Research Statement

Yuandong Cyrus Liu

December 2024

1 Summary

My primary area of research revolves around formal binary analysis, temporal verification, AI security, and blockchain security. Specifically, my focus lies in enhancing software/systems security through programming language-based approaches and automatic verification techniques. The theoretical aspect of my work involves exploring formal logics to reason about low-level programs, with a particular interest in understanding how program properties are preserved in the executable code. Through gaining new insights, the ultimate goal is to develop practical and efficient tools that can effectively apply formal methods in real-world scenarios. This, in turn, will facilitate bug hunting and code optimization during software development and testing, while also offering rigorous proofs for the proposed solutions. My research efforts are guided by three overarching themes:

- **Emphasis on Simplicity**: I am deeply motivated by the pursuit of simplicity in various aspects. Whether it's in the theory behind a problem or the underlying lemmas, I find great fascination in discovering simple and precise principles that lead to various solutions. I believe that simplicity often goes hand in hand with generality, and I strive to communicate and appreciate the elegance of simple theories in my research.
- **Architecting**: I find immense satisfaction in the process of designing straightforward and coherent prototypes and subsequently transforming them into reality. For me, research is not just about theoretical exploration or detailing technical aspects, but rather about harmoniously synthesizing both elements to create practical solutions.
- **Collaborative Approach**: I view research as a collective endeavor within a community of researchers. I actively engage in communication and collaboration through teaching, writing, and seeking opportunities to work with other researchers and students. I believe that the best research outcomes emerge from a dynamic and open dialogue among peers.

The following sections outline several ongoing areas of research and how they align with the aforementioned themes. Additionally, I provide insights into planned future directions for my research.

2 Published Research

My primary research objective centers around developing fully automated tools capable of conducting formal verification on practical software systems and malware detection. To achieve this, my work encompasses several key areas:

- **Hybrid Analysis for Program Verification**: When dealing with practical programs, static analysis often encounters the state explosion problem. To address this issue, I have designed and implemented a hybrid analysis tool called DrNLA [1, 4]. DrNLA leverages both static and dynamic analysis to mitigate the state explosion problem, enabling static analyzers to reason about programs involving nonlinear computations. Within DrNLA, I devised a dual rewrite algorithm that combines dynamic and static analysis. This algorithm collects program execution traces and infers candidate invariants based on the trace data. Subsequently, a static verifier validates these candidates, terminating the process once a sound linear candidate is identified. The dual rewriting algorithm not only extends temporal verification to the nonlinear domain but also shows promise in enhancing program slicing and compiler optimizations.

- **Binary Verification**: Recognizing that compilers may not always be entirely trustworthy, my research delves into understanding the changes made during the compilation process from source code to binary (machine code). Instead of focusing on studying the compiler itself, I propose verifying the differences between the binary and its source code. My work on DarkSea [6], a toolchain for binary verification, aims to expand the capabilities of existing verifiers to encompass low-level system software verification. I developed a bit-wise branching strategy [5] for bit-vector reasoning, which was integrated into the open-source program analysis framework Ultimate. This enhancement was subsequently incorporated as a verification module in DarkSea, enabling automated verification of temporal properties for lifted binaries.
- **Mobile Network Security**: In the current era, mobile devices have become indispensable gadgets for nearly everyone, with an increasing number of mobile apps running on personal devices. This prevalence raises concerns regarding personal data security and privacy violations. To address these issues, I have developed an Android dynamic tool to monitor app behaviors, including URL requests, camera usage, microphone access, SMS interactions, and more [7, 8]. Collaborating with my colleagues, we designed a machine learning algorithm employing CNN (Convolutional Neural Network) to detect malicious applications. The algorithm learns from app behavior data, mapping out an abnormal behavior space. Apps exhibiting behavior within this space are reported as potentially malicious. Additionally, I have created an NLP model for mobile app review mining [3]. This model effectively identifies the most valued features of various apps based on user reviews.

By pursuing these research directions, my ultimate goal is to contribute to the development of practical and automated formal verification tools while addressing crucial challenges in program analysis and mobile network security.

3 Ongoing and Future Research

In the pursuit of my long-term research objectives, I aim to develop advanced techniques for automated reasoning of vulnerable systems and bug discovery. As technology continually evolves and finds widespread adoption in various industries, I have developed a strong interest in the following research directions:

- **Formal Verification for AI Systems**: The prevalence of AI-powered applications across diverse domains, such as coding, image/video processing, prediction, and recommendation, has significantly increased with the availability of domain-specific data. While these applications have become more precise and efficient, the complexity and computational demands of model training, such as in large language models (LLM), have also grown. In this context, ensuring the correctness of AI algorithms and the implementations of generative AI is crucial to prevent AI prejudice and privacy violations. Developing formal verification techniques for AI systems is vital to guarantee their reliability and safety. My current role primarily focuses on researching and developing correct and reliable compilers for Artificial General Intelligence (AGI) computing systems. My research spans topics such as intermediate representation (IR) verification, translation validation, and more.
- **Blockchain Security and Practical Applications**: The rapid growth of distributed computing systems, fueled by blockchain technology, has led to various practical applications, including the integration of smart contracts into distributed ledgers through different implementation protocols. However, this progress has also raised significant security concerns. Smart contracts, as autonomous programs managing digital assets on a distributed ledger, are vulnerable to unintended bugs that could result in substantial capital losses. Ensuring the intended temporal behaviors of smart contracts before their deployment is of paramount importance, given that all transactions on the blockchain are immutable, and any asset loss would be permanent. To address this challenge, the development of formal verification tools [2] for smart contracts and distributed protocols is essential to mitigate the risk of asset loss and enhance the security of emerging financial systems.
- **Data-driven Program Analysis**: Traditional program analysis methods have primarily focused on verifying safety and liveness properties. Two challenging tasks in program correctness verification involve identifying loop invariants to establish safety properties and finding ranking functions to prove program termination. Although static analysis has shown promising results in these areas, scalability remains a significant concern. A promising direction involves leveraging dynamic execution data to learn invariants and ranking functions through machine learning algorithms. This data-driven approach has the potential to offer more effective and scalable solutions for a broader range of software domains.

- **Program Synthesis with Type Systems**: Contemporary program synthesis methodologies extensively rely on the utilization of templates or sketches, in conjunction with input-output examples. The result of these synthesis procedures is contingent upon examples that necessitate users possessing specialized expertise in the synthesized domain, thereby impeding the scalability of these methodologies. Conversely, the emergence of large language models (LLMs) has yielded substantial advancements across various domains including program synthesis, although their capacity to achieve a high degree of accuracy, LLMs operate as probabilistic models, rendering their outcomes intricately linked to the training dataset. Consequently, concerns regarding data privacy and the potential propagation of biased information from the training repository have emerged, thus imperiling the veracity of the synthesis outcomes. It is evident that the aforementioned approaches do not fundamentally address the core challenge of program synthesis. In this context, the development of a synthesis calculus rooted in type systems represents a promising approach to effectively address this predicament at its foundational level.

These envisioned research directions aim to contribute to the advancement of automated reasoning techniques, bolstering the security and reliability of AI systems, enhancing program analysis, and securing distributed computing systems enabled by blockchain technology. By addressing these critical areas, my research endeavors seek to make significant strides towards ensuring the integrity and dependability of cutting-edge technologies in real-world applications.

References

[1] Cyrus Liu, Y., Le, T.C., Antonopoulos, T., Koskinen, E., Nguyen, T.: Drnla: Extending verification to non-linear programs through dual re-writing. arXiv e-prints arXiv:2306.15584 (Jun 2023). https://doi.org/10.48550/arXiv.2306.15584

[2] Larsen, S., Johanson, K., Liu, Y.C.: Vesc: Towards temporal verification of smart contracts. In: Companion Proceedings of the 2024 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity. p. 46–48. SPLASH Companion '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3689491.3689974, `https://doi.org/10.1145/3689491.3689974`

[3] Liu, Y., Li, Y., Guo, Y., Zhang, M.: Stratify mobile app reviews: E-lda model based on hot "entity" discovery. In: 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). pp. 581–588 (2016). https://doi.org/10.1109/SITIS.2016.97

[4] Liu, Y.C.: Temporal Verification of Nonlinear Programs. Ph.D. thesis, Stevens Institute of Technology (2022)

[5] Liu, Y.C., Le, T., Koskinen, E.: Source-level bitwise branching for temporal verification. CoRR **abs/2111.02938** (2021), `https://arxiv.org/abs/2111.02938`

[6] Liu, Y.C., Pang, C., Dietsch, D., Koskinen, E., Le, T.C., Portokalidis, G., Xu, J.: Proving ltl properties of bitvector programs and decompiled binaries. In: Programming Languages and Systems: 19th Asian Symposium, APLAS 2021, Chicago, IL, USA, October 17–18, 2021, Proceedings 19. pp. 285–304. Springer (2021)

[7] Wang, S., Guo, Y., Wu, Q., Liu, Y.: A detection method of android application malicious behaviors based on xposed framework. In: Sciencepaper Online (2015), `https://www.paper.edu.cn/releasepaper/content/201512-1264`

[8] Xu, S., Ma, X., Liu, Y., Sheng, Q.: Malicious application dynamic detection in real-time api analysis. In: 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 788–794 (2016). https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2016.166