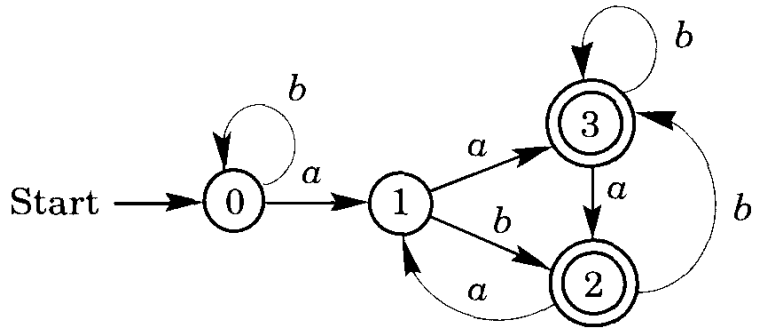
Theoretical Computer Science (M21276)

Part A/4: Deterministic and non-deterministic finite automata

(Oct 2-6, 2023)

Question 1. We are given the following DFA over the alphabet $\{a, b\}$:



(i) Decide which of the following strings are accepted by the DFA:
aaa and *aba*.

Answer: *aaa*

(ii) Give an example of three stings of length at least 5 which are accepted by the DFA:

Answer: E.g. *aabbb*, *aaabb*, *abaaa*

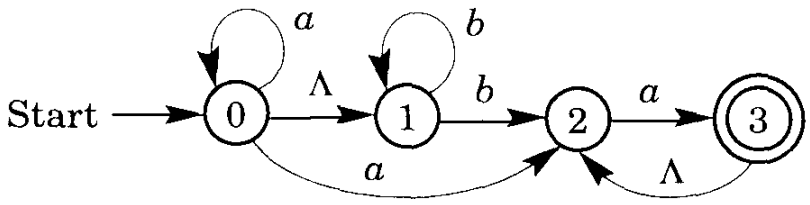
(iii) Give an example of three stings of length at least 5 which are not accepted by the DFA:

Answer: E.g. *bbbba*, *bbaba*, *bbbbba*

(iv) Write down the transition function T for the DFA.

Answer: $T(0, a) = T(2, a) = 1$,
$T(0, b) = 0$,
$T(1, a) = T(2, b) = T(3, b) = 3$,
$T(1, b) = T(3, a) = 2$, where 0 is the start state and both 2 and 3 are final states.

Question 2. We are given the following NFA over the alphabet $\{a, b\}$:



(i) Decide which of the following strings are accepted by the NFA:
aaa, ba, ab.

Answer: aaa, ba

(ii) Give an example of a string of length at least 5 which is accepted by the NFA:

Answer: E.g. $aaaaa$

(iii) Give an example of a string of length at least 5 which is not accepted by the NFA:

Answer: E.g. $aaaab$

(iv) Write down the transition function T for the given NFA:

Answer: $T(0, a) = \{0, 2\}$, $T(0, \Lambda) = \{1\}$, $T(1, b) = \{1, 2\}$, $T(2, a) = \{3\}$, $T(3, \Lambda) = \{2\}$, where 0 is the start state and 3 is the final state.

Question 3. Use your wits to construct a DFA for each of the following regular expressions. You can draw a directed graph or use a formal definition of DFA. Is it important to mention an alphabet?

(i) $a + b$ over the alphabet $\Sigma = \{a, b\}$.

Answer: States 0 (start), 1 (final), and 2.
$T(0, a) = T(0, b) = 1$ and all other transitions go to state 2.

(ii) $a + b^*$ over the alphabet $\Sigma = \{a, b\}$.

Answer: States 0, 1, 2, 3, with start state 0 and final states 0, 1, and 2.
$T(0, a) = 1$, $T(0, b) = 2$, $T(2, b) = 2$, and all other transitions go to state 3.

(iii) $ab^* + bc^*$ over the alphabet $\Sigma = \{a, b, c\}$.

Answer: States 0 (start), 1 (final), 2 (final), and 3.
$T(0, a) = T(1, b) = 1$, $T(0, b) = T(2, c) = 2$, and all other transitions go to state 3.

Question 4. Use your wits to construct an NFA for each of the following regular expressions.

(i) $a^*bc^* + ac$

Answer: States 0 (start), 1, 2 (final), 3, and 4 (final). $T(0, a) = \{1\}$, $T(1, c) = \{2\}$, $T(0, \lambda) = T(3, a) = \{3\}$, $T(3, b) = T(4, c) = \{4\}$ and all other transitions are equal $\emptyset$.

(ii) $(a + b)^*a$

Answer: States 0 (start), and 1 (final). $T(0, a) = \{0, 1\}$, $T(0, b) = \{0\}$ and all other transitions are equal $\emptyset$.

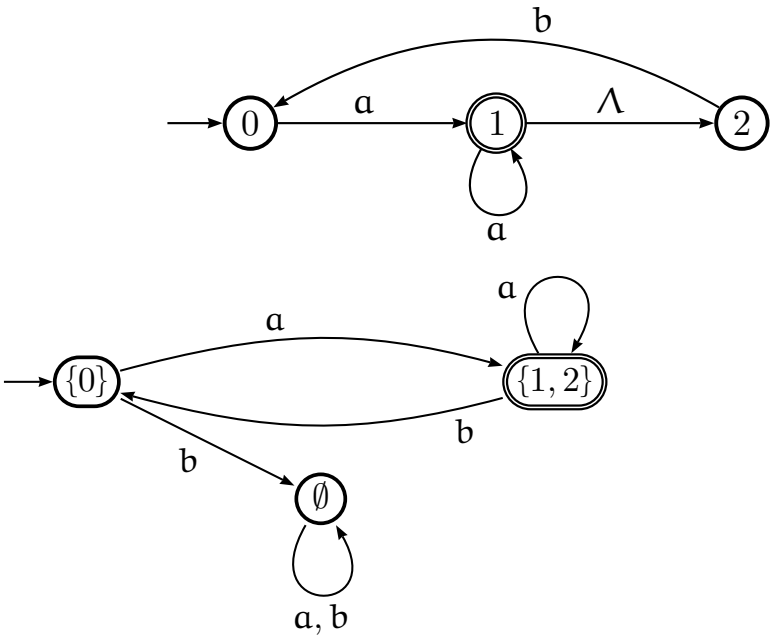
(iii) $a^* + ab$

Answer: States 0 (start), l, 2 (final), and 3 (final). $T(0, a) = \{1\}$, $T(1, b) = \{2\}$, $T(0, \Lambda) = T(3, a) = \{3\}$, and all other transitions are equal $\emptyset$.

Question 5. Use your wits (or use conversion of Q4i) to construct an DFA for the following regular expression $a^*bc^* + ac$ over the alphabet $\Sigma = \{a, b, c\}$.

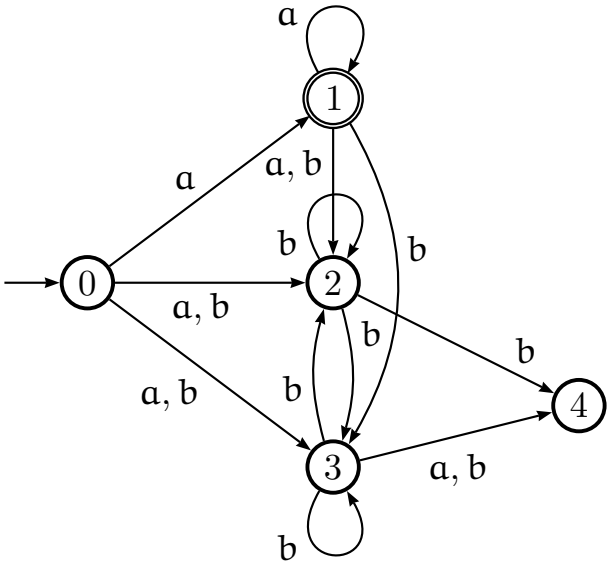
Answer: States 0 (start), 1, 2 (final), 3 (final), 4, and 5.
$T(0, a) = 1$, $T(1, c) = 2$, $T(0, b) = T(1, b) = T(4, b) = T(3, c) = 3$, $T(1, a) = T(4, a) = 4$, and all other transitions go to state 5.

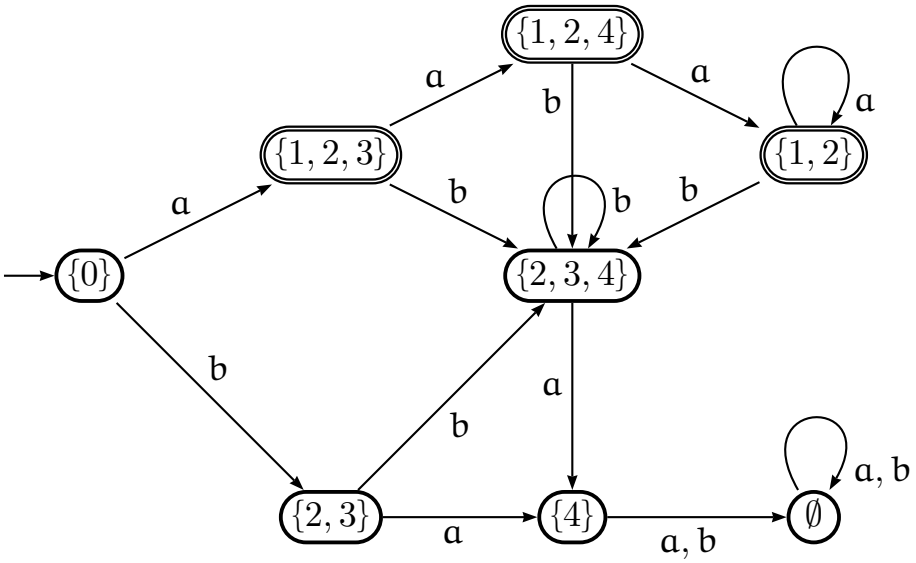
Question 6. Convert the following NFA to an equivalent DFA over the alphabet $\{a, b\}$.



Answer:

Question 7. Convert the following NFA to an equivalent DFA over the alphabet $\{a, b\}$.





Answer:

This was good as an exercise to practise with the algorithm, but you might also look at the language which is accepted by the NFA and try to construct a DFA from scratch. In this case it is quite easy! Only the strings aa^* are accepted!