Test cases reduction technique considering the time and cost as evaluation standards

ROOPALI SALWAN
A.S.E.T, AMITY UNIVERSITY
UTTAR PRADESH,INDIA
salwan_roopali@yahoo.co.in

RAJINI SEHGAL
A.S.E.T, AMITY UNIVERSITY
UTTAR PRADESH,INDIA
rsehgal@amity.edu

***Abstract*—Main aim of test cases reduction is to achieve the same reliability and same assurance in lesser number of test cases, we all know that testing walks hand in hand with development .We all know " WHY WE CREATE TEST CASES?". keep a track on whether the end product is achieved as per the requirements ,our aim is full filled ,the model which was formed theoretically on paper matches the actual end product and is performing the task as intended by SRS(software requirement specification). Initially testing was just considered small and unimportant part of development but now this is considered to be a myth as testing consume major part of cost and time .**

Thus reducing it can make a better product in term of its cost as certain industries considered time and cost as an important evaluation standard of a product/product under testing(put). Making test cases are considered an initial and most important step of testing thus it is the the root /whole and soul of our testing procedure whether it is primitive manual testing and newly accepted technique ie Automated testing which make use of new software tools help in analyzing ,running and executing test cases.

In this paper I would like to through light on the technique till now proposed and the best out of them which can really gives the better cost(in terms of money and time) for test suite reduction

***Index Terms*—Testing, Test cases ,Test suite, test cases reduction, manual testing, automated testing , product under test**

INTRODUCTION

Development of software ,because with changing time not only size but importance of software is increasing. Now days development is not considered as small job where small source code is written in any of the languages say C,C++ for printing your name on console or
reversing the number but it is , becoming tough and critical job .Thus as development is making its mark, limits of testing is also increasing ,software testing which is an efficient technique of locating the faults in the program. Software testing monitors the performance of the software by keeping an eye on efficiency ,requirement and functionality of the software. Few years back testing is only considered as small part of development to where testing is used to locate the error prone part in the source code thus locating the segment having error . Testing however not bounded to these limits software testing is locating error, identifying it, its analysis reporting it and then coming up with alternate solutions that can remove the error from the code and give the software as per the user requirement.

Testing occur repeatedly at every step of software Development life cycle , as with every change new segment of code is added and thus demands for testing increases, we know fundamental requirement of testing is to make test cases ,which is criteria of testing thus as testing domain increases number of test cases also increases which is a problematic task test cases number may ranges depending on size of the system .These varying size causes the huge loses in term of cost and time because running and executing such huge amount of test cases is the vigorous task not only effect the cost of the overall system but also the performance of tester is hampered due to boredom. Every now and then new techniques are evolved which support test suite reduction (in general leads to test cases minimization),some emphasise on test requirement contraction/optimization ,some give automatic test case generation based on IOTS (Input/output symbolic transition system),test suite reduction problem, which is similar to minimum set cover ,from NP-complete problem. Some time test reduction also leads to reduced fault localization i.e in order to reducing the size of test suite we ignore such test cases which are important for the fault localization, thus there should be reduced test cases thus new test cases is given from redundant test to reduced test ie returned test cases from the redundant test cases that maintain the evenness of distribution

or monitor the complete coverage of the system every path is covered thus assuring that the system is without fault .

Thus in this paper, I would like to analyse the technique that is better giving reduced test suite as well as giving the assurance of fault localization as well.

II BACKGROUND

A.TEST SUITE REDUCTION TECHNIQUES

Various test suite reduction techniques are given but no one fulfilled the basic need i.e reducing the test cases (reducing cost and time) and maintaining the fault localization effectiveness, every technique has its pros and cons if one reduces the test cases effectively than it lacks in even distribution or complete coverage of the system thus one or more segment is left

Untested any now and then, thus now we want a technique that reduced test cases as well as exposed the uncover error thus system is error free and efficient

A.1 TEST SUITE REDUCTION AND REQUIREMENT OPTIMIZATION

This technique was given to reduced the test cases so that lesser time and cost is given to the testing phase we all know that testing acquire 50%of total effort ,time and cost of development and reducing test cases can make this division less whether it is in term of money or time Given T test suite **where T={t_1+t_2+t_3+.....t_n}** and R is the testing **requirement where R={r_1+r_2+r_3.....r_m}** where r_i considered as subset of T (test suite),testing requirement is feasible if there is test case in input domain that satisfies testing requirement .Here relation between T,R and E is given by bigraph where T and E are the two disjoints and E is the edges joining T and EAnd r are the requirements. And E is set of edges connecting T and R In all technique testing requirement optimization is considered as pre-process of test suite generation and reduction .In general test requirement optimization considered in preference to test suite reduction.

t∈ T (∈ means element of T test suite thus t is element of T) It is defined as

Req(t)={ r∈ R|T ∈t } and test suite T' satisfies R if each testing requirement r in R there is one test case in T' satisfying r ie

Req(T')=R **and R=TS(R)** where **T**' is said to be reduced test suite of S if T' is subset of T such that T can satisfy R .Reduced test T' is said to be optimal one of S if for any reduced test suite T''

$|T'| \leq |T''|$

The optimal test suite may not be unique and they have same size

Test Suite Reduction problem to find an optimal test suite which is equivalent to minimum set cover. Here question arises what is minimum set cover problem is ,it is classical question in computer science and complexity theory which leads to Approximation Algorithm

EXPLANATION :

Suppose {123…m}set of elements forming universe i.e elements of universe and n sets whose union comprises the universe the set cover problem is to identify smallest number of sets whose union still contain all element in universe

Suppose C ={12345}

S = **{** {1,2,3},{2,4},{3,4},{4,5} **}**

Here Set Cover = { {123},{4,5}},Thus minimal element forming the universe here S is subset of C. a cover is subfamily .thus by using the MINIMUM SET COVER PROBLEM we can find the reduced test cases that form universe or in other words uncover all unexposed error

A .2 DYANAMIC DOMAIN REDUCTION

This technique focuses that software testing finds error in any software .Effective test is one in which probability of detecting is high i.e it is defining quality of effective test. Computer based system offer tester's with diversity of testing methods and enhance probability of detection various method considered in this scenario are

a. Path Testing –Aims to inspect validity of selected path without testing every path generally used when path under testing is so large where testing each path is next to impossible and is not feasible also.
b. Independent Program Path –In this at leas t one new processing path or any new condition is introduced while traversing or going through a program ex:program control is flown such that a new condition or statement is seen

However Literature Review of this technique gives many technique's like DDR, Ping Pong Technique and newly derived Coverall Technique

DDR EXPLANATTION: DDR based on CBT (Constraint Based Testing)which is control flow analysis it is evaluation of symbols and information about changes or mutants generates test data automatically to satisfy testing criteria .However it has certain drawbacks that is in problem handling loops, array and nested expression where DDR is given as it cover all these problem area which uses part of CBT approach as well as "Domain Reduction" method where source code is transferred to CFG

BASIC TERM USED IN DDR TECHNIQUE:

a. Basic Block- Maximum sequence of program statement there is one entry point and one exit point

b. Decision Point- Is point where control flow can diverge ex DO,WHILE,GOTO

c. Junction- point where control flow merges ex ENDIF,IF

d. Predicate-Boolean expression associated with decision node that determine which edge from the node will be traversed

e. Constraints- Algebraic expression that restricts the space of program variable to certain domain ex A > 0 which sshow A is positive However it uses Split Algorithm which focuses on Split point where split point used to reduce domains of variable ,it encodes our major heuristics for selecting test cases value and is key for our searching process. Search Point takes constraint and two expression with overlapping domain can be split so that constraint is TRUE for all values . In the dynamic domain reduction procedure, loops arehandled dynamically instead of finding all possible paths.The procedure exits the loop and continues traversing the path on the node after the loop. This eliminates the need for loop unrolling, which allows more realistic programs to be handled.

A .3 PING-PONG TECHNIQUE

This technique selects less number of test case by reordering test cases , based on heuristic technique which dosen't promise best solution but give good solution in appropriate time ,where each stage is mark such as initial stage and the stage in problem area and then it is seen final stage achieved is with real value by comparing the set of values of goal state and set of states of achieved values.Here minimal test case reduces the cost of testing also it runs test in different order with difference between the actual and succeeded ordering

Are from end to beginning .Different Procedures can be followed

a. Forward Procedure: suppose test case$<t_{1,}t_{2.....}t_{N>}$ than executing the test cases from starting t_1 to $t_{N.}$

b. Reverse Procedure: If test cases of size N is there than executing test cases in reverse order from t_N to t_1

c. Inside Out Procedure: Run test cases from middle to both the ends

A .4 COVERALL ALGORITHM

Coverall algorithm have certain constraints /rule to follow like every algorithm have its own domain

Firstly find all possible constraints (where constraints is algebraic expression that restricts the space of program variable to certain domain) from initial to final point it dictate condition of variables between start and finish nodes(<,>,=,≥,≠)

Secondly ,identifying and finding variable with maximum and minimum values in the path if any .Using condition dictated by constraint ,two variable one with maximum and other with minimum value can be identified then maximum variable would be set at the highest value within its range and minimum variable at the lowest value of its range A Constant value in the path is detected ,if any constant value for any variable is found ,value is assigned

3) Finding constant values in the path, if any. When constant values can be found for any variable in the path, the values would then be assigned to the given variables at each node.

Using the methodology, the Coverall algorithm would have the following characteristics:

1) Number of test cases. The number of test cases is smaller since each variable has a fixed value, either as maximum, minimum or constant values.

2) Automatic test cases generation. The test cases can be automatically generated with the reduction process.

3) Less time to test run. A single generation of test cases reduces the time of test run and compilation

Iv) TEST CASE REDUCTION TOOLS USED :

Above mention techniques generally offer techniques which can be referred during program implementation to validate the test data using test cases, however test cases can be minimized either reducing there domain ,or reducing requirement ,giving a fixed max –min value to the variable but whether there is an automatic tool that can support reduction of the test cases ,Answer is YES certain tools are presented which support test case reduction ,here are the study and comparison of certain of the tools used. Certain tools which are making there mark in the field of testing providing efficient way to minimizing the human effort of making test cases than ,analysing and choosing/selecting the best one and removing the undesirable one is tough task. However there is an account of some of these tools

A. BUGPOINT- Automatic test case reduction tool

Bugpoint Tool is designed in such a manner that it doesn't **effect the LLVM or hampering the LLVM (low level virtual machine is a infrastructure of compiler written in C/C++) co**nsiderably used for all **passes and code generators does not emphasise on the working i.e how is the system working ids not concern area. Bugpoint deals with one major drawback that is support gcc compiler only and one most important disadvantage is that it takes loads and loads of programmer time ,remove the**

undesired test cases and reduce the number of test cases but however time taken is considerably long Thus ruling out our second priority we want lesser number of test cases but in lesser time ,thus it hamper the resources like cost and space for the cost to give less test cases

B.LITHIUM-Automatic test case reduction tool Lithium that automatically reduces enormous amount of test cases saving time and cost in terms of (less manpower and space i.e memory), such as real-world web pages or test cases produced by jsfunfuzz (where jsfunfuzz is a fuzzer (fuzzer is a program using fuzzing technique, which may automated sometime or may be partially automated ,that gives invalid ,undesired and unexpected data and data anyhow is given randomly as input to computer program. .Then track is kept for the errors ,failure.). Javascript engine in firefox is tested ,reduced 100 line to 8 relatively faster than manually

IV) COMPARISION OF THESE TECHNIQUE TO EVALUATE WHICH YIELD BEST RESULT

S.No	TEST REDUCTION TECHNIQUE NAME	ADVANTAGE	DISADVANTAGE
1	**Requirement Optimization**	Boolean expression used to formalized the requirement ie used in term of true or false test cases are finite	It is local operation as the contraction of requirement optimization highlight the connection between two requirement and on the other hang global operation is always considered best
2	**D.D.R**	It succeeded in situation of problem handling array, loops and nested expression ex CBT not cover these problems	Tool build using DDR approach handles loops and array effectively but in programming language like C and C++ which uses pointers fail in handling pointers
3	**PING PONG TECHNIQUE**	Assure domain coverage anyhow as test case is traversed in every order forward to back ,mid to each starting point.	Time consuming technique aas well as expensive technique more memory ,time and efficiency is required in executing the test cases
4	**Coverall Algorithm**	Compilation of all the above technique states that coverall algorithm yield lessertime.Thus literature states that coverall isthe best	Disadvantage of Coverall algorithm lies in its requirement for identification of fix values for all variables, either as maximum,minimum or constant values.

V) CONCLUSION AND FUTURE WORK

This paper focuses on many test reduction technique used and given by researcher's,initially test reduction is considered as a area where requirement contraction or the functional expectation of the system is reduced critically but this technique leads to contraction of certain test cases where important requirement is filtered and thus efficient system is not achieved anyhow .Other technique's evaluated give's domain reduction where domain is decided or max-min value is decided and the variable can contain these value only but how can we assure by the other values of variable not taken suppose a system is giving correct value with integer but not working efficiently with float value thus this system also not ideally gives correct result as expected. Future work here focuses on enhancing these technique to yield the correct result as per the requirement mention in Software Requirement Specification and as per the requirement of the user ,generally the literature survey review reflects that coverall algorithm yield/gives best result but now some other algorithm are been focussed and extended further for future references ,because till now technique without any loop hole /defection is not achieved which gives 100%assurance of time ,cost and efficiency preservation as well as focuses on the /highlight the error that is fault is not ignored and is exposed fully .Certain tools like lithium and bug detector is also given but we can rely on such tool for our future because these tool need certain platform to support and are not working efficiently on all the system ex some support java while some work on C,C++.With advancement of technology we don't need to be limited to one or other language this technoworld support open source system and thus a system is required that

support every platform ,as this fast running world where data is not reserved to a laptop or personal computer through cloud computing data is moving from your PC'S to web

REFRENCES

[1] Preeyavis Pringsulaka and Jirapun Daengdej **"Coverall Algorithm for Test Case Reduction"** paper #1502, Final Version, Updated December 12, 2005

[2] Donghuo Chen，Xuandong Li，Shizhong Zhao "**Auto-generation and redundancy reduction of test cases for reactive systems"** 978-1-4244-8666-3/10/$26.00 C 2010 IEEE

[3] Xin Zhang, Qing GU,, Xiang Chen, Jingxian Qi, Daoxu Chen**A Study of "Relative Redundancy in Test-Suite Reduction While Retaining or Improving Fault-Localization"** *SAC'10*, March 22-26, 2010, Sierre, Switzerland. Copyright 2010 ACM 978-1-60558-638-0/10/03

[4] B. Beizer. **"Software Testing Techniques**." Van Nostrand Reinhold, 2nd edition, 1990.

[5] B. Korel, **"Automated Software Test Data Generation,"Conference on Software Engineering,** Vol 10, No. 8,Pages 870-879, August 1990.

[6] L. A. Clarke, "**A System to Generate Test Data and Symbolically Execute Programs**," IEEE Transactions on Software Engineering, Vol. SE-2, No. 3, pages 215-222, September 1976.

[7] L. J. Morell. "**A Theory of Error-Based Testing**," PhD thesis, University of Maryland, College Park MD, 1984, Technical Report TR-1395

[8] M. J. Gallagher and V. L. Narsimhan, "ADTEST: A Test Data Generation Suite for Ada Software Systems," IEEE Transactions on Software Engineering, Vol . 23, No. 8, pages 473-484, August 1997.

[9] Neelam Gupta, A. P. Mathur and M. L. Soffa, "**Automated Test Data Generation using An Iterative Relaxation Method**," ACM SIGSOFT Sixth International Symposium on Foundations of Software Engineering (FSE-6), pages 231-244, Orlando, Florida, November 1998.

[10] Offutt A. Jefferson, J. Pan and J. M. Voas**."Procedures for Reducing the Size of Coverage-based Test**

[11] H. Agrawal. **"Efficient coverage testing using global dominator graphs"**. In *Proc. of PASTE'99*, pages 11{20, 1999.

[12] J. Black, E. Melachrinoudis, and D. Kaeli." **Bi-criteria models for all-uses test suite reduction"**. In *Proc. of ICSE'04*, pages 106-115, 2004.