**SSW 567 - Assignment 6**

**Group 5**

Miguel Camacho

Ed Chang

Harmony Sullivan

**Assignment Description**

OATS Testing

Read CopelandCh. 6 Pairwise Testing. Do Practice 1.Document your results.

1. Neither the Brown & Donaldson nor the Stateless University Registration System case studies contain huge numbers of combinations suitable for the pairwise testing approach. As exercises, use the orthogonal array and/or all pairs technique on the other two examples in this chapter. Determine the set of pairwise test cases using the chosen technique.
2. A bank has created a new data processing system that is ready for testing. This bank has different kinds of customers - consumers, very important consumers, businesses and non-profits; different kinds of accounts - checking, savings, mortgages, consumer loans and commercial loans; they operate in different states, each with different regulations - California, Nevada, Utah, Idaho, Arizona and New Mexico.
3. In an object-oriented system, an object of class A can send a message containing a parameter P to an object of class X. Classes B, C and D inherit from A so they too can send the message. Classes Q, R, S and T inherit from P so they too can be passed as the parameter. Classes Y and Z inherit from X so they too can receive the message.

**Results**

We have chosen to use both approaches (Orthogonal Arrays and Allpairs Algorithm) to determine the resulting test cases.

1a. Bank Data Processing System

Orthogonal Arrays:

1. Identify variables:

* kinds of customers
* different kinds of accounts
* different states

2. Determine # of choices for each variable:

* kinds of customers - consumers, very important consumers, businesses, non-profts (4 choices)
* different kinds of accounts - checking, savings, mortgages, consumer loans, commercial loans (5 choices)
* different states - California, Nevada, Utah, Idaho, Arizona, New Mexico (6 choices)

I have 3 factors: 1 4-level factor, 1 5-level factor and 1 6-level factor. This gives 120 combinations for complete testing. The lower bound is 6\*5 = 30.

3.Locate an orthogonal array which has a column for each variable and values within the columns that correspond to the choices for each variable.

In this case, we have located the L36 (28 63) which covers (41 51 61).

This array is available at [http://neilsloane.com/oadir/MA.36.6.3.2.8.txt]

4. Map the test problem onto the orthogonal array.

In this case, we only need the 63 columns, the 28 columns can be deleted.

|  |  |  |
| --- | --- | --- |
| 1 | 0 | 0 |
| 4 | 1 | 1 |
| 2 | 0 | 1 |
| 3 | 1 | 0 |
| 1 | 2 | 4 |
| 2 | 3 | 5 |
| 4 | 2 | 5 |
| 3 | 3 | 4 |
| 3 | 4 | 2 |
| 2 | 5 | 3 |
| 4 | 4 | 3 |
| 1 | 5 | 2 |
| 2 | 2 | 2 |
| 5 | 3 | 3 |
| 0 | 2 | 3 |
| 4 | 3 | 2 |
| 2 | 4 | 0 |
| 0 | 5 | 1 |
| 5 | 4 | 1 |
| 4 | 5 | 0 |
| 4 | 0 | 4 |
| 0 | 1 | 5 |
| 5 | 0 | 5 |
| 2 | 1 | 4 |
| 0 | 4 | 4 |
| 3 | 5 | 5 |
| 1 | 4 | 5 |
| 5 | 5 | 4 |
| 0 | 0 | 2 |
| 1 | 1 | 3 |
| 3 | 0 | 3 |
| 5 | 1 | 2 |
| 5 | 2 | 0 |
| 1 | 3 | 1 |
| 3 | 2 | 1 |
| 0 | 3 | 0 |

Column 1 – Customers (0=consumers, 1=very important consumers, 2=businesses, 3=non-profts, 4-5 don’t matter)

Column 2 – Accounts (0=checking, 1=savings, 2=mortgages, 3=consumer loans, 4=commercial loans, 5 doesn’t matter)

Column 3 – States (0=California, 1=Nevada, 2=Utah, 3=Idaho, 4=Arizona, 5=New Mexico)

|  |  |  |  |
| --- | --- | --- | --- |
| Test Cases (Runs) / Factors (variables) | Customers | Accounts | States |
| 1 | Very important consumers | Checking | California |
| 2 | ~consumers | Savings | Nevada |
| 3 | businesses | Checking | Nevada |
| 4 | Non-profits | Savings | California |
| 5 | Very important consumers | Mortgages | Arizona |
| 6 | Businesses | Consumer loans | New Mexico |
| 7 | ~Very important consumers | Mortgages | New Mexico |
| 8 | Non-profits | Consumer loans | Arizona |
| 9 | Non-profits | Commercial loans | Utah |
| 10 | Businesses | ~checking | Idaho |
| 11 | ~Businesses | Commercial loans | Idaho |
| 12 | Very important consumers | ~savings | Utah |
| 13 | Businesses | Mortgages | Utah |
| 14 | ~Non-profits | Consumer loans | Idaho |
| 15 | Consumers | Mortgages | Idaho |
| 16 | ~consumers | Consumer loans | Utah |
| 17 | Businesses | Commercial loans | California |
| 18 | Consumers | ~mortgages | Nevada |
| 19 | ~Very important consumers | Commercial loans | Nevada |
| 20 | ~Businesses | ~consumer loans | California |
| 21 | ~Non-profits | Checking | Arizona |
| 22 | Consumers | Savings | New Mexico |
| 23 | ~Consumers | Checking | New Mexico |
| 24 | Businesses | Savings | Arizona |
| 25 | Consumers | Commercial loans | Arizona |
| 26 | Non-profits | ~commercial loans | New Mexico |
| 27 | Very important consumers | Commercial loans | New Mexico |
| 28 | ~Very important consumers | ~checking | Arizona |
| 29 | Consumers | Checking | Utah |
| 30 | Very important consumers | Savings | Idaho |
| 31 | Non-profits | Checking | Idaho |
| 32 | ~Businesses | Savings | Utah |
| 33 | ~Non-profits | Mortgages | California |
| 34 | Very important consumers | Consumer loans | Nevada |
| 35 | Non-profits | Mortgages | Nevada |
| 36 | Consumers | Consumer loans | California |

5. Construct the test cases

Test cases would be constructed by applying the rules about each input combination to the runs above and including an expected result for each.

2nd approach: Utilize all pairs algorithm:

## Using ‘ALLPAIRS Test Case Generation Tool (Version 1.2.1)’

[http://www.satisfice.com/tools.shtml]

L30(415161)

That is: 30 rows, 3 columns - 1 column with 4 choices, 1 column with 5 choices, 1 column with 6 choices

Input file:

|  |  |  |
| --- | --- | --- |
| cust | acct | state |
| cons | check | Cal |
| VIcons | sav | Nev |
| bus | mort | Utah |
| non-p | cloan | Idaho |
|  | comloan | Ariz |
|  |  | NewMex |

Resulting test cases:

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASES | |  |  |
| case | Customer | Account | State |
| 1 | Consumer | checking | California |
| 2 | very important consumer | savings | California |
| 3 | Businesses | mortgage | California |
| 4 | non-profits | consumer loans | California |
| 5 | very important consumer | checking | Nevada |
| 6 | Consumer | savings | Nevada |
| 7 | non-profits | mortgage | Nevada |
| 8 | Businesses | consumer loans | Nevada |
| 9 | Businesses | checking | Utah |
| 10 | non-profits | savings | Utah |
| 11 | Consumers | mortgage | Utah |
| 12 | very important consumers | consumer loans | Utah |
| 13 | non-profits | checking | Idaho |
| 14 | Businesses | savings | Idaho |
| 15 | very important consumers | mortgage | Idaho |
| 16 | Consumers | consumer loans | Idaho |
| 17 | Consumers | commercial loans | Arizona |
| 18 | very important consumers | checking | Arizona |
| 19 | Businesses | savings | Arizona |
| 20 | non-profits | mortgage | Arizona |
| 21 | very important consumers | commercial loans | New Mexico |
| 22 | Consumers | checking | New Mexico |
| 23 | non-profits | savings | New Mexico |
| 24 | Businesses | mortgage | New Mexico |
| 25 | Businesses | commercial loans | California |
| 26 | non-profits | commercial loans | Nevada |
| 27 | ~consumers | commercial loans | Utah |
| 28 | ~very important consumers | commercial loans | Idaho |
| 29 | ~consumers | consumer loans | Arizona |
| 30 | ~very important consumers | consumer loans | New Mexico |

Please note that the values marked with ‘~’ do not matter, because all of their pairings have already been selected.

1b. Object Oriented System

Orthogonal Arrays

1 Identify variables

1. senders
2. parameters
3. receivers

2 Determine number of choices for each variable

senders: A, B, C, D (4 choices)

parameters: P, Q, R, S, T (5 choices)

receivers: Y, Z, X (3 choices)

I have 3 factors: 1 4-level factor, 1 5-level factor and 1 3-level factor. This gives 60 combinations for complete testing.

The lower bound is 4\*5 = 20.

3.Locate an orthogonal array which has a column for each variable and values within the columns that correspond to the choices for each variable.

In this case, we have located the L32 (215 43 81) which covers (31 41 51).

The full array is not available in any array directory we could find, such as Neil Sloane. If we were to use L32, 215 and 41 columns could be deleted.

4.Map the test problem onto the orthogonal array. We cannot map this since we do not have the full array with values.

5. Construct the test cases

Test cases would be constructed by applying the rules about each input combination to the runs and including an expected result for each.

2nd approach: Utilize all pairs algorithm:

## Using ‘ALLPAIRS Test Case Generation Tool (Version 1.2.1)’

[<http://www.satisfice.com/tools.shtml>]

Input file:

|  |  |  |
| --- | --- | --- |
| sender | param | receive |
| A | P | X |
| B | Q | Y |
| C | R | Z |
| D | S |  |
|  | T |  |

L20(314151)

That is: 20 rows, 3 columns - 1 column with 4 choices, 1 column with 5 choices, 1 column with 3 choices

Resulting test cases:

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASES | |  |  |
| case | sender | parameter | receiver |
| 1 | A | P | X |
| 2 | B | P | Y |
| 3 | C | P | Z |
| 4 | A | Q | Y |
| 5 | B | Q | X |
| 6 | C | Q | X |
| 7 | D | Q | Z |
| 8 | A | R | Z |
| 9 | B | R | X |
| 10 | C | R | Y |
| 11 | B | S | Z |
| 12 | D | S | X |
| 13 | A | S | Y |
| 14 | D | T | Y |
| 15 | A | T | X |
| 16 | B | T | Z |
| 17 | D | P | ~X |
| 18 | D | R | ~Y |
| 19 | C | S | ~X |
| 20 | C | T | ~Y |

Please note that the values marked with ‘~’ do not matter, because all of their pairings have already been selected.

**Lessons Learned**

The all pairs algorithm seems like a good choice for finding the minimum set of pairwise test cases. Without access to this algorithm, the standard set of orthogonal arrays fits as best it can to the data for each problem. There are only a few documented studies that prove the usefulness of orthogonal arrays. This makes it an interesting choice for organizations to use. But as the text states, the results are better than choosing a random set of test cases from the total set.

**Honor Pledge**

We pledge on our honor that we have not given or received any unauthorized assistance on this assignment/examination. We further pledge that we have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.