Student Information

Name : Kaan Karaçanta

ID : 2448546

Answer 1

a)

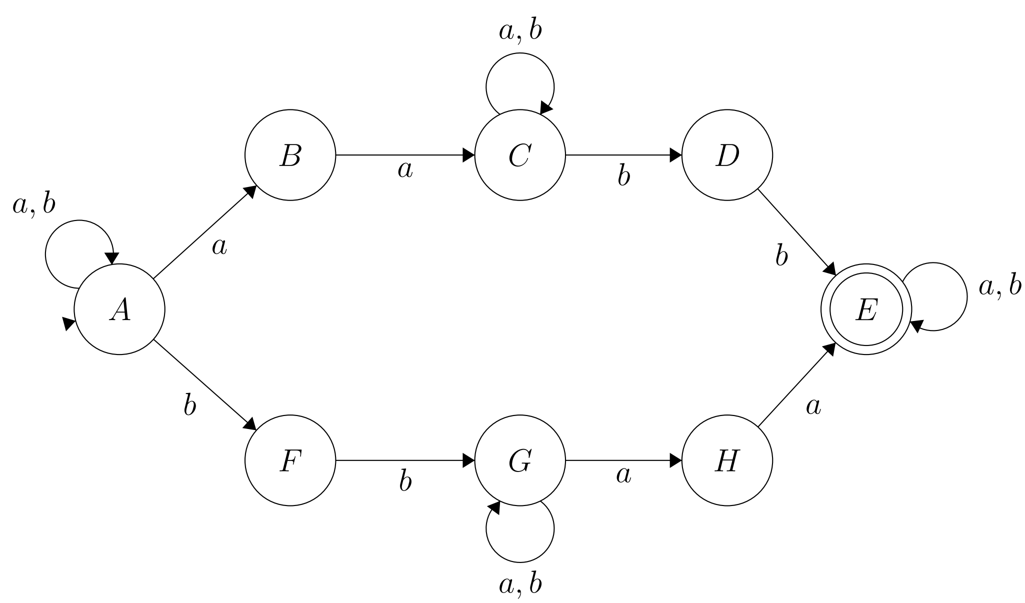
$$((a \cup b)^* aa(a \cup b)^* bb(a \cup b)^*) \cup ((a \cup b)^* bb(a \cup b)^* aa(a \cup b)^*)$$

b)

Let nondeterministic finite automata $M = (K, \Sigma, s, F, \Delta)$ where:

$$\begin{gathered}
K = \{A, B, C, D, E, F, G, H\}, \\
\Sigma = \{a, b\}, \\
s = A, \\
F = \{E\}, \\
\Delta = \{(A, a, A), (A, a, B), (A, b, A), (A, b, F), (B, a, C), \\
(C, a, C), (C, b, C), (C, b, D), (D, b, E), (E, a, E), \\
(E, b, E), (F, b, G), (G, a, G), (G, a, H), (G, b, G), (H, a, E)\},
\end{gathered}$$

and the state diagram of the NFA:



c)

Let the set of states can go without reading an input $E(A) = \{A\}, E(B) = \{B\}, ..., E(H) = \{H\}$ since there are no empty transitions. We can write the equivalent DFA's new relations starting from the initial state of the NFA, and continuing with the new states that are going to be created. In these steps, not to write the same sets and notations again and again, I will shorten some parts. For example, $\delta'(\{A, B\}, a) = \{A, B, C\}$ means that $\delta(\{A\}, a) \cup \delta(\{B\}, a)$. However, since there is no empty transition I will not combine any states with that idea, for E(A), for instance, I will just write A. Plus, I will directly write the final union form of it ({A,B,C} in that example).

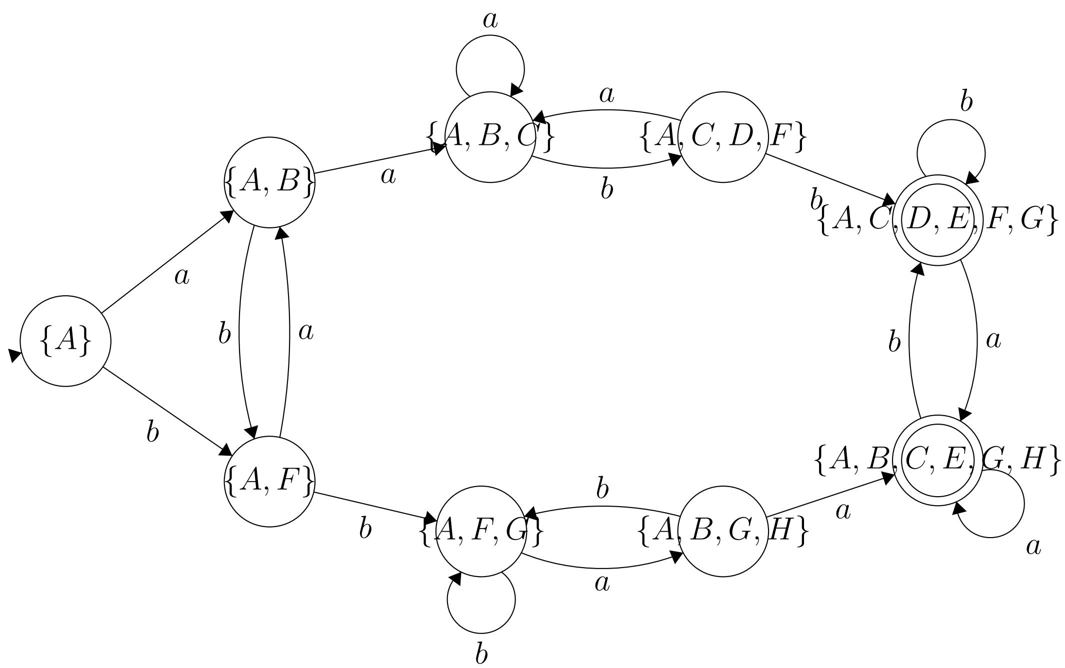
$$\delta'(\{A\}, a) = \{A, B\} \tag{1}$$
$$\delta'(\{A\}, b) = \{A, F\} \tag{2}$$
$$\delta'(\{A, B\}, a) = \{A, B, C\} \tag{3}$$
$$\delta'(\{A, B\}, b) = \{A, F\} \tag{4}$$
$$\delta'(\{A, F\}, a) = \{A, B\} \tag{5}$$
$$\delta'(\{A, F\}, b) = \{A, F, G\} \tag{6}$$
$$\delta'(\{A, B, C\}, a) = \{A, B, C\} \tag{7}$$
$$\delta'(\{A, B, C\}, b) = \{A, C, D, F\} \tag{8}$$
$$\delta'(\{A, F, G\}, a) = \{A, B, G, H\} \tag{9}$$
$$\delta'(\{A, F, G\}, b) = \{A, F, G\} \tag{10}$$
$$\delta'(\{A, C, D, F\}, a) = \{A, B, C\} \tag{11}$$
$$\delta'(\{A, C, D, F\}, b) = \{A, C, D, E, F, G\} \tag{12}$$
$$\delta'(\{A, B, G, H\}, a) = \{A, B, C, E, G, H\} \tag{13}$$
$$\delta'(\{A, B, G, H\}, b) = \{A, F, G\} \tag{14}$$
$$\delta'(\{A, C, D, E, F, G\}, a) = \{A, B, C, E, G, H\} \tag{15}$$
$$\delta'(\{A, C, D, E, F, G\}, b) = \{A, C, D, E, F, G\} \tag{16}$$
$$\delta'(\{A, B, C, E, G, H\}, a) = \{A, B, C, E, G, H\} \tag{17}$$
$$\delta'(\{A, B, C, E, G, H\}, b) = \{A, C, D, E, F, G\} \tag{18}$$

Let deterministic finite automata $M' = (K, \Sigma, s, F, \Delta)$ where:

$$K = \{\{A\}, \{A, B\}, \{A, F\}, \{A, B, C\}, \{A, F, G\}, \{A, C, D, F\}, \{A, B, G, H\},$$
$$\{A, C, D, E, F, G\}, \{A, B, C, E, G, H\}\}$$
$$\Sigma = \{a, b\},$$
$$s = A,$$
$$F = \{\{A, C, D, E, F, G\}, \{A, B, C, E, G, H\}\},$$

and Δ, set of all transition functions, is defined above in the subset construction part with those 18 lines. $\delta'(\{A\}, a) = \{A, B\}$ means ({A}, a, {A, B}) for example.

State diagram of this DFA:



d)

When the NFA M is given the input bbabb, as it is a NFA, there are several different moves for different parts of the input. However, I will only write the ones that will end with empty string in some state of the NFA. In other words, I will not consider the cases whose configuration has still an input different than empty string, yet the state has nowhere to move. For instance, ({B}, bb), because of that only transition the state B has with a, it is the end of the reading that input.

$$(A, bbabb) \vdash_M (A, babb)$$

$$\vdash_M (A, abb)$$
$$\vdash_M (A, bb)$$
$$\vdash_M (A, b)$$
$$\vdash_M (A, e)$$

Since A is not a final state, the automaton cannot accept it with these configurations.

Here 2 other different sets of configurations that are able to read the whole string, yet do not accept since G is not a final state either.

$$(A, bbabb) \vdash_M (A, babb)$$
$$\vdash_M (A, abb)$$
$$\vdash_M (A, bb)$$
$$\vdash_M (F, b)$$
$$\vdash_M (G, e)$$

$$(A, bbabb) \vdash_M (F, babb)$$
$$\vdash_M (G, abb)$$
$$\vdash_M (G, bb)$$
$$\vdash_M (G, b)$$
$$\vdash_M (G, e)$$

Since any of the reading scenarios do not end with a final state, the input string w' is not accepted by this NFA.

If the DFA M' is given the input bbabb, and its initial configuration is ({A},bbabb) then:

$$(\{A\}, bbabb) \vdash_M (\{A, F\}, babb)$$
$$\vdash_M (\{A, F, G\}, abb)$$
$$\vdash_M (\{A, B, G, H\}, bb)$$
$$\vdash_M (\{A, F, G\}, b)$$
$$\vdash_M (\{A, F, G\}, e)$$

As {A,F,G} is not a final state, this string w' is not accepted by the deterministic finite automaton.

Answer 2

a)

Firstly, since regular languages are closed under complement, if L_1 is regular L_2 will be regular, and if not, it will not. Therefore, we need to determine whether L_1 is regular or not. Assume that L_1 is a regular language, so for any string t of at least length n in it, we can say if $t = uvw$, where $|uv| <= l$ and $|v| >= 1$, for all $k >= 0, uv^kw$ must be in the language L_1. Since v is between u and w, and length of it will not be 0, we can say that v only consists of a's or b's or both of them.

For L_1 let v only consists of a's, so $u = a^x$, $v = a^y$, and $w = a^{(m-y-x)}b^n$ where $y > 0$, n is the number of b's in the string, and $x + y + z > n$, so this string is in the language. However, x+z can be less than n, but if we take the case where $k = 0$, the remaining string will be $a^{(m-y)}b^n$, and this does not always satisfy the language's condition because $y > 1$.

Since this is not the only option for v, we need to check the other conditions as well. The string v may consist of some a's and some b's, so let $u = a^x$, $v = a^{(m-x)}b^z$ and $w = b^{(n-z)}$. In the case of $m - x > z$, it becomes the first condition actually; that is, the power of v is 0, and because of that the newly created string will not be in the language. If $z > m - x$ then for the larger values of k we will have $a^x a^{(m-x)k} b^{zk+n-z}$, which, also, does not need to satisfy the condition since number of b's will grow faster as there are more b's in v.

Lastly, v may consist of only b's. In this case it is easier to see that after some values of k the number of b's will be more than number of a's since the number of b's are growing and the number of a's stays the same. Hence, those strings will not be in the language, so v cannot be in that form either.

We checked every condition for v in this string t since none of them is suitable, we can say that this language L_1 is not regular, so its complement, which is L_2, is not regular.

b)

In language L_4 n can take values from 1 to infinity, and it produces the same number of a's and b's, but in L_5 m or n can be any natural numbers in any order, so they can, also, be the same or different. Therefore, we can say that $L_4 \subset L_5$. Also, L_5 can be written as a^*b^* since m and n can take any values including 0. As L_5 is a superset of L_4, we can simplify the given question as $L_5 \cup L_6$.

Besides, we know that L_5 is a regular language since it can be written with the operations of regular expressions (Kleene star and concatenation), and with the elements of the alphabet, which are a and b. Likewise, L_6 is, already, given as a combination of some regular expressions with Kleene star, concatenation and parenthesis, so it is a regular language too. Since union of 2 regular languages is regular, and L_5 and L_6 are a regular language, so this union is a regular language.