Problem

Show that E_{DFA} is NL-complete.

Step-by-step solution

Step 1 of 1

- E_{DFA} is in $co-NL$ (a sufficient certificate is an accepted string-one always exists of length less than the number of states), thus it is in NL. Now it is required to prove that $NL-\text{hardness}$ by a reduction from $\overline{PATH}$ (which is $co-NL-\text{complete}$, and thus $NL-\text{complete}$).
- The idea of the reduction from $\overline{PATH}$ is simple. Given $G=(V,E)$ and vertices s,t and will construct a DFA having state graph is G, s is the initial state and t is the final state. Then the language of this DFA is empty if and only if t is not reachable from s $\left(\text{that is,}\langle G,s,t\rangle \notin \overline{PATH}\right)$.
- So, it sufficient to show this is possible with a log-space transducer.
- First, for each vertex, count the maximum out-degree d. Our alphabet will have $|\Sigma|=d$ (in particular, will take $\Sigma=\{1,\ldots..,d\}$ where one can write each number in $\log d=O(\log m)$ bits). This computation is easily done in log-space, by counting at each vertex and only maintaining a maximum value.
- Each vertex u will be translated (in order vertices appear on the tape) into a state, and each edge (u,v_i) (in the order the edges out of u appear on the tape) into a transition rule $\delta(u,i)=v_i$.
- If reach the end of the list of vertices without giving transitions for all d letters, add copies of the last edge visited so that u has transitions for all letters. This step is done in log-space, since at most one vertex and two edges at a time.
- It is then easy to set s to be the start state, and t to be the final state. So it is possible to do this reduction in log-space, and hence, E_{DFA} is NL-complete.

Comment