Proving Mathematical Theorems Using Neural Networks

Wenda Li

University of Cambridge

wl302@cam.ac.uk

February 11, 2022

Overview

- My experience on applying machine learning to theorem proving.
- Challenges and opportunities in this domain.

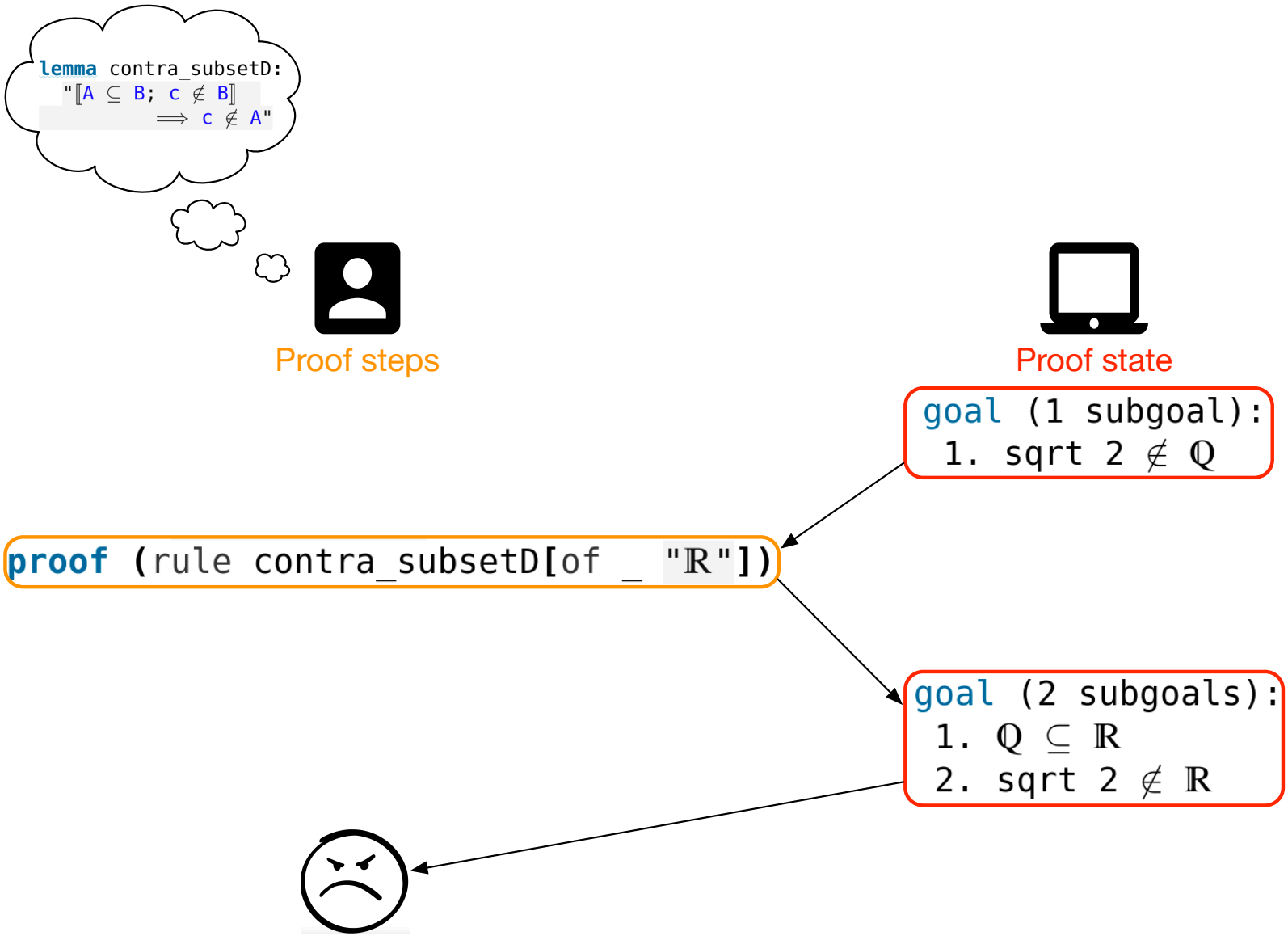
Overview

- What are proof assistants? And what are the challenges?
- Investigating neural seq2seq models' capability of conjecturing the intermediate steps of proofs.
- Improving neural models' capability of doing mathematical reasoning via pretraining.
- Proving with language models (in progress).
- Summary and future work.

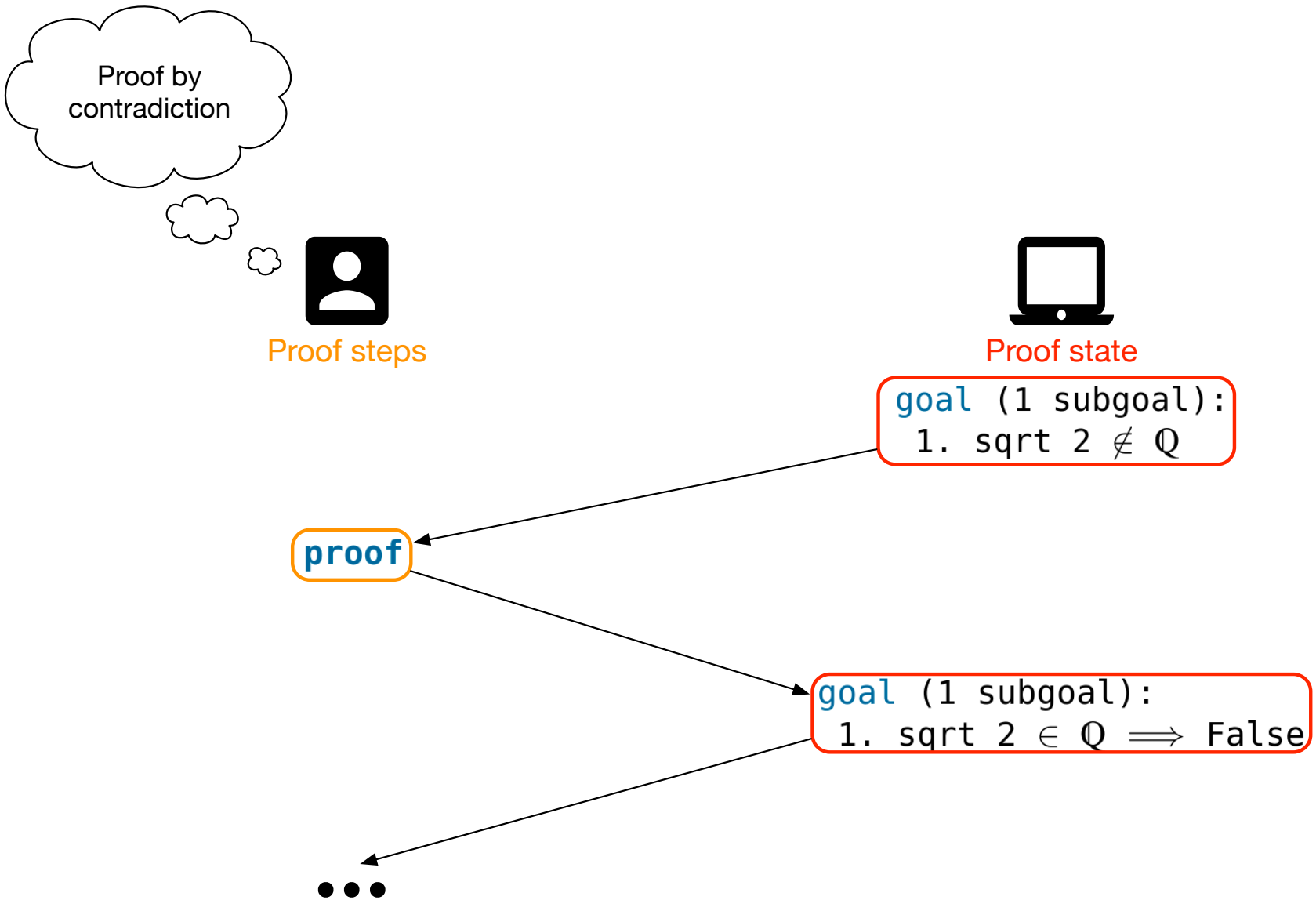
What is a proof assistant

- A framework that allows inferences/derivations being checked by computers, so that everything we derive in a proof assistant is **sound** provided the handful axioms we assumed are consistent.
- The logic (e.g. higher-order logic) is **expressive** enough to encode almost all programming and math problems.
- **Human guidance** may be needed, but state-of-the-art automated theorem provers have been incorporated to reduce the labor.
- Widely applied to **verify** the correctness of hardware, security protocols, safety critical software, and mathematical proofs.

Proving theorems by interacting with a proof assistant



Proving theorems by interacting with a proof assistant



Success stories of proof assistants

- seL4: a verified high-performance microkernel by Isabelle/HOL.
- CompCert: an optimised C compiler certified by Coq.
- Proof of the odd order theorem in Coq.
- Proof of the Kepler conjecture in HOL Light and Isabelle/HOL.
- Modelling and verification of aircraft trajectories in PVS

NEWS | 18 June 2021

Mathematicians welcome computer-assisted proof in 'grand unification' theory

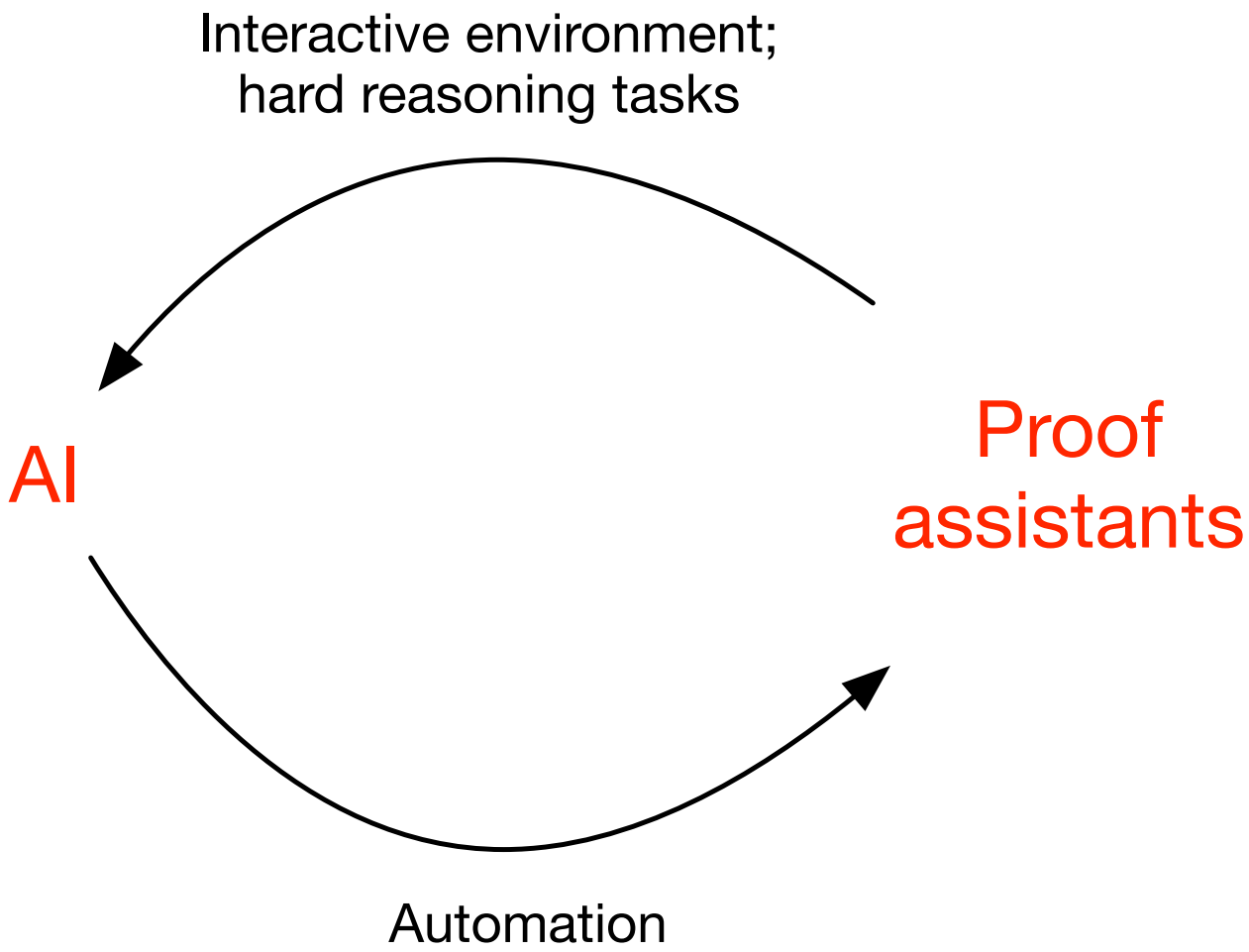
Proof-assistant software handles an abstract concept at the cutting edge of research, revealing a bigger role for software in mathematics.

Why not mechanise more often?

The mechanisation process is still labor intensive:

- seL4 project in Isabelle/HOL: 20 person-years for 8700 lines of C;
- Odd Order theorem in Coq: 250 pages of mathematical proofs, 15 experts, 6 years.

Opportunities for AI researchers



Why conjecture?

Cut-free proofs may need to be superexponentially larger than proofs that contain cut.

Can neural networks conjecture?

IsarStep: a Benchmark for High-level Mathematical Reasoning

- published at ICLR 2021.
- joint work with Lei Yu[1], Yuhuai Wu[2], and Larry Paulson[3].

[1] DeepMind
[2] University of Toronto
[3] University of Cambridge

Proof of the irrationality of $\sqrt{2}$.

Assume $\sqrt{2}$ is rational. Then there exists a pair of coprime integers a and b such that $\sqrt{2} = a/b$, and it follows that $2 = a^2/b^2$ and then $2b^2 = a^2$. Hence a is even. Thus there exists an integer c such that $a = 2c$, which combined with $2b^2 = a^2$ yields $2c^2 = b^2$: hence b is also even. So a and b are both even although they are coprime, contradiction. $\square$

Proof of the irrationality of $\sqrt{2}$.

Assume $\sqrt{2}$ is rational. Then there exists a pair of coprime integers a and b such that $\sqrt{2} = a/b$, and it follows that $2 = a^2/b^2$ and then $2b^2 = a^2$. Hence a is even. Thus there exists an integer c such that $a = 2c$, which combined with $2b^2 = a^2$ yields $2c^2 = b^2$: hence b is also even. So a and b are both even although they are coprime, contradiction. □

An informal (high-level) proof is, mostly, a sequence of intermediate propositions connected with logical relations (e.g. then, thus, yield).

Proving the irrationality of $\sqrt{2}$ in Isabelle/HOL

Proof of the irrationality of $\sqrt{2}$.

Assume $\sqrt{2}$ is rational.

Then there exists a pair of coprime integers a and b such that $\sqrt{2} = a/b$,

and it follows that $2 = a^2/b^2$ and then $2b^2 = a^2$.
Hence a is even.

Thus there exists an integer c such that $a = 2c$, which combined with $2b^2 = a^2$ yields $2c^2 = b^2$: hence b is also even. So a and b are both even although they are coprime, contradiction. □

```
theorem "sqrt 2 ∉ ℚ"
proof
  assume "sqrt 2 ∈ ℚ"
  then obtain a b :: int where
      "sqrt 2 = a/b" "coprime a b"
    \<proof>
  then have "2 = a² / b²" \<proof>
  then have *:"2*b² = a²"
    \<proof>
  then have "even a" \<proof>
  then obtain c::int where "a=2*c"
    \<proof>
  with * have "b² = 2*c²" \<proof>
  then have "even b" \<proof>
  with ‹even a› ‹coprime a b›
  show False \<proof>
qed
```

Bridging the gap

Part of the derivation in the previous example is as follows:

$$\underbrace{2b^2 = a^2}_{(1)} \Rightarrow \underbrace{a \text{ is even}}_{(2)} \Rightarrow \underbrace{\exists c \in \mathbb{Z}.\, a = 2c}_{(3)}.$$

where the proposition (2) bridges the gap between (1) and (3).

In IsarStep, we want to synthesise (2) given (1) and (3).

A conditional text generation task

$$\underbrace{2b^2 = a^2}_{(1)} \Rightarrow \underbrace{a \text{ is even}}_{(2)} \Rightarrow \underbrace{\exists c \in \mathbb{Z}.\, a = 2c}_{(3)}.$$

where the proposition (2) bridges the gap between (1) and (3).

Source:

- a set of propositions to derive the target, (e.g. $2b^2 = a^2$);
- a proposition derived from the target, (e.g. $\exists c \in \mathbb{Z}.\, a = 2c$);
- is a set of other propositions to derive the target;
- a set of lemmas to justify the target (optional).

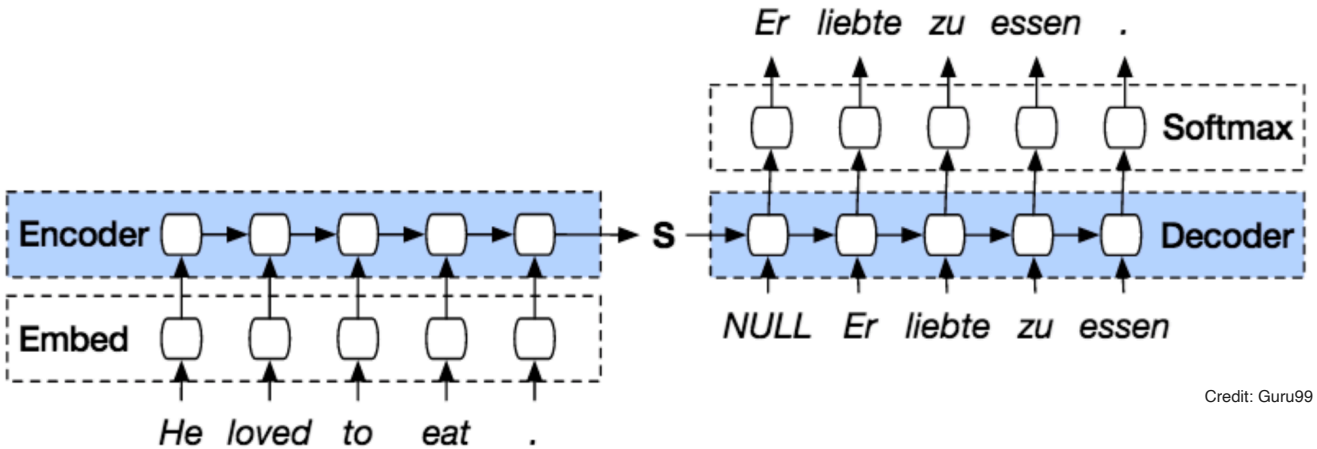
Target: A proposition (e.g. $a \text{ is even}$).

About seq2seq models

A seq2seq model aims to model the following conditional probability:

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{j=1}^{J} p(y_j \mid \mathbf{y}_1^{j-1}, \mathbf{x})$$

meaning the probability of generating the output sequence **y** given **x**.



Dataset statistics

Dataset mined from the Archieve of Formal Proofs and the standard library of Isabelle/HOL:

- 1.2 million raw data points from 204K lemmas.
- Dataset split: 820K/5K/5K for train/valid/test.
- Lengths:
  - Average input length: 310;
  - Average output length: 43.
- Vocabulary size: 30K.

Models

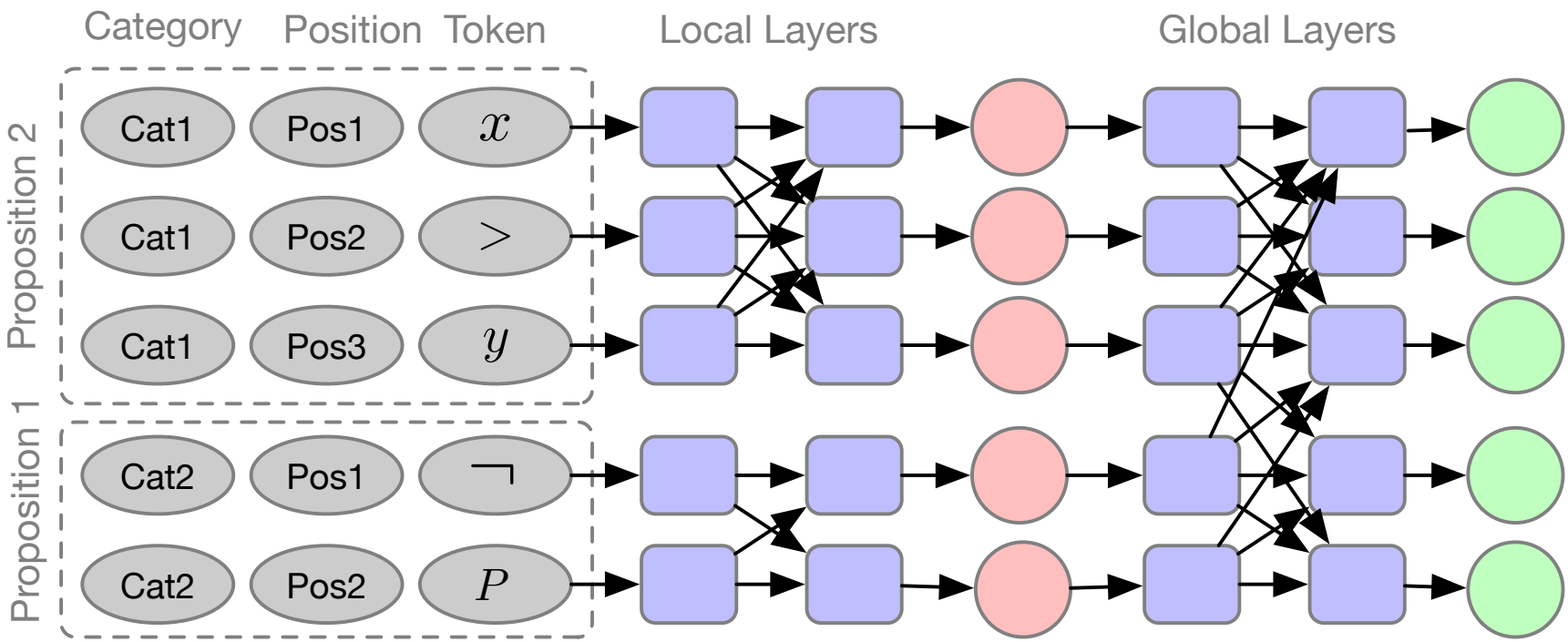
Baseline models:

- LSTM with attention (RNNSearch).
- Transformer base.

Our proposed model:

- Hierarchical Transformer (HAT).

Hierarchical Transformer (HAT)



Evaluation methods

- Exact match at string form.
  - **BLEU score**: calculate the statistics of n-grams overlaps between the system output and reference sequences, ranging from 0 to 100. The higher the better.
  - **Top-1 accuracy**: percentage of the best output sequences that exactly match the references.
  - **Top-10 accuracy**: percentage of references appearing in the top 10 generated sequences.
- **Correctness**:
  - A test suite to bring the generated propositions back to the Isabelle environment and check their correctness using automatic theorem provers (ATPs).
  - A proposition is **correct** if it bridges the gap in the proving environment – it can be different from the ground truth.

Results

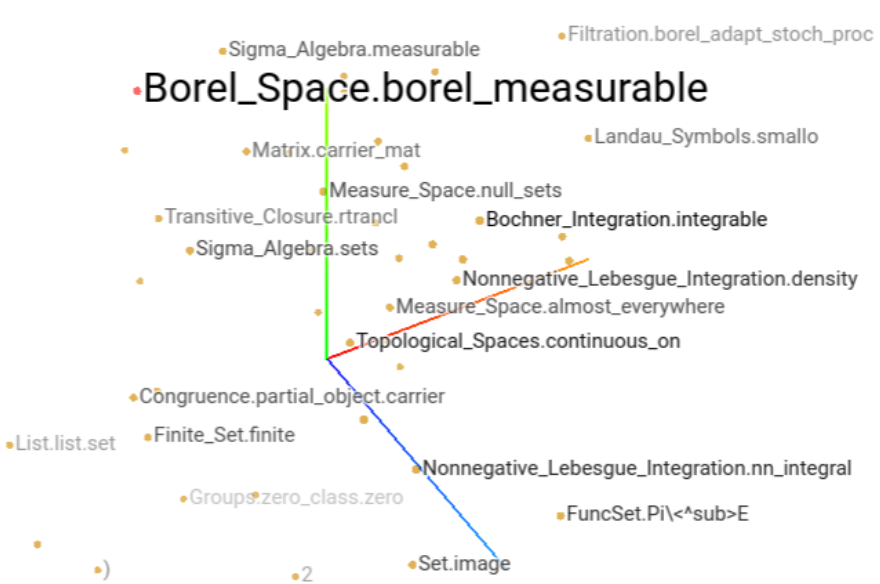
Model	Top-1 Acc.	% Correctness	Top-10 Acc.	BLEU
RNNSearch	13.0	-	26.2	42.3
Transformer	20.4	25.2	33.1	59.6
HAT	**22.8**	**27.6**	**35.2**	**61.8**

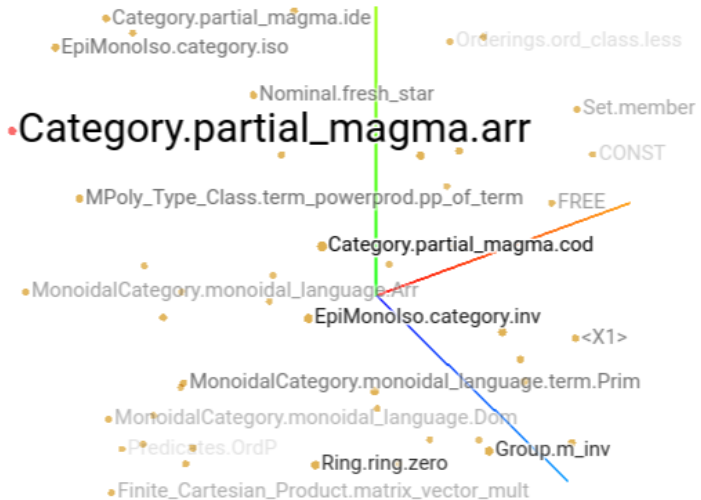
% Correctness: percentage of **alternative valid propositions** + propositions that match the references at surface form.

Findings

- Neural models can learn the syntax of propositions correctly.
- The learned token embeddings are comprehensive: related mathematical concepts are close in cosine space.
- Models can learn non-trivial mathematical reasoning and even more complicated multi-step reasoning.

Embeddings of related mathematical concepts are close





Analysis: derivation by invoking lemmas

Source: $? \wedge S_1 \subseteq S_2 \Rightarrow a \in S_2$

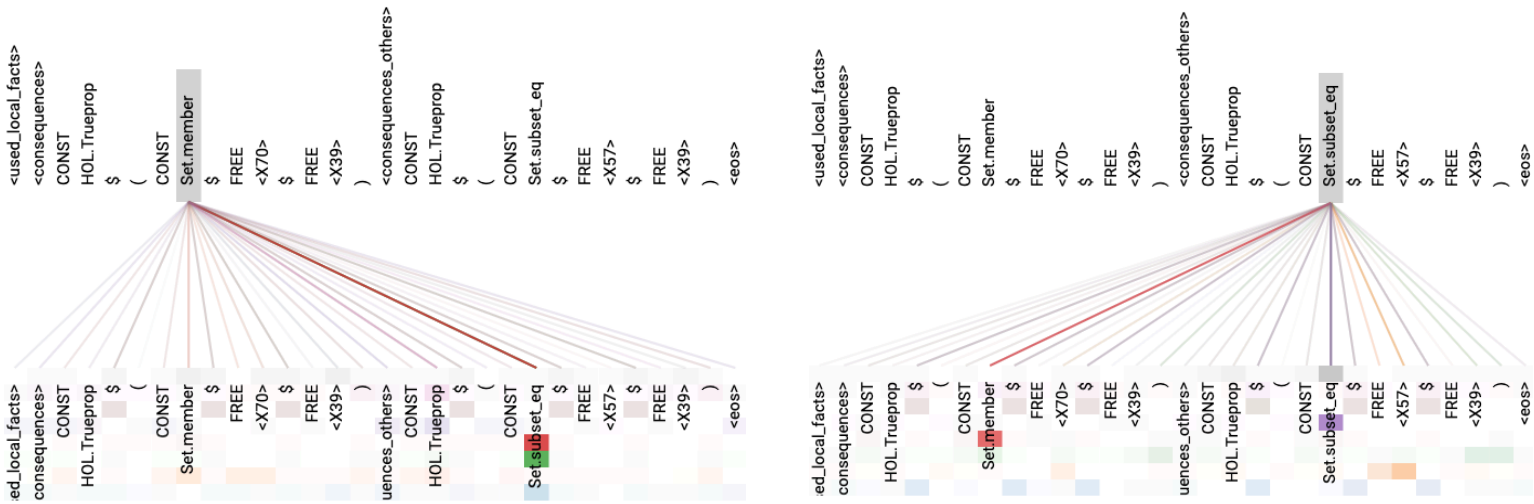
The model gives: $a \in S_1$

The derivation involves the following lemma:

$$x \in A, A \subseteq B \vdash x \in B.$$

Important tokens are likely to attend to each others

The lemma $x \in A, A \subseteq B \vdash x \in B$ relates $\in$ and $\subseteq$, the tokens of which tend to each other:



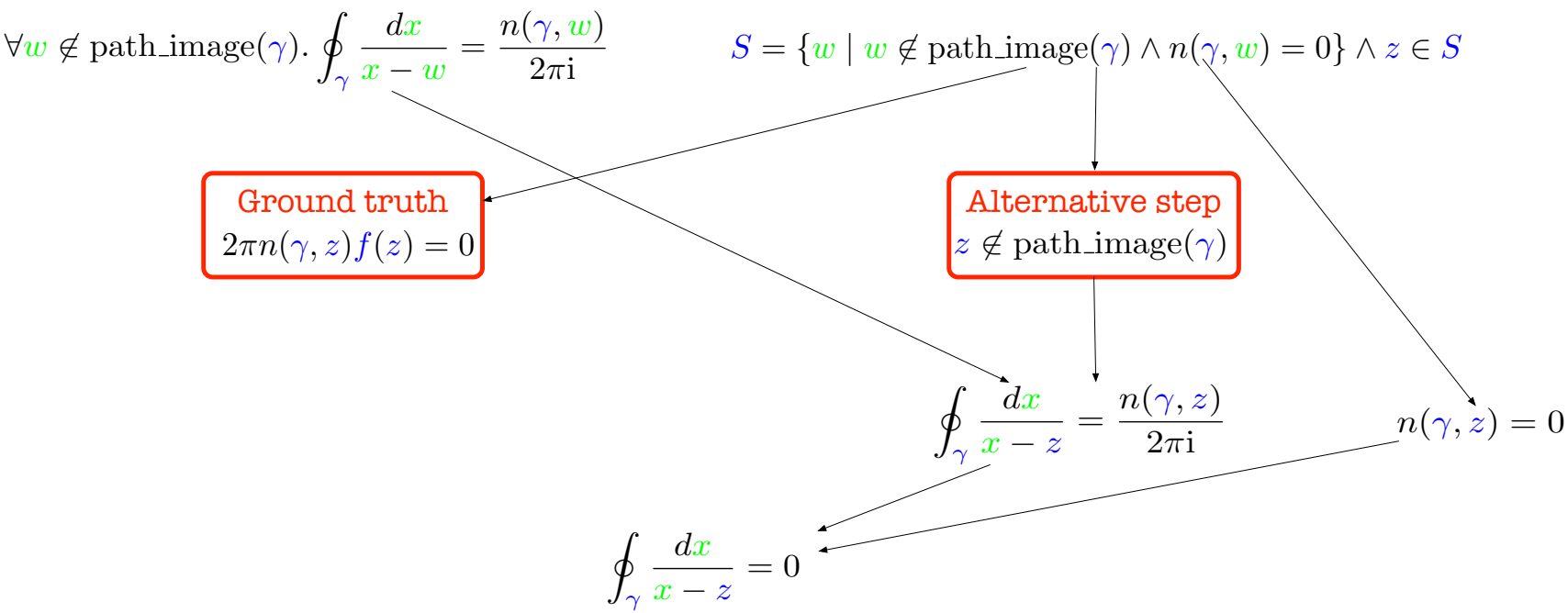
Robustness

Source: $? \wedge \cancel{S_1 \subseteq S_2}\ S_2 \subseteq S_1 \Rightarrow a \in S_2$

The model gives: $\cancel{a \in S_1}\ a \in S_2$

The previous lemma $x \in A, A \subseteq B \vdash x \in B$ no longer applies – to prove $a \in S_2$, we can only discharge itself.

An alternative step is much better than the ground truth



Take-home message of IsarStep

- Neural models are able to learn to conjecture intermediate steps of formal proofs.
- However, they are far from solving theorem proving – only generate 22.8% exact matches and 27.6% correct propositions on our test set.

Can pretraining help conjecturing?

LIME: Learning Inductive Bias for Primitives of Mathematical Reasoning

- published at ICML 2021
- joint work with Yuhuai Wu[1] [2], Markus Rabe[1], Jimmy Ba[3], Roger Grosse[3], and Christian Szegedy[1]

[1] Google Brain
[2] Stanford University
[3] University of Toronto

Charles Peirce: Deduction, Abduction, and Induction

Reasoning Primitive	Inference Map
Deduction	Rule, Case → Result
Abduction	Rule, Result → Case
Induction	Case, Result → Rule

Deduction

Rule: *All* the beans in this bag are white.
Case: These beans are from this bag.
Result: These beans are white.

Abduction

Rule: *All* the beans in this bag are white.
Result: These beans are white.
Case: These beans are from this bag.

Induction

Case: These beans are from this bag.
Result: These beans are white.
Rule: *All* the beans in this bag are white.

LIME: synthetic tasks for reasoning primitives

To pretrain transformer networks on four synthetic tasks (Deduct, Abduct, Induct, and Mix) for learning inductive biases of reasoning.

The task of Deduct:

- Source: Rule string and Case dictionary
- Target: Result string

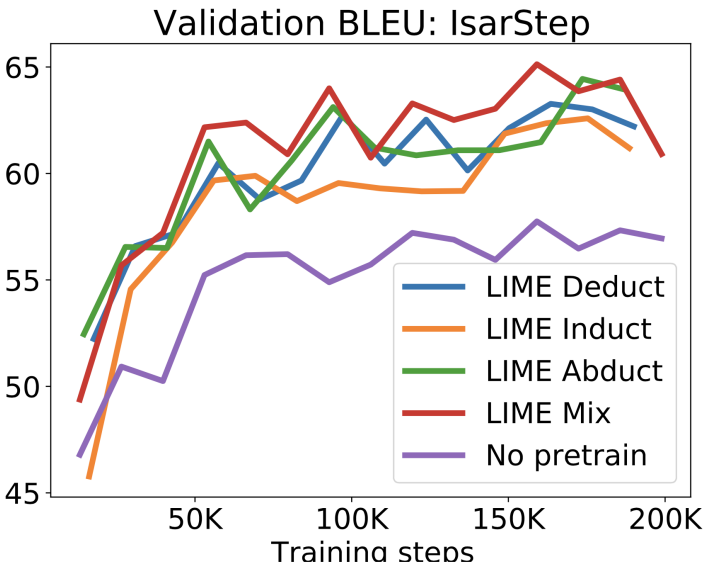
An example of Rule, Case, and Result

Rule: $A + B = B + A$

Case: { A: "xy$%2", B: "8djwh4" }

Result: xy$%2 + 8djwh4 = 8djwh4 + xy$%2

LIME on IsarStep



Model	Top-1 Acc.	Top-10 Acc.
No Pretrain	20.4	33.1
HAT	22.8	35.2
LIME Deduct	24.7	37.7
LIME Abduct	26.7	**41.0**
LIME Induct	23.9	38.8
LIME Mix	**26.9**	40.4

Ablation study

Model	Top-1 Acc.	Top-10 Acc.
No Pretrain	20.4	33.1
LIME Mix	**26.9**	**40.4**
Pretrain on MetaMathStep Deduct	23.1	35.7
Pretrain on WMT En-De	17.2	30.3

Comparisons to other pretraining tasks on IsarStep

LIME on other mathematical reasoning benchmarks

Task	No Pretrain	LIME Best
HOList Skip-Tree/Equation Completion	46.3	51.7
HOList Skip-Tree/Missing Assumptions	41.8	47.9
MetaMathStep	67.7	69.9
LeanStep/Next Lemma Prediction	15.8	29.8

Top-8 Acc. for HOList, and Top-1 Acc. for others.

To conclude LIME

A methodology for designing **synthetic** tasks for pre-training transformer models.

Considerable performance improvement on mathematical reasoning tasks with **negligible** ($< 3\%$) extra computation cost.

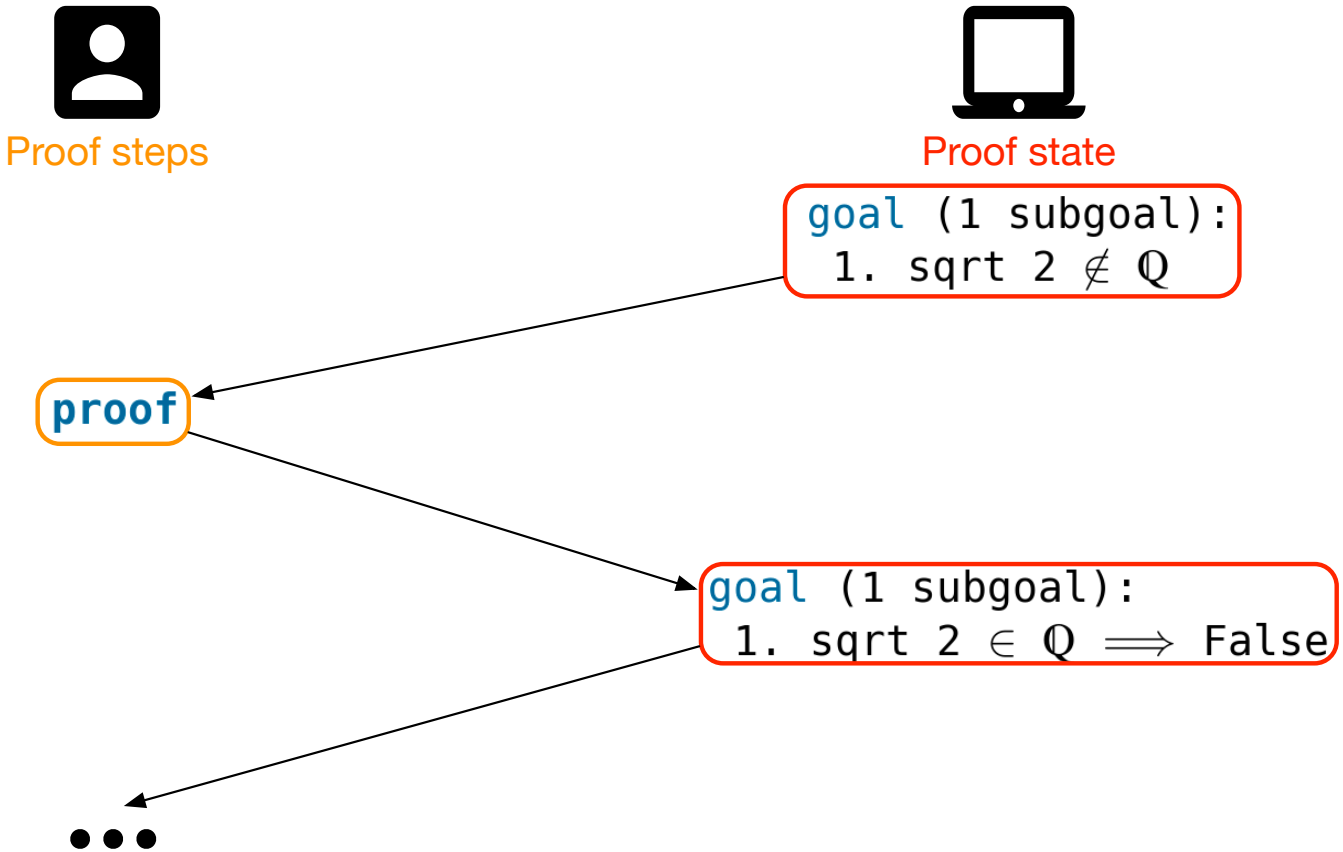
Proving with language models (in progress)

LISA: Language models of ISAbelle proofs

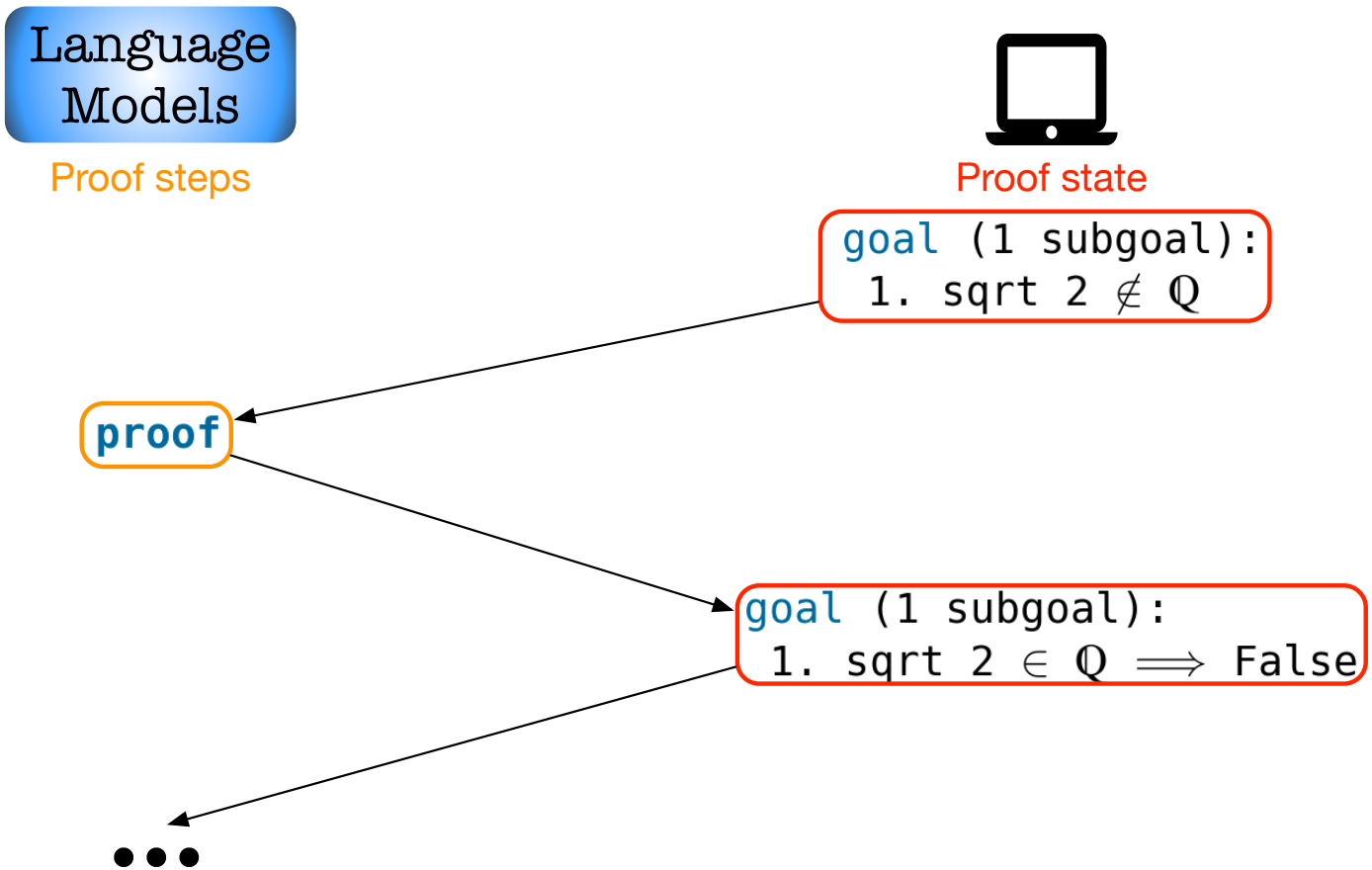
- published at AITP
- joint with Albert Qiaochu Jiang[1], Jesse Michael Han[2], Yuhuai Wu[3] [4]

[1] University of Cambridge
[2] OpenAI
[3] Google Brain
[4] Stanford University

Recap: human experts propose proof steps



Replace humans with language models



Language models & Dataset statistics

A language model models a probability distribution over sequences of words: $p(\mathbf{y}) = \prod_{j=1}^{J} p(y_j \mid \mathbf{y}_1^{j-1})$.

- Decoder-only transformer similar to GPT-3
- Pretrained on the WebMath dataset for 72B tokens
- Fined-tuned on the AFP datatset for 35B elapsed tokens
- Similar setup to GPT-f[1]

For the datatset,

- 2.16M proofs steps, 95%/1%/4% split
- Seq length on average (in characters):
  - Source: 379.6
  - Target: 34.2

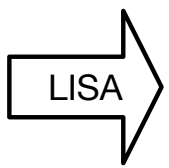
[1] Polu and Sutskever, *Generative Language Modeling for Automated Theorem Proving*.

Initial results

Model/Method	% Completed Proofs
Sledgehammer (baseline, default setup)	30
Language model with 163M parameters	33.2
Language model with 780M parameters	42.7

LISA can shorten proofs

```
lemma foldr_foldr_concat:
  "foldr (foldr f) m a
    = foldr f (concat m) a"
proof (induct m arbitrary: a)
  case Nil
  show ?case by simp
next
  case (Cons v m a)
  show ?case
    unfolding concat.simps
      foldr_Cons o_def Cons
    unfolding foldr_append
      by simp
qed
```




```
lemma foldr_foldr_concat:
  "foldr (foldr f) m a
    = foldr f (concat m) a"
  by (induct m arbitrary: a)
     simp_all
```

Future directions: human-readable proofs

Step-by-step $\rightarrow$ high-level sketches

- a step-by-step proof strategy is unlikely to scale
- introducing conjectures: bringing IsarStep to LISA

Comparison between a step-by-step proof and a human-readable proof

Find the value of $a_2 + a_4 + a_6 + a_8 + \cdots + a_{98}$ if $a_1, a_2, a_3, \cdots$ is an arithmetic progression with common difference 1, and $a_1 + a_2 + a_3 + \cdots + a_{98} = 137$.

```
theorem aime_1984_p1:
  fixes a :: "nat ⇒ rat"
  assumes h0: "∀ n. a (n + 1) = a n + 1"
    and h1: "(∑ k < 98. a (k+1)) = 137"
  shows "(∑ k < 49. a (2*(k+1))) = 93"
```

GPT-f generated proof in Lean

```
begin
  revert_all,
  simp [finset.sum_range_succ],
  ring,
  simp [pow_succ, mul_comm,
     add_left_comm, add_assoc],
  simp [two_mul],
  rintro p,
  revert p,
  simp [add_comm],
  intros p hp,
  simp [hp], ring_nf,
  intros,
  nlinarith
end
```

Human proof in Isabelle

```
proof -
  have "(∑ k < 2*m. a (k+1)) + of_nat m
          = 2*(∑ k < m. a (2 * (k+1)))"
    for m
    by (induct m) (use h0 in simp)+
  from this[of 49]
  have "(∑k<98. a (k + 1))+49
            = 2*(∑k<49. a (2*(k + 1)))"
    by simp
  then have "186 = 2*(∑k<49. a (2*(k+1)))"
    using h1 by simp
  then show ?thesis by auto
qed
```

Summary

In this talk, I covered

- IsarStep: conjecturing benchmarks & HAT
- LIME: introducing inductive bias to the transformer networks for mathematical reasoning
- LISA: a proving environment based on language models

Huge opportunities for both the AI community and the theorem proving community.