Software Testing, Quality Assurance and Maintenance	Winter 2010

Lecture 5 — January 13, 2010

Patrick Lam *version 2*

Let's consider an example of a test set which satisfies node coverage on D, the double-diamond graph from last time.

Start with a test case t_1; assume that executing t_1 gives the test path

$$\text{path}(t_1) = p_1 = [n_0, n_1, n_3, n_4, n_6].$$

Then test set $\{t_1\}$ does not give node coverage on D, because no test case covers node n_2 or n_5. If we can find a test case t_2 with test path

$$\text{path}(t_2) = p_2 = [n_0, n_2, n_3, n_5, n_6],$$

then the test set $T = \{t_1, t_2\}$ satisfies node coverage on D.

What is another test set which satisfies node coverage on D?

Here is a more verbose definition of node coverage.

Definition 1 *Test set T satisfies* node coverage *on graph G if and only if for every syntactically reachable node $n \in N$, there is some path p in path(T) such that p visits n.*

A second standard criterion is that of edge coverage.

Criterion 1 Edge Coverage *(EC). TR contains each reachable path of length up to 1, inclusive, in G.*

We describe edge coverage this way so that, as far as possible, new criteria in a series will subsume previous criteria.

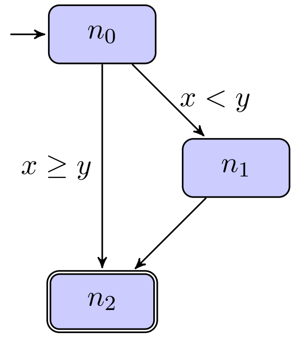
Here are some examples of paths of length ≤ 1:

Note that since we're not talking about *test paths*, these reachable paths need not start in N_0.

In general, paths of length ≤ 1 consist of nodes and edges. (Why not just say edges?)

Saying "edges" on the above graph would not be the same as saying "paths of length ≤ 1".

Here is a more involved example:



Let's define

$$\begin{aligned}
\text{path}(t_1) &= [n_0, n_1, n_2] \\
\text{path}(t_2) &= [n_0, n_2]
\end{aligned}$$

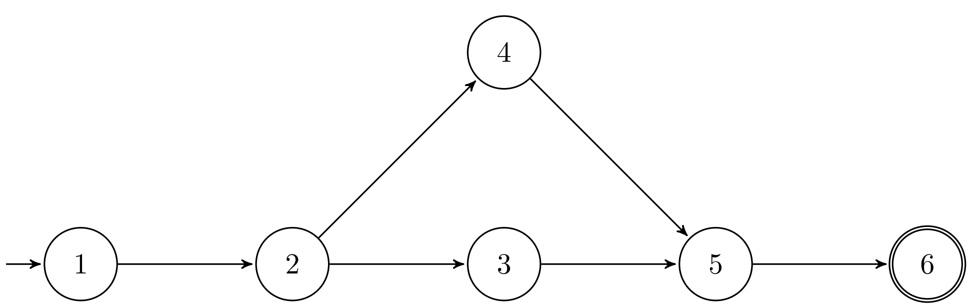
Then

$$\begin{aligned}
T_1 &= \qquad\qquad \textsf{satisfies node coverage} \\
T_2 &= \qquad\qquad \textsf{satisfies edge coverage}
\end{aligned}$$

Going beyond 1. So far we've seen length ≤ 0 (node coverage) and length ≤ 1. Of course, we can go to lengths ≤ 2, etc., but we quickly get diminishing returns. Here is the criterion for length ≤ 2.

Criterion 2 Edge-Pair Coverage. *(EPC) TR contains each reachable path of length up to 2, inclusive, in G.*

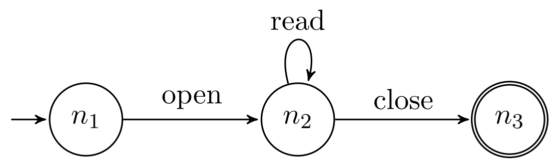
Here's an example.



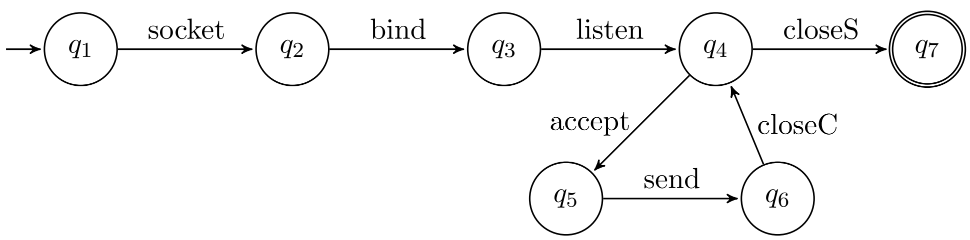
- nodes:
- edges:
- paths of length 2:

Further properties of paths

Let's now move beyond control-flow graphs and think about a different type of graph. For instance:



or perhaps



These graphs are finite state machines rather than control-flow graphs. Our motivation will be to set up criteria that visit round trips in cyclic graphs. We first set up a few definitions:

Definition 2 *A path is* simple *if no node appears more than once in the path, except that the first and last nodes may be the same.*

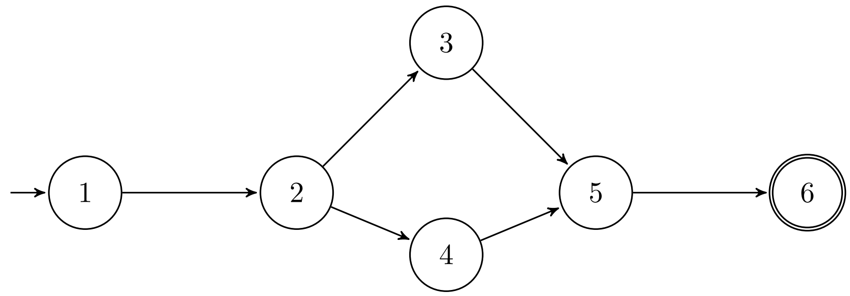
In the graphs above, some simple paths are:

but not:

Some properties of simple paths:

- no internal loops;
- can bound their length;
- can create any path by composing simple paths; and
- many simple paths exist (too many!)

Because there are so many simple paths, let's instead consider *prime* paths, which are simple paths of maximal length. For instance, in the following graph:



- Simple paths:
- Prime paths:

Definition 3 *A path is* prime *if it is simple and does not appear as a proper subpath of any other simple path.*

Criterion 3 Prime Path Coverage. *(PPC)* TR *contains each prime path in G.*

There is a problem with using PPC as a coverage criterion: a prime path may be infeasible but contain feasible simple paths.

Example:

One could replace infeasible prime paths in TR with feasible subpaths, but we won't bother.

Returning to our motivation of covering non-control-flow graph automata, let's talk about round trips.

Definition 4 *A* round trip *path is a prime path of nonzero length that starts and ends at the same node.*

Criterion 4 Simple Round Trip Coverage. *(SRTC)* TR *contains at least one round-trip path for each reachable node in G that begins and ends a round-trip path.*

Criterion 5 Complete Round Trip Coverage. *(CRTC)* TR *contains all round-trip paths for each reachable node in G.*

Here are two more path coverage criteria.

Criterion 6 Complete Path Coverage. *(CPC)* TR *contains all paths in G.*

Note that CPC is impossible to achieve for graphs with loops.

Criterion 7 Specified Path Coverage. *(SPC)* TR *contains a specified set S of paths.*

Specified path coverage might be useful for encoding a set of usage scenarios.