More Scalable LTL Model Checking via Discovering Design-Space Dependencies ($D^3$)
Rohit Dureja and Kristin Yvonne Rozier

1. Motivation

Airspace Allocation (safe separation)

- The design of complex systems often requires analyzing several variants of the system under development for:
  - narrowing in on the final system design, and
  - check capabilities of system with varying features.
- The design choices constitute the system’s design space.

Model checking aids system development via a thorough comparison of all design choices

2. Modeling Design Spaces

Classical Method – Every design choice is a model. very hard to cross-validate as design-space grows

Scalable Method – Every design choice is a parameter. efficient, easier to maintain as design evolves

- Parameters are added as preprocessor directives.
  - works with off-the-shelf checkers, like NUXMV
  - every parameter configuration is a valid model

Combinatorial Transition System (CTS)

- Parameters $P_1, P_2$ enable design choices

3. Problem Statement

- NASA NextGen Air Traffic Control System
  - 4.0x speedup
  - Crossover point (~ 120 models)
- BOEING Wheel Braking System
  - Heuristics:
    H1: Maximum Dependence
    H2: Property Grouping
  - Fast multi-property verification

4. Our Solution

Discover Design-Space Dependencies, or $D^3$
- Reduces design space by finding dependencies between:
  - parameters (number of models to check)
  - properties (number of model-checking runs)
- Is fully automatic, works with off-the-shelf checkers

i) Minimize number of parameter configurations (GENPC)

- CTS with Boolean parameters $P_1, P_2, P_3$
- Same Models
  - If $P_1 = 1$, then $P_2, P_3$ are redundant.

Can be checked together!

- Finds dependencies between parameter settings via reduction to a reachability problem.

ii) Minimize number of model-checking runs (CHECKRP)

- $\varphi_1 = \Box p$
- $\varphi_2 = \Box (p \land q)$
- $\varphi_3 = \Box (p \lor q)$

$M \models \varphi_2$ then $M \models \varphi_1$
- $\varphi_1$ and $\varphi_2$ are dependent

$M \models \varphi_2$ then $M \models \varphi_3$
- $\varphi_2$ and $\varphi_3$ are dependent

- Finds dependencies between properties via fast LTL satisfiability checking.

5. Experimental Results

- Property Table
- Result Array
- One check
- Four results