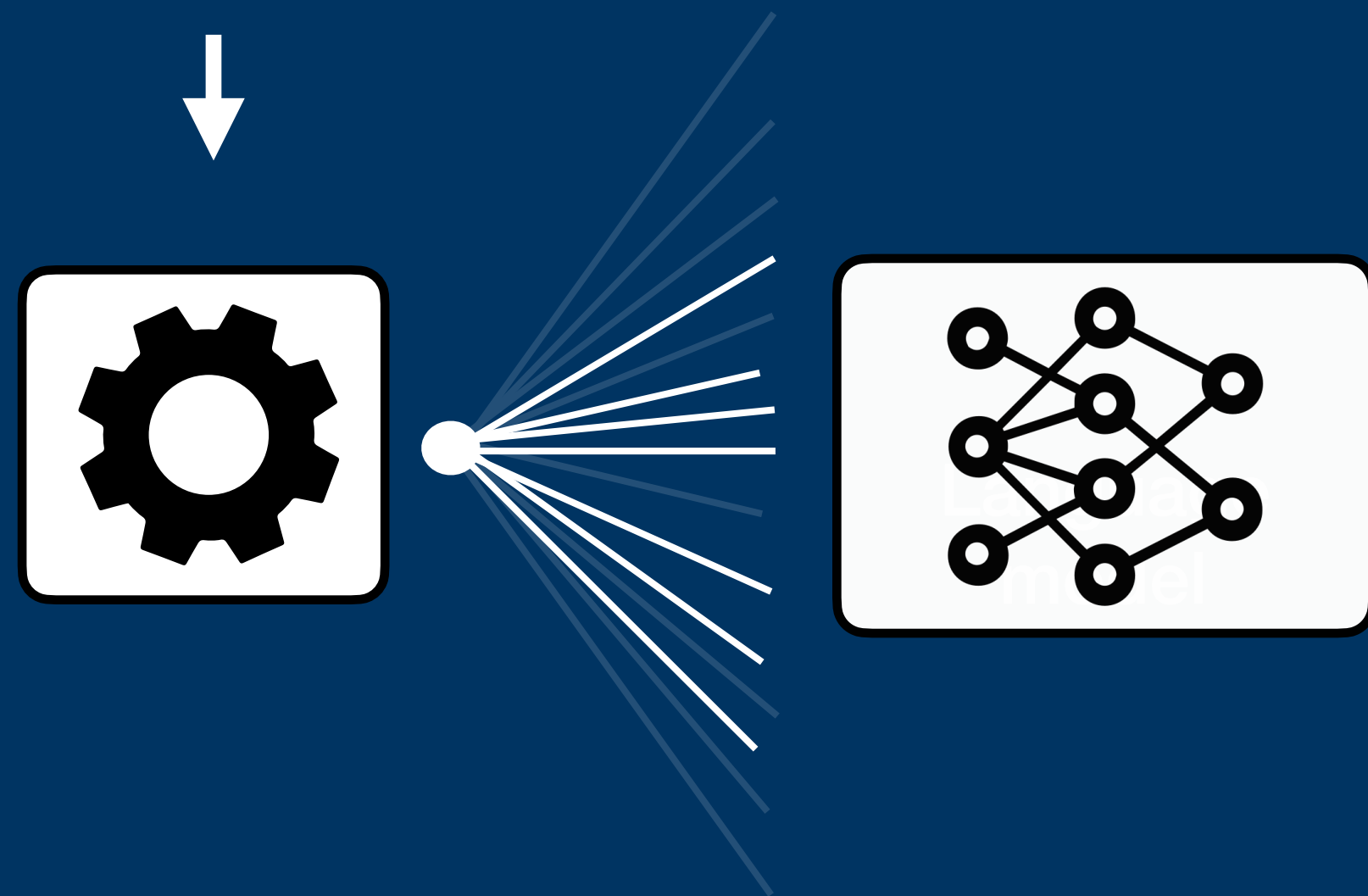
Vast search space!!

```
theorem algebra_absx1pabsxpabsxp1eqxp2_0leqxleq1:
  fixes x ::real assumes "abs (x - 1) + abs x + abs (x + 1) = x + 2"
  shows "0 <le> x <and> x <le> 1"
proof -
  have c0: "x <le> -1 <Longrightarrow> False"
  proof -
    assume c1: "x <le> -1"
    have c2: "abs(x-1) + abs x + abs(x+1) = -(x-1) - x - (x+1)" using c1
      <ATP> by auto </ATP>
    then have c3: "abs(x-1) + abs x + abs(x+1) = -3*x" <ATP> by auto </ATP>
    then have c4: "-3*x = x+2" using assms c3 <ATP> by auto </ATP>
    then have c5: "x = -1/2" <ATP> by auto </ATP>
    then show ?thesis using c1 <ATP> by auto </ATP>
  qed
  have c6: "-1 < x <Longrightarrow> x < 0 <Longrightarrow> False"
  proof -
    assume c7: "-1 < x" assume c8: "x < 0"
    have c9: "abs(x-1) + abs x + abs(x+1) = -(x-1) - x + (x+1)" using c7 c8
      <ATP> by auto </ATP>
    then have c10: "abs(x-1) + abs x + abs(x+1) = 2-x" <ATP> by auto </ATP>
    then have c11: "2-x = x+2" using assms c10 <ATP> by auto </ATP>
    then have c12: "x = 0" <ATP> by auto </ATP>
    then show ?thesis using c8 <ATP> by auto </ATP>
  qed
  have c13: "x > 1 <Longrightarrow> False"
  proof -
    assume c14: "x > 1"
    have c15: "abs(x-1) + abs x + abs(x+1) = x-1 + x + (x+1)" using c14
      <ATP> by auto </ATP>
    then have c16: "abs(x-1) + abs x + abs(x+1) = 3*x" <ATP> by auto </ATP>
    then have c17: "3*x = x+2" using assms c16 <ATP> by auto </ATP>
    then have c18: "x = 1" <ATP> by auto </ATP>
    then show ?thesis using c14 <ATP> by auto </ATP>
  qed
  then show ?thesis using c0 c6 c13 <ATP> by fastforce </ATP>
qed
```

Formal

Proof automation

Real theorem

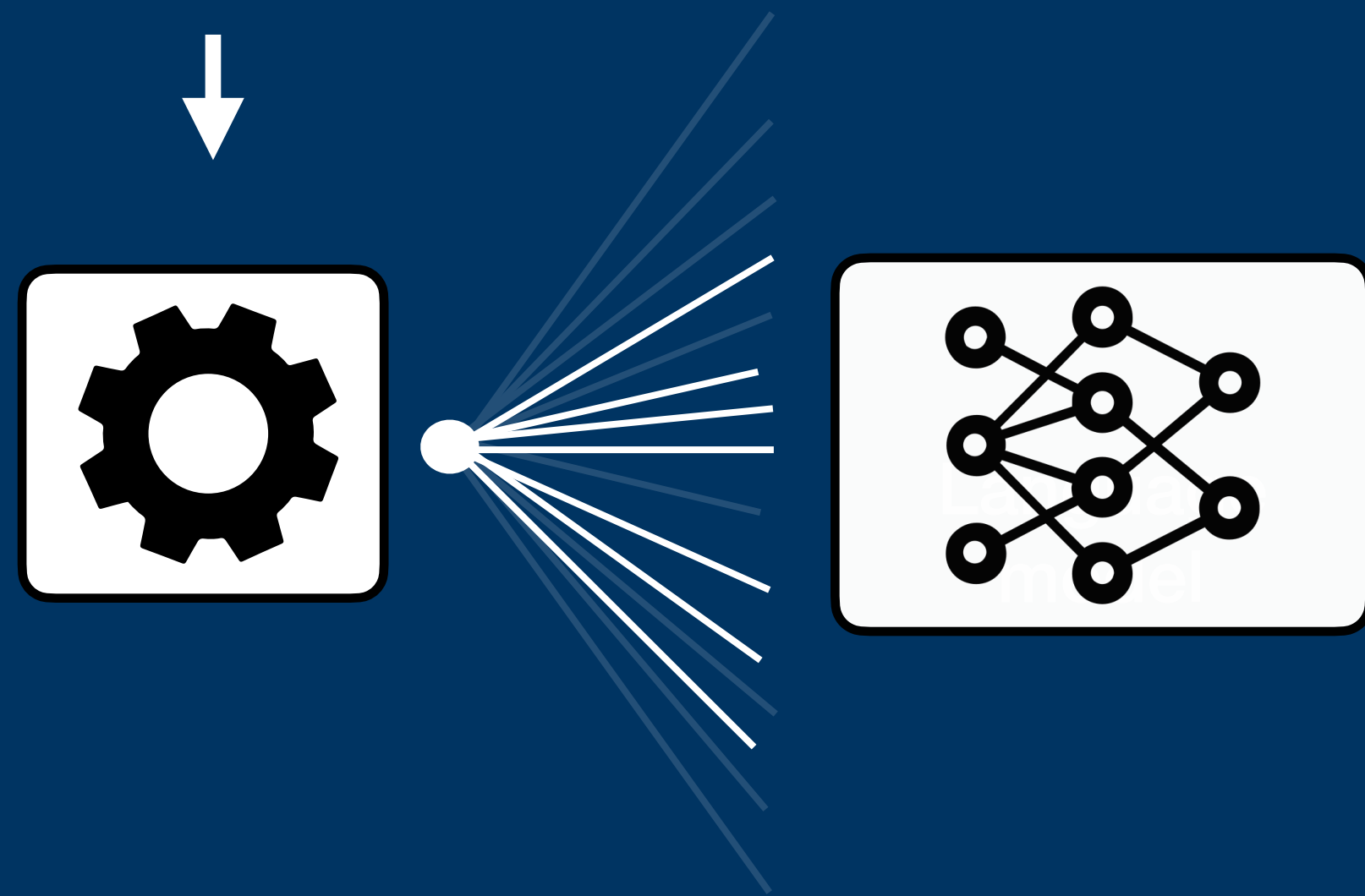


```
theorem algebra_absx1pabsxpabsxp1eqxp2_0leqxleq1:
  fixes x ::real assumes "abs (x - 1) + abs x + abs (x + 1) = x + 2"
  shows "0 <le> x <and> x <le> 1"
proof -
  have c0: "x <le> -1 <Longrightarrow> False"
  proof -
    assume c1: "x <le> -1"
    have c2: "abs(x-1) + abs x + abs(x+1) = -(x-1) - x - (x+1)" using c1
      <ATP> by auto </ATP>
    then have c3: "abs(x-1) + abs x + abs(x+1) = -3*x" <ATP> by auto </ATP>
    then have c4: "-3*x = x+2" using assms c3 <ATP> by auto </ATP>
    then have c5: "x = -1/2" <ATP> by auto </ATP>
    then show ?thesis using c1 <ATP> by auto </ATP>
  qed
  have c6: "-1 < x <Longrightarrow> x < 0 <Longrightarrow> False"
  proof -
    assume c7: "-1 < x" assume c8: "x < 0"
    have c9: "abs(x-1) + abs x + abs(x+1) = -(x-1) - x + (x+1)" using c7 c8
      <ATP> by auto </ATP>
    then have c10: "abs(x-1) + abs x + abs(x+1) = 2-x" <ATP> by auto </ATP>
    then have c11: "2-x = x+2" using assms c10 <ATP> by auto </ATP>
    then have c12: "x = 0" <ATP> by auto </ATP>
    then show ?thesis using c8 <ATP> by auto </ATP>
  qed
  have c13: "x > 1 <Longrightarrow> False"
  proof -
    assume c14: "x > 1"
    have c15: "abs(x-1) + abs x + abs(x+1) = x-1 + x + (x+1)" using c14
      <ATP> by auto </ATP>
    then have c16: "abs(x-1) + abs x + abs(x+1) = 3*x" <ATP> by auto </ATP>
    then have c17: "3*x = x+2" using assms c16 <ATP> by auto </ATP>
    then have c18: "x = 1" <ATP> by auto </ATP>
    then show ?thesis using c14 <ATP> by auto </ATP>
  qed
  then show ?thesis using c0 c6 c13 <ATP> by fastforce </ATP>
qed
```

Formal

Proof automation

Real theorem



```
theorem algebra_absx1pabsxpabsxp1eqxp2_0leqxleq1:
  fixes x ::real assumes "abs (x - 1) + abs x < 2"
  shows "0 <le> x <and> x <le> 1"
proof -
  have c0: "x <le> -1 <Longrightarrow"
  proof -
    assume c1: "x <le> -1"
    have c2: "abs(x-1) + abs x < 2"
      <ATP> by auto </ATP>
    then have c3: "abs(x-1) + abs x < 2"
    then have c4: "-3*x = x+2"
    then have c5: "x = -1/2" <ATP> by auto </ATP>
    then show ?thesis using c1
  qed
  have c6: "-1 < x <Longrightarrow"
  proof -
    assume c7: "-1 < x"
    assume c8: "abs(x-1) + abs x < 2"
    <ATP> by auto </ATP>
    then have c9: "abs(x-1) + abs x < 2"
    then have c10: "abs(x-1) + abs x < 2"
    then have c11: "2-x = x+2"
    then have c12: "x = 0" <ATP> by auto </ATP>
    then show ?thesis using c8
  qed
  have c13: "x < 1 <Longrightarrow"
  proof -
    assume c14: "x > 1"
    assume c15: "abs(x-1) + abs x < 2"
    <ATP> by auto </ATP>
    then have c16: "abs(x-1) + abs x < 2"
    then have c17: "3*x = x+2" using c15
    then have c18: "x = 1" <ATP> by auto </ATP>
    then show ?thesis using c14 <ATP> by auto </ATP>
  qed
  then show ?thesis using c0 c6 c13 <ATP> by fastforce </ATP>
qed
```

Informal Data

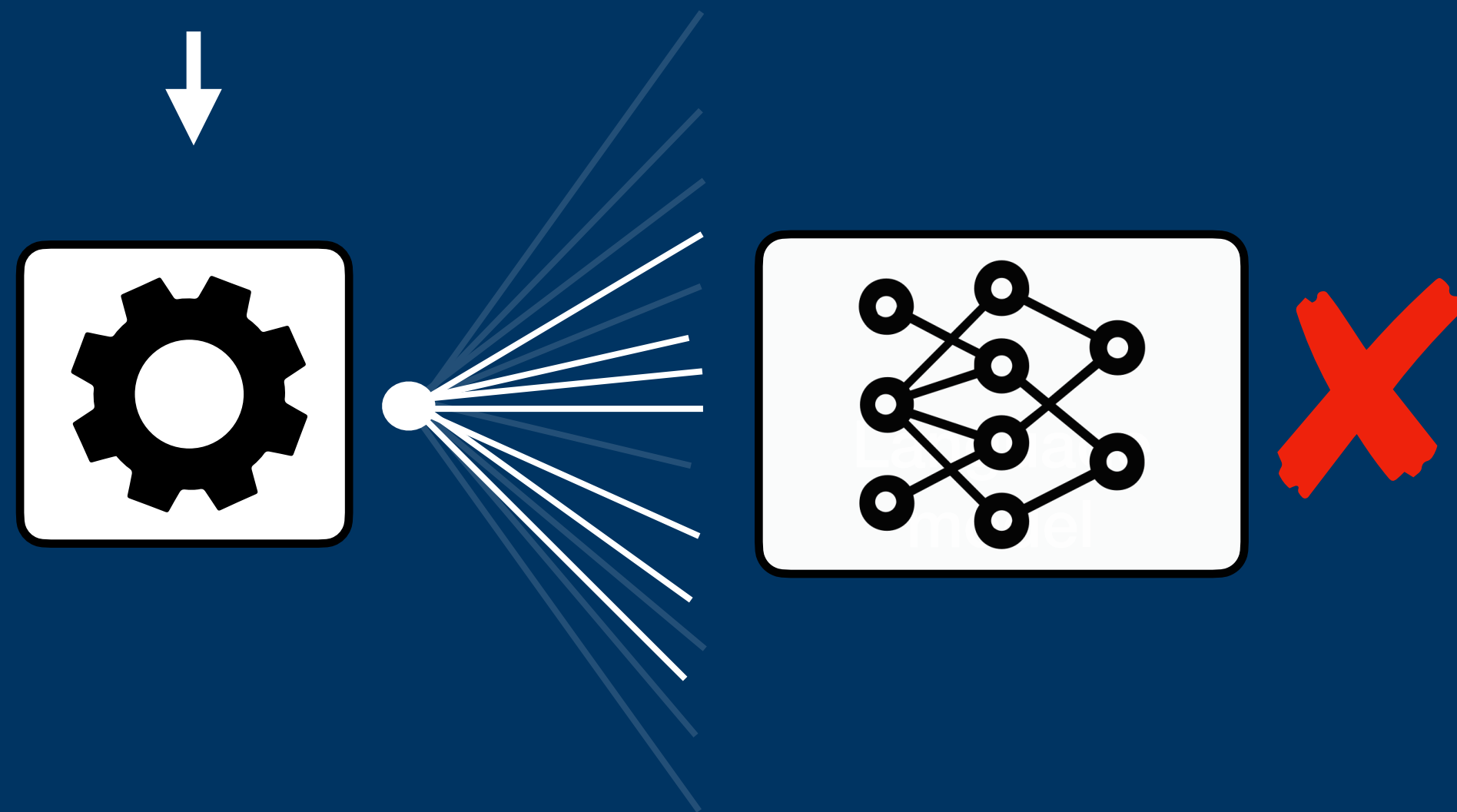
Formal Data

Data scarce



Proof automation

Real theorem



```
theorem algebra_absx1pabsxpabsxp1eqxp2_0leqxleq1:
  fixes x ::real assumes "abs (x - 1) + abs (x + 1) < 2"
  shows "0 <= x <= 1"
proof -
  have c0: "x <= -1 <= Long"
  proof -
    assume c1: "x <= -1"
    have c2: "abs(x-1) + abs x < 2"
      <ATP> by auto </ATP>
    then have c3: "abs(x-1) + abs x < 2"
    then have c4: "-3*x = x+2"
    then have c5: "x = -1/2"
    then show ?thesis using c1 c5
  qed
  have c6: "-1 < x <= Long"
  proof -
    assume c7: "-1 < x"
    have c8: "abs(x-1) + abs x < 2"
      <ATP> by auto </ATP>
    then have c9: "abs(x-1) + abs x < 2"
    then have c10: "abs(x-1) + abs x < 2"
    then have c11: "2-x < 2"
    then have c12: "x < 0"
    then show ?thesis using c7 c12
  qed
  have c13: "x <= 1 <= Long"
  proof -
    assume c14: "x > 1"
    have c15: "abs(x-1) + abs x < 2"
      <ATP> by auto </ATP>
    then have c16: "abs(x-1) + abs x < 2"
    then have c17: "3*x = x+2" using c14
    then have c18: "x = 1" <ATP> by auto
    then show ?thesis using c14 <ATP> by auto </ATP>
  qed
  then show ?thesis using c0 c6 c13 <ATP> by fastforce </ATP>
qed
```

Formal Data

Data scarce



Formal

Informal



Formal

If $\gcd(n, 40) = 10$ and
 $\text{lcm}(n, 4) = 280$,
prove that n is 70.

Informal

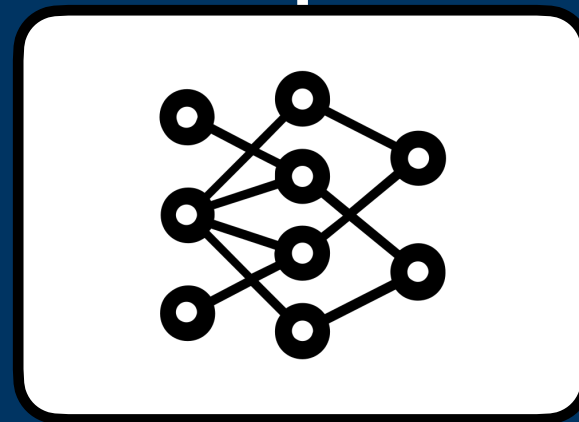
Formal

Informal Proof *Draft*

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

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

completing the proof. ■



If $\gcd(n, 40) = 10$ and
 $\text{lcm}(n, 4) = 280$,
prove that n is 70.

Informal

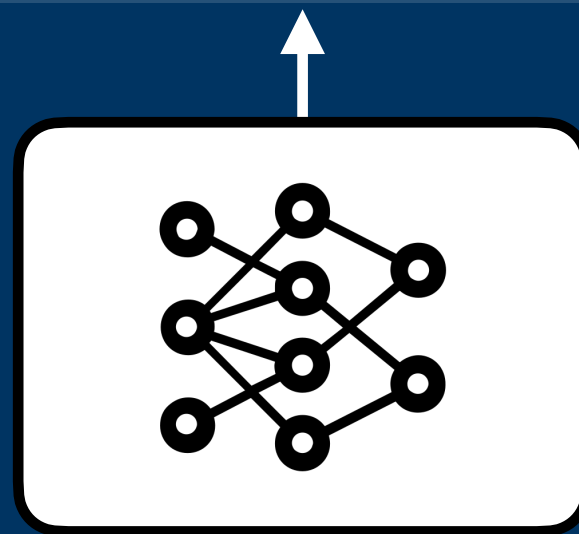
Formal

Informal Proof *Draft*

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

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

completing the proof. ■



If $\gcd(n, 40) = 10$ and
 $\text{lcm}(n, 4) = 280$,
prove that n is 70.

Informal

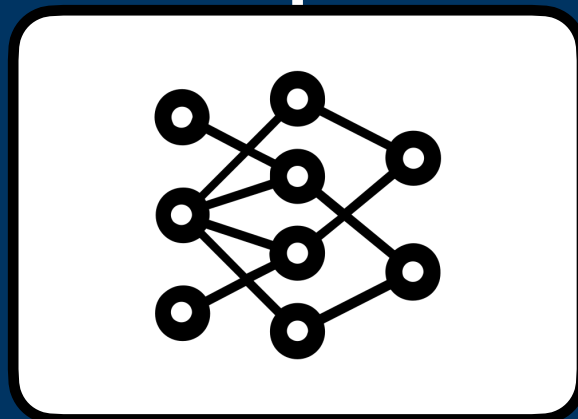
Formal

Informal Proof *Draft*

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

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

completing the proof. ■



If $\gcd(n, 40) = 10$ and
 $\text{lcm}(n, 4) = 280$,
prove that n is 70.

Formal Proof *Sketch*

have c1: $"10 * 280 = n * 40"$
using assms

<proof>

Informal

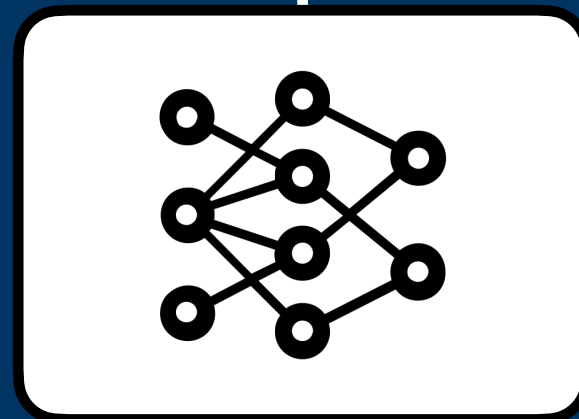
Formal

Informal Proof *Draft*

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

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

completing the proof. ■



If $\gcd(n, 40) = 10$ and
 $\text{lcm}(n, 4) = 280$,
prove that n is 70.

Formal Proof *Sketch*

have c1: $"10 * 280 = n * 40"$
using assms

<proof>

then have c2: $"n = 10 * 280 / 40"$

<proof>

Informal

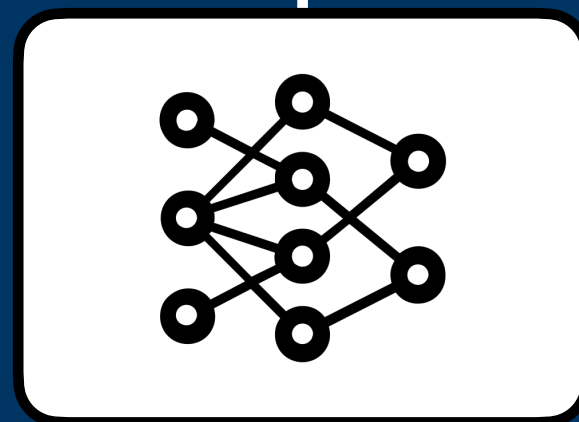
Formal

Informal Proof *Draft*

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

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

completing the proof. ■



If $\gcd(n, 40) = 10$ and
 $\text{lcm}(n, 4) = 280$,
prove that n is 70.

Formal Proof *Sketch*

have c1: $"10 * 280 = n * 40"$
using assms

<proof>

then have c2: $"n = 10 * 280 / 40"$

<proof>

then show ?thesis

<proof>

High-level reasoning structure

Informal

Formal

Natural Proof *Draft*

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

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

completing the proof. ■

Formal Proof *Sketch*

```
have c1: "10*280 = n*40"  
using assms  
  <proof>
```

```
then have c2: "n = 10*280/40"  
  <proof>
```

```
then show ?thesis  
  <proof>
```

Informal

Formal

Natural Proof *Draft*

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

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

completing the proof. ■

Formal Proof *Sketch*

```
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
```



Decomposes search

Informal

Formal

Natural Proof *Draft*

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

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

completing the proof. ■

Formal Proof Sketch

```
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
```



✓ Verified Formal Proof

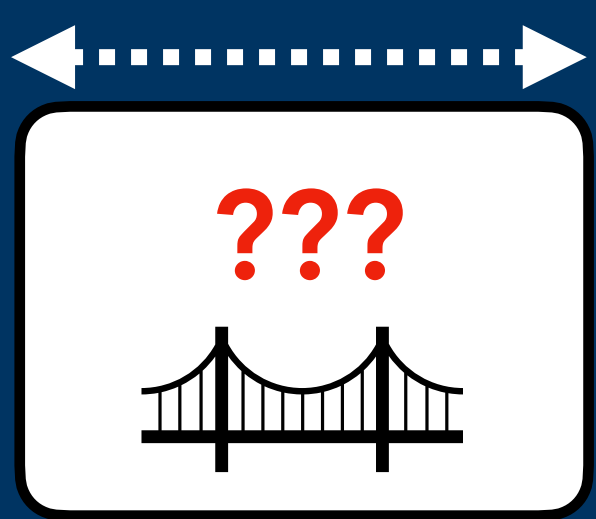
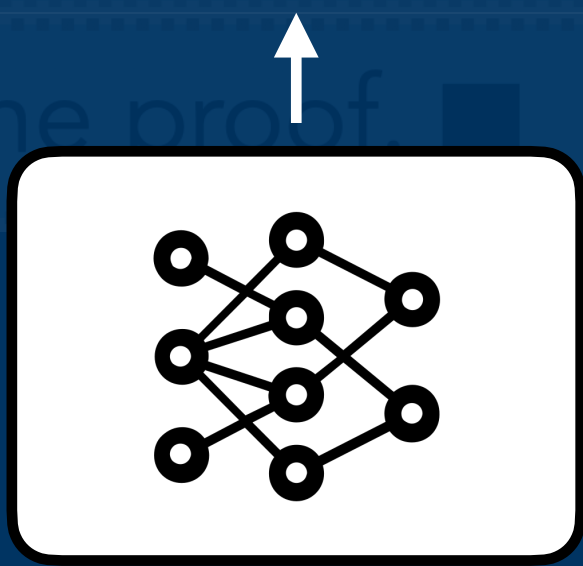
Decomposes search

Informal

Formal

We know that $\text{gcd}(a, b) \cdot \text{lcm}(a, b) = ab$,
hence $10 \cdot 280 = n \cdot 40$.
Then $n = \frac{10 \cdot 280}{40} = 70$,
completing the proof. \square

Natural Proof *Draft*



```
have c1: "10*280 = n*40"  
using assms  
<proof>  
then show c2: "10*280/40 = n" <proof>  
then show c3: "n = 70" <proof>
```

Formal Proof *Sketch*



Verified Formal *Proof*

Informal

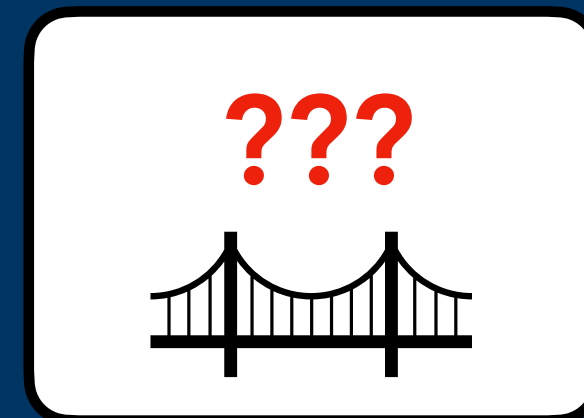
Formal



Paired data doesn't exist!

We know that $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$,
hence $10 \cdot 280 = n \cdot 40$.
Then $n = \frac{10 \cdot 280}{40} = 70$,
completing the proof.

Natural Proof *Draft*



" $10 \cdot 280 = n \cdot 40$ "
using assms
<proof>
then show ?thesis
<proof>

Formal Proof *Sketch*



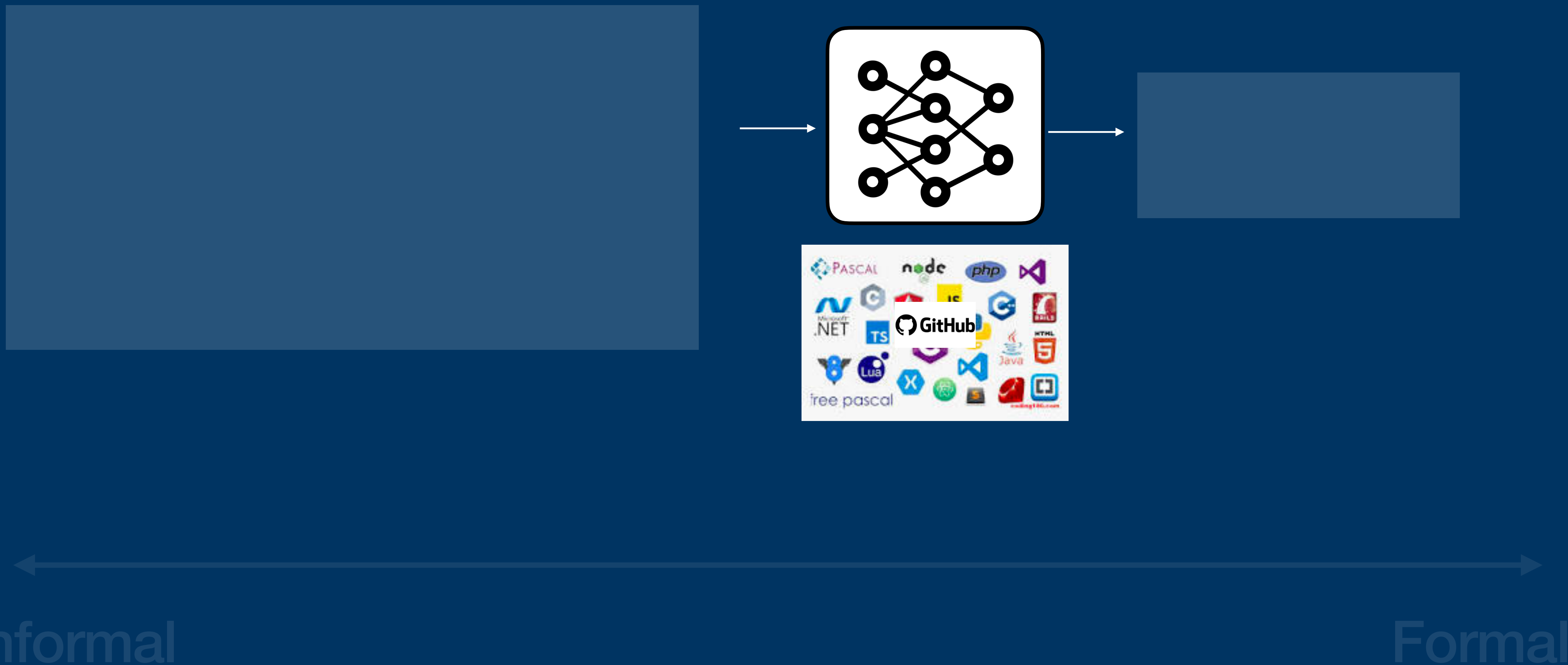
Verified Formal Proof

Informal

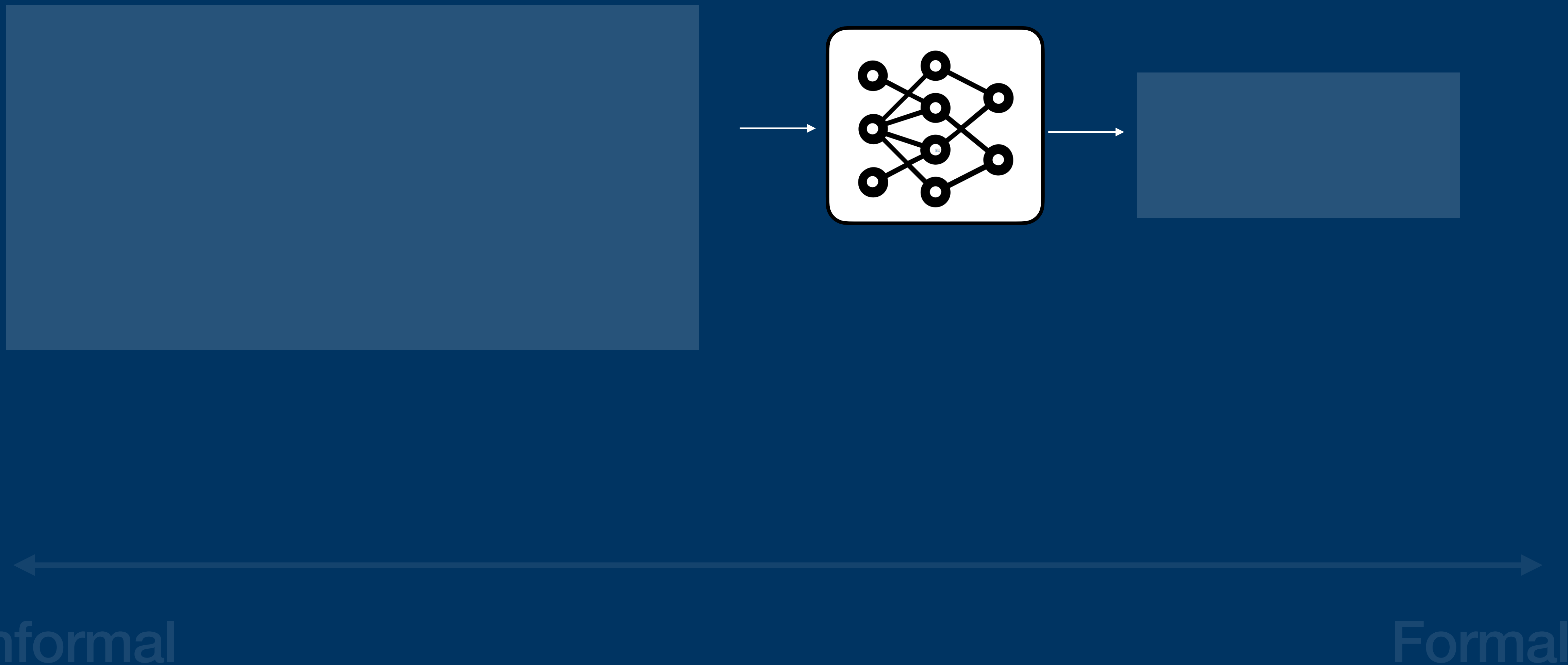
Formal



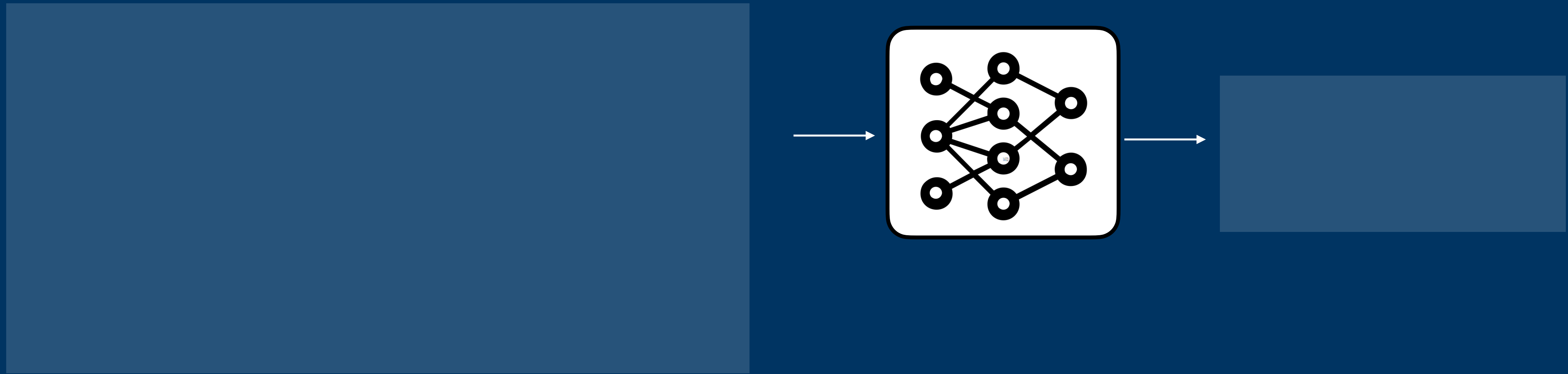
Codex: language model trained on vast amount of code



Codex: language model trained on vast amount of code



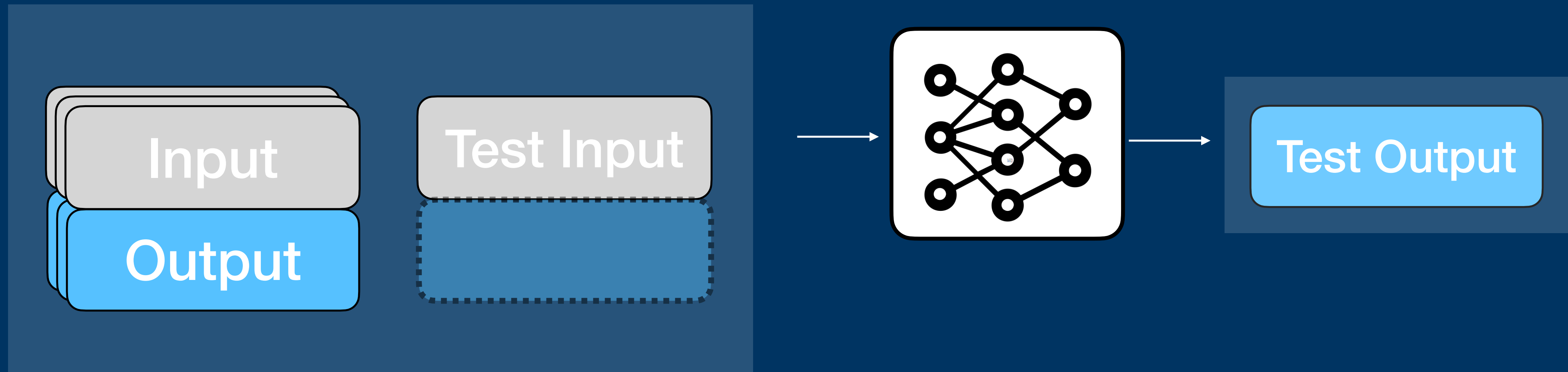
Codex: language model trained on vast amount of code



Can learn tasks *in-context*



Codex: language model trained on vast amount of code



Can learn tasks *in-context*



Theorem $10a \leq 28a^2 + 1$.

Informal Proof:

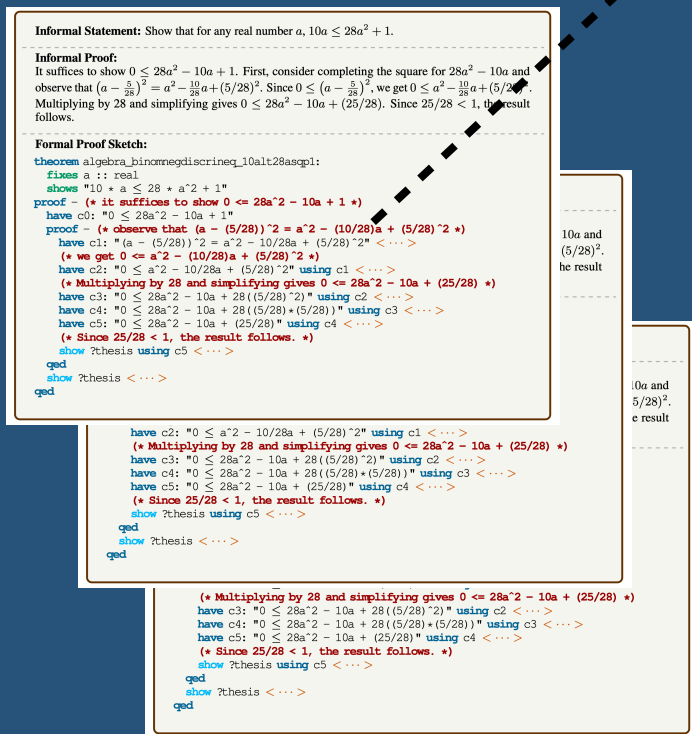
It suffices to show $0 \leq 28a^2 - 10a + 1$. First, consider completing the square for $28a^2 - 10a$ and observe that $(a - \frac{5}{28})^2 = a^2 - \frac{10}{28}a + (5/28)^2$. Multiplying by 28 and simplifying gives $0 \leq 28a^2 - 10a + (25/28)$. Since $25/28 < 1$, the result follows.

Draft

Formal Proof Sketch:

```
theorem algebra_binomnegdiscrineq_10alt28asqpl:
  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)^2" using c3 <...>
  (* Multiplying by 28 and simplifying gives 0 ≤ 28a^2 - 10a + (25/28) *)
  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
```

Sketch



Can learn to sketch *in-context!!!*

Informal

Formal

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 $(a - \frac{5}{28})^2 = a^2 - \frac{10}{28}a + (5/28)^2$. Multiplying by 28 and simplifying gives $0 \leq 28a^2 - 10a + (25/28)$. Since $25/28 < 1$, the result follows.

Draft

Formal Proof Sketch:

```
theorem algebra_binomnegdiscrineq_10alt28asqpl:
  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)^2" using c3 <...>
  (* Multiplying by 28 and simplifying gives 0 ≤ 28a^2 - 10a + (25/28) *)
  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
```

Sketch

Can learn to sketch *in-context!!!*

Informal

Formal

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 $(a - \frac{5}{28})^2 = a^2 - \frac{10}{28}a + (5/28)^2$. Since $0 \leq (a - \frac{5}{28})^2$, 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_10alt28asqpl:
  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 <...>
  (* Since 25/28 < 1, the result follows. *)
  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
```

Sketch

Can learn to sketch *in-context!!!*

Informal

Formal

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

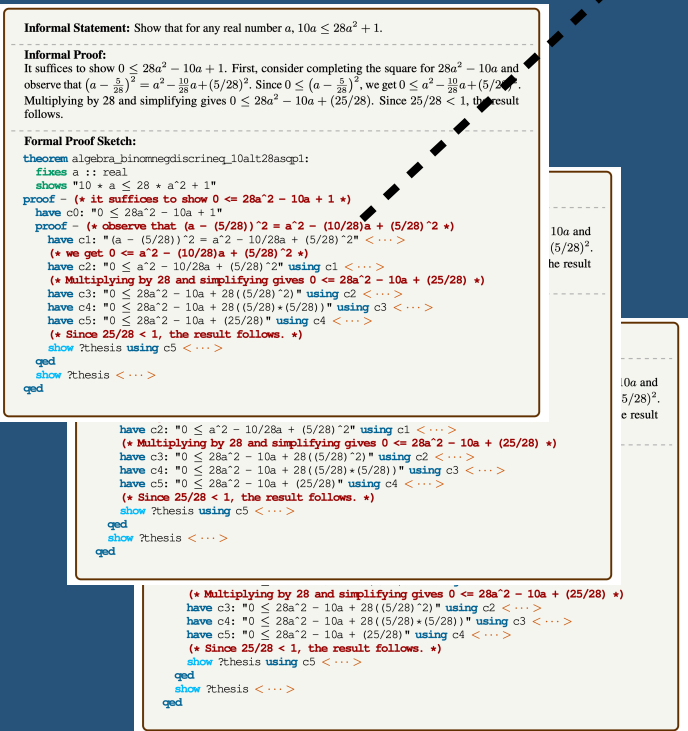
Informal Proof:

It suffices to show $0 \leq 28a^2 - 10a + 1$. First, consider completing the square for $28a^2 - 10a$ and observe that $(a - \frac{5}{28})^2 = a^2 - \frac{10}{28}a + (5/28)^2$. Since $0 \leq (a - \frac{5}{28})^2$, 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_binomnegdiscrined_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 <...>
  (* Multiplying by 28 and simplifying gives 0 ≤ 28a^2 - 10a + (25/28) *)
  have c4: "0 ≤ 28a^2 - 10a + 28((5/28)*(5/28))" using c3 <...>
  (* Since 25/28 < 1, the result follows. *)
  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
  
```



Can learn to sketch *in-context!!!*

Informal

Formal


```
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  $(a - \frac{5}{28})^2 = a^2 - \frac{5}{14}a + (5/28)^2$ . Since  $0 \leq (a - \frac{5}{28})^2$ , we get  $0 \leq a^2 - \frac{5}{14}a + (5/28)^2$ . Multiplying by 28 and simplifying gives  $0 \leq 28a^2 - 10a + (25/28)$ . Since  $25/28 < 1$ , the result follows.

Formal Proof Sketch:
theorem alpha_positivity (a:real) : 10a <= 28a^2 + 1 :=
  show "0 <= 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 - 5/14a + (5/28)^2 *)
      have c1 : 0 <= (5/28)^2 := by norm_num
      have c2 : 0 <= a^2 - 5/14a + (5/28)^2 := by
        (* we get 0 <= a^2 - 5/14a + (5/28)^2 *)
        have c3 : 0 <= a^2 - 5/14a + (5/28)^2 := by
          (* Multiplying by 28 and simplifying gives 0 <= 28a^2 - 10a + (25/28) *)
          have c4 : 0 <= 28a^2 - 10a + 28 * (5/28)^2 := by
            (* Multiplying by 28 and simplifying gives 0 <= 28a^2 - 10a + (25/28) *)
            have c5 : 0 <= 28a^2 - 10a + (25/28) := by
              (* Since 25/28 < 1, the result follows. *)
              show "thesis" using c4 <|>
            qed
          show "thesis" using c5 <|>
        qed
      have c6 : 0 <= a^2 - 5/14a + (5/28)^2 := by
        (* Multiplying by 28 and simplifying gives 0 <= 28a^2 - 10a + (25/28) *)
        have c7 : 0 <= 28a^2 - 10a + 28 * (5/28)^2 := by
          (* Multiplying by 28 and simplifying gives 0 <= 28a^2 - 10a + (25/28) *)
          have c8 : 0 <= 28a^2 - 10a + 28 * (5/28)^2 := by
            (* Since 25/28 < 1, the result follows. *)
            show "thesis" using c7 <|>
          qed
          show "thesis" using c8 <|>
        qed
      have c9 : 0 <= 28a^2 - 10a + (25/28) := by
        (* Since 25/28 < 1, the result follows. *)
        show "thesis" using c9 <|>
      qed
    }
  }
  show "thesis" using c0 <|>
  qed
```

Can learn to sketch *in-context!!!*



Informal

Formal

```

Informal Proof:
It suffices to show  $0 < 28a^2 - 10a + 1$ . First, consider completing the square for  $28a^2 - 10a$  and observe that  $(a - \frac{5}{28})^2 = a^2 - \frac{5}{14}a + (5/28)^2$ . Since  $0 \leq (a - \frac{5}{28})^2$ , we get  $0 \leq a^2 - \frac{5}{14}a + (5/28)^2$ . Multiplying by 28 and simplifying gives  $0 \leq 28a^2 - 10a + (25/28)$ . Since  $25/28 < 1$ , the result follows.

Formal Proof Sketch:
theorem alpha_positivity [alpha_positivity] :
  show "0 < 28 * a^2 - 10 * a + 1"
  proof -
    (* It suffices to show  $0 < 28a^2 - 10a + 1$  *)
    have c0 : "0 < 28 * a^2 - 10 * a + 1"
    proof -
      (* observe that  $(a - \frac{5}{28})^2 = a^2 - \frac{5}{14}a + (5/28)^2$  *)
      have c1 : "0 < (5/28)^2 - \frac{5}{14}a + a^2"
      proof -
        (* we get  $0 \leq a^2 - \frac{5}{14}a + (5/28)^2$  *)
        have c2 : "0 < a^2 - \frac{5}{14}a + (5/28)^2" using c0
        proof -
          (* Multiplying by 28 and simplifying gives  $0 \leq 28a^2 - 10a + (25/28)$  *)
          have c3 : "0 < 28 * a^2 - 10 * a + 28 * (5/28)^2" using c0
          have c4 : "0 < 28 * a^2 - 10 * a + 28 * (5/28)^2" using c3
          (* Since  $25/28 < 1$ , the result follows. *)
          show "thesis" using c4
        qed
      qed
    qed
  qed

```

If $\gcd(n, 40) = 10$
 and $\text{lcm}(n, 40) = 280$
Theorem
 prove that n is 70.

We know that $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$,
 hence $10 \cdot 280 = n \cdot 40$.
Draft
 Then $n = 10 \cdot 280 / 40 = 70$
 completing the proof. ■

Can learn to sketch *in-context!!!*

Informal

Formal

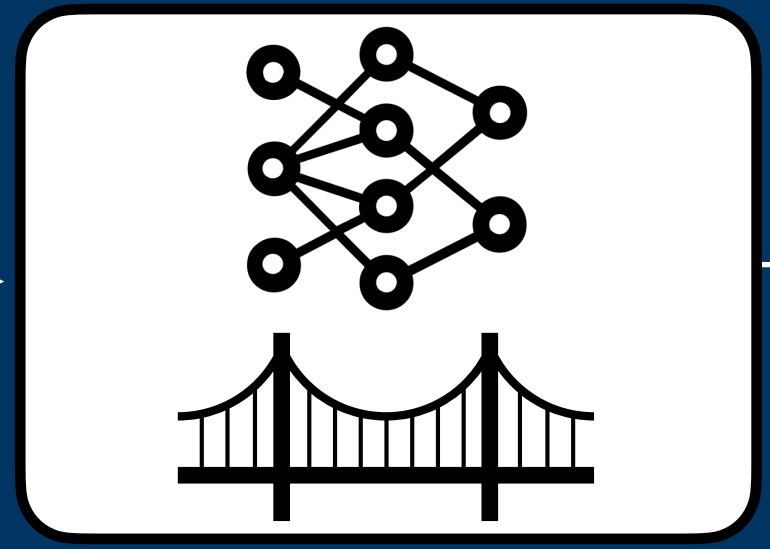
Informal Statement: Show that for any real number a , $10a \leq 28a^2 + 1$.

Informal Proof:
 It suffices to show $0 \leq 28a^2 - 10a + 1$. First, consider completing the square for $28a^2 - 10a$ and observe that $(a - \frac{5}{28})^2 = a^2 - \frac{5}{14}a + (\frac{5}{28})^2$. Since $0 \leq (a - \frac{5}{28})^2$, we get $0 \leq a^2 - \frac{5}{14}a + (\frac{5}{28})^2$. Multiplying by 28 and simplifying gives $0 \leq 28a^2 - 10a + (\frac{25}{28})$. Since $\frac{25}{28} < 1$, the result follows.

Formal Proof Sketch:
 theorem alpha_positivity_lemma_10a128a21
 class a: ℝ
 show 10 * a ≤ 28 * a^2 + 1
 proof - (x) it suffices to show 0 ≤ 28 * a^2 - 10 * a + 1 *
 have c1: 0 ≤ 28 * a^2 - 10 * a + 1
 proof - (x) observe that: 0 = (5/28)^2 * a^2 + a^2 - (10/28) * a + (5/28)^2 * a
 have c1: 0 = (5/28)^2 * a^2 + a^2 - (10/28) * a + (5/28)^2 * a <->
 (x) we get 0 = a^2 - (10/28) * a + (5/28)^2 * a
 have c1: 0 ≤ a^2 - (10/28) * a + (5/28)^2 * a using c1 <->
 (x) Multiplying by 28 and simplifying gives 0 ≤ 28 * a^2 - 10 * a + (25/28) * a
 have c1: 0 ≤ 28 * a^2 - 10 * a + 28 * (5/28)^2 * a using c1 <->
 have c1: 0 ≤ 28 * a^2 - 10 * a + 28 * (5/28) * (5/28) * a using c1 <->
 have c1: 0 ≤ 28 * a^2 - 10 * a + (25/28) * a using c1 <->
 (x) Since 25/28 < 1, the result follows. *)
 show thesis using c1 <->
 qed
 show thesis <->
 qed

If $\gcd(n, 40) = 10$
 and $\text{lcm}(n, 40) = 280$
Theorem
 prove that n is 70.

We know that $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$,
 hence $10 \cdot 280 = n \cdot 40$.
Draft
 Then $n = 10 \cdot 280 / 40 = 70$
 completing the proof. ■



```

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

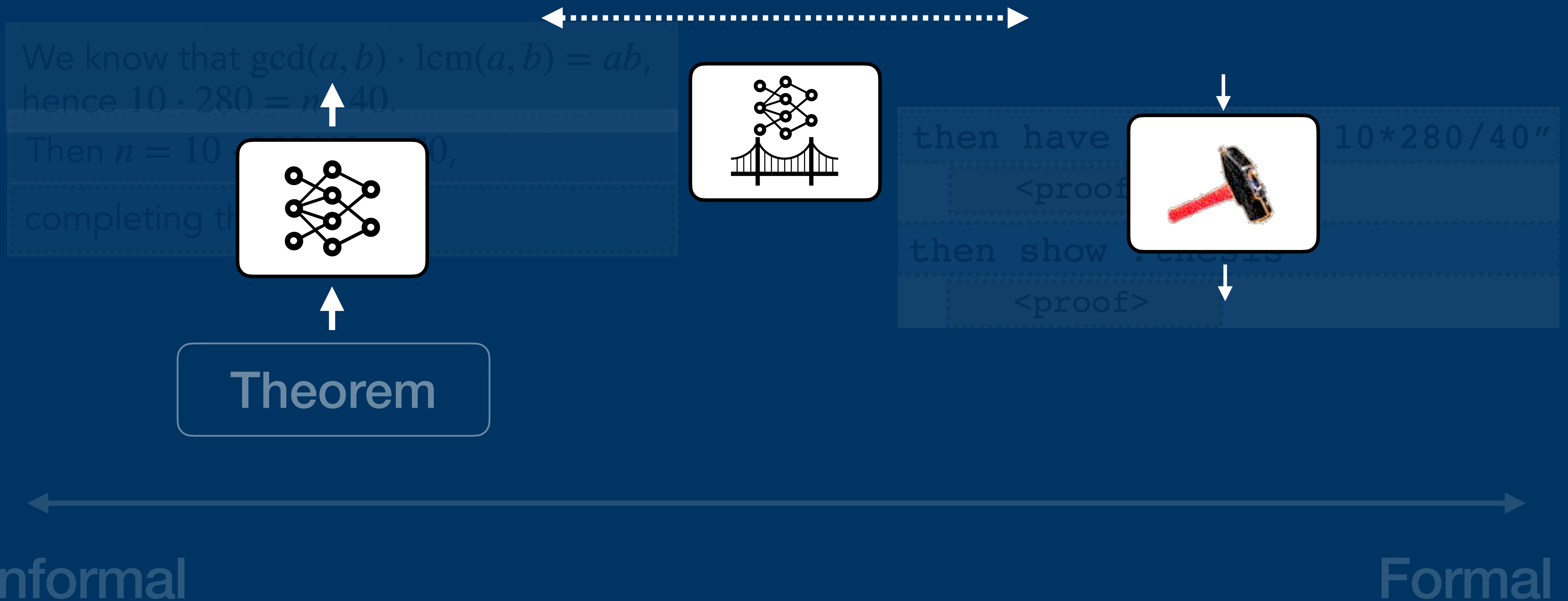
Sketch

Can learn to sketch *in-context!!!*

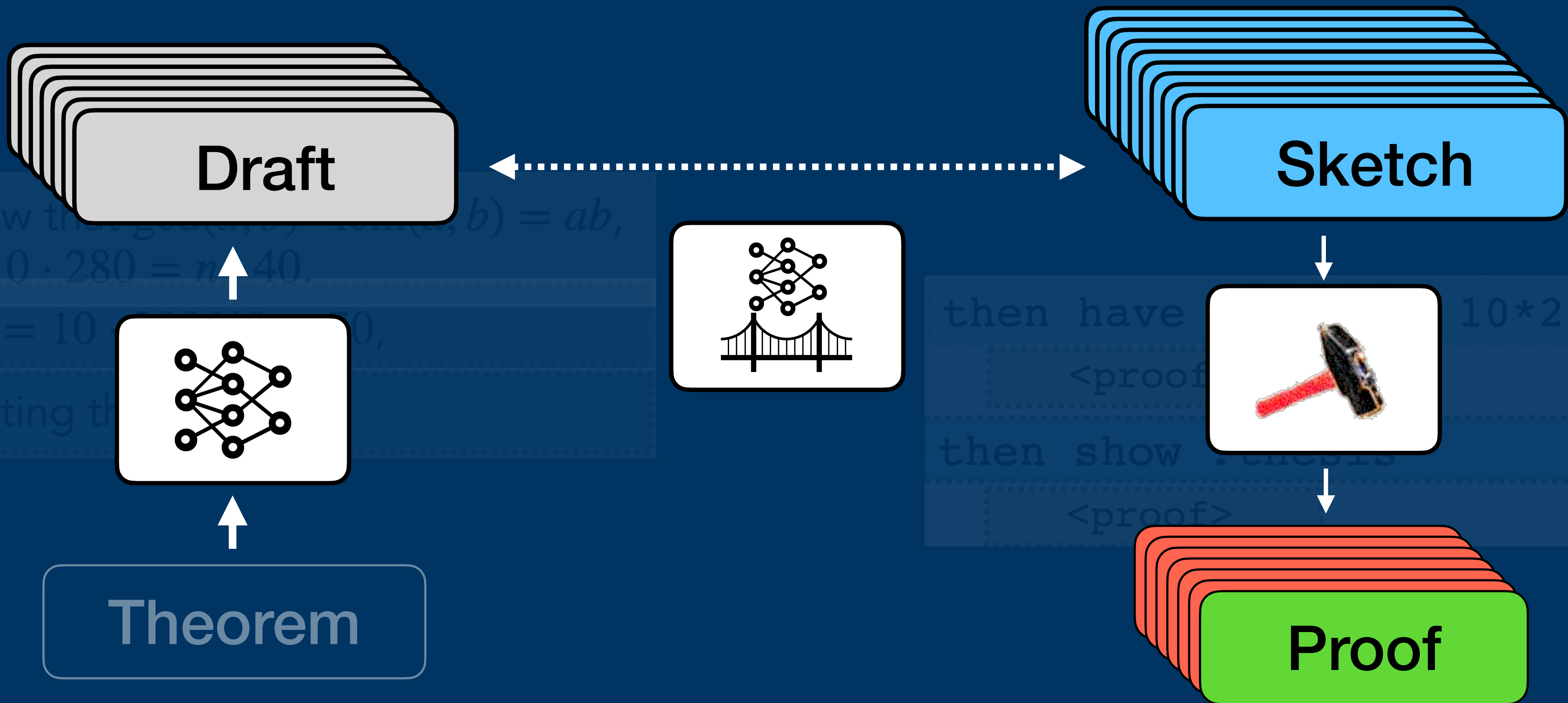
Informal

Formal

Generate & verify



Generate & verify

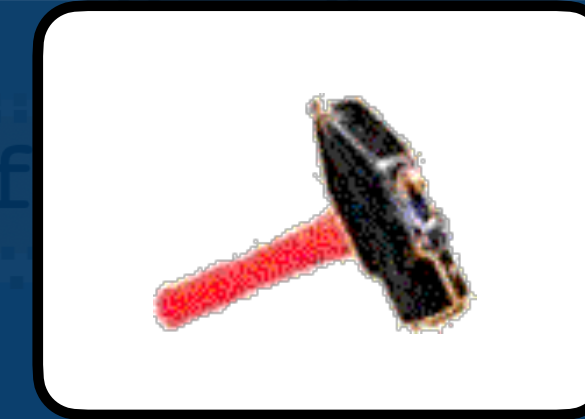
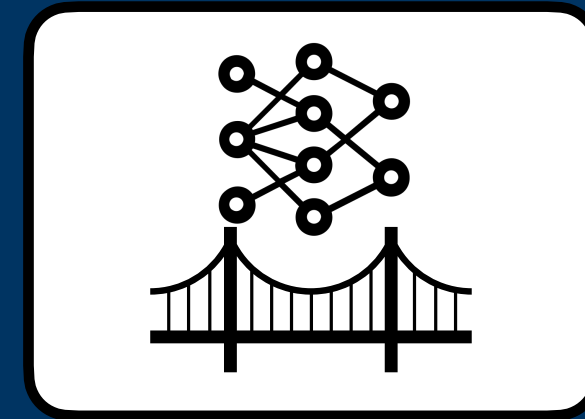
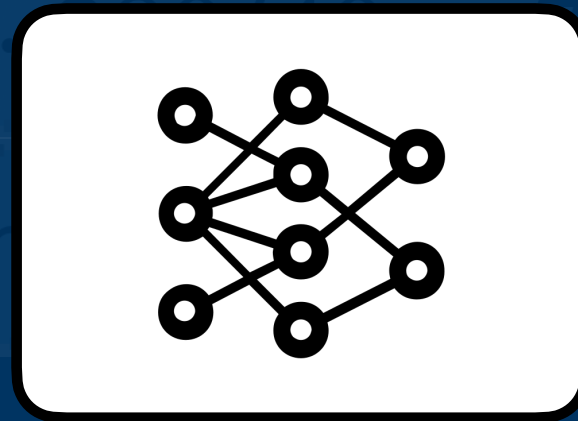


Informal

Formal

Draft, Sketch, Prove

We know that $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$,
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 "n = 10*280/40"  
  <proof>  
then show "n = 70"  
  <proof>
```

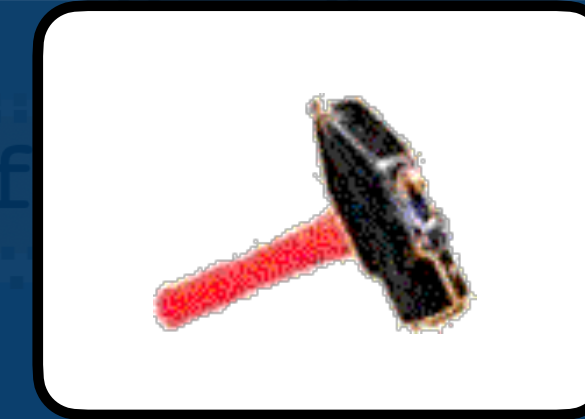
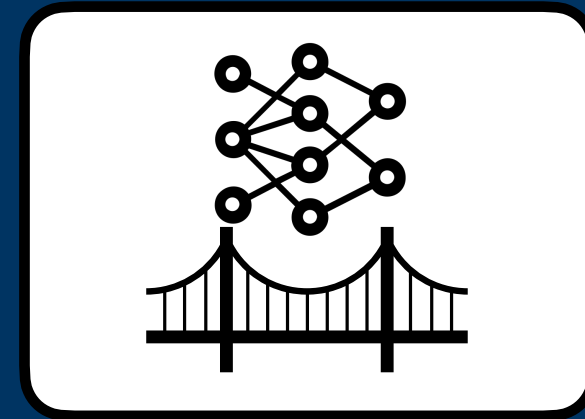
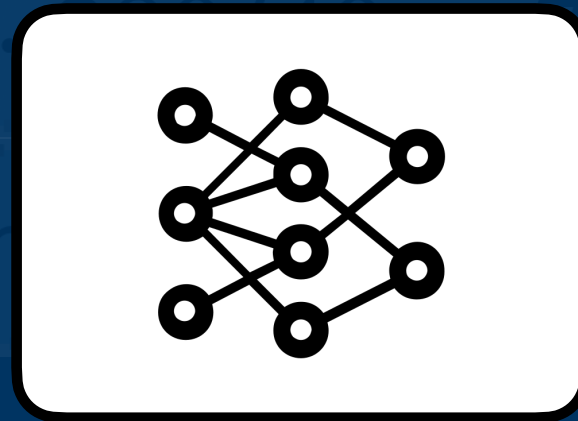


Informal

Formal

Draft, Sketch, Prove

We know that $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$,
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 "n = 10*280/40"  
  <proof>  
then show "n = 70"  
  <proof>
```

Flexibility
Data

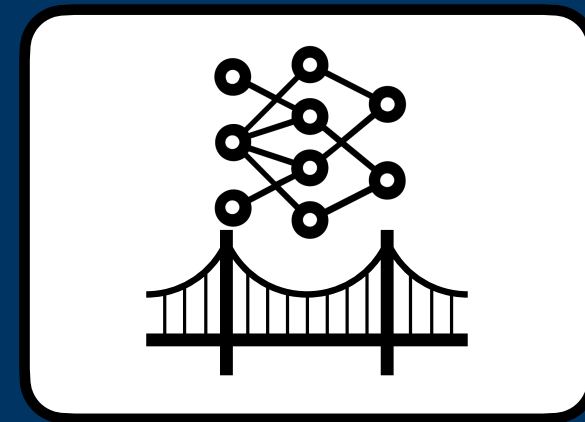
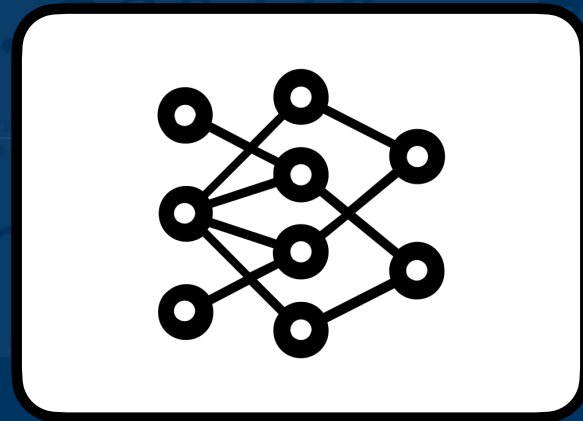


Informal

Formal

Draft, Sketch, Prove

We know that $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$,
hence $10 \cdot 280 = n \cdot 40$.
Then $n = 10 \cdot 280 / 40$,
completing the proof.



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

Flexibility
Data

Grounding

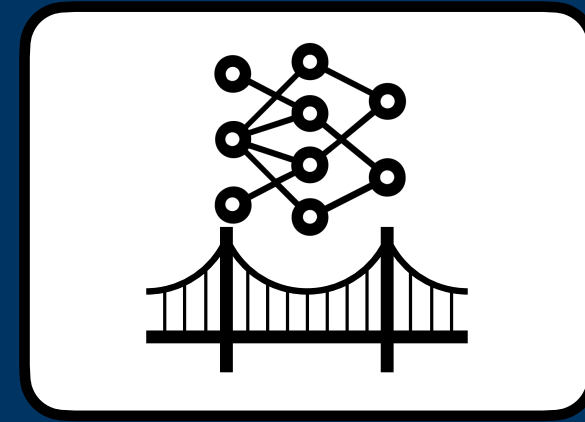
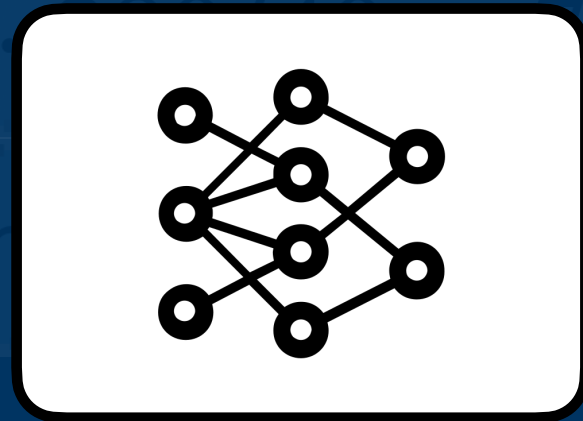


Informal

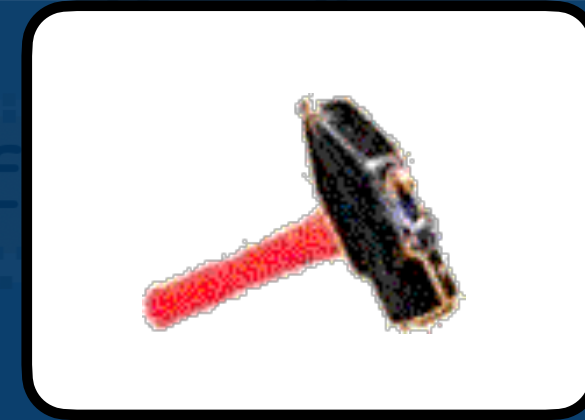
Formal

Draft, Sketch, Prove

We know that $\text{gcd}(a, b) \cdot \text{lcm}(a, b) = ab$,
hence $10 \cdot 280 = n \cdot 40$.
Then $n = 10 \cdot 280 / 40$,
completing the proof.



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



Flexibility
Data

Best of both worlds!

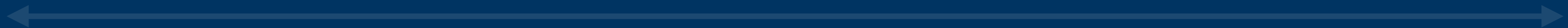
Grounding



Informal

Formal

Evaluation



Informal

Formal

Evaluation

- **miniF2F benchmark:** 488 problems from competitions and undergrad courses



Informal

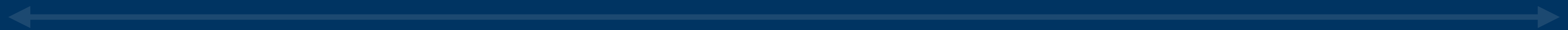
Formal

Evaluation

- **miniF2F benchmark:** 488 problems from competitions and undergrad courses



- Isabelle proof assistant



Informal

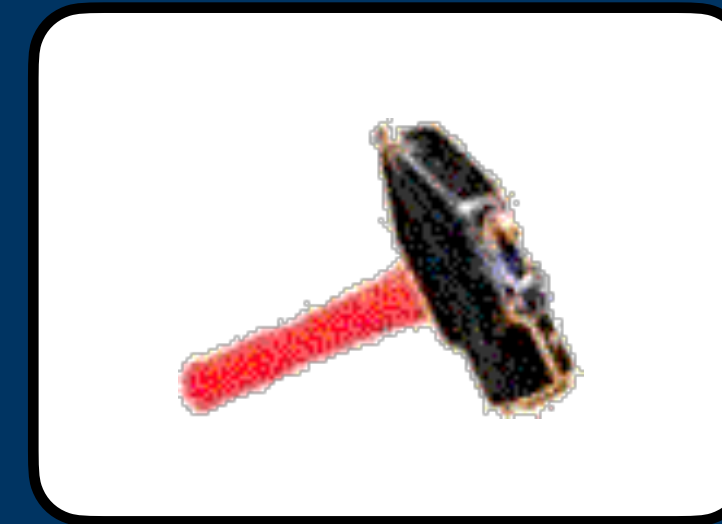
Formal

Evaluation

- **miniF2F benchmark:** 488 problems from competitions and undergrad courses



- Sledgehammer + heuristics



- Isabelle proof assistant



Informal

Formal

Evaluation

- **miniF2F benchmark:** 488 problems from competitions and undergrad courses



- Sledgehammer + heuristics



- Isabelle proof assistant



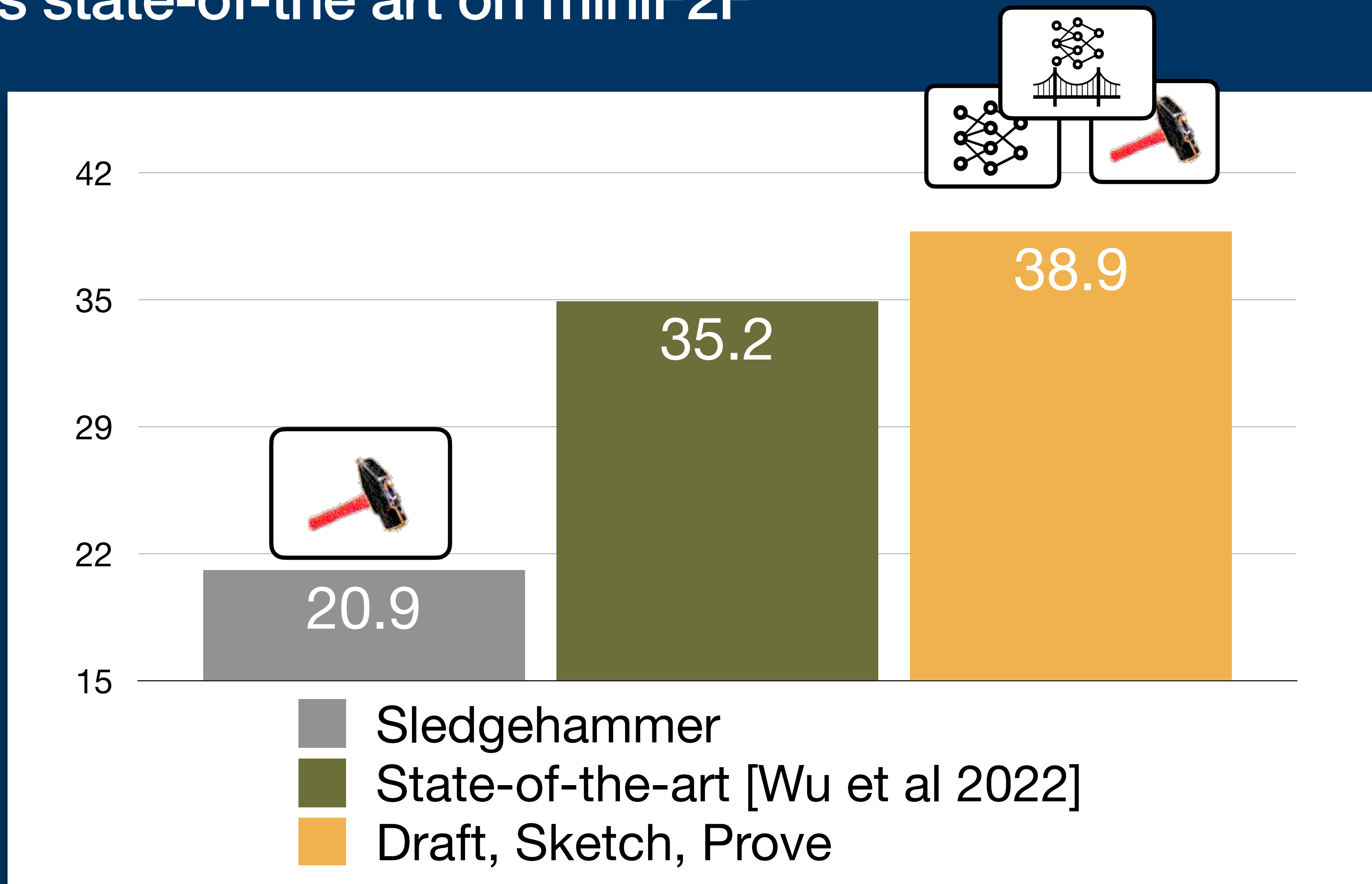
- State-of-the-art neural theorem prover

THOR with Expert Iteration
[Wu et al 2022]

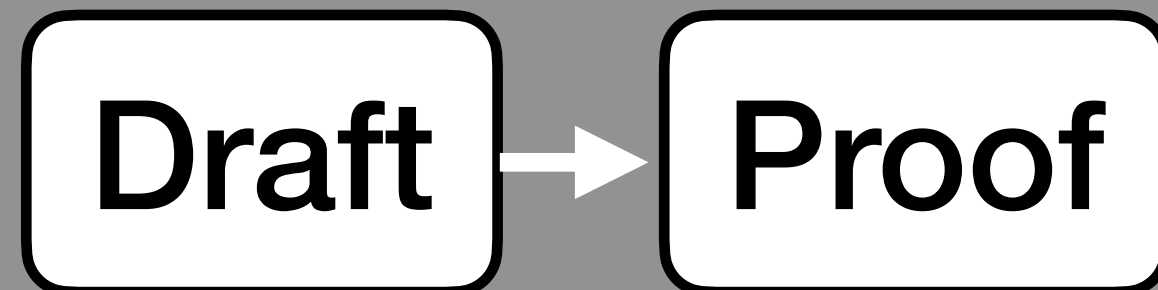
Informal

Formal

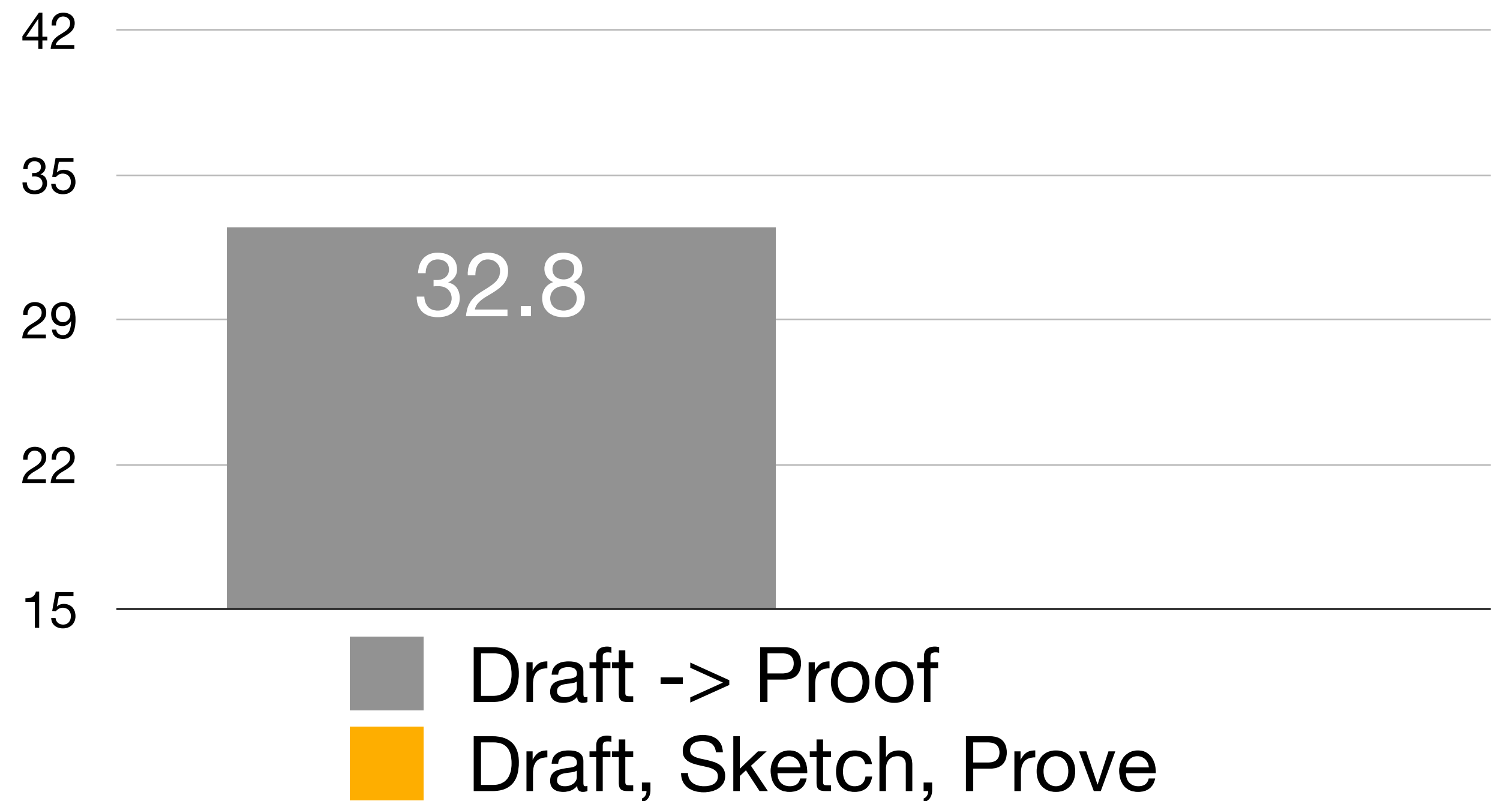
Achieves state-of-the-art on miniF2F



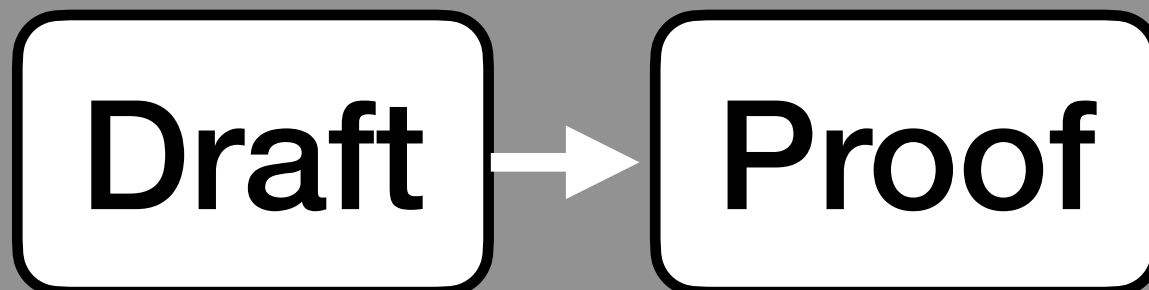
Proving without sketching isn't enough



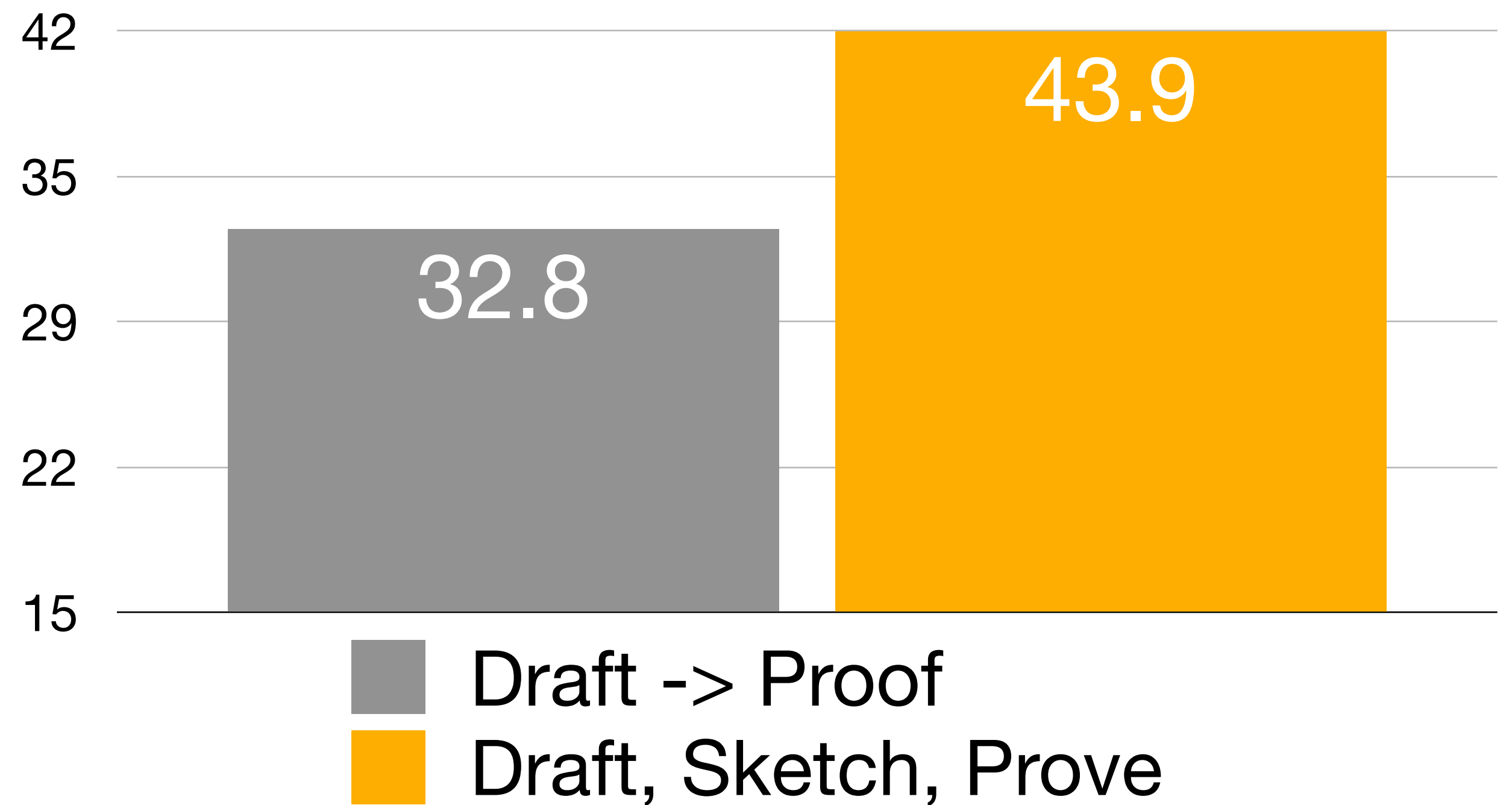
versus



Proving without sketching isn't enough

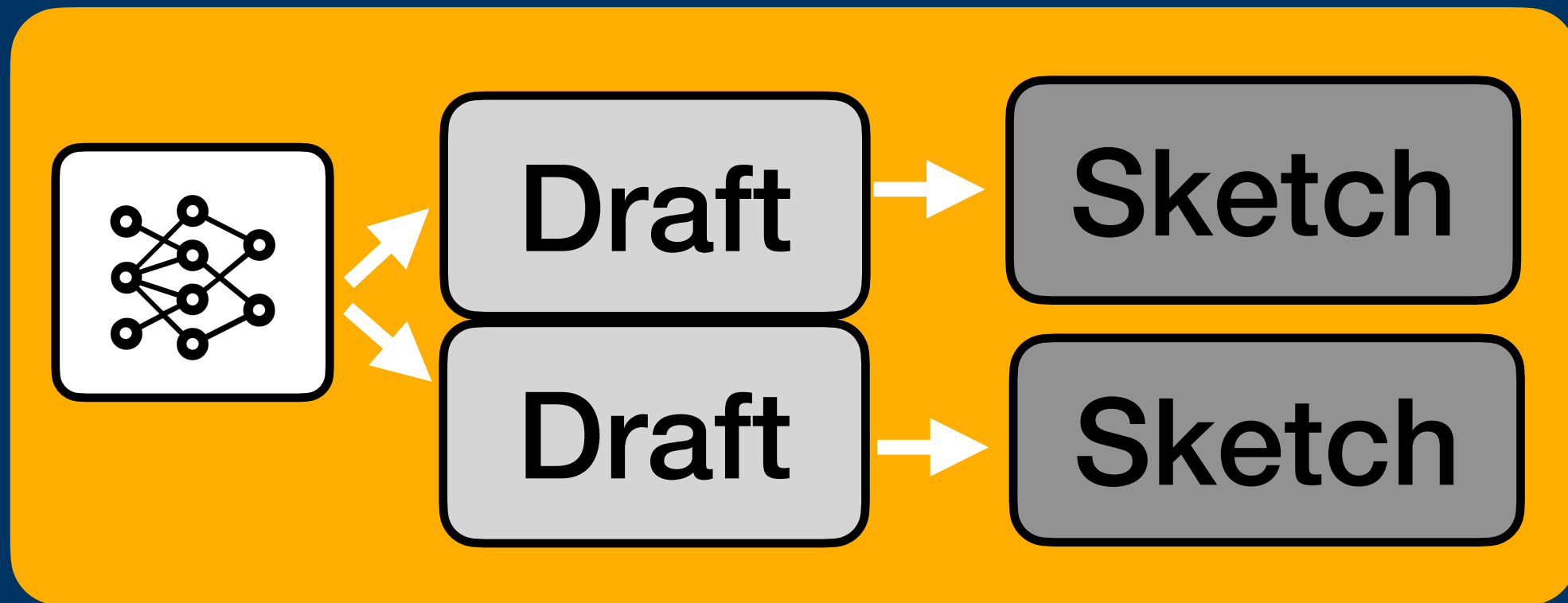


versus

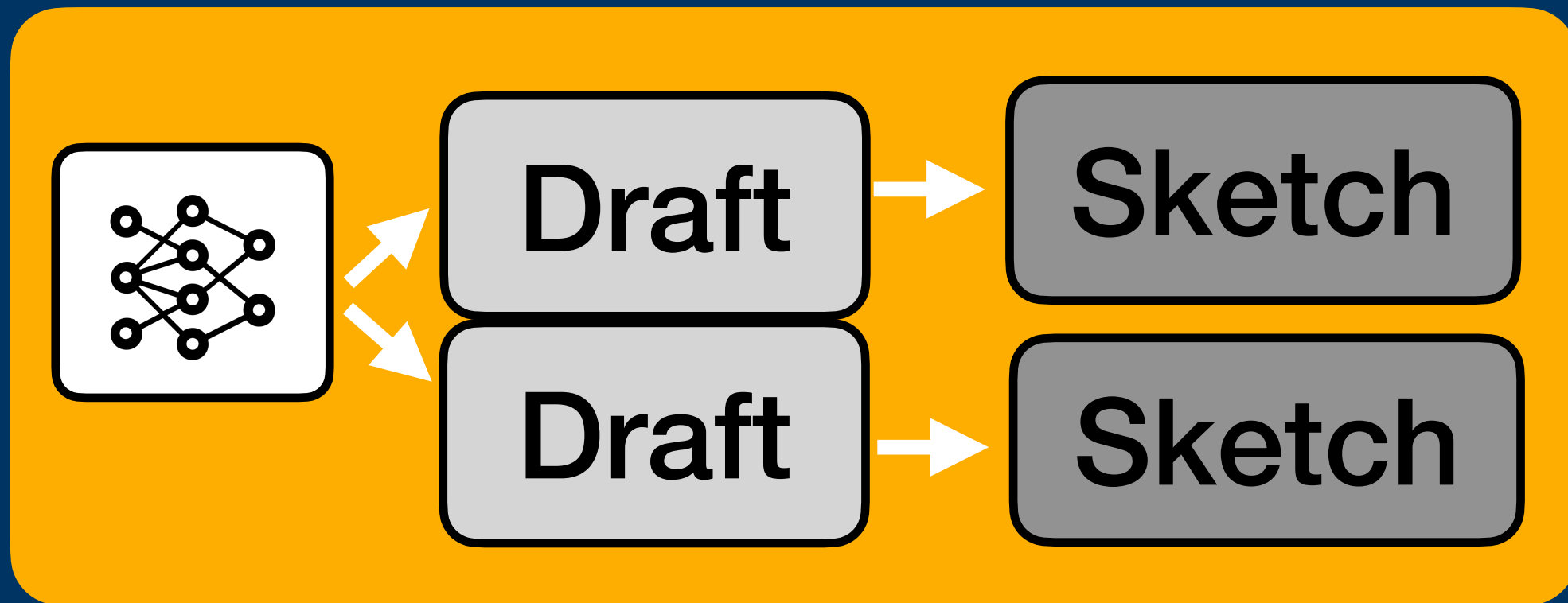


Neural drafts can solve more problems than human-written drafts

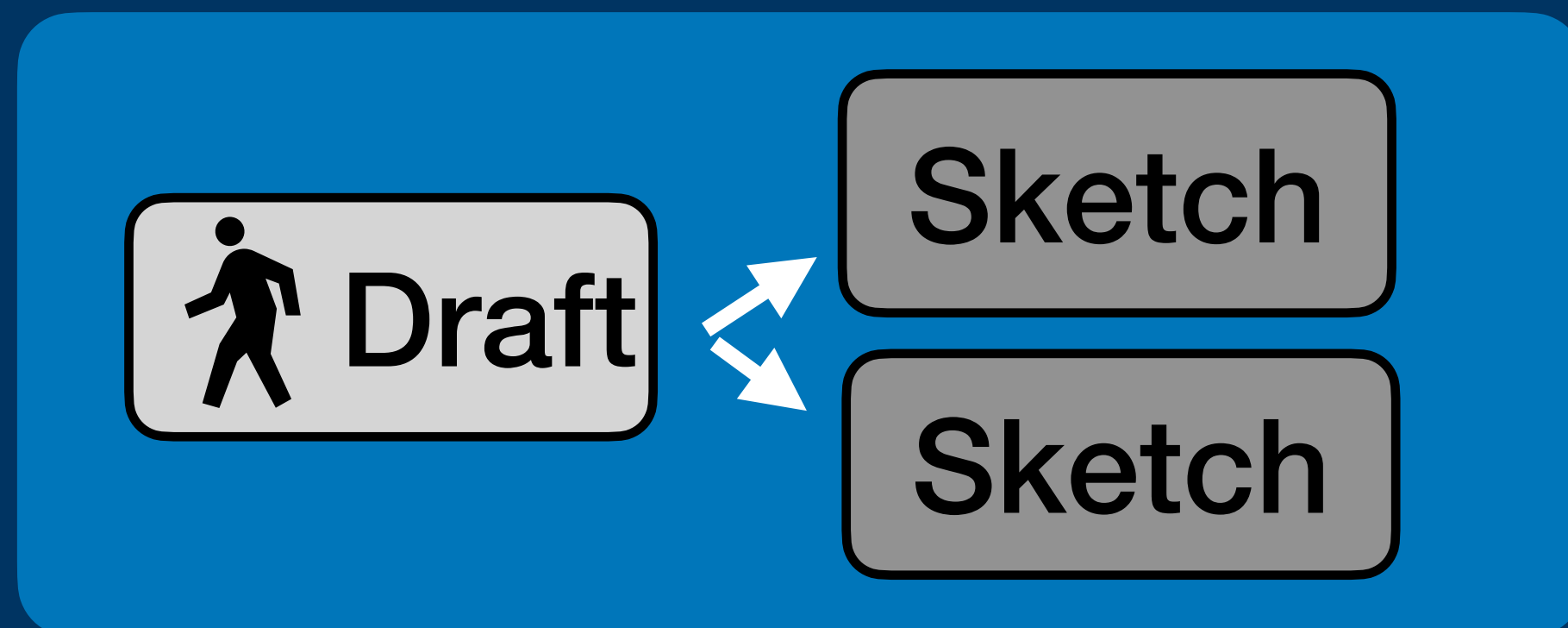
Neural drafts can solve more problems than human-written drafts



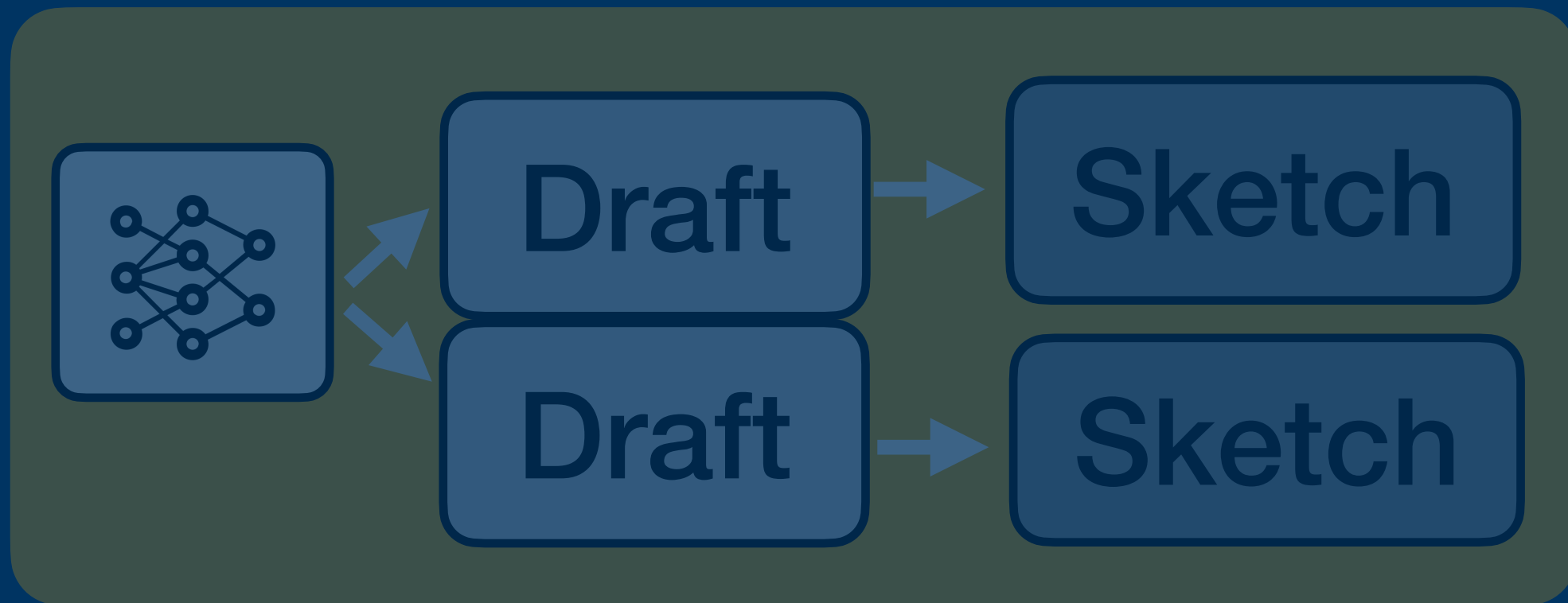
Neural drafts can solve more problems than human-written drafts



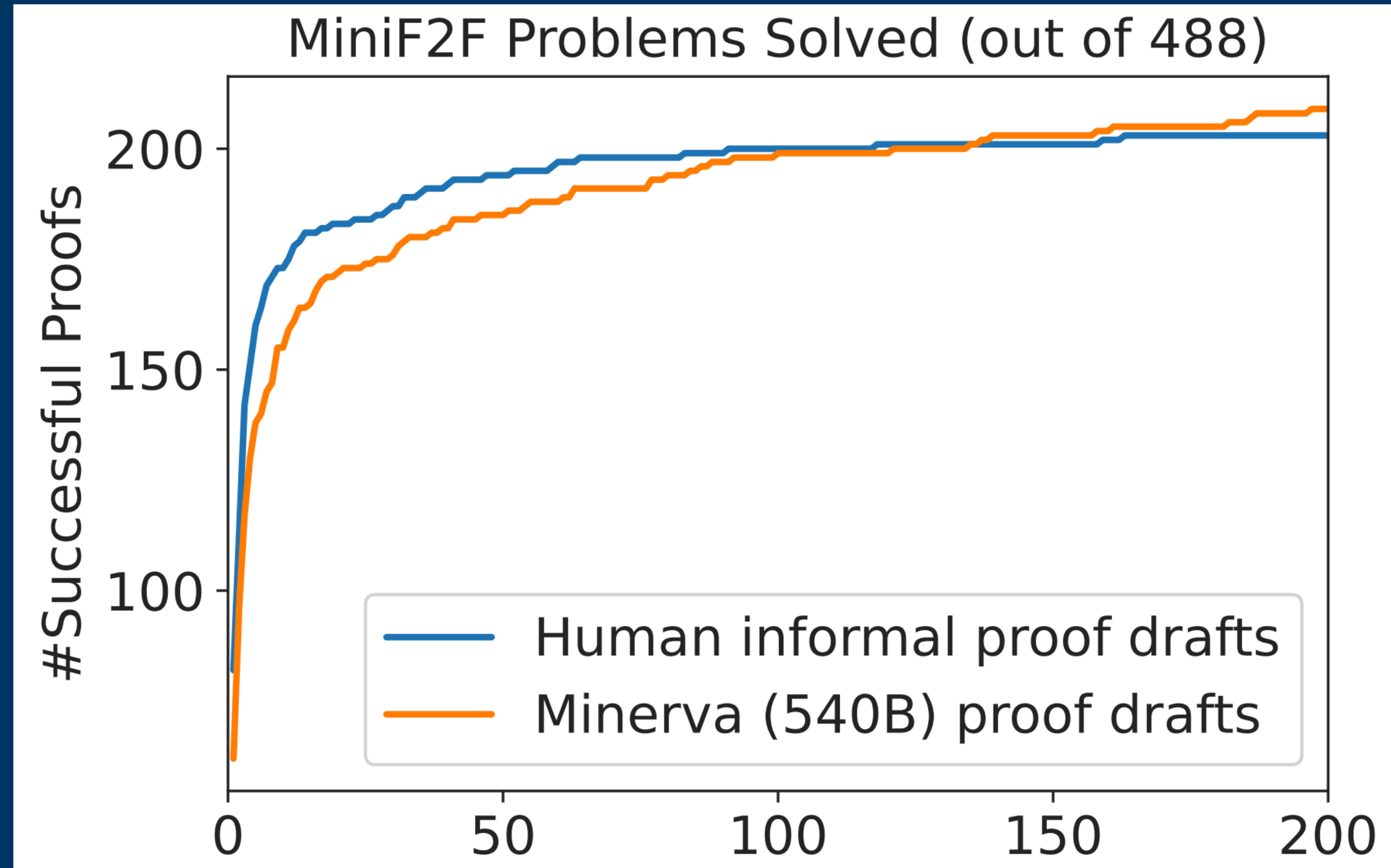
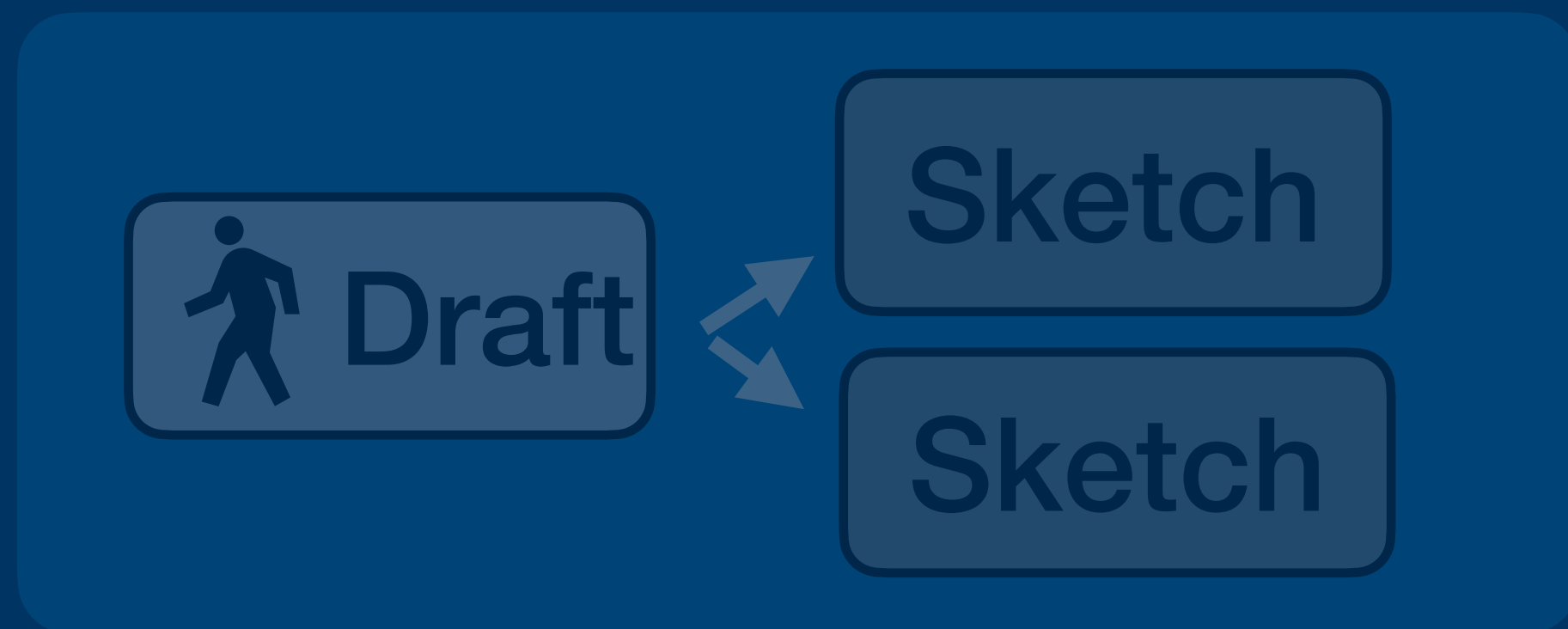
versus



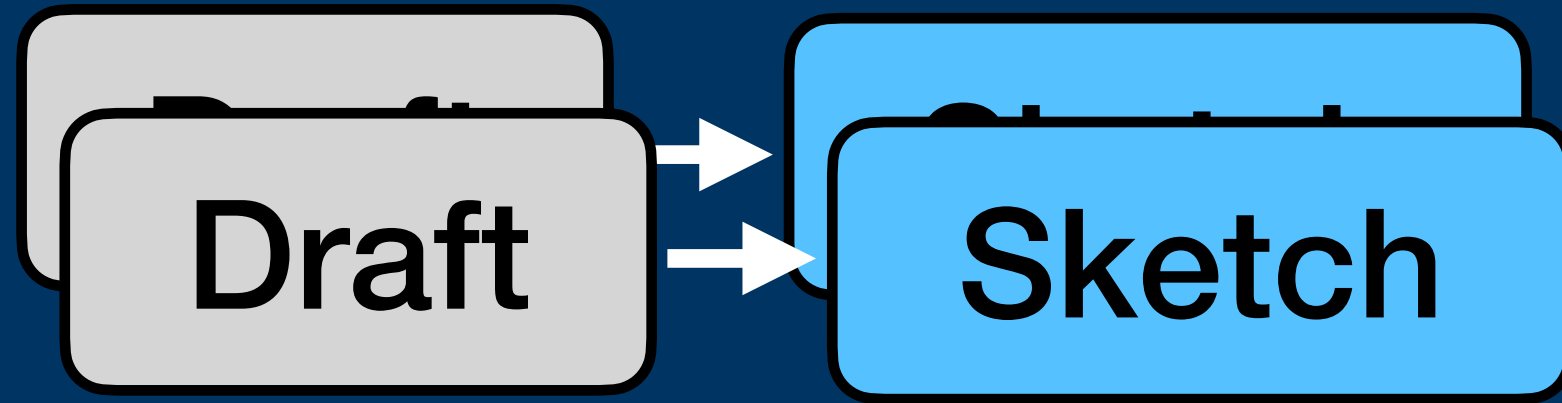
Neural drafts can solve more problems than human-written drafts



versus

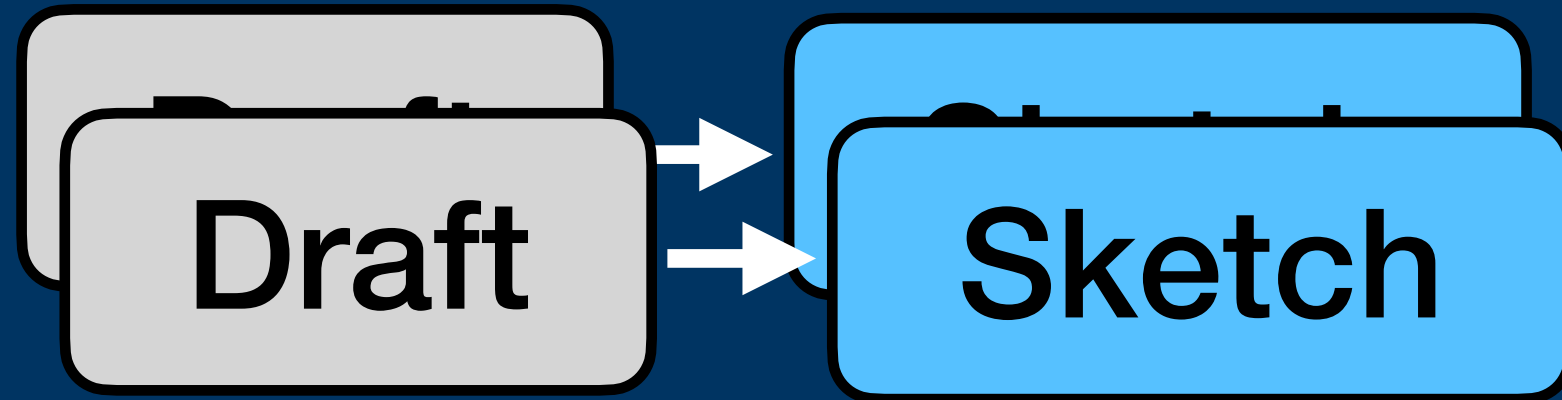


More drafts > more sketches

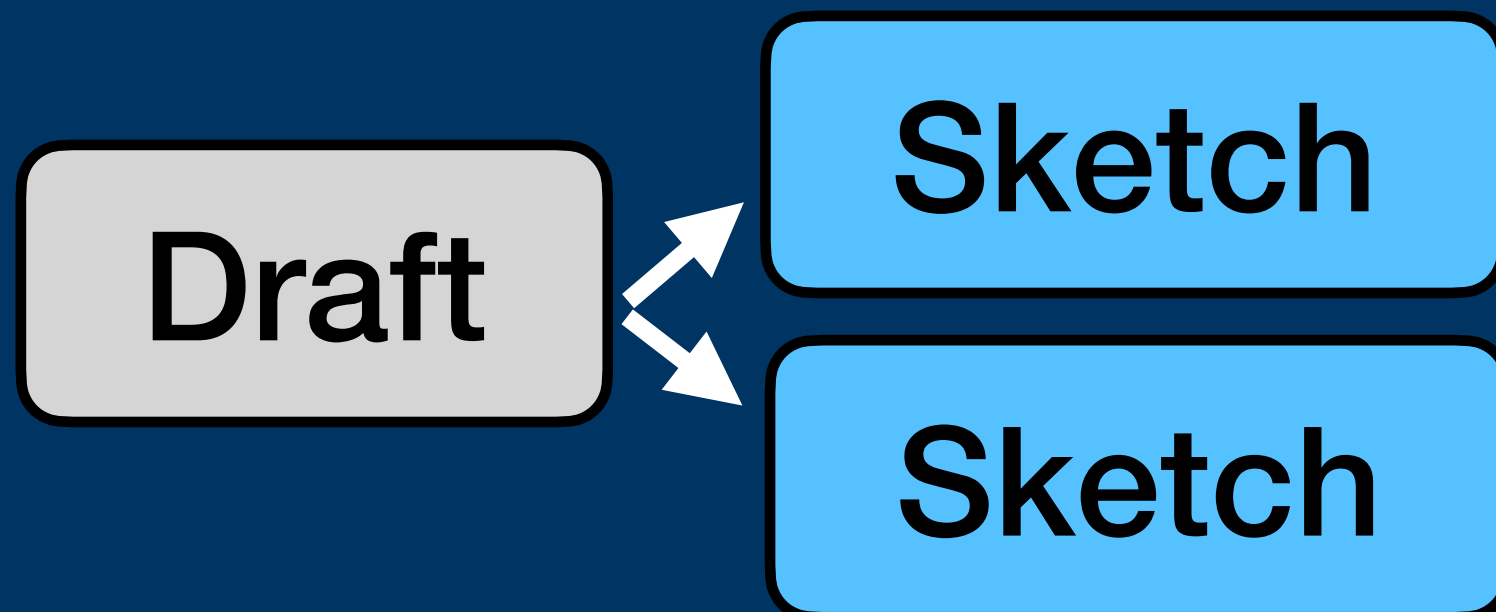


versus

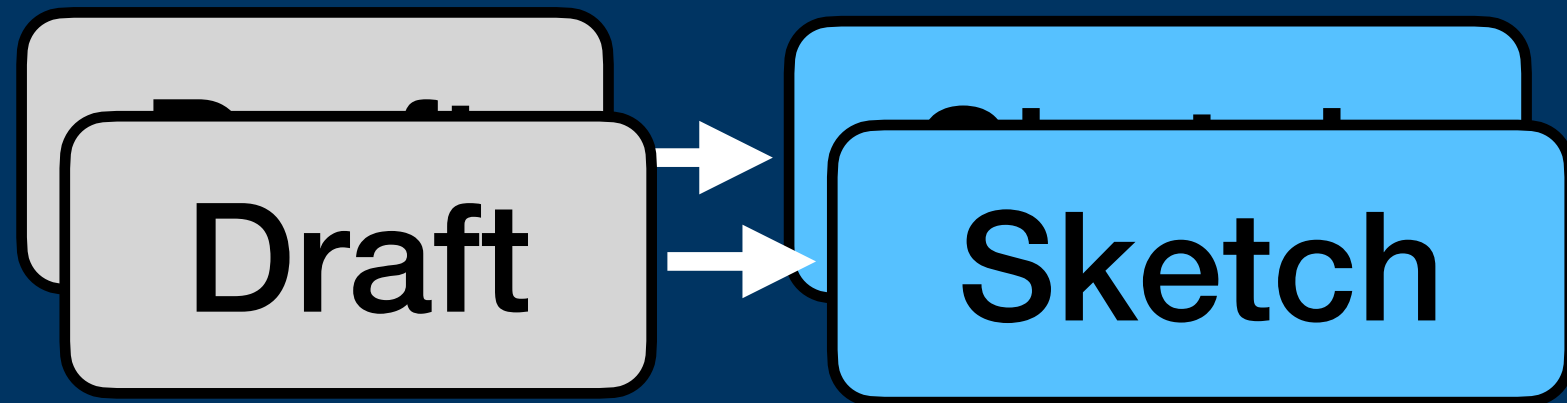
More drafts > more sketches



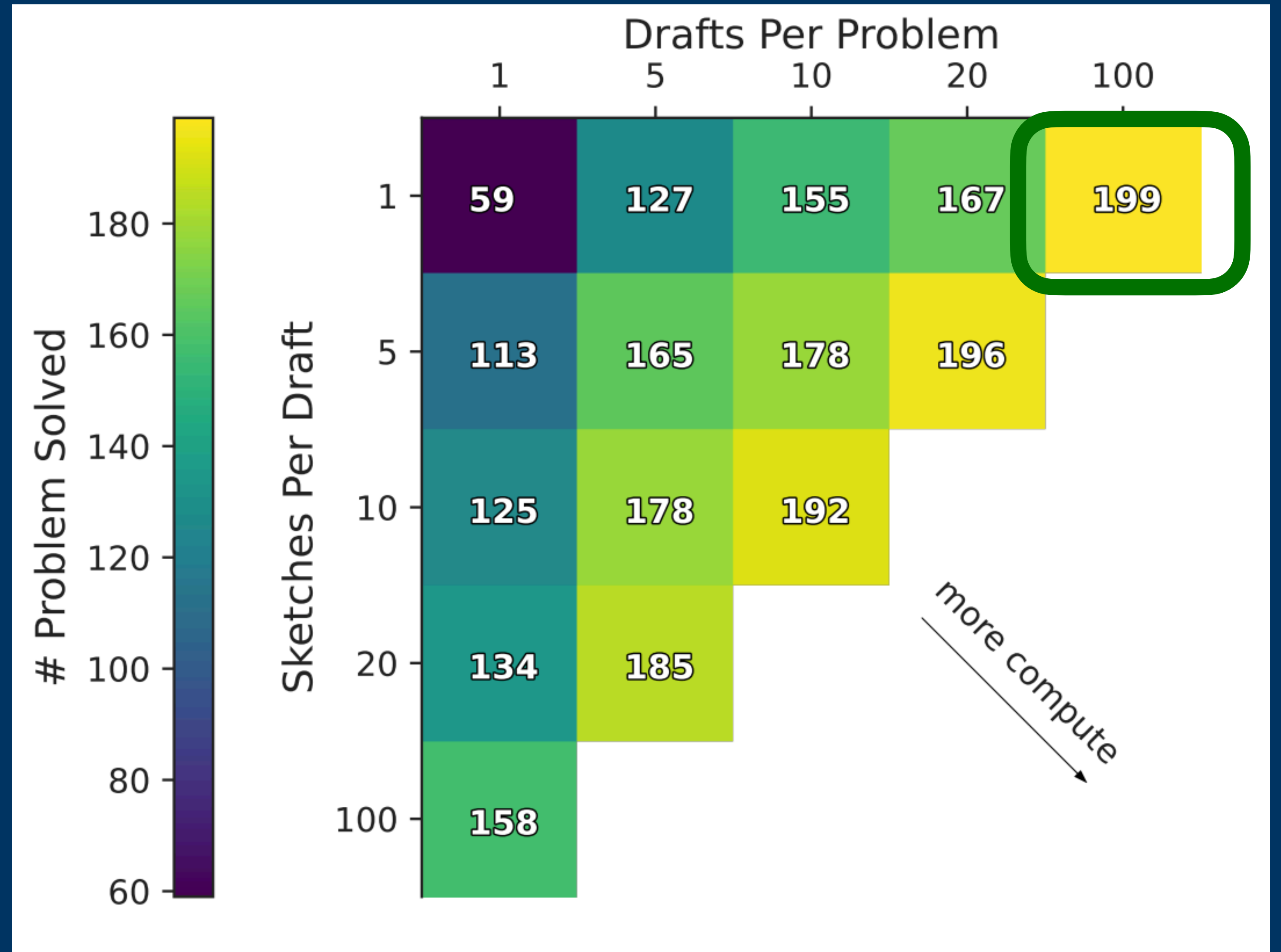
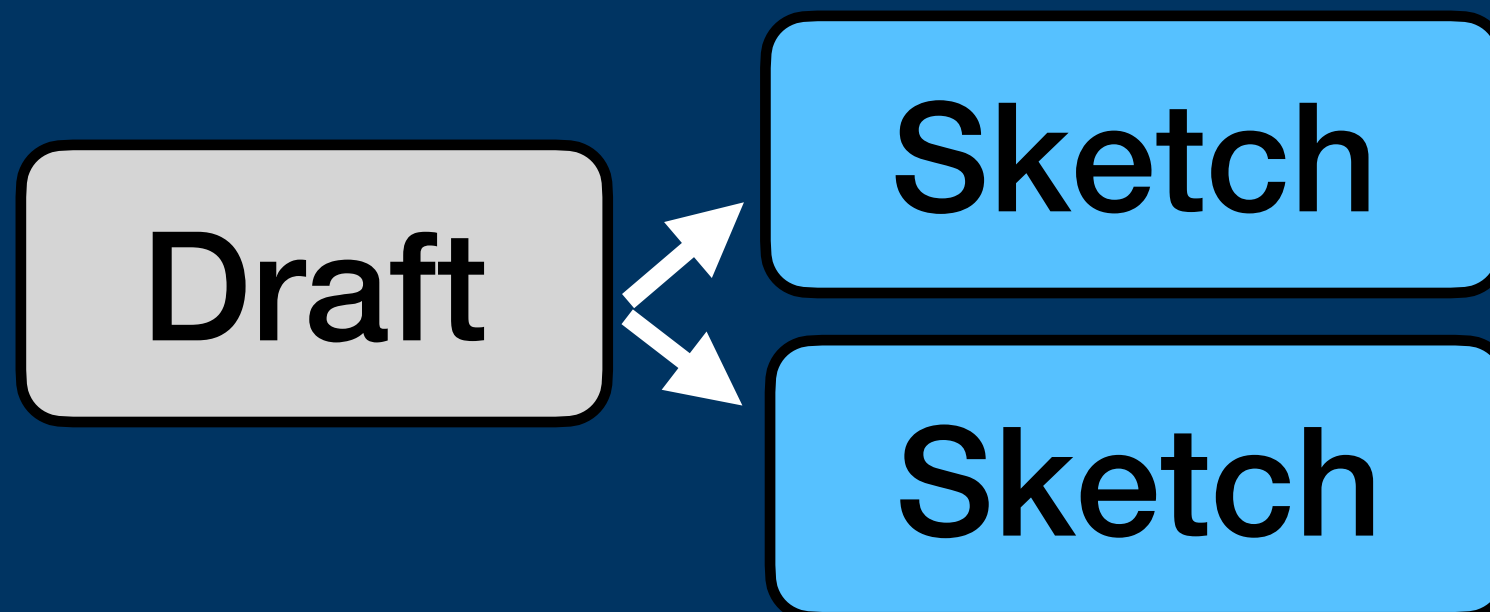
versus



More drafts > more sketches



versus





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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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

Formal Proof:

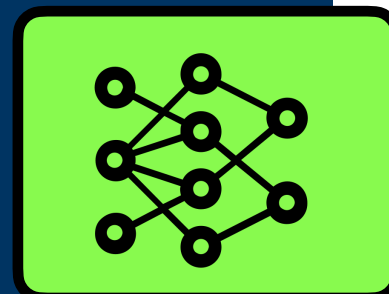
```

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

```

Neural Draft (Minerva)



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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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

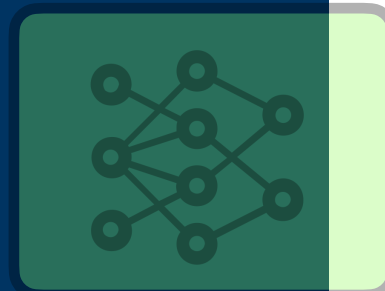
Formal Proof:

```

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

```



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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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]

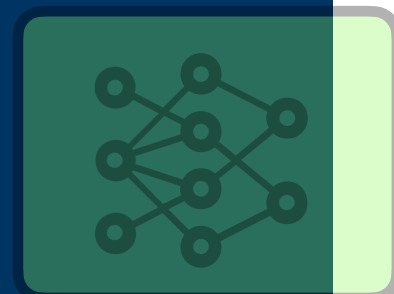
qed

Formal statement

(given)

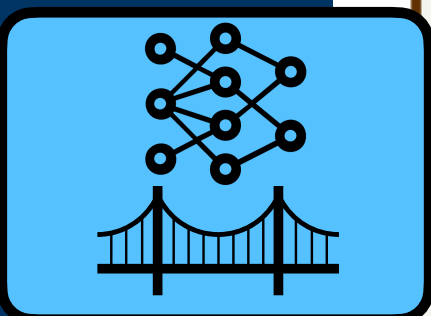
Neural Draft

(Minerva)



Neural Sketch

(Codex)



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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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]

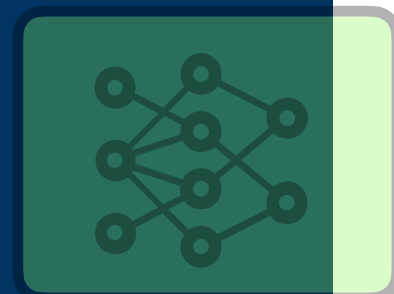
then show ?thesis

using c1

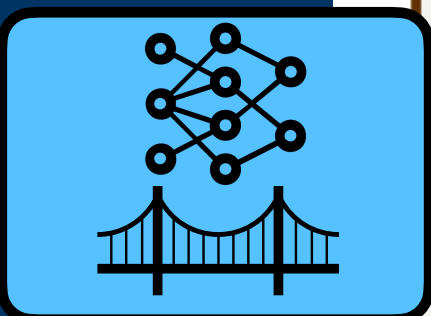
by blast [ATP]

qed

Neural Draft (Minerva)



Neural Sketch (Codex)



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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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]

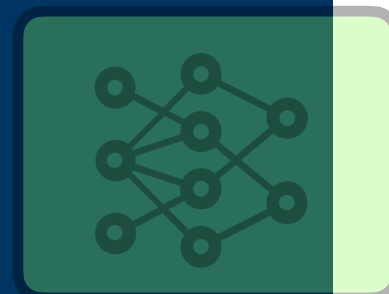
then show ?thesis

using c1

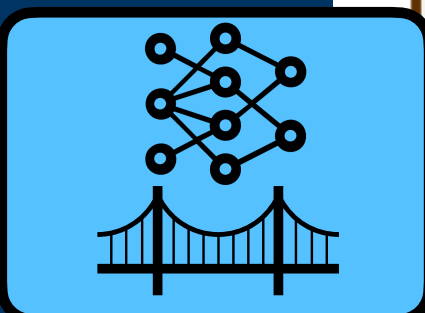
by blast [ATP]

qed

Neural Draft (Minerva)



Neural 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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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]

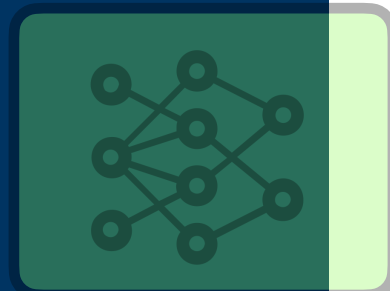
then show ?thesis

using c1

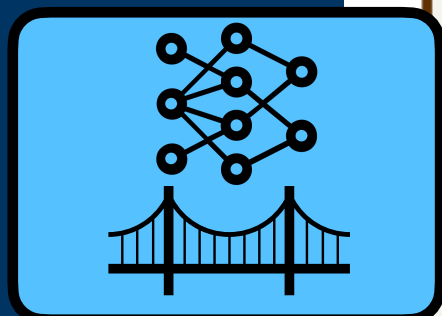
by blast [ATP]

qed

Neural Draft
(Minerva)



Neural 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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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]

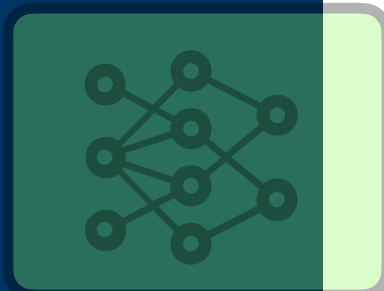
then show ?thesis

using c1

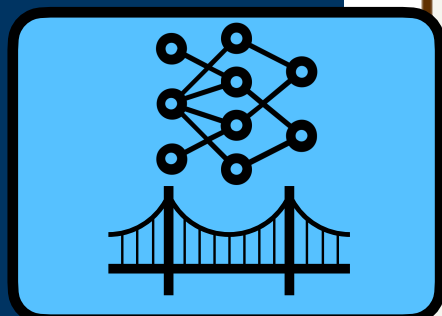
by blast [ATP]

qed

Neural Draft
(Minerva)



Neural 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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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]

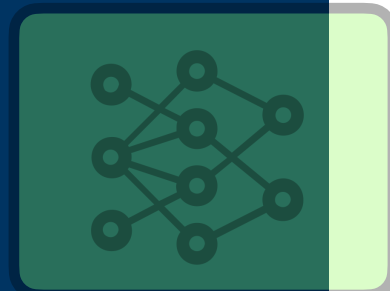
then show ?thesis

using c1

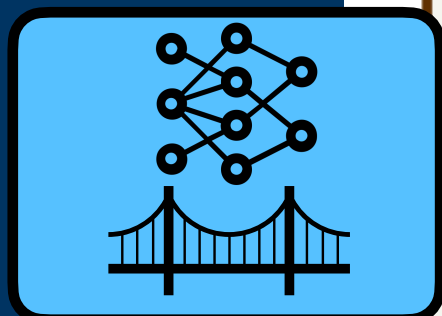
by blast [ATP]

qed

Neural Draft (Minerva)



Neural 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

$$21n + 4 = 1 \cdot (14n + 3) + 7n + 1$$

$$14n + 3 = 2 \cdot (7n + 1) + 1.$$

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

Summary

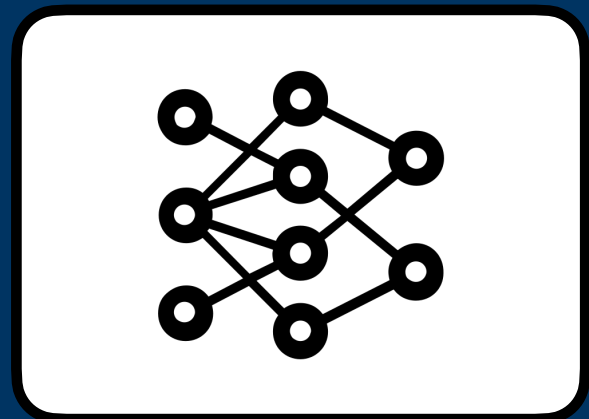


Informal

Formal

Summary

- Language models are capable of proving some sophisticated mathematical theorems

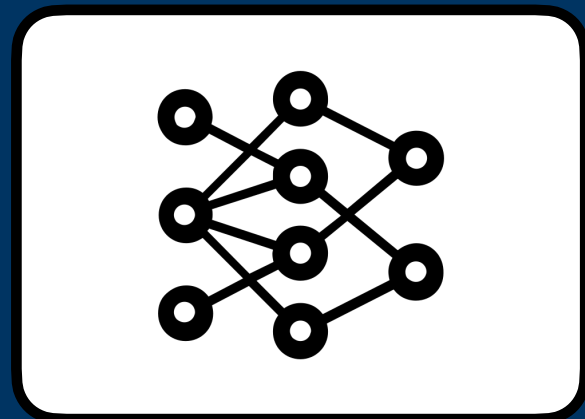


Informal

Formal

Summary

- Language models are capable of proving some sophisticated mathematical theorems
- **Key tradeoff:** flexible vs. trustworthy

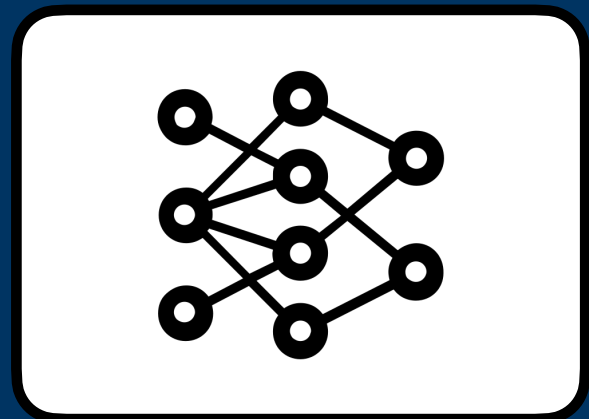


Informal

Formal

Summary

- Language models are capable of proving some sophisticated mathematical theorems
- **Key tradeoff:** flexible vs. trustworthy
- Sketching helps bridge the gap between informal and formal

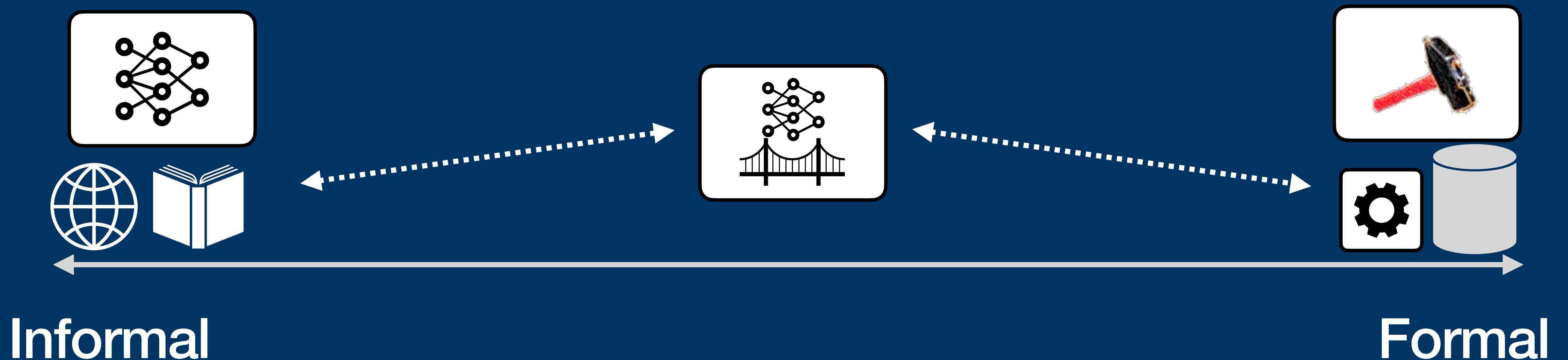


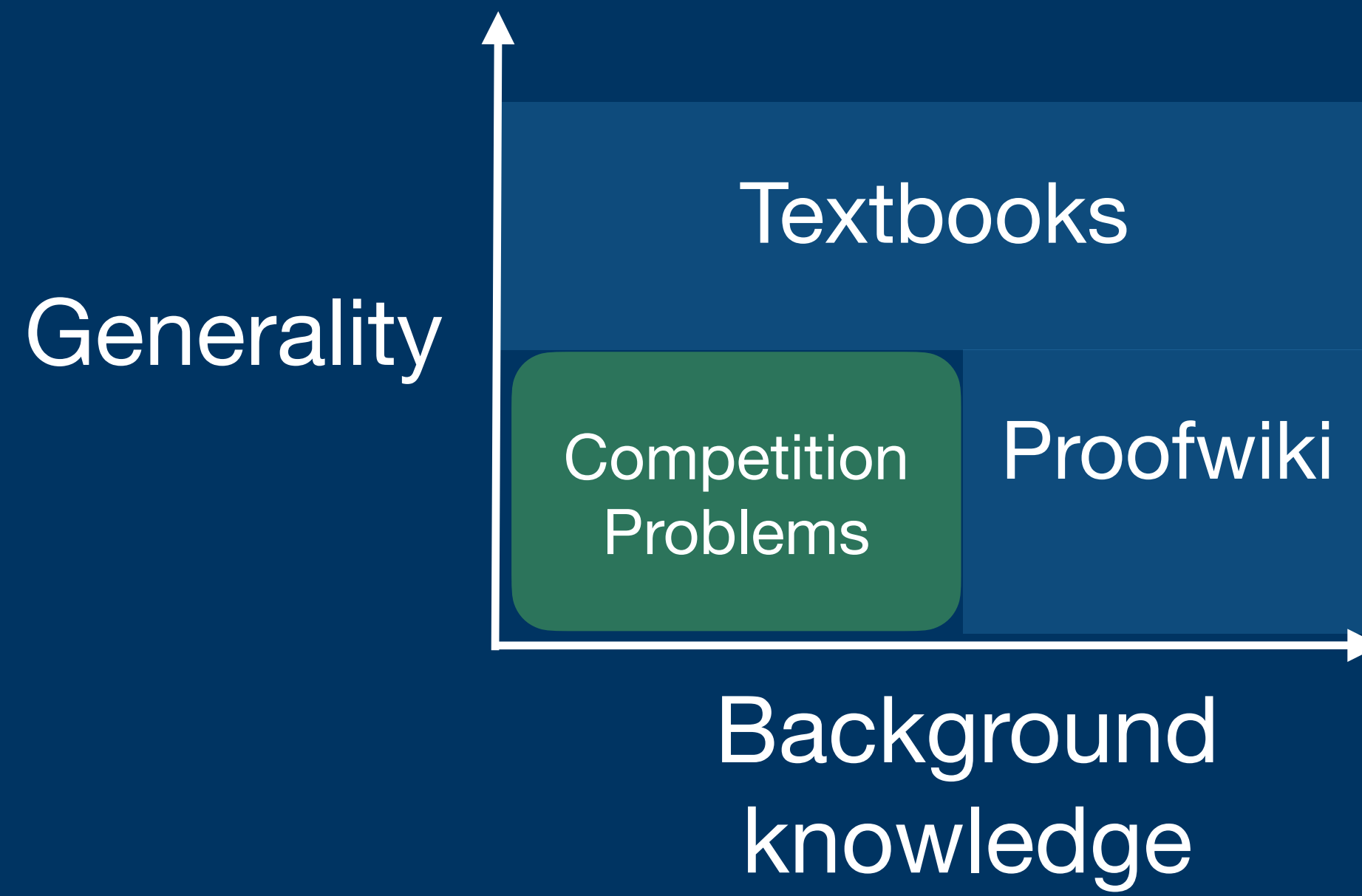
Informal

Formal

Summary

- Language models are capable of proving some sophisticated mathematical theorems
- **Key tradeoff:** flexible vs. trustworthy
- Sketching helps bridge the gap between informal and formal
- Language models serve as “high-level reasoners”





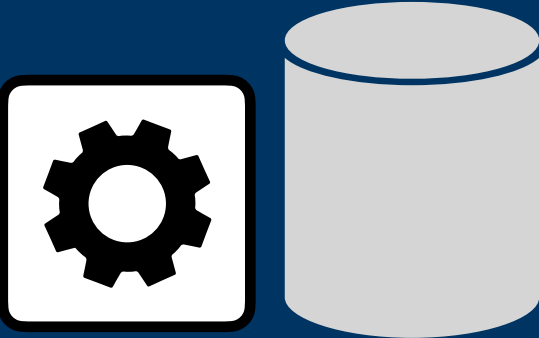
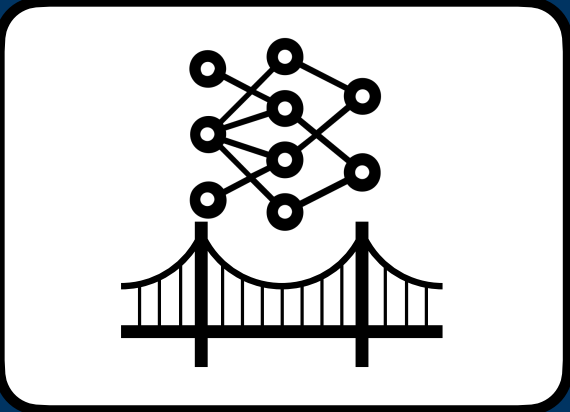
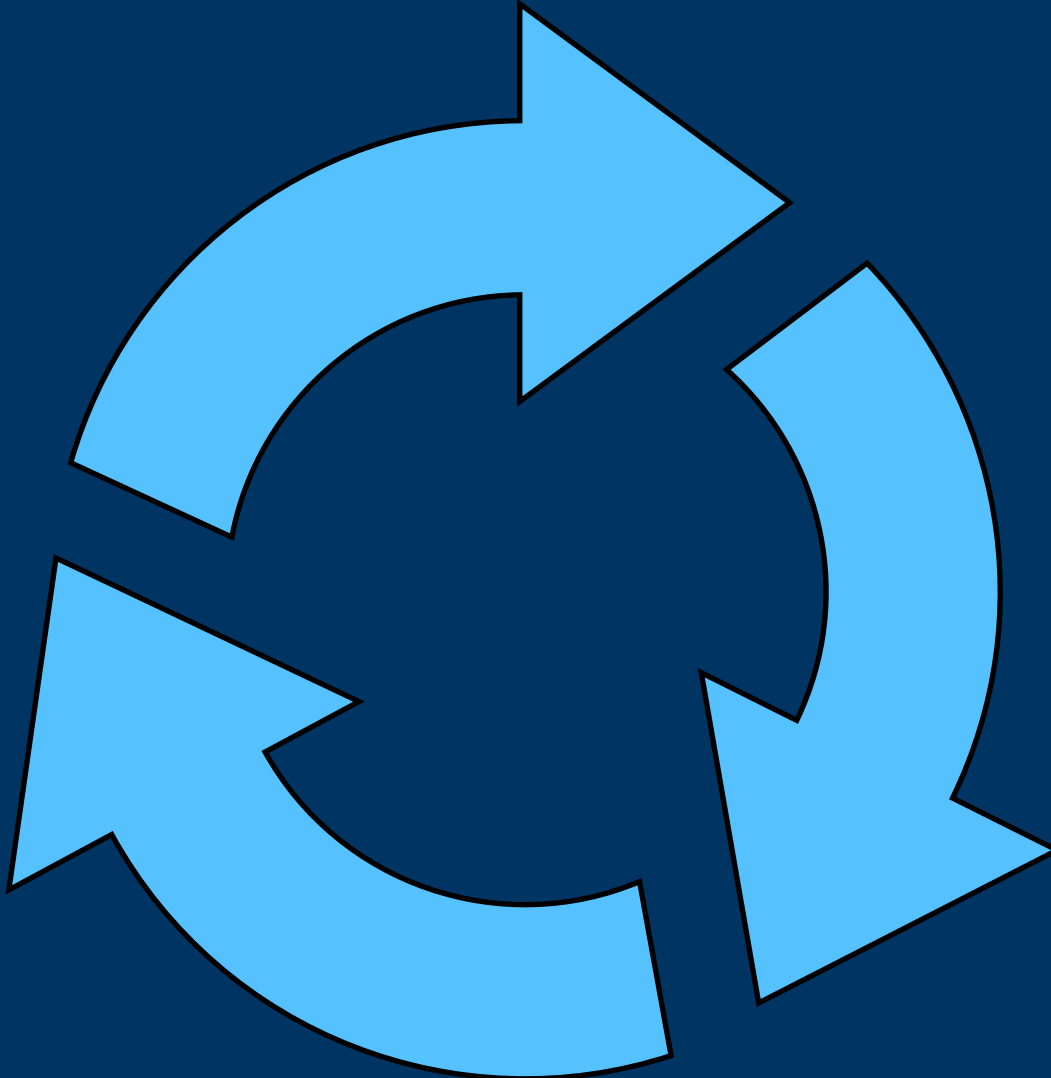
Informal

Formal

Interaction and Feedback

Human feedback

Machine feedback



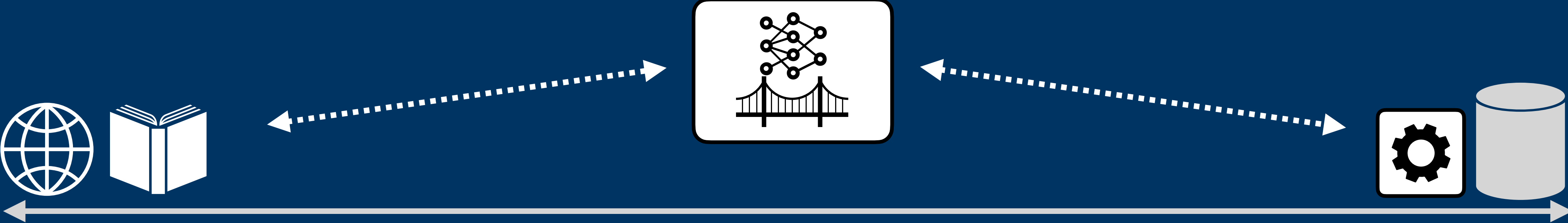
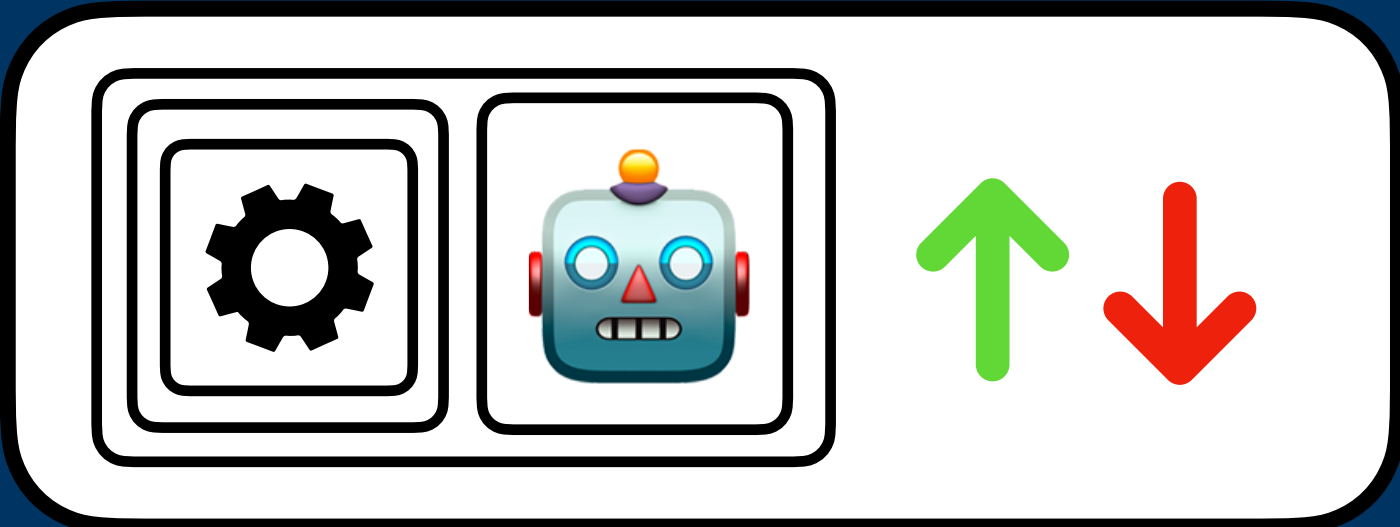
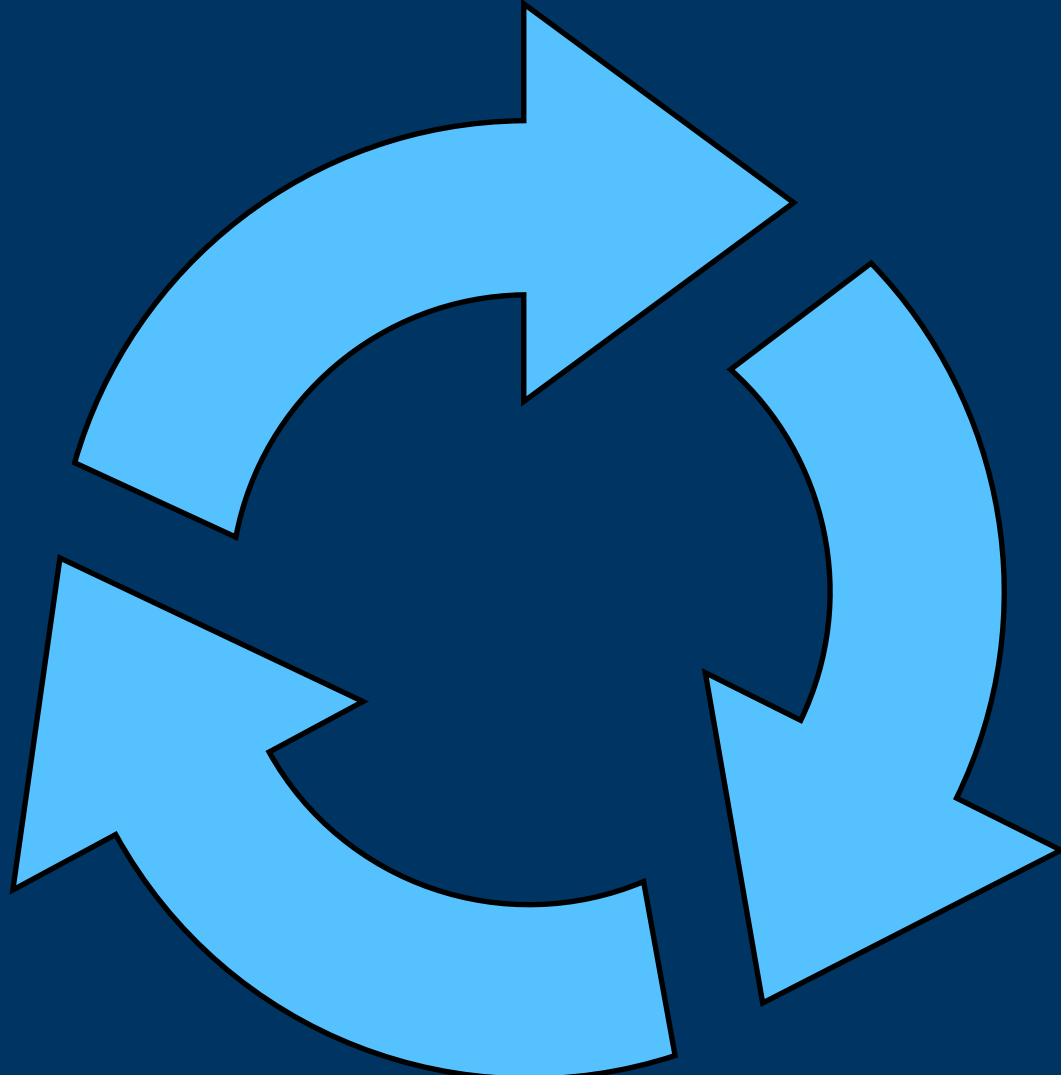
Informal

Formal

Interaction and Feedback

Human feedback

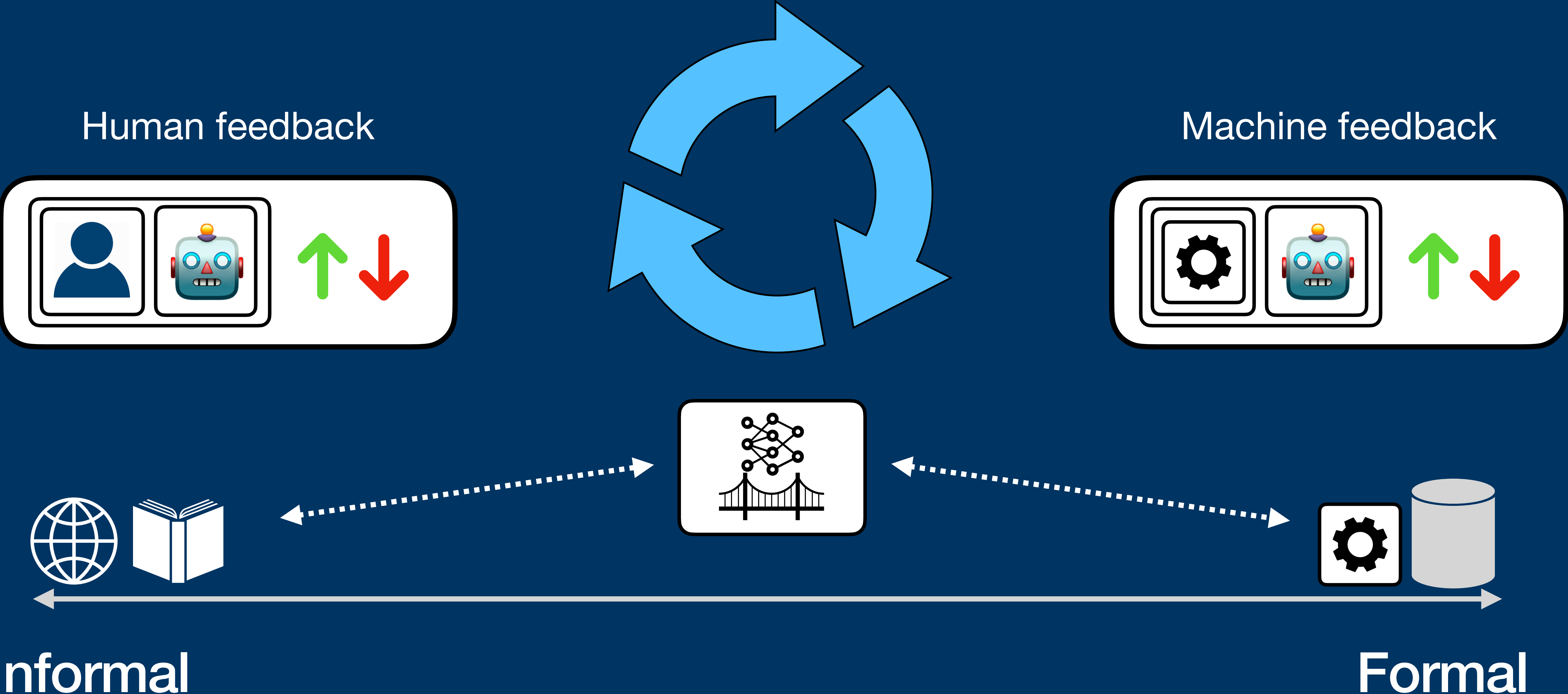
Machine feedback



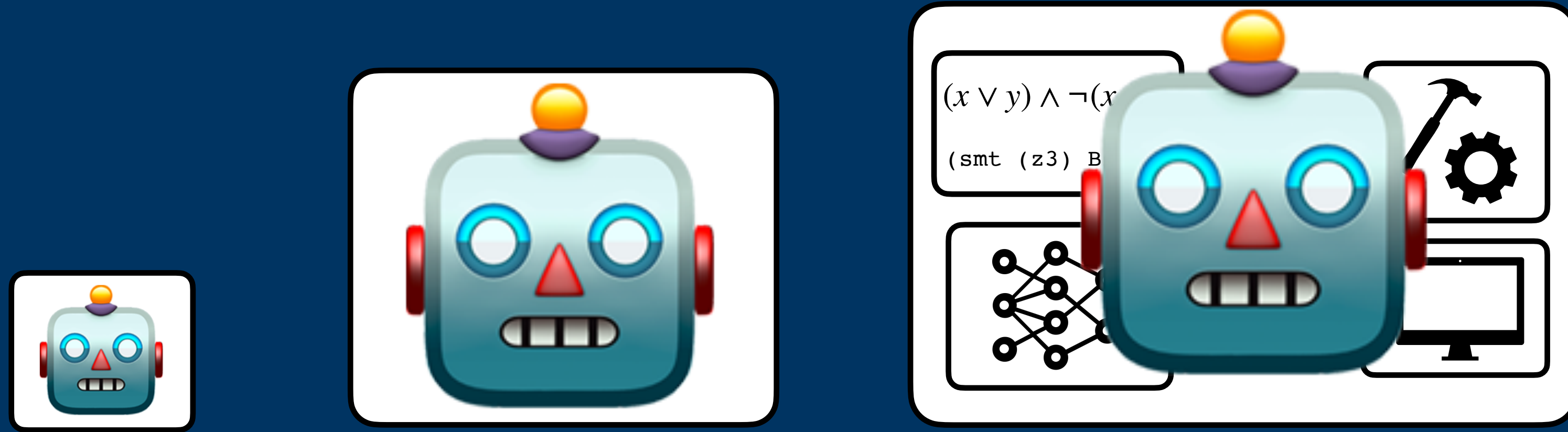
Informal

Formal

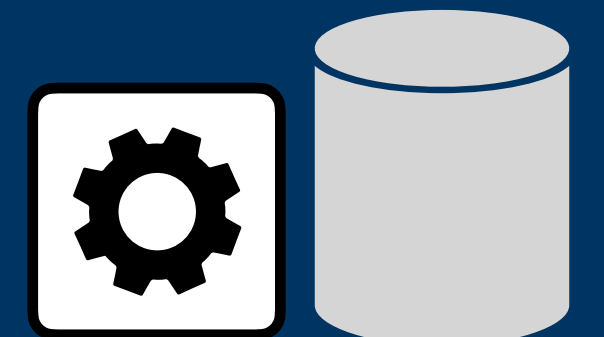
Interaction and Feedback



Modularity



Informal



Formal



NaturalProver: Grounded Mathematical Proof Generation with Language Models
S. Welleck*, J. Liu*, X. Lu, H. Hajishirzi, Y. Choi
NeurIPS 2022.

Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs
A. Jiang*, S. Welleck*, J. Zhou*, T. Lacroix, J. Liu, W. Li, M. Jamnik, Y. Wu, G. Lample
In submission, ICLR 2023