Coverage Discounting: Improved Testbench Qualification by Combining Mutation Analysis with Functional Coverage

Nicole Lesperance, Peter Lisherness, Tim Cheng, University of California, Santa Barbara

1. Problem
   Functional coverage evaluates activation, ignoring propagation and detection

2. An Existing Solution
   Mutation Testing
   Synthetic fault insertion
   ✔ Evaluates propagation
   ✔ Evaluates checker
   ✔ Minimal infrastructure
   ✗ Hard to analyze
   ✗ Long runtime

   Example of a Weak Checker Inhibiting Debug Effort
   Performance bugs can be a blindspot of functional checkers. Suppose burst mode is disabled on a bus controller – does the testbench notice?

3. Coverage Discounting – Our Solution
   Coverage Discounting
   ✔ Evaluates propagation
   ✔ Evaluates checker
   ✔ Minimal infrastructure
   ✗ No special analysis
   ✔ Manageable runtime

   Main Idea
   Use mutation results to revise functional coverage score

4. Discounting Flow
   Discounting = Change in Coverage + Undetected Mutants

5. Experimental Results
   Concise, Functional Results
   Compare to mutation analysis:
   - 1588 Mutants:
     7 not activated
     106 not propagated
     33 not detected
   - Total 146 mutants demand attention
   - 846 Coverpoints:
     4 uncovered
     3 discounted
   - All discounted relate to specific unchecked functions:
     (Loopback, timeout interrupt identification register)

   Identifies Checker Bugs
   OpenRISC SoC

   Confidence Metric (DECO Score)
   ➢ Point confidence: # of times a coverpoint is suppressed
   ➢ DECO(n): percentage of points with point confidence > n

   Q: Is the coverpoint set adequately challenged by the mutants?
   A: Yes, if DECO score is sufficient

   Q: When can we stop simulating?
   A: When DECO score exceeds a predetermined threshold or when a point is discounted

   Efficient Fault/Test Ordering
   Q: What is the optimal mutant simulation ordering for coverage discounting?
   1. Test Selection: Choose test covering the most low confidence points
   2. Mutant Selection: Select mutant activated by the fewest tests

6. Acknowledgment
   This work was supported by the Semiconductor Research Corporation under GRC CADTS Verification.