

Formal Coverage

Moshe Y. Vardi

Rice University

Model Checking: From Falsification to Verification

Traditional Model Checking (1983-1996):

- Given design D and property ϕ , check if $D \models \phi$.
- If answer negative, obtain a counterexample, and use it to debug D .
- If answer positive, you are done.

Main value: *falsification*

New Model Checking: Focus also on verification!

- *Vacuity*: Do I believe a “yes”?
- *Coverage*: Have I checked enough properties?

Relationship with Classical Coverage

Similarity:

- Classical coverage: Have I run enough tests?
- *Formal Coverage*: Have I checked enough properties?

Dissimilarity:

- Classical coverage: Easy to explain what it means for simulation to cover much of the design.
- *Formal Coverage*: What does it mean?

Difficulty of Defining Formal Coverage

Difficulty: Checking that $D \models \phi$ requires an exhaustive search of D 's state space! It seems like 100% coverage!

Solution: Focus on collection ϕ_1, \dots, ϕ_n of properties.

KGG'99, OneSpin : Full coverage means that D is “*equivalent*” to ϕ_1, \dots, ϕ_n .

Difficulty:

- Equivalence unlikely, since many design aspects cannot be captured by formal properties.
- Equivalence either hold or does not hold; quantitative measures not possible.

Measuring Formal Coverage

Formal Coverage

- Define design *aspects*, e.g., states, latches, etc.
- Compute *relevant* aspects, i.e., aspects that *affect* satisfaction of properties.
- Measure coverage of relevant aspects.

This is a framework!

- Depends on *aspects*.
- Depends on *relevance*.

[HKHZ'99]:

- Aspects= states in Kripke structure.
- Affects= change in value of variable in state falsifies property.

General Framework: [CKV'01,CKKV01,CKV'03]

Aspects

Fundamental Lesson from Classical Coverage:

There is no unique way to measure coverage – many syntactic and semantics *metrics*:

- Code coverage
- Latch coverage
- Toggle coverage
- FSM coverage
- ...

Observation: Different metrics correspond to different design aspects.

The Mutation Principle: An aspect is relevant if changing it *affect* satisfaction of properties.

Standard Relevance

The Mutation Principle I: An aspect is relevant if changing it *falsifies* a property.

Difficulty: Often a change results in reduction in number of design behaviors.

- *Universal* properties (e.g., linear-time properties) are preserved under reduction in behaviors – how can we measure relevance?

Vacuity

Intuition: $D \models \phi$ holds *vacuously* if ϕ holds “*too easily*” [BBER’97].

Example: *always* $(p \rightarrow \text{eventually } q)$ holds too easily if p never holds or q always holds.

The Mutation Principle II: An aspect is relevant if changing it *vacuify* a property.

General Framework: Algorithms for computing coverage wrt different notions of aspects and relevance.

From Theory to Practice

Important: Formal coverage is not important unless it is useful. Usefulness is determined by practice.

Note: The “from-falsification-to-verification change” was driven by industry:

- *Vacuity*: IBM
- Formal coverage: Intel

Conclusion: Further academic-industrial interaction required!