Probabilistic reasoning in computation and type theory

Simona Kašterović

Faculty of Technical Sciences, University of Novi Sad Novi Sad, Serbia
simona.k@uns.ac.rs

Over the last decades reasoning about uncertain knowledge has played an important role in computer science and artificial intelligence and the interest in probabilistic programming has been rapidly growing. Formalizing probabilistic reasoning is one of the main topics of my research. Regarding the study of programming with probabilistic features, different approaches have been emerged.

The first approach we consider consists in taking a probabilistic choice as primitive and obtaining a probabilistic computation. The probabilistic computation is a new paradigm which has proved to be extremely applicable and useful in various areas, such as robotics, machine learning, and natural language processing. One of the most challenging problems is checking if two programs enjoy the same behavioural properties, that is proving program equivalence in a probabilistic setting ([3, 9]). Hence, the aim is to find an effective method for checking the behavioural equivalence, that is to find the characterization of the equivalence which enables us to check the equality of programs more easily. In [8], we have proved that in the call-by-name probabilistic lambda calculus extended with let-in operator there is an easier way to check if two programs are equivalent by proving that bisimilarity and behavioural equivalence coincide.

The second approach we are interested in concerns formalization in terms of probabilistic logics ([10]). We have developed formal models for probabilistic reasoning about typed lambda terms. The idea is presented in [5, 6]. We start by defining classical propositional logic over the typed statements, and obtain *a set of basic formulas*. Then, we apply probabilistic operator $P_{\geq s}$ to probabilistic formulas and obtain *a set of atomic probabilistic formulas*. The language of the logic is obtained by closing the set of atomic probabilistic formulas under the classical propositional connectives. The semantics of the logics introduced in [5, 6] have been based on the well-known semantics for the lambda calculus ([1]). In the proposed logic we are able to express statement " The probability that term (program) M has type σ is greater than or equal to s." with formula $P_{\geq s}$. In order to solve issues from [5] and to obtain system which is sound and complete with respect to the proposed semantics we have studied combinatory logic and developed a formal system for reasoning about typed combinatory terms ([11]). We have defined the classical propositional logic over simply typed combinatory terms ([2, 7]), introduced semantics based on applicative structure and proved that the obtained system is sound and complete with respect to the proposed semantics. We plan to further use these results to develop formal models for probabilistic reasoning about simply typed terms of combinatory logic.

Another line of my research is about resource control, that is about control of variable use in computation. In [4] we have achieved resource control by introducing a new notion of types, called L-types, in lambda calculus with implicit names. We worked simultaneously on the development of the L-type calculi and on their implementation in Haskell and Agda. Agda is a dependently typed programming language, which helps us to eliminate all possible errors and shows to be appropriate for the implementations of our calculi

Recently, I became interested in data privacy and blockchain technology. I participate in the project AI4TrustBC-Advanced Artificial Intelligence Techniques for Analysis and Design of System Components Based on Trustworthy BlockChain Technology supported by the Science

Fund Republic of Serbia. Within this project, we investigate privacy protection of blockchain and approaches for preserving privacy that are based on blockchain.

References

[1] Hendrik Pieter Barendregt. *The lambda calculus - its syntax and semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, 1985.

[2] Katalin Bimbó. *Combinatory Logic: Pure, Applied, and Typed.* CRC Press, Taylor & Francis Group, Boca Raton, Florida, 2012.

[3] Raphaëlle Crubillé and Ugo Dal Lago. On probabilistic applicative bisimulation and call-by-value λ-calculi. In Zhong Shao, editor, *Programming Languages and Systems - 23rd European Symposium on Programming, ESOP 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8410 of *Lecture Notes in Computer Science*, pages 209–228. Springer, 2014.

[4] Silvia Ghilezan, Jelena Ivetic, Simona Kasterovic, and Pierre Lescanne. L-types for resource awareness: an implicit name approach. *CoRR*, abs/2112.11062, 2021.

[5] Silvia Ghilezan, Jelena Ivetic, Simona Kasterovic, Zoran Ognjanovic, and Nenad Savic. Probabilistic reasoning about simply typed lambda terms. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2018, Deerfield Beach, FL, USA, January 8-11, 2018, Proceedings*, volume 10703 of *Lecture Notes in Computer Science*, pages 170–189. Springer, 2018.

[6] Silvia Ghilezan, Jelena Ivetic, Simona Kasterovic, Zoran Ognjanovic, and Nenad Savic. Towards probabilistic reasoning in type theory - the intersection type case. In Andreas Herzig and Juha Kontinen, editors, *Foundations of Information and Knowledge Systems - 11th International Symposium, FoIKS 2020, Dortmund, Germany, February 17-21, 2020, Proceedings*, volume 12012 of *Lecture Notes in Computer Science*, pages 122–139. Springer, 2020.

[7] J. Roger Hindley and Jonathan P. Seldin. *Introduction to Combinators and Lambda-Calculus.* Cambridge University Press, 1986.

[8] Simona Kasterovic and Michele Pagani. The discriminating power of the let-in operator in the lazy call-by-name probabilistic lambda-calculus. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 26:1–26:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[9] Ugo Dal Lago, Davide Sangiorgi, and Michele Alberti. On coinductive equivalences for higher-order probabilistic functional programs. In Suresh Jagannathan and Peter Sewell, editors, *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 297–308. ACM, 2014.

[10] Zoran Ognjanovic, Miodrag Raskovic, and Zoran Markovic. *Probability Logics - Probability-Based Formalization of Uncertain Reasoning.* Springer, 2016.

[11] Simona Kašterović and Silvia Ghilezan. Logic of combinatory logic. Submitted for publication, 2022.