



# BİÇİMSEL DİLLER VE OTOMATA TEORİSİ

Biçimsel Diller ve Otomata Teorisi

2. Hafta

DR. ÖĞR. ÜYESİ. HÜSEYİN VURAL



# Ders İzlenesi

---

- Finite Automata (Sonlu Otomata)
- Sonlu Otomata Tanımı
- Sonlu Otomata Tasarımı

# Sonlu Otomata

---

- Sonlu otomatlar çok küçük hafızaya sahip modellerdir.
- Çok küçük hafızaya sahip bu modellerin kullanıldığı sistemleri günlük hayatımızda bir çok yerde kullanıyoruz.
- Örneğin otomatik açılabilen kapılar.

# Sonlu Otomata

- Otomatik kapı sisteminin kontrol mekanizması 2 farklı durumda bulunur. **Açık** veya **kapalı**.



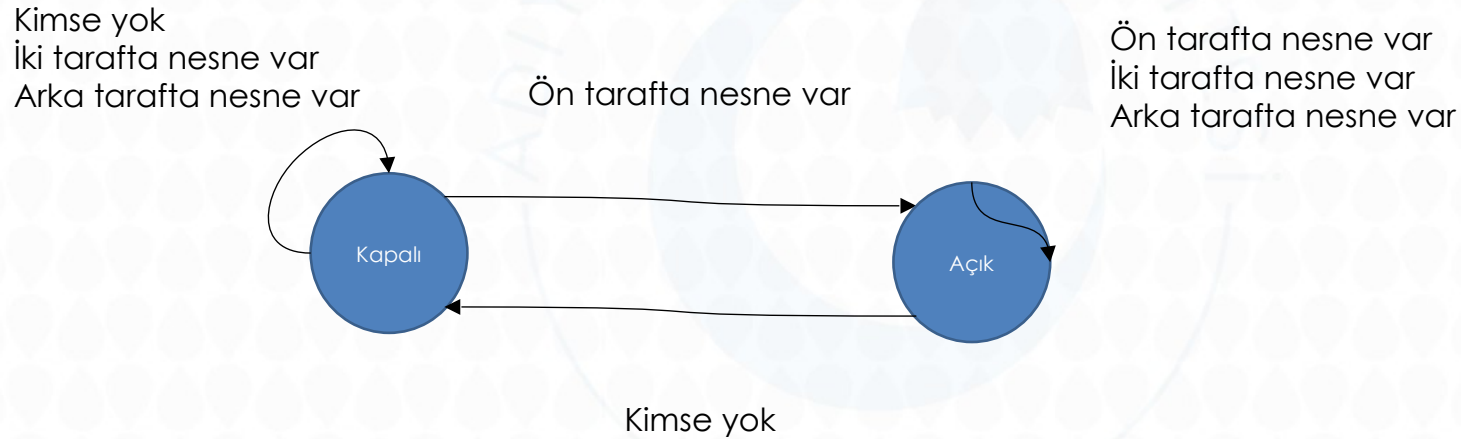
# Sonlu Otomata

- 4 farklı giriş olabilir.



# Sonlu Otomata

- Otomatik açılan kapı için durum diyagramı





# Sonlu Otomata

- Otomatik açılan kapı için durum geçiş tablosu

|        | Kimse yok | Ön tarafta nesne var | Arka tarafta nesne var | İki tarafta nesne var |
|--------|-----------|----------------------|------------------------|-----------------------|
| Kapalı |           |                      |                        |                       |
| Açık   |           |                      |                        |                       |

# Sonlu Otomata

- Otomatik açılan kapı için durum geçiş tablosu

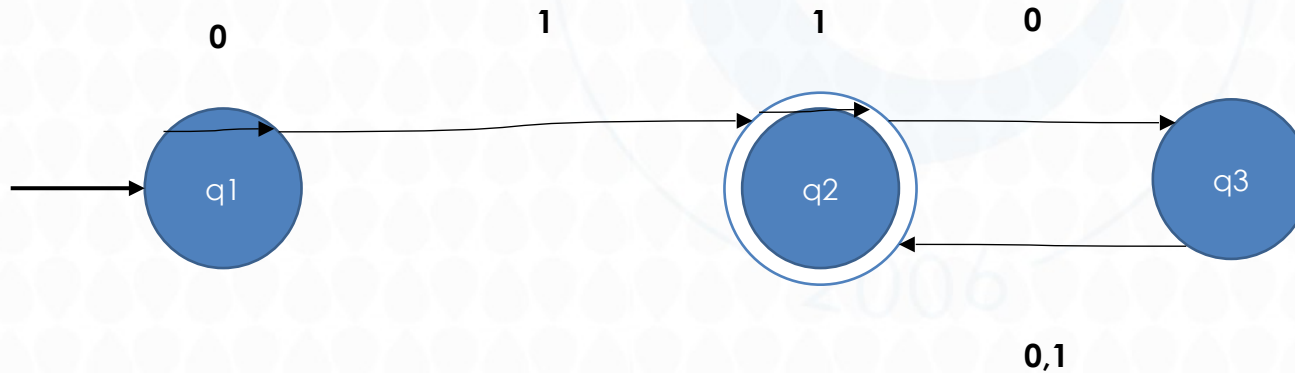
|        | Kimse yok | Ön tarafta nesne var | Arka tarafta nesne var | İki tarafta nesne var |
|--------|-----------|----------------------|------------------------|-----------------------|
| Kapalı | Kapalı    | Açık                 | Kapalı                 | Kapalı                |
| Açık   | Kapalı    | Açık                 | Açık                   | Açık                  |

- Otomatik açılan kapı için 1 bitlik bir hafıza kontrolörü yeterli olacaktır çünkü 1 bit sayesinde 2 durum da hafızada tutulabilir.
- Eğer asansör örneği veya hesap makinesi yapmak istersek 1 bit yeterli olmayacaktır.



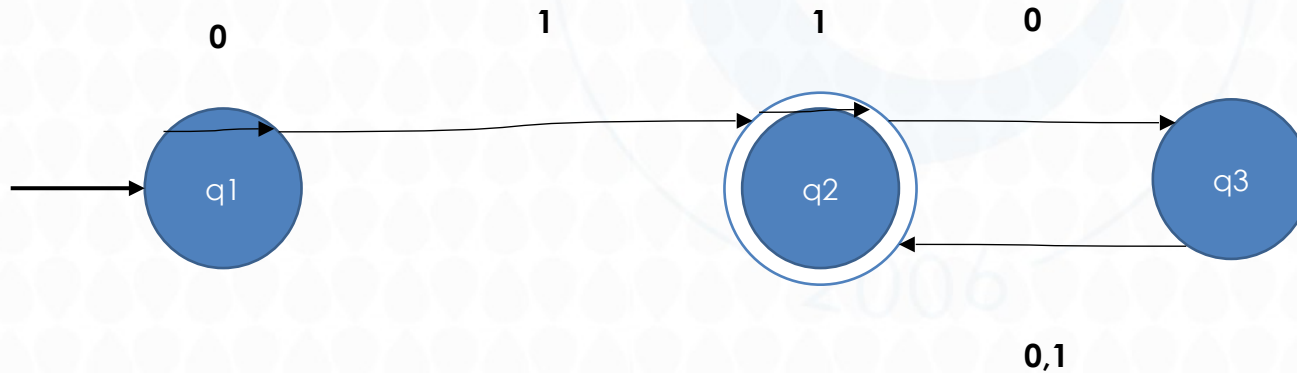
# Sonlu Otomata Tanımı

- Aşağıdaki şekil **M1** otomatasını gösteren durum diyagramıdır.
  - 3 durum içermektedir ( $q_1, q_2, q_3$ ).
  - $q_1$  **başlangıç** durumunu,  $q_2$  ise **kabul** durumunu göstermektedir.
  - Bir durumdan diğerine geçişi gösteren oklara **geçiş** denilmektedir.



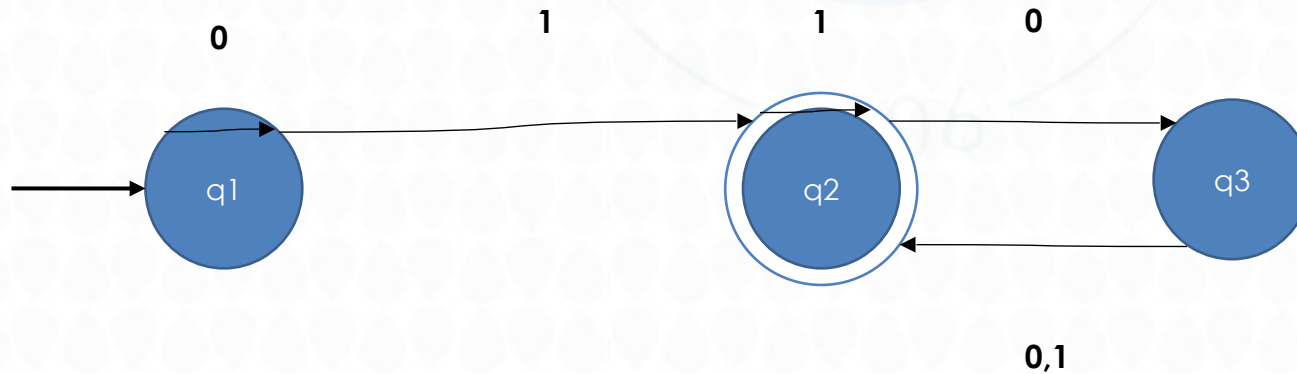
# Sonlu Otomata Tanımı

- **M1** otomatasının 1101 string'ini kabul edip etmeyeceğini inceleyelim.
- Otomat, string'i soldan başlayarak sırasıyla birer birer okur ve en son değeri okuduğunda otomat **çıkışı** üretilir.



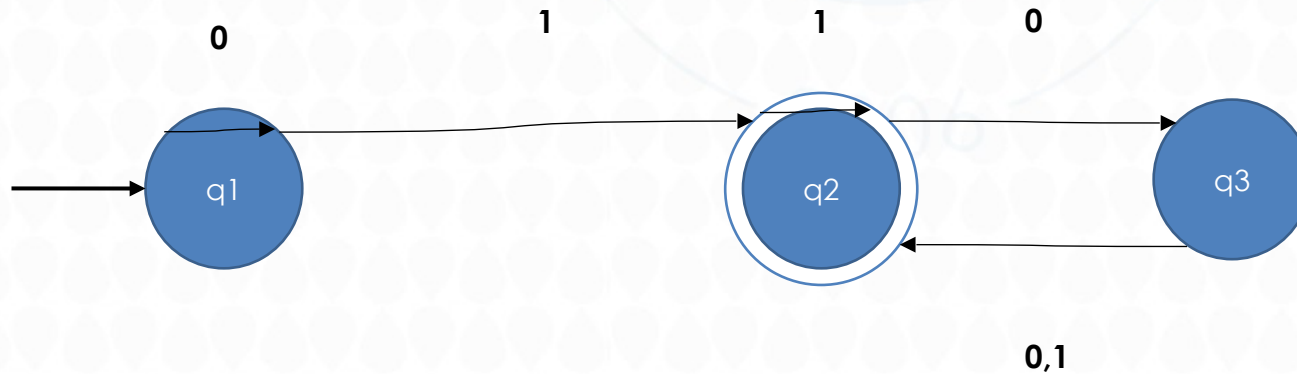
# Sonlu Otomata Tanımı

- 1.) 1101 string'i için  $q_1$  durumundan başlanacaktır.
- 2.) İlk olarak 1 değeri okunacak ve  $q_2$  durumuna geçiş yapılacaktır.  $q_1 \rightarrow q_2$
- 3.) Sonraki 1 değeri okunacak ve  $q_2$  durumunda kalmaya devam edecektir.  $q_2 \rightarrow q_2$



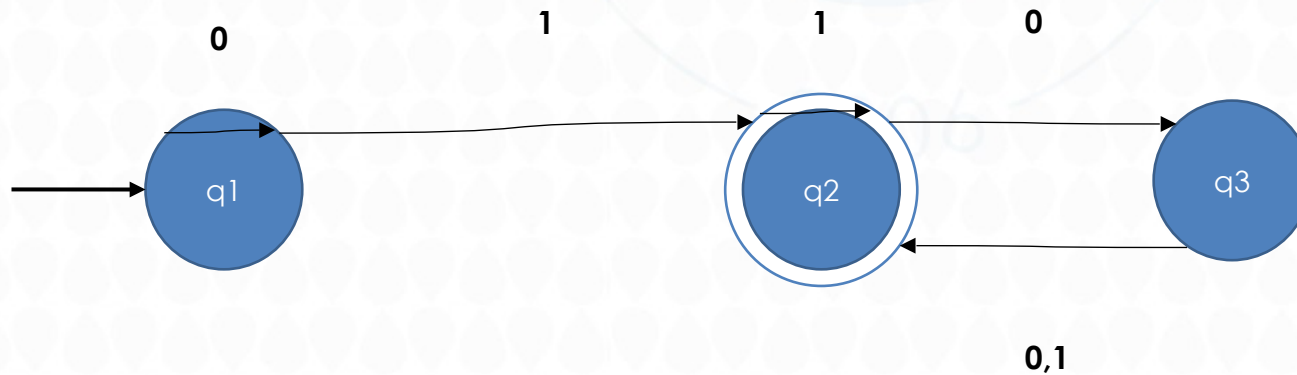
# Sonlu Otomata Tanımı

- 1.) Sonraki 0 değeri okunacak ve  $q3$  durumuna geçiş yapılacaktır.  $q2 \rightarrow q3$
- 2.) Son kalan 1 değeri okunacak ve  $q2$  durumuna geçiş yapılacaktır.  $q3 \rightarrow q2$
- 3.) String kabul edilecektir çünkü  $M1$  otomati son olarak  $q2$  durumundadır.



# Sonlu Otomata Tanımı

- M1 otomati aşağıdaki stringleri kabul eder mi?
  - 1,01,11,0101010101
  - 100,0100,1100000,0101000000





# Sonlu Otomata Tanımı

- Sonlu Otomata'yı matematiksel notasyonla tanımlamak için farklı özellikler gerekmektedir.
  - **Durum kümesi** ve bir durumdan diğerine geçişi gösteren giriş değerleri
  - Giriş değerlerini elde etmek için **alfabe**
  - **Geçiş kuralları**
  - **Başlangıç** durumu
  - **Kabul** durum(ları)



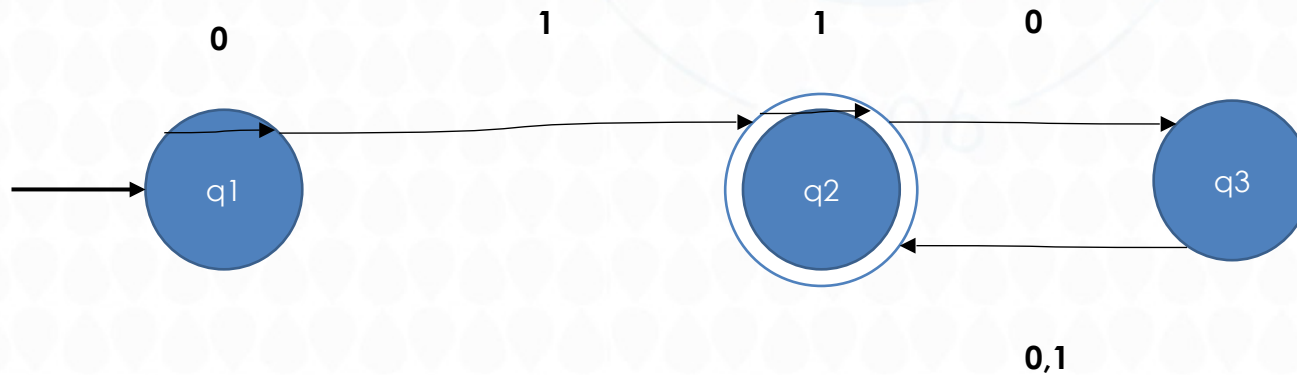
# Sonlu Otomata Tanımı

---

- $M = (Q, \Sigma, \delta, q_0, F)$
- 1.)  $Q$ , sonlu durumlar kümesidir. Otomatın bulunabileceği durumları gösterir.
- 2.)  $\Sigma$ , alfabeyi gösterir.
- 3.)  $\delta : Q \times \Sigma \rightarrow Q$ .  $\delta$ (delta) geçiş fonksiyonudur
- 4.)  $q_0$  başlangıç durumunu gösterir
- 5.)  $F$ , kabul durumlarının kümesini gösterir.

# Sonlu Otomata Tanımı

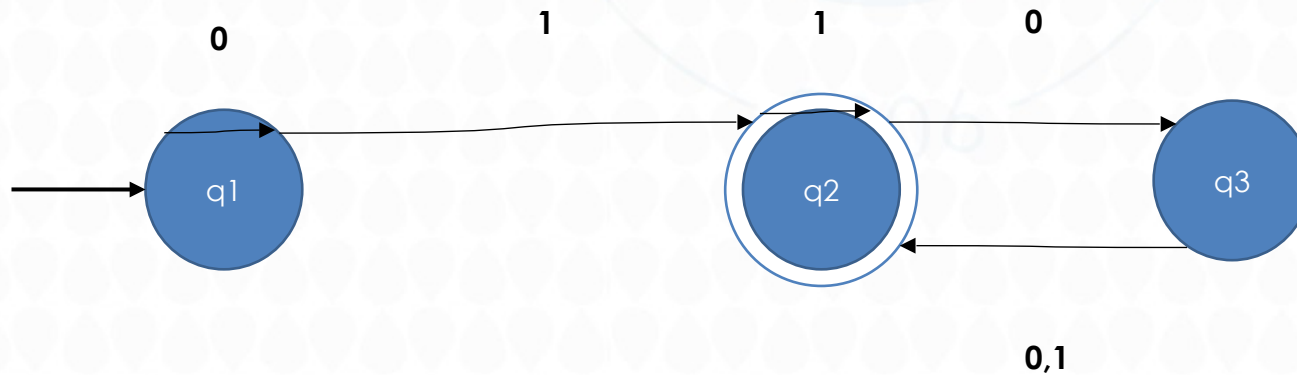
- $M1 = (Q, \Sigma, \delta, q_0, F)$
- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{0,1\}$
- $q_1 =$  başlangıç durumu
- $F = \{q_2\}$



# Sonlu Otomata Tanımı

- $M1 = (Q, \Sigma, \delta, q_0, F)$
- $\delta$  geçiş fonksiyonu şu şekildedir

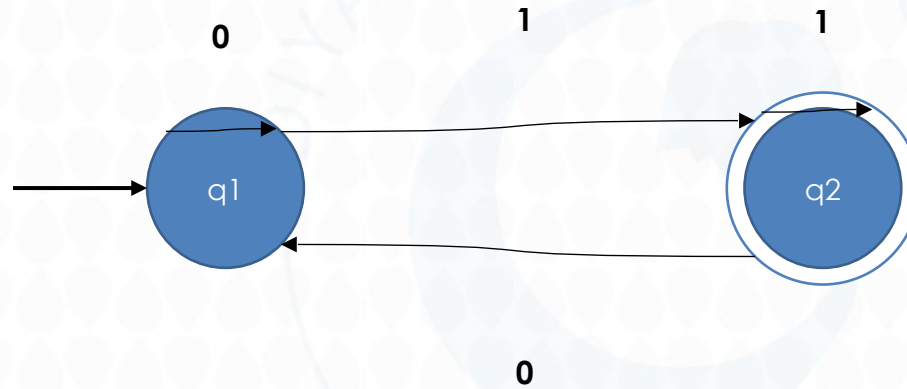
|       | 0     | 1     |
|-------|-------|-------|
| $q_1$ | $q_1$ | $q_2$ |
| $q_2$ | $q_3$ | $q_2$ |
| $q_3$ | $q_2$ | $q_2$ |



# Sonlu Otomata

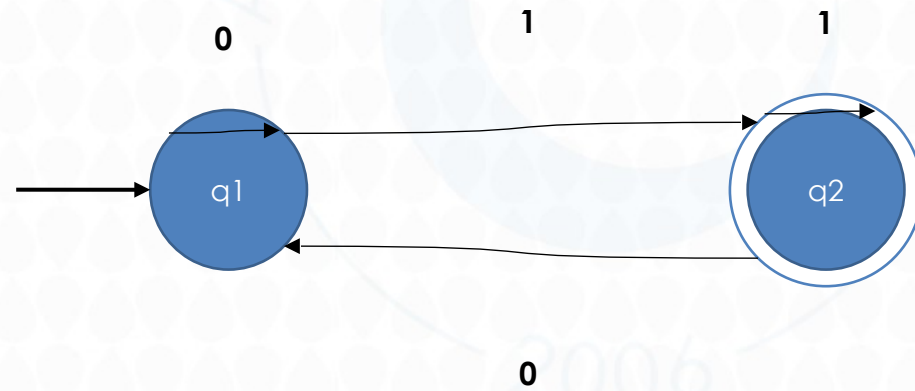
- Eğer  $M$  otomatının kabul ettiği tüm stringlerin kümesine  $A$  dersek o zaman  $M$  otomatının diline  $A$  denilir.
- $L(M) = A$
- $A = \{w \mid w \text{ en az bir tane } 1 \text{ içermelidir veya son } 1 \text{'den sonra gelen sıfırların sayısı çift olmalıdır.}$
- Eğer bir otomatın kabul ettiği her hangi bir string yoksa o zaman bu otomatın dili boş kümedir.

# Sonlu Otomata Örnekleri



# Sonlu Otomata Örnekleri

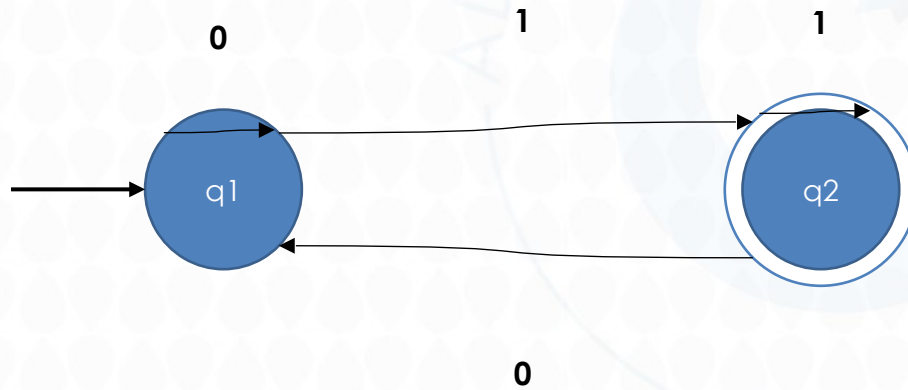
|       | 0     | 1     |
|-------|-------|-------|
| $q_1$ | $q_1$ | $q_2$ |
| $q_2$ | $q_1$ | $q_2$ |





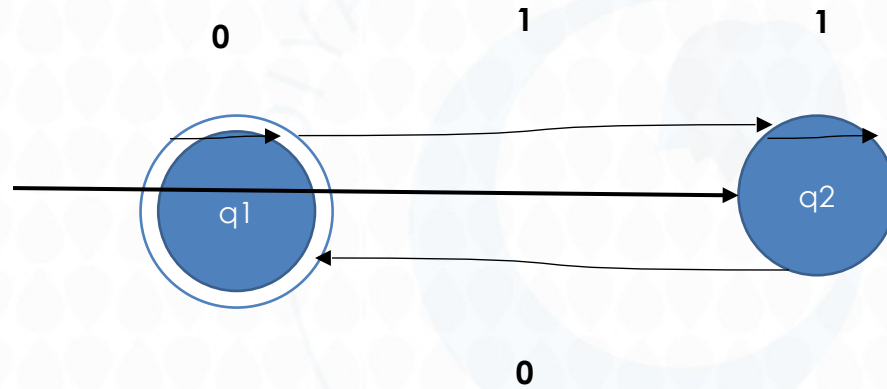
# Sonlu Otomata Örnekleri

|       | 0     | 1     |
|-------|-------|-------|
| $q_1$ | $q_1$ | $q_2$ |
| $q_2$ | $q_1$ | $q_2$ |



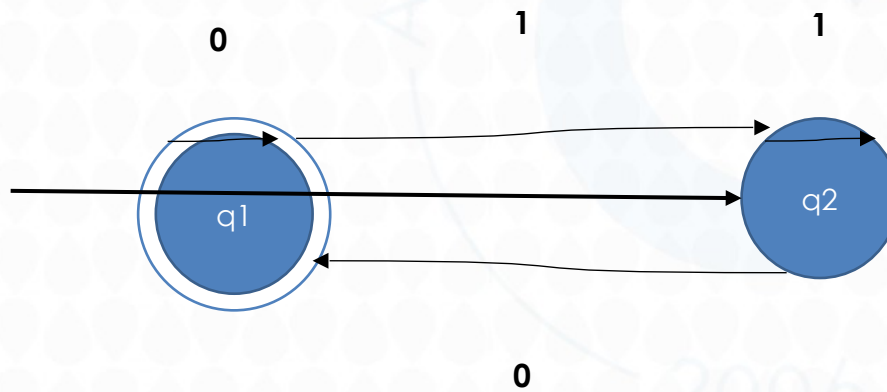
$$L(M) = \{w \mid w, 1 \text{ ile biten kelimeler}\}$$

# Sonlu Otomata Örnekleri



# Sonlu Otomata Örnekleri

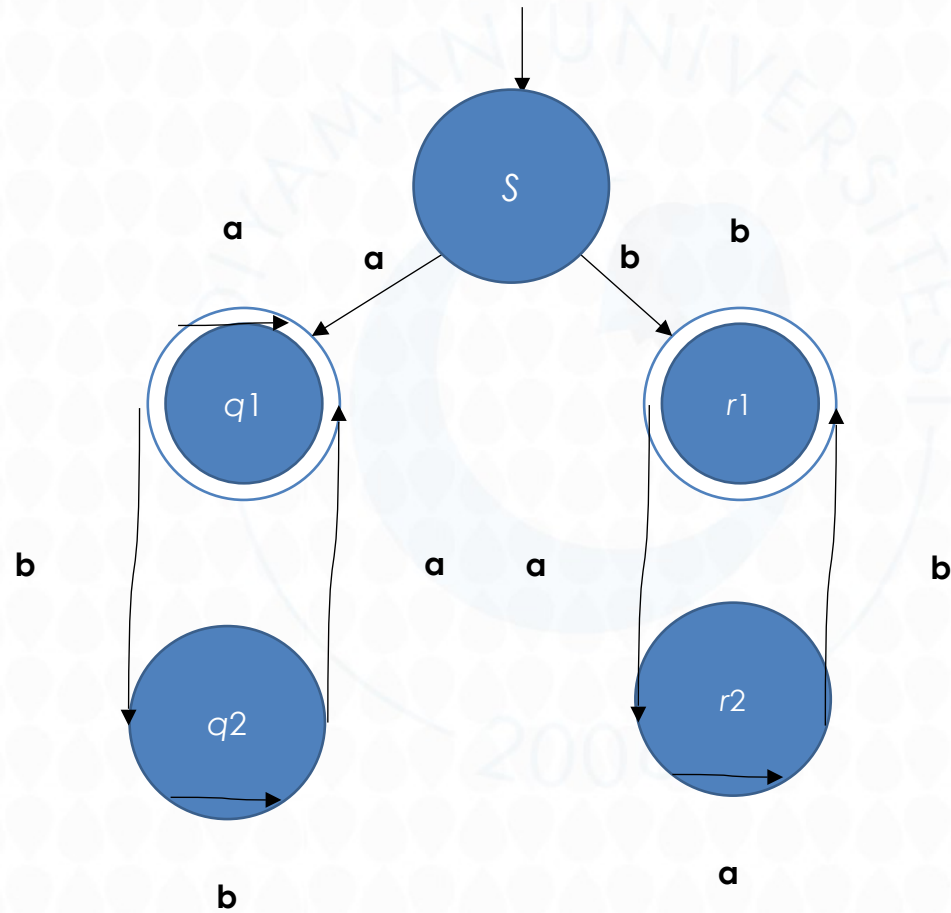
|       | 0     | 1     |
|-------|-------|-------|
| $q_1$ | $q_1$ | $q_2$ |
| $q_2$ | $q_1$ | $q_2$ |



$$L(M) = \{w \mid w, \text{ boş string veya } 0 \text{ ile biten kelimeler}\}$$

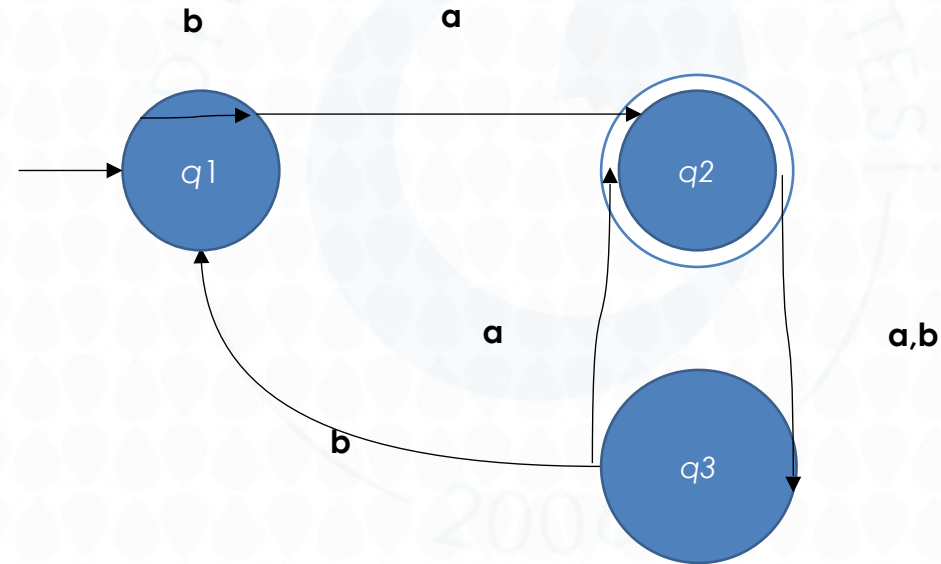
# Sonlu Otomata Örnekleri

- M2 otomati



# Sonlu Otomata Örnekleri

- M3 otomati





# Sonlu Otomat Örnekleri

---

\*1 ile başlayan ve 0 ile biten kelimeleri kabul eden otomat

\*En az bir tane 1 içerecek otomat

\*0101 substring'ini içerecek otomat,  $x0101y$

\*En az 3 uzunluğuna sahip kelimeleri içeren 3. sembolü 0 olacak otomat





## DERS SONU

Finite Automata ve Deterministic Finite Automata

Dr. Öğr. Üyesi Hüseyin VURAL