

CENG 306 Biçimsel Diller ve Otomatlar

Formal Languages and Automata

Konular

- Düzenli ve Düzenli olmayan diller - Regular and Non-regular Languages
- Pumping Lemma and Its Application to RE/Non-RE Languages

Regular and Non-regular Languages

- Düzenli diller bazı işlemler (**birleşim, kesişim, Kleene star, complement, concatenation**) için kapalıdır.
- Düzenli diller, düzenli ifadeler (regular expressions) ile veya sonlu otomatlar ile (deterministic veya nondeterministic) belirlenebilir.

Regular and Non-regular Languages

- Düzenli diller bazı işlemler (**birleşim, kesişim, Kleene star, complement, concatenation**) için kapalıdır.
- Düzenli diller, düzenli ifadeler (regular expressions) ile veya sonlu otomatlar ile (deterministic veya nondeterministic) belirlenebilir.

Örnek:

$\Sigma = \{0, 1, 2, 3, \dots, 9\}$, $L \subseteq \Sigma^*$ olarak tanımlı olsun ve sadece 2'ye veya 3'e bölünebilen ve önünde 0 olmayan pozitif sayılara sahip olsun.

$(0, 3, 6, 244 \in L \text{ ve } 1, 03, 00 \notin L)$

Bu dilin regular olduğunun ispatı 4 kısımda yapılabilir.

Regular and Non-regular Languages

Örnek: (devam)

- $\Sigma = \{0, 1, 2, 3, \dots, 9\}$, $L \subseteq \Sigma^*$, 2'ye veya 3'e bölünebilen ve önünde 0 olmayan pozitif sayılara sahiptir

1 L_1 dili pozitif tamsayıların kümesi olsun

$$L_1 = 0 \cup \{1, 2, \dots, 9\} \Sigma^* \text{ (regular)}$$

Regular and Non-regular Languages

Örnek: (devam)

- $\Sigma = \{0, 1, 2, 3, \dots, 9\}$, $L \subseteq \Sigma^*$, 2'ye veya 3'e bölünebilen ve önünde 0 olmayan pozitif sayılara sahiptir

1) L_1 dili pozitif tamsayıların kümesi olsun

$$L_1 = 0 \cup \{1, 2, \dots, 9\} \Sigma^* \text{ (regular)}$$

2) L_2 dili 2'ye bölünebilen pozitif tamsayıların kümesi olsun

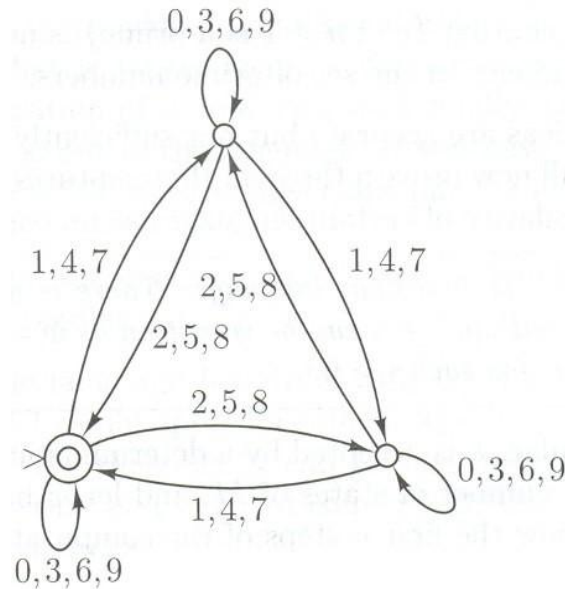
$$L_2 = L_1 \cap \Sigma^* \{0, 2, 4, 6, 8\} \text{ (regular)}$$

Regular and Non-regular Languages

Örnek: (devam)

- $\Sigma = \{0, 1, 2, 3, \dots, 9\}$, $L \subseteq \Sigma^*$, 2'ye veya 3'e bölünebilen ve önünde 0 olmayan pozitif sayılara sahiptir

- 3) 3'e bölünebilen pozitif tamsayıların kümesi olan dili aşağıdaki otomat tanır (*regular*)

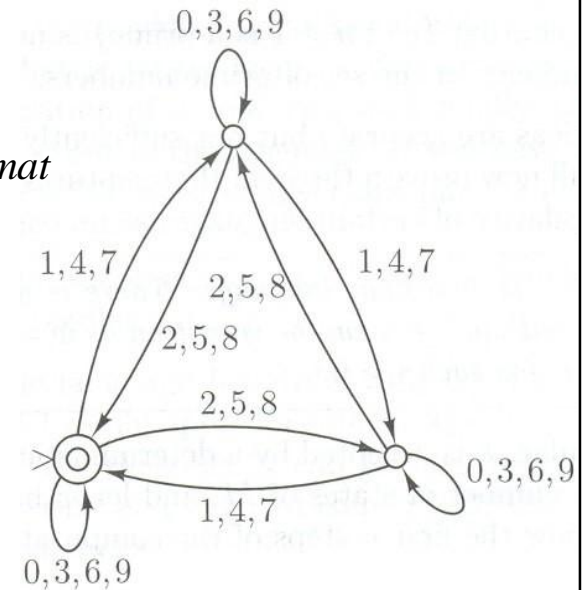


Regular and Non-regular Languages

Örnek: (devam)

- $\Sigma = \{0, 1, 2, 3, \dots, 9\}$, $L \subseteq \Sigma^*$, 2'ye veya 3'e bölünebilen ve önünde 0 olmayan pozitif sayılara sahiptir

- 3) 3'e bölünebilen pozitif tamsayıların kümesi olan dili yandaki otomat tanır (regular)



- 4) L_3 bu otomat ile L_1 'in kesişimidir. Sonuç olarak elde edilen dil regular dildir ve

$L = L_2 \cup L_3$, şeklinde ifade edilir.

Regular and Non-regular Languages

- Bir dilin düzenli dil olduğunu gösteren yöntemler vardır (ve öğrendik) ancak **düzenli olmadığını göstermek için özel araçlara ihtiyaç vardır.**

Regular and Non-regular Languages

- Bir dilin düzenli dil olduğunu gösteren yöntemler vardır ancak düzenli olmadığını göstermek için özel araçlara ihtiyaç vardır.
 - **İki özellik** tüm düzenli dillerde bulunur, ancak düzenli olmayan dillerde bulunmaz;
 - Bir string soldan sağa doğru taranırken o string'in ilgili dile ait olup olmadığını belirlemek için gereken **toplam hafızanın bir sınırı vardır**, bu sınır sabittir ve ilgili string'e değil o dile bağlıdır.
- Örnek:** $L = \{ a^n b^n : n \geq 0 \}$ dili regular değildir. b'leri okumaya başladığında kaç tane a okuduğu belli değildir ve n için bir sınır değeri yoktur.

Regular and Non-regular Languages

- Bir dilin düzenli dil olduğunu gösteren yöntemler vardır ancak düzenli olmadığını göstermek için özel araçlara ihtiyaç vardır.
- **İki özellik** tüm düzenli dillerde bulunur, ancak düzenli olmayan dillerde bulunmaz;
 - Bir string soldan sağa doğru taranırken o string'in ilgili dile ait olup olmadığını belirlemek için gereken **toplam hafızanın bir sınırı vardır**, bu sınır sabittir ve ilgili string'e değil o dile bağlıdır.

Örnek: $L = \{ a^n b^n : n \geq 0 \}$ dili regular değildir. b'leri okumaya başladığında kaç tane a okuduğu belli değildir ve n için sınır değer yoktur.
 - Sonsuz sayıda string'e sahip olan düzenli diller, döngüye sahip **otomatlar** veya Kleene star içeren **regular expression'lar** tarafından gösterilebilir.

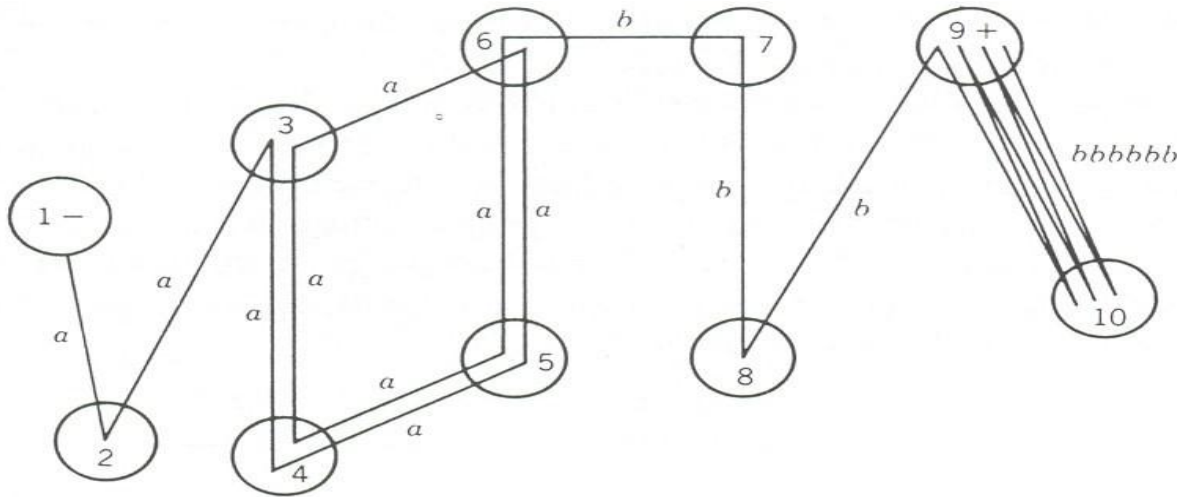
Örnek: $L = \{ a^n : n \geq 1 \text{ asal sayı} \}$ regular değildir. (ispatı daha sonra verilecektir.)

Regular and Non-regular Languages

- $L = \{ a^n b^n : n \geq 0 \}$ şeklinde tanımlanmış olsun. Eğer bu dil regular ise bir sonlu otomat tarafından tanınır.
- $n = 96$ için $a^{96} b^{96}$ olur. Toplam 95 duruma sahip bir otomat bu dili tanıyor olsun.
- En az bir noktada yol daha önce geçtiği durumlara geri döner ve tekrar geçer.
- Bu tekrar geçişlere **loop** adı verilir.

Regular and Non-regular Languages

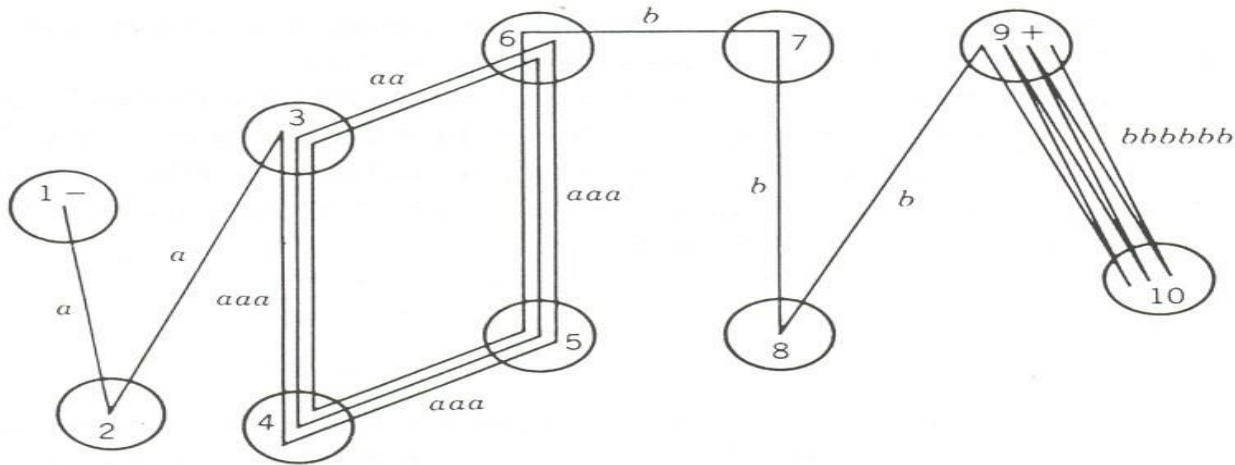
- $L = \{a^n b^n : n \geq 0\}$ şeklinde tanımlanmış olsun. Eğer bu dil regular ise bir sonlu otomat tarafından tanınır.
- $n = 96$ için $a^{96}b^{96}$ olur. Toplam 95 duruma sahip bir otomat bu dili tanıyor olsun.
- En az bir noktada yol daha önce geçtiği durumlara geri döner ve tekrar geçer.
- Bu tekrar geçişlere **loop** adı verilir.
- Aşağıdaki 10 durumlu otomat a^9b^9 için geçişleri vermektedir. Otomatta sadece geçilen yollar verilmiştir.



- $a^{13}b^9$ için nasıl bir yol izlenir ?

Regular and Non-regular Languages

- $a^{13}b^9$ için 6-3-4-5 yolunda bir tur daha atılır.
- $a^9(a^4)^mb^9$, $m \geq 0$ şeklinde tanımlanan tüm stringleri tanır. (Örn : $a^{25}b^9$)
- Bu şekilde bir string'in önündeki ve sonundakine bakmadan ortasına ekleme yapmaya **pumping** denilmektedir.
- string'in önündeki ve/veya sonundaki katar **boş olabilir**.



Regular and Non-regular Languages

Teorem: Sonsuz sayıda string'e sahip bir regular L dilinde, kendisini tanıyan otomatın durum sayısından fazla sembole sahip tüm stringler için x, y, z şeklinde üç substring tanımlanabilir ve tüm stringler xy^nz olarak parçalarla ifade edilebilir. ($n = 1, 2, 3, \dots$)

ispat: L dilinde sonsuz string olduğu için bazı w stringleri kendisini tanıyan otomatın durum sayısından daha fazla sembole sahiptir. Bu stringler için otomat üzerinde loop oluşur.

w string'leri x, y, z olarak üç kısımda incelenebilir;

1. x yeniden geçilen ilk duruma kadar olan sembolleri içersin. x eğer boş ise loop başlangıç durumunu da içine almıştır.
2. y string'i, x 'den hemen sonra başlayıp loop'un sonuna kadar olan kısmı içersin. Bir loop olduğu için y boş olamaz.
3. z string'i loop'tan hemen sonra başlayıp w string'inin sonuna kadar olan kısmı içersin. z boş ise loop sonuç durumunu da içine almıştır.

Regular and Non-regular Languages

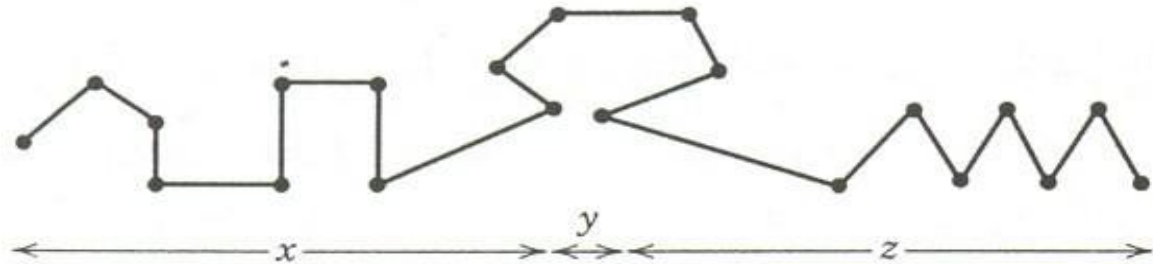
Teorem: (devam) Sonsuz sayıda string'e sahip bir regular L dilindeki tüm stringler için

x, y, z şeklinde üç string tanımlanabilir ve

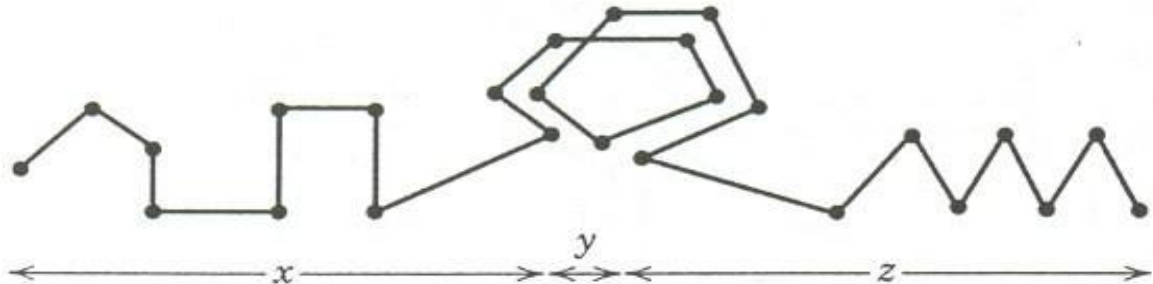
tüm stringler xy^nz olarak parçalarla ifade edilebilir. ($n = 1, 2, 3, \dots$)

$xyz, xyyz, xyxyz, \dots xy^nz$ string'leri tanınır.

xyz

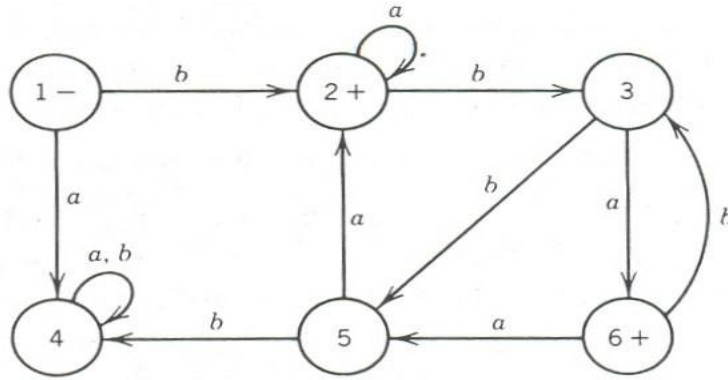


$xyyz$



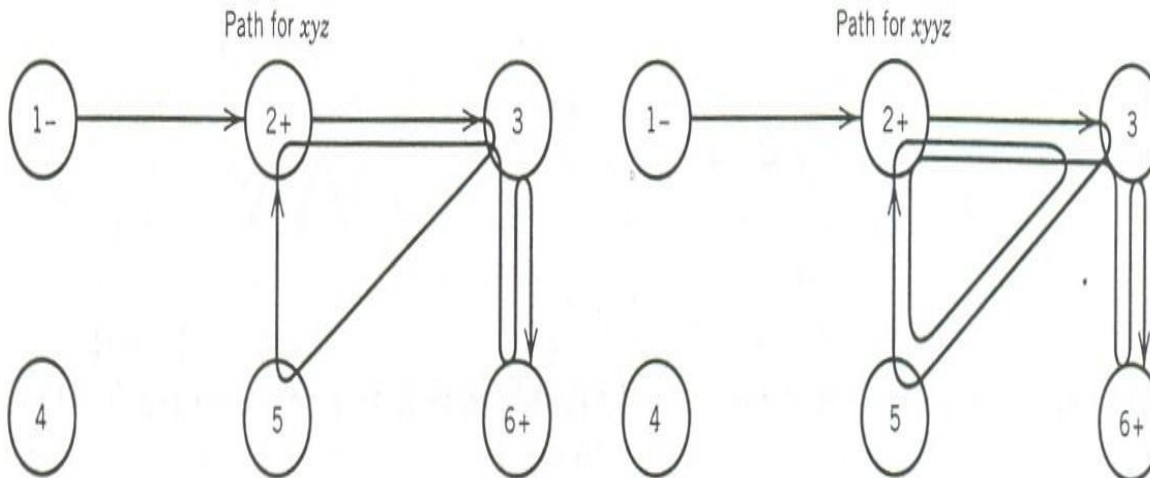
Regular and Non-regular Languages

Örnek : Aşağıdaki otomatta “-” başlangıç ve “+” sonuç durumlarını göstermektedir ve $w = bbbababa$ string’ini tanır.



Durum sayısından fazla sembol olduğu için loop olmak zorundadır.

$x = b$, $y = bba$ ve $z = baba$ alınabilir.



Regular and Non-regular Languages

Örnek: $L = \{a^n b^n : n \geq 0\}$ dili regular değildir. Eger L regular olsaydı tüm string'ler x, y, z olarak üç parçaya ayrılabilirdi.

Tipik bir string $aaa.....aaaabbbb.....bbb$ şeklindedir.

- Eger y , a 'ları içerirse $xyyz$ şeklindeki string daha fazla a içerir ve elde edilen string L diline ait değildir.
- Eger y , b 'leri içerirse $xyyz$ şeklindeki string daha fazla b içerir ve elde edilen string L diline ait değildir.
- Eger y , ortadaki a ve b 'leri içerirse $xyyz$ string'i iki tane " ab " substringine sahiptir. L dilindeki tüm stringler sadece bir tane " ab " substringine sahip olabileceği için bu string L diline ait değildir.

Bunların sonucu olarak L dili regular değildir.

Pumping Lemma

Teorem: L regular dil olsun. Dile bağı olarak seçilen bir

$n \geq 1$ için $|w| \geq n$ olacak şekilde bir $w \in L$ string'i vardır ve

$$w = xyz,$$

$$y \neq \epsilon,$$

$$|xy| \leq n$$

olmak üzere yeniden yazılabilir. Her $i \geq 0$ için $xy^i z \in L$ olur.

İspat: L regular dil olduğundan deterministic finite automata M tarafından

kabul edilir. M automata'nın n duruma sahip olduğunu varsayalım ve

$|w| = m, m \geq n$ olsun.

M automata'nın ilk m adımı aşağıdaki gibidir;

$$(q_0, w_1 w_2 \dots w_m) \xrightarrow{M} (q_1, w_2 \dots w_m) \xrightarrow{M} \dots \xrightarrow{M} (q_m, \epsilon)$$

Pumping Lemma

ispat: (devam)

$$(q_0, w_1w_2...w_m) \vdash_M (q_1, w_2...w_m) \vdash_M \dots \vdash_M (q_m, e)$$

- Burada q_0 başlangıç durumu ve $w_1w_2...w_m$ ilk m semboldür
- M , n adet duruma sahiptir ancak $m \geq n$ adet konfigürasyon vardır.
- Pigeonhole prensibine göre $0 \leq i < j \leq m$ olacak şekilde i ve j sayıları vardır ve $q_i = q_j$ olur.
- $y = w_iw_{i+1}...w_j$ vardır ve q_i durumundan tekrar q_i durumuna geçiş yapar.
- $i < j$ olduğu için y boş olamaz.
- y string'i w 'dan atılarak veya istendigi kadar tekrar edilerek bulunan stringler'de M tarafından tanınır.
- $i \geq 0$ olmak üzere $xy^iz \in L$ olur.

Pumping Lemma

- Bir dilin regular olup olmadığını belirlemek için kullanılır.
 - Öncelikle bir n sayısı belirlenir. (dili tanıyan ve en az duruma sahip otomat)
 - n 'den uzun bir w string'i belirlenir.
 - w string'i xyz şeklinde parçalanır.
 - her $i \geq 0$ degeri için $xy^iz \in L$ olacak şekilde bir xy^iz bulunuyorsa L regulardır..
 - Eger bu şekilde bir değer hiçbir xy^iz için bulunamıyorsa L regular değildir.

Pumping Lemma

Örnek: (tekrar) $L = \{ a^i b^i : i \geq 0 \}$ dili regular değildir.

ispat:

- $w = a^n b^n \in L$ olduğunu varsayalım.
- Pumping teoreminden $w = xyz$ yazılabilir.
- $|xy| \leq n$ alınırsa ve $y \neq \epsilon$, $y = a^i$, $i > 0$ değerleri için,
- y 'nin çıkarıldığı string olan $xz = a^{n-i} b^n$ olur ve L diline ait değildir.
- Bu sonuç $y = b^i$, $i > 0$ için de aynı şekilde geçerlidir.
- Böylece bu dil regular değildir.

Pumping Lemma

Örnek: $L = \{ a^k : k \text{ asal sayı} \}$ dili regular değildir.

ispat:

- Pumping teoreminden $w = xyz$ yazılabilir.
- $p, r \geq 0, q > 0$ için $x = a^p, y = a^q$, ve $z = a^r$ olsun.
- Teoremden $xy^n z \in L$ olduğundan $n \geq 0$ için $p + nq + r$ asal sayı olmak zorundadır. (her n sayısı için sağlanmalıdır!)
- Özel olarak $n = p + 2q + r + 2$ için $p + nq + r = (q + 1)(p + 2q + r)$ olur.
- Burada iki çarpan da 1'den büyüktür ve böylece $p + nq + r$ asal sayı olamaz!
- $n = p + 2q + r + 2$ için elde edilen string L diline ait değildir.
- Öyle ise L dili regular değildir.

Pumping Lemma

Örnek: $L = \{ w \in a, b \}^ : w \text{ eşit sayıda } a \text{ ve } b \text{'ye sahiptir} \}$ dili regular değildir.*

ispat:

- *Bu ispat closure özelliği ile yapılabilir.*
- *Eğer L dili regular ise, regular bir dil ile kesişim işlemi kapalı olur.*
- *Ancak $L \cap a^*b^*$ kesişiminin sonucunda elde edilen dil $\{a^n b^n : n \geq 0\}$ olur.*
- *$\{a^n b^n : n \geq 0\}$ regular dil olmadığı için L dili de regular değildir.*

Örnek: $L = \{ab, abba, aabb, abab, aaabbb, \dots\}$,

*$L(a^*b^*) = \{a, b, aa, ab, aab, bb, aabb, abbbb, aaabbb, \dots\}$*

*$L \cap a^*b^* = \{ab, aabb, aaabbb, \dots\}$*