SAT-Based Model Checking with Interpolation

Orna Grumberg
Technion, Haifa, Israel

iPRA workshop
July 2014

The beautiful slides are mostly borrowed from Yakir Vizel
Focus of the talk

• Interpolants in the propositional logic and their use in verification

• Interpolation for other logics is used, for instance, for software verification
  – Linear arithmetic, Reals, and others
Model Checking

- Given a system and a specification, does the system satisfy the specification?

![Diagram of Model Checking]

- System model
- MC
- No+CEX
- Yes
- Property
Outline

• Background on model checking

• SAT-based model checking with interpolation

• Model checking with interpolation sequence

• Model checking with backward and forward interpolations
System model
 Modeling

• System is modeled as \((V, \text{INIT}, T)\), where:
  - \(V\) is a finite set of variables
  - \(S\) – set of states – all valuations of \(V\)
  - \(\text{INIT} \subseteq 2^V\) is the set of initial states
  - \(T \subseteq 2^V \times 2^V\) is the set of transitions

• A safety property of the form \(\text{AG P}\)
  - “\(P\) holds in every reachable state of the system”
  - \(P\) is a formula over \(V\)
Translation to Propositional Formulas

- Four states:
  - Two Boolean variables: \( v_1, v_2 \)
- INIT: \( \neg v_1 \)
- T:
  - \( v'_1 = \neg v_1 \lor (v_1 \land v_2) \)
  - \( v'_2 = (v_1 \land v_2) \)
- P: \( \neg v_1 \lor \neg v_2 \) (Bad = \( \neg P = v_1 \land v_2 \))
Example

$T$ is a conjunction of constraints, one per component.

$g = a \land b$

$p = g \lor c$

$c' = p$

$T = \land \{\ 
g = a \land b,
p = g \lor c,
c' = p\ \}$
Reachability Analysis

• Problem definition:
  – Does the transition system have a finite run ending in a state satisfying \( \neg P \) ?
  – More precisely, is there a sequence of states \( s_0, \ldots, s_k \) s.t.:
    • \( s_0 \in I \) and \( s_k \in \neg P \)
    • for all \( 0 \leq i < k \), \((s_i, s_{i+1}) \in T\)

• Using automata-theoretic methods, model checking safety properties reduces to reachability analysis.
Forward Reachability Analysis

Does AG P hold?

\[ R_1 = \text{Img}(\text{INIT}, T) \]
\[ R_2 = \text{Img}(R_1, T) \]
\[ \ldots R_n = \text{Img}(R_{n-1}, T) \]
Termination when

- either a bad state satisfying $\neg p$ is found
- or a **fixpoint** is reached: $R_j \subseteq \bigcup_{i=0,j-1} R_i$
Main limitation

The state explosion problem:

Space and time requirements grow with the size of the model
SAT-based Model Checking

Main idea

• **Translate** the model and the specification to propositional formulas

• Use efficient tools (**SAT solvers**) for solving the satisfiability problem

• At the beginning it was **mainly** used for finding CEXs
DPLL-style SAT solvers

GRASP, CHAFF, MiniSAT, Glucose

• Objective:
  – Check whether a CNF formula is satisfiable or not
    • Either return a satisfying assignment
    • Or “UNSAT” and a refutation proof

• Approach:
  – Decision: Choose arbitrary variable+value for an unassigned variable
  – Propagate implications
  – Add conflict clauses to avoid rechecking assignments
Can the circuit output be 1?

\[(a \lor \neg g) \land (b \lor \neg g) \land (\neg a \lor \neg b \lor g)\]

\[(\neg g \lor p) \land (\neg c \lor p) \land (g \lor c \lor \neg p)\]

\[CNF(p)\]

\[p\] is satisfiable when the formula \(CNF(p) \land p\) is satisfiable
Bounded model checking

- Unfold the model $k$ times:
  \[ U = T^{<0>} \land T^{<1>} \land \ldots \land T^{<k-1>} \]

- Use SAT solver to check satisfiability of
  \[ I^{<0>} \land U \land \neg P^{<k>} \]

- If unsatisfiable:
  - property has no counterexample of length $k$
  - can produce a refutation proof
Bounded Model Checking

\[ \text{INIT}(V^0) \land T(V^0, V^1) \land \neg P(V^1) \]
Bounded Model Checking

\[\text{INIT}(V^0) \land T(V^0, V^1) \land T(V^1, V^2) \land \neg P(V^2)\]
Bounded Model Checking

\[ \text{INIT}(V^0) \land T(V^0, V^1) \land \ldots \land T(V^{k-1}, V^k) \land \neg P(V^k) \]
Bounded Model Checking

Terminates
• with a counterexample or
• with time- or memory-out

The method is suitable for falsification, not verification
Outline

• Background on model checking

• SAT-based model checking with interpolation

• Model checking with interpolation sequence

• Model checking with backward and forward interpolations
SAT-Based Verification

unbounded model checking

- Uses **BMC** for falsification
- Simulates forward reachability analysis for verification
- Identifies a termination condition
  - all reachable states has been found: “fixpoint”
Interpolants

- Given an unsatisfiable pair \((A,B)\) of propositional formulas
  - \(A(X,Y) \land B(Y,Z)\) is unsatisfiable

- There exists a formula \(I\) such that:
  - \(A \Rightarrow I\)
  - \(I \land B\) is unsatisfiable
  - \(I\) is over \(Y\), the common variables of \(A\) and \(B\)
Interpolation (cont.)

Interpolants from proofs

• When \( A \land B \) is unsatisfiable, SAT solvers return a proof of unsatisfiability in the form of a resolution graph.

• Given a resolution graph, \( I \) can be derived in linear time.

Pudlak, Krajicek 97, McMillan 03
ITP - Interpolation-based MC

McMillan, CAV 2003

\[ \text{INIT}(V) \land T(V, V^1) \land T(V^1, V^2) \land T(V^2, V^3) \land (\neg P(V^1) \lor \ldots \lor \neg P(V^3)) \]

- I over-approximates the states reachable from INIT in one transition
  - It satisfies P and cannot reach a bad state in two transitions or less
ITP – Interpolation-based MC

McMillan, CAV 2003

\[ I(V) \land T(V, V^1) \land T(V^1, V^2) \land T(V^2, V^3) \land (\neg P(V^1) \lor \ldots \lor \neg P(V^3)) \]

- \(I\) is fed back to the formula
  - A new interpolant \(I'\) is computed
  - Iterative process
Using Interpolation ($i=1$)

$INIT(V_0) \land T(V_0, V_1) \land \neg p(V_1)$

$I_1$

$I_1(V_0) \land T(V_0, V_1) \land \neg p(V_1)$

$I_2$

$I_2(V_0) \land T(V_0, V_1) \land \neg p(V_1)$

BAD

$\neg p$
Using Interpolation (i=2)

\[\begin{align*}
\text{INIT}(V_0) \land T(V_0, V_1) \land T(V_1, V_2) & \land (\neg q(V_1) \lor \neg q(V_2)) \\
\vdots \\
I'_1(V_0) \land T(V_0, V_1) \land T(V_1, V_2) & \land (\neg q(V_1) \lor \neg q(V_2)) \\
I'_2(V_0) \land T(V_0, V_1) \land T(V_1, V_2) & \land (\neg q(V_1) \lor \neg q(V_2)) \\
\vdots \\
I'_k(V_0) \land T(V_0, V_1) \land T(V_1, V_2) & \land (\neg q(V_1) \lor \neg q(V_2))
\end{align*}\]
• In ITP, short BMC formulas can prove the nonexistence of long CEXs
  – INIT is replaced by $I_k$ which over-approximates $S_k$

• If a satisfying assignment is found, the counterexample might be spurious
  – Since INIT is over-approximated

• Increase $k$ and start with the original INIT
• A fixpoint is checked whenever a new interpolant is computed

• For iteration $i$, every new interpolant is checked for inclusion in all previously computed interpolants for the same $i$:

  $I_n \Rightarrow \text{INIT} \lor \bigvee_{j=1,n-1} I_j$
In ITP, a computed interpolant is fed back into the BMC problem.

BMC problem is solved with a SAT solver.

Problems:
1. “Big” interpolant causes the BMC problem to be hard to solve.
2. Non-CNF interpolant needs to be translated to CNF.
Outline

• Background on model checking

• SAT-based model checking with interpolation

• Model checking with interpolation sequence

• Model checking with backward and forward interpolations
Interpolation-Sequence

• If \( A_1 \land A_2 \land \ldots \land A_k \) is unsatisfiable, then there exists an interpolation-sequence \( I_0, I_1, \ldots, I_{k+1} \) for \( (A_1, \ldots, A_k) \) such that:

\[
\begin{align*}
I_0 &= T \quad \text{and} \quad I_{k+1} = F \\
I_j \land A_{j+1} &\Rightarrow I_{j+1}
\end{align*}
\]

\( I_j \) - over common variables of \( A_1, \ldots, A_j \) and \( A_{j+1}, \ldots, A_k \)

• Each \( I_j \) can be computed as the interpolant of 
\[
A = A_1 \land \ldots \land A_j \quad \text{and} \quad B = A_{j+1} \land \ldots \land A_k
\]

– All \( I_j \)'s should be computed on the same resolution graph
Reachability with Interpolation-Sequence

- Unsatisfiable BMC formula partitioned in the following manner:

\[
\text{INIT}(V^0) \land T(V^0, V^1) \land T(V^1, V^2) \land T(V^2, V^3) \land \cdots \land T(V^{k-1}, V^k) \land \neg P(V^k)
\]
Reachability with Interpolation-Sequence

\[
\text{INIT}(V^0) \land T(V^0, V^1) \land T(V^1, V^2) \land T(V^2, V^3) \land \cdots \land T(V^{k-1}, V^k) \land \neg P(V^k)
\]

\[
T \quad I_1 \quad I_2 \quad I_3 \quad I_k \quad F
\]

\[
I_0 = T \quad \text{and} \quad I_{k+1} = F \\
I_j \land A_{j+1} \implies I_{j+1}
\]

\(I_j\) - over common variables of \(A_1,...,A_j\) and \(A_{j+1},...,A_k\)
Reachability with Interpolation-Sequence

- Compute a sequence of reachable states from BMC formulas
  - Forward Sequence: \(<F_0, F_1, \ldots, F_n>\)
- Sequence is over-approximated
  - \(F_i(V) \land T(V, V') \Rightarrow F_{i+1}(V')\)
  - \(F_i \Rightarrow P\)
- Integrated into the BMC loop to detect termination
Using Interpolation-Sequence

\[ INIT(V_0) \land T(V_0, V_1) \land \neg p(V_1) \]

\[ INIT(V_0) \land T(V_0, V_1) \land T(V_1, V_2) \land \neg p(V_2) \]
The Analogy to Forward Reachability Analysis

\[ \text{INIT}(V_0) \land T(V_0, V_1) \land \text{F}(1, 1, 2, 2) \land \text{F}(2, 2, V_3) \land \neg p(V_3) \]
Checking if a “fixpoint” has been reached

• $F_n \Rightarrow V_{j=1,n-1} F_j$

• Similar to checking fixpoint in forward reachability analysis:
  $R_k \subseteq U_{j=1,k-1} R_j$

• But here we check inclusion for every $2 \leq k \leq n$
  – No monotonicity because of the approximation

• “Fixpoint” is checked with a SAT solver
Outline

• Background on model checking

• SAT-based model checking with interpolation

• Model checking with interpolation sequence

• Model checking with **backward and forward interpolations**
Forward Reachability Analysis

\[ R_1 = \text{Img}(\text{INIT}, T) \]
\[ R_2 = \text{Img}(R_1, T) \]
\[ R_n = \text{Img}(R_{n-1}, T) \]

\[ \text{Bad} = \neg \text{P} \]
Interpolants

• Given an unsatisfiable pair \((A,B)\) of propositional formulas

• Then, there exists a formula \(I\) such that:
  – \(A \Rightarrow I\)
  – \(I \land B\) is unsatisfiable
  – \(I\) is over the common variables of \(A\) and \(B\)

• \(I = \text{Itp}(A,B)\)
Approximated Forward Reachability

• $F(V)$ - a set of states
• For the unsatisfiable formula $F(V) \land T(V,V') \land \neg P(V')$, define:
  
  $$A = F(V) \land T(V,V')$$
  $$B = \neg P(V')$$

• Approximated forward reachability
Backward Reachability Analysis

Does $AGp$ hold?

$B_n = \text{PreImg}(B_{n-1}, T)$

$B_2 = \text{PreImg}(B_1, T)$

$B_1 = \text{PreImg}(\neg P, T)$

Bad = $\neg P$
Duality In a SAT Query

• \text{INIT}(V) \land T(V,V') \land \neg P(V')

• We tend to read it "Forward"
  – From left to right
Duality In a SAT Query

- \( \text{INIT}(V) \land T(V,V') \land \neg P(V') \)

- We tend to read it "Forward"
  - From left to right

- We can also read it "Backward"
  - From right to left
  - Does the pre-image of the bad states intersect the initial states?
Approximated Backward Reachability

- $B(V)$ - a set of states
- For the unsatisfiable formula $\text{INIT}(V) \land T(V,V') \land B(V')$, define:
  \[
  A = T(V,V') \land B(V')
  
  B = \text{INIT}(V)
  \]

- Approximated backward reachability
Dual Approximated Reachability (DAR)

(Vizel, Grumberg and Shoham, TACAS 2013)

• Compute two sequences of reachable states
  – Forward Sequence: \(<F_0, F_1, \ldots, F_n>\)
  – Backward Sequence: \(<B_0, B_1, \ldots, B_n>\)

• Sequences are over-approximations
  – For the forward sequence:
    • \(F_i(V) \land T(V, V') \Rightarrow F_{i+1}(V')\)
    • \(F_i \Rightarrow P\)
  – For the backward sequence
    • \(B_{i+1}(V) \Leftarrow T(V, V') \land B_i(V')\)
    • \(B_i \Rightarrow \neg \text{INIT}\)
Dual Approximated Reachability (DAR)

• Two main phases during the computation
  – Local Strengthening
    • No unrolling
  – Global Strengthening
    • Limited unrolling
    • In case the Local Strengthening fails
Dual Approximated Reachability

\[ F_0 = \text{INIT} \]

\[ B_0 = \neg P \]

\[ \text{INIT}(V) \land T(V, V') \land \neg P(V') \]
Local Strengthening

What if $F_1$ and $B_1$ intersect each other?

There might be a counterexample
Local Strengthening

What if $F_1$ and $B_1$ intersect each other?

$F_1(V) \land T(V,V') \land B_0(V')$

$F_0(V) \land T(V,V') \land B_1(V')$
Local Strengthening

\[ B \quad A \quad A \quad B \]

\[ F_1(V) \land T(V, V') \land \neg P(V') \]

- Compute forward and backward interpolants
  - \( F_2 \) is the forward interpolant
  - Backward interpolant strengthens the already existing \( B_1 \)

\[ F_0 = \text{INIT} \]
Local Strengthening

\[ B_0 = \neg P \]

\[ \text{INIT}(V) \land T(V,V') \land B_1(V') \]

**Must be UnSAT**

- Compute forward and backward interpolants
  - \( B_2 \) is the backward interpolant
  - \( F'_1 \) is strengthening the already existing \( F_1 \)
Local Strengthening Fails

\[ F_0(v) \land T(v, v') \land B_0(v') \]
Global Strengthening

• Apply unrolling gradually
  – Start from the initial states
  – Try to reach the backward sequence using an increasing number of T’s
Global Strengthening

\[ F_0(\forall \mathbf{v}(V) \land \forall \mathbf{v'}(V' \land \forall \mathbf{v''}(V'' \land \forall \mathbf{v'''}(V''' \land \forall \mathbf{v'''}(V'''))) \land B_1(VP(V''')) \land \neg \mathbf{P}(V''')) \land T(V, V') \land T(V', V'') \land T(V'', V''') \land T(V''', V''))) \]
Global Strengthening
Interpolation sequence for UNSAT formula

\[ F_0(V) \land T(V, V') \land T(V', V'') \land T(V'', V''') \land B_1(V''') \]

\[ T \quad I_1 \quad I_2 \quad I_3 \quad F \]
Global Strengthening

\[ F_0(V) \land T(V, V') \land T(V', V'') \land T(V'', V''') \land B_1(V''') \]

- **Formula is unsatisfiable**
  - Extract an interpolation-sequence: \( I_1, I_2, I_3 \)
  - \( I_j \) over-approximates states reachable in \( j \) steps

- **Use \( I_j \) to strengthen \( F_j \)**
  - Example: \( F_3' = F_3 \land I_3 \)
  - \( F_3' \land B_1 \) is unsatisfiable

- **Re-Apply Local Strengthening**
  - \( F_3(V) \land T(V, V') \land \neg P(V') \) is unsatisfiable
Global Strengthening

• If a CEX exists – Full unrolling
• Otherwise, gradually unroll the model
  – Try to reach the Backward sequence
• When the backward sequence is not reachable
  – Extract interpolation sequence
  – Strengthen forward sequence
  – Reapply Local Strengthening
Checking if a “fixpoint” has been reached

• $F_k \Rightarrow V_{j=1,n-1} F_j$

• But we also have the backward sequence
  $B_k \Rightarrow V_{j=1,n-1} B_j$

• Same principle applies here check inclusion for every $2 \leq k \leq n$
(Local) Summary

• Use both **Forward** and **Backward** traversals in a **tight** manner

• *Mostly local – No unrolling*
  – *Inspired by IC3/PDR*

• *When unrolling is used, it is restricted*
  – *Experiments confirm*
We presented several methods for SAT-based (unbounded) model checking

- **Over-approximate** the (forward) reachability analysis
- Apply different methods for making the over-approximation **more precise**
  - Reduce number of spurious counterexamples
  - (Hopefully) help termination (fixpoint)
Thank You