Building Symbolic Execution DAGs

Data Structures:

. nodes have label and a (possibly empty) set of identifiers

. associative mapping from operator to list of nodes representing that operator

. associative mapping from variables to nodes representing those variables' values "currently"

. associative mapping from constants to nodes representing those values

. each statement is one of the three forms:

   a. x = y op z
   b. x = op y
   c. x = y

Algorithm:

Complete steps 1-3 for each 3-addr stmt

1. If node(y) not defined make a new node for y. If node(z) not defined make a new node for z.

2. If there is "op" node with left child of node(y) and right child of node(z), add x to the set of identifiers for that node. If not, create a new node for "op", make node(y) the left child, node(z) the right and add x to the list of identifiers

3. change node(x) to point to the node created (or found) in steps 1 and/or 2.