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 depth-first-search (DFS) on the input graph G in order to visit every vertex in search of an odd cycle. The DFS can be applied recursively on the tree by selecting an initial start vertex, v_0, which is set to the current vertex, and then recursively visiting each of the neighbors of the current vertex until all vertices have been visited. At each vertex, the depth of the DFS 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 DFS a vertex is visited more than once, the even/odd parity of the current DFS depth and the already-visited vertex label is compared. If this parity is even then the cycle must be even as well, and so the current DFS recursive path is stopped and the traversal continues as normal. If this parity is odd then the cycle is odd, at which point 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