Data Flow Testing

Contents

[Data Flow Testing 2](#_Toc443476448)

[Steps in Data Flow Testing 2](#_Toc443476449)

[The Data Flow Graph 2](#_Toc443476450)

[Data Flow Testing Criteria 2](#_Toc443476451)

[All-defs 2](#_Toc443476452)

[All-c-uses 3](#_Toc443476453)

[All-p-uses 3](#_Toc443476454)

[All-p-uses/Some-c-uses 3](#_Toc443476455)

[All-c-uses/Some-p-uses 3](#_Toc443476456)

[All-uses 3](#_Toc443476457)

[All-du-paths 3](#_Toc443476458)

[Def() and c-use() Sets of Nodes 4](#_Toc443476459)

[Predicates and p-use() Set of Edges 4](#_Toc443476460)

[All-defs on **tv** 4](#_Toc443476461)

[All-c-uses on **ti** 5](#_Toc443476462)

[All-p-uses on **tv** 5](#_Toc443476463)

[All-p-uses/Some-c-uses on **i** 6](#_Toc443476464)

[All-c-uses/Some-p-use on **AS** 6](#_Toc443476465)

# Data Flow Testing

## Steps in Data Flow Testing

1. Draw a data flow graph (DFG) from the SUT
2. Select one or more data flow testing criteria
3. Identify paths in the data flow graph satisfying the selection criteria
4. Derive path predicate expressions from the selected paths and solve the expressions to derive test data

## The Data Flow Graph

Nodes specify a sequence of ***definitions*** and ***c-uses***. A ***definition*** of a variable occurs when it is assigned a value. A ***c-use*** of a variable occurs when the variable is used in a computation. A variable becomes ***undefined*** or ***killed*** when it becomes deallocated.

Edges specify a set of ***p-uses***. A ***p-use*** of a variable occurs when the variable is used in a predicate.

|  |
| --- |
|  |

## Data Flow Testing Criteria

### All-defs

For each variable and for each node, such that has a ***global definition*** in node, select a ***complete path*** which includes a ***def-clear*** path from node to:

* Node having a ***global c-use*** of or
* Edge having a p-use of

***Global Definition***: A node has a global definition of a variable if the variable is defined in that node and there is a ***def-clear*** path of that variable to another node or edge where the variable is used.

***Complete Path***: A path from the entry node to the exit node.

***Def-clear path***: A path is a def-clear path if there is no definition or undefinition of the given variable.

***Global c-use***: A variable has global c-use in a node if it has been defined in an earlier node.

### All-c-uses

For each variable and for each node, such that has a global definition in node, select complete paths which include def-clear paths from node to all nodes such that there is a global c-use of in.

### All-p-uses

For each variable and for each node such that has a global definition in node, select complete paths which include def-clear paths from node to all edges such that there is a p-use of on edge.

### All-p-uses/Some-c-uses

Identical to all-p-uses except if has no p-use, then this reduces to ***some-c-uses***.

***Some-c-uses***: For each variable and for each node such that has a global definition in node, select complete paths which include def-clear paths from node to *some* nodes such that there is a global c-use of in node.

### All-c-uses/Some-p-uses

Identical to all-c-uses except if has no global c-use, then this reduces to ***some-p-uses***.

***Some-p-uses***: For each variable and for each node i such that has a global definition in node, select complete paths which include def-clear paths from node to some edges such that there is a p-use of on edge .

### All-uses

The conjunction of off all-p-uses and all c-uses.

### All-du-paths

For each variable and each node such that has a global definition in node, select complete paths which include all ***du-paths*** from node:

* To all nodes such that there is a global c-use of in and
* To all edges such that there is a p-use of on

***Du-paths***: A path is a definition-use path w.r.t. variable if the first node has a global definition of and either:

* The last node has a global c-use of and the path is a def-clear ***simple path*** w.r.t. or
* The last edge has a p-use of and the path is a def-clear, loop-free path w.r.t..

***Simple path***: A path in which all nodes, expect possibly the first and last node are distinct.

***Loop-Free path***: A path in which all nodes are distinct.

### Def() and c-use() Sets of Nodes

|  |  |  |
| --- | --- | --- |
| **Node i** | **def(i)** | **c-use(i)** |
| **1** | {values, AS, MIN, MAX} | {} |
| **2** | {ti, i, tv, sum} | {} |
| **3** | {} | {} |
| **4** | {ti} | {ti} |
| **5** | {tv, sum} | {tv, i, sum, value} |
| **6** | {i} | {i} |
| **7** | {average} | {} |
| **8** | {average} | {sum, tv} |
| **9** | {average} | {} |
| **10** | {} | {average} |

### Predicates and p-use() Set of Edges

|  |  |  |
| --- | --- | --- |
| **Edges (i, j)** | **predicate(i, j)** | **p-use(i, j)** |
| **(1, 2)** | True | {} |
| **(2, 3)** | True | {} |
| **(3, 4)** | ((ti < AS) && (values[i] != -999)) | {ti, AS, values} |
| **(4, 5)** | ((values[i] >= MIN) && (values[i] <= MAX)) | {values, MIN, MAX} |
| **(4, 6)** | ~((values[i] >= MIN) && (values[i] <= MAX)) | {values, MIN, MAX} |
| **(5, 6)** | True | {} |
| **(6, 3)** | True | {} |
| **(3, 7)** | ~((ti < AS) && (values[i] != -999)) | {ti, AS, values} |
| **(7, 8)** | ~(tv > 0) | {tv} |
| **(7, 9)** | (tv > 0) | {tv} |
| **(8, 10)** | True | {} |
| **(9, 10)** | True | {} |

### All-defs on **tv**

**Global definitions: 2, 5**

**Global c-use: 5**

**Def-clear path: 2-3-4-5**

**Complete path: 1-2-3-4-5-6-3-7-9-10**

**P-uses: (7, 8)**

**Def-clear path: 2-3-7-8**

**Complete path: 1-2-3-7-8-10**

**Global c-use: 8**

**Def-clear path: 5-6-3-7-8**

**Complete path: 1-2-3-4-5-6-3-7-8-10**

### All-c-uses on **ti**

**Global definitions: 2, 4**

**Node 2: Global c-use: 4**

**Node 4: There is no c-use that corresponds**

**Def-clear path: 2-3-4**

**Complete paths:**

* **1-2-3-4-5-6-3-7-8-10**
* **1-2-3-4-6-3-7-8-10**
* **1-2-3-4-5-6-3-7-9-10**
* **1-2-3-4-6-3-7-8-10**

### All-p-uses on **tv**

**Global definitions: 2, 5**

**Node 2: p-uses: (7, 8) and (7, 9)**

**Def-clear paths: 2-3-7-8, 2-3-7-9**

**Node 5: p-uses: (7, 8) and (7, 9)**

**Def-clear paths: 5-6-3-7-8, 5-6-3-7-9**

**Complete paths:**

* **1-2-3-7-8-10**
* **1-2-3-7-9-10**
* **1-2-3-4-5-6-3-7-8-10**
* **1-2-3-4-5-6-3-7-9-10**

### All-p-uses/Some-c-uses on **i**

**Global definitions: 2, 6**

**No p-use of i 🡪 c-use of i**

**Node 2: c-use in 6**

**Def-clear path: 2-3-4-6**

**Complete path: 1-2-3-4-6-3-7-8-10**

### All-c-uses/Some-p-use on **AS**

**Global definition: 1**

**No global c-use of AS 🡪 p-use of AS**

**Edges: (3, 4) and (3, 7)**

**Def-clear paths: 1-2-3-4, 1-2-3-7**

**Complete path: 1-2-3-4-5-6-3-7-8-10**