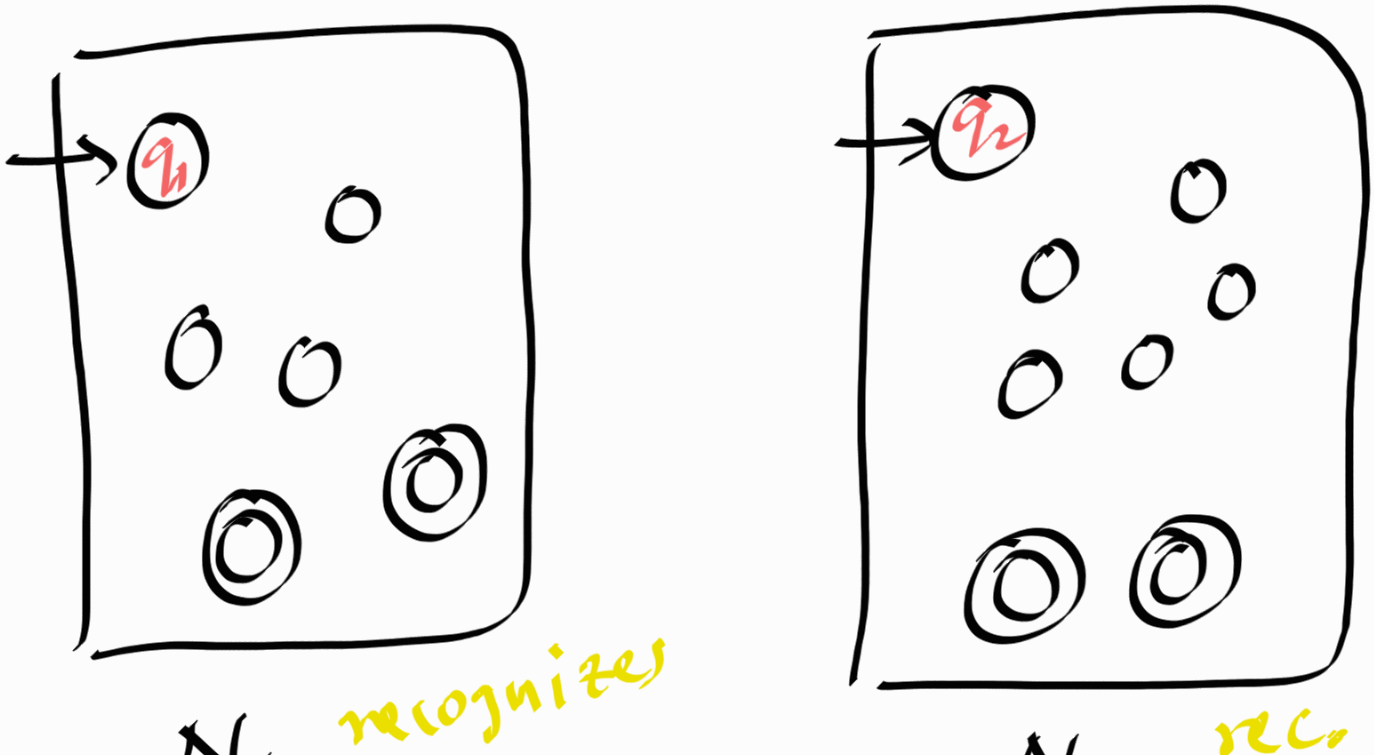
Corollary 1.40

A language is regular iff some NFA accepts it.

Theorem 1.45

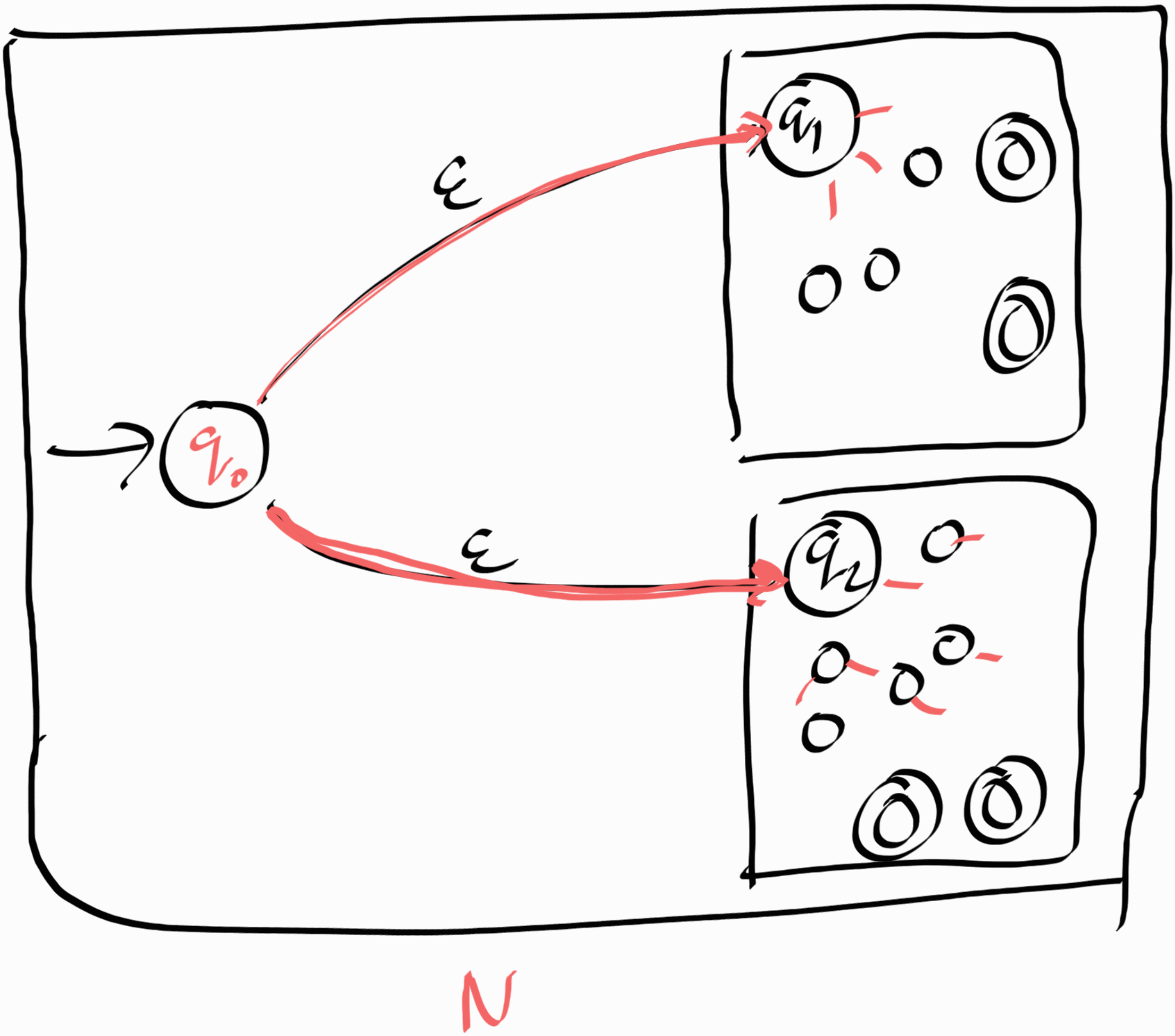
The class of regular languages is closed under the union operation.

Proof Proof idea



N_1 recognizes A_1 N_2 rec. A_2

Basically we want to create a machine that accepts strings in A_1 and strings in A_2



Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1

Let $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognizes A_2

we construct the NFA N,

$N = (Q, \Sigma, \delta, q_0, F)$ from N_1 & N_2

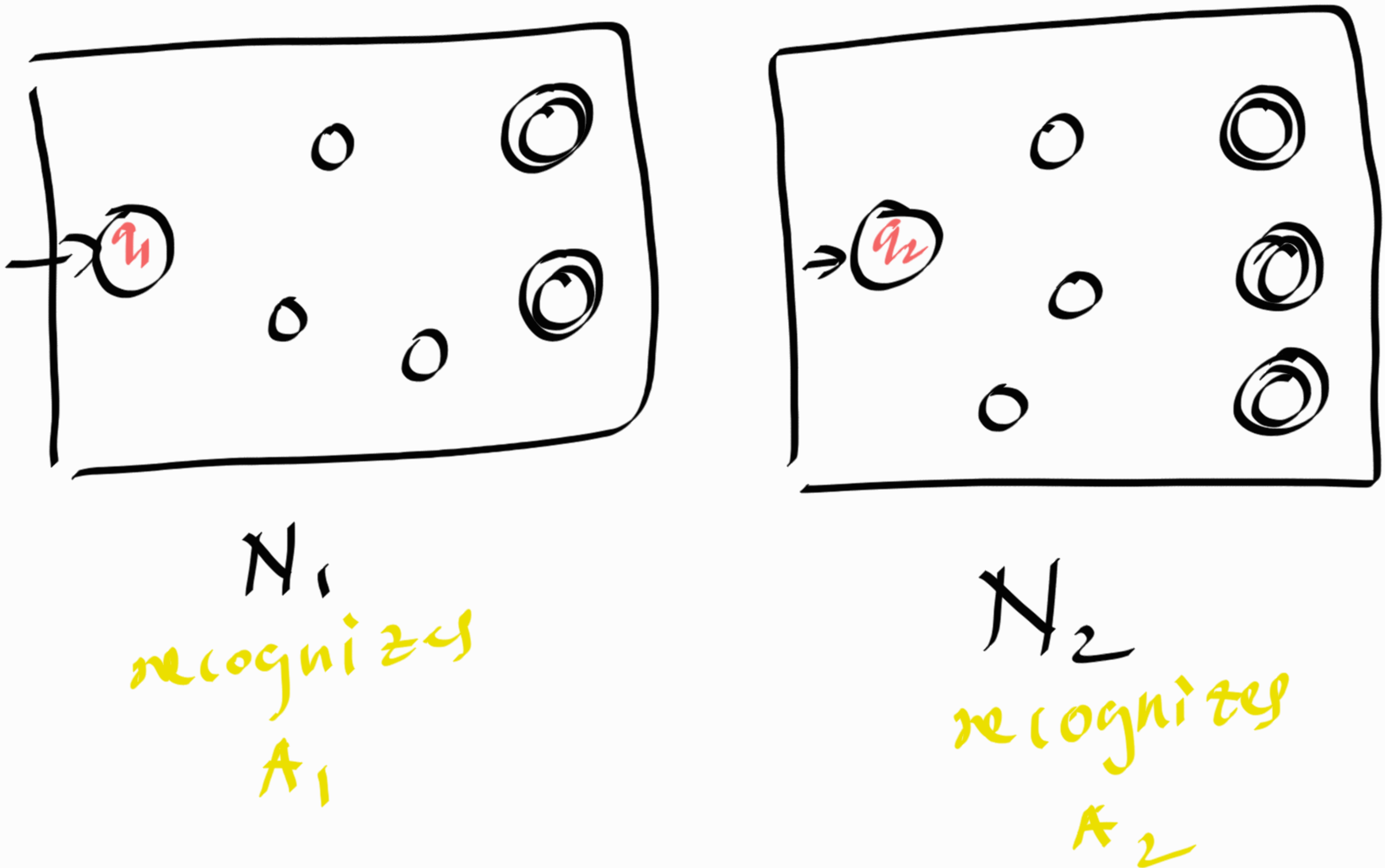
1. $Q = \{q_0\} \cup Q_1 \cup Q_2$
2. $q_0 \in Q$ is the new starting state
3. $F = F_1 \cup F_2$
4.

$$\delta(q,a) \begin{cases} \delta_1(q,a) & \text{if } q \in Q_1 \\ \delta_2(q,a) & \text{if } q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \text{ and } a = \varepsilon \\ \phi & q = q_0 \text{ and } a \neq \varepsilon \end{cases}$$

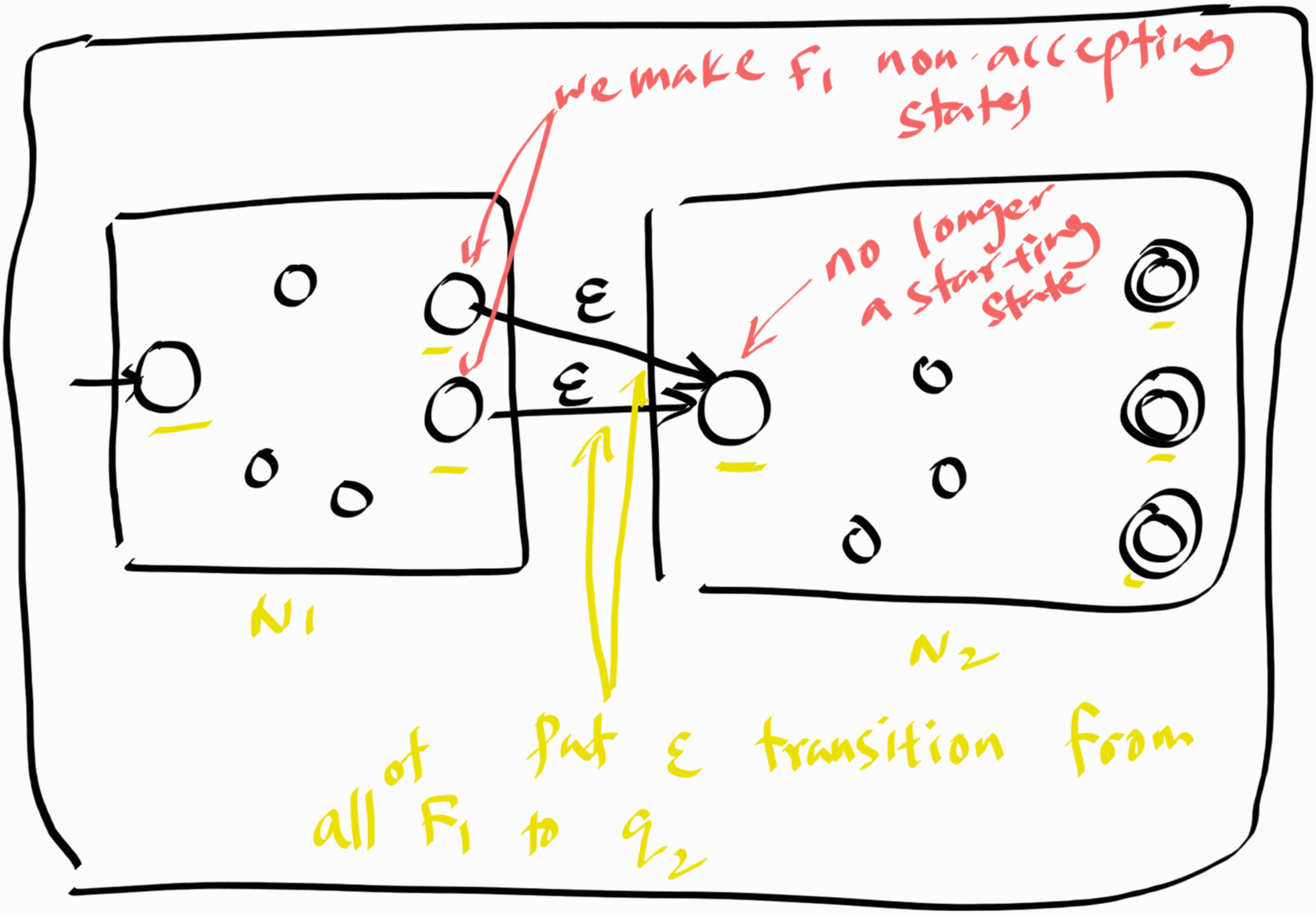
Theorem 1.47

The class of regular languages are closed under concatenation operation.

Proof: Basic Idea.



we want to create a NFA that recognizes $A_1 \circ A_2$



N recognizes $A_1 \circ A_2$

Use the N_1 machine to check the prefix of a input string is accepted.

Then put ε transitions from N_1 accepting state to N_2 starting state.

But we need to states the N_1 accepting states to non-accepting states.

Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1

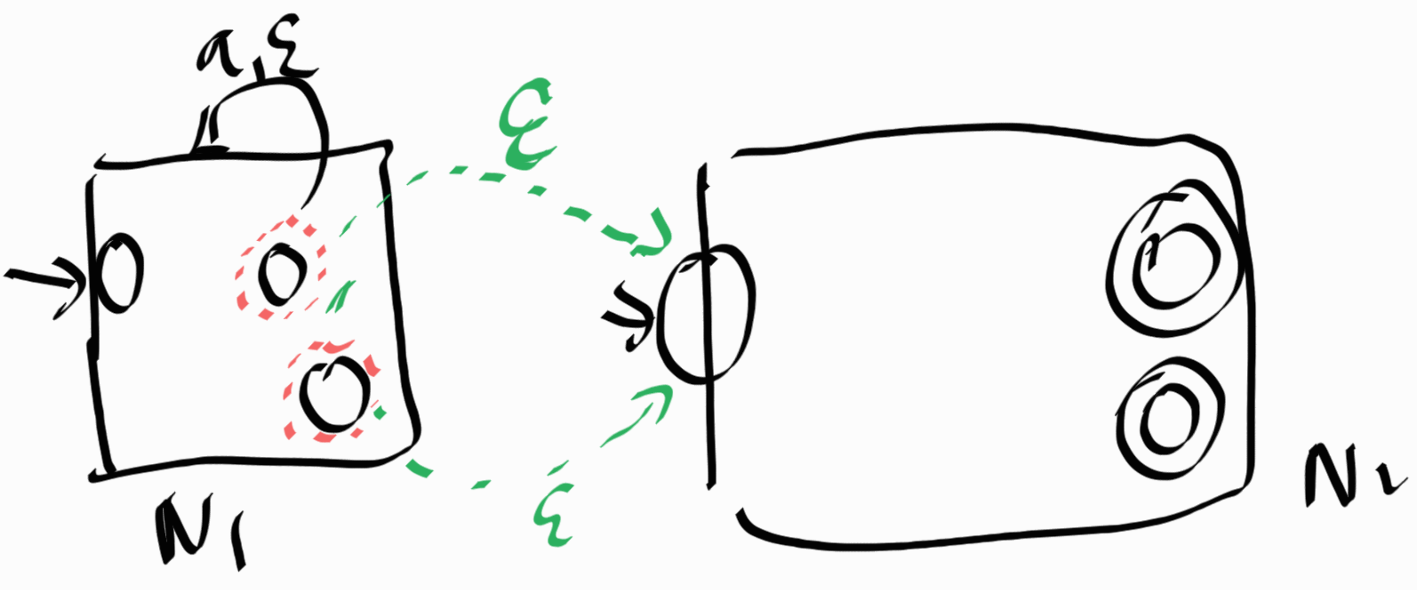
Let $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize A_2

Then we construct $N = (Q, \Sigma, \delta, q_0, F)$ from N_1 and N_2 to recognize $A_1 \circ A_2$

1. $Q = Q \cup Q_2$
2. $q_0 \in q_1$
3. $F = F_2$ * Important

4.

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & \text{if } q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & \text{if } q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & \text{if } q \in F_1 \text{ and } a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$



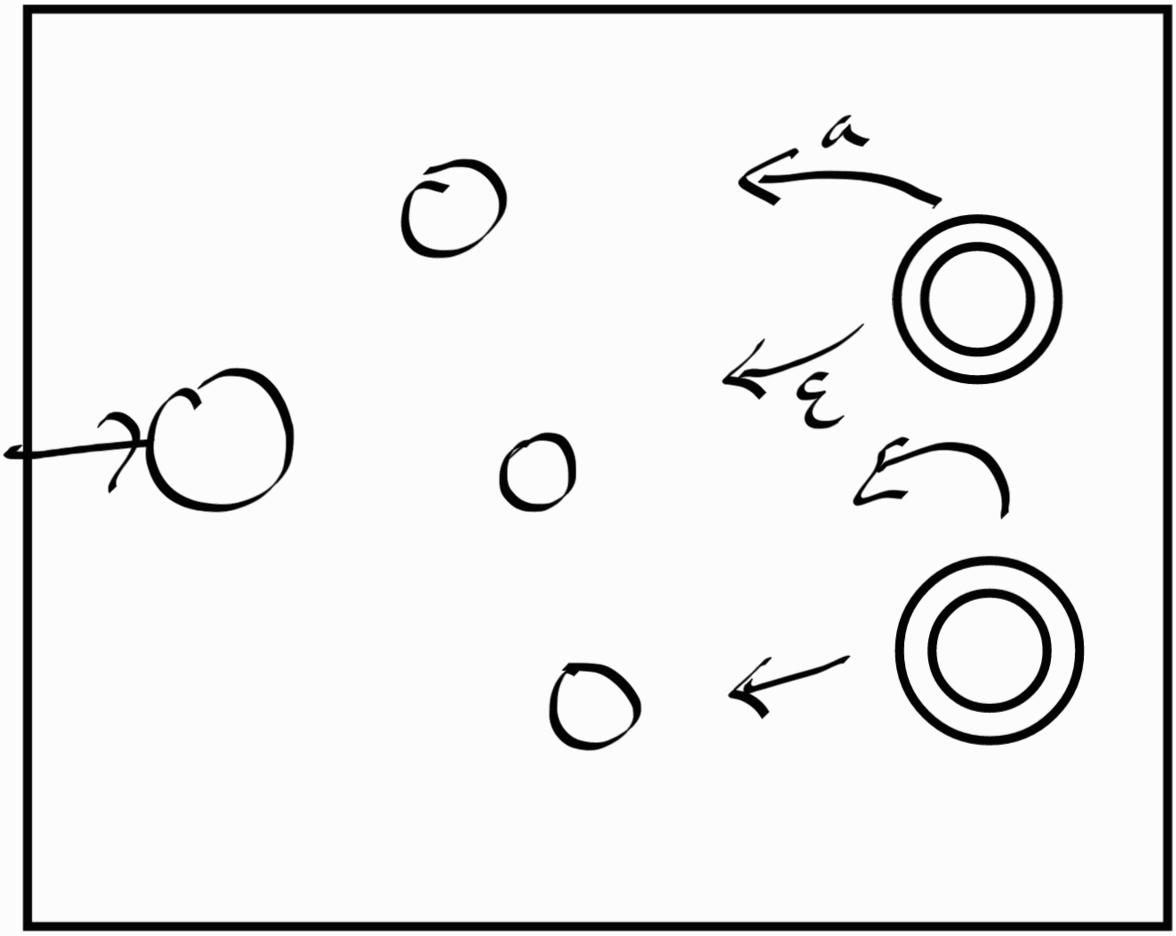
This pooof is kind of like building a larger entity with smaller entities.

Theorem 1.49

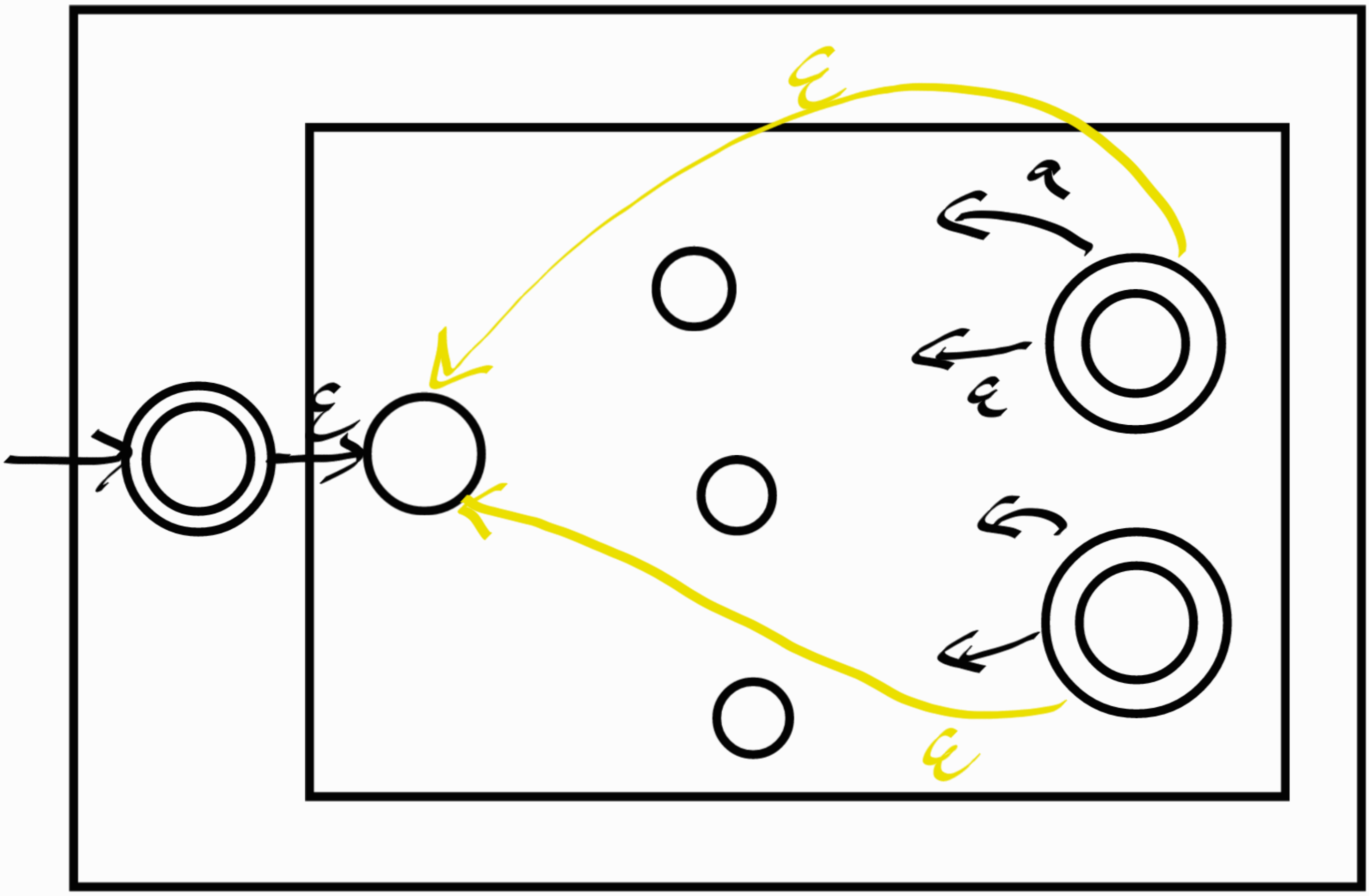
The class of regular languages is closed under star operation.

I will not go into too much details.
Here we learn the basic idea.

Check the book for more details.



N_1



Basically idea is to reset the status of an execution path to starting state whenever it reaches a previously accepting state