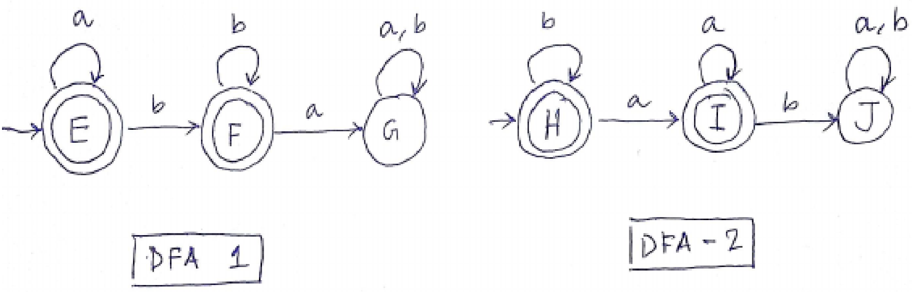
Tutorial 2

Formal Language and Automata Theory

January 12, 2023

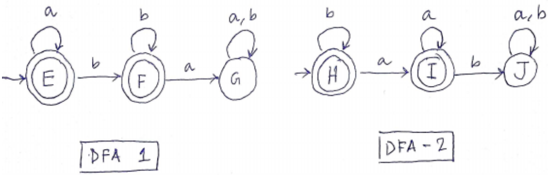
Question 1

Compute the product of DFA 1 and DFA 2.



Question 1

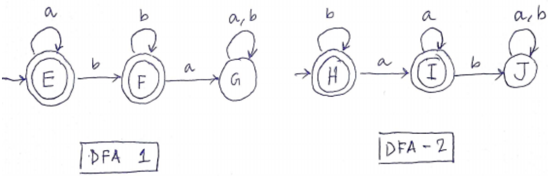
Compute the product of DFA 1 and DFA 2.



Hint 1: Start with combining states by combining the transition

Question 1

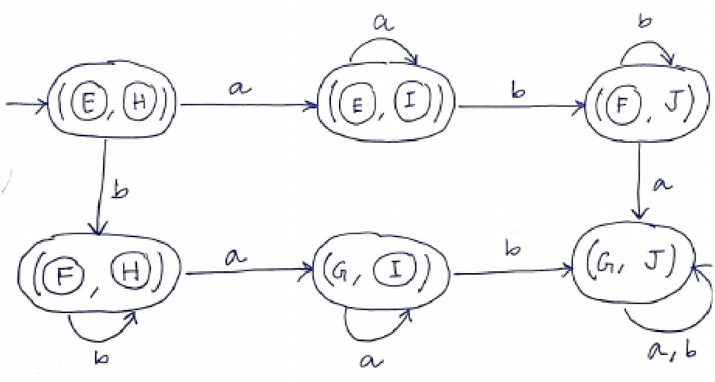
Compute the product of DFA 1 and DFA 2.



Hint 2: For example, combine E and H states. See what happens when there is transition 'a' from E and H state. Similarly see for 'b'. Continue for all other states and transition.

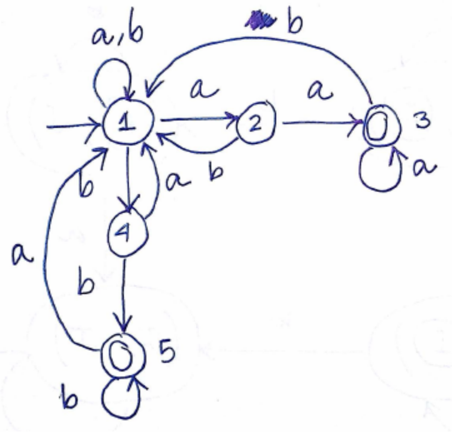
Question 1

Solution:



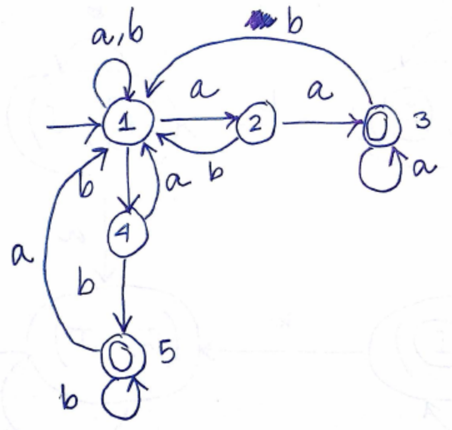
Question 2

Let A be the set of all strings that end with two consecutive a's or two consecutive b's. This is the NFA of the language A.



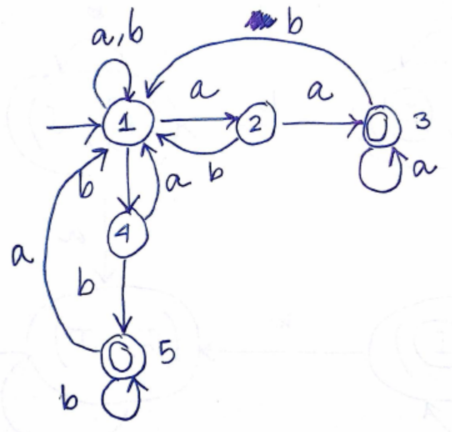
Question 2

Convert the NFA into a DFA.



Question 2

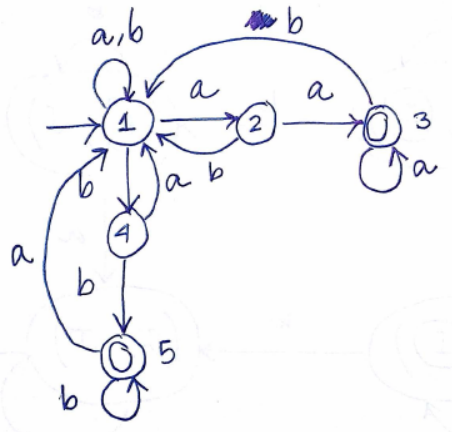
Convert the NFA into a DFA.



Hint 1: Remember that DFA cannot have multiple states with the same input.

Question 2

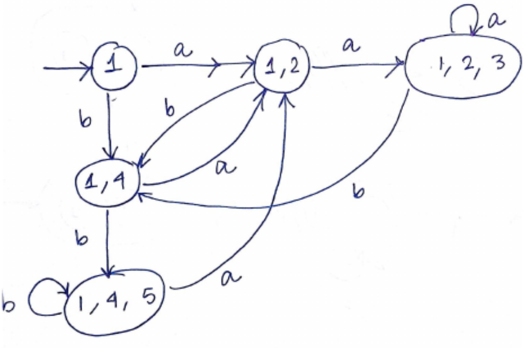
Convert the NFA into a DFA.



Hint 2: Draw the state transition table and combine the states when there are multiple states.

Question 2

Convert the NFA into a DFA.
Solution:



Question 3

Construct DFA for the following languages:
(a) L is the set of all strings containing $bbab$ as a substring

Question 3

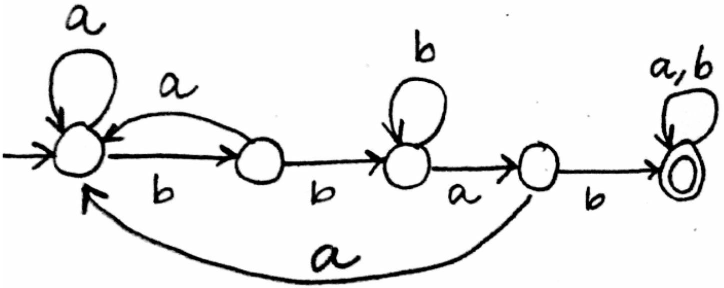
Construct DFA for the following languages:
(a) L is the set of all strings containing *bbab* as a substring
Hint 1: Start with forming the part of the DFA corresponding to the substring.

Question 3

Construct DFA for the following languages. In all parts $\Sigma = \{a, b\}$.
(a) L is the set of all strings containing *bbab* as a substring
Hint 2: What can be the remaining part of the strings containing *bbab* as substring?

Question 3

(a) L is the set of all strings containing *bbab* as a substring
Solution:



Question 3

(b) $L = \{a^n \mid n \geq 0, n \neq 4\}$

Question 3

(b) $L = \{a^n \mid n \geq 0, n \neq 4\}$
Hint 1: Start with forming the part of the DFA corresponding to the less than 3 consecutive a's.

Question 3

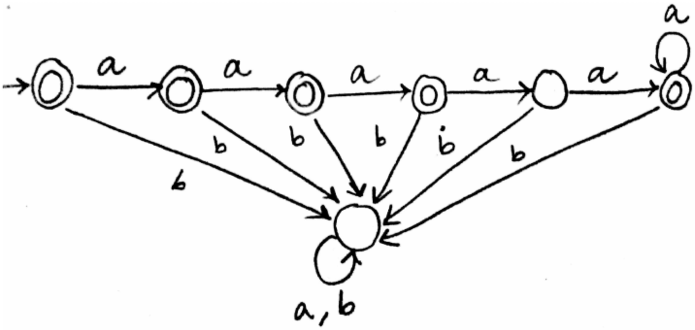
(b) $L = \{a^n \mid n \geq 0, n \neq 4\}$

Hint 2: Do the strings of the given language contain b?

Question 3

(b) $L = \{a^n \mid n \geq 0, n \neq 4\}$

Solution:



Question 3

(c) $L = \{ab^5wb^4 : w \in \{a, b\}^*\}$

Question 3

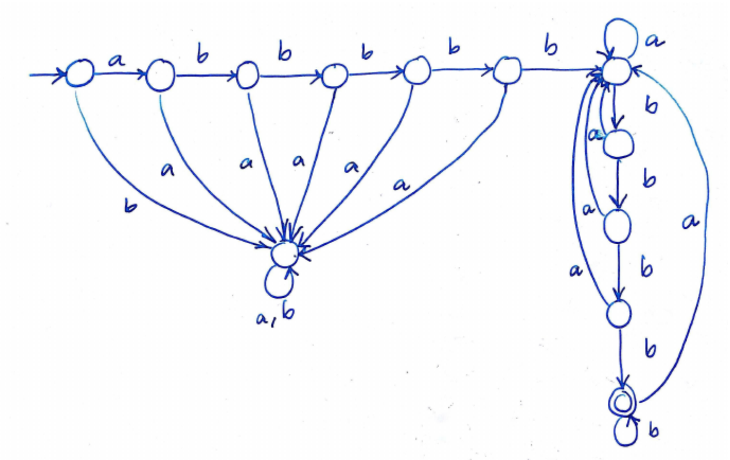
(c) $L = \{ab^5wb^4 : w \in \{a, b\}^*\}$
Hint 1: Start with forming the substring at the beginning and the end.

Question 3

(c) $L = \{ab^5wb^4 : w \in \{a, b\}^*\}$
Hint 2: What would happen if an input symbol causes violation to the prefix substring ab^5 or b^4 at the end?

(c) $L = \{ab^5wb^4 : w \in \{a, b\}^*\}$

Solution:



Question 4

Let $\Sigma = \{a, b\}$. Find a grammar that generates the language $L = \{w | n_a(w) = 2n_b(w)\}$.

Question 4

Let $\Sigma = \{a, b\}$. Find a grammar that generates the language $L = \{w | n_a(w) = 2n_b(w)\}$.

Hint 1: Think of possible ways 'a' can occur twice the number of times 'b' occurs in a string.

Question 4

Let $\Sigma = \{a, b\}$. Find a grammar that generates the language $L = \{w | n_a(w) = 2n_b(w)\}$.

Hint 2: Use two non-terminals to represent the grammar.

Question 4

Let $\Sigma = \{a, b\}$. Find a grammar that generates the language $L = \{w | n_a(w) = 2n_b(w)\}$.

Solution:

The production rules for the grammar would be:

$$S \rightarrow AaAaAbA | AaAbAaA | AbAaAaA$$
$$A \rightarrow AaAaAbA | AaAbAaA | AbAaAaA | \lambda$$

where the set of non-terminals is $V = \{S, A\}$ and S is the usual start symbol.

Question 5

Show that the grammars $S \to SS|aSb|bSa|a$ and $S \to aSb|bSa|\lambda$ are not equivalent.

Question 5

Show that the grammars $S \rightarrow SS|aSb|bSa|a$ and $S \rightarrow aSb|bSa|\lambda$ are not equivalent.
Hint 1: Generate sample strings from the given grammars and try to find the difference.

Question 5

Show that the grammars $S \to SS|aSb|bSa|a$ and $S \to aSb|bSa|\lambda$ are not equivalent.
Hint 2: What about the lengths of the strings generated by the grammars?

Question 5

Show that the grammars $S \to SS|aSb|bSa|a$ and $S \to aSb|bSa|\lambda$ are not equivalent.

Solution:

Let us call the grammars G_1 and G_2. Note that every string $\sigma \in L(G_2)$ is of length $2n$ for some n, i.e. all strings in $L(G_2)$ are of even length. However, G_1 can generate odd length strings, e.g. $S \to aSb \to aab$.