Paths - Software Testing

As a preview of how we will use graph theory, the structural approaches to testing, all center on types of paths in a program. Here we define interpretation-free in a graph.

Definition

A path is a sequence of edges such that for any adjacent pair of edges e, e in the sequence, the edges share a common (node) endpoint.

Paths can be described either as sequence of edges or as sequences of nodes; the node sequence choice is more common.

Some paths in the graph in figure.

Path	Node Sequence	Edge Sequence
Between n_1 and n_5	n_1, n_2, n_5	e_1, e_4
Between n_6 and n_5	n_6, n_4, n_1, n_2, n_5	e_5, e_2, e_1, e_4
Between n_3 and n_2	n_3, n_4, n_1, n_2	e_3, e_2, e_1

Paths can be generated directly from the adjacency matrix of graph using a binary form of matrix multiplication and addition. In our continuing example. Edge e_1 is between nodes n_1 and n_2 and edge e_4 is between nodes n_2 and n_5. In the product of the adjacency matrix with itself the element in position (1, 2) forms a product with the element in position (2, 5) an element in position. (1, 5), which corresponds to the two-edge path between n_1 and n_5. If we multiplied the product matrix by the original adjacency matrix again, we would get all three edge paths and so on. At this point, the pure math folks go into a long digression to determine the length of the longest path in a graph; we will not bother. Instead, we focus our interest on the fact that paths connect "distant" parts of a graph.

The graph in figure predisposes a problem. It is not completely general, because it does not show all the situations that might occur in a graph. In particular no paths exist in which a node occurs twice in the path. If it did, the path would be a loop (or circuit). We could create a circuit by adding an edge between nodes n_3 and n_6.