

PROBABILISTIC VS DETERMINISTIC TESTING

A report on debate Proceeding

Kartheek Arun Sai Ram Chilla

Kach15@students.bth.se

Pr. Number:9304163778

Abstract-In this paper, I submit all Inclusive report related to the discussion made during the report. The debate was about whether the deterministic Testing is preferred over the probabilistic testing. By providing the pragmatic proofs that speak about the deterministic testing is far much better or preferable once when compared to the probabilistic testing. The report also explains about the claims and arguments that were posed against deterministic testing and the evidences that arguments are wrong about the deterministic testing. At the end of the debate I elucidated the area under which the deterministic testing exists and preferably test works much better over the probabilistic testing.

Keywords: Deterministic Testing, Probabilistic Testing, Advantages and Disadvantages.

1 INTRODUCTION:

Deterministic testing is also called stoicistic testing[1] which is very desirable as for known set of same inputs no matter how many times its run unchangeable same results occur. Its input data value are known beforehand In deterministic testing once you start the system you know exactly what is going to happen. Example: If you know the initial deposits and the interest rates than you can determine the amount in one year (its value is measurable accurately) in deterministic testing. Unlikely the probabilistic testing or stochastic testing, in which something will happen don't know what will happen as the outcome is uncertain so you cannot predict the value we can only predict the limit into which the value might fall.[1] The user requirements might not be same as before they are modified so does the criteria for testing has to be molded for greater performance over the previous versions Example: In semiconductor technologies manufacturing of systems with multiple identical ICs but enforces in-field test as manufacturing test only is not Sufficient. It is most desirable to make use of the same test during production test and in-field test; hence, embed the high-quality deterministic tests....so, even at the beginning of the testing might be probabilistic but at the end for requirement based results deterministic criteria should be considered[2]. There are also other important notable advantages of deterministic testing over Probabilistic testing which are further imbibed in the report.

2 ADVANTAGES AND DISADVANTAGES

Based on conducting search in Engineering village...a few studies are being shortlisted that support our discussion on deterministic testing

ADVANTAGES OF DETERMINISTIC TESTING OVER PROBABILISTIC TESTING:

Below several advantages are outlined. The studies show that use of deterministic testing is inevitable. Dr. J Anusz Rajski Mentor Graphics Cooperation ,San Jose, CA introduced the Embedded Deterministic Testing (EDT) to reduce the scan test data and time[3] compared to ATPG tool (design for testability (DFT) based on scan and automatic test pattern generation (ATPG) which had been used for high quality manufacture testing for many years as it was desirable method, in later 2000 as the size of designs increased) in which only a small percent of scan cell are specified and remaining positions are randomly filled . The results show that for industrial circuits with test cubes with low fill rate, ranging from 3% to 0.2%, the EDT method results in compression ratios of 30 to 500 times[3].

Level of Work: The deterministic testing is easy and sometimes even straight forward implementation [1] In random versus the deterministic testing for RAM's The deterministic testing requires an order of $(n \log n)$ compared to random testing which remains linear as function of n ($447n$) for $Qd=.0003$.the above is for 3-coupling the deterministic comparisons will be more impressive for v-coupling $v > 3$ but for small cell sizes like stuck-at faults the deterministic is shorter than probabilistic. A probabilistic behavior can be replaced into deterministic by replacing random key generator with mock proxy that returns deterministic values[4].Cost effective: The deterministic testing are lesser expensive to implement and uses same set of parameters inside the test case and also which don't differ with the test runs. [1]

B) DISADVANTAGES OF DETERMINISTIC TESTING OVER PROBABILISTIC TESTING:

Design: The probabilistic testing reduces the amount of testing time....it help in potentially finding more errors and system bugs.

Cost: The probabilistic estimations when compared to Deterministic estimations are more expensive. The data and

parameters inside the test case will differ between the test runs. [1]

Level of skill: The probabilistic testing requires highly skilled people to implement. It is highly taken into consideration by many of the researches. [1]

Time constraint: They are dependent on software upgrades if new products are designed then testing of different parameters and reconfiguration and recoding is to be made. P. Thevenod-Fosse, H. Waeselynck, Y. Crouzet Toulouse cedex, FRANCE investigated the fault revealing power for 10 various test conditions.[5] The stimulated results in terms of percent show the mutants that were killed by test sets. The statistical testing killed more mutants only 0.2% of the total are alive than that of the deterministic testing. ...[5] pascal Thevenod-Fosse, Helene Weselynck proposed statistical testing as verification technique in complex systems. In complex systems, the usability automatic test generator for revealing high fault revealing is quite a challenge then the probabilistic testing is more pragmatic approach as the input domain versus the random test data graphs reveal the enhanced way of failure probability of program under a test result[5].

Keqin Li and David Lee proposed a way to detect the hierarchical networks as the network communications are expanding to large areas the controlling it is very difficult with Probabilistic testing algorithms it can be easy.[4] Due to size and complexity of the network topology use of network topology equivalence to reduce the testing of each topology equivalence class representation. we used the probabilistic testing to show the high fault coverage can be made ease with polynomial number of test on PSTN, ATM PNNI and other hierarchical networks[4].

3) CONCLUSIONS

Based on advantage there are empirical evidences that the Deterministic testing is better over Probabilistic testing. The Pros team that is opponent team claimed that Q1) Software is mainly project scheduling based and the effect on Time and budget trade off based on the consumer needs how can deterministic effect this scenario?

A) Time and budget trade off are interdependent in every company as the requirement of consumers are bound to them. The scheduling of project in time its deterministic if sample is based on only time and budget as whole rebuilding of project is cost ineffective where budget is crucial factor. Further they posted questions like Testing is done at all stages of software development life cycle for positive evaluations.

Q2) In case of marketing when a software is designed how can you assess the customers point of view for sales up and down such that there won't be any set backs in using deterministic cases?

A) Firstly, debate is on testing scenarios and not case studying more over what kind of software is there to Up and down the sales it's a case but not testing's question is wrong and our explanation to it is justified.

Q3) When there is inadequate or insufficient data how does deterministic testing works?

A) In case of inadequate data the initial assumptions might help in deducing certain results which are sufficient to be considered as deterministic input values for our final results or else our confidence level in assuming values might affect the standard derivation.

Based on example given on an MVC application we can use both probabilistic testing by random test generator which certainly occupies more space requirement for deterministic testing so that to directly select specified row and column in generating the results. Any probabilistic can be made near closet to deterministic by replacing random test generator with mock deterministic values.

Thus there are both advantages and disadvantage of using the deterministic testing in context to the consumer requirements and initial stages for development of product. Deterministic is right when the test cases and test runs should be the same and the probabilistic cases are right when the test cases and test runs need to be differed at every stage[1]. The cons team (our team) are right as the output values obtain are desirable and accurate, but for complex systems and networks the pros (opponents) are right as not easy to get exact results. The pros are wrong in considering vast examples without empirical data and proofs which are unbiased as there should be proof of explanation before posing questions why? Suppose if proof of explanation is there it can be easily referred and empirical methods used can help in understanding the deeper and the original context for greater debates, I support use of probabilistic or deterministic is up to designers and testers...even both for hybrid results. I conclude by saying why Deterministic testing is very important? It might be new picture and its growth of measurement in testing in various fields will increase and the deterministic testing is not any kind of proposal nor an idea to be proved, its already in existence and providing desirable results so, It's the future growth in sector of testing.

4) REFERENCES

- [1] Automated, deterministic testing versus stochastic testing using Quickcheck. .
- [2] M. Majeed, D. Ahlstrom, U. Ingelsson, G. Carlsson, and E. Larsson, "Efficient embedding of deterministic test data," in Test Symposium (ATS), 2010 19th IEEE Asian, 2010, pp. 159–162.
- [3] "Embedded Deterministic Test - CEDA." [Online]. Available: <http://ieee-ceda.org/activities/distinguished-speaker-series/janusz-rajski>. [Accessed: 19-Feb-2015].
- [4] R. David, A. Fuentes, and B. Courtois, "Random pattern testing versus deterministic testing of RAMs," IEEE Trans. Comput., vol. 38, no. 5, pp. 637–650, May 1989.
- [5] P. Thevenod-Fosse, H. Waeselynck, and Y. Crouzet, "An experimental study on software structural testing: deterministic versus random input generation," in Fault-Tolerant Computing, 1991. FTCS-21. Digest of Papers., Twenty-First International Symposium, 1991, pp. 410–417.

