Yun-Rong (Lauren) Luo

Research Statement

+1-774-991-9349
yunrong@umich
lauren-yrluo.github.io

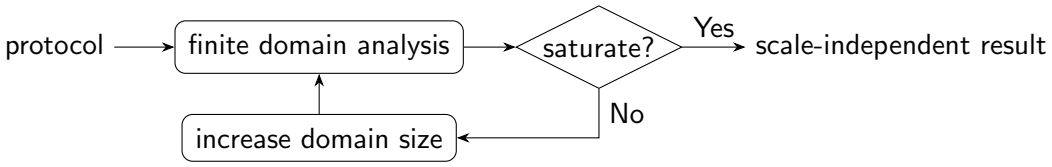
Research Goals

Distributed protocols ensure that the operations of distributed systems are reliable and scalable. For reliability, a distributed protocol should establish certain *safety* properties such as data consistency and consensus between nodes. For scalability, a distributed protocol can model a distributed system of *any size*, potentially with an infinite number of nodes. It is thus important to ask whether the designed distributed protocols are *correct*, i.e., they are actually reliable and scalable. The problem can be formulated as follows.

Problem Statement 1 (Safety Verification for Distributed Protocols) *Given a distributed protocol P and a safety property S, the problem asks whether S holds for any size of P.*

Unfortunately, Problem Statement 1 falls into the category of *parameterized verification*, which is known to be undecidable. Therefore, approaches to this problem still largely rely on *interactive theorem proving*, which demands considerable manual effort and extensive domain knowledge. However, real-world distributed protocol designs often exhibit certain regularity: *structural symmetry* or *temporal repetitiveness*. Prior research [GS23; FGS23] has established practical methodologies for the automated analysis of distributed protocols with such properties, and verified safety for a wide set of useful protocols, including fairly complex models of consensus algorithms, two-phase commit, Chord ring, and simplified versions of Paxos. Continuing this line of research, my research seeks **automated analysis and verification** for this restricted yet useful class of distributed protocols.

The approaches of [GS23; FGS23] are based on the key observation that *a protocol's behavior saturates at a small finite cut-off size*. That is, we can perform repeated automated analyses on a protocol of small finite sizes, gradually increasing the protocol sizes until the analysis saturates, achieving a *scale-independent* result applicable to any protocol size. The flowing flow chart visualizes the procedure.



My research focuses on automatically deriving a quantified first-order logic formula R that describes the *full behavior* of a protocol P. In this way, safety verification can be achieved by checking whether R implies S for any desired safety property S for the protocol P. We have shown that it is possible to derive such formulas R for protocols exhibiting structural symmetry [LGS25; FGS23], i.e., protocols containing types whose values are *indistinguishable* and can be arbitrarily permuted without changing the protocols' behaviors. For example, a type `node` may contain values $\{n_0, n_1, n_2, \ldots\}$ where each n_i is indistinguishable. I envision our approach will also be effective for protocols exhibiting temporal repetitiveness, i.e., protocols with totally-ordered *time-stamps* values and repetitive behaviors across different time-stamps. My research goals are as follows.

- Extend our current work [LGS25; FGS23] to protocols with temporal repetitiveness.
- Establish a theoretical foundation for our approach.

Current Progress

I have developed a prototype `FoRSMin` to synthesize R for distributed protocols, which is an improvement over `QSM` [FGS23]. `FoRSMin` has been evaluated on 19 protocols from ivybench [LGSa] that are structurally symmetric and include fairly complex models of consensus algorithms, two-phase commit, sharded key-value store, etc. Partial results of `FoRSMin` have been published in [LGS25]. I have given a talk about this paper at the 2024 International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLA) [LGSb]. We have submitted a follow-up paper with the complete results to the 2025 International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS).

`FoRSMin` performs two key steps to derive R for distributed protocols:

1. *Forward reachability*: computes the *reachable states* of a finite protocol instance to capture the protocol's full behavior.
2. *Logic minimization*: derives a quantified formula R representing the reachable states that syntactically converges when the protocol's behavior saturates.

I have proposed and implemented improvements in both steps, enabling `FoRSMin` to surpass `QSM` in terms of **scalability** and result **quality**. Specifically, I have developed: (1) an efficient and scalable depth-first search algorithm for forward reachability that allows `FoRSMin` to handle more complex protocol instances; (2) a stronger logic minimization procedure that yields more compact R, providing an easier interpretable analysis of protocol behaviors for their designers.

Future Work

My short-term goal is to further optimize the `FoRSMin` prototype and experiment `FoRSMin` on a wider set of protocols. My long-term goal is to extend `FoRSMin` to handle protocols with totally ordered time-stamp values. Given the effectiveness of using temporal regularity in analyzing protocols with totally-ordered values in [GS23], `FoRSMin` could likely be extended similarly. Another long-term goal is to establish a theoretical foundation for the `FoRSMin` approach. Specifically, I aim to show that for protocols exhibiting structural symmetry or temporal repetitiveness, it is possible to mechanically derive R, which captures the full behavior of these protocols and serves as an indicator of their safety properties.

The current `FoRSMin` prototype has already shown promising results in automating analysis and verification for distributed protocols. With further refinements, `FoRSMin` will become an impactful tool, providing protocol designers and the research community with fast, automated, comprehensive, and easily interpretable protocol analysis. Furthermore, establishing a theoretical foundation for `FoRSMin` will demonstrate the feasibility of automated analysis and verification for a wide range of distributed protocols, paving the way for new approaches and research directions in this field.

Achievements

I have a strong and broad research experience in **automated reasoning**, especially **model checking** and **constraint satisfaction**. For the former, I have worked on both hardware [Wan+21] and software model checking [LGS25]. For the latter, I have worked on both theoretical [LCJ23] and

practical aspects [CLJ24]. These works have been published at top venues (ICCAD, IJCAI, and Journal of Automated Reasoning), where I am listed as either the first or second author.

- Our work "Compatible Equivalence Checking of X-Valued Circuits" [Wan+21] develops an equivalence checker for ternary-valued circuits. Our equivalence checker could solve 9 more cases (out of 30 test cases) compared to an industrial checker and won first place in Problem A of the 2020 CAD Contest (with 186 teams competing in 3 problems.)
- Our work "A Resolution Proof System for Dependency Stochastic Boolean Satisfiability" [LCJ23] develops the first sound and complete proof system for dependency stochastic Boolean satisfiability (DSSAT). As modern SAT/QBF algorithms are closely related to their underlying proof systems, my work in developing a proof system for DSSAT will pave the foundation for the development of DSSAT solvers.
- Our work "Knowledge Compilation for Incremental and Checkable Stochastic Boolean Satisfiability" [CLJ24] develops an SSAT-proof framework to certify the results of an SSAT solver. Our proof framework helped reveal an existing bug in our SSAT solver and our contribution will likely inspire more follow-up work on enhancing the reliability of constraint solvers.

This research record shows my capability of achieving high-quality research in the field of automated reasoning. My diverse research experience (hardware/software, theory/practice) equips me with strong skills in both theory and practice, making me a suitable candidate to carry out my proposed research agenda.

References

[CLJ24] Che Cheng, Yun-Rong Luo, and Jie-Hong R. Jiang. "Knowledge Compilation for Incremental and Checkable Stochastic Boolean Satisfiability". In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*. Ed. by Kate Larson. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2024, pp. 1862–1872. DOI: 10.24963/ijcai.2024/206. URL: https://doi.org/10.24963/ijcai.2024/206.

[FGS23] Katalin Fazekas, Aman Goel, and Karem A Sakallah. "SAT-Based Quantified Symmetric Minimization of the Reachable States of Distributed Protocols." In: *FMCAD*. 2023, pp. 152–161.

[GS23] Aman Goel and Karem A Sakallah. "Regularity and quantification: a new approach to verify distributed protocols". In: *Innovations in Systems and Software Engineering* 19.4 (2023), pp. 359–377.

[LCJ23] Yun-Rong Luo, Che Cheng, and Jie-Hong R Jiang. "A resolution proof system for dependency stochastic boolean satisfiability". In: *Journal of Automated Reasoning* 67.3 (2023), p. 26.

[LGSa] Yun-Rong Luo, Aman Goel, and Karem Sakallah. *A Collection of Distributed Protocol Verification Problems*. https://github.com/lauren-yrluo/ivybench.

[LGSb] Yun-Rong Luo, Aman Goel, and Karem Sakallah. *Slides for the Talk: "SAT-Based Quantified Symmetric Minimization of the Reachable States of Distributed Protocols: An Update"*. https://lauren-yrluo.github.io/files/scaveri_2024_slides.pdf.

[LGS25] Yun-Rong Luo, Aman Goel, and Karem Sakallah. "SAT-Based Quantified Symmetric Minimization of the Reachable States of Distributed Protocols: An Update". In: *Leveraging Applications of Formal Methods, Verification and Validation. Specification and Verification*. 2025, pp. 374–384.

[Wan+21] Yu-Neng Wang et al. "Compatible Equivalence Checking of X-Valued Circuits". In: *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE. 2021, pp. 1–9.