A Test Generation Algorithm for 3-Way Software Testing

Soumen Maity
Indian Institute of Science Education and Research
Sai Trinity Building, Garware Circle, Pashan, Pune 411 008, INDIA
Email: soumen@iiserpune.ac.in

Abstract

In the 21^{st} century our society is becoming more and more dependent on software systems. The safety of these systems and the quality of our lives is increasingly dependent on the quality of such systems. A key element in the manufacture and quality assurance process in software engineering is the testing of software systems. Empirical results show that 2-way or pair-wise testing is efficient for various types of software systems. But software failures may be caused by interactions of more than two parameters. In this paper, we propose recursive technique to generate test cases for 3-way testing when all parameters can take the same number of values. 3-way testing requires that for a given number of input parameters to the system, each possible combination of values for any three parameters be covered by at least one test case.

1 Introduction

Software testing is expensive and time consuming. Given the different input parameters with multiple possible values for each parameter, performing exhaustive testing which tests all possible combinations is practically impossible. Generating an optimal test set which will effectively test the software system is therefore desired. Pair-wise testing is known for its effectiveness in different types of software systems [1, 3]. But software failures may be caused by interactions of more than two parameters. A recent NIST study by Rick Kuhn indicates that failures can be triggered by interactions up to 6 parameters. Increased coverage leads to a higher level of confidence. But the number of test cases may increase rapidly as the degree of interactions increases. For example, assume that each parameter has 10 values. Then pairwise testing requires atleast 100 test cases, 3-way testing atleast 10^3 test cases, 4-way testing atleast 10^4 test cases. Thus test generation algorithms must be more sensitive in terms of both time and space requirements. To illustrate the concept of 3-way testing, consider a system with four two-valued parameters, i.e., parameter A has values A_0 and A_1, parameter B has values B_0 and B_1, parameter C has values C_0 and C_1, and parameter D has values D_0 and D_1. For parameters A, B, C, and D, (A_0, B_0, C_0, D_0), (A_0, B_0, C_1, D_1), (A_0, B_1, C_0, D_1), (A_0, B_1, C_1, D_0), (A_1, B_0, C_0, D_1), (A_1, B_0, C_1, D_0), (A_1, B_1, C_0, D_0), (A_1, B_1, C_1, D_1) is the only 3-way test set of size 8.

2 Main Results

A *covering array* $CA(N; t, k, s)$ is an $N \times k$ array on s symbols such that every $N \times t$ sub-array contains each ordered subset of size t from s symbols at least once, where t is the strength of the array. Covering arrays are used to generate software test cases to cover all t-sets of component interactions. For our purpose, N is the number of test cases, k the number of parameters, s the number of values of each parameter, t the degree of interaction. For pair-wise or 2-way testing $t = 2$ and for 3-way testing $t = 3$. An *orthogonal array* $OA(t, k, s)$ is an $s^t \times k$ array on s symbols such that every $s^t \times t$ sub-array contains each ordered subset of size t from s symbols exactly once. For example,

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

is a $CA(8; 3, 4, 2)$ as well as an $OA(3, 4, 2)$. Note that the rows of $CA(8; 3, 4, 2)$ have been considered as the test cases for 4 2-valued parameters in the previous illustration of 3-way testing. In practice, a common situation is systems with large number of parameters, each of which has only few possible values. Thus, construction of test cases for an arbitrary number n of s-valued parameters for 3-way testing is an interesting combinatorial open problem.

A recursive construction for a covering array is a method for constructing a covering array from one or more covering arrays with smaller parameter sets. We will now describe one recursive construction that shows how to square the number of columns in 3- wise covering arrays. Let A be a 3-wise covering array with k columns, N rows, s levels, and let A^i be the i-th column of A. Let $B = B[i, j]$ be an orthogonal array of strength 2 with $T(k)+1$ columns and symbols from $\{1, 2, \ldots, k\}$, where $T(k) - 1$ is the number of *mutually orthogonal Latin squares* of size k. If k is prime or prime power then $T(k) = k$. The full table of mutually orthogonal Latin squares appears in [2]. We will construct a block array C with k^2 columns and $T(k) + 1$ rows. Each element in C will be a column of A. Let $A^{B[j,i]}$ be the block in the i-th row and j-th column of C.

Now we will illustrate how to construct recursively a 3-wise covering array with 16 columns from a 3-wise covering array with 4 columns. Consider the above mentioned $CA(8; 3, 4, 2)$. Let A^i be the i-th column of A, $i = 1, 2, 3, 4$. Take an orthogonal array $B = B[i, j]$ of strength 2 with 5 columns, 16 rows, and entries from $\{1, 2, 3, 4\}$, i.e., $B^t =$

$$\begin{pmatrix} 1\,2\,3\,4\,2\,1\,4\,3\,3\,4\,1\,2\,4\,3\,2\,1 \\ 1\,2\,3\,4\,3\,4\,1\,2\,4\,3\,2\,1\,2\,1\,4\,3 \\ 1\,2\,3\,4\,4\,3\,2\,1\,2\,1\,4\,3\,3\,4\,1\,2 \\ 1\,2\,3\,4\,1\,2\,3\,4\,1\,2\,3\,4\,1\,2\,3\,4 \\ 1\,1\,1\,1\,2\,2\,2\,2\,3\,3\,3\,3\,4\,4\,4\,4 \end{pmatrix}$$

where B^t denotes the transpose of B. Then $C = C[i, j] = A^{B[j,i]} =$

$$\begin{pmatrix} A^1 & A^2 & A^3 & A^4 & A^2 & A^1 & A^4 & A^3 & A^3 & A^4 & A^1 & A^2 & A^4 & A^3 & A^2 & A^1 \\ A^1 & A^2 & A^3 & A^4 & A^3 & A^4 & A^1 & A^2 & A^4 & A^3 & A^2 & A^1 & A^2 & A^1 & A^4 & A^3 \\ A^1 & A^2 & A^3 & A^4 & A^4 & A^3 & A^2 & A^1 & A^2 & A^1 & A^4 & A^3 & A^3 & A^4 & A^1 & A^2 \\ A^1 & A^2 & A^3 & A^4 & A^1 & A^2 & A^3 & A^4 & A^1 & A^2 & A^3 & A^4 & A^1 & A^2 & A^3 & A^4 \\ A^1 & A^1 & A^1 & A^1 & A^2 & A^2 & A^2 & A^2 & A^3 & A^3 & A^3 & A^3 & A^4 & A^4 & A^4 & A^4 \end{pmatrix}$$

is a 3-wise covering array with 16 columns and 40 rows; that is, C is a $CA(40; 3, 16, 2)$. Thus the number of test cases in 3-wise covering test suit with k^2 columns is no more than $T(k) + 1$ times the the number of test cases in 3-wise covering test suit with k columns.

Consider T, an arbitrary 3-tuple of members of $\{1, 2, \ldots, s\}$. Let $\bar{C}$ be a sub-matrix of C induced by an arbitrary choice of 3 columns. It can be shown that T is a row of $\bar{C}$. The columns of $\bar{C}$ correspond to 3 columns of B^t. Let $\bar{B}^t$ be the sub-matrix of B^t induced by those 3 columns. We can always find a row in $\bar{B}^t$ with distinct values. This would guarantee that T is a row in $\bar{C}$ using the properties of the base array A.

Algorithm TestGenerator generates test cases for 3-way testing. Inputs to TestGenerator are the number of parameters n and the value s for each parameter and outputs are test cases. It may be noted that the algorithm makes use of covering array of small parameters from Charlie Colbourn's CATables: http://www.public.asu.edu/ ccolbou/src/tabby/catable.html

Algorithm: TestGenerator
Input: The number of parameters n and the value s.
Output: Test cases for 3-way testing.
1. Find the smallest integer k greater than or equal to $\sqrt{n}$
2. Take one 3-wise covering array $CA(N; 3, k, s)$ from Charlie Colbourn's CATables.
3. Construct an orthogonal array $OA(2, T(k) + 1, k)$ of strength two.
4. Construct a 3-wise covering array $C = CA(N; 3, k^2, s)$ using the technique described before. Let $\bar{C}$ be a sub-matrix of C induced by an arbitrary choice of n columns. The rows of $\bar{C}$ are the test cases for n s-valued parameters.

3 Conclusions

In this paper, we have proposed a recursive test generation strategy for 3-way testing. As an illustration, consider the case of generating test cases for 24 2-valued parameters. Here $n = 24$ and hence $k = 5$. We make use of a covering array $CA(10; 3, 5, 2)$ and an orthogonal array $OA(2, 6, 5)$. The following table shows the number of test cases obtained by TestGenerator.

Systems for 3-way testing	No. of test cases
16 2-valued parameters	40
17-25 2-valued parameters	60
121 2-valued parameters	72

References

[1] Cohen, D. M., Dalal, S. R., Fredman, M. L., and Patton, G. C., "The AETG system: An Approach to Testing Based on Combinatorial Design", *IEEE Transaction on Software Engineering*, 23 (7), 1997, pp. 437-443.

[2] Colbourn, C. and Dinitz, J. (Ed.) *The CRC Handbook of Combinatorial Designs*, CRC Press, 1996.

[3] Hartman, A., "Software and hardware testing using combinatorial covering suites", in *"Graph Theory, Combinatorics and Algorithms: Interdisciplinary Applications"*, Kluwer Academic Publishers, 34, 2006, pp. 237-266.