Problem

Recall that a directed graph is ***strongly connected*** if every two nodes are connected by a directed path in each direction. Let

$$STRONGLY\text{-}CONNECTED = \{\langle G\rangle \mid G \text{ is a strongly connected graph}\}.$$

Show that $STRONGLY\text{-}CONNECTED$ is NL-complete.

Step-by-step solution

Step 1 of 4

Specified that

A directed graph is strongly connected if every two nodes are connected by a directed path in each direction.

Let $STRONGLY\_CONNETED =\{ \langle G\rangle \mid G$ is a strongly connected graph}

We have to show that $STRONGLY\_CONNETED$ is $NL-$complete

<u>*NL* -completeness:</u> A language '*B*' is *NL*_ complete if

1. $B \in NL$, and

2. Every *A* in *NL* is log space reducible to *B*.

So, to show that $STRONGLY\_CONNECTED$ is *NL* _ complete

We need to prove the 2 conditions of $NL-$completeness.

Comment

Step 2 of 4

(1) $\underline{STRONGLY\_CONNECTED \in NL}$:

We know that

"*NL* is the class of languages that are decidable in logarithmic space on non-deterministic Turing machine (NTM)„

So to prove $STRONGLY\_CONNECTED \in NL$, we need to construct a *NTM N* that decides $\overline{STRONGLY\_CONNECTED}$ in logarithmic space.

The construction of *N* is as follows:

$N_1 =$"On input $\langle G\rangle$:

1. Select two nodes *a* and *b* non-deterministically.

2. Run $PATH(a,b)$.

• If it rejects, then the graph is not strongly connected, so accept.

• Otherwise, reject.

Since storing the node numbers *a* and *b* only takes log space, and *PATH* uses only log space, so

$\overline{STRONGLY\_CONNECTED} \in NL$.

We know that $NL = CONL$.

Thereof $STRONGLY\_CONNECTED \in NL$

Comment

Step 3 of 4

(2) Next we must show that every language in *L* is log space reducible to $STRONGLY\_CONNECTED$.

We do this by reducing $PATH$ to $STRONGLY\_CONNECTED$

The $NTM\ \ N_2$ will do this procedure.

$N_2 =$ "On input $\langle G, s, t \rangle$, where *G* is a graph and *s, t* are vertices in *G*

1. Copy all of *G* onto the output tape
2. Additionally for each node *i* in *G*.
3. Output on edge from *i* to *s*.
4. Output an edge from *t* to *i.* "

Comment

Step 4 of 4

This algorithm only needs log space to store the counter for *i*

• If there is a path from *s* to *t*, then the constructed graph is strongly connected because every node can now get to every other node by going through the path $s-t$.

• If there is no path from *s* to *t*, then the graph is not strongly connected. Because the only additional edges in the constructed graph go into *s* and out of *t* , so there can be no new ways of getting from *s* to *t.*

So, from (1) and (2)

$STRONGLY\_CONNECTED$ is *NL* complete

Comment