TOC PROBLEM SET-5

SIDDHANT CHAUDHARY
BMC201953

1. Let $\mathcal{A}$ be a DFA with n states that accept a word of length $\geq n$, prove that it accepts a word of length m where $n \leq m < 2n$.

Solution: Let w be a word that $\mathcal{A}$ accepts, and suppose $|w| = m' \geq n$. If $n \leq m' < 2n$, then we are done. So, we assume that $m' \geq 2n$. We first show that

$$w = xyz$$

where $0 < |y| \leq n$ such that xz is also accepted by $\mathcal{A}$. Let $w = a_1 a_2 ... a_{m'}$, and let

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} ... \xrightarrow{a'_m} q_{m'}$$

be an accepting run for a. Now, $q_0, ...q_{m'}$ are $m' + 1 \geq 2n$ states. So by the pigeonhole principle, there are $0 \leq i < j \leq m'$ such that $q_i = q_j$. Now, let x be the word in the run $q_0 \xrightarrow{x} q_i$, y be the run $q_i \xrightarrow{y} q_j$ and let z be the run $q_j \xrightarrow{z} q_{m'}$. Since $i \neq j$, $|y| > 0$, and we can write

$$w = xyz$$

and observe that xz is acceptable, since the run is

$$q_0 \xrightarrow{x} q_i \xrightarrow{z} q'_m$$

Now, let y be such a word of *minimal* length. We claim that $0 < |y| \leq n$. If $|y| > n$ was true, then we could apply the pigeonhole principle in the run for y again, obtaining a smaller such y, a contradiction.

Finally, because $0 < |y| \leq n$, observe that

$$m' > |xz| = |xyz| - |y| \geq m' - n \geq n$$

and hence we have found a *smaller word* of size $\geq n$ that is accepted by $\mathcal{A}$. So, unless the length of the word is less than $2n$, we can keep repeating this process, until we get a word whose length is in the range $[n, 2n)$. This completes the proof.

2. Are the following languages regular or not? Explain each answer with proof.

(a) $L = \{0^k u 0^k | k \in \mathbb{N}, u \in \{0,1\}\}^*$. We show that this language is regular. We claim that L is that set of all those words which start and end with a zero. To see this, suppose w is any such word. Consider the largest prefix of w which only contains 0's, and consider the largest suffix of w that only contains $0'$s. Let k be the minimum of the lengths of this prefix and suffix. so that $k > 0$. So, w is of the form

$$0^k u 0^k$$

and hence $w \in L$. Conversely, if $w = 0^k u 0^k$ for some $u \in \{0,1\}^*$ and $k > 0$, then w starts and ends with a 0. So, L is described by the regular expression

$$00^*(0+1)^*0^*0$$

Date: September 2020.

so that L is regular.

(b) $L = \{1^k y | k \in \mathbb{N}, y \in \{0,1\}^*$ and y has $\geq k$ 1's$\}$. We show that L is regular in this case as well. We claim that L is the set of all words that start with a 1 and contain atleast two $1'$s. First, suppose $w \in L$, so that $w = 1^k y$ for $k > 0$, and where $|y|_1 \geq k$. Clearly, w starts with a 1 and contains atleast two $1'$s. Conversely, suppose w starts with a 1 and contains atleast two $1'$s. So,

$$w = 1y$$

where y contains atleast one 1, and hence $w \in L$. So, the regular expression for L is

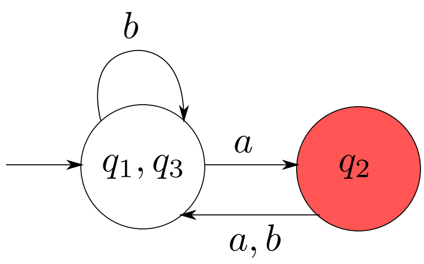
$$L = 1(0+1)^* 1(0+1)^*$$

and hence L is regular.

3. Minimise the following DFAs using partition refinement.

Solution: We refer to the images given in the problem set.

(a) Observe that q_1 and q_2 cannot be merged, because starting at q_1, the word a is accepted, but the same is not true for q_2. For the same reason, q_2 and q_3 cannot be merged. However, q_2 and q_3 can be merged together, and the result of merging is the following.



(b) We claim that the given DFA is already the minimal DFA. q_1 and q_2 cannot be merged, because starting at q_1 a is accepted, but the same is not true for q_2. For the same reason, q_1, q_3 and q_1, q_4 cannot be merged.

q_2, q_4 and q_3, q_4 cannot be merged because q_4 is a final state, and hence accepts ϵ.

Finally, q_2 and q_3 cannot be merged because starting at q_3, ba is accepted, but the same is not true for q_2. Hence, the given DFA is already isomorphic to the Nerode DFA.

Update: This technique, though it works, it *not* partition refinement.

4. Construct Nerode automata for $L_1 = a(abb)^* + b$ and $L_2 = (a + b^*)a^*b^*$.

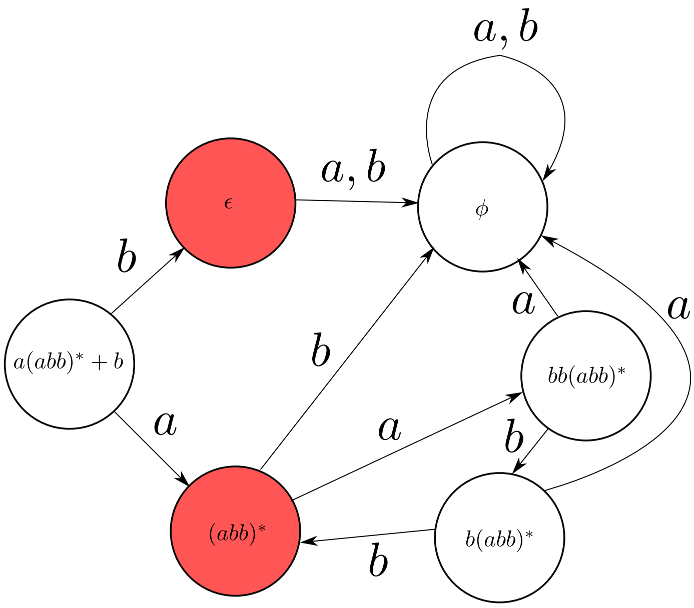
Solution: Since both languages are written using regular expressions, they are regular. So, they have finitely many quotients. To compute the Nerode Automata, we compute the quotients of L. Here, we use the formula

$$(uv)^{-1}L = v^{-1}(u^{-1}L)$$

(a) $L = a(abb)^* + b$. We have the following:

$$\begin{aligned}
\epsilon^{-1}L_1 &= L_1\\
a^{-1}L_1 &= (abb)^*\\
b^{-1}L_1 &= \epsilon\\
(aa)^{-1}L_1 &= bb \cdot a^{-1}L_1\\
(ab)^{-1}L_1 &= (ba)^{-1}L_1 = (bb)^{-1}L_1 = \phi\\
(aab)^{-1}L_1 &= b \cdot a^{-1}L_1
\end{aligned}$$

and it is easy to see that these are all the quotients. The Nerode Automaton for L_1 is given below (the leftmost state is the starting state).

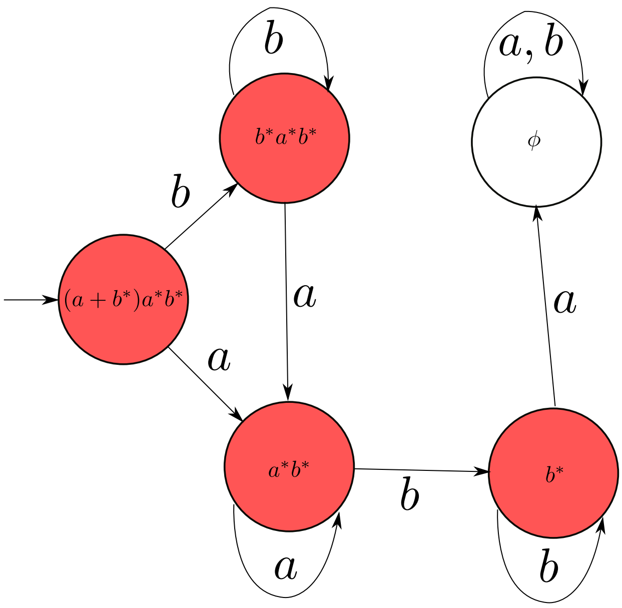


(b) $L_2 = (a + b^*)a^*b^*$. We follow the same approach as above.

$$\begin{aligned}
\epsilon^{-1}L_2 &= L_2\\
a^{-1}L_2 &= a^*b^*\\
b^{-1}L_2 &= b^*a^*b^*\\
(aa)^{-1}L_2 &= a^*b^*\\
(ab)^{-1}L_2 &= b^*\\
(ba)^{-1}L_2 &= a^*b^*\\
(bb)^{-1}L_2 &= b^*a^*b^*\\
(aba)^{-1}L_2 &= \phi
\end{aligned}$$

and again it is seen that all quotients of L_2 have been enumerated. So, the Nerode Automaton for L_2 is given below.

Update: This automaton is *not* minimal. Observe that $(a + b^*)a^*b^* = b^*a^*b^*$.



5. Prove that for each $n > 0$, a language B_n exists where

(1) B_n is recognizable by an NFA that has n states, and

(2) if $B_n = A_1 \cup ... \cup A_k$ for regular languages A_i, then atleast one of the A_i requires a DFA with atleast $2^{\lfloor n/k \rfloor}$ states.

Solution: Let $n > 0$, and let $\Sigma = \{a_1, ..., a_n\}$ be an alphabet containing n letters. Let

$$L = \{w | w \text{ does not contain all the letters from } \Sigma\}$$

In PSET-4 problem 1, I showed that any DFA accepting L requires atleast 2^n states. We now construct an NFA with n states that accepts L.

Make n states, say q_i for $1 \leq i \leq n$. Each q_i is initial, and each q_i has $n-1$ self loops, the only missing self loop being the one with label a_i. Also, each state is final. It is clear that this NFA accepts all those words which don't contain all letters.

Now, suppose

$$B_n = A_1 \cup ... \cup A_k$$

where each A_i is a regular language. For the sake of contradiction, suppose the minimal DFA required for A_i has r_i states, where

$$r_i < 2^{\lfloor n/k \rfloor}$$

for each $1 \leq i \leq k$. So, we can construct a DFA for the language $A_1 \cup ... \cup A_k$ using the cartesian product of these DFAs. The number of states in the cartesian product will be

$$r_1 r_2 ... r_k < (2^{\lfloor n/k \rfloor})^k \leq 2^n$$

which contradicts the fact that the minimal DFA required for B_n contains atleast 2^n states. So, there is atleast one i for which $r_i \geq 2^{\lfloor n/k \rfloor}$, completing the proof.