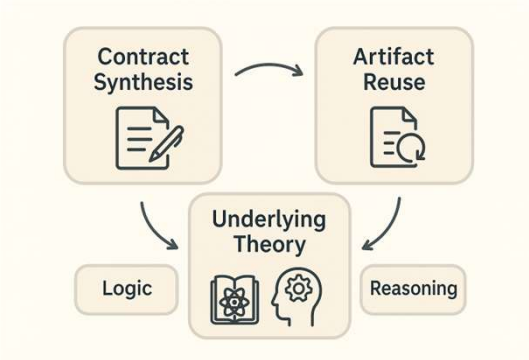
Learning-Infused Formal Reasoning: Contract Synthesis, Artefact Reuse and Semantic Foundations

Arshad Beg, Diarmuid O'Donoghue and Rosemary Monahan

Principles of Programming Group, Maynooth University

LIFR: Three Dimensions



1. Workflow for Contract Synthesis

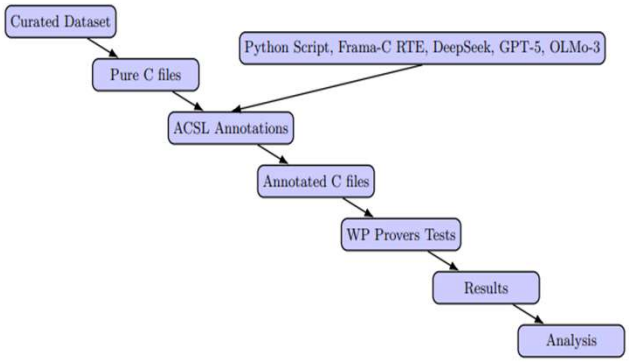
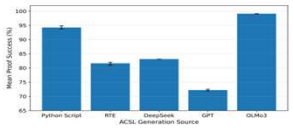
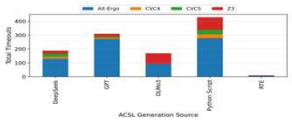


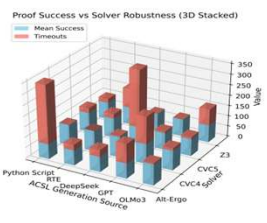
Figure: End-to-end research workflow illustrating how a curated dataset is progressively transformed into pure C files, enriched with ACSL annotations generated via a combination of automated scripts, Frama-C RTE, and large language models (DeepSeek-V3.2, GPT-5.2, and OLMo-3.1 32B Instruct), then consolidated into annotated C files that are evaluated using weakest-precondition (WP) prover tests to produce verification results that are finally subjected to systematic analysis.



(a) Mean proof success rate with solver-dependent variability.



(b) Distribution of solver timeouts per ACSL source.



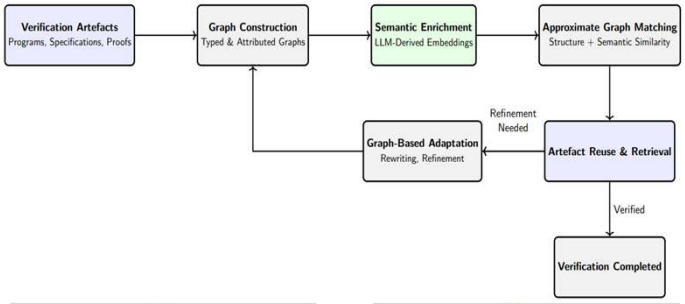
(c) Trade-off between proof success and solver robustness.

Figure: Comparison of WP verification outcomes across ACSL generation methods. The figure contrasts proof success, solver robustness, and their interaction for tool-generated and LLM-generated specifications.

Source	Solver	Runs	Mean	TO	Qed
Python Script	AltErgo	207	71.93	278	1.25
	CVC4	205	72.23	27	1.16
	CVC5	203	72.19	33	1.16
	Z3	205	72.57	91	1.22
RTE	AltErgo	115	99.15	6	7.50
	CVC4	113	99.12	0	8.50
	CVC5	111	99.09	0	7.50
	Z3	113	99.12	4	6.50
DeepSeek	AltErgo	149	94.89	130	6.17
	CVC4	147	93.76	13	6.53
	CVC5	145	93.80	23	6.32
	Z3	147	94.56	22	5.04
GPT	AltErgo	177	80.98	272	1.86
	CVC4	175	81.63	8	2.00
	CVC5	173	81.90	10	2.30
	Z3	175	82.07	20	1.97
OLMo3	AltErgo	134	83.14	92	2.92
	CVC4	133	83.14	0	2.81
	CVC5	132	83.14	0	3.05
	Z3	133	83.14	77	2.87

Table: WP verification results for ACSL annotations generated by different sources. Mean success denotes the average percentage of proved goals over generated goals. Qed time is reported in milliseconds.

2. Workflow for Verification Artefact Reuse



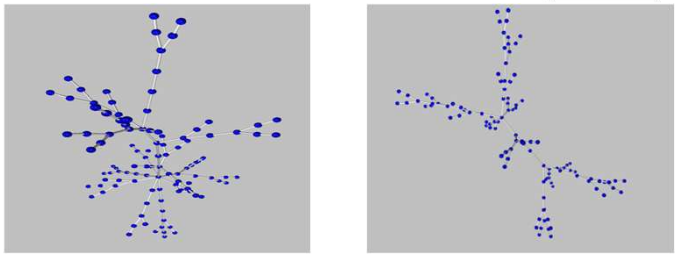


Figure: Graph constructed for ACSL annotated C programs for **quicksort** and **merge** using workflow of verification artefact reuse.

3. Theoretical / Semantic Foundations

- Strong mathematical semantics are essential for trustworthy verification, motivating the use of unified frameworks that support heterogeneous modelling languages and tools.
- The Unifying Theories of Programming (UTP) provides a relational, algebraic framework that unifies multiple paradigms (imperative, concurrent, state-based) and enables consistent reasoning via refinement, composition, and behavioural equivalence.
- The Theory of Institutions offers an abstract, category-theoretic foundation for representing and relating logical systems, supporting translations and interoperability across diverse specification languages.
- Together, UTP and institutional semantics act as "semantic governance" for AI-assisted verification, ensuring that generated artefacts remain logically sound while enabling cross-tool and cross-language integration.

This research was conducted with the financial support of Research Ireland under Grant Agreement No. 13/RC/2106_P2 at the ADAPT Centre at Maynooth University