University of Edinburgh, School of Informatics

Informatics 3: Software Testing: Tutorial 5

Data Flow Testing

Consider the following program:

```java
static int find (int list[], int n, int key)
{
        // binary search of ordered list
        int lo = 0;
        int hi = n - 1;
        int result = -1;
        while ((hi >= lo) && (result == -1)) {
                final int mid = (lo + hi) / 2;
                if (list[mid] == key)
                        result = mid;
                else if (list[mid] > key)
                        hi = mid - 1;
                else // list[mid] < key
                        lo = mid + 1;
        }
        return result;
}
```

This is *not* a particularly good example of programming but it is useful for the purposes of this tutorial.

Prerequisites

Review the material on Data-Flow based testing in Lectures 7 and 8 and the paper by Frankl and Weyuker.

Preparation

Review the code above; please try to ensure you understand the method and the particular implementation. It is an implementation of binary search of an ordered array.

Activities

1. (10 Minutes) First individually construct the flow graph corresponding to this program.
2. (5 Minutes) Find a partner to work with in the group and check that you agree on the structure of the flow-graph for the program.
3. (10 Minutes) For each block calculate the *defs* and *c-use* sets and for each condition calculate the *p-use* sets for each out arc.
4. (5 Minutes) Check with your partner that you agree on the sets for your flowgraph.
5. (10 Minutes) Calculate paths that satisfy the following coverage criteria:
    a. All uses.
    b. All DU paths.

6. (10 Minutes) Work with your partner to devise test sets that explore the sets of paths you devised in the previous activity (modify the paths if they are infeasible).
7. If time is available, as a whole class devise a test set that satisfies the "all uses" criterion but fails the "all DU paths" criterion.
8. If time is available discuss in the whole group what a best test set for each of the criteria might look like.