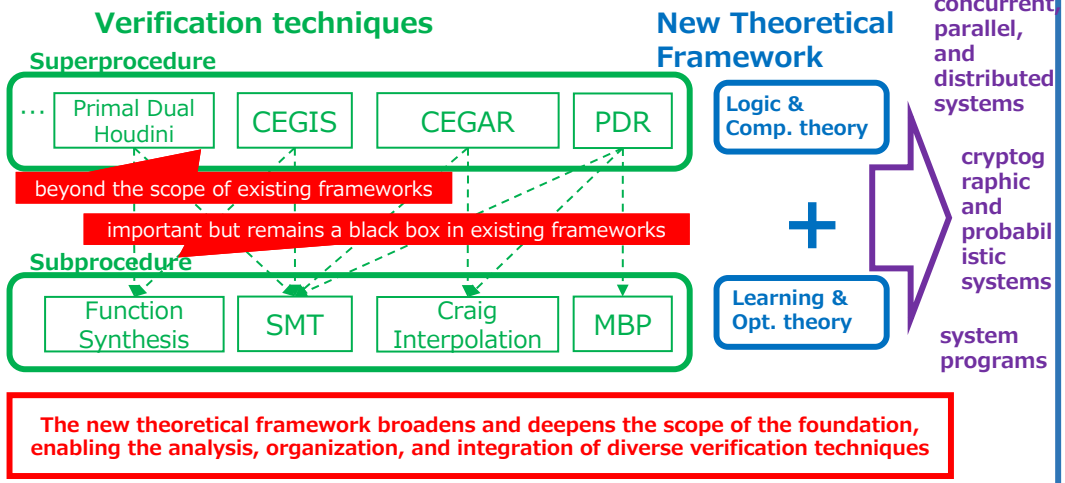
【Grant-in-Aid for Scientific Research (S)】

Foundations and Applications of Program Verification Techniques

Principal Investigator	Tohoku University, Research Institute of Electrical Communication, Professor UNNO Hiroshi　Researcher Number : 80569575
Project Information	Project Number : 25H00446　Project Period (FY) : 2025-2029 Keywords : Program Verification, Formal Logic, Theory of Computation, Optimization Theory, and Learning Theory

Purpose and Background of the Research

- **Outline of the Research: Program verification** is a technology for mathematically ensuring that the software controlling a computer behaves as intended. Current techniques have become increasingly complex due to accumulated optimizations for efficiency. This complexity makes the overall structure hard to grasp, resulting in issues such as mathematical unsoundness and limited extensibility. To address these problems, this research establishes a mathematically concise theoretical foundation based on **logic**, **theory of computation**, **optimization theory**, and **learning theory**, and develops verification techniques combining **correctness**, **extensibility**, and **efficiency**. The foundational techniques will then be extended in response to practical demands. In particular, we will develop theory-based automated verification tools for **concurrent, parallel, and distributed systems**, for **cryptographic and probabilistic systems** essential to security and privacy, and for **system programs** written in languages such as C and Rust.



The new theoretical framework broadens and deepens the scope of the foundation, enabling the analysis, organization, and integration of diverse verification techniques

Figure 1. The Proposed New Theoretical Framework

- **Research Background:** Program verification is used to verify systems such as aircraft control, CPU, OS drivers, and cloud networks. Though often unnoticed, it plays a vital role in social infrastructure. As many verification problems are undecidable, heuristics are essential for handling practical cases. Years of research have produced numerous techniques, but their accumulation has made tools highly complex—often beyond expert understanding. In one case, a minor change broke termination, and the issue went unnoticed for nearly a decade. This complexity hinders assurance of correctness and extensibility, and makes verification time unpredictable.
- **Previous Research:** To address the increasing complexity of program verification tools, the principal investigator and collaborators have worked to reconstruct verification techniques on concise mathematical foundations, drawing on logic and computation theory. By formulating various verification problems as validity checking problems for fixed-point logic formulas, we enabled cross-application of techniques and built efficient tools. We also clarified the link between procedural program verification and proof search in cyclic proof systems, organizing existing methods via proof theory. The aforementioned nontermination bug was uncovered through this.
- **Research Objectives:** This research aims to establish a concise theoretical foundation for program verification that supports deeper integration of diverse techniques, including heuristics. By incorporating mathematical optimization and learning theory, we will provide a theoretical foundation for heuristics beyond the scope of formal logic and computation theory. We will also apply the resulting framework to critical domains such as concurrent, parallel, and distributed systems, cryptographic and probabilistic computation, and low-level software.

Expected Research Achievements

- **Expanding and Deepening the Scope of the Foundational Framework:** This research aims to build a concise mathematical foundation for program verification, combining correctness, extensibility, and efficiency. By broadening the scope of this foundation, we will support advanced verification across key domains such as concurrency, parallelism, distribution, cryptography, probabilistic computation, and low-level programming. We will apply optimization theory—especially Lagrangian duality—to analyze high- and low-level procedures in modern techniques, enhancing their soundness and extensibility. In addition, learning theory will be used to study convergence behavior in iterative methods, improving efficiency.
- **Application to Real-World Software Verification:** To demonstrate its effectiveness, we will formally verify query and transaction processing—key components of web and cloud systems—previously difficult to handle due to parallel and distributed execution, cryptographic and probabilistic behavior for security, and low-level programming in C and Rust. By extending verification tools through the new theoretical foundation, we address these challenges.

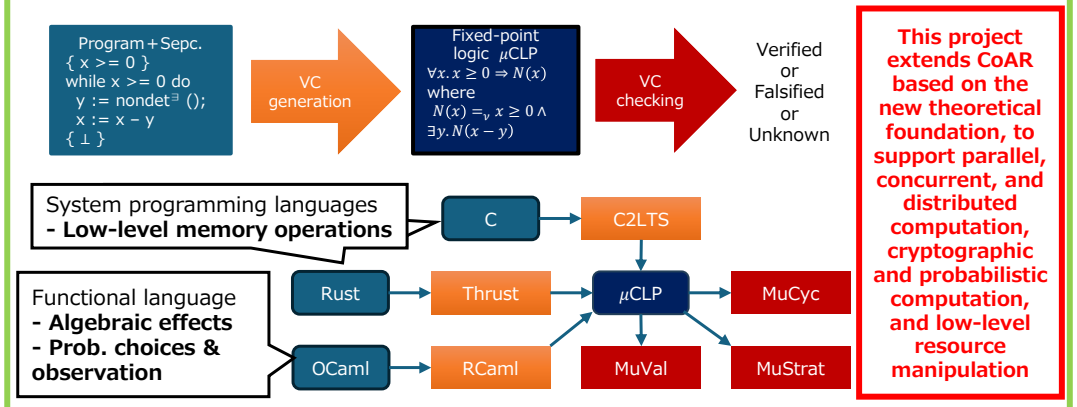


Figure 2. CoAR, the suite of program verification tools proposed in this project

Homepage Address, etc.	https://www.riec.tohoku.ac.jp/~unno/ https://github.com/hiroshi-unno/coar