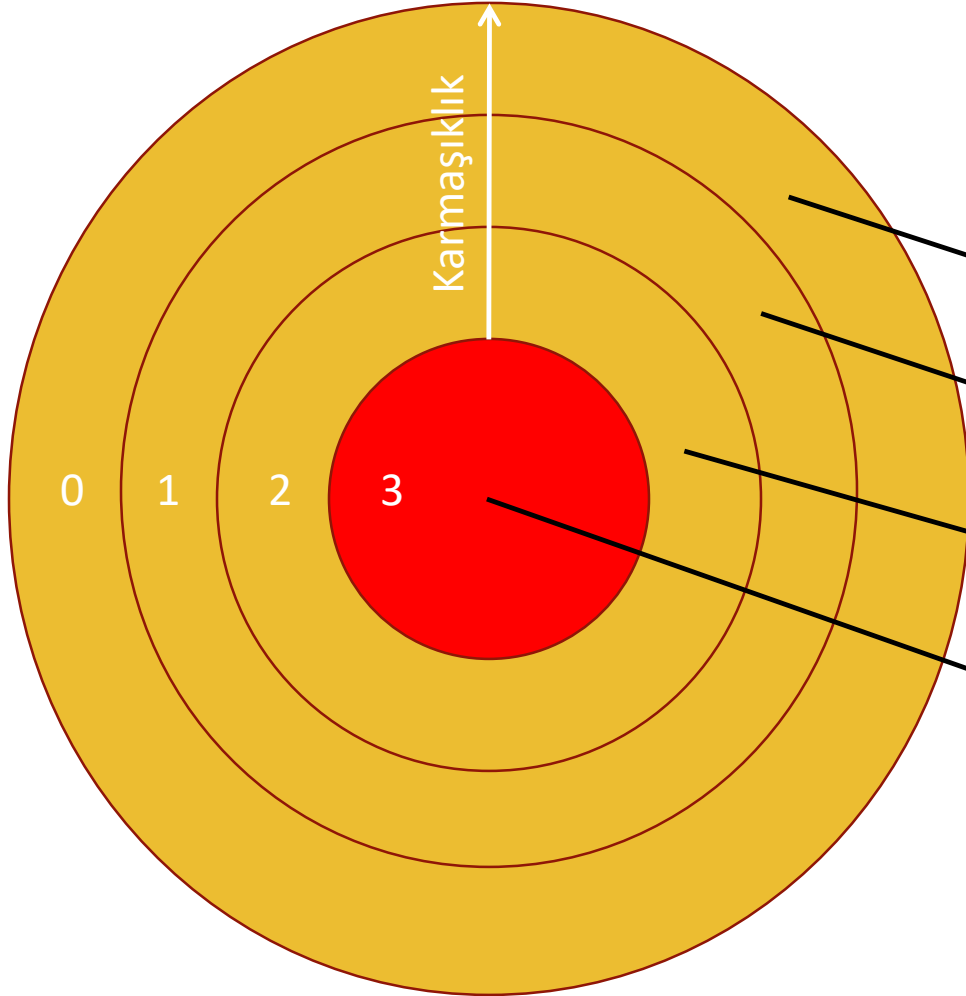


FORMEL DİLLER VE SOYUT MAKİNALAR

Hafta 3

CHOMSKY HİYERARŞİSİ



Özyinelemeli - Sayılabilir Diller : Turing Makinesi
(Recursively – Enumerable Languages : Turing Machine)

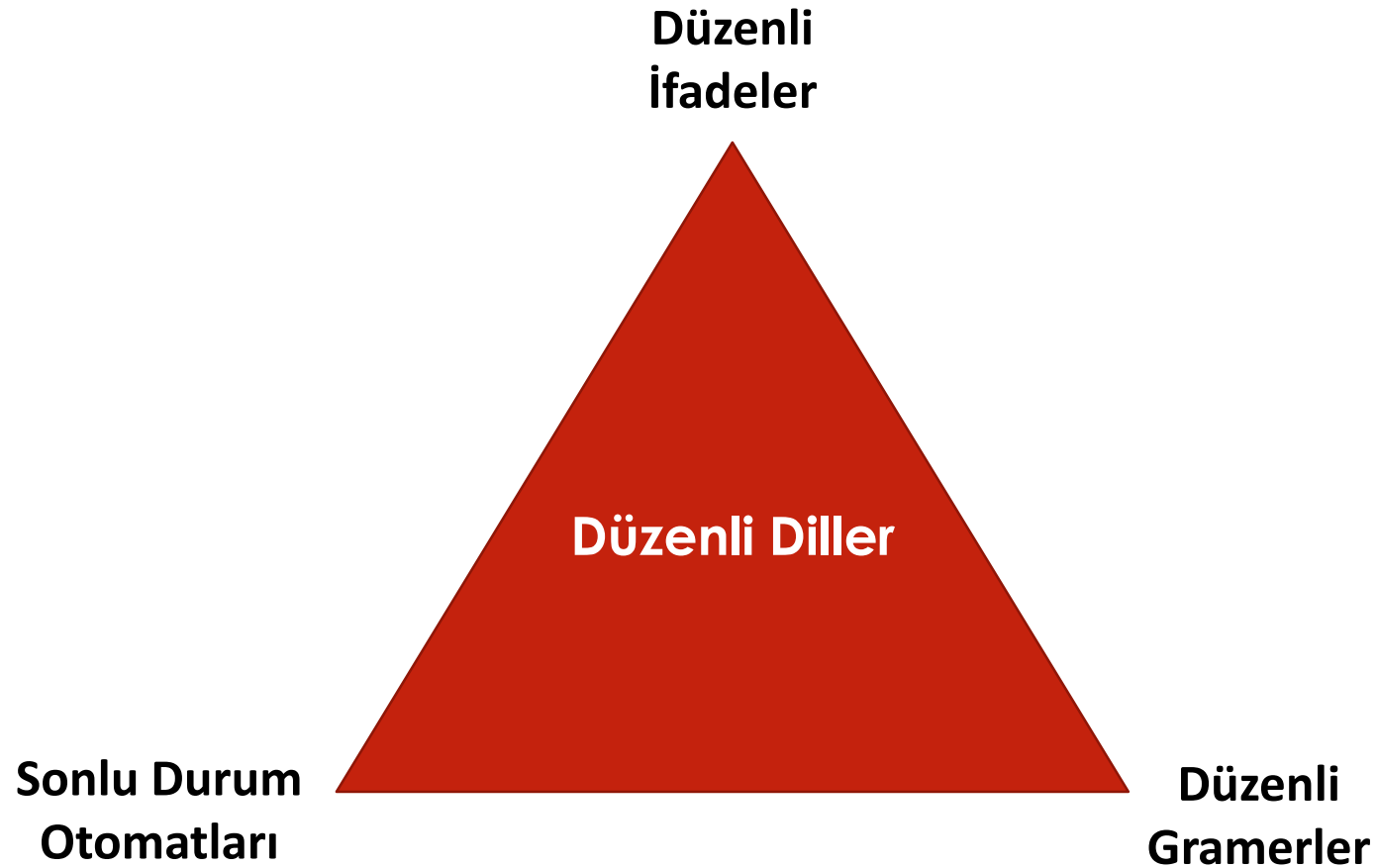
Bağlama - Duyarlı Diller : Doğrusal – Sınırlandırılmış otomata
(Context – Sensitive Languages : Linear – Bounded Automata)

Bağlam - Bağımsız Diller : Bas – Bırak otomata
(Context – Free Languages : Push – Down Automata)

Düzenli Diller : Sonlu - Durum otomata
(Regular Language : Finite – State Automata)

*(Dil : otomata)

CHOMSKY HİYERARŞİSİ - TİP 3





DÜZENLİ İFADELER

(REGULAR EXPRESSIONS)

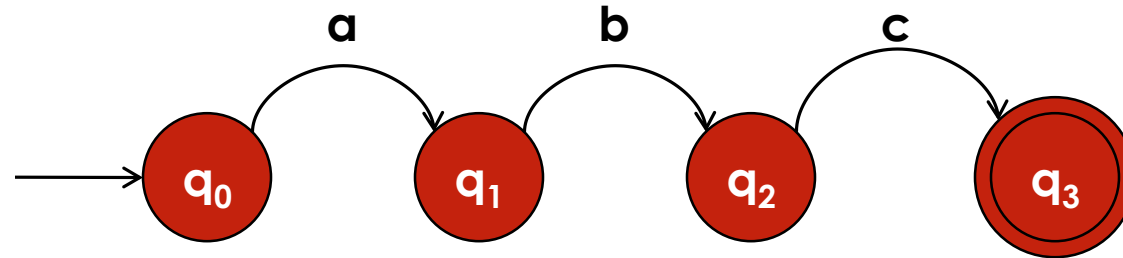
DÜZENLİ İFADELER

- **Biçimsel Tanım:** Karakter katarı kümelerinin cebirsel olarak tanımlanmasını sağlayan bir gösterim yöntemidir (*Stephen C. Kleene, 1956*).
- Karakter dizileri içinde, belirlenen örüntüleri (pattern) taramaya yarar.
- Kullanım alanları:
 - Arama motorları
 - Bilgi edinimi (Information retrieval)
 - Kelime işleme (Word processing)
 - Veri doğrulama (Data validation)
 - ...
- ✓ Herhangi bir düzenli ifade doğrudan NFSA'ya, buradan da DFSA'ya dönüştürülebilir.

DÜZENLİ İFADELER

- En basit düzenli ifadeler karakterlerin sıralı biçimde dizilmesiyle oluşur.
- Düzenli ifadeler büyük – küçük harf duyarlıdır.

- /abc/

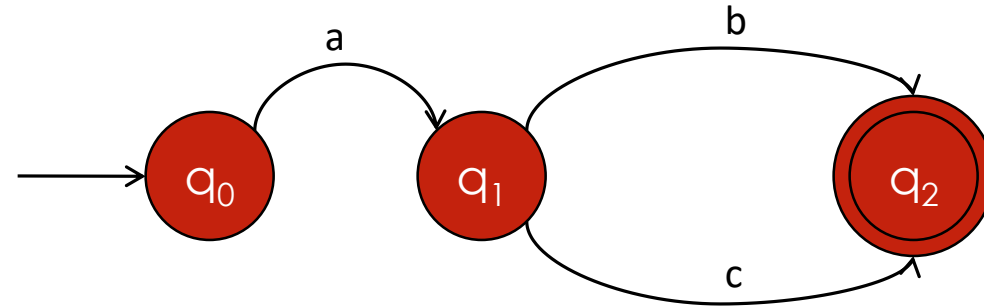


- /okula/ → 'Ali okula gelmedi.'
- /okul/ → 'Ali okula gelmedi.'
- /a/ → 'Ali okua gelmedi.'
- /okula gelmedi/ → 'Ali okula gelmedi.'

KARAKTER SINIFLARI

- Kare parantezler arasına yazılan karakterler, 'veya' (disjunction) işlemine tabi olurlar.

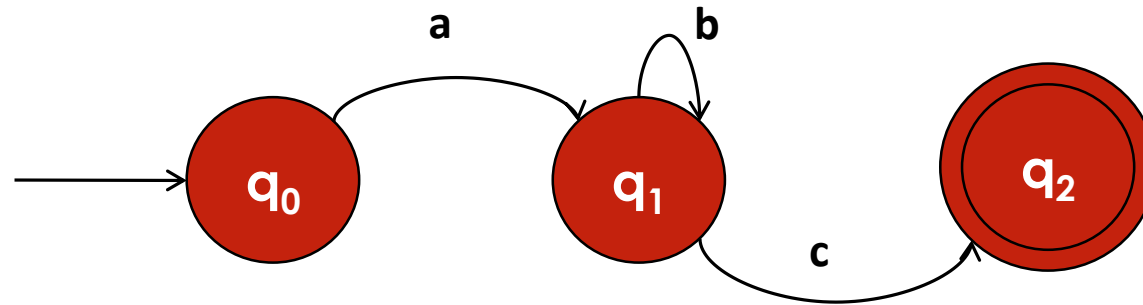
- $/a[bc]/$



- '-' işareti ile kare parantez içinde bir aralık belirtilebilir.
 - $/[Bb]ah\text{e}/ \rightarrow$ Bahe veya bahe
 - $/[abc]/ \rightarrow$ a veya b veya c
 - $/[1234567890]/ \rightarrow$ Herhangi bir rakam
 - $/[A-Z]/ \rightarrow$ Herhangi bir byk harf (İngiliz alfabesindeki 26 harften biri)
 - $/[a-z]/ \rightarrow$ Herhangi bir kk harf (İngiliz alfabesindeki 26 harften biri)
 - $/[0-9]/ \rightarrow$ Herhangi bir rakam

* (KLEENE STAR)

- '*' karakteri kendinden önce gelen karakterin veya düzenli ifadenin 0 veya daha fazla kez ardışık olarak tekrarlanabileceğini belirtir.
- $/ab^*c/ \rightarrow ac, abc, abbc, abbbc, \dots$

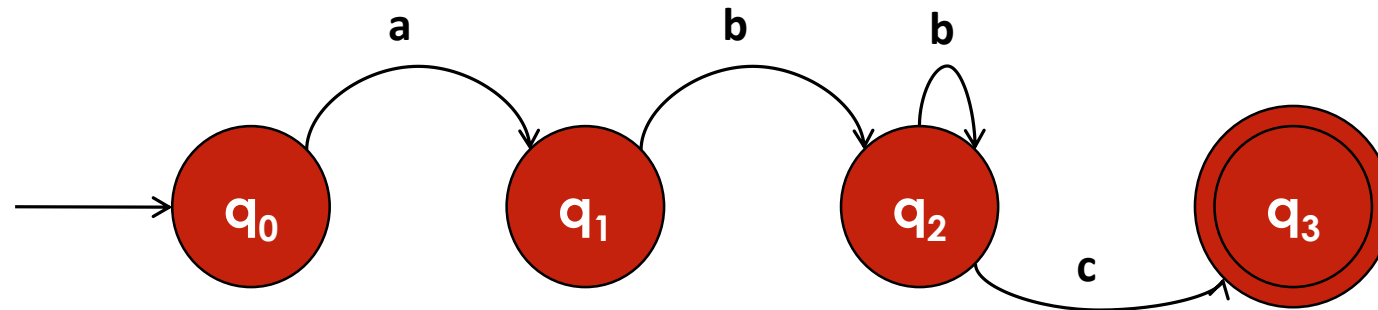


- $/[0-9][0-9]^*/ \rightarrow$ bir veya daha fazla sayıda ardışık rakam

+ (KLEENE PLUS)

- '+' karakteri kendinden önce gelen karakterin veya düzenli ifadenin 1 veya daha fazla kez ardışık olarak tekrarlanabileceğini belirtir.

- /ab+c/

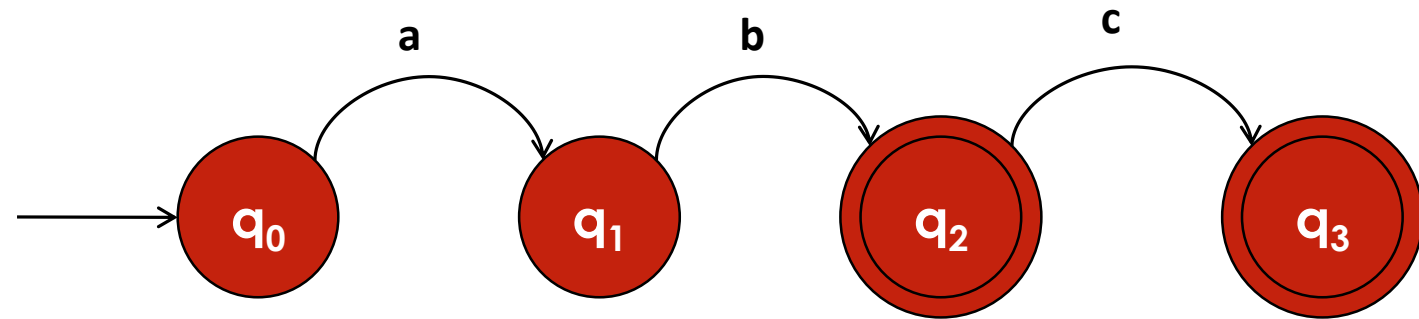


- /[0-9]+/ → bir veya daha fazla sayıda ardışık rakam

| (PIPE)

- Bu karakter örüntüler arasında 'veya' işlevi görür.

- /abc | ab/

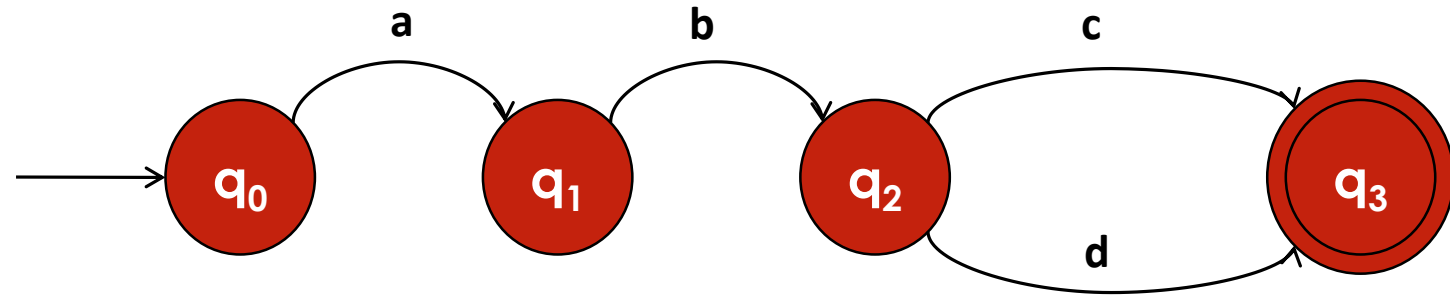


- /elma | portakal/ → elma veya portakal

| (PIPE)

- ‘()’ şeklindeki normal parantezler içine alınan ifadeler tek bir karaktermiş gibi işlenir.

- /a(bc | bd)/

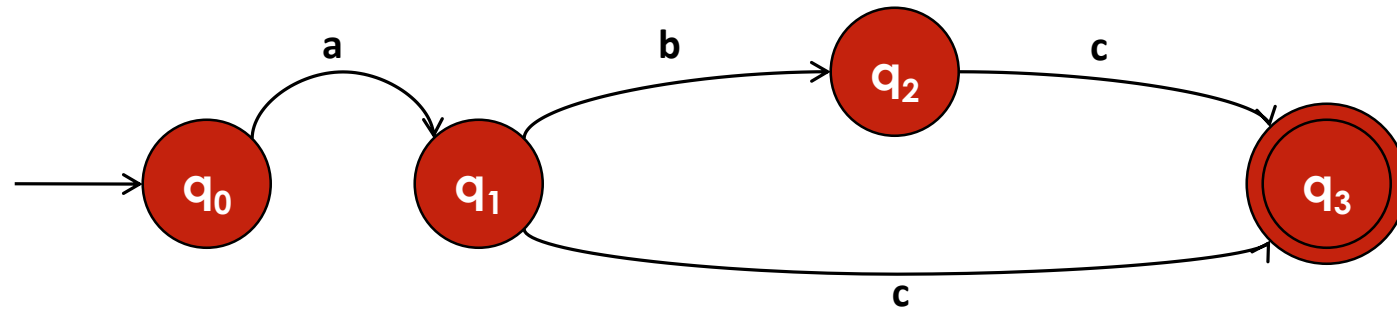


- /yüz(er | erler)/ \rightarrow yüzer veya yüzerler

? VE .

- ‘?’ karakteri kendinden önce gelen karakterin veya düzenli ifadenin seçimlik olduğunu belirtir.

- /ab?c/



- /evleri?/ → evler veya evleri
 - /colou?r/ → color veya colour
- ‘.’ karakteri tekil herhangi bir karakterin yerine geçebilir.
 - /dö.me/ → dökme veya dövme gibi

İFADE SINIRLARI

- **\b** özel karakteri, kullanıldığı yere göre, aranan ifadenin önünde veya arkasında sınırlayıcı (boşluk gibi) karakterleri sınır olarak kabul eder.
 - /\b eli\b/ → önünde ve arkasında boşluk olan 'eli' ifadesini bulur.
- **\B** karakteri sınırlandırma olmayan durumu belirtir.

^ (CARET)

- ^ karakteri 3 farklı şekilde kullanılabilir.
 - 1) Aralık için olumsuzlama
 - /^[^A-Z]/ → büyük harf harici karakter
 - /^[^Ss]/ → S veya s harici karakter
 - /^[^\.]/ → nokta harici karakter
 - 2) Satır başına bağlama
 - /^Avrupa/ → satır başında 'Avrupa' olan durum
 - 3) Normal karakter olarak kullanım
 - /[e^]/ → e veya ^
 - /a^b/ → a^b örüntüsü



\$

- \$ karakteri ise normal bir karakter olarak kullanılabileceği gibi satır sonuna bağlama için de kullanılabilir.
 - /gittiler\.\$/ → satır sonunda 'gittiler' olan durum

SAYAÇLAR

- Herhangi bir düzenli ifadenin ne kadar tekrarlanacağını belirtirler.
 - $\{n\} \rightarrow$ kendinden önceki düzenli ifade n defa ardışık olmalıdır.
 - $\{n, m\} \rightarrow$ kendinden önceki düzenli ifade en az n , en çok m defa ardışık olmalıdır.
 - $\{n, \}$ \rightarrow kendinden önceki düzenli ifade en az n defa ardışık olmalıdır.
- Örneğin **Trakya** kelimesi önünde 3 karakter ve devamında 4 rakam gelecek şekilde bulunmak istendiğinde, şu şekilde düzenli ifade ile gösterilebilir:

$[a-z]\{3\}[0-9]\{4\}Trakya$

OPERATÖR ÖNCELİK HİYERARŞİSİ

- En yüksekten en düşük öncelikli operatöre doğru sıralama şu şekildedir:
 1. Parantez $\rightarrow ()$
 2. Sayaçlar $\rightarrow * + ? \{ \}$
 3. Seriler veya bağlayıcılar $\rightarrow \text{evler} \wedge \text{Yarın gelecek} \$$
 4. Veya $\rightarrow |$ (pipe)

ÖZEL OPERATÖRLER

- \d → herhangi bir rakam ([0-9])
- \D → rakam olmayan bir karakter ([^0-9])
- \w → alfanümerik karakterler ve _ karakteri ([a-zA-Z0-9_])
- \W → alfanümerik olmayan karakterler ([^\w])
- \s → boşluk karakterleri ([\r\t\n\f])
- \S → boşluk olmayan karakter ([^\s])
- \n → newline
- \r → carriage return
- \t → tab
- \f → form feed
- \. → nokta

ÖRNEKLER

- Tarih → 02/04/2010
 - $(\backslash d\{1,2\} \backslash / \backslash d\{1,2\} \backslash / \backslash d\{4\})$
- E-posta adresi
 - $(\backslash w + @ [a-zA-Z_] + ? \backslash . [a-zA-Z] \{2,6\})$
- IP adresi
 - $\backslash b ([0-9] \{1,3\}) \backslash . ([0-9] \{1,3\}) \backslash . ([0-9] \{1,3\}) \backslash . ([0-9] \{1,3\}) \backslash b$

ÖRNEK

- $\Sigma = \{a, b\}$ alfabetiyle oluşturulan, **b** ile başlayıp **a** ile biten tüm karakter katarlarını tanıyan dilin **düzenli ifadesini (regular expression)** yazınız.

ÖRNEK

- $ab(a \mid b)^*$ düzenli ifadesini tanıyacak DFA'yı çiziniz.

ÖRNEK

- $(a \mid b)^*ab(a \mid b)^*$ düzenli ifadesini tanıyacak DFA'yı çiziniz.

KAYNAKLAR

- Hopcroft, J.E. and Ullman J.D. (1979). Introduction to Automata Theory, Languages, and Computation (1st ed.). Addison-Wesley.
- Yarımağan, Ünal, “Özdevinirler Kuramı ve Biçimsel Diller”, Bıçaklar Kitabevi, 2003, ISBN# 975-8695-05-3
- Prof. Dr. Yılmaz KILIÇASLAN, Formel Diller ve Soyut Makinalar Ders Sunumları
- Namık Kemal Üniversitesi, Yrd. Doç. Dr. E. Serdar Güner Ders Sunumları
- Sakarya Üniversitesi, Biçimsel Diller ve Soyut Makineler Ders Sunumları
- Gazi Üniversitesi, Biçimsel Diller ve Otomatlar Ders Sunumları