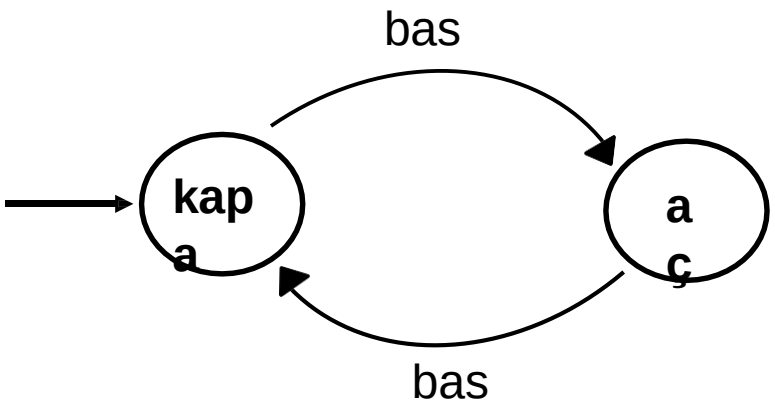
Otomata Teorisi

Hafta 2:
Sonlu Otomata (1.Bölüm)



Hafta 2
Plan

1. Bir Sonlu Otomata Orneği
2. Sonlu Otomatanin Esasları
3. Sonlu Otomatanın Resmi Gösterimi
4. Nondeterministik Sonlu Otomata

I Bir Sonlu Otomasyon Örneği

Farz edelim ki bir makineye bağlı bır turnikemiz var ve bu makineye 25 kuruş atılınca turnike açılıyor.

Makine 5, 10 ve 25 kuruşluk madeni para kabul ediyor ve para üstü vermiyor.

Müşteri 5, 10 ve 25 kuruşluk paraları atarken her bir anda turnikenin açık yada kapalı oluşuna karar veren bır sonlu otomasyon tasarlıyalım.

Durumlar:

q_0: Makinede para yok.

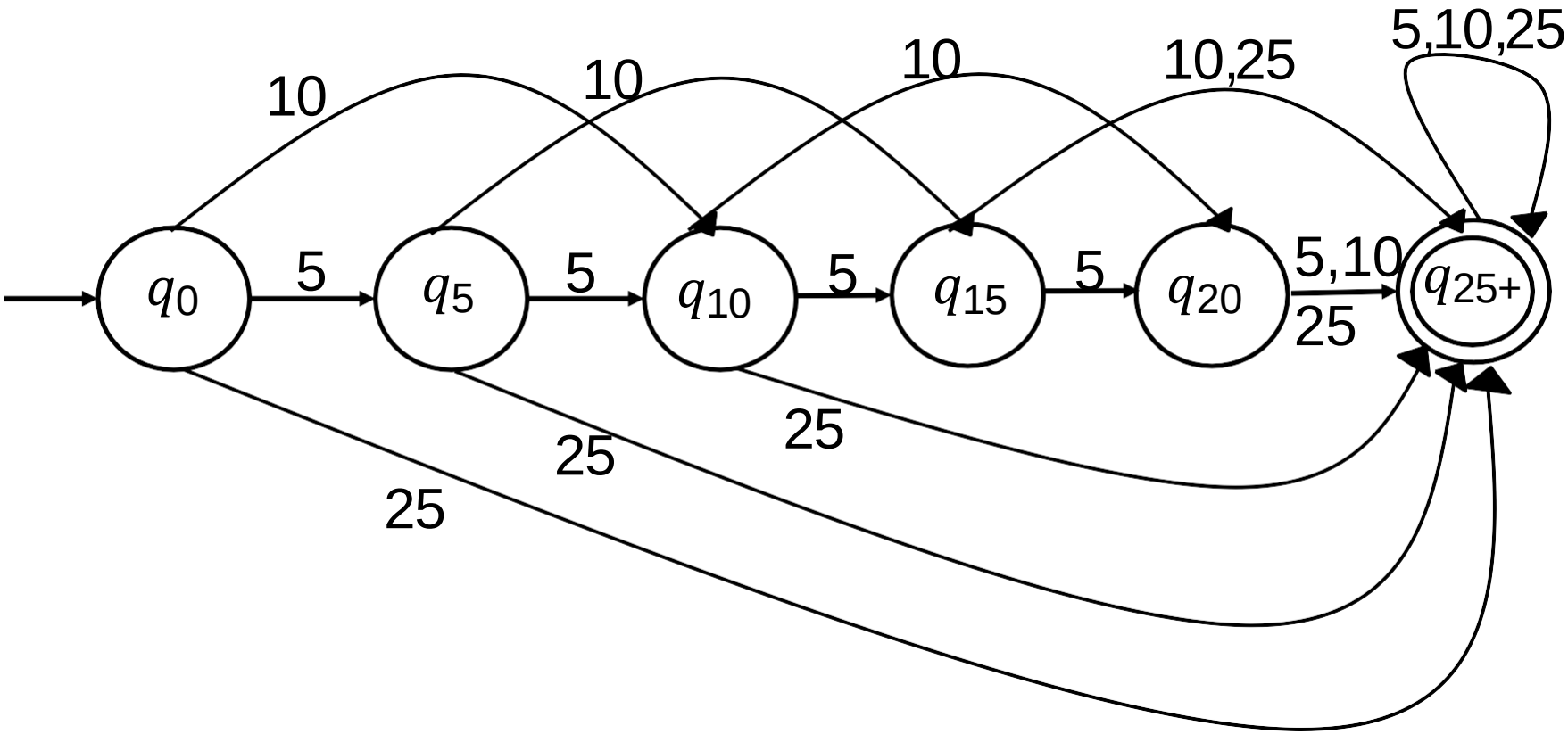
q_5: Makinede 5 kuruş var.

q_{10}: Makinede 10 kuruş var.

q_{15}: Makinede 15 kuruş var.

q_{20}: Makinede 20 kuruş var.

q_{25+}: Makinede 25 kuruş yada daha fazlası var.



Burada makine (bilgisayar) yalnizca herhangi bir anda hangi durumda olduğunu hatırlamak zorunda. O yuzden yalnizca çok küçük bır hafızaya (log 6 ≈ 3 bit) ihtiyaç duyar.

Dil: Bir Σ alfabesinden elde edilebilecek tum kelimelerin kumesi olan Σ^*un bir alt kumesine dil denir.

or. $\Sigma = \{0,1\}$ alfabesi icin bu alfabeden uretilebilecek tum kelimelerin kumesi:

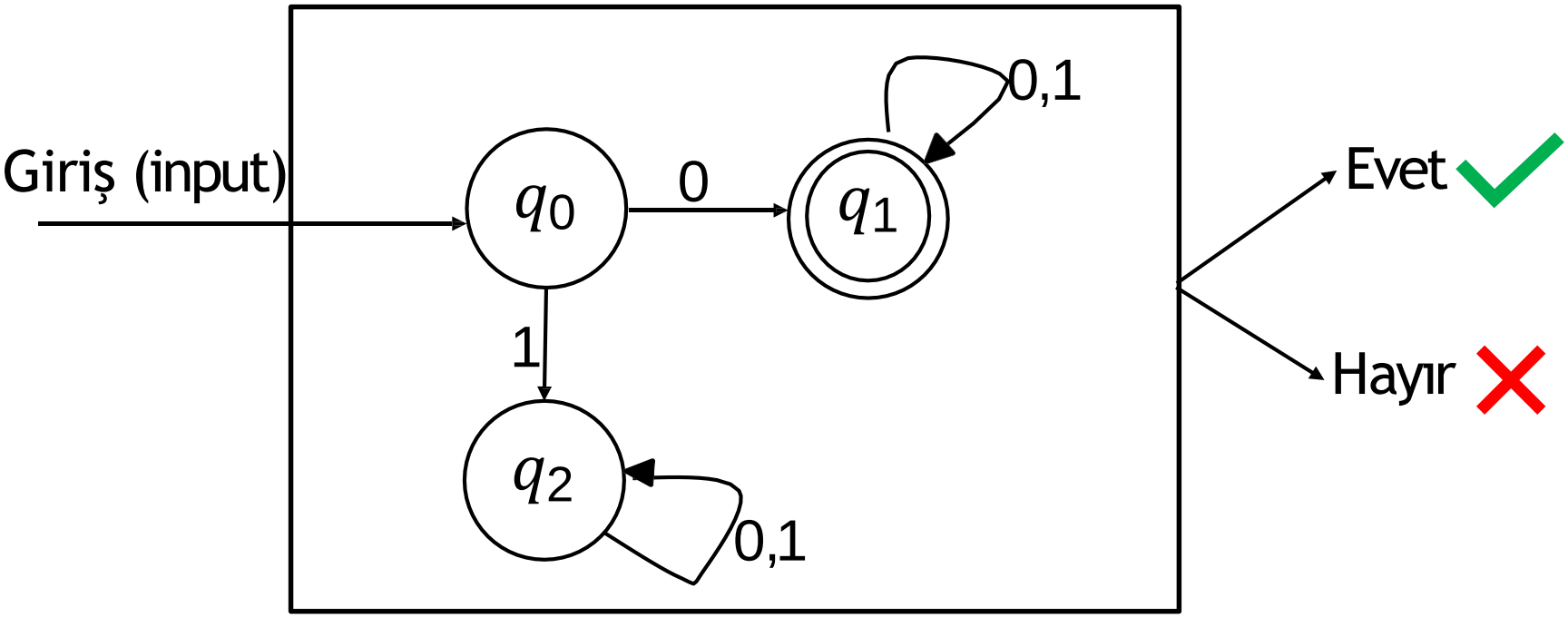
$$\Sigma^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 111, 010, \ldots\}$$

Buradan bir L dili olusturulalim, oyleki bu L dili "0" ile baslayan kelimlerin dili olsun:

$$L = \{0, 00, 01, 000, 0101, \ldots\}$$

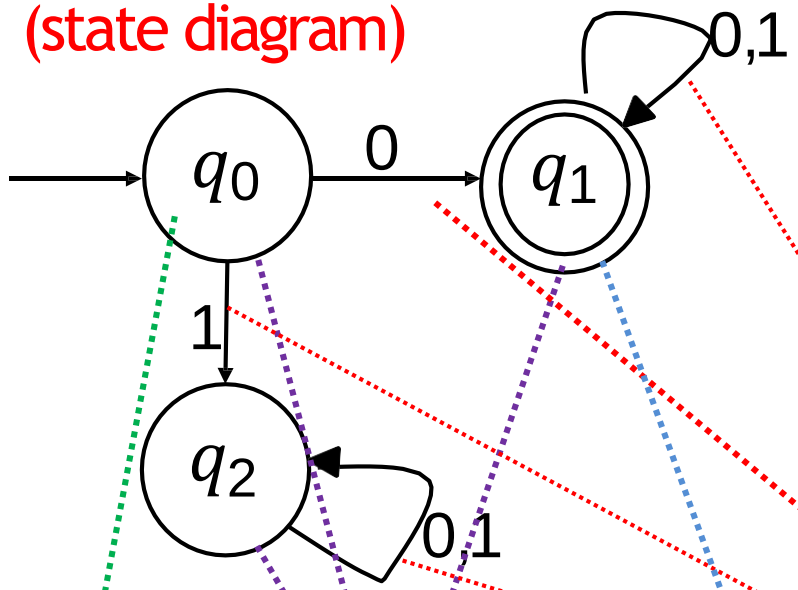
($L = \{s \in \Sigma^* | \; s'nin\ ilk harfi\ 0\}$ seklinde ortak ozellikler yontemi ile de gosterebiliriz.)

İşte otomasyonda amacımız verilen bir dilin kelimelerini otomatik olarak algilayan-taniyan (recognize eden) bir soyut makine icat etmektir. İcat edilen makine verilen bir kelimeyi işler (process eder), ve sonuçta bu kelime dile ait mi diye karar verir (evet yada hayır der).

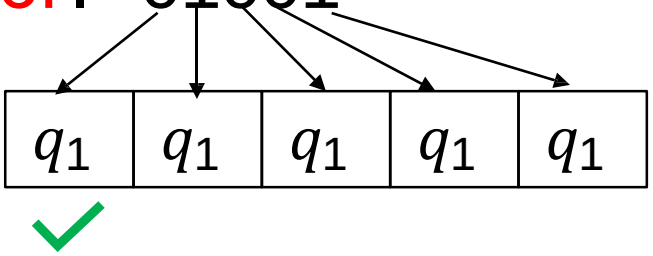


durum diagrami

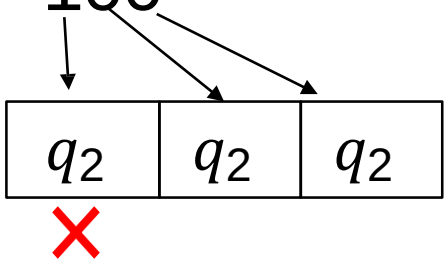
(state diagram)



ör. 01001



ör. 100



durum (state)

geciş (transition)

baslangıç durumu

(initial state)

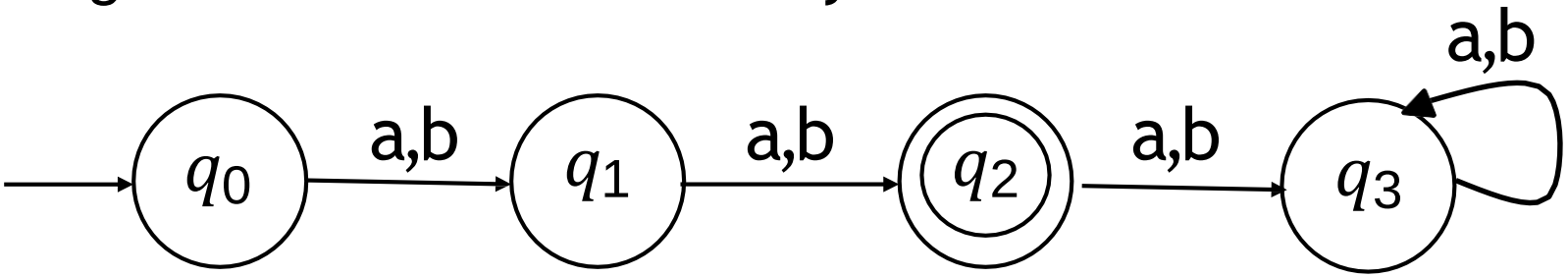
(boş ok ile gösterilir)

final (kabul) durumu

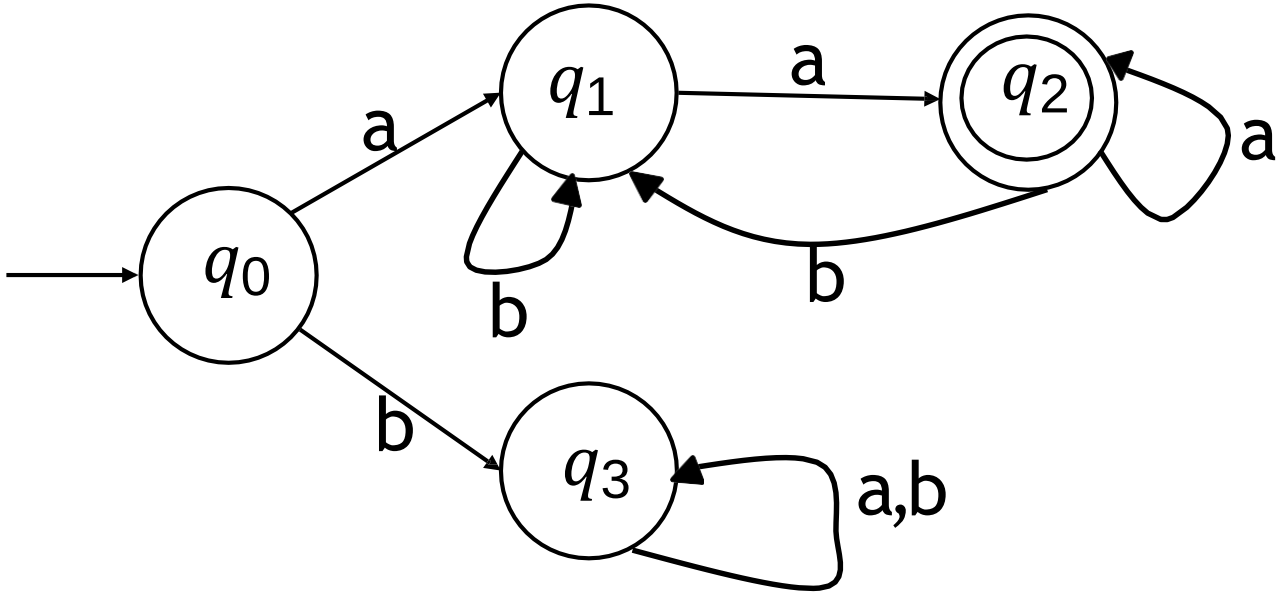
(final (acceptance) state)

(çift halka ile gosterilir)

ör. Σ = $\{a, b\}$ alfabesi kullanılarak üretilen kelimelerden 2 uzunluğundaki kelimelerin dilini tanıyan sonlu otomata:



ör. Σ = $\{a, b\}$ alfabesi kullanılarak üretilen kelimerlerden a ile başlayıp a ile biten kelimerlerin dilini tanıyan sonlu otomata:



Sonlu Otomatinin Formal Tanımı:

Bir sonlu otomata 5-li sıradır ve $(Q, \Sigma, \delta, q_0, F)$ ile gösterilir. Burada:

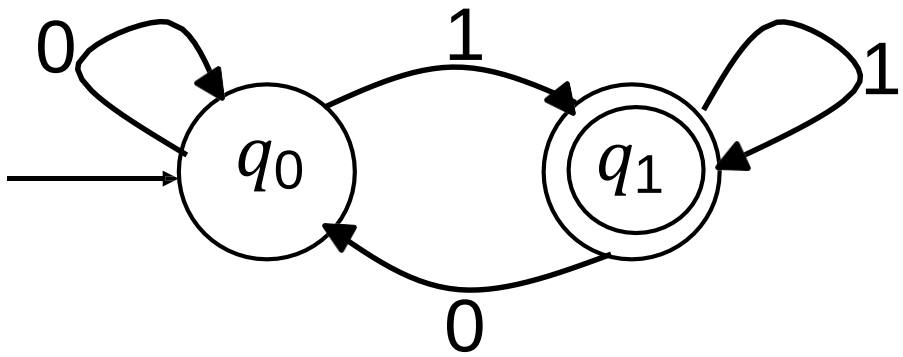
1. Q tüm durumları içeren sonlu bir kümedir,
2. Σ kullandığımız harfleri (inputları) içeren alfabedir,
3. $\delta: Q \times \Sigma \rightarrow Q$ geçiş fonksiyonudur (transition function),
4. $q_0 \in Q$ baslangıç durumudur,
5. $F \subseteq Q$ final durumlari içeren kümedir.

Not 1. δ (geciş fonksiyonu) bir fonksiyondur. Fonksiyon olma tanımı gereği $Q \times \Sigma$ tanım kümesindeki her elemanı (her sıralı ikiliyi) Q'da bir elemana götürür. Yani her durum ve her harf ikilisine karşılık bir durum vardir.

Not 2. F bir kümedir. Yani eleman sayisi 1'den fazla olabilir. Yani bir otomatanin 1'den fazla final durumu (kabul durumu) olabilir.

Not 3. Sonlu otomata dememizin nedeni sonlu sayıda durum içermesi

ör.



$Q = \{q_0, q_1\}$

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

$F = \{q_1\}$

harfler

δ	0	1
q_0	q_0	q_1
q_1	q_0	q_1

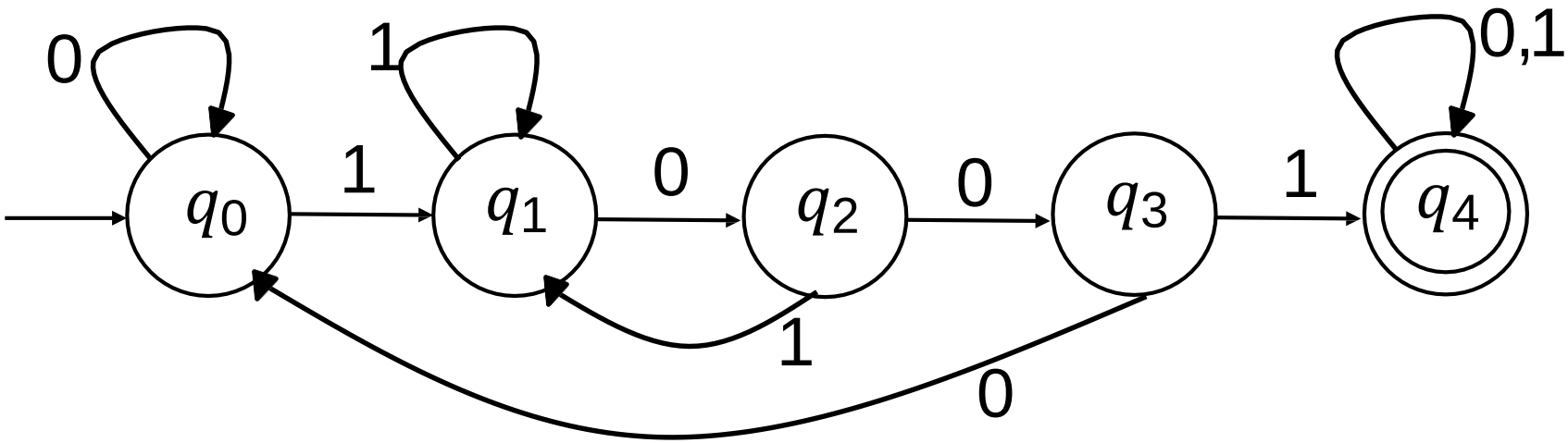
durumlar

$\delta(q_0, 1) = q_1$

geçiş tablosu

$$M = \{\{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_1\}\}$$

ör. $\Sigma = \{0,1\}$ alfabesi kullanilarak uretilen kelimerlerden, içinde '1001' altkelimesi olan kelimeleri tanıyan otomasyon.



$Q = \{q_0, q_1, q_2, q_3, q_4\}$

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

$F = \{q_4\}$

δ	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_3	q_1
q_3	q_0	q_4
q_4	q_4	q_4

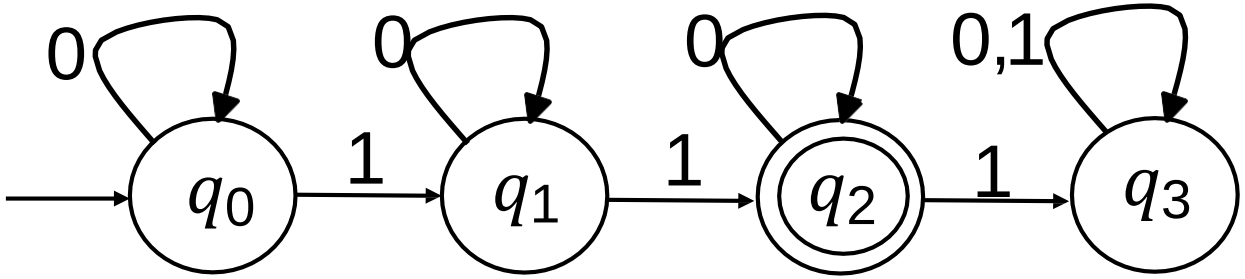
Bir kelimenin bir sonlu otomata tarafından kabul edilirliği

giriş: $M = (Q, \Sigma, \delta, q_0, F)$ makinesi, w kelimesi

çıkış: Evet yada Hayır

1. $q_{şimdiki} := q_0$
2. for i= 1: $|w|$ //w'nun her bir harfi için
3. $q_{yeni} := \delta(q_{şimdiki}, w_i)$; //şimdiki durumdan w_i ile yeni duruma geç
4. $q_{şimdiki} := q_{yeni}$ //yeni durumu şımdiki durum yap (şimdiki durumu güncelle)
5. end for
6. if $q_{şimdiki} \in F$: //eğer son durum final durumlarından biriyse
7. cout<<" Evet"; //Evet yazdır
8. else //değilse
9. cout<<"Hayır"; //Hayır yazdır
10. end if

ör. $\Sigma = \{0,1\}$ alfabesi kullanılarak üretilen kelimerlerden, içinde 2 tane 1 olan kelimeleri tanıyan sonlu otomasyon:



Yukarıdaki otomasyon w=1011 kelimesini kabul eder mi?

1. $q_{şimdiki} := q_0$
2. for'a başla:

w

1

$i = 1$ $\begin{cases} q_{yeni} := \delta(q_0, 1) = q_1 \\ q_{şimdiki} = q_1 \end{cases}$

2

$i = 2$ $\begin{cases} q_{yeni} := \delta(q_1, 0) = q_1 \\ q_{şimdiki} = q_1 \end{cases}$

3

$i = 3$ $\begin{cases} q_{yeni} := \delta(q_1, 1) = q_2 \\ q_{şimdiki} = q_2 \end{cases}$

4

$i = 4$ $\begin{cases} q_{yeni} := \delta(q_2, 1) = q_3 \\ q_{şimdiki} = q_3 \end{cases}$

3. for'u bitir.

4. $q_{şimdiki} = q_3 \notin F$ olduğundan cevap

Nondeterministik (Belirsiz) Sonlu Otomata (NSO)

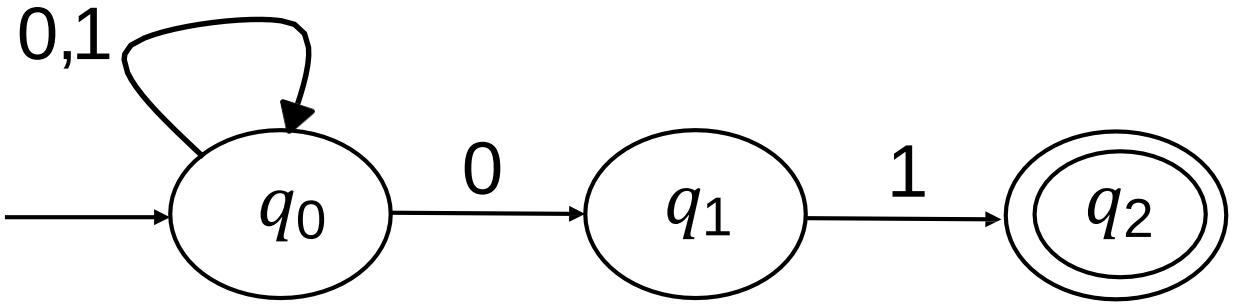
Şuana kadar gördüğümüz sonlu otomatalar "Deterministik Sonlu Otomata (DSO) " idi. (Deterministic Finite Automata (DFA)) (Deterministik rastegele olmayan, kararlı, belirli şeklinde Türkçe'ye çevriliyor.)

DSO'nun ana karakteristiği, belirli bir anda makinenin yalnızca bir durumda bulunabilir olmasıdır (iki durum ayni anda aktif olamaz).

NSO, DSO kavramının genelleştrilmesidir: DSO $\subseteq$ NSO, yani her DSO bir NSO'dur. Ayrıca her NSO, bir DSO ya dönüştürülebilir.

Bir NSO da makine ayni anda birden fazla durumda olabilir. Bu NSO'nun ana karakteristiği dir. Bu bize daha karmaşık dilleri tanıyabilen makineler üretebilmeyi sağlar.

ör. $\Sigma = \{0,1\}$ alfabesi kullanılarak üretilen kelimerlerden sonu "01" ile biten kelimerleri tanıyan sonlu otomata:



soru: bu otomatada dikkatimizi çeken şeyler neler?

1. q_0' da 2 tane 0 oku var.Yani q_0'da iken 0 gelirse hem q_0'a, hem q_1'e geçeriz.
2. q_1'de 0 oku yok.Yani q_1'de 0 gelirse makine (otomata) çıkmaza girer (ölür).
3. q_2'de ne 0 oku nede 1 oku var.Yani makine q_2'de sonlanmaz harf almaya (0,1) almaya devam ederse, makine çıkmaza girer.

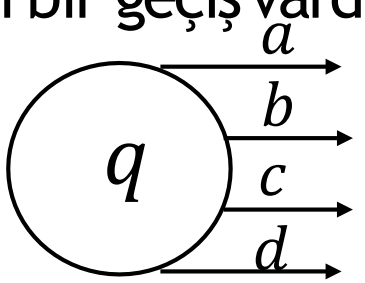
Hatırla: Daha önce gördüğümüz deterministik sonlu otomatada (DSO) geçiş fonksiyonu:

$$\delta: Q \times \Sigma \rightarrow Q$$

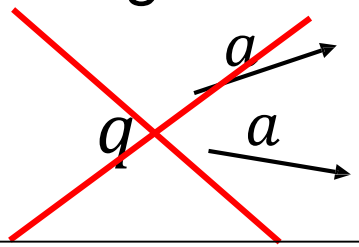
1.Burada δ bir fonksiyon olduğundan her bir (durum, harf) ikilisi için

fonksiyon tanımlıdır.Yani her bir durum ($q \in Q$) da bütün harfler ($\forall \sigma \in \Sigma$) için bir ok yani bir geçiş vardır:

Ör. $\Sigma = \{a, b, c, d\}$ için



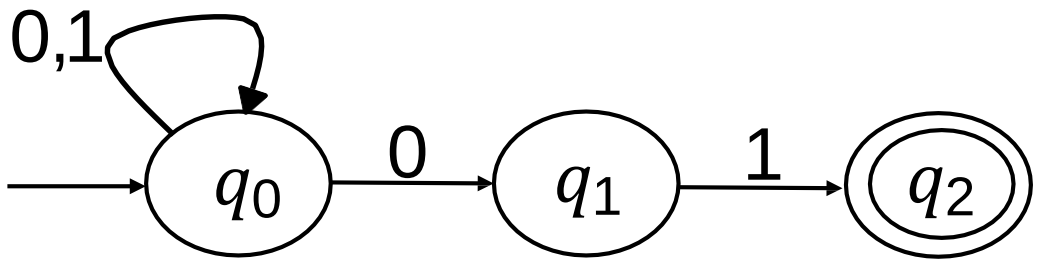
2.Yine, δ bir fonksiyon olduğundan aldığı her bir (durum, harf) ikilisini bir ve yalnız bir duruma götürür.

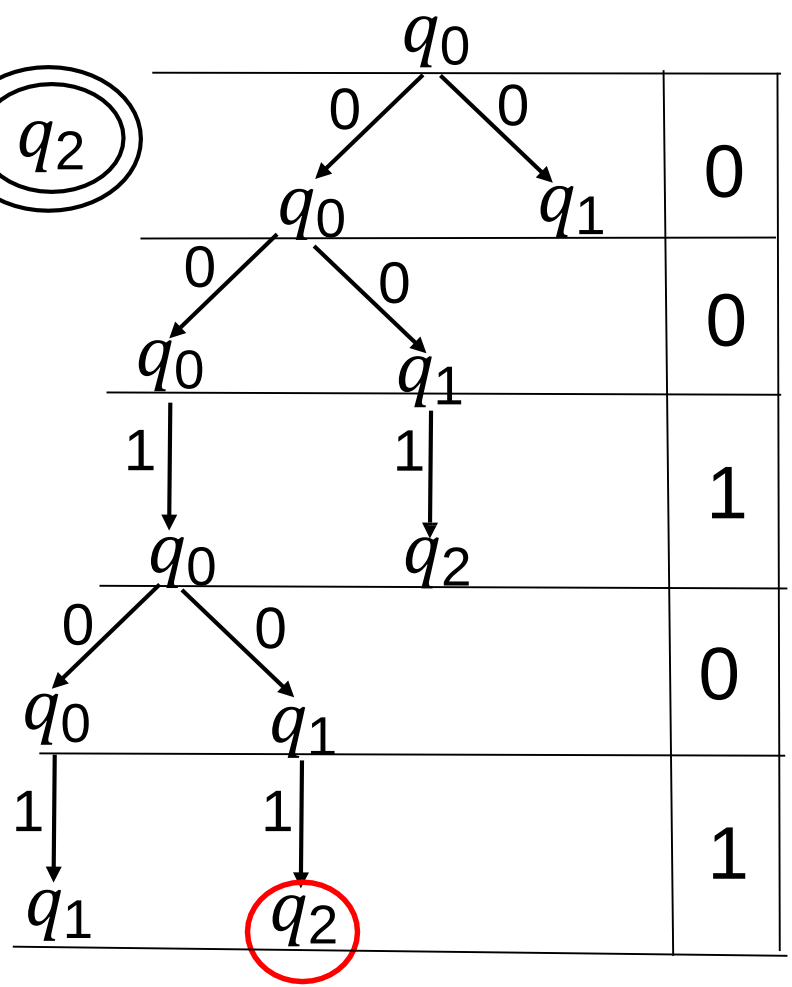


!! NSO'da bu yukarıdaki durumların ikiside geçerli değildir.

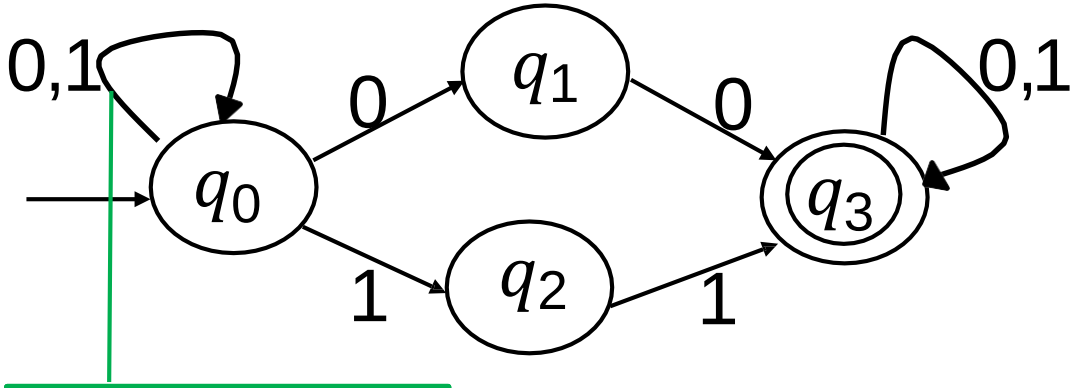
Bir NSO'nun İşleyişi

w = 00101 kelimesi aşağıdaki NSO şu şekilde işlenir:



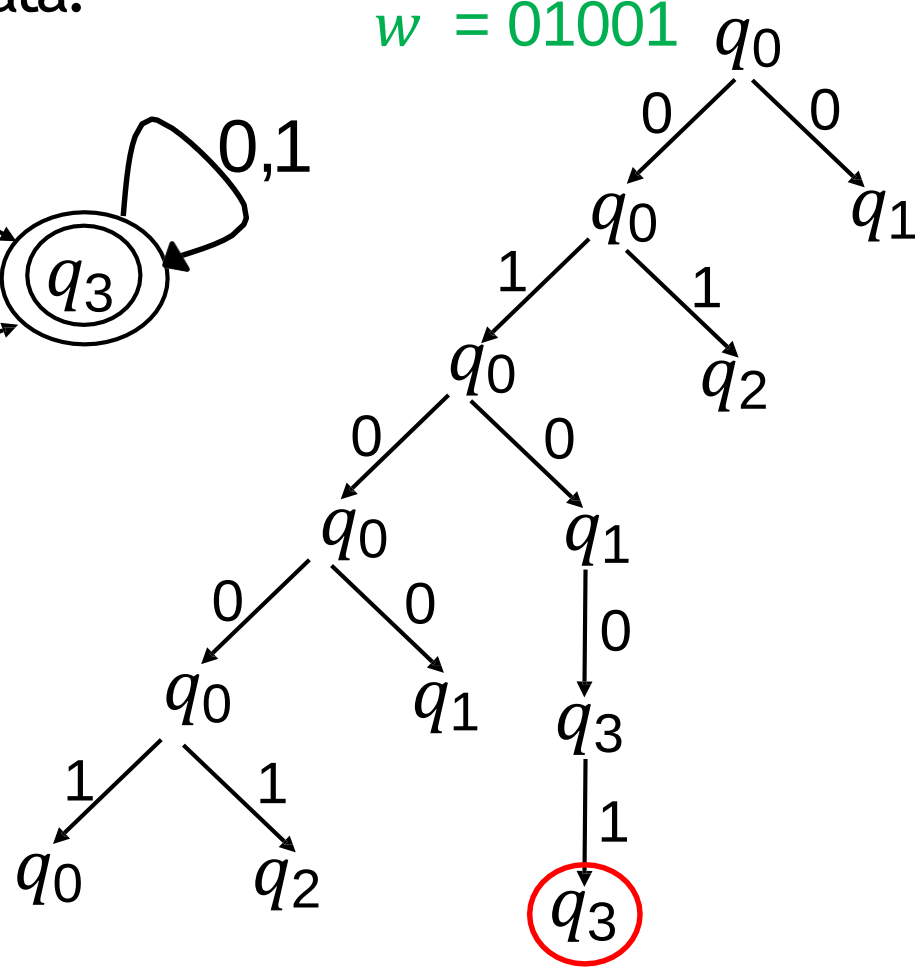


ör. $\Sigma = \{0,1\}$ alfabesi kullanılarak üretilen kelimerlerden <u>içinde</u> "00" yada "11" alt kelimerleri geçen kelimeleri tanıyan nondeterministik sonlu otomata:



Eğer q_0'ın bu oku olmazsa, makine yalnızca 00 yada 11 ile başlayan kelimeleri kabul eder!

w = 01001



Bu işlemin sonunda makine q_0, $q_2$ve q_3 durumlarında duruyor.