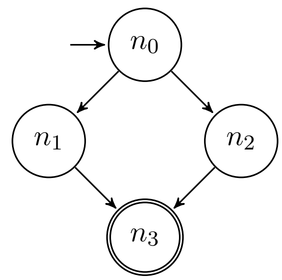
Software Testing, Quality Assurance and Maintenance	Winter 2010

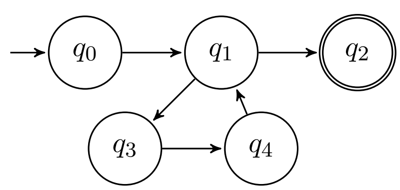
Lecture 6 — January 15, 2010

Patrick Lam *version 3*

Prime Path Coverage versus Complete Path Coverage.

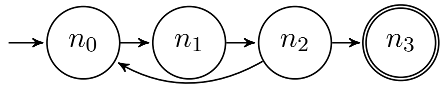


- Prime paths:
- path(t_1) =
- path(t_2) =
- $T_1 = \{t_1, t_2\}$ satisfies both PPC and CPC.



- Prime paths:
- path(t_3) =
- path(t_4) =
- $T_1 = \{t_3, t_4\}$ satisfies both PPC but not CPC.

Specifying versus meeting test requirements. Consider this graph.



The following simple (and loop-free) path is, in fact, prime:

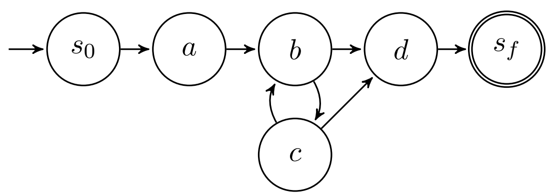
$$p =$$

PPC includes this path as a test requirement. The *test path*

meets the test requirement induced by p even though it is not prime. Note that a test path may satisfy the prime path test requirement even though it is not prime.

Sidetrips and Detours

Let's expand the notion of a tour. Recall that our definition of "tour" is strict: for path p to tour subpath $p' = [n_1, n_2, n_3]$, p must contain exactly the sequence n_1–n_2–n_3 in that order. There may not be any extra edges or nodes in p if it is to tour p'.



The path

$$q = [a, b, d]$$

is never toured by any path containing c because:

For instance, the paths

$$\begin{aligned} p &= [s_0, a, b, c, b, d, s_f] \\ p' &= [s_0, a, b, c, d, s_f] \end{aligned}$$

don't tour q using our definition of "tour". But they does tour q in some more general sense. Hence we'll define "sidetrips", which leave and return to the subpath on the same node (e.g. p) and "detours", which leave and return to the subpath on the next node (e.g. q).

For the following definitions, assume that q is simple.

Definition 1 *(Tour) Test path p* tours *subpath q iff q is a subpath of p.*

Definition 2 *(Tour with sidetrips) Test path p* tours *subpath q* with sidetrips *iff every edge in q is also in p, in the same order.*

Definition 3 *(Tour with detours) Test path p* tours *subpath q* with detours *iff every node in q is also in p, in the same order.*

Example of a sidetrip.

Example of a detour.

Missing edges can (even more than nodes) change test behaviour significantly.

Refining coverage criteria. We could define each graph coverage criterion and explicitly include the kinds of tours allowed, e.g.

- prime paths with tours;
- prime paths, sidetrips allowed;
- prime paths, detours allowed.

However, we won't do this. We do need sidetrips sometimes, or too many TRs are infeasible.

For instance,

We'd rather not use sidetrips when we don't have to, so we use:

Definition 4 *(Best Effort Touring) Let TR_{tour} be the subset of test requirements that can be toured and $TR_{sidetrip}$ the subset of test requirements that can be toured with sidetrips. A set T of test paths achieves* best effort touring *if for every path p in TR_{tour}, some path in T tours p (directly) and for every path p' in $TR_{sidetrip}$, some path in T tours p either directly or with a sidetrip.*

Note that $\mathrm{TR}_{\mathrm{tour}} \subseteq \mathrm{TR}_{\mathrm{sidetrip}}$. We also drop infeasible test requirements from this definition.

Best-effort touring meets as many TRs as possible, each in the strictest possible way. We'll use best-effort touring for our subsumption chart, which is coming up later.