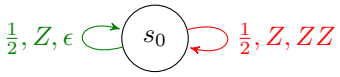
Software Modeling and Verification Chair | RWTH AACHEN UNIVERSITY

— Bachelor's or Master's Thesis —

Proving Termination of Probabilistic Recursive Programs via SMT-Solving

What is it all about?

Probabilistic Push-Down Automata (pPDA) [EKM04] are a universal model for discrete recursive stochastic processes. Unlike classic Push-Down Automata, their behaviour is purely probabilistic, i.e., they do not act as word-acceptors. Their main purpose is to provide a model that enables reasoning about properties such (almost-sure) termination, expected runtimes, failure probabilities and many more of the underlying system.
Recently, termination analyses of probabilistic programs have received lots of attention in the literature[1]. Finite-state pPDA can faithfully model a sub-class of probabilistic programs with recursive procedure calls. As usual, the basic idea is to encode the programs's call or procedure stack into the automaton's stack. Since termination of pPDA is decidable, it is clear that we can also decide termination of the encoded programs.
The standard technique for deciding pPDA termination relies on encoding the automaton as a polynomial system of constraints. This, in turn, can be seen as a first-order logical formula over the real numbers with addition, multiplication and comparison. It is well-known that the validity of such formulae is decidable, and decision procedures are nowadays implemented in SMT-solvers.



$$t = \frac{1}{2} + \frac{1}{2}t^2$$

Figure 1: Left: Example pPDA that either pops symbol Z from the stack (green) or pops Z and then pushes ZZ (red). This is repeated until the stack is empty in which case we say that the automaton has terminated. Right: The polynomial equation characterizing the termination probability. More precisely, the termination probability is exactly the least non-negative solution. Can you figure out the termination probability if the automaton takes the red transition with probability $\frac{2}{3}$ instead?

What is to be done?

The goal of this project is to automatically prove pPDA termination via SMT solving:

1. Implement a translation (aka encoding) of pPDA to non-linear SMT
2. Propose optimizations for the encoding
3. Compare the approach to other techniques for proving termination of probabilistic programs

This list is of course non-exhaustive! All of the above suggestions require both theoretical and practical (implementation) effort.

What we expect:

- Solid background in theoretical computer science and maths – ideally you have already taken theoretical CS electives
- Passion and endurance for solving difficult theoretical problems
- Basic Java skills are helpful but not required

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 decent coffee machine

Apply

- Tobias Winkler (tobias.winkler@cs.rwth-aachen.de)
  Please introduce yourself briefly and say why you're interested in this topic!

[1] https://dblp.org/search?q=probabilistic+program+termination

References

[EKM04] Javier Esparza, Antonín Kucera, and Richard Mayr. Model checking probabilistic pushdown automata. In *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*, pages 12–21. IEEE Computer Society, 2004.