Exploiting Craig Interpolants in Unbounded Model Checking of Hardware Designs

Danilo Vendraminnetto

Formal Methods Group, Politecnico di Torino, Italy.
Outline

• Background
  • Verification of hardware designs
  • Craig Interpolants in UMC

• Contributions
  • Redundancy removal and reduction in UNSAT proofs and ITPs
    • Heuristic procedure for scalable *ITP compaction*
  • Abstraction and refinement techniques for ITPs
    • Heuristic procedure for abstracting *without resorting to resolution proofs*

• Experimental results & Conclusions
ITP Proof Compaction

- Proof reduction
  - Recycle-pivots [Bar-Inal & al. HVCo8]
    - Exploiting proof topology:
      *proof node chains*

- Logic synthesis manipulations on the proof
  - Constant propagation
  - BDD-based sweeping (for equivalences)
  - Observability Don’t Care (lightweight)
ITP Circuit Compaction

- Proof into AIG
  - ODC (structural)
  - Logic synthesis
    - rewrite / refactor
      - using ABC tool
    - AIG balance
  - ITE-based decomposition
    - iff necessary
Interpolant Abstraction

- ITP+: take on improved Craig’s interpolation
- Incremental computation of interpolants using alternative techniques
  - Equivalence classes, mutual implications of state variables
  - Cube-based over-approximation, based on the detection of those state variables that are stuck at constant values

\[ \text{To}^+ (V') = \text{ITP}(\text{From} \land T, \text{Cone}^k) \]
**ITP⁺ - Abst. by iterative refinement**

\[
\text{IMG}^+_{\text{Adq}} \ (\text{From}, \ T, \ \text{Cone}^k) \\
\text{To}^+ = \text{Full\_state\_space} \\
\text{Foreach Class} \in \text{Abstraction\_classes} \\
\quad \text{Select abstraction} \\
\quad \text{To}^+_{\text{Class}} = \text{IMG}^+ \text{ using abstraction} \\
\quad \text{To}^+ = \text{To}^+ \land \text{To}^+_{\text{Class}} \\
\quad \text{if UNSAT}(\text{To}^+ \land \text{Cone}^k) \text{ return } \text{To}^+ \\
\text{return } (\text{To}^+ \land \text{ITP}(\text{From} \land T, \ \text{Cone}^k))
\]
ITP+ - Abstraction classes

- **Tightening** abstraction classes
  - Equivalent state variables
  - Constant state variables
  - SAT-based enumeration

- **Loosening** abstraction classes
  - Localization abstraction
  - Ternary abstraction
ITP+ - Constant state variables

From $T$

$V, V', W$

$V_i' = 1 \ ?$

If yes, simplify...

Refine: literal invariant

$T_{o+} = T_{o+} \land V_i'$

Simplify cone as well (constant prop.)...
Some experimental results...

- Interpolant abstraction
Come to my poster presentation for more details and experimental results
Thanks for your attention
Some experimental results...

- Interpolant compaction
Conclusions

- ITP-based MC heavily relies on scalability
- We developed effective techniques to improve standard ITPs.
  - Scalable techniques, applied incrementally
- Best suited as a second engine
  - Hard-to-prove properties (hard for IC3)
  - Explosion of standard interpolation
  - Can afford extra time (for memory)