Proof of Correctness for Bipartite Test Algorithm

Christopher Wood

12/21/11

1 Bipartite Test Algorithm

Input: The adjacency matrix for a graph, G.

Output: Boolean value indicating whether or not G is bipartite.

Description: The idea of the algorithm is to perform a breadth-first-search (BFS) on the input graph G in order to visit every vertex in search of an odd cycle. The BFS can be applied recursively on the tree by selecting an initial start vertex, v_0, which is set to the current vertex, and then iteratively visiting each of the neighbors of the current vertex until all vertices have been visited. At each vertex, the depth of the BFS traversal is assigned to the vertex in such a way so that v_0 would receive a label of 0. Using these visited labels, if at any point during the BFS a vertex is visited more than once, the labels of the immediate predecessor vertices of the already-visited vertex are compared for even and odd parity. If the parity of these two vertices matches, then the cycle that has been detected is even length, and the BFS traversal continues. If the parity of these two vertices does not match then an odd cycle has been detected, and the algorithm terminates and returns false. Once all vertices in $V(G)$ have been visited and no odd cycle has been detected, the algorithm terminates and returns true.

2 Theorem

Applying the BTA algorithm on a given graph G will indicate whether or not G is bipartite.

2.1 Proof

TODO: walk through algorithm