# Exercise for chapter 4 (Part 2)
# Determinization, Optimization and Applications

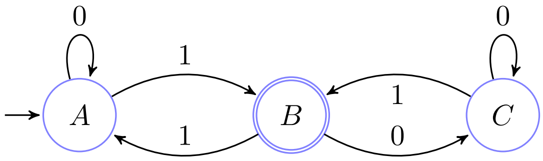
## 1 Introduction

In this exercise, we will pratice mainly on automata determinization - from NFA (nondeterministic finite automata) to DFA (deterministic finite automata). Students should review the slide and related theoretical documents before doing the exercises below.

## 2 Example

### Question 1.

Give an execution of the following DFA on 0001, 01001, and 0110.



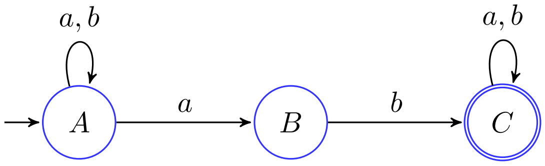
### Solution.

- $(A, 0001) \rightarrow (A, 001) \rightarrow (A, 01) \rightarrow (A, 1) \rightarrow (B, -) \Rightarrow 0001$ *is a valid word.*
- $(A, 01001) \rightarrow (A, 1001) \rightarrow (B, 001) \rightarrow (C, 01) \rightarrow (C, 1) \rightarrow (B, -)$. *Then,* 01001 *is a valid word.*
- $(A, 0110) \rightarrow (A, 110) \rightarrow (B, 10) \rightarrow (A, 0) \rightarrow (A, -)$. *Since A is not an accepting state, then* 0110 *is an invalid word.*

□

### Question 2.

Convert the following NFA into DFA.
Give an execution of the DFA on *aaba*, *bbabbbaa*, *bababaa* and *bbabbbabbbabba*.

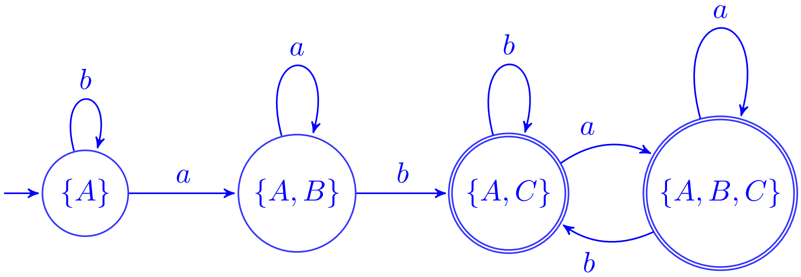


### Solution.

*First, we need define transition table containing useful sets of states as follows.*

| | $a$ | $b$ |
|---|---|---|
| $\rightarrow \{A\}$ | $\{A, B\}$ | $\{A\}$ |
| $\{A, B\}$ | $\{A, B\}$ | $\{A, C\}$ |
| $\{A, C\}^*$ | $\{A, B, C\}$ | $\{A, C\}$ |
| $\{A, B, C\}^*$ | $\{A, B, C\}$ | $\{A, C\}$ |

*DFA could be determined in which each state refers a useful sets of NFA state.*

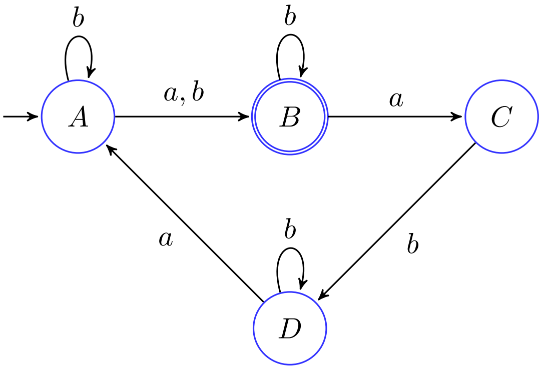


□

## 3 Exercise

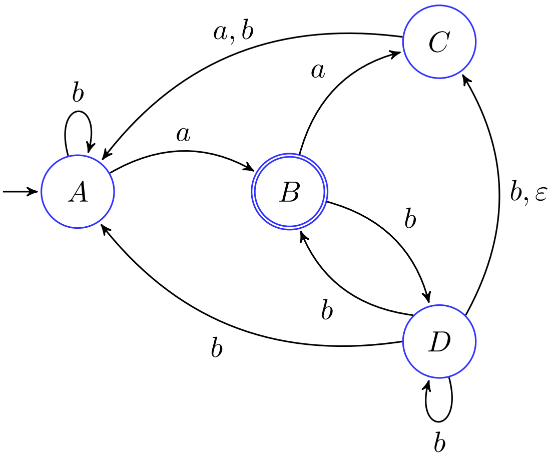
### Question 3.

Convert the following NFA into DFA.



### Question 4.

Convert the following NFA into DFA.



### Question 5.

Find a regular expression for the set $\{a^n b^m : (n+m) \textit{ is even }\}$.
Determine the corresponding DFA (or NFA and then convert NFA to DFA).

### Question 6.

Give a regular expression for the language on $\Sigma = \{a, b, c\}$ containing no any sequence of $a$ with length

greater than two.
Determine the corresponding DFA (or NFA and then convert NFA to DFA).

### Question 7.

Give a regular expression for the language on $\Sigma = \{a, b\}$ containing all strings not ending in $ab$.
Determine the corresponding DFA (or NFA and then convert NFA to DFA).

### Question 8.

Let $\Sigma = \{a, b, c\}$. Give complet DFA's for the sets consisting of

a) all strings with exactly two '$a$'.

b) all strings of odd length.

c) all strings which the number of appearances of both '$b$' and '$c$' is divisible by 3.

d) all strings ending with $ca$.

e) all strings not ending with '$a$' and any $aa$ appreared after $bc$.

f) all non-empty strings not ending with '$ca$'.

g) all strings with at least one '$b$'.

h) all strings with at most one '$a$' and at least one '$b$'.

i) all strings without any '$a$' and at most one '$b$'.

j) all strings including at least one $a$ and whose the first appearance of '$a$' is not followed by a '$b$'.

### Question 9.

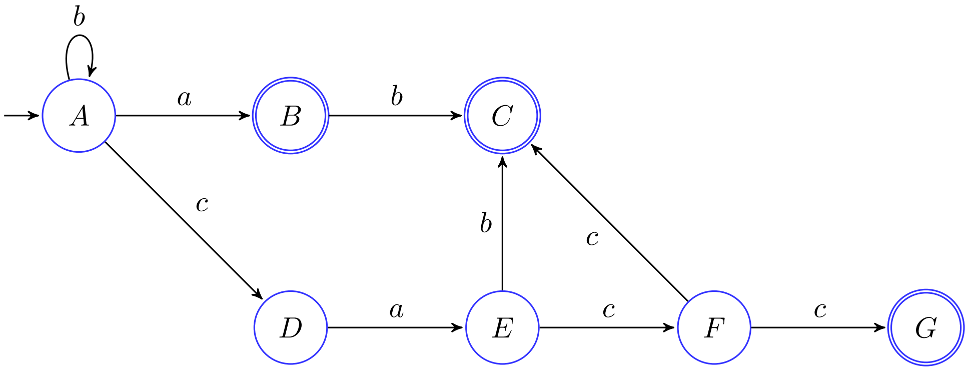
Give a DFA that accepts language which represent by regular expression:

- $E_1 = ((a + b)^*b(a + ab)^*)$,
- $E_2 = b^*(a + b) + aa^*aba^*$,
- $E_3 = (aa + ca)b^*b + +cab^*a^*c$,
- $E_4 = b(ca + ac)(aa)^* + a^*(ca + ac)$,
- $E_5 = (ab)^{2*}c + (a + b)c^*$,
- $E_6 = b(b^* + a^*b)ac + a^*(b^* + a^*b)$,
- $E_7 = (b + c)ab + ba(c + ab)^*$,
- $E_8 = (b + c)^*ba + a(c + b)^*$,
- $E_9 = (a(b + c)^* + bc^*)^*$.

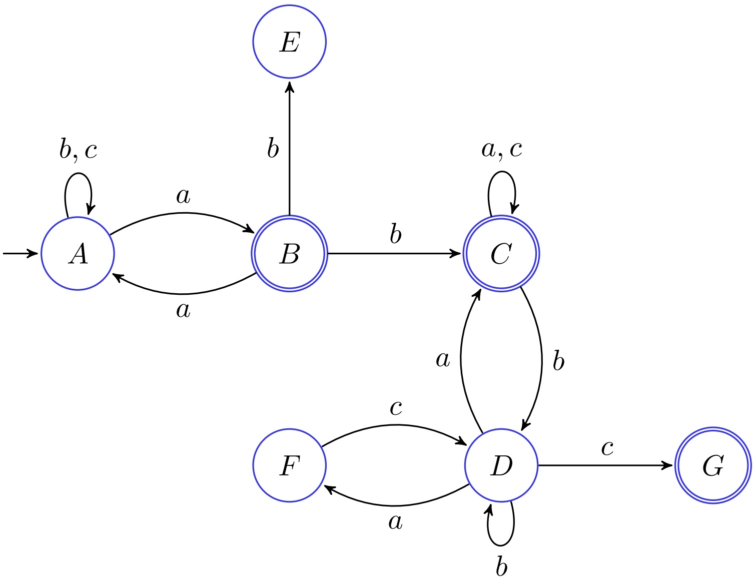
### Question 10.

Minimize the following automatas.

1.



2.



## Question 11.

Propose an automata to describe a vehicular multi-information display system with a given number of buttons.

For example, digital speedo meter of Honda Lead motor with only one button can display information about: petroleum level, speed, trip, date, time, engine oil life. (Hint: we distinguish two different actions: quickly press the button, press the button and hold-down over two seconds.)