Progess Report 4

Xie Li

September 22, 2020

Overview of the Progress

- Sample-based Algorithm.
- Reading a chapter of BDD of handbook.
- A brief summary of algorithms and techniques.
- Build the gap between LLVM IR and our own data structure.
- Configured the use of CMake and LLVM API for developing.
- Investigate SPOT.

Sample based Algorithm

- LLVM IR $\Rightarrow$ CFG $\Rightarrow$ Automaton $\mathcal{A}$
- LTL/LTLf $\Rightarrow$ Automaton(BA/DFA) $\mathcal{P}$
- Sample paths of $\mathcal{A} \times \mathcal{P}$ on the fly

  $\Rightarrow$

  check feasibility of the path

Properties

- Reachability problem: avoidance of errornous states. Here the "error" may be given by statement `error()` in the source. LTL: **G**!`error()`
- Overflow problem: can be encoded as

$$-\texttt{maxInt} \leq x \leq \texttt{maxInt}$$

- Free and Deref of pointers: can be represented as LTL formula or reachability problem.
- Devided by zero.
- Use after free, use without definition.
- Termination.
- No Deadlocks.

Algorithms

- Sampled-based MC: SMT solver for feasibility checking, sampling strategy, on-the-fly product.
- Symbolic MC: BDD library for the encoding of the formula and conditions, SMT solver for checking.
- Bounded MC:

Current Progress of Coding

Source code available at

`https://github.com/SpencerL-Y/LLVMADT`