Disambiguating natural language with probabilistic inference

Mauricio Barba da Costa, Katherine Collins, Fabian Zaiser, Romir Patel, Alexander K. Lew, Vikash Mansinghka, Timothy O'Donnell, Joshua Tenenbaum, and Cameron Freer

Massachusetts Institute of Technology

EuroProofNet

Auto-formalization

- Given a (potentially imprecise) informal statement, can you extract the formal meaning of the statement?
- How can we resolve ambiguity in Lean 4?

Types of ambiguity in Lean

Type/Domain Ambiguity	Pronoun Ambiguity	Quantifier Scope Ambiguity
For all x, there exists y such that $y^2=x$.	If a function has a derivative, it is continuous.	Each f is bounded by some g.
`/-- x and y are natural numbers -/` `∀ x : ℕ, ∃ y : ℕ, y^2 = x`	`/-- “it” refers to the function -/` `∀ (f : ℝ → ℝ), Differentiable ℝ f → Continuous f`	`/-- every f has its own bound g -/` `∀ f, ∃ g, f ≤ g`
`/-- x and y are complex numbers -/` `∀ x : ℂ, ∃ y : ℂ, y^2 = x`	`/-- “it” refers to the derivative -/` `∀ (f : ℝ → ℝ), Differentiable ℝ f → Continuous (deriv f)`	`/-- one g bounds every f -/` `∃ g, ∀ f, f ≤ g`

Why auto-formalization?

- Can assess how well machines understand the intents of their users. Better auto-formalization means better thought partners
- Understanding how humans alternate between precise reasoning and rough draft type thinking

Defining success for auto-formalization is difficult

- Traditional machine learning approach: compare label to ground truth.
  - (for propositions) Any true statement implies any other true statement
  - (for predicates) is undecidable

$$
\begin{array}{l}
f : \mathbb{C} \to \mathbb{C} \text{ is a polynomial} \\
\stackrel{?}{=} \\
f : \mathbb{C} \to \mathbb{C} \text{ is a polynomial with number of roots equal to its degree.}
\end{array}
$$

Defining success for auto-formalization is difficult

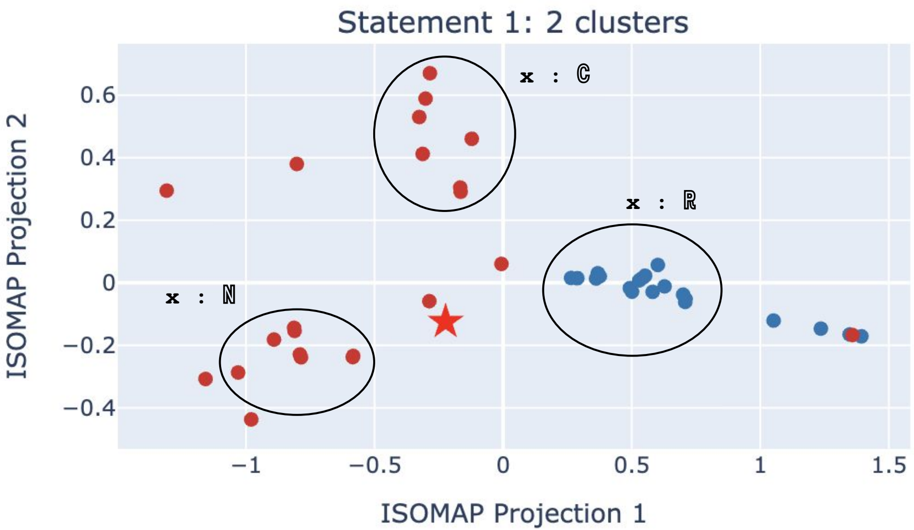
- Traditional machine learning approach: compare label to ground truth.
  - (for propositions) Any true statement implies any other true statement
  - (for predicates) is undecidable
- Despite this, humans still have an intuition for when two statements are equivalent

$$
\begin{aligned}
&f : \mathbb{C} \to \mathbb{C} \text{ is a polynomial} \\
&\stackrel{?}{=} \\
&f : \mathbb{C} \to \mathbb{C} \text{ is a polynomial with number of roots equal to its degree.}
\end{aligned}
$$

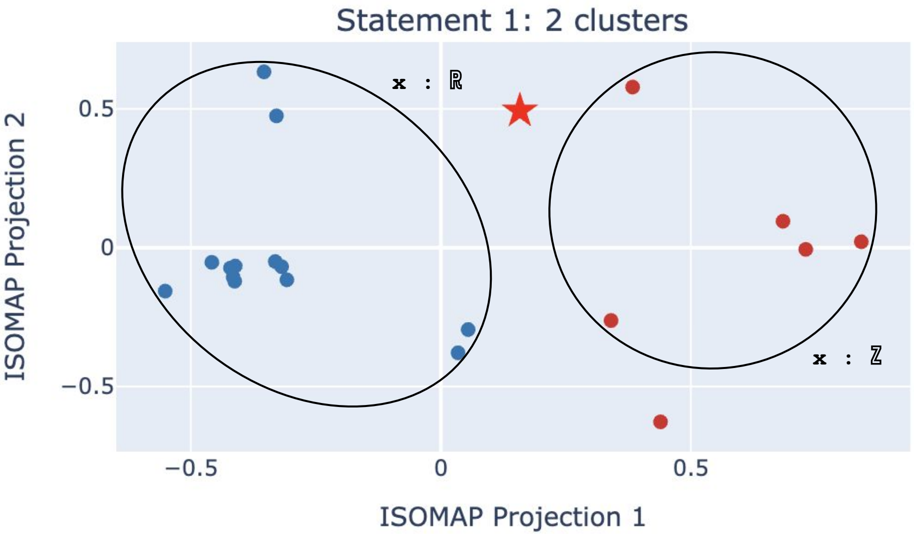
Ambiguity via LM

- How well is a language embedding model at distinguishing between unequal formalizations?
- LM attempts to formalize statement 50 times
- Embedding model converts statement to vector
- Reduce dimensionality to view in 2 dimensions.

Ambiguity via LLM



“For all x, there exists y such that y^2 = x”



“For any x, 0 <= x^2”

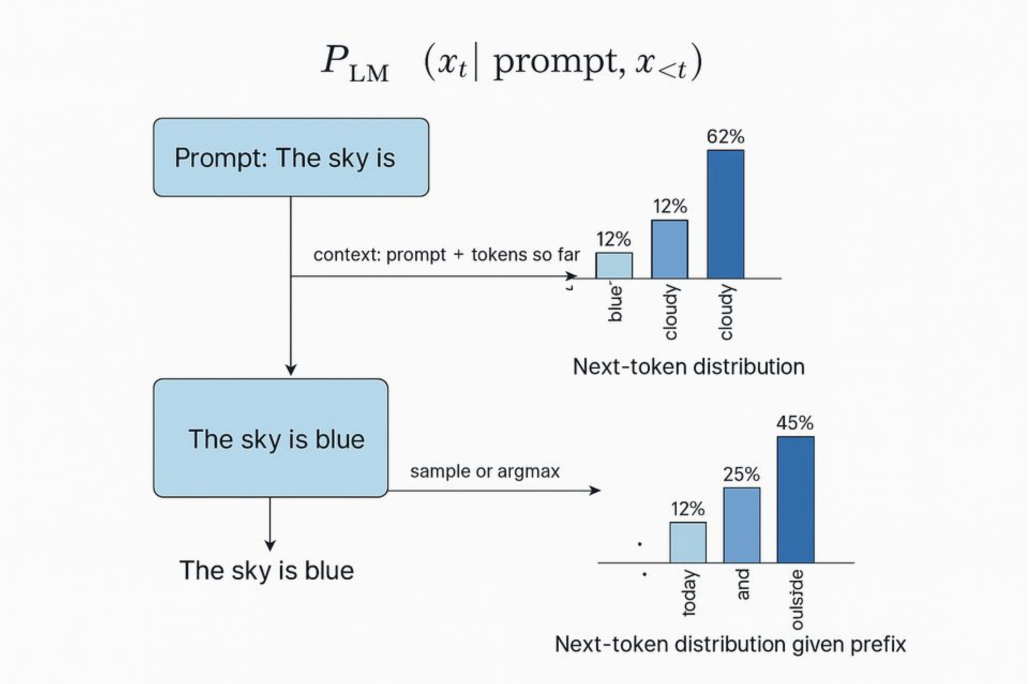
Isomap from Tenenbaum, de Silva, and Langford, Science, 2000

Ambiguity via LLM

- Some structure is preserved!
  - Robust to semantic-preserving transformations like reordering hypotheses, renaming hypotheses
  - Gives natural clustering boundaries
  - Different disambiguations are represented
  - Informal statement is embedded roughly between all the formalization attempts

Can we do something more principled?

- LLMs are trained using next-token prediction. Why should we expect that they can reason about math?
- LMs define conditional distributions for sequences of text



Outline

- Introducing autoformalization as inference
- Preliminary experiments and simple case studies

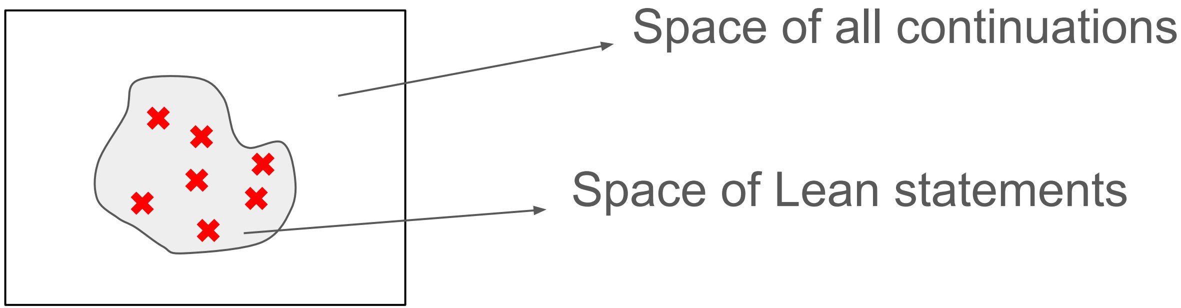
This talk

- Useful constraints/signals for autoformalization?
- How systematically combine these ingredients?
- Preliminary experimental results

Next talk

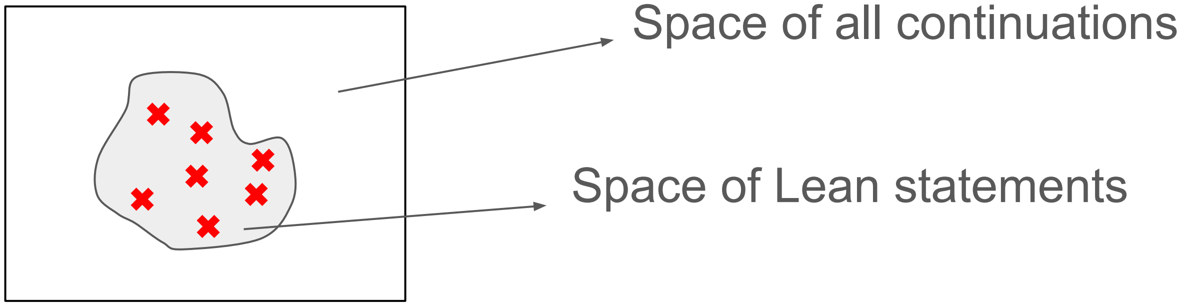
Posterior inference

- **Generate samples from a target distribution** that is often difficult to compute.
- For instance, “the distribution given by sentences in the English language” conditioned on “all the words must not contain the letter e”.



Can we frame auto-formalization as posterior inference?

- Can we use an LLM as a **proposal distribution** (which is what it was designed for) for sampling from a target distribution?
- What might that target distribution look like?



Auto-formalization as posterior inference

- Proposal: autoformalization by sampling from a distribution that adjusts for multiple factors

$$P^*(X_{\text{formal}}|X_{\text{informal}}) \propto P_{\rightarrow}(X_{\text{formal}}|X_{\text{informal}})P_{\leftarrow}(X_{\text{informal}}|X_{\text{formal}})\mathbf{1}_{\text{well-typed}}\mathbf{1}_{\text{plausible}}$$

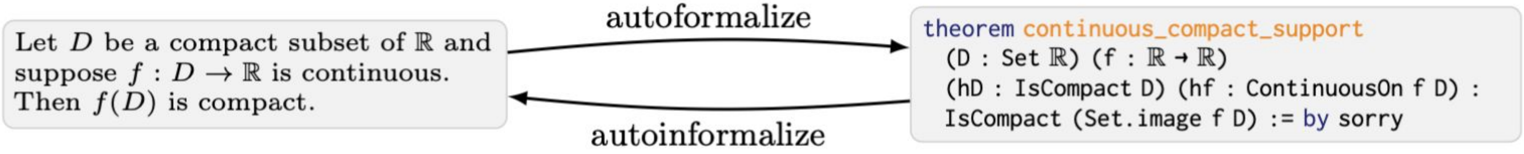
- We can **sample approximately** from this distribution
- See more in next talk to see how this is done in practice!

Auto-formalization as posterior inference

- The first two terms correspond to cycle consistency

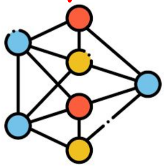
$$P^*(X_{\text{formal}}|X_{\text{informal}}) \propto P_{\rightarrow}(X_{\text{formal}}|X_{\text{informal}})P_{\leftarrow}(X_{\text{informal}}|X_{\text{formal}})\mathbf{1}_{\text{well-typed}}\mathbf{1}_{\text{plausible}}$$

  - **autoformalization**: translate an informal math statement to corresponding formal statement, specifically: English LaTeX → LEAN



Auto-formalization as posterior inference

$$P^*(X_{\text{formal}}|X_{\text{informal}}) \propto \underbrace{P_{\rightarrow}(X_{\text{formal}}|X_{\text{informal}})P_{\leftarrow}(X_{\text{informal}}|X_{\text{formal}})}_{} \mathbf{1}_{\text{well-typed}}\mathbf{1}_{\text{plausible}}$$



- Forward LLM prompt: "**Formalize**…"
- Reverse LLM prompt: "State this statement **in natural language**: "
    - Evaluate the likelihood of the continuation

Case Study

Example disambiguating with forward and reverse kernels

- “If f is continuous on a closed interval, then it is bounded.”
  - Which definitions do I use?
  - What is the domain of the interval?
  - What is the domain of f ?
  - What is the quantifier of the interval and of f ?
  - What is the quantifier of the interval?
  - The statement is False.
- Natural language is underspecified, so these sorts of questions need to be answered by an autoformalizer

“If *f* is continuous on a closed interval, then it is bounded”

	log P(formal | informal)	log P(informal | formal)
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b → ContinuousOn f (Set.Icc a b) → ∃ M : ℝ, ∀ x ∈ Set.Icc a b, |f x| ≤ M BEST	**-85.4375** +	**-10.2656**

“If *f* is continuous on a closed interval, then it is bounded”

	log P(formal | informal)	log P(informal | formal)
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →… BEST	-85.4375	-10.2656
∀ f : ℝ → ℝ, ContinuousOn f (Set.univ) → ∀ a b : ℝ, a ≤ b → BoundedOn f (Set.Icc a b)	**-96.0625**	**-13.2656**

Contrived formalization

“If *f* is continuous on a closed interval, then it is bounded”

	log P(formal | informal)	log P(informal | formal)
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →… BEST	-85.4375	-10.2656
∀ f : ℝ → ℝ, ContinuousOn f (Set.univ) → ∀ a b : ℝ, a ≤ b → BoundedOn f (Set.Icc a b)	**-96.0625**	**-13.2656**

Reverse direction is saying something different

“If *f* is continuous on a closed interval, then it is bounded”

	log P(formal | informal)	log P(informal | formal)
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →… BEST	-85.4375	-10.2656
∀ f : ℝ → ℝ, ContinuousOn f…	-96.0625	-13.2656
∀ f : ℝ → ℝ, (∃ a b : ℝ, a ≤ b ∧ ContinuousOn f (Set.Icc a b)) → ∃ a b : ℝ, a ≤ b ∧ BoundedOn f (Set.Icc a b)	**-100.3750**	**-11.3906**

Unlikely quantifier

“If *f* is continuous on a closed interval, then it is bounded”

	log P(formal | informal)	log P(informal | formal)
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →… BEST	-85.4375	-10.2656
∀ f : ℝ → ℝ, ContinuousOn f…	-96.0625	-13.2656
∀ f : ℝ → ℝ, (∃ a b : ℝ, a ≤ b ∧…	-100.3750	-11.3906
∀ f : ℝ → ℝ, ∀ K : Set ℝ, IsCompact ContinuousOn f K → BoundedOn f K	**-100.1250**	**-18.5000**

IsCompact describes something different

“If *f* is continuous on a closed interval, then it is bounded”

	log P(formal | informal)	log P(informal | formal)
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →…	-85.4375	-10.2656
∀ f : ℝ → ℝ, ContinuousOn f…	-96.0625	-13.2656
∀ f : ℝ → ℝ, (∃ a b : ℝ, a ≤ b ∧…	-100.3750	-11.3906
∀ f : ℝ → ℝ, ∀ K : Set ℝ, IsCompact…	-100.1250	-18.5000
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b → ContinuousOn f (Set.Icc a b) → BoundedOn f (Set.Icc a b) BEST	-86.1875 +	-8.7500

“If *f* is continuous on a closed interval, then it is bounded”

	log P(formal | informal)	log P(informal | formal)
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b → ContinuousOn f (Set.Icc a b) → ∃ M : ℝ, ∀ x ∈ Set.Icc a b, |f x| ≤ M	-85.4375	-10.2656
∀ f : ℝ → ℝ, ContinuousOn f…	-96.0625	-13.2656
∀ f : ℝ → ℝ, (∃ a b : ℝ, a ≤ b ∧…	-100.3750	-11.3906
∀ f : ℝ → ℝ, ∀ K : Set ℝ, IsCompact…	-100.1250	-18.5000
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b → ContinuousOn f (Set.Icc a b) → BoundedOn f (Set.Icc a b) BEST	-86.1875	-8.7500

Informalization would likely spell out the bound *M*.

“If *f* is continuous on a closed interval, then it is bounded”

	log P(formal | informal)	log P(informal | formal)
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →…	-85.4375	-10.2656
∀ f : ℝ → ℝ, ContinuousOn f…	-96.0625	-13.2656
∀ f : ℝ → ℝ, (∃ a b : ℝ, a ≤ b ∧…	-100.3750	-11.3906
∀ f : ℝ → ℝ, ∀ K : Set ℝ, IsCompa	-100.1250	-18.5000
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →…	-86.1875	-8.7500
∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b → ContinuousOn f (Set.Icc a b) → Bdd.above (f '' Set.Icc a b) ∧ Bdd.below (f '' Set.Icc a b)	**-98.4375**	**-10.8438**

Forward direction overly complicated

Auformalization as posterior inference

- Language model outputs aren't guaranteed to be well-typed

$$P^*(X_{\text{formal}}|X_{\text{informal}}) \propto P_{\rightarrow}(X_{\text{formal}}|X_{\text{informal}})P_{\leftarrow}(X_{\text{informal}}|X_{\text{formal}})\mathbf{1}_{\text{well-typed}}\mathbf{1}_{\text{plausible}}$$

Well-typed check

- $\forall\ f : \mathbb{R} \to \mathbb{R},\ \forall\ a\ b : \mathbb{R},\ a \le b \to \ldots$
- $\forall\ f : \mathbb{R} \to \mathbb{R}$, ContinuousOn f…
- $\forall\ f : \mathbb{R} \to \mathbb{R},\ (\exists\ a\ b : \mathbb{R},\ a \le b\ \land \ldots$
- $\forall\ f : \mathbb{R} \to \mathbb{R},\ \forall\ K :$ Set $\mathbb{R}$, IsCompact…
- $\forall\ f : \mathbb{R} \to \mathbb{R},\ \forall\ a\ b : \mathbb{R},\ a \le b \to \ldots$
- **$\forall\ f : \mathbb{R} \to \mathbb{R},\ \forall\ a\ b : \mathbb{R},\ a \le b \to$ ContinuousOn f (Set.Icc a b) $\to$ Bdd.above (f '' Set.Icc a b) $\land$ Bdd.below (f '' Set.Icc a b)**

The Bdd.above and Bdd.below predicates were hallucinated!

Well-typed check

- ∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →…
- ∀ f : ℝ → ℝ, ContinuousOn f…
- ∀ f : ℝ → ℝ, (∃ a b : ℝ, a ≤ b ∧…
- ∀ f : ℝ → ℝ, ∀ K : Set ℝ, IsCompact…
- ∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b →…
- ∀ f : ℝ → ℝ, ∀ a b : ℝ, a ≤ b → ContinuousOn f (Set.Icc a b) → Bdd.above (f '' Set.Icc a b) ∧ Bdd.below (f '' Set.Icc a b)

P(informal | formal)	P(formal | informal)
-86.1875	-8.7500
~~-96.0625~~	~~-13.2656~~
~~-100.3750~~	~~-11.3906~~
~~-100.1250~~	~~-18.5000~~
~~-85.4375~~	~~-10.2656~~
~~**-98.4375**~~	~~**-10.8438**~~

Actually, a lot of things were hallucinated

Plausibility check

$$P^*(X_{\text{formal}}|X_{\text{informal}}) \propto P_{\rightarrow}(X_{\text{formal}}|X_{\text{informal}})P_{\leftarrow}(X_{\text{informal}}|X_{\text{formal}})\mathbf{1}_{\text{well-typed}}\mathbf{1}_{\text{plausible}}$$

Biases during formalization

- Correct statements are more likely to be what was intended
    - This statement “If *f* is continuous on a closed interval, then it is bounded” is false! but you probably know what I meant, or how to easily salvage the statement to make it correct.
- **Statements that are falsifiable via a counterexample are less likely to be what was intended**
- Statements that cohere with earlier context are more likely to be what was intended
- Non-trivial statements are more likely to be what was intended
- Statements that match with statements stored in my memory are more likely to be what was intended

Biases during formalization

- What did the author mean here?
  - Can I disambiguate what they meant by coming up with a counterexample?
- “Every positive number has a square root”
  - $\forall x : \mathbb{R}, \exists y : \mathbb{R}, y^2 = x$
  - $\exists y : \mathbb{R}, \forall x : \mathbb{R}, y^2 = x$
- “Well, 1 and 2 are positive real numbers and they have different square roots so the correct formalization is more likely to be the first statement”

Thought experiment operationalizing plausibility bias

```
example (a : ℤ) : a ≥ 0 := by
  plausible
```

```
▼ All Messages (1)
 ▼ test.lean:5:2

  ===================
  Found a counter-example!
  a := -1
  issue: 0 ≤ -1 does not hold
  (0 shrinks)
  -------------------
```

```
example (a : ℕ) : a ≥ 0 := by
  plausible
```

```
▼ Messages (1)
 ▼ test.lean:5:2

   Unable to find a counter-example

▶ All Messages (2)
```

Toy example: Formalize "For all x, x ≥ 0"

example (a : ℕ) : a ≥ 0 := by plausible	✅
example (a : ℤ) : a ≥ 0 := by plausible	❌
example : ∀ x : ℕ, x ≥ 0 := by plausible	✅
example : ∀ x : ℕ, 0 ≤ x := by plausible	✅

Plausibility as assessed by plausible tactic

Toy example: Formalize the associative law

Subtraction is really monus for natural numbers.

example : ∀ (x y z : ℕ), x + y - z = x + (y - z) := by plausible	❌
example : ∀ (x y z : ℤ), x + y - z = x + (y - z) := by plausible	✅
example : ∀ (x y z : ℚ), x + y - z = x + (y - z) := by plausible	✅
example {x y z : ℤ} : x + y - z = x + (y - z) := by plausible	✅

Auto-formalization as posterior inference

$$P^*(X_{\text{formal}}|X_{\text{informal}}) \propto P_{\rightarrow}(X_{\text{formal}}|X_{\text{informal}})P_{\leftarrow}(X_{\text{informal}}|X_{\text{formal}})\mathbf{1}_{\text{well-typed}}\mathbf{1}_{\text{plausible}}$$

$$\times \exp(\mathbf{1}_{\text{provable with hammer}})$$

$$\times \mathbf{1}_{\text{nontrivial}}$$

$$\times P(X_{\text{formal}}|Y_{\text{surrounding context}})$$

Outline

- Introducing autoformalization as inference
- Preliminary experiments and simple case studies

This talk

- Useful constraints/signals for autoformalization?
- How systematically combine these ingredients?
- Preliminary experimental results

Next talk

Thank you!