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 is a technique that uses formal specifications to automatically generate systems (such as functions, programs, or circuits) that provably satisfy the requirements of the specification. In my dissertation, I developed a scalable data-driven approach, Manthan, that combines advances in formal methods and machine learning to significantly improve upon the state-of-the-art. Our approach uses constrained sampling to generate data, which is then fed into a machine learning pipeline to generate an initial candidate system. We leverage automated reasoning to repair the candidate system and synthesize a final system that provably satisfies the given specification. Our proposed method achieved significant scalability on real-world instances, and the corresponding publication received a **best paper award nomination** at ICCAD-21. We also extended our general-purpose monolithic synthesis approach to modular designs, in which different components of the required system handle different inputs. This has various applications in fields such as circuit repair and controller synthesis, and the corresponding publication received a **best paper award nomination** at DATE-23.

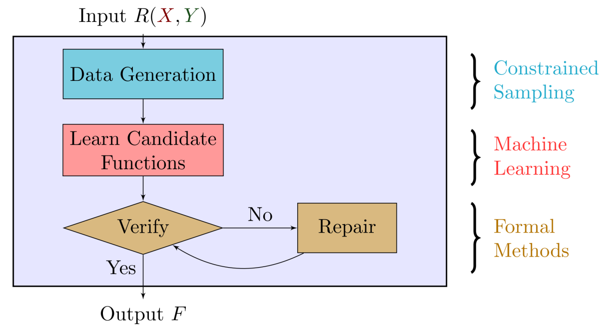
Our work on automated synthesis emphasizes the importance of constrained sampling, which has applications in testing of systems. Constrained sampling is a powerful way of generating test suites. To increase the trustworthiness of the system under test, we need to ensure that the generated test suite effectively captures the given system's functionality, which is generally referred to as the quality of the test suite. In recent years, sampling techniques have been developed that are either scalable to real-world problems or accompanied by theoretical analysis on the distribution of produced samples. However, achieving both scalability and high-quality samples remains a challenge, particularly in applications like verifying the fairness of a deep neural network. In my research, I pioneered a tuneable constrained sampler in a test-driven method to achieve the balance between scalability and quality of produced samples. Moreover, we demonstrate a virtuous cycle between testing and design, and equipped the tuneable constrained 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 is a critical part of ensuring that computing systems function as intended. One common approach to generating test suites for a system is to use constrained sampling, where samples are selected randomly from the set of solutions to a given set of input constraints. Existing constrained sampling techniques either provide theoretical guarantees on the samples they produce or are able to scale to real-world problems. In addition, there are sampler testers such as Barbarik that can determine whether a given sampler produces samples from a distribution close to a specified distribution. To design a constrained sampler that passes the Barbarik test and is highly scalable, we employed a test-driven development methodology to come up with a tunable sampler CMSGen [6], which is a randomized variation of the Conflict-Driven Clause Learning (CDCL) framework used by modern SAT solvers. We then equipped CMSGen with a computational hardness-based tester, called ScalBarbarik [2], which allows for an expressive measurement of the quality of samples produced by the sampler.

In terms of scalability and quality, CMSGen has demonstrated impressive performance. CMSGen has achieved a geometric speed-up of 420× over the prior state-of-the-art samplers. It could produce high-quality samples that could achieve average coverage of at least 99 %, whereas, the prior state-of-the-art samplers achieved at most 85.48% coverage for all our considered instances. This makes CMSGen a promising tool for verifying systems where quality and scalability are critical bottlenecks.

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.