

PAMUKKALE ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
2021 BAHAR

# Biçimsel Diller ve Otomata Teorisi

## Formal languages and automata theory

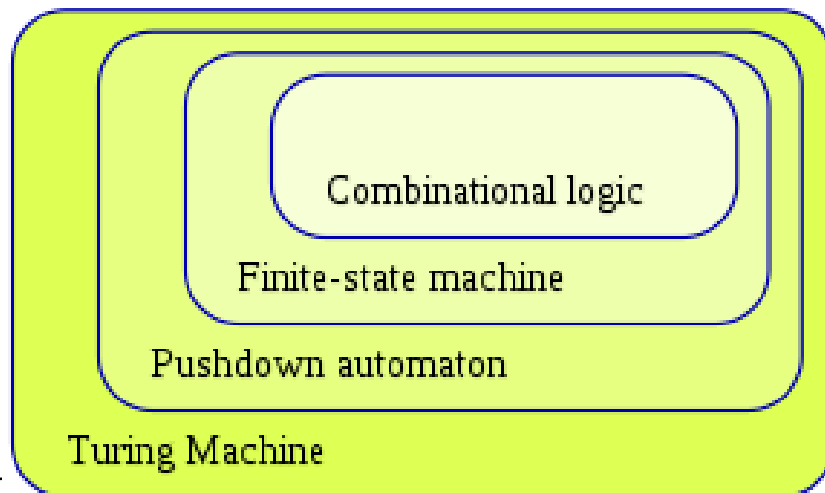
Deterministik Sonlu Otomat

---

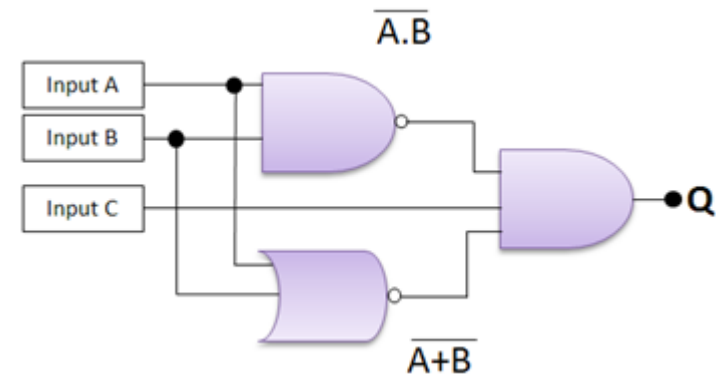
# Deterministik Sonlu Otomatlar- Deterministic Finite Automata

- **Deterministic:** Deterministik: (özgür irade eksikliği anlamına gelen determinizmden) rastgele bir olayın tam tersidir. Bize gelecekteki bazı olayların rastgelelik dahil olmadan tam olarak hesaplanabileceğini söyler.
- **Otomata teorisi (Otomatlar Teorisi)**, soyut makineler ve otomatların yanı sıra bunlar kullanılarak çözülebilen hesaplama problemlerinin incelenmesidir.

Automata theory



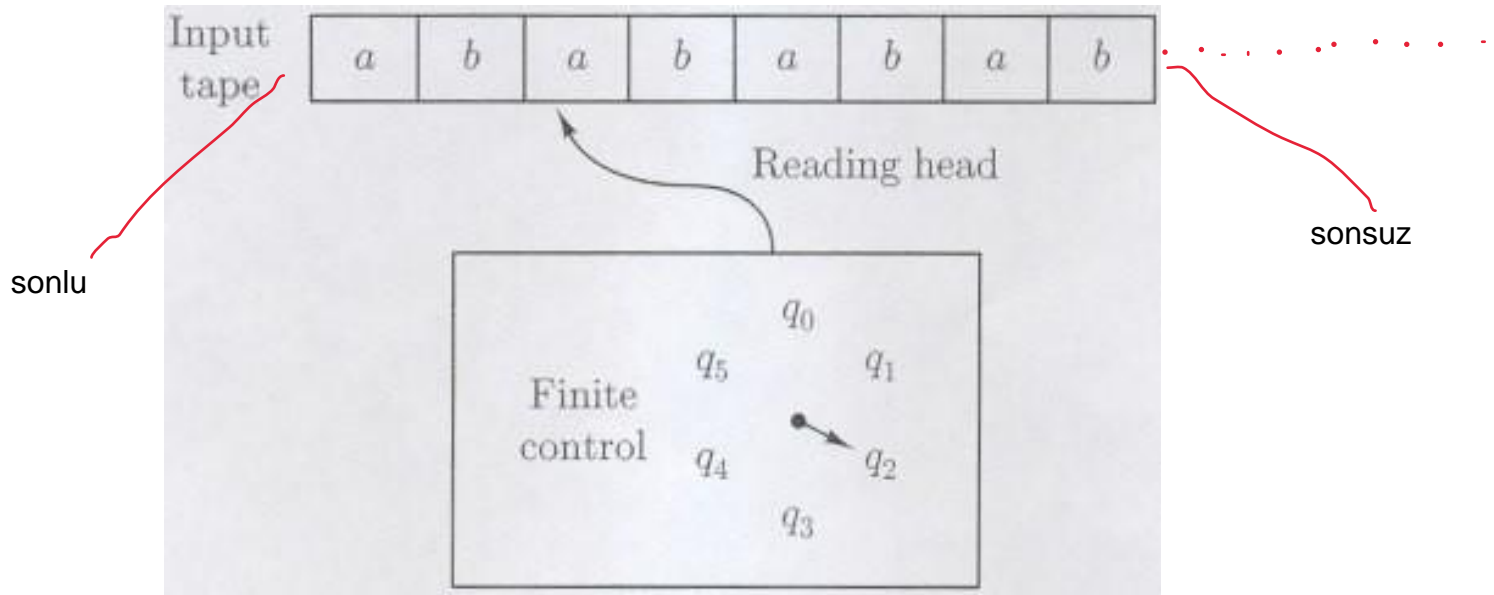
Combinational Logic



# Deterministic Finite Automata

- Bir sonlu otomat (**finite automaton**) (**otomatlar-automata: çoğul**) sabit ve sonlu kapasiteye sahip bir merkezi işlem birimine sahiptir.
- Bilgisayarın oldukça sınırlı ve basit bir modelidir.
- Giriş bilgisini katar (string) olarak bir giriş şeridi (**input tape**) üzerinden alır.
- Bir **çıkış üretmez**, giriş bilgisinin **kabul edilip edilmediğini** gösterir.
- **Language recognition device (dil tanıyıcı cihaz)** olarak işlem yapar.
- Compiler'da **lexical analysis** (begin, +, for gibi program birimleri belirlenir) aşamasında kullanılır.
- Protokol tanımlamalarında kullanılır.

# Deterministic Finite Automata



- Giriş bilgisi, input tape üzerinden string olarak elde edilir.
- Makinenin ana kısmı "black box" tır ve sonlu sayıda farklı duruma sahiptir.
- Bu black box "finite control" olarak adlandırılır ve hareketli **reading head**(okuma kafası) ile input tape üzerinde herhangi bir pozisyonda bulunan sembolü algılar.
- Başlangıçta okuma kafası en soldaki kare üzerinde bulunur ve finite control **initial state**(başlangıç durumu) durumundadır.

# Deterministic Finite Automata

- Automaton her seferinde input tape'ten bir sembol okur ve yeni bir duruma geçer.
- Yeni duruma geçme sadece ve sadece
  - **mevcut sonlu sayıda duruma ve okunan sembole**

bağlıdır !!!
- Bu yüzden **deterministic finite automaton (DFA)**'dur.
- Her okumadan sonra okuma kafası bir sağa geçer ve string sonuna kadar okuma devam eder.
- Eğer string bittiğinde DFA **final state(s) (sonuc durumları)**'lerden birisinde ise bu string kabul edilir.
- Bu DFA tarafından kabul edilen dil, kabul edilen stringler kümesidir.

# Deterministic Finite Automata

## Tanım:

Bir DFA beşli demet (quintuple) olarak tanımlanır.  $M = (K, \Sigma, \delta, s, F)$

$K$  sonlu sayıda durumlar kümesi  $\{q_0, q_1, q_2, \dots\}$

$\Sigma$  alfabe  $\{a, b\}$

$\delta$  transition function (geçiş fonksiyonu)  $K \times \Sigma \rightarrow K'$ ,

$s \in K$  başlangıç durumu (sadece bir tane)

$F \subseteq K$  final state(s) kümesi

- $M$  otomatının sonraki duruma geçişi, geçiş fonksiyonu ( transition function) ile belirlenir.

- Eger  $M$  otomatı  $q \in K$  durumunda iken input tape'ten

$a \in \Sigma$  okumuşsa,  $\delta(q, a) \in K$  durumuna geçer ve bu durum unique (tek)'tir (function)

# Deterministic Finite Automata

Örnek:

$M$  bir DFA ve  $M = (K, \Sigma, \delta, s, F)$  şeklinde tanımlanmış olsun:

$$\begin{aligned} K &= \{q_0, q_1\}, \\ \Sigma &= \{a, b\}, \\ s &= q_0 \\ F &= \{q_0\} \end{aligned}$$

$q$	$a$	$\delta(q, a)$
$q_0$	$a$	$q_0$
$q_0$	$b$	$q_1$
$q_1$	$a$	$q_1$
$q_1$	$b$	$q_0$

$L(M)$  dilini sözel olarak tanımlayalım

# Deterministic Finite Automata

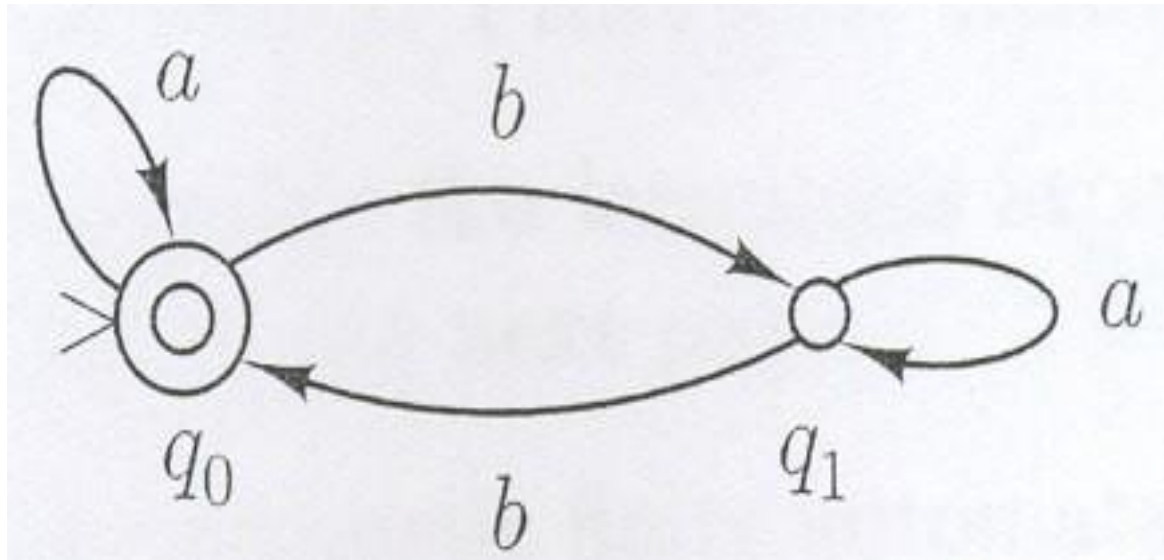
Örnek:

$M$  bir DFA ve  $M = (K, \Sigma, \delta, s, F)$  şeklinde tanımlanmış olsun:

$$\begin{aligned} K &= \{q_0, q_1\}, \\ \Sigma &= \{a, b\}, \\ s &= q_0 \\ F &= \{q_0\} \end{aligned}$$

$q$	$a$	$\delta(q, a)$
$q_0$	$a$	$q_0$
$q_0$	$b$	$q_1$
$q_1$	$a$	$q_1$
$q_1$	$b$	$q_0$

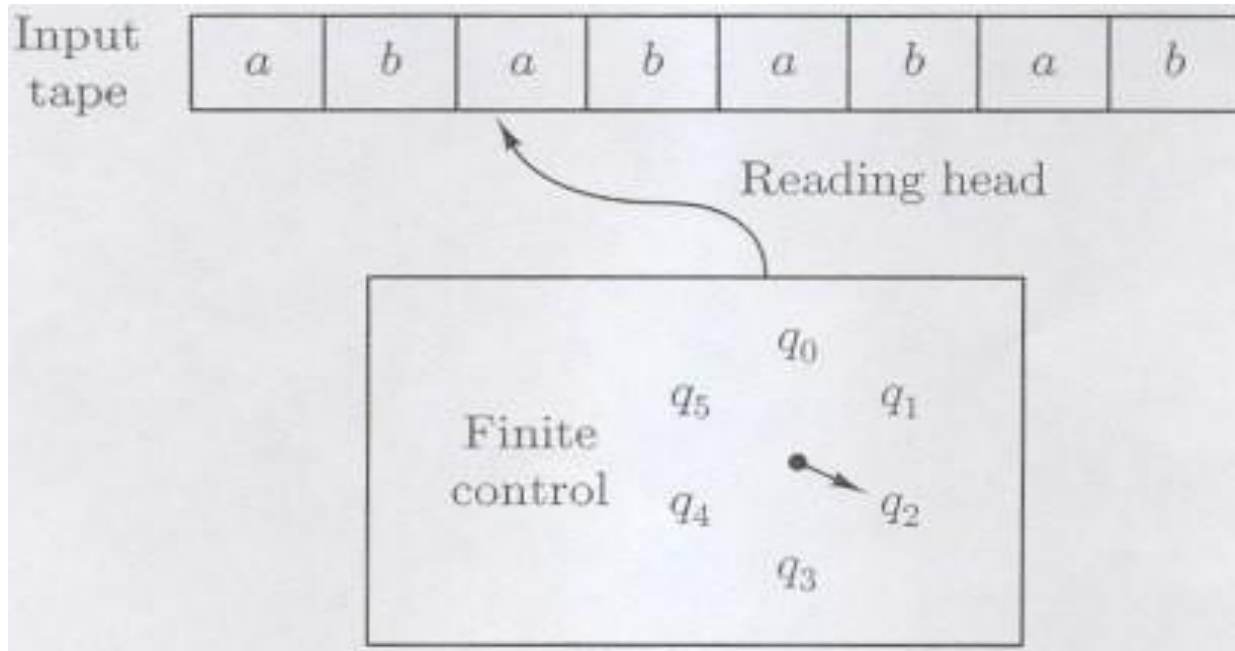
$L(M)$  içerisinde çift sayıda  $b$  bulunduran tüm stringlerin kümesidir.





# Deterministic Finite Automata

- **Configuration**, otomatın herhangi bir andaki durumu ile input tape'te sağ kısımdaki henüz okunmamış olan string'i ifade eder.
- Configuration  $K \times \Sigma^*$  'ın bir elemanıdır. Aşağıdaki otomat için konfigürasyon  $(q_2, ababab)$  'dır.



- $\vdash_M$  ardarda iki konfigürasyon arasındaki binary relation'ı ifade eder.

# Deterministic Finite Automata

- $(q, w)$  ve  $(q', w')$  ardarda iki konfigürasyon ise  $(q, w) \vdash_M (q', w')$  şeklinde belirtilir.  
*burada  $w = aw'$ ,  $a \in \Sigma$  ve  $(q, a) = q'$  olmak zorundadır.*
- $\vdash_M$  fonksiyonu  $K \times \Sigma^+$  'dan  $K \times \Sigma^*$  'ya bir fonksiyondur.
- $(q, e)$  konfigürasyonu giriş string'inin sonunu gösterir ve otomat işlemini bitirir.
- $\vdash_M$  fonksiyonunun reflexive, transitive closure'u  $\vdash_M^*$  şeklinde tanımlanır.
- Bir string  $w \in \Sigma^*$  kabul edilir, eger sadece ve sadece  $(s, w) \vdash_M^* (q, e)$  ve  $q \in F$  ise
- Sonuç olarak bir  $M$  otomatı tarafından tanınan dil  $L(M)$  olarak gösterilir ve tüm kabul edilen string'ler kümesidir.

# Deterministic Finite Automata

Örnek:

$M$  bir DFA ve  $M = (K, \Sigma, \delta, s, F)$  şeklinde tanımlanmış olsun.

$$K = \{q_0, q_1\},$$

$$\Sigma = \{a, b\},$$

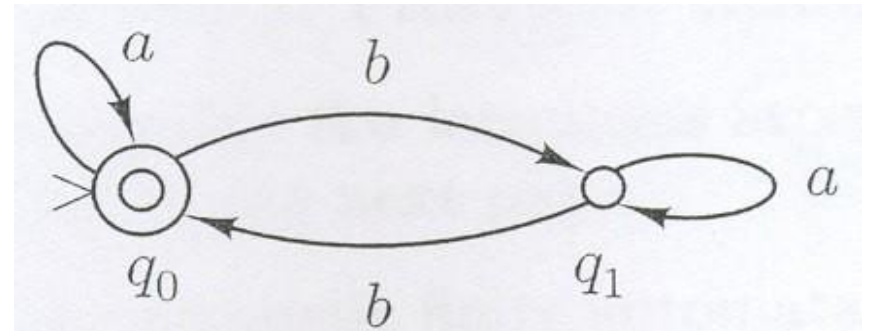
$$s = q_0$$

$$F = \{q_0\}$$

$q$	$a$	$\delta(q, a)$
$q_0$	$a$	$q_0$
$q_0$	$b$	$q_1$
$q_1$	$a$	$q_1$
$q_1$	$b$	$q_0$

$L(M)$  içerisinde çift sayıda  $b$  bulunduran tüm stringlerin kümesidir.

$$(q_0, aabba) \vdash_M (q_0, abba)$$



# Deterministic Finite Automata

Örnek:

$M$  bir DFA ve  $M = (K, \Sigma, \delta, s, F)$  şeklinde tanımlanmış olsun.

$$K = \{q_0, q_1\},$$

$$\Sigma = \{a, b\},$$

$$s = q_0$$

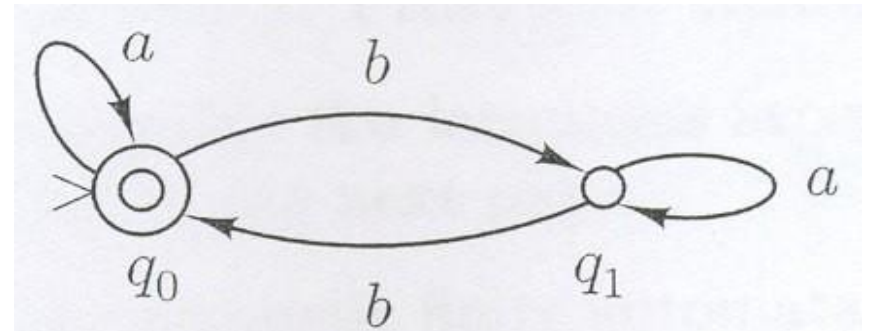
$$F = \{q_0\}$$

$q$	$a$	$\delta(q, a)$
$q_0$	$a$	$q_0$
$q_0$	$b$	$q_1$
$q_1$	$a$	$q_1$
$q_1$	$b$	$q_0$

$L(M)$  içerisinde çift sayıda  $b$  bulunduran tüm stringlerin kümesidir.

$$(q_0, aabba) \vdash_M (q_0, abba)$$

$$\vdash_M (q_0, bba)$$



# Deterministic Finite Automata

Örnek:

$M$  bir DFA ve  $M = (K, \Sigma, \delta, s, F)$  şeklinde tanımlanmış olsun.

$$K = \{q_0, q_1\},$$

$$\Sigma = \{a, b\},$$

$$s = q_0$$

$$F = \{q_0\}$$

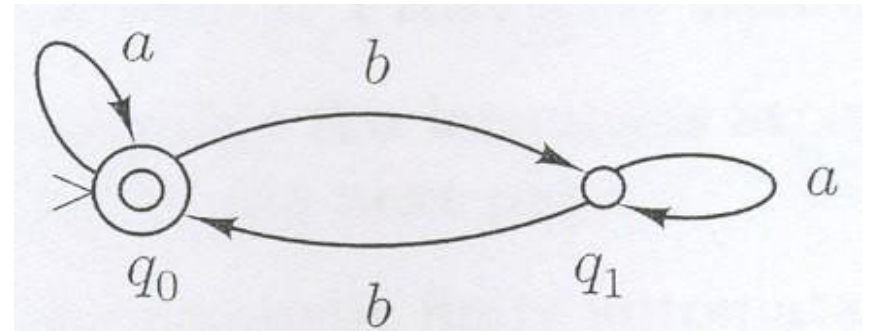
$q$	$a$	$\delta(q, a)$
$q_0$	$a$	$q_0$
$q_0$	$b$	$q_1$
$q_1$	$a$	$q_1$
$q_1$	$b$	$q_0$

$L(M)$  içerisinde çift sayıda  $b$  bulunduran tüm stringlerin kümesidir.

$$(q_0, aabba) \vdash_M (q_0, abba)$$

$$\vdash_M (q_0, bba)$$

$$\vdash_M (q_1, ba)$$



# Deterministic Finite Automata

Örnek:

$M$  bir DFA ve  $M = (K, \Sigma, \delta, s, F)$  şeklinde tanımlanmış olsun.

$$K = \{q_0, q_1\},$$

$$\Sigma = \{a, b\},$$

$$s = q_0$$

$$F = \{q_0\}$$

$q$	$a$	$\delta(q, a)$
$q_0$	$a$	$q_0$
$q_0$	$b$	$q_1$
$q_1$	$a$	$q_1$
$q_1$	$b$	$q_0$

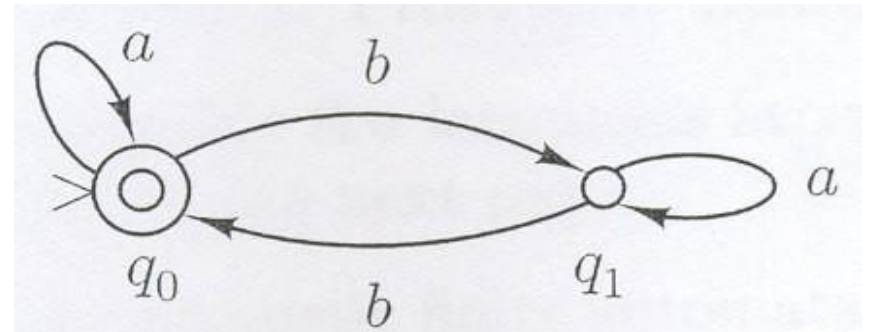
$L(M)$  içerisinde çift sayıda  $b$  bulunduran tüm stringlerin kümesidir.

$$(q_0, aabba) \vdash_M (q_0, abba)$$

$$\vdash_M (q_0, bba)$$

$$\vdash_M (q_1, ba)$$

$$\vdash_M (q_0, a)$$



# Deterministic Finite Automata

Örnek:

$M$  bir DFA ve  $M = (K, \Sigma, \delta, s, F)$  şeklinde tanımlanmış olsun.

$$K = \{q_0, q_1\},$$

$$\Sigma = \{a, b\},$$

$$s = q_0$$

$$F = \{q_0\}$$

$q$	$a$	$\delta(q, a)$
$q_0$	$a$	$q_0$
$q_0$	$b$	$q_1$
$q_1$	$a$	$q_1$
$q_1$	$b$	$q_0$

$L(M)$  içerisinde çift sayıda  $b$  bulunduran tüm stringlerin kümesidir.

$$(q_0, aabba) \vdash_M (q_0, abba)$$

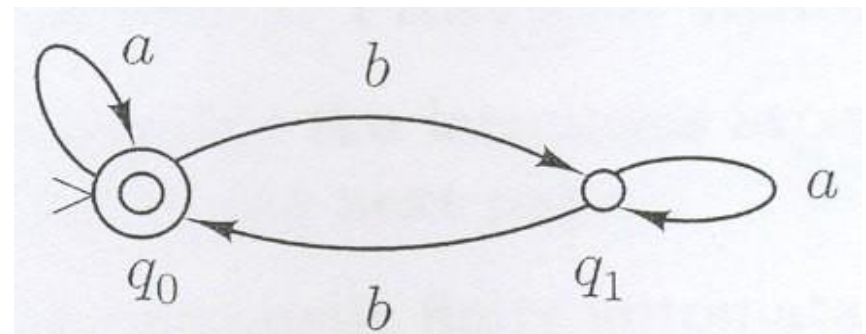
$$\vdash_M (q_0, bba)$$

$$\vdash_M (q_1, ba)$$

$$\vdash_M (q_0, a)$$

$$\vdash_M (q_0, e)$$

$(q_0, aabba) \vdash_M^* (q_0, e)$  olduğu için  $aabba$  stringi  $M$  makinesi tarafından kabul edilir.



# Deterministic Finite Automata

Örnek:

$M$  bir DFA ve  $M = (K, \Sigma, \delta, s, F)$  şeklinde tanımlanmış olsun.

$M$  otomati  $L(M) = \{w \in \{a, b\}^* : w \text{ ardarda üç tane } b \text{ bulundurmaz}\}$  dilini tanır.

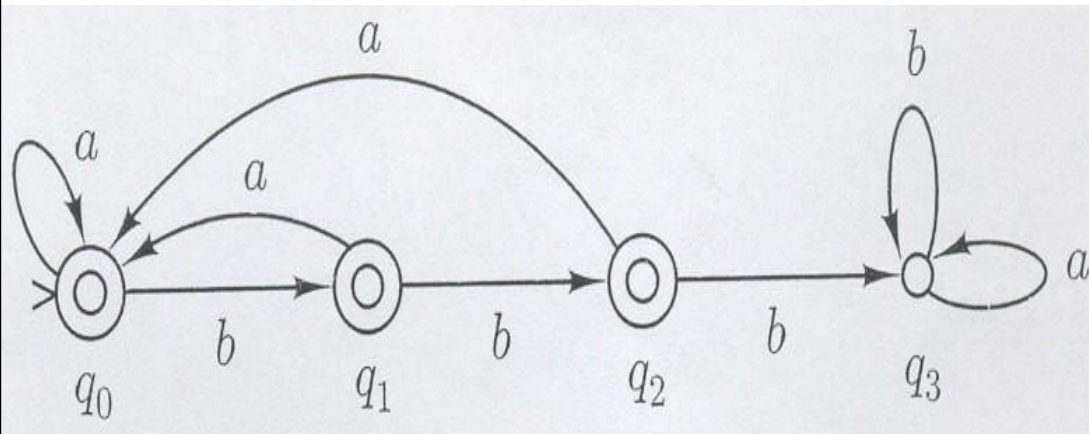


$$K = \{q_0, q_1, q_2, q_3\},$$

$$\Sigma = \{a, b\},$$

$$s = q_0$$

$$F = \{q_0, q_1, q_2\}$$



$q$	$a$	$\delta(q, a)$
$q_0$	$a$	$q_0$
$q_0$	$b$	$q_1$
$q_1$	$a$	$q_0$
$q_1$	$b$	$q_2$
$q_2$	$a$	$q_0$
$q_2$	$b$	$q_3$
$q_3$	$a$	$q_3$
$q_3$	$b$	$q_3$

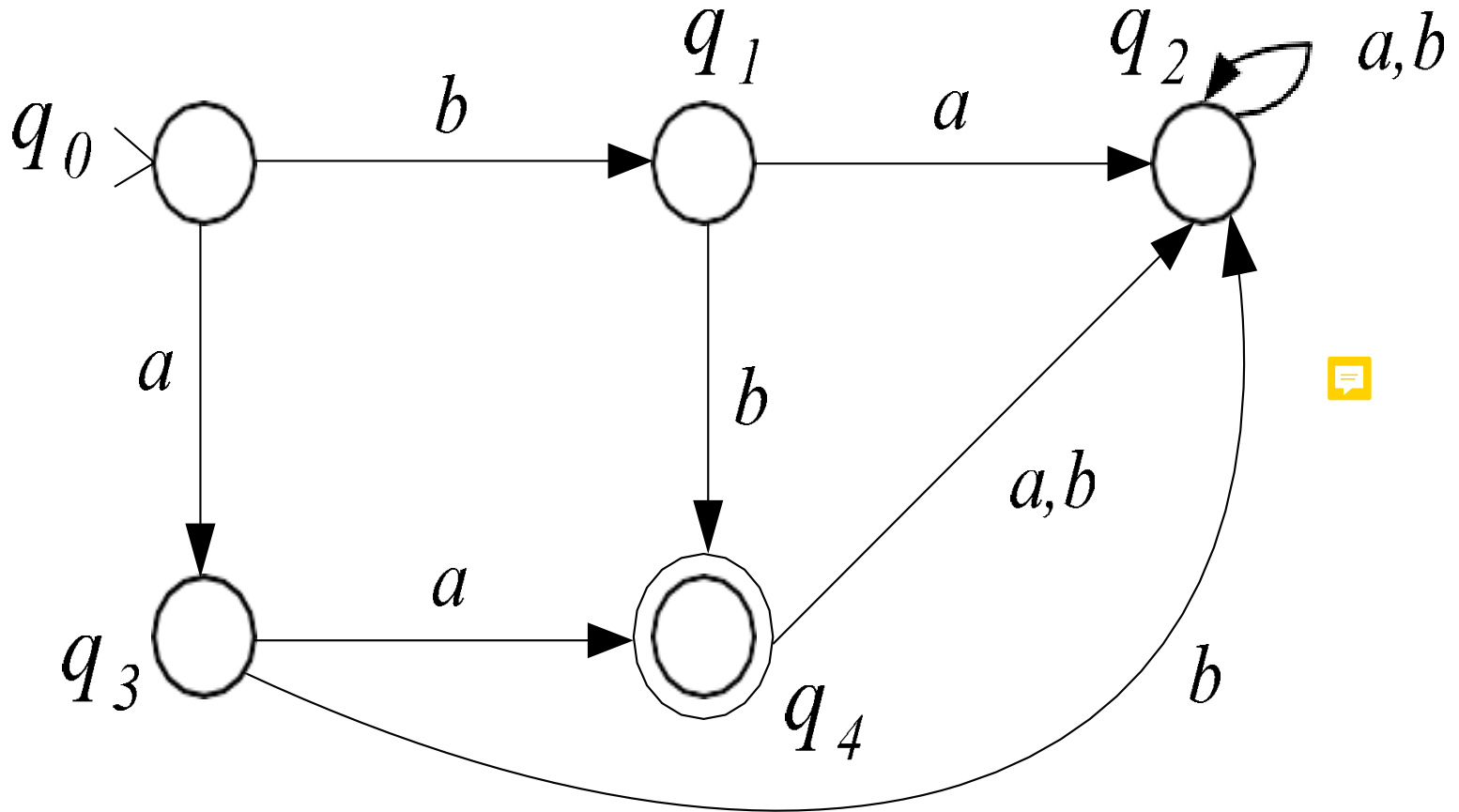
burada  $q_3$  **dead state** olarak adlandırılır.



# Deterministic Finite Automata

Örnek:

Aşağıdaki otomatın tanıdığı dili regular expression ile ifade ediniz.



# Deterministic Finite Automata

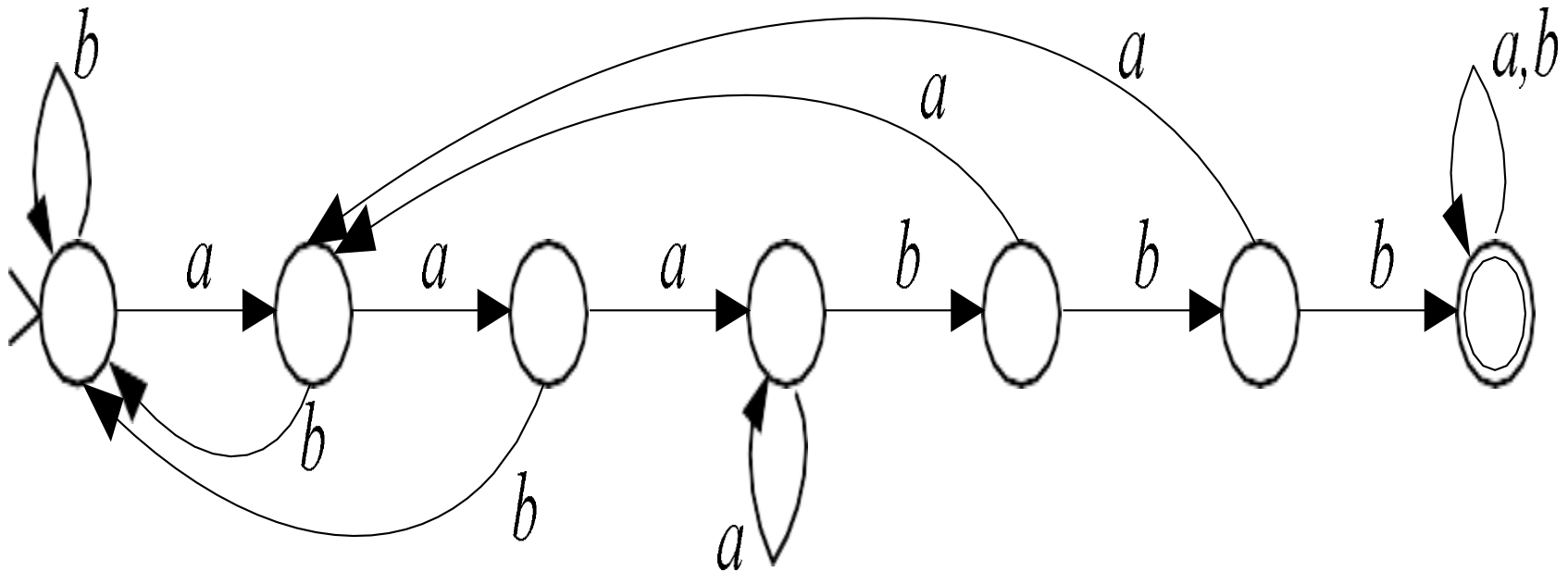
Örnek:

$L(M) = \{w \in \{a, b\}^* : w \text{ içerisinde } aaabbb \text{ substring' i bulunur}\}$  dilini sağlayan DFA'yı elde ediniz.

# Deterministic Finite Automata

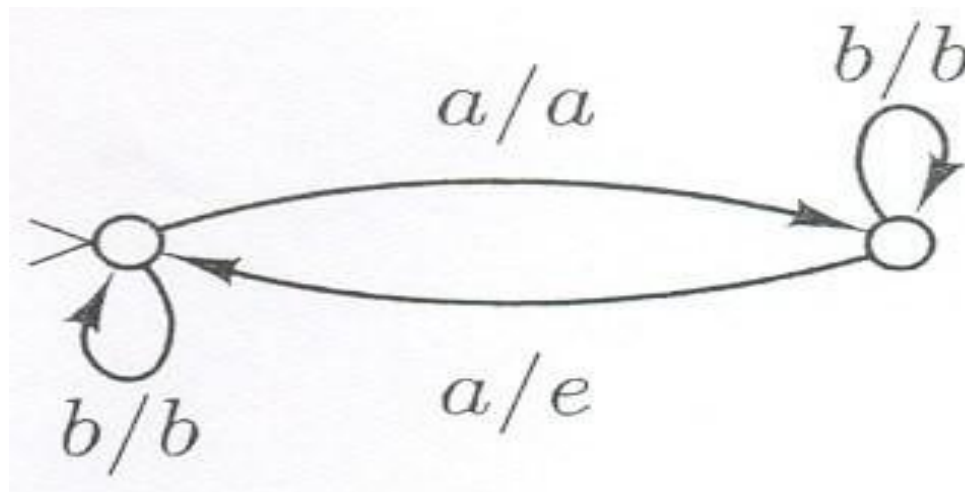
Örnek:

$L(M) = \{w \in \{a, b\}^* : w \text{ içerisinde } aaabbb \text{ substring'ı bulunur}\}$



# Deterministic Finite-State Transducer

- **Deterministic finite-state transducer**, giriş string'ini çıkış string'ine çeviren bir sonlu otomattır.
- Transducer her durum geçişinde output tape üzerine bir sembol, boş sembol veya string yazar.
- $a/w$  girişten  $a$  okur ve çıkışta  $w$  yazar.

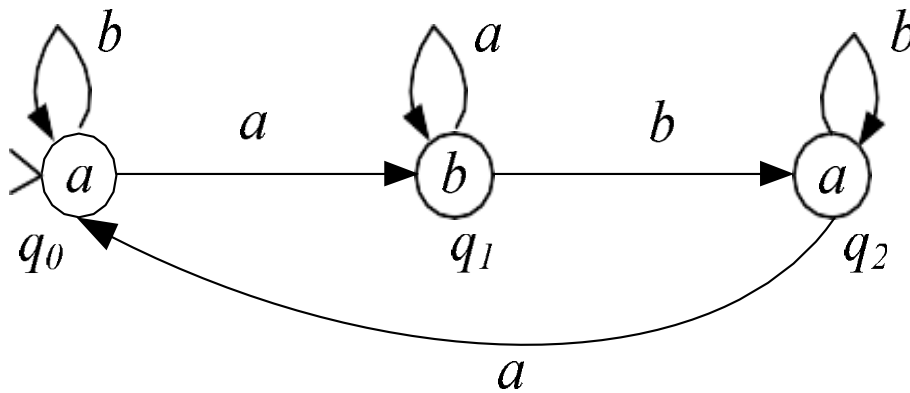


- Yukarıdaki otomat bütün  $b$ 'leri geçer ve her iki  $a$ 'dan ikincisinin yerine  $e$  yazar.

# Deterministic Finite-State Transducer

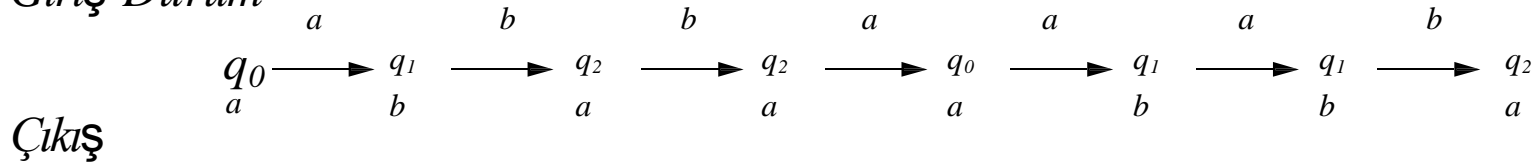
- Deterministic finite-state transducer, **MOORE** ve **MEALY** makinesi olarak iki türdedir.

## Moore makinesi



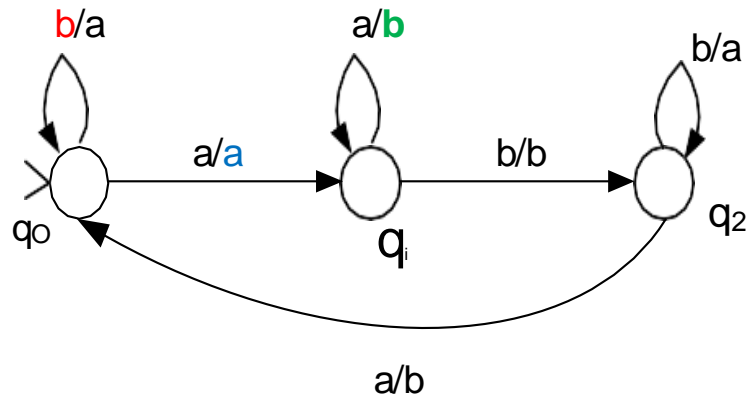
state	input	next state	output
<b>q<sub>0</sub></b>	a	q <sub>1</sub>	<b>a</b>
<b>q<sub>0</sub></b>	b	q <sub>0</sub>	<b>a</b>
<b>q<sub>1</sub></b>	a	q <sub>1</sub>	<b>b</b>
<b>q<sub>1</sub></b>	b	q <sub>2</sub>	<b>b</b>
<b>q<sub>2</sub></b>	a	q <sub>0</sub>	<b>a</b>
<b>q<sub>2</sub></b>	b	q <sub>2</sub>	<b>a</b>

Giriş Durum

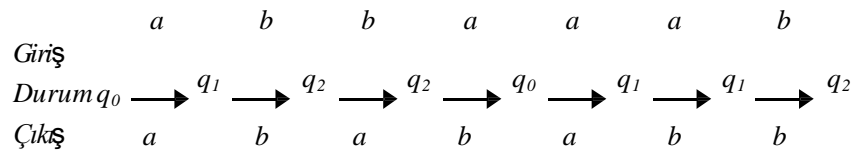


# Deterministic Finite-State Transducer

## Mealy makinesi



<i>state</i>	<i>input</i>	<i>next state</i>	<i>output</i>
$q_0$	$a$	$q_1$	$a$
$q_0$	$b$	$q_0$	$a$
$q_1$	$a$	$q_1$	$b$
$q_1$	$b$	$q_2$	$b$
$q_2$	$a$	$q_0$	$b$
$q_2$	$b$	$q_2$	$a$

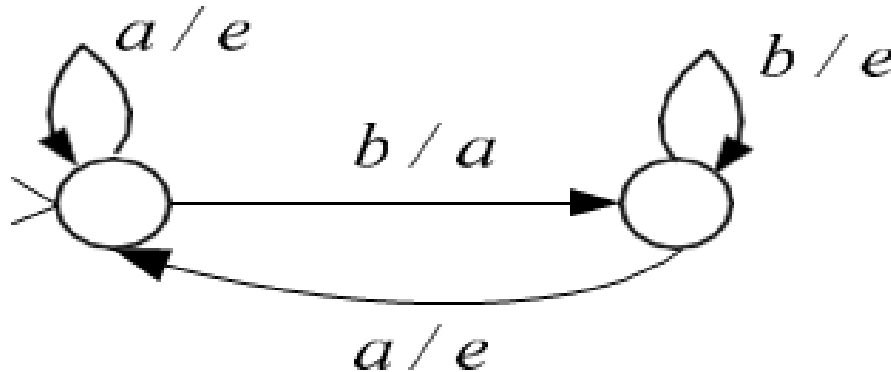


# Deterministic Finite-State Transducer

Örnek: (2.1.4.a(i))

- $w$  giriş için  $a^n$  üreten transducer durum çizeneğini oluşturunuz.  $n$  sayısı  $a$ 'nın tekrar sayısını gösterir ve  $w$  içinde  $ab$  substring'inin sayısına eşittir.

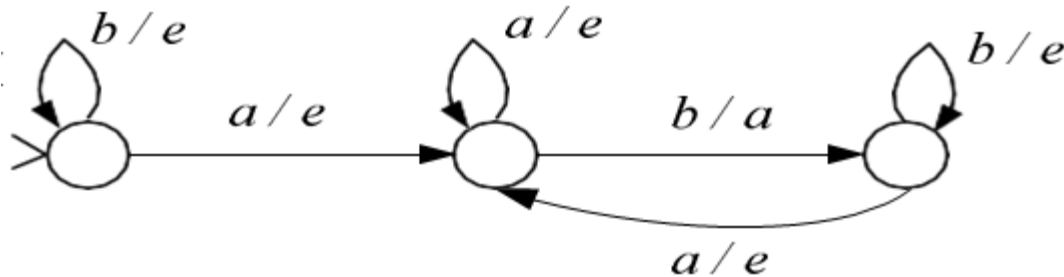
Hangi girişler için hatalı çıkış üretir ?



# Deterministic Finite-State Transducer

Örnek: (2.1.4.a(i))

- $w$  giriş için  $a^n$  üreten transducer durum çizeneğini oluşturunuz.  $n$  sayısı  $a$ 'nın tekrar sayısını gösterir ve  $w$  içinde  $ab$  substring'inin sayısına eşittir.

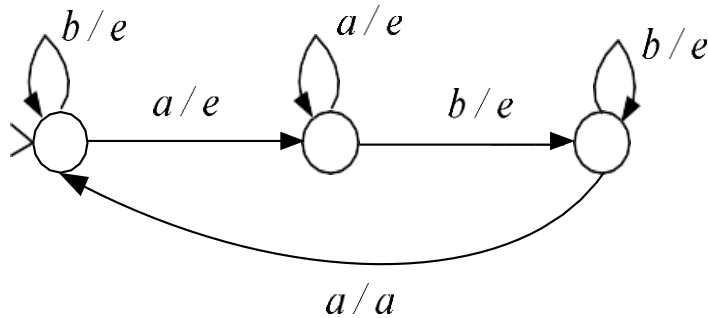




# Deterministic Finite-State Transducer

Örnek: (2.1.4.a(ii))

- $w$  giriş için  $a^n$  üreten transducer durumu çizeneğini oluşturunuz.  
 $n$  sayısı  $a$ 'nın tekrar sayısını gösterir ve  $w$  içinde  $aba$  substring'inin sayısına eşittir.

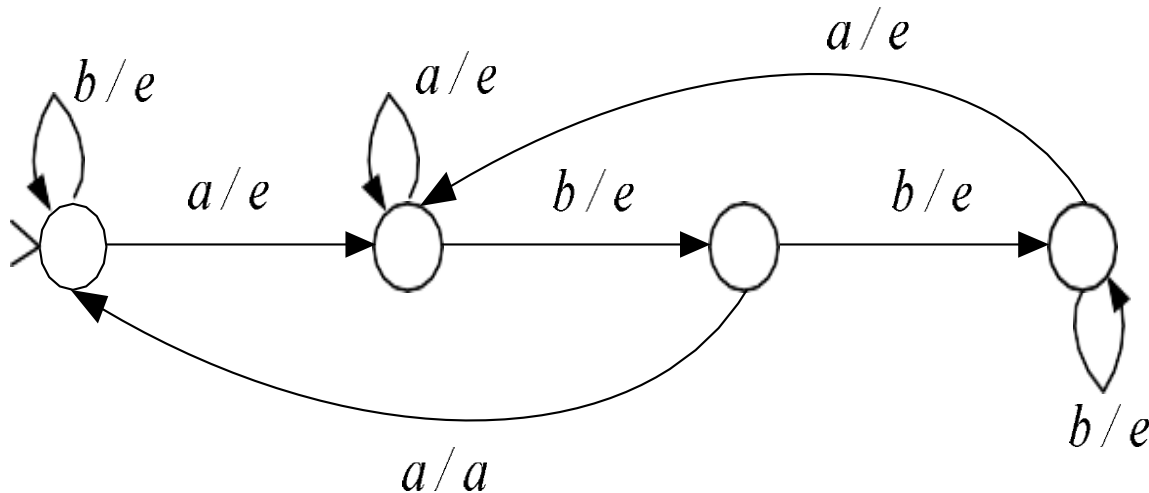


Hangi giriş için  
hatalı çıkış üretir ?

# Deterministic Finite-State Transducer

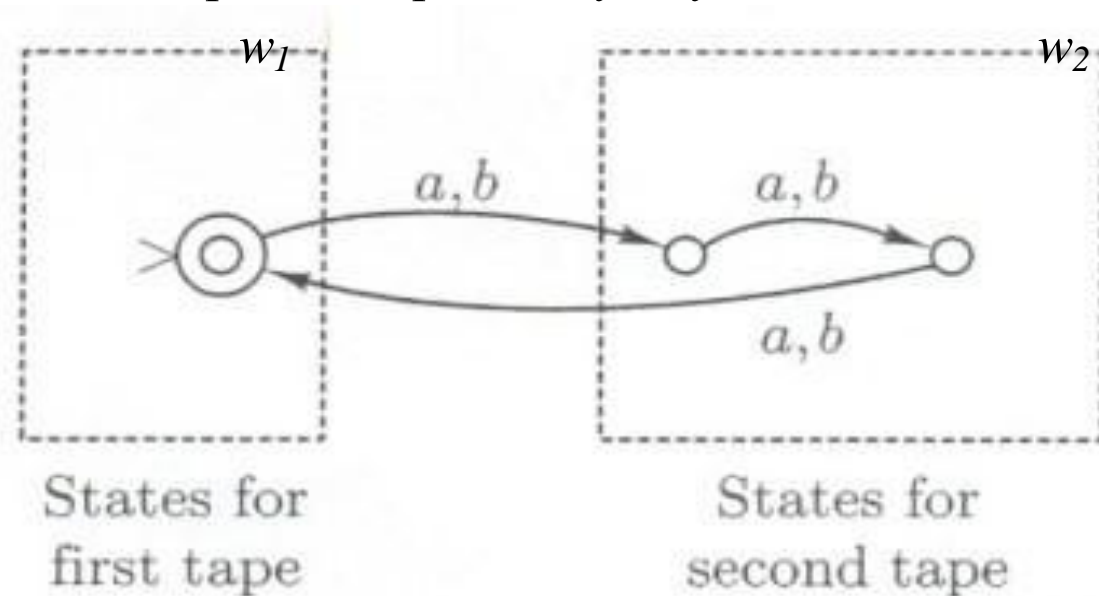
Örnek: (2.1.4.a(ii))

- $w$  giriş için  $a^n$  üreten transducer durumu çizeneğini oluşturunuz.  
 $n$  sayısı  $a$ 'nın tekrar sayısını gösterir ve  $w$  içinde  $aba$  substring'inin sayısına eşittir.



# Deterministic 2-Tape Finite Automata

- **Deterministic 2-tape finite automaton**, string çiftlerini kabul eden ve iki girişe sahip olan sonlu otomattır.
- Transition function 1.tape ve 2.tape için ayrı ayrı tanımlıdır.

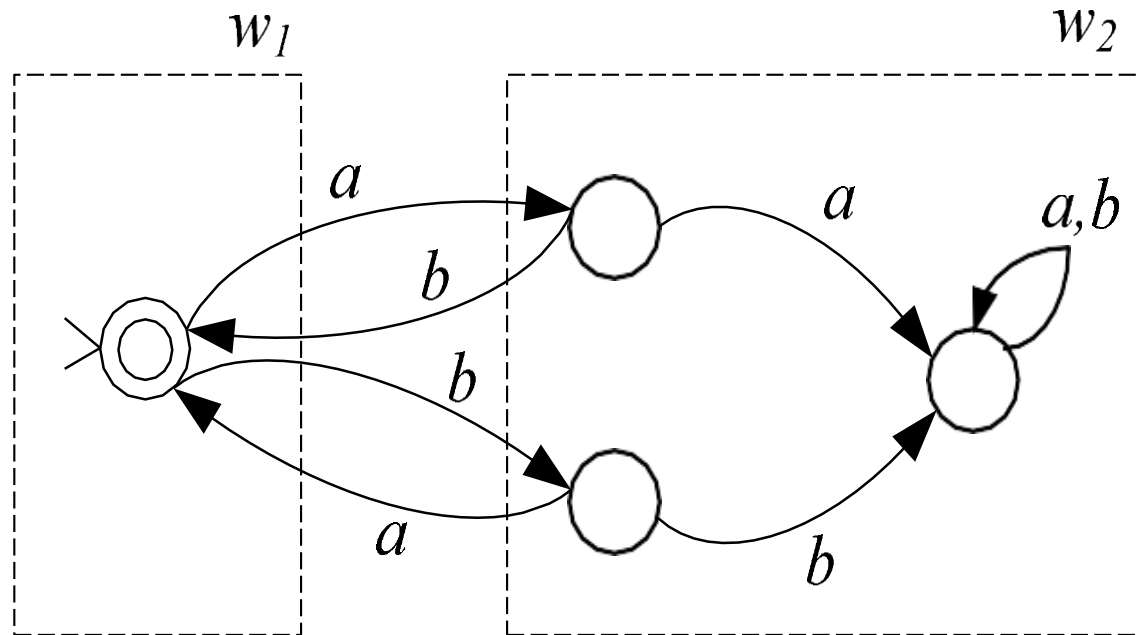


- $(w_1, w_2) \in \{a, b\}^* \times \{a, b\}^*$  stringlerini  $|w_2| = 2 |w_1|$  olması koşuluyla kabul eder.

# Deterministic 2-Tape Finite Automata

## Örnek:

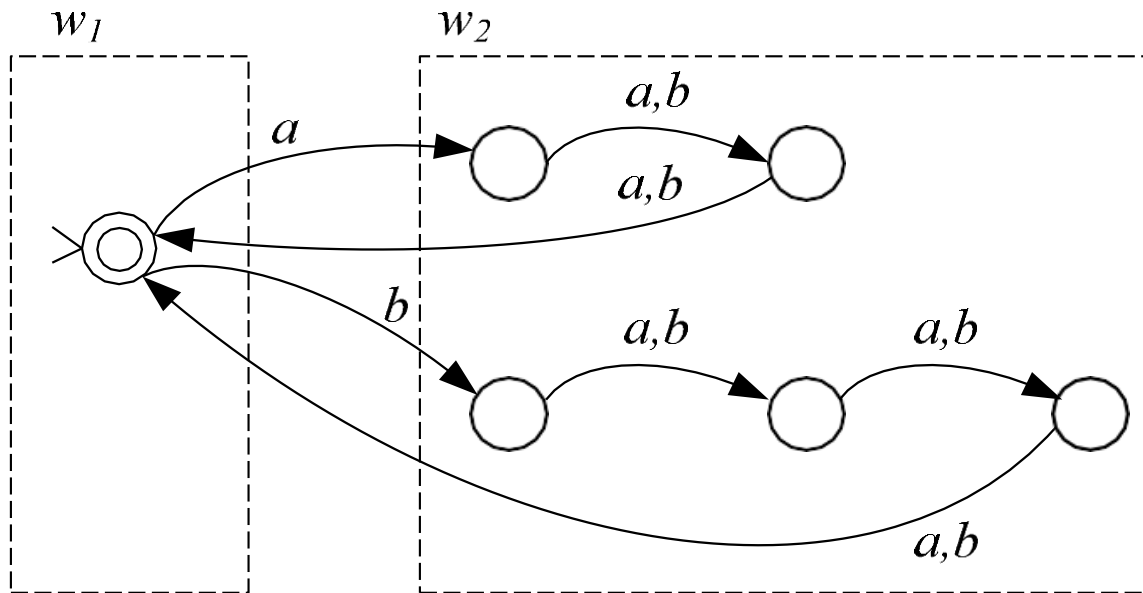
Bir 2-tape sonlu otomat, tüm  $(w_1, w_2) \in \{a, b\}^* \times \{a, b\}^*$  string çiftlerini  $|w_1| = |w_2|$  olması ve tüm  $i$ 'ler için  $w_1(i) \neq w_2(i)$  olması koşuluyla kabul eder.



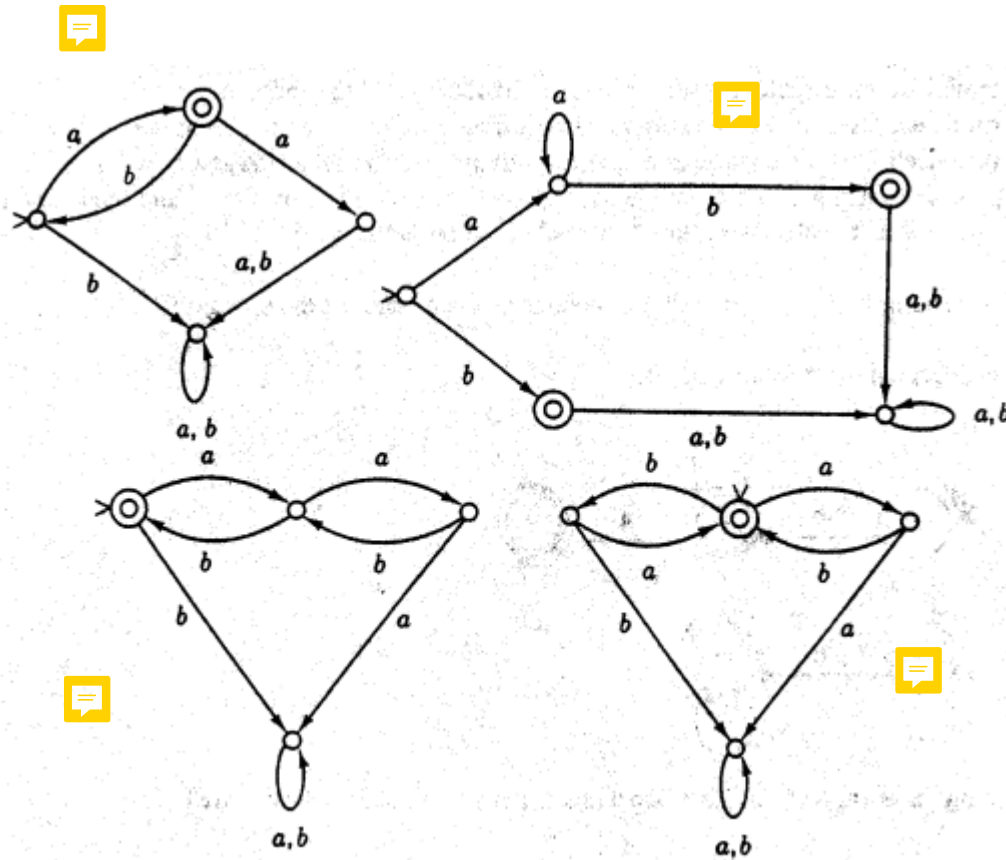
# Deterministic 2-Tape Finite Automata

## Örnek:

Bir 2-tape sonlu otomat, tüm  $(w_1, w_2) \in \{a, b\}^* \times \{a, b\}^*$  string çiftlerini  $|w_2|$  değerinin,  $w_1$  içindeki  $a$  sayısının iki katı artı  $b$  sayısının üç katına eşit veya büyük olması koşuluyla kabul eder.

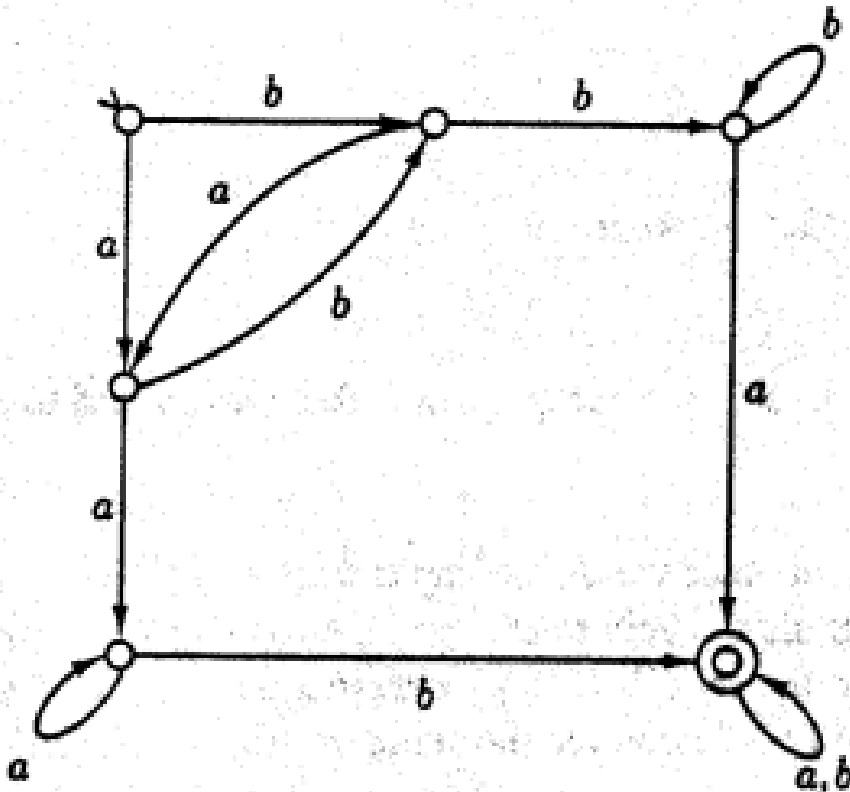


# Exercise



Yandaki herbir DFA ile hangi dilin kabul edildiğini sözel olarak ifade ediniz.

# Exercise



- Yandaki DFA ile hangi dilin kabul edildiğini sözel olarak ifade ediniz.

# Exercise

$\{w \in \{a, b\}^* \mid w\text{'da tek sayıda a'lar ve çift sayıda b'ler olsun}\}$

(even,even)

(even,odd) start state

(odd,even) final state

(odd,odd)



# Ödev

- Problemleri çözünüz 2.1.2, 2.1.3, 2.1.7 (sayfa 60-63)

- JFLAP indiriniz.

<https://www.jflap.org/> önce form doldurup sonra indirip kurunuz.

[https://www.youtube.com/playlist?list=PLeaAjeNjt7tTAH3LvVMVeR\\_rOVOgLLx6D](https://www.youtube.com/playlist?list=PLeaAjeNjt7tTAH3LvVMVeR_rOVOgLLx6D) izleyerek

<https://www.jflap.org/jflapbook/> veya okuyarak öğreniniz.

- JFLAP kullanarak  $\{0, 1\}$  alfabesi üzerinde

$$L(M) = \{w \mid |w| \text{ 5'in katıdır} \}$$

dilini tanıyan bir DFA oluşturun. JFLAP diyagramının yanı sıra, her bir durumun neden dahil edildiğine ve neden her bir durumu neden kabul edip veya etmediğinizi belirleyen kısa bir açıklama (Türkçe sözel olarak) ekleyin.