Towards Flexible Interpolation

Motivation

- Rich use of interpolants in model checking – abstractions, inductive invariants, …
- Problem
  - Traditionally, model checkers and interpolating SMT solvers developed independently
  - Usage of the interpolator: black box, one-size-fits-all approach
- Our approach
  - Tight cooperation between model checker and interpolator
  - Flexible interpolation framework
  - Smart model checker

Towards Flexible Interpolation

- Decomposition of Farkas Interpolants
- Interpolation procedure for LRA conflicts
- Generalization of interpolation procedure based on Farkas coefficients – flexibility in logical strength

Experiments

- Sally\(^1\) with OpenSMT\(^2\) as interpolator
- Sally benchmark set – 1107 transition systems

Comparison of the performance of a portfolio against only Farkas interpolation. Runtime in seconds.

Portfolio consisting of four approaches: Farkas interpolants, decomposed interpolants, and their duals.

1. \(\text{http://sri-csl.github.io/sally}\)
2. \(\text{https://github.com/usi-verification-and-security/opensmt/}\)

Craig Interpolation

Definition [Craig'57]: For an unsatisfiable \(A \land B\), interpolant is a formula \(I\) such that:
- \(A \Rightarrow I\)
- \(B \Rightarrow \neg I\)
- \(I\) contains only common symbols of \(A\) and \(B\)

Craig Interpolation

Towards Smart Model Checker

- Utilizing flexible interpolation
- Parallelization – different interpolation strategies
- Multiple interpolants from a single query
  - Choose one with appropriate strength
  - Delay choice until more information available

Towards Smart Model Checker

Envisioned Approach – Tight Cooperation

Smart Model Checker

SMT query

query-specific information

interpolants of different strength

SMT solver

Flexible Interpolator

proof