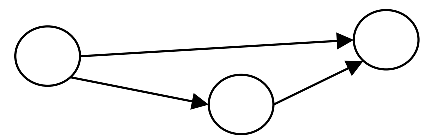
SYSC 4101 / 5105

Graph Criteria (Part I)

Test Criteria Based on Structure [Offutt]

- Graphs

Method body
Methods and calls
Components interactions
State and transitions
…



- Logical Expressions

[not X or not Y] and A and B

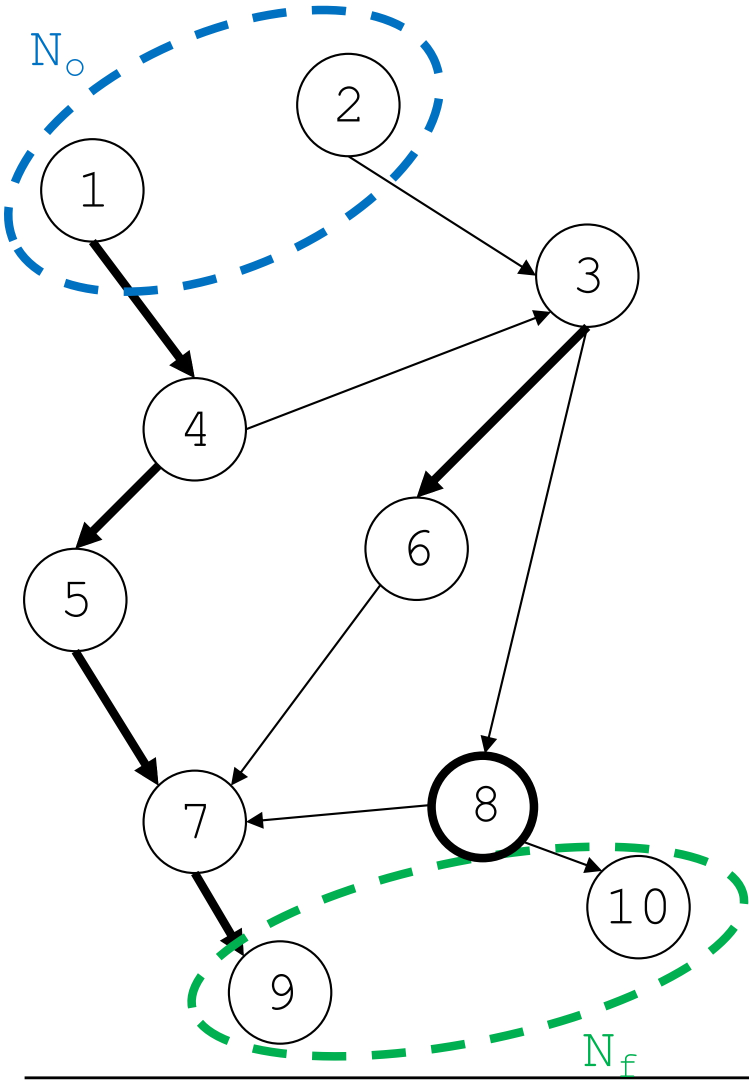
- Input Domain Characterization

A: {0,1,>1}
B: {600,700,800}
C: {swe,cs,isa,infs}

- Syntactic Structures

```
if [x>y]
    z = x - y;
else
    z = 2 * x
```

Graph Principles



- A **directed graph** is defined by:
  - a set N of **nodes**
  - a set N_0 of **initial nodes**, where $N_0 \subseteq N$: $|N_0| \geq 1$
  - a set N_f of **final nodes**, where $N_f \subseteq N$: $|N_f| \geq 1$
  - a set E of **edges**, where $E \subseteq N \times N$
- **Path** = a sequence $[n_1, n_2, \ldots, n_M]$ of nodes, where each pair of adjacent nodes $[n_i, n_{i+1}]$, $1 \leq i < M-1$, is in E.
  - [n] is a path of length zero
  - Two nodes may share several edges, in which case a path definition needs to indicate edges and nodes
- The **length** of path p is the number of its edges.
- A **sub-path** of path p is a subsequence of p (possibly p itself).
  - $[n_2,n_3]$ is a sub-path of $[n_2,n_3,n_6,n_7]$
  - $[n_1,n_4,n_5]$ is a sub-path of $[n_1,n_4,n_5,n_7,n_9]$
  - $[n_4]$ is a sub-path of $[n_1,n_4,n_5]$

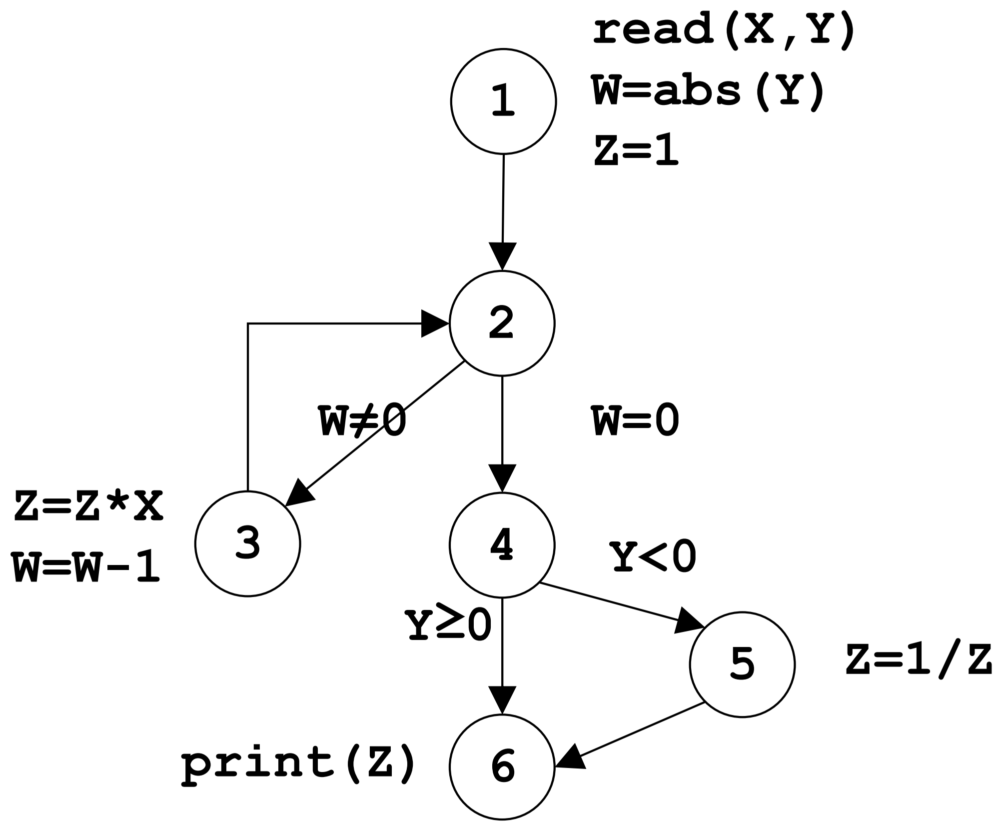
Graph Testing—Criteria

- Graph criteria define a set of test requirements to be achieved
  - These requirements are stated in terms of elements of the graph to be exercised.
  - The requirements then translate into paths from an initial to a final node [to exercise the required elements]: a.k.a. test paths.
  - The paths then become test cases: one identifies (inputs) values to execute the path.
- ➢ Graph criteria, therefore, consider the syntax of the graph
  - ➢ Not its semantics!
- ➢ A criterion may result in infeasible objectives (and therefore paths)
  - See example next.
- Reachability:
  - Node n [or edge e] is syntactically reachable from node n_i if there exists a path from node n_i to n [or e]
  - Node n [or edge e] is also semantically reachable if it is possible to execute at least one of the paths with some input.

Example: Power Function

Program computing Z=X^Y

```
BEGIN
  read (X, Y) ;
  W = abs(Y) ;
  Z = 1 ;
  WHILE (W <> 0) DO
    Z = Z * X ;
    W = W - 1 ;
  END
  IF (Y < 0) THEN
    Z = 1 / Z ;
  END
  print (Z) ;
END
```



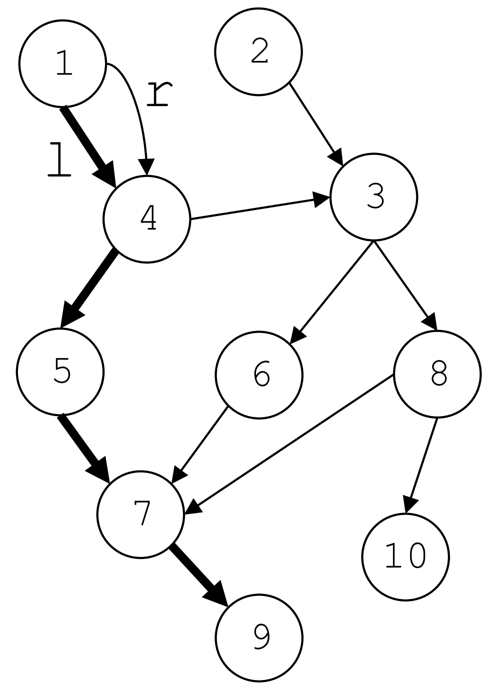
Test requirement [1,2,4,5] is:

- Syntactically feasible/reachable: path [1,2,4,5,6] is possible in the graph.
- Semantically infeasible/unreachable: need to satisfy y=0 and y<0.

Graph Testing—Terminology

- Test Path
  - A path p, possibly of length zero, that starts at some node in N_0 and ends at some node in N_f.
    - A complete traversal of the graph, from an initial node to a final node.
  - A test path p **visits** node n [resp. edge e] if n [resp. e] is in p.
  - A test path p **tours** sub-path q if q is a sub-path of p.

- Unless the system is non-deterministic, a test case executes <u>one</u> test path.
  - path[T] is the set of test paths executed by test set T.



Path [1, l, 4, 5, 7, 9]:

- Visits node 5
- Visits edge (7,9)
- Tours path [5, 7, 9]

Graph Testing—Two Families of Criteria

- Control Flow Criteria
  - Only consider the flow of nodes and edges
  - Seven criteria (many more criteria exist)

- Data Flow Criteria
  - Considers the definitions and usages of data along paths
  - Three criteria (many more criteria exist)

Reminder: a criterion can be an selection criterion or a coverage criterion.

- One can use selection control flow criterion A and coverage control flow criterion B
- One can use selection control flow criterion A and coverage data flow criterion B
- One can use selection data flow criterion A and coverage control flow criterion B
- …

Graph Testing—Control Flow Criteria [textbook]

- Graph Criteria (generic/template definition)
  - Given a set TR of test requirements for a graph criterion C, a test set T satisfies C on graph G if and only if for every test requirement tr in TR, there is at least one test path p in path[T] such that p meets tr.
    - Overall, all the tr in TR are exercised by paths in path[T], the paths exercised by T.
  - **Recall the procedure:**

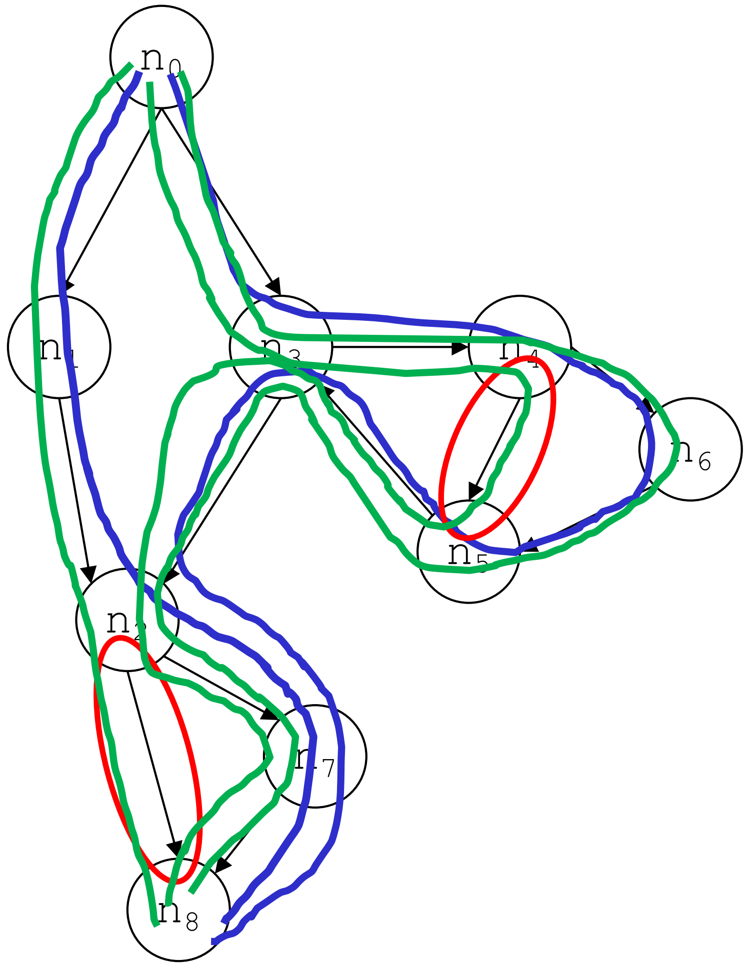
    Criterion → Test Requirements (paths) → Test Paths → Test inputs
- All-Nodes Criterion [NC]:
  - TR contains each reachable node in G.
- All-Edges Criterion [EC]:
  - TR contains each reachable [sub-]path of length up to 1, inclusive, in G.
    - Length of 0 or 1, i.e., nodes and edges. (All-edges subsumes All-Nodes by construction.)
- All-Edge-Pairs Criterion [EPC]:
  - TR contains each reachable path of length up to 2, inclusive, in G.
    - Length of 0, 1, or 2, i.e., nodes, edges and pairs of consecutive edges. (All-Edge-Pairs subsumes All-Edges by construction.)

Graph Testing—Control Flow Criteria [2]

- Prime-Path Criterion [PPC]: (textbook illustrates an algorithm to find them)
  - TR contains each prime path in G.
  - A path from n_i to n_j is a prime path if it is a simple path and it does not appear as a proper sub-path of any other simple path
  - A path from n_i to n_j is simple if no node appears more than once in the path.
    - Exception: the first and last nodes of the path may be identical (the sub-path is a loop).
  - A prime path is a simple path of maximum length.
- Simple-Round-Trip Criterion [SRTC]:
  - TR contains at least one round-trip path for each reachable node in G that begins and ends a round-trip path.
  - A round-trip path is a nonzero length prime path that starts and ends at the same node, i.e., a loop with at least one edge.
- Complete-Round-Trip Criterion [CRTC]:
  - TR contains all round-trip paths for each reachable node in G.
- Complete-Path Criterion [CPC]:
  - TR contains all paths in G.

Example

Each time, we list the set of test requirements and then test paths that are adequate for the criterion.



- All-Nodes Criterion
  - Test requirements: $[n_0]$, $[n_1]$, $[n_2]$, $[n_3]$, $[n_4]$, $[n_5]$, $[n_6]$, $[n_7]$, $[n_8]$
  - $path[t_1]=[n_0,n_1,n_2,n_7,n_8]$
  - $path[t_2]=[n_0,n_3,n_4,n_6,n_5,n_3,n_2,n_7,n_8]$
  - $T_{NC}=\{t_1,t_2\}$ satisfies the All-Nodes criterion
    - Many other possibilities…
- All-Edges Criterion
  - Test requirements

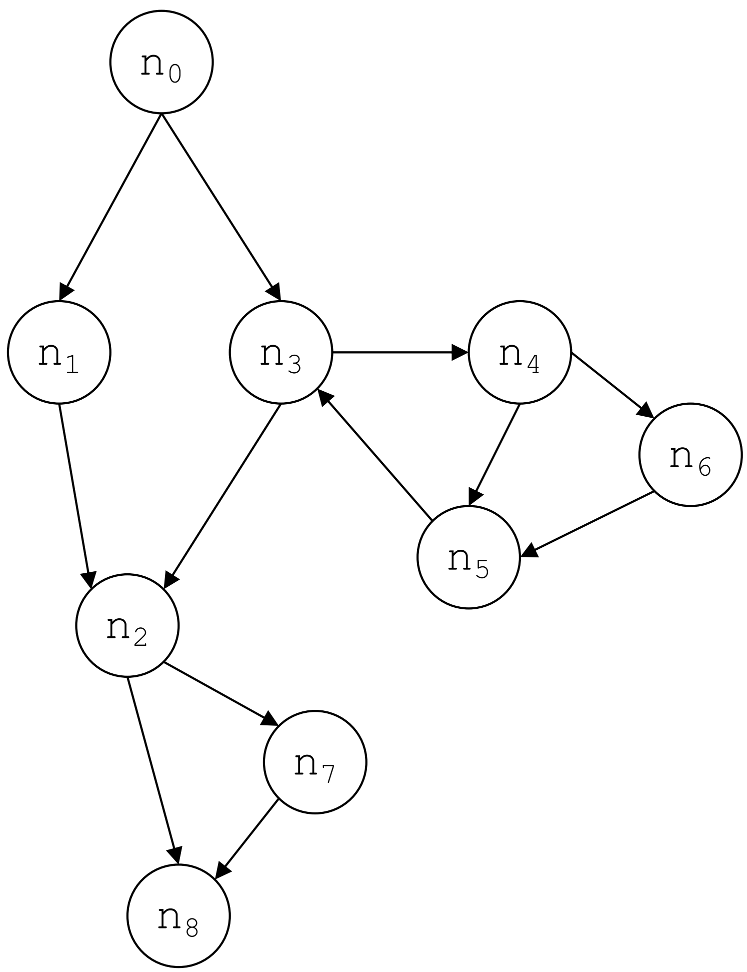
    $[n_0]$, $[n_1]$, $[n_2]$, $[n_3]$, $[n_4]$, $[n_5]$, $[n_6]$, $[n_7]$, $[n_8]$

    $[n_0,n_1]$ $[n_0,n_3]$ $[n_1,n_2]$ $[n_2,n_7]$ $[n_2,n_8]$

    $[n_3,n_4]$ $[n_3,n_2]$ $[n_4,n_5]$ $[n_4,n_6]$ $[n_5,n_3]$

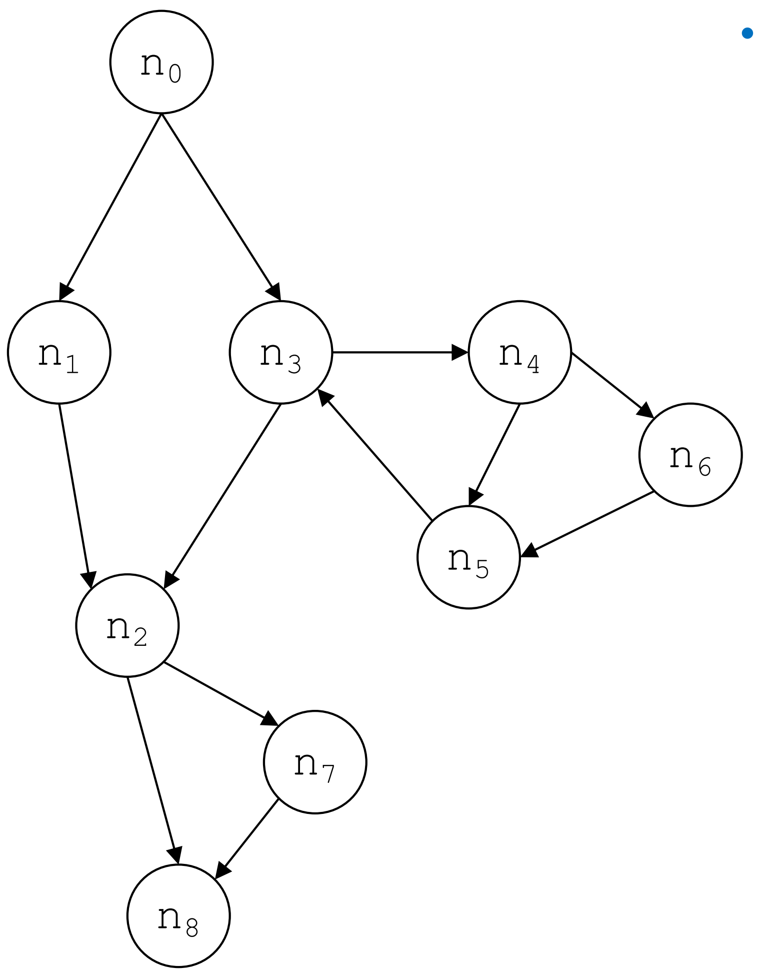
    $[n_6,n_5]$ $[n_7,n_8]$
  - $path[t_3]=[n_0,n_1,n_2,n_8]$
  - $path[t_2]=[n_0,n_3,n_4,n_6,n_5,n_3,n_2,n_7,n_8]$
  - $path[t_4]=[n_0,n_3,n_4,n_5,n_3,n_2,n_7,n_8]$
  - $T_{EC}=\{t_2,t_3,t_4\}$ satisfies the All-Edges criterion

Example [cont.]



- All-Edge-Pairs Criterion
  - Test requirements
    [n_0], [n_1], [n_2], [n_3], [n_4], [n_5], [n_6], [n_7], [n_8]
    [n_0,n_1] [n_0,n_3] [n_1,n_2] [n_2,n_7] [n_2,n_8]
    [n_3,n_4] [n_3,n_2] [n_4,n_5] [n_4,n_6] [n_5,n_3]
    [n_6,n_5] [n_7,n_8]
    [n_0,n_1,n_2] [n_0,n_3,n_2] [n_0,n_3,n_4]
    [n_1,n_2,n_7] [n_1,n_2,n_8] [n_2,n_7,n_8]
    [n_3,n_4,n_6] [n_3,n_4,n_5] [n_3,n_2,n_7] [n_3,n_2,n_8]
    [n_4,n_5,n_3] [n_4,n_6,n_5] [n_5,n_3,n_2] [n_5,n_3,n_4]
    [n_6,n_5,n_3]
  - path[t_1]=[n_0,n_1,n_2,n_7,n_8]
  - path[t_3]=[n_0,n_1,n_2,n_8]
  - path[t_5]=[n_0,n_3,n_2,n_7,n_8]
  - path[t_6]=[n_0,n_3,n_4,n_6,n_5,n_3,n_4,n_5 ,n_3,n_2,n_8]
  - T_{EPC}={t_1,t_3,t_5, t_6} satisfies the All-Edge-Pairs criterion

Example [cont.]



- Prime-Path Criterion
  - Test requirements

    $[n_0,n_1,n_2,n_8]$ $[n_0,n_1,n_2,n_7,n_8]$ $[n_0,n_3,n_2,n_8]$

    $[n_0,n_3,n_2,n_7,n_8]$ $[n_0,n_3,n_4,n_5]$ $[n_0,n_3,n_4,n_6,n_5]$

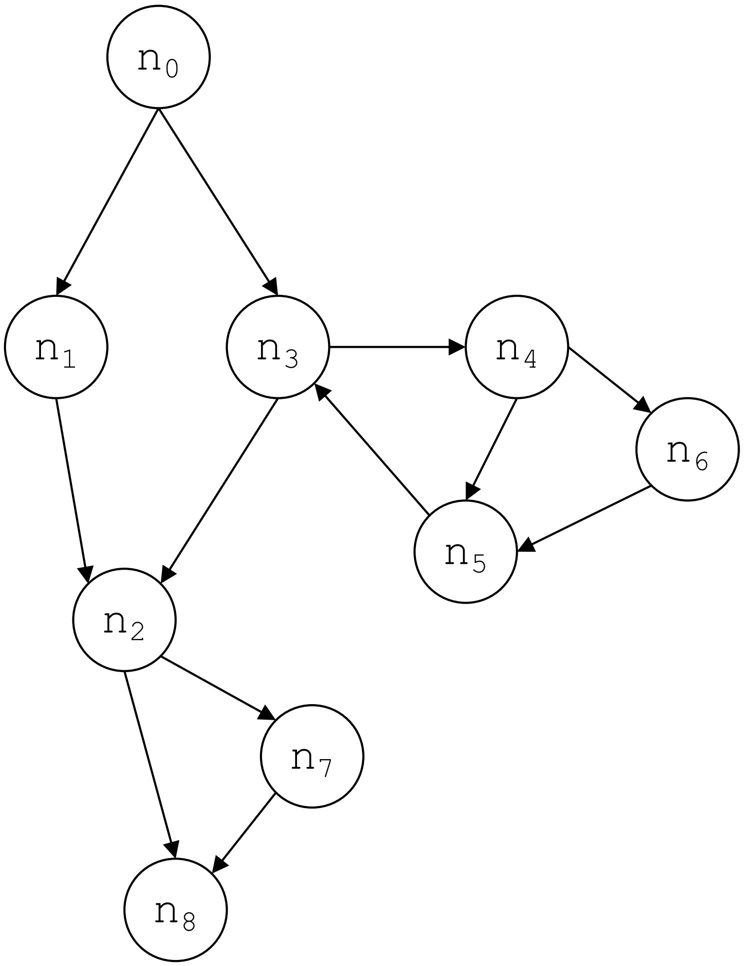
    $[n_3,n_4,n_5,n_3]$ $[n_3,n_4,n_6,n_5,n_3]$

    $[n_4,n_5,n_3,n_2,n_8]$ $[n_4,n_5,n_3,n_2,n_7,n_8]$ $[n_4,n_6,n_5,n_3,n_4]$

    $[n_4,n_6,n_5,n_3,n_2,n_8]$ $[n_4,n_6,n_5,n_3,n_2,n_7,n_8]$ $[n_4,n_5,n_3,n_4]$

    $[n_5,n_3,n_4,n_5]$ $[n_6,n_5,n_3,n_4,n_6]$ $[n_5,n_3,n_4,n_6,n_5]$
  - $path[t_1]=[n_0,n_1,n_2,n_7,n_8]$
  - $path[t_3]=[n_0,n_1,n_2,n_8]$
  - $path[t_5]=[n_0,n_3,n_2,n_7,n_8]$
  - $path[t_6]=[n_0,n_3,n_4,n_6,n_5,n_3,n_4,n_5 ,n_3,n_2,n_8]$
  - $path[t_7]=[n_0,n_3,n_2,n_8]$
  - $path[t_8]=[n_0,n_3,n_4,n_5,n_3,n_2,n_7,n_8]$
  - $path[t_9]=[n_0,n_3,n_4,n_6,n_5,n_3,n_2,n_7,n_8]$
  - $path[t_{10}]=[n_0,n_3,n_4,n_6,n_5,n_3,n_2,n_8]$
  - $path[t_{11}]=[n_0,n_3,n_4,n_5,n_3,n_4,n_6,n_5,n_3,n_4,n_6,n_5,n_3,n_2,n_8]$
  - $T_{PPC}=\{t_1,t_3,t_5,t_6,t_7,t_8,t_9,t_{10},t_{11}\}$ satisfies the Prime-Path criterion

Example [cont.]



- Simple-Round-Trip Criterion
  - Test requirements

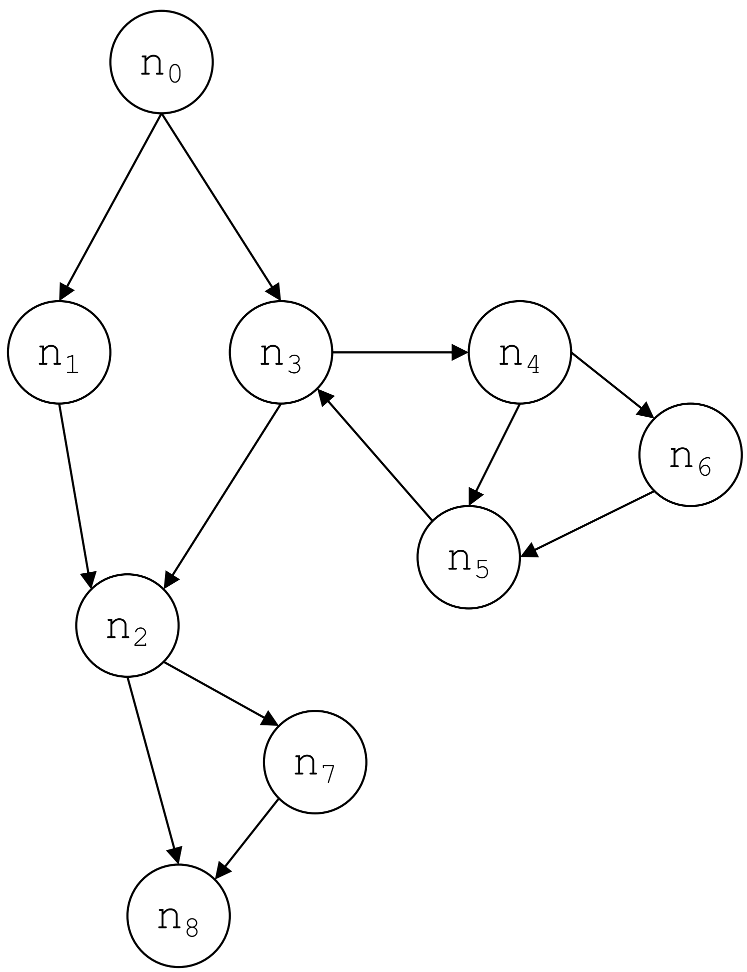
    $\underline{[n_3,n_4,n_5,n_3]}$ $[n_3,n_4,n_6,n_5,n_3]$ (one of these 2)

    $\underline{[n_4,n_5,n_3,n_4]}$ $[n_4,n_6,n_5,n_3,n_4]$ (one of these 2)

    $[n_5,n_3,n_4,n_5]$ $\underline{[n_5,n_3,n_4,n_6,n_5]}$ (one of these 2)

    $\underline{[n_6,n_5,n_3,n_4,n_6]}$
  - path$[t_{12}]=[n_0,n_3,n_4,n_5,n_3,n_4,n_6,n_5,n_3,n_4,n_6,n_5,n_3,n_2,n_7,n_8]$
    - Covered requirements are underlined.
  - $T_{SRTC}=\{t_{12}\}$ satisfies the Simple-Round-Trip criterion
- Complete-Round-Trip Criterion
  - Test requirements
    - See above: all the seven round trips must be exercises
  - path$[t_{13}]=[n_0,n_3,n_4,n_5,n_3,n_4,n_6,n_5,n_3,n_4,n_6,n_5,n_3,n_4,n_5,n_3,n_2,n_7,n_8]$
  - $T_{SRTC}=\{t_{13}\}$ satisfies the Complete-Round-Trip criterion

Example [cont.]



- Complete Path Coverage

  $[n_0,n_1,n_2,n_8]$ $[n_0,n_1,n_2,n_7,n_8]$

  $[n_0,n_3,n_2,n_8]$ $[n_0,n_3,n_2,n_7,n_8]$

  infinite number of paths involving loops!

- Alternative to Complete Path Coverage when there are loops!
- Specified Path Coverage [SPC]
  - TR contains a set S of test paths, where S is supplied as a parameter (judgement call by test engineer).
  - For instance, for each loop, S contains
    - a path that bypasses the loop,
    - a path that takes the loop once,
    - a path that takes the loop a reasonable number of times (say, 5).

    (if these are feasible)

Example [cont.]

- T_{NC} does not satisfy the All-Edges Criterion.
  - It misses edges $[n_2,n_8]$ and $[n_4,n_5]$.
- T_{EC} does not satisfy the Edge-Pair Criterion.
  - It misses $[n_0,n_3,n_2]$, $[n_1,n_2,n_7]$, $[n_3,n_2,n_8]$, $[n_5,n_3,n_4]$.
- T_{EPC} does not satisfy the Prime-Path Criterion.
  - It misses $[n_0,n_3,n_2,n_8]$, $[n_0,n_3,n_4,n_5]$, $[n_4,n_5,n_3,n_2,n_7,n_8]$, $[n_4,n_6,n_5,n_3,n_2,n_8]$, $[n_4,n_6,n_5,n_3,n_2,n_7,n_8]$.
- T_{PPC} does not satisfy the Complete-Path Criterion.
  - It misses many loops involving nodes n_3, n_4, n_5, and n_6.
- The Simple-Round-Trip Criterion and the Complete-Round-Trip Criterion do not even exercise all the transitions
  - But they are useful when testing state machines, which are often connected.
  - Exercising round-trips (SRTC, CRTC) must be combined with another criterion that exercises at least all the edges.
- The Complete-Path Criterion if often not achievable when there are loops.
  - An alternative exists.