Software Modeling and Verification Chair | RWTH AACHEN UNIVERSITY

— Master's Thesis —

Approximating the Output of a Probabilistic Program

What is it all about?

Probabilistic programs extend deterministic programs by a random choice about which code branch is executed next. They can be defined by the following grammar:

$$c \coloneqq \text{skip} \mid x \coloneqq \text{a} \mid \{c\}[p]\{c\} \mid c; c \mid \text{if b then c else c end} \mid \text{while b do c end}.$$

Distributions over program states describe what values the program variables hold at a certain time, and with which probability. Our goal is to reason about the output distribution of a probabilistic program when given a certain initial distribution. Consider the following two examples:

$$// \; 1 : x \mapsto 1$$
$$x := x - 1; [\tfrac{1}{2}] x := x + 1;$$
$$// \; \tfrac{1}{2} : x \mapsto 0 \text{ and } \tfrac{1}{2} : x \mapsto 1$$

$$// \; 1 : x \geq 42$$
$$x := 0;$$
$$// \; 1 : x \mapsto 0$$

The output distribution can be computed by the denotational semantics (as defined e.g. in [dH99]). For loops, a least fixed point (LFP) is therefore employed. In general, the LFP of a function can be approximated by Knaster-Tarski's theorem: (taken from [CQS+25]).

Theorem 1 (Knaster-Tarski)**.** *Let $(X, \preceq)$ be a complete lattice and $\mathcal{F} \colon X \to X$ be monotone. Then, the set of fixed points $(\{a \in X \mid \mathcal{F}(a) = a\}, \preceq)$ is also a complete lattice. In particular, $\mathcal{F}$ has a least and a greatest fixed point given by $\operatorname{lfp} \mathcal{F} = \inf \{a \in X \mid \mathcal{F}(a) \preceq a\}$ and, dually, $\operatorname{gfp} \mathcal{F} = \sup \{a \in X \mid a \preceq \mathcal{F}(a)\}$. As a consequence, the following fixed point induction rules are sound: $\forall a \in X$:*

- $\mathcal{F}(a) \preceq a \implies \operatorname{lfp} \mathcal{F} \preceq a$ (fixed point induction)
- $a \preceq \mathcal{F}(a) \implies a \preceq \operatorname{gfp} \mathcal{F}$ (fixed point co-induction)

What is to be done?

The goals of this project are:

1. Apply Knaster-Tarski's theorem to multiple examples in order to verify upper bounds on the post distribution of a probabilistic loop
2. Examine what information an upper bound actually provides for the verification of probabilistic programs
3. Optional: Develop a greatest fixed-point (GFP) characterization for the post distribution. This would open the possibility to verify lower bounds on the post distribution using Knaster-Tarski as well.

This list is of course non-exhaustive! The above suggestions may be changed, shortened and/or extended while we work on our project and gain more insights on how difficult the topic is.

What we expect:

- Solid background in theoretical computer science and maths – ideally you have already taken theoretical CS electives
- Passion and endurance for solving theoretical problems

What you can expect:

- Get a chance to work on relevant problems of both theoretical and practical nature
- You can work in the student room at our chair – we have a coffee machine, lots of tea and sometimes cookies :)

Apply

- Daniel Zilken (daniel.zilken@cs.rwth-aachen.de)
  Please introduce yourself briefly and say why you're interested in this topic!

References

[CQS+25] Krishnendu Chatterjee, Tim Quatmann, Maximilian Schäffeler, Maximilian Weininger, Tobias Winkler, and Daniel Zilken. Fixed point certificates for reachability and expected rewards in mdps, 2025.

[dH99] J. I. den Hartog. Verifying probabilistic programs using a hoare like logic. In P. S. Thiagarajan and Roland Yap, editors, *Advances in Computing Science — ASIAN'99*, pages 113–125, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.