Probabilistic Functional Programs: Termination and Verification

Charles Grellois
LIS, Aix-Marseille University

2020-2021

Title. Probabilistic Functional Programs: Termination and Verification

Topic. Theoretical Computer Science, Probabilistic Lambda-Calculus, Type Systems, Termination Analysis, Verification

City and country. Aix-Marseille, France.

Team or project in the lab. Team LIRICA: `https://www.lis-lab.fr/lirica/`

Name and mail of the advisor. Charles Grellois — `charles.grellois@lis-lab.fr`

Name and mail of the head of the department. Frédéric Béchet — `frederic.bechet@lis-lab.fr`

General presentation of the topic. Functional programming is a successful paradigm originating in λ-calculus. It has led to the development of programming languages such as OCaml and Haskell.

On the other hand, probabilistic computation is more and more used nowadays, in cryptography, robotics or for machine learning-related approaches.

Both approaches have been combined in languages such as Church [2] or Anglican [5]. We propose to study termination and verification problems for probabilistic λ-calculi, which mix functional programming with probabilistic programming.

Objective of the internship. The purpose of this internship is to work on the analysis of probabilistic λ-calculi. In recent work, Dal Lago and Grellois [4] defined a type system which ensures that a typable probabilistic λ-term terminates with probability 1: divergence is improbable. A first possible objective of the internship would be to work on type inference for this type system: how to

compute a typing derivation for a typable term? For the deterministic case, a type inference procedure already exists [1].

Another objective would be to devise a type system in the spirit of [4], but aiming to do probabilistic verification of the probabilistic λ-term of interest. This would combine intersection types, in the spirit of Kobayashi [3], with the distribution types introduced by Dal Lago and Grellois in [4]. A typable term would satisfy a property of interest with probability 1: failure would be improbable.

For more information and discussion on these matters, please contact me at `charles.grellois@univ-amu.fr`

Expected ability of the student. The student should have a strong interest in theoretical computer science. A prior knowledge of the basics of λ-calculus and of type systems would be very helpful.

References

[1] Gilles Barthe, Benjamin Grégoire, and Colin Riba. A tutorial on type-based termination. In Ana Bove, Luís Soares Barbosa, Alberto Pardo, and Jorge Sousa Pinto, editors, *Language Engineering and Rigorous Software Development, International LerNet ALFA Summer School 2008, Piriapolis, Uruguay, February 24 - March 1, 2008, Revised Tutorial Lectures*, volume 5520 of *Lecture Notes in Computer Science*, pages 100–152. Springer, 2008.

[2] Noah D. Goodman, Vikash K. Mansinghka, Daniel Roy, Keith Bonawitz, and Joshua B. Tenenbaum. Church: A language for generative models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, page 220229, Arlington, Virginia, USA, 2008. AUAI Press.

[3] Naoki Kobayashi. Types and higher-order recursion schemes for verification of higher-order programs. In Zhong Shao and Benjamin C. Pierce, editors, *Proceedings of the 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2009, Savannah, GA, USA, January 21-23, 2009*, pages 416–428. ACM, 2009.

[4] Ugo Dal Lago and Charles Grellois. Probabilistic termination by monadic affine sized typing. *ACM Trans. Program. Lang. Syst.*, 41(2):10:1–10:65, 2019.

[5] Frank Wood, Jan Willem van de Meent, and Vikash Mansinghka. A new approach to probabilistic programming inference. In *Proceedings of the 17th International conference on Artificial Intelligence and Statistics*, pages 1024–1032, 2014.