Research Statement

Paulius Dilkas

University of Toronto & Vector Institute, Toronto, Canada
`https://dilkas.github.io/`

Efficient probabilistic inference remains a significant challenge, with applications including probabilistic programming, graphical models, statistical-relational systems, and neuro-symbolic systems [1]. Many of these systems exhibit patterns of independence and symmetry that traditional inference methods often fail to recognise. My research addresses this issue by developing *model counting* algorithms for propositional and first-order logic (FO), which use these patterns to enhance inference efficiency. By leveraging the inherent structure of these problems, my algorithms can manage larger datasets and find polynomial-time solutions for scenarios that might otherwise require exponential time.

Model counting is a dynamic research area with extensive connections across computer science and mathematics. It benefits from and contributes to fundamental data structures like automata, circuits, and decision diagrams and intersects with database theory, descriptive complexity, combinatorics, and finite model theory. The applications of model counting extend beyond probabilistic inference and learning to areas including formal verification, quantum computing, and explainable artificial intelligence (AI). For instance, it aids in assessing software reliability, analysing properties of neural networks like fairness and robustness, simulating quantum circuits on classical machines, verifying circuit equivalence, and understanding the complexity of explainability algorithms. However, the efficiency of a solution to a counting problem often depends on how we formulate the problem.

My research focuses on developing novel counting algorithms and knowledge representation formats that are sufficiently rich to capture the simplicity of any given problem. In other words, if a human can examine the problem, perform a few simple calculations, and provide the answer, then a computer should not require significantly more resources. My contributions include a scalable *first-order model counting* algorithm that constructs *recursive functions* to encapsulate the solution to the counting problem. I have also developed a top-performing algorithm for probabilistic inference in *Bayesian networks* based on a generalised version of (weighted) model counting. My work in experimental algorithmics involves two new random instance generators for logic programs and model-counting instances based on *constraint solvers* and a new technique for controlling *treewidth*. In the future, I aim to develop efficient algorithms for model *sampling*, theories employed by *satisfiability modulo theories* solvers, fragments of FO that surpass the capabilities of current model counting algorithms, and more expressive logics such as monadic second-order logic and FO with counting.

My long-term vision is to facilitate efficient counting and sampling across various input formats. Advancements can be made by adapting counting techniques to new domains and improving existing algorithms to make them more effective and versatile. These developments will expand the application of probabilistic and quantitative methods across AI systems, formal methods, and beyond.

Research Thrust 1: Propositional Model Counting and Probabilistic Inference

One of the primary focuses of my work is *weighted model counting* (WMC) and its applications in probabilistic inference. WMC extends the well-known *Boolean satisfiability* (SAT) problem. SAT asks whether a given propositional formula has a model. The *propositional model counting* problem seeks to determine the number of models. WMC goes a step further by asking for the *weighted* sum of these models.

My research revisits the foundations of WMC and explores generalisations of key definitions to enhance conceptual clarity and practical efficiency. We begin by developing a measure-theoretic perspective on WMC, which introduces a new and more general method for defining the weights of an instance [5]. This new representation can be as concise as standard WMC or expand to accommodate less-structured probability distributions. We demonstrate the performance benefits of this new format by creating a novel WMC encoding for Bayesian networks.

Next, we show how to transform existing WMC encodings for Bayesian networks into this more general format [6]. By combining the strengths of this more flexible representation with the techniques of existing encodings, we achieve further efficiency improvements in probabilistic inference, surpassing all other WMC-based approaches on most benchmarks.

Research Thrust 2: First-Order Model Counting and Lifted Inference

In propositional logic, encoding a claim such as "after flipping n coins, exactly one came out as heads" involves iterating over every pair of coins and asserting that at most one of them showed heads. In FO, this can be expressed more succinctly as $(\exists x \in \mathrm{Coins.\ Heads}(x)) \land (\forall x, y \in \mathrm{Coins.}\ (\mathrm{Heads}(x) \land \mathrm{Heads}(y)) \Rightarrow x = y)$. The distinction is not merely conceptual; this formulation allows the algorithm to recognise that all coins are exchangeable and leverage this symmetry for greater efficiency. The performance improvement can be exponential: while propositional model counting is #P-complete, counting in FO—known as *first-order model counting* (FOMC)—can often be performed in polynomial time with respect to the domain size. Similarly to WMC, the *weighted* variant of FOMC is used for polynomial-time inference on statistical-relational models, such as Markov logic networks. More broadly, the practice of accelerating probabilistic inference by exploiting symmetries is termed *lifted inference*.

My early work on FOMC introduced a new *knowledge compilation*-based model counter, Crane, demonstrating its ability to handle fragments of FO that other FOMC algorithms cannot address [7]. Knowledge compilation algorithms build a *circuit* by iteratively applying *compilation rules* to formulas. This circuit can then be processed in polynomial time to compute the model count. Crane makes two fundamental contributions to the field. First, it enables *recursive* solutions by constructing *circuits with cycles*, where each cycle-inducing edge acts as a recursive function call. Second, we show how to interpret these circuits as collections of function definitions whose evaluation encodes the desired model count. This approach enables us to produce an algebraic expression for the model count as a function of arbitrary domain sizes.

Recently, I supervised several student projects that built upon this work. First, we presented a new version of Crane that compiles each circuit into a C++ program, demonstrating significant scalability improvements over other algorithms [2]. Another student project focused on employing FOMC-based techniques to tackle FO formulas beyond the capabilities of existing FOMC algorithms. This work introduces an approach that is up to an order of magnitude more efficient than the current state of the art for such formulas. Additionally, one of my ongoing projects aims to extend Crane's applicability to larger fragments of FO by introducing new compilation rules and generalising existing techniques.

Research Thrust 3: Synthetic Data Generation and Experimental Algorithmics

Quality algorithms generally emerge from a blend of theoretical reasoning and extensive experimental work. It is crucial to test algorithms across a broad range of problem instances to substantiate claims regarding one algorithm's superiority over another. However, existing benchmarks often fall short, failing to highlight key differences in algorithm performance. To gain a deeper understanding of the performance characteristics of WMC algorithms, I have undertaken several research projects aimed at developing novel random models and conducting comprehensive experimental analyses.

First, we have developed a *constraint model* for generating random probabilistic logic programs [4]. These programs are typically evaluated using WMC algorithms. We also introduce a new constraint to manage the independence structure of the underlying probability distribution. This model enables us to experimentally investigate inference algorithms on a significantly larger set of instances than before.

Second, we present a random model for WMC instances that includes a parameter that indirectly controls treewidth [3]. We show that all WMC algorithms scale exponentially with treewidth, albeit at different rates. Additionally, we demonstrate that the easy-hard-easy pattern concerning clause density differs for algorithms based on dynamic programming and algebraic decision diagrams compared to other solvers.

Future Directions

In the future, model counting-based algorithms and systems will become increasingly essential for two key reasons. First, these algorithms are experiencing a transformation similar to the one that allowed SAT—initially a prototypical NP-complete problem—to be solved efficiently on practical benchmarks. This transformation will enable a growing number of applications to move beyond simply determining whether an event is *possible* to *quantifying* its probability or frequency. Second, model counting algorithms will continue to undergird efficient inference in symbolic AI systems. These systems will serve as tools, collaborators, supervisors, or verifiers of subsymbolic AI methods, working together to provide trustworthy answers to complex queries. The remainder of this section outlines three primary directions for my future research, focused on improving the capabilities, efficiency, and reliability of model counting algorithms to support these widespread applications.

Knowledge Compilation for Counting and Sampling. I plan to continue my work on knowledge compilation-based algorithms. Two ongoing research projects align with this focus: the effort to enhance Crane's capabilities for larger fragments of FO by introducing new compilation rules and the development of a randomised algorithm for first-order model sampling. A long-term goal in this domain is to design a model counting algorithm for *first-order logic with counting* ($\mathsf{FO} + \mathsf{C}$). While the capabilities of such an algorithm remain largely uncharted, $\mathsf{FO} + \mathsf{C}$ would facilitate reasoning about the number of variable instantiations that satisfy a formula. For example, $\mathsf{FO} + \mathsf{C}$ could model the problem of counting the number of directed graphs with n vertices, where for each edge e, the source of e has a higher degree than the target.

Automata-Based Counting. Addressing counting problems by reducing them to counting the accepting paths of an automaton and deriving the corresponding generating functions is an under-explored area of research. In particular, *multi-track* automata hold the potential to become a widely used data structure in this setting akin to decision diagrams. My ongoing work on counting the combinations of bit-vector-valued variables that satisfy arbitrary arithmetic and number-theoretic constraints fits this theme. In the future, I intend to apply automata-based counting techniques to *monadic second-order logic*. Another long-term project involves enriching these automata with a structure allowing an automaton to encode counting problems independently of bit vector length or domain size.

Certificate Generation. A *certified* algorithm produces its final answer with an externally-verifiable proof of its correctness. Certification is a vibrant area of research in SAT and, more recently, propositional model counting. However, FOMC necessitates entirely different certification techniques, such as equipping an *automated theorem prover* with the ability to reason about FO formulas and their models. Implementing certification for FOMC will also deepen our understanding of the problem, which could inspire new advancements in counting algorithms.

References

1. Feldstein, J., **Dilkas, P.**, Belle, V., Tsamoura, E.: Mapping the neuro-symbolic AI landscape by architectures: A handbook on augmenting deep learning through symbolic reasoning. CoRR **abs/2410.22077** (2024)
2. Kidambi, A.K., Singh, G., **Dilkas, P.**, Meel, K.S.: Towards practical first-order model counting. In: SAT. LIPIcs, vol. 341, pp. 6:1–6:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2025)
3. **Dilkas, P.**: Generating random instances of weighted model counting - An empirical analysis with varying primal treewidth. In: CPAIOR. Lecture Notes in Computer Science, vol. 13884, pp. 395–416. Springer (2023)
4. **Dilkas, P.**, Belle, V.: Generating random logic programs using constraint programming. In: CP. Lecture Notes in Computer Science, vol. 12333, pp. 828–845. Springer (2020)
5. **Dilkas, P.**, Belle, V.: Weighted model counting with conditional weights for Bayesian networks. In: UAI. Proceedings of Machine Learning Research, vol. 161, pp. 386–396. AUAI Press (2021)
6. **Dilkas, P.**, Belle, V.: Weighted model counting without parameter variables. In: SAT. Lecture Notes in Computer Science, vol. 12831, pp. 134–151. Springer (2021)
7. **Dilkas, P.**, Belle, V.: Synthesising recursive functions for first-order model counting: Challenges, progress, and conjectures. In: KR. pp. 198–207 (2023)