TOC PROBLEM SET-4

SIDDHANT CHAUDHARY
BMC201953

1. Given an alphabet $\Sigma = \{a_1, a_2, ..., a_k\}$, show that any DFA that accepts the language $L = \{w| \ w$ does not contain all the letters from $\Sigma\}$ has atleast 2^k states.

Bonus: What can you say about the minimum number of words required for an NFA?

Solution: We will show that the language L has 2^k distinct quotients, and hence it will follow that any DFA accepting L must contain atleast 2^k distinct states.

To show this, suppose A, B are two *distinct* subsets of Σ. Because they are distinct, one contains an element that is not in the other, i.e suppose wlog that

$$A - B \neq \phi$$

Let w_1 be a word formed by using all letters in A exactly once, and let w_2 be a word formed by using all letters in B exactly once (as we will see, the order in which the letters are used is not significant). We will show that

$$w_1^{-1}L \neq w_2^{-1}L$$

Consider A^c, and let w_3 be a word formed by using all letters of A^c exactly once (w_3 can be the empty word). Then, observe that

$$w_2w_3 \in L$$

because w_2w_3 does not contain any word from the set $A - B$. This shows that

$$w_3 \in w_2^{-1}L$$

However, observe that $w_1w_3 \notin L$, because all letters of Σ are present in this word. So,

$$w_3 \notin w_1^{-1}L$$

and hence

$$w_1^{-1}L \neq w_2^{-1}L$$

Now, there are 2^k possible distinct subsets of Σ, this proves the claim.

We can provide a simple bound for NFAs as well. Suppose M is an NFA accepting L containing l states. Using the subset construction, we can form a DFA out of M, which will contain 2^l states. By what we have shown,

$$2^l \geq 2^k$$

and hence $l \geq k$. So, any NFA must contain atleast k states.

Date: September 2020.

2. Construct an NFA for the language L of all words over $\Sigma = \{a, b\}$ such that the number of a's and the number of b's appearing in the word are both even. Using this NFA, give a rational expression for L.

Solution: We will construct a DFA and convert the same into a rational expression. We maintain four states, each state labeled by an ordered pair (x, y), where, $x, y \in \{0, 1\}$. The first coordinate represents the number of a's modulo 2 and the y coordinate represents the number of b's modulo 2. So, the DFA is the following (where the red state is the final state).

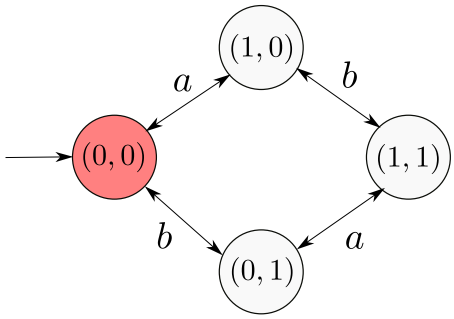
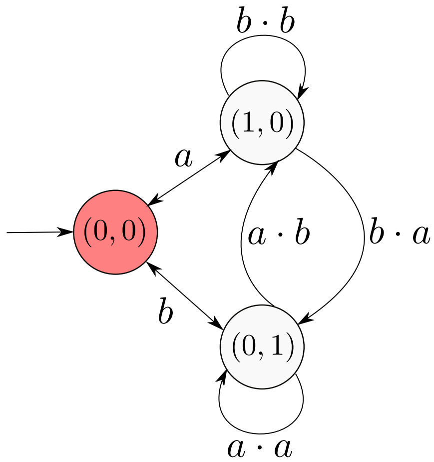
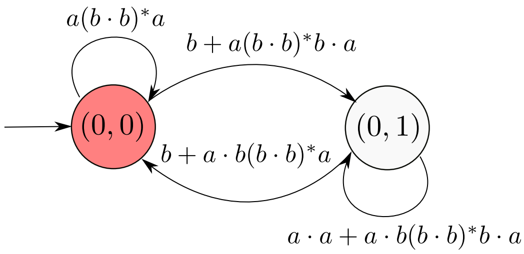


Figure 1. $(0, 0)$ is both the initial and the final state

Now, we eliminate the state $(1, 1)$, and get the following GNFA.



Next, we eliminate the state $(1, 0)$ and get the following GNFA.



Using this GNFA, we get that the regular expression for the language is

$$[(a(bb)^*a)^* + (b + a(bb)^*ba)(aa + ab(bb)^*ba)^*(b + ab(bb)^*a)]^*$$

Note: Just like in the lecture on GNFAs, I have assumed that in GNFAs, the initial state can also be the final state, and that transitions are allowed to start at the final state and go into the initial state. I mention this here because this is opposite to the definition given in the fourth problem.

3. The Dyck language L is the set of all balanced strings of square brackets, i.e it is a language over the alphabet $\Sigma = \{[,]\}$ such that for any word $u \in L$, the number of $['$s and $]'$s are equal and for any prefix of u, the number of $['$s is greater than or equal to the number of $]'$s. Show that the Dyck language is not regular.

Solution: We show that L has infinitely many quotients, thereby proving that L is not regular.

For any $n \in \mathbb{N}$, put

$$[^n = [[[[...[\ (n \text{ times})$$

We claim that if $n \neq m$, then

$$([^n)^{-1}L \neq ([^m)^{-1}(L)$$

which will complete the proof.

So suppose $n \neq m$, and without loss of generality suppose $n < m$. Then, observe that

$$[^n \cdot]^n \in L$$

which means that $]^n \in ([^n)^{-1}L$. However, observe that

$$[^m \cdot]^n \notin L$$

because $n < m$, and hence $]^n \notin ([^m)^{-1}L$. Hence, it follows that these two quotients are different, and hence it shows that L has infinitely many quotients.

4. A generalised NFA(GNFA) is similar to NFAs except in the fact that it has a single initial state i, a single final state f (i and f are distinct) and its transition function δ is of the following form

$$\delta : (Q - \{f\}) \to (Q - \{i\}) \to R$$

where R is the set of rational expressions over Σ. Prove that the class of languages recognized by GNFAs is the same for NFAs.

Solution: First, let L be a language recognized by DFA. We do the following: add a new state to the DFA, and make it initial, and put an epsilon transition from this state to the old initial state. This ensures that the initial state is not the final state anymore. Also, add a new state, make it final, and add epsilon transitions from all old final states to this one. This ensures that there is only one initial state and only one final state (and that they are distinct). Next, by state elimination, we obtain a GNFA recognizing L.

Conversely, suppose there is a GNFA recognizing some language L. We show how to convert this GNFA to an ϵ-NFA, and that will complete the proof. Note that regular expressions are composed only of the operations $+$, $\cdot$ and *. So we show how to break down each operation. Suppose in the GNFA, there is an edge of the form

$$q_1 \xrightarrow{e_1+e_2} q_2$$

Then, we add two new states to the GNFA say q and q', remove this transition, and add the following transitions:

$$q_1 \xrightarrow{\epsilon} q \xrightarrow{e_1} q_2$$
$$q_1 \xrightarrow{\epsilon} q' \xrightarrow{e_2} q_2$$

Next, if there is an edge of the form

$$q_1 \xrightarrow{e_1 \cdot e_2} q_2$$

then we add a new state q to the GNFA, remove this transition, and add the following transitions

$$q_1 \xrightarrow{e_1} q \xrightarrow{e_2} q_2$$

Finally, if there is an edge of the form

$$q_1 \xrightarrow{(e)^*} q_2$$

we add a new state q to the GNFA, remove this transition, and add the following transitions

$$q_1 \xrightarrow{\epsilon} q \xrightarrow{e} q \xrightarrow{\epsilon} q_2$$

i.e, we add a self-loop. We keep repeating this process untill all expressions reduce to single letter expressions. Note that, at every stage, we are preserving the language that is recognized. So, after a finite number of these operations, we get an NFA accepting L. This completes the proof.

5. The set of star-free languages S over an alphabet Σ is defined as the smallest set of languages such that it contains the single-ton languages ($\{a\}$ for all $a \in \Sigma$), ϕ and is closed under union, concatenation, and complementation.

(1) Is Σ^* star-free?
(2) Is a^* star free? Justify.
(3) Is $(ab)^*$ star free? Justify.

Solution:

(1) Observe that ϕ is star-free, and since star free languages are closed under complementation, it follows that $\Sigma^* = (\phi)^c$ is also star-free.
(2) Yes, a^* is star free. Suppose the alphabet is

$$\Sigma = \{a_1, a_2, a_3, ..., a_k\}$$

and suppose $a = a_1$. Observe that $(a^*)^c$ is the language consisting of all words which contain atleast one letter from $\{a_2, a_3, ..., a_k\}$. So, we can write

$$(a^*)^c = (\Sigma^* \cdot a_2 \cdot \Sigma^*) \cup (\Sigma^* \cdot a_3 \cdot \Sigma^*) \cup ... \cup (\Sigma^* \cdot a_k \cdot \Sigma^*)$$

Now, Σ^* is star-free, and by using the fact that this set is closed under concatenation, we see that

$$\Sigma^* \cdot a_i \cdot \Sigma^*$$

is star-free for every $2 \leq i \leq k$. So, $(a^*)^c$ being a finite union of star-free languages is star-free, and hence a^* is star-free.
(3) Yes, $(ab)^*$ is also star-free. First, because this set is closed under complementation, it is also closed under intersection by De-Morgan's Law. By part (2),

$$\{\epsilon\} = a^* \cap b^*$$

is also star-free. Now, let L_1 be the language of all words that start with a. Observe that

$$L_1^c = \epsilon \cup (\Sigma - \{a\}) \cdot \Sigma^*$$

Now, Σ^* is star free, and $\Sigma - \{a\}$, being finite, is also star free. We have just seen that ϵ is star-free. So, L_1^c is also star free, and hence L_1 is star free. Next, let L_2 be the set of all languages that end with a b. We have

$$L_2^c = \epsilon \cup \Sigma^* \cdot (\Sigma - \{b\})$$

and by similar reasoning as above, we have that L_2^c is also star free. Let L_3 be the language of all words that do not contain aa. Now,

$$L_3^c = \Sigma^* \cdot aa \cdot \Sigma^*$$

and hence L_3^c is star-free, implying that L_3 is star free. Similarly, le L_4 be the language of all words that do not contain bb. So, L_4 is also star free. Finally, observe that

$$(ab)^* = \epsilon \cup (L_1 \cap L_2 \cap L_3 \cap L_4)$$

and hence $(ab)^*$ is star free, since only finite intersections are involved.