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) 2025 (Year)

Background

Formal methods are mathematically rigorous techniques for provably correct specification, verification and synthesis of computing systems. Classical formal methods achieve impressive results in reasoning about deterministic computing systems and providing YES or NO answers about whether the system satisfies the property of interest. However, such boolean reasoning becomes insufficient in the presence of probabilistic uncertainty. Probabilistic systems arise in many areas of computer science, including randomized algorithms, stochastic networks, security and privacy protocols, cyber-physical systems, and robotics. The importance of probabilistic systems has been further proliferated by the rapid advance of artificial intelligence (AI) technologies, where uncertainty arises due to e.g. learning from data or planning under uncertainty. In the presence of probabilistic uncertainty, computing system behavior is no longer deterministic, and their formal analysis and verification require more fine-grained reasoning about the probability with which a property is satisfied or the average-case behavior. However, existing formal methods for probabilistic systems are significantly lagging behind their counterparts for non-probabilistic systems, both in terms of the class of properties and in terms of the size of systems that they can formally analyze. This raises concerns regarding the correctness of software and AI solutions used in safety-critical settings that exhibit uncertain behavior, such as critical infrastructure, aircraft and autonomous vehicle software, or healthcare devices.

The long-term goal of my work is to contribute to laying *theoretical and algorithmic foundations of formal reasoning about general probabilistic systems*. By general probabilistic systems, we mean probabilistic systems that may be either finite- or infinite-state, going beyond the classical probabilistic model checking paradigm which is restricted to finite-state probabilistic systems. On the other hand, the central application domains that I focus on are *trustworthy AI and safe autonomy*. Reasoning about probabilistic uncertainty presents one of the key challenges in solving these problems, and my goal is to use formal methods to make AI and autonomous systems provably safe and trustworthy even in the presence of probabilistic uncertainty. Finally, I am also interested in *broader application domains* where formal methods for probabilistic systems can make an impact. This gives rise to the three research directions that I pursue:

1. **Foundations: Formal methods for probabilistic models and programs.**
2. **Main application domains: Trustworthy AI and safe autonomy.**
3. **Broader application domains and interdisciplinarity.**

My research lies at the interplay of several areas of computer science, with **formal methods** and **programming languages** constituting the theoretical core of my work, and **artificial intelligence**, **machine learning** and **control theory** being the key application domains.

Research Areas

1. Formal Methods for Program Analysis and Verification

Probabilistic system verification has for long been recognized as an important problem in the formal methods community. However, most existing methods and tools focus on probabilistic model checking and consider finite-state probabilistic systems, while not being readily applicable to infinite-state probabilistic systems. Many probabilistic systems arising in practice are infinite-state, with examples including any probabilistic protocol that performs sampling from a continuous distribution such as uniform or normal distribution, or any control or planning task in a continuous stochastic environment. To address this gap, recent years have seen increased interest in formal verification of probabilistic programs (PPs). 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 both finite- and infinite-state probabilistic systems. 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 write the probabilistic system of interest as a PP to be analyzed. However, the state of the art of formal and automated verification of probabilistic programs is significantly lagging behind its counterpart for non-probabilistic programs, and closing this gap remains a challenging open problem. The goal of my research is to bring us closer to closing this gap and to lay *theoretical and algorithmic foundations of formal reasoning about general (i.e. both finite- and infinite-state) probabilistic models and programs*.

Methodology: Supermartingale certificates. My approach to formal and automated reasoning about probabilistic models and programs is based on the use of supermartingale certificates. Supermartingale certificates are formal and locally checkable proof rules for reasoning about different properties of probabilistic systems. Their name is due to their connection to supermartingale processes from advanced probability theory, which lie at the core of formal correctness guarantees provided by supermartingale certificates that apply to both finite- and infinite-state probabilistic systems. On the other hand, supermartingale certificates have a simple syntactic form, which allows for their fully automated computation by using classical template-based synthesis techniques from program verification (see non-probabilistic program verification paragraph below) or by using deep learning (a novel method introduced in my work, which will be discussed in Section 2). Hence, supermartingale certificates provide the following three desirable features: (1) formal correctness guarantees, (2) applicability to both finite- and infinite-state probabilistic systems, and (3) fully automated reasoning.

Past research highlights. Early works on formal verification via supermartingale certificates focused either on proving probability 1 (a.k.a. qualitative) termination and reachability in probabilistic programs, or on probability $p \in [0,1]$ (a.k.a. quantitative) safety in stochastic dynamical and control systems. My work introduced the *first supermartingale certificates and automated algorithms for a significantly larger class of specifications*, thus enabling their broader applicability and adoption beyond these two contexts:

- **Quantitative reachability, safety and reach-avoid specifications**. My early work focused on designing supermartingale certificates for quantitative reachability, safety, and reach-avoid specifications, thus unifying and significantly generalizing previous supermartingale certificates. At the core of my approach lies the idea to decompose a quantitative reachability or reach-avoid specification into (1) qualitative reachability and (2) quantitative safety specifications. This decomposition is achieved through the novel notion of stochastic invariants [1] that generalize classical program invariants to the setting of probabilistic models and programs. We proved that such a decoupling provides a sound [1] and complete [2] approach for reasoning about quantitative reachability and reach-avoidance. Based on this decoupling, we designed novel supermartingale certificates and fully automated algorithms for formal verification of quantitative reachability and reach-avoid specifications [1,2,9]. More recently, we applied this idea to cost (or resource) analysis in probabilistic models and programs and designed a supermartingale certificate for cumulative expected cost analysis and a fully automated algorithm for its computation [3]. This method was able to solve the cost analysis problem for a large class of PPs that no prior method could handle, including interesting and novel applications to blockchain protocols.

- **Quantitative omega-regular specifications.** In one of our most recent works [24], we generalized the ideas above to design the first supermartingale certificate and fully automated algorithm for the *general class of quantitative omega-regular specifications* in probabilistic models and programs. Omega-regular specifications are a rich class of specifications that subsume linear temporal logic (LTL) and computation tree logic (CTL), hence this result makes a significant step forward in terms of the class of properties that formal methods for infinite-state probabilistic systems can analyze.

- **Compositional algorithms for qualitative reachability**. Compositionality is one of the key factors that contributed to the scalability of modern non-probabilistic program verifiers. However, it remains largely unexploited and unexplored in the context of probabilistic system verification. My work resulted in a compositional framework for proving probability 1 termination in PPs via novel generalized lexicographic ranking supermartingale (GLexRSM) certificates [4]. These generalize lexicographic ranking functions for non-probabilistic programs, which are a classical certificate for proving non-probabilistic program termination that lie at the core of many termination provers. This work presents a first step towards a much bigger goal of designing compositional certificates and algorithms for general properties in probabilistic models and programs.

- **Non-probabilistic program verification**. When it comes to automating PP verification, my work builds upon the classical template-based synthesis method from non-probabilistic program verification, which reduces program verification to a constraint-solving problem. However, my work also led to some new state of the art approaches for non-probabilistic program verification and to new algorithms and tooling support for template-based synthesis. We 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 excellent practical performance. Our prototype 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. I also 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, we 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 [5]. This work has sparked interest in both academia and industry -- it was featured in the Amazon Science blog and was presented at the Infer Practitioners 2021 workshop that is organized by the Infer static analyzer team at Meta. Finally, our recent work [5] introduces a method for program verification with respect to linear 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. My work also led to novel algorithms for more efficient template-based synthesis [25] and to PolyQEnt [26] which is the first tool that automates the reduction performed by template-based synthesis from polynomial templates to polynomial constraint-solving.

Future perspective. Recent years have seen great progress in PP verification. However, there is still a large gap between modern non-probabilistic program verifiers and existing methods for probabilistic model and program verification. My long-term goal is to bring us closer to closing this gap and to advance the theory and automation of formal analysis of probabilistic models and programs along three different axes:

- **Programming language expressivity**. Prior work on automated PP verification has predominantly focused on programs with numerical datatypes. However, many important applications of PPs involve arrays and heap operations. One of my future research goals is to study how we could extend the above methods to PPs with arrays and heap operations.
- **Compositional algorithms for PP verification**. While recent years have seen a rapid growth of interest in PP verification, existing automated methods are mostly capable of handling academic examples and do not scale to large PPs. The key reason behind this scalability bottleneck is that existing algorithms for PP verification are not compositional, meaning that they need to perform a one-shot analysis of the whole PP. In contrast, compositionality is one of the key factors that contributed to the scale at which modern program verifiers perform. As one of my future research directions, I plan to explore the possibility of performing compositional PP verification.
- **Relational PP verification**. Much of the existing work on (infinite-state) probabilistic model and program verification focuses on properties such as termination, reachability, safety, or cost. These are all examples of unary properties that are defined with respect to a single program and a single program execution. However, there are also many important applications in which one needs to consider relational properties that are defined with respect to a program or an execution pair. An important example of a

relational property is program equivalence, which is significant in the context of PP compiler correctness. In two recent works, we proposed the first automated method for equivalence refutation in PPs that provides formal correctness guarantees [27,28].

2. Formal Methods for Trustworthy AI and Safe Autonomy

The tremendous success of AI has sparked interest in deploying AI-enabled solutions in a broad range of application domains, with safety-critical applications such as cyber-physical systems not being an exception. However, the lack of correctness guarantees and interpretability of many learned models raises serious concerns regarding their safety and trustworthiness. To eliminate these concerns and to provide the necessary level of trust, we need methods that help ML algorithms learn models that are correct with respect to the desired specification and allow us to guarantee that the learned models are truly correct. My work on this front focuses on the development of methods for guaranteed safe AI-enabled autonomy, by combining (1) neural control methods which leverage the power of deep learning to solve challenging control problems that have been out of the reach of classical control methods with (2) formal methods that can provide formal guarantees on the correctness of learned neural controllers.

Methodology: Neural control with certificates. Neural control with certificates is a promising approach to formally ensuring the correctness of neural controllers that has seen a surge of interest within both control theory and AI communities. The key idea behind this paradigm is that neural controllers should not be learned and verified alone -- rather, they should be learned and verified together with a neural certificate of their correctness. A certificate function, which provides a formal proof that the property of interest is satisfied, is parametrized as a neural network and learned jointly with the neural controller. This is achieved by designing a loss function which incorporates the defining properties of the certificate function, whose minimization then gives rise both to a neural controller and a neural certificate. This way, the learning process is guided towards learning a neural controller that satisfies the specification of interest. Formal guarantees on the neural controller are then provided by checking that the neural certificate satisfies all certificate conditions. My work builds upon this paradigm to design methods for guaranteed safe AI-enabled autonomy.

Past research highlights. While neural control with certificates has received significant attention over the past few years, prior methods were restricted to deterministic control tasks and did not take environment uncertainty into account. My work addresses this issue by giving rise to the *first framework for neural stochastic control with certificates*. This is achieved through a novel combination of the neural control with certificates paradigm and the supermartingale certificates for probabilistic models and programs. This combination highlights an application on my foundational work on formal methods for probabilistic models and programs in Section 1 to a timely and relevant problem in trustworthy AI and safe autonomy:

- **Neural stochastic control with supermartingale certificates.** Our work resulted in the first framework for neural stochastic control with certificates in discrete-time stochastic control systems [8,9,10,11,12]. As mentioned above, this is achieved through a novel combination of the neural control with certificates paradigm and the supermartingale certificates for probabilistic models and programs. The framework is applicable to neural stochastic control with respect to several classes of specifications, including quantitative reachability, safety, reach-avoidance and qualitative stability [8,9,10,11]. More recently, we extended this framework to a compositional framework for neural control with respect to specifications written in the SpectRL specification logic, which allows all sequential and boolean combinations of reach-avoid specifications [12]. Our implementation is able to successfully learn and formally verify neural controllers and supermartingale 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 as we showed most recently to even repair incorrect neural controllers [29].

- **Feed-forward neural network verification.** While my primary focus is on certified learning and formal verification of neural controllers, in my work I also considered certified learning and formal verification of feed-forward neural networks with respect to adversarial robustness and safety specifications. There is a large body of work on analyzing these two properties in feed-forward neural networks. 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. Our work considers two popular architectures that address these problems, namely quantized neural networks (QNNs) [13,14] and Bayesian neural networks (BNNs) [15].

Future perspective. The synergy of ML and formal methods has the potential to revolutionize control under safety constraints and how we design autonomous systems. On one hand, deep learning allows us to fit neural controllers to extremely complicated environments by learning from data. On the other hand, formal methods allow us to guarantee the correctness of neural controllers, ultimately making them safe and trustworthy. My research goal is to realize the potential of this synergy of ML and formal methods for guaranteed safe AI-enabled autonomy, by advancing it along three axes:

- **Neural control with certificates under richer specifications**. My past work on neural control with certificates considered some of the most relevant specifications in control theory, such as reach-avoidance or stability. However, one can specify many other different control tasks. Designing new methods for each new specification is not a feasible approach, and ideally we would like to have a single framework for a sufficiently expressive specification logic which would then be applicable to the whole family of control tasks. To that end, my first goal for future work is to extend the above framework to neural stochastic control with certificates under a probabilistic specification logic, such as pLTL or pCTL. In recent work [24], we introduced the first supermartingale certificates for quantitative omega-regular specifications, which may provide the necessary theoretical foundations for solving this challenge.

- **Multi-agent systems**. Semantically, stochastic control systems are Markov decision processes (MDPs). Thus, the analyses discussed above can be viewed as quantitative analyses for infinite-state MDPs. This raises a natural question of whether we can use supermartingale certificates to formally analyze infinite-state systems that involve more than one agent, i.e. infinite-state stochastic games. The aim of this research direction is to explore the possibility of combining neural control and supermartingale certificates towards obtaining formal guarantees in multi-agent systems
- **Safe reinforcement learning with certificates**. In controller synthesis tasks, one typically assumes a model of the system and solves the problem with respect to the model. In contrast, the goal of reinforcement learning is to learn good controllers from data alone, without assuming the knowledge of the model. My goal here is to explore how we could improve the performance of existing safe reinforcement learning algorithms by building on the above ideas and making them learn not only controllers, but also certificates of correctness for the safety property of interest.

3. Formal Policy Synthesis in Markov Models

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 *''<u>at every time</u>, 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. In the most recent work, we extended our method to a framework for general distributional omega-regular specifications [30].

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)

[24] T. A. Henzinger, K. Mallik, P. Sadeghi, Đ. Žikelić. *Supermartingale Certificates for Quantitative Omega-regular Verification and Control*. In 37th International Conference on Computer Aided Verification (CAV 2025)

[25] K. Chatterjee, E. K. Goharshady, M. Karrabi, H. J. Motwani, M. Seeliger, Đ. Žikelić. *Quantified Linear and Polynomial Arithmetic Satisfiability via Template-based Skolemization.* In 39th AAAI Conference on Artificial Intelligence (AAAI 2025)

[26] K. Chatterjee, A, K. Goharshady, E. K. Goharshady, M. Karrabi, M. Saadat, M. Seeliger, Đ. Žikelić. *PolyQEnt: A Polynomial Quantified Entailment Solver.* 23rd edition of the International Symposium on Automated Technology for Verification and Analysis (ATVA 2025)

[27] K. Chatterjee, E. K. Goharshady, P. Novotný, Đ. Žikelić. *Equivalence and Similarity Refutation for Probabilistic Programs.* In 46th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2024)

[28] K. Chatterjee, E. K. Goharshady, P. Novotný, Đ. Žikelić. *Refuting Equivalence in Probabilistic Programs with Conditioning.* In 31st International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2025)

[29] E. Yu, Đ. Žikelić, T. A. Henzinger. *Neural Control and Certificate Repair via Runtime Monitoring.* In 39th AAAI Conference on Artificial Intelligence (AAAI 2025)

[30] S. Akshay, O. Neysari, Đ. Žikelić. *Omega-regular Verification and Control for Distributional Specifications in MDPs.* In 36th International Conference on Concurrency Theory (CONCUR 2025)