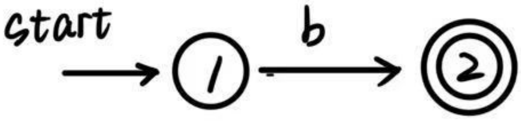
Exercise 1:

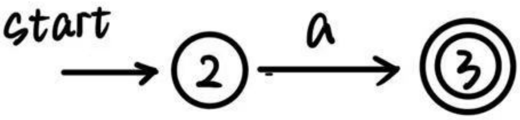
Convert the above two regular expressions to NFAs using the Thompson’s Construction Algorithm. Please put down the detailed steps and DO NOT optimize the NFAs.

- $a(ba)^*c$

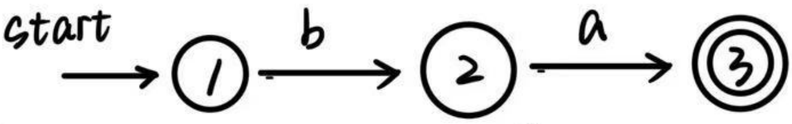
b



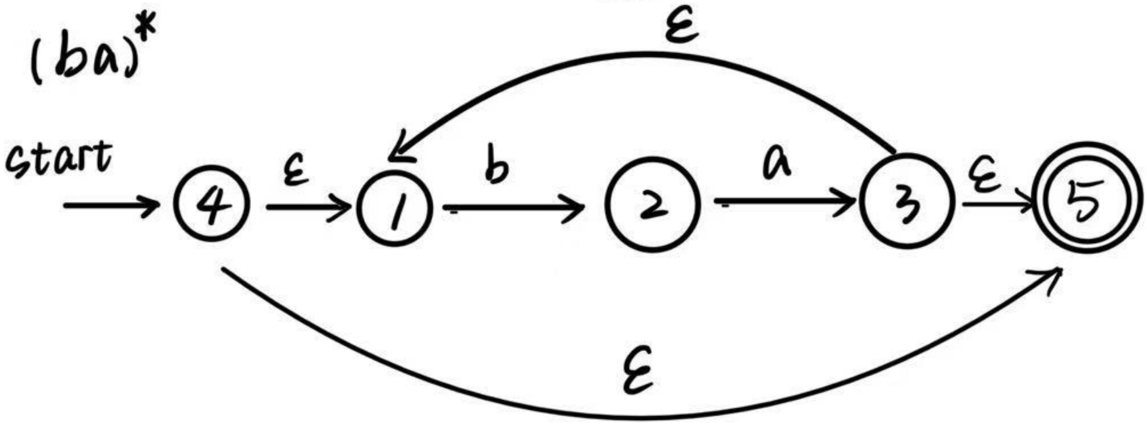
a



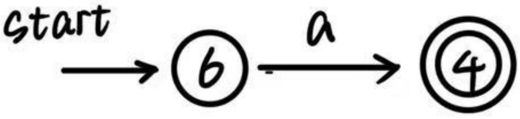
ba



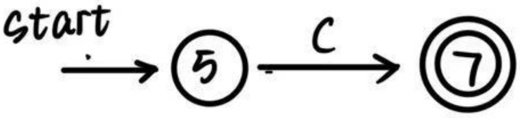
(ba)*

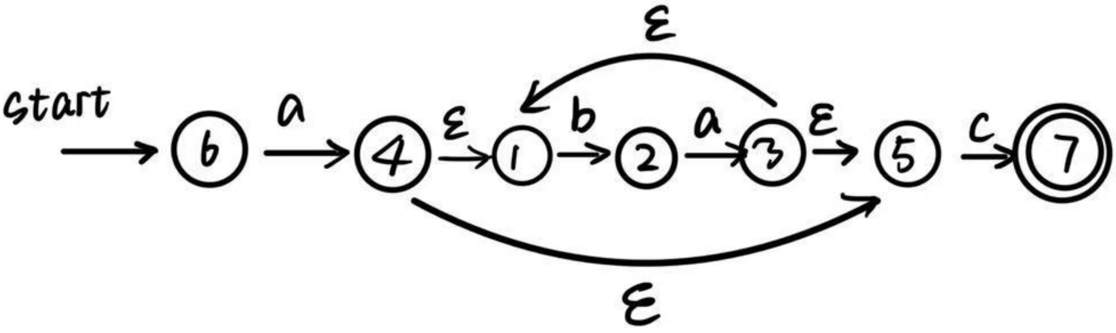


a



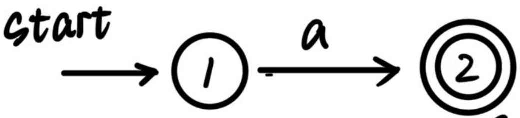
c



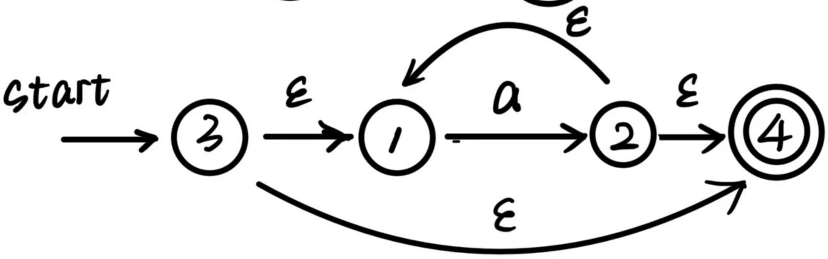


- $ba^+|ab^*$

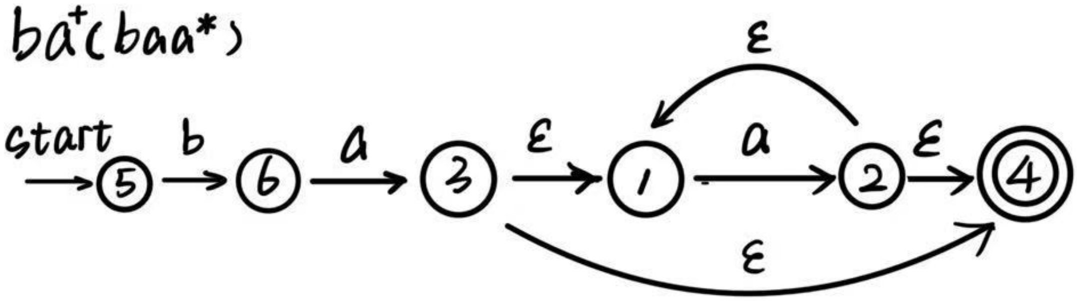
a



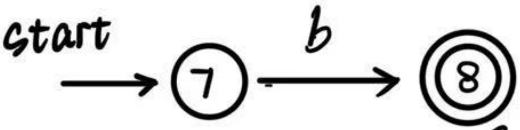
a*



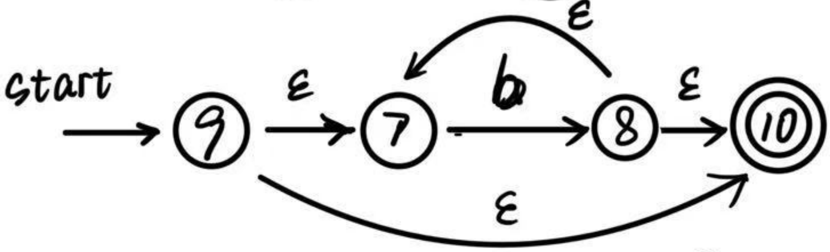
ba+(baa*)



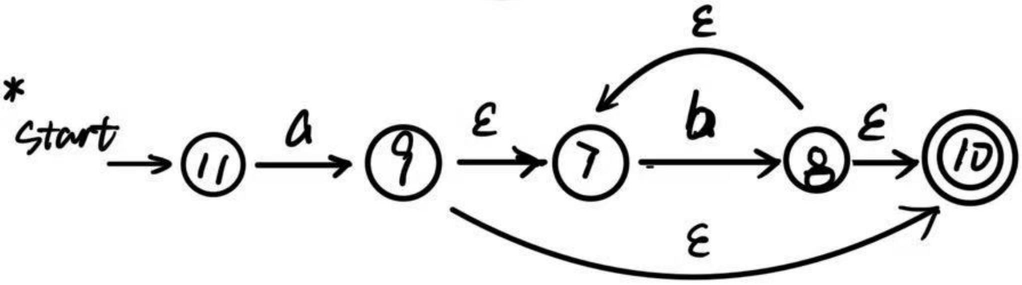
b



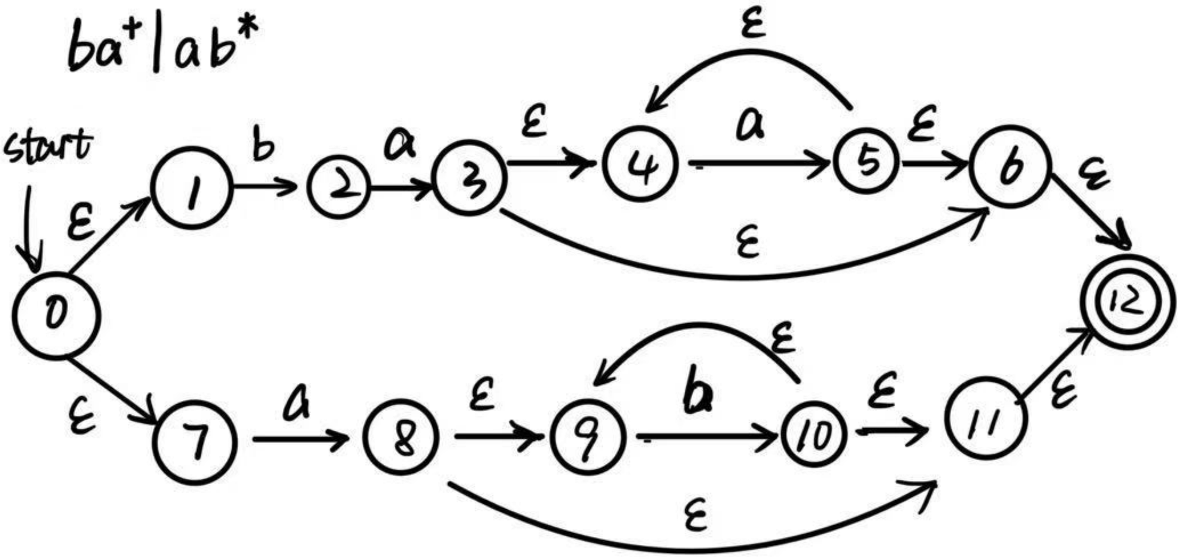
b*



ab*



ba+|ab*



Exercise 2:

Convert the NFAs constructed in Exercise 1 to DFAs using the Subset Construction Algorithm (Algorithm 3.20 in the dragon book). Please put down the detailed steps and DO NOT optimize the DFAs.

$a(ba)^*c$

DTran Table

NFA-State	DFA-State	a	b	c
$\{0\}$	A	B	X	X
$\{1,2,5\}$	B	X	C	E
$\{3\}$	C	D	X	X
$\{2,4,5\}$	D	X	C	E
$\{6\}$	E	X	X	X
$\varnothing$	X	X	X	X

1. Start state $A = \epsilon - closure(0) = \{0\}$
2. State A is unmarked, we check a, b, c in alphabet
   - $\epsilon - closure(move(A, a)) = \{1, 2, 5\}$, we note this as B and push it into stack
   - $\epsilon - closure(move(A, b)) = \varnothing$
   - $\epsilon - closure(move(A, c)) = \varnothing$
   - Now A is marked
3. State B is pop and is unmarked, we check:
   - $\epsilon - closure(move(B, a)) = \varnothing$
   - $\epsilon - closure(move(B, b)) = \{3\}$, denote it as C and push it into stack
   - $\epsilon - closure(move(B, c)) = \{6\}$, denote it as E and push stack
   - B is marked
4. State C is pop, we check:
   - $\epsilon - closure(move(C, a)) = \{2, 4, 5\}$, denote it as D and push stack
   - $\epsilon - closure(move(C, b)) = \varnothing$
   - $\epsilon - closure(move(C, c)) = E$,
   - C is marked

5. State D is pop, we check:
    - $\epsilon - closure(move(D, a)) = \varnothing$
    - $\epsilon - closure(move(D, b)) = \{3\}$, that is state C
    - $\epsilon - closure(move(D, c)) = \{6\}$, denote it as E
    - D is marked
6. E is the end state
    - $\epsilon - closure(move(E, a)) = \varnothing$
    - $\epsilon - closure(move(E, b)) = \varnothing$
    - $\epsilon - closure(move(E, c)) = \varnothing$

$ba^+|ab^*$

DTran Table

NFA-State	DFA-State	a	b	c
$\{0, 1, 7\}$	A	B	C	X
$\{8, 9, 11, 12\}$	B	X	D	X
$\{2\}$	C	E	X	X
$\{9, 10, 11, 12\}$	D	X	D	X
$\{3, 4, 6, 12\}$	E	F	X	X
$\{4, 5, 6, 12\}$	F	F	X	X
$\varnothing$	X	X	X	X

1. Start from state $A = \epsilon - closure(0)$, that is $\{0, 1, 7\}$
2. Pop state A and mark it, check the alphabet:
    - $\epsilon - closure(move(A, a)) = \{8, 9, 11, 12\}$, denote is as state B
    - $\epsilon - closure(move(A, b)) = \{2\}$, denote it as state C
    - $\epsilon - closure(move(A, c)) = \varnothing$
3. Pop state B and mark it:
    - $\epsilon - closure(move(B, a)) = \varnothing$
    - $\epsilon - closure(move(B, b)) = \{9, 10, 11, 12\}$, denote it as D
    - $\epsilon - closure(move(B, c)) = \varnothing$
4. Pop state C and mark it
    - $\epsilon - closure(move(C, a)) = \{3, 4, 6\}$, denote it as E

- $\epsilon - closure(move(C, b)) = \varnothing$
- $\epsilon - closure(move(C, c)) = \varnothing$

5. Pop state D and mark it
   - $\epsilon - closure(move(D, a)) = \varnothing$
   - $\epsilon - closure(move(D, b)) = D$
   - $\epsilon - closure(move(D, c)) = \varnothing$
6. Pop state E and mark it
   - $\epsilon - closure(move(E, a)) = \{4, 5, 6, 12\}$, denote it as F
   - $\epsilon - closure(move(E, b)) = \varnothing$
   - $\epsilon - closure(move(E, c)) = \varnothing$
7. Pop state F and mark it
   - $\epsilon - closure(move(F, a)) = F$
   - $\epsilon - closure(move(F, b)) = \varnothing$
   - $\epsilon - closure(move(F, c)) = \varnothing$

Optional Excercises

Exercise1:

Please pick a DFA you have constructed for the above two languages and follow the State-Minimization Algorithm (Algorithm 3.39 in the dragon book) to minimize the number of states in the DFA. There might be chances that the built DFA is already minimum and in that case you should justify why it is already minimum. Note that the algorithm is not covered during lectures and you need to study it by yourself.

Given DFA Transition Table:

NFA-State	DFA-State	a	b	c
$\{0\}$	A	B	X	X
$\{1, 2, 5\}$	B	X	C	X
$\{3\}$	C	D	X	X
$\{2, 4, 5\}$	D	X	C	E
$\{6\}$	E	X	X	X

NFA-State	DFA-State	a	b	c
∅	X	X	X	X

Step 1: Initial Partition

We start by partitioning the DFA states into two groups:

- **Group 1 (Accepting States)**: {E}
- **Group 2 (Non-Accepting States)**: {A, B, C, D, X}

Step 2: Refining the Non-Accepting States

Constructing the Transition Map for Group 2:

State	a	b	c
A	B	X	X
B	X	C	E
C	D	X	X
D	X	C	E
X	X	X	X

We map the transitions to their respective groups:

State	a's Target Group	b's Target Group	c's Target Group
A	Group 2 (B)	Group 2 (X)	Group 2 (X)
B	Group 2 (X)	Group 2 (C)	Group 1 (E)
C	Group 2 (D)	Group 2 (X)	Group 2 (X)
D	Group 2 (X)	Group 2 (C)	**Group 1 (E)**
X	Group 2 (X)	Group 2 (X)	Group 2 (X)

- State **B**, **D** transitions to an accepting state (Group 1) on input $'c'$, while other states do not. Therefore, state **B**, **D** must be separated from the other non-accepting states and they should be combined into one state **B**.

New Partition:

- **Group 1 (Accepting States)**: {E}
- **Group 2a**: {B, D}
- **Group 2b**: {A, C, X}

Step 3: Refining Group 2b

DFA-State	a	b	c
A	B	X	X
B	X	C	X
C	D	X	X
X	X	X	X

We map the transitions to their respective groups:

State	a's Target Group	b's Target Group	c's Target Group
A	Group 2b (B)	Group 2b (X)	Group 2b (X)
C	**Group 2a (D)**	Group 2b (X)	Group 2b (X)
X	Group 2b (X)	Group 2b (X)	Group 2b (X)

- State C transitions to Group $2a(D)$ on input $'a'$, while others do not.
- Therefore, state A, C must be separated from states X and they should be merged

Final Partition:

- **Group 1 (Accepting States)**: $\{E\}$
- **Group 2a**: $\{B, D\}$
- **Group 2b1**: $\{A, C\}$
- **Group 2b2**: $\{X\}$

Minimized DFA Transition Table:

State	**a**	**b**	**c**	**Accepting**
A	B	X	X	No
B	X	A	X	No
E	X	X	X	**Yes**
X	X	X	X	No