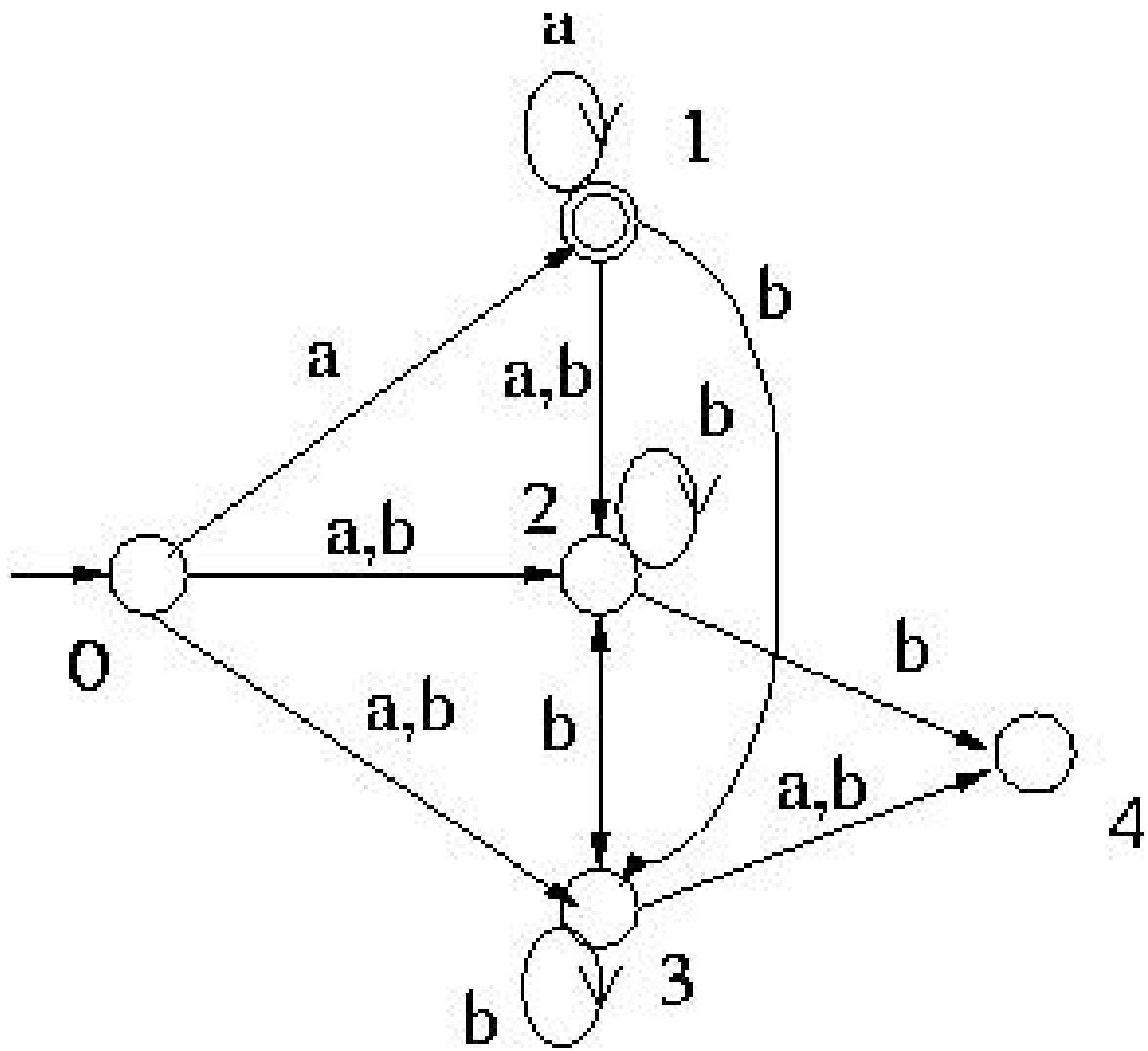
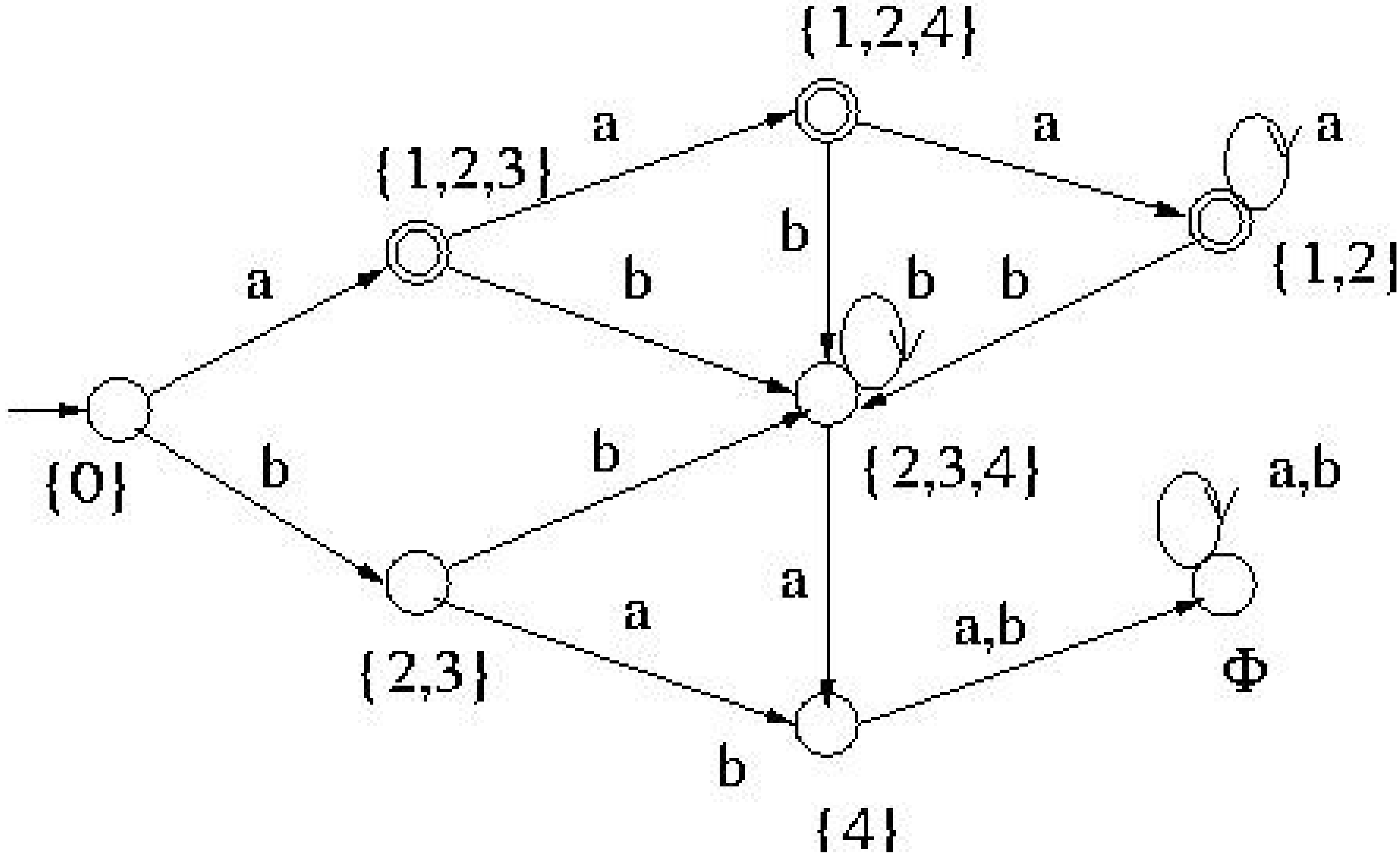
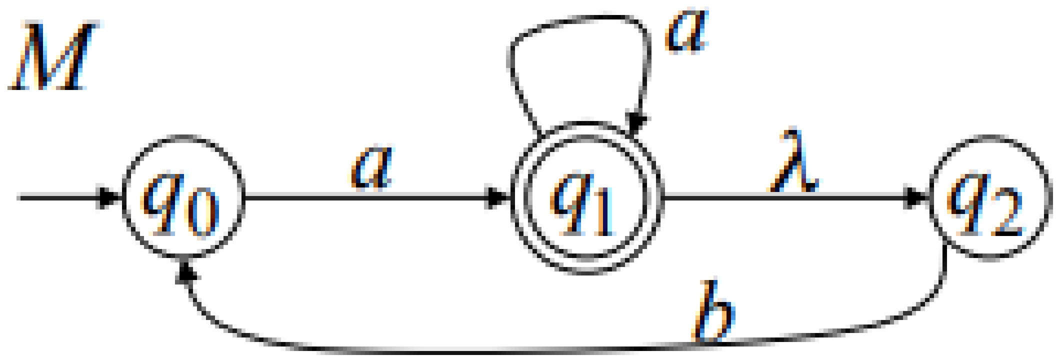
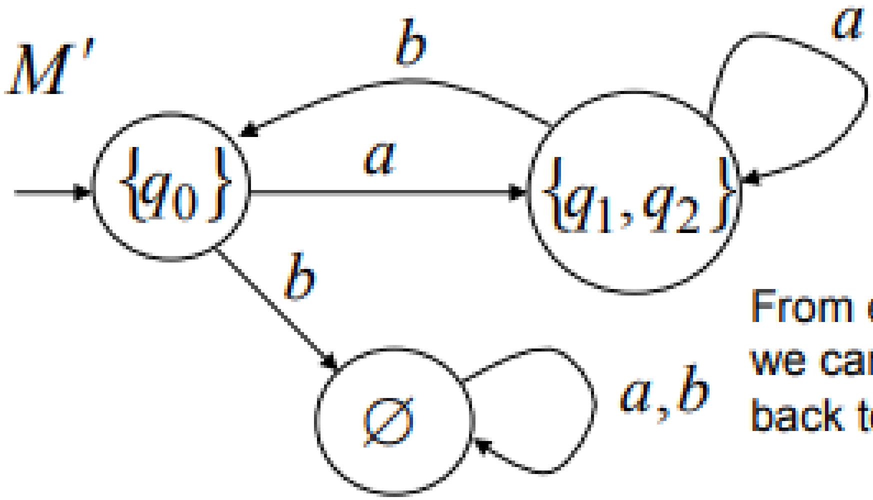
Aşağıdaki NFA'yı DFA'ya çeviriniz.




{1,2,4}
a
a
a
{1,2,3}
b
b
b
b
{1,2}
a
b
{0}
b
{2,3,4}
a,b
a
a
a,b
{2,3}
Φ
b
{4}

Aşağıdaki NFA'yı DFA'ya çeviriniz.





From empty set,
we can only get
back to empyt set.

PAMUKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
2021 BAHAR

Biçimsel Diller ve Otomata Teorisi
Formal languages and automata theory

FA to RE Conversion

Öğrendiklerimiz:

A language is regular iff it is defined by some regular expression.

RE < -- > FA

Sonlu durum makineleri ve düzenli ifadeler aynı dil sınıfını tanımlar.Bunu kanıtlamak için şunları göstermeliyiz:

Teorem: Bir düzenli ifade ile tanımlanabilen herhangi bir dil, bazı FA'lar tarafından kabul edilebilir ve bu nedenle düzenlidir.(RE—>FA)

(Bunu gördük)

Teorem: Her düzenli dil (yani, bazı FA tarafından kabul edilebilen her dil) bir düzenli ifade ile tanımlanabilir. (FA→RE)

Teorem: Her düzenli dil (yani, bazı FA tarafından kabul edilebilen her dil) bir düzenli ifade ile tanımlanabilir. (FA→RE)

- *$M = (K, \sum, \Delta, s, F)$ bir automat olsun (DFA veya NFA olabilir).*
- *Bu automat için $L(R) = L(M)$ olacak şekilde bir regular expression R* ***her zaman*** *oluşturulabilir.*
- *L(M) dili sonlu sayıda basit dillerin birleşimi olsun.*
- *$K = \{q_1, ..., q_n\}$ ve $s = q_1$ olsun.*
- *R(i,j,k), M makinesini qi durumundan qj durumuna* ***k+1 veya daha büyük numaralanmış ara durumlara uğramadan götüren*** *$\sum^*$ içerisindeki tüm katarlardır. Burada q_i ve q_j k'dan büyük numaralı olabilir: sadece ara düğümler için koşul var.*
- *$i, j =$ **1**$, ..., n$ ve $k =$ **0**$, ..., n$ için $\sum^*$ üzerinde bir R(**i, j**, **k**) regular expression tanımlanabilir.*

Bu durumda «uğramadan" girmeden / çıkmadan anlamına gelir. i veya j'nin (veya her ikisinin) k'den büyük olabileceğini unutmayın.

Teorem: Her düzenli dil (yani, bazı FA tarafından kabul edilebilen her dil) bir düzenli ifade ile tanımlanabilir. (FA→RE)

- *R(i,j,k), M makinesini qi durumundan qj durumuna k+1 veya daha büyük numaralanmış ara durumlara uğramadan götüren ∑* içerisindeki tüm katarlardır. Burada qi ve qj k'dan büyük numaralı olabilir: sadece ara düğümler için koşul var.*
- *Öyleyse:*
- $R(i, j, n) = \{w \in \Sigma^* : (q_i, w) \vdash_M^* (q_j, e)\}$

Kabul edilen dil ise aşagıdaki gibi tanımlanır;

$L(M) = \bigcup \{R(1, j, n) : q_j \in F$ $\quad s = q_1$

Özyinelemeli olarak elde edilir:

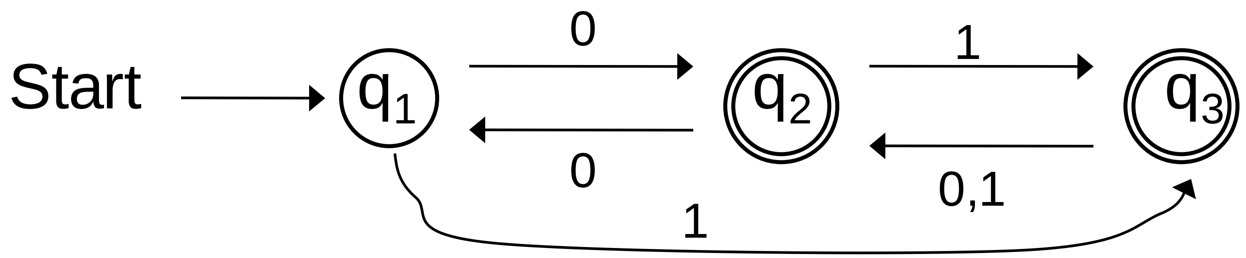
- Başlangıç durumu (k = 0):
  - If $i \neq j$, $R_{ij0} = \{a \in \Sigma : \delta(q_i, a) = q_j\}$
  - If $i = j$, $R_{ii0} = \{a \in \Sigma : \delta(q_i, a) = q_i\} \cup \{\varepsilon\}$

- Özyinelemeli durum (k > 0):

  $R_{ijk} = R_{ij(k-1)} \cup R_{ik(k-1)}(R_{kk(k-1)})^{*}R_{kj(k-1)}$

Her R_{ijk} r_{ijk} şeklinde bir düzenli ifade olarak yazılabilir.

Örnek (r_{ijk} is $r_{ij(k-1)} \cup r_{ik(k-1)}(r_{kk(k-1)})^* r_{kj(k-1)}$)



	k=0	k=1	k=2
r_{11k}	ε	ε	(00)*
r_{12k}	0	0	0(00)*
r_{13k}	1	1	00*1U1= 0*1
r_{21k}	0	0	0(00)*
r_{22k}	ε	ε ∪ 00	(00)*
r_{23k}	1	1 ∪ 01	0*1
r_{31k}	∅	∅	(0 ∪ 1)(00)*0
r_{32k}	0 ∪ 1	0 ∪ 1	(0 ∪ 1)(00)*
r_{33k}	ε	ε	ε ∪(0 ∪ 1)0*1

FA to RE dönüştürme

- **Çözüm 1. Durum Eleme Yöntemi**
- **Çözüm 2.** Arden Teoremi

FA to RE dönüştürme

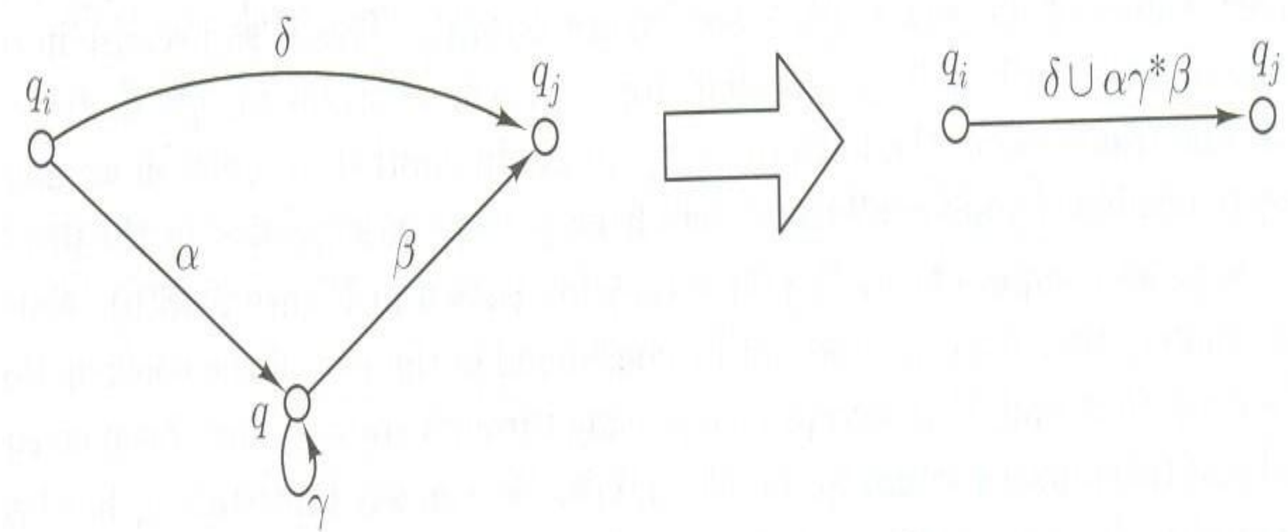
- ÇÖZÜM 1.

Durum Eleme Yöntemi –

- **Adım 1** -Başlangıç durumu bir kabul durumuysa veya kendisine gelen geçişler varsa, yeni bir kabul etmeyen başlangıç durumu ekleyin ve yeni başlangıç durumu ile önceki başlangıç durumu arasına bir e-geçişi ekleyin.
- **Adım 2** -Birden fazla kabul durumu varsa veya tek kabul durumunda kendisinden çıkan geçişler varsa, yeni bir kabul durumu ekleyin, diğer tüm durumları kabul etmeyecek hale getirin ve her önceki kabul durumundan yeni kabul durumuna bir e-geçişi ekleyin. Böylece tek final state elde edin.
- **Adım 3** -Sırasıyla, her başlangıç olmayan final olmayan durum için, durumu ortadan kaldırın ve geçişleri buna göre güncelleyin.

FA to RE dönüştürme

Örnek: iki durum arasındaki geçişin regular expression ile ifade edilmesi



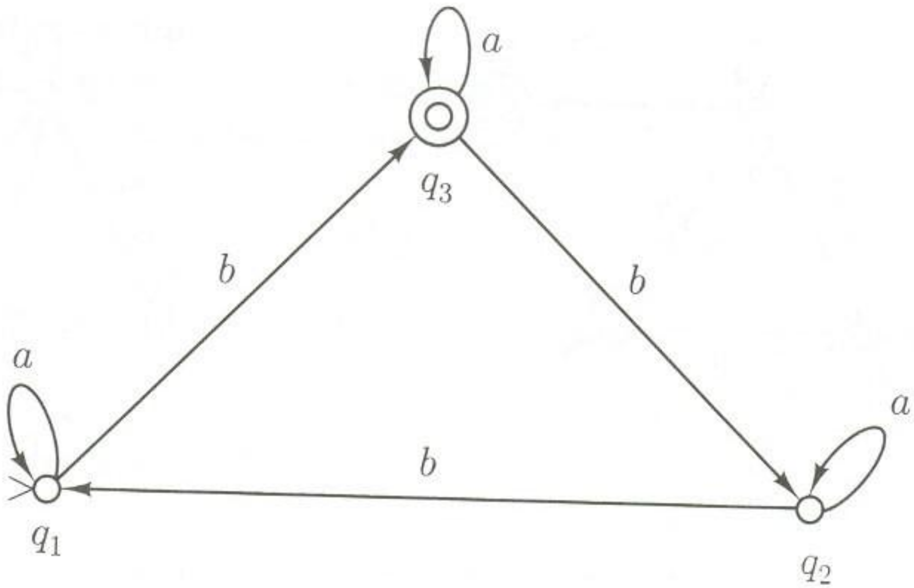
- *iki durum arasındaki alternatif yollar* ∪ *ile birleştirilir.*
- *Kendi kendisine dönen geçişler * ile ifade edilir.*
- *Ardarda geçişler concatenation ile ifade edilir.*

FA to RE dönüştürme

Örnek-1

$L = \{w \in \{a, b\}^* :$ *w içindeki b sayısı 3k+1 şeklinde olan tüm stringler}*

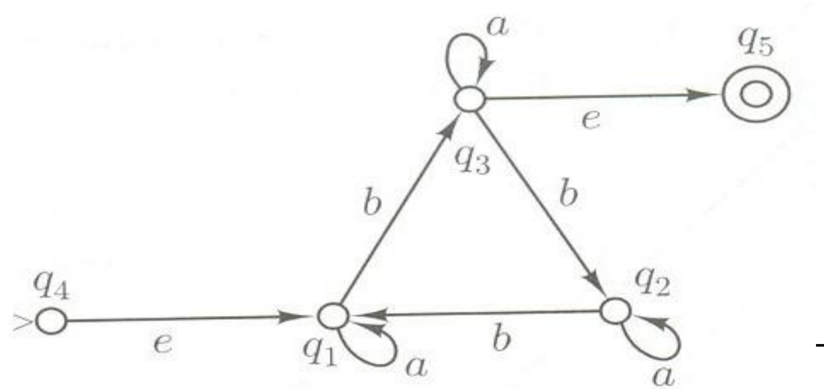
şeklinde *tanımlanan dili kabul eden otomat için regular expression* oluşturunuz.



FA to RE dönüştürme

Örnek-1: (Devam)

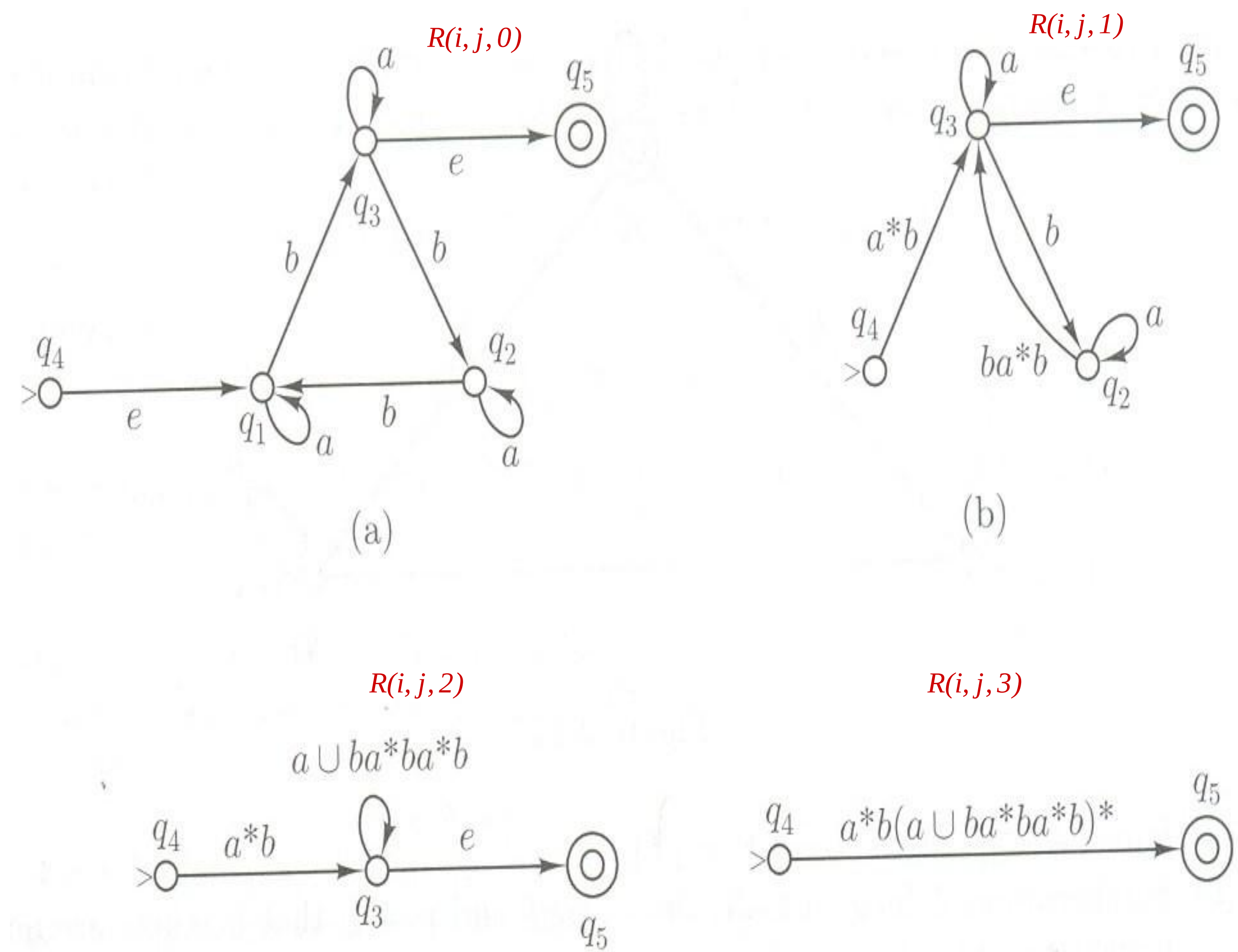
- *Başlangıç ve bitiş durumlarının önüne sonuna e-transition'larla geçişe sahip olan yeni başlangıç ve bitiş durumları q_{n-1} ve q_n durumları olarak eklenir.*
- *$s = q_{n-1}$ ve $f = q_n$ olarak belirlenir. Sonuçta elde edilecek regular expression $R(n-1, n, n)$ şeklinde ifade edilecektir.*
- *İlk önce $R(i, j, 0)$, sonra $R(i, j, 1)$ olacak şekilde tüm basit regular expression'lar belirlenir.*
- *Her aşamada bir state kaldırılır. ($R(i, j, 1)$ için q_1, $R(i, j, 2)$ için q_2, ..., $R(n-1, n, n-2)$)*



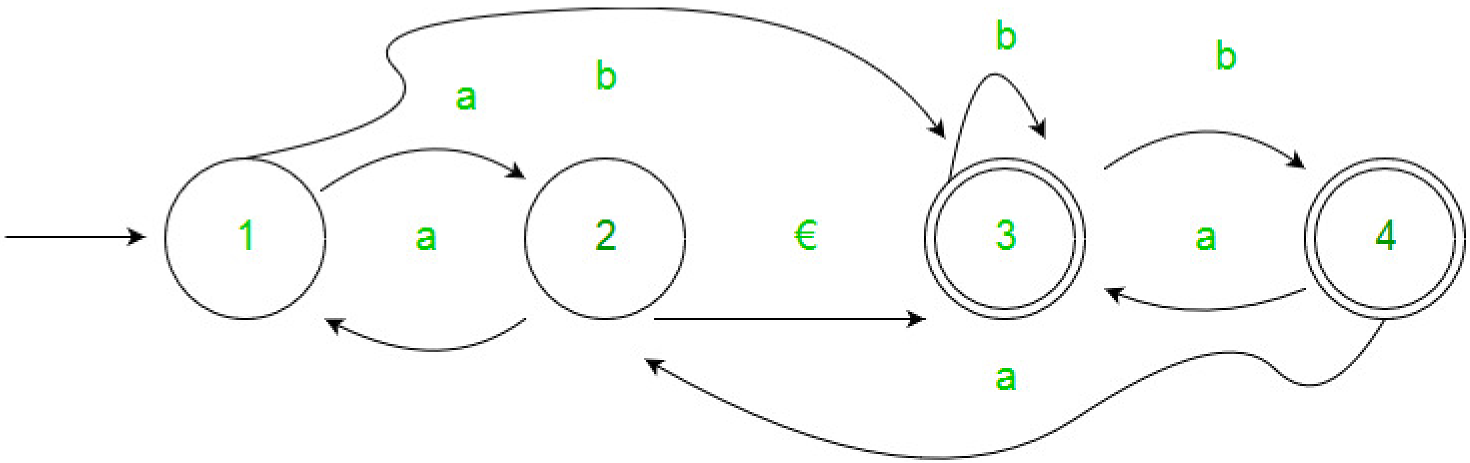
$R(i, j, 0)$

FA to RE dönüştürme

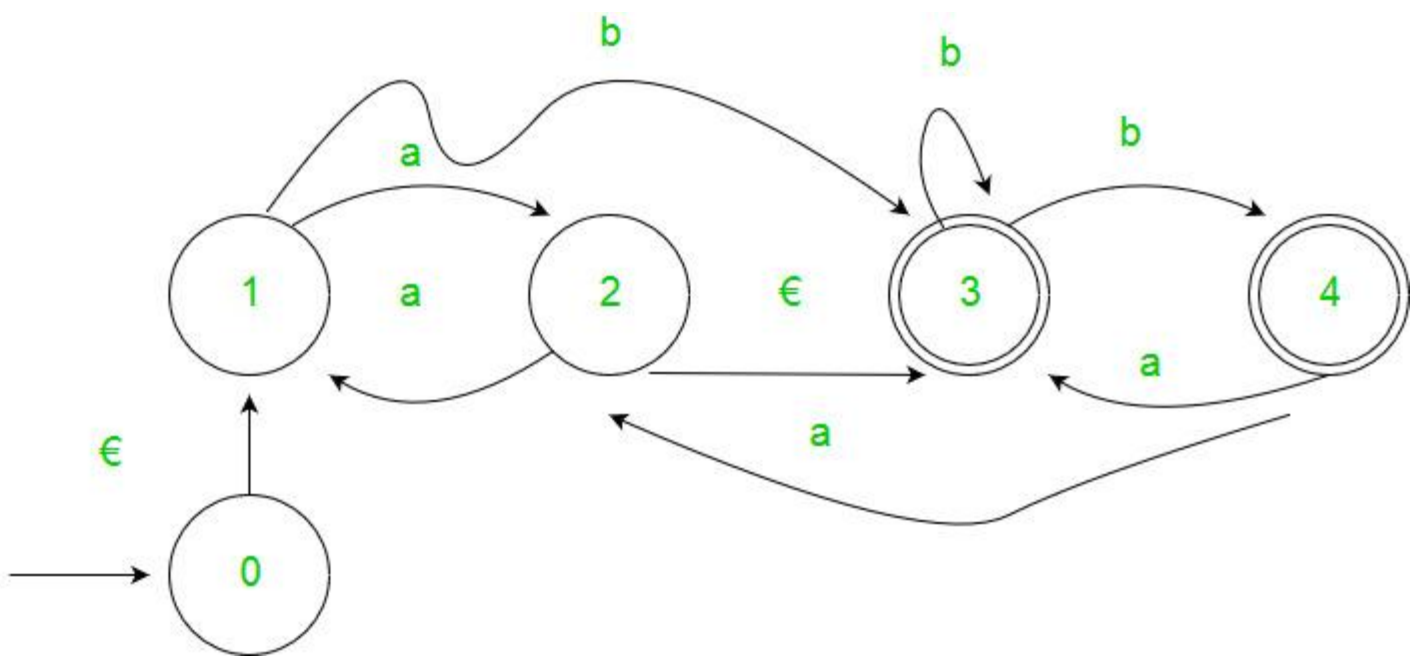
Örnek-1: (Devam)



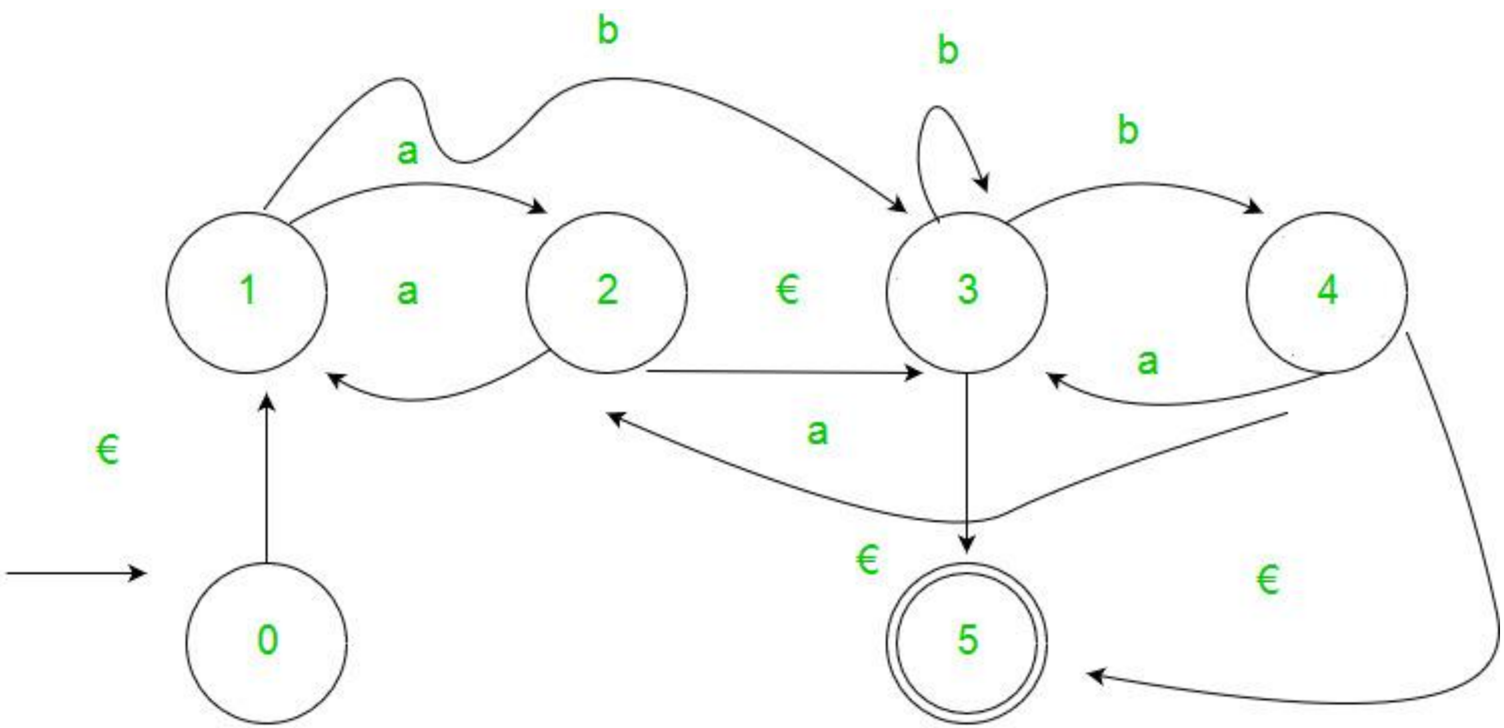
Örnek-2: birden fazla final state

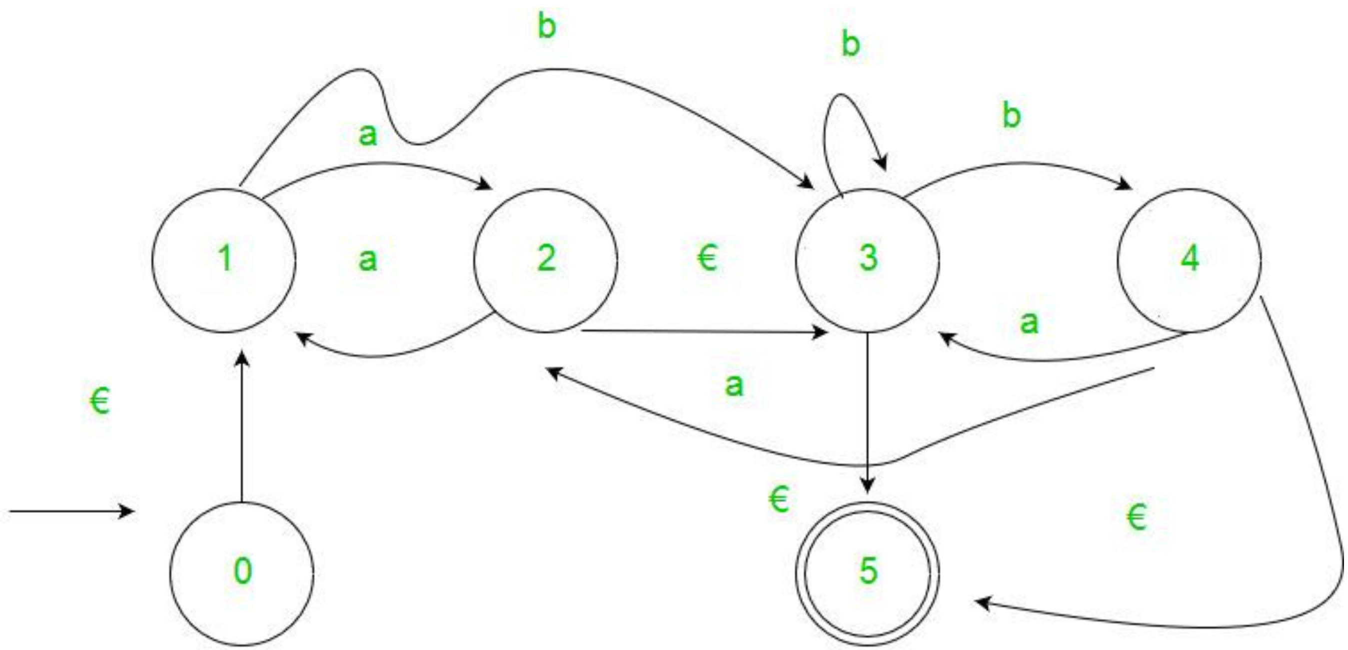


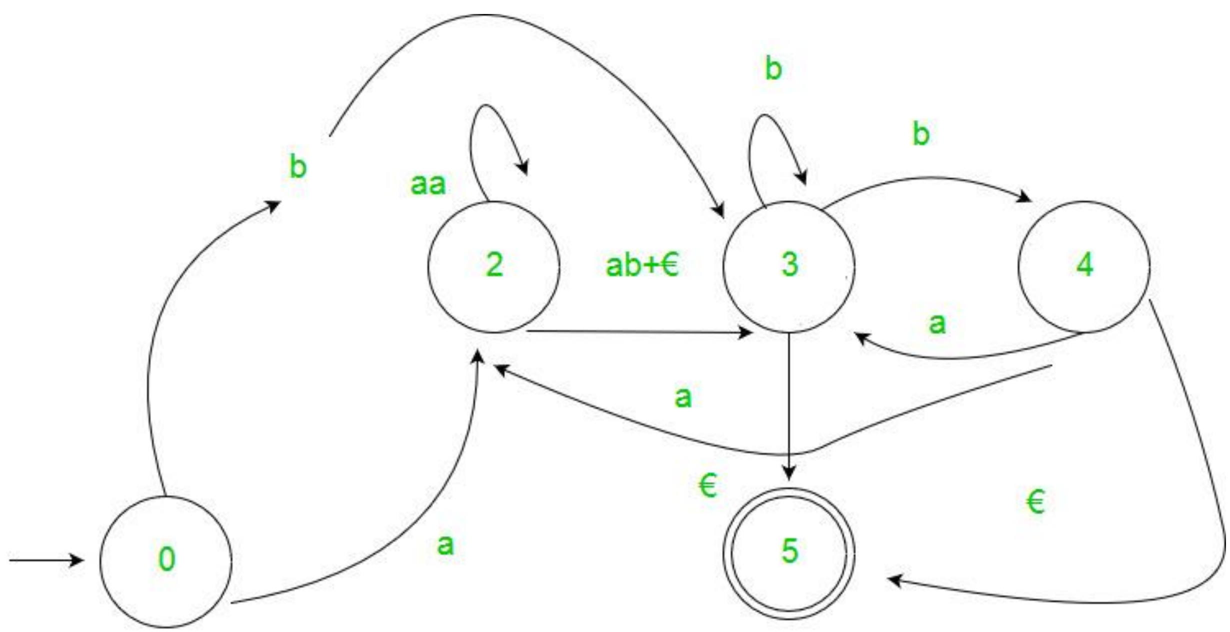
Step-1

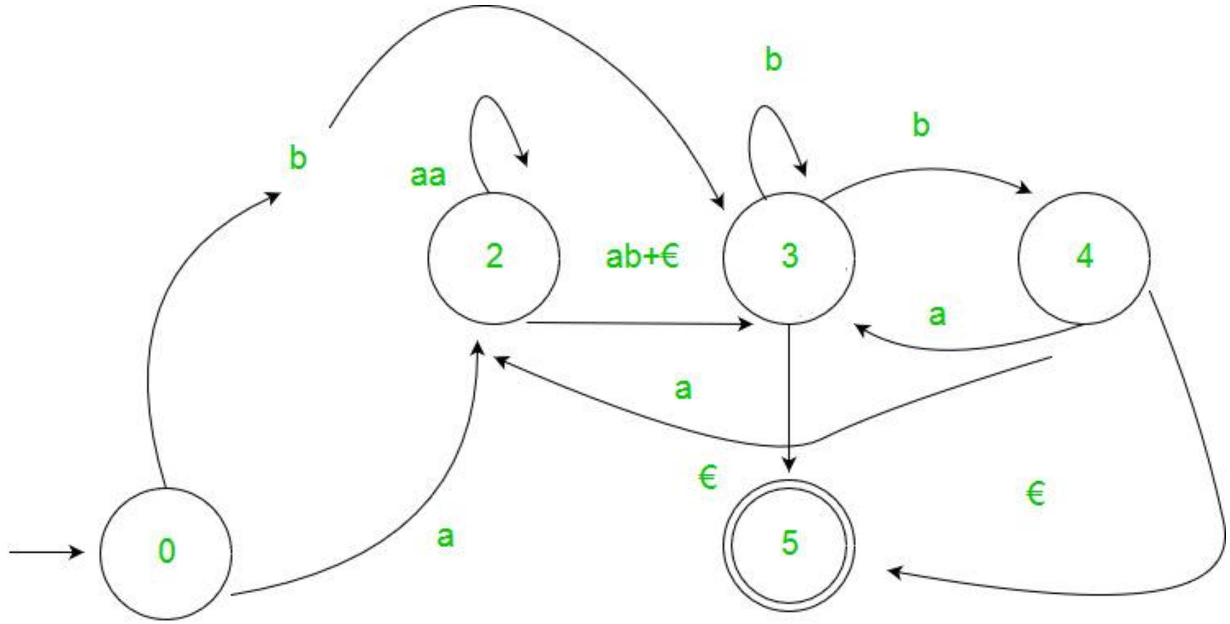


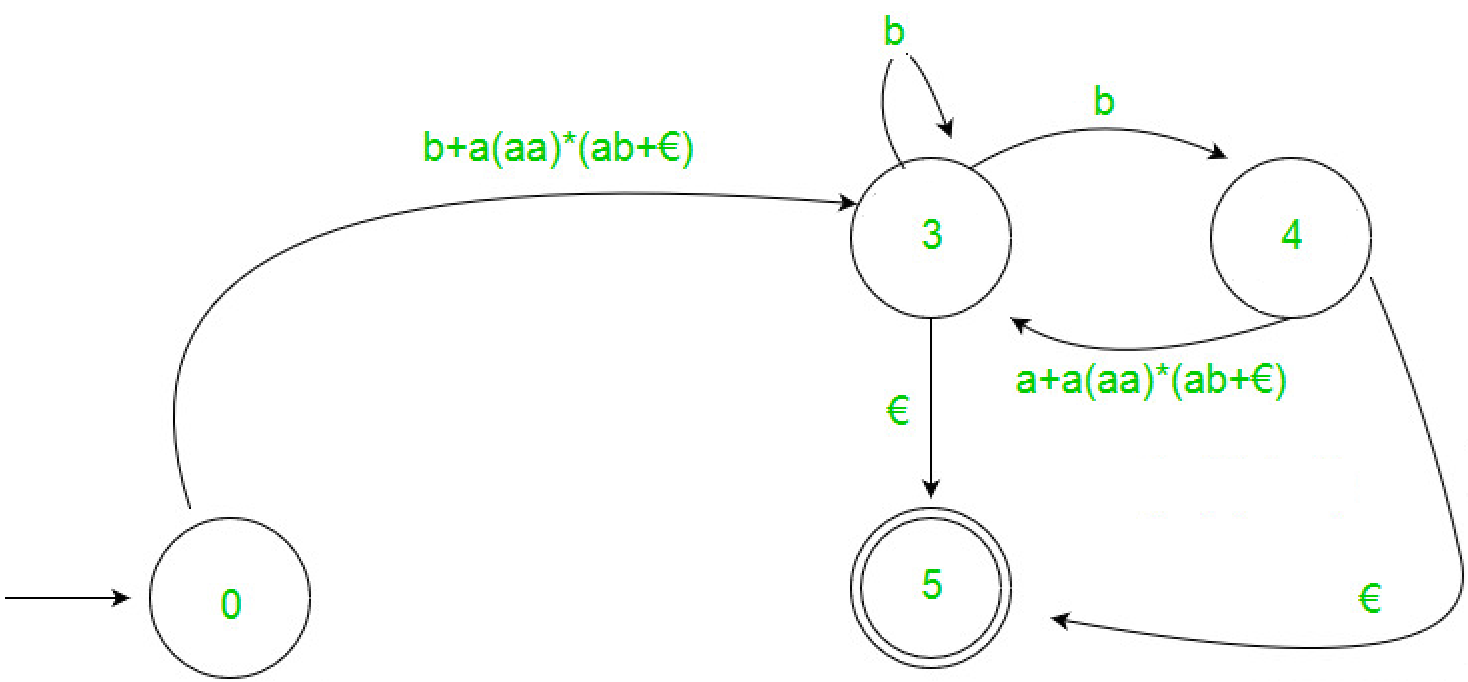
Step-2

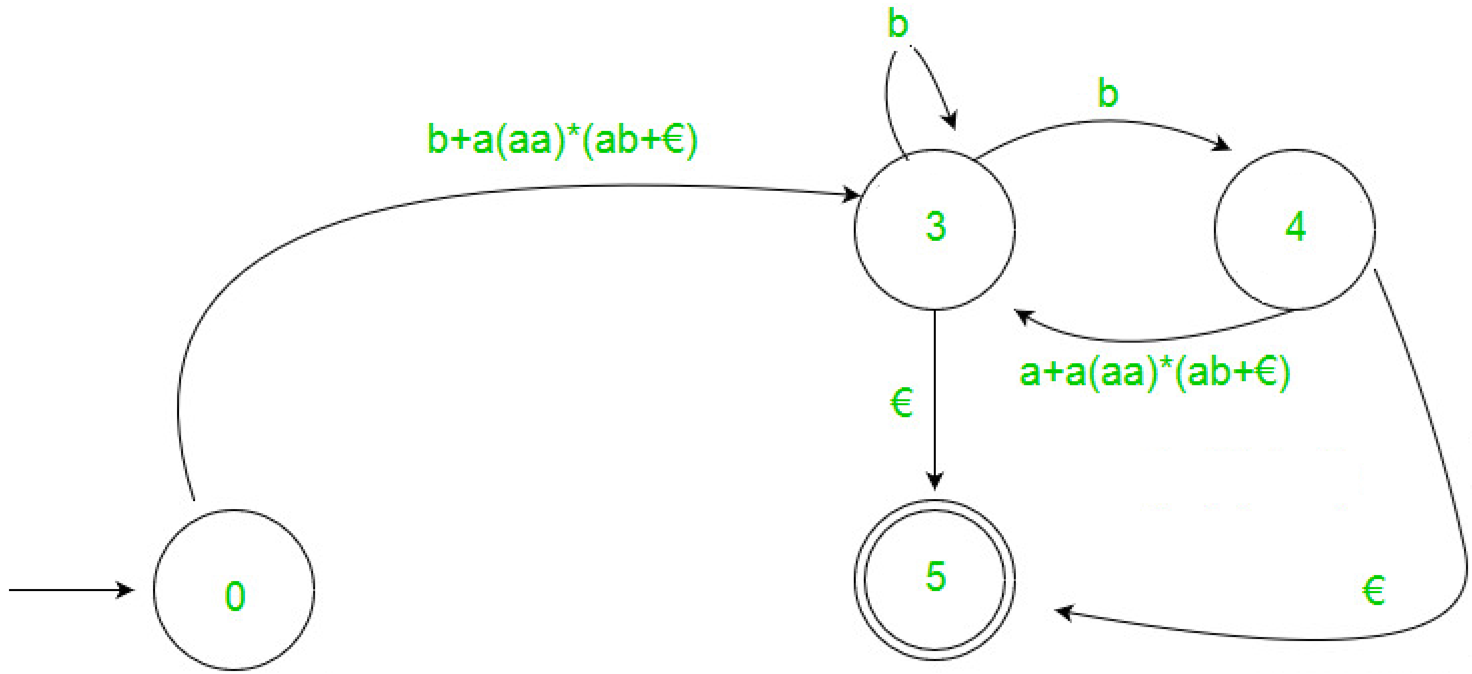


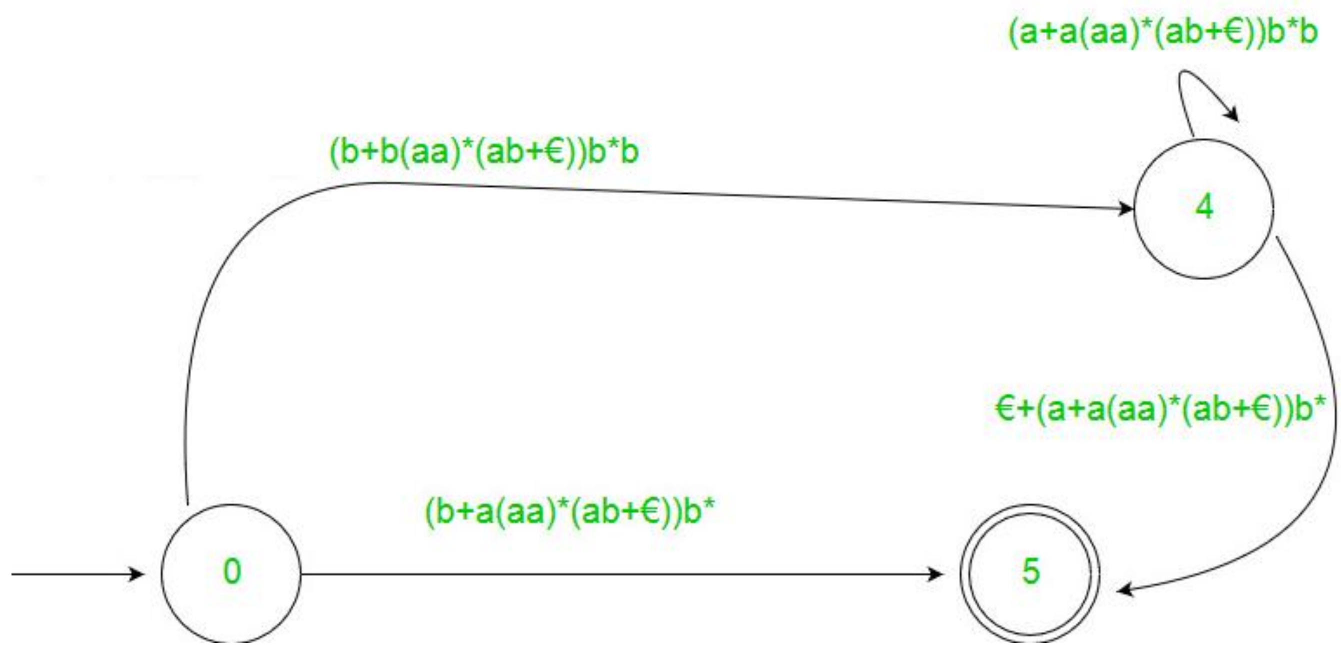

b
b
a
b
1
a
2
€
3
4
a
€
a
€
€
0
5

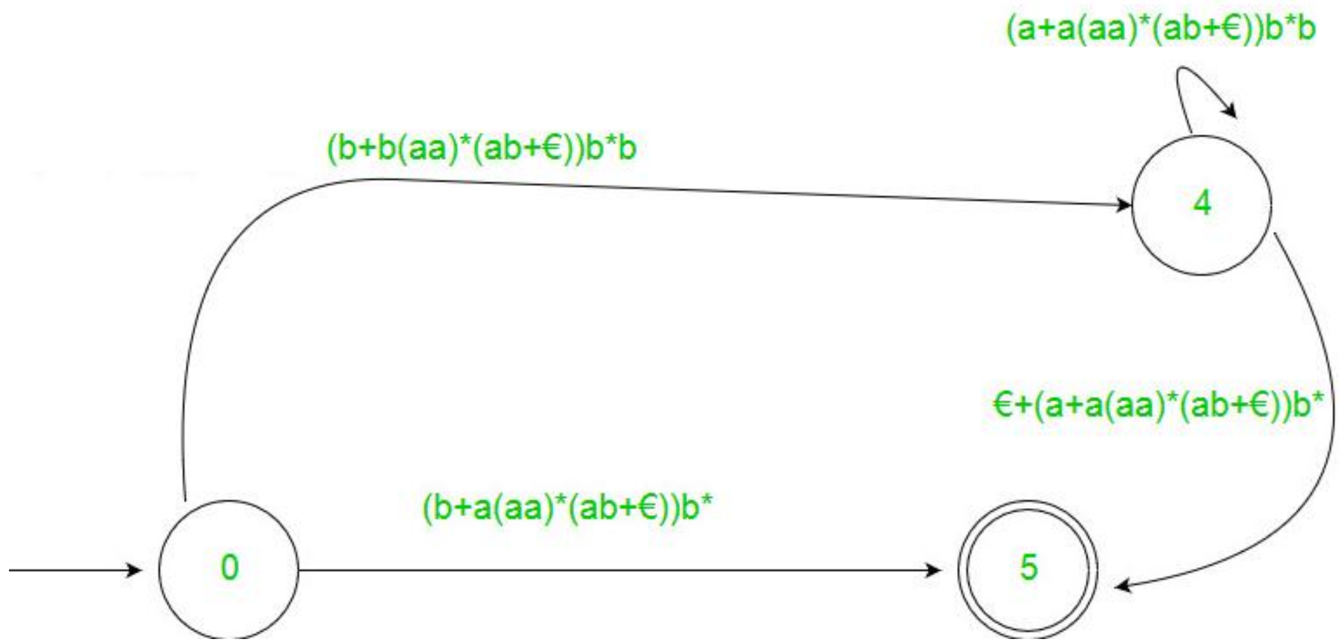

b
b
b
aa
2
ab+€
3
4
a
a
€
€
0
a
5


b
aa
b
b
2
ab+€
3
4
a
a
€
€
0
a
5


b
b
b+a(aa)*(ab+€)
3
4
a+a(aa)*(ab+€)
€
€
0
5


b
b+a(aa)*(ab+€)
b
3
4
a+a(aa)*(ab+€)
€
0
5
€


(a+a(aa)*(ab+€))b*b
(b+b(aa)*(ab+€))b*b
4
€+(a+a(aa)*(ab+€))b*
(b+a(aa)*(ab+€))b*
0
5



(b+a((aa)*(ab+€))b*+((b+a(aa)*(ab+€))b*b((a+a(aa)*
(ab+€))b*b)*(€+(a+a(aa)*(ab+€))b*)

ÇÖZÜM-2: ARDEN’s THEOREM

- **2. Arden’s Theorem –** P ve Q iki düzenli ifade olsun. Eğer P boş katar değilse, R = Q U RP denkleminin tek çözümü vardır ve R = QP* ‘dir.

Varsayımlar

- e-geçişler olmamalı (NFA olabilir fakat e-geçiş olamaz, e-geçiş varsa Çözüm-1 ile elde edebiliriz).
- Sadece tek başlangıç durumu olmalıdır.

İspat

- $R = Q \cup (Q \cup RP)P$ [$R = Q \cup RP$ değeri sağda yerine koyulursa]
- $= Q \cup QP \cup RPP$
- Özyinelemeli olarak tekrar tekrar yerine koyarsak:
- $R = Q \cup QP \cup QP^2 \cup QP^3 .....$
- $R = Q (\varepsilon \cup P \cup P^2 \cup P^3 \cup ....)$
- $R = QP^*$ [$P^* == (\varepsilon \cup P \cup P^2 \cup P^3 \cup ....)$]
- Böylece ispatlanmış olur.

ARDEN's THEOREM

Step-1 For getting the regular expression for the automata we first create equations of the given form for all the states

$\mathbf{q_1 = q_1w_{11}\ Uq_2w_{21}\ U\ldots Uq_nw_{n1}\ Ue}$ **(q_1 is the initial state)**

$\mathbf{q_2 = q_1w_{12}\ Uq_2w_{22}\ U\ldots Uq_nw_{n2}}$

.

.

.

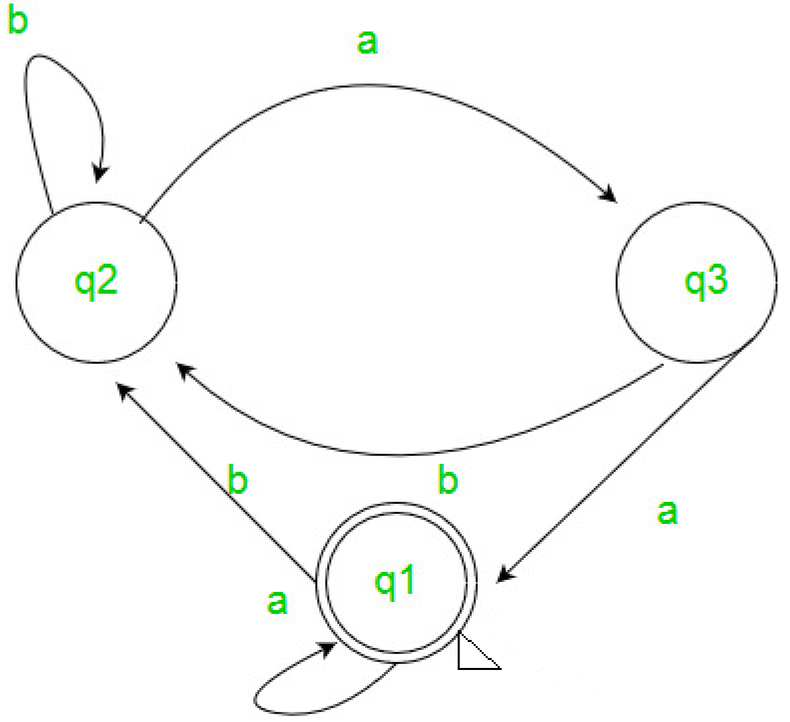
$\mathbf{q_n = q_1w_{1n}\ Uq_2w_{2n}\ U\ldots Uq_nw_{nn}}$

w_{ij} is the regular expression representing the set of labels of edges from q_i to q_j

- **Note –** For parallel edges there will be that many expressions for that state in the expression.

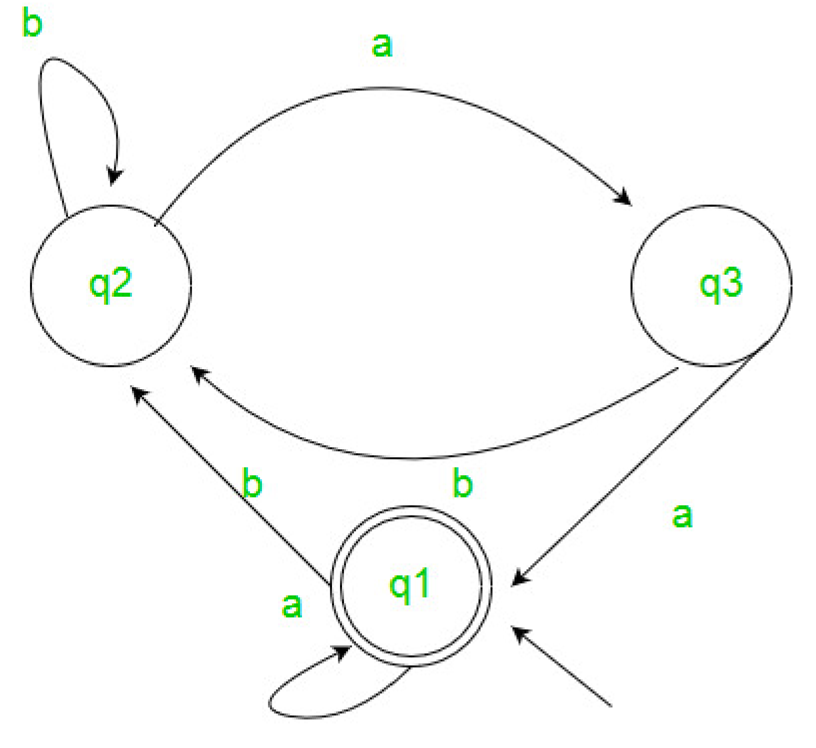
Step-2 Then we solve these equations to get the equation for q_i in terms of w_{ij} and that expression is the required solution, where q_i is a final state.

Örnek-1



Örnek-1

- Burada başlangıç durumu q_1 ve final durumu q_1.
  q_1, q_2, ve q_3 için denklemler şu şekilde yazılabilir
- $q_1 = q_1a \cup q_3a \cup e$ (empty string eklendi çünkü q_1 başlangıç durumu)
  $q_2 = q_1b \cup q_2b \cup q_3b$
  $q_3 = q_2a$



Aşağıdaki adımlarla çözülür:

- $q_2 = q_1b \cup q_2b \cup q_3b$

  $= q_1b \cup q_2b \cup (q_2a)b$ ($q_3=q_2a$ değerini yerine koyarsak)

  $= q_1b \cup q_2(b \cup ab)$

  $= q_1b\,(b \cup ab)^*$ (Arden's Theorem'i: $R = Q \cup RP = QP^*$)

  $q_1 = q_1a \cup q_3a \cup e$

  $= q_1a \cup q_2aa \cup e$ ($q_3=q_2a$ değerini yerine koyarsak)

  $= q_1a \cup q_1b(b \cup ab^*)aa \cup e$ (bulduğumuz q_2 değerini yerine koyarsak))

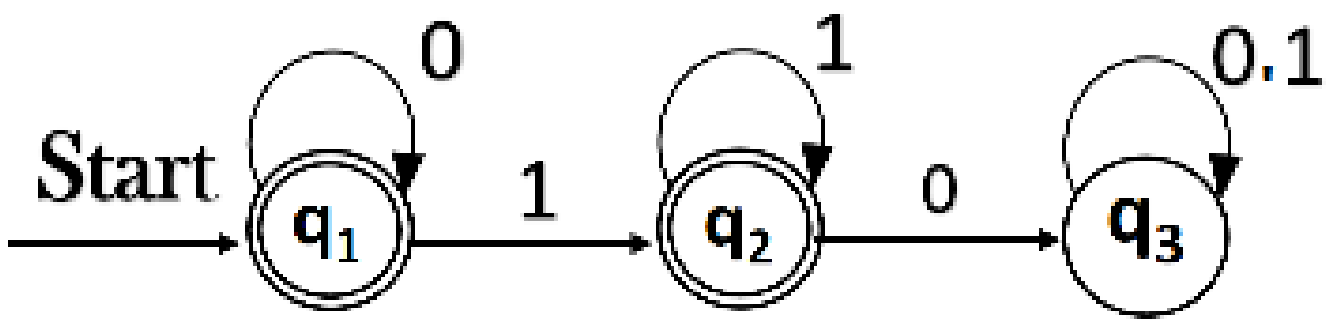
  $= q_1(a \cup b(b \cup ab)^*aa) \cup e$

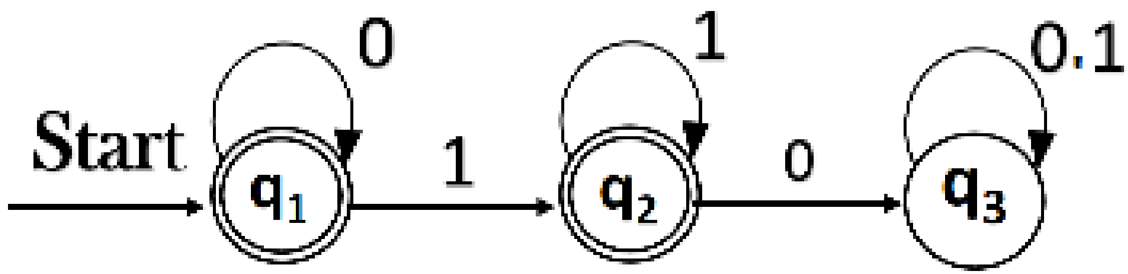
  $= e\,(a\cup b(b \cup ab)^*aa)^*$

  $= (a \cup b(b \cup ab)^*aa)^*$ (Arden's Theorem'i: $R = Q \cup RP = QP^*$)

Böylece düzenli ifade $(a \cup b(b \cup ab)^*aa)^*$ olarak elde edilir.

Örnek-2





$q_1 = q_1\, 0 \cup e$ (q_1 başlangıç durumu olduğu için e ekledik, giriş 0 q_1'e q_1'den geliyor,
State = source state of input × input coming to it şeklinde yazdığımız için bu şekilde elde ettik). Benzer şekilde

$q_2 = q_1\, 1 \cup q_2\, 1$

$q_3 = q_2\ 0 \cup q_3\, (0 \cup 1)$ (final durumu q_1 ve q_2 olduğu için bunları çözmeye çalışıyoruz.)

$q_1 = q_1\, 0 \cup e = e \cup q_1\, 0$ (Bu $R = Q \cup RP$ şeklinde $\rightarrow$ $R = QP^*$: Arden Teoreminden).

$R = q_1$, $Q = e$, $P = 0$ dersek:

$q_1 = e.(0)^* = 0^*$ ($e.R^* = R^*$ olduğu için)

Bunu q_2'de yerine koyarsak,

$q_2 = 0^*\, 1 \cup q_2\, 1$

$q_2 = 0^*\, 1\, (1)^*$ ($R = Q \cup RP \rightarrow Q\, P^*$)

Bu durumda $r = q_1 \cup q_2$ olduğundan

- $r = q_1 \cup q_2 = 0^* \cup 0^*\, 1.1^*$ $r = 0^* \cup 0^*\, 1^+$ ($1.1^* = 1^+$ olduğu için) elde edilir