Software Testing

Sibylle Schupp[1]

[1]Institute for Software Systems/Institut für Softwaresysteme
Hamburg University of Technology (TUHH)

Spring 2022

Lecture 4

Outline

Outline

Introduction

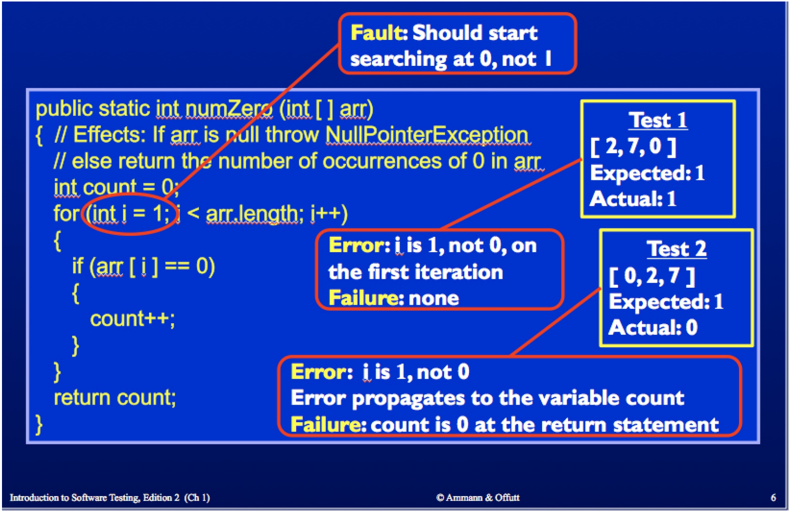
- Recall the differences between faults, failure, and error
  - Fault: defect. Failure: missed requirements. Error: internal manifestation of a fault. Thus, failure $\Rightarrow$ error $\Rightarrow$ fault
- Recall the RIPR model for faults and failures:
  - Reachability, infection, propagation, revealability
- Recall the notion of coverage:

  A coverage criterion is a rule or collection of rules that impose test requirements on a test set (AO, Ch.5)

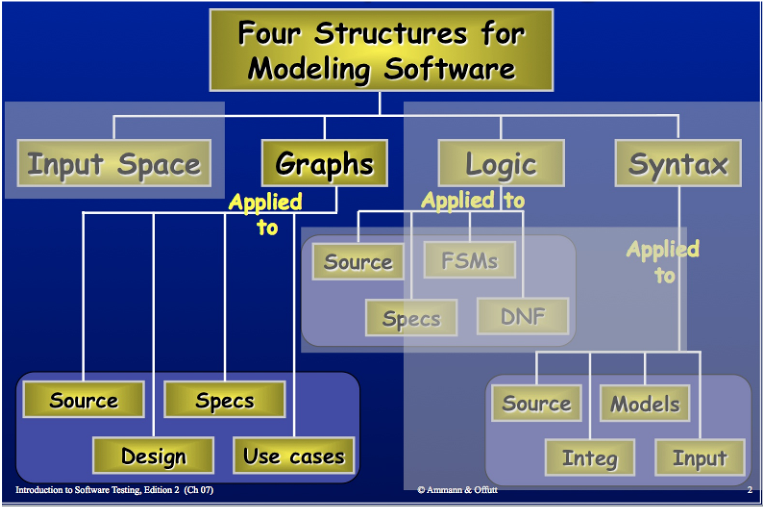
- Subsumption

  A coverage criterion C_1 subsumes C_2 iff every test set that satisfies C_1 also satisfies C_2.

Recall the example



Graph coverage



Graph coverage for testing

- Graphs are the most common structure for testing.
- Graphs can be extracted from many sources.
  - Code (or directly: control-flow graph)
  - Finite state machines, statecharts
  - Use cases
  - Module hierarchies
- Tests cover the graph in various ways.

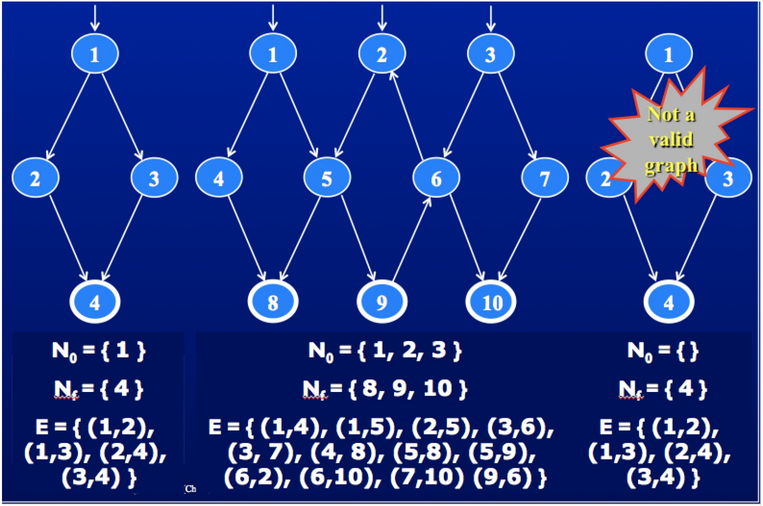
What is a graph?

Definition

A graph $G = (N, E, N_o, N_f)$ is a non-empty set of nodes, N, a set E of edges between pairs of nodes, a non-empty set of initial nodes, N_o, and a set N_f of final nodes.

- For an edge (n_i, n_j), n_i is called the predecessor of n_j and n_j is called the successor of n_i.

Examples (graphs)



Paths

Definition

- A path in a graph G is a sequence of nodes, $[n_1, n_2, \ldots, n_m]$, such that each adjacent pair of nodes is an edge in G.
- The length of a path is the number of its edges.
- A subpath of a path p is a path that is a subsequence of the nodes in p.

Ex (see previous slide):

- [1,4,8]; [2,5,9,6,2]; [3,7,10]

Test paths and SESE graphs

Definition

A test path is a path that starts at an initial node and ends at a final node.

- Test paths represent the execution of test cases.
  - Some test paths can be executed by many tests.
  - Some test paths cannot be executed by any tests.
- A graph with a single initial node and a single final node is called single entry/singly exit (SESE) graph. Test paths in SESE graphs all start and end in the same node.

Visits and tours

Definition

A test path p visits node n if n is a node in p. A test path p visits edge e if e is an edge in p. A test path p tours subpath q if q is a subpath in p.

Example

- Test path [1,2,4,5,7]
- Visits nodes {1,2,4,5,7}
- Visits edges {(1,2),(2,4),(4,5),(5,7)}
- Tours subpaths {[1,2,4],[2,4,5],[4,5,7],[1,2,4,5],[2,4,5,7],[1,2,4,5,7]}

Tests and test paths

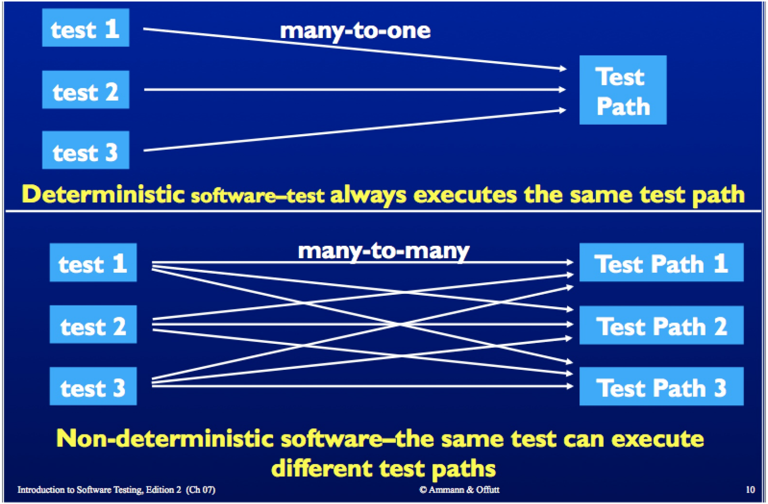
- Each test executes exactly one test path.
  - The test path executed by test t is denoted by path(t).
  - The set of test paths executed by the set of tests T is denoted by path(T).
- Infeasible test paths often result from locations that are unreachable. Formally:

Definition

A location in a graph (node or edge) can be reached from another location if there is a sequence of edges from the first location to the second.

  - Syntactic reach: a subpath exists in a graph
  - Semantic reach: a test exists that can execute that subpath

Tests and test paths (cont'd)



In-class exercise

$$
\begin{aligned}
N &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\
N_0 &= \{1, 2, 3\} \\
N_f &= \{8, 9, 10\} \\
E &= \{(1, 4), (1, 5), (4, 8), (5, 8), (2, 5), (5, 9), \\
&\quad (9, 6), (6, 2), (3, 6), (6, 10), (3, 7), (7, 10)\}
\end{aligned}
$$

Outline

1. Graph coverage I
   - Terminology
   - Structural coverage criteria
   - Touring and sidetrips

Graph coverage

- A test is represented as a test path. Test requirements impose constraints on test paths. Test criteria are rules for test requirements.
- The general notion of coverage for (general) graphs:

Definition

Given a coverage criterion C, a graph G, and a set TR of test requirements for that criterion on G. A test set T satisfies C on the graph G iff for every test requirement *tr* in TR, there is a path in *path*(T) ("test path") that covers *tr*.

- Different views on graphs refine that definition further.
- Major test criteria
  - Structural criteria: defined directly on the graph, in terms of nodes and edges
  - Data-flow criteria: defined on the graph annotated with variable information

Node coverage

Definition

A test set T satisfies node coverage (NC) on a graph $G = (N, E, N_o, N_f)$ if for every syntactically reachable node n in N there exists a path p in path(T) ("test path") such that p visits n.

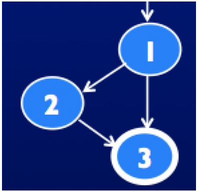
- Alternative definition: "Given a graph G. In the node coverage (NC) criterion, the set TR of test requirements contains all syntactically reachable nodes."
- The simplest and very common criteria

Edge coverage

Definition

Given a graph $G = (N, E, N_o, N_f)$. In the edge coverage (EC) criterion, TR contains each syntactically reachable path in G of length up to 1.

- Example



  - TR={[1,2], [1,3], [2,3]}. (Strictly speaking: TR={[1,2], [1,3], [2,3], [1], [2], [3]})
  - Then T= {[1,2,3], [1,3]} is a set of test paths that meets *TR*.
- Admits graphs with one node and no edge.

Node coverage and edge coverage

- Recall subsumption: a coverage criterion C_1 subsumes C_2 iff every test set that satisfies C_1 also satisfies C_2.
- Edge coverage subsumes node coverage.
  - Without the clause "length up to 1", subsumption would not hold.
  - Ex.: consider a graph of one node: $G_1 = (\{n\}, \emptyset, \{n\}, \{n\})$.
    For G_1, TR_{NC} (node coverage) : $\{n\}$, satisfied by test set $T_{NC} = \{t\}$ with path(t) = [n]
  - Assume we had defined edge coverage as

    " ... TR contains each syntactically reachable path in G of length = 1".

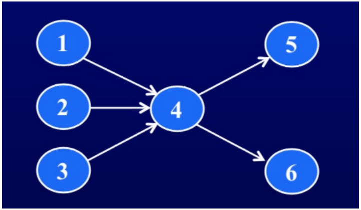
    Then, for G_1, TR_{EC}(edge coverage): $\emptyset$, satisfied by test set $T_{EC} = \emptyset$, and EC would not subsume NC (empty path does not visit n).

Edge-pair coverage

Definition

Given a graph $G = (N, E, N_o, N_f)$. In the edge-pair coverage (EPC) criterion, TR contains each syntactically reachable path in G of length up to 2.

- Example:



- TR={[1,4,5], [1,4,6], [2,4,5], [2,4,6], [3,4,5], [3,4,6]}
- The clause "length up to 2" matters for graphs without paths of length ≥ 2.

Complete path coverage

Definition

Let $G = (N, E, N_o, N_f)$ be a graph. In the complete path coverage (CPC) criterion, TR contains all paths in G.

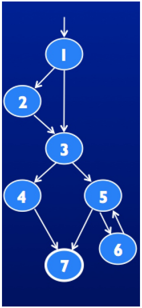
- For graphs with loops, the test set is infeasible.
- Ad-hoc definition as possible workaround:

Definition

Let $G = (N, E, N_o, N_f)$ be a graph. In the specified path coverage (SPC) criterion, TR contains a set S of test paths in G, where S is externally defined.

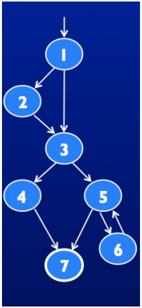
But SPC lacks objectivity. More on loops below.

Example (structural coverage)



- Node coverage
  - TR = {1,2,3,4,5,6,7}, test paths ={[1,2,3,4,7], [1,2,3,5,6,5,7]}
- Edge coverage
  - TR = {[1,2], [1,3], [2,3], [3,4], [3,5], [4,7], [5,6] [5,7], [6,5]}, test paths ={[1,2,3,4,7], [1,3,5,6,5,7]}

Example (structural coverage)



- Edge-pair coverage
  - TR = {[1,2,3], [1,3,4], [1,3,5], [2,3,4], [2,3,5], [3,4,7], [3,5,6], [3,5,7], [5,6,5], [6,5,6], [6,5,7]}, test paths ={[1,2,3,4,7], [1,2,3,5,7], [1,3,4,7], [1,3,5,6,5,6,5,7]}
- Complete-path coverage
  - TR ={[1,2,3,4,7], [1,2,3,5,7], [1,2,3,5,6,5,6] [1,2,3,5,6,5,6,5,7] [1,2,3,5,6,5,6,5,6,5,7] . . . }

How to deal with a loop?

It is surprisingly difficult to deal with loops in graphs. Historically:

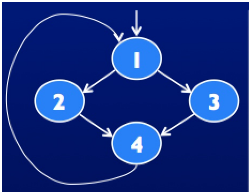
- 1970s: execute cycle once (e.g., [5,6,5])
- 1980s: execute each loop exactly once (formally)
- 1990s: execute loops 0, once, more than once (informally)
- 2000s: prime path (touring, sidetrips, detours)

Simple paths

Definition

A path from node n_i to n_j is simple if no node appears more than once, except for the first and last nodes, which may be the same.

- No nested loops.
- Example:



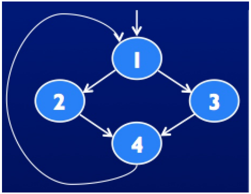
- Set of simple paths: { [1,2,4,1], [1,3,4,1], [2,4,1,2], [2,4,1,3], [3,4,1,2], [3,4,1,3], [4,1,2,4], [4,1,3,4], [1,2,4], [1,3,4], [2,4,1], [3,4,1], [4,1,2], [4,1,3], [1,2], [1,3], [2,4], [3,4], [4,1], [1], [2], [3], [4] }

Prime paths

Definition

Given a set P of paths. A simple path that does not appear as a proper subpath of any other simple path in P is called a prime path.

- Example:



- Set of prime paths: {[1,2,4,1], [1,3,4,1], [2,4,1,2], [2,4,1,3], [3,4,1,2], [3,4,1,3], [4,1,3,4], [4,1,2,4] }
- In the example, all simple (and prime) paths visit the loop. In the general case, simple paths exist that skip the loop.

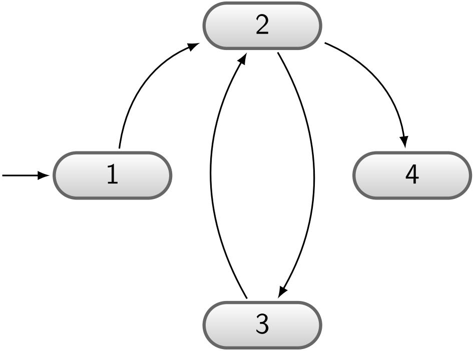
Prime path coverage

Definition

Let $G = (N, E, N_o, N_f)$ be a graph. In the prime path coverage (PPC) criterion, TR contains each prime path in G.

- Loops are executed 0x (if possible), 1x, $> 1x$
- Tours all paths of length 0, 1

In-class exercise

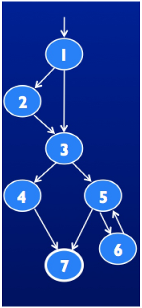


Outline

Prime path coverage and edge-pair coverage

- PPC does not subsume EPC.
- Consider a node n with a self-edge and an edge (n, m).
  - EPC requires [n,n,m].
  - But [n,n,m] is not prime ([n,n,m] is not simple).
- Example: N={1,2,3}, E={(1,2), (2,2), (2,3) }
  - EPC: TR = {[1,2,3], [1,2,2], [2,2,3], [2,2,2]}
  - PP: TR={[1,2,3], [2,2] }

Example (prime paths)



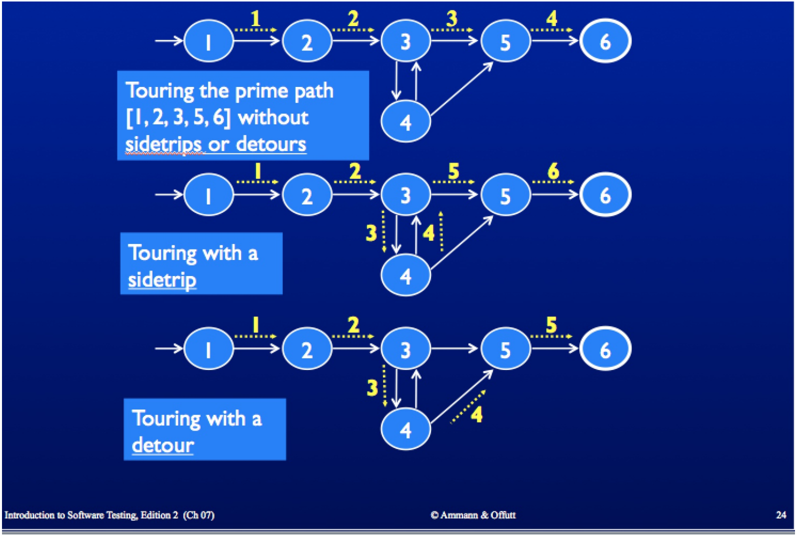
- 38 simple paths, 9 prime paths
- Prime paths: [1,2,3,4,7], [1,2,3,5,7], [1,2,3,5,6], [1,3,4,7], [1,3,5,7], [1,3,5,6], [6,5,7], [6,5,6] [5,6,5]
- Loop is executed 0 times, once, and more than once.

Tours, sidetrips, detours

Definition

A test path p tours subpath q if q is a subpath of p.
A test path p tours subpath q with sidetrips iff every edge in q is also in p, in the same order.
A test path p tours subpath q with detours iff every node in q is also in p, in the same order.

Example (tours, sidetrips, detours)



Infeasible requirements

- Recall the notion of infeasiblity:

Definition

A test requirement that cannot be satisfied is called infeasible

  - Most test criteria can result in infeasible test requirements.
  - Ex.: dead code; subpaths with inconsistent conditions
- Sidetrips allow turning (some) infeasible requirements into feasible ones.
- Ex.: assume *TR* contains [2,3,5], which cannot be covered by test path [1,2,3,4,3,5,6].
  - Assume edge [3,5] not executable. Then test path tours [2,3,5] with the detour [3,4,3].

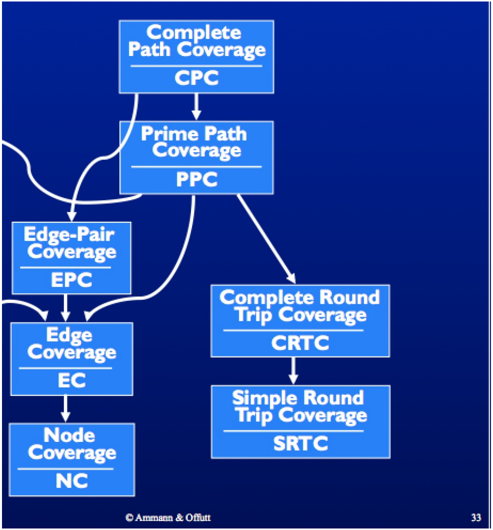
Round-trip coverage

Definition

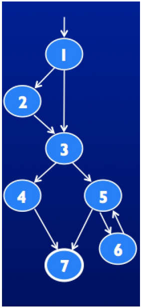
A prime path that starts and ends at the same node is called a round-trip path.
Let $G = (N, E, N_o, N_f)$ be a graph. In the simple round-trip coverage (SRTC) criterion, TR contains at least one round-trip path for each reachable node in G that begins and ends a round-trip path.
If TR contains all round-trip paths, the coverage is called complete round-trip converage (CRTC).

- Special case of PPC, reduces the number of tests
- Focus on loops
- Ex.: nodes 5,6 start round-trip paths [6,5,6], [5,6,5]

Subsumption (structural coverage)



Appendix: Constructing prime paths



- Construct simple paths of length 0, 1, 2, ... Mark terminating path using !, and cycles using *. Eliminate all others.
  - Len 0: [1], [2], [3], [4], [5], [6], [7]!
  - Len 1: [1,2], [1,3], [2,3], [3,4], [3,5], [4,7]!, [5,7]!, [5,6], [6,5]
  - Len 2: [1,2,3], [1,3,4], [1,3,5], [2,3,4], [2,3,5], [3,4,7]!, [3,5,7]!, [3,5,6]!, [5,6,5]*, [6,5,7]!, [6,5,6]*
- Check for subpath relationships in the remaining ones.

References

- AO, Ch. 7.1 and 7.2.1-7.2.2