Safe Probabilistic Programming by Inductive Synthesis

2023-08-02 19:43:09

Authors

Joost-Pieter Katoen

Abstract

Probabilistic programs steer AI robots, encode randomised algorithms and are used to improve the training of neural networks. They are even used to investigate seismologic effects on our globe. Probabilistic programs are however very hard to understand, in particular it is hard to prove that they are in some way "safe". Model checking can be used for finite state programs, but what about infinite, loopy programs? Deductive verification of probabilistic programs gets quite some attention and has recently led to powerful proof rules enabling proving loops to be "safe". These rules are all of the shape: if your program satisfies certain -- hopefully easy-to-check -- conditions, then it is "safe". We present a learning technique synthesising a class of loop invariants on probabilistic programs enabling proving that a program is "safe" in a fully automated manner. We will explain the underlying technology and presents some promising results using a prototypical implementation. In particular, it outperforms state-of-the-art probabilistic model checkers such as Storm on some examples, is competitive to the expected runtime analyser AbSynth, and outperforms data-driven ML techniques for obtaining lower bounds with several orders of magnitude.

Keywords

--