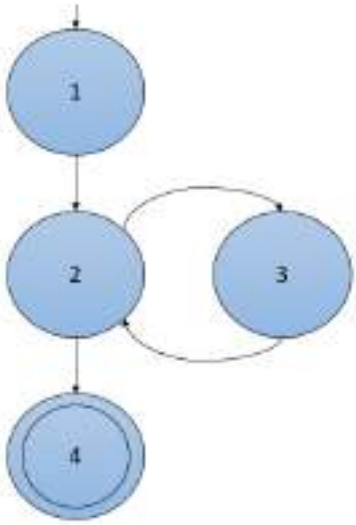
Prime Path Coverage (PPC) Tutorial



Using the graph above, PPC will be applied to it. But before that a discussion of notation that will be present in this tutorial will be quickly discussed.

!: represents that the simple path cannot be extended further. The two possible reasons for this is that either it was reached a final node or if it is extended by one more it would create an internal loop/cycle, which means it would no longer be a simple path.

*: represents that the simple path is a loop/cycle and should not be extended further because then the simple path would have an internal loop/cycle, which simple paths cannot have.

1. Start with all nodes of the graph created as a list as below (Note L # is just denoting length of simple path):

L1
[1]
[2]
[3]
[4]!

2. Expand each of the paths without a symbol to another node via a valid edge and see if any of the simple paths need to be marked with an ‘!’ or ‘*’.

L1	L2
[1]	[1, 2]
[2]	[2, 3]
[3]	[2, 4]!
[4]!	[3, 2]

3. Continue step 2 until all simple paths have a symbol next to them (i.e. they cannot be expanded further)

Iteration 2 of Step 2:

L1	L2	L3
[1]	[1, 2]	[1, 2, 3]!
[2]	[2, 3]	[1, 2, 4]!
[3]	[2, 4]!	[2, 3, 2]*
[4]!	[3, 2]	[3, 2, 3]*
		[3, 2, 4]!

NOTE [1, 2, 3] cannot be extended any further because the only edge that it can go is to node to creating the path [1, 2, 3, 2]. However, this gives us an internal loop and simple paths cannot have internal loops, thus [1, 2, 3] cannot be extended further.

4. Cross out all paths that are a sub path of another. A useful technique is to start from longest length and go to shortest length.

L1	L2	L3
~~[1]~~	~~[1, 2]~~	[1, 2, 3]!
~~[2]~~	~~[2, 3]~~	[1, 2, 4]!
~~[3]~~	~~[2, 4]!~~	[2, 3, 2]*
~~[4]!~~	~~[3, 2]~~	[3, 2, 3]*
		[3, 2, 4]!

5. The remaining paths are the simple paths that make up the test requirements that are needed to satisfy PPC for the given graph.

TR = {[1, 2, 3], [1, 2, 4], [2, 3, 2], [3, 2, 3], [3, 2, 4]}

6. If test paths are needed to be created, expand the given path as needed.

Reminder a testpath must start with an initial node and end with a final node.
Multiple test requirements can be satisfied by one test path.

Example: [1, 2, 3] made in a test path is [1, 2, 3, 2, 4]. This test path would satisfy [1, 2, 3], [2, 3, 2], and [3, 2, 4].

Repeat this until your set of testpaths satisfy all of the test requirements.