



# Düzenli Diller (Regular Languages)

Otomata Teorisi  
Konya Teknik Üniversitesi  
Bilgisayar Mühendisliği Bölümü



# Tanım

- Düzenli ifade ile tanımlanabilen bir dil düzenli dil "regular language" olarak adlandırılmaktadır.
  - Düzenli bir dile karşılık gelen bir sonlu otomata modeli vardır.
  - Düzenli ifade ile tanımlanamayan diller de bulunmaktadır. Bu diller ileri konularda incelenecektir.

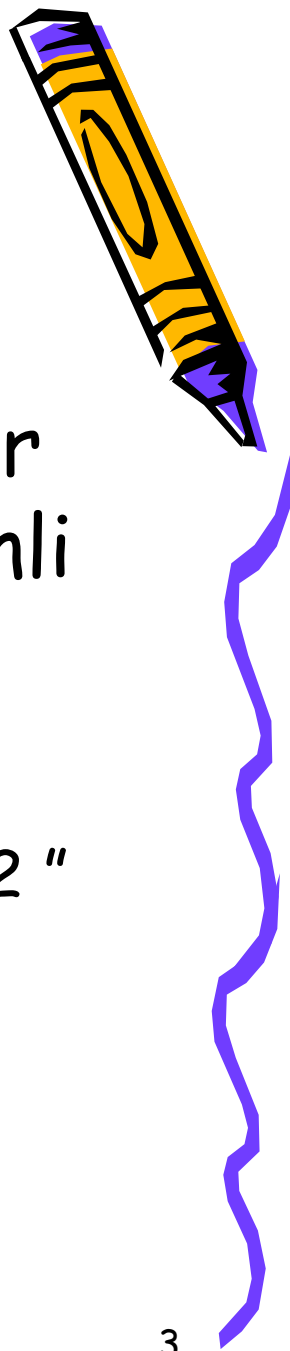


# Teorem

- Eğer  $L1$  ve  $L2$  düzenli diller (regular languages) ise,  $L1 + L2$  dili de düzenli bir dildir.

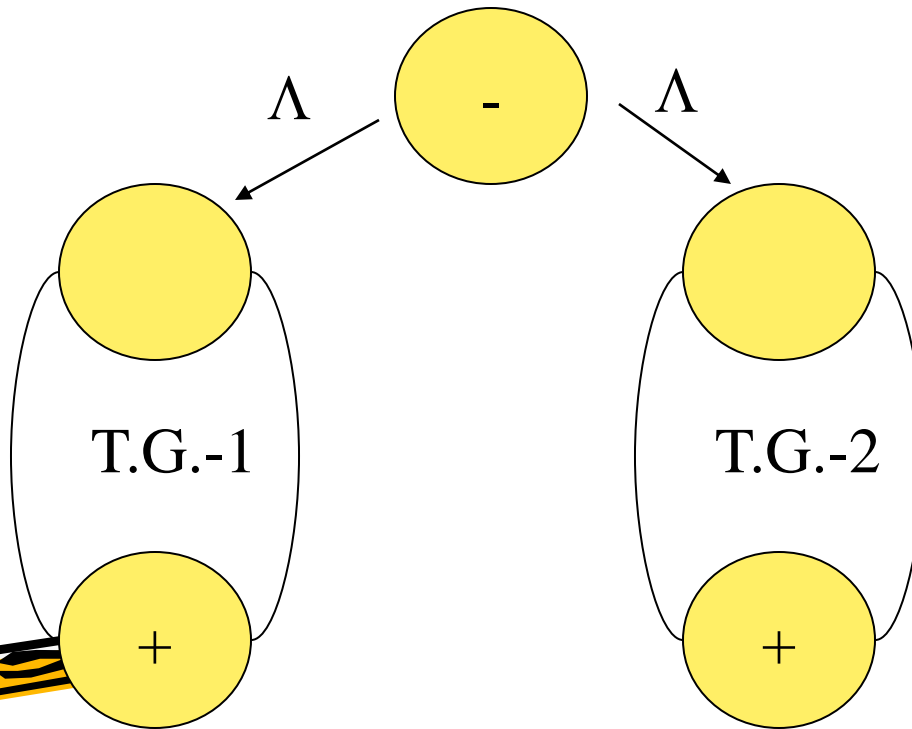
$L1 + L2$  :

"The language of all words in either  $L1$  or  $L2$  "  
(Ya  $L1$  kümesi ya da  $L2$  kümesinde olan kelimelerin dili)



# İspat

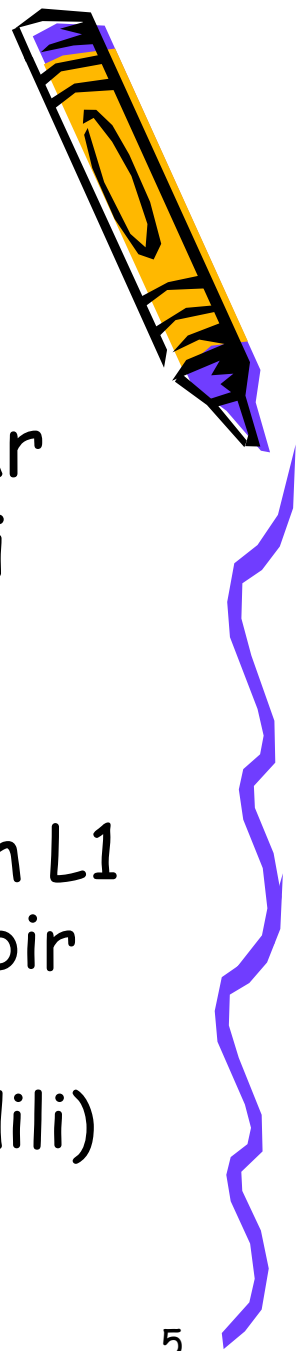
- T.G. Kullanarak ispat yapılabilir.



**L1 + L2 için T.G.**

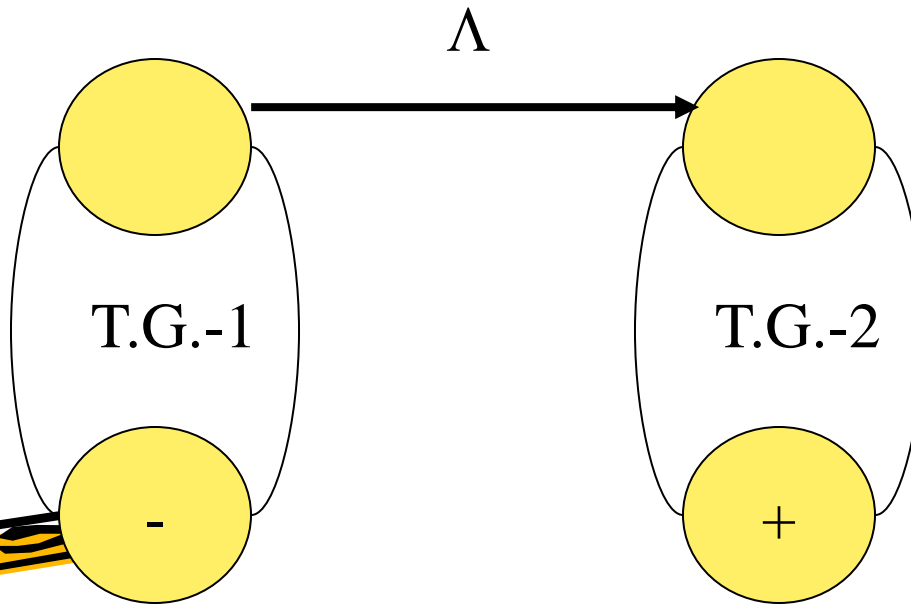
# Teorem

- Eğer  $L_1$  ve  $L_2$  düzenli diller (regular languages) ise,  $L_1 L_2$  dili de düzenli bir dildir.
  - $L_1 L_2$  : "The language of all words formed by concatenating a word from  $L_1$  with a word from  $L_2$ ." ( $L_1$  den alınan bir kelime ile  $L_2$  den alınan bir kelimenin birleştirilmesi ile oluşan kelimelerin dili)



# İspat

- T.G. Kullanarak ispat yapılabilir.



**L1 L2 için T.G.**

# Teorem

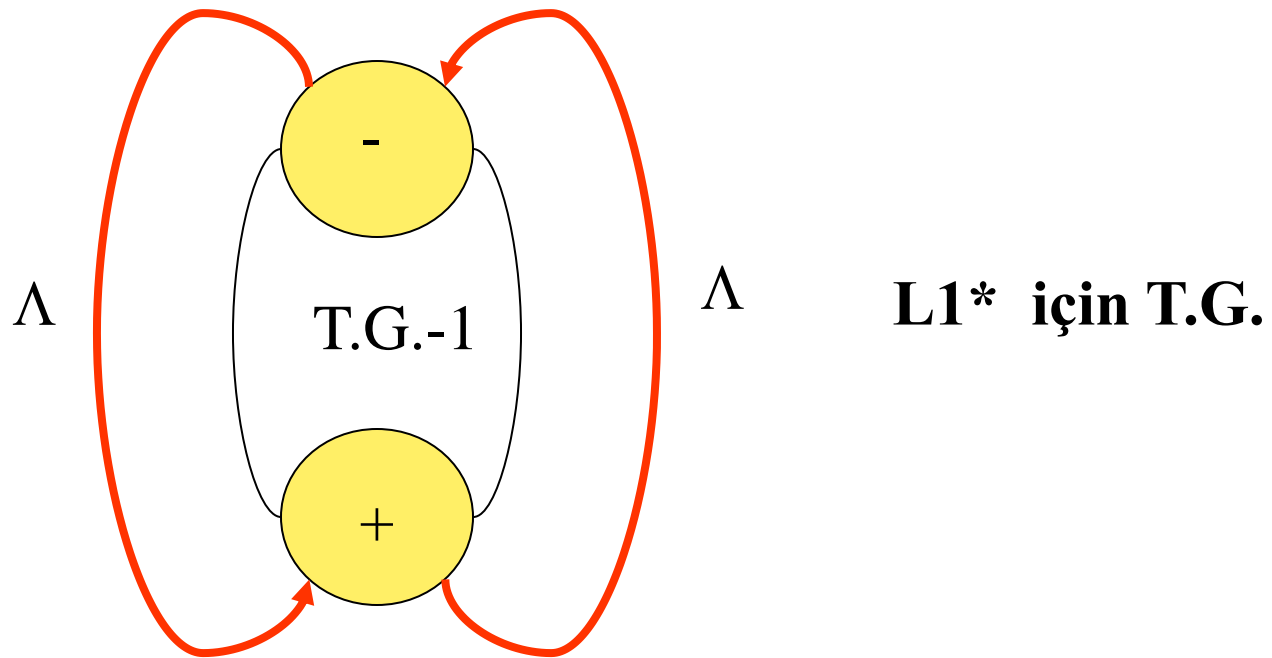


- Eğer  $L_1$  ve  $L_2$  düzenli diller (regular languages) ise,  $L_1^*$  dili de düzenli bir dildir.
  - $L_1^*$  : "The language of all words that are the concatenation of arbitrarily many factors from  $L_1$ ". (L1 den alınan kelimelerin çeşitli defalar birleştirilmesi ile oluşan kelimelerin dili).



# İspat

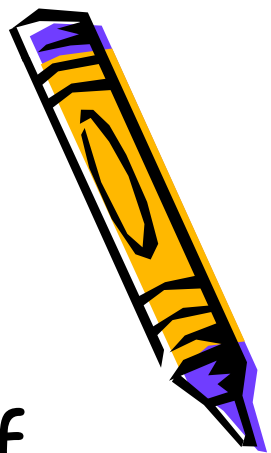
- T.G. Kullanarak ispat yapılabilir.





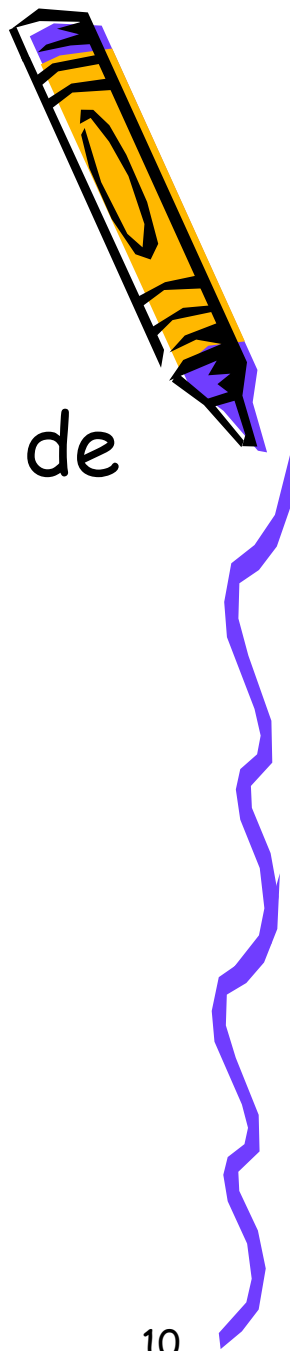
# Tanım

- L dilinin Tümleyeni ( Complement of L):
  - $L'$  ile gösterilir.  $L'$  dili, L dilinde olmayan tüm kelimeleri içerir.
  - $(L')' = L$



# Teorem

- Eğer  $L$  dili düzgün bir dil ise,  $L'$  dili de düzgün bir dildir.
  - İspatı bir sonraki slaytta yer almaktadır.



# İspat



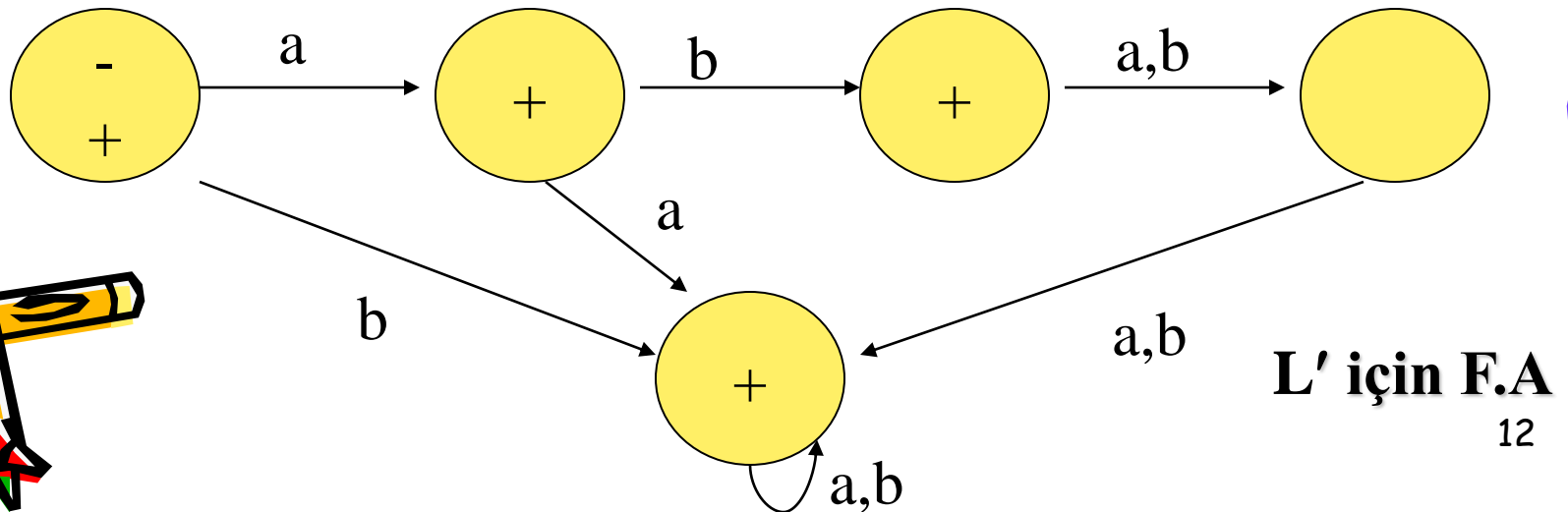
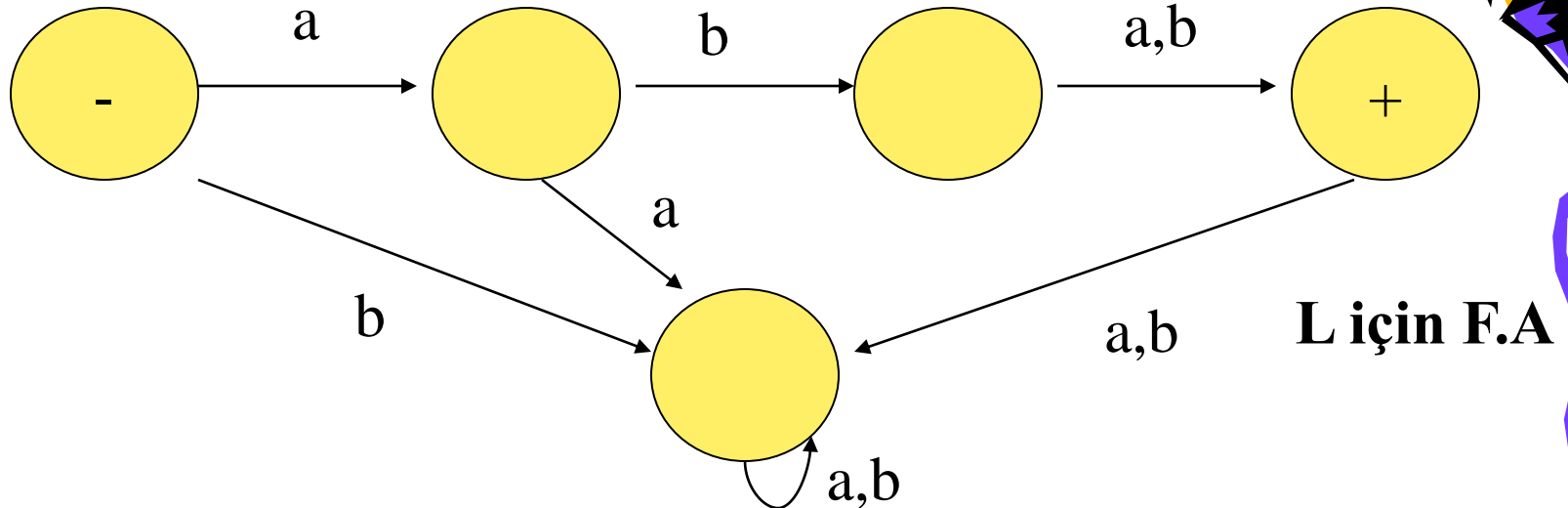
- L dilini kabul eden bir F.A. vardır. L' dilini kabul eden bir F.A. modeli ortaya konabilirse ispat yapılmış olacaktır.
- Yöntem: Bir F.A.'nın sonuç durumlarını normal durumlara, normal durumlarını da sonuç durumlarına çevirmek ile, L' dilini kabul eden bir F.A. Modeli çıkacaktır.

Uyarı:  $- \leftrightarrow \pm$



# İspat

- $L$  ve  $L'$  için örnek F.A. modelleri:



# Teorem



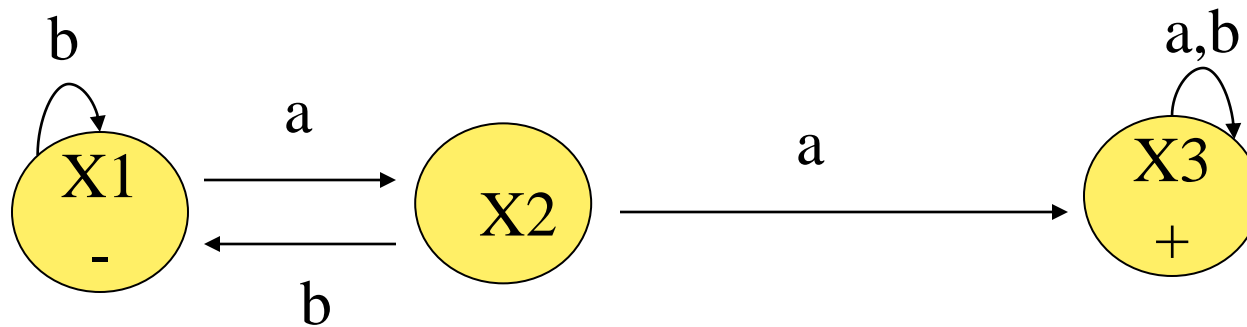
- Eğer  $L1$  ve  $L2$  düzenli diller ise,  $L1 \cap L2$  dili de düzenli bir dildir.
- $L1 \cap L2 = (L1' + L2')'$  DeMorgan kuralı.
  - Bu açılıma göre ve Kleene's teoremi ispatında anlatılan yöntemlerden birisini kullanarak kesişim kümesi nasıl bulunabilir ... ?



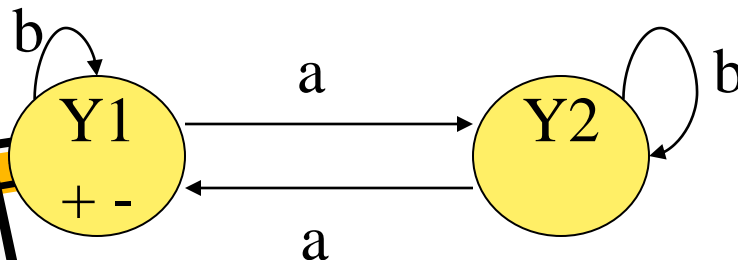
# Örnek



- L1: All strings with double a. (Çift a sembolü içeren kelimeler)
- L2: All strings with an even number of a's. (Çift sayıda a sembolü içeren kelimeler)



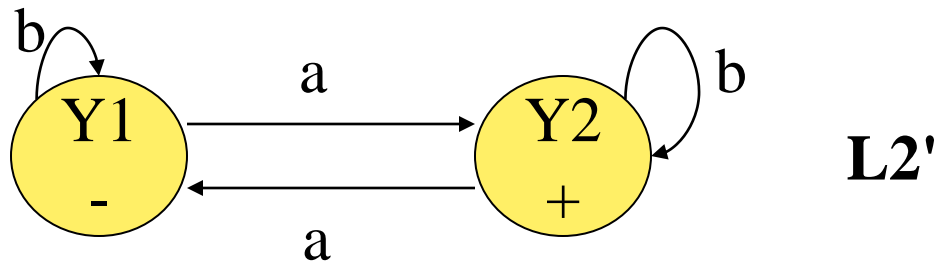
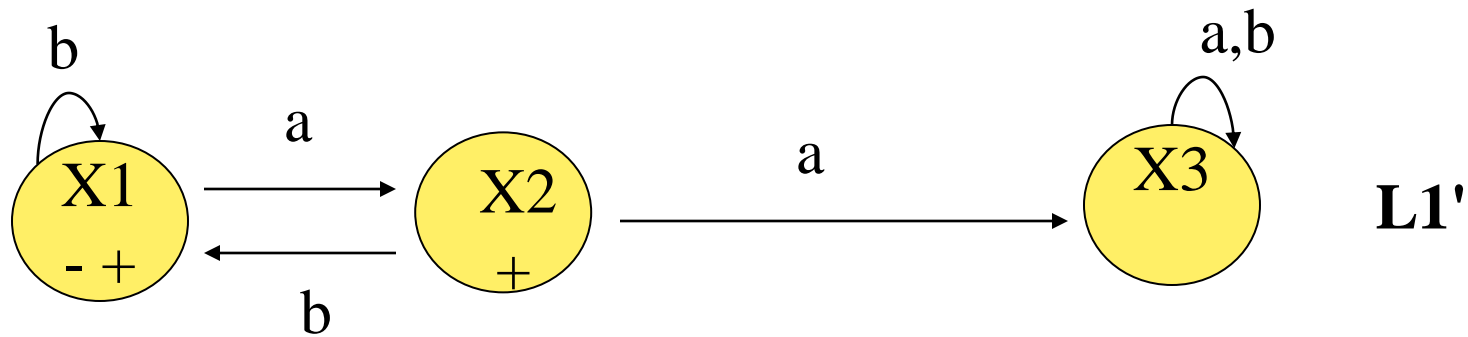
$(a+b)^*aa(a+b)^*$



$b^*(ab^*ab^*)^*$



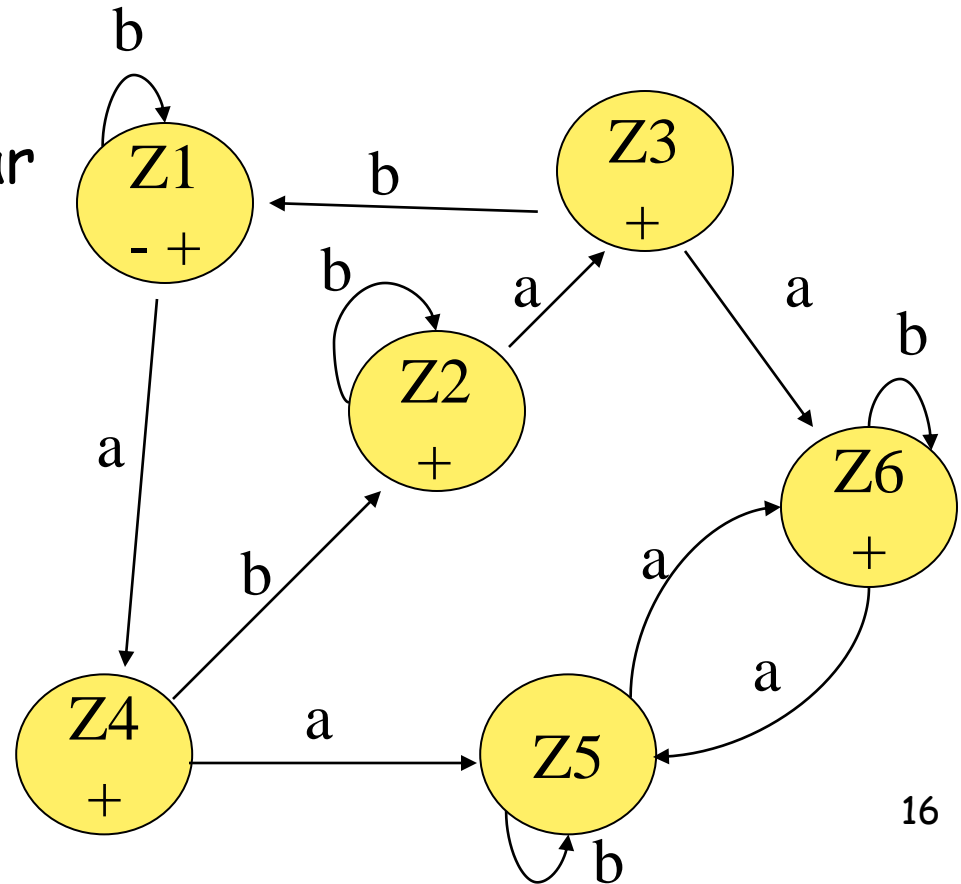
# Örnek - Tümlenileri Almak



# Örnek

- $L1' + L2'$  için F.A. bulunmalıdır. Kleene teoreminde verilen yöntem kullanılabilir ve aşağıdaki durumlar bulunur:

- $\pm Z1 = X1 \text{ OR } Y1$
- $+ Z2 = X1 \text{ OR } Y2$
- $+ Z3 = X2 \text{ OR } Y1$
- $+ Z4 = X2 \text{ OR } Y2$
- $Z5 = X3 \text{ OR } Y1$
- $+ Z6 = X3 \text{ OR } Y2$





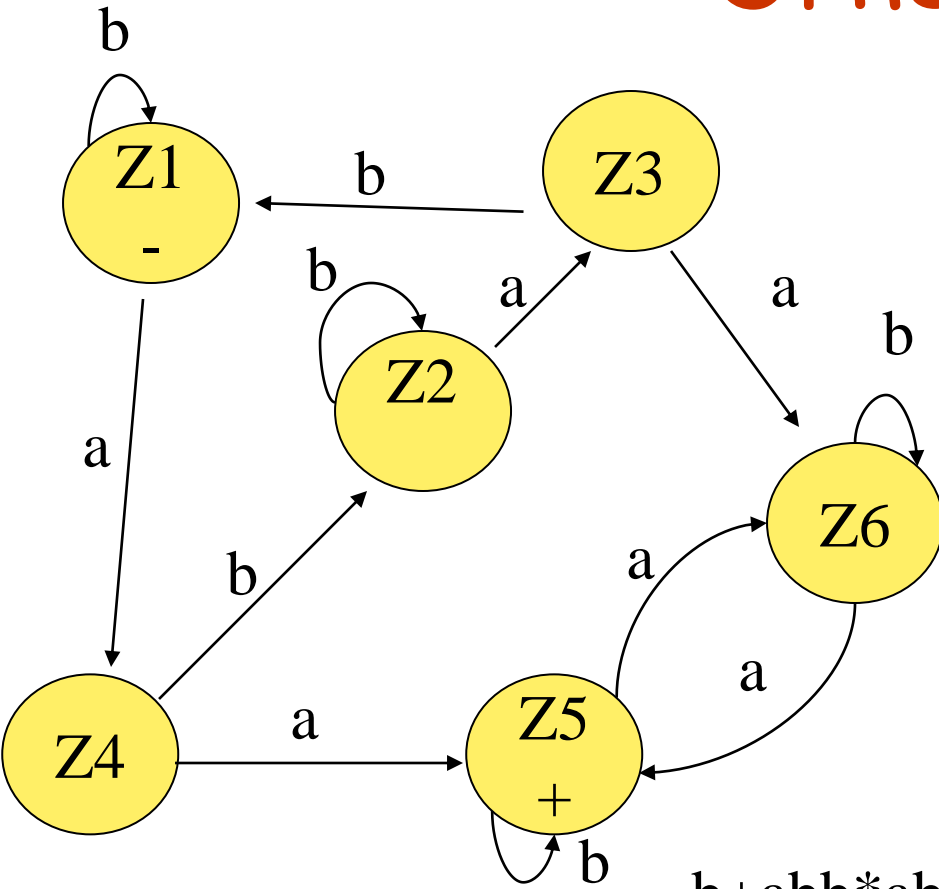
# Örnek



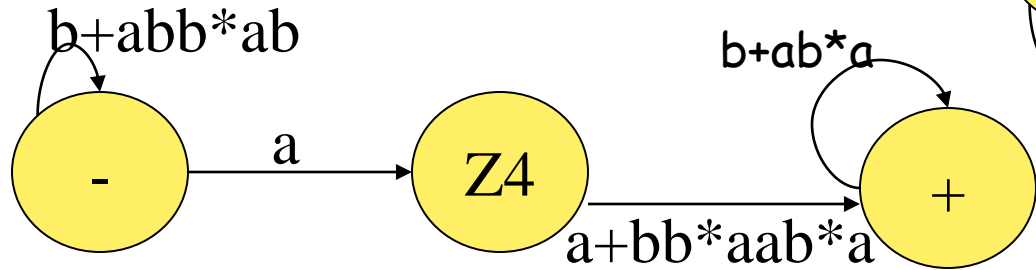
- Bir önceki adımda bulunan,  $L1' + L2'$  dilidir.  $L1 \cap L2 = (L1' + L2')'$  olduğu hatırlanırsa, bulunan F.A. modelinde + ve normal durumlar yer değiştirilmelidir. Böylece istenen kesişim kümesine karşılık gelen F.A. elde edilmiş olacaktır.
- Sonraki slayda bakınız..



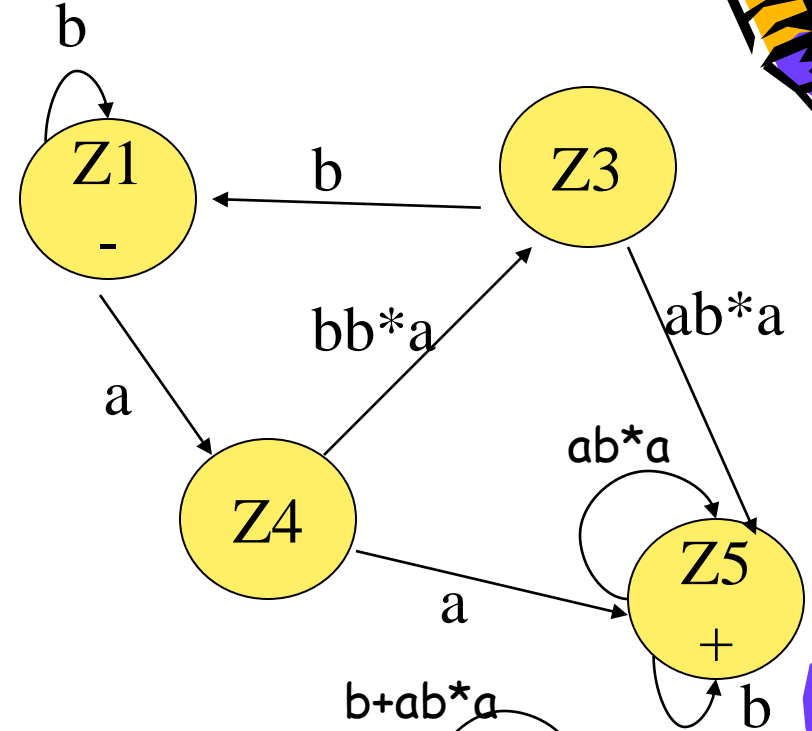
# Örnek $L1 \cap L2$



Z3 ü by-pass'larsak;



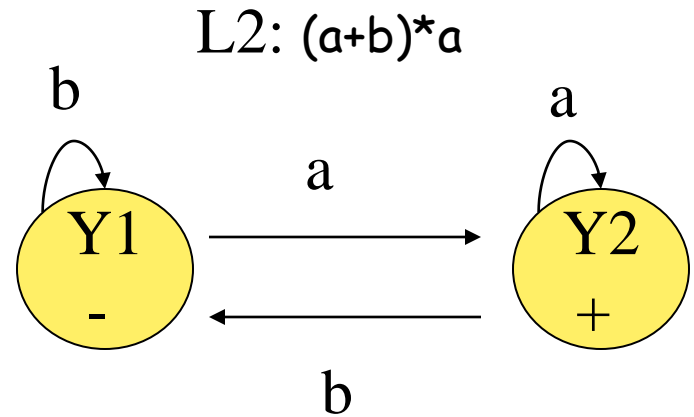
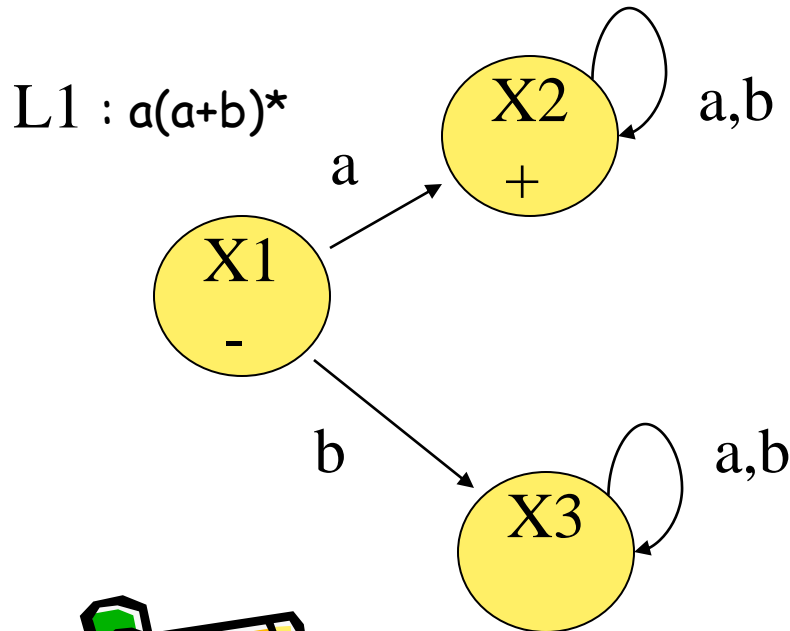
Z2 ve Z6 yı by-pass'larsak;



Tüm makine aşağıdaki R.E. Ye indirgenir:  
 $(b+abb^*ab)^*a(a+bb^*aab^*a)(b+ab^*a)^*$

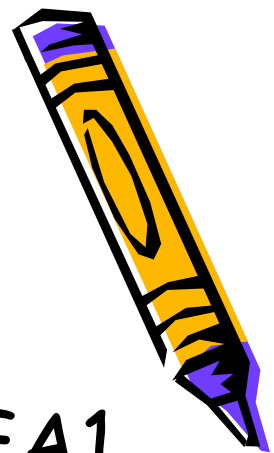
# Dil Kesişimi için Kısa Yol

- L1: a ile başlayan kelimelerin dili.
- L2: a ile biten kelimelerin dili.



$L1 \cap L2$ : a ile başlayan ve biten kelimelerin dilidir.

# Dil Kesişimi için Kısa Yol



- Her iki F.A., Kleene teoremindeki  $FA1 + FA2$  işlemine tabi tutulur. Ancak, amaç birleşim değil, kesişim kümesi elde etmek olduğu için "+" durumları belirlerken her iki modelde de + olanlar dikkate alınır.
- Buna göre elde edilen çözüm izleyen slaytta verilmektedir.

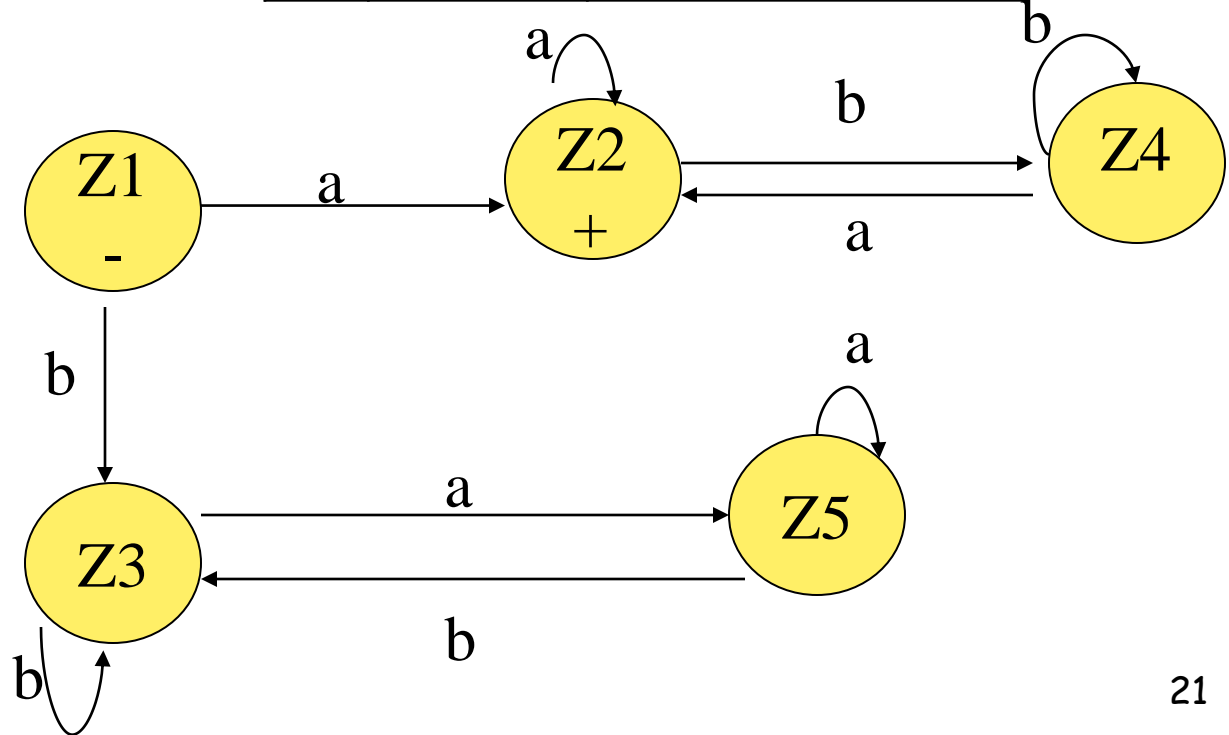


# Dil Kesişimi için Kısa Yol

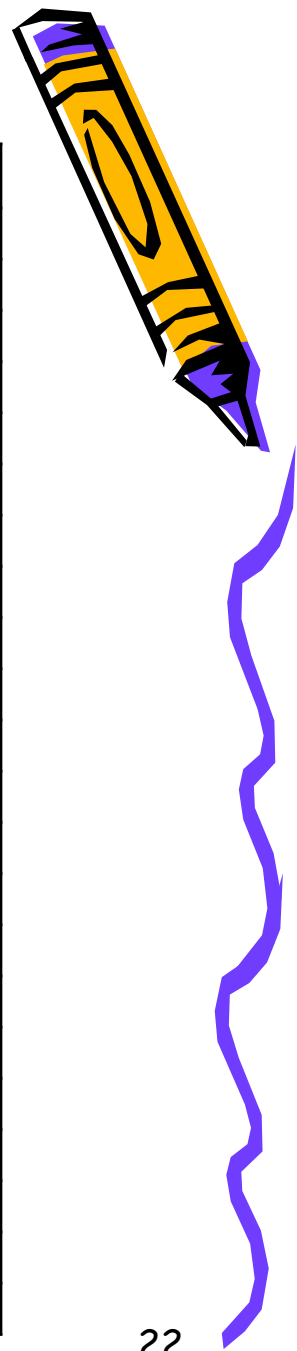
- $Z1 = X1 \text{ OR } Y1$
- $Z1:a \text{ } X2 \text{ OR } Y2 = +Z2$   
 $Z1:b \text{ } X3 \text{ OR } Y1 = Z3$
- $Z2:a \text{ } X2 \text{ OR } Y2 = +Z2$   
 $Z2:b \text{ } X2 \text{ OR } Y1 = Z4$
- $Z3:a \text{ } X3 \text{ OR } Y2 = Z5$   
 $Z3:b \text{ } X3 \text{ OR } Y1 = Z3$
- $Z4:a \text{ } X2 \text{ OR } Y2 = +Z2$   
 $Z4:b \text{ } X2 \text{ OR } Y1 = Z4$
- $Z5:a \text{ } X3 \text{ OR } Y2 = Z5$   
 $Z5:b \text{ } X3 \text{ OR } Y1 = Z3$

Bu durumlara göre, çizilen durum geçiş diyagramı kesişim dili içindir

	State	Read a	Read b
-Z1	$X_1 \text{ yada } Y_1$	$X_2 \text{ yada } Y_2$	$X_3 \text{ yada } Y_1$
Z2	$X_2 \text{ yada } Y_2$	$X_2 \text{ yada } Y_2$	$X_2 \text{ yada } Y_1$
Z3	$X_3 \text{ yada } Y_1$	$X_3 \text{ yada } Y_2$	$X_3 \text{ yada } Y_1$
Z4	$X_2 \text{ yada } Y_1$	$X_2 \text{ yada } Y_2$	$X_2 \text{ yada } Y_1$
Z5	$X_3 \text{ yada } Y_2$	$X_3 \text{ yada } Y_2$	$X_3 \text{ yada } Y_1$



# Problemler



Aşağıdaki düzenli dil çiftleri için; $L_1 \cap L_2$ tanımlayan R.E. Ve F.A. bulunuz		
	$L_1$	$L_2$
1	$(a+b)^*a$	$b(a+b)^*$
2	$(a+b)^*a$	$(a+b)^*aa(a+b)^*$
3	$(a+b)^*a$	$(a+b)^*b$
4	$(a+b)b(a+b)^*$	$b(a+b)^*$
5	$(a+b)b(a+b)^*$	$(a+b)^*aa(a+b)^*$
6	$(a+b)b(a+b)^*$	$(a+b)^*b$
7	$(b+ab)^*(a+\Lambda)$	$(a+b)^*aa(a+b)^*$
8	$(b+ab)^*(a+\Lambda)$	$(b+ab^*a)^*ab^*$
9	$(b+ab)^*(a+\Lambda)$	$(a+ba)^*a$
10	$(ab^*)^*$	$b(a+b)^*$
11	$(ab^*)^*$	$a(a+b)^*$
12	$(ab^*)^*$	$(a+b)^*aa(a+b)^*$
13	All strings of even length	$a(a+b)^*$
14	Even-Length strings	$(a+b)^*aa(a+b)^*$
15	Even-Length strings	$(b+ab)^*(a+\Lambda)$
16	Odd-Length strings	$a(a+b)^*$
17	Even-Length strings	EVEN-EVEN
18	i) Even-Length strings	Strings with an even number of a's
	ii) Even-Length strings	Strings with an odd number of a's
19	i) Even-Length strings	Strings with an odd number of a's an and an odd number of b's
	ii) Even-Length strings	Strings with an odd number of a's an and an even number of b's

