Random Testing: Outline

- RT: advantages and tradeoffs
- RT vs pseudorandom testing (PR)
- Coverage and detectability profile
- Hardware and software DPs
- C(L) for random and pseudorandom tests
- High and low testability faults during early & late testing
- Implications of a late asymmetric profile
Random Testing

- Extensively used for both hardware and software
- Ideally each input is selected randomly. PR (Pseudorandom) schemes approximate random.
- Generally quite effective for moderate coverage.
  - Coverage hard to determine a priori.
  - Ineffective for random-pattern-resistant faults.
  - Coverage tools: Random (functional) followed by Structural testing.

Random Testing: Advantage

- No test generation using structural information needed.
- Test set-up using comparison:

  - Alternative: Is response reasonable? (software testing)
Pseudorandom (PR) Testing

- Unlike true random, reproducible.
- Will not repeat until all combinations applied.
- **Generation:** usually just-in-time (not stored).
  - Autonomous linear feedback shift register (ALFSR).
  - Cellular automata etc possible.
- **Some randomness properties** satisfied, but not all.

Coverage Achieved

- Coverage grows fast in the beginning, saturates near end.
- Is it described by
  - $C(L) = 1 - e^{-aL}$?
  - No, doesn’t fit.
- It is controlled by distribution of detectability of faults.
- Detectability profile (Malaiya & Yang ’84):
  - $H = \{h_1, h_2, \ldots, h_N\}$
    - $N$: total possible vectors
    - $h_k$: number of faults detected by exactly $k$ vectors.
  - Total faults $M = \sum h_k$
  - $h_1$: number of least testable faults
Detectability Profiles: Ex

- **CECL Full adder**
  Inputs=4 (N=16), M=90
  \[ H=(h_1,h_2,h_3,h_4,h_5,h_6,h_8) = (1,11,2,43,21,4,8) \]

- **Schneider’s counterexample:**
  Inputs=4 (N=16), M=44
  \[ H=(h_1,h_2,h_3,h_{14}) = (23,19,1,1) \]

Coverage with L random vectors

- \( h_k \) out of \( M \) defects detectable by exactly \( k \) vectors: detection probability \( k/N \)
- \[ P\{\text{a defect with dp } k/N \text{ not detected by a vector}\} = (1 - \frac{k}{N}) \]
- \[ P\{\text{a defect with dp } k/N \text{ not detected by } L \text{ vectors}\} = (1 - \frac{k}{N})^L \]
- Of \( h_k \) faults, expected number not covered is \( (1 - \frac{k}{N})^L h_k \)
- Expected test coverage with \( L \) vectors
  \[ C(L) = 1 - \sum_{k=1}^{\infty} \left(1 - \frac{k}{N}\right)^L \frac{h_k}{M} \]
Ex: C(L) and components for CECL Full Adder

<table>
<thead>
<tr>
<th>Hk</th>
<th>1</th>
<th>11</th>
<th>2</th>
<th>43</th>
<th>21</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.1250</td>
<td>0.1875</td>
<td>0.2500</td>
<td>0.3125</td>
<td>0.3750</td>
<td>0.5000</td>
</tr>
<tr>
<td>5</td>
<td>0.2756</td>
<td>0.4978</td>
<td>0.6459</td>
<td>0.7627</td>
<td>0.8484</td>
<td>0.9046</td>
<td>0.9688</td>
</tr>
<tr>
<td>10</td>
<td>0.4755</td>
<td>0.7309</td>
<td>0.8746</td>
<td>0.9427</td>
<td>0.9794</td>
<td>0.9960</td>
<td>0.9990</td>
</tr>
<tr>
<td>15</td>
<td>0.6202</td>
<td>0.8651</td>
<td>0.9556</td>
<td>0.9866</td>
<td>0.9964</td>
<td>0.9991</td>
<td>1.0000</td>
</tr>
<tr>
<td>20</td>
<td>0.7249</td>
<td>0.9308</td>
<td>0.9943</td>
<td>0.9999</td>
<td>0.9999</td>
<td>1.0000</td>
<td>0.9965</td>
</tr>
</tbody>
</table>

After 20 vectors:

<table>
<thead>
<tr>
<th></th>
<th>covered</th>
<th>remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.72</td>
<td>1.00</td>
<td>0.28</td>
</tr>
<tr>
<td>10.24</td>
<td>0.76</td>
<td>0.03</td>
</tr>
<tr>
<td>1.97</td>
<td>0.14</td>
<td>0.01</td>
</tr>
<tr>
<td>42.86</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>20.99</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Coverage of partitions
Shift in profile with progress in testing

Coverage Obtained by L Vectors

• For PR tests (McClusky 87)
  \[ C(L) = 1 - \sum_{k=0}^{N-L} \frac{N-L \cdot C_k \cdot h_k}{\sum C_k} \cdot \frac{h_k}{M} \]
  \[ = 1 - \sum_{k=1}^{N} (1 - \frac{k \cdot h_k}{N \cdot M}) \text{ (for Random)} \]
  
  • For large L, terms with only low k (i.e. faults that are hard to test) have an impact. Thus only lower elements of H need to be estimated.
  
  • For CECL Full Adder,
  \[ C(15) = 1 - [4.2 + 16.4 + 0.9 + 6.3 + 0.84 + 0.03 + 0 + \ldots] \cdot 10^{-3} \]
 Detectability Profile: software

- Regardless of initial profile, after some initial testing, the profile will become asymmetric.
- Dunham’s data based on NASA experiments for 16 faults.

![Graph showing error rate vs. number of faults]

 Detectability Profile: software

- Adam’s Data

![Graph showing defect detection rates and error rates]
Detectability Profile: Software

- Software detectability profile is exponential (Adam’s data, IBM).
- Justification: Early testing will find & remove easy-to-test faults.
- Testing methods need to focus on hard-to-find faults.

Implications: Fault Seeding

- A program has \( x \) defects. We want to estimate \( x \).
- Seed \( j \) new faults.
- Do some testing. Let faults found be \( j_1 \) seeded faults and \( x_1 \) original faults.
- Assuming \( j_1/j = x_1/x \) we get \( x = x_1 \frac{j}{j_1} \)

However, in reality the \( x \) faults include harder faults to test,

\[
\frac{j_1}{j} > \frac{x_1}{x} \quad \text{hence} \quad x > \frac{x_1 j}{j_1}
\]
Implications: Estimation by Inspection Sampling

- Software with x bugs is inspected by two separate teams that finds $x_1$ and $x_2$ bugs respectively, of which $x_3$ are shared.
- Assuming $x_1/x = x_2/x$, we get
  $$x = \frac{x_1 x_2}{x_3}$$

- However actually since x includes more harder to test faults,
  $$\frac{x_3}{x_2} > \frac{x_1}{x} \text{ hence } x > \frac{x_1 x_2}{x_3}$$

Implications: fault exposure ratio

Let $N(t)$ be the number of bugs at time $t$ during testing, then if $a$ is a parameter,
$$\frac{dN(t)}{dt} = -aN(t)$$

If $a$ is constant, then $N(t) = N(0)e^{-at}$ [expo SRGM]
However in random testing a should decline as faults get harder to find.
If testing is intelligent, then a can rise, which can give rise to Logarithmic SRGM.
References