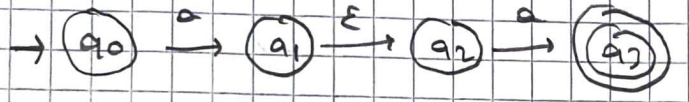
→ At any time, it is not possible to say that NFA is only this state, it can be at multiple states at a given time.

Epsilon (Null) Transitions:

→ Epsilon transitionda NFA herhangi bir input alphabet harcamadan, bir stateden bir başka state'e gecebilmektedir.

Example:

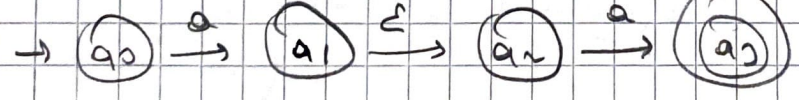
aa



* İlk önce ilk symbol okunur ve NFA $q_0 \rightarrow q_1$'e gecer.
* Okunan symbol ε oldığından NFA direkt olarak $q_1 \rightarrow q_2$ transitionu yapar.
* ε transitionda "machine is simultaneously in q_1 and q_2 at the same time" denir.

Example:

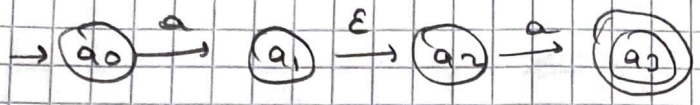
aaa



* Bu durumda NFA reject olur, accept state'e tüm symboller harcanadan gelmiyor.

Example:

How many computational path exists in this NFA? L = {aa}



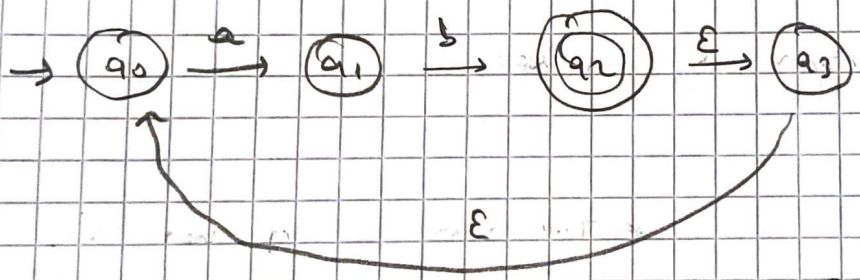
Path 1: q0, q1, q2, q3

Path 2: q0, q1, halts

↳ We don't have to follow epsilon transition.

Example:

What is the language recognized by this NFA?



Path 1: ab

Path 2: ab ab

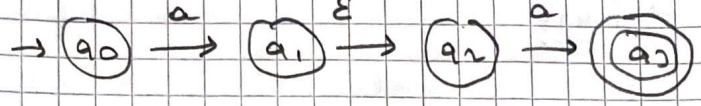
⋮

* Therefore, language recognized by this NFA is:

$$L = \{ (ab)^n : n \geq 1 \}$$

n must be ≥ 1 since "0" means that stay in q0 and q0 is not an accept state.

→ Epsilon transitionda, machine aynı anda iki tane state içerisinde bulunabilmektedir.

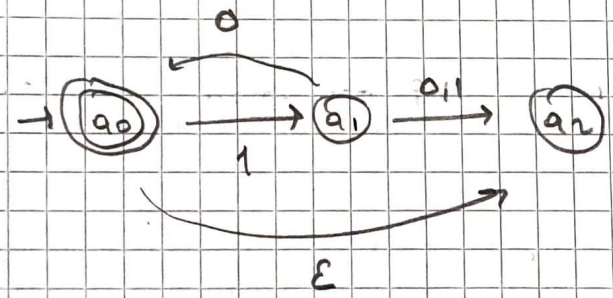


* Geçişlerin visual representation'u ise

$$\{q_0\} \xrightarrow{a} \{q_1, q_2\} \xrightarrow{a} \{q_3\}$$

şeklinde olur fakat q_1'den sonra gidecek başka bir yer olmadığından automaton halts in q_1 branch denir.

Example: what is the language recognized by this NFA.



* The smallest string that recognized by this NFA is empty string (ε)

* ε için 2 durum geçerlidir. Machine çalışmaya başladığında durması, ya da ε yolunu takip ederek q_2'ye ulaşması.

$$L = \{\varepsilon, 10, 1010, 101010, \dots\} = \{10\}^*$$

! How many computational path exists for the string 10?

Answer: • for string 10

Path 1: $q_0 \to q_1 \to q_0$ (Accept path)

Path 2: $q_0 \to q_1 \to q_2$ (Reject path)

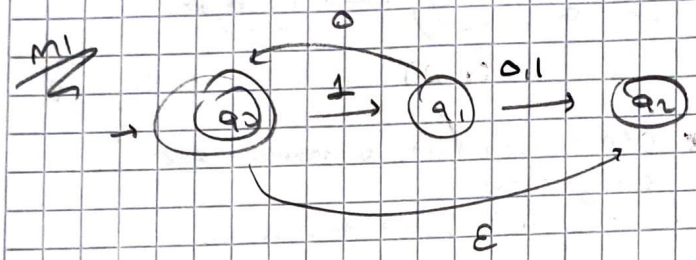
Path 3: $q_0 \to q_2$ (Halts, reject)

Path 4: $q_0 \to q_1 \to q_0 \to q_2$ (Reject)

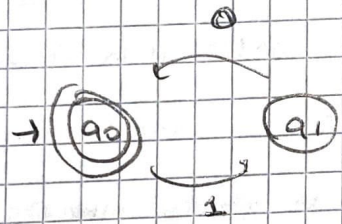
Not: Eğer $L = \{10\}^*$ yazdığımızda * operatörünü de yapıyorsak, ε transition'ı da kabul etmiş oluruz.

→ Bize iki tane NFA verilmiş olsun. Bu NFA'lar aşağıdaki gibi:

M1



M2

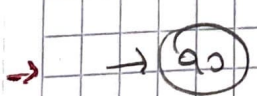


⚠ M_1 ve M_2 birbirinden tamamen farklı 2 NFA'dır. Ancak M_1 ve M_2 aynı language üzerinde tanımlıdır.

- Dolayısıyla M_1 is equivalent to M_2 denir.

* Farklı NFA'lar ya da aynı language üzerinde tanımlanan birimlere equivalent denmektedir.

Simple NFAs:

→ 

$L(M_1) = \{\}$

* This machine will halt for any input string including ε.

→ 

$L(M_2) = \{\varepsilon\}$

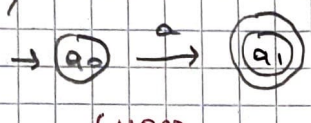
* This machine will halt for any string except ε.

W4-L2

→ Daha az state kullanarak bir DFA'i NFA'e dönüştürebiliriz.

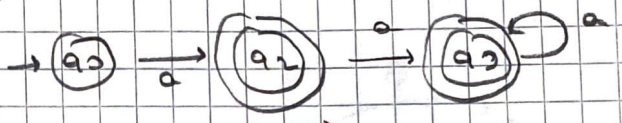
→ Mesela M_1 bir NFA ve M_2 bir DFA olsun. M_1 ve M_2'nin tanımlandığı diller ise $L(M_1) = L(M_2) = \{a\}$ olsun.

M_1:



(NFA)

M_2:



(DFA)

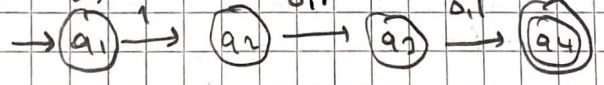
* DFA requires that a transition for every state.

Example:

Let A be the language consisting of all strings over $\{0,1\}$ containing a 1 in the third position from the end. (e.g, 000100 is in A but 0011 (UNDF) Construct both NFA and DFA for it.

* Means that if the last third element is 1, it will be accepted, otherwise it's rejected.

NFA:



DFA:

Formal Definition of NFAs:

→ NFA'ler de 5 tane component'ten oluşmaktadır.

NFA: $(Q, \Sigma, \delta, q_0, F)$

Q: set of states

Σ: alphabets

δ: transition function

q_0: initial state

F: set of accept state

2. Transition function δ

→ DFA durumunda transition function'ın birinci parametresinde current state, ikinci parametresinde ise symbol vardı. İşlemin sonucunda ise o symbol'ü takip ederek ulaşacağımız state mevcuttu.

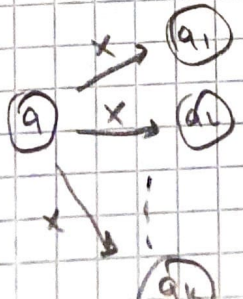
$$\delta(q_0, x) = \{q_1\}$$

* There is exactly one way in DFA.

→ NFA'de ise current state'ten ilgili sembolü okuduğumuzda kendisi başka bir state'le gidebilirim.

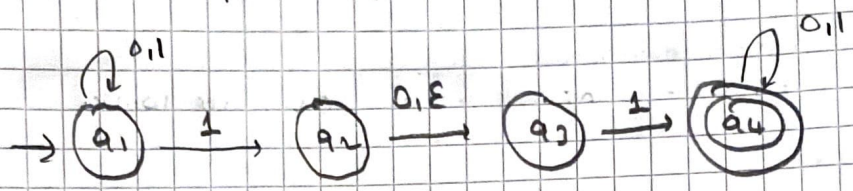
$$\delta(q, x) = \{q_1, q_2, q_3, \dots, q_k\}$$

$$0 \leq |\delta(q, x)| \leq |Q| \quad \rightarrow \text{\# of states}$$



Example:

Find all components of this NFA M_1



1: Q: is the set of all states

$$Q = \{q_1, q_2, q_3, q_4\}$$

2: Σ: is the alphabet.

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

3: δ: Transition function

	0	1	ε
q_1	q_1	$\{q_1, q_2\}$	∅
q_2	q_3	∅	q_3
q_3	∅	q_4	∅
q_4	q_4	q_4	∅

4: q_1 is the start state.

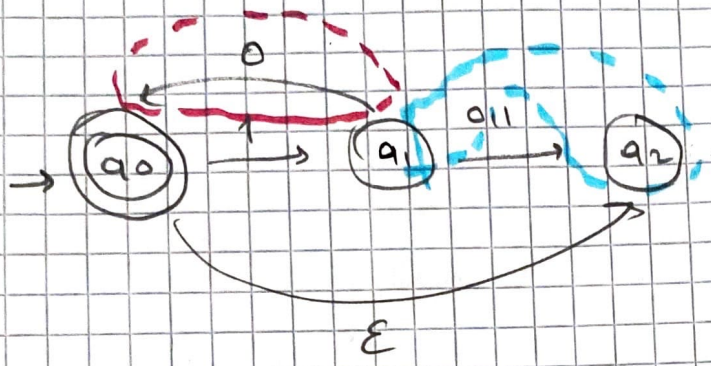
5: F: accept states

$$F = \{q_4\}$$

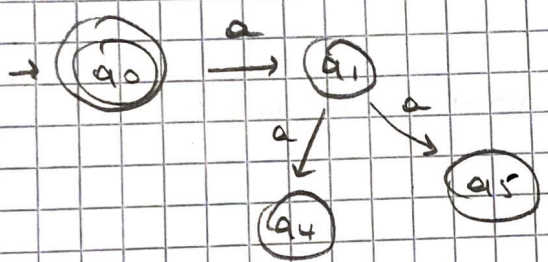
Example:

$$\delta(q_1, 0) = \{q_0, q_2\}$$

* q_1'den 0 okunarak hem q_0'a hem de q_2'a gidilebilir.



Example:

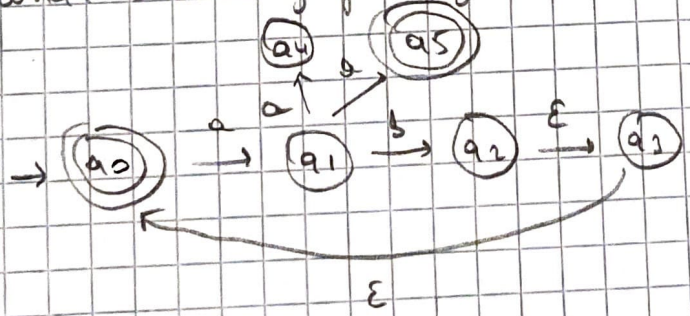


$$\delta^*(q_0, aa) = \{q_4, q_5\}$$

* Starts from q_0, read aa & machine can be in both q_4 and q_5 simultaneously.

Example:

what is the language recognized by this NFA?



$$L(M) = \{\varepsilon, ab, abaa, abab, ababaa, \cdots\}$$

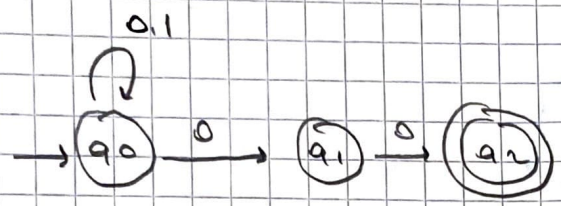
$$L(M) = \{ab\}^* \cup \{ab\}^* \{aa\}$$

* including ε.

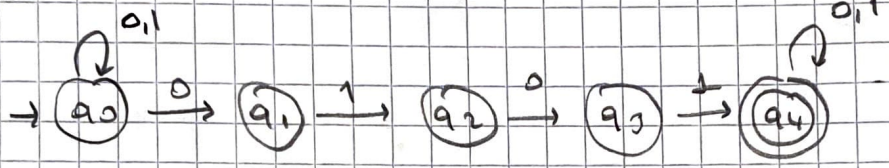
Example:

Give state diagrams of NFAs. In all parts, the alphabet is $\{0,1\}$

a. $L = \{ w \mid w \text{ ends with } 00 \}$ with $\Sigma = \{0,1\}$



b. $L = \{ w \mid w \text{ contains substring } 0101 \}$ with 5 states.



c. $L = \{ w \mid w \text{ contains an even \# of 0s, or contains exactly two 1s} \}$ with 6 states.