An Incremental Approach to Model Checking Progress Properties

Paper authors: Aaron R. Bradley, Fabio Somenzi, Zyad Hassan, Yan Zhang
Slides by Enrico Magnago

April 10, 2022

Context, problem and generic solution

Context

- Finite state discrete transition system:
  $S := (\bar{i}, \bar{x}, I(\bar{x}), T(\bar{i}, \bar{x}, \bar{x}'))$.
- Set of fairness conditions $\mathcal{B} := \{B_1(\bar{x}), \ldots, B_l(\bar{x})\}$

Problem

Is the language of S empty? Is there a fair path?

Solution

Look for a "lasso" shaped path.

Fair: overview

Iterate the following steps:

1. Obtain a set of states that together satisfy the fairness conditions $\mathcal{B}$, *skeleton*.
2. Check if from the initial states it is possible to reach one of the states of the *skeleton*, (stem).
3. Each state of the *skeleton* can reach the following, for some order, (loop).

If we find such *skeleton* states, the stem and the loop, their concatenation is a fair path of S.

Learning

1: find skeleton states

SAT obtain set of states,

UNSAT no fair states then no fair path in S;

2: find stem

SAT trace leading to one of the *skeleton* states,

UNSAT learn reachability predicate P that separates initial states from current *skeleton* states; collect them into a set $\mathcal{P}$.

3: find loop

SAT obtain loop between the *skeleton* states,

UNSAT learn predicate W (wall): all states in the loop must agree on the truth assignment of W; collect them into a set $\mathcal{W}$.

1. Find skeleton states

1. Basic formulation

- For each fairness condition $B \in \mathcal{B}$ create a copy $\bar{x}_B$ of $\bar{x}$;
- For each wall $W \in \mathcal{W}$ add a boolean variable c_W: choose the side of the wall.

Every model of the following formulae provides a set of *skeleton* states:

$$\bigwedge_{B \in \mathcal{B}} \Big[B(\bar{x}_B) \wedge \bigwedge_{R \in \mathcal{R}} R(\bar{x}_B) \wedge \bigwedge_{W \in \mathcal{W}} (c_W \rightarrow W(\bar{x}_B)) \wedge (\neg c_W \rightarrow \neg W(\bar{x}_B)) \Big]$$

1. Optimisation a: reduce number of skeleton states

Let $j : \mathcal{B} \to \{1 \dots |\mathcal{B}|\}$ map from fairness condition to index.

$$\bigwedge_{B \in \mathcal{B}} \Bigg[B(\bar{x}_{j(B)}) \wedge \bigwedge_{R \in \mathcal{R}} R(\bar{x}_{j(B)}) \wedge \bigwedge_{W \in \mathcal{W}} (c_W \to W(\bar{x}_{j(B)})) \wedge (\neg c_W \to \neg W(\bar{x}_{j(B)})) \Bigg]$$

- The map decides which fairness conditions should be satisfied by the same skeleton state.
- If the map is bijective this formula is equivalent to the previous one.

1. Optimisation b: skeleton states with at least K successors and predecessors

- Reduce number of potential skeletons by considering only those with at least K predecessors and successors;
- all states should remain within the same walls;
- the $2K$ long unrolling should be either loop-free or loops on the skeleton-state.

2. Find stem

2. Possible formulations

1. Let s_0 be a state of the *skeleton*:

$$reach(S, \bigwedge_{R \in \mathcal{R}} R(\bar{x}), I, s_0)$$

2. Repeat 1 for all states in the *skeleton*;
3. Check if the disjunction of all states in the *skeleton* is reachable.

3. Find loop

3. Naive formulation

Let $\oplus_n$ be the sum modulo n.

Simple reachability

$$reach(S, \top, s_i, s_{i \oplus_n 1})$$

if UNSAT, we learn an inductive proof P:

- $s_i \rightarrow P$,
- $P \wedge T \rightarrow P'$,
- $P \rightarrow \neg s_{i \oplus_n 1}$

P is a wall: can cross only once from $\neg P$ to P.

3. Consider walls

Create $\mathcal{C}$ as follows

for every $W \in \mathcal{W}$ (W is inductive)

- if no W-skeleton exists, add $\neg W'$ to $\mathcal{C}$;
- if no $\neg W$-skeleton exists, add W to $\mathcal{C}$;
- otherwise add $W' \to W$ to $\mathcal{C}$.

Considers only one wall at a time, doesn't try to exclude regions defined by multiple walls.

Cycle query becomes

$$reach(S, \bigwedge_{R \in \mathcal{R}} R \wedge \bigwedge_{C \in \mathcal{C}} C, s_i, s_{i \oplus_n 1})$$

3. Special case: single state skeleton

- Need to find a non-trivial loop: successor of s_0 can reach s_0.

$$reach(S, \bigwedge_{R \in \mathcal{R}} R \wedge \bigwedge_{C \in \mathcal{C}} C, post(S, s_0), s_0)$$

- If we get a proof P, we also need to remove s_0 from the possible skeletons, get unsat core $d \subseteq s_0$ of:

$$s_0 \wedge \neg P \wedge T \wedge \bigwedge_{R \in \mathcal{R}} R \wedge \bigwedge_{C \in \mathcal{C}} C \wedge \neg P'$$

- The wall associated with P now has 2 different sides: P, $\neg P \wedge \neg d$.
- Otherwise use "optimisation" 1.b: skeleton states in $2K$ long path.

Extra optimisations

Extra: minimise unreachability proofs

- IC3 proofs are in CNF: large;
- Strengthening - weakening loop: reduce number of literals and clauses (MIC algorithm);
- Obtain shorter, more informative proof.

Extra: skeleton-independent proofs

Pick a predicate l (from S), l is a wall if the following formula is UNSAT:

$$l \wedge \bigwedge_{R \in \mathcal{R}} R \wedge \bigwedge_{C \in \mathcal{C}} C \wedge T \wedge \neg l'$$

Motivating example

- Bit-counter with output bit o,
- o becomes $\top$ when all bits are 1 and then remains $\top$,
- fairness: $GF\neg o$.

Further generalisation of Infinite Traces

Reachability between fairness conditions

Re-formulation using reachability

For all $0 \leq i < |\mathcal{B}|$ there is no state s in the set of fair states B_i such that s can not reach any of the fair states in the next fair condition $B_{i \oplus_{|\mathcal{B}|} 1}$:

$$\forall i.0 \leq i < |\mathcal{B}| \; \forall s_0 \; \exists s_1 : B_i(s_0) \rightarrow reach(S, P, s_0, s_1) \wedge B_{i \oplus_{|\mathcal{B}|} 1}(s_1)$$

for some predicates P.
P can include $\mathcal{R}$ and $\mathcal{W}$.