Research Statement

Djordje Zikelic
School of Computing and Information Systems, Singapore Management University
Tel: (65) 6828-0973; Email: dzikelic@smu.edu.sg
19 (Day) 12 (Month) 2024 (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 AI systems also means that we are becoming increasingly dependent on their inherent correctness and safety. This is highlighted by safety-critical applications such as automated critical infrastructure, autonomous driving, healthcare or digital finance, where it is imperative to ensure correctness of software and AI systems since incorrect behavior can lead to fatal consequences.

My research is concerned with helping programmers ensure that software and AI systems are correct, safe, and trustworthy. To that end, I study Formal Methods and their applications to Program Analysis and Verification, as well as Trustworthy AI and Safe Autonomy. The long term goal of my work is to advance the theory and automation of formal methods for trustworthy software and AI, especially in the presence of *probabilistic uncertainty*. The two guiding principles in my work are mathematically rigorous correctness guarantees and full automation.

Classical formal methods achieve impressive results in reasoning about deterministic systems and providing YES or NO answers about whether the system satisfies some property of interest. However, uncertainty in software and AI systems may arise due to a number of reasons, including interaction with unknown or noisy 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-case (i.e. expected) behavior. My research goal is to contribute to laying theoretical and algorithmic foundations of automated formal reasoning about probabilistic systems. The long term vision of my work is to make formal methods applicable to probabilistic systems at the same level and scale at which they are currently applicable to non-probabilistic systems, thus making software and AI systems more safe, robust and trustworthy even in the presence of probabilistic uncertainty.

Research Areas

In order to achieve this research vision, I consider automated formal verification, synthesis and certified learning for several general classes of software and AI

systems in which probabilistic uncertainty naturally arises. The following sections outline my research highlights and discuss future perspective.

1. Formal Methods for Program Analysis and Verification (Formal Methods meet Programming Languages)

Much of my work on program analysis and verification focuses on infinite-state probabilistic systems modeled as probabilistic programs. 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 probabilistic 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 are used many applications including stochastic networks, security and privacy protocols, machine learning, robotics 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 probabilistic model of interest as a PP and then focus on its analysis. My work focuses on program analysis with respect to temporal properties such as *termination, reachability or safety,* as well as *cost* properties in PPs.

Quantitative analysis. While most previous works focused on qualitative analysis of PPs and proving probability 1 termination, my key contributions to PP analysis concern quantitative analysis and the computation of bounds on the probability of termination or safety. These bounds allow us to reason about the probability of the system modeled as a PP reaching some desired or undesired sets of states. My work resulted in some of the first fully automated methods for solving these problems in PPs with general unbounded loops. At the core of my approach lie stochastic invariants [1], a notion that we introduced as a generalization of 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 [2], as well as fully automated algorithms and prototype tools for these problems [1,2]. In a more recent work, we also utilized stochastic invariants to design a new method for cost analysis in PPs, which is able to tackle a large class of examples that prior methods could not handle [3]. It should be noted that, while the work on PP analysis is of significant theoretical and algorithmic interest, it also has significant application potential. For instance, we showed that our cost analysis method can formally analyze security of several blockchain protocols, to which prior methods are not applicable [3].

Almost-sure termination. Probability 1 (a.k.a. almost-sure) termination is a fundamental property of probabilistic 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)* [4]. 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.

Non-deterministic programs. I also work on static analysis of non-probabilistic numerical programs. Many existing algorithms for PP analysis and verification build on prior algorithms for non-probabilistic numerical programs. Hence, the program analysis and verification problems for these two classes of programs are deeply connected. In my work, I proposed the first method for detecting non-termination bugs in polynomial arithmetic programs that provides relative completeness guarantees [5]. This means that the method is guaranteed to catch non-termination bugs of a certain form. These appealing theoretical guarantees translate to excellentpractical performance. Our prototype tool RevTerm outperforms all termination tools that competed in the TermComp'19 competition, both in terms of the number of detected non-termination bugs and in terms of runtime. During my internship at Amazon, I worked on differential cost analysis where the goal is to compute a bound on the difference in cost usage between two program versions and detect potential performance regressions induced by code change. To address this problem, I proposed the first sound method for differential cost analysis that does not require two program versions to be syntactically aligned but is applicable to general program pairs [6]. This work has sparked interest in both academia and industry – it was featured in the Amazon Science blog and it was presented at the Infer Practitioners 2021 workshop that is organized by the Infer static analyzer team at Meta. Finally, our recent work [7] introduces a method for program analysis with respect to linear-time temporal logic (LTL) properties. Again, it gives rise to the first method that provides relative completeness guarantees for polynomial arithmetic programs, while also showing excellent practical performance and outperforming state of the art tools.

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. Formal Methods for Trustworthy AI and Safe Autonomy (Formal Methods meet AI and ML)

The tremendous success of AI has sparked interest in deploying AI-enabled solutions in a broad range of application domains, with safety-critical applications not being an exception. However, the lack of correctness guarantees and interpretability of many learned models raises serious concerns regarding their safety and trustworthiness. In order to eliminate these concerns and provide the

necessary level of trust, we need methods that (1) help ML algorithms learn models that are correct with respect to the desired specification, and (2) allow us to guarantee that learned models are truly correct. 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 control for safe autonomy, and that of analysis of feed-forward neural networks in isolation.

Neural control for safe autonomy. Learning-based methods, and in particular reinforcement learning (RL), have shown enormous potential for solving challenging control tasks that classical control methods cannot tackle. However, they also raise concerns regarding the correctness 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 [8,9]. The core idea behind the framework is to learn a neural controller together with a neural certificate of correctness, which provides a proof that the property of interest is satisfied. The neural certificate is then formally verified to be correct. We designed certificates and a framework for their learning and formal verification for several classes of properties, including reachability, safety, reach-avoidance and stability [8,9,10,11]. In each case, the certificate is a carefully designed martingale-like object. Martingales are a class of stochastic processes from probability theory, and the design of martingale certificates builds on deep mathematical results from probability theory. We also proposed a compositional framework for properties defined as compositions of different objectives [12]. 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, the 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) [13,14] and Bayesian neural networks (BNNs) [15]:

- **QNNs.** Quantization reduces the computational cost of evaluating a neural network by reducing the arithmetic precision of its computations and it has been widely adopted in industry. We studied formal verification of QNNs. On the theory side, we proved that the formal verification problem for QNNs over bit-vector specifications and linear arithmetic is PSPACE-hard [13], in contrast to the formal verification problem for real arithmetic neural networks and linear arithmetic specifications which is known to be NP-complete. On the practical side, we designed quantization-aware interval bound propagation (QA-IBP), the first procedure for training provably robust QNNs

[14]. Our certifiable training and verification procedures for QNNs present the state of the art in this line of work.

- **BNNs.** BNNs have established themselves as a go-to architecture for learning uncertainty in the network's prediction. We proposed the first method for certifiable training of BNNs with respect to safety specifications, by first computing a set of safe weight vectors and then altering the BNN's weight posterior to reject samples outside this set [15].

Future perspective. The synergy of ML and formal methods has the 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 [12].
- **Safe RL with certificates**. 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 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 with respect to distributional specifications. My work on this problem resulted in the first automated method for formal policy verification and synthesis in finite-state MDPs with provable guarantees on *distributional reachability, safety and reach-avoidance specifications* [16,17], 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 certificate* that formally proves distributional specification. 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 automated formal policy verification and synthesis with respect to distributional specifications in two ways. First, my aim is to consider *richer specifications* going beyond distributional reachability and safety. Second, our method for distributional reachability and 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 two primary research areas are formal methods for program analysis and verification and formal methods for trustworthy AI and safe autonomy, I am also interested in other application domains where probabilistic system verification can make an impact. To that end, I enjoy engaging in discussions and collaborating with researchers from diverse areas. This has lead to some exciting research and novel applications of probabilistic system verification.

One thread of my past work is on bidding games on graphs, which provide a natural model for stateful and ongoing auctions. Bidding games have been used to model auctions for online advertisement slots, scheduling of concurrent processes, and there were even efforts to formalize some blockchain attacks as bidding games. In my work, I studied several bidding mechanisms as well as games with partially observable bids [18,19,20,21], resulting e.g. in a somewhat surprising use of martingale theory for the design of optimal bidding strategies [19]. 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 [22]. Finally, in collaboration with cryptography researchers, we showed that the analysis of selfish mining attacks on efficient proof system blockchains (e.g. those based on

Proof-of-Stakeand Proof-of-Space protocols) can be modeled as a probabilistic model checking problem. This lead to the first fully automated analysisof selfish mining attacks on efficient proof system blockchains and some very interesting observations of practical relevance [23]. In contrast, all prior analyses were based on tedious pen-and-paper work, which quickly becomes intractable.

Selected Publications and Outputs

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

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

[2] 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)

[2] K. Chatterjee, A. K. Goharshady, T. Meggendorfer, Đ. Žikelić. *Quantitative Bounds on Resource Usage of Probabilistic Programs*. In ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, (OOPSLA 2024)

[4] 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)

[5] 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)

[6] 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)

[7] K. Chatterjee, A. K. Goharshady, E. K. Goharshady, M. Karrabi, Đ. Žikelić. *Sound and Complete Witnesses for Template-based Verification of LTL Properties on Polynomial Programs*. In 26th International Symposium on Formal Methods, (FM 2024)

[8] 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)

[9] 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)

[10] 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)

[11] 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)

[12] 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)

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

[14] 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)

[15] 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)

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

[17] S. Akshay, K. Chatterjee, T. Meggendorfer, Đ. Žikelić. *Certified Policy Verification and Synthesis for MDPs under Distributional Reach-Avoidance Properties.* In 33rd International Joint Conference on Artificial Intelligence, (IJCAI 2024)

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

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

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

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

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

[23] K. Chatterjee, A. Ebrahimzadeh, M. Karrabi, K. Pietrzak, M. Yeo, Đ. Žikelić. *Fully Automated Selfish Mining Analysis in Efficient Proof Systems Blockchains.* In 43rd ACM Symposium on Principles of Distributed Computing, (PODC 2024)