To Split or to Group: From Divide-and-Conquer to Sub-Task Sharing in Verifying Multiple Properties

Paolo Enrico Camurati, Carmelo Loiacono, Paolo Pasini, Denis Patti, Stefano Quer
Dip. di Automatica ed Informatica, Politecnico di Torino, Torino, Italy
• Multiple properties/targets for same model
  – As primary inputs
  – Generated by decomposition
• Handle different properties as sub-problems
  – Target sorting and/or grouping
• Interaction and synergy among proofs
  – Reuse reduction
  – Propagate learning
• Focus on large circuits with several properties
  – Between 500 and 50K properties
  – Between 500 and 500K latches
• Subset of HWMCC’13 (multiple and single tracks)
• Motivation

• **Property grouping**
  – clustering
  – verification with learning

• Property decomposition
  – partial verification
  – coverage estimation

• Conclusions and future works
Multiple properties verification

One at a time

State Reg_{n-1}
• Straightforward verification
  – sequential
  – individual checks
• Overhead
  – initialization and finalization of single properties
• Repetition of shared sub-tasks
• Group properties together $\mathcal{P}: p = \bigwedge_i p_i$
• Tuning to avoid scalability issues

Cooperation: share CEXes, invariants

Grouping & Sorting Properties
• Several strategies
  – sort properties by expected verification effort
  – classify properties according to mutual affinity
• Group properties in subsets
  – tune verification within subset
• Address scalability issues
  – COIs size explosion
• Exploit learning
  – reuse discovered invariants
    • cluster to cluster
    • target to target
  – reuse reductions and simplifications
  – trade off between usability and size/costs
• Filter CEXes
  – reorganize clusters removing failed properties
• One hard property may hinder whole cluster verification
• Affinity estimated based on support variables $V_p$ within COIs

• Jacquard Index $\alpha = \frac{|V_j \cap V_k|}{|V_j \cup V_k|}$

• Grouping performed if resulting value is above a chosen threshold

• Verification starts from properties with smaller COIs
• Comparison between our sequential and cluster based approaches
• Best result among different clustering thresholds
• Usually at least as good as sequential verification
• COIs sizes tend not to grow so much to become intractable
• Values normalized considering only non-constant properties
• The number of allowed clusters influences verification outcome
• Automatic tuning of thresholds is an on-going effort
• The number of allowed clusters influences verification outcome
• Automatic tuning of thresholds is an on-going effort
• The number of allowed clusters influences verification outcome
• Automatic tuning of thresholds is an on-going effort
• The number of allowed clusters influences verification outcome
• Automatic tuning of thresholds is an on-going effort
• The number of allowed clusters influences verification outcome
• Automatic tuning of thresholds is an on-going effort
• Motivation
• Property grouping
  – clustering
  – verification with learning
• Property decomposition
  – partial verification
  – coverage estimation
• Conclusions and future works
• Property decomposition aimed at full verification

• Easy-to-solve properties of little interest
  – introducing overhead
  – no information to gain

• Hard-to-solve still unsolvable as a whole
  – sub-problems can be as hard as the original
• Compositional verification of monolithic properties

• Relax goal of full verification
  – infer information from covered parts (bounds, CEXes, ...)
  – better than nothing at all
Divide & Conquer approach for hard-to-solve properties $P = \bigwedge p_i$

Identify a subset of “easier” properties
- smaller COIs
- sub-space constrained
- only describing sub-behaviors

Treat original property as a grouped instance

SAT solvers as sub-target enumerator
• Derive target from invariant \( t = \neg p \)

• Consider a minterm as first sub-target

• Acquire over-approximated stateset representations as sub-product of previous verification \( R_0, \ldots, R_k \)

• Iteratively select targets that hit the innermost reachable state ring

• Terminate upon
  – identifying a partial target as reachable, disproving the property
  – acquiring a strong enough \( R \) set to prove the original property
Coverage estimation

- Based on size/percentage of reachable states
- State space estimation based on graph-based algorithm
- Derived from life sciences and “capture-mark-recapture” approaches
- Inherently difficult to produce almost exact estimation
- Ongoing work in this direction
• Focus on hard-to-solve single property designs

• SAT properties:
  – BMC runs to identify CEX bounds

• UNSAT properties:
  – Standard verification to identify pass bounds

• Partial verification
  – Diminished time limit for sub-properties verification through UMC
  – Bound estimation derived from these runs

Partial/Exact Bound Ratio
• Motivation
• Property grouping
  – clustering
  – sequential verification with learning
• Property decomposition
  – partial verification
  – coverage estimation
• Conclusions and future works
Conclusions and future work

• Preliminary results are promising and show room for improvement
• Further investments in clustering techniques and heuristics
• Automatization of threshold selection and cluster parametrization
• Further research in partial verification as indicator for currently untreatable instances
To Split or to Group: From Divide-and-Conquer to Sub-Task Sharing in Verifying Multiple Properties

Paolo Enrico Camurati, Carmelo Loiacono, Paolo Pasini, Denis Patti, Stefano Quer
Dip. di Automatica ed Informatica, Politecnico di Torino, Torino, Italy