

Aria: An Agent for Retrieval and Iterative Auto-Formalization via Dependency Graph

Hanyu Wang, Ruohan Xie, Yutong Wang, Guoxiong Gao
Xintao Yu, Bin Dong

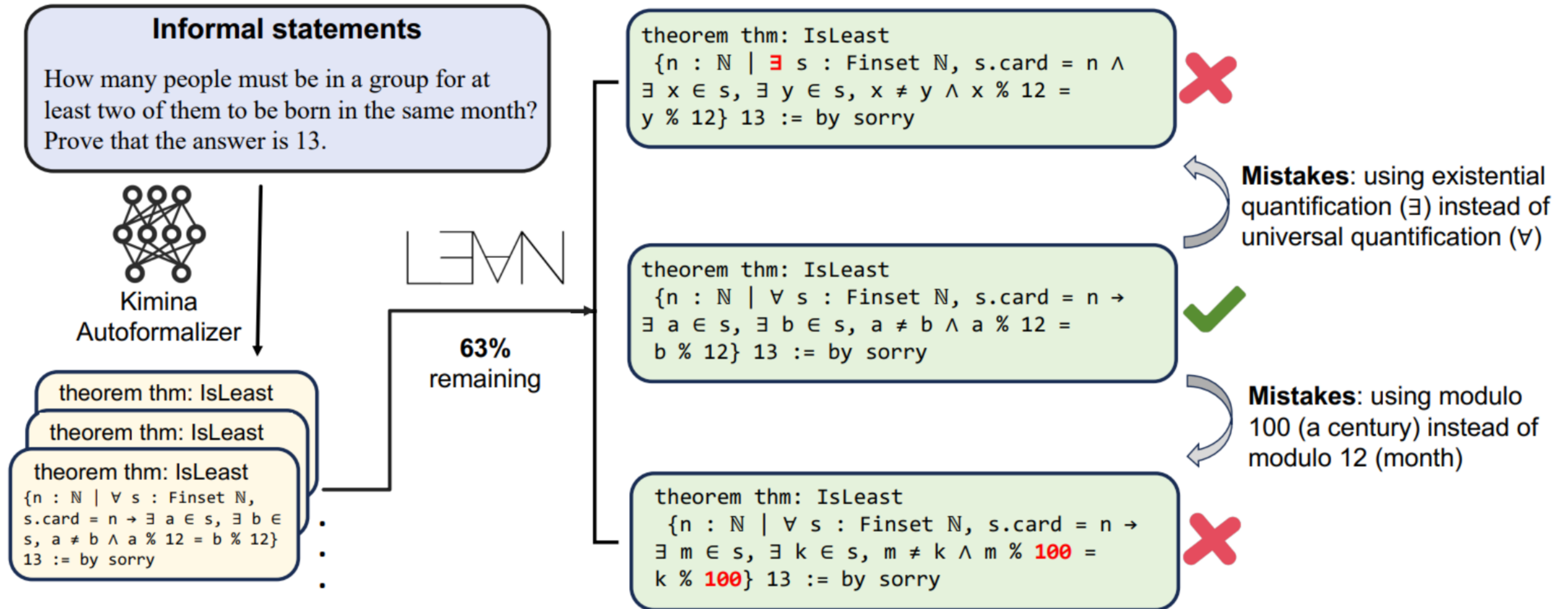
AI4M team, BICMR, Peking University

Outline

- **Background**
- **Aria–Autoformalizer**
 - Methodology
 - Experiments
- **AriaScorer**
 - Methodology
 - Experiments
- **Conclusion**

Background

Challenges




Challenges: Syntactic Error

- Generated formal statement by Herald

```
1 import Mathlib
2 open Ideal Set
3 /-- Let  $R$  be a UFD with two nonassociate prime elements  $p$  and  $q$  such that every prime element is an associate of either  $p$  or  $q$ . -/
4 | Prove that  $R$  is a PID. -/
5 theorem of_exists_prime_of_ufd_of_unique_irreducible {R} [CommRing R] [IsDomain R] [UniqueFactorizationMonoid R] (p q : R)
6 | (hp : Irreducible p) (hq : Irreducible q) (h_prime :  $\forall \{x : R\}, \text{Irreducible } x \rightarrow x \sim_u p \vee x \sim_u q$ ) :
7 | | | IsPrincipalIdealRing R := sorry
```

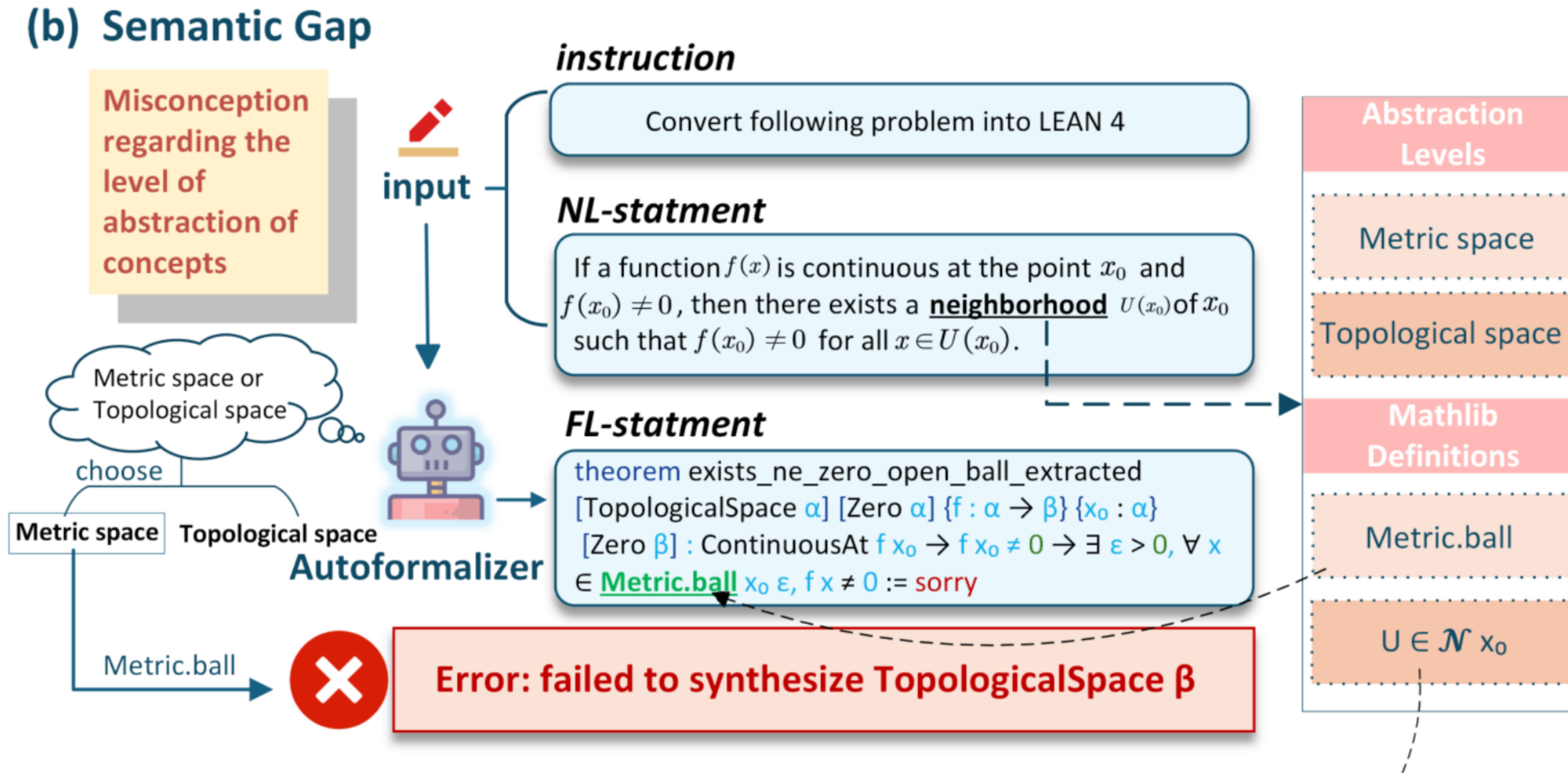
- Generated formal statement by Aria

```
1 import Mathlib
2 /--
3 Let  $R$  be a UFD with two nonassociate prime elements  $p$  and  $q$  such that every prime
4 element is an associate of either  $p$  or  $q$ . Then  $R$  is a PID.
5 -/
6 theorem UFD_with_two_prime_classes_is_PID
7 | {R : Type*} [CommRing R] [IsDomain R] [UniqueFactorizationMonoid R]
8 | (p q : R)
9 | (hp : Prime p)
10 | (hq : Prime q)
11 | (hpq_non_assoc :  $\neg$  Associated p q)
12 | (h_all_primes :  $\forall r : R, \text{Prime } r \rightarrow \text{Associated } r p \vee \text{Associated } r q$ ) :
13 | | | IsPrincipalIdealRing R := sorry
```



▼ All Messages (1)
▼ **MathlibDemo.lean:6:86**
expected token

Challenges: Semantic Mismatch



Challenges: Complexity Barrier

- Generated formal statement by **Kimina-Autoformalizer**

```
1 import Mathlib
2 open Fintype Set Real Ideal Polynomial
3 open scoped BigOperators
4 /-
5 Let  $(G)$  be a finite group acting as automorphisms of an algebra  $(R)$  over a field of characteristic  $(\theta)$ . Show that if
 $(R)$  is Cohen-Macaulay, then the ring of invariants  $(R^G)$  is Cohen-Macaulay.
6 -/
7 theorem algebra_447377 {IsCohenMacaulay} {R : Type*} [Ring R] [Field R] (hR : CharP R  $\theta$ )
8   {G : Type*} [Group G] [Fintype G] (hG : IsCohenMacaulay R) :
9     IsCohenMacaulay (R / span ({R} : Set G)) :=
10    SORRY
```

▼ All Messages (2)

▼ MathlibDemo.lean:8:42

Function expected at
IsCohenMacaulay
but this term has type
?m.1

Note: Expected a function because this term is
being applied to the argument
R

▼ MathlibDemo.lean:9:4

Function expected at
IsCohenMacaulay
but this term has type
?m.1

Note: Expected a function because this term is
being applied to the argument
(R / span ({ R } : Set G))

Challenges: Semantic Consistency Checking

- An inconsistent example from **Olympiad**

Let positive integers K and d be given. Prove that there exists a positive integer n and a sequence of K positive integers b_1, b_2, \dots, b_K such that the number n is a d -digit palindrome in all number bases b_1, b_2, \dots, b_K .

```
1 import Mathlib
2
3 theorem olymid_ref_base_296 (K d : ℕ) (hd : d > 0) :
4   ∃ n : ℕ, n > 0 ∧ ∃ b : Fin K → ℕ,
5     (∀ i, (Nat.digits (b i) n).length = d) ∧
6     (∀ i, List.Palindrome (Nat.digits (b i) n)) := by sorry
```

Aria–Autoformalizer

- Methodology

Solution: From One-Pass Model to Agent Pipeline

- Syntactic Error: Using **REPL** error message feedback for **self-reflection**. Using **RAG** to establish Mathlib knowledge.
- Semantic Mismatch: Using strongest close-source model like **Gemini-2.5-pro** for better mathematical understanding.
- Complexity Barrier: Using **Graph-of-Thought (GoT)** to decompose and synthesis the complex dependency of mathematical concepts.

Self-Reflection in Formal Math

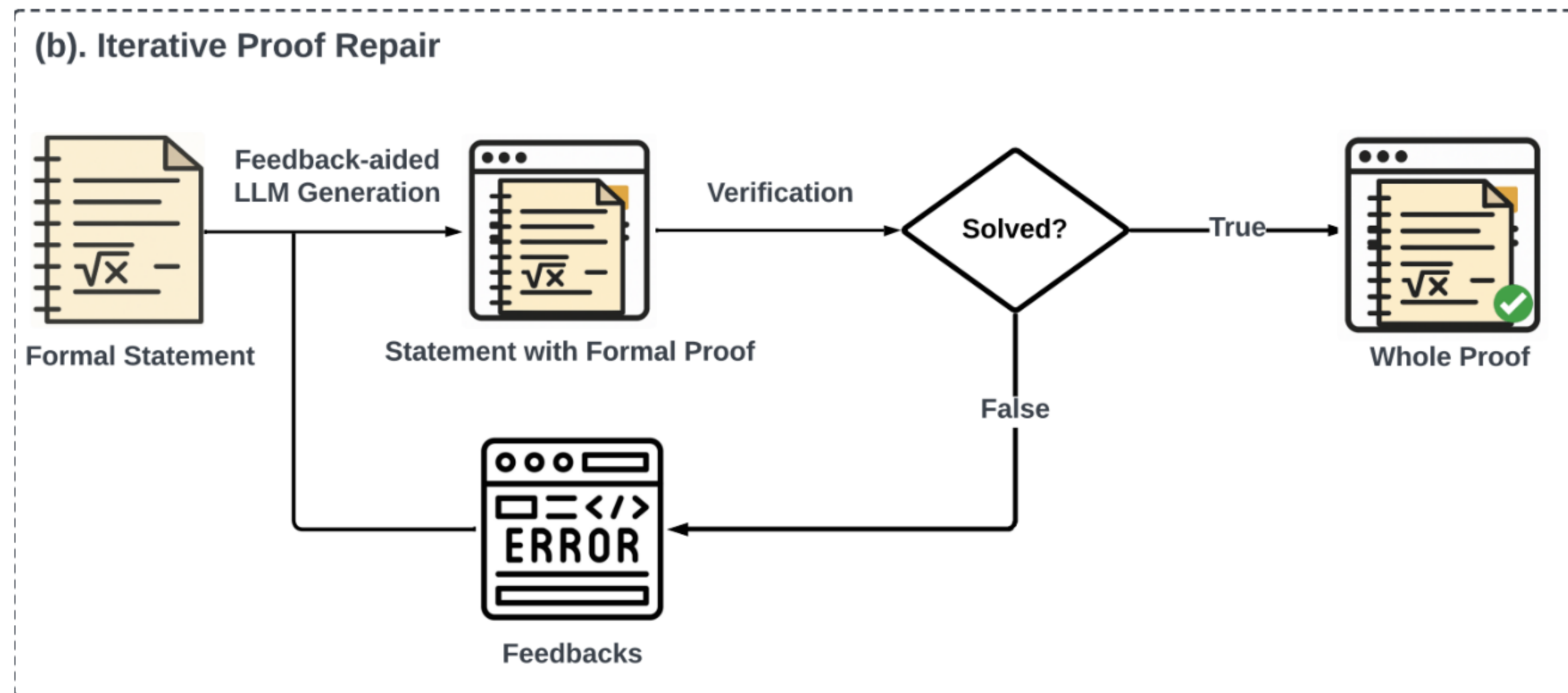
ByteDance | Seed

Delta-Prover Seed-Prover

Solving Formal Math Problems by Decomposition and Iterative Reflection

ByteDance | Seed

Seed-Prover: Deep and Broad Reasoning for Automated Theorem Proving



LeanSearch

- <https://leansearch.net/>

LeanSearch

Find theorems in Mathlib4 using natural language query

Query Name or description of the theorem or definition you are looking for

Example: $\dim(V) = \dim(\ker) + \dim(\text{im})$

Number of results

Clear

Query Augmentation

Search

Tip: **Query Augmentation** augments your query to increase the chance to find relevant results.

LeanSearch: An Example

LeanSearch

Find theorems in Mathlib4 using natural language query

Query Name or description of the theorem or definition you are looking for

Number of results

Clear

Query Augmentation

Search

Tip: **Query Augmentation** augments your query to increase the chance to find relevant results.

Polynomial.uniqueFactorizationMonoid

instance

: UniqueFactorizationMonoid $D[X]$

► Polynomial Rings over UFDs are UFDs

Doc Doc-Next-Gen (beta) Similar

↑ ↓ ×

MvPolynomial.uniqueFactorizationMonoid

instance

: UniqueFactorizationMonoid (MvPolynomial σ D)

► Multivariate Polynomial Ring over a UFD is a UFD

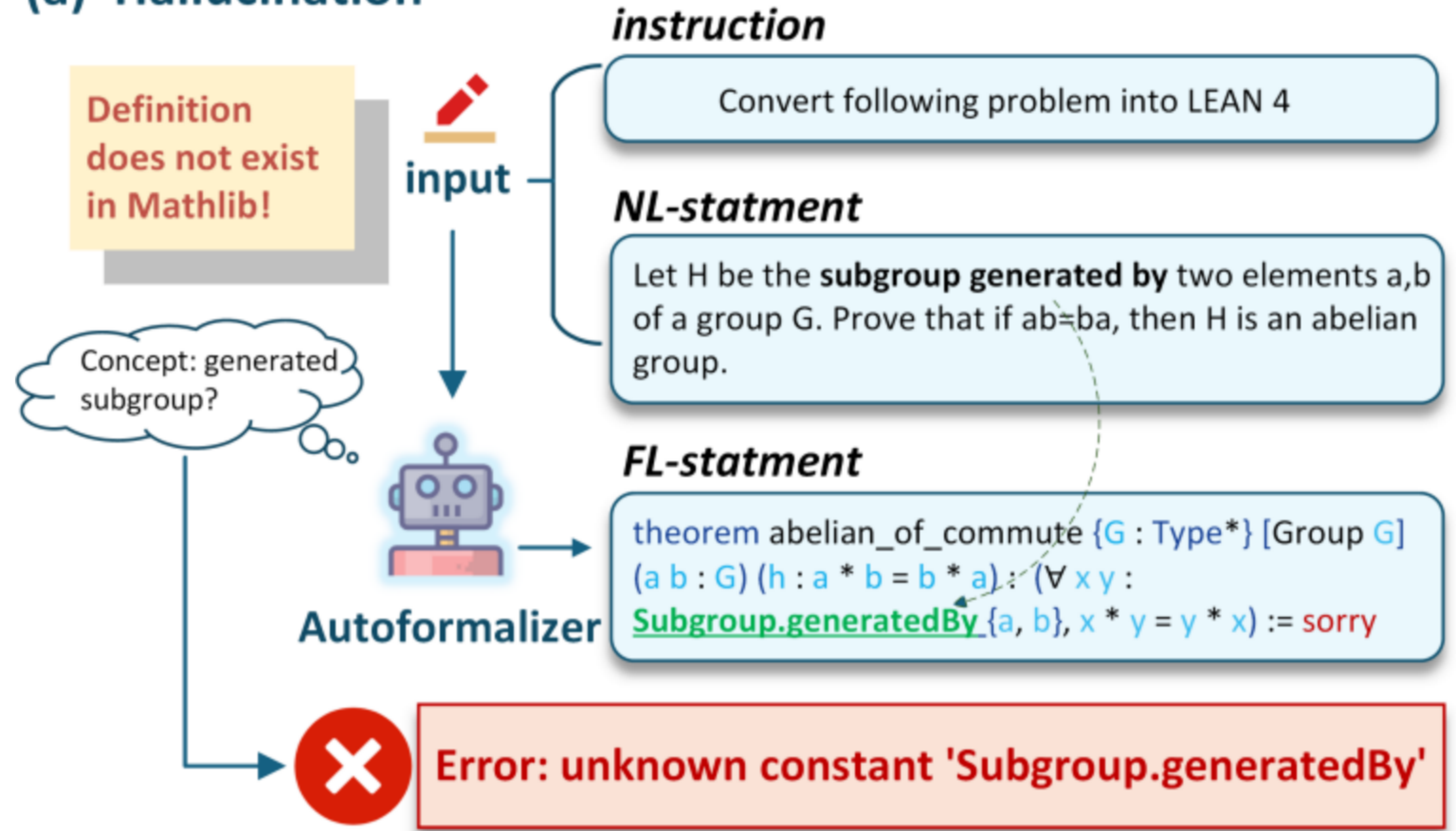
Doc Doc-Next-Gen (beta) Similar

↑ ↓ ×

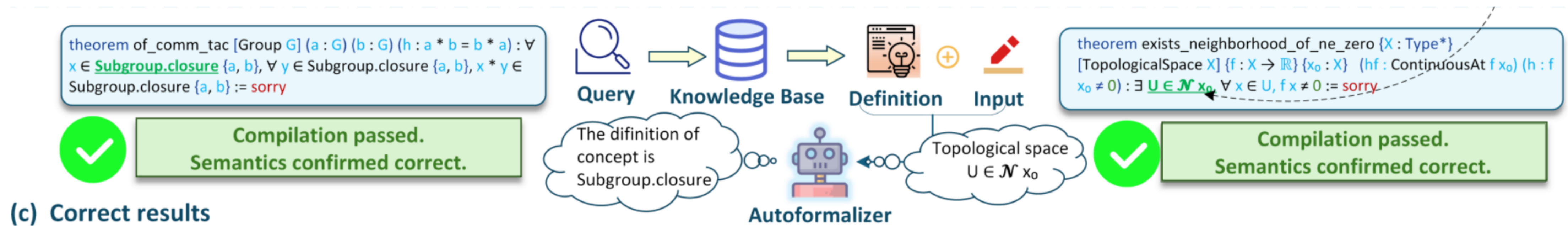
Retrieval-Augmented Generation (RAG)

- Retrieve Concepts in Mathlib4

(a) Hallucination



Retrieval-Augmented Generation

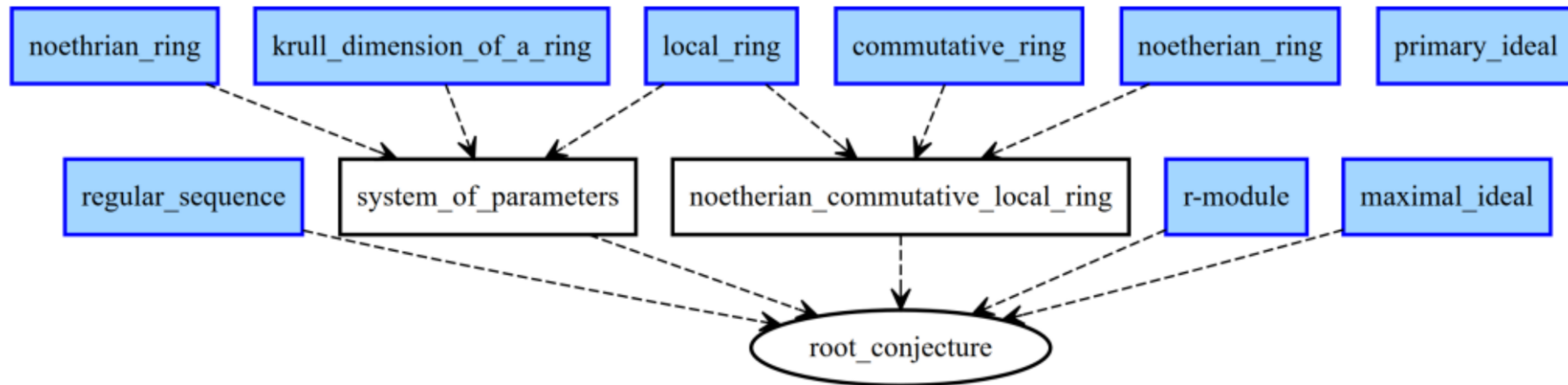


Graph-of-Thought (GoT) Mechanism

- Mirrors the nature of mathematical reasoning.

INFORMAL STATEMENT

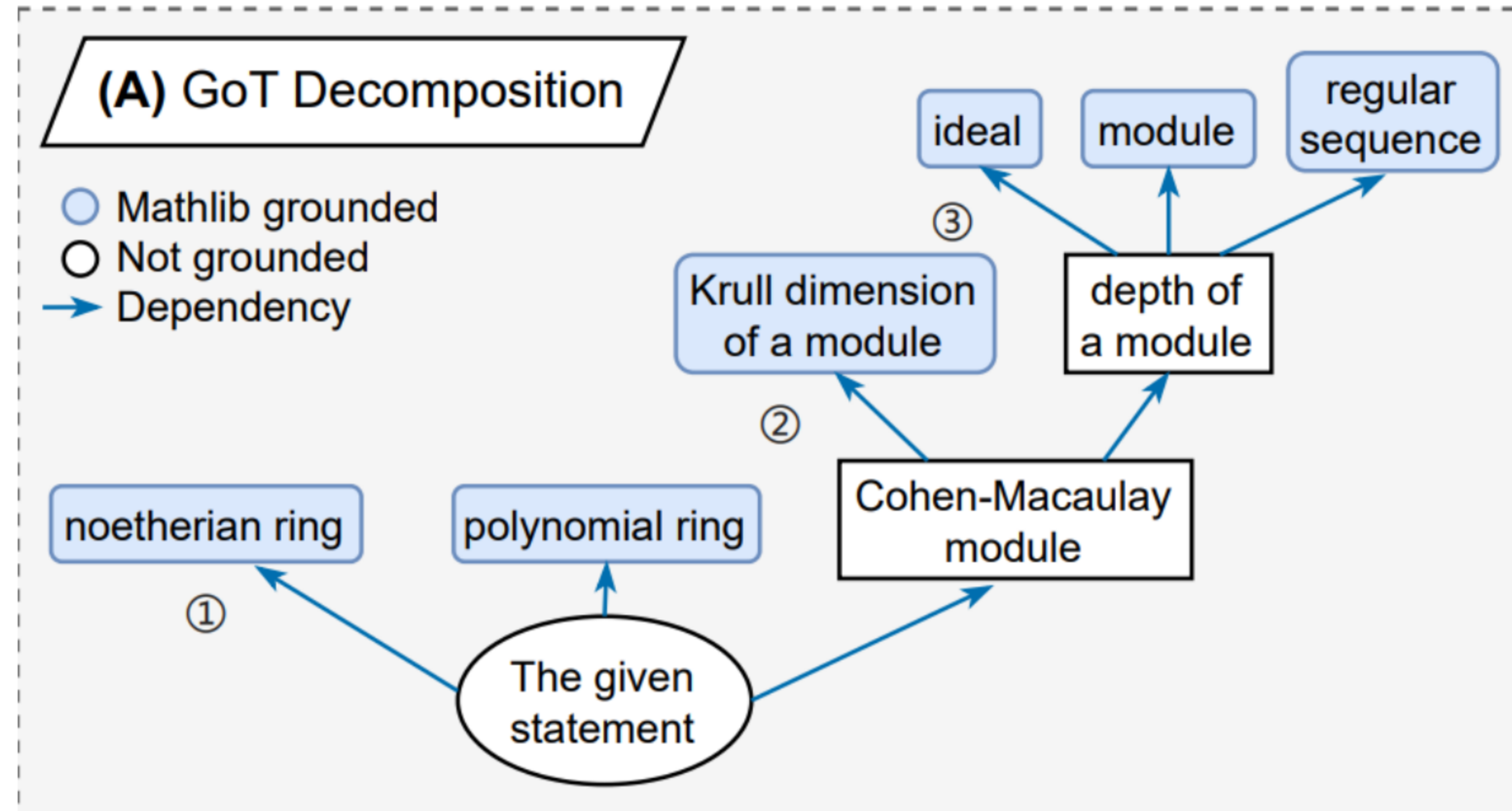
Let R be a Noetherian commutative local ring with maximal ideal m_R . There exists a R -module W such that $m_R W \neq W$ and every system of parameters for R is a regular sequence on W .



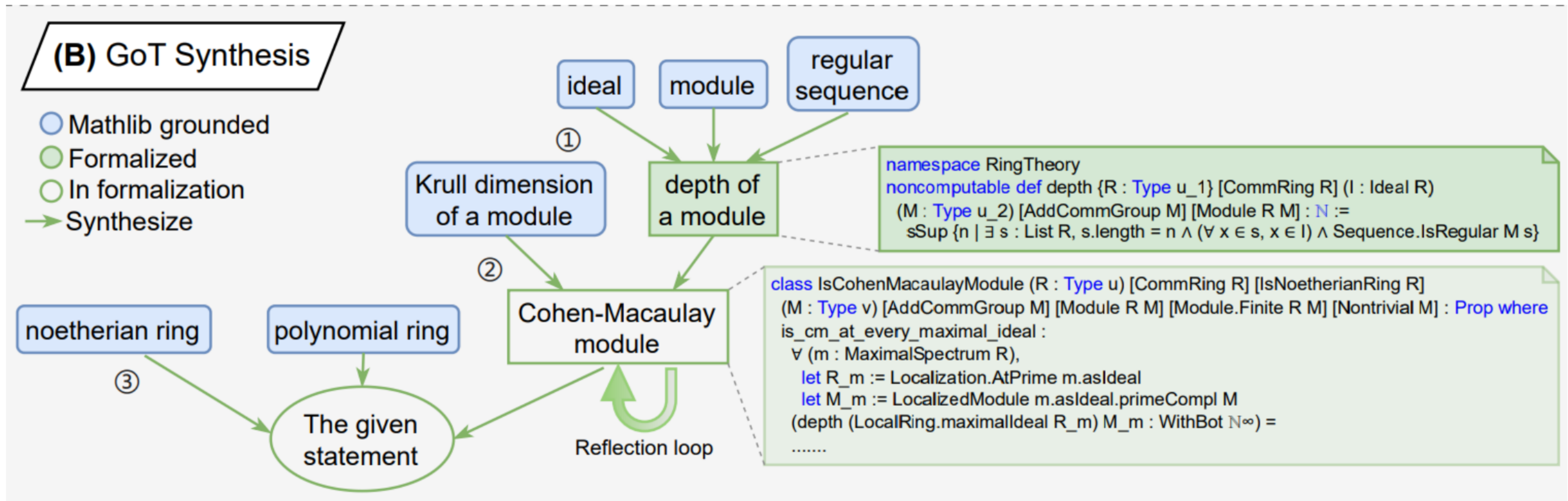
Graph-of-Thought (GoT) Decomposition

Let R be a Noetherian ring. Let M be a Cohen-Macaulay module over R . Then $M \otimes_R R[x_1, \dots, x_n]$ is a Cohen-Macaulay module over $R[x_1, \dots, x_n]$.

Informal Statement



Graph-of-Thought (GoT) Synthesis



Graph-of-Thought (GoT) Synthesis

```
import Mathlib
namespace RingTheory
noncomputable def depth <...>
class IsCohenMacaulayModule <...>
open TensorProduct
theorem isCohenMacaulayModule_tensor_mvPolynomial (n : ℕ)
  (R : Type u) [CommRing R] [IsNoetherianRing R]
  (M : Type v) [AddCommGroup M] [Module R M] [Module.Finite R M] [Nontrivial M]
  [IsCohenMacaulayModule R M]
  [h_alg : Algebra R (MvPolynomial (Fin n) R)]
  [h_mod : Module (MvPolynomial (Fin n) R) (M ⊗[R] MvPolynomial (Fin n) R)]
  [h_mod_fin : Module.Finite (MvPolynomial (Fin n) R)
    (M ⊗[R] MvPolynomial (Fin n) R)]
  [h_nontrivial : Nontrivial (M ⊗[R] MvPolynomial (Fin n) R)]:
  IsCohenMacaulayModule (MvPolynomial (Fin n) R)
  (M ⊗[R] MvPolynomial (Fin n) R) := by
  sorry
end RingTheory
```

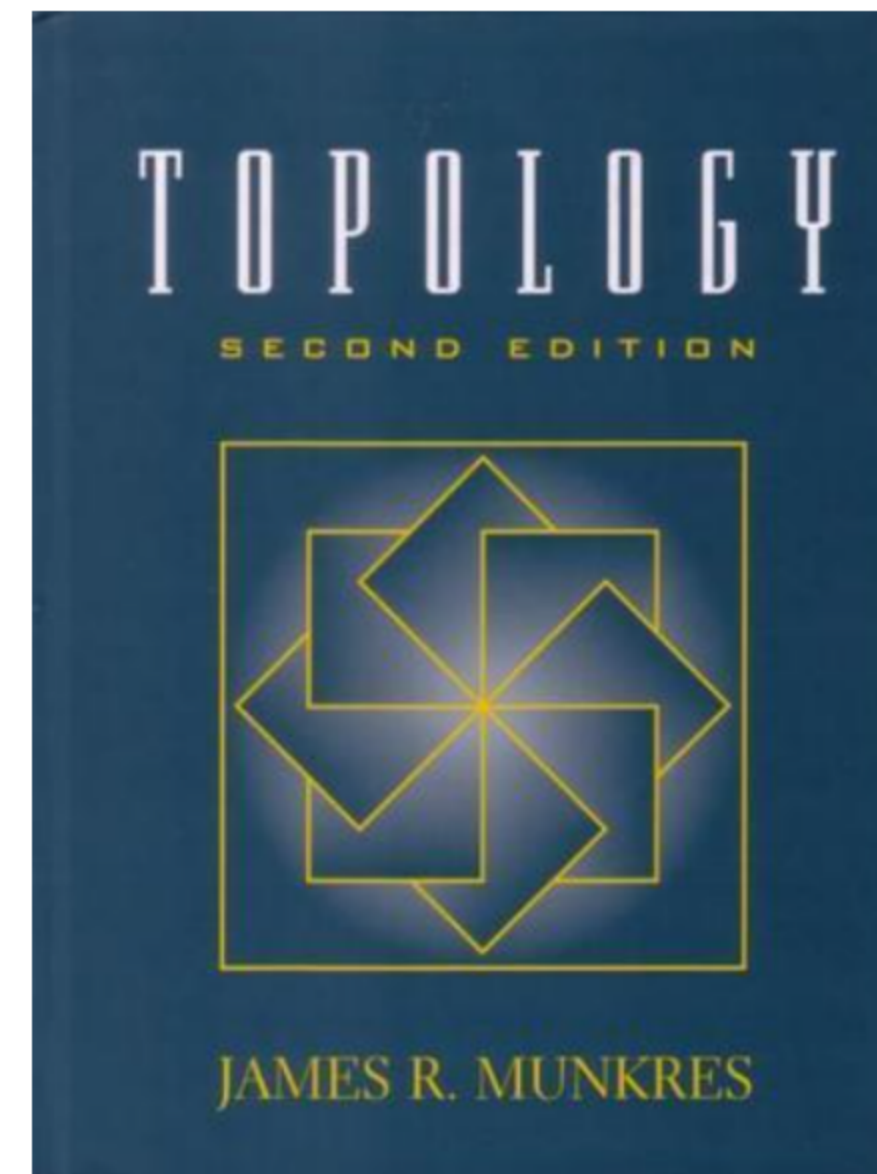
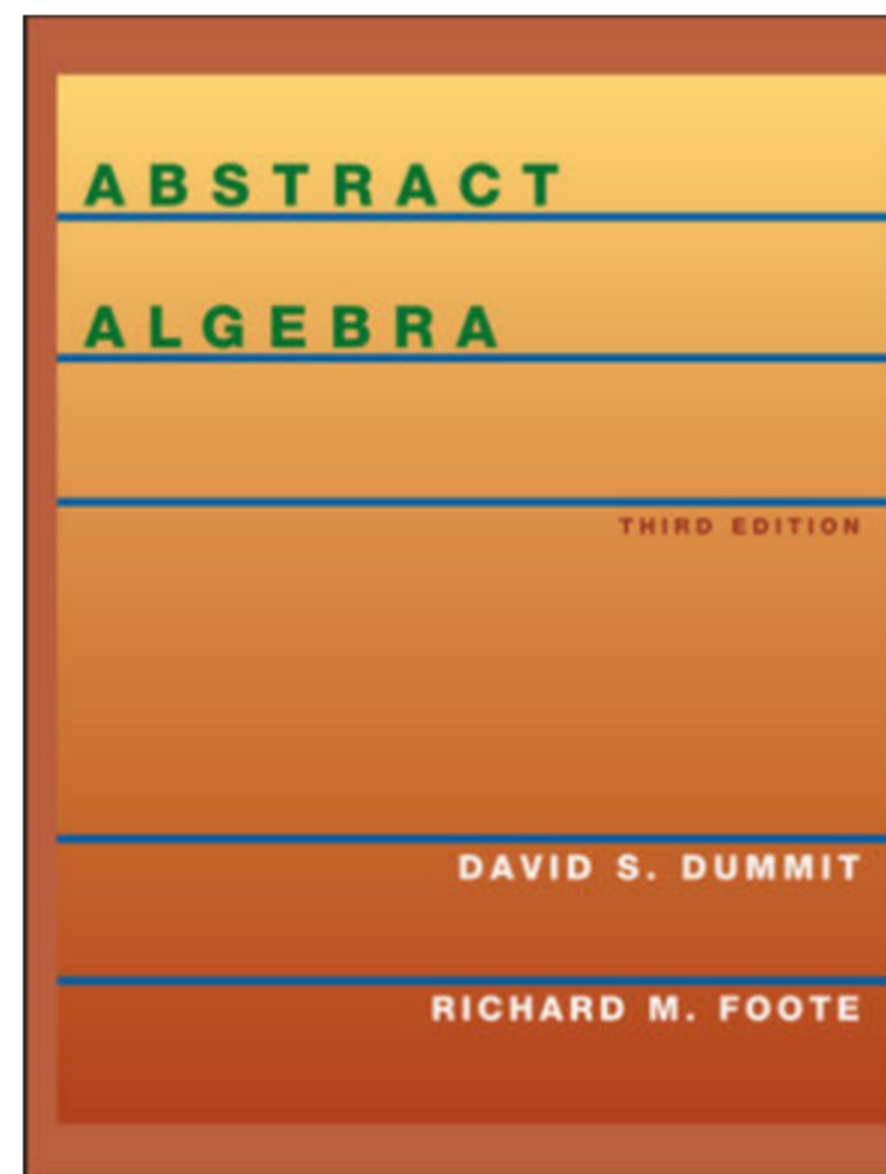
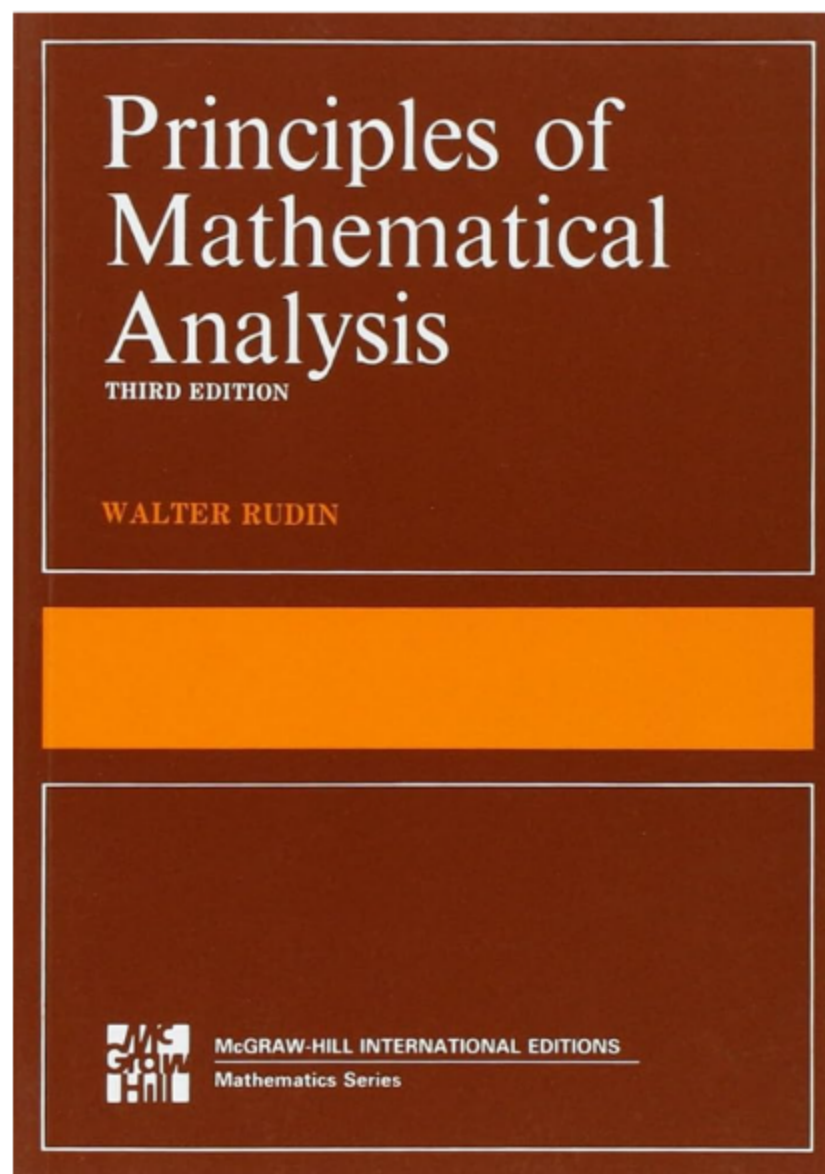
Formalized Statement

Aria–Autoformalizer

- Experiments

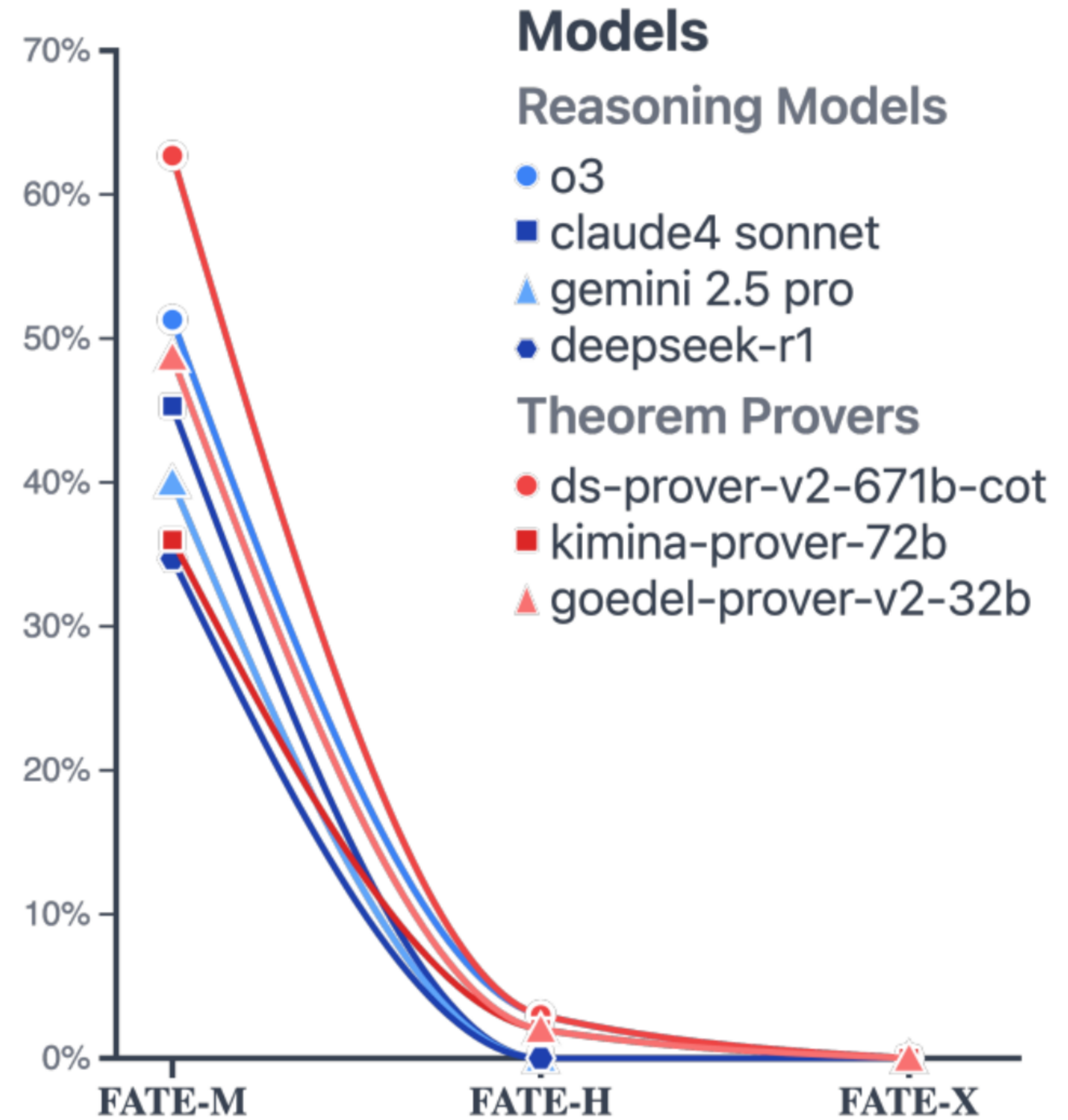
Benchmarks

- **ProofNet:** The problems are primarily drawn from popular undergraduate pure mathematics textbooks and cover topics such as real and complex analysis, linear algebra, abstract algebra, and topology.



Benchmarks

- FATE (Formal Algebra Theorem Evaluation)
- FATE-H, FATE-X





FATE-H



FATE-X

Problem distributions by mathematical topics in FATE-H/X.

Benchmarks

• Conjectures (Homological Conjectures in Commutative Algebra)

7. **The Canonical Element Conjecture.** Let x_1, \dots, x_d be a [system of parameters](#) for R , let F_\bullet be a free R -resolution of the [residue field](#) of R with $F_0 = R$, and let K_\bullet denote the [Koszul complex](#) of R with respect to x_1, \dots, x_d . Lift the identity map $R = K_0 \rightarrow F_0 = R$ to a map of complexes. Then no matter what the choice of system of parameters or lifting, the last map from $R = K_d \rightarrow F_d$ is not 0.
8. **Existence of Balanced Big Cohen–Macaulay Modules Conjecture.** There exists a (not necessarily finitely generated) R -module W such that $m_R W \neq W$ and every system of parameters for R is a regular sequence on W .
9. **Cohen–Macaulayness of Direct Summands Conjecture.** If R is a direct summand of a regular ring S as an R -module, then R is Cohen–Macaulay (R need not be local, but the result reduces at once to the case where R is local).
10. **The Vanishing Conjecture for Maps of Tor.** Let $A \subseteq R \rightarrow S$ be homomorphisms where R is not necessarily local (one can reduce to that case however), with A, S regular and R finitely generated as an A -module. Let W be any A -module. Then the map $\mathrm{Tor}_i^A(W, R) \rightarrow \mathrm{Tor}_i^A(W, S)$ is zero for all $i \geq 1$.
11. **The Strong Direct Summand Conjecture.** Let $R \subseteq S$ be a map of complete local domains, and let Q be a height one prime ideal of S lying over xR , where R and R/xR are both regular. Then xR is a [direct summand](#) of Q considered as R -modules.
12. **Existence of Weakly Functorial Big Cohen–Macaulay Algebras Conjecture.** Let $R \rightarrow S$ be a local homomorphism of complete local domains. Then there exists an R -algebra B_R that is a balanced big Cohen–Macaulay algebra for R , an S -algebra B_S that is a balanced big Cohen–Macaulay algebra for S , and a homomorphism $B_R \rightarrow B_S$ such that the natural square given by these maps commutes.
13. **Serre's Conjecture on Multiplicities.** (cf. [Serre's multiplicity conjectures](#).) Suppose that R is regular of dimension d and that $M \otimes_R N$ has finite length. Then $\chi(M, N)$, defined as the alternating sum of the lengths of the modules $\mathrm{Tor}_i^R(M, N)$ is 0 if $\dim M + \dim N < d$, and is positive if the sum is equal to d . (N.B. [Jean-Pierre Serre](#) proved that the sum cannot exceed d .)
14. **Small Cohen–Macaulay Modules Conjecture.** If R is complete, then there exists a finitely-generated R -module $M \neq 0$ such that some (equivalently every) system of parameters for R is a [regular sequence](#) on M .

Main Results

Table 1: End-to-end auto-formalization results comparing Aria against specialized models. All values are success rates (%); we report Compiler success rate and the stricter Final accuracy (passing both compilation and our AriaScorer semantic check). Results for the Conjectures dataset were manually verified. Kimina’s score on ProofNet is marked due to potential data contamination*.

Method	ProofNet		FATE-H		FATE-X		Conjectures
	Compiler	Final acc.	Compiler	Final acc.	Compiler	Final acc.	
Aria	91.6	68.5	89.0	71.0	69.0	44.0	42.9
Goedel-V2 (pass@16)	–	–	77.0	–	37.0	–	0
Goedel-V2 (pass@32)	–	–	82.0	–	49.0	–	0
Goedel-V2 (pass@64)	–	–	88.0	–	58.0	–	0
Goedel-V2 (pass@128)	–	–	91.0	43.0	63.0	24.0	0
Gemini-2.5-Pro (pass@1)	55.8	27.8	35.0	31.0	27.0	21.0	0
Goedel-V2 (pass@1)	59.6	32.0	35.0	27.0	27.0	16.0	0
Kimina (pass@1)	70.4*	24.7*	10.0	0.0	5.0	1.0	0
Herald (pass@1)	48.5	18.3	24.0	12.0	8.0	5.0	0

* Kimina was trained on the ProofNet dataset, so its reported score may not reflect true generalization capabilities.

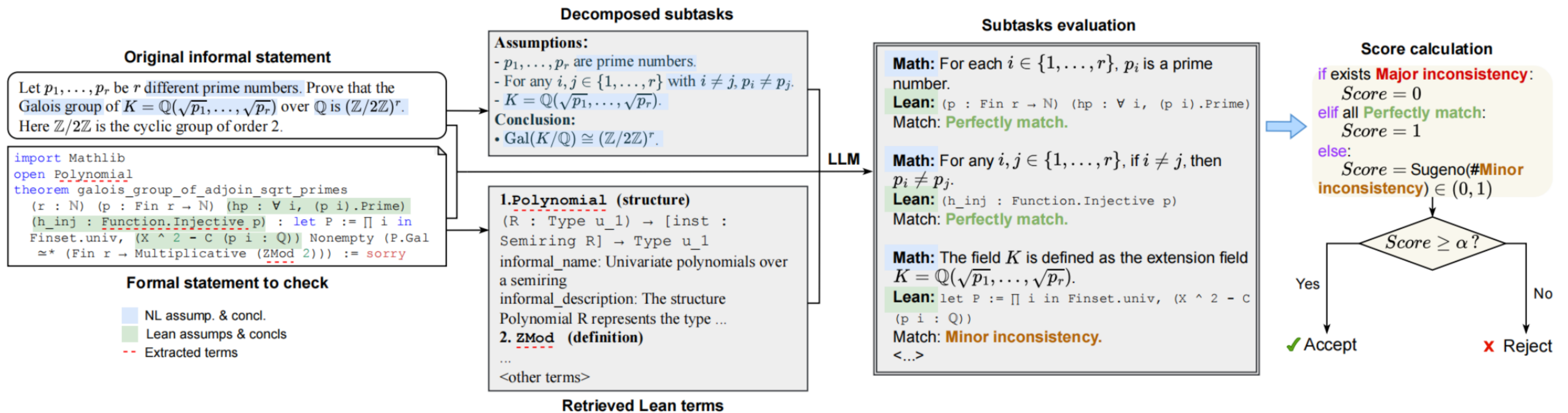
AriaScorer

- Methodology

Solution: A Term–Grounded Semantic Checker

- Subtask Decomposition: Decompose informal statement into atomic assumptions and conclusions.
- Term–level Grounding: Extract Lean terms from formal statements and retrieve their authoritative definitions.
- Matching and Assessment: Assign a consistency label (e.g., Match, Major Inconsistency) to each subtask.
- Score Aggregation: Aggregate labels into a score between 0 and 1.

Solution: A Term-Grounded Semantic Checker



Subtask Decomposition

- More fine-grained filtering of incorrect formal statements.

Original informal statement

Let p_1, \dots, p_r be r different prime numbers. Prove that the Galois group of $K = \mathbb{Q}(\sqrt{p_1}, \dots, \sqrt{p_r})$ over \mathbb{Q} is $(\mathbb{Z}/2\mathbb{Z})^r$. Here $\mathbb{Z}/2\mathbb{Z}$ is the cyclic group of order 2.

Decomposed subtasks

Assumptions:

- p_1, \dots, p_r are prime numbers.
- For any $i, j \in \{1, \dots, r\}$ with $i \neq j$, $p_i \neq p_j$.
- $K = \mathbb{Q}(\sqrt{p_1}, \dots, \sqrt{p_r})$.

Conclusion:

- $\text{Gal}(K/\mathbb{Q}) \cong (\mathbb{Z}/2\mathbb{Z})^r$.

Subtask Decomposition

Let p_1, \dots, p_r be r different prime numbers. Prove that the Galois group of $K = \mathbb{Q}(\sqrt{p_1}, \dots, \sqrt{p_r})$ over \mathbb{Q} is $(\mathbb{Z}/2\mathbb{Z})^r$, here $\mathbb{Z}/2\mathbb{Z}$ is the cyclic group of order 2.

Assumptions:

1. p_1, \dots, p_r are prime numbers.
2. For any $i, j \in \{1, \dots, r\}$ with $i \neq j$, $p_i \neq p_j$.
3. $K = \mathbb{Q}(\sqrt{p_1}, \dots, \sqrt{p_r})$.

Conclusion:

- The Galois group of K over \mathbb{Q} is isomorphic to $(\mathbb{Z}/2\mathbb{Z})^r$. In mathematical notation, this is $\text{Gal}(K/\mathbb{Q}) \cong (\mathbb{Z}/2\mathbb{Z})^r$, where $\text{Gal}(K/\mathbb{Q}) = \{\sigma \in \text{Aut}(K) \mid \forall x \in \mathbb{Q}, \sigma(x) = x\}$.

Term-level Semantic Grounding

- Reasoning over the true meaning of Lean terms rather than their surface meaning.

```
import Mathlib
open Polynomial
theorem galois_group_of_adjoin_sqrt_primes
  (r : ℕ) (p : Fin r → ℕ) (hp : ∀ i, (p i).Prime)
  (h_inj : Function.Injective p) : let P := ∏ i in
  Finset.univ, (X ^ 2 - C (p i : ℚ)) Nonempty (P.Gal
  ≃* (Fin r → Multiplicative (ZMod 2))) := sorry
```

Formal statement to check

- NL assump. & concl.
- Lean assumps & concls
- Extracted terms

```
1. Polynomial (structure)
(R : Type u_1) → [inst :
Semiring R] → Type u_1
informal_name: Univariate polynomials over
a semiring
informal_description: The structure
Polynomial R represents the type ...
2. ZMod (definition)
...
<other terms>
```

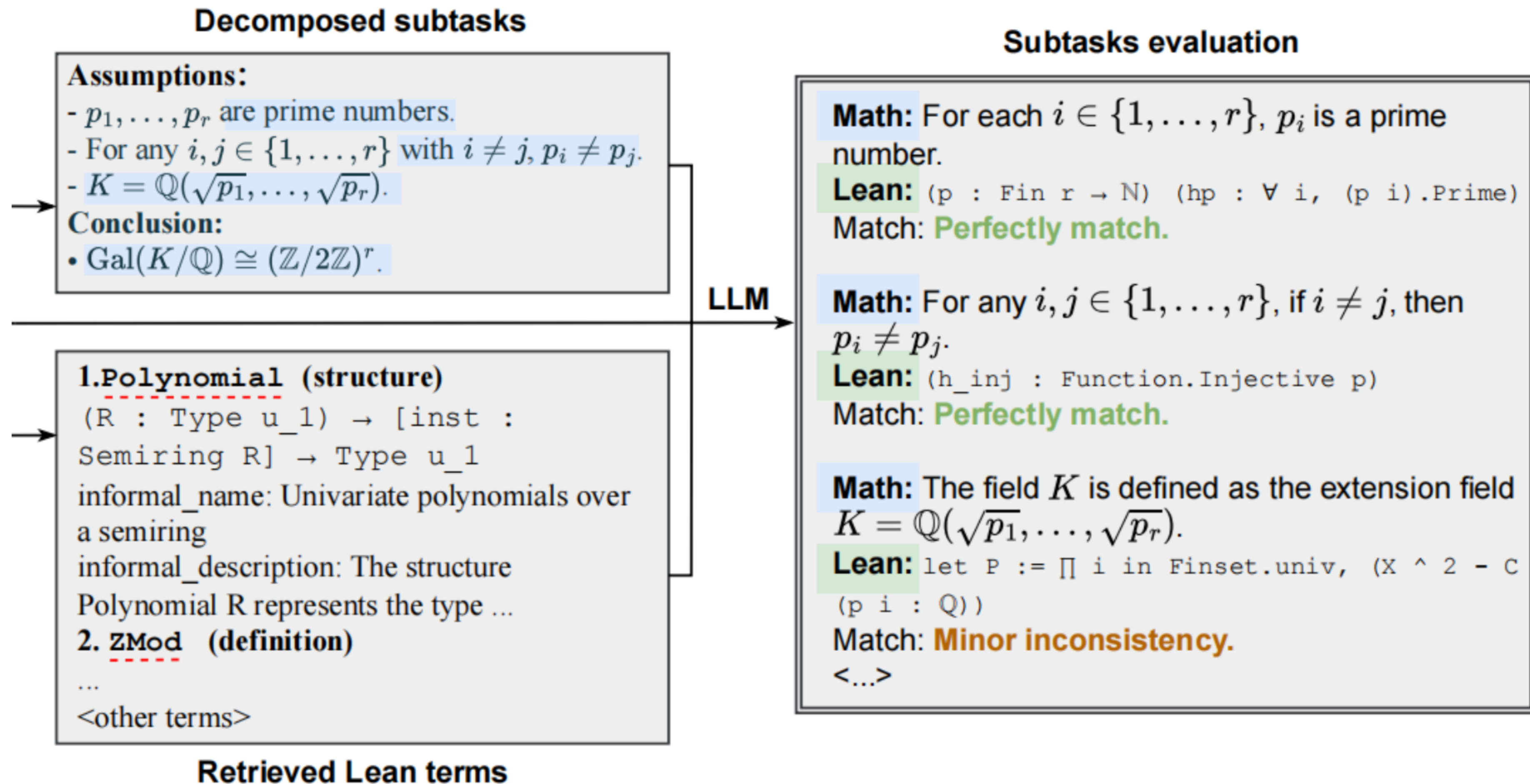
Retrieved Lean terms

Term-level Semantic Grounding

```
[['RingHom'], ['Polynomial', 'sub'], ['Pi', 'instMul'], ['Multiplicative', 'mul'], ['Fin'], ['NormedRing', 'toRing'], ['Semiring', 'toNonAssocSemiring'], ['Rat', 'instNatCast'], ['NonUnitalNonAssocCommRing', 'toNonUnitalNonAssocCommSemiring'], ['HSub', 'hSub'], ['MulOneClass', 'toMul'], ['NormedField', 'toNormedCommRing'], ['Nat', 'Prime'], ['Group', 'toDivInvMonoid'], ['Rat', 'commRing'], ['instHPow'], ['Rat', 'instNormedField'], ['MulEquiv'], ['NonUnitalNonAssocSemiring', 'toDistrib'], ['NonUnitalCommRing', 'toNonUnitalNonAssocCommRing'], ['ZMod', 'commRing'], ['Nonempty'], ['NonUnitalNonAssocCommSemiring', 'toNonUnitalNonAssocSemiring'], ['Polynomial', 'Gal'], ['Polynomial'], ['Polynomial', 'commRing'], ['Nat'], ['DivInvMonoid', 'toMonoid'], ['Rat'], ['Rat', 'semiring'], ['Nat', 'cast'], ['RingHom', 'instFunLike'], ['CommRing', 'toNonUnitalCommRing'], ['Fin', 'fintype'], ['DFunLike', 'coe'], ['instOfNatNat'], ['MonoidWithZero', 'toMonoid'], ['Monoid', 'toMulOneClass'], ['NormedCommRing', 'toNormedRing'], ['Rat', 'instField'], ['ZMod'], ['Finset', 'univ'], ['Distrib', 'toAdd'], ['Polynomial', 'X'], ['Polynomial', 'C'], ['Semiring', 'toMonoidWithZero'], ['Polynomial', 'semiring'], ['HPow', 'hPow'], ['OfNat', 'ofNat'], ['Polynomial', 'Gal', 'instGroup'], ['CommRing', 'toCommMonoid'], ['Finset', 'prod'], ['Monoid', 'toNatPow'], ['Multiplicative'], ['instHSub'], ['Function', 'Injective']]
```

```
{  
  "name": [  
    "ZMod"  
  ],  
  "kind": "definition",  
  "value": "| 0 => ℤ\n | n + 1 => Fin (n + 1)",  
  "informal_name": "Integers modulo  $(n)$  ( $(\mathbb{Z}/n\mathbb{Z})$ )",  
  "informal_description": "The integers modulo  $(n)$ , denoted  $(\mathbb{Z}/n\mathbb{Z})$ , is defined as:\n- The integers  $(\mathbb{Z})$  when  $(n = 0)$ \n- The finite type with  $(n)$  elements (represented as  $(\text{Fin } n)$ ) when  $(n \geq 1)$ "  
}
```

Matching and Assessment



Matching and Assessment

Let's compare the mathematical conditions and conclusions with the Lean 4 formal statement one by one:

1. **For each $i \in \{1, \dots, r\}$, p_i is a prime number:**

- Math: For each $i \in \{1, \dots, r\}$, p_i is a prime number.
- Lean: `(p : Fin r → ℕ) (hp : ∀ i, (p i).Prime)`.
- Match: Perfectly match.

2. **For any $i, j \in \{1, \dots, r\}$, if $i \neq j$, then $p_i \neq p_j$:**

- Math: For any $i, j \in \{1, \dots, r\}$, if $i \neq j$, then $p_i \neq p_j$.
- Lean: `(h_inj : Function.Injective p)`.
- Match: Perfectly match.

3. **The field K is defined as the extension field $K = \mathbb{Q}(\sqrt{p_1}, \dots, \sqrt{p_r})$:**

- Math: The field K is defined as the extension field $K = \mathbb{Q}(\sqrt{p_1}, \dots, \sqrt{p_r})$.
- Lean: `let P := ∏ i in Finset.univ, (X ^ 2 - C (p i : ℚ))`.
- Match: Minor inconsistency.

4. **The base field is \mathbb{Q} :**

- Math: The base field is \mathbb{Q} .
- Lean: The coefficients of the polynomial `P` are defined over the rationals via `C (p i : ℚ)`. The type `P.Gal` is defined as the Galois group over the base field of `P`, which is thus \mathbb{Q} .
- Match: Perfectly match.

5. **The Galois group of the field extension K/\mathbb{Q} , denoted $\text{Gal}(K/\mathbb{Q})$, is isomorphic to the direct product of r copies of the cyclic group of order 2, i.e., $\text{Gal}(K/\mathbb{Q}) \cong (\mathbb{Z}/2\mathbb{Z})^r$:**

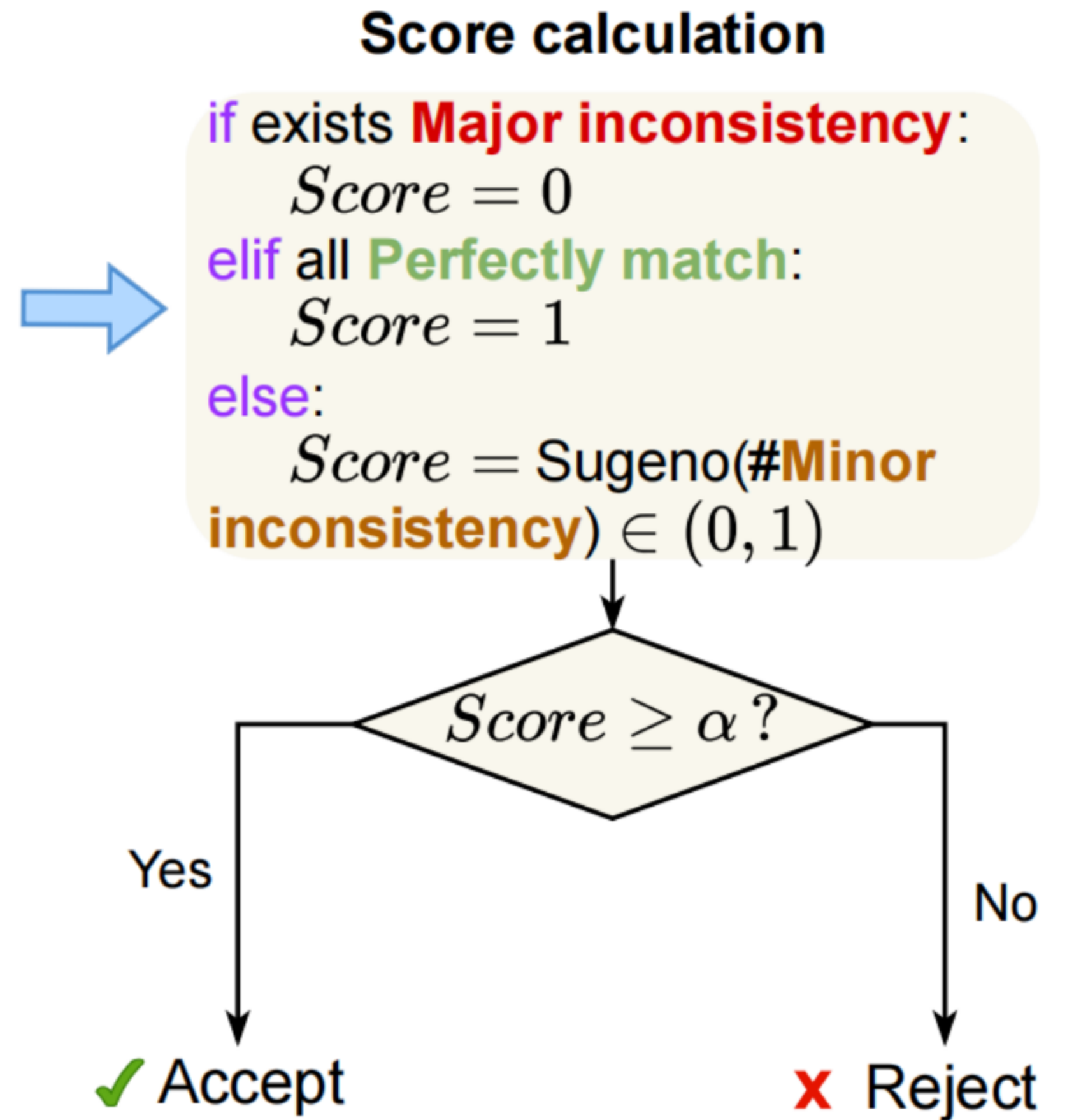
- Math: $\text{Gal}(K/\mathbb{Q}) \cong (\mathbb{Z}/2\mathbb{Z})^r$.
- Lean: `Nonempty (P.Gal ≃* (Fin r → Multiplicative (ZMod 2)))`.
- Match: Minor inconsistency.

Check for missing conditions / implicit conditions:

- The mathematical fact that the field $K = \mathbb{Q}(\sqrt{p_1}, \dots, \sqrt{p_r})$ is the splitting field of the polynomial $P = \prod_{i=1}^r (X^2 - p_i)$ is implicitly used to connect the formal statement about `P.Gal` to the mathematical problem about $\text{Gal}(K/\mathbb{Q})$.

Score Aggregation

- The score decays gradually from 1 as minor inconsistencies accumulate.

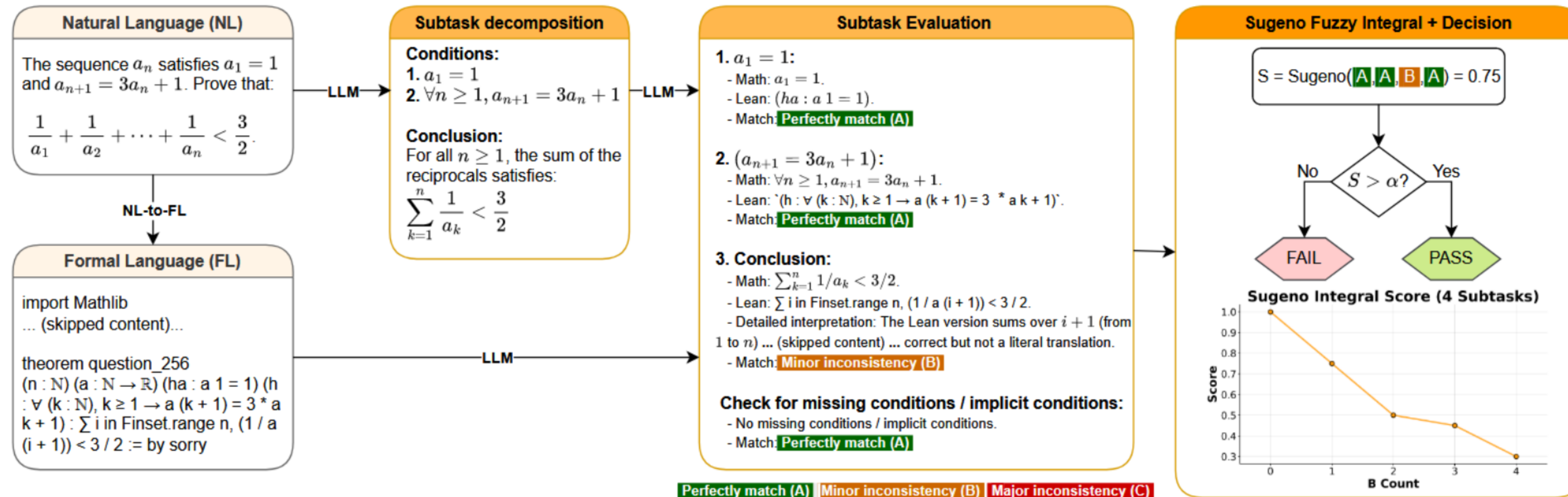


AriaScorer

- Experiments

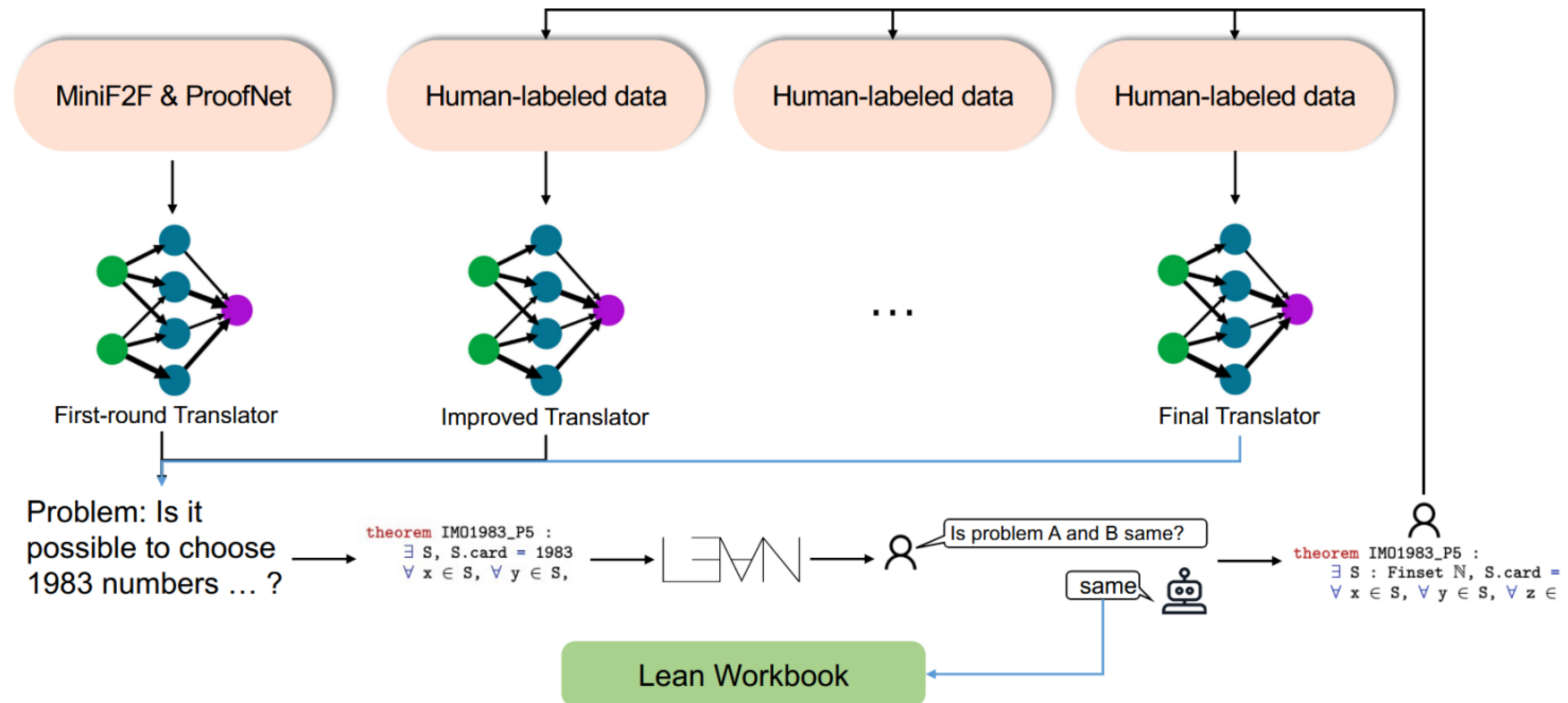
Baselines

- **LeanScorer**: A semantic consistency checker using decomposition and matching, which is an ablation study of AriaScorer without the term-level grounding step.



Baselines

- **Back Translation:** A widely-used pipeline that translates the formal statement back to an informal one and uses an LLM to judge their similarity.



Performance of AriaScorer

	AriaScorer ($\alpha = 0$)	AriaScorer ($\alpha = 0.9$)	LeanScorer ($\alpha = 0$)	LeanScorer ($\alpha = 0.9$)	Back Translation	Gemini
TP	50	42	46	44	7	45
TN	12	15	3	7	16	8
FP	5	2	14	10	1	9
FN	2	10	6	8	45	7
Accuracy	89.9%	82.6%	71.0%	73.9%	33.3%	76.8%
Precision	90.9%	95.5%	77.6%	81.5%	87.5%	83.3%
Recall	96.2%	80.8%	88.5%	84.6%	13.5%	86.5%
F1	93.5%	87.5%	82.1%	83.0%	23.3%	84.9%

Key Findings

- Implicit Semantic Inclusion

2. **R is a Unique Factorization Domain (UFD):**

- **Math:** The ring R is an integral domain, and every non-zero, non-unit element has a unique factorization into irreducible elements (up to order and multiplication by units).
- **Lean:** The conclusion is formalized in two parts. First, `local instance : IsDomain CxyzModKleinian` asserts that the ring is an integral domain. Second, the main theorem `theorem isUFD_of_C_xyz_mod_x2_y3_z7 : UniqueFactorizationMonoid CxyzModKleinian` asserts the property of unique factorization. In Mathlib, a ring is a UFD if and only if it satisfies both `IsDomain` and `UniqueFactorizationMonoid`.
- **Match:** Perfectly match.

Key Findings

- Definition Discrepancy Detection

3. **The multiplication in $D_{A,B,\mathbb{R}}$ is defined by the relations: $i^2 = A$, $j^2 = B$, and $ij = -ji = k$.**

- Math: The multiplication rules are $i^2 = A$ and $j^2 = B$.
- Lean: The multiplication is determined by `QuaternionAlgebra ℝ (A : ℝ) (B : ℝ) 1`.
- According to the provided Mathlib entry, `QuaternionAlgebra R a b c` has multiplication rules $i^2 = a + bi$ and $j^2 = c$. For the Lean statement, `R` is \mathbb{R} , `a` is $(A : \mathbb{R})$, `b` is $(B : \mathbb{R})$, and `c` is 1. This implies the rules are $i^2 = A + Bi$ and $j^2 = 1$. These rules do not match the mathematical rules.
- Match: Major inconsistency.

Key Findings

- Hallucination
Suppression
via
Grounding

↓ with Lean terms

Conclusion: $\text{Gal}(E/\mathbb{Q}) \cong Q_8$:

- Math: The Galois group of the extension E over \mathbb{Q} is isomorphic to Q_8 , the quaternion group of order 8.
- Lean: `Nonempty (P.Gal \simeq^* QuaternionGroup 1)`. `P.Gal` corresponds to $\text{Gal}(E/\mathbb{Q})$. `Nonempty (G \simeq^* H)` means G is isomorphic to H . However, `QuaternionGroup 1` is the generalized quaternion group with $n=1$, which is isomorphic to the cyclic group of order 4 (`C_4`). The quaternion group of order 8, `Q_8`, corresponds to `QuaternionGroup 2`.
- Match: Major Inconsistency.

↓ without Lean terms

Conclusion:

- Math: $\text{Gal}(E/\mathbb{Q}) \cong Q_8$, i.e., the Galois group is isomorphic to the quaternion group.
- Lean: `Nonempty (P.Gal \simeq^* QuaternionGroup 1)`. This states that there exists a group isomorphism between `P.Gal` and `QuaternionGroup 1`. The `Nonempty` wrapper is a standard way in Lean to express existence in a proposition.
- Match: Perfectly match.

Conclusion

Thank You!