Problem

An undirected graph is ***bipartite*** if its nodes may be divided into two sets so that all edges go from a node in one set to a node in the other set. Show that a graph is bipartite if and only if it doesn't contain a cycle that has an odd number of nodes. Let *BIPARTITE* = { $\langle G \rangle$ | G is a bipartite graph}. Show that *BIPARTITE* ? NL.

Step-by-step solution

Step 1 of 4

Given that

$BIPARTITE = \{ \langle G \rangle | G$ is a bipartite graph}

An undirected graph is bipartite if its nodes may be divided into two sets so that all edges go from a node in one set to a node in the other set.

Now we have to show that

- A graph is bipartite if and only if it doesn't contain a cycle that has an odd number of nodes, and
- $BIPARTITE \in NL$

Comment

Step 2 of 4

(i) First we show that if a graph is bipartite then it doesn't contain a cycle that has an odd number of nodes:

- Let *G* be a graph which is bipartite.
- Assume that *G* contain a cycle that has odd number of nodes
- $n_1, n_2, n_3 \ldots, n_{2k}, n_{2k+1}$
- By the definition of bipartite graph, as *G* is a bipartite we have n_1 is in some set *A*, and n_2 must be in some set *B*, then n_3 must be in set *A* etc.
- By induction, all the nodes with an odd subscript must be in set *A*, and all those with an even subscript must be in set *B*
- This implies that n_1 and n_{2k+1} are both in set *A*.
- This is a contradiction because they are connected.
- Thus our assumption that *G* contains a cycle that has odd number of nodes is wrong.
- Hence *G* does not contain a cycle with odd number of nodes.

Comment

Step 3 of 4

(ii) Second, if a graph doesn't contain a cycle with an odd number of nodes then the graph is bipartite:

- Suppose a graph does not contain any odd cycles.
- Pick a node, and label it *A*.
- Label all of its neighbors *B*.
- Label all of their unlabeled neighbors *A*, etc, until all nodes are labeled.
- Suppose that this construction caused two adjacent nodes *x* and *y* to have the same label.
- Then that would mean that both *x* and *y* were reached by taking an even number of steps.
- In either case, the total number of nodes traversed in getting to *x* from the start node, and getting to *y* from the start node (excluding the start node itself), is even.
- But adding the start node in makes the total number of nodes odd, contradicting the hypothesis that there were no odd cycles.
- Thus, the construction succeeds in properly dividing the nodes, so the graph is bipartite.

Comment

Step 4 of 4

(iii) Finally, we show that $BIPARTITE \in NL$

• We know that $NL = CONL$

• So if we prove that $\overline{BIPARTITE} \in NL$ then than implies that $BIPARTITE \in NL$

• $\overline{BIPARTITE}$ ={ $\langle G \rangle | G$ is a graph that contain an add cycle}

We know that

" *NL* is the class of languages that are decidable in logarithmic space on a non deterministic Turing machine "

Let *M* be the *NTM* that decides $\langle G \rangle | G$ $\overline{BIPARTITE}$ in logarithmic space. *M* can be constructed as follows:

$M =$ "on input $\langle G \rangle$:

1. Counter : = 0
2. start: = Nondeterministically select a starting node.
3. Successor := Nondeterministically select a successor of *s*.
4. while (counter $\leq$ the number of nodes)
5. If successor = start and if counter is odd accept
6. otherwise successor $:=$ nondeterministically select a successor of successor
7. If counter has increased above the number of nodes, reject."

Since, the only space used by the above algorithm is for keeping track of the node numbers 'start' and 'Successor' and keeping track of the 'counter', the algorithm clearly only uses log space on any one branch.

• There fore $\overline{BIPARTITE} \in NL$

• Because $NL = CONL, BIPARTITE \in NL$

Comment