COL750: Foundations of Automatic Verification (Jan-May 2023)

Lectures 19 & 20 (Interpolation and SAT-Based Model Checking)

Kumar Madhukar

madhukar@cse.iitd.ac.in

Mar 27th and 29th

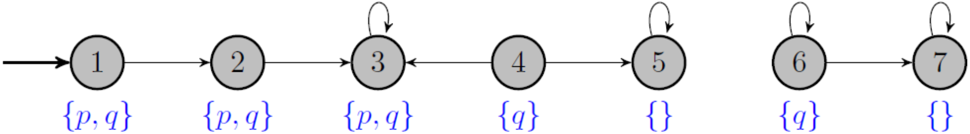
Bounded Model Checking: Recap

- primarily a bug finding technique
- but what to do when no bugs are being found
- use k-induction to obtain proofs
  - *strengthen* the criteria for the base case [i.e., p holds in the first k states starting from the initial state]
  - *weaken* the criteria for the step case [i.e., if p holds in all states in any sequence of k states on any path, then it also holds in the $(k+1)^{th}$ state]

k-induction

- base case
  $I(s_0) \wedge T(s_0, s_1) \wedge T(s_1, s_2) \wedge \ldots \wedge T(s_{k-1}, s_k) \wedge \neg p(s_k)$
- step case
  $p_j \wedge T(s_j, s_{j+1}) \wedge p_{j+1} \wedge T(s_{j+1}, s_{j+2}) \wedge \ldots \wedge p_{j+k-1} \wedge T(s_{j+k-1}, s_{j+k}) \wedge \neg p(s_{j+k})$
- if base and step cases both are unsat for any k, then p holds globally along all paths
- if base case is sat (for some k), we get a counterexample (of length k)
- if step case is sat (then no conclusion can be made about the property, because s_j was arbitrary and therefore may not have been reachable), increase k
- case for incremental sat solving (base and step case formulas have a lot of overlap)

Recall example



- 6 and 7 are neither initial states nor reachable; so `AG q` holds
- but the step case of *k* induction is bound to fail for any *k*
- to obtain a complete variant of *k*-induction for `AG p` properties, we add a conjunct that all states on any counterexample to the step-case are pairwise different (why? – *exercise*)

Model Checking with Inductive Invariants

- inductive reasoning can be applied to prove properties of the form `AG p`

- given a model M, the post-image of a set of states Q is the set of states that are reachable from Q in one transition (in M)

  $$\mathit{post\text{-}image}(Q) = \{s' \mid \exists s \in Q.(s, s') \in T_M\}$$

- we say I to be an inductive invariant for the property `AG p` if the following conditions hold:

  1. I includes all initial states [*initiation*]
  2. I must be closed under the transition relation (i.e., $\mathit{post\text{-}image}(I) \subseteq I$ holds) [*consecution*]
  3. I must not include a $\neg p$ state [*safety*]

Algorithmically computing an inductive invariant

- recall the BMC for `AG p`, for bound k

  $I(s_0) \wedge \bigwedge_{i=0}^{k-1} T(s_i, s_{i+1}) \wedge \bigvee_{i=0}^{k} \neg p(s_i)$

- let us omit the check for $p(s_0)$ from here and do this separately, and also replace the set of initial states I with an arbitrary set of states Q

  $Q(s_0) \wedge \bigwedge_{i=0}^{k-1} T(s_i, s_{i+1}) \wedge \bigvee_{i=1}^{k} \neg p(s_i)$

- and now let us rewrite this by splitting the formula into two parts

  $Q(s_0) \wedge T(s_0, s_1) \wedge \bigwedge_{i=1}^{k-1} T(s_i, s_{i+1}) \wedge \bigvee_{i=1}^{k} \neg p(s_i)$

Algorithm

if $S_0 \wedge \neg p$ is SAT **return** $M \not\models AG\ p$ — `S₀ is the initial set of states`

k := 1; Q := S_0;

while *true* **do**

  A := $Q(s_0) \wedge T(s_0, s_1)$;   B := $\bigwedge_{i=1}^{k-1} T(s_i, s_{i+1}) \wedge \bigvee_{i=1}^{k} \neg p(s_i)$

  if (A $\wedge$ B) is SAT then

    if $Q = S_0$ **return** $M \not\models AG\ p$

    increase k; $Q := S_0$ — the over-approximate Q is not corrected, but reset

  else

    I := *compute-interpolant*(A, B)

    if $I \subseteq Q$ **return** $M \models AG\ p$ — Q is a safe inductive invariant

    $Q := Q \cup I$

  end if

end while

Why is this correct?

- whenever it returns $M \not\models AG\ p$, it is because of a counterexample returned from a concrete BMC instance
- Assumption 1: I does not contain any state labelled with $\neg p$
- Assumption 2: I over-approximates the post-image of Q
- these two assumptions imply that Q is indeed a *safe inductive invariant*
- but what about the assumptions? (they will be guaranteed by the way we generate I)

Why is it complete (for finite-state systems)?

- if Q stops increasing (with the augmentation of I) then the algorithm stops
- otherwise Q strictly increases each time in the else branch
- this cannot go on; so, k must increase eventually
- if the property does not hold, k must eventually be increased to the length of the shortest counterexample (and in that case, the immediate next SAT query will give us that counterexample)
- if the property holds, k will eventually reach the diameter of the model M and then post-image will not be able to able any new state (Q will stop increasing)

Interpolation

- A and B first-order formulas, such that $A \wedge B$ is unsat
- an interpolant I for A and B is a first-order formula such that

  $$A \Rightarrow I \qquad \text{and} \qquad I \Rightarrow \neg B$$

- Craig showed that interpolants exist for any two inconsistent first-order formulas A and B

Craig's Interpolation Theorem

Given an inconsistent pair of first-order formulas A and B, there exists an interpolant I such that

1. A implies I,
2. I is inconsistent with B, and
3. I uses only symbols that are both in A and B.

Algorithmic techniques for computing interpolants from unsat proofs (of $A \wedge B$) exist for many fragments of first-order logic.

We will restrict ourselves, here, to resolution proofs and propositional logic formulas.

Proving the two assumptions

given that I is an interpolant

- Assumption 1: I does not contain any state labelled with $\neg p$

  Note that I must be inconsistent with B. Now, assume that there is a $s \in I$ such that $\neg p(s)$. But then B will be satisfied. (Why? Because the right conjunct of B gets satisfied because of s, and the left conjunct is satisfied because we work under the assumption that there is an outgoing transition from every state in the model.)

- Assumption 2: I over-approximates the post-image of Q

  Suppose not. Let there be a state $s \in \textit{post-image}(Q)$ such that $s \notin I$. But this also means that s cannot be in A (because $A \Rightarrow I$). But look at the structure of A – it has all the states that are reachable from Q in one step (i.e., $\textit{post-image}(Q)$). So, s cannot be in $\textit{post-image}(Q)$.

Computing interpolant

The notes on interpolation (uploaded on Teams as `itp-notes.pdf`), which is nothing but Section 18.6 from the Handbook of Satisfiability, summarizes what was covered in the class.

Thank you!