```
Algorithm 1: Path-insensitive Inner Component Model Construction
Input: app program source code
Output: path-insensitive inner component model, including nodes
        and edges for each component
1  begin
2      auxNodes <- geneAuxNodes(comp)
3      entireLife <- constructLifecycleGraph(comp)
4      nodes <- entireLife.nodes
5      edges <- entireLife.edges
6      lifeNodes, nonLifeNodes <- getAllValidCallbacks()
7      foreach hiddenNode in (entireLife.nodes - lifeNodes) do
8          edges <- edges U geneEdges(hiddenNode.outers, hiddenNode.inners)
9          nodes <- nodes - hiddenNode
10         edges <- edges - hiddenNode.edges
11     end
12     RA <- getRA()
13     foreach ra in RA do
14         if ra.ivr Blongs lifeNodes and ra.ive blongs nonLifeNodes do
15             nodes <- nodes U ra.ive
16             edges <- edges U geneEdge(auxNodes.activeStart, ra.ive)
17             edges <- edges U geneEdge(ra.ive, auxNodes.activeEnd )
18         elif ra.ivr Blongs nonLifeNodes and ra.ive blongs nonLifeNodes do
19             nodes <- nodes U ra.ive
20             edges <- edges U geneEdge(ra.ivr, ra.ive)
21             edges <- edges U geneEdge(ra.ive, auxNodes.activeEnd )
22         else pass
23         end
24     end
25 end
```