Research Statement

Priyanka Golia

Ph.D. Candidate
National University of Singapore &
Indian Institute of Technology Kanpur
https://priyanka-golia.github.io

We entrust large parts of our daily lives to automated systems, which are becoming increasingly more complex. Designing and testing such systems is extremely challenging, and there is a emanate need to develop automated systems that are scalable and reliable. An overarching theme of my research is to employ formal methods and artificial intelligence to acquire the critical balance between trustworthiness and scalability in designing and testing automated systems.

Automated synthesis aims to synthesize the desired system (functions, programs, circuits) that provably satisfies the requirements represented as formal specifications. In my dissertation, I designed a scalable data-driven approach that relies upon advances in formal methods and machine learning to achieve significant performance improvement over the prior state-of-the-art. We rely on constrained sampling to generate data which are further fed to a machine learning pipeline to generate the initial candidate system. We leverage the advances in automated reasoning to repair the candidate system to synthesize a system that provably satisfies the given underlying specification. The proposed method achieved significant scalability for real-world instances, and the corresponding publication has received a **best paper award nomination** at ICCAD-21. As a next step, we extended the general-purpose monolithic synthesis to modular designs in which different components of the required system deal with different inputs. Such a setting has various applications in circuit repair, controller synthesis, and more. The corresponding publication has received a **best paper award nomination** at DATE-23.

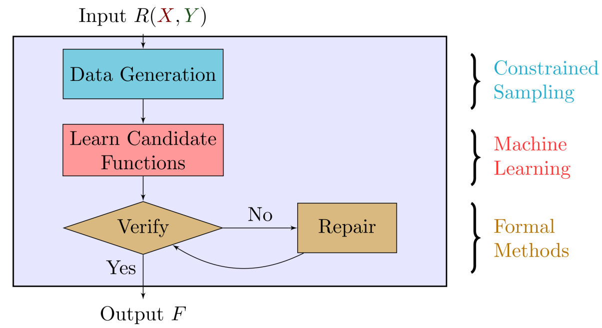
Our work on automated synthesis emphasizes the importance of constrained sampling, which has applications in testing of systems. Constrained sampling is a powerful way of generating test suites. Recent years have seen the development of sampling techniques that are either scaled to real-world problems or accompanied by theoretical analysis on the distribution of produced samples. To increase the trust in the system under test, we need to ensure that the generated test suite captures the given system's functionality, generally referred to as the quality of the test suites. In applications like verifying the fairness of a deep neural network, scalability in generating samples and quality are both key bottlenecks. In my research, I pioneered a test-driven method to design a tuneable constrained sampler that achieves the balance between scalability and distribution of produced samples. In addition to that, we demonstrate a virtuous cycle between testing and design, and equip the sampler with a testing framework that can provide a more nuanced quantitative certification of the quality of samples produced.

My research has led to the release of **five** open-sourced tools. We have presented **tutorial on automated synthesis** at AAAI-22 and IJCAI-22. I was named one of the EECS Rising Stars in 2022.

Research Thrust 1: Automated Synthesis

Automated synthesis deals with the synthesis of programs, functions, and circuits that provably meet the user's given requirement. Given a relation specification $R(X, Y)$ over input X and output Y, the task is to synthesize output Y in terms of X, that is, $Y := F(X)$ such that the given specification is met. Synthesis has been studied for over 150 years, dating back to Boole in 1850's, yet scalability remains a core challenge. To tackle the scalability challenge, we proposed a data-driven approach, Manthan, to efficiently synthesize functions (also represented as circuits) [5, 7].

Manthan first exploits the advances in constrained sampling to generate samples from a given relational specification. Manthan casts the functional synthesis as a classification problem wherein the input variables in samples correspond to features while output variables correspond to labels. The generated samples are fed as training data to learn a classifier encoded as a Boolean function. Since machine learning techniques often produce good but inexact approximations, Manthan leverages advances in automated reasoning and relies on a proof-guided approach that seeks to identify and apply minor repairs to the candidate functions in an iterative manner until it converges to a provably correct functions.



Overview of Manthan.

To this end, we rely on advances in SAT and MaxSAT to diagnose and repair of candidate functions. The corresponding publication received a **best paper award nomination** at ICCAD 2021.

Manthan was able to synthesize functions for 509 instances out of a total of 609 standard suites of instances — to give a perspective, the prior state-of-the-art techniques ranged from 210 to 280 instances. Manthan improved the state-of-the-art by solving an additional 40% of instances. Motivated by the impressive scalability, we turned our attention to program synthesis. We demonstrated that the problem of program synthesis reduces to functional synthesis when there are no syntactic restrictions [4]. Our reduction allows us to transform Manthan as a state of the art approach for program synthesis over bit-vector theory. Furthermore, as a next step, we lift the data-driven approach to contrive modular designs that enabled Manthan to push the envelope in synthesis with explicit input dependencies. Manthan handles additional 26 instances for which the state-of-the-art tools could not synthesize a system [1]. The corresponding publication received a **best paper award nomination** at DATE 2023.

Research Thrust 2: Testing of Systems

Testing ensures that computing systems satisfy their intended functionality. Constrained sampling is a popular way to generate test suites for the system, which is to sample randomly, subject to a given weight function, from the set of solutions of input constraints. The state-of-the-art constrained sampling techniques either provide theoretical guarantees on produced samples or scale to real-world problems. Moreover, there are sampler testers such as Barbarik that can test whether the underlying sampler produces samples from a distribution close to the given distribution. Intending to design a constrained sampler that passes Barbarik test and achieves significant scalability, we pursued a *test-driven* development methodology to come up with CMSGen [6]. CMSGen is a randomized variation of the Conflict-Driven Clause Learning (CDCL) framework used by the modern SAT solver. As a next step, we equipped the tunable constrained sampler CMSGen with a *computational hardness-based* tester, called ScalBarbarik [2], which allows an expressive measurement of the quality of samples produced by the underlying samplers.

CMSGen can achieve scalability while maintaining the quality of samples — it attained the geometric speed-up of 420× in generating the samples over the prior state-of-the-art samplers. Regarding verifying systems where quality and scalability are vital bottlenecks, the average coverage achieved by the prior state-of-the-art samplers was at most 85.48%; however, samples produced by CMSGen achieved at least 99% coverage for all our considered instances.

Moving towards the testing of a safety-critical system, we proposed a method to quantitatively analyse the information leakage in a system by approximately estimating the entropy [3]. We relied on constrained sampling and counting to design a scalable approach to estimate probabilistically approximate entropy, which could handle instances beyond the reach of baseline approaches.

Future Directions and Outlook

I plan to extend the ideas developed in my research and create new foundations to build explainable, verifiable, scalable, and trustworthy systems. The long-term goal of my research is to make formal methods hand in hand with artificial intelligence a ubiquitous technique in designing a verifiable, efficient and accessible system. This would require an influx of ideas from foundational research and practice. Toward this, I will continue to seek academic and industrial collaborations. A few concrete themes have been described below.

Satisficing Synthesis. Designing and testing system techniques generally work with the unsaid principle of "all-or-nothing" — either provide rigorous theoretical guarantees or do not provide any guarantee at all. Methods with rigorous guarantees sacrifice scalability, and scalable techniques generally do not have guarantees. This "all-or-nothing" approach is the crucial bottleneck in the wide adaptation of synthesis and verification techniques in a real-world setting. Users should have the power to decide the satisficing measure for their requirements depending on the availability of the resources. Accordingly, they should be able to tune the available methods. Our work on designing tunable samplers is just the first step in overcoming this bottleneck. Moving forward, I intend to work towards developing efficient domain-agnostic tunable methods to balance scalability and trustfulness. Given the widespread applicability of tunable automated systems, the successful execution of this research direction promises to have a broader impact.

Interactive Synthesis and Testing. In my research so far, we have assumed that given a complete specification, we want to synthesize an automated system that satisfies the specification. However, this assumption needs to be validated in various real-world applications such as banking, educational, and critical safety systems. We might need to modify or update the specification timely for many different reasons, such as user feedback , sensor updates and more. In such a setting, we would not like to synthesize a new system all over again; instead, we would want to *repair* our synthesized system to handle the updated specification. Moving forward, I intend to develop synthesis techniques that interacts with specifications updates and modify the synthesized system accordingly. Moreover, in regards to the testing, the setting is exciting and challenging. Instead of testing the whole system, again and again, we would like to ensure that the *patch* in the synthesized system works as expected. Moreover, this will require us to devise different sampling strategies to generate the test suites.

References

[1] P. Golia, S. Roy, and K. S. Meel. Synthesis with explicit dependencies. In *Proceedings of Design, Automation and Test in Europe (DATE)*, 2023. Best Paper Award Nomination.

[2] M. Soos, P. Golia, S. Chakraborty, and K. S. Meel. On quantitative testing of samplers. In *Principles and Practice of Constraint Programming (CP)*, 2022.

[3] P. Golia, B. Juba, and K. S. Meel. A scalable shannon entropy estimator. In *Proceedings of International Conference on Computer Aided Verification (CAV)*, 2022.

[4] P. Golia, S. Roy, and K. S. Meel. Program synthesis as dependency quantified formula modulo theory. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[5] P. Golia, F. Slivovsky, S. Roy, and K. S. Meel. Engineering an efficient boolean functional synthesis engine. In *Proceedings of International Conference On Computer Aided Design (ICCAD)*, 2021. Best Paper Award Nomination.

[6] P. Golia, M. Soos, S. Chakraborty, and K. S. Meel. Designing samplers is easy: The boon of testers. In *Proceedings of Formal Methods in Computer Aided Design (FMCAD)*, 2021.

[7] P. Golia, S. Roy, and K. S. Meel. Manthan: A data-driven approach for boolean function synthesis. In *Proceedings of International Conference on Computer-Aided Verification (CAV)*, 2020.