Research Statement

Djordje Zikelic
School of Computing and Information Systems, Singapore Management University
Tel: (65) 6828-0973; Email: dzikelic@smu.edu.sg
6 (Day) 12 (Month) 2023 (Year)

Background

Many industries are undergoing digital transformation and software systems have become ubiquitous in almost all aspects of daily life, including critical infrastructure and business operations. This was additionally fueled by the success of artificial intelligence (AI) and machine learning (ML) technologies, which created the desire to deploy them in general software development. However, the widespread adoption of software and intelligent systems also means that we are becoming increasingly dependent on their inherent safety and security. This is highlighted by safety-critical applications of software and intelligent systems such as automated critical infrastructure, autonomous driving, healthcare or digital finance. For such applications, it is imperative to ensure correctness of software and intelligent systems since incorrect behavior can lead to fatal consequences.

My research focuses on developing techniques for *formally verifying correctness of programs and for certified synthesis and learning of intelligent systems*, towards making software and intelligent systems more safe, secure, robust and trustworthy. By utilizing formal methods which use rigorous mathematical tools and logic to formally reason about systems, my goal is to design *automated methods* for providing *provable guarantees* about system correctness. My work lies at the interface of formal methods (FM), trustworthy AI and ML and programming languages (PL) research.

The central theme of my work is formal automated reasoning about software and intelligent systems in the presence of *uncertainty*. Classical formal methods achieve impressive results in reasoning about deterministic systems and providing YES or NO answers about whether the system satisfies some property. However, uncertainty in software and intelligent systems may arise due to a number of reasons, including interaction with unknown environments, inference from data, randomization, process interleaving or multi-agent systems. In the presence of uncertainty, the behavior of systems is no longer deterministic and their analysis requires more fine-grained reasoning about e.g. the probability with which some property is satisfied or the average, best and worst case behavior. My research vision is to contribute to laying foundations of *automated formal reasoning under uncertainty* towards making software and intelligent systems more safe, secure, robust and trustworthy even in the presence of uncertainty.

Research Areas

In order to achieve this research vision, I consider formal verification, synthesis and certified learning for several general classes of software and intelligent systems in which uncertainty naturally arises. The following four sections outline my research highlights and discuss future perspective for these different classes of systems.

1. Program Analysis of Probabilistic and Non-deterministic Programs (Formal Methods meet Programming Languages)

Probabilistic programs (PPs) are classical programs extended with the ability to sample values from probability distributions and to condition program executions on observed data. They provide a universally expressive framework for specifying and writing stochastic models. Recent years have seen the development of many PP languages (see the Wikipedia page on Probabilistic Programming for a non-exhaustive list), and PPs have found many applications including stochastic networks, security, machine learning and generative AI. The expressivity of PPs makes them a general model for formal analysis since, rather than designing different verification algorithms for each application domain, one can first write the stochastic model of interest as a PP and then focus on its analysis. In my work, I also study the extension of PPs with non-determinism, which allows us to model unknown user inputs or interleaving of processes in cases where these cannot be modeled via probability distributions, or to abstract away parts of programs towards simplifying program analysis. My work focuses on program analysis of *termination, reachability, safety* and *cost* properties in PPs.

Almost-sure termination. Probability 1 (a.k.a. almost-sure) termination is a fundamental property of stochastic models and PPs that is necessary for the correctness of most statistical inference algorithms. However, this property is typically not checked in the existing statistical inference tools, which raises concerns regarding their use for analysis and decision making in safety-critical applications. My work resulted in a compositional framework for proving almost-sure termination in PPs via the novel notion of *generalized lexicographic ranking supermartingales (GLexRSMs)* [1]. These generalize lexicographic ranking functions for non-probabilistic programs to the setting of PPs, which are a classical certificate for proving termination and lie at the core of many modern termination provers for non-probabilistic programs. Our method fully automates the computation of GLexRSMs in affine arithmetic PPs and is able to prove almost-sure termination in classes of examples that no prior automated method could handle.

Quantitative termination, reachability and safety. Computation of bounds on termination, reachability and safety probability (a.k.a. quantitative analysis) allows us to reason about the probability of the system modeled as a PP reaching some specified target or unsafe states. My work resulted in some of the first fully automated methods for solving these problems in PPs. At the core of my approach lie *stochastic invariants* [2], which generalize classical program invariants to the setting of PPs. We showed that stochastic invariants can be used to design sound and complete certificates for computing bounds on the probability of termination, reachability and safety in PPs [3] and designed some of the first fully automated algorithms and prototype tools for the computation of these certificates in PPs [2,3].

Non-deterministic programs. I also work on program analysis of non-deterministic programs, which more appropriately capture uncertainty due to e.g. interleaving of processes, unknown user input, missing procedure implementation or abstraction.
On this front, my work lead to the first method for detecting non-termination bugs in polynomial arithmetic programs that provides relative completeness guarantees [4]. This means that our method is *guaranteed* to catch non-termination bugs of certain

form. Apart from being theoretically appealing, these guarantees also translate to impressive practical performance and our tool RevTerm catches more non-termination bugs than any state of the art tool in the TermComp 2019 competition.
In collaboration with researchers at Amazon, I also worked on computing bounds on the difference in cost usage between two program versions towards automatically detecting performance regressions induced by code changes [5]. Our method is the first sound method for this problem that does not require two program versions to be syntactically aligned but is applicable to general program pairs. This project sparked interest in both academia and industry – it was featured in the Amazon Science blog and was also invited for presentation at the Infer Practitioners 2021 workshop that is organized by program analysis researchers at Meta.

Future perspective. While recent years have seen a lot of work on PP analysis, there is still a significant gap between what modern non-probabilistic program analyzers have achieved and what current methods for PP analysis can do. My long-term goal is to close this gap and to advance the analysis of PPs with non-determinism along 3 axes: *language expressivity, richer properties* and *scalability*. With respect to language expressivity, prior work on automated PP analysis has predominantly focused on programs with numerical datatypes. My goal is to extend the existing approaches to support PPs with arrays and heap manipulation. With respect to richer properties, my goal is to consider more expressive properties of PPs going beyond termination, reachability and safety. Finally, in order to scale PP analysis to very large PPs, I believe that we should develop compositional methods as the ones that achieved impressive results in non-probabilistic program analysis.

2. Certified Learning and Formal Verification of Learned Systems (Formal Methods meet Trustworthy AI and ML)

The tremendous success of ML has sparked interest into deploying ML solutions in a broad range of application domains, with safety-critical applications not being an exception. However, the lack of explainability and interpretability of many learned models raises serious concerns regarding their safety and security. In order to eliminate these concerns and provide the necessary level of trust, we need methods that (1) help ML algorithms to learn models that are correct with respect to the desired specification, and (2) allow us to prove that learned models are truly safe. My work on this front focuses on the development of formal methods for certifiable learning and for formal verification of learned models, with a particular focus on neural networks. I study this problem in two settings – that of neural controllers for stochastic environments and that of neural networks in isolation.

Neural controllers for stochastic environments. Learning-based methods, and in particular reinforcement learning (RL), have shown enormous potential for solving challenging control problems in continuous environments that classical control methods struggle with. However, they also raise concerns regarding safety and explainability of learned controllers. While recent years have seen increased interest in the certification of learned controllers, most existing methods study control in deterministic environments without taking environment uncertainty into account.
My work resulted in the *first framework for certified learning and formal verification of neural controllers in discrete-time stochastic control systems* [6,7]. The core idea behind the framework is to learn a neural controller *together with a neural certificate,*

which proves that the property of interest is satisfied. The neural certificate is then formally verified to be correct. We designed certificates and a method for learning neural controllers and certificates in stochastic systems for several classes of properties, including reachability, safety, reach-avoidance and stability [6,7,8,9]. In each case, the certificate is a carefully designed martingale-like object. Martingales are a class of stochastic processes from probability theory and our design of certificates builds on deep mathematical results from probability theory. We also proposed a compositional framework for properties defined as compositions of different objectives [10]. Our implementation is able to successfully learn and formally verify neural controllers and certificates for a range of highly non-linear stochastic control tasks and properties that were beyond the reach of prior methods. Furthermore, our method can also be used to *formally verify* neural controllers learned via other methods or even to *repair* incorrect neural controllers.

Neural networks in isolation. My work also studies certified learning and formal verification of adversarial robustness and safety properties in neural networks in isolation. There is a large body of work on analyzing these two properties. However, most works consider real arithmetic idealizations of neural networks in which the values of all neurons are treated as real numbers and where rounding errors in computations or inherent uncertainty in network's prediction are ignored. My work considers two popular architectures that address these problems, namely quantized neural networks (QNNs) [11,12] and Bayesian neural networks (BNNs) [13].

Future perspective. The synergy of ML and formal methods has potential to revolutionize control under safety constraints. On one hand, (deep) learning allows us to fit neural controllers to extremely complicated environments by learning from data. Learning alone already produces very promising controllers, as evidenced by empirical studies. On the other hand, formal methods allow us to formally verify these controllers, ultimately making them safe and trustworthy. My research goal is to realize the potential of this synergy of ML and FM by advancing it along 2 axes:

- **Learning-based stochastic control**. In order to get us closer to deployable methods for certified learning-based control, my plan is to consider *richer classes of models*, *better architectures for neural controllers and certificates* and to provide support for *richer specifications,* the latter going beyond reachability, safety and stability, ideally allowing users to specify properties belonging to some general temporal logic such as pLTL. I am also interested in compositional aspects of learning, where hard problems can be solved by decomposing them into a series of simpler subtasks, as we did in [10].
- **Safe RL**. In control theory, one typically assumes a model of the system and solves the problem with respect to the model. In contrast, the goal of RL is to learn good controllers from data alone, without assuming the model. My goal here is to explore how we could improve performance of existing safe RL algorithms or design novel ones by making them learn controllers together with certificates of safety constraint satisfaction.

3. Formal Policy Synthesis in Markov Models (Formal Methods meet Trustworthy AI and Planning)

The work in the previous section uses the synergy of ML and FM to solve control problems in *continuous* stochastic environments that are beyond the reach of

classical control theory and formal methods approaches. In this section, we consider an orthogonal problem of solving control problems in *finite-state* stochastic environments. Formal methods have been used extensively in this area, particularly in solving risk-averse planning problems in finite-state Markov models such as MDPs, POMDPs and stochastic games. In finite-state Markov models, formal methods achieve impressive scalability and can synthesize policies with formal guarantees on a rich class specifications belonging to classical temporal logics such as pLTL or pCTL. For instance, one can synthesize policies which guarantee that ‘’*the probability of a system run ever reaching an unsafe state is at most 0.01%*’’. Such specifications are defined over *system runs*.

However, existing methods do not allow synthesis of policies with guarantees on specifications defined *over probability distributions over system states* that the system semantics induce at each time step. In this view, we treat Markov models as discrete-time transformers which give rise to a new probability distribution over states at each time step, and specify properties with respect to these distributions. For instance, existing methods cannot solve formal policy synthesis problem with respect to the specification *‘’at every time, the probability of the system being in an unsafe state is at most 0.01%’’*. As it turns out, this specification is not expressible in pCTL*. However, such safety constraints naturally arise in certain applications such as control of chemical networks, robot swarms or traffic networks. The problem that has recently captivated my interest is how to enable formal policy synthesis in Markov models with respect to *distributional specifications* such as the one above.

Formal policy synthesis for distributional safety. My initial work on this problem resulted in the first automated method for formal policy synthesis in finite-state MDPs with provable guarantees on *distributional safety specifications* [14], such as the example above. As we show in our work, this turns out to be an incredibly hard problem that may even require randomized and infinite memory policies. In order to solve this problem, our method combines insights from template-based synthesis and invariant generation in programs and it simultaneously synthesizes a policy together with a *distributional invariant* that formally proves distributional safety. Our method reduces to two algorithms that differ in their efficiency and generality – the first which considers positional policies but allows for a more efficient synthesis, and the second can synthesize symbolic representations of infinite-memory policies.

Future perspective. My research goal is to provide foundations of formal policy synthesis with respect to distributional specifications in two ways. First, my aim is to consider *richer specifications* going beyond distributional safety. Second, our method for distributional safety provides the first step towards solving this problem but is not very scalable. My goal is to improve *scalability* by coupling it with different search strategies or considering different synthesis techniques.

4. Broader Perspective and Interdisciplinarity

While my primary research focuses on formal verification and synthesis and certified learning for programs and intelligent systems, I have broad research interests and enjoy engaging in discussions with researchers working on different problems and finding other applications of automated formal reasoning under uncertainty. One thread of my work is on *bidding games on graphs*, which present a natural model for

stateful and ongoing auctions. Bidding games were used to model auctions for online advertisement slots, scheduling, and there were even efforts to formalize some blockchain attacks as bidding games. As it turns out, although defined as deterministic models, bidding games admit a beautiful connection to *stochastic games*. We studied this connection and solved bidding games on graphs under several bidding mechanisms [15,16,17,18]. I also contributed to the study of social balance on networks in statistical physics, where the analysis can be reduced to studying Markov chains and evolutionary graph theory [19].

Selected Publications and Outputs

See my DBLP or Google Scholar pages for a complete publication list.

[1] K. Chatterjee, E. K. Goharshady, P. Novotný, J. Zárevúcky, Đ. Žikelić. *On Lexicographic Proof Rules for Probabilistic Termination*. In Formal Aspects of Computing 35(2), (FAC 2023)

[2] K. Chatterjee, P. Novotný, Đ. Žikelić. *Stochastic Invariants for Probabilistic Termination*. In 44th ACM SIGPLAN Symposium on Principles of Programming Languages, (POPL 2017)

[3] K. Chatterjee, A. K. Goharshady, T. Meggendorfer, Đ. Žikelić. *Sound and Complete Certificates for Quantitative Termination Analysis of Probabilistic Programs*. In 34th International Conference on Computer Aided Verification, (CAV 2022)

[4] K. Chatterjee, E. K. Goharshady, P. Novotný, Đ. Žikelić. *Proving Non-termination by Program Reversal.* In 43rd ACM SIGPLAN Conference on Programming Language Design and Implementation, (PLDI 2021)

[5] Đ. Žikelić, B. Y. E. Chang, P. Bolignano, F. Raimondi. *Differential Cost Analysis with Simultaneous Potentials and Anti-potentials.* In *44th ACM SIGPLAN Conference on Programming Language Design and Implementation*, (PLDI 2022)

[6] M. Lechner, Đ. Žikelić, K. Chatterjee, T. A. Henzinger. *Stability Verification in Stochastic Control Systems via Neural Network Supermartingales.* In 36th AAAI Conference on Artificial Intelligence, (AAAI 2022)

[7] Đ. Žikelić, M. Lechner, T. A. Henzinger, K. Chatterjee. *Learning Control Policies for Stochastic Systems with Reach-avoid Guarantees.* In 37th AAAI Conference on Artificial Intelligence, (AAAI 2023)

[8] K. Chatterjee, T. A. Henzinger, M. Lechner, Đ. Žikelić. *A Learner-Verifier Framework for Neural Network Controllers and Certificates of Stochastic Systems.* In 29th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, (TACAS 2023)

[9] M. Ansaripour, K. Chatterjee, T. A. Henzinger, M. Lechner, Đ. Žikelić. *Learning Provably Stabilizing Neural Controllers for Discrete-Time Stochastic Systems.* In 21st International Symposium on Automated Technology for Verification and Analysis, (ATVA 2023)

[10] Đ. Žikelić, M. Lechner, A. Verma, K. Chatterjee, T. A. Henzinger. C*ompositional Policy Learning in Stochastic Control Systems with Formal Guarantees*. In 37th Conference on Neural Information Processing Systems, (NeurIPS 2023)

[11] T. A. Henzinger, M. Lechner, Đ. Žikelić. *Scalable Verification of Quantized Neural Networks.* In 35th AAAI Conference on Artificial Intelligence, (AAAI 2021)

[12] M. Lechner, Đ. Žikelić, K. Chatterjee, T. A. Henzinger, D. Rus. Quantization-aware Interval Bound Propagation for Training Certifiably Robust Quantized Neural Networks. In 37th AAAI Conference on Artificial Intelligence, (AAAI 2023)

[13] M. Lechner, Đ. Žikelić, K. Chatterjee, T. A. Henzinger. *Infinite Time Horizon Safety of Bayesian Neural Networks.* In 35th Conference on Neural Information Processing Systems, (NeurIPS 2021)

[14] S. Akshay, K. Chatterjee, T. Meggendorfer, Đ. Žikelić. *MDPs as Distribution Transformers: Affine Invariant Synthesis for Safety Objectives.* 35th International Conference on Computer Aided Verification, (CAV 2023)

[15] G. Anvi, T. A. Henzinger, Đ. Žikelić. *Bidding Mechanisms in Graph Games*. Journal of Computer and System Sciences 119, (JCSS 2021)

[16] G. Anvi, I. Jecker, Đ. Žikelić. *Infinite-Duration All-Pay Bidding Games.* ACM-SIAM Symposium on Discrete Algorithms, (SODA 2021)

[17] G. Anvi, I. Jecker, Đ. Žikelić. *Bidding Graph Games with Partially-Observable Budgets.* 37th AAAI Conference on Artificial Intelligence, (AAAI 2023)

[18] G. Anvi, T. Meggendorfer, S. Sadhukhan, J. Tkadlec, Đ. Žikelić. *Reachability Poorman Discrete-Bidding Games.* 26th European Conference on Artificial Intelligence, (ECAI 2023)

[19] K. Chatterjee, J. Svoboda, Đ. Žikelić, A. Pavlogiannis, J. Tkadlec. *Social Balance on Networks: Local Minima and Best-edge Dynamics.* Physical Review E 106, (PRE 2022)