Structural Testing

Mohammad Mousavi

Department of Informatics, King's College London

Software Testing and Measurement,
October 11, 2021

Outline

Functional Testing: Pros and Cons

Pros:

- Straightforward test-case generation
- Based on specification (early test-case generation)

Cons:

- No use of program information
- Gaps and redundancies

Structural Testing

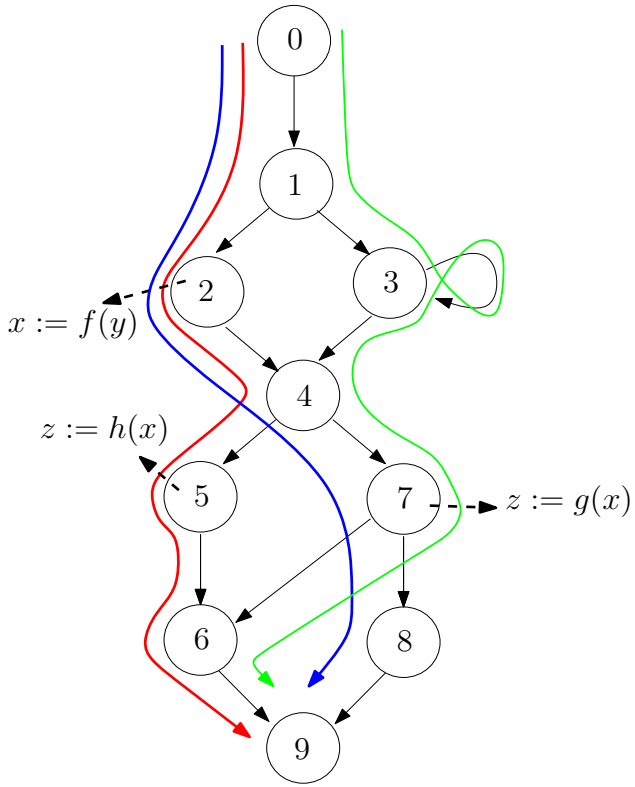
Idea

- Derive structural abstractions from programs
  Example: flow graphs
- Use them to measure the adequacy of the test-set

Test Adequacy Criteria

The test-set covers, in the flow graph,

1. all nodes (statement coverage)
2. all edges (DD-path coverage)
3. all prime paths (single-loop coverage)
4. all edges + all combinations of data-flow dependent edges (dependent pairs coverage)



Finite Feasibility

An adequacy criteria should be
satisfiable by some finite test-set.

Question: Which of the aforementioned criteria are finitely feasible?

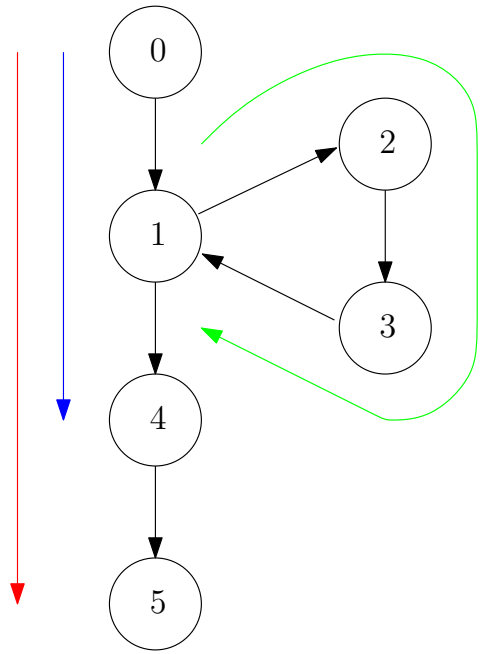
Outline

Simple Path: Definition

A simple path $n_0, \ldots, n_t$, with $0 \leq t$, is a list of nodes s.t.

1. $n_j \rightarrow n_{j+1}$ for each $j < t$,
2. for each $0 \leq i < j \leq i$, $n_i \neq n_j$ or $(n_i = n_0$ and $n_j = n_t)$

Informally: a simple path visits a node at most once, except that the start and the ending node may be the same.

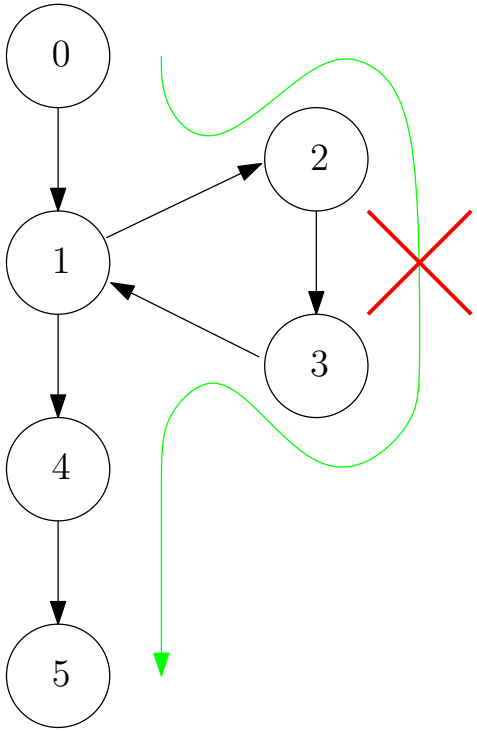


Simple Path: Definition

A simple path $n_0, \ldots, n_i$, with $0 \leq i$, is a list of nodes s.t.

1. $n_j \rightarrow n_{j+1}$ for each $j < i$,
2. for each $0 < i < j \leq i$, $n_i \neq n_j$

Informally: a simple path visits a node at most once, except that the start and the ending node may be the same.

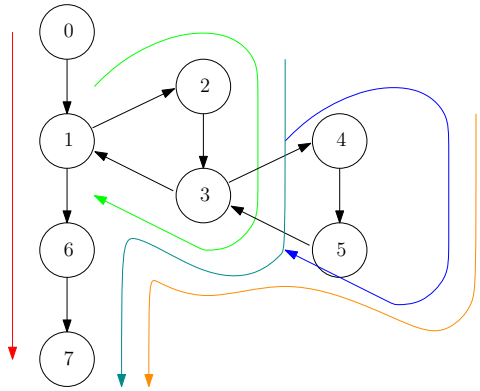


Prime Path: Definition

A prime path is:

- a simple path that
- does not appear as a proper sub-path of any other simple path.

Informally: a prime path is a complete path from start to end, or a complete and simple iteration of a loop (infeasibility issue set aside)

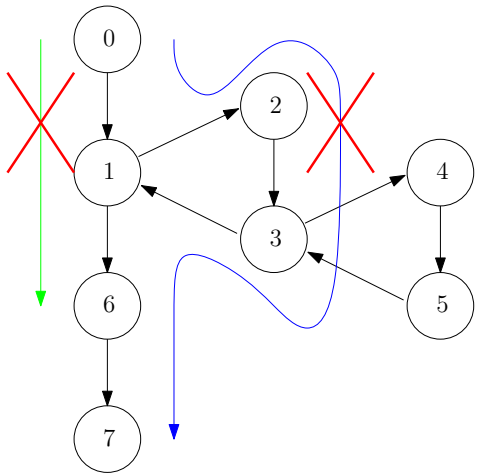


Prime Path: Definition

A prime path is:

- a simple path that
- does not appear as a proper sub-path of any other simple path.

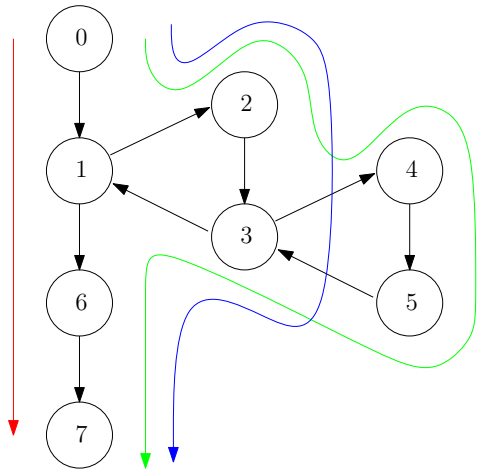
Informally: a prime path is a complete path from start to end, or a complete and simple iteration of a loop



Prime Path Coverage

A test set is adequate if for each prime path, there is a test case covering it (as a sub-path).

Informally: all complete simple paths and up to two iteration of each loop

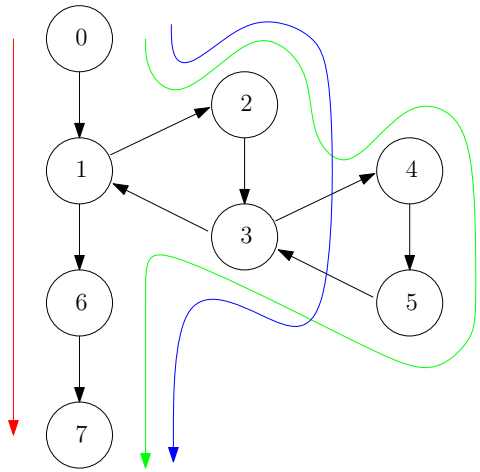


Prime Path Coverage

A test set is adequate if for each prime path, there is a test case covering it (as a sub-path).

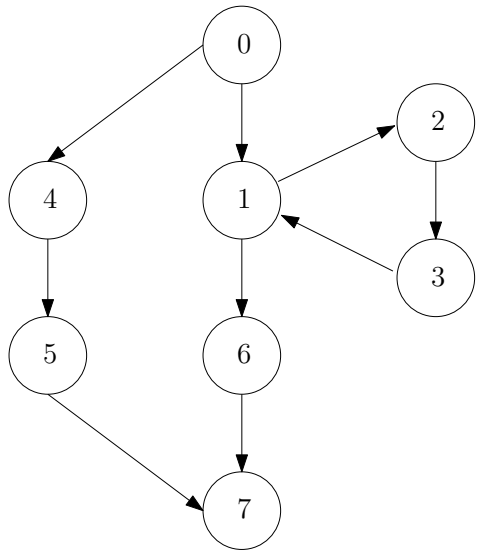
Informally: all complete simple paths and up to two iteration of each loop

Variants with tours, detours and side-trips



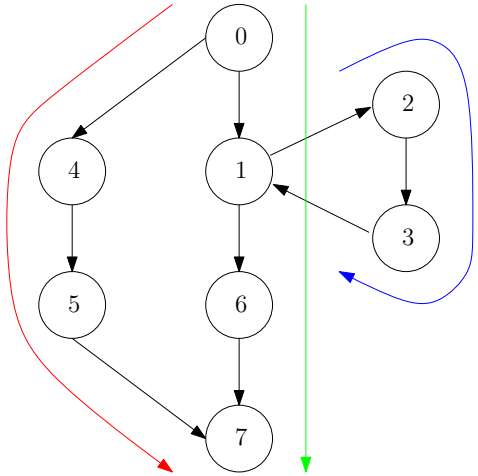
Prime Path Coverage: Exercise

Propose a set of test cases that is adequate for prime path coverage.



Prime Path Coverage: Solution

Some prime paths (not all)



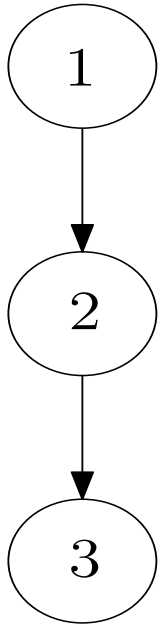
Outline

Testability: Cyclomatic number

1. Idea (very informal):
   Take one path from start to exit,
   count the number of alternatives by
   flipping one condition at a time.
2. Also called: McCabe complexity, nullity, first Betti number, dimension of cycle space

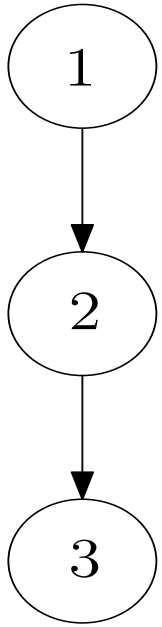
Cyclomatic number: Examples

Cyclomatic number:



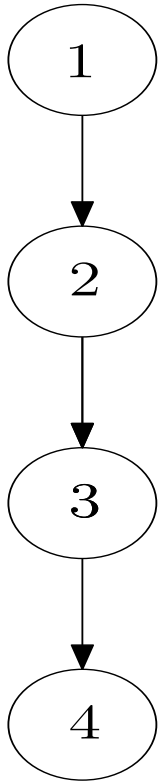
Cyclomatic number: Examples

Cyclomatic number: 1



Cyclomatic number: Examples

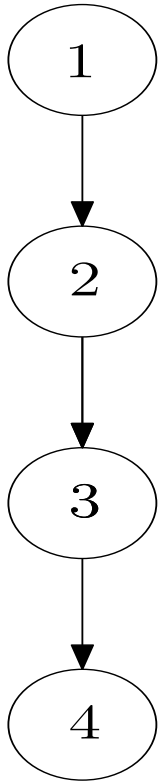
Cyclomatic number:



Cyclomatic number: Examples

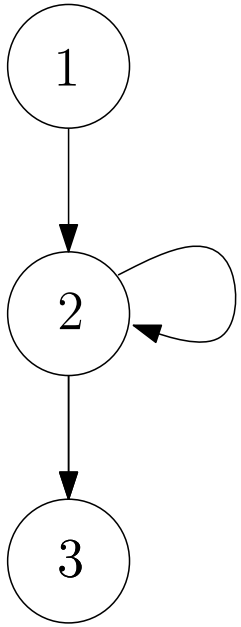
Cyclomatic number: 1

Observation: Cyclomatic nr. is size independent...



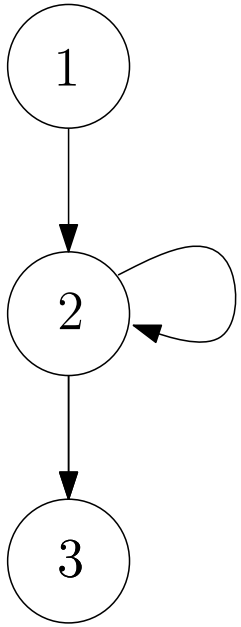
Cyclomatic number: Examples

Cyclomatic number:



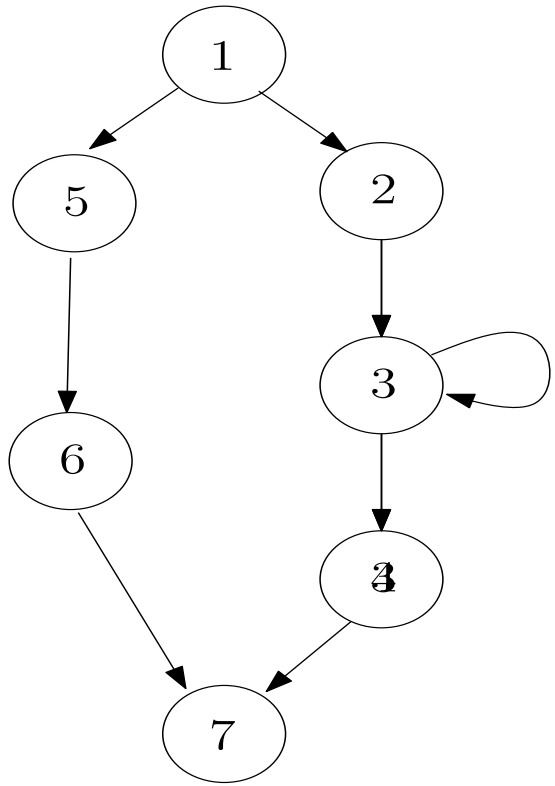
Cyclomatic number: Examples

Cyclomatic number: 2



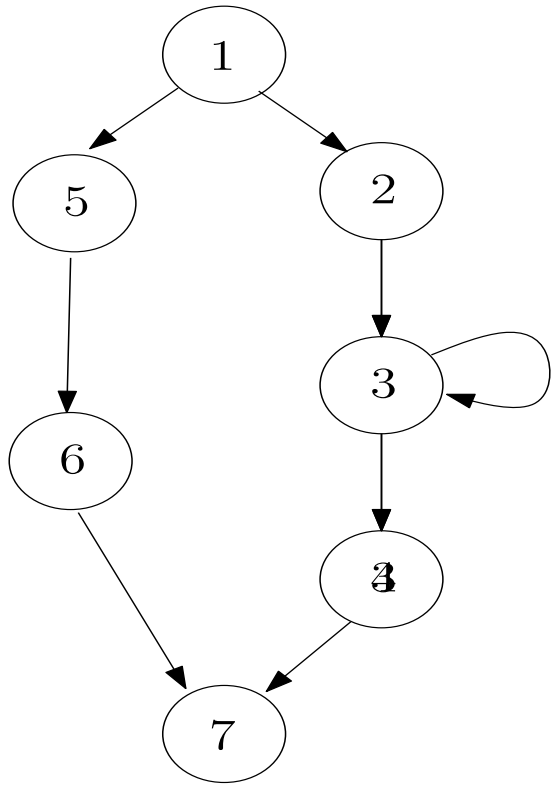
Cyclomatic number: Examples

Cyclomatic number:



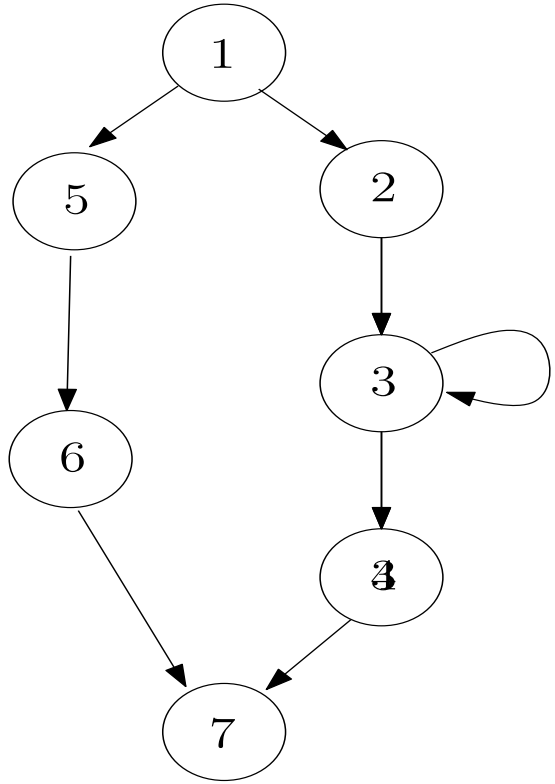
Cyclomatic number: Examples

Cyclomatic number: 3



Cyclomatic number: Examples

Cyclomatic number:



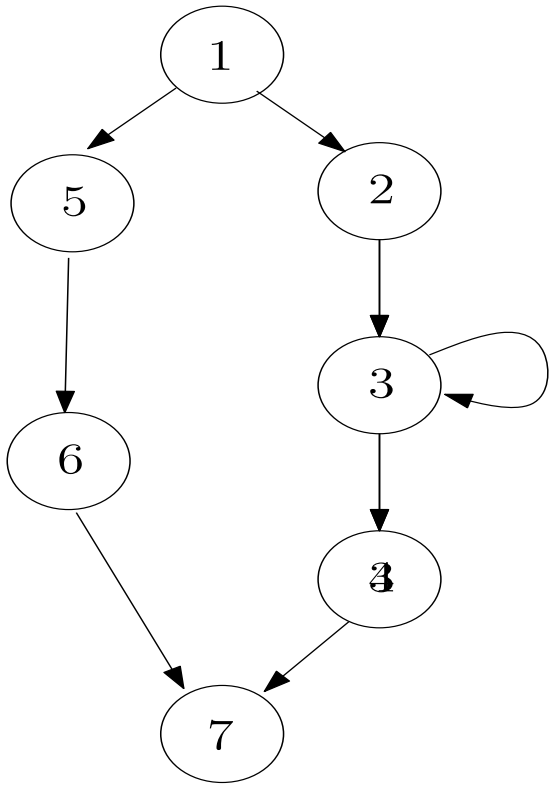
Cyclomatic number: Examples

Cyclomatic number: 3

Only for programs with:

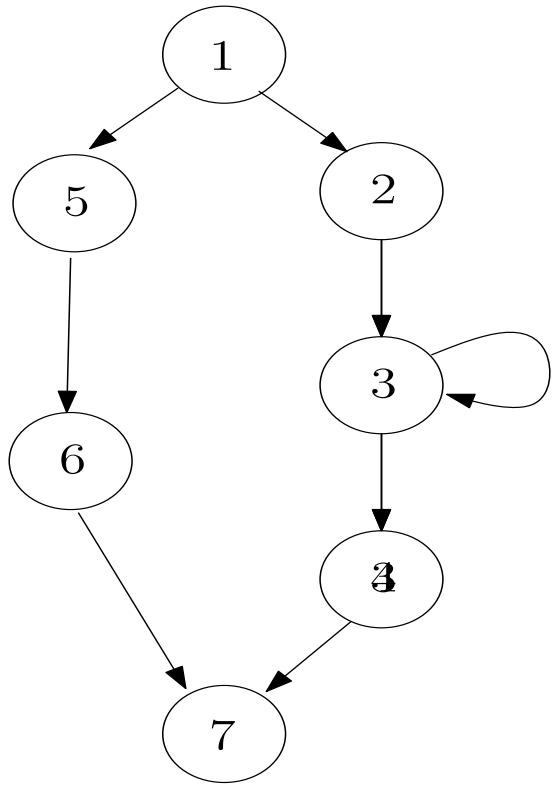
1. one connected component,
2. one starting state, and
3. one terminal state:

$V(G) = \#edges - \#nodes + 2$



Cyclomatic number: Calculation

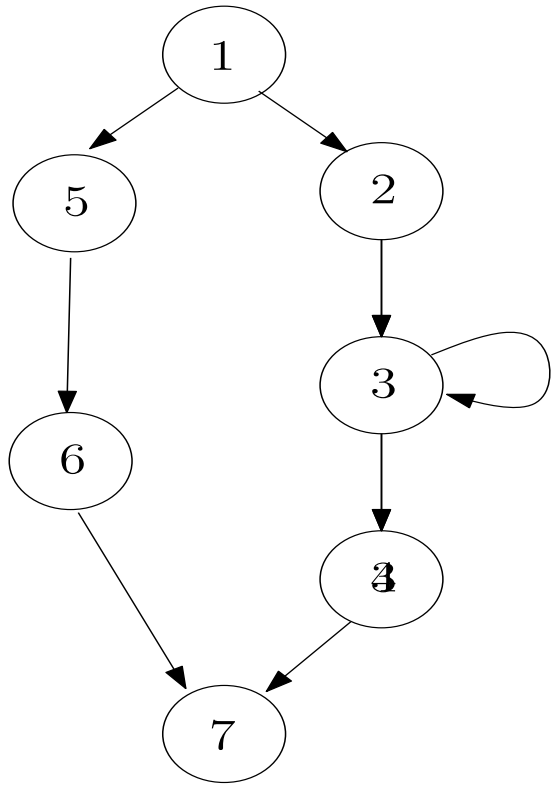
Cyclomatic number:



Cyclomatic number: Calculation

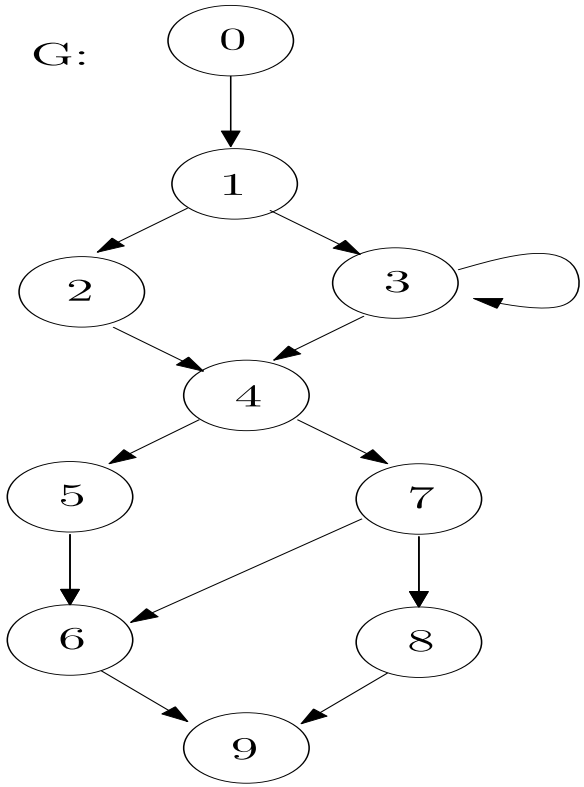
Cyclomatic number: 3

For (connected) planar graphs:
$V(G) = \#regions\ in\ the\ plane$



Cyclomatic number: Example

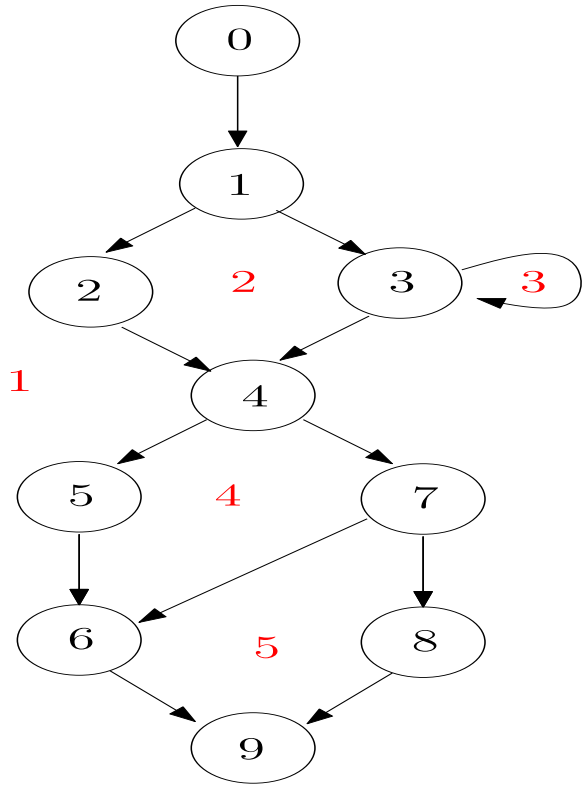
1. the cyclomatic number of G?
2. the cyclomatic number of G sequentially composed with itself?
3. the cyclomatic number of G *n*-times sequentially composed with itself?



Cyclomatic number: Example

What is

1. the cyclomatic number of G? 5
2. the cyclomatic number of G sequentially composed with itself? $9 = 5 + 5 - 1$
3. the cyclomatic number of G n-times sequentially composed with itself? $4 * n + 1$



Cyclomatic number: Implications

Cyclomatic Complexity	Risk Evaluation
1-10	a simple program, without much risk
11-20	more complex, moderate risk
21-50	complex, high risk program
>50	untestable program (very high risk)

Conclusions

1. Cyclomatic number: a measure for software complexity and testability watch out for programs with $V(G) > 10$!
2. implemented in several tools (see the course web-site for examples)
3. a measure of test-set adequacy (originally invented for this purpose!)

Conclusions

1. Cyclomatic number: a measure for software complexity and testability watch out for programs with $V(G) > 10$!
2. implemented in several tools (see the course web-site for examples)
3. a measure of test-set adequacy (originally invented for this purpose!) Let's have a better look...