
Minif2f in Rocq: Automatic Translation Between Proof Assistants — A Case Study

Jules Vienne

IRIF, Université Paris Cité, Inria, CNRS

Guillaume Baudart

IRIF, Université Paris Cité, Inria, CNRS

Emilio Jesús Gallego Arias

IRIF, Université Paris Cité, CNRS

Marc Lelarge

DI ENS, PSL University, Inria

Abstract

While the Minif2f dataset exists for Lean, Isabelle/HOL, and MetaMath, it has not been formalized in Rocq, limiting cross-system comparisons in automated theorem proving. We investigate whether state-of-the-art LLMs can automatically translate formal theorems between proof assistants. Using a three-stage methodology from basic prompting to multi-turn conversations with error feedback, we successfully translated 478 out of 488 theorems (98%) from Minif2f to Rocq. Expert validation of 150 translations confirmed high accuracy, with only three errors. This work provides a complete Rocq formalization of Minif2f and demonstrates the viability of LLM-based cross-proof-assistant translation.

1 Introduction

Recent advances in Large Language Models (LLMs) have shown remarkable progress in automated theorem proving using interactive theorem provers (ITPs) such as Isabelle [Wu et al., 2022, First et al., 2023], Lean [Polu et al., 2023, Yang et al., 2023], and Rocq [Zhang et al., 2023, Thompson et al., 2024]. However, the landscape of formal mathematics remains fragmented across different proof assistants, each with distinct syntactic conventions, type systems, and mathematical libraries. This fragmentation poses significant barriers to knowledge transfer and comparative evaluation.

The challenge of cross-system compatibility is particularly acute in the evaluation of machine learning approaches to theorem proving. Researchers developing techniques for different proof assistants often work with incompatible datasets, making it difficult to fairly compare methodologies or transfer insights across systems. While manual translation efforts exist, they are time-consuming, error-prone, and do not scale with the growing volume of formalized mathematics.

LLMs have demonstrated particular aptitude for translation tasks between programming languages, especially when extensive shared resources exist [Xu and Zhu, 2022]. This capability suggests potential for automated translation between formal proof languages, which share many structural similarities despite their syntactic differences. Such automated translation could unlock significant value by enabling: (1) fair comparison of automated proving techniques across systems, (2) rapid porting of benchmark datasets, and (3) leveraging the unique strengths of different proof assistants for the same mathematical content.

In this work, we investigate whether state-of-the-art LLMs can effectively translate formal mathematical theorems between proof assistants. We focus specifically on translating the Minif2f dataset [Zheng et al., 2021, Jiang et al., 2022] from its existing formalizations in Lean

and Isabelle to Rocq. Minif2f contains 488 high-school-level mathematical problems with existing formalizations in multiple systems, making it a popular benchmark for evaluating automated proof techniques [Polu and Sutskever, 2020, Thakur et al., 2024, Mikula et al., 2023, Wang et al., 2024].

Our work is available at <https://github.com/LLM4Rocq/miniF2F-rocq>.

2 Methodology

We focus on translating Minif2f to Rocq, as this system lacks a complete formalization despite previous community efforts.¹ The dataset contains 488 theorems spanning various mathematical domains including algebra, number theory, and geometry.

Our translation task generates Rocq theorem statements based on three input sources: (1) natural language descriptions of the mathematical problems, (2) existing Lean formalizations, and (3) existing Isabelle formalizations. We deliberately focus only on theorem statements, ignoring proofs to isolate the translation challenge from proof generation complexity.

All generated Rocq statements are automatically verified using Petanque and its dedicated interface for python pytanque [Teodorescu et al., 2024], a machine-to-machine interactive environment for Rocq. This ensures that our translations are both syntactically and type-theoretically correct within the Rocq system. Then, valid translations are reviewed by a human to ensure their semantic correctness with regards to the three input sources.

We designed a systematic approach with three stages of increasing complexity. To manage computational costs while maximizing translation success, we employ a cascading approach: each stage only processes theorems that remained untranslated in previous stages. This ensures that expensive model calls are focused on the most challenging cases while simpler theorems are handled efficiently in earlier stages.

Stage 1: one-shot prompting In this baseline stage, we provide models with a single prompt containing the natural language description and existing formalizations, requesting a direct Rocq translation. We evaluate four state-of-the-art models: GPT-4o mini (4o mini), Claude-3.5-Sonnet (claude), o1-mini (o1 mini), and o1 (o1). This stage assesses the models’ inherent translation capabilities without interactive refinement.

Stage 2: multi-turn with error feedback Building on Stage 1 failures, we implement an interactive approach where models can attempt up to three translations per theorem. Each subsequent attempt incorporates the error messages from Petanque verification of previous attempts. This stage tests whether models can learn from their mistakes and iteratively improve translations. We focus on claude and o1 mini for this stage based on their Stage 1 performance and cost considerations.

Stage 3: refined prompting with extended attempts For the most challenging remaining theorems, we implement targeted improvements using claude. Based on error analysis from earlier stages, we refine our prompts to specifically address common failures. We progressively increase the number of attempts from 6 to 24, allowing more extensive exploration of the solution space for difficult cases.

3 Results

Figure 1 presents our cumulative translation results across all stages. The progression demonstrates the value of our multi-stage approach.

One-shot prompting in Stage 1 achieved translation rates of up to 68%, showing that models already possess strong base capabilities for translation between proof-assistants. Adding iterative attempts with error feedback in Stage 2 provided significant improvements: claude successfully translated 31% of the theorems remaining after Stage 1, demonstrating that

¹<https://github.com/openai/miniF2F/issues/66>

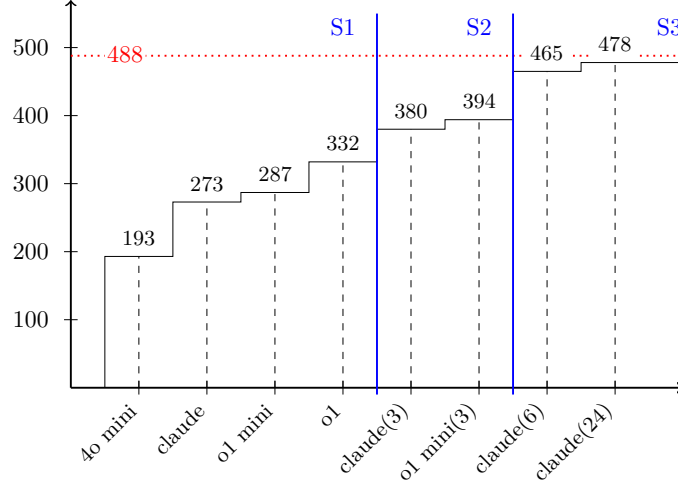


Figure 1: Cumulative translation results for Minif2f to Rocq across all experimental stages. Numbers in parentheses indicate the maximum number of attempts allowed per theorem in multi-turn stages.

models can learn from Rocq error messages. In Stage 3, refining the prompt and increasing the number of attempts yielded the most substantial gains, leaving only ten theorems un-translated. This highlights the importance of providing models with targeted information to overcome their limitations.

Quality assessment To validate translation quality, we conducted an expert audit on a random sample of 150 theorems (approximately 30% of successful translations). Each expert reviewed a batch of 25 translations, comparing them against the natural language description as well as the Lean and Isabelle formalizations.

We classify the answers into three categories:

Error: the translation does not correspond to the original problem. For example, a translation with hypothesis x of type Q of a problem requiring the numerator and denominator of x to be relatively prime is false in Rocq, as this property is not guaranteed for elements of type Q (see also Appendix A.1).

Perfectible: the translation is correct but could be improved. For example, the Rocq statements $\dots (x > 0 \wedge y > 0) \rightarrow \dots$ could be written $\dots x > 0 \rightarrow y > 0 \rightarrow \dots$ (see also Appendix A.2).

Valid: the translation requires no modification.

Table 1: Expert audit results for 150 randomly sampled translations.

Answers	Number of theorems
Error	3
Perfectible	32
Valid	115

Results are presented in Table 1. The low error rate (2%) and high rate of perfect translations (77%) indicate that LLM-based translation can achieve human-level quality for the majority of cases.

4 Discussion

To better understand model capabilities for formal language translation, we now focus on four research questions:

- RQ1:** Do supposedly superior models actually perform better on translation tasks?
RQ2: Does the amount of information available to the model affect its performance?
RQ3: Is the generated code faithful to the existing formalizations?
RQ4: Can a model assess the semantic correctness of translations?

4.1 RQ1: models comparison

To assess whether model rankings correlate with translation performance, we compared 4o mini and o1 mini on a subset of 100 theorems. We evaluated both pass@1 (single attempt, equivalent to Stage 1) and pass@3 (three attempts) scenarios. As for Stage 1, a human ensured the semantic correctness of all proposed translations. Results are presented in Figure 2.

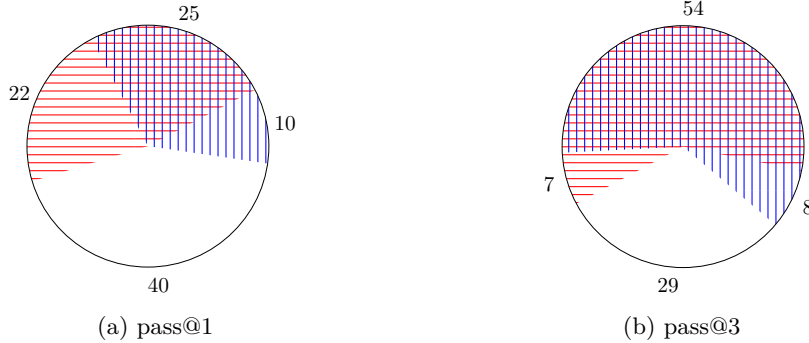


Figure 2: Comparison between 4o mini and o1 mini performance. The circle area amounts to the 100 theorems. Red horizontal lines denote theorems translated by 4o mini. Blue vertical lines denote theorems translated by o1 mini. This defines four zones (untranslated, translated by 4o mini only, translated by o1 mini only, translated by both models).

Despite o1 mini’s chain-of-thought capabilities, 4o mini achieved superior pass@1 performance. However, both models converged to similar performance levels at pass@3, suggesting that superior model architecture does not guarantee better translation performance. Notably, both models tend to succeed and fail on the same theorems.

4.2 RQ2: ablation study

We conducted an ablation study using o1 mini on the same 100 theorems, systematically varying the input information: informal description only, as it is the reference content on which the Rocq version must be based; formal versions only, to test pure translation between proof assistants; Lean version only as Lean is most similar to Rocq; or everything at once (our initial set up). The same methodology as in the models comparison is employed to compute pass@1 and pass@3 performance.

Table 2: Ablation study showing the effect of input information on translation.

Information in the prompt	pass@1	pass@3
informal description + isabelle version + lean version	35%	62%
informal description	43%	65%
isabelle version + lean version	41%	62%
lean version	40%	56%

Results are presented in Table 2. Surprisingly, varying the input information does not substantially influence performance. Providing only the informal description achieved the best performance, suggesting that natural language descriptions constitute the most crucial information for models, while additional formal representations may introduce confusion rather than clarity.

4.3 RQ3: faithfulness

When evaluating semantic correctness of translations, we observed that a formalization can be valid for a theorem prover while failing to capture the complete intent of the natural language problem. For example, when expressing that m is the maximum of a function f , formalizing only that f is bounded above by m is insufficient, the statement must also ensure that this bound is attained.

When both Lean and Isabelle formalizations were provided, mismatches originated from these reference versions (e.g., the maximum example above; see also Appendix A.3) in all but one case. This indicates that residual inaccuracies exist in the original Minif2f formalizations, likely due to human error, and that our translated dataset achieves quality comparable to the original versions. To investigate these discrepancies, we analyzed results from the ablation study, focusing on the two prompting strategies containing informal descriptions: informal description only versus everything (Lean, Isabelle, and informal description).

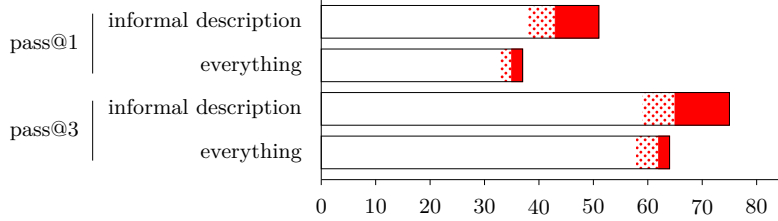


Figure 3: Effect of input information on translation faithfulness. Black bars represent errors, dotted bars indicate faithfulness errors, and white bars show valid translations.

Figure 3 reveals that prompting models with only informal descriptions produces more Rocq-accepted theorems but exhibits a higher overall error rate compared to including both Lean and Isabelle formalizations. Additionally, faithfulness errors constitute a larger proportion of total errors when formal versions are provided in the prompt.

4.4 RQ4: LLM-as-a-judge

To assess whether an LLM can perform semantic verification in place of human reviewers, we compared model judgments against human evaluations from the RQ1 and RQ2 experiments. For this task, we selected DeepSeek R1, a model distinct from those used for translation and known for strong reasoning capabilities. Detailed results are provided in Appendix A.4.

DeepSeek R1 and human reviewers agreed on 95.2% of translations. However, the model demonstrated limited accuracy in error detection: it failed to recognize 41.7% of errors identified by human reviewers. Since the verification step aims to identify semantic errors, DeepSeek R1’s poor error identification performance indicates that human review remains necessary for this task.

5 Conclusion

We successfully translated 478 of 488 theorems (98%) from the Minif2f dataset to Rocq using state-of-the-art LLMs, providing the first complete Rocq formalization of this important benchmark. Our three-stage methodology demonstrates that interactive approaches with error feedback substantially improve one-shot translation, with expert validation confirming high translation quality (only 2% error rate). This work establishes LLM-based translation as a viable approach for translation between proof assistants.

Acknowledgements The research leading to these results has received funding from the Inria “Défi” LLM4Code, and the European Research Council (ERC) under the European Union’s Ninth Framework Programme Horizon Europe (ERC Synergy Project Malinca, Grant Agreement n.101167526).

References

- Yuhuai Wu, Albert Qiaoju Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models. In *NeurIPS*, 2022.
- Emily First, Markus N. Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-proof generation and repair with large language models. *CoRR*, abs/2303.04910, 2023. doi: 10.48550/arXiv.2303.04910. URL <https://doi.org/10.48550/arXiv.2303.04910>.
- Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal mathematics statement curriculum learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL <https://openreview.net/pdf?id=-P7G-8dmSh4>.
- Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. Leandro: Theorem proving with retrieval-augmented language models. *CoRR*, abs/2306.15626, 2023.
- Shizhuo Dylan Zhang, Talia Ringer, and Emily First. Getting more out of large language models for proofs. *CoRR*, abs/2305.04369, 2023.
- Kyle Thompson, Nuno Saavedra, Pedro Carrott, Kevin Fisher, Alex Sanchez-Stern, Yuriy Brun, João F. Ferreira, Sorin Lerner, and Emily First. Rango: Adaptive retrieval-augmented proving for automated software verification. *CoRR*, abs/2412.14063, 2024.
- Yichen Xu and Yanqiao Zhu. A survey on pretrained language models for neural code intelligence. *CoRR*, abs/2212.10079, 2022.
- Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021.
- Albert Q. Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. In *Submitted to The Eleventh International Conference on Learning Representations*, 2022. URL <https://arxiv.org/abs/2210.12283>.
- Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.
- Amitayush Thakur, George Tsoukalas, Yeming Wen, Jimmy Xin, and Swarat Chaudhuri. An in-context learning agent for formal theorem-proving. In *First Conference on Language Modeling*, 2024.
- Maciej Mikula, Szymon Tworowski, Szymon Antoniak, Bartosz Piotrowski, Albert Qiaoju Jiang, Jin Peng Zhou, Christian Szegedy, Łukasz Kuciński, Piotr Miłoś, and Yuhuai Wu. Magnushammer: A transformer-based approach to premise selection. *arXiv preprint arXiv:2303.04488*, 2023.
- Haiming Wang, Huajian Xin, Zhengying Liu, Wenda Li, Yinya Huang, Jianqiao Lu, Zhicheng Yang, Jing Tang, Jian Yin, Zhenguo Li, et al. Proving theorems recursively. *arXiv preprint arXiv:2405.14414*, 2024.
- Laetitia Teodorescu, Guillaume Baudart, Emilio Jesús Gallego Arias, and Marc Lelarge. Nlir: Natural language intermediate representation for mechanized theorem proving. In *MathAI@NeurIPS*, 2024.

A Appendix

A.1 Example of an answer with an *Error* and its correction

The natural language statement of the problem is: The number $a = \frac{p}{q}$, where p and q are relatively prime positive integers, has the property that the sum of all real numbers x satisfying $\lfloor x \rfloor \cdot \{x\} = a \cdot x^2$ is 420, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x and $\{x\} = x - \lfloor x \rfloor$ denotes the fractional part of x . What is $p + q$? Show that it is 929.

Theorem amc12a_2020_p25 :

```
forall (a : Q),

  forall (S : list R),
    (forall x : R, In x S <=>
      (IZR (Int_part x) *
       (x - IZR (Int_part x))
       = Q2R a * Rpower x 2))
  -> NoDup S
  -> fold_left Rplus S 0 = 420
  -> (Z.pos (Qden a) + Qnum a = 929)%Z.
```

(a) Rocq formalization before the audit.

Theorem amc12a_2020_p25 :

```
forall (p q : nat),
  Nat.gcd p q = 1%nat ->
  forall (S : list R),
    (forall x : R, In x S <=>
      (IZR (Int_part x) *
       (x - IZR (Int_part x))
       = INR p / INR q * Rpower x 2))
  -> NoDup S
  -> fold_left Rplus S 0 = 420
  -> (p + q = 929)%nat.
```

(b) Rocq formalization after the audit.

Example 1: an answer with an *Error* and its correction: a is replaced by the ratio of its numerator p and its denominator q , and a hypothesis ensuring they are relatively prime is added.

A.2 Example of a *Perfectible* problem and its editing

The natural language statement of the problem is: Let a and b be two positive real numbers, and n be a positive integer. Show that $\left(\frac{a+b}{2}\right)^n \leq \frac{a^n + b^n}{2}$.

Theorem

```
algebra_apbon2pownleqapownpbpowon2 :
  forall (a b : R) (n : nat),
    0 < a & 0 < b ->

    (0 < n)%nat ->
    ((a + b) / 2)^n
    <= (a ^ n + b ^ n) / 2.
```

(a) Rocq formalization before the audit.

Theorem

```
algebra_apbon2pownleqapownpbpowon2 :
  forall (a b : R) (n : nat),
    0 < a ->
    0 < b ->
    (0 < n)%nat ->
    ((a + b) / 2)^n
    <= (a ^ n + b ^ n) / 2.
```

(b) Rocq formalization after the audit.

Example 2: a *Perfectible* problem and its editing: the conjunction $a < 0 \wedge b < 0$ is curried into two separate hypotheses $a < 0$ and $b < 0$.

A.3 Example of an answer that is not *faithful* and its adjusment

The natural language statement of the problem is: What is the maximum value of $\frac{(2^t - 3t)t}{4^t}$ for real values of t ? Show that it is $\frac{1}{12}$.

This additional statement is not present in the Lean version² where only a proof for the upper bound is required:

theorem amc12b_2020_p22

```
(t : ℝ) :
  ((2^t - 3 * t) * t) / (4^t) ≤ 1 / 12 :=
...
```

²Lean formalization from <https://github.com/facebookresearch/minif2f>

```

Theorem amc12b_2020_p22 :
  forall t : R,
    ((exp (t * ln 2) - 3 * t) * t)
      / (exp (t * ln 4))
    <= 1 / 12.

```

(a) Rocq formalization before the audit.

```

Theorem amc12b_2020_p22 :
  forall t : R,
    ((exp (t * ln 2) - 3 * t) * t) /
      (exp (t * ln 4))
    <= 1 / 12
  ^
  exists t : R,
    ((exp (t * ln 2) - 3 * t) * t) /
      (exp (t * ln 4))
    = 1 / 12.

```

(b) Rocq formalization after the audit.

Example 3: An answer the is not *faithful* and its adjustment: an additional proof that the maximum is indeed reached is required.

We consider it as an *Faithfulness* issue: the formal statement in Lean or answered (before the audit) is valid but not as strong as the informal statement.

Here is another example where the natural language statement is: Solve the system of equations

$$\begin{aligned}
 |a_1 - a_2|x_2 + |a_1 - a_3|x_3 + |a_1 - a_4|x_4 &= 1 \\
 |a_2 - a_1|x_1 + |a_2 - a_3|x_3 + |a_2 - a_4|x_4 &= 1 \\
 |a_3 - a_1|x_1 + |a_3 - a_2|x_2 + |a_3 - a_4|x_4 &= 1 \\
 |a_4 - a_1|x_1 + |a_4 - a_2|x_2 + |a_4 - a_3|x_3 &= 1
 \end{aligned}$$

where a_1, a_2, a_3, a_4 are four different real numbers.

In this case, the formalization process requires to have a look at the solution. However, the informal proof (in <https://github.com/facebookresearch/miniF2F>) assumes that $a_1 > a_2 > a_3 > a_4$ and shows that in this case, $x_2 = x_3 = 0$, and $x_1 = x_4 = 1/(a_1 - a_4)$. This informal proof only solves a particular case. It turns out that the general solution can be written as follows: define $m = \arg \max_i a_i$ and $n = \arg \min_i a_i$, then $x_m = x_n = \frac{1}{a_m - a_n}$ and for all $i \neq n, m$, $x_i = 0$.

The Lean formalization relies on the weak informal proof and it is acknowledged in the file that this formal statetment is weaker than the informal original problem:

```

-- Solution encoded in the theorem statement.
-- Conclusion too weak. It doesn't show "if and only if"
theorem imo_1966_p5 (x a : ℕ → ℝ) (h₀ : a 1 ≠ a 2) (h₁ : a 1 ≠ a 3) (h₂ : a 1 ≠ a 4)
  (h₃ : a 2 ≠ a 3) (h₄ : a 2 ≠ a 4) (h₅ : a 3 ≠ a 4) (h₆ : a 1 > a 2) (h₇ : a 2 > a 3)
  (h₈ : a 3 > a 4)
  (h₉ : abs (a 1 - a 2) * x 2 + abs (a 1 - a 3) * x 3 + abs (a 1 - a 4) * x 4 = 1)
  (h₁₀ : abs (a 2 - a 1) * x 1 + abs (a 2 - a 3) * x 3 + abs (a 2 - a 4) * x 4 = 1)
  (h₁₁ : abs (a 3 - a 1) * x 1 + abs (a 3 - a 2) * x 2 + abs (a 3 - a 4) * x 4 = 1)
  (h₁₂ : abs (a 4 - a 1) * x 1 + abs (a 4 - a 2) * x 2 + abs (a 4 - a 3) * x 3 = 1) :
  x 2 = 0 ∧ x 3 = 0 ∧ x 1 = 1 / abs (a 1 - a 4) ∧ x 4 = 1 / abs (a 1 - a 4) := by
  sorry

```

Thanks to our audit, we were able to get a stronger formal version closer to the original informal statement.

```

Theorem imo_1966_p5' :
  forall (m n : nat) (x a : nat -> R),
    (forall i j, a i = a j -> i = j) ->
    (Rabs (a 1%nat - a 2%nat) * x 2%nat + Rabs (a 1%nat - a 3%nat) * x 3%nat
      + Rabs (a 1%nat - a 4%nat) * x 4%nat = INR 1) ->
    (Rabs (a 2%nat - a 1%nat) * x 1%nat + Rabs (a 2%nat - a 3%nat) * x 3%nat
      + Rabs (a 2%nat - a 4%nat) * x 4%nat = INR 1) ->
    (Rabs (a 3%nat - a 1%nat) * x 1%nat + Rabs (a 3%nat - a 2%nat) * x 2%nat
      + Rabs (a 3%nat - a 4%nat) * x 4%nat = INR 1) ->

```

```

(Rabs (a 4%nat - a 1%nat) * x 1%nat + Rabs (a 4%nat - a 2%nat) * x 2%nat
+ Rabs (a 4%nat - a 3%nat) * x 3%nat = INR 1) ->
(1 <= m <= 4)%nat -> (1 <= n <= 4)%nat ->
(forall i : nat, a m >= a i) ->
(forall i, a n <= a i) ->
(x m = 1 / (a m - a n)) ∧ (x n = x m)
∧ (forall i : nat , (i<=4)%nat -> i <> m -> i <> n -> x i = R0).

```

A.4 Complete report of research questions results

Tables on the following pages present all detailed results computed in RQ1 and RQ2, listed by theorem. All experimental configurations are represented: 4o mini versus o1 mini comparisons, and various prompt information conditions for o1 mini. For each configuration, we computed pass@3 results, generating three translations per theorem. Within each table cell, results for the three translation attempts are presented and separated by blank spaces. A dash indicates that the model failed to produce a valid Rocq statement. For translations that successfully type-checked, results from the semantic verification phase are shown. A V indicates a valid statement, a F denotes a faithfulness error, and a E represents an error. Human reviews are displayed in black, while DeepSeek R1 answers are shown in gray. The reviews where the assessments of humans and DeepSeek R1 diverged are highlighted in bold (F classifications by humans are considered valid for DeepSeek R1, as they align with the Lean and Isabelle formalizations in most cases).

Theorems	4o mini	o1 mini			
	everything	everything	formal versions	lean version	informal description
aime_1983_p1	-- -- VV	-- -- --	-- -- --	-- -- --	-- -- --
aime_1990_p4	-- VV VV	-- VV --	VV VV VV	VV VE --	EE VV VV
aime_1991_p6	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
aimeII_2001_p3	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
algebra_3rootspoly_...	-- -- VV	VV -- VV	VV VV VV	VV VV VV	VV -- VV
algebra_9onxpyzleq_...	VV VV VV	VV VV --	-- VV VV	-- -- --	-- VV VV
algebra_apb4leq8ta4pb4	VV VV VV	VV VV VV	VV -- VV	-- VV VV	VV VV VV
algebra_others_exir_...	-- -- --	-- -- --	-- -- --	EE -- --	-- -- --
algebra_sqineq_at2m_...	VV VV VV	VV VV VV	VV -- VV	VV VV VV	VV VV VV
amc12_2000_p6	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
amc12_2001_p9	VV -- VV	VV VV VV	VV VV VV	-- -- VV	VV VV VV
amc12a_2002_p21	FV -- --	-- -- --	FE FE FE	FV -- --	-- -- --
amc12a_2003_p5	-- -- EE	VV VV VV	-- -- --	EV -- --	-- -- EV
amc12a_2008_p15	VV VV VV	-- VV VV	VV -- VV	VV VV VV	VV -- VV
amc12a_2009_p2	EV -- VV	-- VV VV	VV -- VV	VV VV EV	VV VV VV
amc12a_2009_p9	VV VV VV	-- -- --	VV -- --	VV -- --	VV VV VV
amc12a_2010_p11	-- -- --	-- -- --	-- -- VV	-- -- --	-- -- --
amc12a_2013_p7	-- -- --	-- -- --	-- -- --	-- -- --	-- -- VV
amc12a_2013_p8	VV -- --	VV VV VV	-- -- --	-- VV --	-- VV VV
amc12a_2017_p2	VV VV VV	VV VV --	VV VV VV	-- -- --	-- VV VV
amc12a_2020_p21	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
amc12a_2020_p25	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
amc12a_2021_p8	-- -- --	-- -- VV	-- -- --	-- -- --	-- EE --
amc12b_2003_p9	-- VV VV	-- VV VV	-- -- VV	-- -- --	VV -- --
amc12b_2004_p3	VV VV VV	VV VV VV	-- VV VV	VV -- VV	-- VV VV
amc12b_2020_p13	-- -- --	-- -- --	-- -- --	-- VV --	-- -- --
amc12b_2020_p22	-- -- FV	-- FV --	-- -- --	-- -- --	VV -- --
amc12b_2021_p18	-- -- --	-- -- --	-- VV VV	-- -- --	-- -- --
imo_1977_p5	-- -- --	-- -- --	VV -- --	-- -- --	EV -- EE
imo_1977_p6	VV -- VV	VV VV VV	-- -- VV	VV -- --	EE EE EE
imo_1981_p6	FV FV --	-- FV --	VV FV FV	FV FV --	-- -- --
imo_1997_p5	VV -- VV	VV -- --	VV -- --	VV -- VV	VV -- VV
imo_2001_p6	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
imosl_2007_algebra_p6	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
induction_nfactltne_...	-- -- --	-- VV VV	-- VV --	-- -- --	-- -- VV
induction_seq_mul2pnp1	VV VV VV	-- VV VV	-- -- --	-- -- --	VV -- --
induction_sum_1oktkp1	-- -- --	-- -- --	-- -- --	-- -- VV	-- -- --
mathd_algebra_13	VV VV VV	-- VV VV	VV -- --	VV -- --	VV VV EE
mathd_algebra_15	VV -- VV	-- VV VV	-- -- --	VV -- --	VV -- --
mathd_algebra_24	VV VV VV	-- VV VV	VV VV VV	VV VV VV	VV VV VV
mathd_algebra_48	-- VV VV	-- VV VV	VV VV VE	VV VE VV	VV VV --
mathd_algebra_51	VV VV VV	-- -- --	-- VV VV	VV -- VV	VV VV VV
mathd_algebra_67	VV VV VV	VV VV --	VV -- VV	VV -- --	VV -- --
mathd_algebra_77	VV VV --	-- VV VV	VV VV --	-- -- VV	VV VV VV
mathd_algebra_104	VV VV VV	VV -- VV	VV VV VV	VV VE VV	VV VV VV
mathd_algebra_107	FV FV FV	VV FV FV	FV FV --	FV FV FE	FV FV EV
mathd_algebra_119	VV VV --	VV VV VV	VV VV --	VV VV VV	VV VV VV
mathd_algebra_123	VV VE VV	VV -- VV	VV VV VV	VV VV VV	EV VV VV
mathd_algebra_131	VV VV VV	VV VV VV	-- -- --	-- -- VV	EV -- EV
mathd_algebra_149	-- -- --	-- -- --	-- -- --	-- -- --	EE -- --

Theorems	4o mini	o1 mini			
	everything	everything	formal versions	lean version	informal description
mathd_algebra_153	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
mathd_algebra_158	-- -- --	-- -- EE	-- -- --	-- -- --	VV VV FV
mathd_algebra_188	-- EE --	EE -- --	-- -- --	-- -- VV	VV VV --
mathd_algebra_192	-- -- VV	VV VV VV	VV -- VV	VV -- VV	-- VV --
mathd_algebra_209	-- -- EE	VV -- EV	-- -- --	-- VV VV	VV -- VV
mathd_algebra_263	VV -- --	VV VV VV	VV VV VV	VV VV VV	-- VV VV
mathd_algebra_282	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
mathd_algebra_304	-- VE VV	-- -- --	VV VV VV	VV VV VV	VV VV --
mathd_algebra_320	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
mathd_algebra_405	FV -- --	-- -- --	-- -- VV	-- -- EE	-- -- FV
mathd_algebra_410	FV -- --	FV FV FV	FV -- FV	FV -- --	-- VV --
mathd_algebra_419	VV VV VV	VV VV VV	-- VV VV	VV VV VE	VV VV VV
mathd_algebra_440	VV VV VV	VV VV VV	VV VV VV	VV VV VV	VV VV VV
mathd_algebra_455	-- VV VV	VV VV VV	VV -- --	VV -- VV	-- VV VV
mathd_algebra_487	VV VV --	VV VV --	VV VV VV	VV VV VV	FV EV FV
mathd_algebra_493	VV VV VV	-- VV VV	-- VV VV	VV -- --	VV VV VV
mathd_algebra_509	-- VV VV	VV -- VV	VV VV VV	-- VV VV	-- VV VV
mathd_algebra_513	VV -- --	-- VV VV	VV VV VV	VV VV VV	VV VV VV
mathd_numbertheory_22	EE -- --	FV FV FV	FV FV FV	-- -- --	-- EE --
mathd_numbertheory_24	-- -- --	VV VV VV	-- -- --	VV -- --	-- VV --
mathd_numbertheory_42	VV -- VV	-- VV VV	-- -- --	-- -- --	-- EE --
mathd_numbertheory_45	VV VV --	-- -- VV	-- VV VV	-- -- VV	-- VV VV
mathd_numbertheory_64	-- -- --	-- VV VV	VV -- --	-- -- --	-- VV VV
mathd_numbertheory_127	-- FV VV	VV -- FV	VV VV --	-- FV --	FV FV FV
mathd_numbertheory_149	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
mathd_numbertheory_150	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
mathd_numbertheory_175	VV VV VV	-- VV --	-- VV VV	-- VV VV	VV VV VV
mathd_numbertheory_185	-- VV VV	VV VV --	VV VV VV	VV -- --	VV VV --
mathd_numbertheory_188	VV VV --	VV -- --	VV VV --	-- -- --	-- VV --
mathd_numbertheory_207	VV VV VV	-- -- VV	VV VV --	VV -- VV	VE VV VV
mathd_numbertheory_212	VV VV VV	-- VV VV	-- VV VV	VV VV VV	VV VV VV
mathd_numbertheory_221	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
mathd_numbertheory_234	VV -- --	VV VV --	-- -- VV	VV -- VV	VV VV --
mathd_numbertheory_252	-- -- --	-- VV VV	VV VV --	-- -- --	VV VV --
mathd_numbertheory_293	-- -- --	-- -- --	-- -- --	-- -- --	EV VV --
mathd_numbertheory_296	EE FV --	EE VV FV	-- VV --	-- -- --	VV VV --
mathd_numbertheory_299	VV VV VV	-- -- VV	-- VV VV	VV VV VV	VV VV VV
mathd_numbertheory_321	VE VE VE	VV -- VV	-- VV VV	-- EV --	FV -- --
mathd_numbertheory_328	VV VV VV	-- -- VV	-- VV VV	VV VV --	VV -- VV
mathd_numbertheory_342	VV -- VV	-- -- --	VV VV --	-- VV VV	-- VV --
mathd_numbertheory_543	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
mathd_numbertheory_552	-- -- --	-- -- --	-- -- --	-- -- --	EV EE --
mathd_numbertheory_629	-- -- --	-- -- --	VV VV --	-- -- --	-- -- FE
mathd_numbertheory_640	VV VV VV	VV VV --	-- VV --	VV VV --	VV VV VV
mathd_numbertheory_765	FV FV FV	-- VV VE	-- -- --	-- -- FV	FV VV --
numbertheory_2pownm ...	-- -- --	-- -- --	-- -- --	-- -- --	-- -- --
numbertheory_exk2po ...	VV -- --	VV -- --	-- VV VV	VV -- VV	-- -- --
numbertheory_fxeq4p ...	-- -- --	-- -- --	-- -- --	-- -- --	-- VV VV
numbertheory_notequ ...	-- -- --	-- -- --	VV -- VV	-- -- --	-- -- EV
numbertheory_sumkmu ...	-- -- --	-- -- --	-- -- --	-- -- VV	-- -- --