Exercise

- lexical analysis of “if i == 0”? Write the token sequence.
  - <keyword, ‘if’>, <id, ‘i’ >, <op, ‘==’>, <num, ‘0’>
- Usage of RE and FA in lexical analysis?
  - RE: specify the token pattern; FA: implement the token recognizer
- Regular expression (x | y)(x | y) denotes the set
  - {xx, xy, yx, yy}
- The languages over the {0,1} described by (0|1)*0(0|1)*0(0|1)*
  - Strings with at least two 0’s

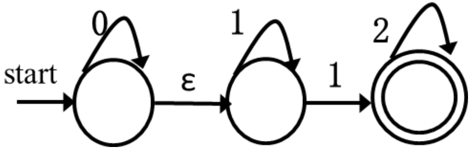
Exercise

- The graph describes NFA or DFA? Why?

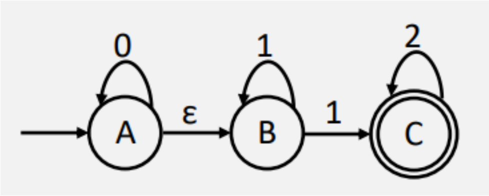
  NFA.

  A: ε-transition,

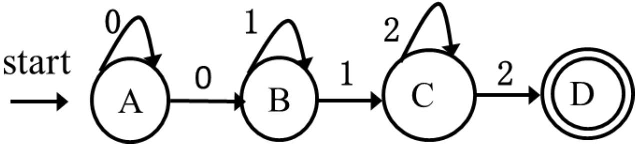
  B: multiple transitions for input ‘1’



- What is the RE?
  - 0*1+2*



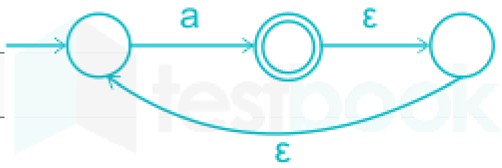
- Then, what is the NFA of 0+1+2+?



Exercise

Que.	The behavior of a NFA can be simulated by a DFA
a.	always
b.	sometime
c.	Never
d.	Depend on NFA
	Answer: always
Que.	Which of the following are Lexemes?
a.	Identifiers
b.	Constants
c.	Keywords
d.	All of the mentioned
	Answer: All of the mentioned

Exercise

Que.	Regular expression a|b denotes the set
a.	{a}
b.	{ε, a, b}
c.	{a, b}
d.	{a b}
	Answer: {a, b}
Que.	What is the complement[补集] of the language accepted by the NFA shown below? Assume ∑ = {a} and ε is the empty string
a.	Φ 
b.	ε
c.	a
d.	{ε, a}
	Answer: ε

Exercise

- Write regular definitions for the following languages
  1. All strings of a's and b's which contains the substring aba

     (a|b)*aba(a|b)*
  2. A language comprising all possible strings of even[偶数] length over the alphabet {a,b}

     (aa | ab| ba | bb)*

  3*. All strings of a‘s and b’s that DO NOT contain the substring abb

     b*(a | ab)*

ε

bbb...

aaa...

a..ba..

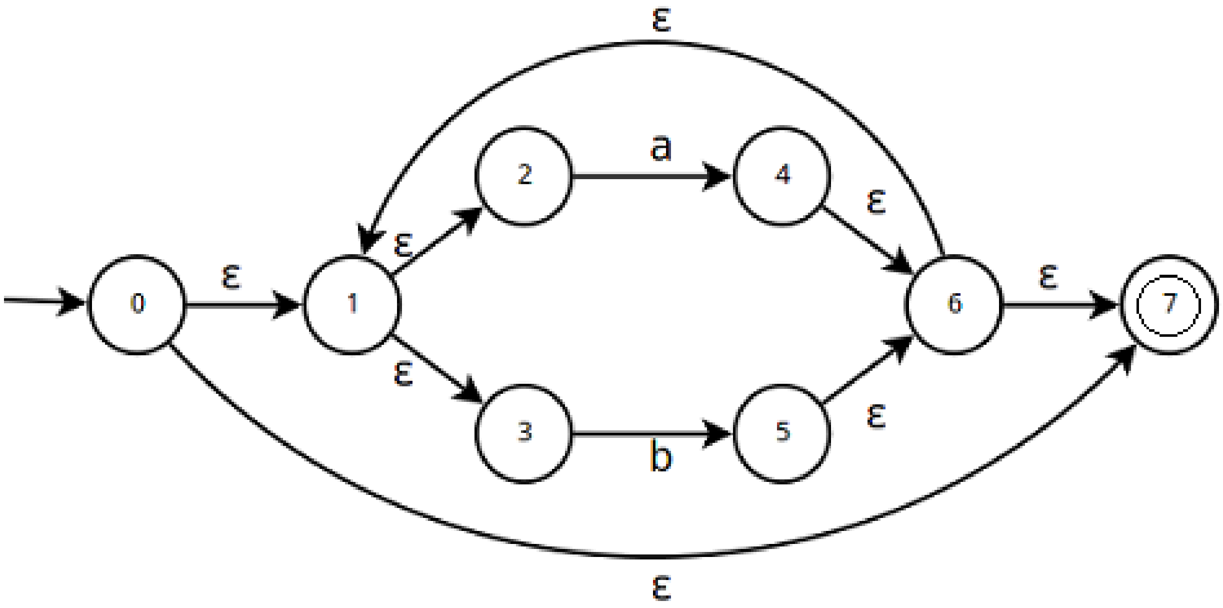
Exercise

- Convert (a|b)*abb(a|b)* into NFA
  - Thompson construction: RE→NFA

    (a|b)→(a|b)*

    abb

    (a|b)*abb(a|b)*



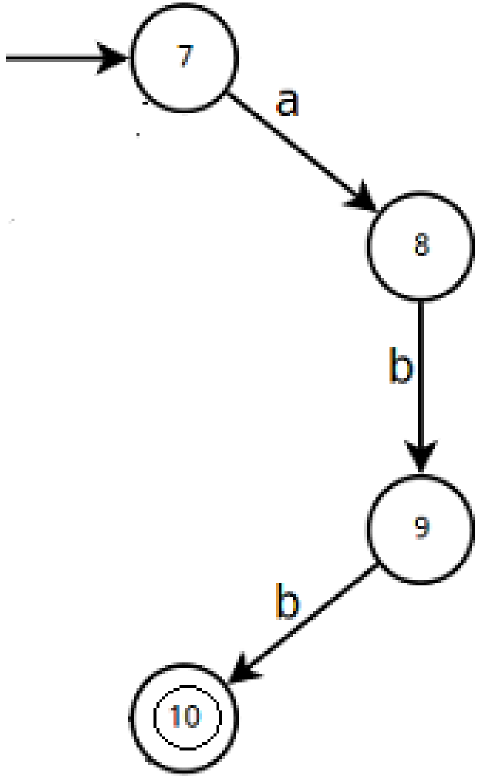
Exercise

- Convert (a|b)*abb(a|b)* into NFA
  - Thompson construction: RE→NFA

    (a|b)→(a|b)*

    abb

    (a|b)*abb(a|b)*



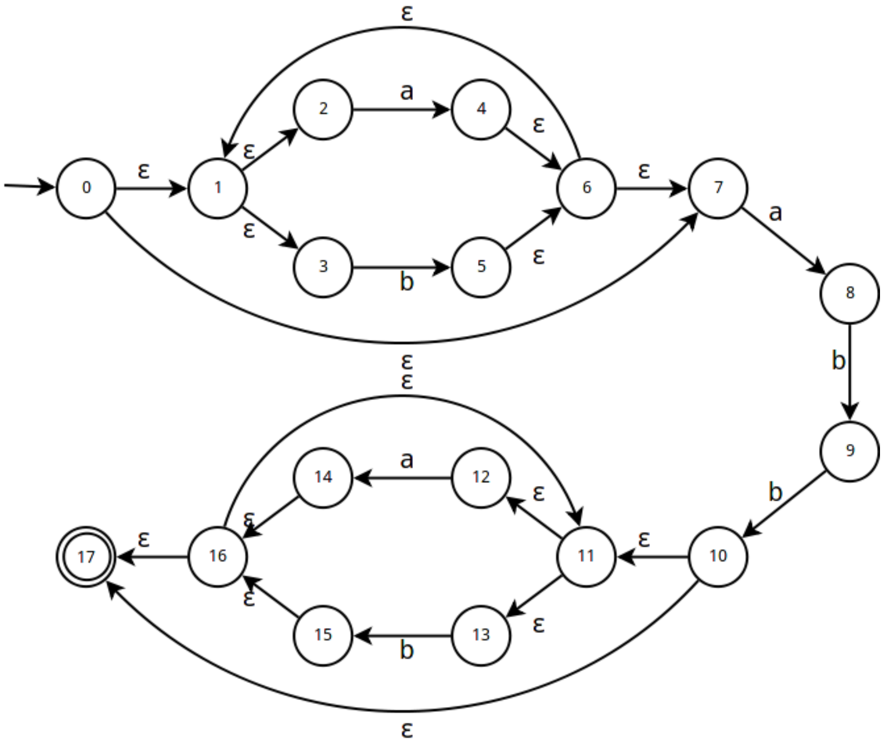
Exercise

- Convert (a|b)*abb(a|b)* into NFA
  - Thompson construction: RE→NFA

    (a|b)→(a|b)*

    abb

    (a|b)*abb(a|b)*



Exercise: from RE to minimized FA

- Construct a DFA for a minion language with $\sum = \{a, b\}$ that does not contain "abb":

1. Build the <u>regular expression</u> for the minion's language
2. Convert the regular expression into <u>NFA</u> first
3. Convert the NFA into <u>DFA</u> by subset construction.
4. <u>Minimize</u> the state of DFA

https://www.sensacine.com/peliculas/pelicula-210493/fotos/detalle/?cmediafile=21209746

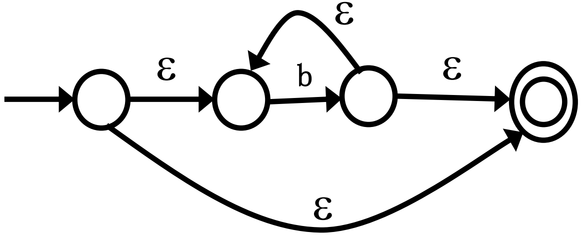
- Build the RE for this language

  b*(a | ab)*

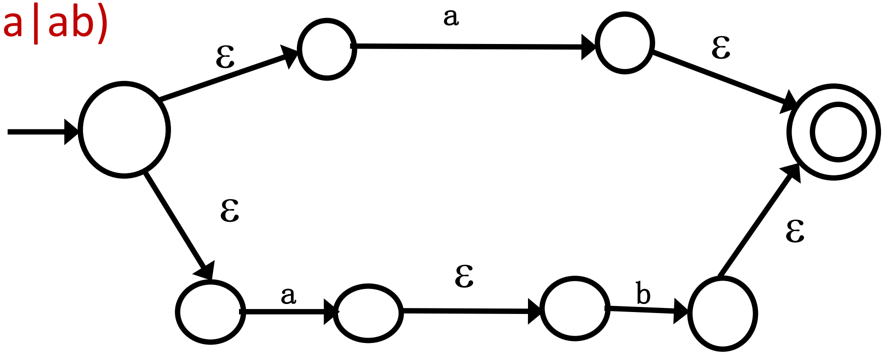
Exercise

- Convert b*(a | ab)* into NFA

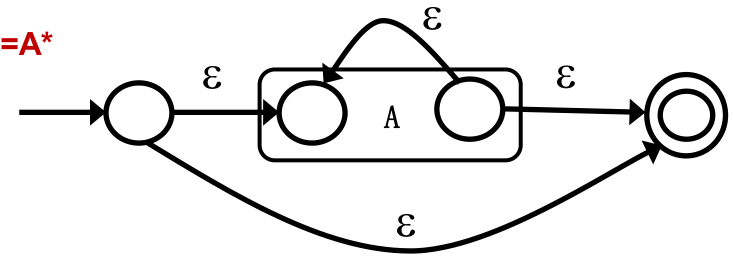
b*



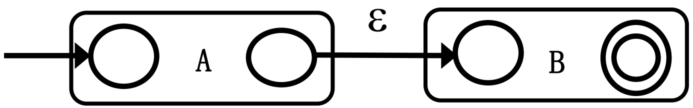
(a|ab)



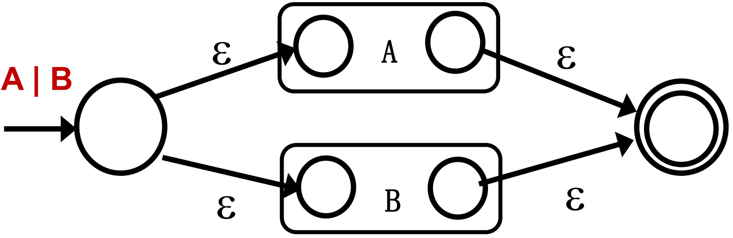
R=A*



R=AB

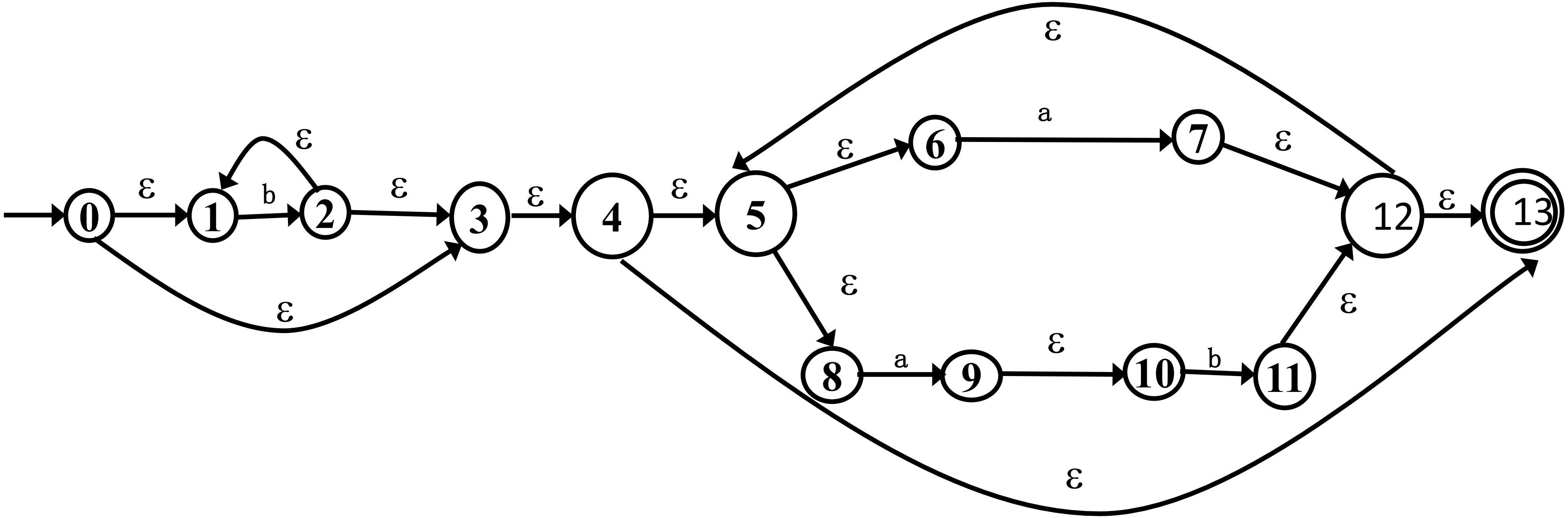


R=A | B

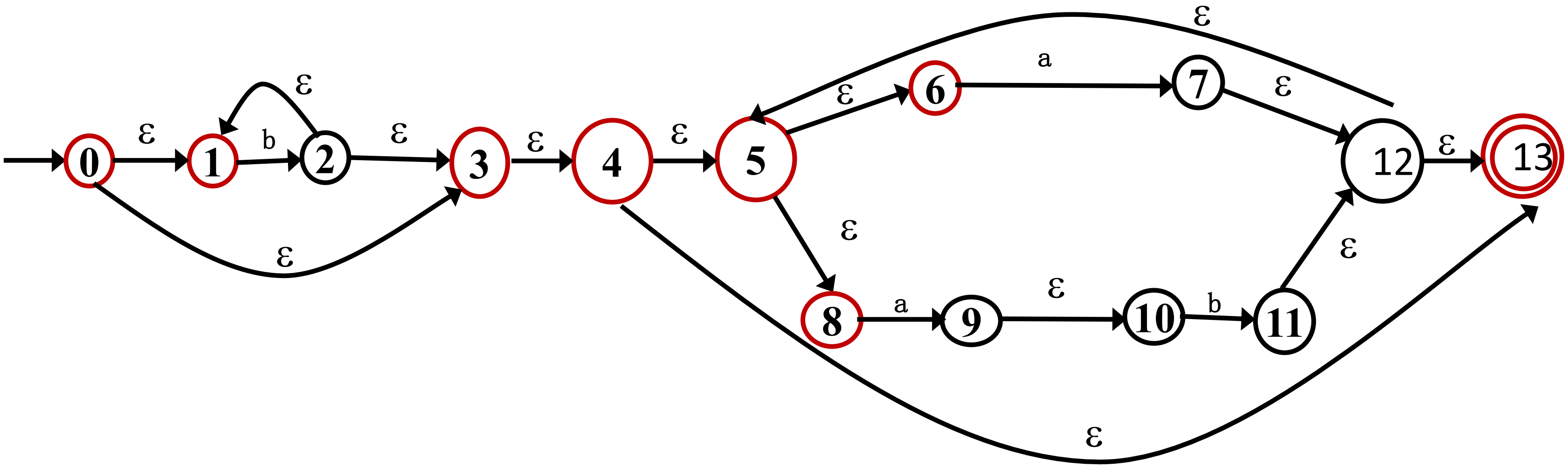


Exercise

- Convert the NFA into DFA by subset construction



Exercise

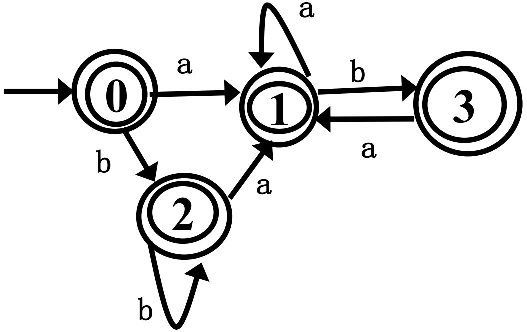


I	I_a	I_b	Accept
{0,1,3,4,5,6,8,13} 0	{7,9,12,13,5,6,8,10} 1	{2,1,3,4,5,6,8,13} 2	Yes
{5,6,7,8,9,10,12,13} 1	{5,6,7,8,9,10,12,13} 1	{11,12,13,5,6,8} 3	Yes
{1,2,3,4,5,6,8,13} 2	{5,6,7,8,9,10,12,13} 1	{1,2,3,4,5,6,8,13} 2	Yes
{5,6,8,11,12,13} 3	{5,6,7,8,9,10,12,13} 1	{}	Yes

Exercise

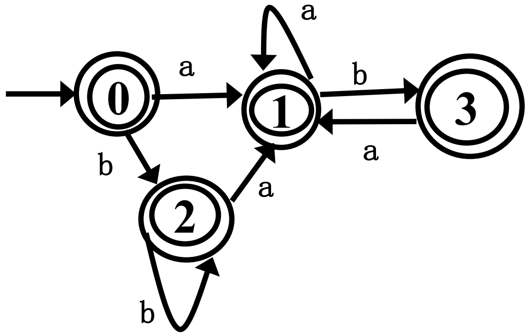
- Draw DFA according to the transition table

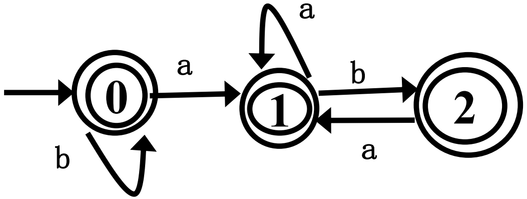
I	I_a	I_b
{0,1,3,4,5,6,8,13} 0	{7,9,12,13,5,6,8,10} 1	{2,1,3,4,5,6,8,13} 2
{5,6,7,8,9,10,12,13} 1	{5,6,7,8,9,10,12,13} 1	{11,12,13,5,6,8} 3
{1,2,3,4,5,6,8,13} 2	{5,6,7,8,9,10,12,13} 1	{1,2,3,4,5,6,8,13} 2
{5,6,8,11,12,13} 3	{5,6,7,8,9,10,12,13} 1	{}



Exercise

- Minimization DFA





- Initial: {0,1,2,3}

  {3} have no 'b' transition， split
- {0,1,2} {3}

  {1} on 'b' ->3， {0,2} on 'b' ->{2}， split
- {0,2} {1} {3}

  {0,2} on 'a' -> {1}

  {0,2} on 'b' -> {2}

  No way to distinguish {0,2} on any transition with '0' or '1'
- Final: {0,2} {1} {3}