Software Modeling and Verification Chair

RWTH AACHEN UNIVERSITY

— Bachelor's / Master's Thesis —

Metrics and Equivalences on Formal Power Series for Probabilistic Program Analysis

What is it all about?

Machine Learning is *the* topic nowadays, but how do we teach machines besides deep learning mechanisms and neural networks? Mostly, Bayesian sometimes also called probabilistic machine learning is used. In case of neural networks, the teaching itself heavily relies on gradient-based methods for regression and classification. On the other hand, for probabilistic machine learning, inference is a crucial ingredient. Automated differentiation tools are to the former, as probabilistic programming is to the latter.
Probabilistic programming is fundamentally about developing languages and semantics that enables the denotation of inference programs and simple evaluation methods that "solve" those inference problems. More about probabilistic programming can be found here [vdMPYW18, Kam19].
In our approach, we use the probabilistic variant of Guarded Command Language (`pGCL`) developed by Dijkstra, McIver and Morgan et. al. to model probabilistic programs. The semantics, however, is defined on the concept of formal power series and generating functions [Wil06]. The main benefit of using these encodings is the usage of so called closed-forms. For example the infinite series $\sum_{i=0}^{\infty} X^i$ can be uniquely represented as $\frac{1}{1-X}$.
Measures and metrics between measures or distributions is already a well-known field in Stochastics. The question is, how these metrics can be brought into the probabilistic programming analysis using generating functions? For a first introduction, you can get more information here [ABH+19].

What is to be done?

1. Develop different metrics on generating functions wrt. probabilistic programs in `pGCL`.
2. Define a notion of equivalence of `pGCL` programs.
3. How do these metrics behave if there are parameters involved?

What you can expect:

- Get a chance to work on relevant open problems of theoretical nature.
- You can always work in our student's room.
- We have a very good coffee machine.

What we expect:

- Solid background in theoretical computer science. — ideally you have taken the lectures Semantics and Verification and Probabilistic Programming
- Solid mathematical skills in Algebra, Calculus and Stochastics — ideally maths as your minor / elective
- Passion and endurance for solving difficult (mathematical) problems.

Contact

- Lutz Klinkenberg, lutz.klinkenberg@cs.rwth-aachen.de, Tel. 0241/80-21221.

References

[ABH+19] Alejandro Aguirre, Gilles Barthe, Justin Hsu, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja, Kantorovich continuity of probabilistic programs, CoRR **abs/1901.06540** (2019).

[Kam19] Benjamin Lucien Kaminski, Advanced weakest precondition calculi for probabilistic programs, Ph.D. thesis, RWTH Aachen University, Germany, 2019.

[vdMPYW18] Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood, An introduction to probabilistic programming, 2018.

[Wil06] Herbert S. Wilf, Generatingfunctionology, A. K. Peters, Ltd., USA, 2006.