

Otomata Teorisi (BİL 2114)

Fırat İsmailoğlu

Hafta 5: Düzenli İfadeler (II. Bölüm)



Hafta 5

Plan

1. Düzenli İfadeler ve Sonlu Otomatanın Birbirine Denkliği
2. Düzenli İfadelerden Sonlu Otomataya Geçiş
3. Sonlu Otomatadan Düzenli İfadeye Geçiş
 - 3.1 Genelleştirilmiş Nondeterministik Sonlu Otomata (GNSO)
4. GNSO'nun Formal Gosterimi
5. Dönüştür Algoritması

Düzenli İfadeler ve Sonlu Otomatanın Denkliği

Düzenli İfade \equiv Sonlu Otomata

1. Bir düzenli ifade tarafından tanınan her dil aynı zamanda bir sonlu otomata tarafından da tanımlanır.
 2. Bir sonlu otomata tarafından tanınan her dil aynı zamanda bir düzenli ifade tarafından da tanımlanır.
1. durumu düzenli ifadelerden sonlu otomataya geçiş; 2. durumu sonlu otomatadan düzenli ifadeye geçiş başlığında işleyeceğiz.



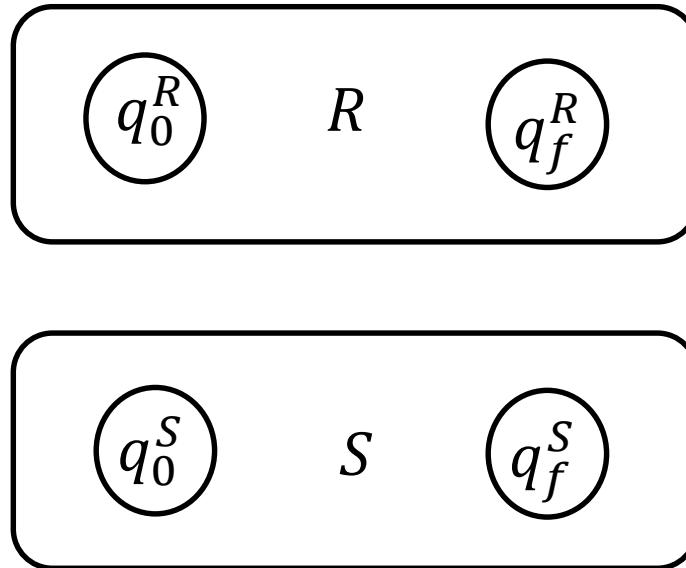
Düzenli İfadelerden Sonlu Otomataya Geçiş

Düzenli İfadelerde üç temel operatör vardı: birleşim, bitleştirme ve yıldız operatörü. Bunların NSO'da gösterimi şöyledir.

R ve S iki düzenli ifade olsun.

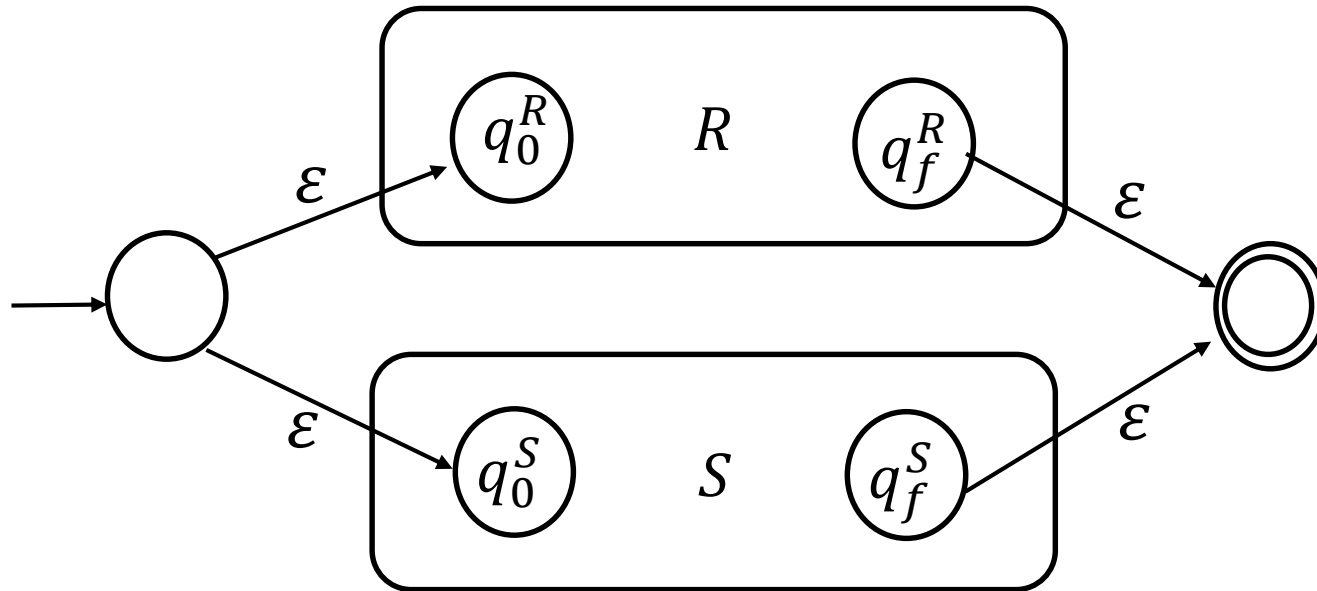
R 'nin başlangıç ve kabul durumunu sırasıyla q_0^R ve q_f^R olsun.

S 'nin başlangıç ve kabul durumunu sırasıyla q_0^S ve q_f^S olsun.



Düzenli İfadelerden Sonlu Otomataya Geçiş

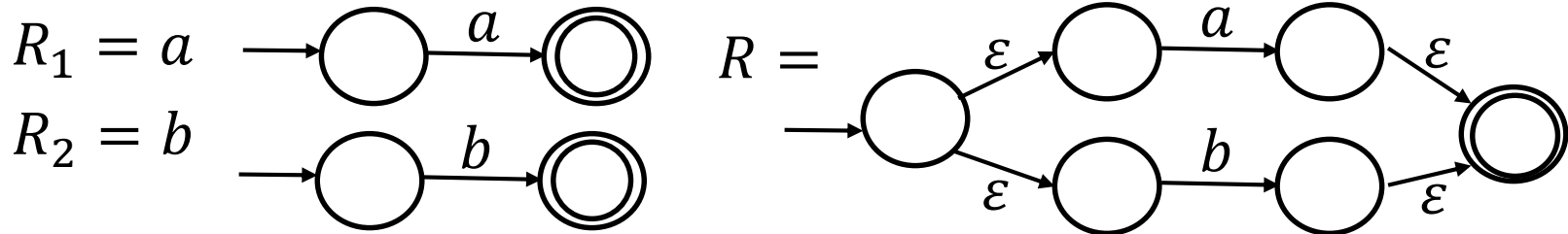
I. Birleşim



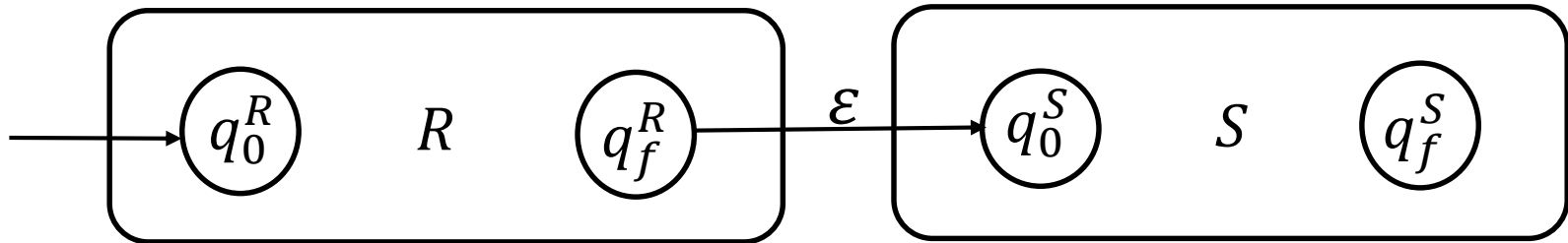
Yeni bir başlangıç ve yeni bir final durumu oluşturuyoruz. Ve bu durumları ϵ okları ile R 'nin ve S 'nin başlangıç ve final durumlarına bağlıyoruz.

Düzenli İfadelerden Sonlu Otomataya Geçiş

ör. $R = a \cup b$



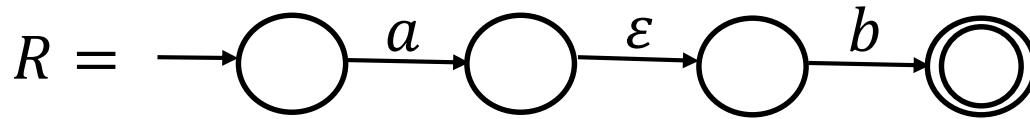
2. Bitiştirme



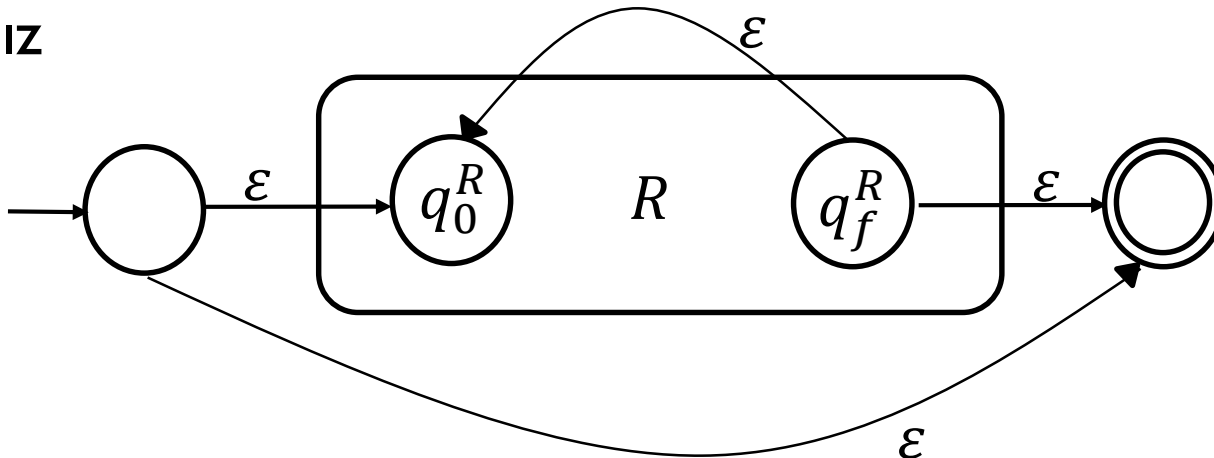
İlk sonlu otomatanın başlangıç durumu tamamın başlangıç durumu, ikinci otomatanın kabul durumu tamamın kabul durumu olur. İki otomat birbirine ϵ ile bağlanır.

Düzenli İfadelerden Sonlu Otomataya Geçiş

ör. $R = ab$



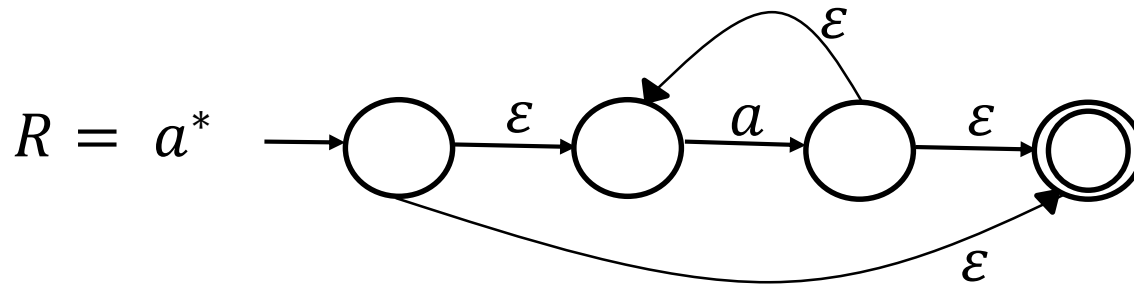
3. Yıldız



Düzenli İfadelerden Sonlu Otomataya Geçiş

- i. Yeni başlangıç ve kabul durumları oluşturulur ve bunlar ε okları ile önceki başlangıç ve kabul durumlarına bağlanır.
- ii. Yeni başlangıç durumundan yeni kabul durumuna bir ε oku eklenerek yıldız operatörünün ε (boş kelimeyi) kabul etmesi sağlanır. ($a^* = \{\varepsilon, a, aa, aaa, \dots\}$)
- iii. Eski kabul durumundan eski başlangıç durumuna bir ε oku eklenerek düzenli ifadenin tekrarlarının kabulü sağlanır.

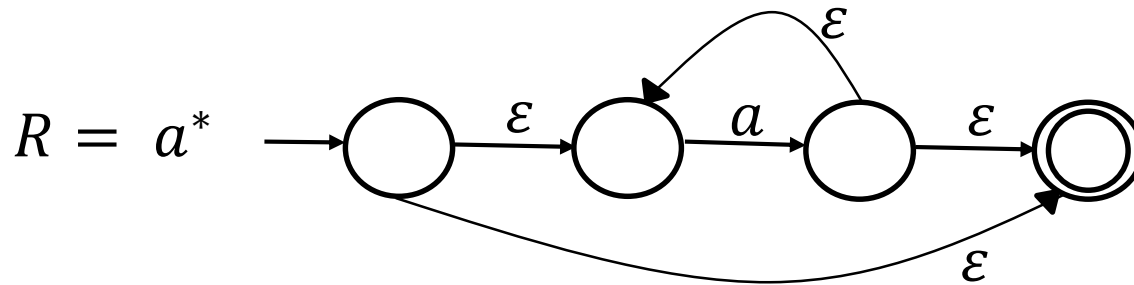
ör.



Düzenli İfadelerden Sonlu Otomataya Geçiş

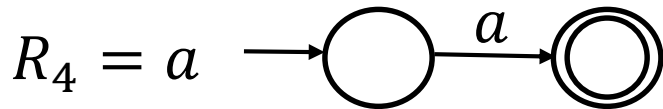
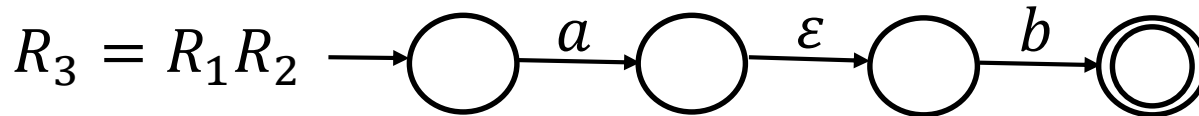
- i. Yeni başlangıç ve kabul durumları oluşturulur ve bunlar ε okları ile önceki başlangıç ve kabul durumlarına bağlanır.
- ii. Yeni başlangıç durumundan yeni kabul durumuna bir ε oku eklenerek yıldız operatörünün ε (boş kelimeyi) kabul etmesi sağlanır. ($a^* = \{\varepsilon, a, aa, aaa, \dots\}$)
- iii. Eski kabul durumundan eski başlangıç durumuna bir ε oku eklenerek düzenli ifadenin tekrarlarının kabulü sağlanır.

ör.

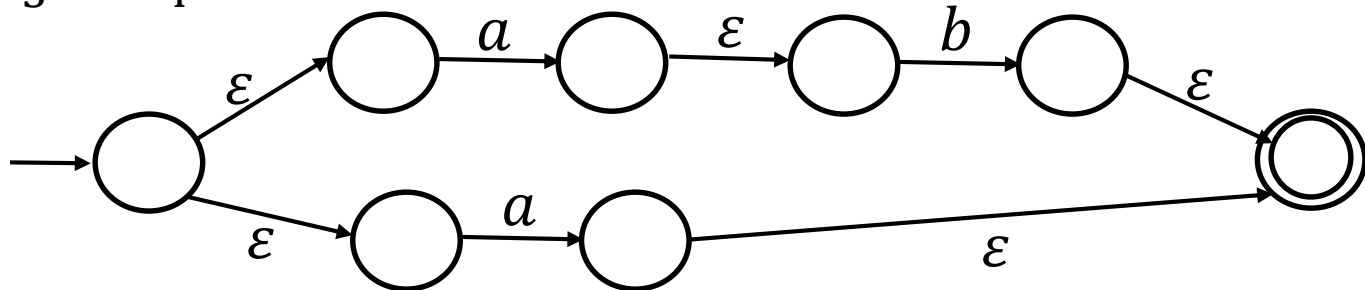


ör. $R = (ab \cup a)^*$ düzenli ifadesinin denk olduğu sonlu otomatayı tasarlayınız.

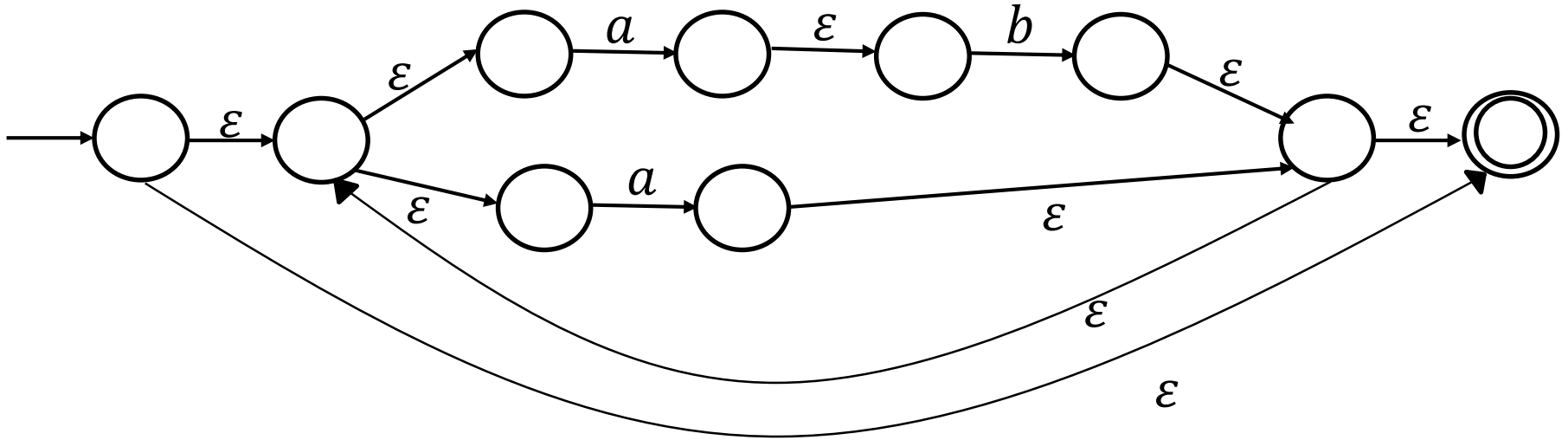
$$R_1 = a, R_2 = b, R_3 = R_1 R_2, R_4 = a, R_5 = R_3 \cup R_4, R_6 = R_5^*$$



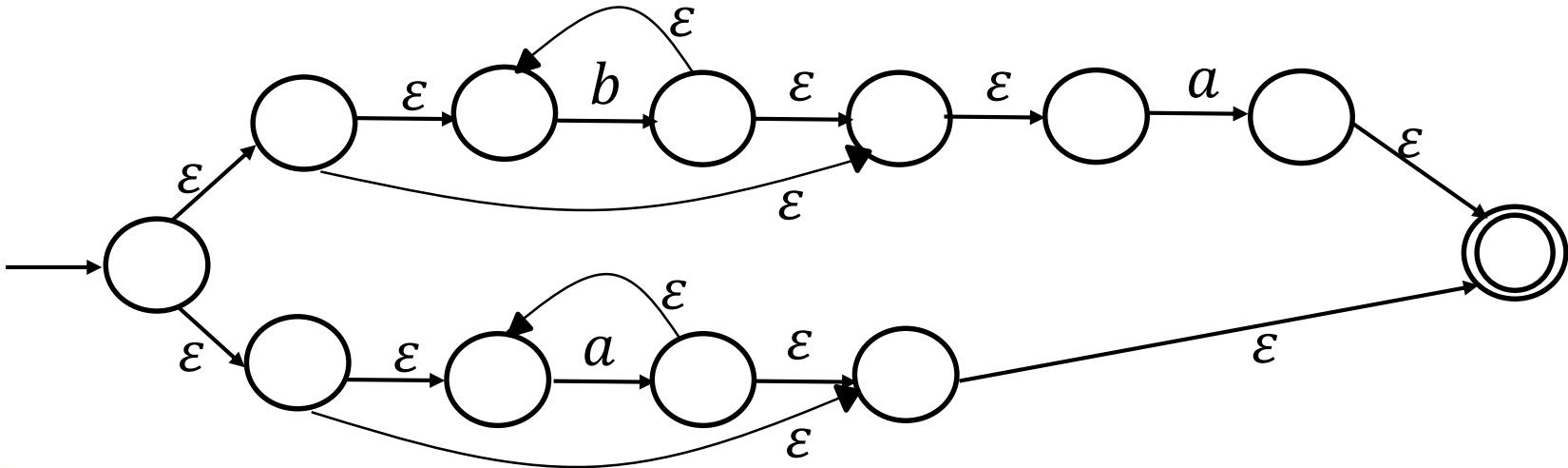
$$R_5 = R_3 \cup R_4$$



ör. $R_6 = R_5^* = R = (ab \cup a)^*$



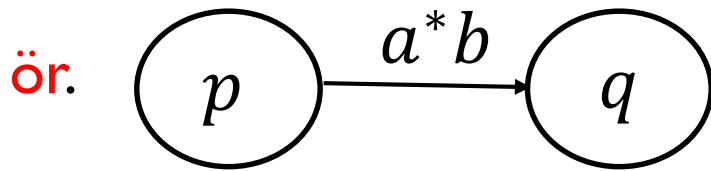
ör. $R = b^*a \cup a^*$



Sonlu Otomatadan Düzenli İfadelere Geçiş

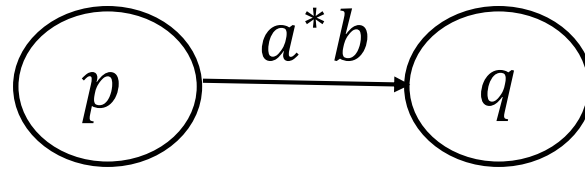
Verilen bir sonlu otomaya denk bir düzenli ifade bulmak için öncelikle *Genelleştirilmiş Nondeterministik Sonlu Otomata* (GNSO) kavramını bilmemiz gerekir.

GNSO'da şimdiye dek gördüğümüz NSO'ların aksine iki durum arasındaki geçiş bir harfle değil; bir düzenli ifade ile olur. Böylece verilen bir kelimeyi teker teker, harf harf değil 'blok' olarak okuruz.



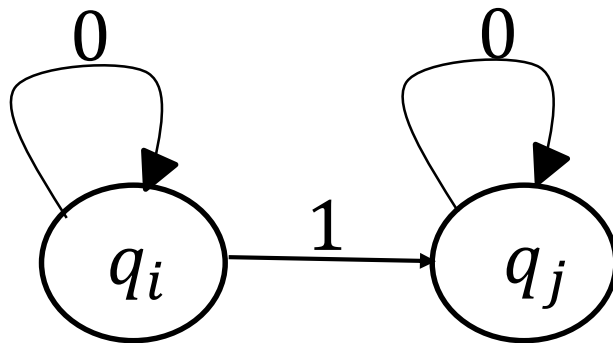
Burada p durumuna vardığımızda, verilen kelimenin kalan kısmının herhangi bir uzunluktaki öneki a^*b ile eşleşiyorsa q durumuna geçeriz.



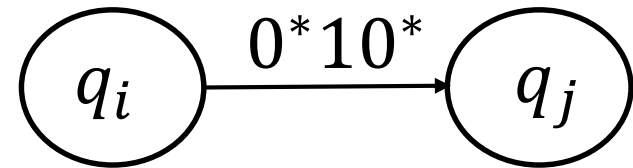


örneğin diyelim ki p ye vardığımızda okuduğumuz kelimenin kalan kısmı $aaaabccc$ olsun. Bu durumda $aaaab$ bloğunu harcayarak q ya geçeriz ve elimizde kelimenin ccc kısmı kalır.

ör.



NSO



GNSO

Sonlu Otomatayı Düzenli İfadeye Çevirme

Verilen bir sonlu otomata bir düzenli ifadeye çevrilirken:

1. Önce genelleştirilmiş sonlu otomataya,
2. Ardından düzenli ifadeye çevrilir.

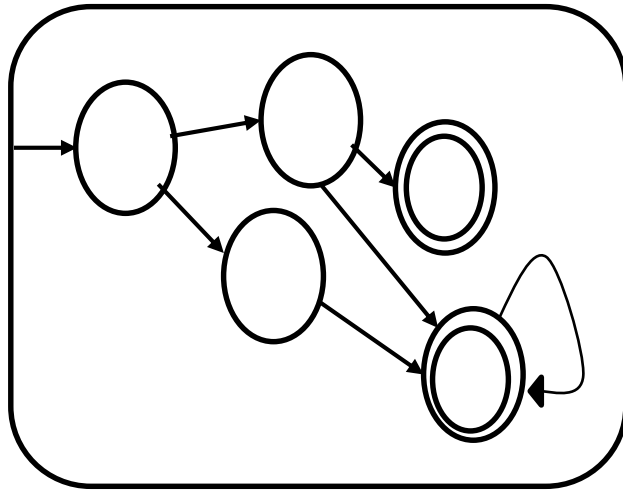


Sonlu otomata genelleştirilmiş sonlu otomataya çevrilirken:

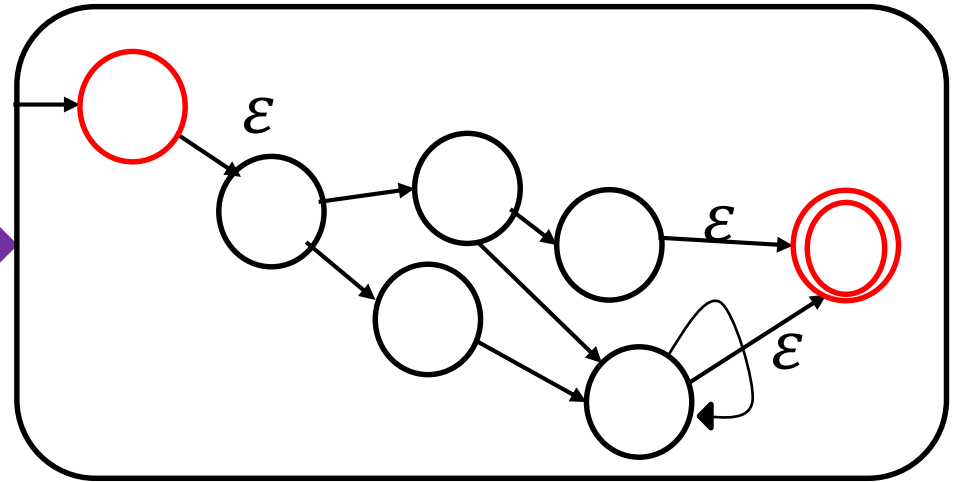
Yeni bir başlangıç ve yeni bir kabul durumu eklenir. Bu yeni başlangıç durumu eski başlangıç durumuna ε ile eklenir.

Önceki kabul durumları ise (ki bunlar 1 den fazla olabilir) yeni kabul durumuna ε ile eklenir.

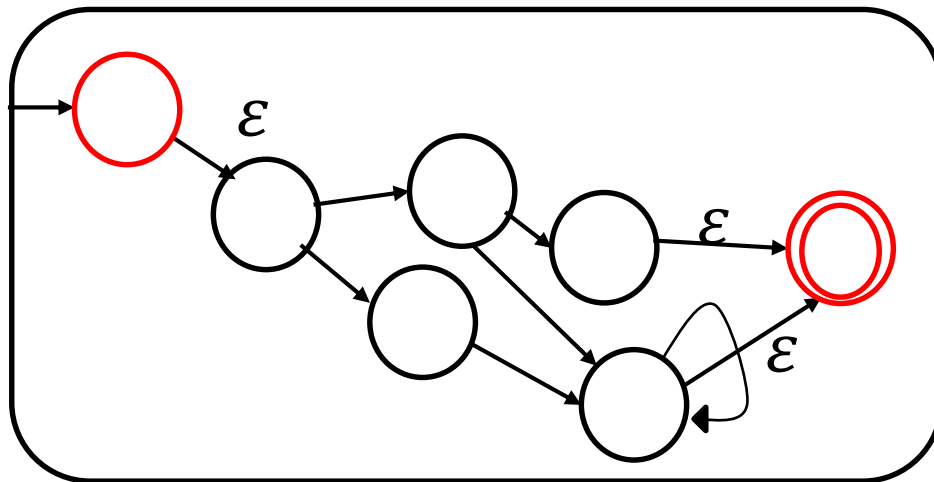
Sonlu Otomatayı Düzenli İfadeye Çevirme



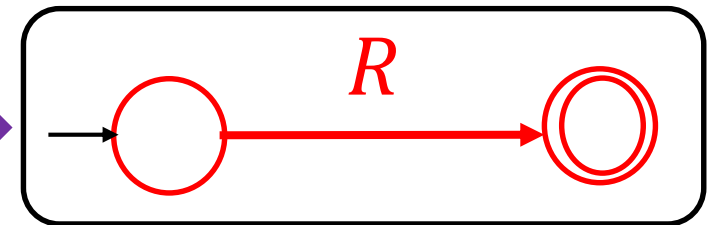
NSO



GNSO



GNSO

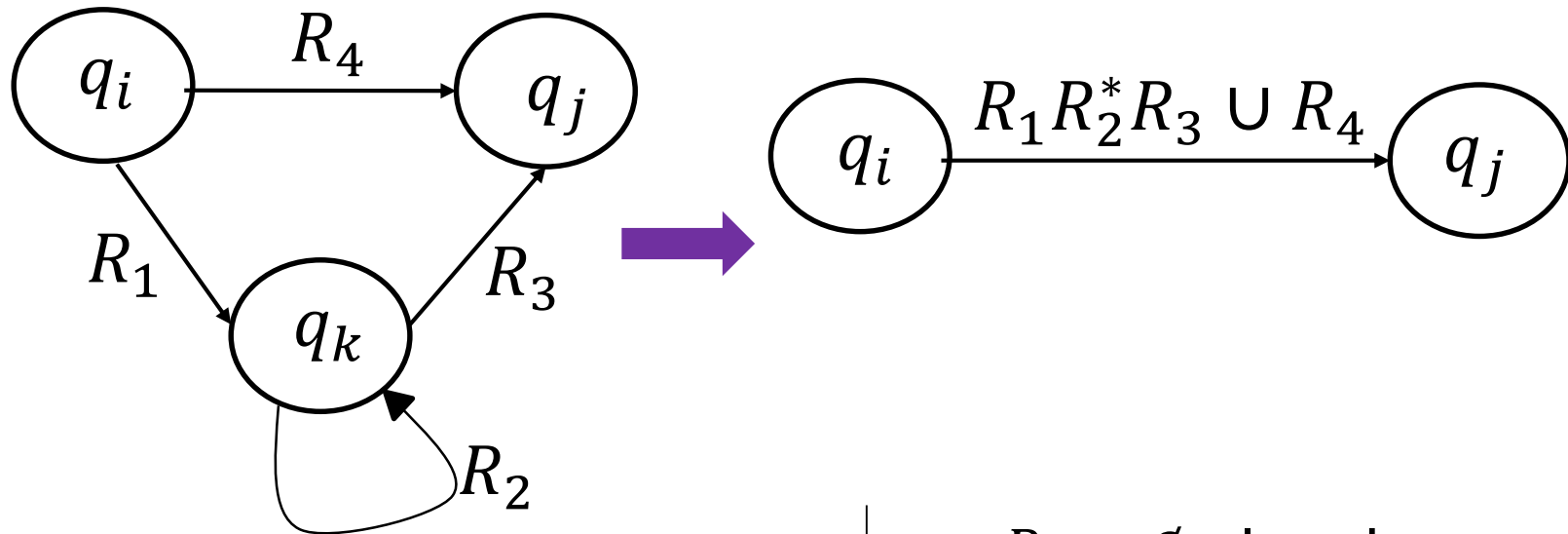


Aradığımız
Düzenli İfade

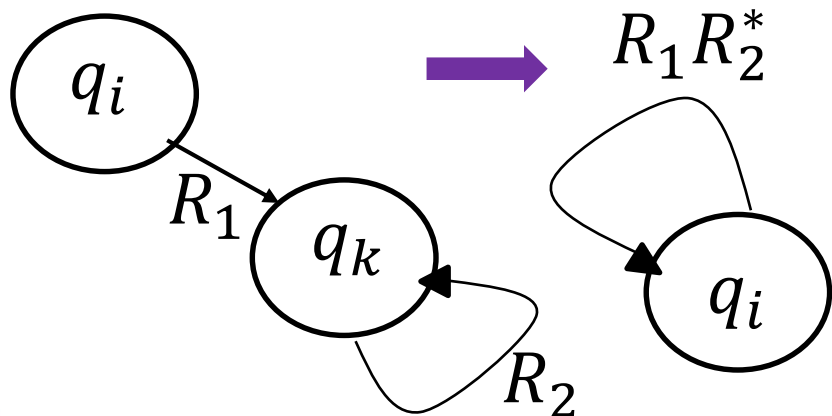


Durum Eleme Yöntemi ile GNSO'yu R'ye Çevirme:

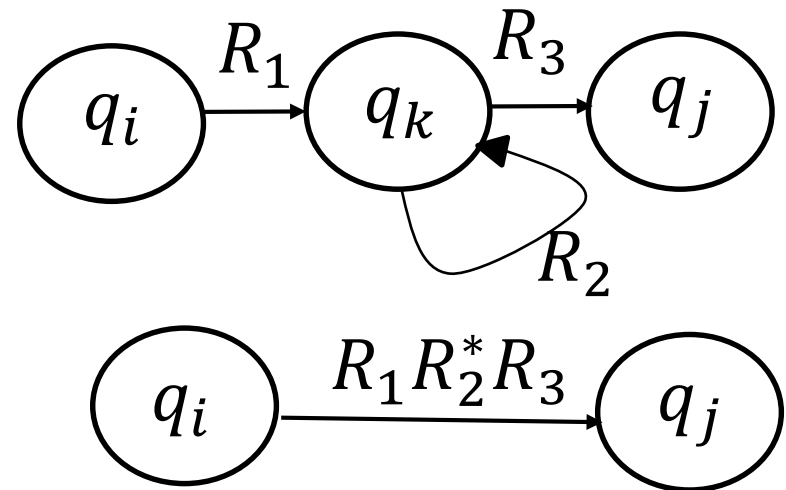
Diyelimki q_k 'yı elemek istiyoruz:



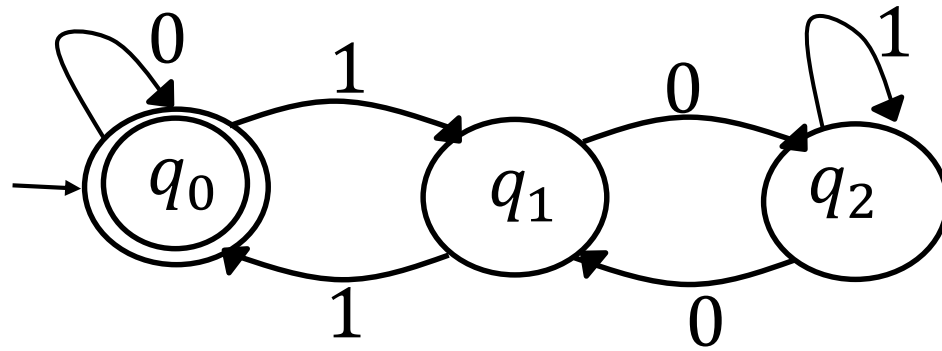
$q_i = q_j$ olma durumu:



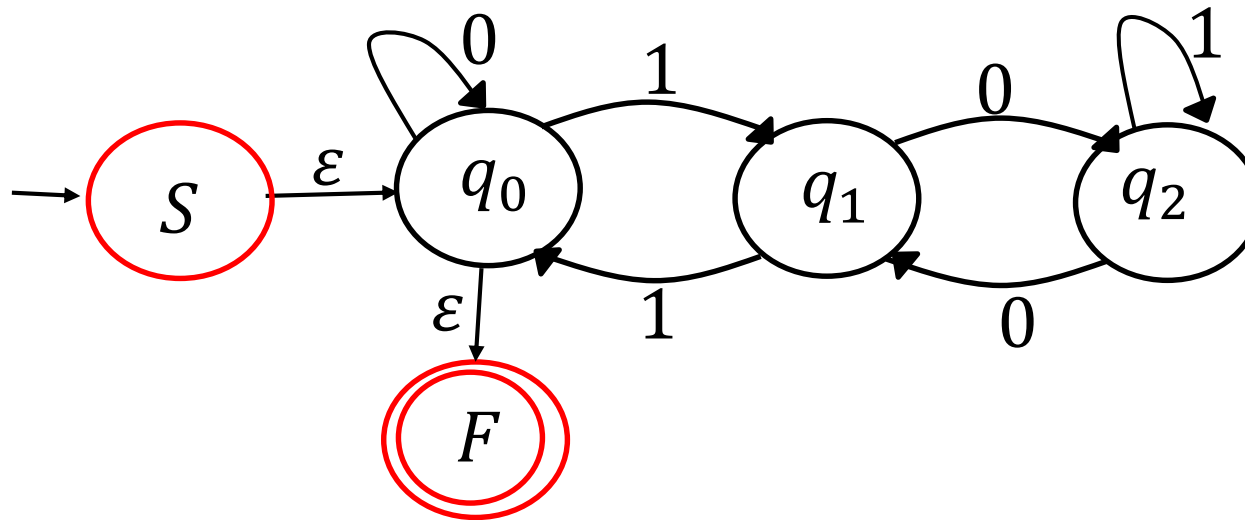
$R_4 = \emptyset$ olma durumu:



or.

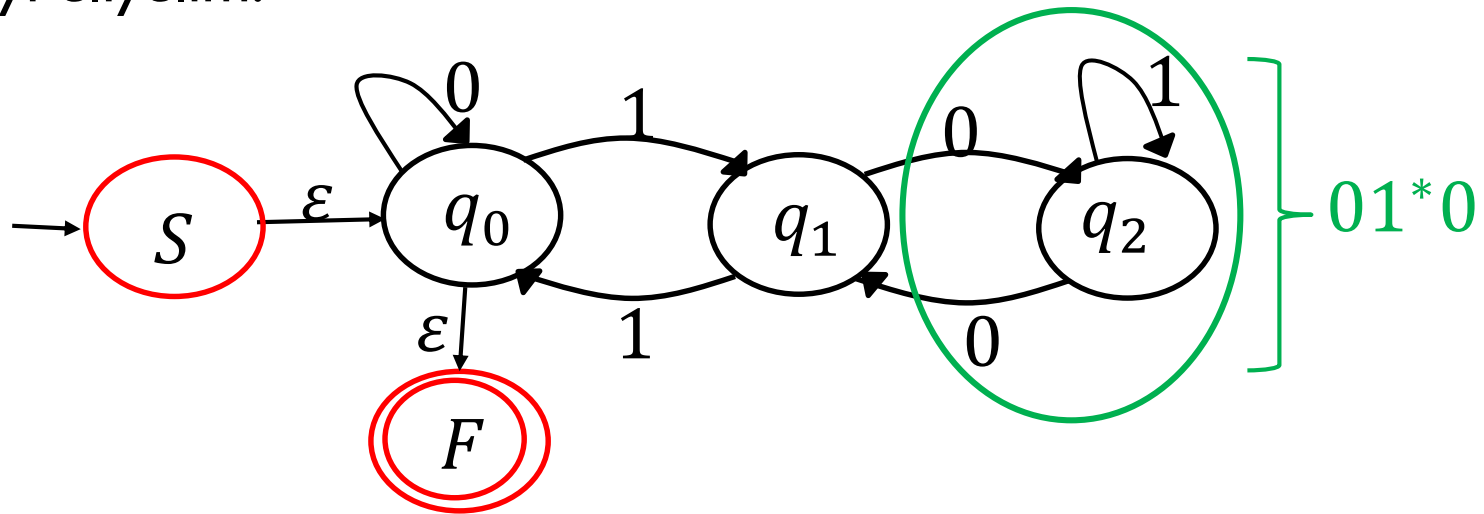


Verilen
NSO

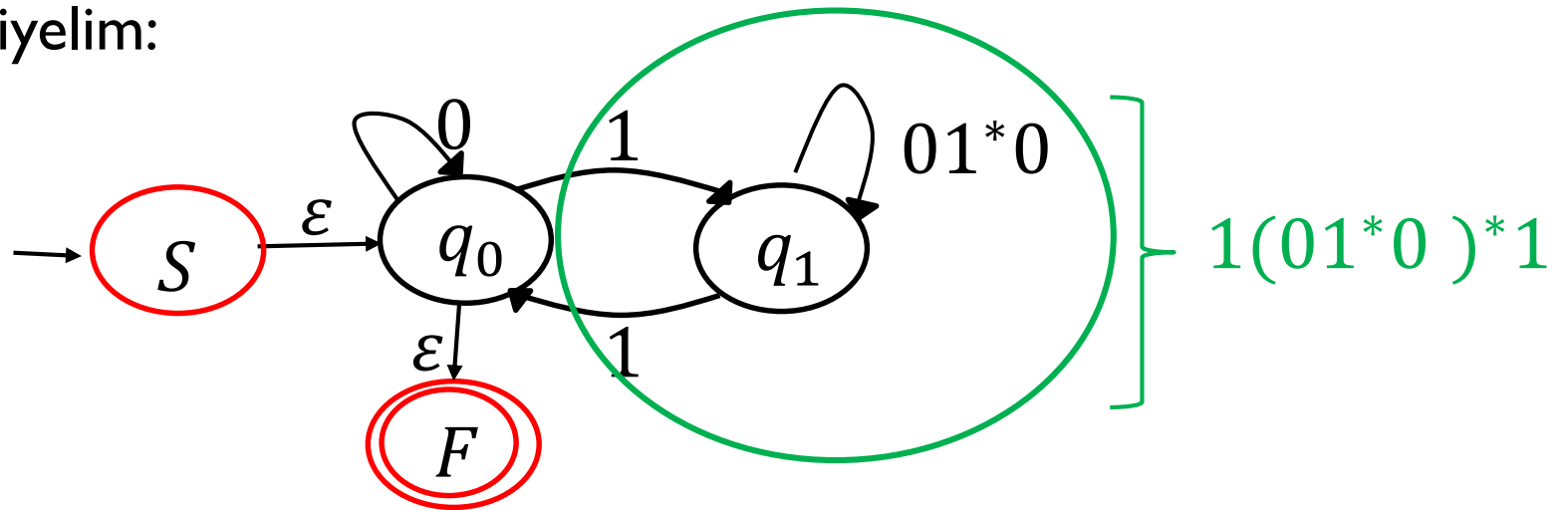


Elde edilen
GNSO

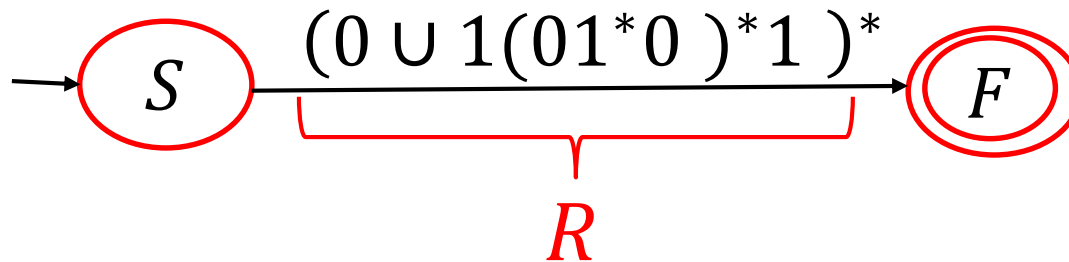
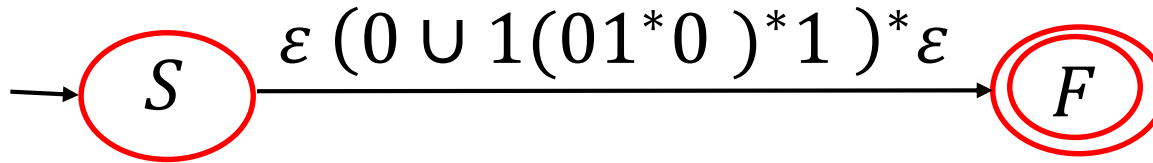
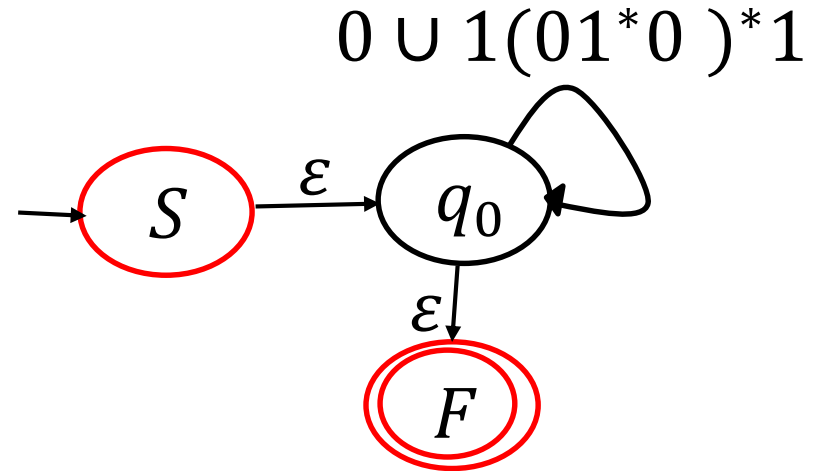
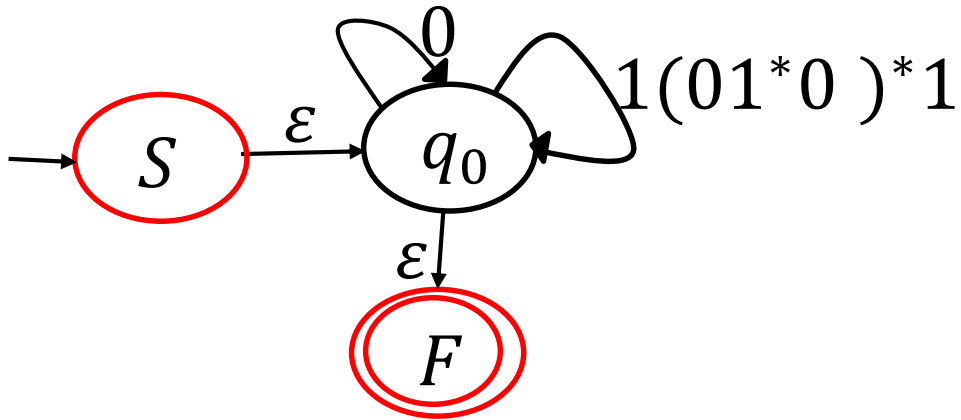
q_2 'yi eliyelim:



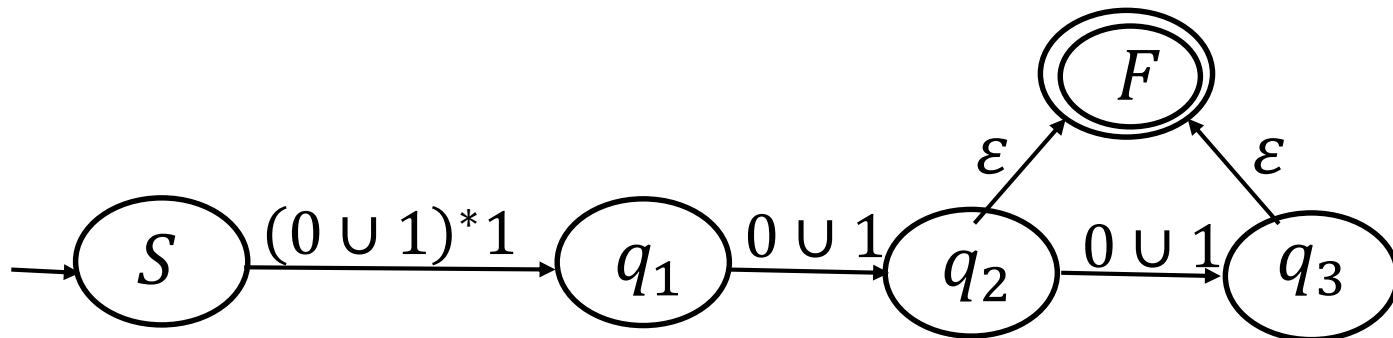
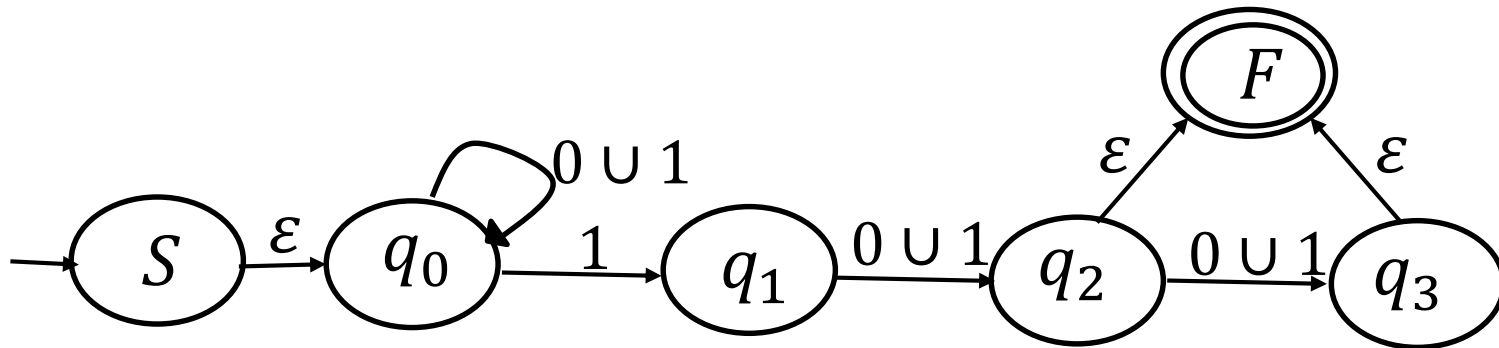
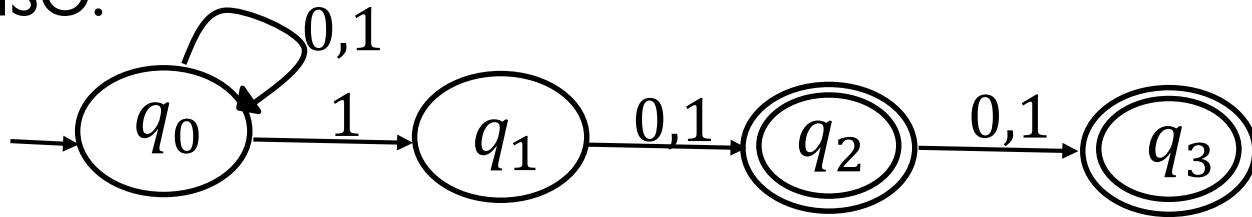
q_1 'yi eliyelim:

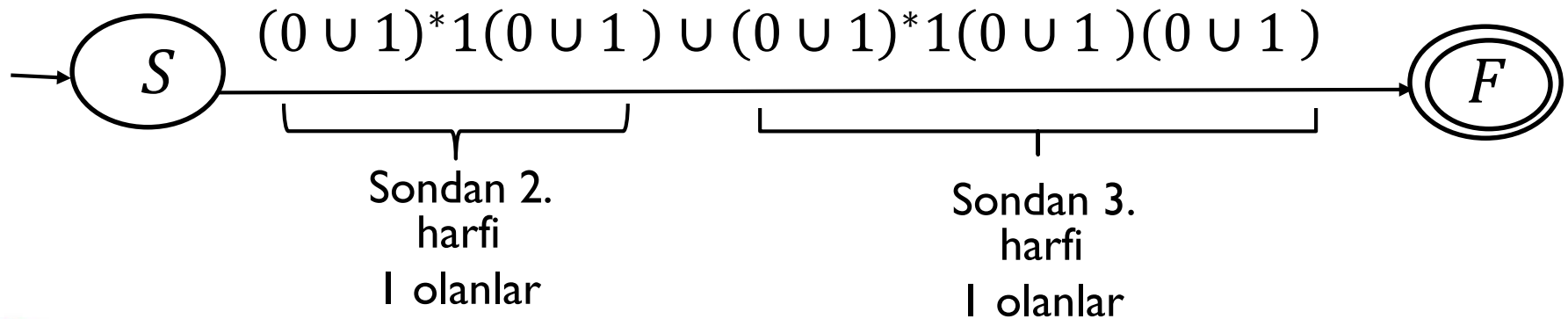
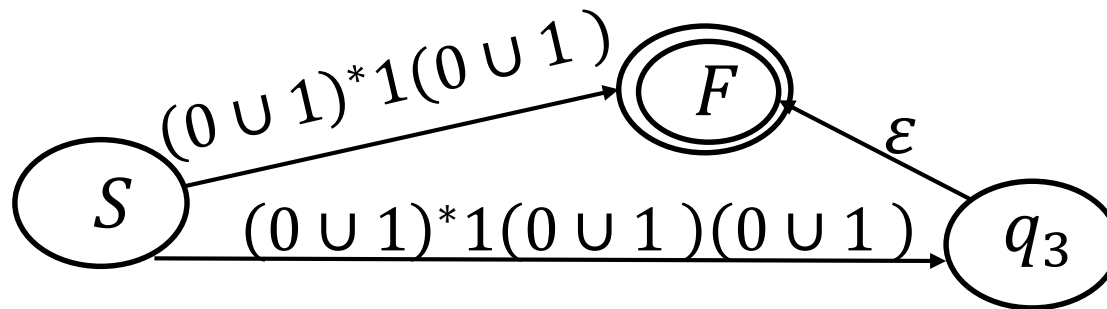
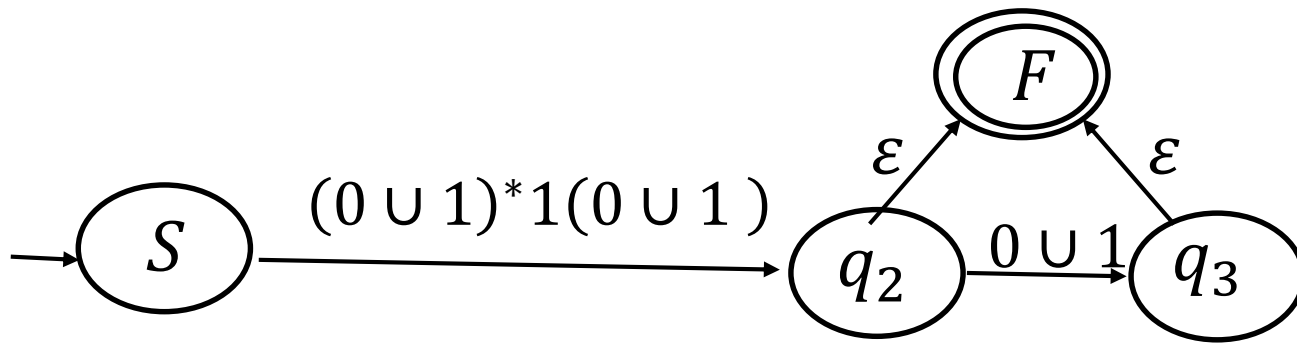


q_0 'i eliyelim:



or. Sondan üçüncü yada sondan ikinci harfi 1 olan kelimeleri kabul eden NSO:





Bir GNSO'nun Formal Tanımı:

Bir GNSO 5-li sıradır ve $G = (Q, \Sigma, \delta, q_{start}, q_{final})$ ile gösterilir.

1. Q tüm durumları içeren sonlu bir kümedir,
2. Σ kullandığımız harfleri (inputları) içeren alfabadir,
3. $\delta: (Q - \{q_{final}\}) \times (Q - \{q_{start}\}) \rightarrow R$ geçiş fonksiyonudur
4. $q_{start} \in Q$ başlangıç durumudur,
5. $q_{final} \in Q$ final/kabul durumudur.

Not 1. Geçiş fonksiyonu bir duruma değil, bir düzenli ifadeye döner.

Not 2. Final ve start durumları arasında geçiş fonksiyonu tanımlı değildir.

Not 3. Yalnızca 1 tane final durumu var.



Dönüştür (G) algoritması:

giriş: $G = (Q, \Sigma, \delta, q_{start}, q_{final})$

çıkış: R düzenli ifadesi

1. $k := |Q|$ // durum sayısı
2. while $k > 2$:
 - seç $q_{ölü} \in Q - \{q_{start}, q_{final}\}$
 - $Q := Q - \{q_{ölü}\}$ // Q 'yu güncelle
 - her $q_i \in Q - \{q_{final}\}$ ve her $q_j \in Q - \{q_{start}\}$ için
 - $\delta(q_i, q_j) := (R_1)(R_2)^*(R_3) \cup (R_4)$
 - öyleki
 - $R_1 = \delta(q_i, q_{ölü}), R_2 = \delta(q_{ölü}, q_{ölü}), R_3 = \delta(q_{ölü}, q_j), R_4 = \delta(q_i, q_j)$
 - $k := k - 1$ // k 'yı güncelle
- end while
3. $R = \delta(q_{start}, q_{final})$
4. return R

Durum eksiltme yönteminin algoritmik gösterimi

