Latticed Craig Interpolation with an Application to Probabilistic Verification

Mingqi Yang[1], Kevin Batz[2], Mingshuai Chen[1], Joost-Pieter Katoen[2], Zhiang Wu[3], and Jianwei Yin[1]

[1] Zhejiang University, Hangzhou, China
{mingqiyang,m.chen,zjuyjw}@zju.edu.cn
[2] RWTH Aachen University, Aachen, Germany
{kevin.batz,katoen}@cs.rwth-aachen.de
[3] University of Waterloo, Waterloo, Canada

Abstract

Due to its inherent capability of local and modular reasoning, *Craig interpolation* serves as a pivotal tool in multiple formal verification techniques, e.g., model checking, theorem proving, and abstract interpretation. In model checking, for instance, the study of interpolation was pioneered by McMillan who proposed to augment SAT-based model checking with the ability to overapproximate the reachable state space via Craig interpolants, thus facilitating *unbounded* symbolic model checking [2]. In particular, Craig interpolation has proved useful in discovering *loop invariants* that are essential to program verification.

As an emerging paradigm, *probabilistic programming* – which describes stochastic models as executable and inferrable computer programs – has undergone a recent surge of interest due to prominent applications in, e.g., cryptography, approximate computing, machine learning, statistical data analysis, and beyond. Unlike verification of deterministic programs against *qualitative* properties, reasoning about probabilistic programs requires addressing various *quantities* such as expected values, assertion-violation probabilities, high-order moments, and expected runtimes. Computing these quantities amounts to inferring the quantitative (least) fixed point $\mathsf{lfp}\ \Phi$ of some monotonic operator Φ capturing the semantics of a possibly unbounded loopy program, which is in general highly intractable.

In this work, we are primarily concerned with the question: *Is Craig interpolation applicable to the automatic, quantitative verification of (infinite-state) probabilistic programs with potentially unbounded loops?* Our preliminary results indicate an affirmative answer:

- *Quantitative Craig interpolants.* We propose a quantitative version of Craig interpolants by extending predicates to expectations (expected values), which can be used to discover quantitative loop invariants that suffice to establish upper bounds on $\mathsf{lfp}\ \Phi$;
- *Latticed Craig interpolation.* We present latticed Craig interpolation by exploiting quantitative interpolants over complete lattices, which conservatively extends both McMillan's interpolation-based SAT model checking [2] (to the *quantitative* setting) and Batz et al.'s latticed bounded model checking [1] (to the *unbounded* case);
- *Soundness and Completeness.* We show that our latticed interpolation procedure is *sound* and establish sufficient conditions under which it is further *complete*.
- *Synthesizing quantitative interpolants.* We (semi-)automated our verification procedure by employing a counterexample-guided inductive synthesis framework to automatically generate quantitative interpolants. Our implementation shows promise: It finds invariants for non-trivial infinite-state programs with unbounded loops.

References

[1] K. Batz, M. Chen, B. L. Kaminski, J.-P. Katoen, C. Matheja, and P. Schröer. Latticed k-induction with an application to probabilistic programs. In *CAV (I)*, LNCS 12760, pages 524–549, 2021.

[2] K. L. McMillan. Interpolation and SAT-based model checking. In *CAV*, LNCS 2725, pages 1–13, 2003.