Software Modeling
and Verification Chair

RWTH AACHEN
UNIVERSITY

— Bachelor's Thesis —

Semantic Equivalences for MDPs and (Nondeterministic) Probabilistic Programs

What is it all about?

Probabilistic programs [BKWZ25] extend deterministic programs by a random choice about which code branch is executed next. They can be defined by the following grammar:

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

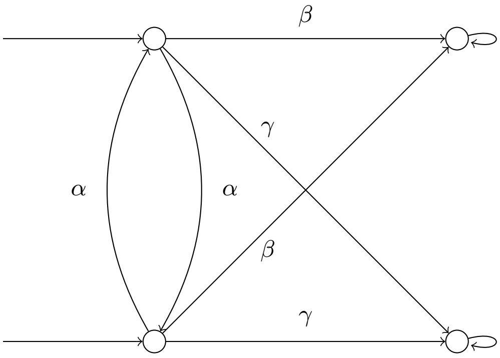
Distributions over program states describe what values the program variables hold at a certain time, and with which probability. Our goal is to reason about distributions as pre- and postconditions of probabilistic programs. Consider the following two examples:

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

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

A function mapping an initial distribution to the post distribution that the probabilistic program outputs is called the denotational semantics of probabilistic programs (see e.g. [dH99]). Programs can be extended by nondeterministic choices (we would do so in a second step of the project).

Markov Decision Processes (MDPs) [BBKW24] are models of nondeterministic and probabilistic systems. An example of an MDP is:



The nondeterminism is modeled by different actions α, β, γ. Uncertainty is modeled by probabilisties on transtitions. A MDP without nondeterminism is simply called a Markov chain (MC) (in this project, we will start with MCs). For Markov chains, we can define a semantic (equivalent to the denotational semantic for programs) by considering the limiting distribution of applying the Markov chains' transitions to some initial distribution.

What is to be done?

The goals of this project are:

1. Understand the relationship between probabilistic programs and Markov chains
2. Prove equivalence of the two semantics: the denotational semantics for programs and the limiting distribution semantics for MCs
3. Prove further semantic equivalences (e.g. when extending both, programs and MCs, by nondeterminism)

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

[BBKW24] Kevin Batz, Tom Jannik Biskup, Joost-Pieter Katoen, and Tobias Winkler. Programmatic strategy synthesis: Resolving nondeterminism in probabilistic programs. *Proc. ACM Program. Lang.*, 8(POPL), January 2024.

[BKWZ25] Kevin Batz, Joost-Pieter Katoen, Tobias Winkler, and Daniel Zilken. Verifying sampling algorithms via distributional invariants, 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.