Formal Verification and Security of Smart Contracts in Decentralized Systems

Souheib Baarir Uli Fahrenberg Ghiles Ziat

February 17, 2025

Abstract

Smart contracts, especially those used in decentralized finance (DeFi), have become a central feature in blockchain systems but remain vulnerable to errors and security risks. This thesis will explore the use of formal verification techniques to ensure the safety, correctness, and security of smart contracts. The focus will be on employing a variety of formal methods—such as model checking, theorem proving, and abstract interpretation, along with bounded model checking to verify smart contract logic and ensure critical properties like privacy, composability, and manipulation resistance.

Objectives

1. Formal Verification of Smart Contracts

   Develop a framework for verifying the correctness of smart contracts using a range of formal methods, including model checking, theorem proving, symbolic execution, and bounded model checking. These methods will help identify vulnerabilities early in the development process and ensure smart contracts perform as expected.

2. Semantic Property Verification for DeFi Protocols

   Focus on verifying essential properties of DeFi protocols, such as composability (the ability for protocols to safely interact), privacy, and resistance to manipulation. These properties will be analyzed using formal methods to ensure DeFi systems remain secure and functional when integrated with other protocols.

3. Verification Techniques Beyond Bounded Model Checking

   Explore a combination of verification techniques, including:

   - Model Checking: For verifying all possible execution paths of smart contracts to ensure that no errors occur.
   - Theorem Proving: To mathematically prove the correctness of contract behaviors based on formal specifications.

- Abstract Interpretation: Implement abstract interpretation to analyze the behavior of smart contracts, approximating execution paths to detect potential issues, such as overflows, underflows, and logical errors, without needing to explore every possible state. This technique will be useful for contracts with large or infinite state spaces.

4. Development of an Automated Verification Tool

   Create a tool that combines these formal verification techniques, enabling developers to easily check their smart contracts for correctness and security. The tool will be user-friendly and capable of identifying critical vulnerabilities, such as reentrancy attacks and overflow issues, across various DeFi protocols.

5. Case Studies of Popular DeFi Protocols

   Apply the verification framework and tool to real-world DeFi protocols such as Uniswap, MakerDAO, and Aave. The case studies will help validate the approach, detect vulnerabilities, and improve the security of these systems.