

Biçimsel Diller ve Otomata Teorisi

Hafta 6: Düzenli İfadeler (I. Bölüm)

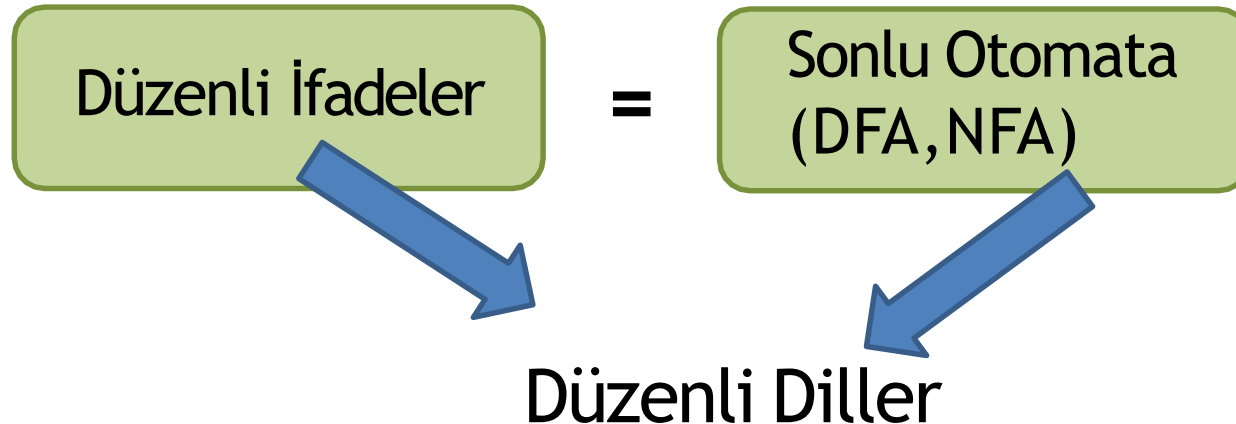
Hafta 6

Plan

1. Düzenli Diller
2. Düzenli Operatörler
3. Düzenli İfade Örnekleri
 - i. R' 'den L' 'ye
 - ii. L' 'den R' 'ye
4. Online Düzenli İfade Programları
5. Sonlu Otomatalardan Düzenli İfadelere

Düzenli İfadeler (Regular Expressions) (Regex)

- ➡ Düzenli ifadeler otomatalar gibi dilleri ifade etmenin bir yoludur. Fakat otomatalar daha makine gibi iken, düzenli ifadeler, daha çok program sintaksı gibidir. Hatta düzenli ifadeler metin içi arama yapmak için kullanılan bir programlama dili olarak düşünülebilir.
- ➡ Düzenli ifadeler aslında sonlu otomataya denktir; sonlu otomatalarda düzenli ifadelere denktir (Kleene Teoremi)
- ➡ Düzenli ifadeler ve sonlu otomata birlikte düşünüldüğünde ‘Düzenli Diller’ (Regular Languages) oluştururlar.



Düzenli Diller (Regular Languages)

Düzenli diller, özyinelemeli (recursive) olarak tanımlanır:
 Σ , alfabetini kullanarak L düzenli dilini oluşturalım.

1. Temel: ε (boş kelime) L dilinin elemanıdır.
2. Tümevarım: her $w \in L$, her $k \in \Sigma$, için $wk \in L$ dir.

ör. $\Sigma = \{0,1\}$ olsun. Bu alfabeden üretilecek düzenli dil.

1. $L = \{\varepsilon\}$
2. $\varepsilon \in L$, için $\varepsilon 0 = 0 \in L$
 $\varepsilon 1 = 1 \in L$ } $L = \{\varepsilon, 0, 1\}$
3. $0 \in L$, için , $00 \in L$
 $01 \in L$
 $1 \in L$, için , $10 \in L$
 $11 \in L$ } $L = \{\varepsilon, 0, 1, 00, 01, 10, 11\}$
4.

Düzenli Operatörler (Regular Operations)

Düzenli ifadeler, düzenli operasyonlar kullanılarak inşa edilirler. Bu operasyonlar 3 tanedir.

L ve M iki dil olsun.

1. Birleşim (Union): $L \cup M = \{x \mid x \in L \vee x \in M\}$ (Birleşimin, elemanları ya L 'nin, ya M 'nin, yada her ikisinin birden elemanıdır).

Not bazı kaynaklarda birleşim işareti olarak '+' kullanılıyor.

2. Bitiştirme (Concatenation): $LM = L \circ M = \{xy \mid x \in L \wedge y \in M\}$
(Bitiştirmenin elemanları L 'den ve M 'den birer elemanın yanyana getirilmesiyle oluşur.)

ör. $L = \{0, 01\}, M = \{\varepsilon, b, bb\}; LM = \{0, 0b, 0bb, 01, 01b, 01bb\}$

3. Yıldız (Kleene Star): $L^* = \{x_1x_2 \dots x_k \mid k \geq 0, \forall x_i \in L\}$

Yani L^* 'ın elemanları, L 'nin herhangi elemanlarının herhangi sayıda bitleştirilmesiyle oluşur.

Yıldız (Kleene Star)

ör. $L = \{0,1\}$ iken $L^* = \{\varepsilon, 0,1,00,11,01,10,000,111,011,101,\dots\}$

ör. $L = \{\text{iyi, kötü}\}$

$L^* = \{\varepsilon, \text{iyi, kötü, iyiyi, kötükötü, iyikötü, kötüiyi, iyiiyyi, \dots\}$

Not 1. L dili ne olursa olsun L^* her zaman boş kelime ε içerir.

Not 2. 1. hafta notlarından Σ^* ' ı hatırlayın. Σ^* , Σ alfabesinin harfleri kullanılarak oluşturulabilecek bütün kelimelerin kümesi idi. Aynı şekilde L^* ' da, L dilinin kelimeleri kullanılarak oluşturulabilecek bütün kelimelerin kümesidir.

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots$$



ör. $L = \{0,11\}$ iken $L^2 = \{00,1111,011,110.\}$



11 elemanı iki defa **tekrar** ediyor!

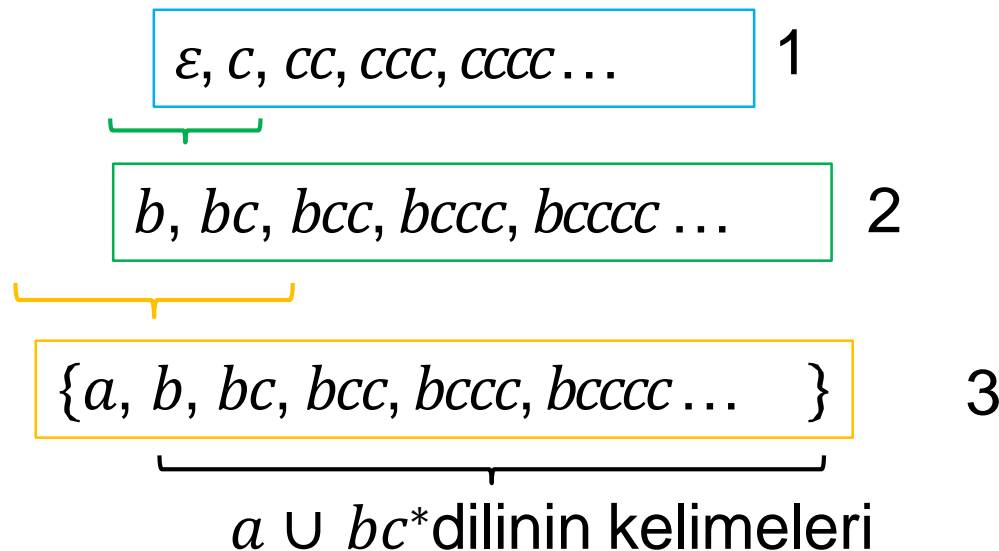
ör. $A = \{0\}$ ise $A^* = \{\varepsilon, 0, 00, 000, 0000, 00000, \dots\} = 0^*$

Not. Düzenli ifadelerde daha çok tek harfli diller, örneğin $A = \{0\}$ (yada bunu direk 0 ile göstereceğiz) ve bu tek harfli dillerin yukarıda saydığımız operatörler yardımıyla oluşturdukları yeni dillerle ilgileneceğiz, örneğin, 0 ve 1 dilleriyle oluşturulmuş $0 \cup 1^*$ dili.

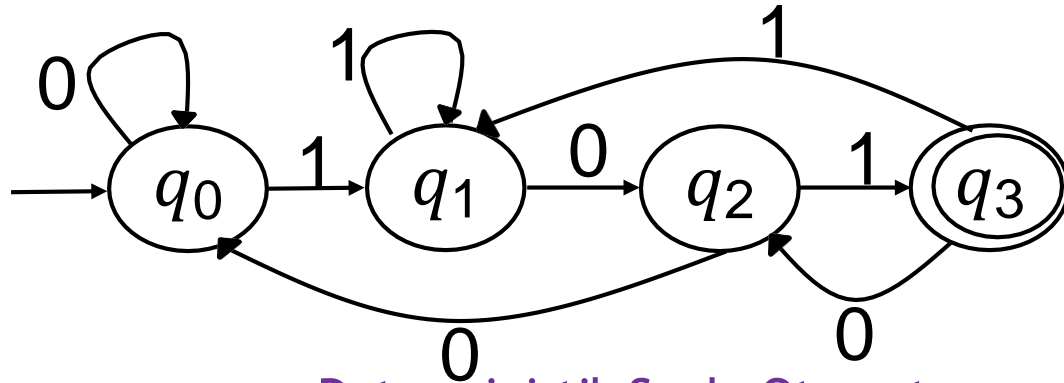
Düzenli Operasyonların İşlem Sırası:

1. Yıldız, 2. Bitiştirme, 3. Birleşim ($*$ > \circ > \cup)

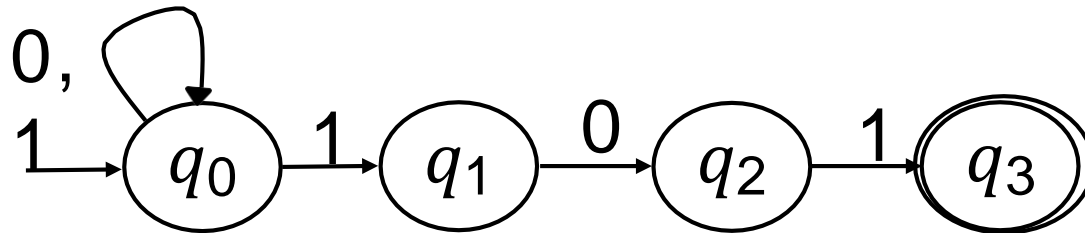
ör. $a \cup bc^* = a \cup (b(c^*))$



ör. $\Sigma = \{0,1\}$ alfabesi kullanılarak üretilen kelimelerden sonu '101' ile biten kelimeleri kabul eden deterministik ve nondeterministik sonlu otomata ve düzenli ifade:



Deterministik Sonlu Otomata



Nondeterministik Sonlu Otomata

$$R = (0 \cup 1)^* 101$$

Düzenli İfade



Düzenli İfade Örnekleri - I (R'den L'ye):

1. $R = (0 \cup 1)01^*$: 0 yada 1 ile başlayan, ikinci harfi 0 olan, sonra (varsa) tüm harfleri 1 olan kelimelerin dilini ifade eden düzenli ifade. Bu dil şu şekilde gösterilebilir:

$$L = \{00, 10, 001, 101, 0011, 1011, 00111, 10111, \dots\}$$

2. $R = 0^*10^* = \{\varepsilon, 0, 00, 000, \dots\} \{1\} \{\varepsilon, 0, 00, 000, \dots\}$
 $= \{\varepsilon 1 \varepsilon, \varepsilon 1 0, \varepsilon 1 00, \dots, 01 \varepsilon, 01 0, 01 00, \dots, 001 \varepsilon, 001 0, \dots\}$
 $= \{1, 10, 100, \dots, 01, 010, 0100, \dots, 001, 0010, \dots\}$
 $= L = \{w \mid w \text{ yalnızca 1 defa 1 içerir}\}.$

3. $\Sigma = 0, 1$ iken $R = \Sigma^*1\Sigma^* = \{0, 1\}^*1\{0, 1\}^*$
 $R = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000\} \{1\} \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, \dots\}$
 $R = \{1 \ 10 \ 11, 100, \dots, 01, 010, 0100, \dots, 11, 110, 111, 1100, \dots\}$
 $= L = \{w \mid w \text{ en az 1 defa 1 içerir}\}.$

Düzenli İfade Örnekleri - I (R'den L'ye):

$$\begin{aligned} 4. R &= ((0 \cup 10 \cup 1))^* = (\{0, 1\}\{0, 1\})^* = \{00, 01, 10, 11\}^* \\ &= \{\varepsilon, 00, 01, 10, 11, 0000, 0101, 1010, 1111, 0001, \dots, 000110, \dots\} \\ &= L = \{w \mid w' \text{nin uzunluğu çift sayıdır.}\}. \end{aligned}$$

5. $\Sigma = \{a, b, c, \dots, z\}$ olsun. Bu durumda
 $\Sigma^* = \{\varepsilon, a, \dots, z, aa, \dots, zz, ab, \dots, az, \dots, bct, \dots, sampiyon, \dots\}$
 Σ alfabetiyle oluşturulabilecek tüm kelimelerin kümesi olur.
Böylece

$$R = \Sigma^* \text{sivas} \Sigma^*$$

içinde 'sivas' altkelimesi geçen tüm kelimelerin dilini temsil eder.

Düzenli İfade Örnekleri - II (L'den R'ye):

1. $L = \{w \mid w \text{ yalnızca 2 defa 0 içerir}\}$. Bu durumda L 'nin kelimeleri şu formda olur:

$$\underbrace{1 \dots 1}_{1^*} 0 \underbrace{1 \dots 1}_{1^*} 0 \underbrace{1 \dots 1}_{1^*}$$

$$R = 1^*01^*01^*$$

2. $L = \{w \mid w \text{'nin ilk ve son harfi aynıdır}\}$.

Düşünmemiz gereken durumlar:

i. 0 ile başlayıp 0 ile biten kelimeler: $0\Sigma^*0$

ii. 1 ile başlayıp 1 ile biten kelimeler: $1\Sigma^*1$

iii. 0 kelimesi

iv. 1 kelimesi

Su halde cevap bu dört durumun birleşimidir:

$$R = 0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$$

Düzenli İfade Örnekleri - II (L'den R'ye):

3. $L = \{w \mid w \text{ 'da yanyana 0 ve yanyana 1 olmaz}\}$. Bu durumu sağlayan dört durum vardır:

- i. $01010101 \dots = (01)^* \text{ (0 ile başlayıp 1 ile bitenler)}$
- ii. $010101 \dots 0 = (01)^* 0 \text{ (0 ile başlayıp 0 ile bitenler)}$
- iii. $10101010 \dots = (10)^* \text{ (1 ile başlayıp 0 ile bitenler)}$
- v. $101010 \dots 1 = (10)^* 1 \text{ (1 ile başlayıp 1 ile bitenler)}$

cevap bu dört durumun birlesimidir:

$$R = (01)^* \cup (01)^* 0 \cup (10)^* \cup (10)^* 1$$

Online Düzenli İfade Programları

Oluşturduğumuz düzenli ifadeleri aşağıdaki online programları kullanarak test edebiliriz

- <http://regex101.com>
- <http://rubular.com>

Not. Bu programlarda { } için []; birleşim \cup için | kullanın.

Düzenli İfadelerin Gerçek Hayattaki Kullanımları

- Düzenli İfadeler, Linux, Perl, Python gibi dillerde kaynak kodun program sintaksına uygunluğu kontrol edilirken derleyici (compiler) tarafından kullanılırlar.
- Kullanıcı tarafından girilen email adresinin istenilen formata uygunluğu test edilirken kullanılırlar. Bunun için kullanılan düz. ifade aşağı yukarı şöyle bir şeydir:

`/^([a-z0-9_\. -]+)@([a-z0-9_\. -]+\.)([a-z]{2,6})$/`

Uygulama Alanları

Arama motorları

- Bilgi edinim (Information retrieval)
- Kelime işleme (Word processing)
- Derlem (corpus) içinde frekans hesaplama
- Veri doğrulama (Data validation)
- Sözdizim belirginleştirme (Syntax highlighting)
- ...

Kullanım Amaçları

- Nasıl yazıldığını tam olarak bilmediğimiz bir sözcüğü aratabiliriz.
- Bir metnin içindeki, özel bir forma uyan parçaları bulabiliriz.
- Aldığımız bir metnin, belli bir yapıya uyup uymadığını kontrol edebiliriz.
- Belli formatta aldığımız bir veriyi, istediğimiz başka bir formata çevirebiliriz.

Sonlu Otomatalardan Duzenli Ifadelere

Daha once gordugumuz sonlu otomatalari (DFA, NFA) düzenli ifadelerle gosterebiliriz.

