FVEL: Interactive Formal Verification Environment with Large Language Models via Theorem Proving

Xiaohan Lin*[1], Qingxing Cao*[1], Yinya Huang*[2], Haiming Wang*[3], Jianqiao Lu[4], Zhengying Liu[5], Linqi Song[2], Xiaodan Liang[1,6]

[1]Shenzhen Campus of Sun Yat-sen University, [2]City Univeristy of Hong Kong,
[3]Sun Yat-sen University, [4]The University of Hong Kong
[5]Huawei Noah's Ark Lab [6]DarkMatter AI Research

FVEL: Motivation and Contributions

Motivation:

- **Formal verification*** has witnessed growing significance with emerging **program synthesis** by the evolving large language models.
- Current formal verification mainly resorts to **symbolic verifiers** or **hand-craft rules**, resulting in limitations for **extensive and flexible** verification.
- To utilizes the LLMs' ability of theorem proving for **rigorous and interactive** formal verification.

Contributions:

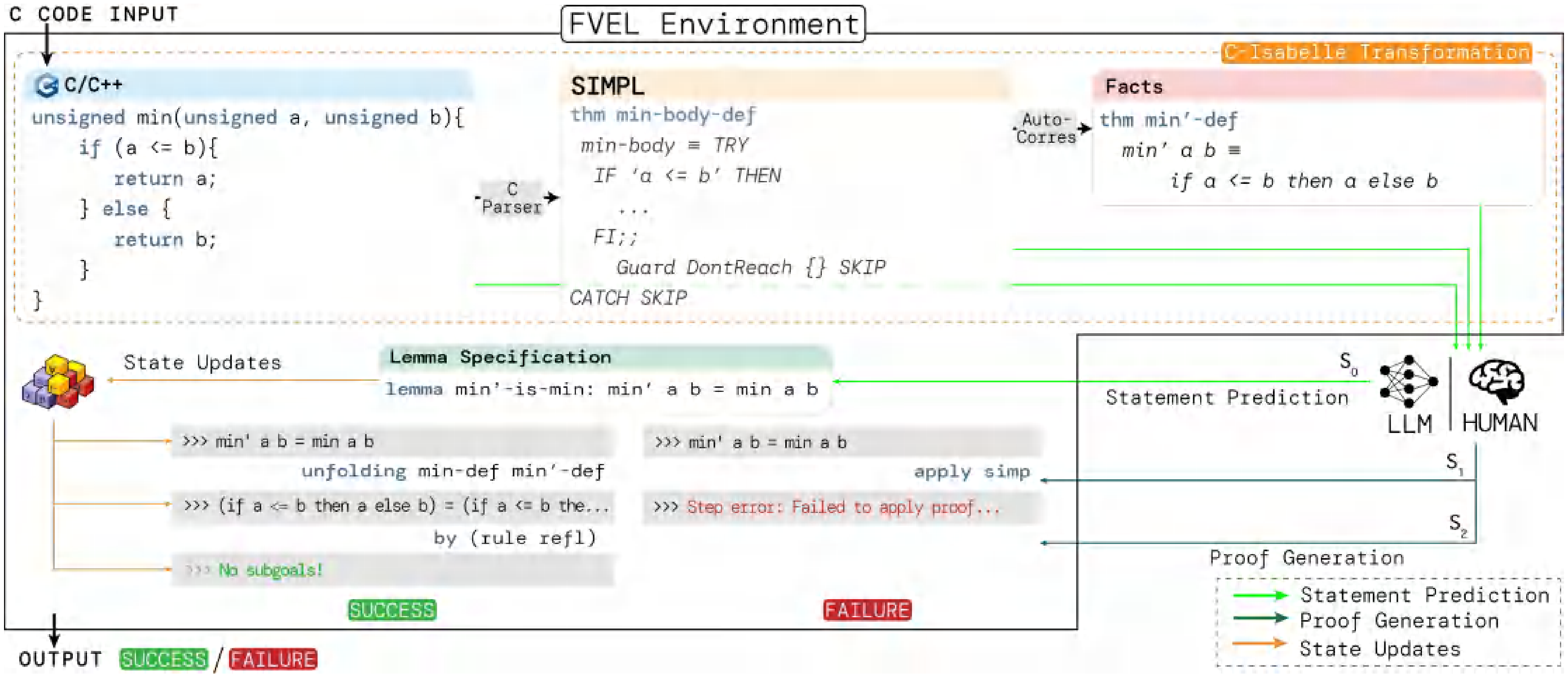
- FVEL: an interactive formal verification environment with LLMs.
- FVELer: a large-scale verification dataset with 758 theories, 29,304 lemmas, and 201,498 proof steps in total that contain deep dependencies.
- The fine-tuned LLMs with FVELer outperform on Code2Inv and SV-Comp datasets, and successfully verify translated Python code.

* Formal verification is the process of mathematically checking that a system's behavior satisfies a given property.

FVEL: Workflow

FVEL provides an interactive environment with LLMs that leverage rigorous theorem-proving processes:

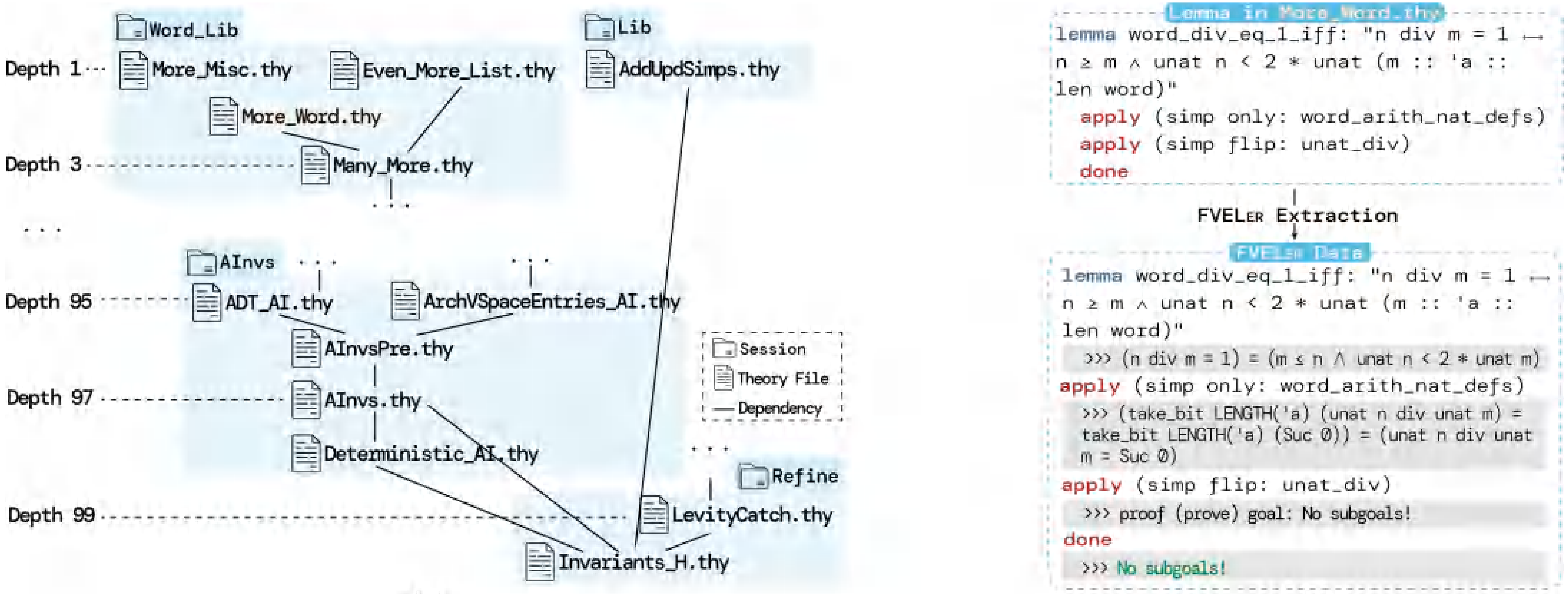
1. Transforms the input **C code** into facts, and then provides the facts to the LLM.
2. The LLM **generates a lemma in Isabelle** (a formal system for theorem proving) as a formal description of the code specification;
3. The LLM **generates proof steps** with feedbacks from Isabelle;
4. The output is a binary result indicating the success or failure of the verification.



FVELer: Dataset

FVELer has two main components:

1. **Theories dependencies** (Figure a). A resource for dependencies among theories, lemmas, and C code specified by SeL4 (a micro-kernel operating system) verification.
2. **Lemmas from theories with their Isabelle proof states** (Figure b), which support step-wise proving process in Isabelle.



(a) (b)

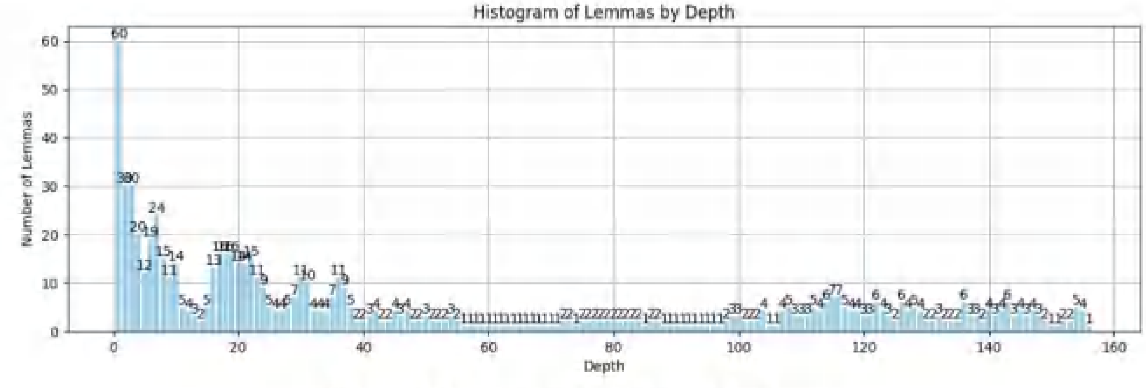
FVELer: Dataset Statistics

FVELer contains **758 theories, 29,304 lemmas, and 201,498 proof steps**. We randomly split FVELer according to lemmas. Lemmas in the “test-hard set” are in higher depths in the dependency relationship.

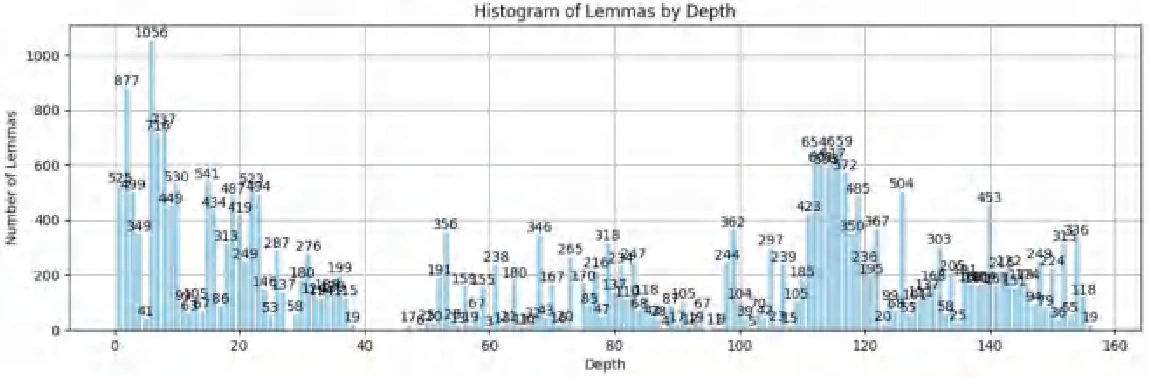
Table 1: FVELER Statistics. A *theory* is a `.thy` file in seL4 that contains multiple *lemmas*. Each *lemma* has multiple *proof steps*. The train/val/test/test-hard data split is based on *lemmas*.

	Total	Train	Val	Test	Test-Hard
▷ *Theory*					
Number of Theories	758	-	-	-	-
Average depth*	-	73.687	73.732	73.958	31.476
Maximum depth	156	156	155	155	115
▷ *Lemma*					
Number of Lemmas	29,304	26,192	1,145	1,115	852
▷ *Proof Step*					
Number of proof steps**	201,498	181,887	6,931	8,036	4,644
Average proof steps	-	6.944	6.053	7.207	5.450
Maximum proof steps	963	963	188	574	107

* Depth: Degree of the theory dependency graph by `import` relationship.
** Proof step: A single step in Isabelle producing a valid statement for interaction."



(a) Distribution of dependency by theory.



(b) Distribution of dependency by lemma.

FVEL: Experiments

FVELer fine-tuned Llama3-8B solves **17.39% (69→81)** more problems, and Mistral-7B **12% (75→84)** more problems in SV-COMP dataset.

For Python code verification, the fine-tuned LLMs are able to verify more Python code with translation to C code.

Table 2: Result on formal verification task. FT: Fine-tuned.

Model	Code2Inv (#=133)	SV-COMP-47 (#=47)	SV-COMP (#=1,000)
▷ *Symbolic Solver*			
UAUTOMIZER [10]	92	1	374
ESBMC [6]	68	1	358
▷ *LLM-based Solver*			
Lemur-GPT-3.5-turbo [40]	103	14	-
Lemur-GPT-4 [40]	107	25	-
Mistral-7B [14]	37	10	75
Mistral-7B-FT	40	14	84
Llama3-8B[4]	46	11	69
Llama3-8B-FT	46	16	81

Table 4: Result on Python (Translated to C) Code Verification.

Model	# Verified
Mistral-7B	35 / 93
Mistral-7B-FT	42 / 93
Llama3-8B	38 / 93
Llama3-8B-FT	43 / 93

Please refer to our paper for more analyses and implement details.

Thank you for listening!

Contact: linxh55@mail2.sysu.edu.cn