Fault Tolerant Computing

CS 530

Random Testing

Yashwant K. Malaiya

Colorado State University

Random Testing: Outline

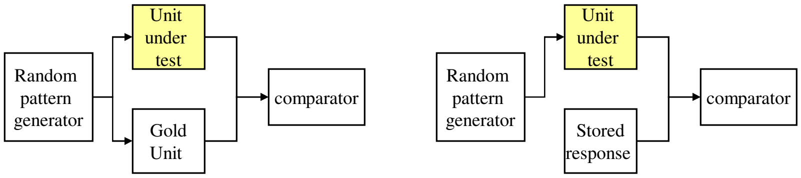
- **RT: advantages and tradeoffs**
- **RT vs pseudorandom testing (PR)**
- **Coverage and detectability profile**
- **Hardware and software DPs**
- **C(L) for random and pseudorandom tests**
- **High and low testability faults during early & late testing**
- **Implications of a late asymmetric profile**

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:**



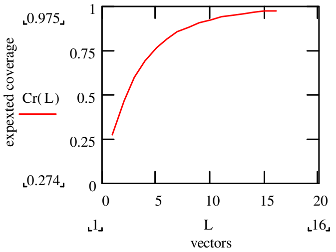
- **Alternative: Is response reasonable?{software testing}**

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

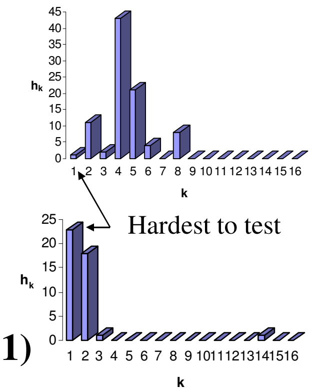
- **Coverage grows fast in the beginning, saturates near end.**
- **Is it described by**
    - $C(L)=1-e^{-aL}$?
    - No, doesn't fit.
- **It is controlled by distribution of detectability of faults.**
- **Detectability profile (Malaiya &Yang '84):**
- **$H=\{h_1, h_2, .. h_N\}$**
    - N: total possible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



- **Total faults $M=\Sigma h_k$**
- **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
  $H=(h_1,h_2,h_3,h_{14})=(23,19,1,1)$



Hardest to test

Coverage with L random vectors

- **h_k out of M defects detectable by exactly k vectors: detection probability k/N**
- **P{a defect with dp k/N not detected by a vector} =** $(1-\frac{k}{N})$
- **P{a defect with dp k/N not detected by L 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**

$$\mathrm{C(L)} = 1-\sum_{k=1}^{N}(1-\frac{k}{N})^L \frac{h_k}{M}$$

Ex: C(L) and components for CECL Full Adder

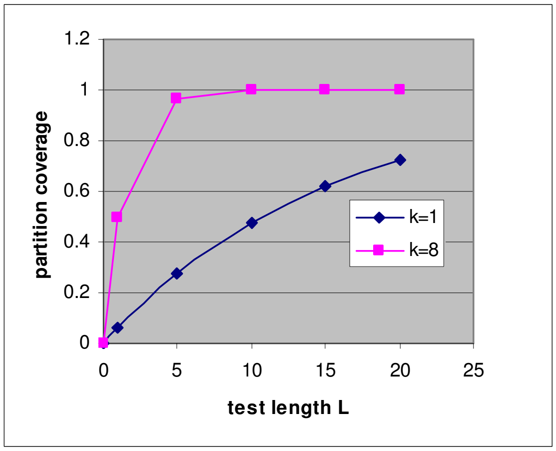
CECL full adder **N** **16** **M** **90**

Hk	**1**	**11**	**2**	**43**	**21**	**4**	**8**	
k =>	1	2	3	4	5	6	8	**Coverage**
L								
0	0	0	0	0	0	0	0	0
1	0.0625	0.1250	0.1875	0.2500	0.3125	0.3750	0.5000	0.2736
5	0.2758	0.4871	0.6459	0.7627	0.8464	0.9046	0.9688	0.7652
10	0.4755	0.7369	0.8746	0.9437	0.9764	0.9909	0.9990	0.9263
15	0.6202	0.8651	0.9556	0.9866	0.9964	0.9991	1.0000	0.9710
20	0.7249	0.9308	0.9843	0.9968	0.9994	0.9999	1.0000	0.9865

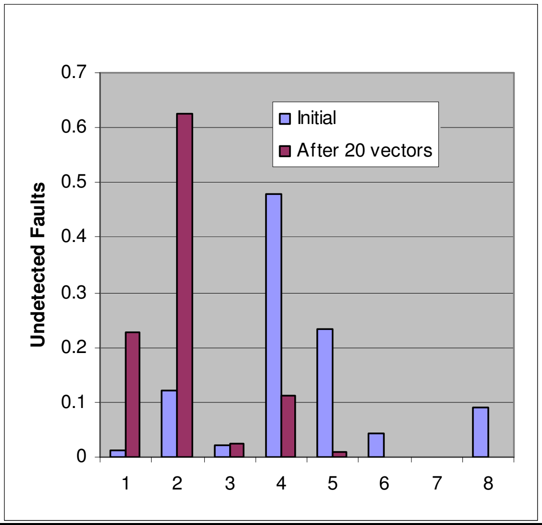
After 20 vectors:

covered	0.72	10.24	1.97	42.86	20.99	4.00	8.00
remaining	0.28	0.76	0.03	0.14	0.01	0.00	0.00

Coverage of partitions



Shift in profile with progress in testing



Coverage Obtained by L Vectors

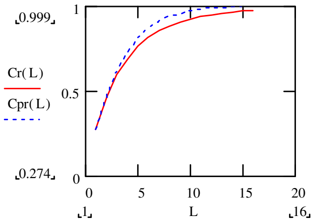
- For PR tests (McClusky 87)

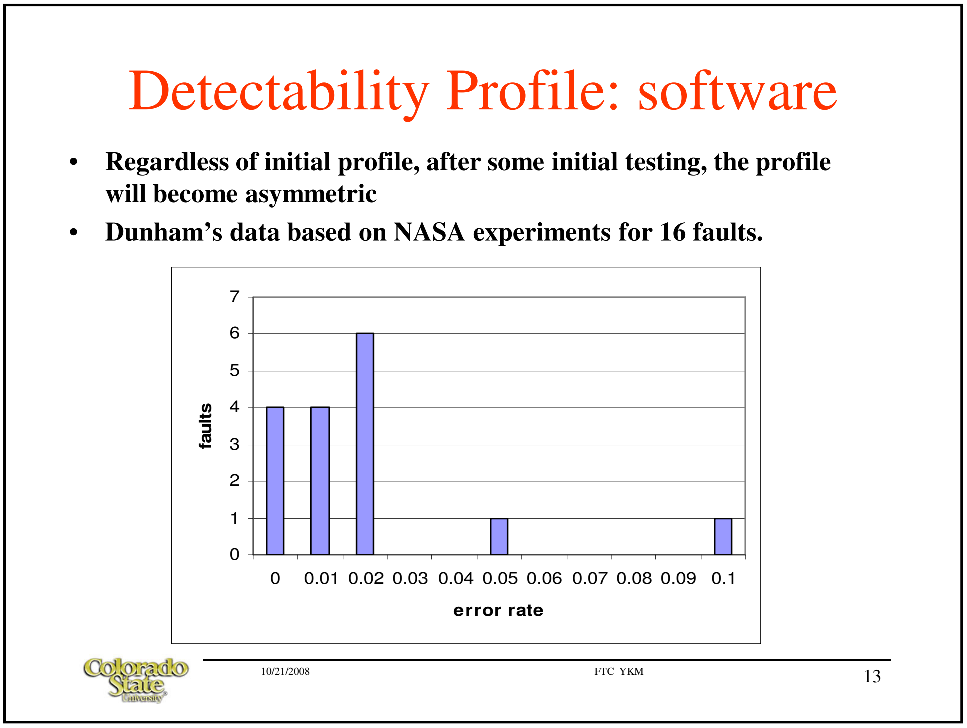
$$C(L) = 1 - \sum_{k=1}^{N-L} \frac{{}^{N-L}C_k}{{}^{N}C_k} \frac{h_k}{M}$$

$$\approx 1 - \sum_{k=1}^{N} \left(1 - \frac{k}{N}\right)^L \frac{h_k}{M} \text{ (for Random)}$$

- 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,

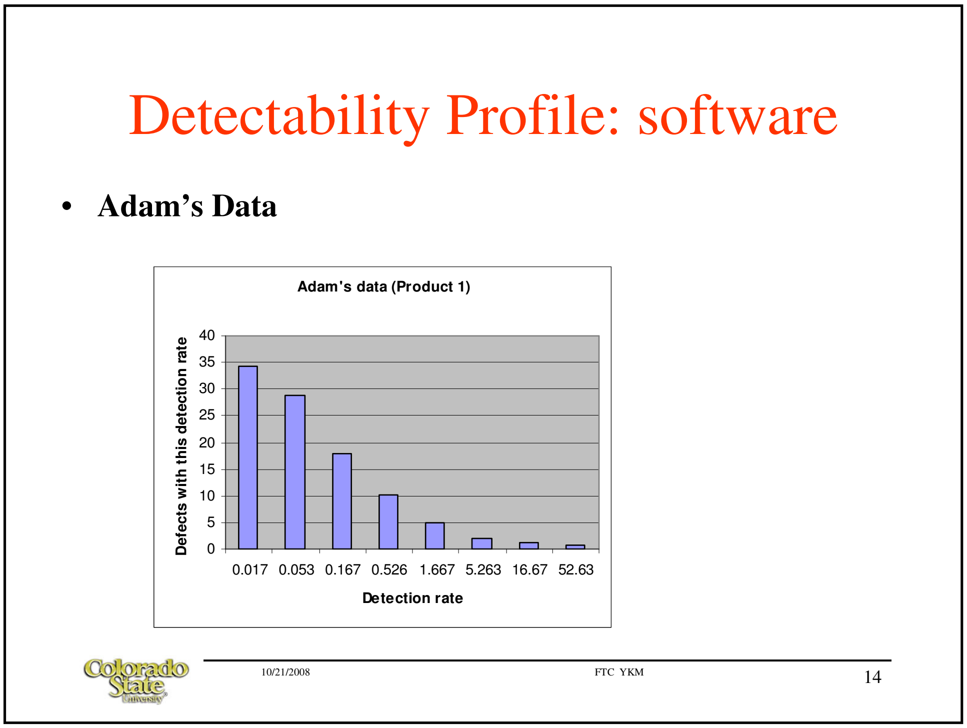
$$C(15) = 1 - [4.2 + 16.4 + 0.9 + 6.3 + 0.84 + 0.03 + 0 + \ldots] \cdot 10^{-3}$$





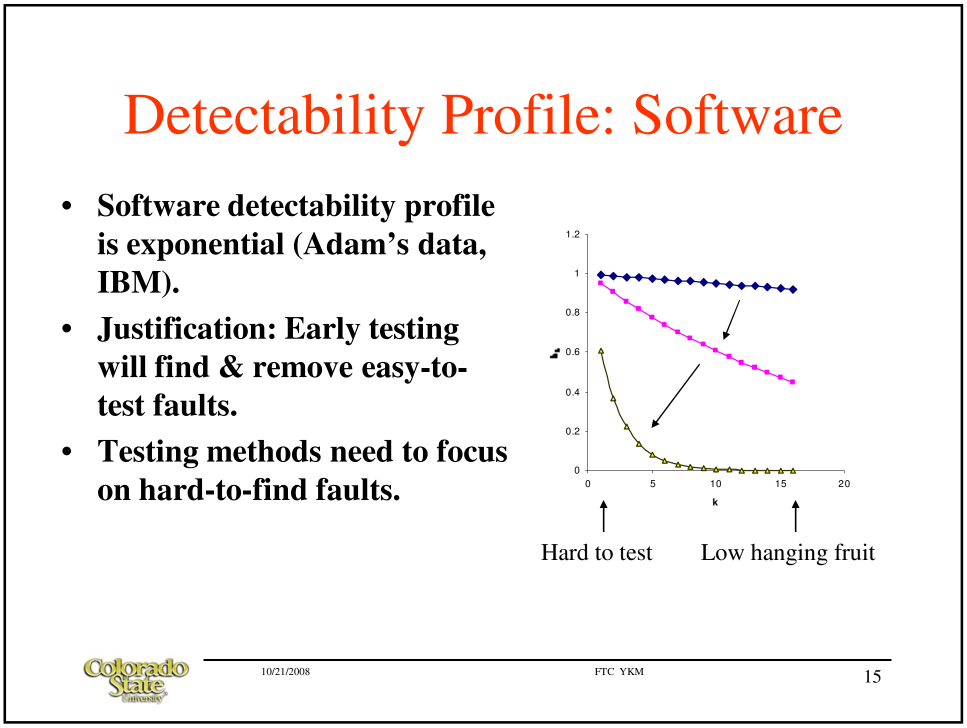
Detectability Profile: software

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



Detectability Profile: software

- Adam’s Data



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.**

Hard to test

Low hanging fruit

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,**

$$\frac{j_1}{j} > \frac{x_1}{x} \quad hence\ x > \frac{x_1 j}{j_1}$$

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 $x_1/x = x_3/x_2$ we get**

$$x = \frac{x_1 x_2}{x_3}$$

- **However actually since x includes more harder to test faults,**

$$\frac{x_3}{x_2} > \frac{x_1}{x} \; hence \; x > \frac{x_1 x_2}{x_3}$$

Implications: fault exposure ratio

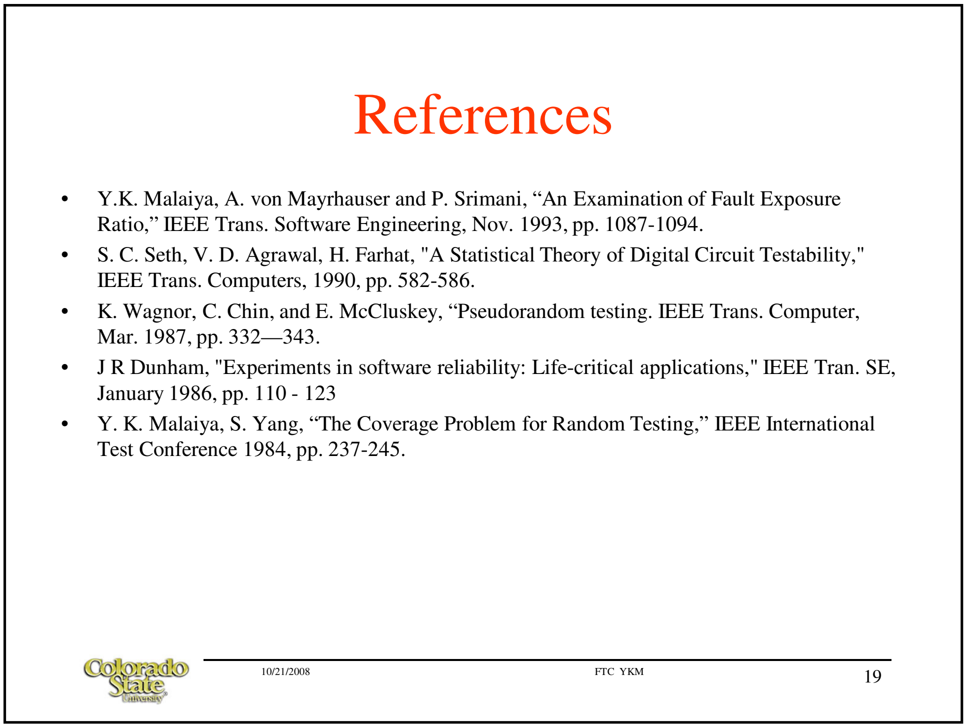
Let N(t) be the number of bugs at time t during testing, then if a is a parameter,

$$\frac{dN(t)}{dt} = -aN(t)$$

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

- Y.K. Malaiya, A. von Mayrhauser and P. Srimani, “An Examination of Fault Exposure Ratio,” IEEE Trans. Software Engineering, Nov. 1993, pp. 1087-1094.
- S. C. Seth, V. D. Agrawal, H. Farhat, "A Statistical Theory of Digital Circuit Testability," IEEE Trans. Computers, 1990, pp. 582-586.
- K. Wagnor, C. Chin, and E. McCluskey, “Pseudorandom testing. IEEE Trans. Computer, Mar. 1987, pp. 332—343.
- J R Dunham, "Experiments in software reliability: Life-critical applications," IEEE Tran. SE, January 1986, pp. 110 - 123
- Y. K. Malaiya, S. Yang, “The Coverage Problem for Random Testing,” IEEE International Test Conference 1984, pp. 237-245.