# Biçimsel Diller ve Otomata Teorisi

*Dr. Öğr. Üyesi Hayri Volkan Agun*

*Bilgisayar Mühendisliği Bölümü*

*Bursa Teknik Üniversitesi*

# Kaynaklar

## Ders Kitabı

- An Introduction to Formal Languages and Autotamata, Peter Linz, 6th Edition, 2017.
- An Introduction to Computer Theory, Daniel Isaac Aryeh Cohen, 2nd Edition, 1996.

## İçerik

- %100 Teorik
- Klasik sınav
- Vize %40, Final %60

# Düzenli ifadeler

- Sonlu durum ototmatı veya düzenli diller tanımlamak için kullanılan hesaplama elemanları sonlu durum otomatlarının bilgisayarların anlayacağı dilden kısa yoldan tanımlamak için tanımlama ifadeleri geliştirilmiştir.
- Bu tanımlama şekli otomatlar için düzenli ifadeler ve dil için düzenli gramerler olarak adlandırılmaktadır.
- Bir sonlu durum otomatı birden farklı düzenli ifade ile tanımlanabilir.
- Düzenli ifade ve düzenli gramerler bir dile ait karakter katarının kabul edilip edilmeyeceğini test etmek için de kullanılabilir.

# Düzenli ifadeler

- ∅, λ, ve a ∈ Σ ile tanımlanan herşey bir düzenli ifadedir. Bu tanımlamalar en temel düzenli ifade olarak adlandırılırlar.
- Programlama dillerinde düzenli ifadeler ile çalışmak için çeşitli kütüphaneler yer almaktadır. Düzenli ifadelerde tanımlama için kullanılan operatörler programlama diline özgü olabilir ancak her operatör için bir tanımlama her programlama dilinde yer almaktadır.
- Bu derste;
  - + operatörü birleşim için kullanılmıştır. Otomat tanımlamada bir geçiş sembolü yada başka bir geçiş sembolü olarak yorumlanabilir.
  - * operatörü sıfır, bir ve birden fazla tekrarı belirtmek için kullanılır.
  - • operatörü ardışıklığı ifade etmek için kullanılır.
  - () gruplama operatörüdür.

# Düzenli ifadeler

- (a + (b • c))* ->  düzenli ifadesinde a yada bc ya hiç yada birden fazla kez geçişi kabul edilir.
- Buna göre
  - λ , a, bc, aa, abc, bca, bcbc, aaa, aabc, gibi karakter katarları bu düzenli ifade tarafından kabul edilebilir.

# Düzenli ifadeler

- Düzenli ifadeler birleştirilebilirler, birleşimin sonucu yine bir düzenli ifadedir. Kısaca ne kadar çok düzenli ifadeyi birleştirirsek birleştirelim oluşan sonlu durum otomatı düzenli dilden daha karmaşık bir ifadeyi kabul edemez.
- Eğer $r_1$ ve $r_2$ düzenli ifade ise $r_1 + r_2$ , $r_1 \bullet r_2$ , ve $r_1{}^*$ ifadeleri de düzenli ifadedir.
- Örneğin $(a + b \bullet c)^* \bullet (c + \emptyset)$ ifadesi düzenli ifade midir? Çözüm için $r_1$ = c olsun bu durumda sırasıyla
  - $r_2=(c + \emptyset)$ düzenli ifadedir.
  - $r_1= b \bullet c$ düzenli ifadedir.
  - $r_3= a + r_1$ düzenli ifadedir.
  - $r_4= r_3{}^*$ düzenli ifadedir.
  - $r = r_4 \bullet r_2$ düzenli ifadedir.

# Düzenli diller

- Ø boş kümeyi temsil ediyorsa
- λ boş düzenli ifadeyi temsil ediyorsa
- Eğer $r_1$ ve $r_2$ düzenli ifade ise bu durumda
  - $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
  - $L(r_1 \bullet r_2) = L(r_1)\ L(r_2)$
  - $L((r_1)) = L(r_1)$
  - $L(r_1^*) = (L(r_1))^*$

# Örnek

- L(a* • (a + b)) ifadesi ile belirtilen dili açınız?
- Çözüm:
  - L(a* • (a + b)) = L(a*) L (a + b)
  - (L(a))* (L(a) U L(b))
  - {λ, a, aa, aaa, ...} {a,b}
  - {a, aa, aaa, ..., b, ab, aab,...}.
  - L = {$a^n$ $b^m$ : n >= 0; m>=0}

# Örnek

- L( (a + b)* •(a + bb) ifadesi ile belirtilen dili açınız?
- Çözüm:
  - L( (a + b)* (a + bb)) = L((a+b)*) L (a + bb)
  - (L(a+b))* (L(a) U L(bb))
  - {λ, a, b, ab, aab, abab, …} {a, bb}
  - aba, abbb, aababa, aabbb
  - $L=\{a^n b^m a^w b^q a^k (bb)^p \;:\; n>=0, m>=0, w>=0, q>=0, k \in \{0, 1\}, p \in \{0, 1\}\}$

# Örnek

- r = (aa)*(bb)* b düzenli ifadesi ile belirtilen dili yazınız?

- Bu dilde
  - çift sayıda a yı tek sayıda b takip etmektedir.
  - $L(r) = \{a^{2n} b^{2m+1} : n>=0, m>= 0\}$

# Soru

- $\sum = \{0, 1\}$ alfabesine sahip aşağıdaki düzenli dil ifade edilmektedir. Bu düzenli dile karşılık gelen düzenli ifade nedir?
- $L(r) = \{w \in \sum^* : w$ en az bir çift ardışık 0 barındırmalıdır$\}$
- L(r) nin kabul ettiği tüm karakter katarları (string) 00 barındırmalıdır. Bu durumda aşağıdaki düzenli ifade tanımlanabilir.
  - $r = (0 + 1)^*00(0+1)^*$

# Soru

- $\sum = \{0, 1\}$ alfabesine sahip aşağıdaki düzenli dil ifade edilmektedir. Bu düzenli dile karşılık gelen düzenli ifade nedir?
- $L(r) = \{w \in \sum^* :$ w ardışık 0 barındırma**ma**lıdır$\}$
- L(r) nin kabul ettiği tüm karakter katarları (string) 00 barındırmamalıdır. 0 geçecek ise önünde veya sonunda tekrar eden 0 geçemez. Bu durumda aşağıdaki düzenli ifade tanımlanabilir.
  - $(1^*011^*)^*$ yada
  - $r = (1 + 01)^* (0 + \lambda)$

# Düzenli diller ve düzenli ifadeler

- Düzenli diller ile düzenli ifadeler aynı şeyi ifade etmek için kullanılırlar. Her bir düzenli dil için bu dili yakalayan bir düzenli ifade vardır.
- Bir düzenli ifadeyi basit formda yazdığımızda düzenli dilin gramerini yazmış oluruz. Örneğin aşağıda 3 farklı kararsız sonlu durum otomatı verilmiştir. Bu gösterimi genelleyelim.

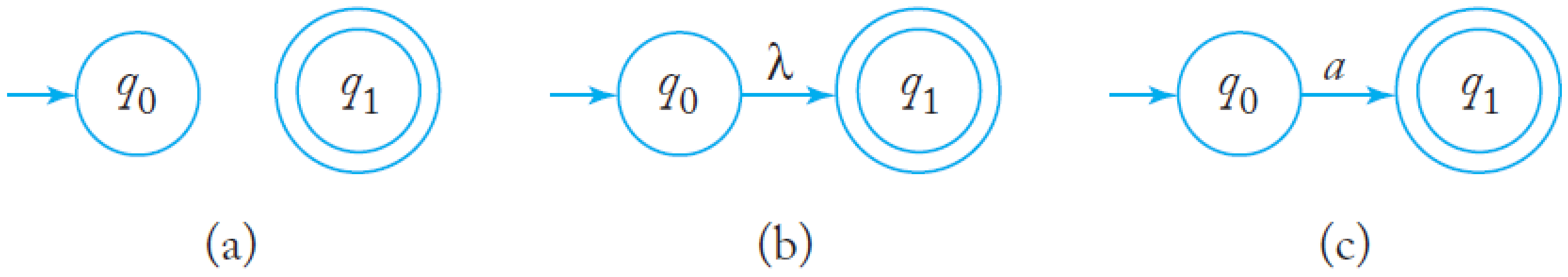
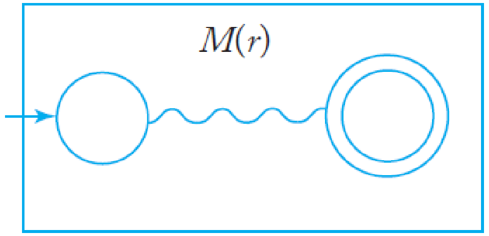



FIGURE 3.1 (a) nfa accepts $\varnothing$. (b) nfa accepts $\{\lambda\}$. (c) nfa accepts $\{a\}$.

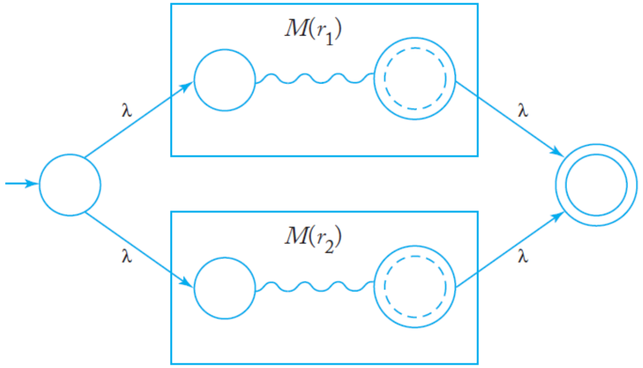
# Düzenli diller ve düzenli ifadeler

- Düzenli dillerin genlleştirilmiş şematik gösterimi verilmiş olsun. Aşağıda L(r) düzenli dilini ifade eden kararsız sonlu durum otomatı verilmiştir. M(r) bir önceki şekildeki 3 öğeden herhangi biri veya bu öğelerin birleşimi olabilir. Bu gösterime genel geçiş şeması (GTG) denilir.
- Bu öğelerin her biri ayrı bir düzenli dil yada düzenli dillerin birleşimi olabilir.
- Sonuçta ne kadar çok düzenli dili karmaşık birşekilde birleştirsek de oluşan dil düzenli bir dil olacaktır.

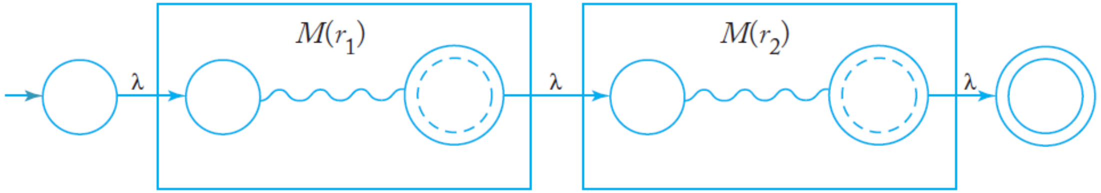


# Düzenli diller ve düzenli ifadeler

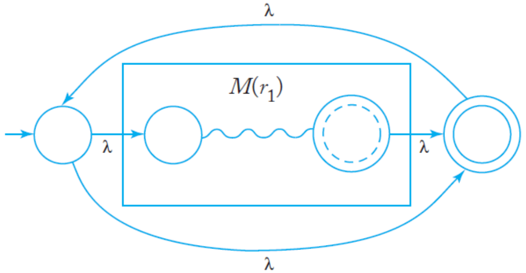
- **Kural 1:** L(r1 + r2)




- **Kural 2:** L(r1 r2)




- **Kural 3:** L(r1*)



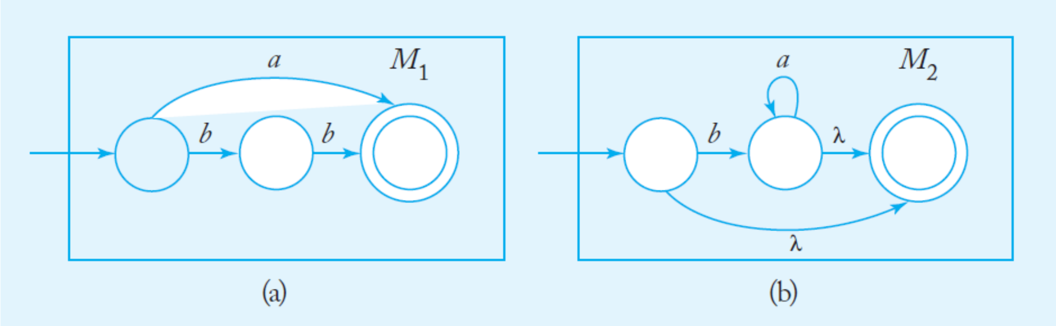
# Düzenli diller ve düzenli ifadeler

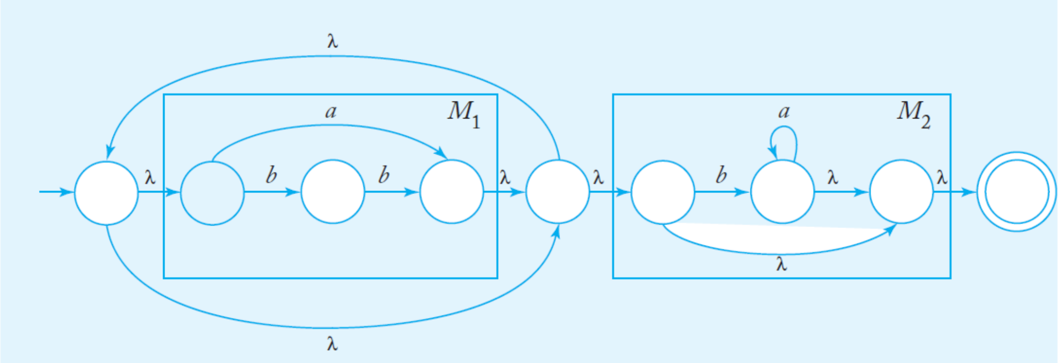
- Soru: L(r) dilinin düzenli bir dil olduğunu parçalayarak gösteriniz?

$$r = (a + bb)^* \, (ba^* + \lambda) \, .$$

- Cevap: (a) için L($M_1$) ve (b) için L($M_2$) önceki sayfada belirtilen yöntemle türetildiği için bu iki dil düzenli bir dildir.
- $M_1$= (a+bb) ve $M_2$=(ba*+λ)
- Bu durumda $M_1$ ve $M_2$ nin birleşimi L($M_1$ *$M_2$ *) şeklinde 3 ve 2 numaralı kural ile billeştirildiğinden sonuçta düzenli bir dildir.

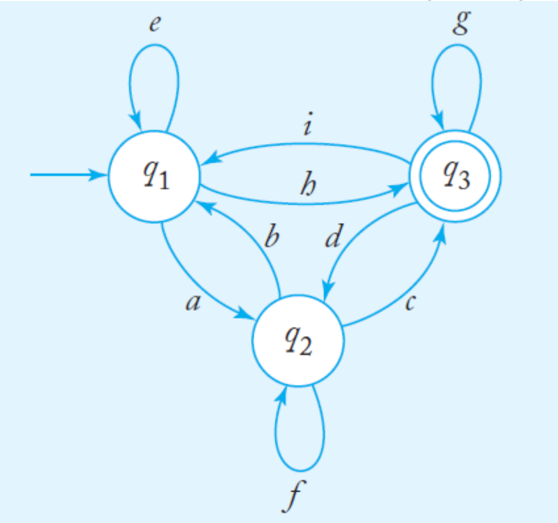
$$L\left((a + bb)^* \;\; (ba^* + \lambda)\right).$$

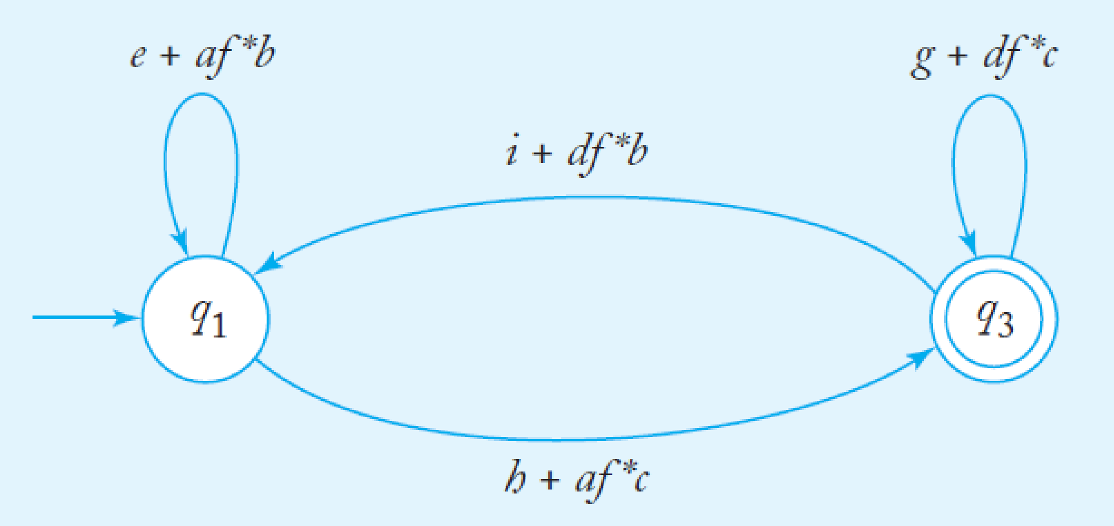


# Düzenli diller ve düzenli ifadeler

- Bahsettiğimiz bu kuralları tersine kullanarak kararsız sonlu durum otomatından düzenli bir ifade türetebiliriz.
- Bu nfa2rex yöntemi için önce genel geçiş şeması oluşturulur.
- Yandaki örnekte genel geçiş şeması verilmiştir. Buna GTG yani general transition graph ile düzenli ifadeyi basitçe oluşturabiliriz..



# Düzenli diller ve düzenli ifadeler

**Algoritma**

- GTG elde edilir.
- Eğer GTG üzerinde 2 adet durum varsa o zaman; i ve j düğümlerini kapsayan geçişler $r_{ii}$, $r_{ij}$, ve $r_{jj}$ ise düzenli ifade aşağıdaki gibidir.

$$r = r_{ii}^* r_{ij} (r_{jj} + r_{ji} r_{ii}^* r_{ij})^* .$$

- Eğer GTG üzerinde 3 düğüm varsa; bu düğümler i, j ve k olsun o zaman; p geçişi i ve j yi kapsasın, q geçişi de i ve j yi kapsasın. O zaman düzenli ifadede k düğümünü ve ilişkili düğümleri kaldırabiliriz. Bu durumda düzenli ifade
- 

$$r_{pq} + r_{pk} r_{kk}^* r_{kq}$$

# Düzenli diller ve düzenli ifadeler

**Algoritma**

- Eğer düğüm sayısı 4 veya üzeri ise herhangi bir k düğümü seçilir ve bu düğümden olmayan düğümler için aşağıdaki basitleştirme adımları uygulanır.

$$(q_i, q_j), i \neq k, j \neq k.$$

- Yukarıda k için i ve j geçişlerini barındırmamalıdır. Aşağıdaki basitleştirme adımları geçerli ise uygula.

$$r + \varnothing = r,$$

$$r\varnothing = \varnothing,$$

$$\varnothing^* = \lambda,$$

- K düğümünü kaldır ve tüm önceki adımlar tekrarlanır.

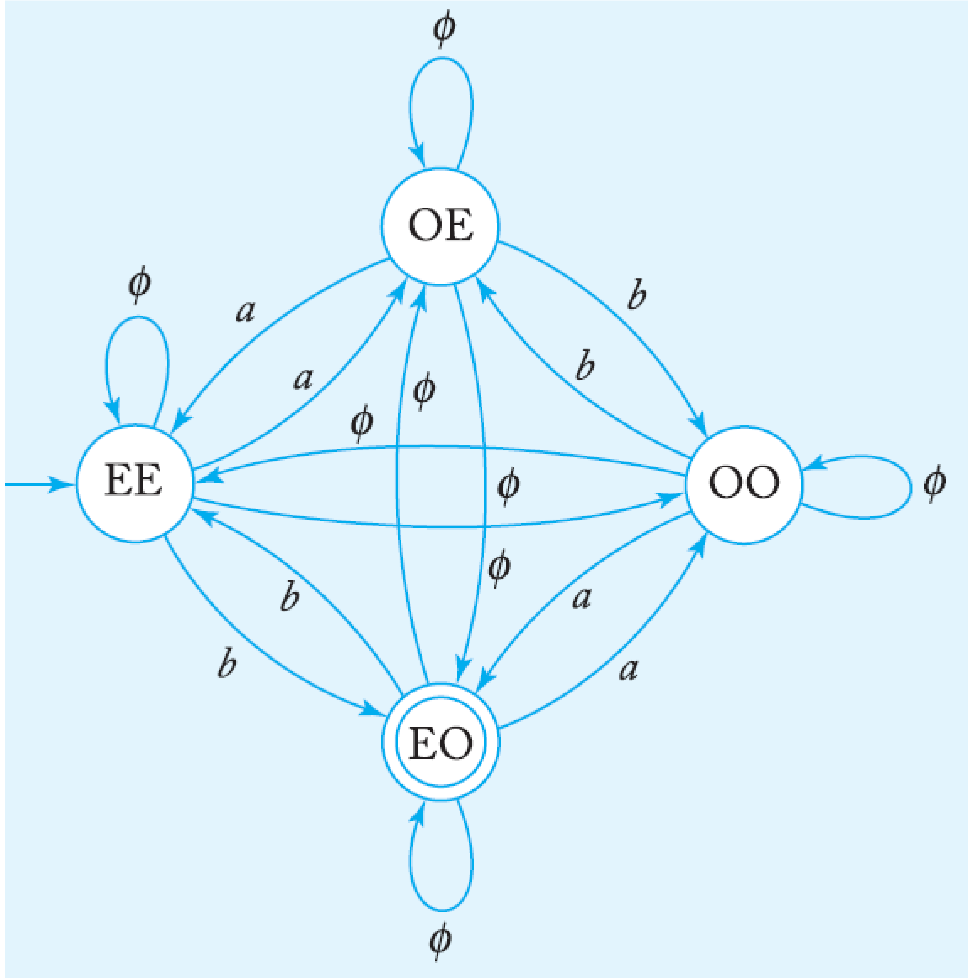
# Düzenli diller ve düzenli ifadeler

## Örnek

- Tek sayıda a veya b geçen bir dili ve çift sayıda a veya b geçen dili biribirinden ayıran düzenli ifadeyi bulunuz.
- Bunu yapabilmek için sonlu durumu çift sayıda a ve b içeren için EE durumu, tek sayıda a ve çift sayıda b içieren durum için OE durumu, tek sayıda a ve b içeren durum için OO durumu, çift sayıda a ve tek sayıda a içeren EO durumu olan bir sonlu durum otomatı oluşturalım.
- Bu durumda tek sayıda b’den b ile çift sayıda b’ye geçiş olur ve tek sayıda a’dan a ile çift sayıda a’ya geçiş olur. Benzer şekilde çift sayıda a’dan a ile tek sayıda a’ya ve çift sayıda b’den b ile tek sayıda b durumuna geçiş olacaktır.

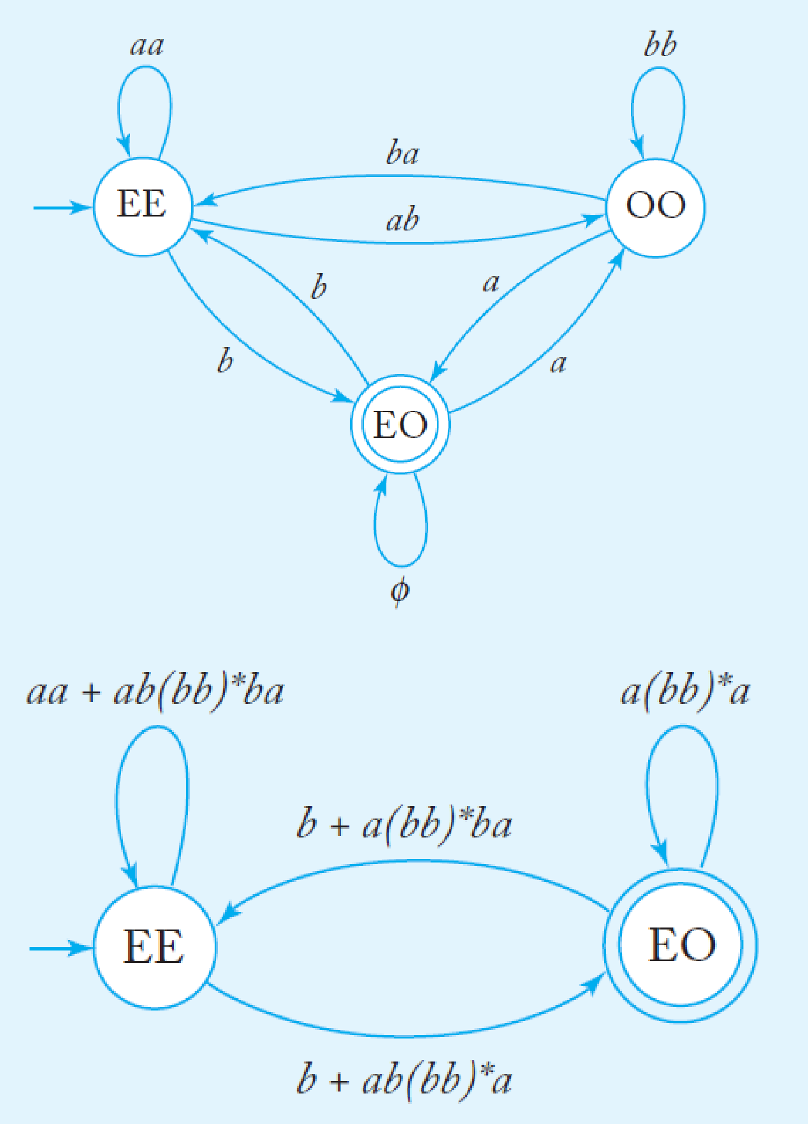
# Düzenli diller ve düzenli ifadeler

Örnek



# Düzenli diller ve düzenli ifadeler

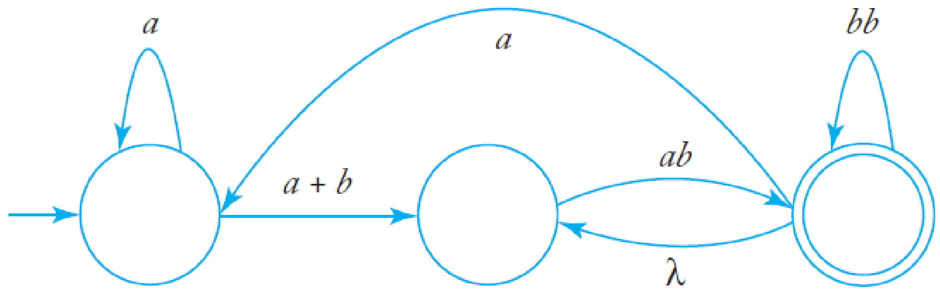
Örnek



# Düzenli diller ve düzenli ifadeler

## Örnek

- Aşağıdaki GTG şemasını sadeleştirerek iki durumlu GTG bulunuz ve düzenli ifadeyi oluşturunuz?




- Bu 3 durumlu bir şema olduğu için 3 durumlu şema koşulu sağlanmalıdır. Burada ortadaki düğüm k düğümünü kaldıralım. i düğümü başlangıç düğümü ve j düğümü de bitiş düğümü olsun. i düğümünden j düğümüne k düğümü üzerinden a+b (ab)* ile gidilir. j düğümünden  i düğümüne a ile gidilir. Bu durumda iki düğüm ile bunu ifade etmiş oluruz.

# Düzenli diller ve düzenli ifadeler

## Örnek

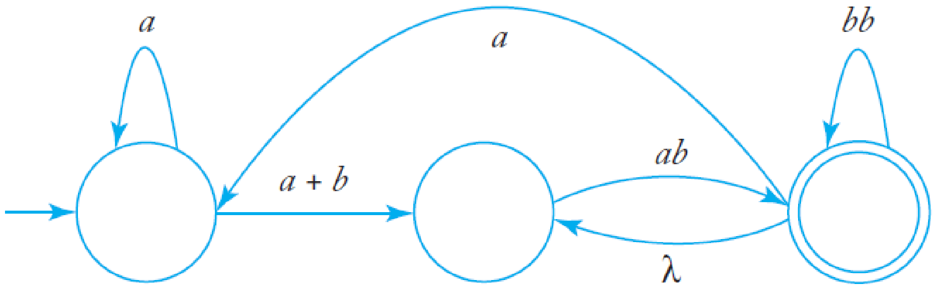
- 3 düğüm ile beliritilen kurala göre düzenli ifade:

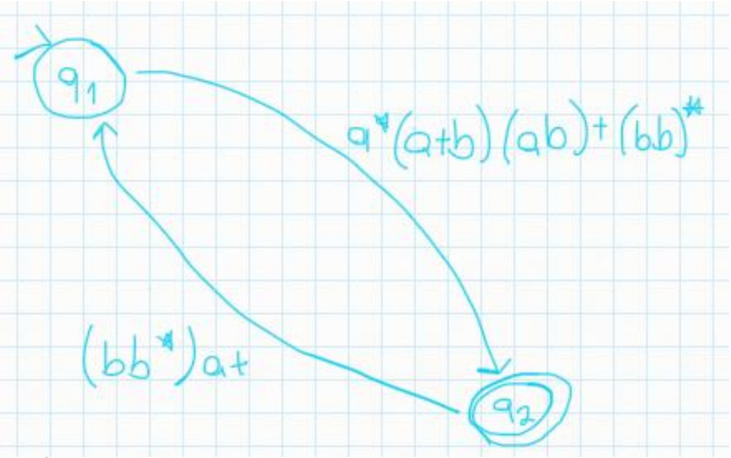
$$r_{pq} + r_{pk} r_{kk}^{*} r_{kq}$$

- p ve q için i ve j'ye ifadesi olarak belirtilir.  Burada $r_{kk}$ için (ab), $r_{pk}$ için a+b, $r_{kq}$ için ise bb ve böylece sağdaki ifade a+b (ab)* olacaktır. $r_{pq}$ için ise  a kullanılır.
- (a+b (ab)*a)+(bb)* bizim için geçerli bir düzenli ifade olacaktır. Burada adımları özetleyelim.
- Kaldırmak istediğimiz ortadaki düğüm üzerinden başlangıç düğümüne, ve son düğüme gelen düzenli ifadeleri yazarsak yukarıdaki ifadeyi bulabiliriz.

# Düzenli diller ve düzenli ifadeler

$$r = r_{ii}^{*} r_{ij} (r_{jj} + r_{ji} r_{ii}^{*} r_{ij})^{*}.$$




- Başlangıç düğümüne gelen düzenli ifadeler (bb*)a+ ile ifade edilebilir. Burada kaldırılan ortanca düğümden başlangıç düğümüne gelen bir geçiş yoktur. O yüzden yazılmaz.
- Bitiş düğümü için a*(a+b)(ab)+(bb)* yazılır. Burada k düğümü yani ortanca düğüm üzerinden geçen düğümler Kural 2 ile birleştirilir.

# Düzenli Gramerler

- Gramerler dili tanımlamada kullanılan alternatif özyinelemeli kurallardır. Gramerler sol yada sağdan doğrusal (lineer) olacak şekilde tanımlanabilir. Örneğin: $G = (V, T, S, P)$ tanımlı gramer için tüm kurallar genel olarak
  - Aşağıdaki gibi tanımlanırsa sağdan lineer gramer olur.

$$A \to xB,$$
$$A \to x,$$

  - Eğer aşağıdaki gibi tanımlanırsa soldan lineer gramer olacaktır.

$$A \to Bx,$$

$$A \to x.$$

  - Yukarıdaki gramer tanımlamalarında B başka bir açılım olabilir. Ancak x somut olarak alfabede geçen bir sembol olmak durumundadır.

# Düzenli Gramerler

- Düzenli gramerler düzenli bir ifade şeklinde ifade edilebilir. Örneğin;

$$G_1 = (\{S\}, \{a, b\}, S, P_1),$$

- Yukarıdaki gramer için kurallar $P_1$

$$S \rightarrow abS|a$$

- Bu gramer (ab)*a düzenli ifadesine denktir.

# Düzenli Olmayan Gramerler

- Her bir gramer tanımı düzenli olmak zorunda değildir. Ama tüm düzenli gramerler doğrusal gramedir. Burada doğrusal gramer için kurallarda sağda en fazla bir adet sembol olmayan kural açılımı olmalıdır. Örneğin;

$$A \to xB,$$
$$A \to x,$$

- A kural açılımında sağda bir adet kural (B) mevcuttur. Bu durumda gramer doğrusaldır. Eğer gramer hem soldan hem de sağdan doğrusal kurallar barındırıyorsa o zaman düzenli değildir. Bu her bir kural için iki farklı türde açılıma neden olacağından düzenli olmaktan çıkar.

# Düzenli Olmayan Gramerler

- A kural açılımında sağda bir adet kural (B) mevcuttur. Bu durumda gramer doğrusaldır. Eğer gramer hem soldan hem de sağdan doğrusal kurallar barındırıyorsa o zaman düzenli değildir. Bu her bir kural için iki farklı türde açılıma neden olacağından düzenli olmaktan çıkar.

$$G = (\{S, A, B\}, \{a, b\}, S, P)$$

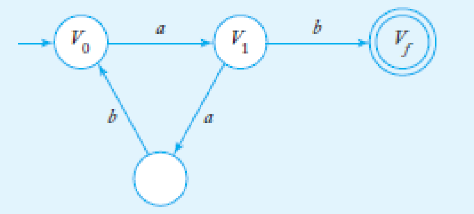
$$\begin{aligned} S &\rightarrow A, \\ A &\rightarrow aB|\lambda, \\ B &\rightarrow Ab, \end{aligned}$$

# Düzenli Gramerler

- Aşağıdaki gramer kuralları için sonlu durum otomatı oluşturulabilir mi? Böyle bir otomat varsa çiziniz?

$$
\begin{aligned}
V_0 &\rightarrow aV_1, \\
V_1 &\rightarrow abV_0 | b,
\end{aligned}
$$

- Bu kurallar sağdan lineer bir gramere aittir. Bu durumda sonlu durum otomatı vardır. Çünkü düzenli bir gramerdir ve düzenli bir ifadeye sahiptir.
- $V_0$ ve $V_1$ için birer düğüm oluşturalım. Bu durumda $V_0$ ‘dan a ile $V_1$ ‘e gidilir. $V_1$ ‘den ab ile $V_0$ ‘a gidilir. Aynı zamanda $V_1$ ‘den b ile son duruma gidilir.



# Düzenli Gramerler

- Düzenli diller düzenli gramerler ile ifade edilirler. Bu durumda dilin düzenli olması için o dili tanımlayan en az bir düzenli gramer olmalıdır.
- Düzenli gramerler belirsiz sonlu durum otomatları ile ifade edilebilir ve aynı zamanda düzenli ifadeler ile tanımlanabilir.

# Düzenli Gramerler

- Aşağıda kuralları verilen gramer ne tür bir gramerdir? Bu gramerin kararlı sonlu durum otomatını çiziniz?

$$
\begin{aligned}
S &\rightarrow abA, \\
A &\rightarrow baB, \\
B &\rightarrow aA|bb.
\end{aligned}
$$

- Bu gramer sağdan doğrusal bir gramerdir. Bu gramer düzenli gramerdir. S başlangıç durumu A düğümüne ab ile gitsin. ab çift sembol olduğundan ara bir düğüm daha ekleyrek önce a ile ara düğüme sonra b ile A düğümüne gidilir. A düğümünden de benzer şekilde ba ile B düğümüne gidilir. B düüğmünden a ile A düğümüne dönülür yada bb ile benzer şekilde ara düğüm kullanarak son düğüme gidilir. Bu anlatılanları çizerek görselleştiriniz.