Temporal Specification and Verification of Smart Contracts

Yuanodng Cyrus Liu *

Jan. 2024

1 Description

Smart contracts are programmable entities designed to autonomously manage digital assets, serving as integral components of blockchain technology. These contracts often involve temporal behaviors, such as time-dependent transactions, which are recorded immutably on the blockchain. Ensuring the correctness of these smart contracts prior to deployment is crucial. This research project aims to address the challenge of formally verifying temporal behaviors of smart contracts without human interactions, providing a comprehensive correctness proof.

Firstly, formal specification of temporal behaviors in smart contracts requires attention. Unlike traditional programs, smart contracts employ unique constructs (e.g., account model, wallet, balance) for frequent transactions submission to blockchains. While temporal logic proficiently describes program behaviors (safety and liveness), its adaptation to the smart contract domain remains unexplored. Thus, designing a temporal specification language precisely capturing smart contract temporal behaviors marks the initial step toward achieving our objective.

Subsequently, the research aims to design a verification algorithm capable of processing temporal specifications and smart contracts, thereby determining whether the specified property holds true. Although existing Linear Temporal Logic (LTL) verifiers can parse standard LTL formulas, smart contracts, predominantly coded in languages like Solidity, pose unique challenges. Hence, the second aspect of this research focuses on developing an approach integrating tailored smart contract temporal formulae with state-of-the-art temporal verification algorithms. This fusion aims to achieve the ultimate goal of formally verifying temporal behaviors inherent in smart contracts.

This research project contributes to the emerging field of blockchain technology by providing a robust framework for the formal verification of temporal behaviors in smart contracts. In collaboration with the faculty, the students will undertake a thorough literature review and conduct extensive experiments leveraging existing verification tools. The project aims to extend the proposed temporal specification language and verification algorithm, thereby bolstering the reliability and security of smart contracts, paving the way for broader adoption in diverse applications.

2 Expected outcomes

2.1 Deliverables

1. Exploration of Related Work and Experimental Analysis:
   - Conduct an extensive literature review by analyzing relevant research papers[1].
   - Construct a comprehensive review of existing methodologies and open source approaches[2].
   - Establish and execute experiments to evaluate the effectiveness of the verification toolchain among the literature[3][4], submit a 5 page report.
2. Development of a Temporal Logic-Based Specification Language for Smart Contracts:
   - Extend a specialized smart contract specification language grounded in temporal logic.
   - Provide comprehensive semantics for each rule within the specification language to ensure clarity and precision.

*MAP at Grinnell.

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9519387&tag=1

[2] https://github.com/utopia-group/SmartPulseTool

[3] https://www.microsoft.com/en-us/research/uploads/prod/2019/05/1812.08829.pdf

[4] https://github.com/ultimate-pa/ultimate

3. Formulation of Temporal Properties in Smart Contracts:

    - Formulate a series of temporal formulae aligned with the temporal properties inherent in smart contracts.
    - Derive temporal formulae that correspond directly to the temporal behaviors addressing the security concerns and requirements of smart contract programs.

2.2 Public presentation and artifact

1. Public Presentation Platforms:

    - Presenting at the Grinnell CS Research Symposium to engage with local academic community and receive feedback.
    - Showcase the research findings at esteemed Student Research Competition (SRC) events hosted within international conferences such as Programming Language Design and Implementation (PLDI), Symposium on Principles of Programming Languages (POPL), International Conference on Functional Programming (ICFP), among others.

2. Software Artifact:

    - Deliver a robust open source toolchain capable of automating the verification process for a defined set of smart contracts, ensuring adherence to specified temporal properties.
    - The toolchain represents a tangible outcome of the research efforts, facilitating the seamless validation of smart contracts in real-world scenarios.

3 Budget

N/A