



# BİÇİMSEL DİLLER VE OTOMATA TEORİSİ

Biçimsel Diller ve Otomata Teorisi

6. Hafta

DR. ÖĞR. ÜYESİ. HÜSEYİN VURAL



# Ders İzlenesi

---

- Düzenli Diller(Regular Languages)
- Düzenli İfadeler(Regular expressions)

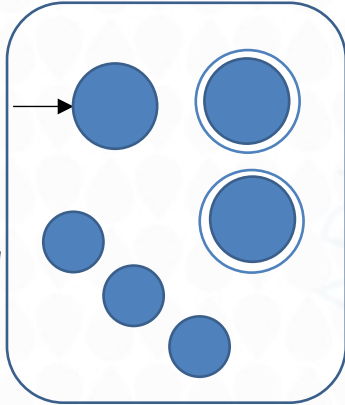
# Düzenli Diller

---

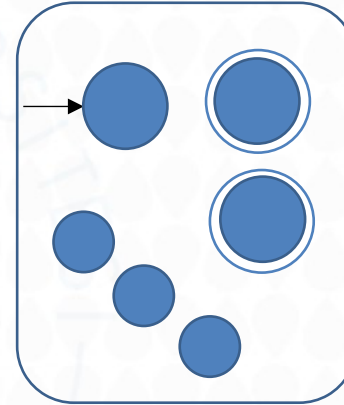
- Belirli bir düzenli dil sınıfı belirli işlemler uygulandığında yine aynı sınıf içinde kalmaktadır
- Bu işlemlerden bazıları union(birleşim), concatenation(ard arda sıralanma) ve kleene star(yıldız) işlemleridir

# Düzenli Dillerde Birleşim

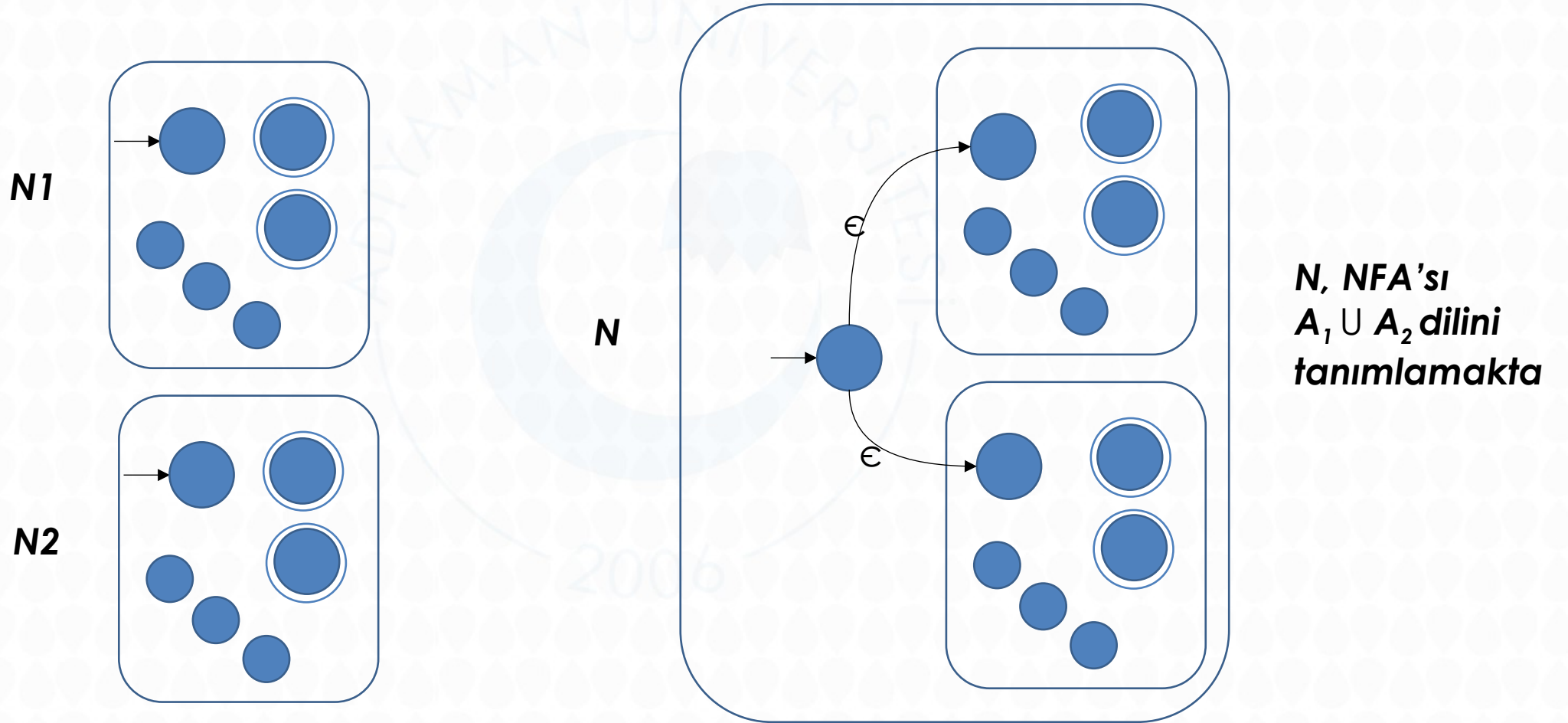
***N1 NFA'sı  
 $A_1$  dilini  
tanımlamakta***



***N2 NFA'sı  
 $A_2$  dilini  
tanımlamakta***

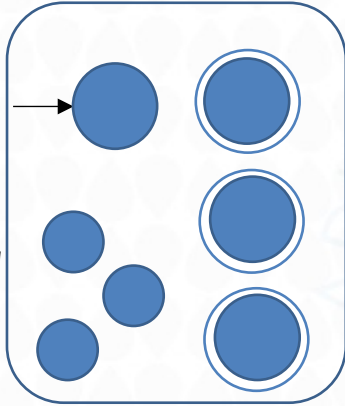


# Düzenli Dillerde Birleşim

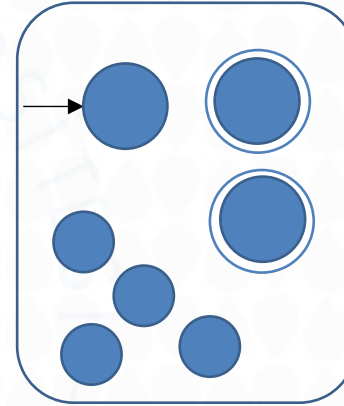


# Düzenli Dillerde Sıralama

***N1 NFA'sı  
 $A_1$  dilini  
tanımlamakta***

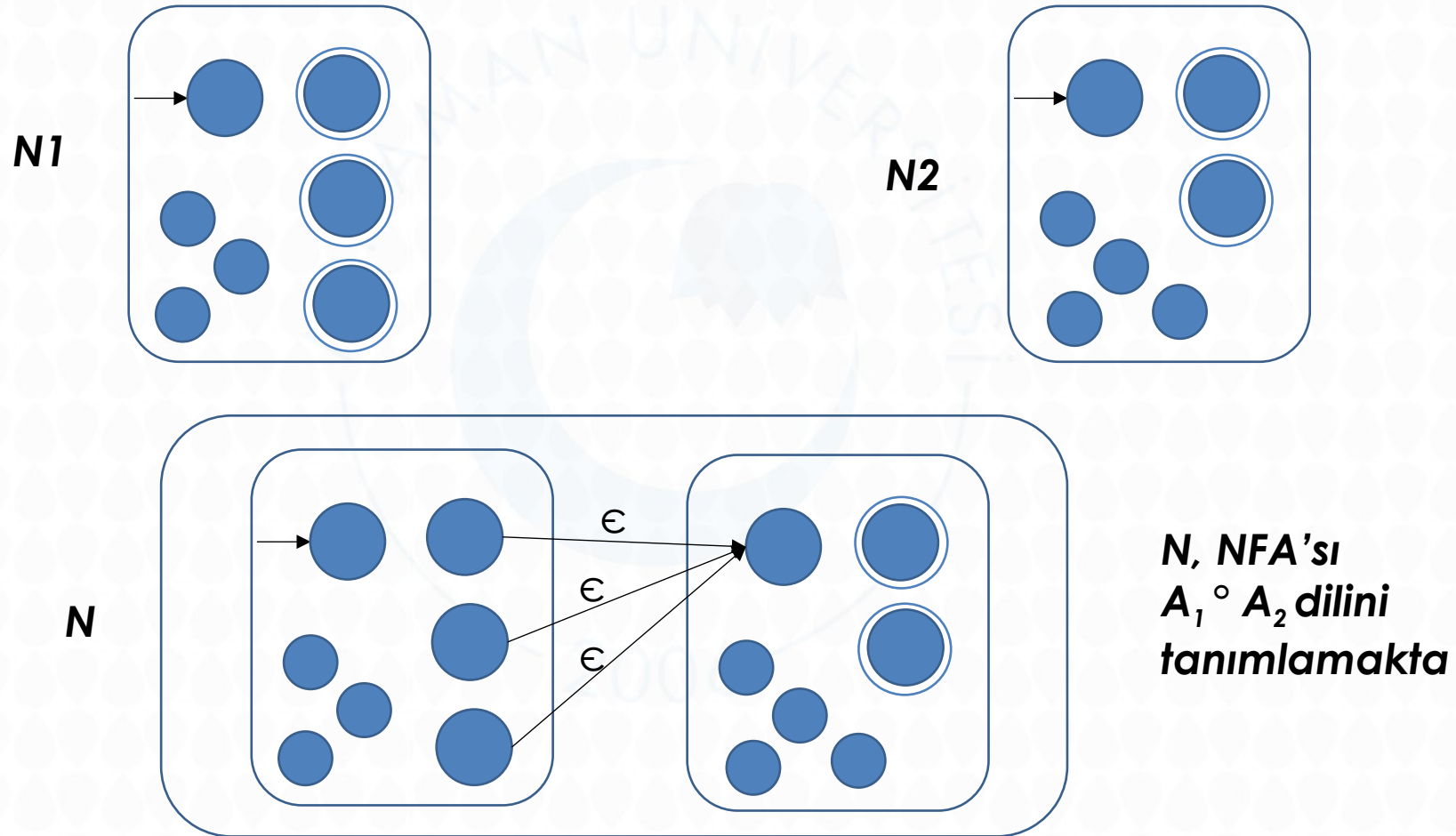


***N2 NFA'sı  
 $A_2$  dilini  
tanımlamakta***



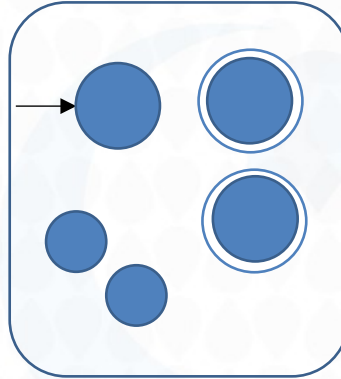


# Düzenli Dillerde Sıralama



# Düzenli Dillerde Yıldız

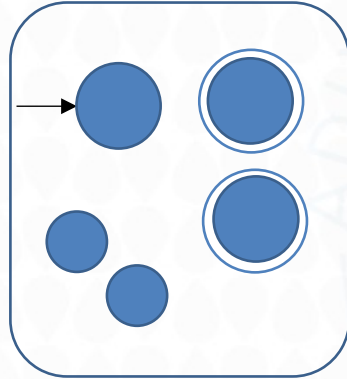
***N1 NFA'sı  
A dilini  
tanımlamakta***



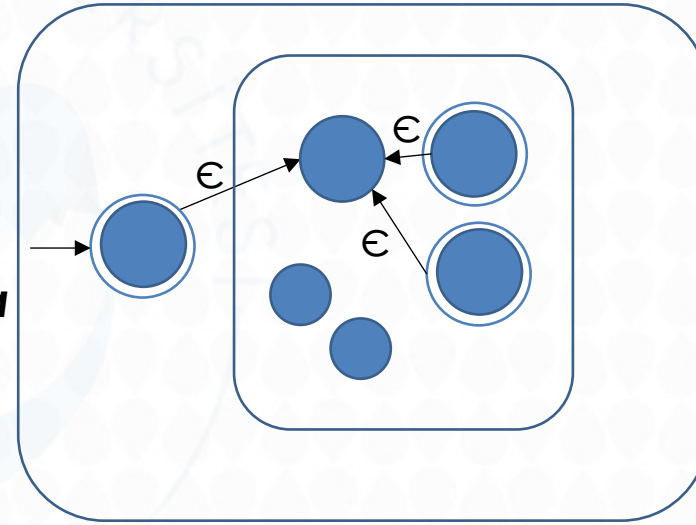


# Düzenli Dillerde Yıldız

***N1***



***N, NFA'sı  
A\* dilini  
tanımlamakta***



# Düzenli İfadeler

- Matematik'te  $+$  ve  $\times$  gibi operatörleri kullanarak  $(5+3)\times 4$  gibi matematiksel ifadeler tanımlayabiliriz.
- Benzer şekilde, bu operatörleri kullanarak dilleri tanımlayan düzenli ifadeler de tanımlayabiliriz.
  - Ör:  $(0 \cup 1)^*$
- Üstteki matematiksel ifadenin sonucu 32'dir. Altta ki düzenli ifadenin sonucuda bir dildir.

# Düzenli İfadeler

- $(0 \cup 1)0^*$
- Bu örnekte ki ifadenin tanımladığı dil; 0 veya 1 ile başlayan ve sonsuz 0 içerebilecek bir dildir.
- $(0 \cup 1)0^*$  ifadesinde concatenation kullanılmıştır.
- $(0 \cup 1)0^* = (0 \cup 1) \circ 0^*$

# Düzenli İfadeler

- Matematik'te parantez en öncelikli işlemdir daha sonra çarpma daha sonrasında toplama operatörü önceliklidir.
- Düzenli ifadelerde ise en öncelikli operatör  $*$  (star)'dır. Daha sonra concatenation daha sonrasında birleştirme operatörü gelmektedir.
- Bir ifadenin **düzeli ifade olarak** kabul edilebilmesi için belli şartlar taşınması gerekmektedir.

# Düzenli İfadeler

- **1.** Bir alfabedeki ( $\Sigma$ ) bir  $a$  sembolü düzenli ifadedir,
- **2.**  $\epsilon$ , boş bir string (null) içerebilir
- **3.**  $\emptyset$ , düzenli ifade boş küme dili içerebilir
- **4.**  $R_1 \cup R_2$  ifadesi de bir düzenli ifadedir
- **5.**  $R_1 \circ R_2$  ifadesi de bir düzenli ifadedir
- **6.**  $(R_1^*)$  ifadesi de bir düzenli ifadedir





# Düzenli İfadeler

- $\Sigma = \{0,1\}$ , alfabetini göz önünde bulunduralım
- $0^*10^*$
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $(01^+)^*$
- $(\Sigma\Sigma)^*$
- $(\Sigma\Sigma\Sigma)^*$
- $01 \cup 10 = \{01, 10\}$





# Düzenli İfadeler

- $\Sigma = \{0, 1\}$ , alfabetini göz önünde bulunduralım
- $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$
- $(0 \cup \epsilon)1^* = 01^* \cup 1^*$
- Bir dil eğer düzenli ifadeler tarafından tanımlanabiliyorsa o zaman düzenli dil olarak kabul edilebilir.

# Düzenli İfadeler

- $\Sigma = \{0, 1\}$ , alfabetini göz önünde bulunduralım
- $0^*10^*$  = sadece birtane 1 içerir
- $\Sigma^*1\Sigma^*$  = en az bir tane 1 içerir
- $\Sigma^*001\Sigma^*$  = 001 substringini mutlaka içerir
- $(01^+)^*$  = Her 0'dan sonra mutlaka 1 gelmelidir
- $(\Sigma\Sigma)^*$  = çift uzunluktaki stringleri içerir
- $(\Sigma\Sigma\Sigma)^*$  = 3 veya 3'ün katı uzunluktaki stringleri içerir
- $01 \cup 10 = \{01, 10\}$

# Düzenli İfadeler

- $\Sigma = \{0, 1\}$ , alfabetini göz önünde bulunduralım
- $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = 0$  ile başlayıp 1 ile biten stringleri içerir
- $(0 \cup \epsilon)1^* = 01^* \cup 1^* = 0$  ile başlar ve sonsuz 1 içerebilir
- Bir dil eğer düzenli ifadeler tarafından tanımlanabiliyorsa o zaman düzenli dil olarak kabul edilebilir.



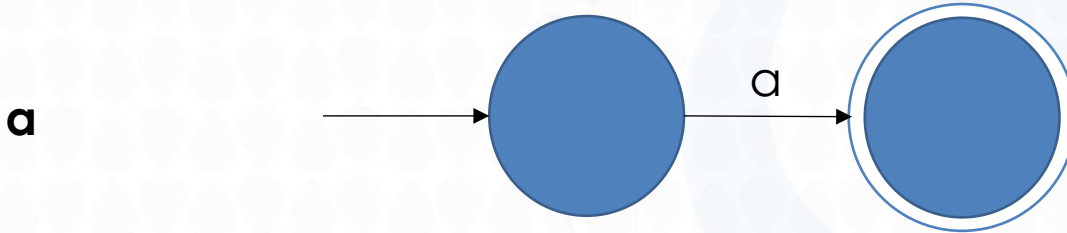
# Düzenli İfadeler

---

- $(abua)^*$  düzenli ifadesi için NFA oluşturalım

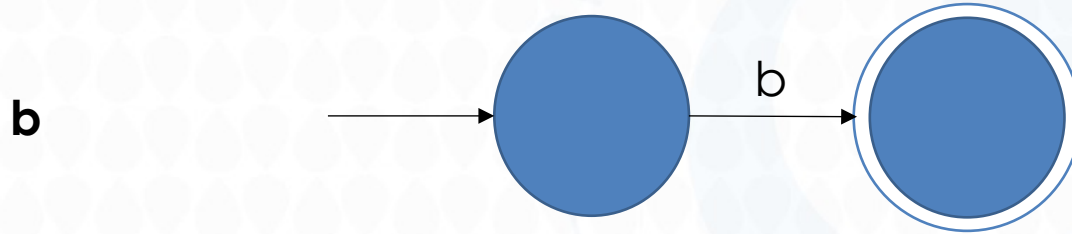
# Düzenli İfadeler

- $(abua)^*$  düzenli ifadesi için NFA oluşturalım



# Düzenli İfadeler

- $(abua)^*$  düzenli ifadesi için NFA oluşturalım

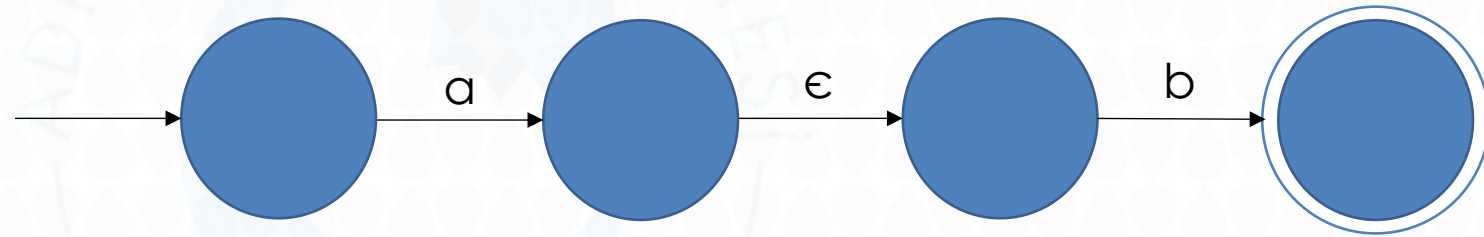




# Düzenli İfadeler

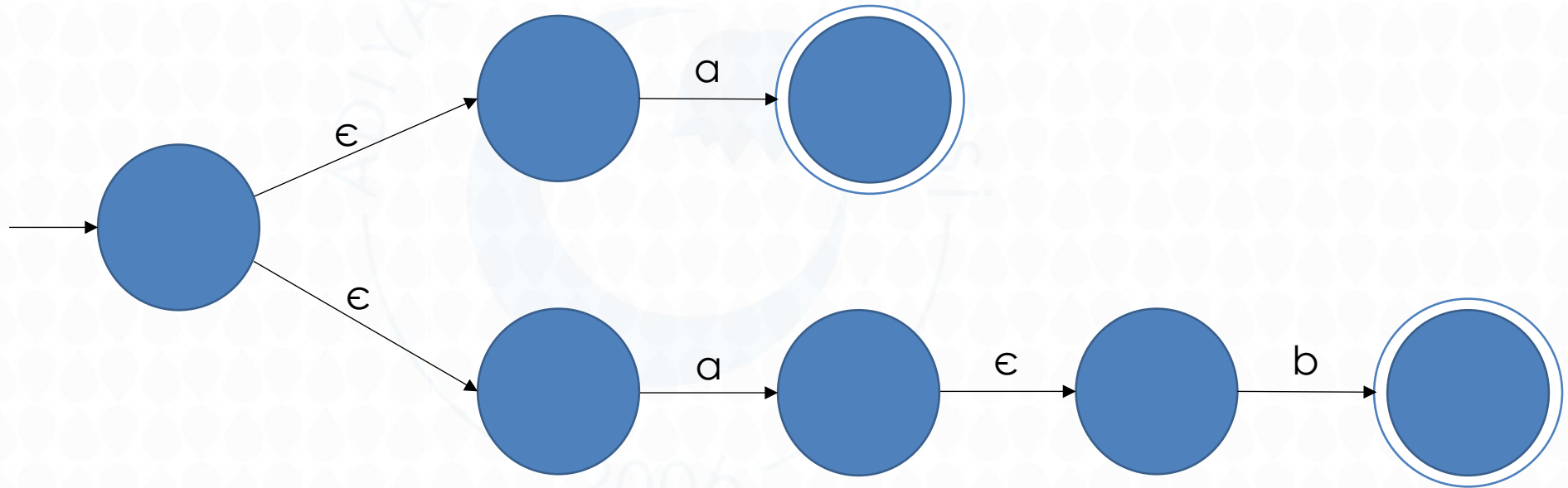
- $(abua)^*$  düzenli ifadesi için NFA oluşturalım

**ab**



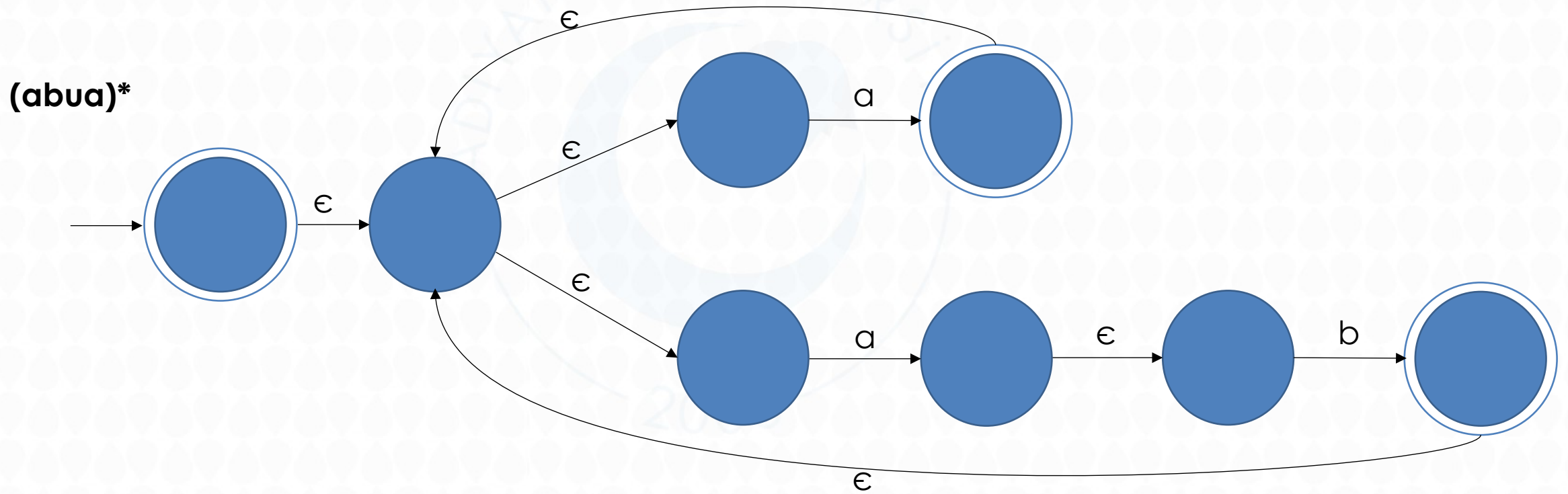
# Düzenli İfadeler

- $(abua)^*$  düzenli ifadesi için NFA oluşturalım



# Düzenli İfadeler

- $(abua)^*$  düzenli ifadesi için NFA oluşturalım





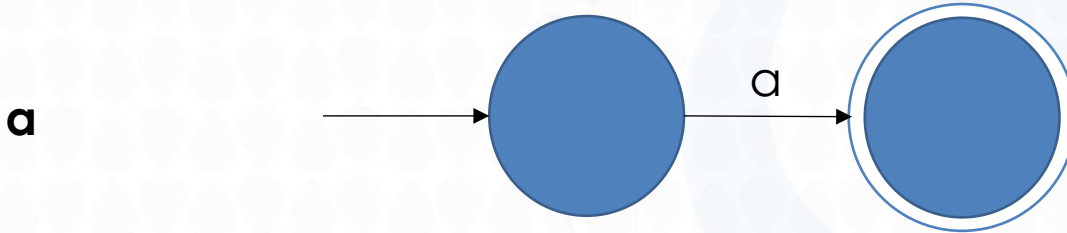
# Düzenli İfadeler

---

- $(aub)^*aba$  düzenli ifadesi için NFA oluşturalım

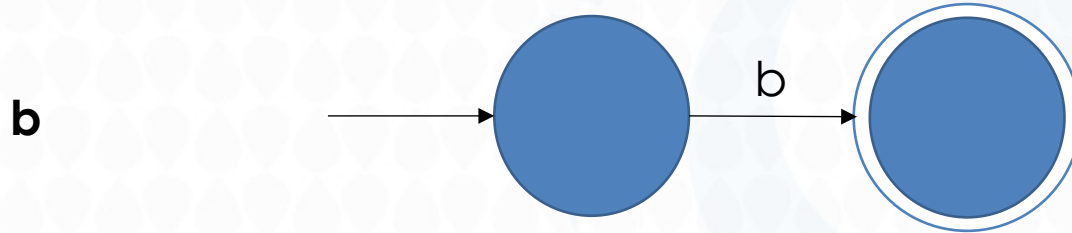
# Düzenli İfadeler

- $(aub)^*aba$  düzenli ifadesi için NFA oluşturalım



# Düzenli İfadeler

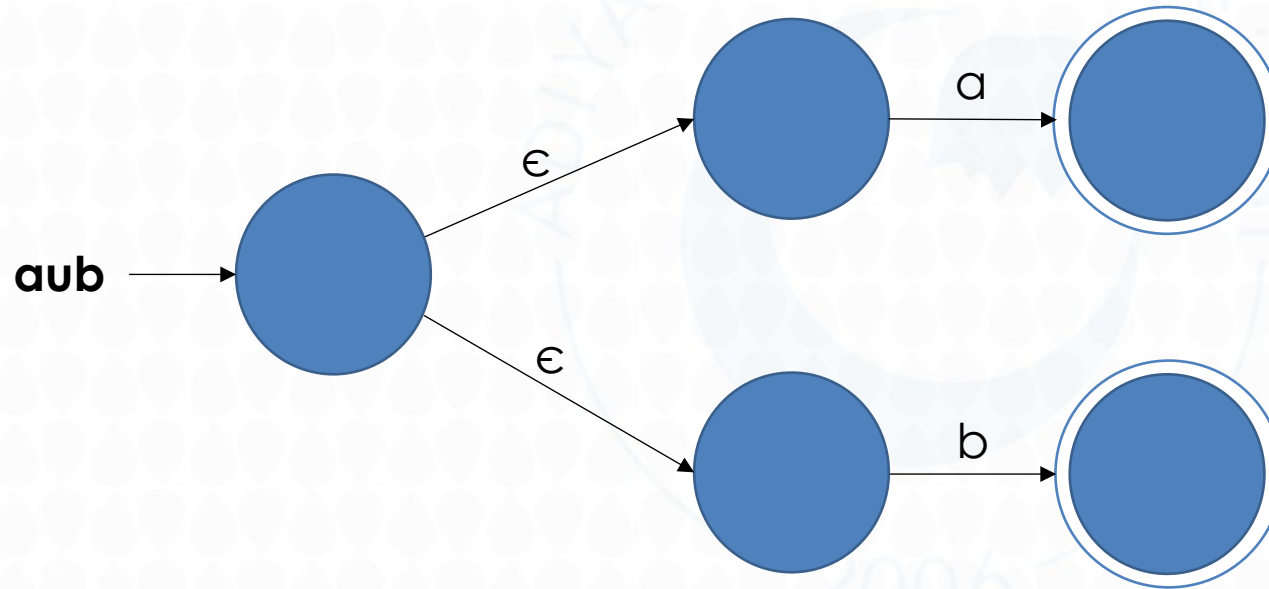
- $(aub)^*aba$  düzenli ifadesi için NFA oluşturalım





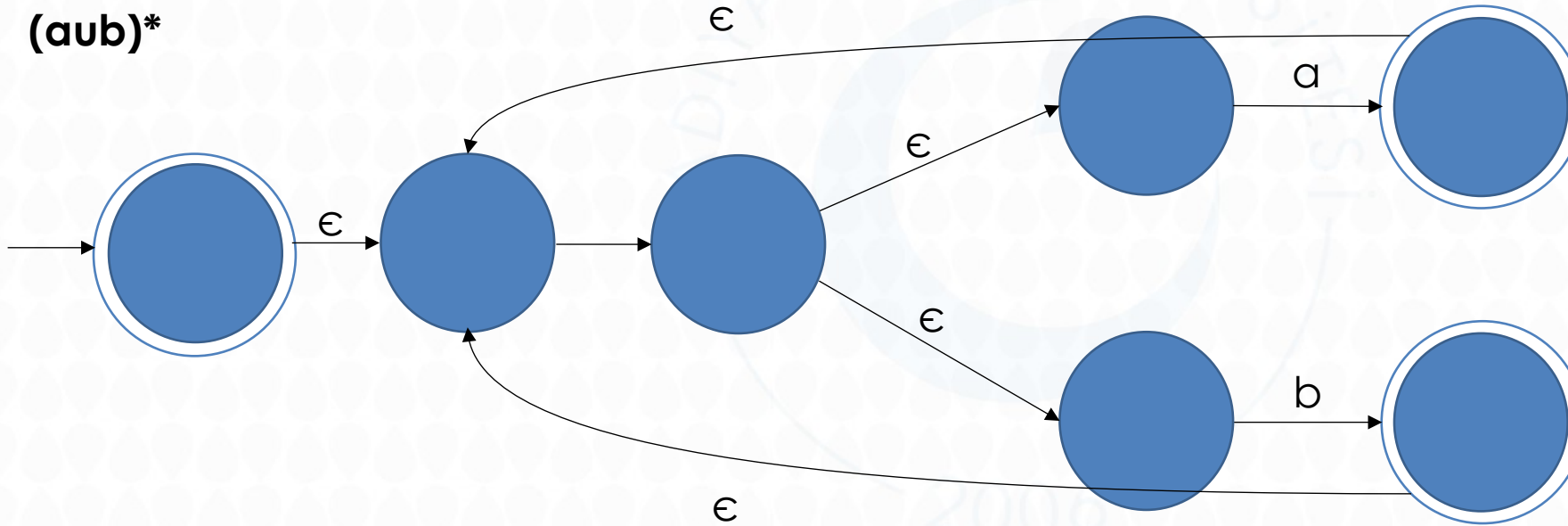
# Düzenli İfadeler

- $(aub)^*aba$  düzenli ifadesi için NFA oluşturalım



# Düzenli İfadeler

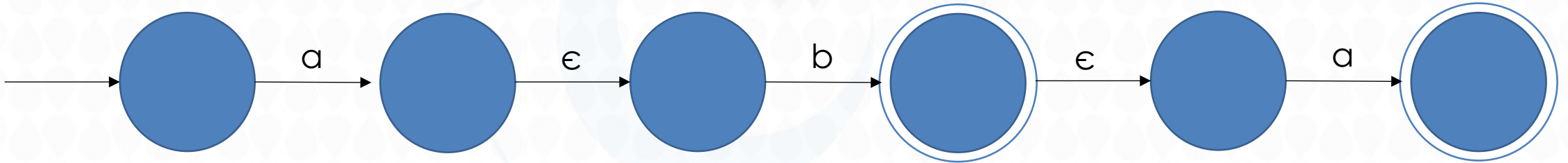
- $(aub)^*aba$  düzenli ifadesi için NFA oluşturalım



# Düzenli İfadeler

- $(aub)^*aba$  düzenli ifadesi için NFA oluşturalım

**aba**



# Düzenli İfadeler

$(a|b)^*aba$

