Problem

Let

$$BOTH_{\text{NFA}} = \{\langle M_1, M_2\rangle |\ M_1 \text{ and } M_2 \text{ are NFAs where } L(M_1) \cap L(M_2) \neq \emptyset\}.$$

Show that $BOTH_{\text{NFA}}$ is NL-complete.

Step-by-step solution

Step 1 of 3

Consider the given language, $BOTH_{NFA} = \{\langle M_1, M_2\rangle | M_1$ and M_2 are NFAs where, $L(M_1) \cap L(M_2) \neq \varnothing\}$. A non-deterministic log space algorithm is given here, which determines that "there are some strings which both M_1 and M_2 accepts". Consider that q_1 and q_2 are the number of states in both NFAs accordingly.

M = "On input $\langle M_1, M_2\rangle$, where M_1, M_2 are NFAs:

1. Place markers on the start states of both of the NFAs.

2. Repeat $2^{q_1 \times q_2}$ times:

3. In each step select non-deterministically an input symbol and accordingly change the marker positions on M_1 and M_2's states to simulate reading of that symbol.

4. **Accept;** if some string is found in stage 2 and 3 on which both M_1 and M_2 attends any accepting state that is if at some point on input string both the markers are on accepting states of M_1 and M_2. Otherwise **Reject**."

The only space requirement for this algorithm is to store the location of the markers and the repeat counter. This requirement will be possible in logarithmic space. Therefore, $BOTH_{NFA} \in NL$. Next user need to prove that $BOTH_{NFA}$ is also NL-Hard.

Comment

Step 2 of 3

Now, consider a log space reduction from some language A in NL to $BOTH_{NFA}$. For this purpose, consider a Nondeterministic TM M which decides A in $O(\log n)$.

• Given an input string ω, construct two different NFAs M_1 and M_2 in log space which accepts ω if and only if M accepts ω.

• The states of M_1 and M_2 are the configurations of M on ω. For configuration q_1 and q_2 of M on ω, the pair (q_1, q_2) are two states both in M_1 and M_2 if q_2 is the possible next configuration of M starting from q_1.

• The above statement shows, if M's transition function follows that state q_1 together with its state symbols under corresponding input symbol.

• Also, each of the work tape heads can find the next state and also the head actions to take from q_1 to q_2, then q_1 and q_2 are two states in both M_1 ands M_2.

This configuration reduces A to $BOTH_{NFA}$ because whenever M accepts an input, some its computational branch accepts some state transition in both of the machines M_1 and M_2.

Comments (1)

Step 3 of 3

Again conversely, if M_1 and M_2 accepts a string ω some computational branch follows in M and accepts ω. Now to show that the given reduction works in log space, consider the log space transducer that outputs $\langle M_1, M_2\rangle$ on input ω.

• Consider describing M_1 and M_2 by listing their states. Here each state is a configuration of M on input of ω and it can be denoted by $c \log n$ space where c is some constant.

• Now, ıt can be said, transducer attends via all possible strings of length $c \log n$ sequentially and checks if each one is a legal configuration of M on ω and outputs only those which passes the checking.

• Hence, it is sufficient to verify in log space that any configuration q_1 of M on input ω can find another configuration q_2. This is because, the transducer only needs to check for the actual tape contents under the each head locations given in q_1 and used to determine that M's transition function would produce q_2 as a result.

• The transducer verifies all pairs (q_1, q_2) in each turn to find the states in M_1 and M_2. **Hence it is proved that** $BOTH_{NFA}$ **is** $NL-Complete$.

Comment