Let AI Explain Math

(Ongoing) AbductiveTP: Abduction via Analogical Divide-and-conquer Planning Facilitates Formal Mathematical Reasoning

Presented by Zory Zhang @ I

Apr 25, 2024

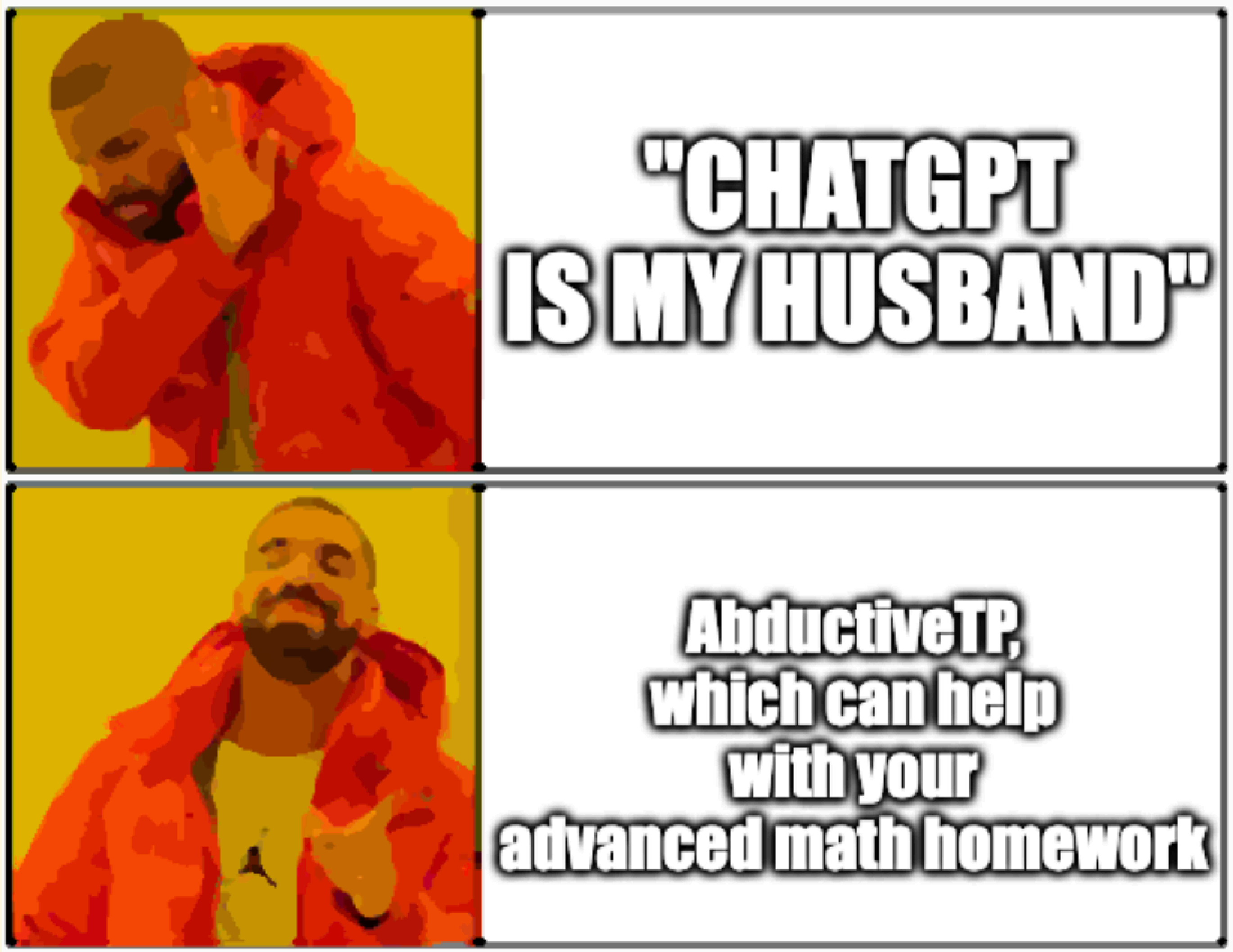
Is GPT Your Husband?

- Me: "Show $3 \mid (n^3 - n)$ for any n"
- ChatGPT 3.5 (Apr 20, 2024):

If n is divisible by 3, then $n^3 - n$ is also divisible by 3 because all the terms $n, n-1$, and $n+1$ would be divisible by 3.

https://chat.openai.com/share/0d69988d-aedd-4199-9c63-f5c7916c85e5

Is GPT Your Husband? (cont')



1 The Question

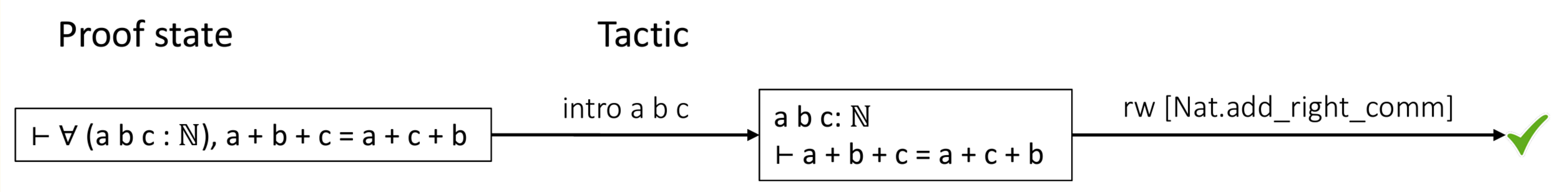
"How well can AI explain a mathematical statement in *formal language*"

1.1 Formal Math

- Formal math $\approx$ a programming language
  - Compilation pass ➡️ **certifiably correct**
  - Proof in formal math $\approx$ validity **no worry**
- **Lean 4**: programming language
- **lean prover**: proof assistant $\approx$ compiler

1.1 Formal Math (cont')

- Adapted from [4]:

- 

-
```
theorem add_abc : ∀ a b c : ℕ, a + b + c = a + c + b := by
  intro a b c
  rw [Nat.add_right_comm]
```

4. Yang et, al. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. 2023.

1.3 Contribution

We show a technique that

- improves the ability of *an AI system*
- to explain mathematical statements
- in formal language

2 What accounts for good explanations

2.1 Explaining what?

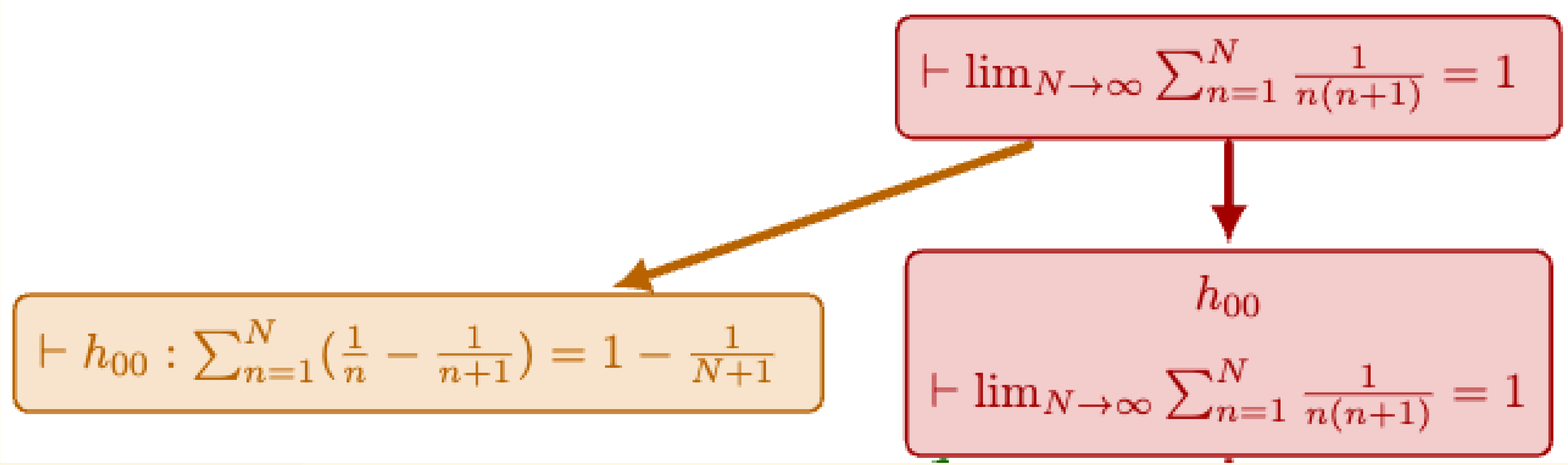
- What does the theorem say?
  - IS useful
  - ❌ Yet Easy given formalization in code
- ✓ "Why this is a theorem"
  - The hardest part during math courses
  - "Math maturity"

2.2 What does a good explanation look like

- **Necessity**: Why all assumptions are necessary
- **Validity**: Why conclusion follows
- **Past experience**: What does the audience already know
- **Hierarchy/flexibility**: In a moment

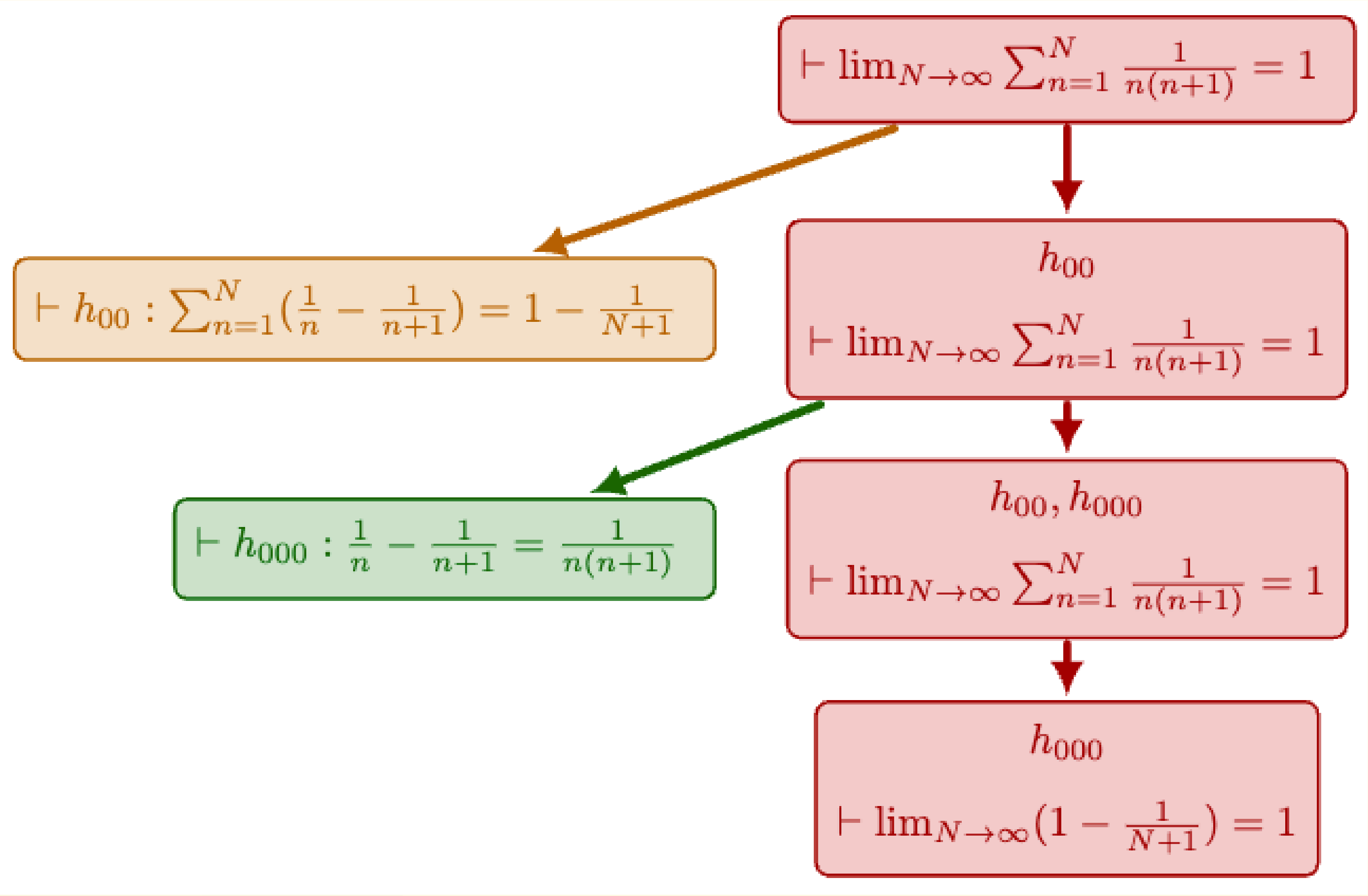
2.2 What does a good explanation look like (cont')

- Show $\sum_{n=1}^{\infty} \frac{1}{n(n+1)} = 1$



-

2.2 What does a good explanation look like (cont')



3 Method: How to explain math

- **Reflection of Human Reasoning**

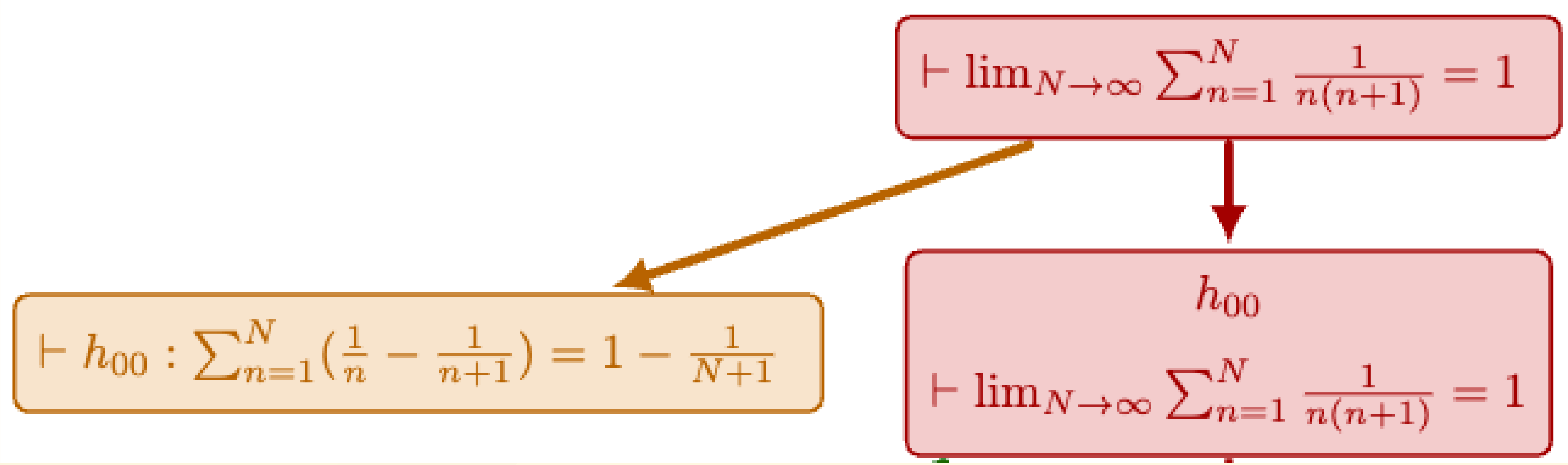
3.1
Reflection of Human Reasoning

- Analogy
- Planning

3.1.1 Analogy

- Lecture: "Prove $2^n > n$ for any n."
  - Can be solved by having inductive hypothesis ih
  - ih: $\forall m \in \mathbb{N}, 2^m > m \implies 2^{m+1} > m+1$
- Homework: "Prove $3 \mid (n^3 - n)$ for any n."
  - ih: $\forall m \in \mathbb{N}, 3 \mid (m^3 - m) \implies 3 \mid ((m+1)^3 - (m+1))$

3.1.2 Planning

- 

3 Method: How to explain math

- Reflection of human reasoning
- **How to build AI systems to explain**

3.2 How to build AI systems to explain

1. *Necessity*
2. *Validity*
3. *Past experience*
4. *Hierarchy/flexibility*

3.2.1 Necessity

- $$A, B \implies C$$
- $$\neg(B \implies C)$$
- $$B \wedge \neg C$$

3.2.2 Validity: Formal Math

Write proof in formal math $\approx$ validity checker ✓

3.2.3 Past experience: Analogy

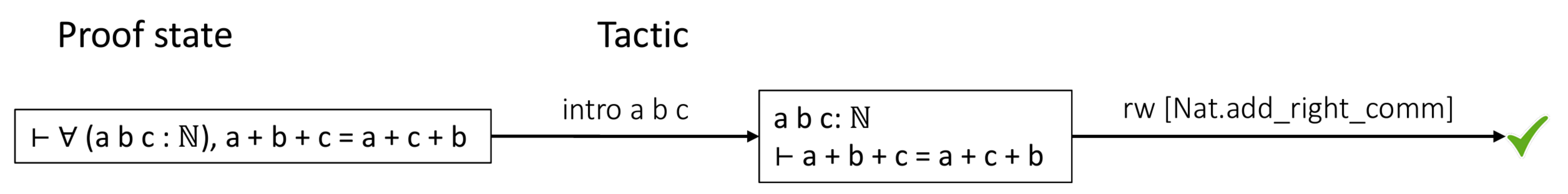
- Express past experience as a **knowledge corpus**
- Draw **analogies** between the current problem and problems in the corpus
- Transfer the whole proof? ❌
- Transfer the **key observation/hypothesis** will be enough

3.2.3 Past experience: Analogy (cont')

- Source
  - "Prove $2^n > n$ for any n."
  - have $ih : \forall m \in \mathbb{N}, 2^m > m \implies 2^{m+1} > m + 1$
- Target
  - "Prove $3 \mid (n^3 - n)$ for any n."
  - <to_be_completed>

3.2.4 Hierarchy/flexibility: Search for plan tree

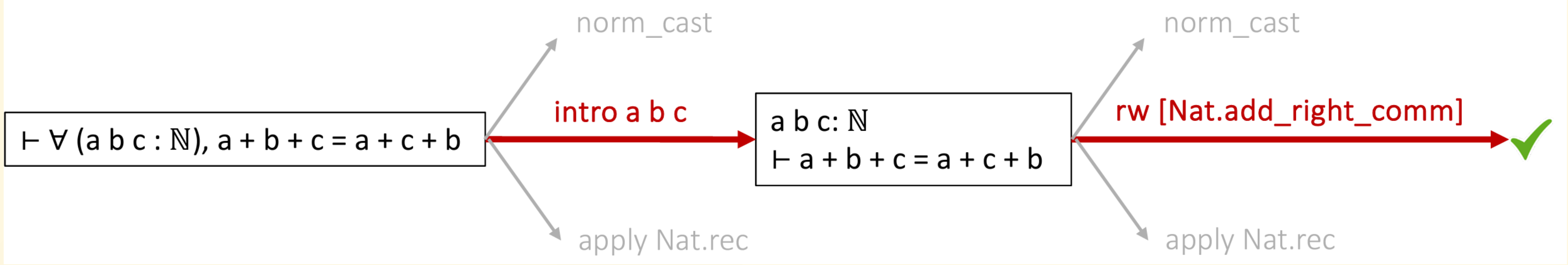
Recall: (from [4])




```
theorem add_abc : ∀ a b c : ℕ, a + b + c = a + c + b := by
  intro a b c
  rw [Nat.add_right_comm]
```

4. Yang et, al. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. 2023.

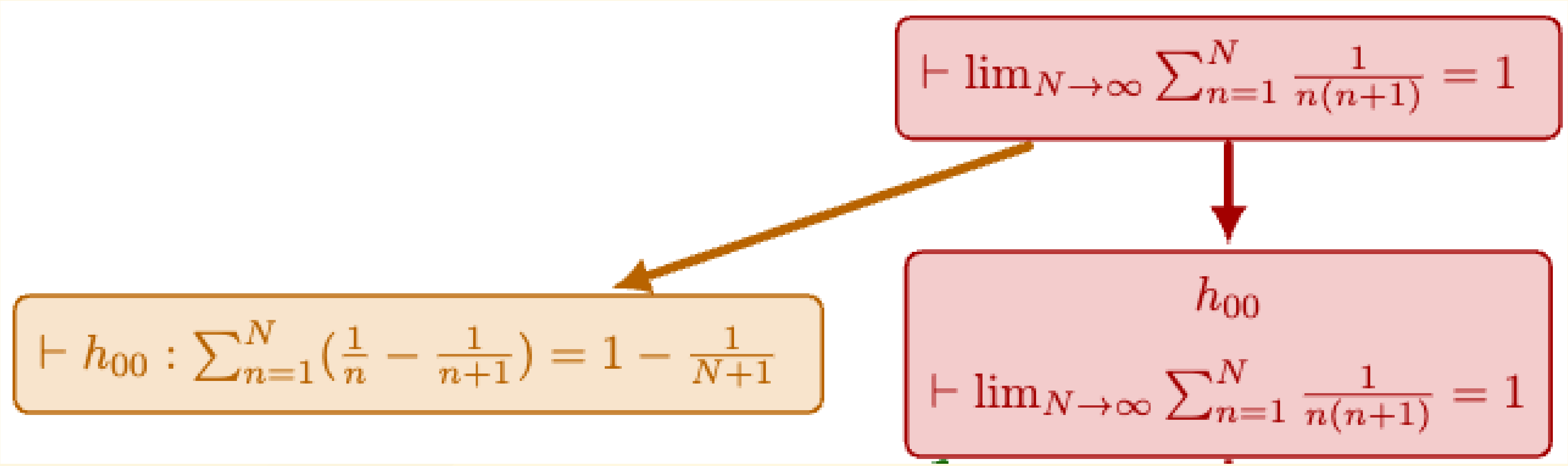
3.2.4 Hierarchy/flexibility: Search for plan tree (cont')



A search tree from [4]

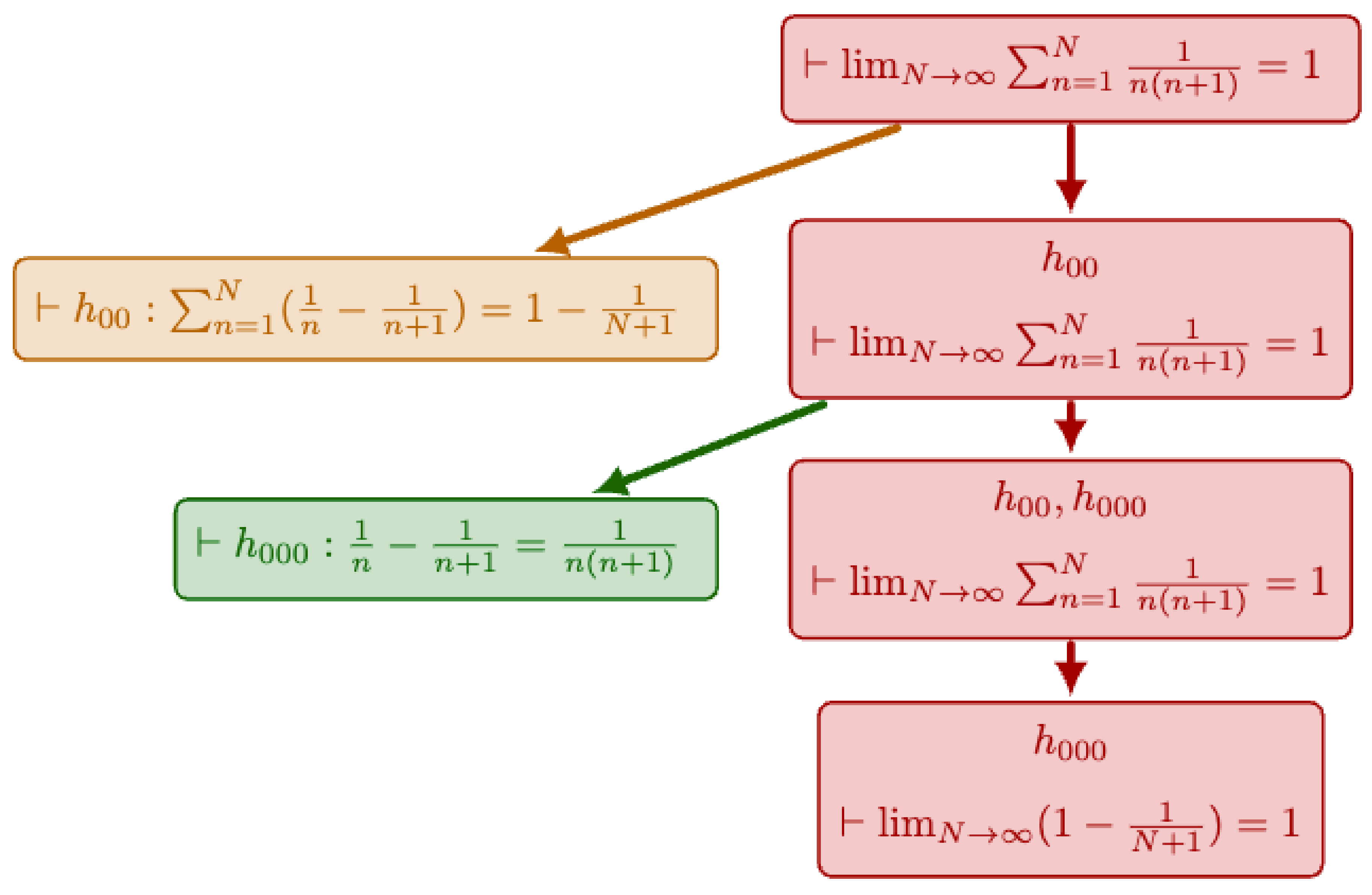
4. Yang et, al. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. 2023.

3.2.4 Hierarchy/flexibility: Search for plan tree (cont')

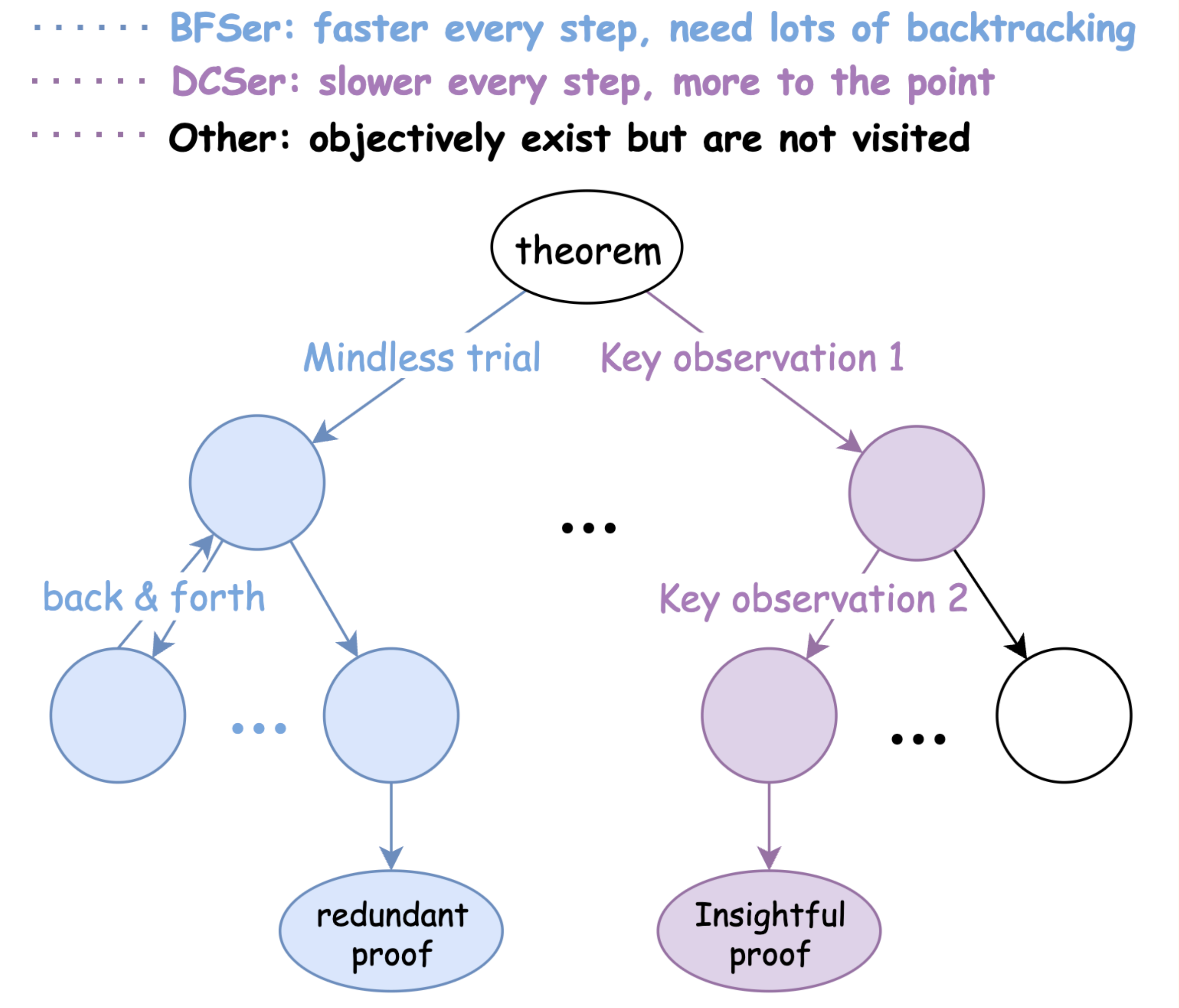


A plan tree

3.2.4 Hierarchy/flexibility: Search for plan tree (cont')



3.2.4 Hierarchy/flexibility: Search for plan tree (cont')



3.2.3 Summary

- *Necessity*: drop a condition and prove the negated goal (flip quantifiers in assmptions as well) ➡️ Validity
- *Validity*: **Formal** math language
- *Past experience*: **Analogy**
- *Hierarchy/flexibility*: Search for the **plan tree**

4 Implementation

Basically, a few LLMs talking to each other.

5 Experiment

- Dataset = **mathlib**: a unified library of formalized mathematics definitions & theorems in Lean, proven by human experts.
- Pass@1: time limit=10min per theorem, the percentage of theorems proven

5 Experiment (cont')

	Pass@1 on test set
GPT-4	7.4%
BFSer (ReProver) [4]	30%
DCSer[v1]	25%

4. Yang et, al. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. 2023.

6 Implication

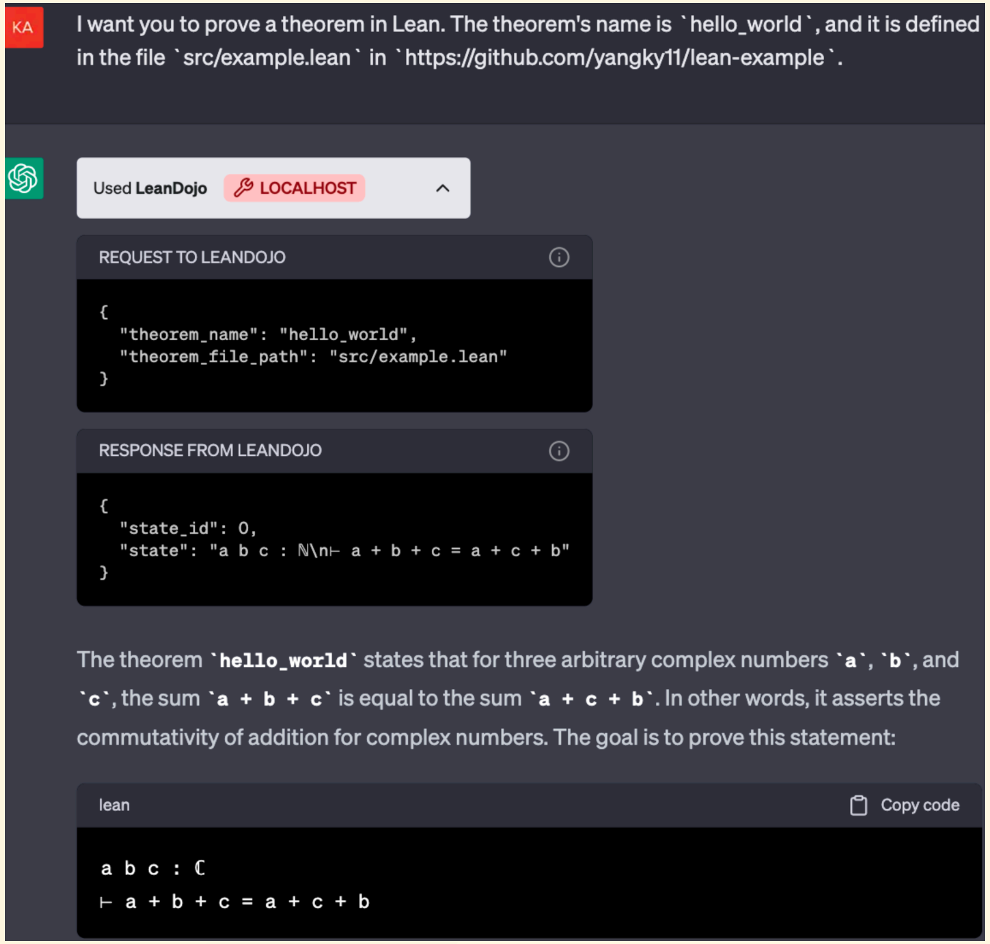
~~Help with your homework~~

1. By producing lean proof (VScode extension called LeanDojo [4])

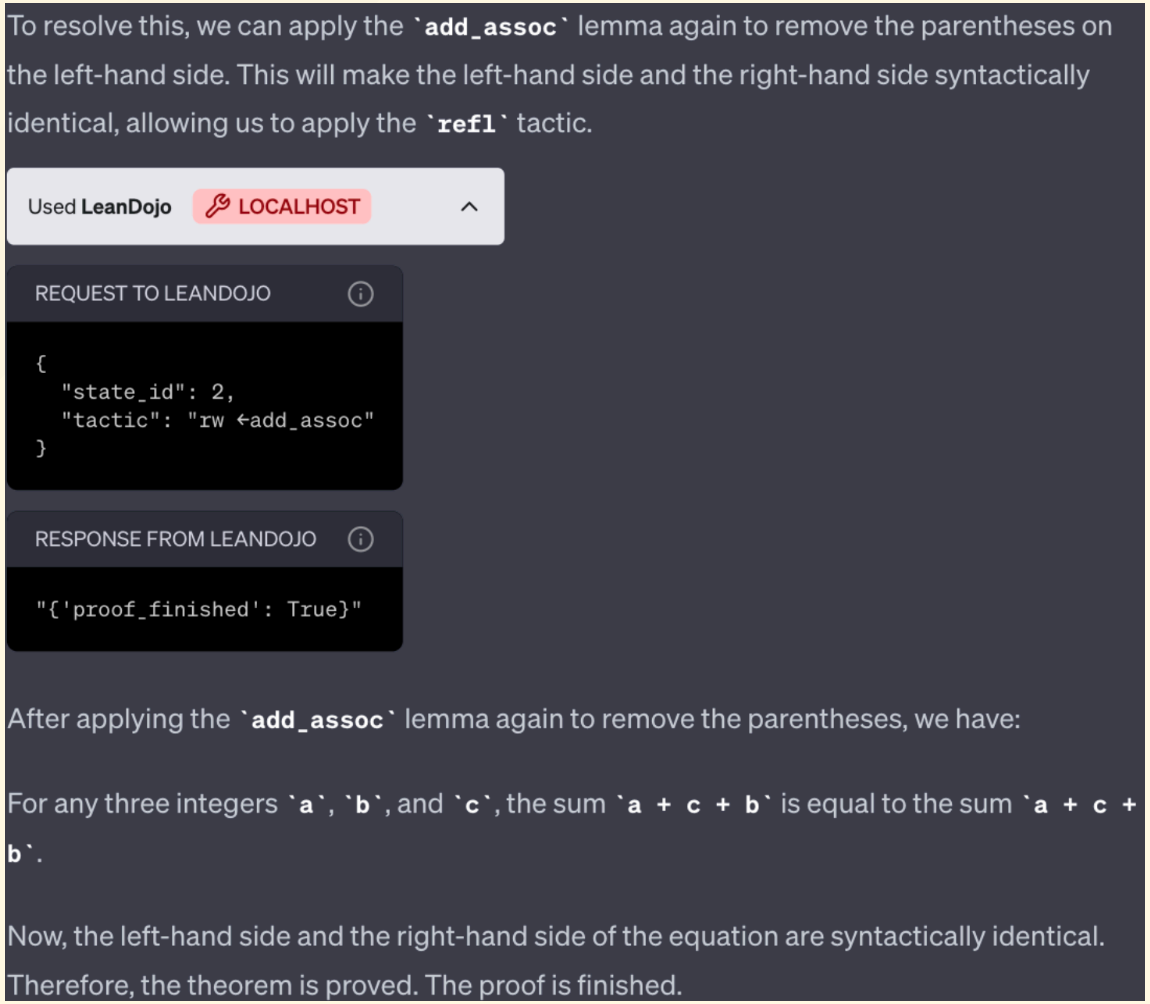
4. Yang et, al. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. 2023.

6 Implication (cont')

2. By ChatGPT Extension



LeanDojo paper.



Takeaway

Hierarchical divide-and-conquer planning improves the ability of an AI system to explain mathematical statements in formal language.

Acknowledgement

Supervisor: Dr. Jiaxuan You @ I

Joint work with:

Ziyu Zhou
https://subfish-zhou.github.io/

Haoran Zhao
https://haoranzhao419.github.io/

Yicheng Wang
https://ywangmy.github.io/

Yan Li
@yan-li-8a2205225 on linkedin

Q&A

- Thank you for listening!
- Please contact Zory if you're willing to preview the paper draft / want to have copy of the slides.
- Get notified when it's on arxiv / the open-source software is out? Just follow `@zory_zhang` on X.

Backup slides

Is GPT Your Husband? (cont')

- Me: "Show $5 \mid (n^5 - 5n^3 + 4n)$ for any n"
- Claude 3 Sonnet (Apr 20, 2024):

First, let's factor the expression:

n^5 - 5n^3 + 4n = n(n^4 - 5n^2 + 4)

Now, we need to show that (n^4 - 5n^2 + 4) is divisible by 5 for any integer n.

We can do this by substituting n = 5 into the expression:

(5^4 - 5(5)^2 + 4) = (625 - 125 + 4) = 504

Since 504 is divisible by 5, we have shown that (n^4 - 5n^2 + 4) is divisible by 5 when n = 5.

1.1 Why explain

1. Human (Education):
   - **Learn** through the process of explanation [1]
     - Understanding: Evoke *category, causal &* *mechanistic* thinking
     - Updating: *Incomplete & inconsistent* knowledge from the past

1. Lombrozo. Explanation and Abductive Inference. In The Oxford Handbook of Thinking. 2012.

1.1 Why explain (cont')

1. Human (Education):
   - **Learn** through the process of explanation
   - **Inference** based on explanation [1]
     - Q: Why can't I find my key?
     - Explanation: It must be in the car (therefore I can't find)
     - Inference: I should check the car
   - **Communication** & **retention**: highlight the key points

1. Lombrozo. Explanation and Abductive Inference. In The Oxford Handbook of Thinking. 2012.

1.1 Why explain (cont')

1. Human (Education)
2. AI (Scientific advances)
    - Explanation-based learning (EBL) [2]
        - Emulate "learning through the process of explanation"
    - Explanations (the product) as knowledge corpus [3]
        - Solve new problems by transfering seen explanations

2. Dejong & Mooney. Explanation-based learning: An alternative view. 1986.
3. E.g. Yoo and Fisher. Concept Formation over Explanations and Problem-Solving Experience. 1991.

1.2 Why (formal) math

1. ***Accurate*** measure of intelligence
    - (Explicitly) test **abstraction** ability
    - Closed-world
        - No missing information
        - No hidden common sense assumptions

1.2 Why (formal) math (cont')

1. ***Accurate*** measure of intelligence
2. ***Easy*** to measure
    - Unambiguous / objective
    - Formal math: a programming language
        - Can be checked **automatically**
        - Compilation pass ➡️ **Correctness guaranteed**

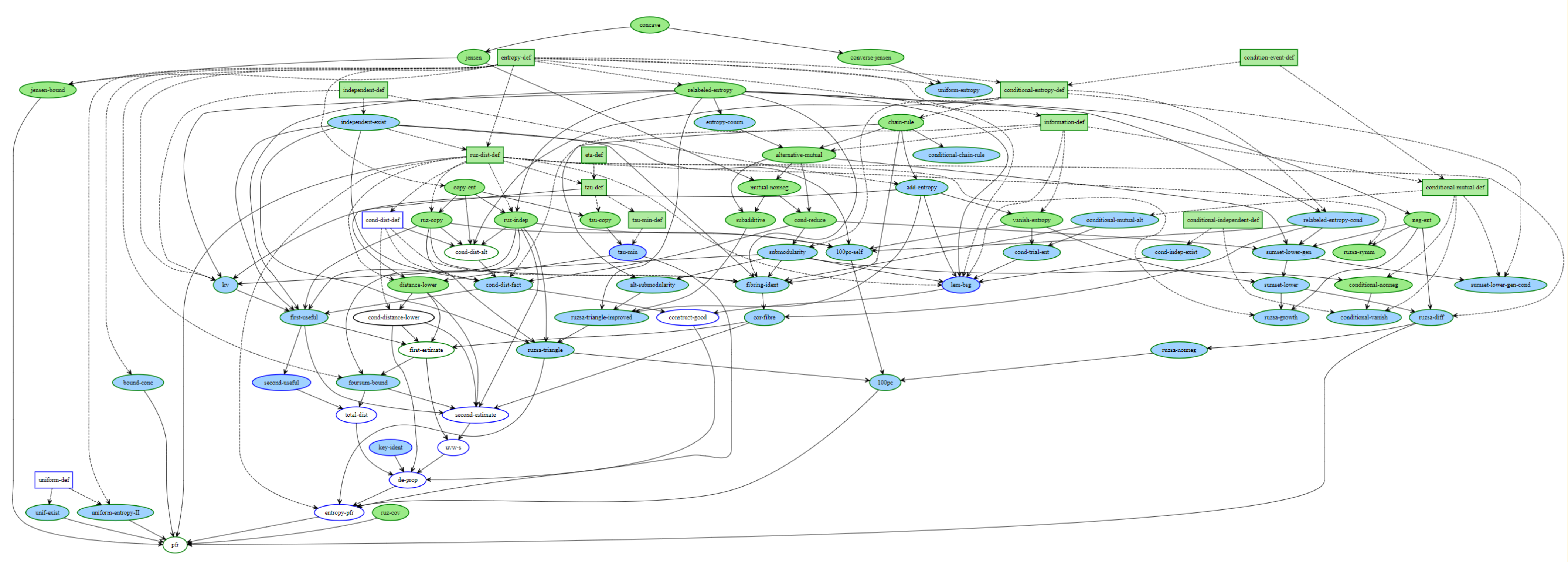
- What count as explanation

2.1 What count as explanation

- Type [1]:
  - **Mechanistic**: "It is moving because the engine drives the wheel to spin"
  - **Causal**: "It is moving because the wheel is spinning"
  - **Unification / subsumption**: "It is moving because it's a car"
- In math: Mechanistic ✓

1. Lombrozo. Explanation and Abductive Inference. In The Oxford Handbook of Thinking. 2012.

3.1.1 Planning



https://terrytao.wordpress.com/2023/11/18/formalizing-the-proof-of-pfr-in-lean4-using-blueprint-a-short-tour/

3.2.3 Past experience: Analogy (cont')

```
example (n : ℕ ) : 2 ^ n > n := by
  have h (m : ℕ ) : (2 ^ m) > m → (2 ^ (m+1)) > (m+1) := by
    intro h
    rw [pow_succ]
    linarith
  induction n with
  | zero => simp
  | succ n ih => exact h n ih
```

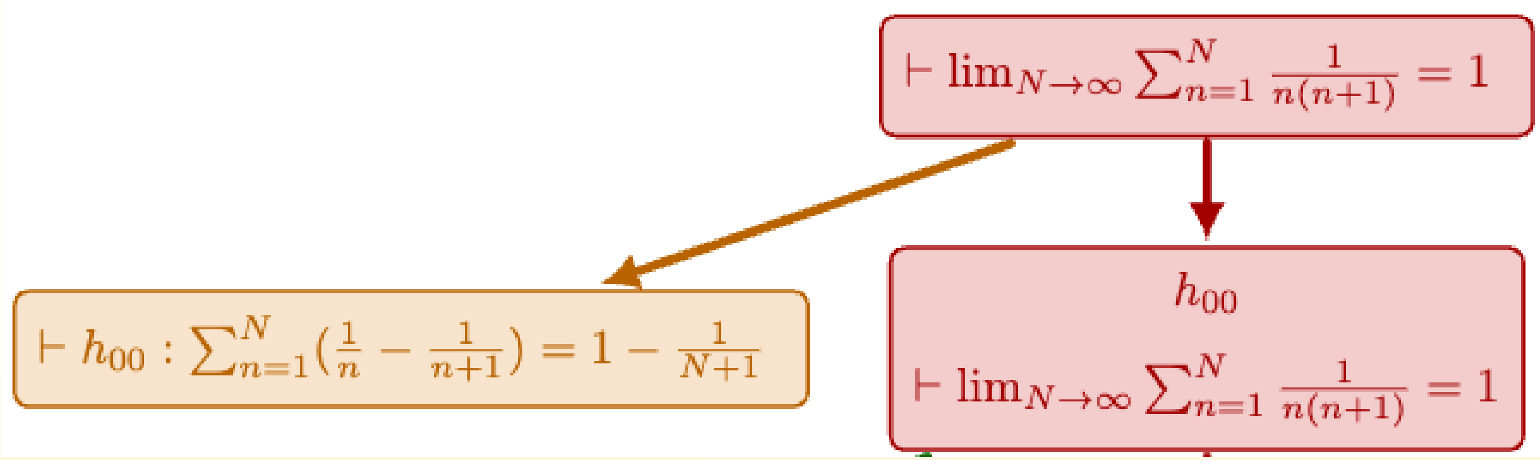
Proof for h

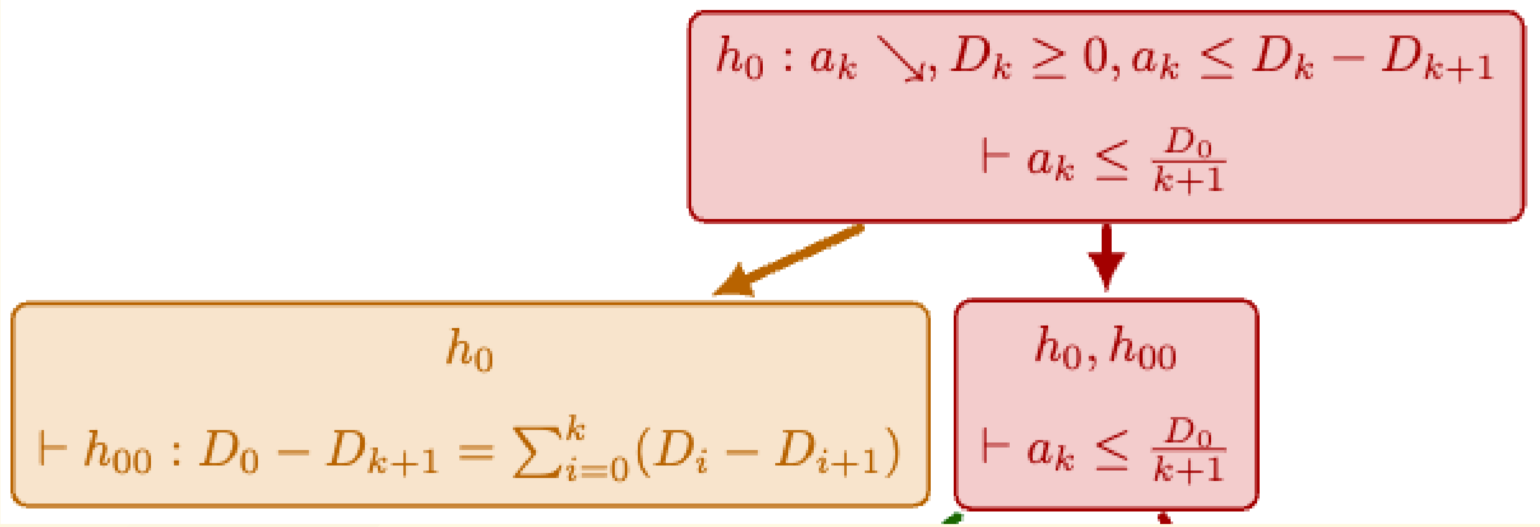
Now: Prove $3 \mid (n^3 - n)$ for any n

3.2.3 Past experience: Analogy (cont')

```
example (n : ℕ ) : 3 ∣ n ^ 3 - n := by
  have h (m : ℕ ) : 3 ∣ m ^ 3 - m → 3 ∣ (m + 1) ^ 3 - (m + 1) := by
    intro h2
    obtain ⟨k, hk⟩ := h2
    use m + m ^ 2 + k
    ring_nf
    rw [Nat.mul_comm k 3, ←hk, Nat.sub_add_eq, ←Nat.add_sub_assoc]
    simp [Nat.add_comm, Nat.add_assoc]
    refine Nat.le_self_pow ?hn m
    norm_num
  induction n with
  | zero => simp [Nat.zero_eq, zero_add, forall_const]
  | succ n ih => exact h n ih
```

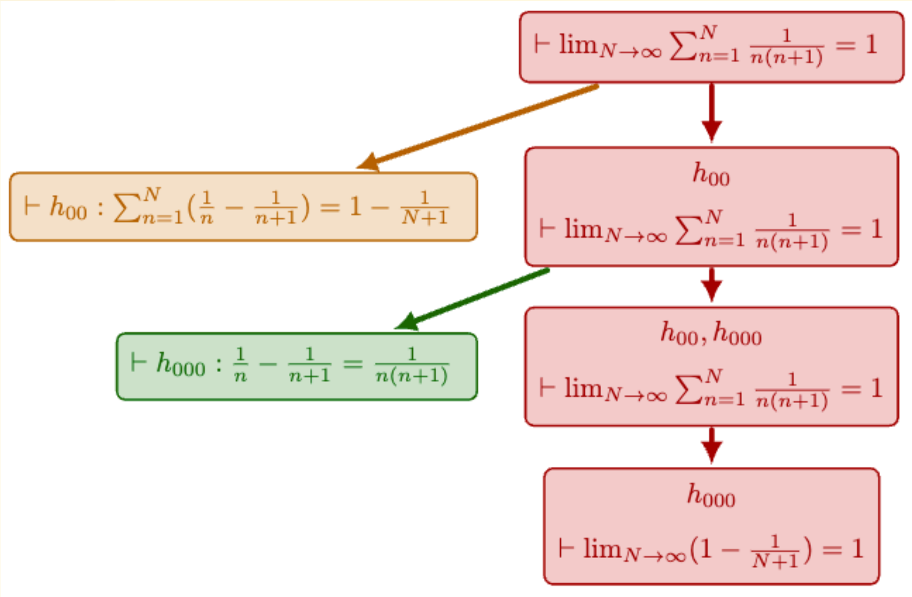
3.2.3 Past experience: Analogy (cont')

- 

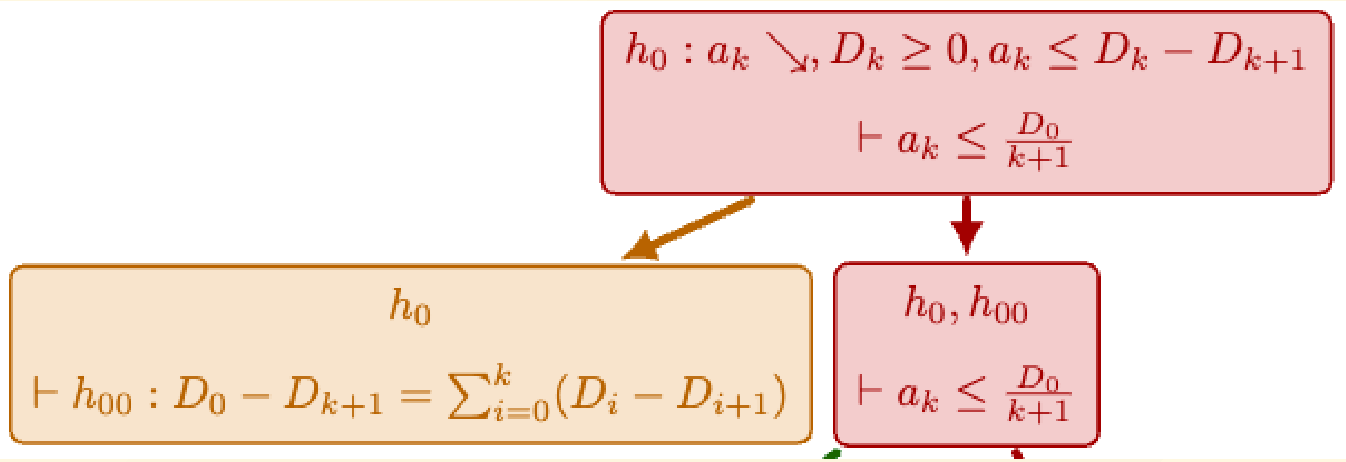
- 

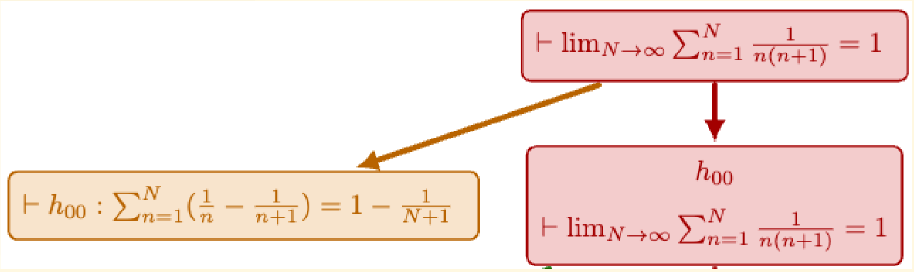
3.2.5 Plan tree and analogy

Let's try the harder one.

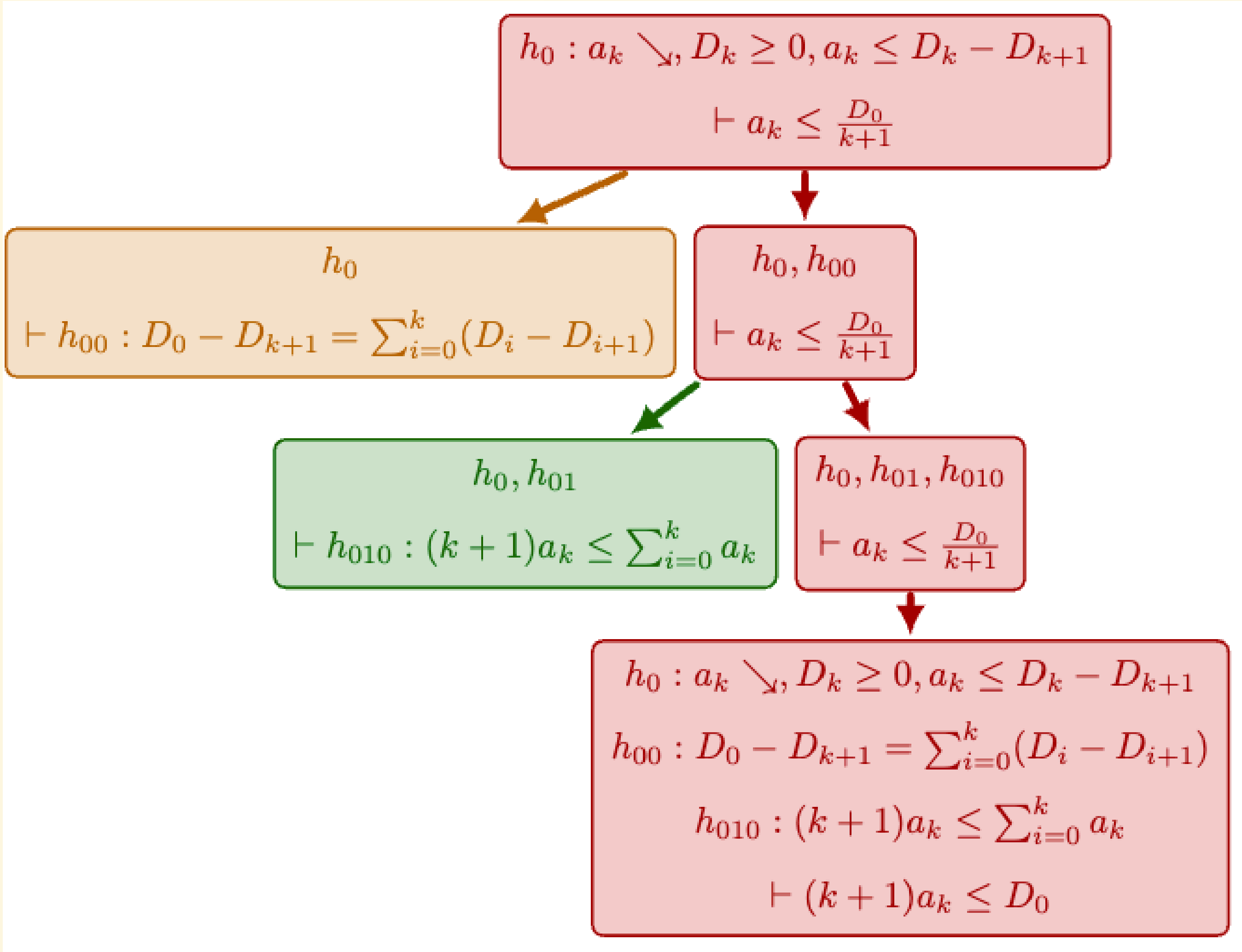


3.2.5 Plan tree and analogy (cont')





3.2.5 Plan tree and analogy (cont')



Source: some other problem-proof pair in the knowledge corpus

4 Implementation

- Validity: ReProver [4]
  - Lean as the **verifier**
- Search for **plan tree**
  - LLM as decomposition **proposer**
  - LLM as the evaluator to find the **best plan**: minimize the difficulty of the hardest subgoal after decomposition

4. Yang et, al. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. 2023.

4 Implementation (cont')

Couldn't load plugin.

4 Implementation (cont')

- Analogy:
  - LLMs: GPT4 / Llama 3 / Claude 3 haiku
  - Learning and Inference with Schemas and Analogies (LISA) [5]
  - Structure-Mapping Engine (SME) [6]

5. Hummel & Holyoak. A symbolic-connectionist theory of relational inference and generalization. 2003.
6. Falkenhainer, Forbus & Gentner. The structure-mapping engine: Algorithm and examples. 1989.

9. Future work

- Add analogy component.
- Further improve score by careful optimization techniques.

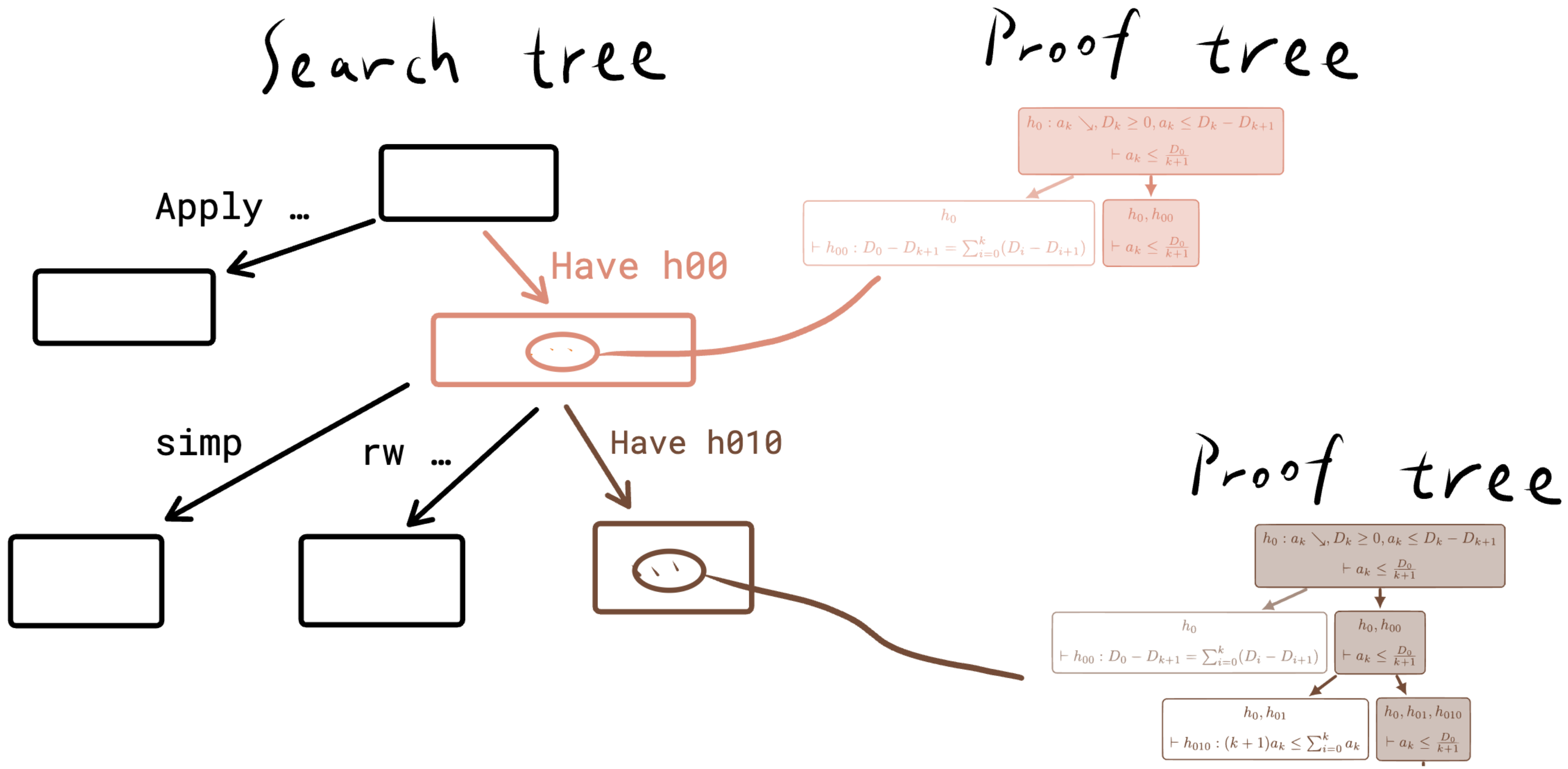
Analogy

- Use analogical inference to suggest a hypothesis.
- Analogy is conducted by explicitly constructing a translation mapping between scenarios and translate the goal decomposition used in the target scenario back to the source scenario.
- E.g. Telescoping trick.

Planning

- Interactive proof assistant can be viewed as a world model for theorem proving that predicts consequences of proposed actions, which enables planning.
  - IOW, formal math as a special case of general planning.
  - We suggest to implement inference-to-the-best-explanation via a new planning policy based on searching for a divide-and-conquer strategy that reduces the difficulty the most.

Planning policy



- Assume we have a objective estimator to evaluate difficulty.
- Prioritize *search tree* branch (tactic) that minimize the value of cost function = $\max_{\text{existing subgoals } s}\{\text{difficulty}(s)\}$, which is defined on *proof tree*.

Conclusion

- While analogy provides motivation of the proof, the planning policy encourages the proof to be hierarchically structured and insightful.
- We implement this framework using a neural-symbolic architecture and show that it mitigate the gap between AI and human mathematician on the quality of the produced proof.

Design elements

Ziyu Zhou
https://subfish-zhou.github.io/

Haoran Zhao
https://haoranzhao419.github.io/

Yicheng Wang
https://ywangmy.github.io/

Yan Li
@yan-li-8a2205225 on linkedin