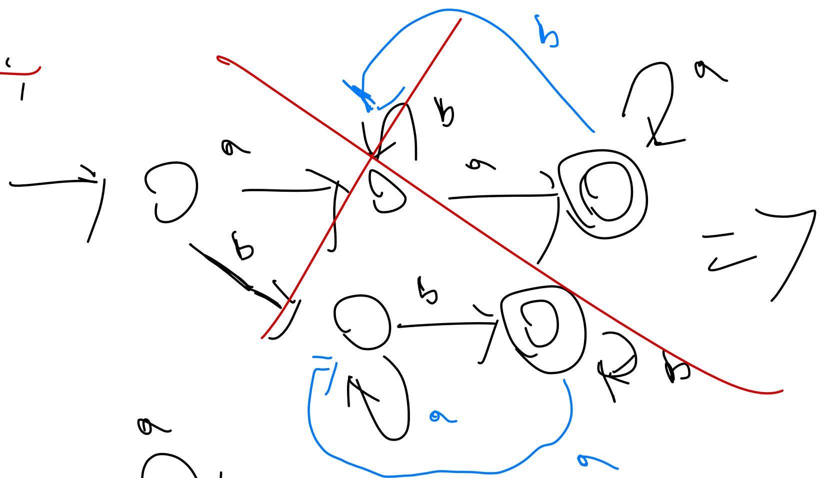
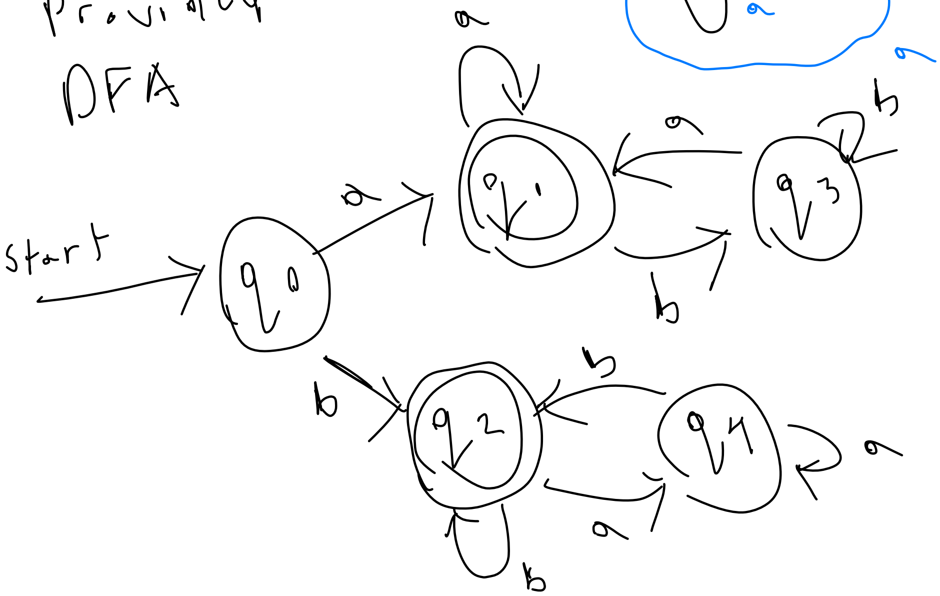
1, $\{a, b\}$

L_1 = set of words that start and end with same symbol

~~NFA:~~

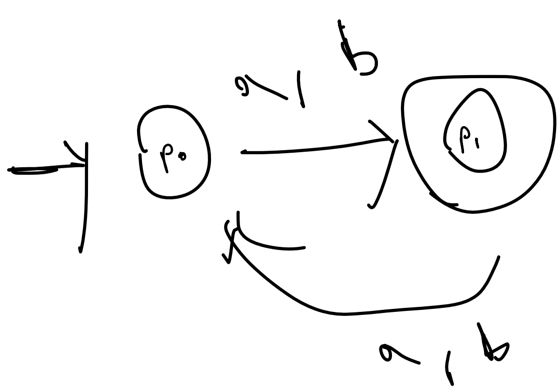


Provided DFA

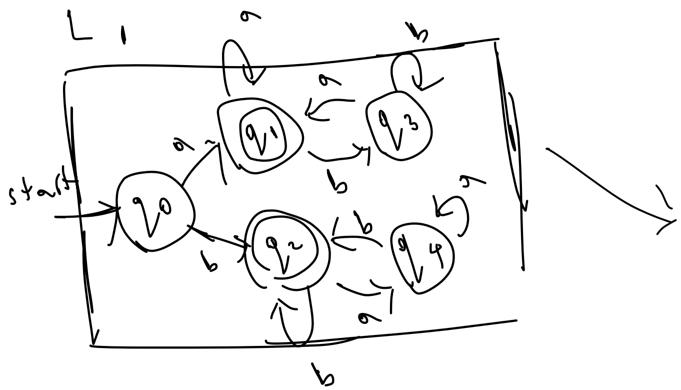


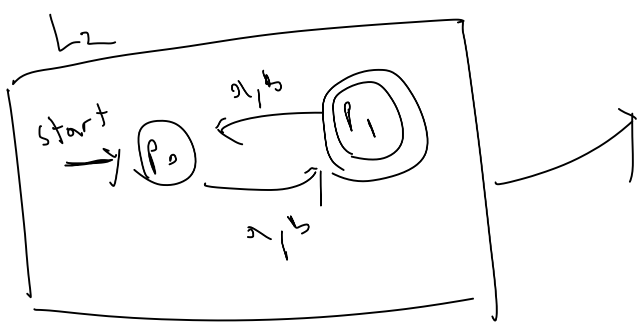
L_2 = set of words of odd length

DFA:

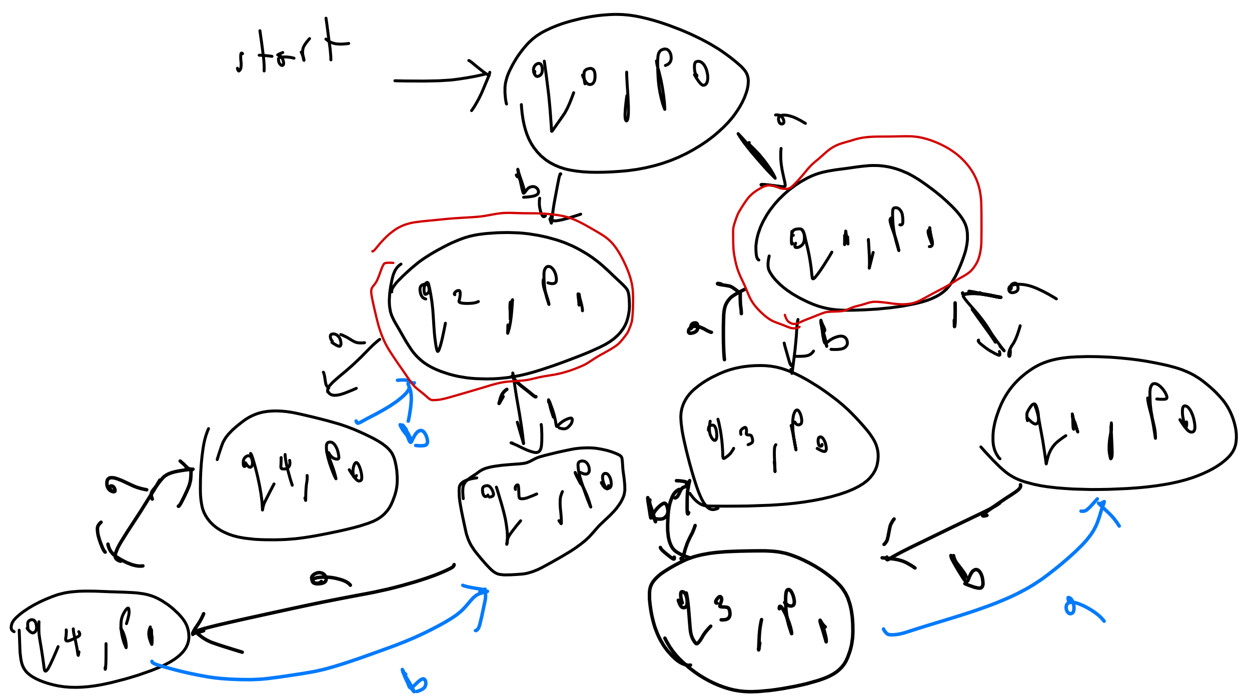


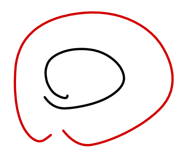
1.a. Using product DFA to prove intersection





Product



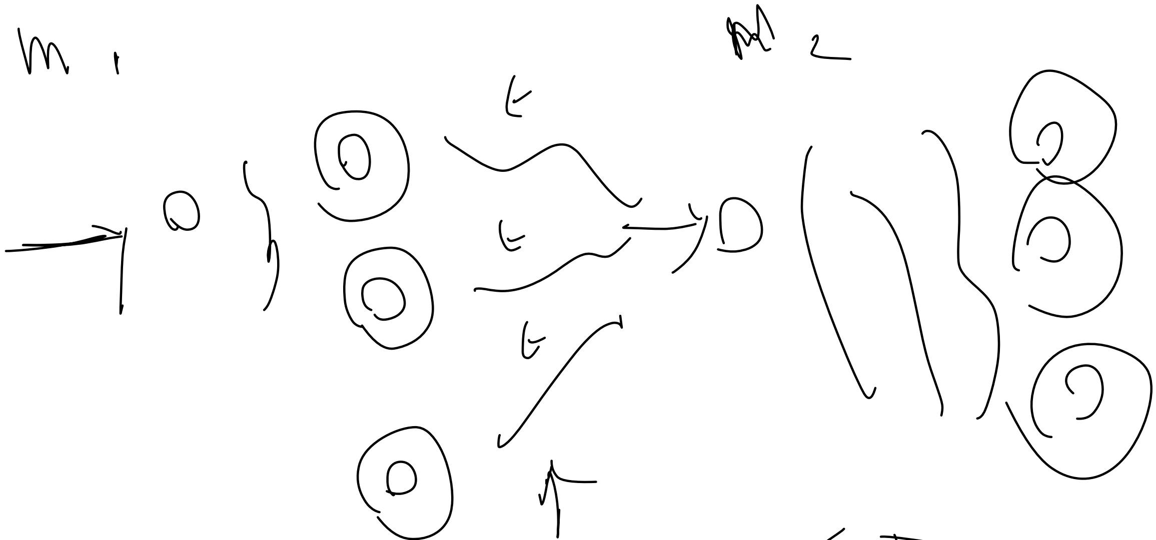
 - Final State

* Worst case = 10 states

1.b. $L_1 \cdot L_2$

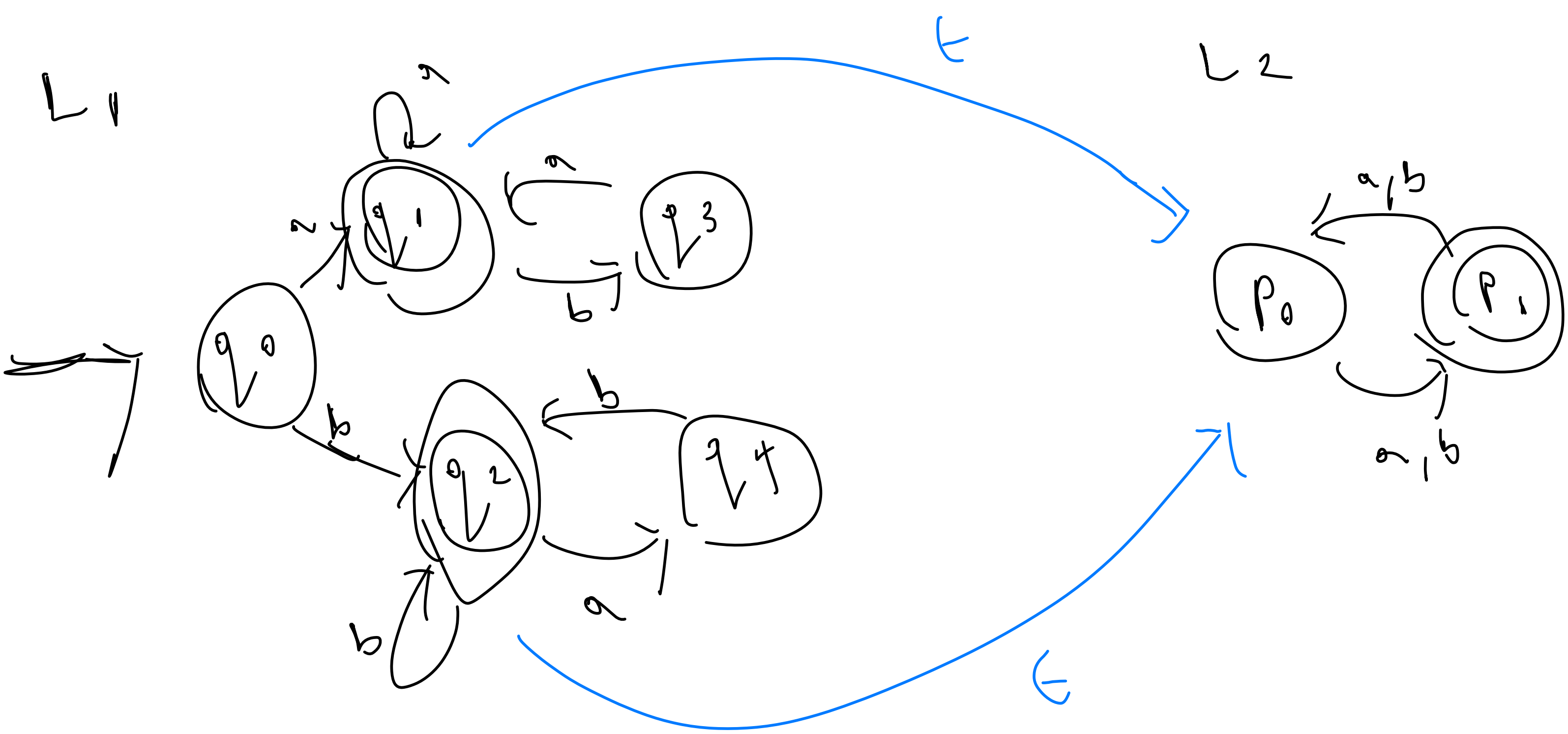
concatenation

machine concatenation:

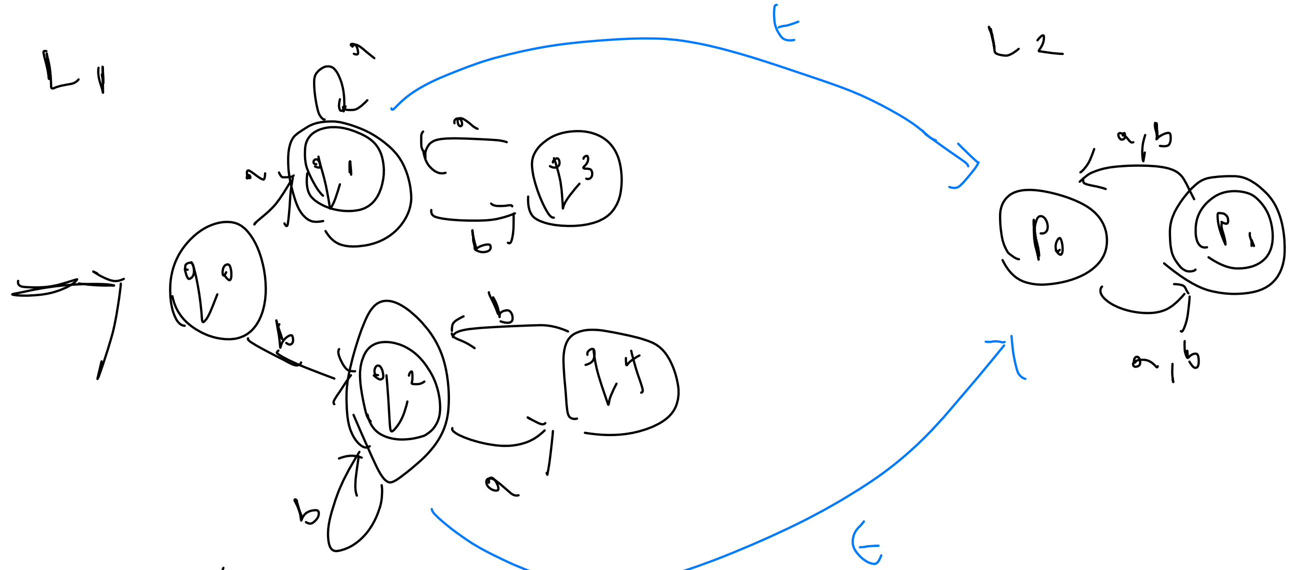


use RMET

to remove ε

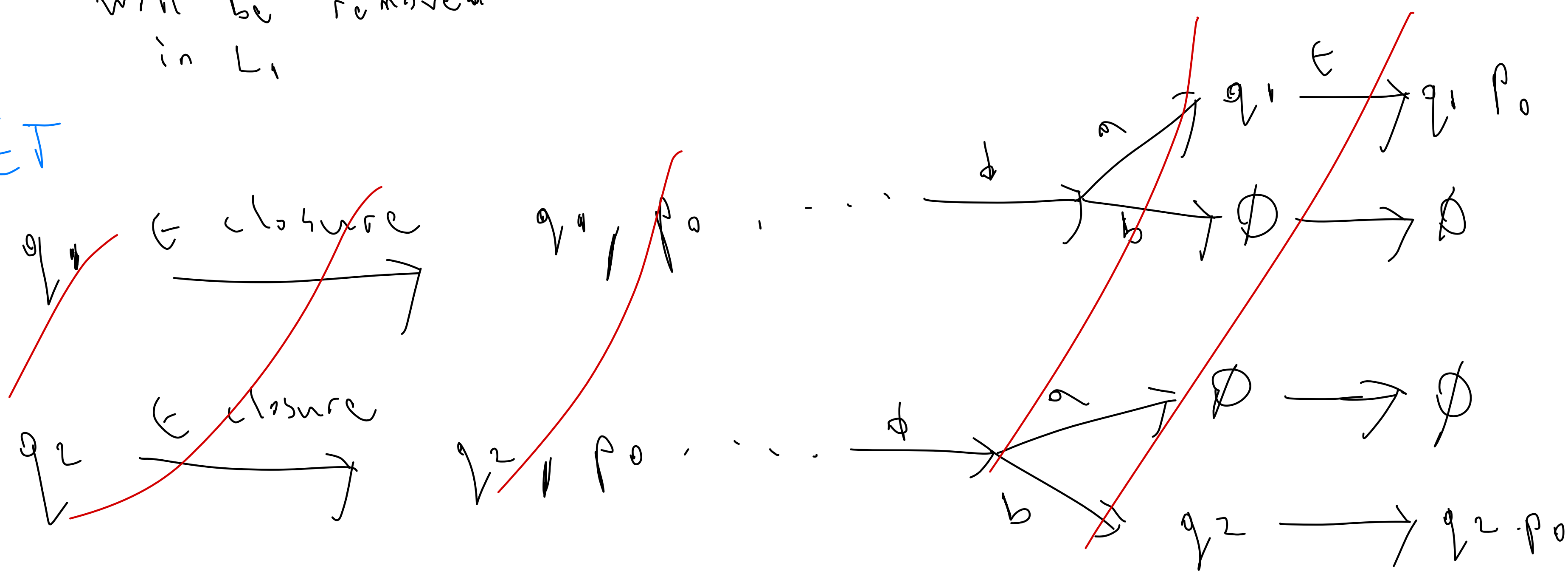


1.b. $L_1 \cdot L_2$



* Final states will be removed in L_1

RMET



1.b RMET

	ε-closure
p_0	p_0
p_1	p_1
q_0	q_0
q_1	p_0, q_1
q_2	p_0, q_2
q_3	q_3
q_4	q_4

$$p_0 \xrightarrow{\epsilon} p_0 \xrightarrow{a} p_1 \xrightarrow{\epsilon} p_1$$

$$p_0 \xrightarrow{\epsilon} p_0 \xrightarrow{b} p_1 \xrightarrow{\epsilon} p_1$$

$$p_1 \xrightarrow{\epsilon} p_1 \xrightarrow{a} p_0 \xrightarrow{\epsilon} p_0$$

$$p_1 \xrightarrow{\epsilon} p_1 \xrightarrow{b} p_0 \xrightarrow{\epsilon} p_0$$

$$\begin{tikzcd} q_0 \arrow[r, "\epsilon"] & q_0 \arrow[r, "a"] & q_1 \arrow[r, "\epsilon"] \arrow[dr, "\epsilon"] & q_1 \\ & & & p_0 \end{tikzcd}$$

$$\begin{tikzcd} q_0 \arrow[r, "\epsilon"] & q_0 \arrow[r, "b"] & q_2 \arrow[r, "\epsilon"] \arrow[dr, "\epsilon"] & p_0 \\ & & & q_2 \end{tikzcd}$$

1. b RMET

$q_1 \xrightarrow{\varepsilon} q_1 \xrightarrow{a} q_1 \xrightarrow{\varepsilon} q_1$, $q_1 \xrightarrow{\varepsilon} p_0$

$q_1 \xrightarrow{\varepsilon} p_0 \xrightarrow{a} p_1 \xrightarrow{\varepsilon} p_1$

$q_1 \xrightarrow{\varepsilon} q_1 \xrightarrow{b} q_3 \xrightarrow{\varepsilon} q_3$

$q_1 \xrightarrow{\varepsilon} p_0 \xrightarrow{b} p_1 \xrightarrow{\varepsilon} p_1$

$q_2 \xrightarrow{\varepsilon} q_2 \xrightarrow{a} q_4 \xrightarrow{\varepsilon} q_4$

$q_2 \xrightarrow{\varepsilon} p_0 \xrightarrow{a} p_1 \xrightarrow{\varepsilon} p_1$

$q_2 \xrightarrow{\varepsilon} q_2 \xrightarrow{b} q_2 \xrightarrow{\varepsilon} q_2$, $q_2 \xrightarrow{\varepsilon} p_0$

$q_2 \xrightarrow{\varepsilon} p_0 \xrightarrow{b} p_1 \xrightarrow{\varepsilon} p_1$

$q_3 \xrightarrow{\varepsilon} q_3 \xrightarrow{a} q_1 \xrightarrow{\varepsilon} q_1$, $q_1 \xrightarrow{\varepsilon} p_0$

$q_3 \xrightarrow{\varepsilon} q_3 \xrightarrow{b} \emptyset$

??? can we say it goes to empty state?

???

$q_4 \xrightarrow{\varepsilon} q_4 \xrightarrow{a} \emptyset$

$q_4 \xrightarrow{\varepsilon} q_4 \xrightarrow{b} q_2 \xrightarrow{\varepsilon} p_0$, $q_2 \xrightarrow{\varepsilon} q_2$

1. b. cont

	a	b
p_0	p_1	p_1
F p_1	p_0	p_0
q_0	q_1, p_0	q_2, p_0
q_1	q_1, p_0, p_1	q_3, p_1
q_2	q_4, p_1	q_2, p_0, p_1
q_3	p_0, q_1	$\emptyset$
q_4	$\emptyset$	p_0, q_2

Note: In NFD, include [ ] in steps for q_3 & q_4

1. b. NFD

Queue: [[q_0]]

$[q_0] \xrightarrow{a} [q_1, p_0]$

$[q_0] \xrightarrow{b} [q_2, p_0]$

Queue: [[q_1, p_0], [q_2, p_0]]

$[q_1, p_0] \xrightarrow{a} [q_1, p_0, p_1]$

$[q_1, p_0] \xrightarrow{b} [q_3, p_1]$

✓ — Already in Queue

✓ — Already defined

Queue: [[q_2, p_0], [q_1, p_0, p_1], [p_1], [q_3, p_1]]

$[q_2, p_0] \xrightarrow{a} [q_4, p_1]$

$[q_2, p_0] \xrightarrow{b} [q_2, p_0, p_1]$

Queue: [[q_1, p_0, p_1], [p_1], [q_3, p_1], [q_4, p_1], [q_2, p_0, p_1]]

$[q_1, p_0, p_1] \xrightarrow{a} [q_1, p_0, p_1]$ ✓

$[q_1, p_0, p_1] \xrightarrow{b} [q_3, p_1, p_0]$ ✓

Queue: [~~[p_1]~~, ~~[q_3, p_1]~~, [q_4, p_1], [q_2, p_0, p_1], [p_0]]

$[p_1] \xrightarrow{a} [p_1]$ ✓

$[p_1] \xrightarrow{b} [p_0]$

$[q_3, p_1] \xrightarrow{a} [p_0, q_1]$ ✓

$[q_3, p_1] \xrightarrow{b} [p_0]$ ✓

Queue: [[q_4, p_1], [q_2, p_0, p_1], [p_0], []]

$[q_4, p_1] \xrightarrow{a} [p_0]$

$[q_4, p_1] \xrightarrow{b} [p_0, q_2]$

Queue: [[~~q_2~~, p_0, p_1], [~~p_0~~], [~~ ~~]]

$[q_2, p_0, p_1] \xrightarrow{a} [q_4, p_1, p_0]$

$[q_2, p_0, p_1] \xrightarrow{b} [q_2, p_0, p_1]$

$[p_0] \xrightarrow{a} [p_1]$

$[p_0] \xrightarrow{b} [p_1]$

1. b.

Final DFA

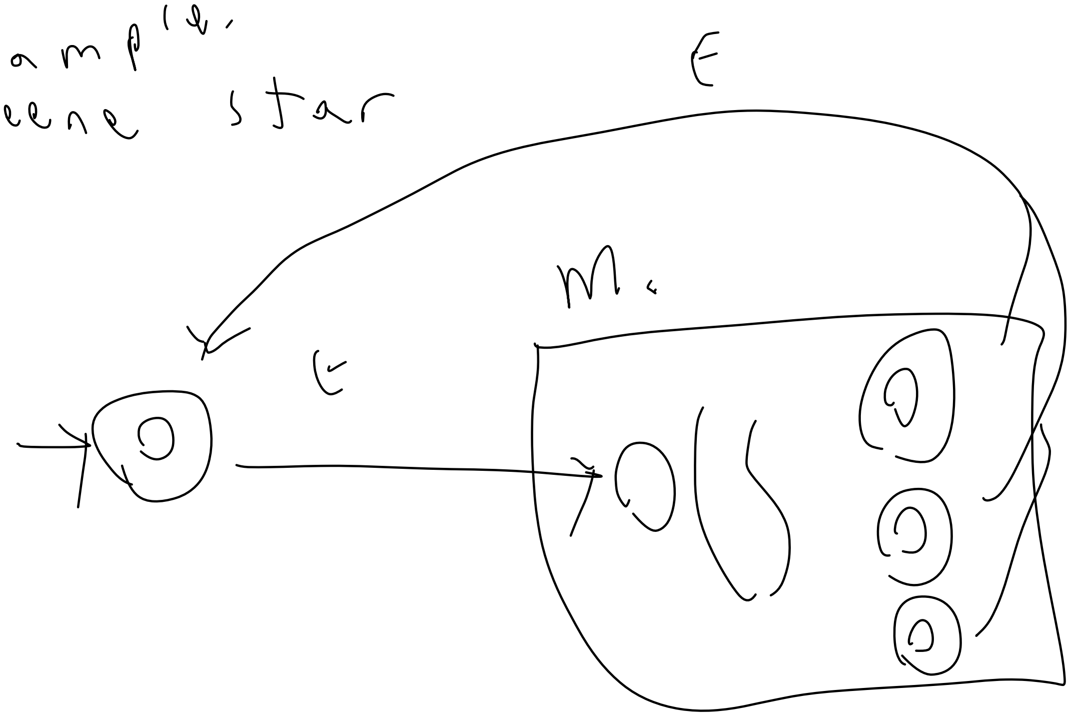
S - Start state

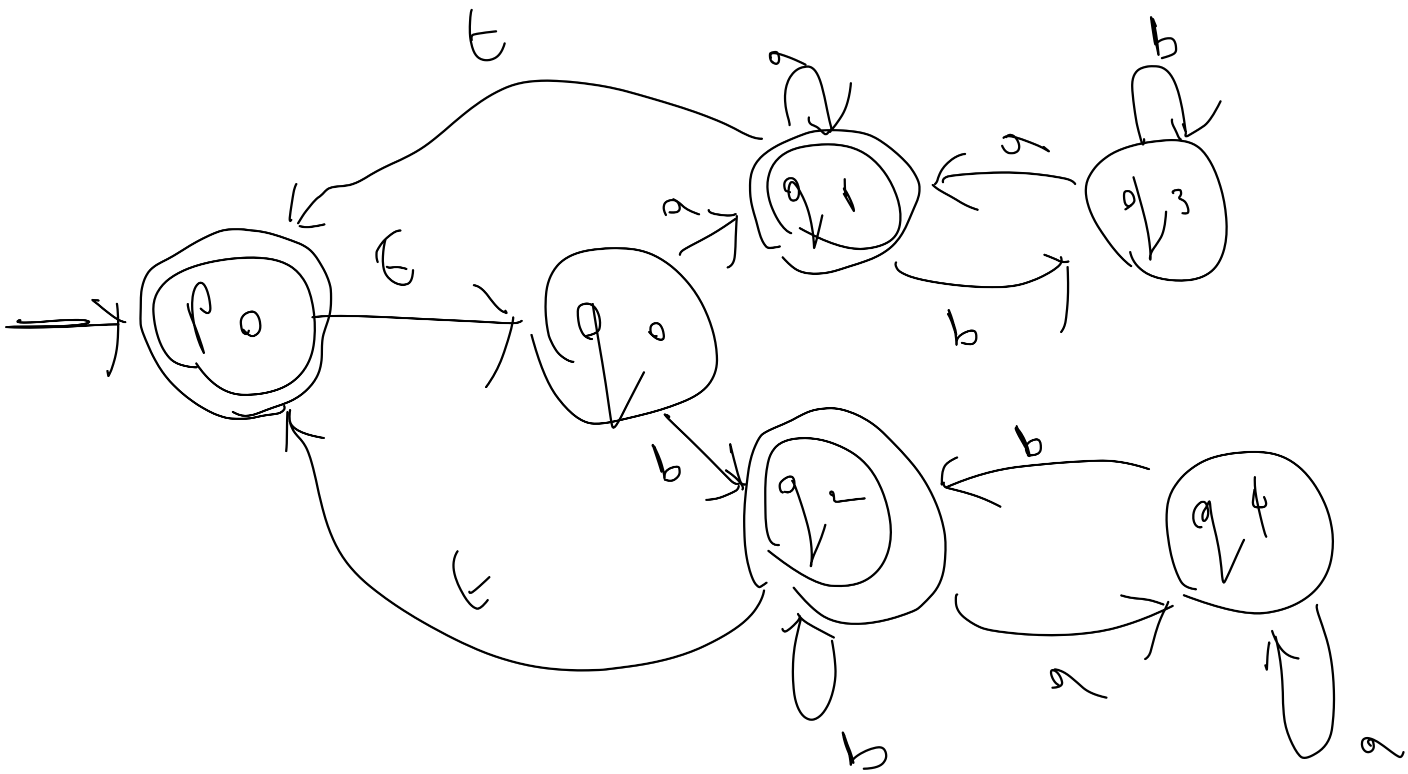
F - Final state

		a	b
S	[q0]	[q1, p0]	[q2, p0]
	[q1, p0]	[q1, p0, p1]	[q3, p1]
	[q2, p0]	[q4, p1]	[q2, p0, p1]
F	[q1, p0, p1]	[q1, p0, p1]	[q3, p1, p0]
F	[p1]	[p0]	[p0]
F	[q3, p1]	[q1, p0]	[p0]
F	[q4, p1]	[p0]	[p0, q4]
F	[q2, p0, p1]	[q4, p1, p0]	[q2, p0, p1]
	[p0]	[p1]	[p1]

1. c. L^* ←

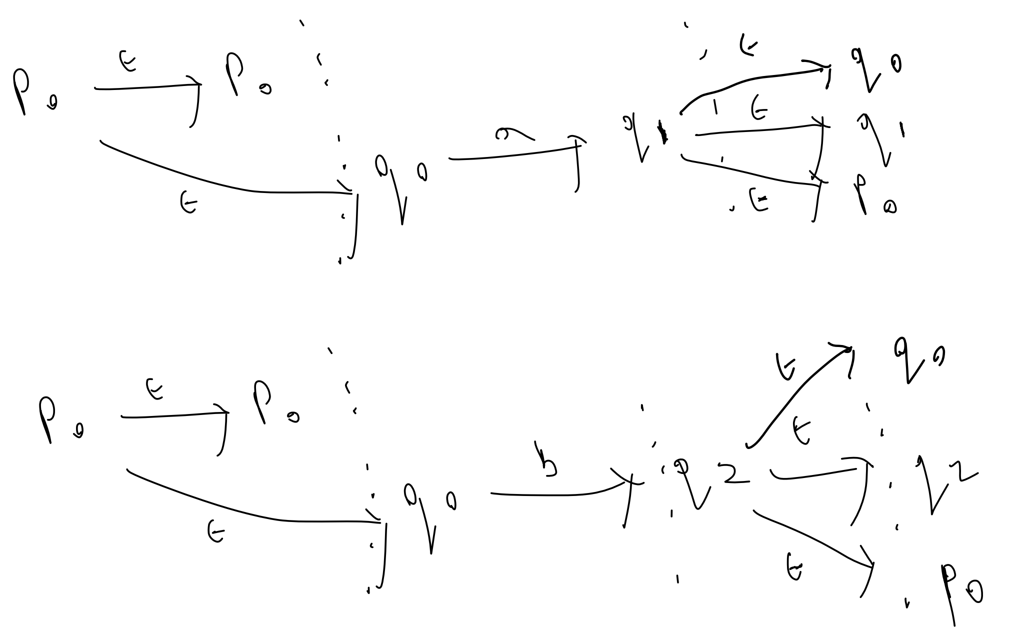
Example: Kleene star



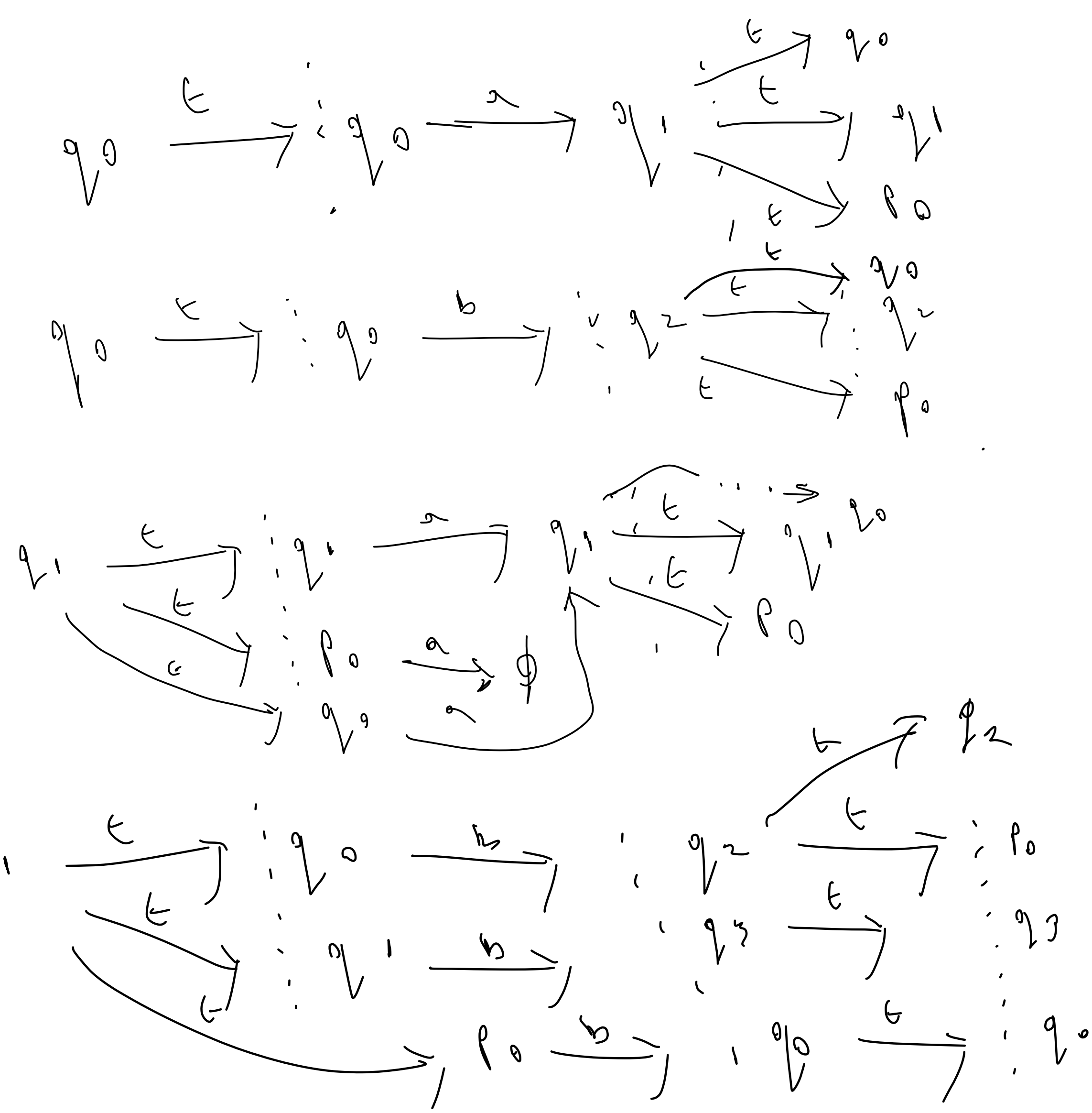


1.c. cont. RMET

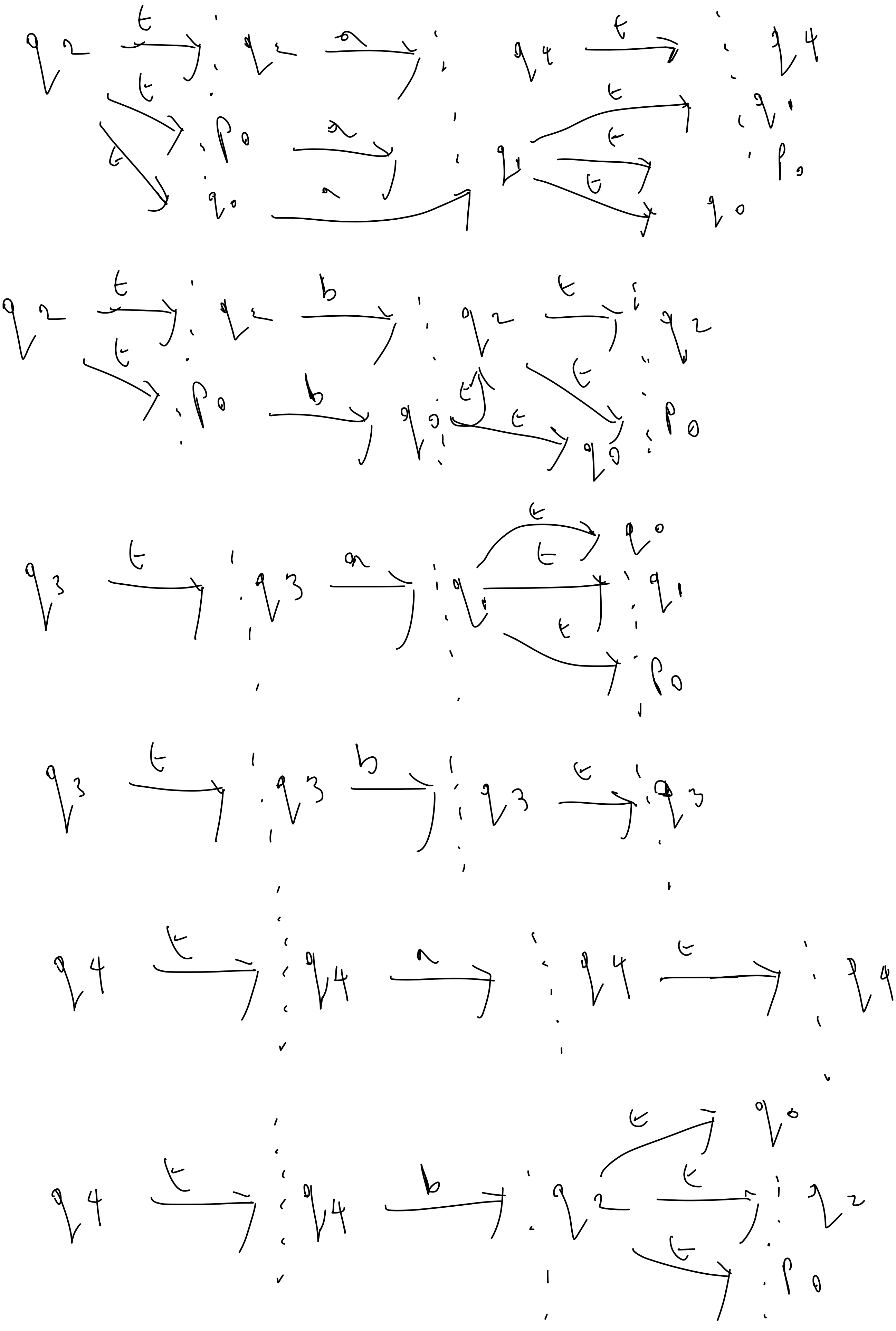
ε-closure (direct trans (ε-closure (p), symbol))



Note: 1st & 2nd ε closures can have many levels!



RMET cont.



1.c. cont

$\epsilon\text{-closure}(\underbrace{d_{in}}_{\text{direct trans}}(\epsilon\text{-clos}(q), a))$

	ε-closure
F p_0	q_0, p_0
q_0	q_0
F q_1	p_0, q_1, q_0
F q_2	p_0, q_2, q_0
q_3	q_3
q_4	q_4

NFA From here

	a	b
F p_0	q_0, q_1, p_0	q_0, q_2, p_0
q_0	q_0, q_1, p_0	q_0, q_2, p_0
F q_1	q_0, q_1, p_0	q_0, p_0, q_3, q_2
F q_2	q_0, p_0, q_1, q_4	q_2, q_0, p_0
?? q_3	q_0, q_1, p_0	q_3
q_4	q_4	q_0, q_2, p_0

* Note: include implicit empty cases

1.c.

NFD

√ — Already in Queue

√ — Already defined

Queue: [[p_0]]

$[p_0] \xrightarrow{a} [q_0, q_1, p_0]$

$[p_0] \xrightarrow{b} [q_0, q_2, p_0]$

Queue: [[q_0, q_1, p_0], [q_0, q_2, p_0]]

$[q_0, q_1, p_0] \xrightarrow{a} [q_0, q_1, p_0]$

$[q_0, q_1, p_0] \xrightarrow{b} [q_0, p_0, q_3, q_2]$

Queue: [[q_0, q_2, p_0], [q_0, p_0, q_3, q_2]]

$[q_0, q_2, p_0] \xrightarrow{a} [q_0, p_0, q_1, q_4]$

$[q_0, q_2, p_0] \xrightarrow{b} [q_0, q_2, p_0]$

Queue: [[$q_0, \cancel{p_0, q_3}, q_2$], [$q_0, \cancel{p_0}, q_1, q_4$]]

$[q_0, p_0, q_3, q_2] \xrightarrow{a} [q_0, p_0, q_1, q_4]$

$[q_0, p_0, q_3, q_2] \xrightarrow{b} [q_0, q_2, p_0, q_3]$

$[q_0, p_0, q_1, q_4] \xrightarrow{a} [q_0, q_1, p_0, q_4]$

$[q_0, p_0, q_1, q_4] \xrightarrow{b} (q_0, p_0, q_3, q_2)$

1. c.

Final DFA:

F - final state
S - start state

	a	b
S F [p0]	[q0, q1, p0]	[q0, q2, p0]
F [q0, q1, p0]	[q0, q1, p0]	[q0, p0, q3, q2]
F [q0, q2, p0]	[q0, p0, q1, q4]	[q0, q2, p0]
[q0, p0, q3, q2] F	[q0, p0, q1, q4]	[q0, q2, p0, q3]
[q0, p0, q1, q4] F	[q0, q1, p0, q4]	[q0, p0, q3, q2]

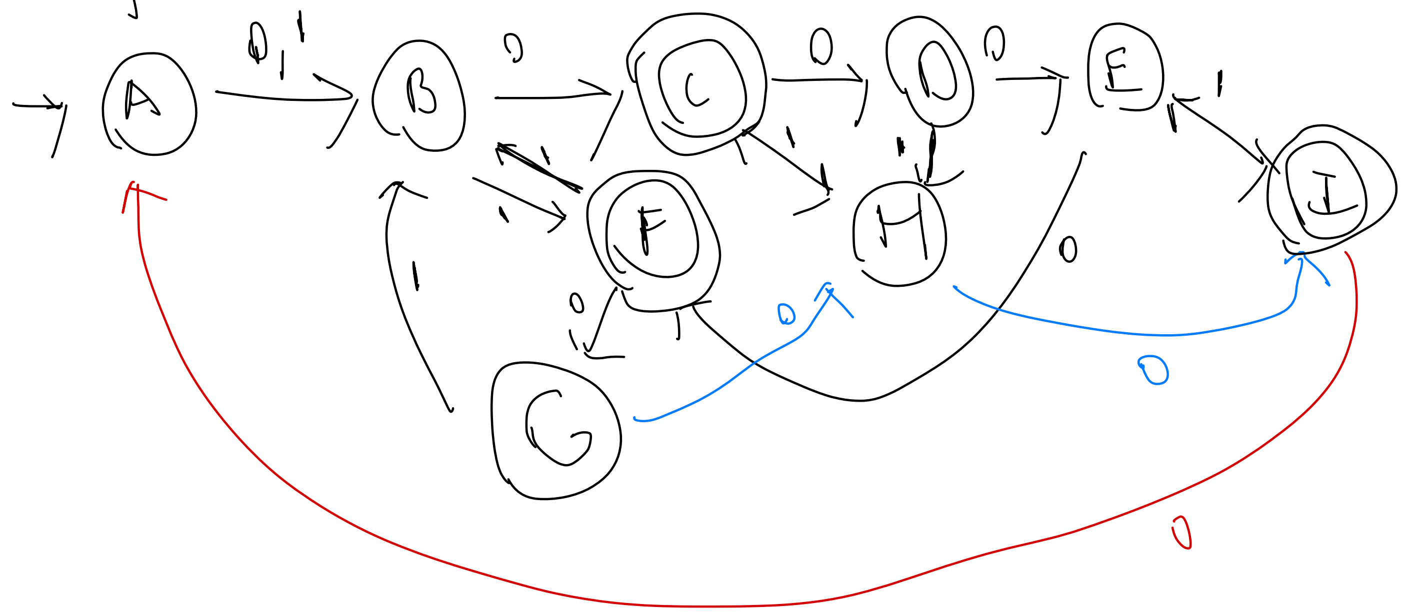
Program Output for reference:

```
DFA.obj __str__ method: Debug print:
start [p0]
final [[p0,q0,q1,q4],[p0,q0,q1],[p0,q0,q2,q3],[p0,q0,q2],[p0]]
trans [p0]:b:[p0,q0,q2]
trans [p0]:a:[p0,q0,q1]
trans [p0,q0,q2]:b:[p0,q0,q2]
trans [p0,q0,q2]:a:[p0,q0,q1,q4]
trans [p0,q0,q1]:b:[p0,q0,q2,q3]
trans [p0,q0,q1]:a:[p0,q0,q1]
trans [p0,q0,q1,q4]:b:[p0,q0,q2,q3]
trans [p0,q0,q1,q4]:a:[p0,q0,q1,q4]
trans [p0,q0,q2,q3]:b:[p0,q0,q2,q3]
trans [p0,q0,q2,q3]:a:[p0,q0,q1,q4]
```

2. Min following DFA:

$\Sigma = \{0, 1\}$

Original DFA:



staircase

	A	B	C	D	E	F	G	H
B	•							
C	x	x						
D	•	•	x					
E	•	•	x	•				
F	x	x	•	x	x			
G	•	•	x	•	•	x		
H	•	•	x	•	•	x	•	
I	x	x	•	x	x	•	x	x

$$\frac{9^2 - 9}{2} = 36$$

• - valid indisting. pair

x - disting.

Final:

C F I

$(A, B) \xrightarrow{0} (B, C)$ ✓

Pair Validation

$(A, B) \xrightarrow{0} (B, C)$ ✓

$(A, B) \xrightarrow{1} (B, F)$

$(G, H) \xrightarrow{0} (H, I)$

$(F, G) \xrightarrow{0} (G, H)$

B

$(D, H) \xrightarrow{1} (H, C)$ ✓

$$\begin{tikzcd} & (D, H) \arrow[dl, "0"'] \arrow[dr, "0"] & \\ E \arrow[rr, bend right, "1"'] & & I \end{tikzcd}$$

✓

$$\begin{tikzcd} & (E, G) \arrow[dl, "1"'] \arrow[dr, "1"] & \\ I & & B \arrow[d, "0"] \\ & & C \end{tikzcd}$$

(Marked

$$\begin{tikzcd} & (B, G) \arrow[dl, "0"'] \arrow[dr, "0"] & \\ C \arrow[rr, "1"] & & H \arrow[d, "0"] \\ & & I \end{tikzcd}$$

(marked)

Pair Validation

(B, D)

$B \xrightarrow{a} C \qquad D \xrightarrow{b} E$

(D, E)

$D \xrightarrow{d} H \qquad E \xrightarrow{b} I$

$H \xrightarrow{a} I$

(A, E)

$A \xrightarrow{a} B \qquad E \xrightarrow{b} I$

$B \rightarrow C, F$

•'s → X's

staircase

• – valid indisting. pair

X – disting.

$$\frac{9^2 - 9}{2} = 36$$

	A	B	C	D	E	F	G	H
B	X							
C	X	X						
D	•	X	X					
E	X	•	X	X				
F	X	X	•	X	X			
G	•	X	X	•	X	X		
H	X	•	X	X	•	X	X	
I	X	X	•	X	X	•	X	X

Final:

C F I

Remaining indisting. pairs:

{ (A, G), (A, D), (B, E), (C, I), (C, F), (D, G), (E, H),

(F, I), (B, H) }

Remaining indisting. pairs:

{ (A, G), (A, D), (B, E), (C, I), (C, F), (D, G), (E, H), (F, I), (B, H) }

(A, G), (A, D), (D, G) => {A, G, D}

(B, E), (E, H), (B, H) => {B, E, H}

(C, I), (C, F), (F, I) => {C, I, F}

Final min DFA:

3.

Figure out states first

RMET code:

- create new NFA obj
- two deltas
- transfer states & alphabet
- suggestion: write ε-closure method to find ε-closures
  - take state as a parameter

first ε-closure on solution should look like:

$q_0: \{q_1, q_4, q_6\} \ldots$

see networkx package

first elipison closure