BİÇİMSEL DİLLER VE OTOMATA TEORİSİ

Biçimsel Diller ve Otomata Teorisi

1. Hafta

DR. ÖĞR. ÜYESİ. HÜSEYİN VURAL

Ders İzlencesi

- Kümeler
- Fonksiyonlar
- Biçimsel Diller

Otomata Teorisi (Automata Theory)

- **Otomata:** Hesaplama süreçlerini modelleyen matematiksel sistemlerdir.

- **Biçimsel Diller:** Otomatlar tarafından tanınan veya üretilen kelime dizileridir.

Kümeler

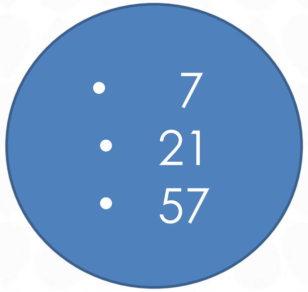
- Bir grup nesnenin bir araya gelmesi ile **küme** oluşur.
- Nesneler; sayı, sembol veya başka küme(ler) olabilir.
- Kümenin içerisindeki nesnelere kümenin elemanları veya üyeleri denilir.

Kümeler

- Kümeler birden fazla şekilde gösterilebilir.

- Örneğin
  - {7,21,57}

- Venn diagramı

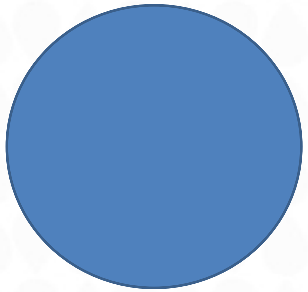


Kümeler

- Küme elemanlarının sıralaması önemli değildir.
- Örneğin
  - {7,21,57} ile {21,7,57} veya {57,21,7} aynıdır
- $\in$ ve $\notin$ sembolü kümenin elemanı olup olmadığını gösterir.
- $7 \in \{7,21,57\}$ ve $8 \notin \{7,21,57\}$

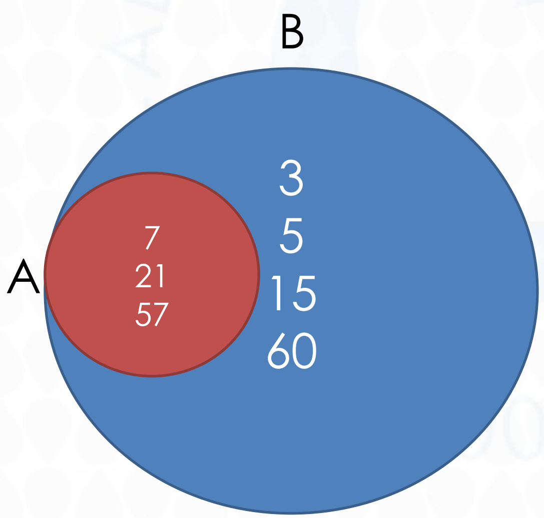
Kümeler

- Boş küme aşağıdaki şekillerde gösterilebilir
- { }
- ∅



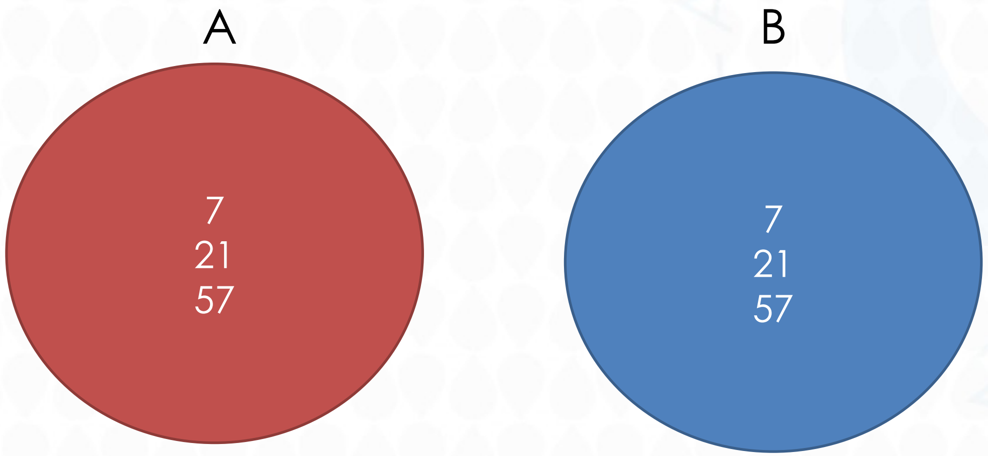
Kümeler

- Eğer A kümesinin tüm elemanları B kümesinde bulunuyorsa o zaman A kümesi B kümesinin **alt kümesidir**.



Kümeler

- Eğer A kümesinin tüm elemanları B kümesinde de bulunuyorsa, A kümesi B kümesi ile eşittir. A=B



Kümeler

- **Birleşim**: İki kümenin tüm elemanlarını içeren kümedir. $A \cup B = \{x | x \in A \text{ veya } x \in B\}$

- **Kesişim**: İki kümenin ortak elemanlarını içerir.
- $A \cap B = \{x | x \in A \text{ ve } x \in B\}$

Kümeler

- **Küme Farkı:** İki kümeden birinin diğerinden farklı elemanlarını içermesidir
- $A - B = \{x \mid x \in A \text{ ve } \notin B\}$

Kümeler

- **Kartezyen Çarpım:** İki kümenin her elemanının diğer kümedeki her elemanla eşleştirilmesiyle elde edilen kümedir.
- $A \times B = \{(a,b) \mid a \in A, b \in B\}$
- Örnek: $A=\{3,5\}$ ve $B=\{a,b\}$
- $A \times B = \{(3,a),(3,b),(5,a),(5,b)\}$

Otomata Teorisi (Automata Theory)

- Otomata teorisi, özdevinirler veya özdevinim kuramı olarak da bilinir

- Otomata teorisi, soyut matematiksel sistemleri ve bu sistemleri kullanarak hesaplama problemlerinin çözümlerini araştırır

- Bahsi geçen soyut sistemlere **otomat** denilmektedir

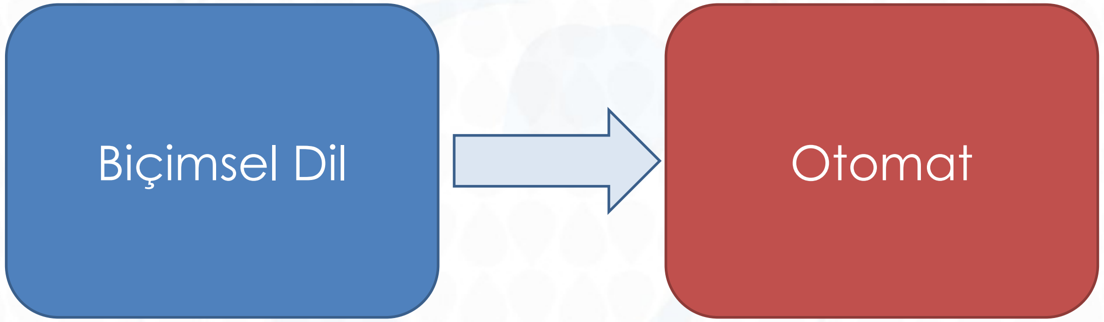
Otomata Teorisi (Automata Theory)

- Günümüzde bir işi kendi kendine yapabilen çay, kahve vb. otomatları görmekteyiz
- Otomatlar, durum veya konfigürasyonda ilerleyerek bir girdi üzerinde hesaplamalar yapabilen soyut makinelerdir
- Hesaplamaları yapılan her durumda bir geçiş fonksiyonu bulunur

Otomata Teorisi (Automata Theory)

- Bilinen en güçlü otomat **Turing makinesidir** (**Turing Machine**).
- Otomata Teorisi birçok sınıflandırılmaya ayrılmaktadır.
- Bu sınıflardan bahsetmeden önce Otomata teorisinin yakından ilgilendiği Biçimsel Dil Kuramı hakkında bilgi verelim.

Otomata Teorisi (Automata Theory)



Otomat Örneği

Otomata Teorisi (Automata Theory)

- Biçimsel diller bilgisayar bilimlerinde, mantıkta ve dil bilim çalışmalarında kullanılan bir dil ailesidir.
- Dilde bulunan bütün öğeler ve dilin ulaşabileceği sınırlar belirli kurallar dahilinde tanımlanabiliyorsa bu dillere **Biçimsel Dil** ismini verebiliriz.
- Bu anlamda bilgisayar bilimlerinde bulunan bütün programlama dillerini bu ailede düşünmek mümkündür.

Biçimsel Diller

- Biçimsel dili tanımlamak için dilin en küçük yapısı olan alfabe, kelime kavramlarını öğrenelim.

- Bir dilde kullanılan sembollere harf denilir ve harflerin listesine de alfabe (alphabet) denilir.

- ∑ dilin en küçük yapı taşı olan alfabeyi temsil etmektedir
  - Örneğin ∑ **= {x}**
  - ∑ alfabesi "x" harfini içermektedir. Bu harf dışındaki hiçbir sembol bu alfabede tanımlı değildir.

Biçimsel Diller

- $L_1 = \{\lambda, x, xx, xxx, xxxx, \ldots\}$, $L_1 = \{x^n\}$ $n = 1, 2, 3 \ldots$
- L_1 dilinde yer alan terimler ise kelimelerdir.
- Harf içermeme durumu ise null string ‘λ ‘ ile gösterilir.

Biçimsel Diller

- Örnek, $\sum = \{a,b\}$
- **Length** ile bir kelimenin uzunluğunu ölçeriz
- length(ab) =2 , length(λ)=0
- **Reverse**, bir kelimenin tersini yazmamızı sağlar

Biçimsel Diller

- Kleene Star (Kleene yıldızı) '*' ile λ dahil, alfabedeki tüm karakterlerin mümkün olan tüm birleşimlerini ifade eder
- $\sum = \{a,b\}$
- $\sum^* = \{\lambda, a, b, ab, ba, aa, bb, aaa, aaabbb, \ldots.\}$
- +(Plus): En az bir kere o karakterden gelmesi anlamına gelir
- $\sum^+ = \{a, b, ab, ba, aa, bb, aaa, \ldots\}$

Düzenli İfadeler (Regular Expressions)

- Bir dilde üretilebilecek olan ifadelerin gösterim biçimidir

- Alfabemizin a ve b harflerinden oluştuğunu düşünelim ve bu harflerden oluşan çeşitli diller tanımlayalım

- **S=(a+b)**; bu dilin kabul ettiği kelimeler yalnızca <u>a veya b'dir</u>. => **{a,b}**

Düzenli İfadeler (Regular Expressions)

- **S= (a+b)(a+b) = (a+b)²** ; bu dilin kabul ettiği kelimeler ; **{aa,ab,ba,bb}** yani a ve b harfleri kullanılarak yazılabilecek **2** harfli stringlerdir.

- **S=(a+b)(a+b)(a+b) = (a+b)³** ; dilinin kabul ettiği kelimeler; **{aaa,aba,bba,bbb}** a ve b harfleriyle yazılabilecek **3** harfli tüm stringlerdir.

- **S=(a+b)*** = a ve b harfleriyle oluşturulabilecek tüm kombinasyonları

{λ,**a,b,ba,bb,aa,aab,aabb,aaaa**} (**sonsuz sayıda**) kabul etmektedir

Düzenli İfadeler (Regular Expressions)

- X* = {λ,x,xx,xxx,xxxx…}

- ab* = {a,ab,abb,abbb,….}
  - a ile başlayacak ve sonsuz tane b gelebilir

- (ab)* = {λ,ab,abab,ababab,abababab,…..}

- a*b* = {λ,a,b,aa,bb,ab,aab,abb,aabb,….}

Düzenli İfadeler (Regular Expressions)

- 1. a ile biten kelimeleri içerecek sonsuz sayıda a veya b içerebilecek regular expression yazın ?
- 2. b ile başlayacak sonsuz sayıda a veya b içerebilecek regular expression yazın ?
- 3. En az 1 tane a ve en az 1 tane b içeren regular expression yazın ?
- 4. Yan yana aynı harf gelmeyecek kelimeleri yazın ?

Düzenli İfadeler (Regular Expressions)

- 1. (a+b)*a
- 2. b(a+b)*
- 3. a(a+b)*b(a+b)*
  - ab*ba*
- 4. a(ba)*b(ab)*

DERS SONU

Düzenli Diller

Dr. Öğr. Üyesi Hüseyin VURAL