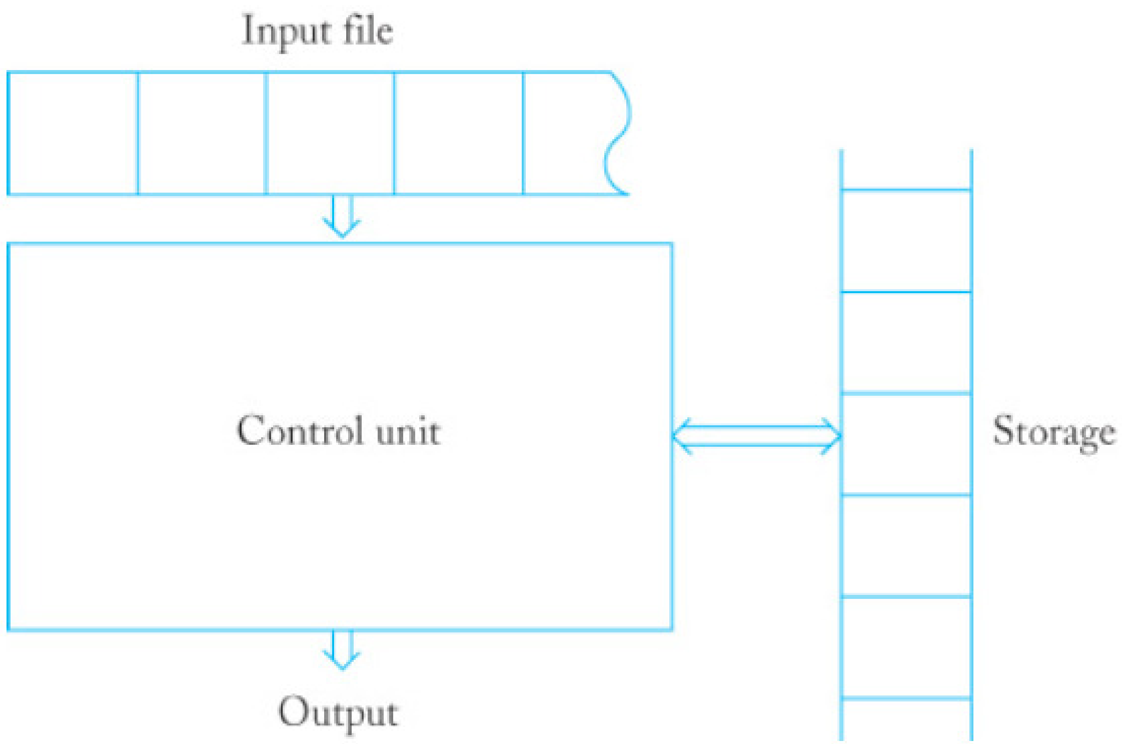
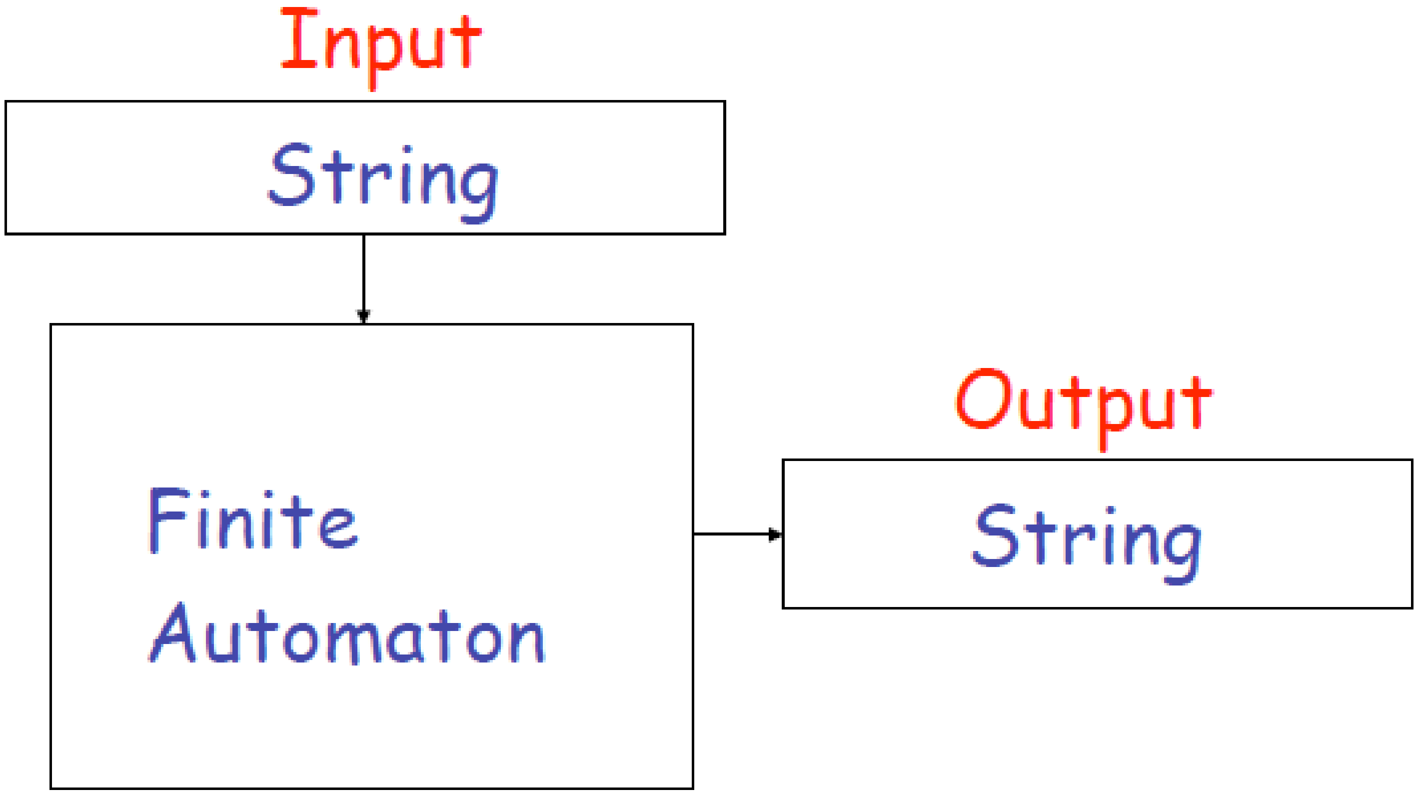
FORMAL DİLLER VE OTOMATLAR

Automata



Otomatların en temel özellikleri girişleri okuyabilen mekanizmalarıdır. Giriş, bir alfabe üzerine oluşturulur. Otomat giriş dosyasından bir dizgiyi okuyabilir ancak değiştiremez. Giriş mekanizması bu dosyayı soldan sağa doğru her seferde tek bir sembol olarak okur. Geçici bir çalışma belleği ve kontrol birimine sahiptir.

Finite Automaton



FINITE AUTOMATA (SONLU OTOMATLAR)

Deterministic Finite Automata (DFA) (Gerekirci Sonlu Otomat)

M = (Q, Σ, δ, q0, F) ile tanımlanır.

Q → durumların sonlu kümesi

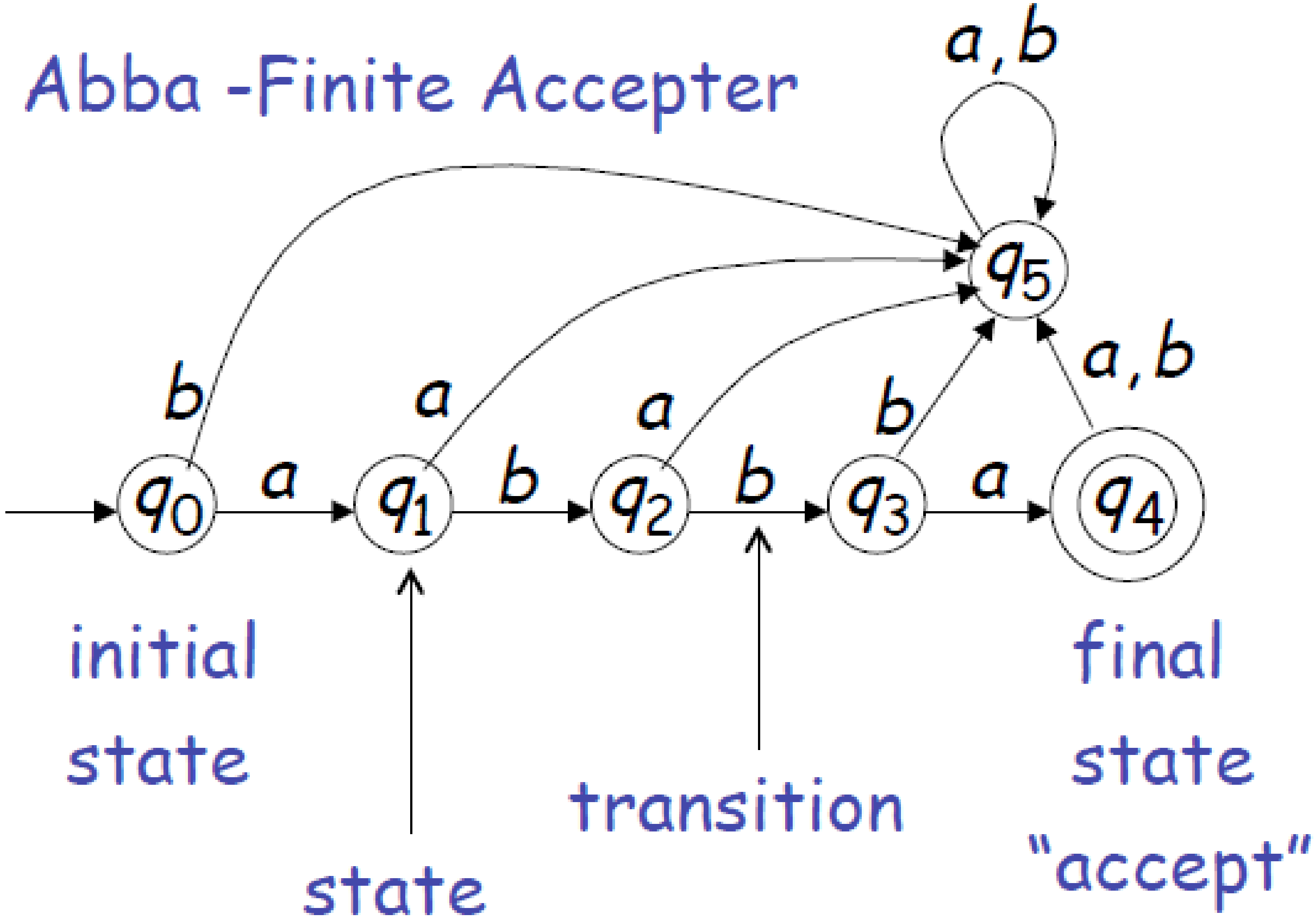
Σ → sembollerin sonlu kümesi

q_0 →başlangıç durumu (initial state, q0 is in Q)

F →kabul durumları kümesi (final/accepting states, which is a subset of Q)

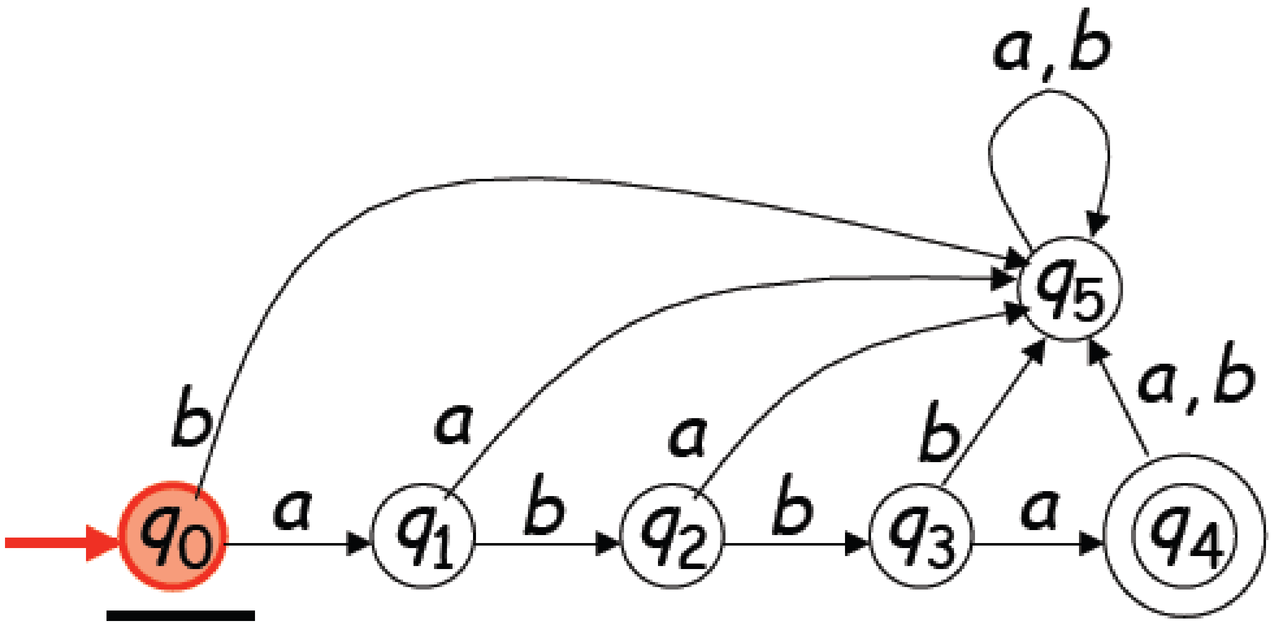
δ →geçiş fonksiyonu (transition function, which is a total function from Q x Σ to Q)

Transition Graph

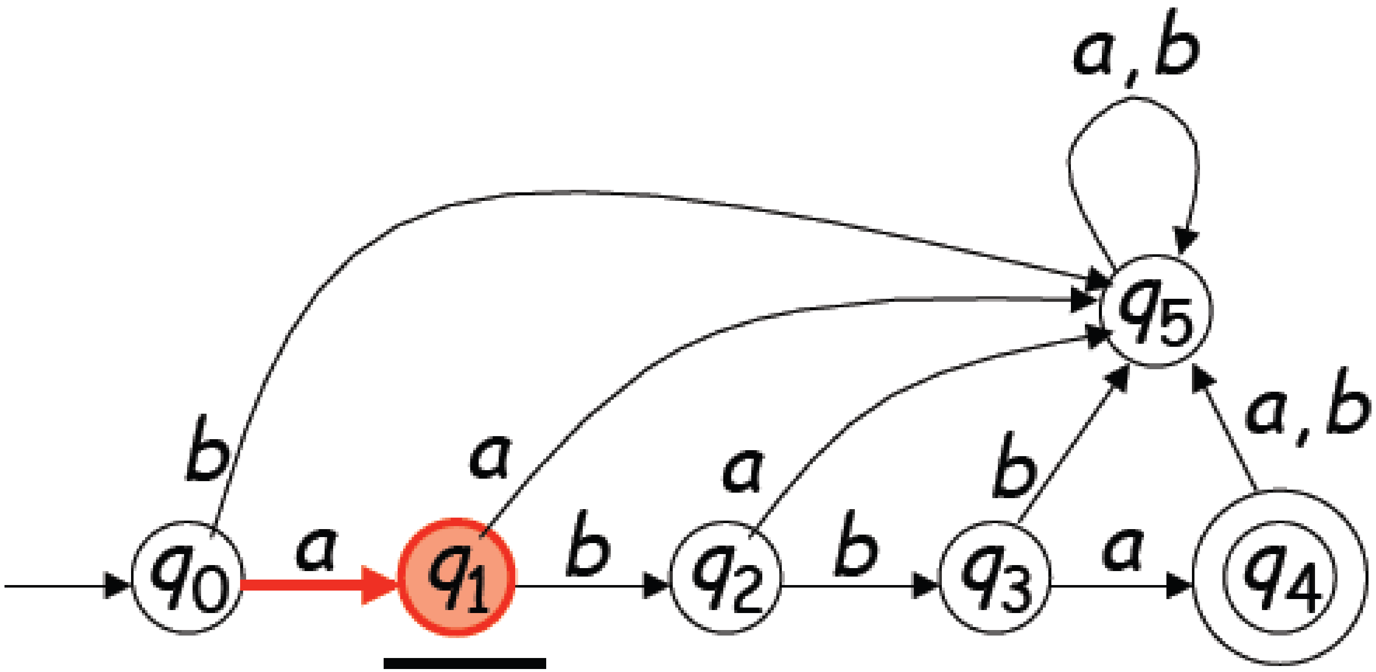


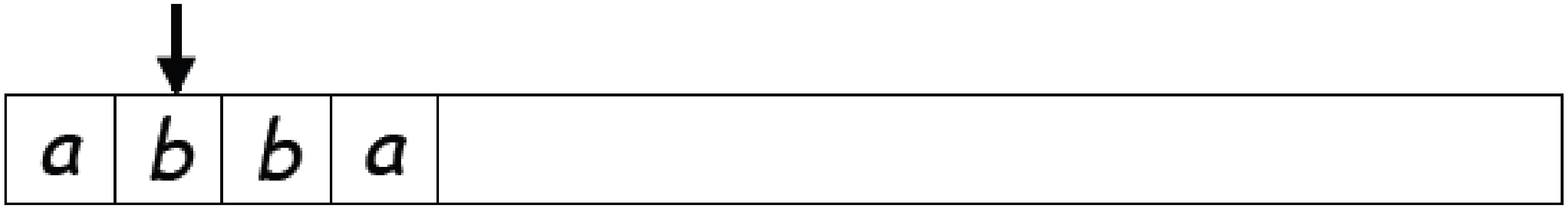
Initial Configuration

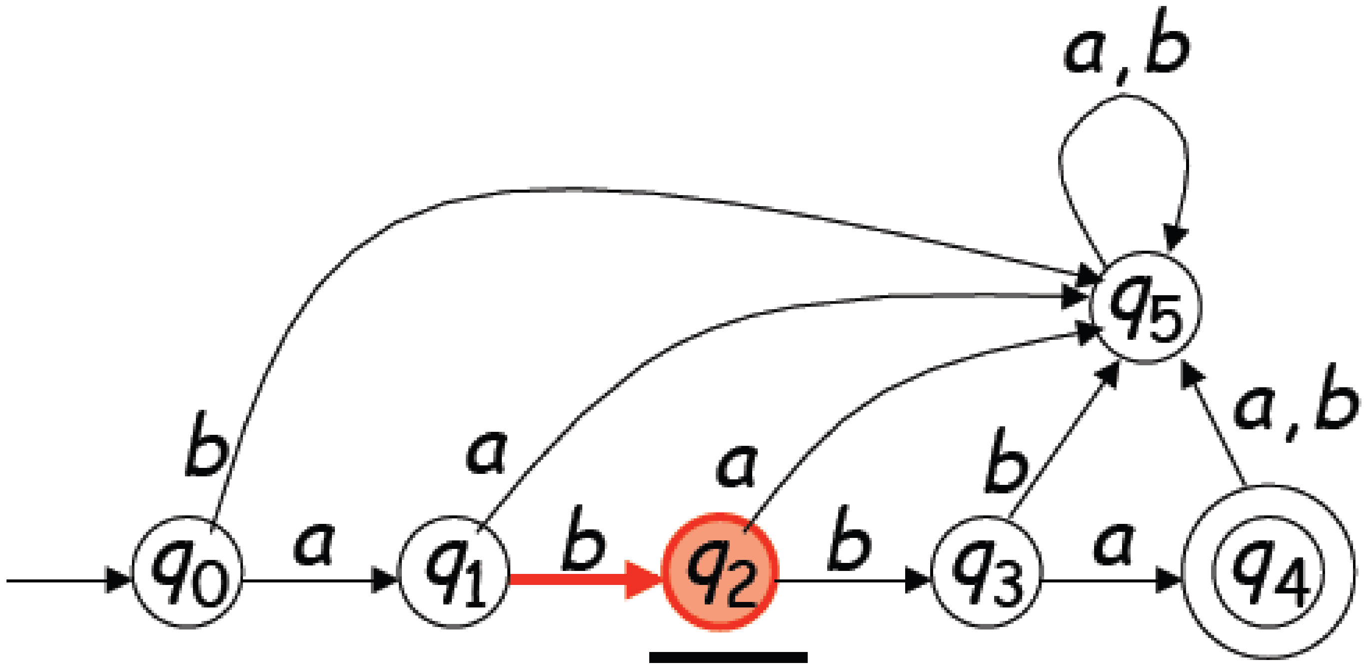
Input String

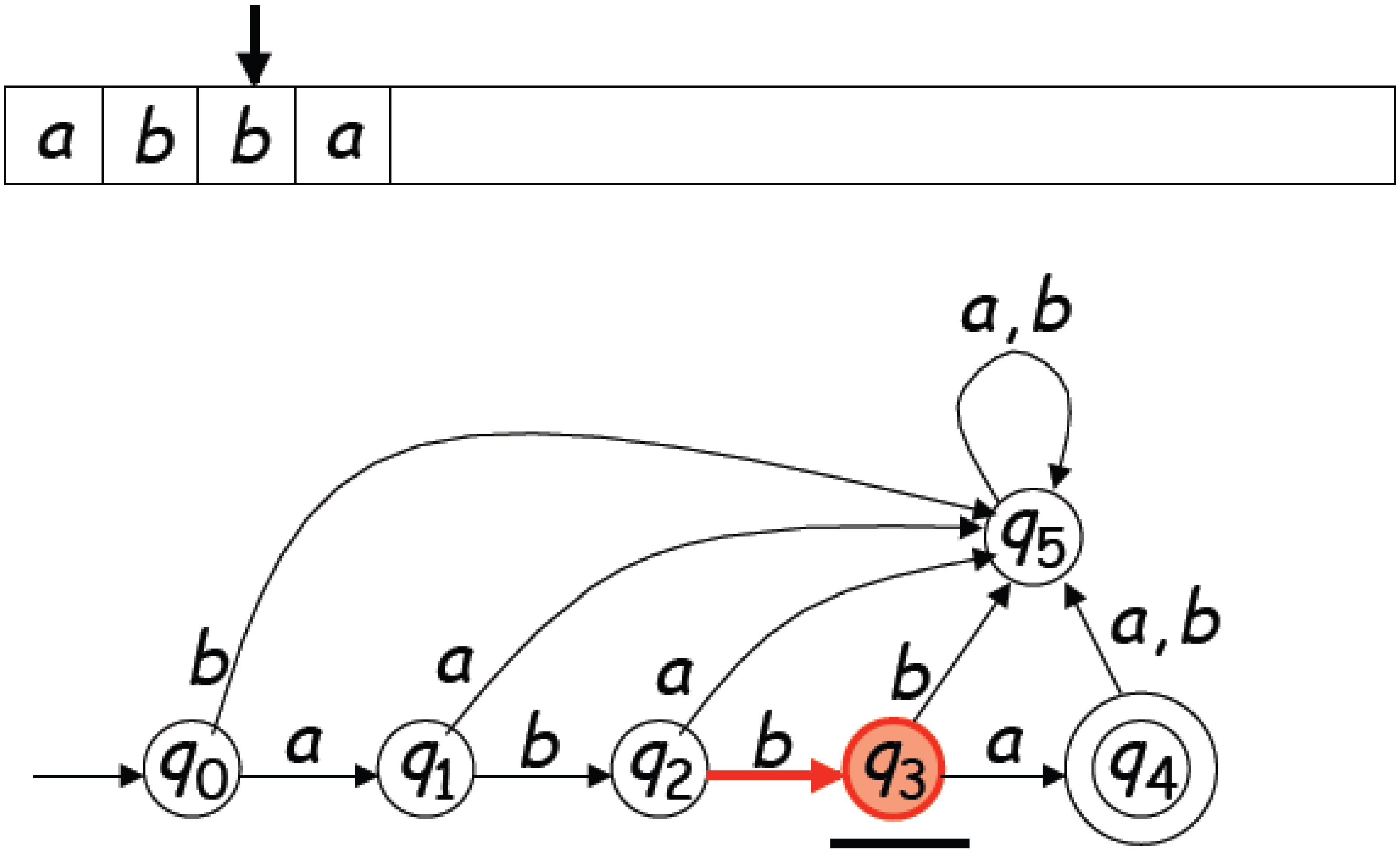


Reading the Input

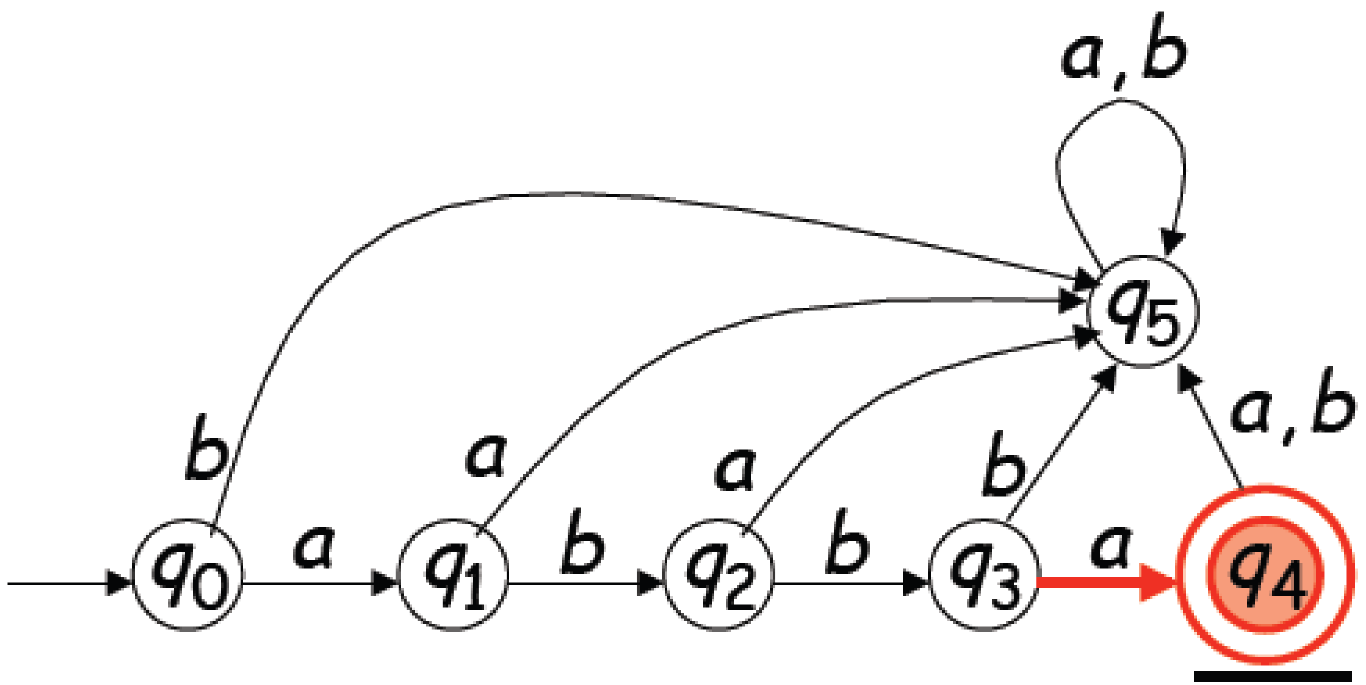



a
b
b
a

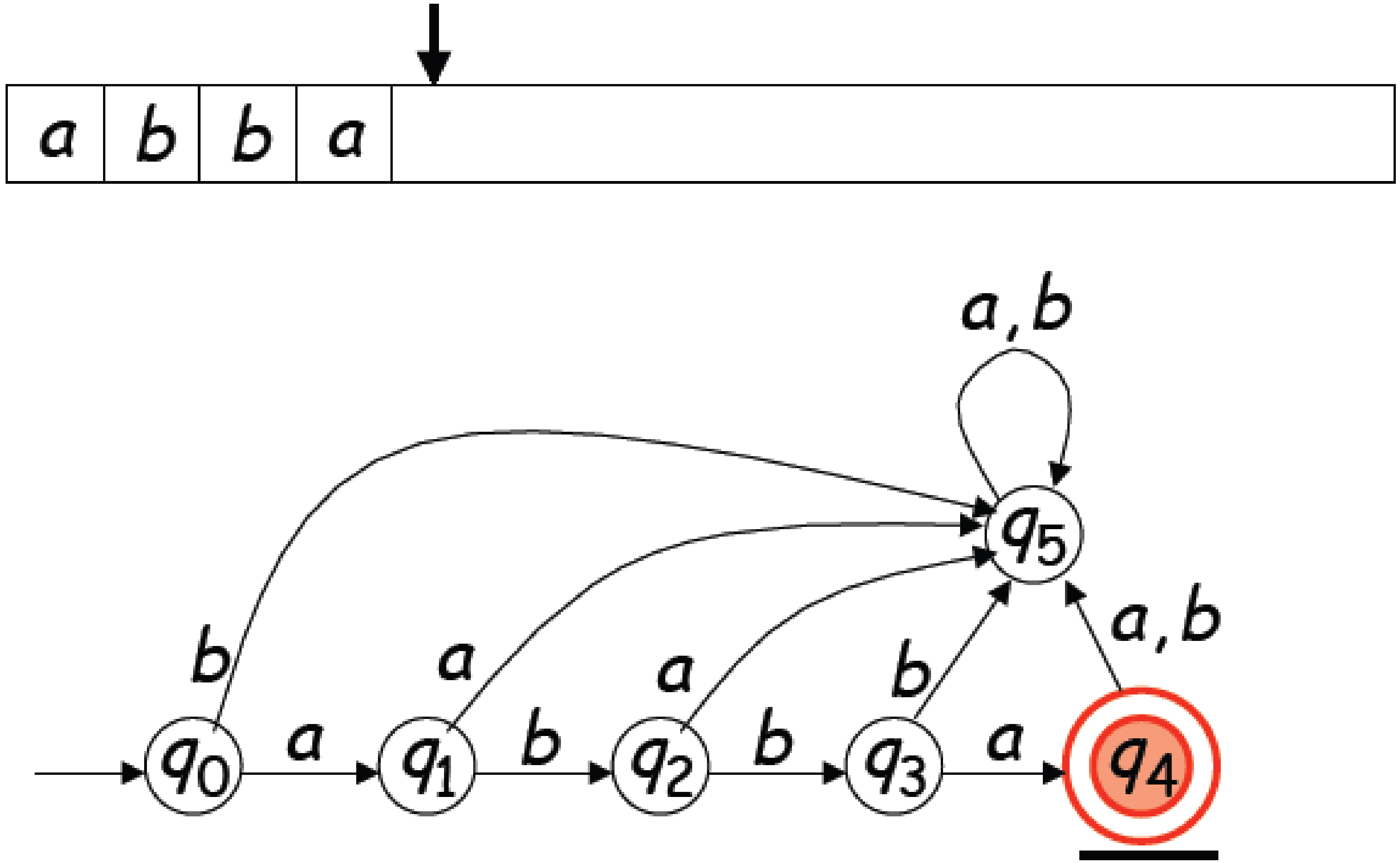

a,b
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4
q5


a
b
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4

a	b	b	a	

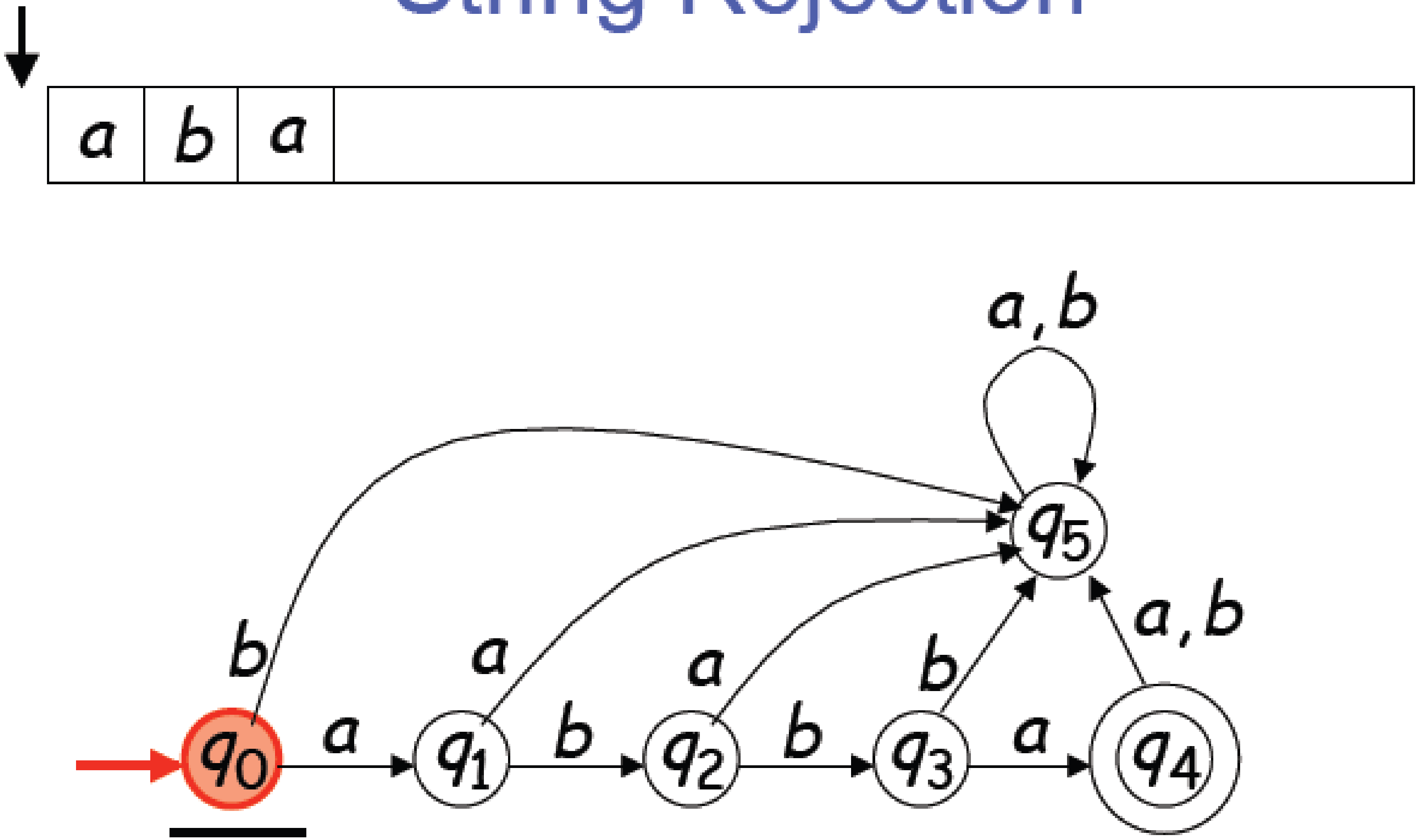


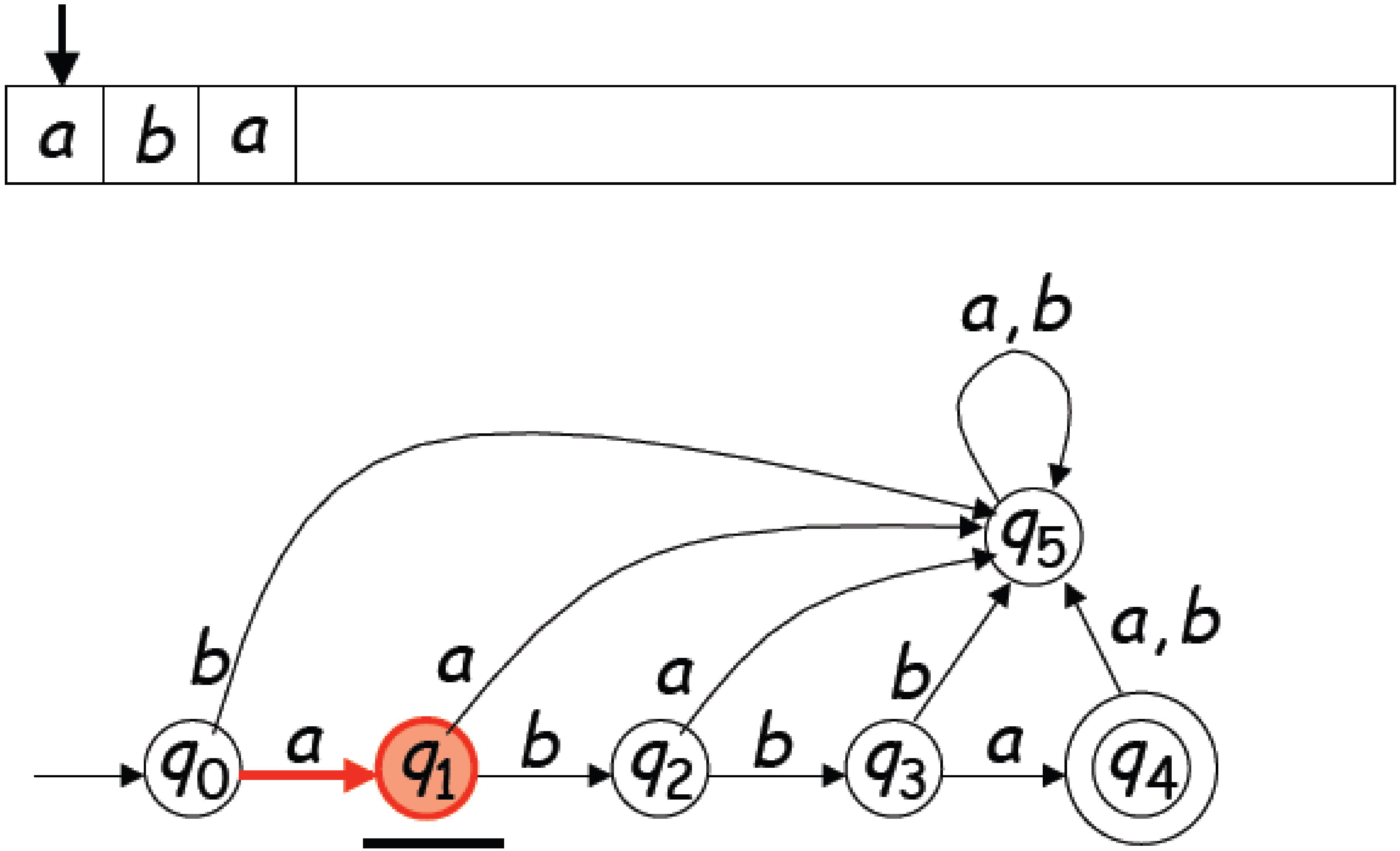
Input finished

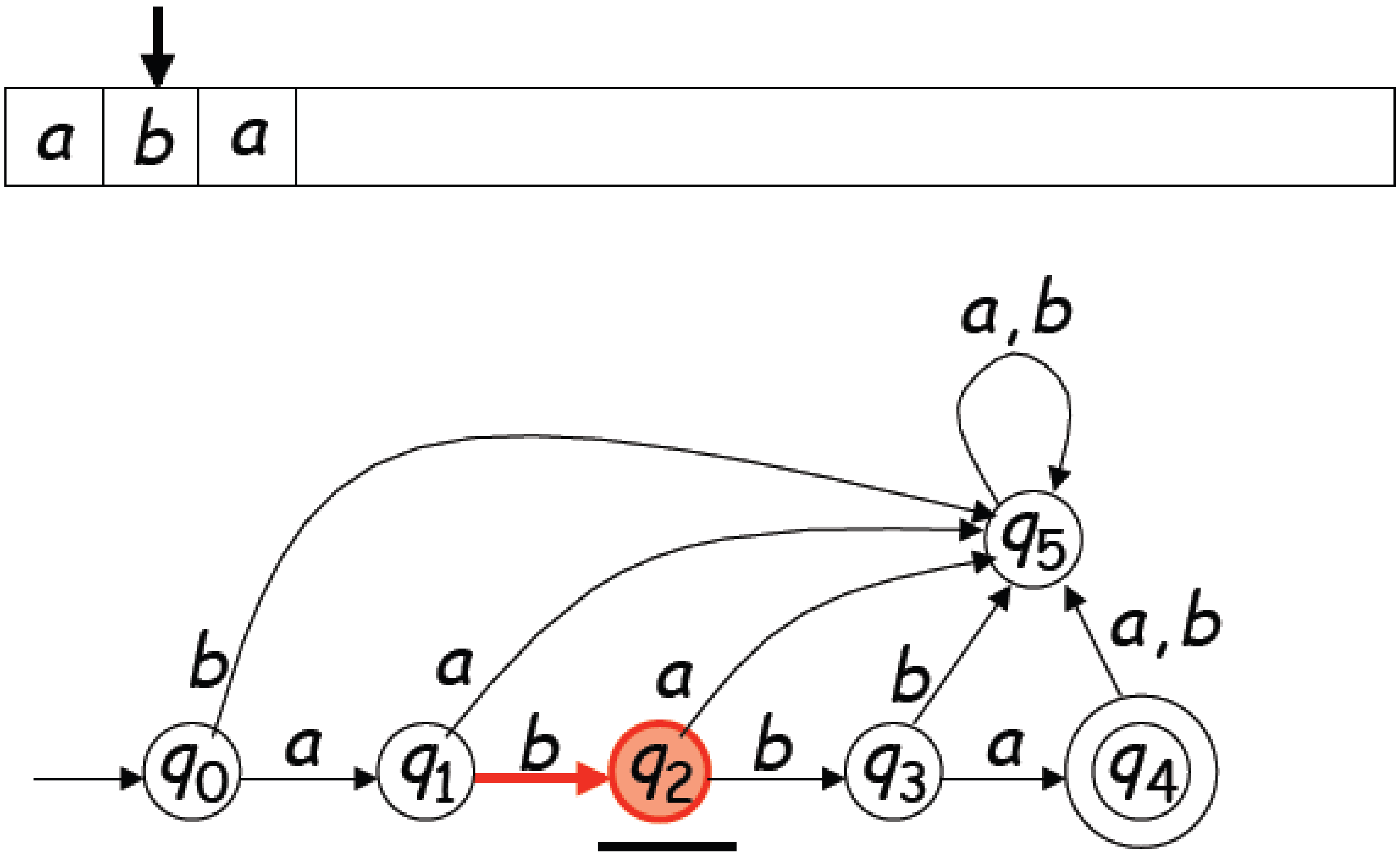


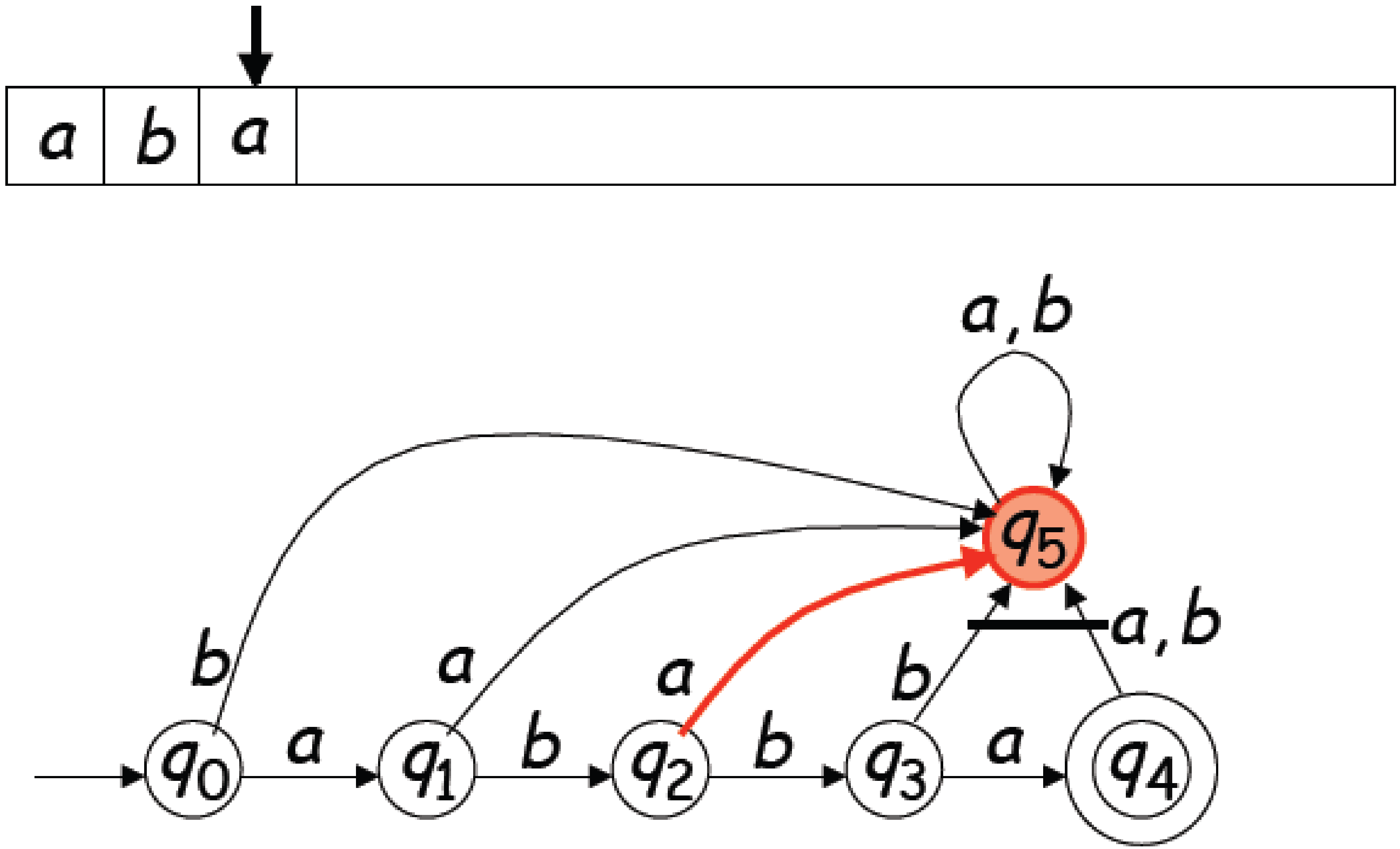
Output: "accept"

String Rejection

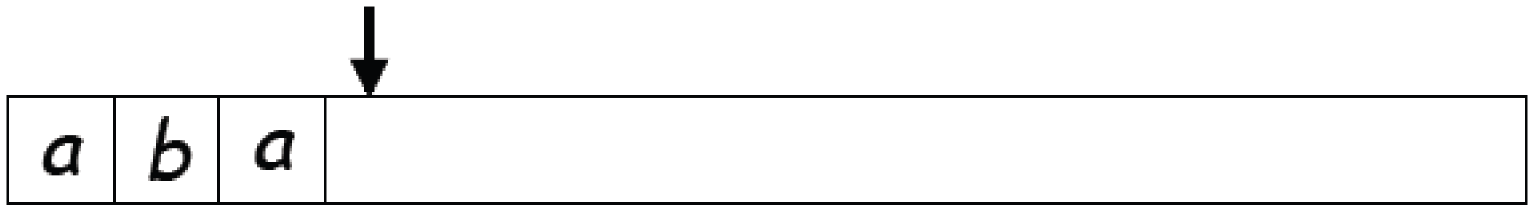


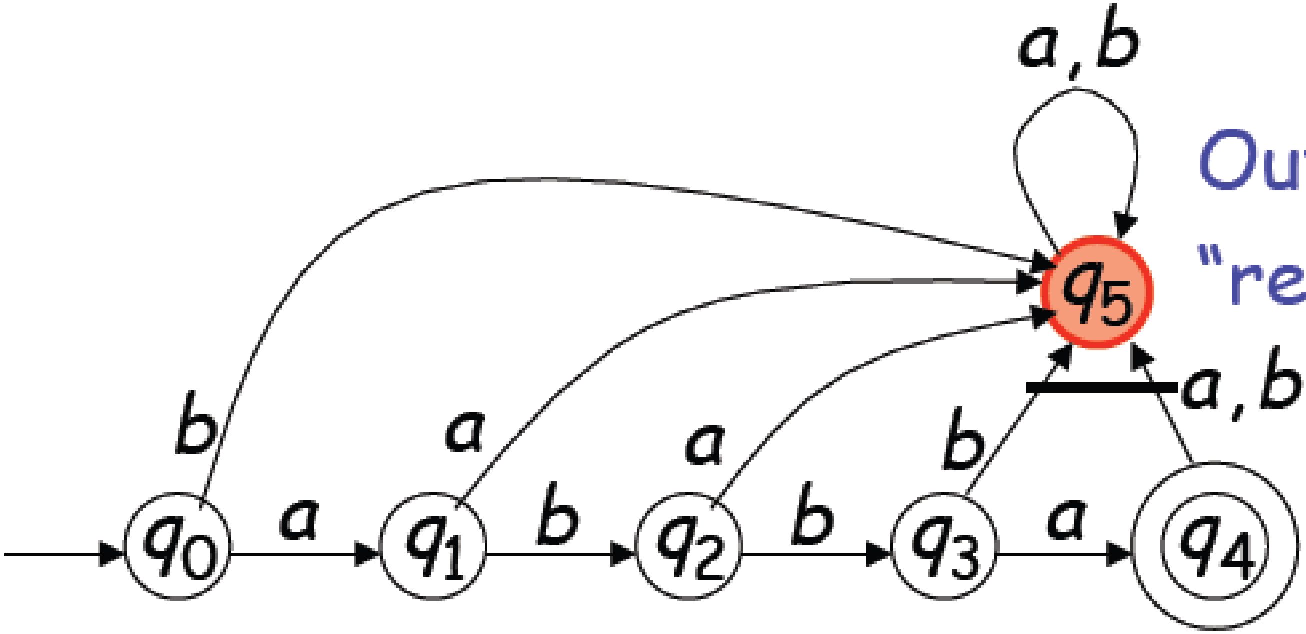

a b a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4


a
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4


a
b
a
a,b
b
a
a
b
a,b
q_0
a
q_1
b
q_2
b
q_3
a
q_4
q_5

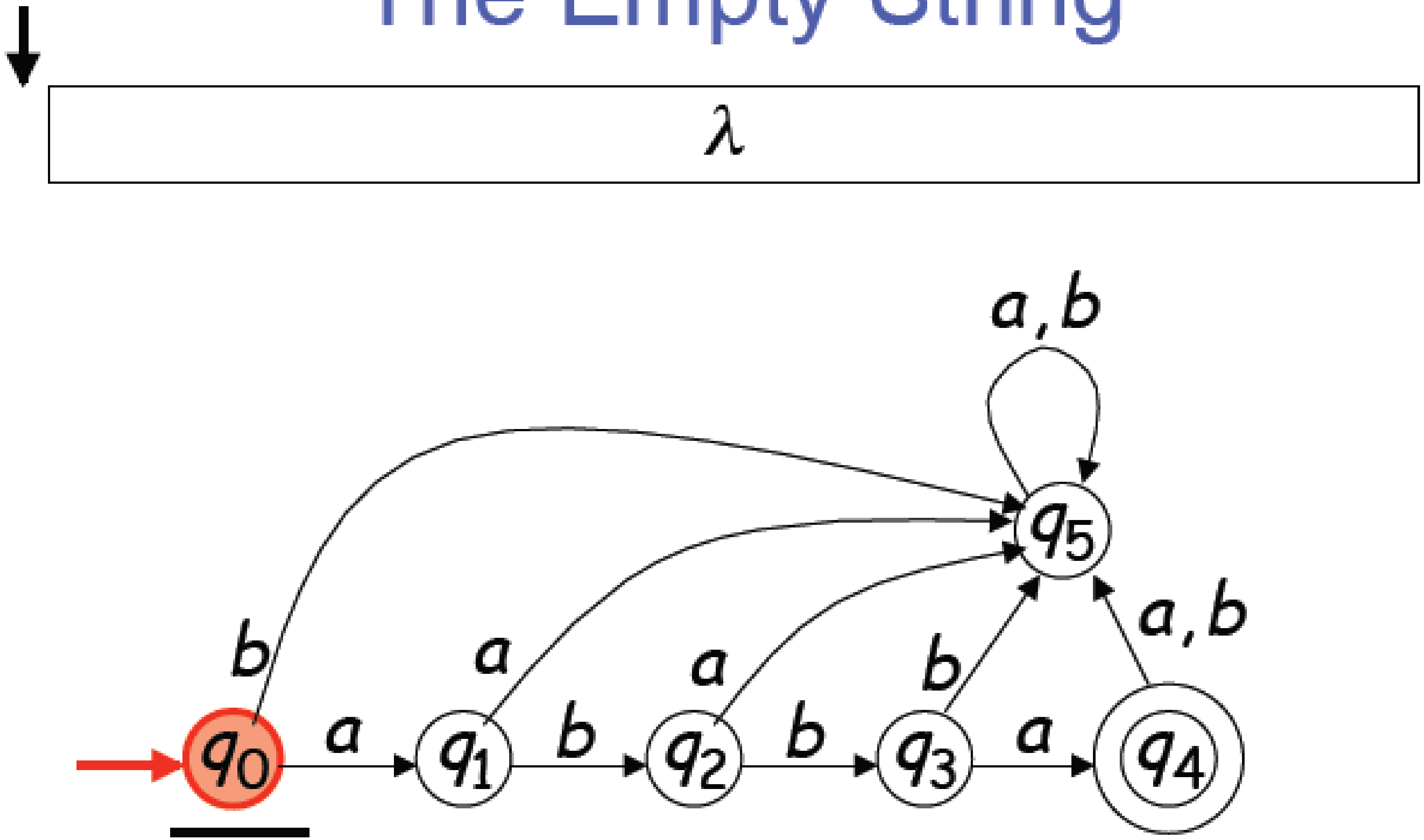
Input finished

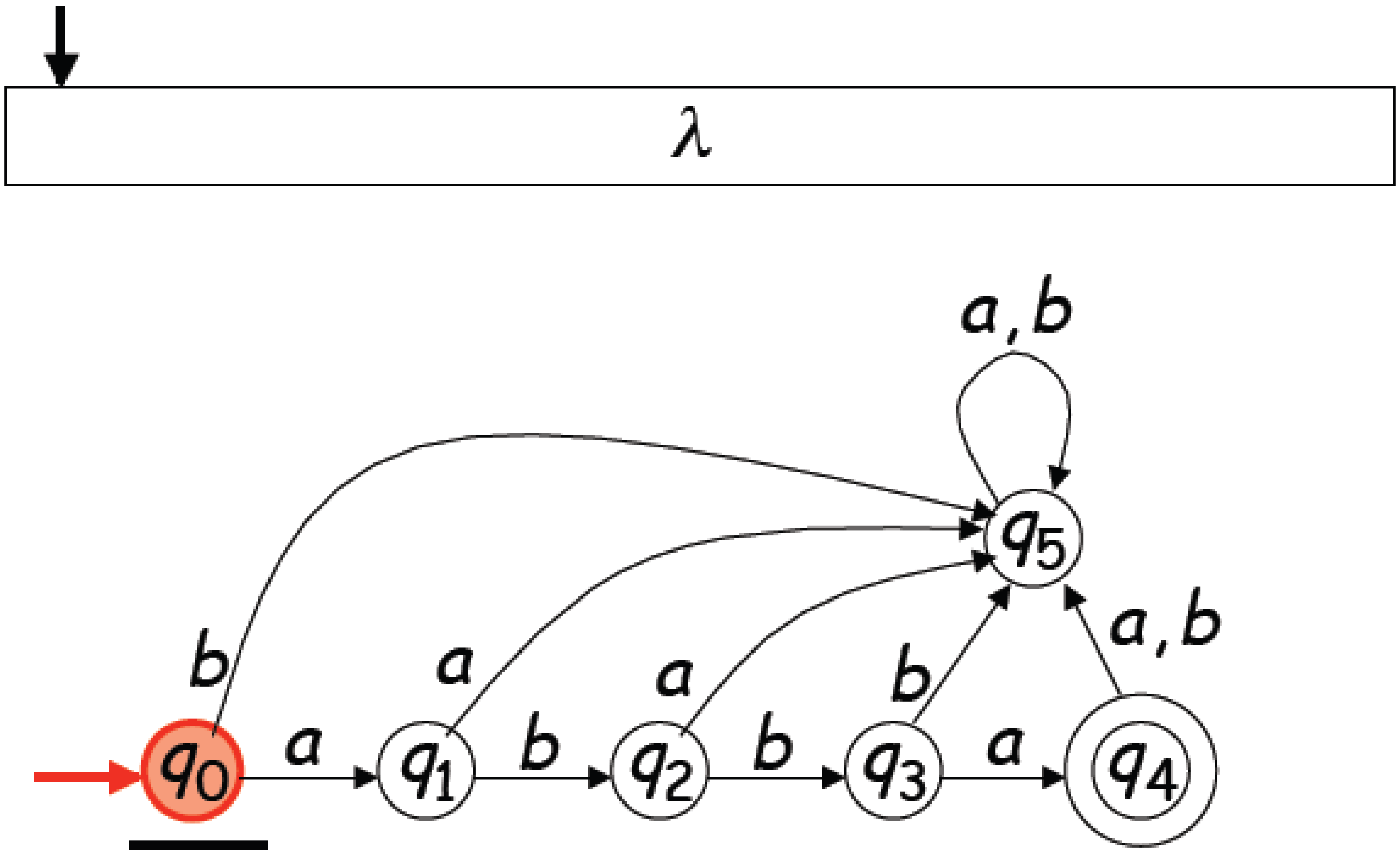




Output: "reject"

The Empty String



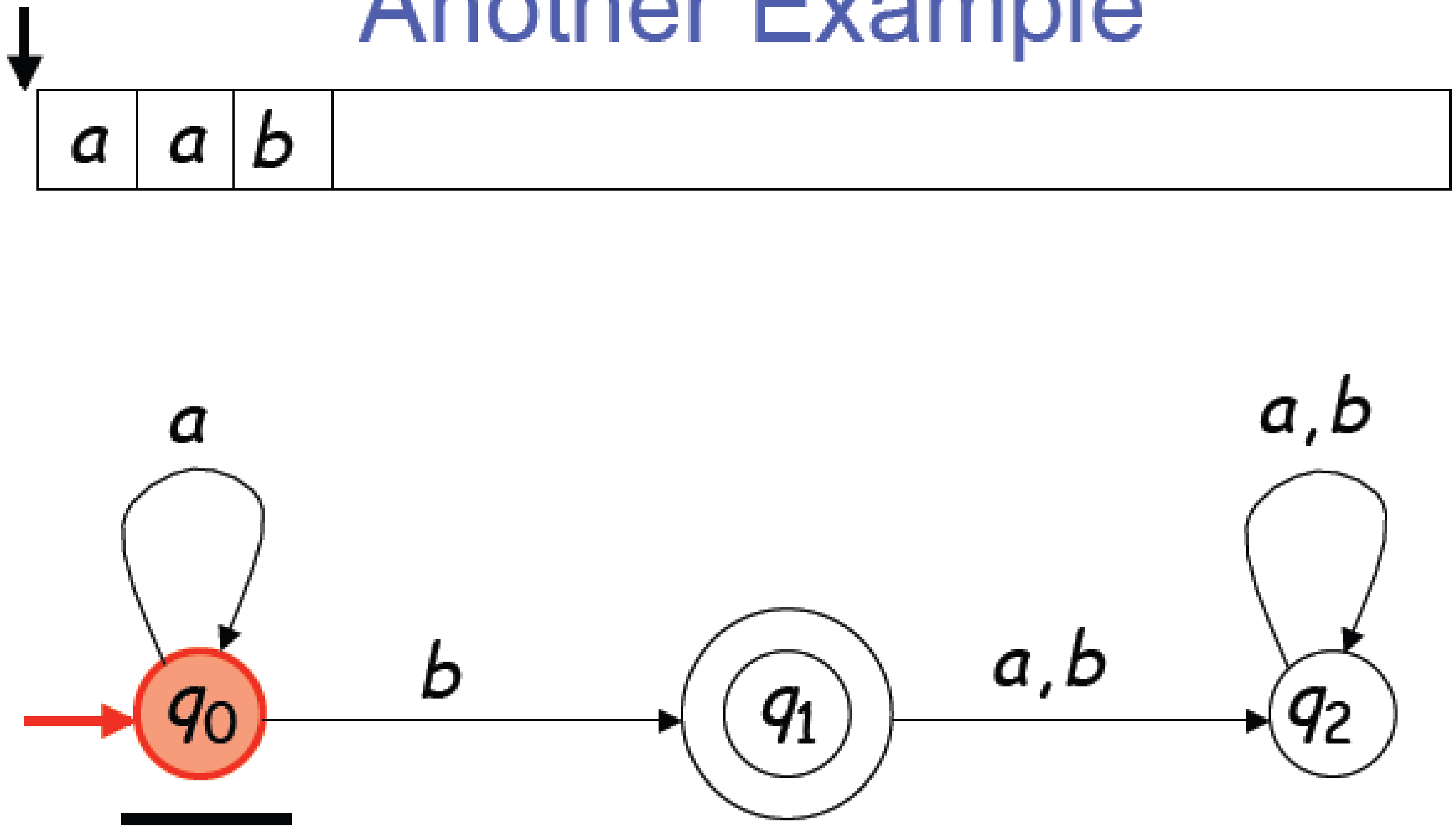


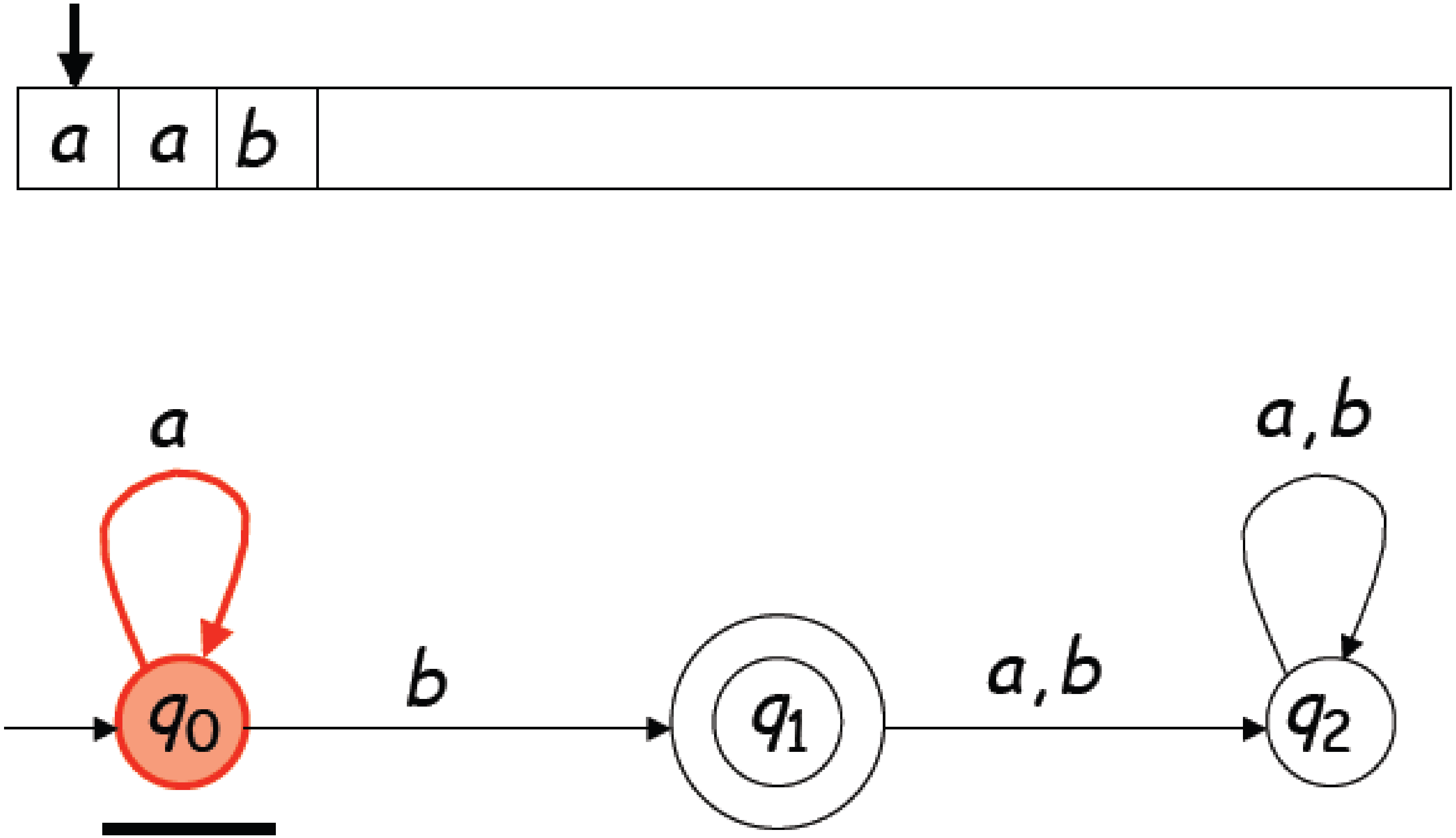
Output:

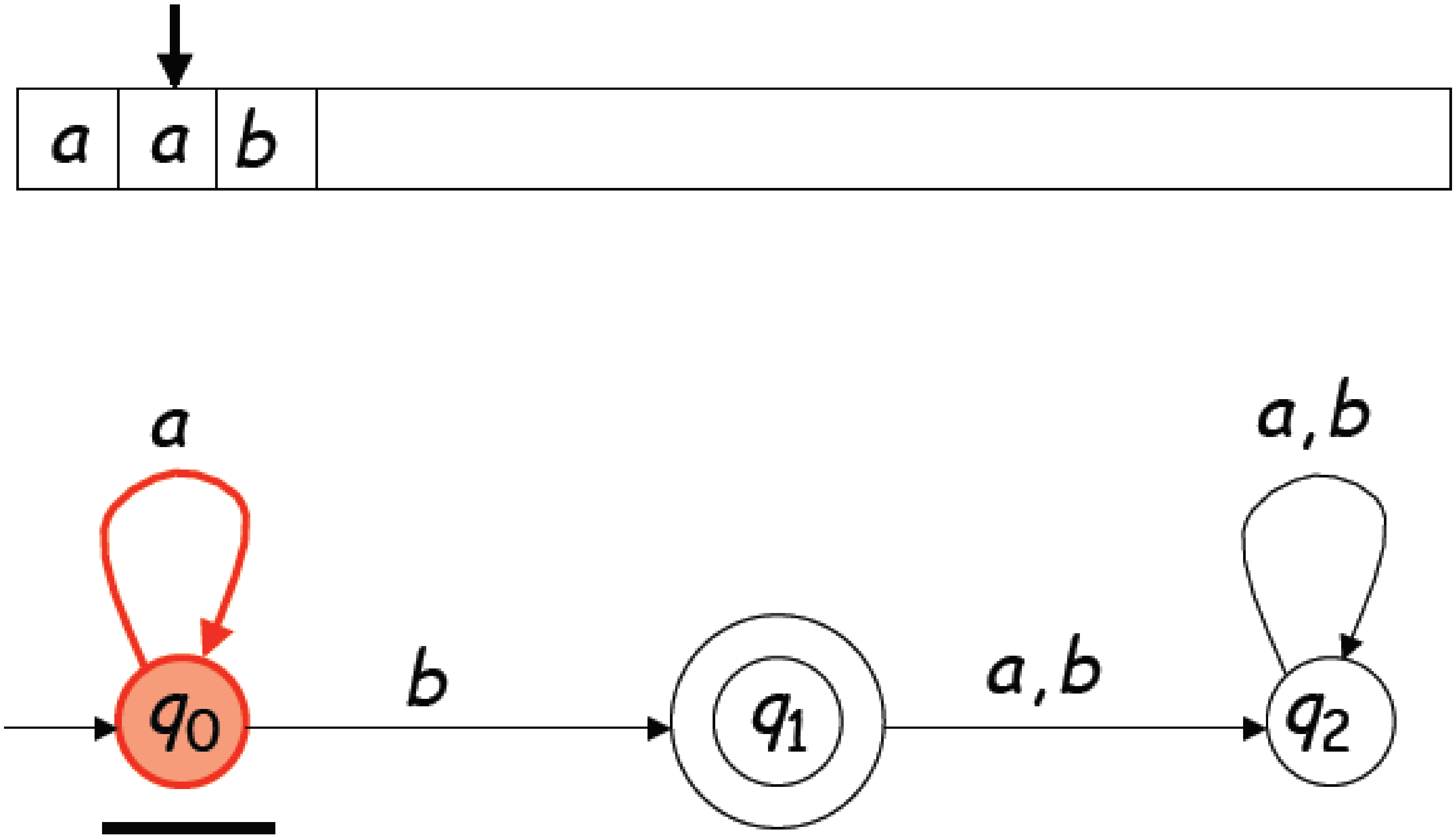
"reject"

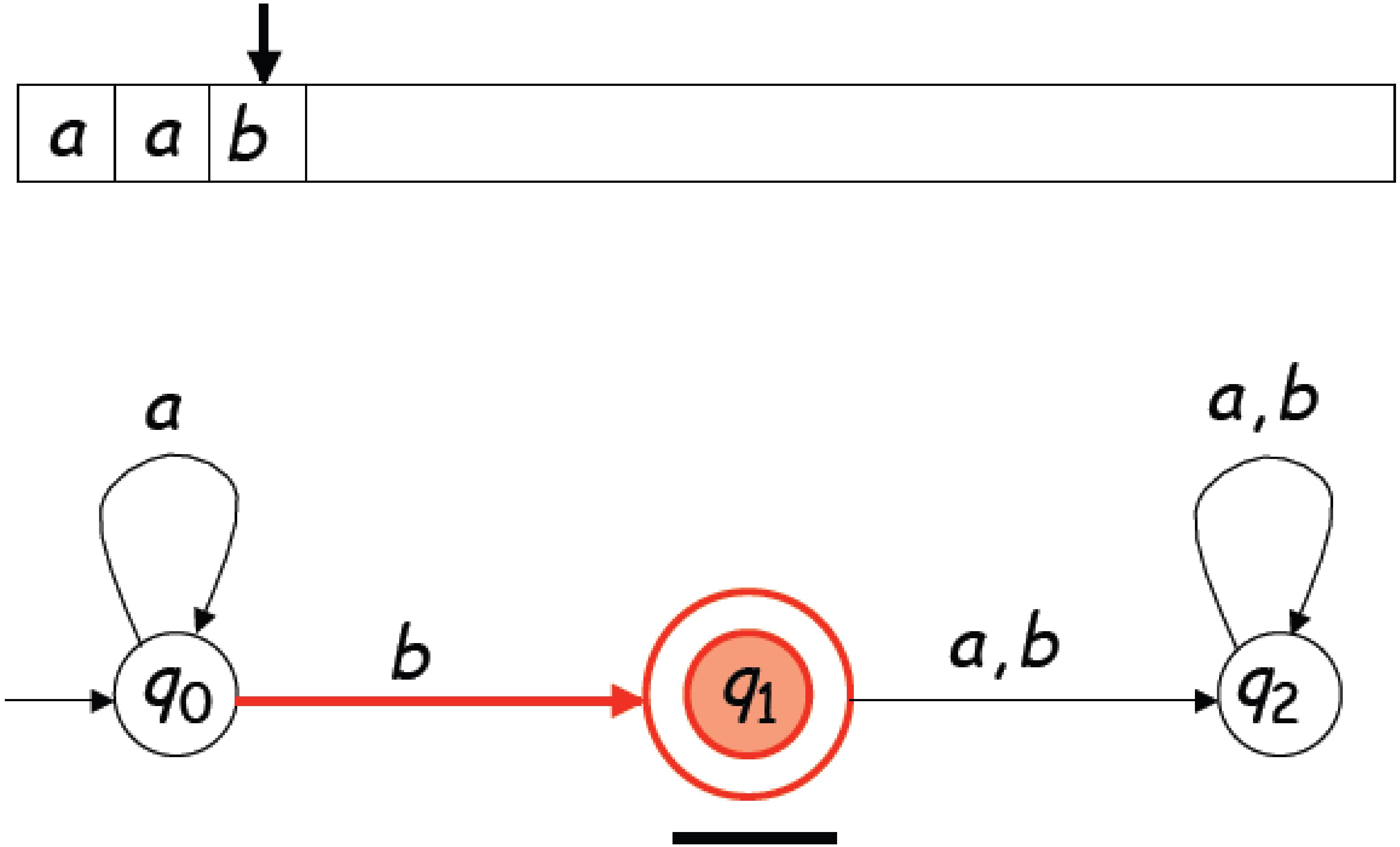
Would it be possible to accept the empty string?

Another Example

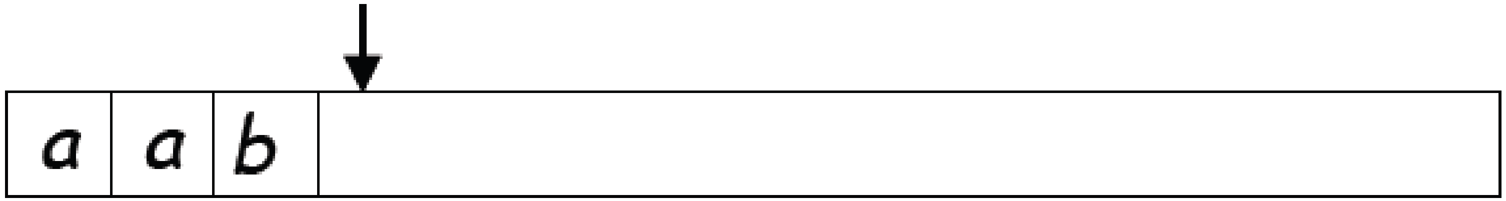



a
a
b
a
a,b
b
a,b
q_0
q_1
q_2

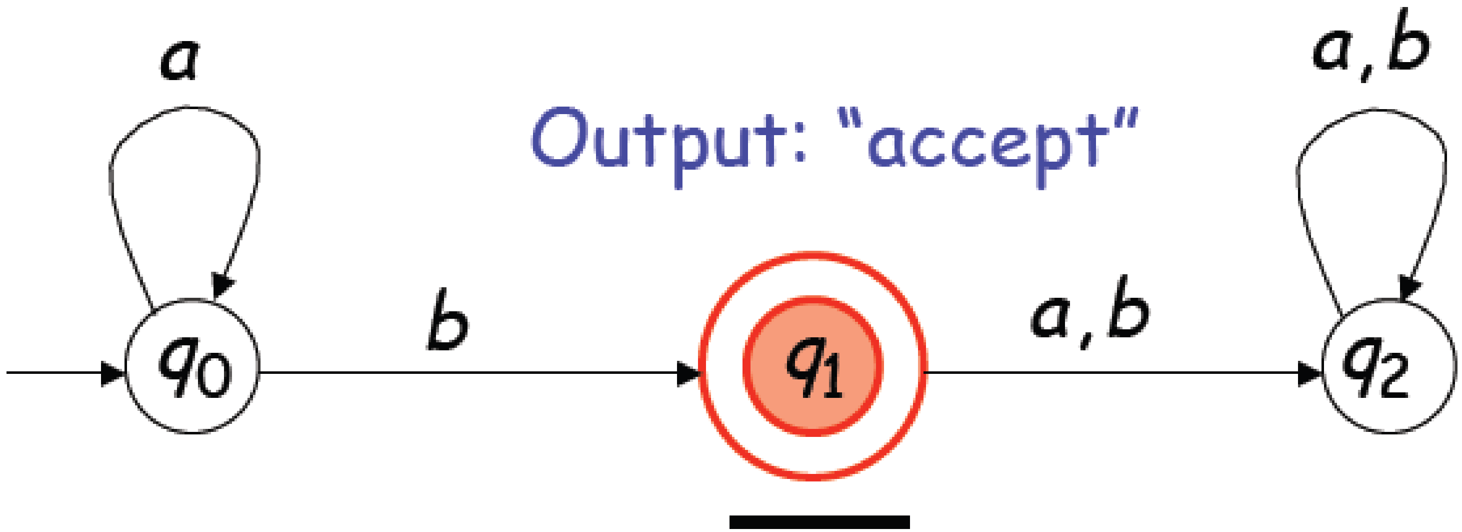

a
a
b
a
b
q0
q1
a, b
a, b
q2


a
a
b
a
a,b
b
a,b
q_0
q_1
q_2

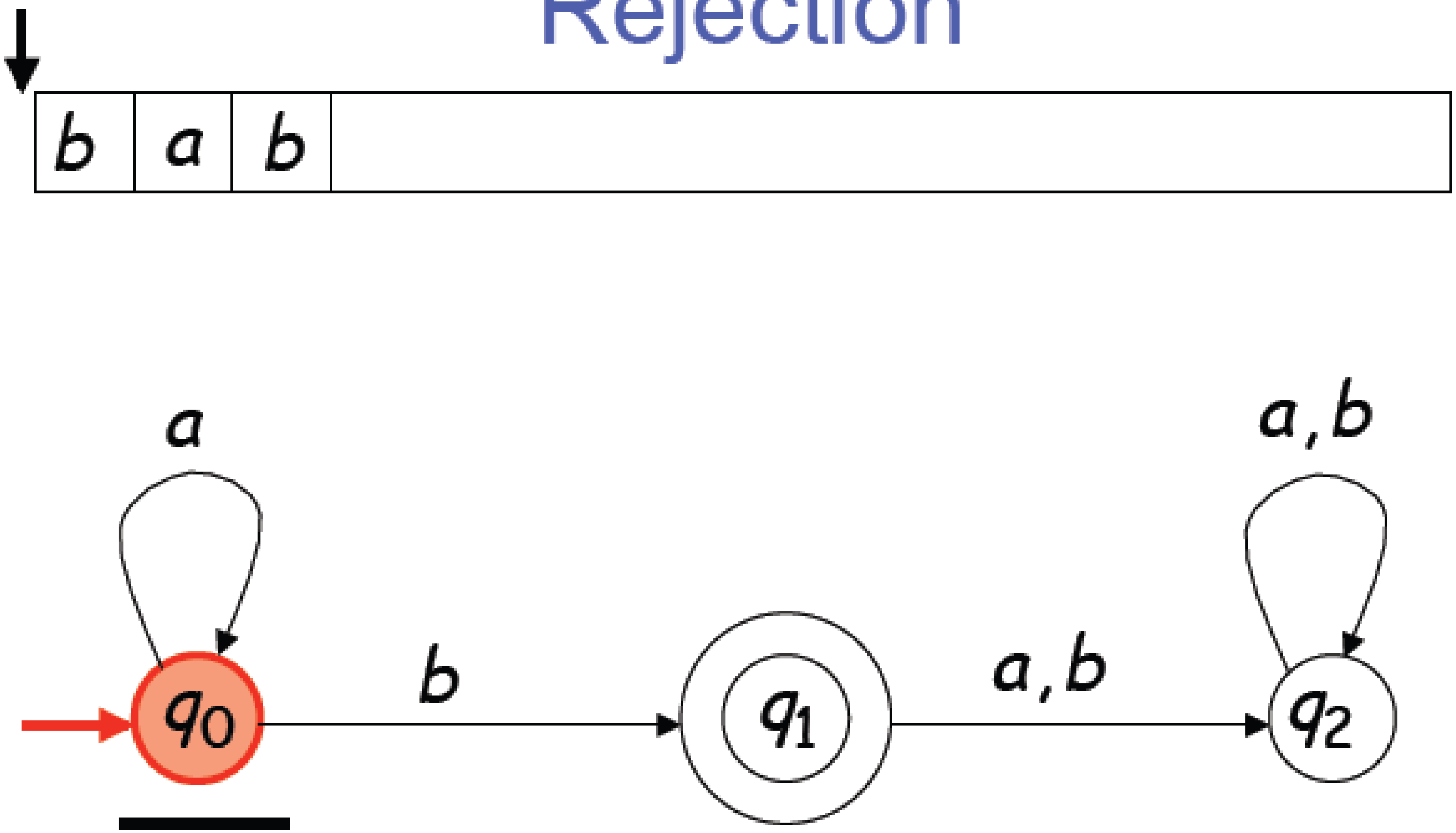
Input finished

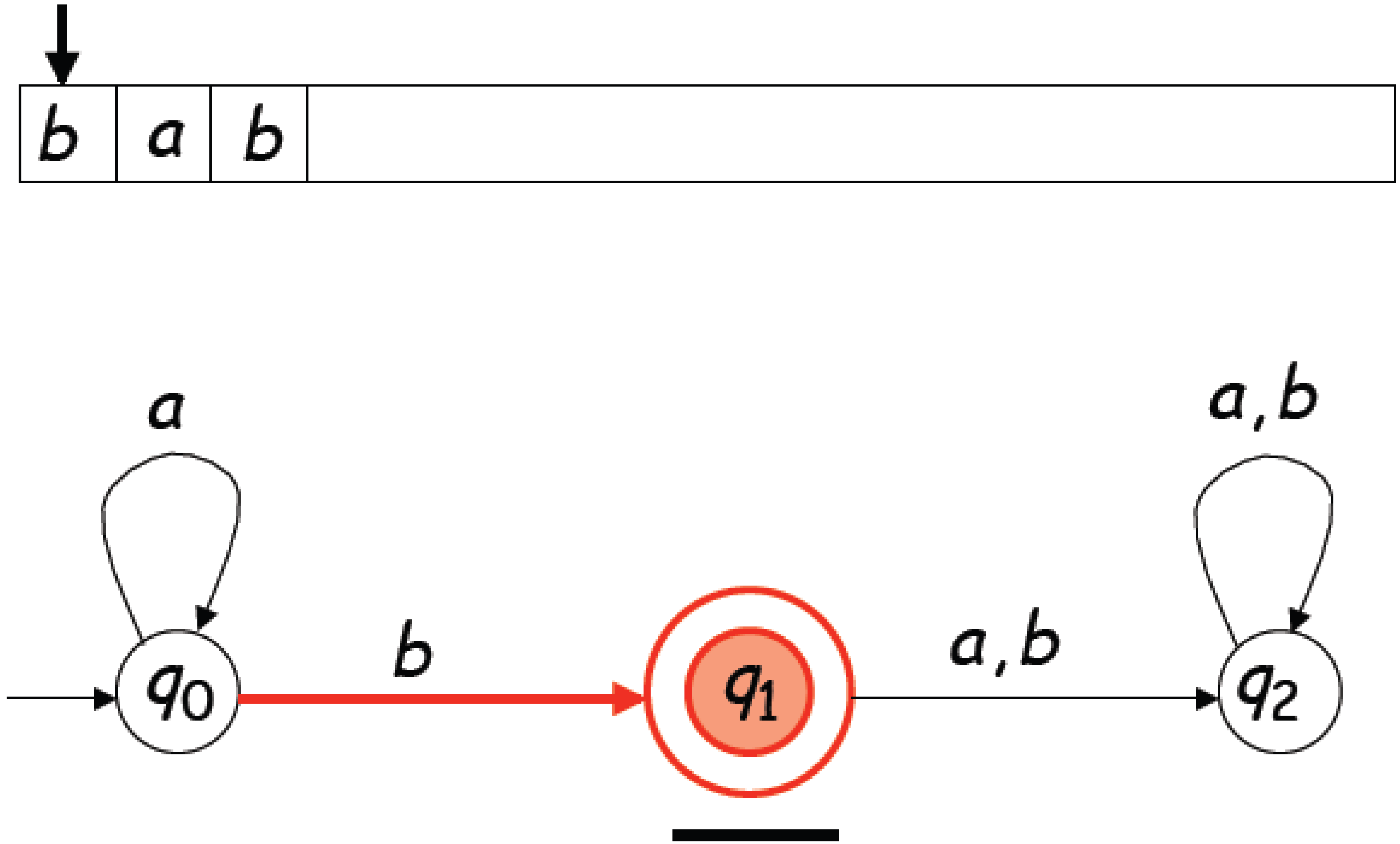


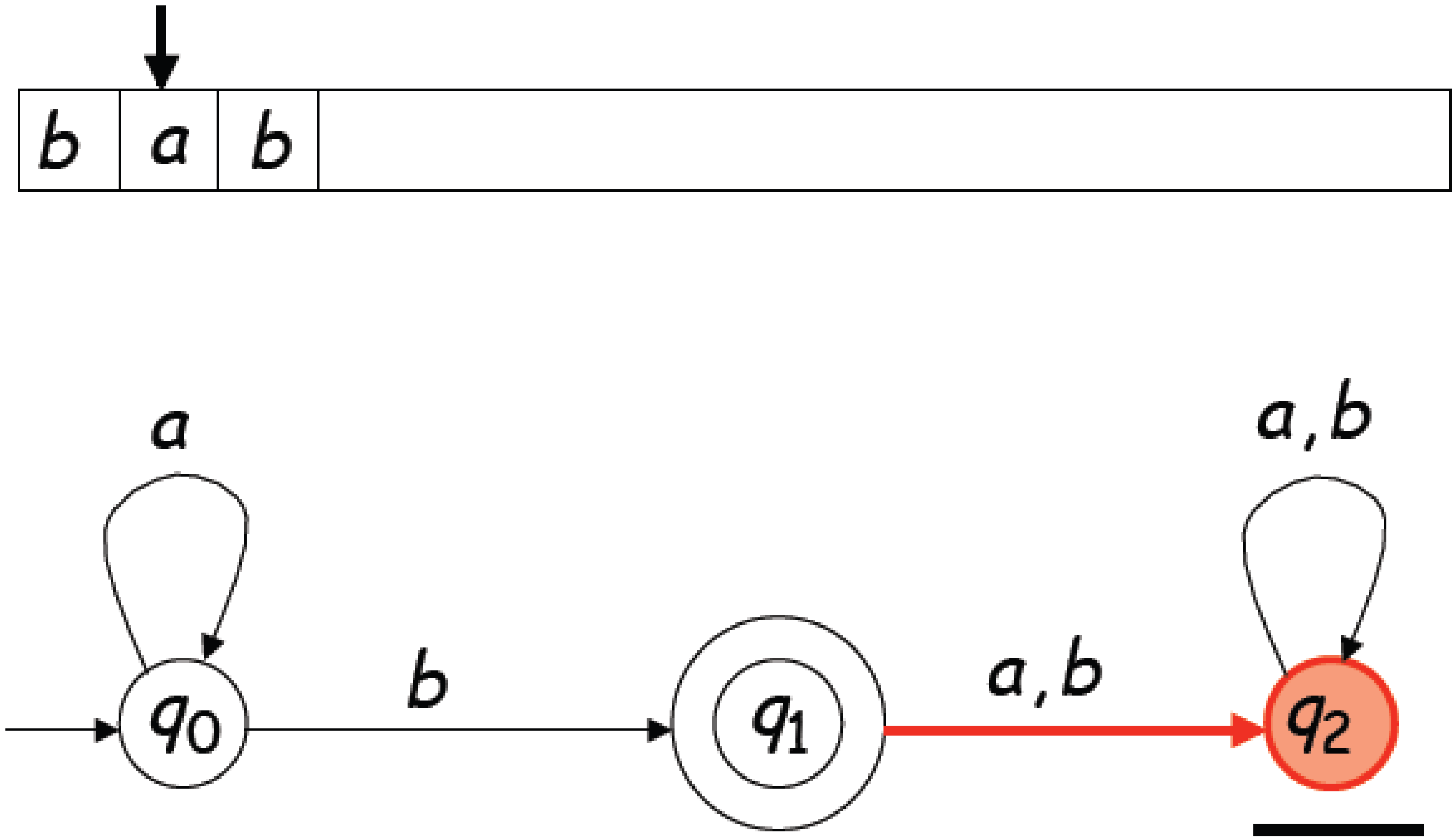
Output: "accept"

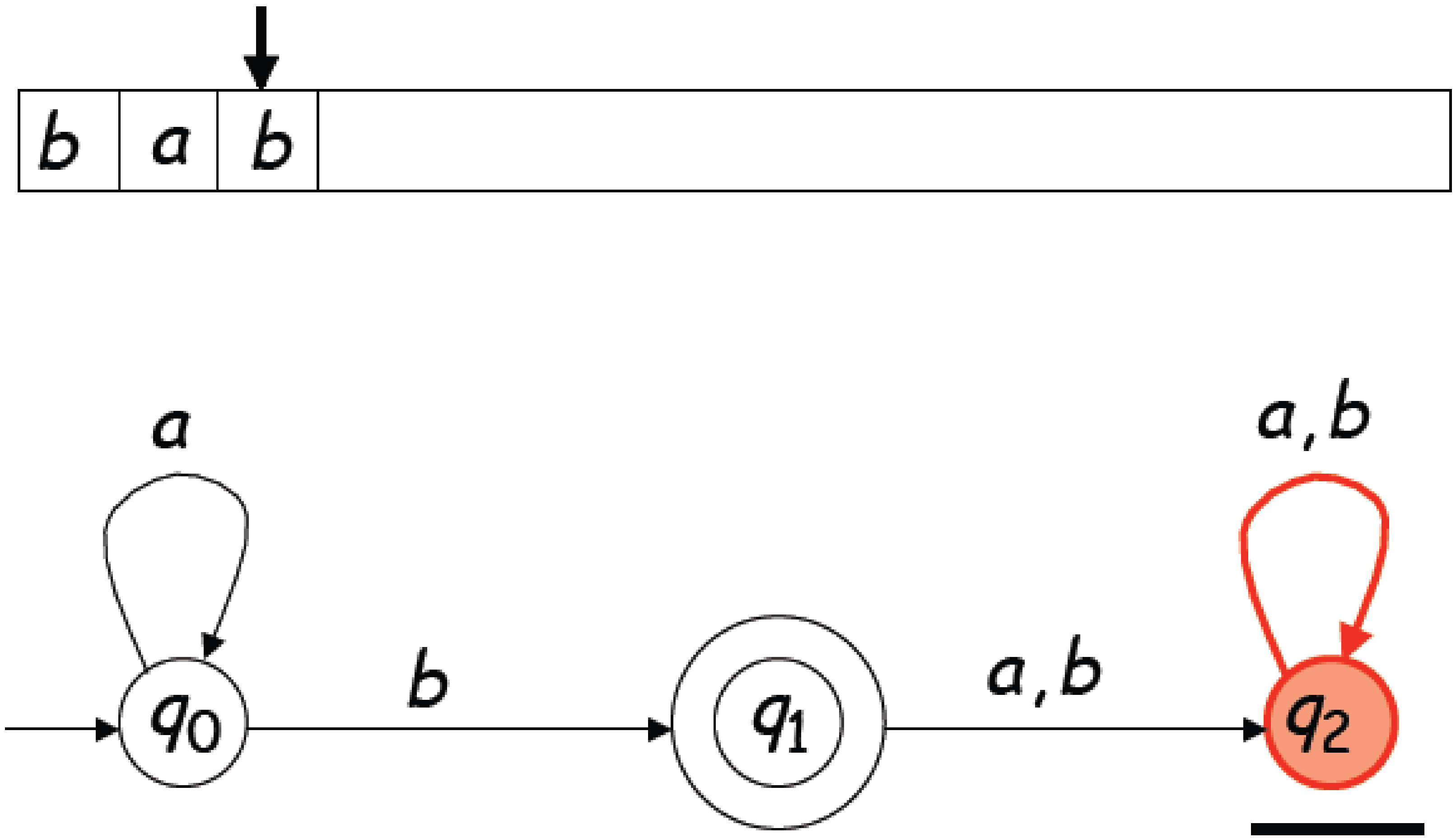


Rejection

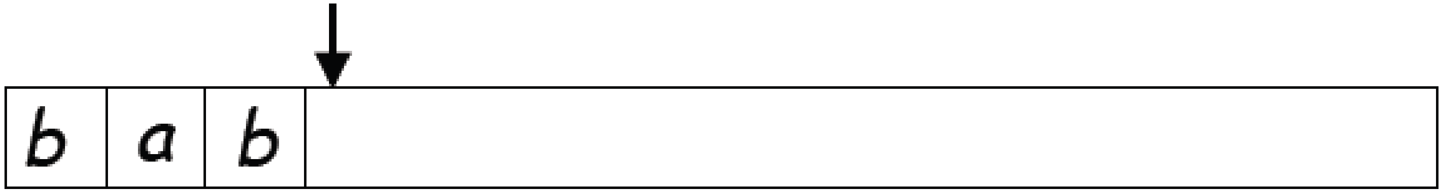



b a b
a
a,b
b
a,b
q_0
q_1
q_2

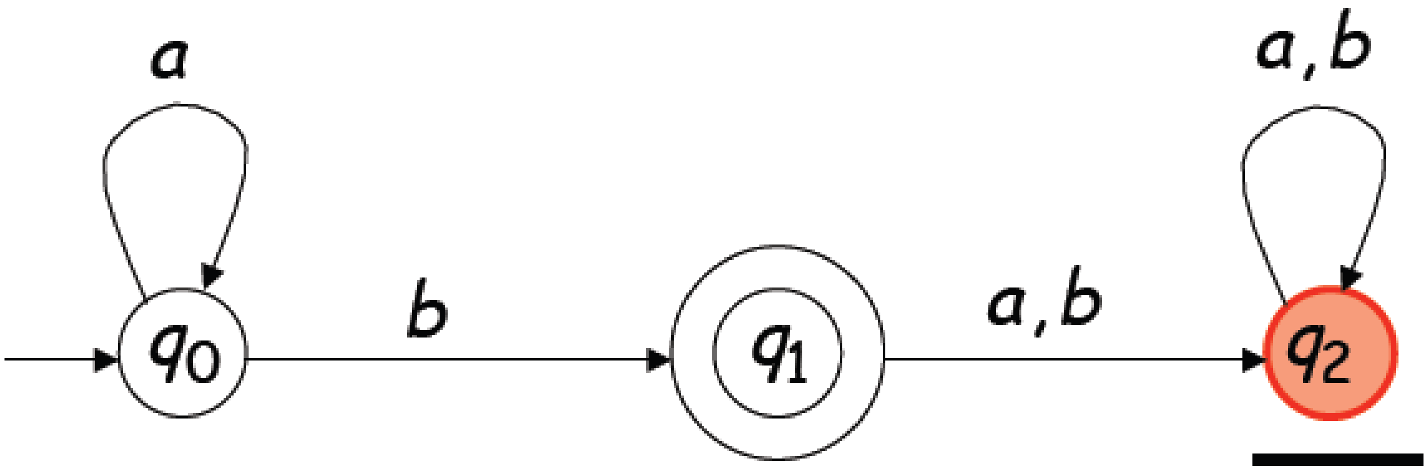

b
a
b
a
a,b
b
q_0
q_1
a,b
q_2


b
a
b
a
a,b
b
q0
q1
a,b
q2

Input finished



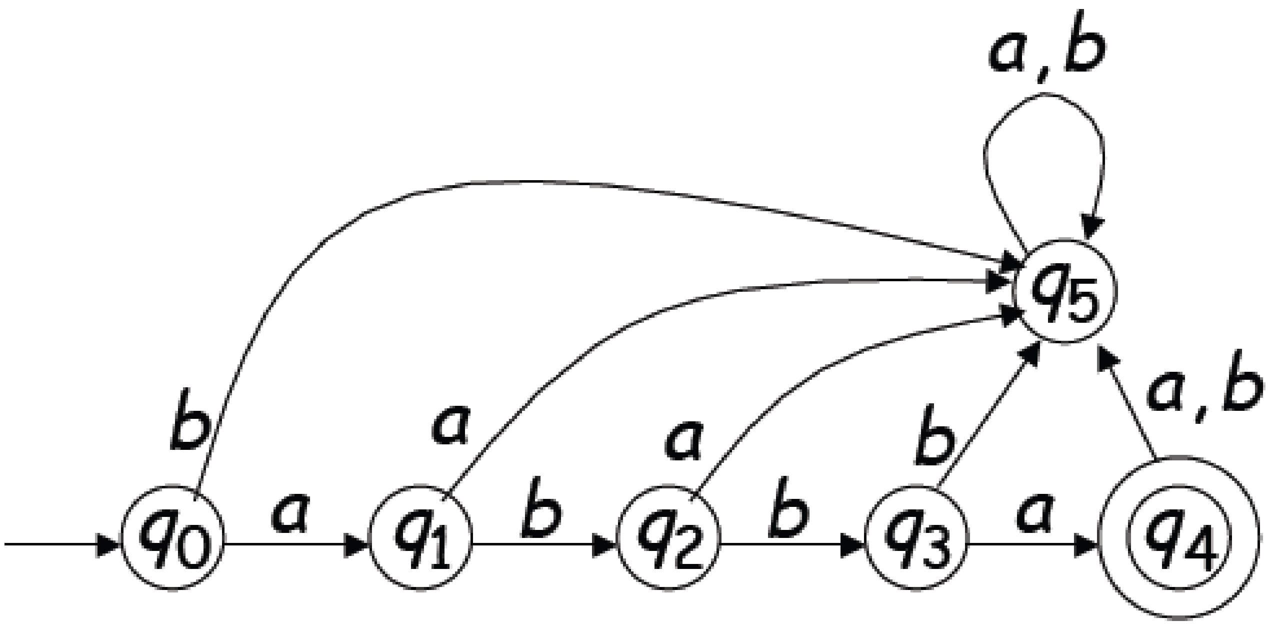
Which strings are accepted?



Output: “reject”

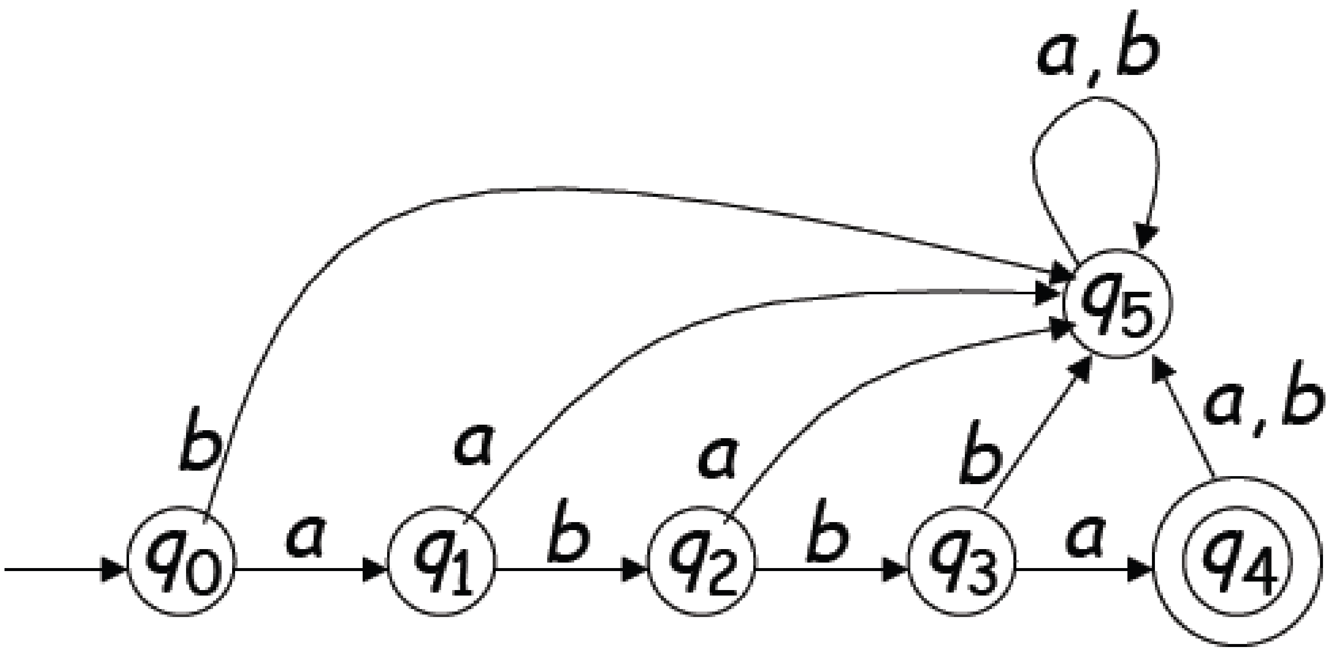
Input Alphabet Σ

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

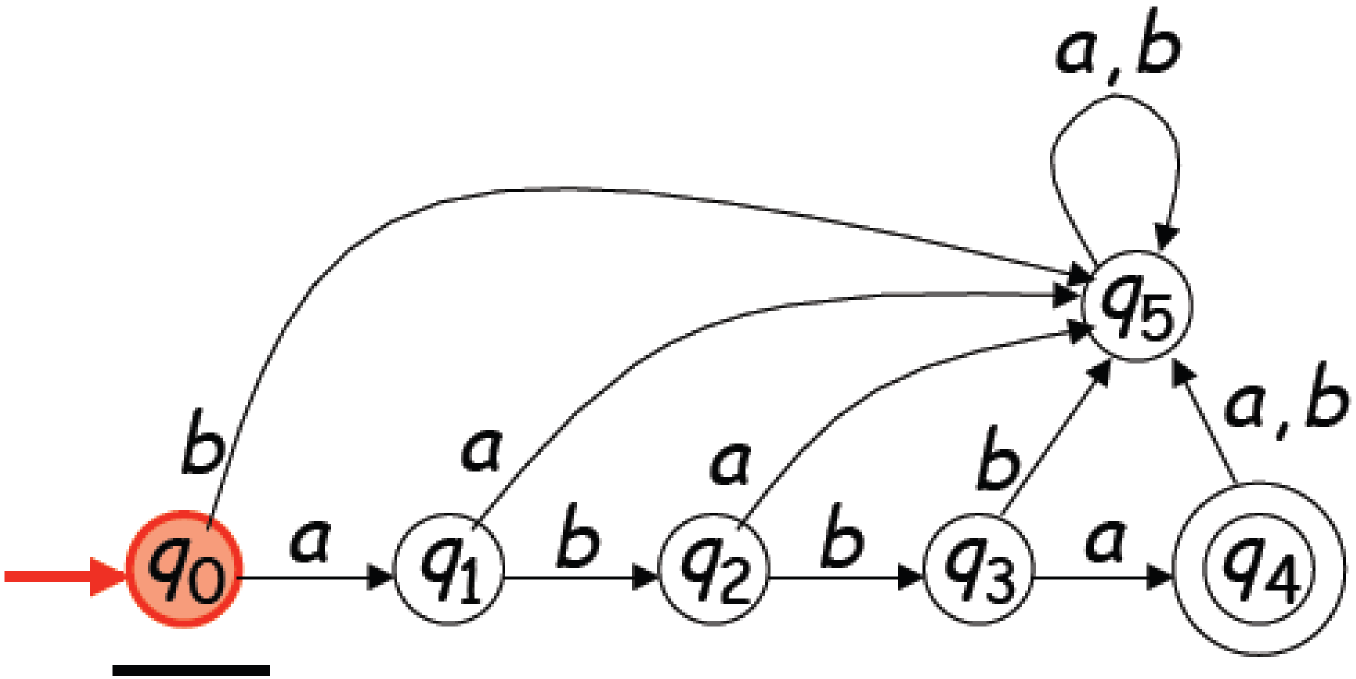


Set of States Q

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$

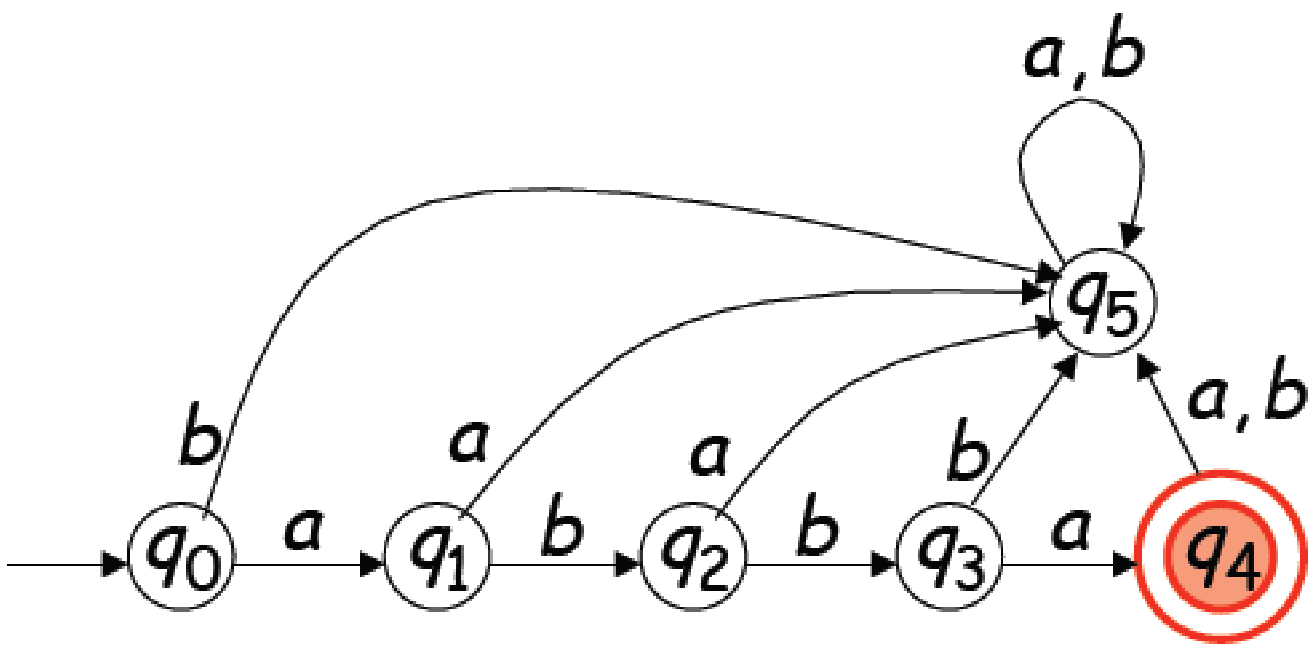


Initial State q_0



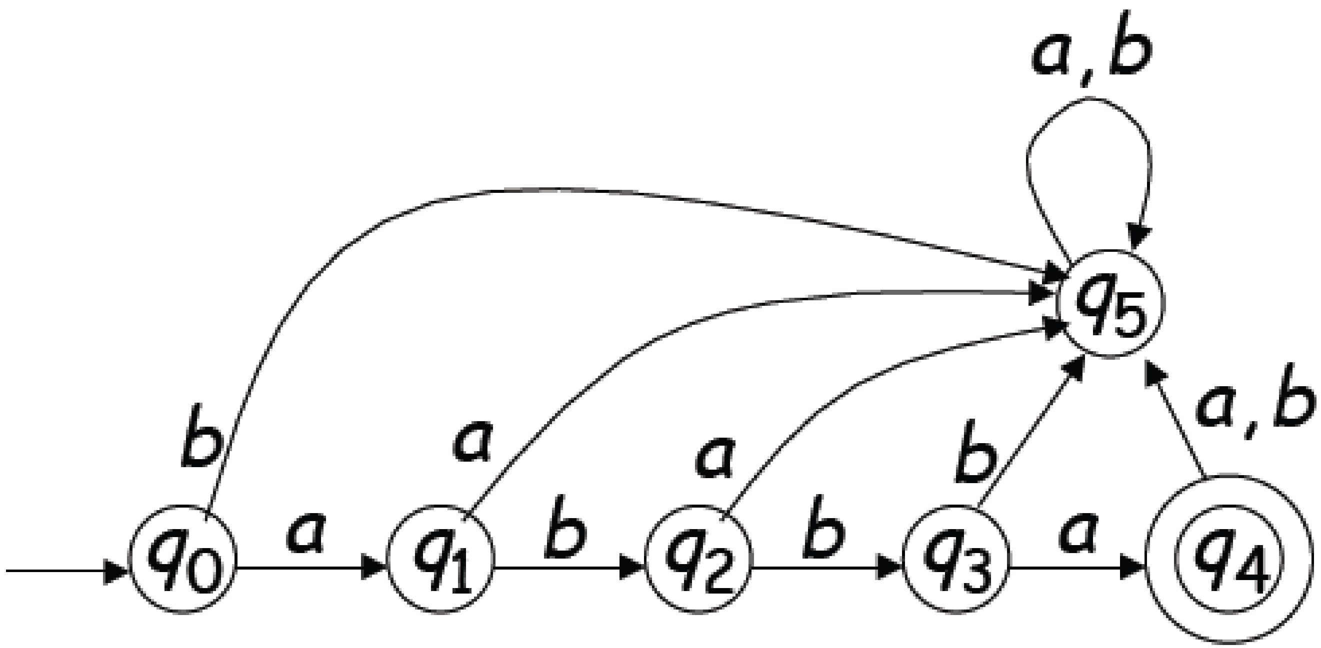
Set of Final States F

$F = \{q_4\}$



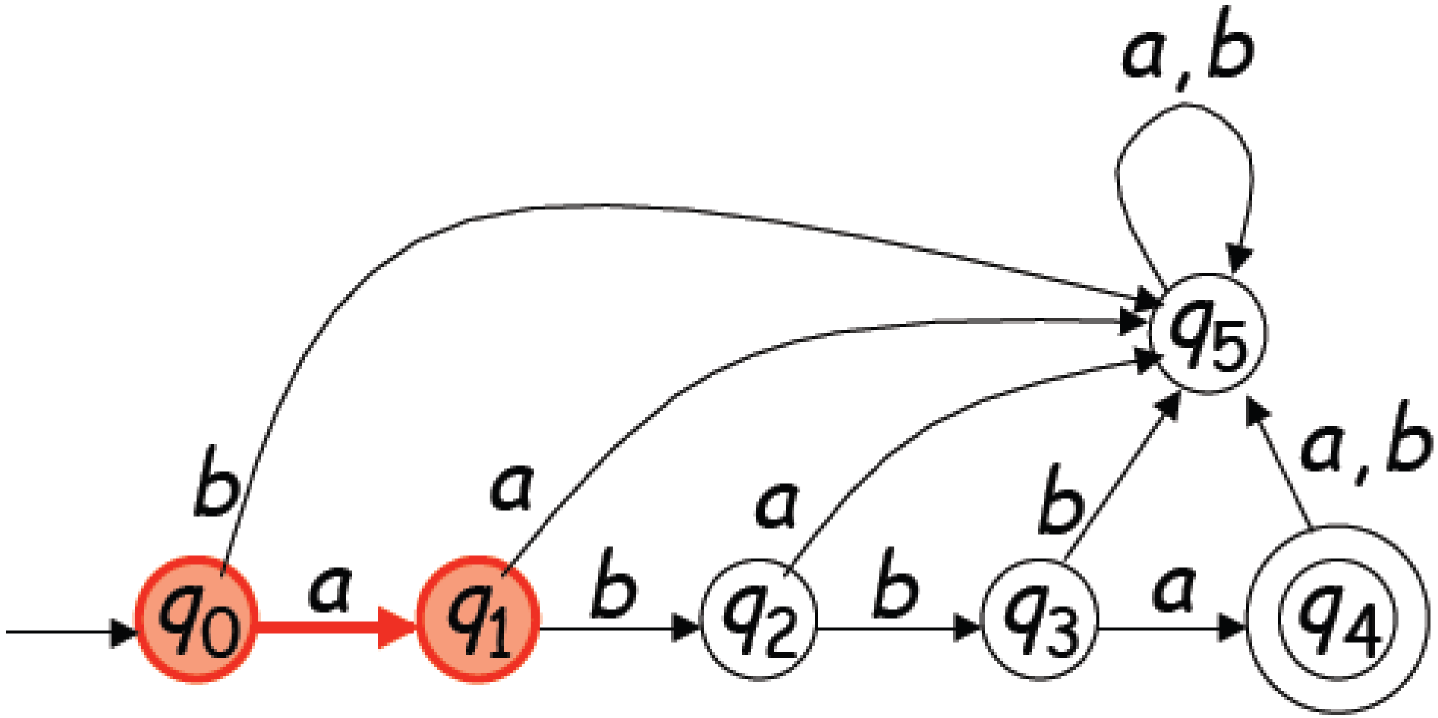
Transition Function δ

$$\delta : Q \times \Sigma \to Q$$

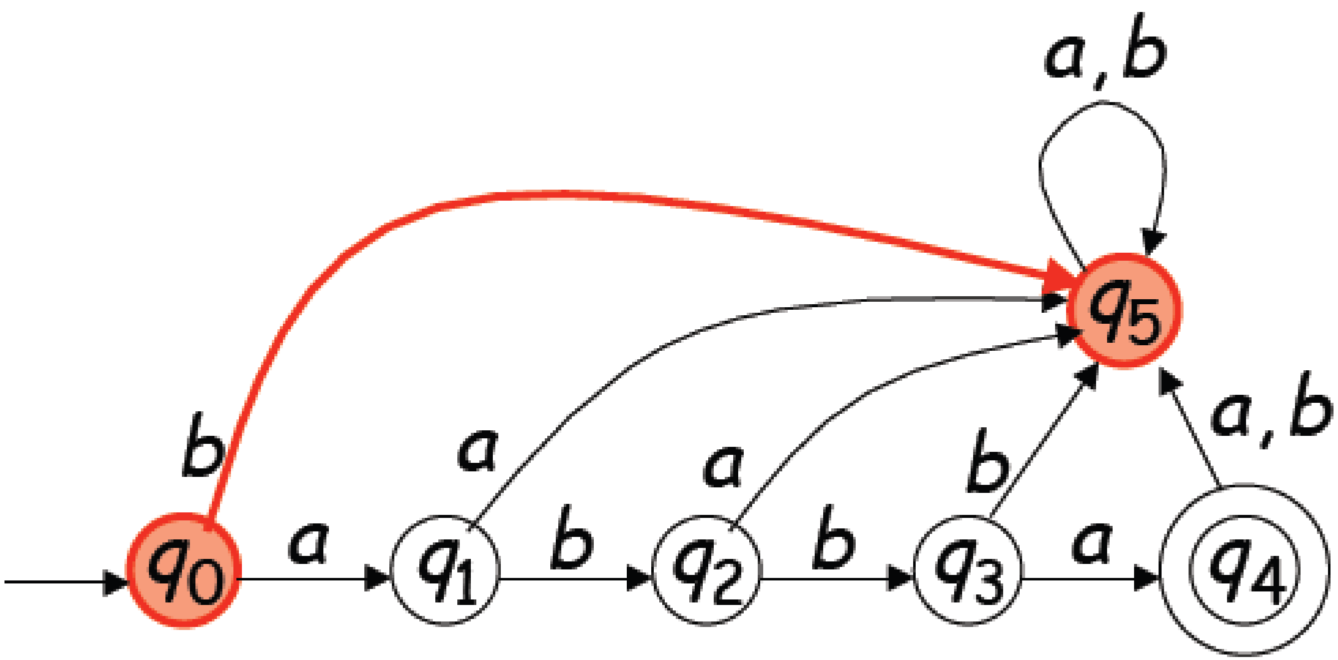


$$\delta(q_0, a) = q_1$$

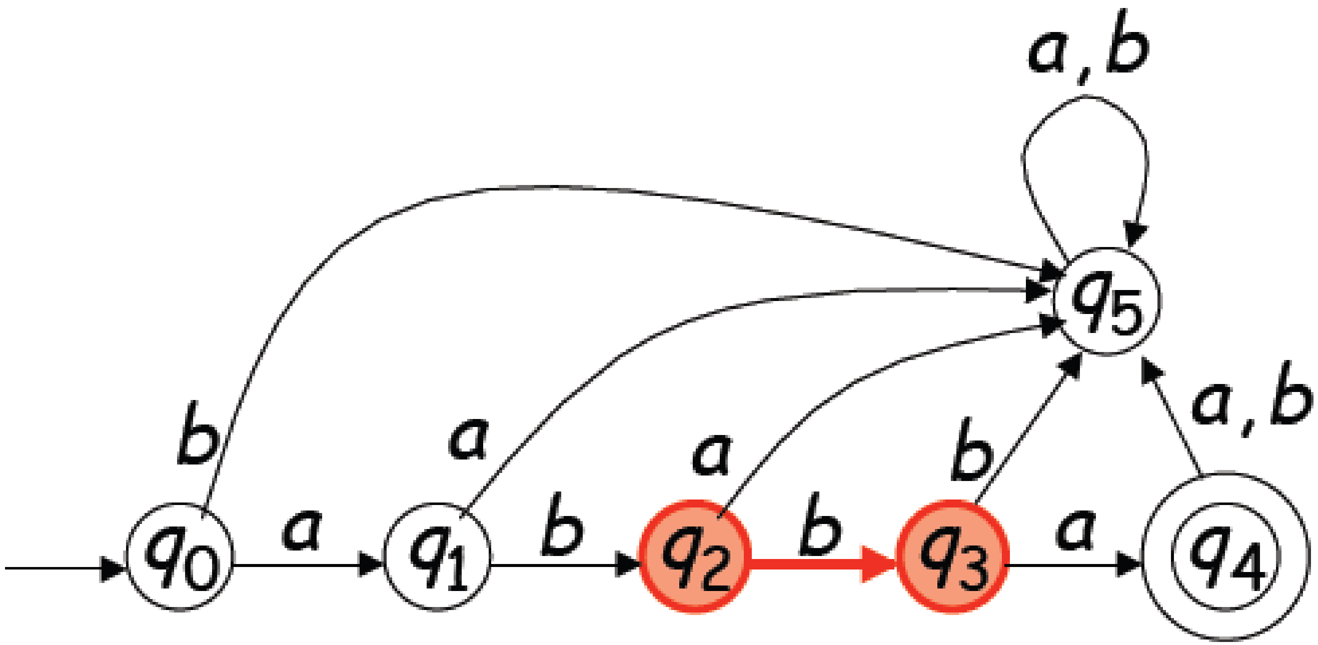
DFA q_0 durumundadır ve o anki girdi sembolü a'dır. Bir sonraki adımda q_1 durumuna gidecektir.



$$\delta(q_0, b) = q_5$$



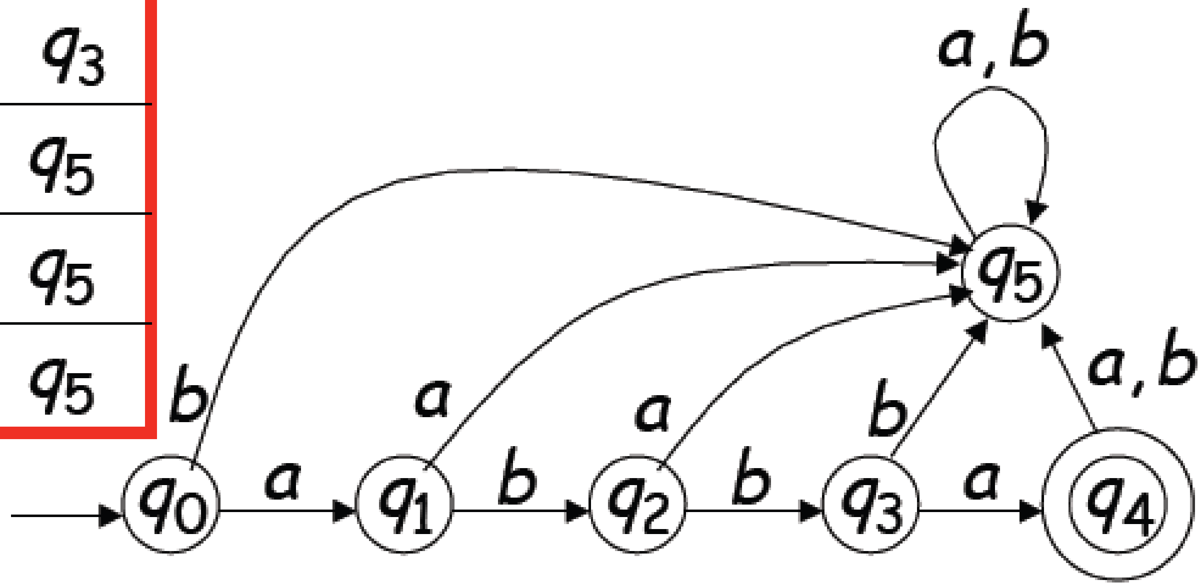
$$\delta(q_2, b) = q_3$$



Transition Function δ

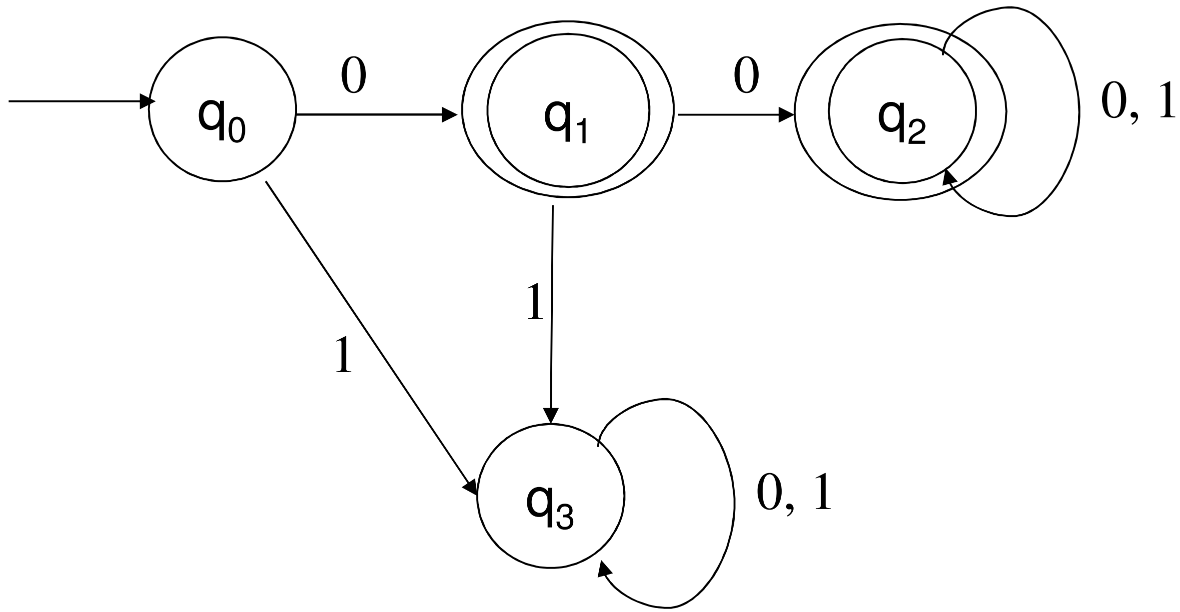
δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5

→ Durum geçiş tablosu



ÖRNEK: $M = (Q, \Sigma, \delta, q_0, F)$ DFA'sı için $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma=\{0, 1\}$, q_0 başlangıç durumu, $F = \{q_1, q_2\}$, durum geçiş tablosu:

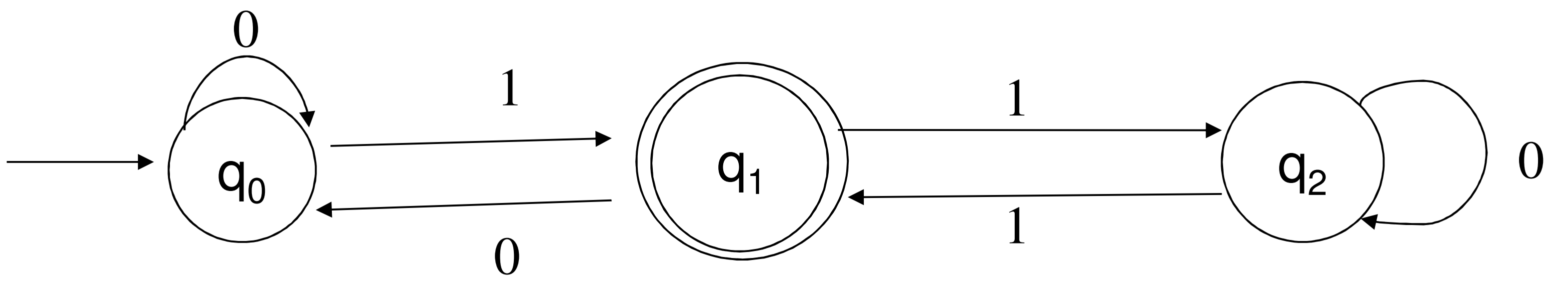
	0	1
q_0	q_1	q_3
$*q_1$	q_2	q_3
$*q_2$	q_2	q_2
q_3	q_3	q_3



Bu otomat tarafından kabul edilen dizgiler:

{0, 00, 000, 001, 0010, 0011, 0001,.....}

ÖRNEK: $M = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$



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

$\delta(q_0, 1) = q_1$ $\delta(q_1, 1) = q_2$ $\delta(q_2, 1) = q_1$

	0	1
q_0	q_0	q_1
$*q_1$	q_0	q_2
q_2	q_2	q_1

Bu DFA tarafından kabul edilen bazı dizgiler:

01, 101, 0111, 11001

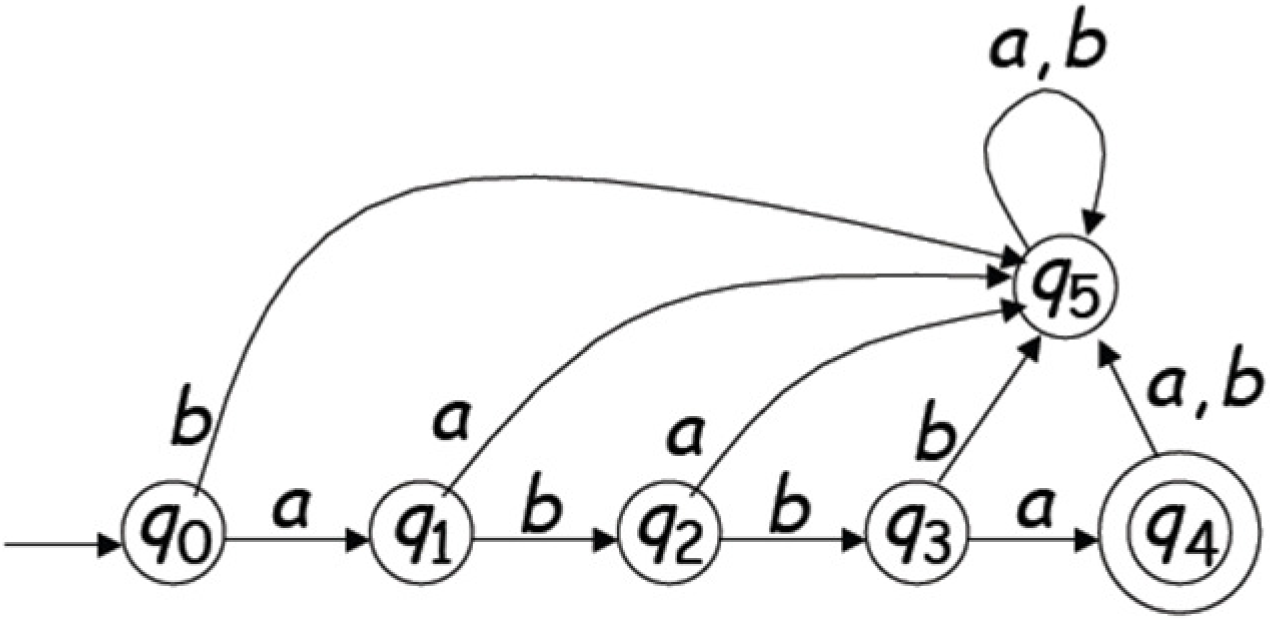
Kabul edilmeyen bazı dizgiler:

11, 00, 100, 1100

Genişletilmiş Geçiş Fonksiyonu
Extended Transition Function δ^*

$$\delta^* : Q \times \Sigma^* \to Q$$

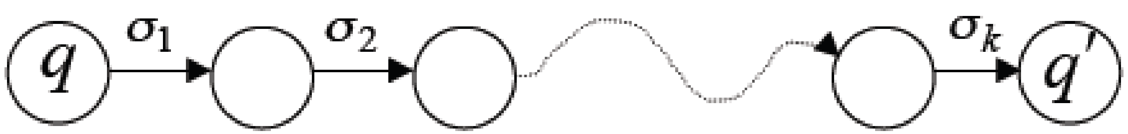
δ* fonksiyonunun ikinci argümanı tek bir sembol değil, bir dizgidir.



Observation: There is a walk from q to q' with label w

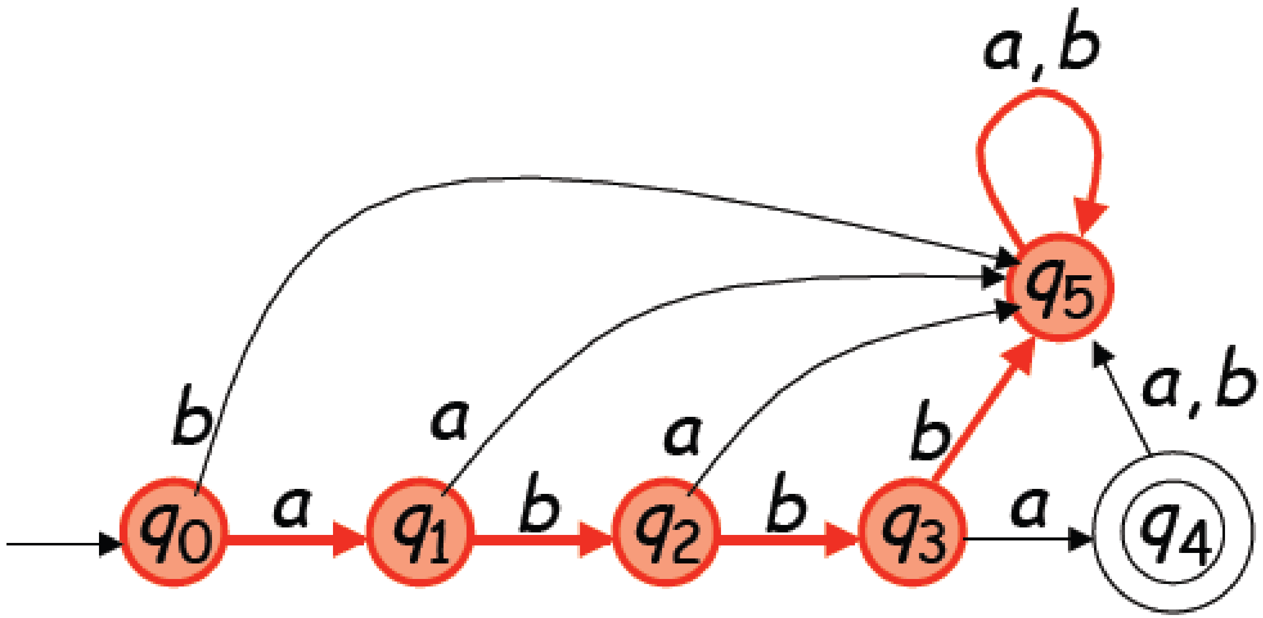
$$\delta^*(q,w) = q'$$

$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



Example: There is a walk from q_0 to q_5 with label $abbbaa$

$$\delta^*(q_0, abbbaa) = q_5$$



Recursive Definition

$$\delta^*(q,\lambda) = q$$

$$\delta^*(q,w\sigma) = \delta(\delta^*(q,w),\sigma)$$

$$\left.\begin{array}{l}\delta^*(q,w\sigma) = q' \\ \\ \delta(q_1,\sigma) = q'\end{array}\right\} \Rightarrow \left.\begin{array}{l}\delta^*(q,w\sigma) = \delta(q_1,\sigma) \\ \\ \delta^*(q,w) = q_1\end{array}\right\} \Rightarrow \delta^*(q,w\sigma) = \delta(\delta^*(q,w),\sigma)$$

$$\delta^*(q_0, ab) =$$

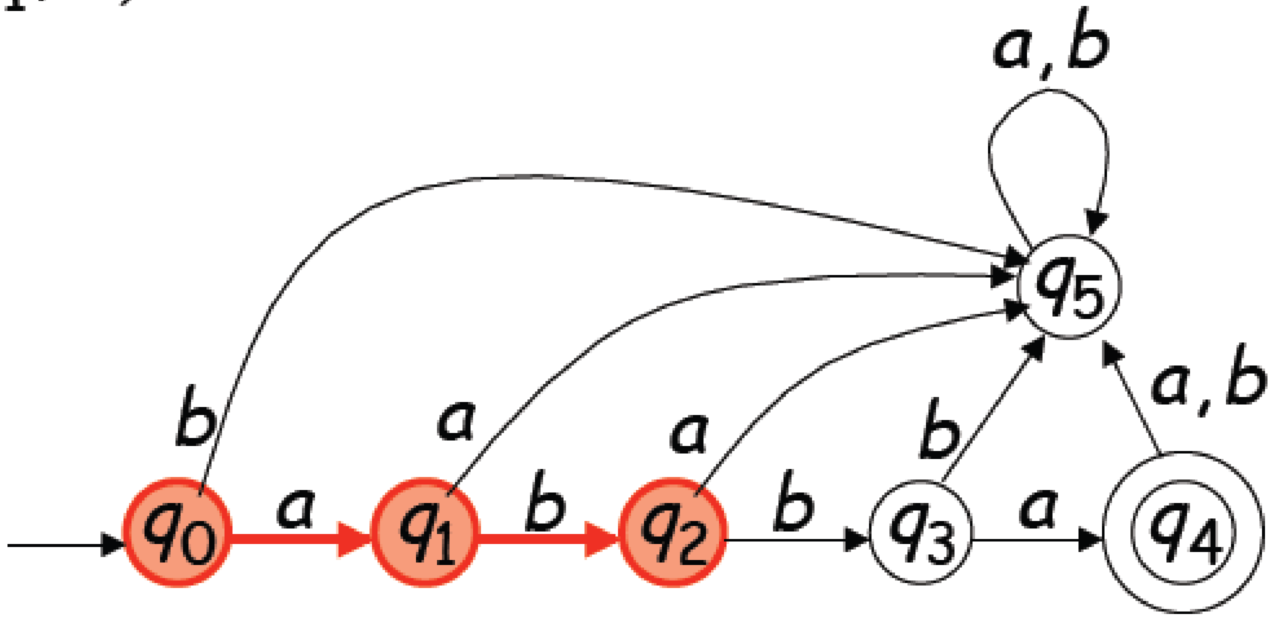
$$\delta(\delta^*(q_0, a), b) =$$

$$\delta(\delta(\delta^*(q_0, \lambda), a), b) =$$

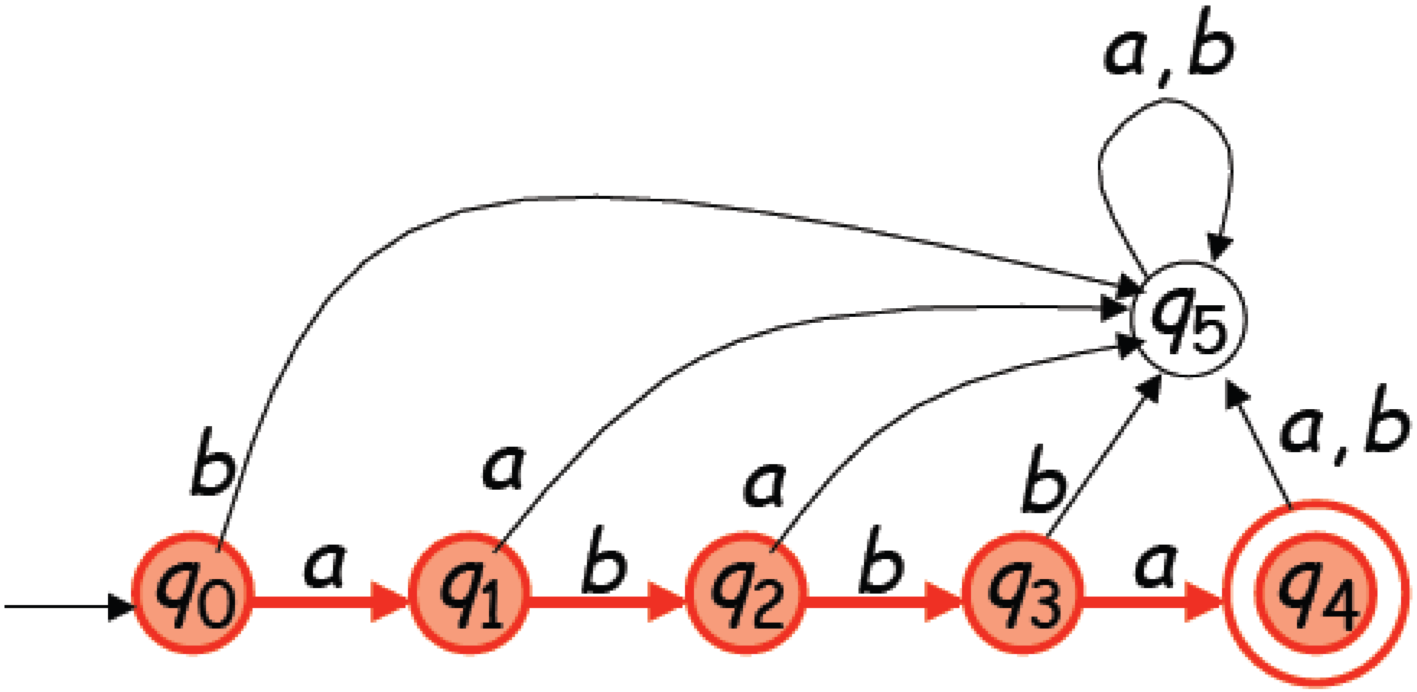
$$\delta(\delta(q_0, a), b) =$$

$$\delta(q_1, b) =$$

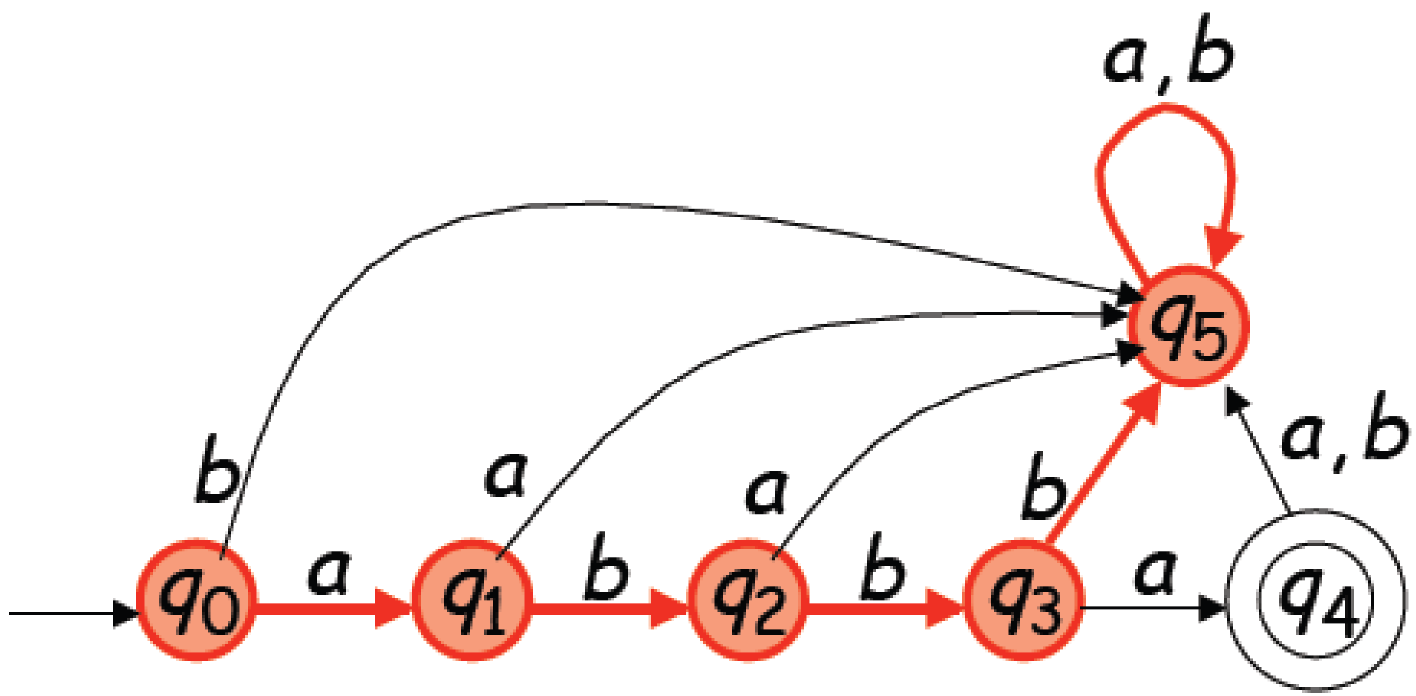
$$q_2$$



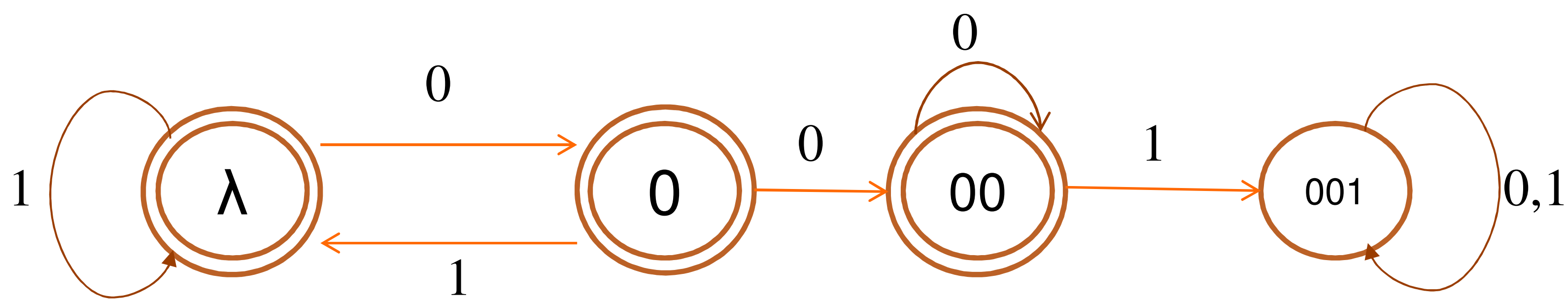
$$\delta^*(q_0, abba) = q_4$$



$$\delta^*(q_0, abbbaa) = q_5$$



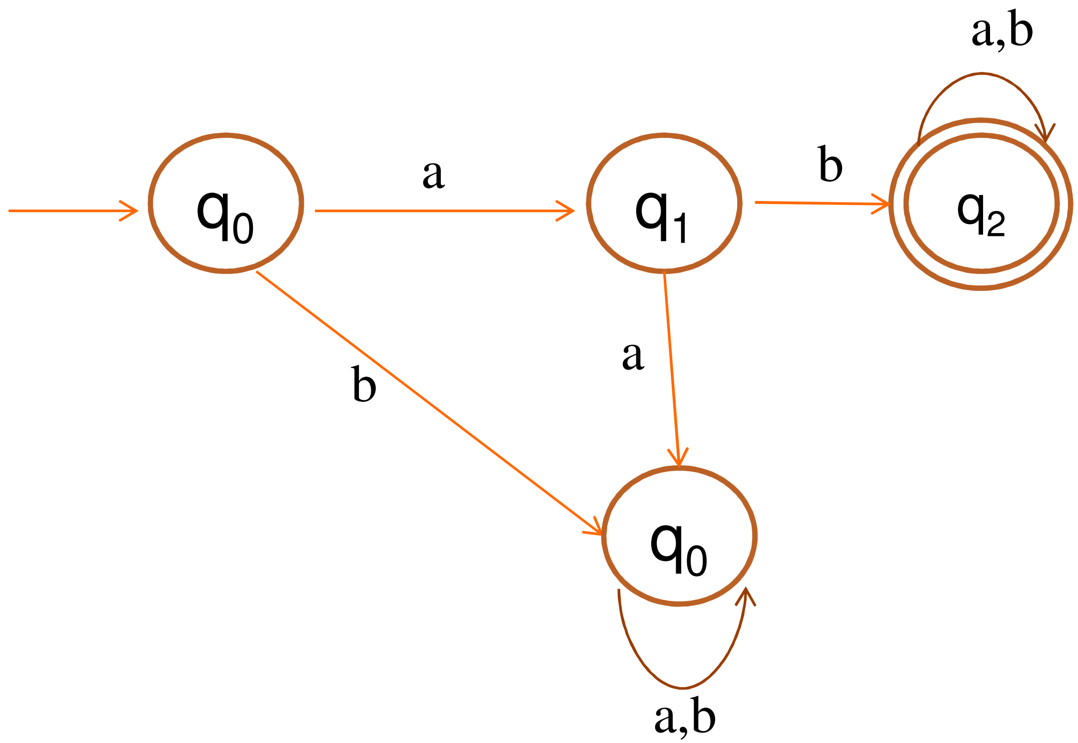
Örnek: İçerisinde 001 dizgisi içeren dizgiler dışında {0, 1} üzerinde tanımlanmış tüm dizgileri kabul eden bir DFA tasarlayınız.



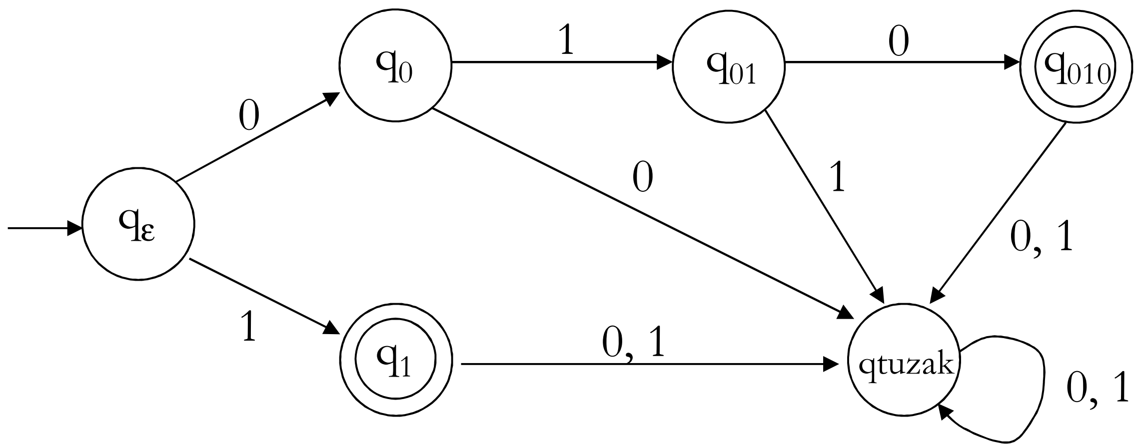
λ dizgisi bu DFA tarafından kabul edilir. Bu nedenle başlangıç durumu aynı zamanda bir kabul durumudur.

001 dizgisi bu DFA tarafından reddedilecektir. Bu nedenle başlangıç durumundan tuzak durumuna giden bir yol olmalıdır.

Örnek: $\Sigma=\{a,b\}$ üzerinde tanımlanmış ab ile başlayan tüm dizgilerin kümesini kabul eden gerekirci sonlu otomatı bulunuz.



- Örnek: Σ = {0, 1} üzerinde tanımlı olan *L* = {010, 1} dilini kabul eden bir DFA tasarlayınız.



DILLER VE DFA'LAR

$M = (Q, \Sigma, \delta, q_0, F)$ DFA'sı tarafından kabul edilen dil, M tarafından kabul edilen tüm dizgilerin kümesidir.

$$L(M) = \{w \mid w \in \Sigma^* : \quad \delta^*(q_0, w) \in F\}$$

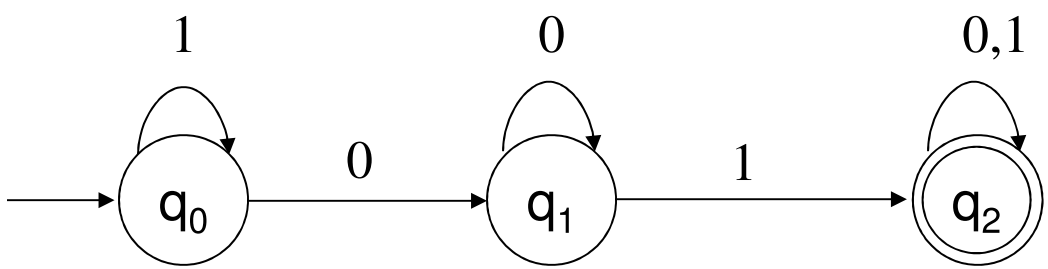
DFA tarafından kabul edilmeyen dil:

$$\overline{L(M)} = \{w \in \Sigma^* : \quad \delta^*(q_0, w) \notin F\}$$

Örnek: Herhangi bir yerinde 01 olan ve 0,1'lerden meydana gelen dizgileri kabul eden DFA'yı çiziniz.

Bu DFA için L dili:

$L = \{ x01y \mid x, y \in \Sigma^* \}$



Geçiş tablosu:

	0	1
q_0	q_1	q_0
q_1	q_1	q_2
$^*q_2$	q_2	q_2

Örnek: L={w| w çift sayıda 0 ve 1'e sahip} Bu dili kabul eden bir DFA tasarlayın.

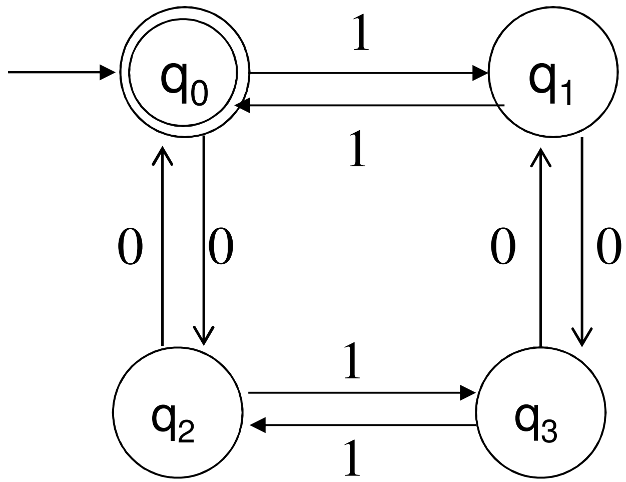
Bu DFA'da 4 durum olabilir:

q_0 : 0 ve 1'lerin sayısı çift

q_1 : 0'ların sayısı çift, 1'lerin sayısı tek

q_2 : 1'lerin sayısı çift, 0'ların sayısı tek

q_3 : 0 ve 1'lerin sayısı tek



Örneğin 110101 dizgisi çift sayıda 1 ve 0 içerdiği için bu dil tarafından kabul edilir. Yani $\delta^*(q_0,110101)=q_0$ olmalıdır.

$\delta^*(q_0, \lambda\} = q_0$

$\delta^*(q_0, 1\} = \delta(\ \delta^*(q_0, \lambda\}, 1\} = \delta(q_0, 1\} = q_1$

$\delta^*(q_0, 11\} = \delta(\ \delta^*(q_0, 1\}, 1\} = \delta(q_1, 1\} = q_0$

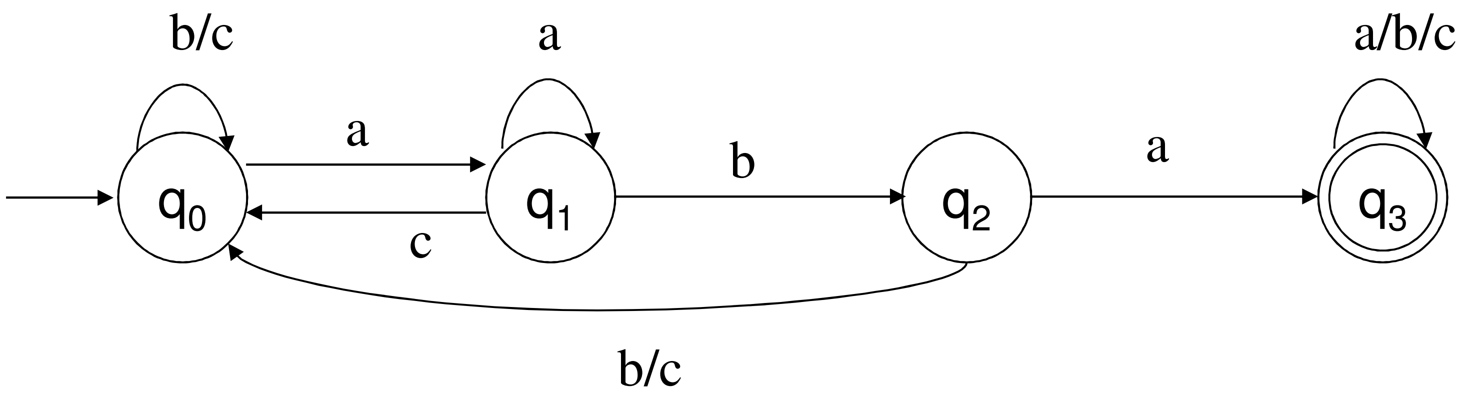
$\delta^*(q_0, 110\} = \delta(\ \delta^*(q_0, 11\}, 0\} = \delta(q_0, 0\} = q_2$

$\delta^*(q_0, 1101\} = \delta(\ \delta^*(q_0, 110\}, 1\} = \delta(q_2, 1\} = q_3$

$\delta^*(q_0, 11010\} = \delta(\ \delta^*(q_0, 1101\}, 0\} = \delta(q_3, 0\} = q_1$

$\delta^*(q_0, 110101\} = \delta(\ \delta^*(q_0, 11010\}, 1\} = \delta(q_1, 1\} = q_0$

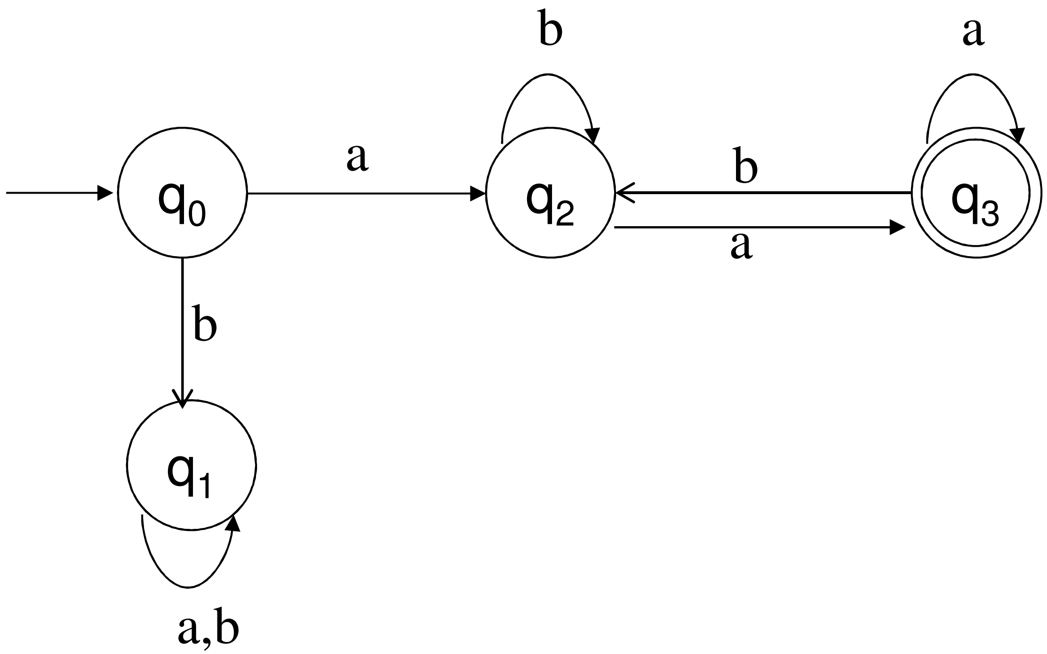
Örnek: L(M) = {x | x; a, b ve c'lerden oluşur ve *aba* alt stringini içerir} Bu dil için DFA'yı çiziniz.



DÜZGÜN DILLER (REGULAR LANGUAGES)

Bir L dilinin düzgün olması için L=L(M) eşitliğini sağlayan bir DFA olmalıdır. Yani L dilini kabul eden bir DFA olmalıdır.

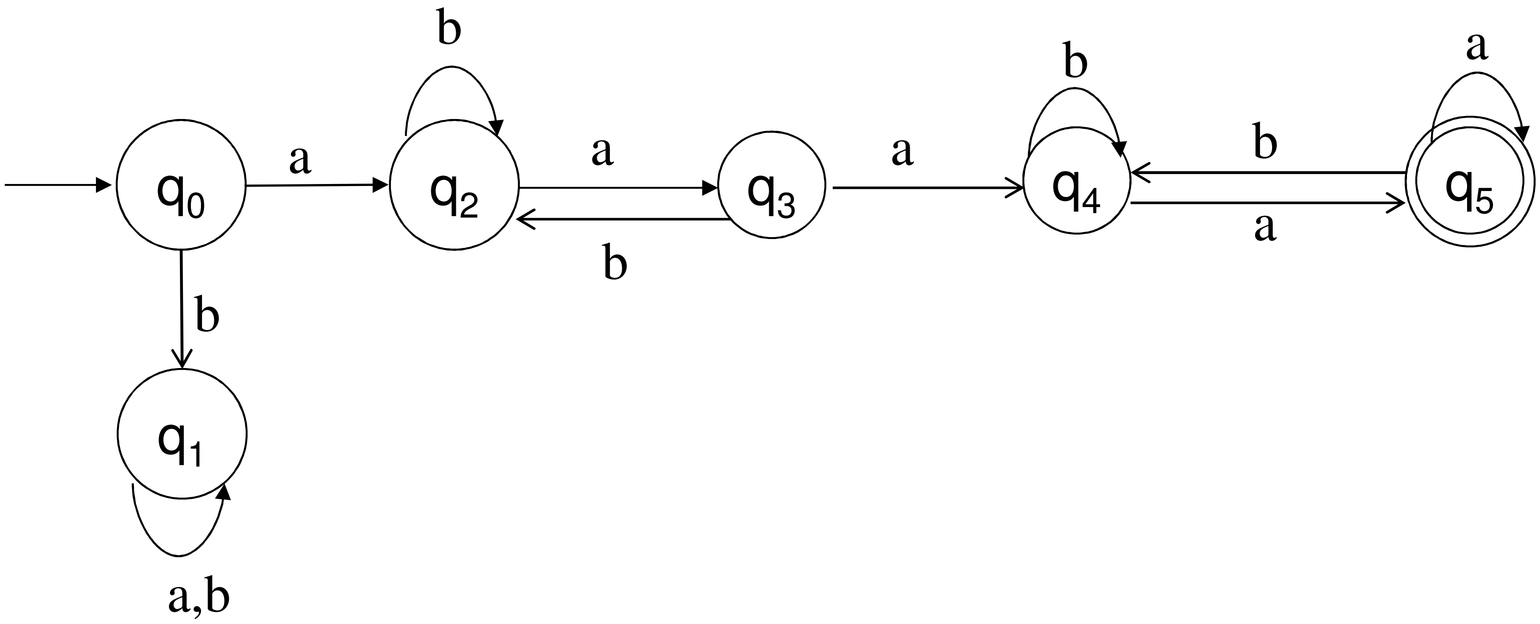
Örneğin; $L=\{awa:\ w \in \{a,b\}^*\}$ dilinin düzgün olduğunu gösteriniz.



aa……a
abb…..a
ababbb…a

L^2 dilinin de düzgün bir dil olduğunu gösterelim:

$L^2 = \{aw_1aaw_2a: w_1, w_2 \in \{a,b\}^* \}$



Sonuç olarak bir L dili düzgün ise L^2, L^3, … dilleri de düzgündür.

L1 U L2 = { w | w ϵ L1 or w ϵ L2 }

İki düzgün dilin birleşimi de yine düzgün bir dil oluşturur.

Düzgün olmayan diller de vardır:

$$L = \{a^n b^n : n \geq 0\}$$

Böyle bir dili kabul eden DFA çizmek mümkün değildir.

NONDETERMINISTIC FINITE AUTOMATA (NFA)
GEREKIRCI OLMAYAN SONLU OTOMAT

- Bir NFA $M = (Q, \Sigma, \delta, q_0, F)$ beşlisi ile tanımlanır.

Q: Mevcut durumların sonlu kümesi

Σ : Sembollerin sonlu kümesi

$q_0 \in Q$: Başlangıç durumu

$F \subseteq Q$: Kabul durumları kümesi

DFA'dan farkı geçiş fonksiyonudur:

NFA'da $\delta: Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$(Q'nun altkümeleri)

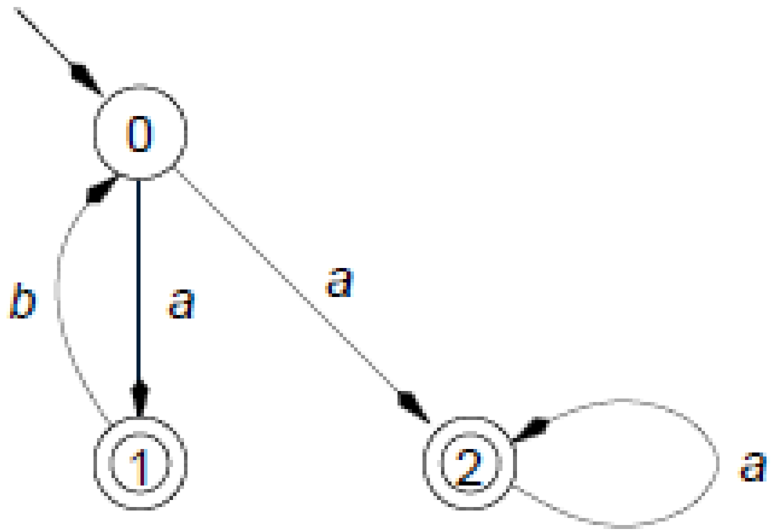
DFA'da: $\delta: Q \times \Sigma \rightarrow Q$

NFA'da geçiş fonksiyonu ikinci argüman olarak λ-geçişi alabilir. Yani NFA bir girdi sembolü almadan da bir durumdan diğerine geçebilir.

NFA'da $\delta(q_1, a) = \{q_0, q_2\}$

NFA'da $\delta(q_i, a)$ kümesi boş olabilir. Yani belirli bir durum için geçiş tanımlanmayabilir.

Aşağıdaki NFA örneğinde 0 durumu, a sembolünü alarak 2 farklı duruma gitmiştir.



DFA VE NFA'NIN FARKLARI

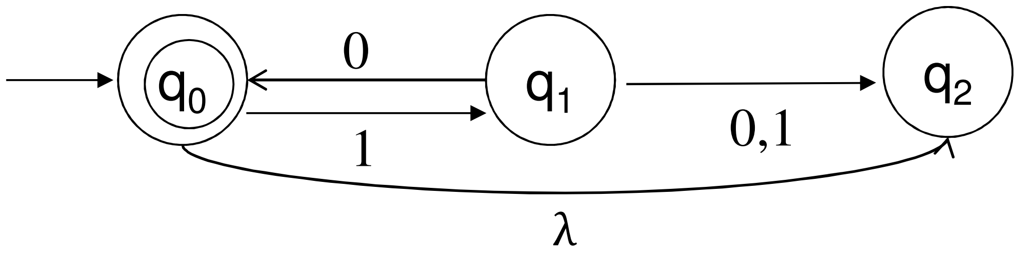
DFA

1. Tüm geçişler gerekircidir
   - Her geçiş sadece tek bir duruma gider
2. Her durum için mümkün olan tüm sembollerin (alfabe) geçişi tanımlanmalıdır
3. Son durum F'in elemanıysa input kabul edilir
4. Durumların sayısından dolayı bazen oluşturmak daha zordur
5. Pratik uygulaması yapılabilir

NFA

1. Bazı geçişler non-deterministic olabilir
   - Bir geçiş, durumların altkümesine gidebilir
2. Tüm semboller için geçiş tanımlanması gerekmez
3. Son durumların bir tanesi F'in elemanıysa input kabul edilir
4. Oluşturmak DFA'ya göre daha kolaydır
5. Pratik uygulamasının deterministic olması gerekir (DFA'ya çevrilir)

Örnek:



λ-geçişi var

$\delta(q_2, 0)$ yok → $\delta(q_2, 0) = \{\ \}$

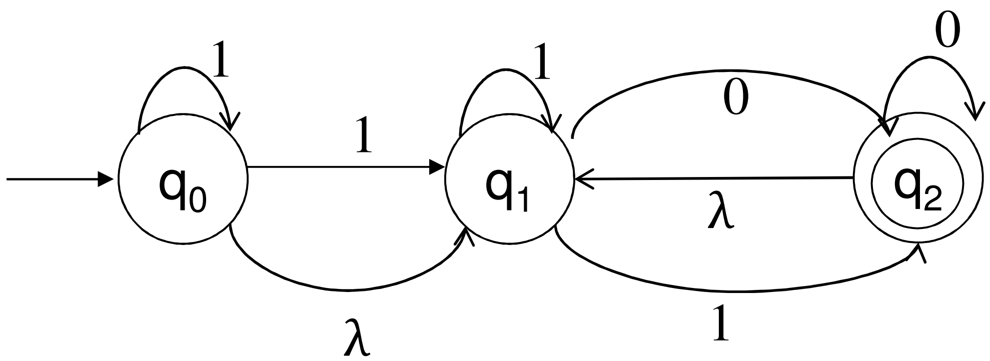
Bu otomatın kabul ettiği dizgiler: λ , 1010 , 101010

Kabul edilmeyen dizgiler: 110, 10100

10 için iki alternatif var. Bir tanesi q_0'a diğeri de q_2 durumuna gider. q_2 kabul durumu olmadığı halde bu dizgi kabul edilmektedir.

Örnek: δ geçiş fonksiyonu şu şekilde verilmiş olsun:

	0	1	λ
q_0	Ø	$\{q_0, q_1\}$	$\{q_1\}$
q_1	$\{q_2\}$	$\{q_1, q_2\}$	Ø
$^*q_2$	$\{q_2\}$	Ø	$\{q_1\}$

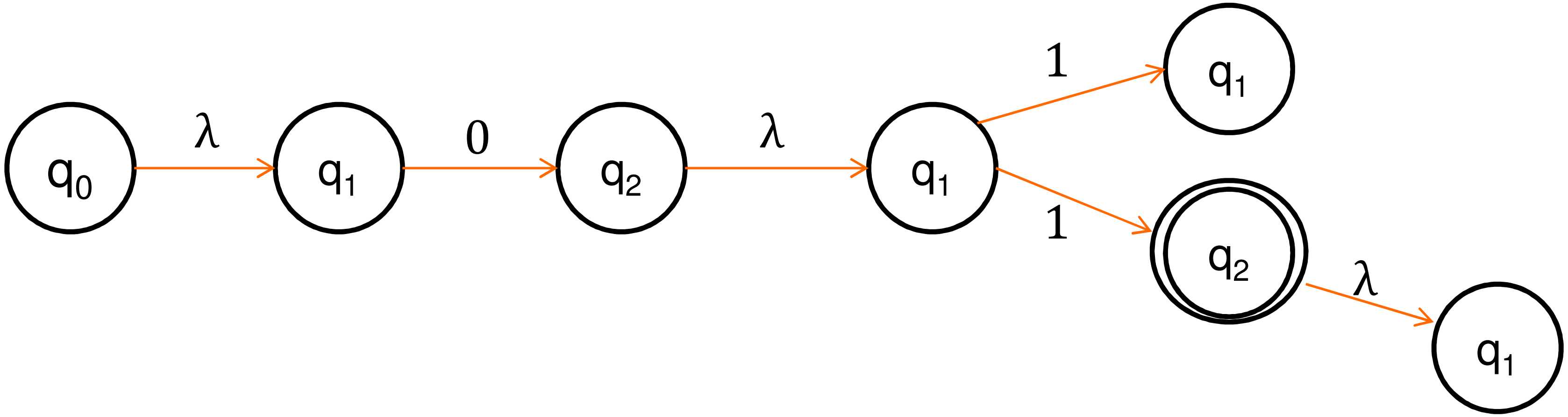


Örneğin; 01 girdisi için NFA'da 3 hesaplama yolu vardır:

- $q_0 \xrightarrow{\lambda} q_1 \xrightarrow{0} q_2 \xrightarrow{\lambda} q_1 \xrightarrow{1} q_1$
- $q_0 \xrightarrow{\lambda} q_1 \xrightarrow{0} q_2 \xrightarrow{\lambda} q_1 \xrightarrow{1} q_2$
- $q_0 \xrightarrow{\lambda} q_1 \xrightarrow{0} q_2 \xrightarrow{\lambda} q_1 \xrightarrow{1} q_2 \xrightarrow{\lambda} q_1$

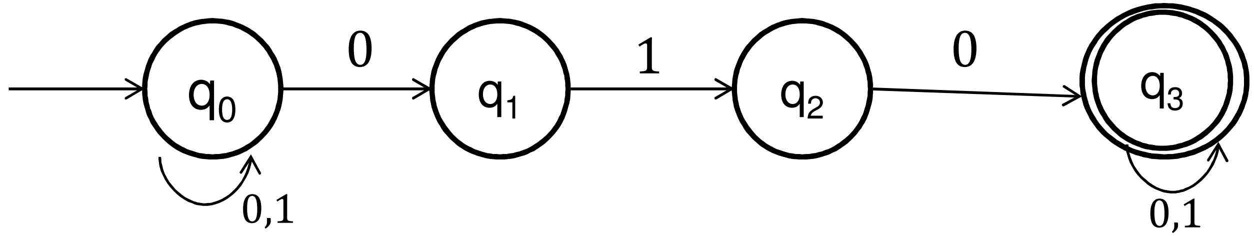
Burada sadece 2. yol kabul durumu ile biter.

Bu örnek için ağaç yapısı şu şekildedir:



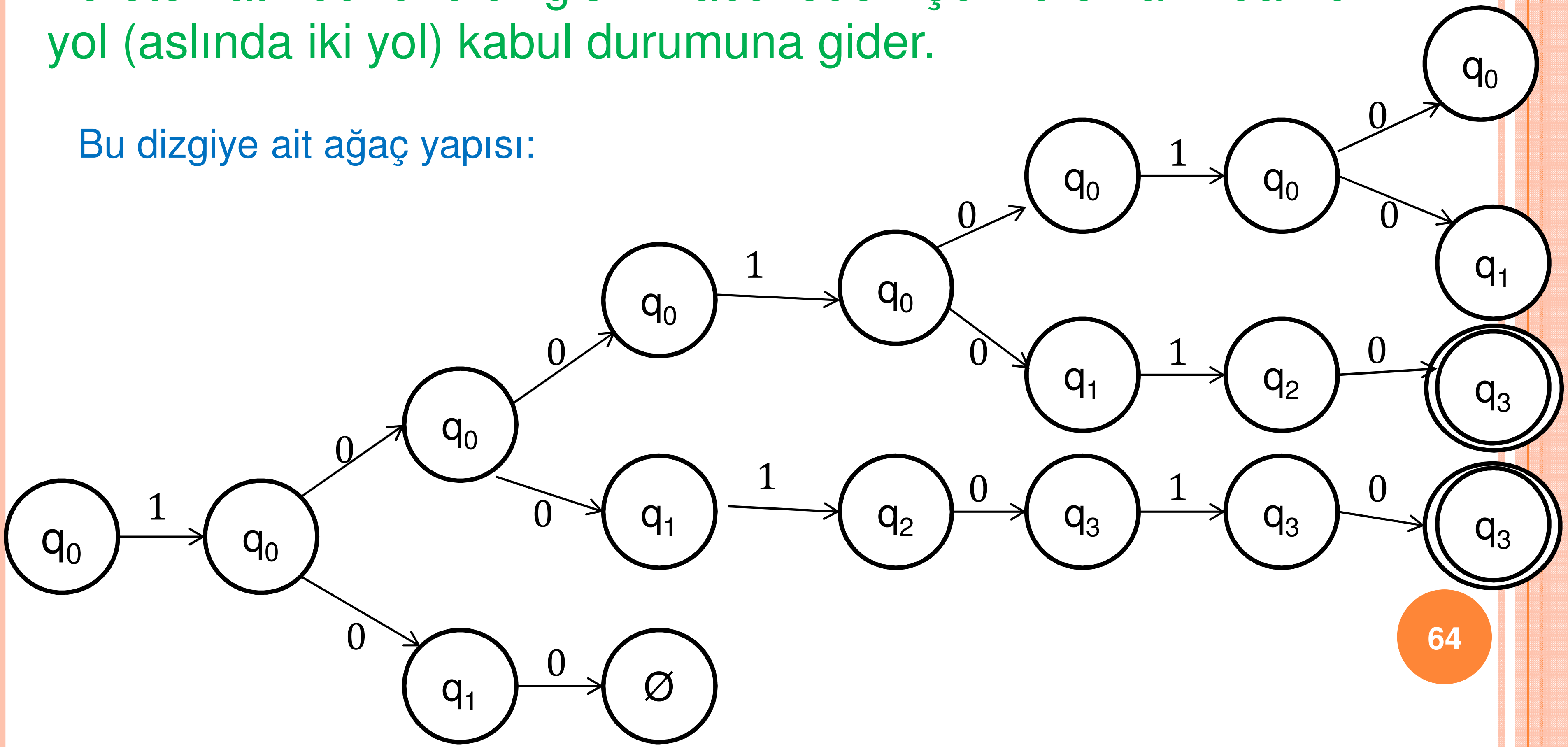
En az bir yol kabul durumuna gittiği için 01 girdisi bu NFA tarafından kabul edilir.

Örnek: 010 alt dizgisini içeren (kabul eden) NFA'yı çiziniz.

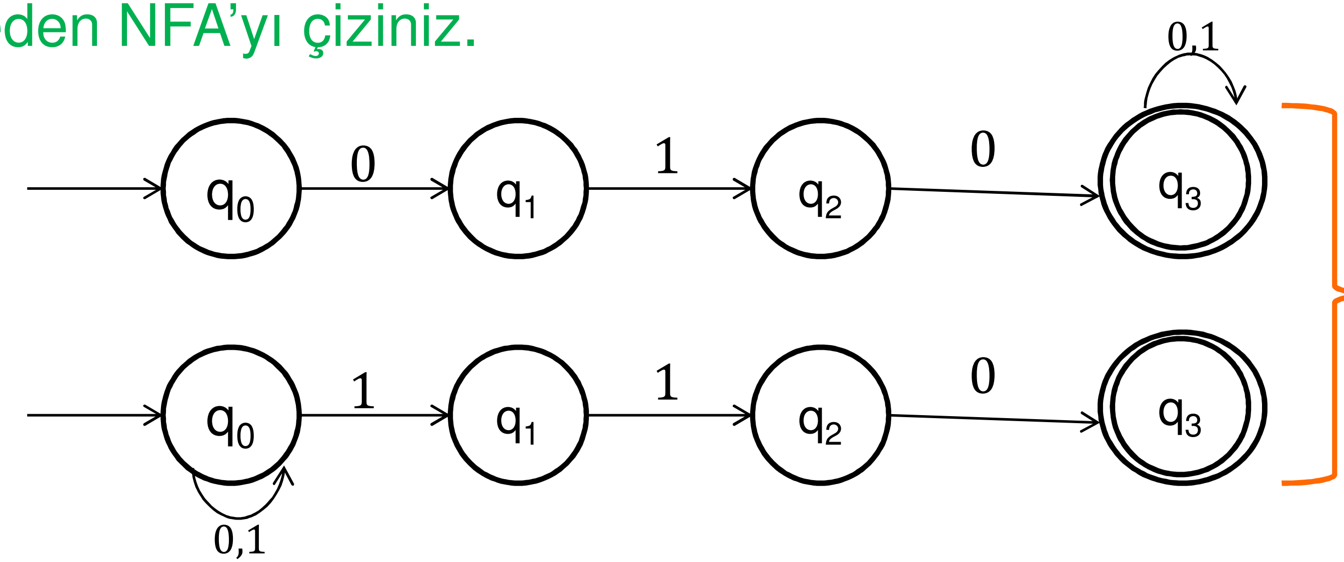


Bu otomat 1001010 dizgisini kabul eder. Çünkü en azından bir yol (aslında iki yol) kabul durumuna gider.

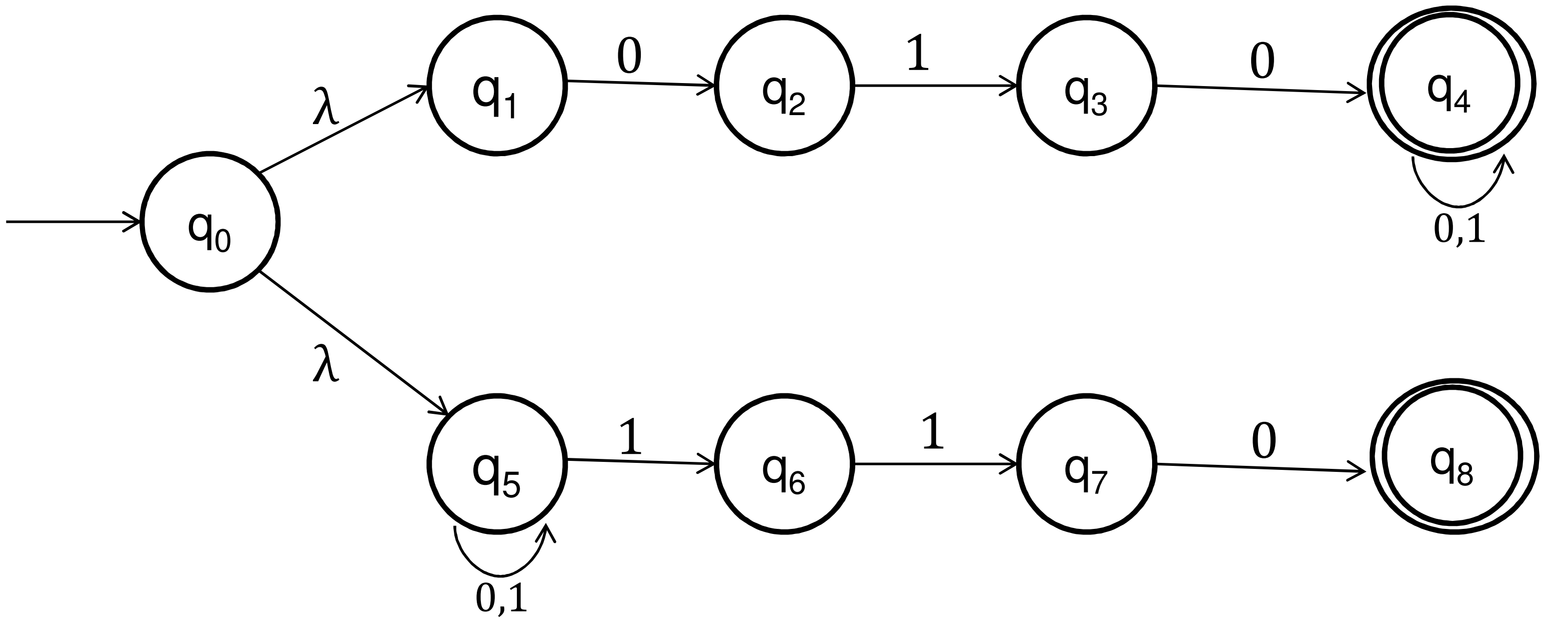
Bu dizgiye ait ağaç yapısı:



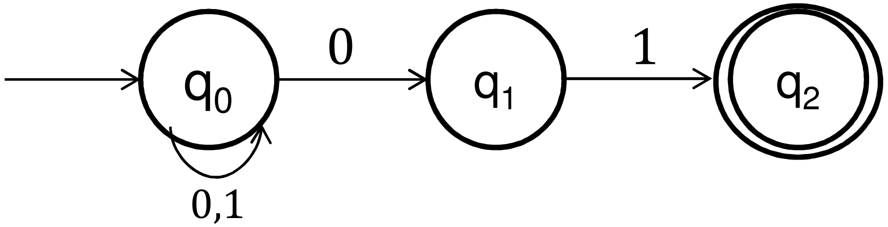
Örnek: 010 ile başlayan ya da 110 ile biten tüm dizgileri kabul eden NFA'yı çiziniz.



λ geçişi ile birleştiririz



Genişletilmiş Geçiş Fonksiyonu (NFA için)



01 ile biten tüm dizgileri kabul eden NFA

δ^* kullanarak bu NFA için 00101 girdisinin nasıl işleneceğini tanımlayalım:

$\delta^*(q_0 , \lambda) = \{q_0\}$

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

$\delta^*(q_0 , 00) = \delta(q_0 , 0) \cup \delta(q_1 , 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$

$\delta^*(q_0 , 001) = \delta(q_0 , 1) \cup \delta(q_1 , 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

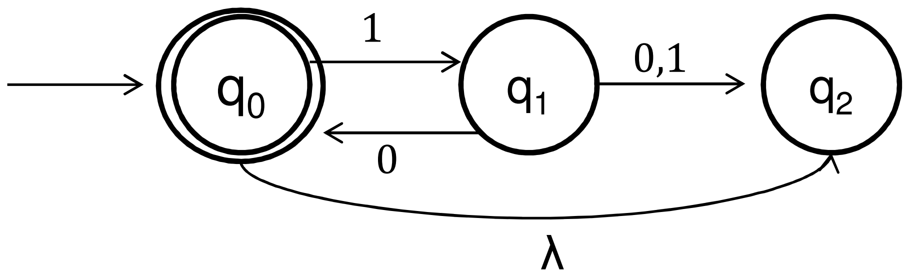
$\delta^*(q_0 , 0010) = \delta(q_0 , 0) \cup \delta(q_2 , 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$

$\delta^*(q_0 , 00101) = \delta(q_0 , 1) \cup \delta(q_1 , 1) = \{q_0, q_2\}$

Kabul durumu

Dil: Bir M= $(Q, \Sigma, \delta, q_0, F)$ NFA'sı tarafından kabul edilen L dili şu şekilde tanımlanır:

$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \cap F \neq \emptyset\}$



Bu otomat tarafından kabul edilen dil?

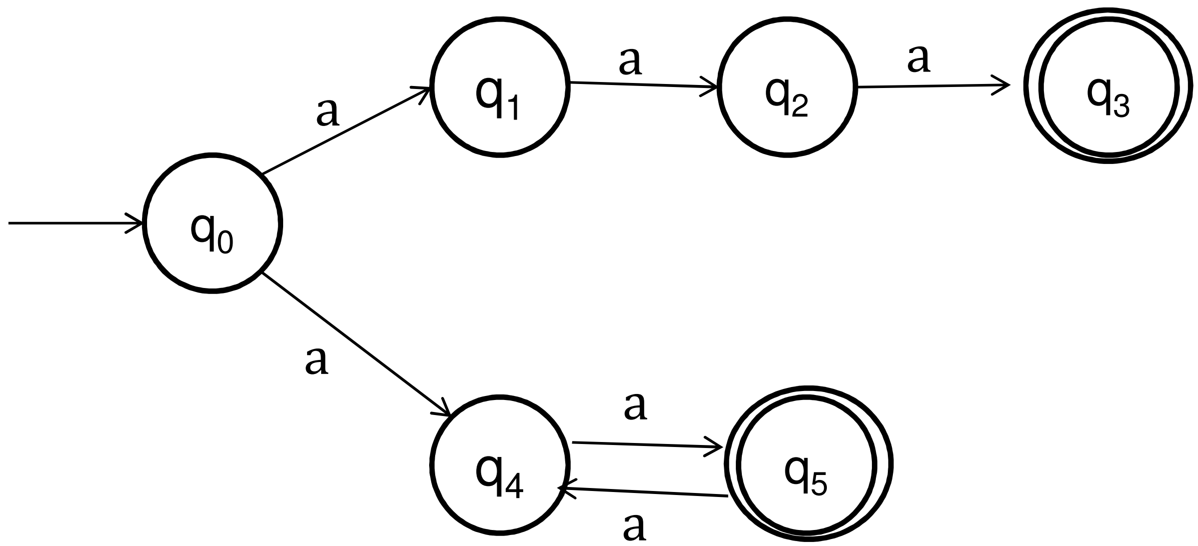
Grafikten görüldüğü gibi bu NFA'nın kabul durumunda durabilmesinin tek yolu 10 dizgisinin tekrarlı şekli ya da boş dizgidir.

$L = \{ (10)^n : n \geq 0\}$

Örneğin 110 dizgisi için $\delta(q_2, 0)$ tanımlanmadığından q_2 durumunda kalır. Böyle bir duruma ölü konfigürasyon denir.

$\delta^*(q_0, 110) = \emptyset$ bu dizgi kabul edilmez.

Örnek:



Bu NFA’nın kabul ettiği dil $L = \{a^3\} \cup \{a^{2n} : n \geq 1\}$

NFA'dan DFA Elde Edilmesi

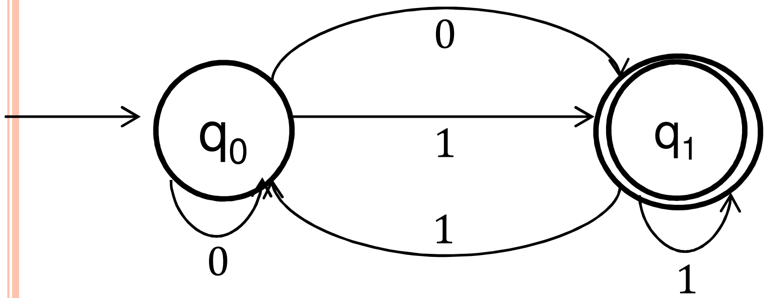
Üzerinde tanımlanmış bir L dilini kabul eden herhangi bir NFA'dan
$\{ M = (Q, \Sigma, \delta, q_0, F) \}$ aynı L dilini kabul eden bir
$N = (Q', \Sigma', \delta', q_0', F')$ DFA'sı elde edilebilir.

Q' = Q nun tüm alt kümelerinin kümesi

Örneğin $Q= \{q_0, q_1\} \rightarrow Q' = \{ \{ \}, \{q_0\}, \{q_1\}, \{q_0, q_1\} \}$

$q_0' = \{q_0\}$
F ' = M'deki bir kabul durumunu içeren Q' 'nün tüm durumlarının kümesi.

Örnek : Aşağıdaki NFA'ya ait DFA'yı oluşturunuz.



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

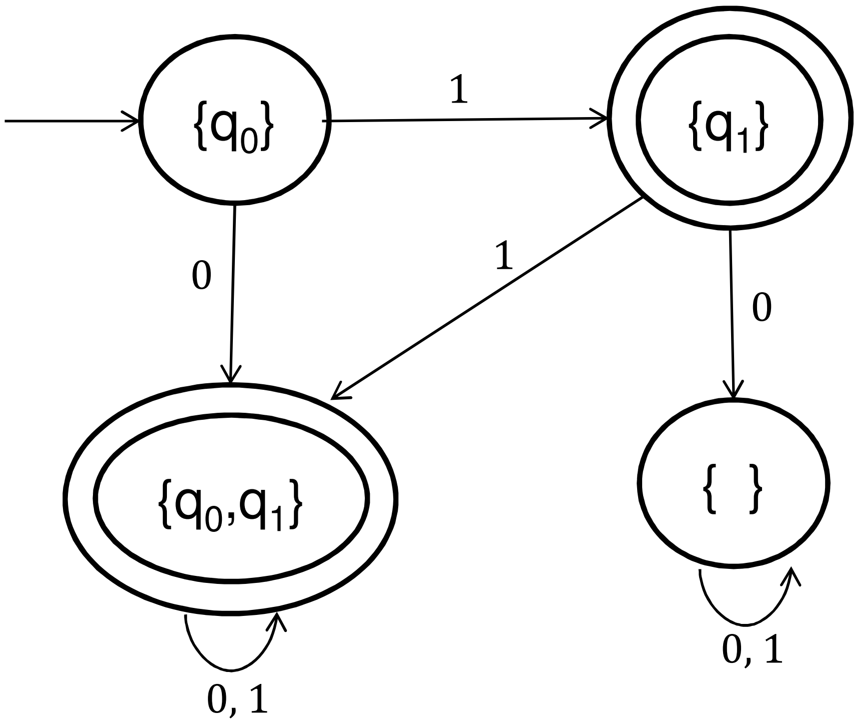
$\delta(q_0, 1) = \{q_1\}$

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

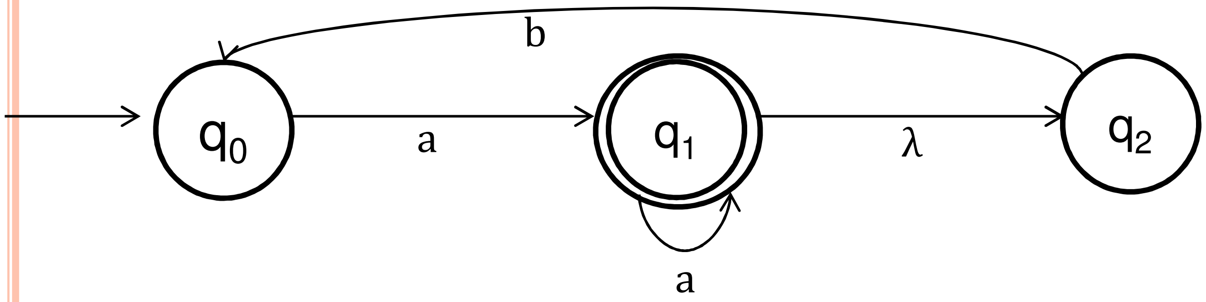
$\delta(\{q_0, q_1\},1) = \delta(q_0,1) \cup \delta(q_1,1) = \{q_0, q_1\}$

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

$\delta(q_1, 1) = \{q_0, q_1\}$

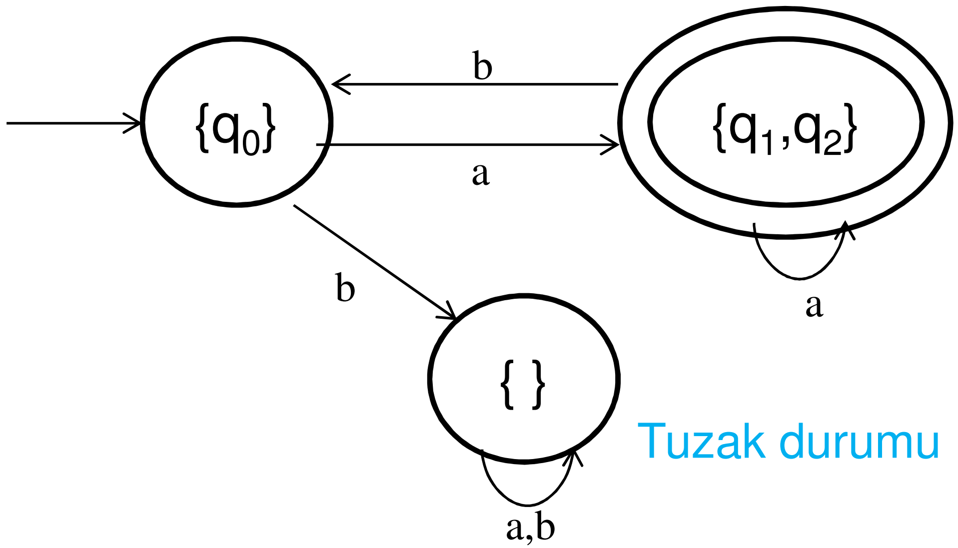


Örnek : Aşağıdaki NFA’yı eşdeğer DFA’ya dönüştürün.



NFA q_0 durumu ile başladığı için DFA’nın başlangıcı $\{q_0\}$ olacaktır.

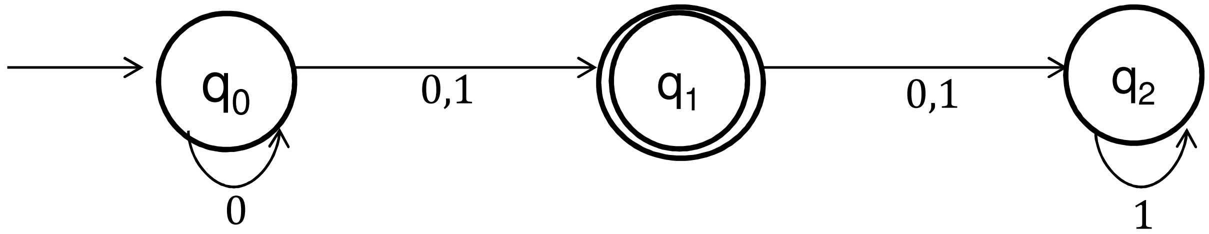
$\delta(q_0, a) = \{q_1, q_2\}$
$\delta(q_0, b) = \{\ \}$
$\delta(\{q_1, q_2\},a) = \{q_1, q_2\}$
$\delta(\{q_1, q_2\},b) = \{q_0\}$



DFA'nın kabul durumları NFA'nın bir kabul durumunu içerdiği için her ikisi de aynı dizgi kümesini kabul edecektir.

Burdan diyebiliriz ki NFA tarafından kabul edilen her dil DÜZGÜN DİL'dir.

Örnek : Aşağıdaki NFA'yı eşdeğer DFA'ya dönüştürün.



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

$\delta(q_0, 1) = \{q_1\}$

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

$\delta(q_1, 1) = \{q_2\}$

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

$\delta(q_2, 1) = \{q_2\}$

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

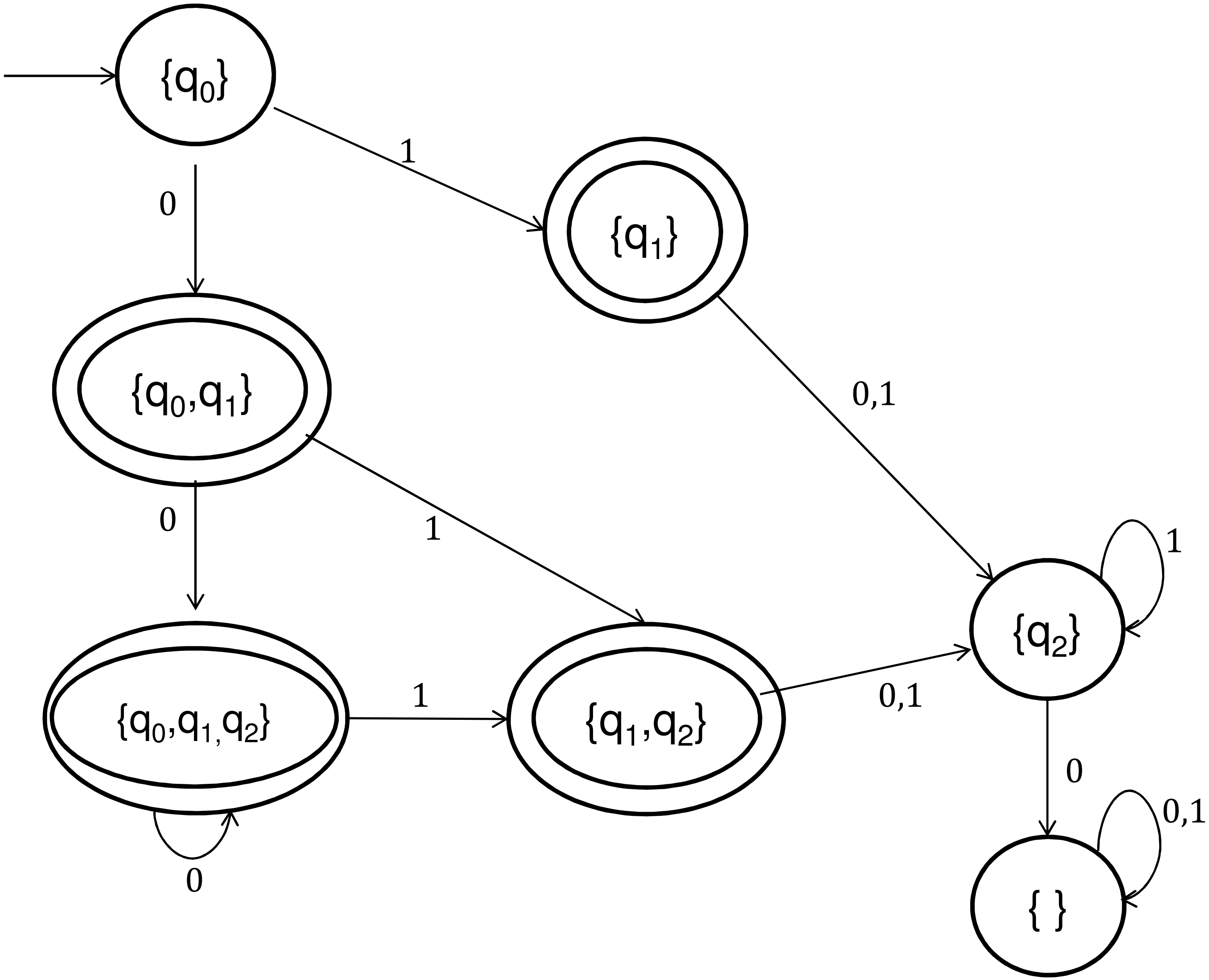
$\delta(\{q_0, q_1\},1) = \{q_1, q_2\}$

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

$\delta(\{q_0, q_1, q_2\},1) = \{q_1, q_2\}$

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

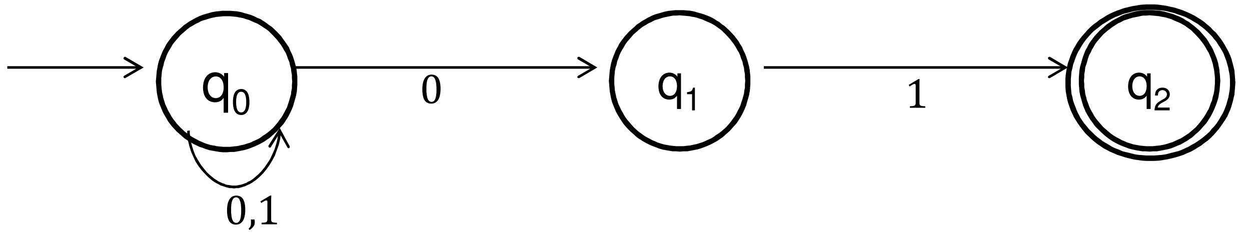
$\delta(\{q_1, q_2\},1) = \{q_2\}$



{q0}
1
0
{q1}
{q0,q1}
0,1
0
1
1
{q2}
{q0,q1,q2}
1
{q1,q2}
0,1
0
0,1
0
{ }

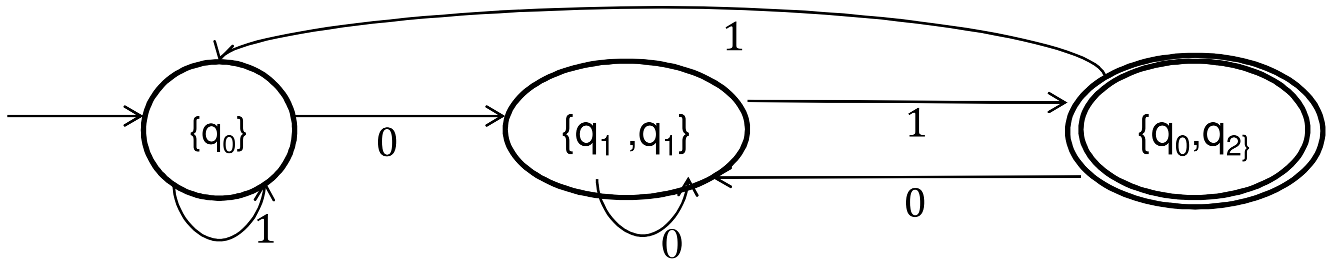
Verilen NFA’nın daha karmaşık olması durumunda NFA’nın tüm durumlarının alt kümelerini ele almaya gerek kalmayabilir. Böyle durumlarda, $\{q_0\}$ başlangıç durumundan başlayarak tek tek DFA’nın durumlarını oluşturabiliriz.

Örnek : Aşağıdaki NFA'yı eşdeğer DFA'ya dönüştürün.



	δ	0	1
A	Ø	Ø	Ø
B	$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
C	$\{q_1\}$	Ø	$\{q_2\}$
*D	$\{q_2\}$	Ø	Ø
E	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
*F	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
*G	$\{q_1, q_2\}$	Ø	$\{q_2\}$
*H	$\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

δ	0	1
A	A	A
→B	E	B
C	A	D
*D	A	A
E	E	F
*F	E	B
*G	A	D
*H	E	F



λ-NFA'DAN NFA ELDE EDİLMESİ

Σ üzerinde tanımlanmış bir L dilini kabul eden herhangi bir $M = (Q, \Sigma, \delta, q_0, F)$ λ-NFA'dan aynı L dilini kabul eden bir $N = (Q', \Sigma', \delta', q_0', F')$ NFA elde edilebilir.

M'de sadece tek bir a sembolü ve sıfır ya da daha fazla λ alarak q_i'den q_j'ye giden bir yol varsa

$$\delta'(q_i, a) = q_j \text{ yazabiliriz.}$$

önemli

$F' = F \cup \{q_0\}$ → {M'de λ geçişleri kullanarak q_0'dan bir son duruma erişilebiliyorsa }

→ Aksi durumda $F' = F$ olur.

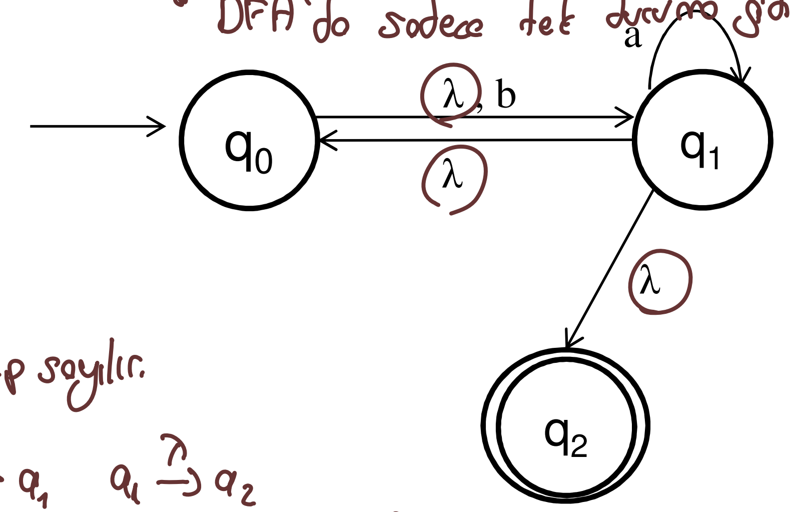
⇒ Sonraki örnekte var.

* NFA'da birden fazla duruma gidilebilir.

* DFA'da sadece tek duruma gidilir.

Örnek:

M



İlk bu kısmı buluyoruz

$\delta'(q_0, a) = \{q_0, q_1, q_2\}$
$\delta'(q_0, b) = \{q_0, q_1, q_2\}$
$\delta'(q_1, a) = \{q_0, q_1, q_2\}$
$\delta'(q_1, b) = \{q_0, q_1, q_2\}$

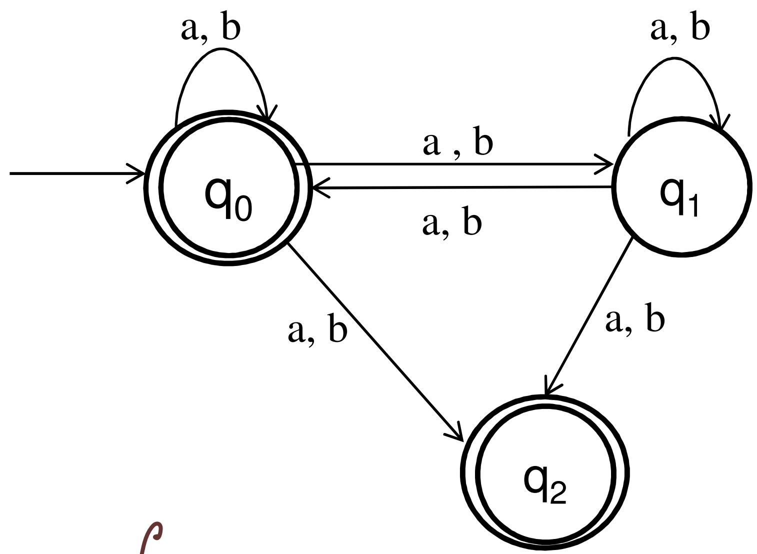
→ λ'lar hep sayılır.

$(q_0, a) = q_0 \xrightarrow{\lambda} q_1 \quad q_1 \xrightarrow{\lambda} q_2$
$q_1 \xrightarrow{\lambda} q_0$

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

λ -NFA'dan elde edilen NFA ile λ -NFA aynı dizgi kümesini kabul eder.

N



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

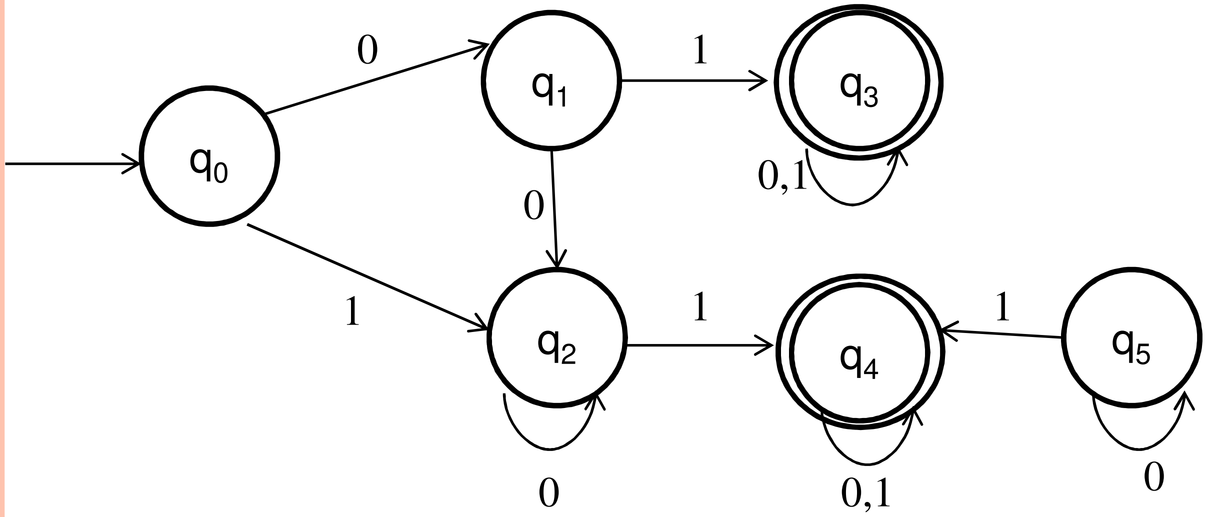
q1'de olabilir (?)

⇒ Her a ile de b ile de her q'ya gidecek.

otomat

Her DFA tek bir dil tanımlar, ancak bunun tersi doğru değildir. Verilen bir dil için, bu dili kabul eden birçok DFA vardır. Bu eşdeğer otomatlar arasındaki en önemli farklılık durumlarının sayısıdır. Örneğin;

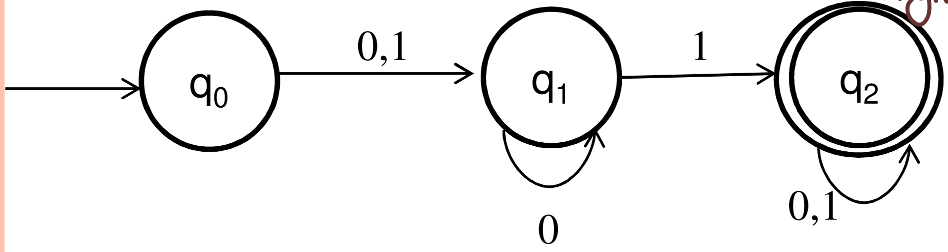
→ Durum sayısı min. olan daha iyidir.



$(q_5)^{\times}$ → atıyoruz.

→ Ulaşılamayan durum var mı

→ Ayırt edilebilen durumlar



Bu iki DFA birbirine eşdeğerdir. q_5 durumu otomatta hiçbir işe yaramaz, çünkü q_0 başlangıç durumundan hiçbir zaman q_5 durumuna erişilemez. İkinci DFA'nın tercih edilme sebebi basitliği ve depolama kolaylığıdır. Bu nedenle DFA'larda mümkün olduğunca durum sayısını azaltmak gerekir.

Tanım: DFA'nın iki durumu p ve q için, tüm $w \in \Sigma^*$ olmak üzere

$$\delta^*(p, w) \in F \quad \text{iken} \quad \delta^*(q, w) \in F$$

ve

$$\delta^*(p, w) \notin F \quad \text{iken} \quad \delta^*(q, w) \notin F$$

ise p ve q durumları ayırt edilemeyen (indistinguishable) durumlardır.

$$\delta^*(p, w) \in F \quad \text{ve} \quad \delta^*(q, w) \notin F \text{ ise}$$

p ve q durumları ayırt edilebilir (distinguishable) durumlardır.

p ve q durumları ayırt edilemeyen durumlar ve
q ve r durumları da ayırt edilemeyen durumlar ise
p ve r durumları da ayırt edilemez.

Bir DFA'nın durumlarının sayısını azaltmanın yöntemlerinden bir tanesi ayırt edilemeyen durumları bulmaya ve bunları birleştirmeye dayanır.

→Ayırt edilemeyen durum çiftlerinin bulunması yöntemi:

mark → işaretle yöntemi

1- Erişilemeyen tüm durumları DFA'dan çıkar. (Başlangıç durumundan başlayarak tüm basit yolları bul. Bu yollardan herhangi birine ait olmayan durum, erişilemeyen durumdur.)

2- Tüm (p,q) durum çiftlerini incele. Eğer $p \in F$ ve $q \notin F$ ya da tam tersi ise (p,q) çiftini ayırt edebilen durum olarak işaretle.

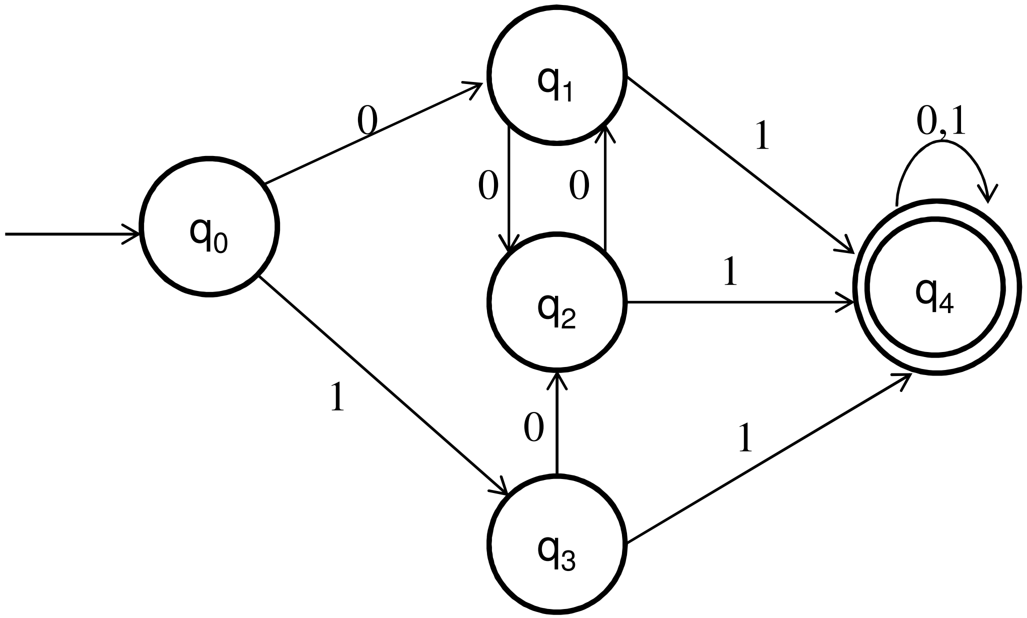
3- Daha önceden (2. aşamada) işaretlenmemiş tüm durum çiftleri işaretlenene kadar bu adımı tekrarla:

Tüm (p,q) çiftleri ve $a \in \Sigma$'lar için

$\delta(p, a) = p_a$ ve $\delta(q, a) = q_a$ hesapla.

Eğer (p_a, q_a) çifti ayırt edilebilen olarak işaretlenmişse (p, q) çiftini de işaretle.

Örnek:



İşaretle yönteminin 2. aşamasına göre;

(q_0, q_4) (q_1, q_4) (q_2, q_4) (q_3, q_4) → ayırt edilebilen çiftler

Kalan çiftler:

(q_0, q_1) , (q_0, q_2) , (q_0, q_3) , (q_1, q_2) , (q_1, q_3) , (q_2, q_3)

3. aşamaya göre de;

$$\left.\begin{array}{l}\delta(q_0, 1) = q_3 \\ \delta(q_1, 1) = q_4\end{array}\right\}$$ (q_3, q_4) ayırt edilebilen çiftler olduğu için (q_0, q_1) de ayırt edilebilir.

* Biri kabul durumu diğeri kabul değilse, ayırt edilebilir çift olur. (Kenara alıyoruz.)

→ ayırt edilemeyenleri nasıl birleştiririz ona bakıyoruz.

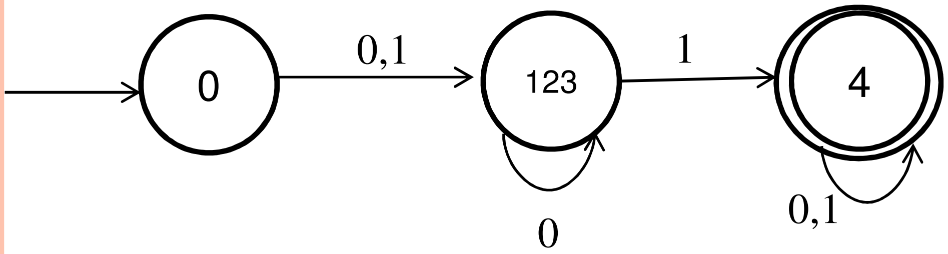
Bu şekilde devam ederek;

(q_0, q_1) , (q_0, q_2) , (q_0, q_3) , (q_0, q_4) , (q_1, q_4) , (q_2, q_4),(q_3, q_4) durumları ayırt edilebilen durumlar olarak işaretlenir.

Buna göre ayırt edilemeyen durumlar:

(q_1, q_2) , (q_1, q_3) , (q_2, q_3) $\rightarrow$ q_1, q_2, q_3 durumları ayırt edilemeyen durumlardır. (Birleştirilir)

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



minimum durumlu DFA

(q_0, q_2)

$$\left.\begin{array}{l}\delta(q_0, 1) = q_3 \\ \delta(q_2, 1) = q_4\end{array}\right\} (q_3, q_4)$$

(q_0, q_3)

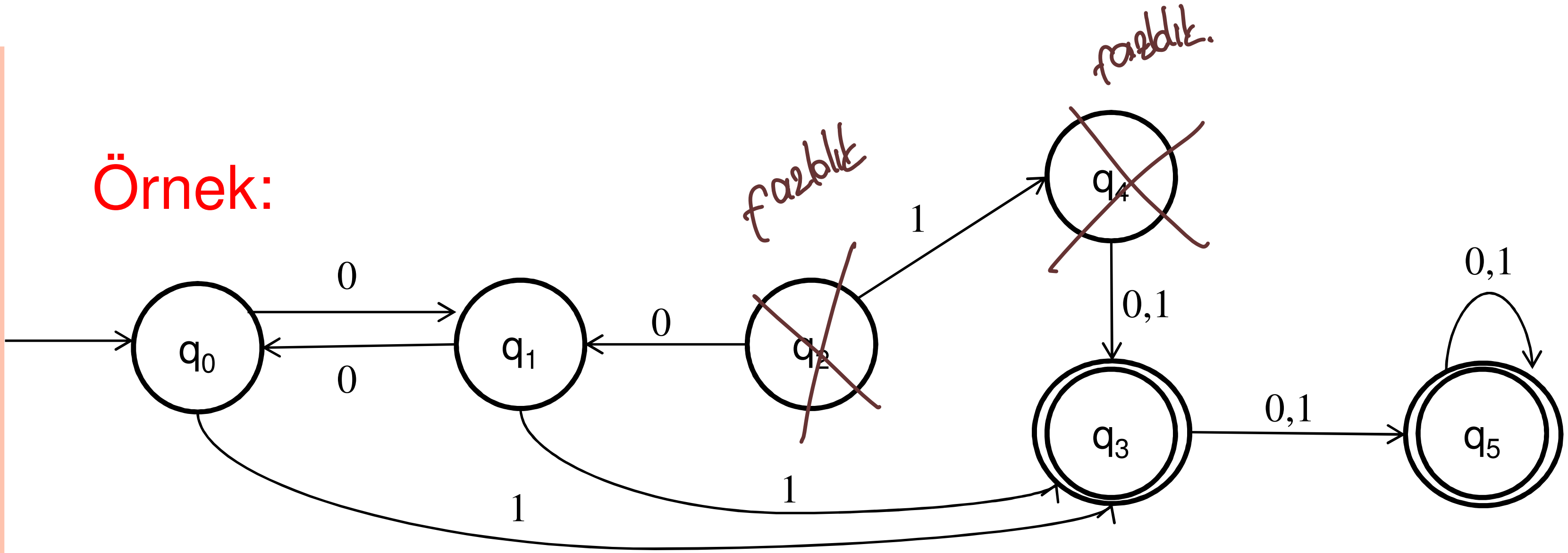
$$\left.\begin{array}{l}\delta(q_0, 1) = q_3 \\ \delta(q_3, 1) = q_4\end{array}\right\} (q_3, q_4)$$

(q_1, q_2)

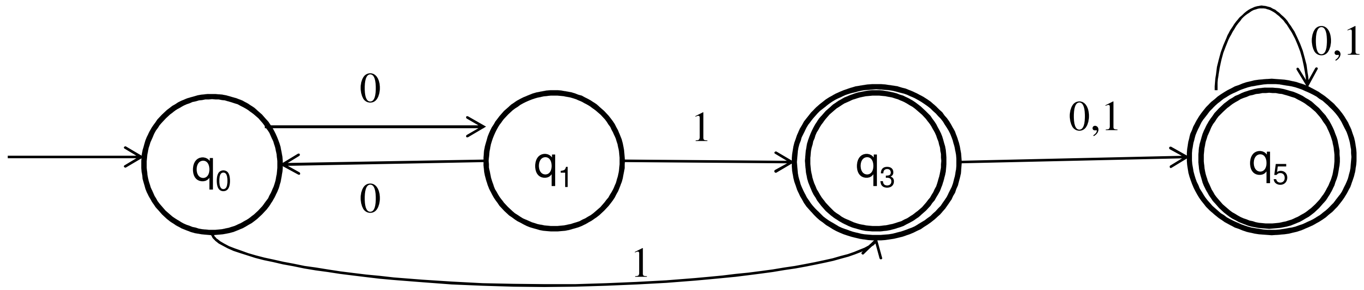
$\delta(q_1, 1) = q_4$ $\quad$ $\delta(q_1, 0) = q_2$

$\delta(q_2, 1) = q_4$ $\quad$ $\delta(q_2, 0) = q_1$

Örnek:



q_2 ve q_4 durumları erişilemeyen durumlar olduğu için öncelikle bu durumlar DFA'dan çıkarılır.



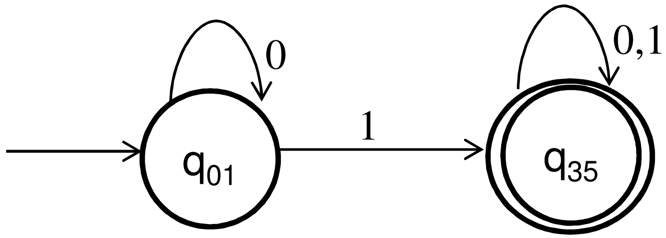
(q_0, q_3) , (q_1, q_3) , (q_0, q_5) , (q_1, q_5) ayırt edilebilen çiftler olarak belirlenir. *

ikisinin de çıktıları aynı. Yani => $\underline{\delta(q_0, -) = \delta(q_1, -)}$

(q_0, q_1) , (q_3, q_5) ayırt edilemeyen durumlar olarak belirlenir. *

→ Ayırt edilemeyenler birleştirilir.

- Bu şekilde minimal DFA şu şekilde belirlenir:



2 çizgi kabul durumu demek.