Formalising Software Requirements with Large Language Models

Arshad Beg, Diarmuid O'Donoghue, and Rosemary Monahan
Principles of Programming Group, Department of Computer Science
National University of Ireland, Maynooth.

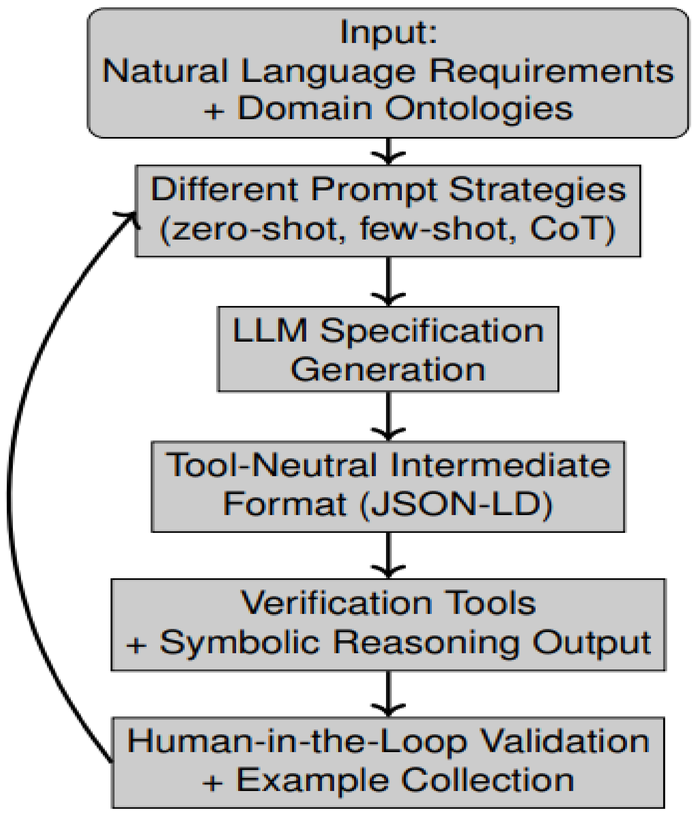
Introduction

Software development faces challenges in requirements traceability and verification. Many specifications are written in natural language, which can be ambiguous. The VERIFAI project leverages Large Language Models (LLMs) to generate formal specifications, bridging the gap between informal requirements and verifiable code.

Motivation: Tackling Ambiguity in Software Requirements

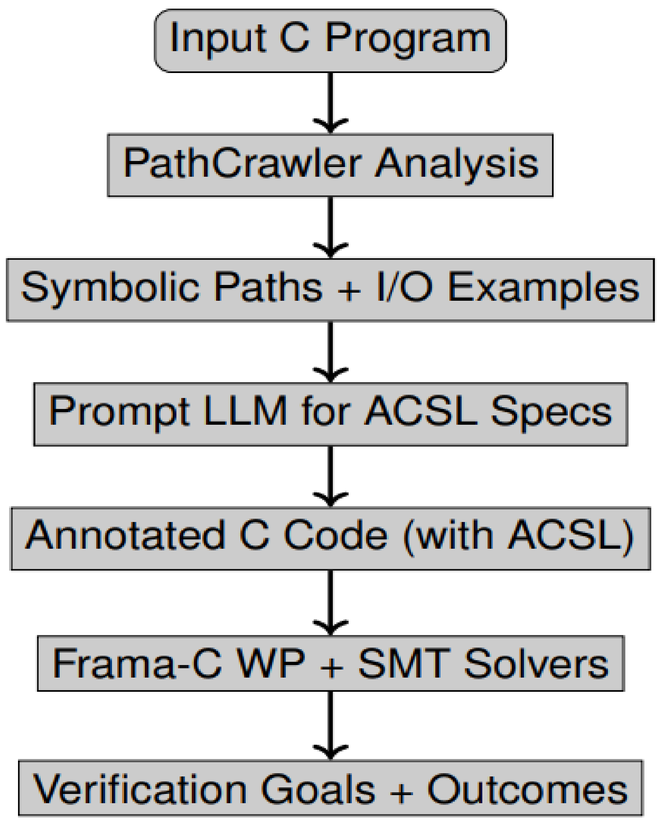
Software requirements are generally written in natural language making them easy to write but hard to verify. Natural language can be vague and open to different interpretations. Misunderstandings at this stage can cause problems leading to bugs, safety issues and failed expectations. Another challenge is traceability where a requirement originates and undergoes varied interpretations and refinements.

VERIFAI Workflow Pipeline



Proposed workflow for the VERIFAI pipeline: Natural-language requirements and domain ontologies are combined with prompt strategies to generate formal specifications via LLMs. Outputs in JSON-LD feed verification tools, with symbolic and human feedback refining results.

Initial Experiment Setup with Frama-C Ecosystem



Methodology of the initial experiments: Combining LLM with symbolic analysis tools in the Frama-C ecosystem. The workflow integrates path-based examples and verification outputs to guide the generation of context-aware ACSL specifications.

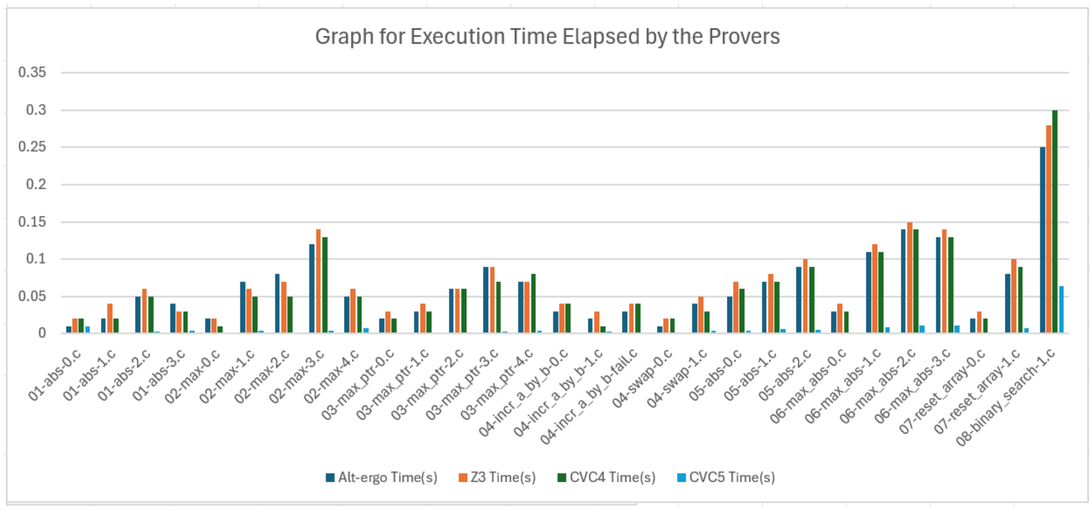
Output of Verification Toolchain for a set of C Programs

C File	Goals	Proved	Qed	Timeout	Term.	Unreach.
01-abs-0.c	0	2 / 2	1	1	0	0
01-abs-1.c	1	3 / 3	1	1	1	0
01-abs-2.c	3	5 / 5	1	1	3	0
01-abs-3.c	4	6 / 6	1	1	4	0
02-max-0.c	2	4 / 4	1	1	2	0
02-max-1.c	5	5 / 7	1	1	3	0
02-max-2.c	1	3 / 3	1	1	0	1
02-max-3.c	6	7 / 8	1	1	4	1
02-max-4.c	8	10 / 10	1	1	7	1
03-max_ptr-0.c	0	2 / 2	1	1	0	0
03-max_ptr-1.c	2	4 / 4	1	1	2	0
03-max_ptr-2.c	2	4 / 4	1	1	2	0
03-max_ptr-3.c	2	4 / 4	1	1	1	1
03-max_ptr-4.c	4	6 / 6	1	1	4	0
04-incr_a_by_b-0.c	0	2 / 2	1	1	0	0
04-incr_a_by_b-1.c	2	4 / 4	1	1	1	1
04-incr_a_by_b-fail.c	2	3 / 4	1	1	1	0
04-swap-0.c	2	4 / 4	1	1	2	0
04-swap-1.c	8	10 / 10	1	1	7	1
05-abs-0.c	0	2 / 2	1	1	0	0
05-abs-1.c	6	8 / 8	1	1	6	0
05-abs-2.c	5	7 / 7	1	1	5	0
06-max_abs-0.c	2	2 / 2	0	0	2	0
06-max_abs-1.c	13	11 / 13	0	0	9	2
06-max_abs-2.c	13	12 / 13	0	0	11	1
06-max_abs-3.c	13	13 / 13	0	0	11	2
07-reset_array-0.c	1	1 / 2	0	1	0	0
07-reset_array-1.c	11	13 / 13	1	1	8	3
08-binary_search-1.c	18	18 / 20	1	1	11	5

Conclusion drawn from output of Z3 prover

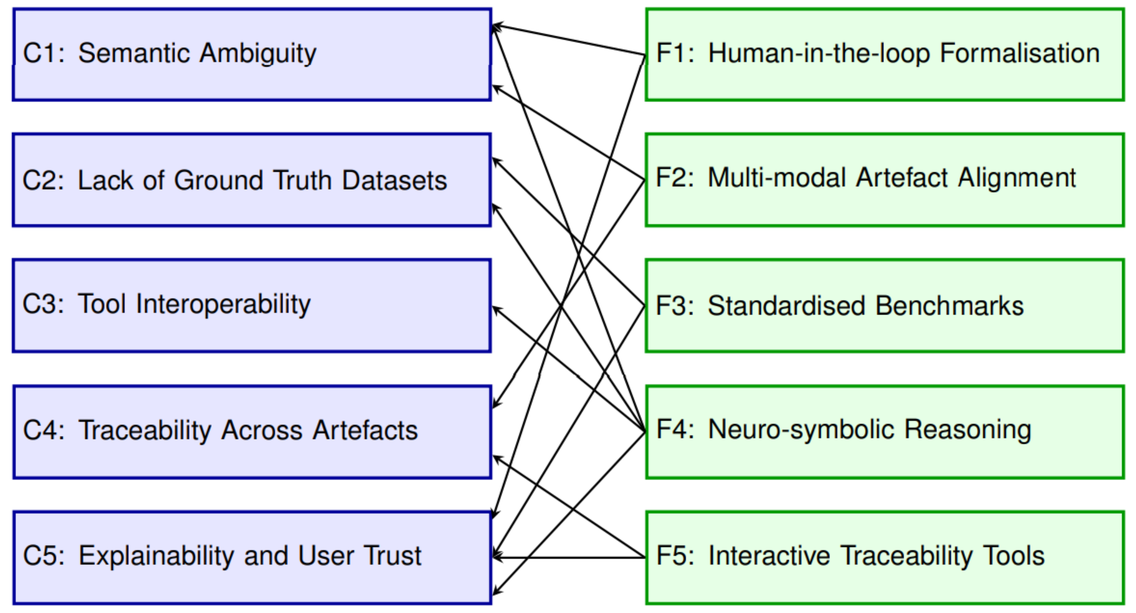
- Across the evaluated 28 C programs, approximately 94% of all verification goals were successfully proven, demonstrating high reliability of the verification toolchain.
- Failures were primarily due to timeout or partial prover coverage on complex arithmetic and array-manipulation examples. Overall, the combined prover strategy offers robust verification performance with only marginal cases requiring further proof engineering or model refinement.

Execution Time Comparison for Provers (Alt-Ergo, Z3, CVC4 and CVC5)



Here, CVC5 delivered the fastest execution times across the evaluated C programs, while Alt-Ergo, Z3, and CVC4 remain competitive but generally slower on larger or more complex C programs.

VERIFAI Challenges and Prospects



Mapping between challenges (C1–C5) and future research (F1–F5) directions in LLM-based formalisation

Publications

1. A. Beg, D. O'Donoghue, and R. Monahan, "Leveraging LLMs for Formal Software Requirements – Challenges and Prospects," Accepted at 7th Int. Workshop on Artificial Intelligence and fOrmal VERification, Logic, Automata, and sYnthesis (OVERLAY) @ the 28th European Conference on Artificial Intelligence (ECAI 2025), Bologna, Italy, to be held on Oct. 26, 2025. Available: https://arxiv.org/abs/2507.14330.

2. A. Beg, D. O'Donoghue, and R. Monahan, "A Short Survey on Formalising Software Requirements using Large Language Models," Draft submitted to Symposium on AI Verification (SAIV), 2025. Available: https://arxiv.org/abs/2506.11874.

3. A. Beg, D. O'Donoghue, and R. Monahan, "Formalising Software Requirements with Large Language Models," ADAPT Annual Conference 2025, Dublin City University, May 15, 2025. [Accepted and Presented Poster]. Available: https://arxiv.org/abs/2506.10704.

This research was conducted with the financial support of Science Foundation Ireland at ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology at Maynooth University [13/RC/2106_P2].