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

Lectures 21 & 22 (IC3 – SAT-Based Model Checking without Unrolling)

Kumar Madhukar

madhukar@cse.iitd.ac.in

Apr 3rd and 6th

Quick remarks about Interpolation

- recall the example from the last class (please see `itp-notes.pdf`, uploaded on Teams)
- the strongest interpolant of A is obtained from A by existentially quantifying over all local variables in A
- thus, interpolation can be seen as an over-approximation of quantifier elimination
- in our example, we had obtained the interpolant $c \vee d$, where the strongest interpolation would have been $c \oplus d$

Quick remarks about Interpolation

Here is a somewhat easier-to-remember method for annotating the resolution proof to obtain an interpolant:

1. for an initial node corresponding to a clause $c \in A$, annotate with c' where c' is obtained from c by keeping only those literals whose variables occur in B
2. for an initial node corresponding to a clause $c \in B$, annotate with *true*
3. for a derived node (with the pivot variable x occurring in B), annotate with the conjunction of its parents' annotations
4. for a derived node (with the pivot variable x not occurring in B), annotate with the disjunction of its parents' annotations

Interpolation and SAT-Based MC

- keeps only one candidate invariant (Q)
- when a bad state is reachable from the over-approximation, the over-approximation is not refined
- instead, the over-approximation is discarded completely and the transition system is unrolled further

SAT-Based Model Checking without Unrolling

- without making copies of the transition relation
- computes over-approximation of the post-image of the set of reachable states
- maintains multiple candidate invariants

Frames and Invariants

- done by maintaining frames – $F_0, F_1, \ldots, F_k$ – which are step-wise assumptions (or over-approximations)

- the frames maintain the following invariants

1. $I_0 \rightarrow F_0$ (F_0 contains the initial set of states)
2. $F_i \rightarrow F_{i+1} \quad (0 \leq i < k)$ (frames are monotonic)
3. $F_i \rightarrow P \quad (0 \leq i \leq k)$ (none of the frames contain a bad, i.e. $\neg P$, state)
4. $F_i \wedge T \rightarrow F'_{i+1} \quad (0 \leq i < k)$ (F_i over-approximates i-step reachability)

Inductive Reasoning

to prove that P is an invariant (that every reachable state satisfies P), it suffices to prove that

1. all initial states satisfy P
   $init(x) \rightarrow P(x)$ (*initiation*)
2. a P-state can only be followed by a P-state
   $P(x) \wedge trans(x, x') \rightarrow P(x')$ (*consecution*)

however, P itself may not be inductive; it may help to have a stronger assertion in that case

1. $init(x) \rightarrow f(x)$ (*initiation*)
2. $f(x) \wedge trans(x, x') \rightarrow f(x')$ (*consecution*)
3. $f(x) \rightarrow P(x)$ (*safety*)

Example

```
x = 1;
y = 1;

while(*)
  x, y = x + 1, y + x
```

suppose we want to prove the property, P, that $y \geq 1$ is an invariant

Example

- $(y \geq 1)$ is not an inductive invariant (why? the consecution check fails)
- so, we must look for a strengthening of $(y \geq 1)$
- $(x \geq 0 \wedge y \geq 1)$ is an inductive invariant; but how do we obtain this?
- counterexample to induction (CTI) from the failed consecution check: $[x = -1, y = 1]$
- the strengthening $(x \geq 0)$ must *eliminate* the CTI
- $(x \geq 0)$ is an inductive invariant
- $(y \geq 1)$ is inductive *relative to* $(x \geq 0)$

$$(x \geq 0) \quad \wedge \quad (y \geq 1) \quad \wedge \quad y' = y + x \quad \wedge \quad x' = x + 1 \quad \rightarrow \quad (y' \geq 1)$$

- thus, an incremental proof is possible

Another example

```
x = 1;
y = 1;

while(*)
  x, y = x + y, y + x
```

suppose we want to prove the property, P, that $y \geq 1$ is an invariant

Another example

- as in case of previous example, $y \geq 1$ is an invariant but not inductive
- we get a CTI last like time: $[x = -1, y = 1]$
- $(x \geq 0)$ eliminates the CTI but isn't inductive (unlike the last time)
- but it is inductive relative to the property

$$(y \geq 1) \quad \wedge \quad (x \geq 0) \quad \wedge \quad y' = y + x \quad \wedge \quad x' = x + y \quad \rightarrow \quad (x' \geq 0)$$

- seemingly circular reasoning, but not actually so

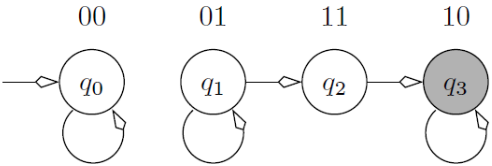
  $P \wedge \psi \wedge T \rightarrow \psi'$ and $\psi \wedge P \wedge T \rightarrow P'$
  together imply that $\psi \wedge P$ is an inductive invariant

- thus, an incremental proof is still possible (though it may not be possible in every case; exercise – construct an example where the entire inductive strengthening must be obtained at once)

Back to frames and invariants

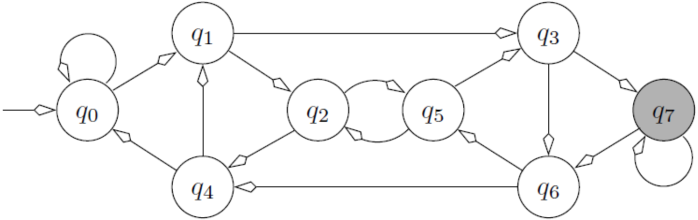
- check that $I \to P$ (that none of the initial states are bad), and set F_0 to I
- check $(I =)\ F_0 \wedge T \to P'$ (that bad is not 1-step reachable), and set F_1 to P
- now, we check $F_1 \wedge T \to P'$
- if not, there must be a CTI $s \in F_1$ that can reach $\neg P$ in one step
- but $s \notin F_0$, else it would have been discovered earlier (while checking $F_0 \wedge T \to P'$)
- so, we check if s is reachable from F_0 in one step $(F_0 \wedge \neg s \wedge T \to \neg s')$
- if yes, then s has a predecessor s_{pre} in F_0 (we need to check if s_{pre} is an initial state, or if it has a predecessor, and so on..)
- if not, then $F_1 := (F_1 \wedge \neg s)$ [it may be better to generalize the CTI instead of just eliminating one state at a time]

IC3 on a safe example[1]



[1] Reference: https://theory.stanford.edu/~arbrad/papers/ic3_tut.pdf

IC3 on an unsafe example[2]



[2]Reference: https://theory.stanford.edu/~arbrad/papers/ic3_tut.pdf

Algorithm

procedure PDR (model M, property P)

if $(I_0 \wedge \neg P)$ is SAT, **return** "P does not hold"
$F_0 \leftarrow I_0$; $k \leftarrow 0$;

while true **do**

$extendFrontier(M, k)$
$propagateClauses(M, k)$
if $F_i = F_{i+1}$ for some i, **return** "P holds"
$k \leftarrow k + 1$

end while

end procedure

Algorithm

procedure extendFrontier (M, k)

$F_{k+1} \leftarrow P$

while $F_k \wedge T \wedge \neg P'$ is SAT **do**

$s' \leftarrow$ state labelled with $\neg P$ extracted from the satisfying assignment
$s \leftarrow$ predecessor of s' extracted from the satisfying assignment
$removeCTI(M, s, k)$

end while

end procedure

Algorithm

procedure removeCTI (M, s, i)

if $I_0 \wedge s$ is SAT, **return** "P does not hold"

while $F_i \wedge T \wedge \neg s \wedge s'$ is SAT **do**

for $j \in [0, i]$
$F_j \leftarrow F_j \wedge \neg s$
end for

$t \leftarrow$ predecessor of s extracted from the SAT witness
$removeCTI(M, t, i - 1)$

end while

end procedure

Algorithm

procedure propagateClauses (M, k)

for $i \in [1, k]$
for every clause $c \in F_i$
if $F_i \land T \land \neg c'$ is UNSAT
$F_{i+1} \leftarrow F_{i+1} \land c$
end if
end for
end for

end procedure

Thank you!