Software Verification & Validation

Natthapong Jungteerapanich

Handout 3

Covering Arrays

Acknowledgement

- **http://math.nist.gov/coveringarrays/coveringarray.html**

- **Software and Hardware Testing Using Combinatorial Covering Suites**, **A. Hartman**, a chapter in "Graph Theory, Combinatorics and Algorithms: Interdisciplinary Applications", published by Kluwer Academic Publishers.

- **Lee Copeland's** "**A Practitioner's Guide to Software Test Design**". Artech House, 2004.

Orthogonal Arrays

- A **(t,v,k,λ)-orthogonal array** ($t \le k$) is a $\lambda v^t \times k$ array whose entries are chosen from a set X with v points such that in every subset of t columns of the array, every t-tuple of points of X appears in exactly λ rows.
- In many applications these parameters are given the following names:
  - v is the number of **levels**,
  - k is the number of **columns** or **factors**,
  - λv^t is the number of **rows** or **runs**,
  - t is the **strength**, and
  - λ is the **index**.
- An orthogonal array is **simple** if it does not contain any repeated rows.

From https://en.wikipedia.org/wiki/Orthogonal_array (Retrieved 2016/2/9)

Orthogonal Arrays

- A (2, 2, 3, 1)-orthogonal array

1	1	1
2	2	1
1	2	2
2	1	2

- A (2, 4, 5, 1)-orthogonal array

1	1	1	1	1
1	2	2	2	2
1	3	3	3	3
1	4	4	4	4
2	1	4	2	3
2	2	3	1	4
2	3	2	4	1
2	4	1	3	2
3	1	2	3	4
3	2	1	4	3
3	3	4	1	2
3	4	3	2	1
4	1	3	4	2
4	2	4	3	1
4	3	1	2	4
4	4	2	1	3

From https://en.wikipedia.org/wiki/Orthogonal_array (Retrieved 2016/2/9)

Orthogonal Arrays

- A (2, 3, 5, 3)-orthogonal array (transposed for easy viewing)

0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2
0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
0	0	0	1	1	1	2	2	2	2	2	2	0	0	0	1	1	1	1	1	1	2	2	2	0	0	0
0	1	2	1	2	0	2	0	1	0	1	2	1	2	0	2	0	1	0	1	2	1	2	0	2	0	1

From https://en.wikipedia.org/wiki/Orthogonal_array (Retrieved 2016/2/9)

Covering Arrays

- A **(t,v,k)-covering array** ($t \le k$) is an array with k columns whose entries are chosen from a set X with v points such that in every subset of t columns of the array, every t-tuple of points of X appears at least once.

- For each triple (t, v, k), there can be many possible (t,v,k)-covering arrays with different numbers of rows.

- **CAN(t,k,v)** denotes the number of rows in a (t,v,k)-covering array with fewest rows.

- See http://www.public.asu.edu/~ccolbou/src/tabby/catable.html for the best known CAN(t,k,v).

Covering Arrays

A (2, 2, 4)-covering array

0 0 0 0
0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0

A (3, 2, 4)-covering array

0 0 0 1
0 0 1 0
0 1 0 0
0 1 1 1
1 0 0 0
1 0 1 0
1 1 0 1
1 1 1 0
1 0 1 1

Are these covering arrays optimal (i.e. smallest)? If not, can you find smaller ones?

Covering Arrays

- The IPOG-F algorithm can be used to construct a “small” covering array for a given triple (t,v,k) efficiently. However, the covering arrays constructed may not be optimal.

- See “**Refining the In-Parameter-Order Strategy for Constructing Covering Arrays**'' by **M. Forbes, J. Lawrence, Y. Lei, R. N. Kacker and D.R. Kuhn**.

- The following website lists a number covering arrays computed using the IPOG-F algorithm:
  - http://nvlpubs.nist.gov/nistpubs/jres/113/5/V113.N05.A04.pdf