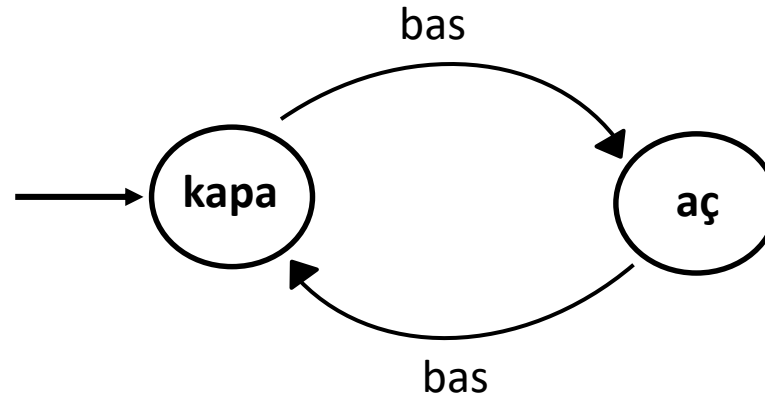


# Otomata Teorisi (BIL 2114)

## Hafta 1: Amaç ve Genel Kavramlar



# Hafta I

## Plan

1. İletişim ve Ders Bilgisi
2. Otomata Teorisi Genel Bakış
3. Hedeflenen Kazanımlar
4. Matematiksel Nosyonlar ve Terminoloji
5. Otomata İçin Genel Kavramlar



# I. İletişim ve ders bilgisi

## Dersi veren öğretim üyesi:

Arş. Gör. Dr. Fırat İsmailođlu

email: [fismailoglu@cumhuriyet.edu.tr](mailto:fismailoglu@cumhuriyet.edu.tr)

Tel: 0346 210 1010 - 2462

Bilgisayar Müh. Bölümü, Oda No:212

## Ders içeriđi:

Her hafta: 2 saat teorik ders + 1 saat uygulama (örnek soru çözümleri)

## Puanlama:

%40 vize + %60 final (ödev verilmeyecek) (yaz okulu olmayacak)



## II. Otomata teorisi genel bakış:

Otomata ismi, Yunanca “*kendi başına hareket*” kelimesinden gelmektedir.

Otomata teorisi, hesaplamanın ve hesaplanabilirliğin prensiplerini anlamaya çalışır. Hatta bir diğer adı ‘Hesaplama Teorisi (Theory of Computation)’ dir.

Ne tür şeyleri mekanik olarak hesaplayabiliriz, hesaplayabildiğimiz şeyleri ne kadar hızlı hesaplarız, bunun için ne kadar hafızaya ihtiyacımız vardır sorularını sorar.

Otomata bu sorular için basit, soyut modeller ortaya koyar.

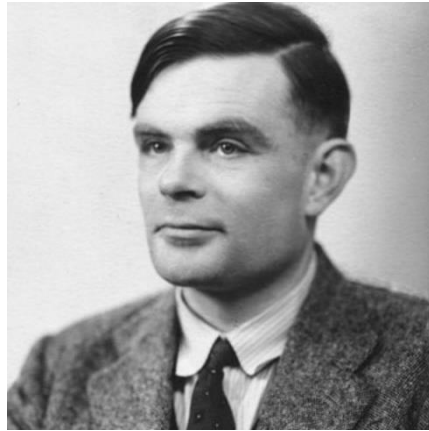
Bu modellere “*soyut makineler*” denir.



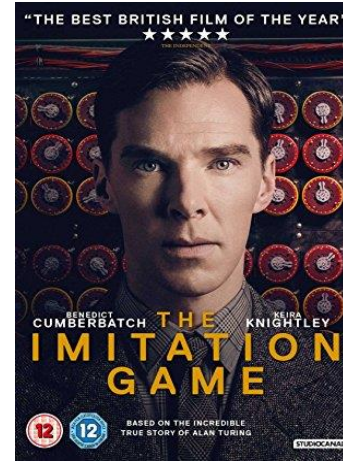
## II. Otomata teorisi genel bakış:



Bu alanda ilk çalışmalar **Alan Turing** tarafından 1930'lu yıllarda yapılmıştır ve Alan Turing'in geliştirdiği "Turing Makinesi" günümüz bilgisayarlarının atası kabul edilir.



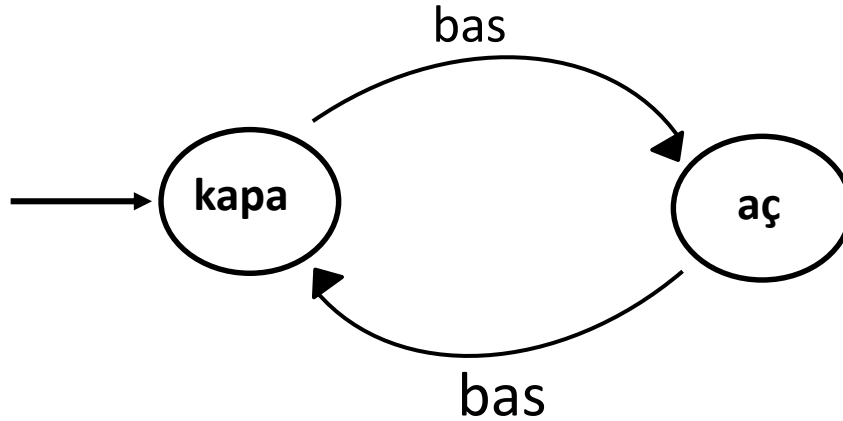
Alan Turing



### III. Hedeflenen Kazanımlar :

+ Problemleri soyutlayabilme, basitçe ifade edebilme

ör.



Ac/kapa düğmesi

+ Sınırlı kapasite ile çalışabilme deneyimi elde etme.

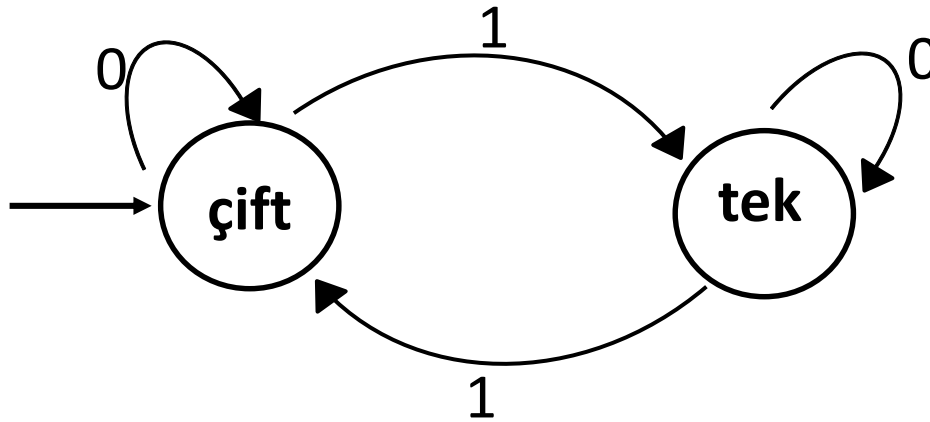
+ İlk bilgisayarların nasıl çalıştığı hakkında bilgi sahibi olma.

+ Bilgisayar mimarisi dersine temel oluşturma

### III. Hedeflenen Kazanımlar :

Sınırlı kapasite ile çalışabilme örneği:

Verilen 001001000....011 gibi büyük bir 0-1 dizisindeki toplam 1 sayısı tek mi diye kontrol etmek istiyoruz. Fakat makinemiz bu sayının tamamını belleğinde tutamıyor. Bu durumda aşağıdaki gibi basit bir makine bu görevi yerine getirebilir.



# IV. Matematiksel Nosyonlar ve Terminoliji

**Küme** (set): (iyi tanımlanmış) objeler topluluğu

ör. Doğal sayılar kümesi:  $\mathbb{N} = \{1,2,3, \dots\}$

Tam sayılar kümesi:  $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$

**Çoklu küme** (multiset - bag): aynı elemanın birden fazla kez bulunmasına izin veren küme  
ör.  $A = \{-1, 3, -1, 4\}$

**Boş küme** (empty set): Hiçbir elemanı olmayan küme,  $\emptyset$  ile gösterilir.

**Altküme** (subset): Bir kümenin elemanlarından oluşturulan yeni küme.

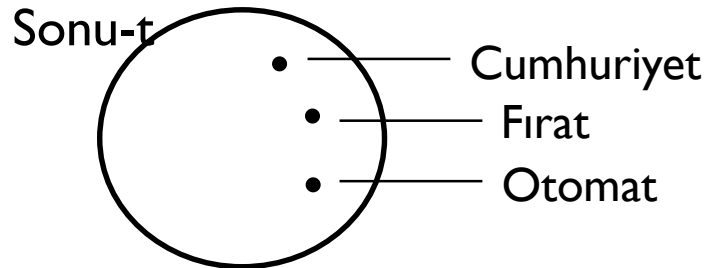
$$A \subseteq B \Rightarrow \forall x \in A, x \in B.$$

**Kümelerin gösterimi:**

i) Liste ile: ör.  $A = \{-1, 3, 4\}$

ii) Ortak özellikler (bir kural ile) ör.  $A = \{x \mid x = 2k, k \in \mathbb{N}\}$

iii) Venn semasi ile:





# IV. Matematiksel Nosyonlar ve Terminoliji

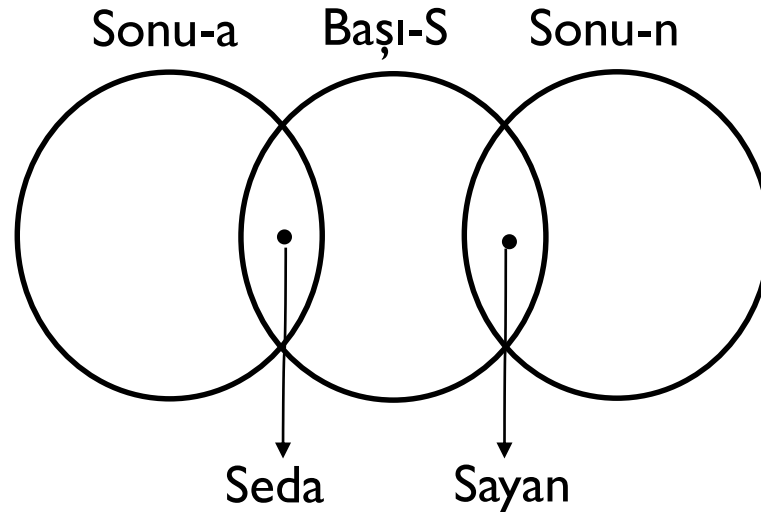
## Küme Operasyonlari:

i) **Birleşme** (union): Birleştirelen kümelerin elamanlarını tek bir kümede toplar.

$$x \in A \cup B \Rightarrow x \in A \vee x \in B$$

ii) **Kesişme** (intersection): Kesiştirilen kümerledeki ortak elemanları yeni bir kümede toplar.  $x \in A \cap B \Rightarrow x \in A \wedge x \in B$

iii) **Tamamlama** (complement): Tamamlanan kümenin elemanları dışında kalan elemanlardan yeni bir küme oluşturur.  $\bar{A} = \{x \mid x \notin A\}$



## IV. Matematiksel Nosyonlar ve Terminoliji

**Güç kümesi:** Bir kümenin bütün alt kümelerinin bir araya gelmesiyle oluşmuş yeni küme. **ör.**  $A = \{0,1\}$  iken  $A'$  nin güç kümesi  $\mathcal{P}(A) = \{\emptyset, \{0\}, \{1\}, \{0,1\}\}$ .

(fark et! Güç kümesinin elemanlarının kendisi de bir küme)

Not:  $k$  elemanlı bir kümenin güç kümesinin eleman sayısı  $2^k$  dir.

**Dizi:** Objelerin belirli bir sırada dizildiği liste. **ör.**  $(4, -12, 1)$   
En meşhur dizi sıralı ikili (pair) dir, iki elemanlı diziye denir. **ör.**  $(4, 1)$   
(fark et! kümelerde elemanların sırası önemli değil, dizi elemanların sırası önemli)

**Kartezyen Çarpım:**  $A$  ve  $B$  gibi iki kümenin kartezyen çarpımı birinci elemanı  $A'$  dan, ikinci elemanı  $B'$  den olan tüm sıralı ikililerin kümesidir.

$$A \times B = \{(a, b) \mid a \in A, b \in B\}$$



## IV. Matematiksel Nosyonlar ve Terminoliji

**ör.**  $A = \{\text{kola, ayran}\}$ ,  $B = \{\text{kebab, döner}\}$  iken

$A \times B = \{(\text{kola, kebab}), (\text{kola, döner}), (\text{ayran, kebab}), (\text{ayran, döner})\}$

**Kartezyan Çarpımın Genelleştirilmesi:**  $A_1, A_2, \dots, A_k$  tane kümenin kartezyen çarpımı

$$A_1 \times A_2 \times \dots \times A_k = \{(a_1, a_2, \dots, a_k) \mid a_i \in A_i, i = 1, \dots, k\}$$

**k-lı sıra** (k-tuple) : Tüm  $A_i$  kümeleri birbirine eşit iken  $A \times A \times \dots \times A = A^k$  kartezyen çarpımının elemanlarına k-lı sıra denir,  $(a_1, a_2, \dots, a_k)$  ile gösterilir.



# IV. Matematiksel Nosyonlar ve Terminoliji

## Fonksiyonlar ve İlişkiler

**Fonksiyon:** Tanım kümesindeki **her** bir elemanı, değer kümesindeki **yalnız bir** elemana eşleyen kurala fonksiyon denir.

$$f : \text{Tanım} \rightarrow \text{Değer}$$

ör. toplama:  $\mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$ ,  $\text{toplama}(3,4) = 7$ .

**Doğrulama fonksiyonu** (predicate/property): Değer kümesi  $\{\text{doğru}, \text{yanlış}\}$  olan fonk.

ör.  $\text{çift}(4) = \text{doğru}$ ,  $\text{çift}(5) = \text{yanlış}$

$\text{yener}(\text{taş}, \text{makas}) = \text{doğru}$

## IV. Matematiksel Nosyonlar ve Terminoliji

**İlişki fonksiyonu** (relation): Tanım kümesi bir k-lı sıra olan doğrulama fonksiyonudur ve  $R$  ile gösterilir.

$$R : A \times \cdots \times A \rightarrow \{\text{doğru}, \text{yanlış}\}$$

**ör.**  $A = \{\text{taş}, \text{kağıt}, \text{makas}\}$  iken *yener* ilişkisi:

<i>yener</i>	<i>taş</i>	<i>kağıt</i>	<i>makas</i>
<i>taş</i>	<i>yanlış</i>	<i>yanlış</i>	<i>doğru</i>
<i>kağıt</i>	<i>doğru</i>	<i>yanlış</i>	<i>yanlış</i>
<i>makas</i>	<i>yanlış</i>	<i>doğru</i>	<i>yanlış</i>

$yener(\text{taş}, \text{makas}) = \text{doğru}$






## V. Otomata için Genel Kavramlar

### Diller ve Kelimeler

**Alfabe** ( $\Sigma$ ): Sonlu, boş olamayan kümeye alfabe denir.

**Harf**: Alfabenin elemanlarına harf denir.

ör.  $\Sigma_1 = \{0,1\}$ ,  $\Sigma_2 = \{a, b, c, d\}$ ,  $\Sigma_3 = \{$      $\}$

**Kelime** (string) : Sonlu harf dizisi ör.  $k_1 = 01001$ ,  $k_2 = dddbcb$ ,  
 $k_3 =$    

Kelime uzunluğu  $| \cdot |$  ile verilir. ör.  $|k_1| = 5$

**Boş kelime** (empty string) : 0 uzunluğundaki kelime.  $\varepsilon$  ile gösterilir.  
Yani hiç harfi yoktur.

Kelimelerin bitleştirilmesi (concatenation): kelimelerin uç uca eklenmesi ile oluşur.

ör.  $k_1 = ba$  ve  $k_2 = na$  iken  $k_1 k_2 = bana$ ,  $k_1 k_1 = k_1^2 = baba$ ,  
 $k_1 k_2 k_2 = banana$ .

## V. Otomata için Genel Kavramlar

### Alfabelerin güçleri

$\Sigma$  bir alfabe iken  $\Sigma^k$ ,  $\Sigma$  alfabesinin harflerinden oluşan  $k$  uzunluğundaki kelimelerin kümesidir.

ör.  $\Sigma = \{0,1\}$  iken  $\Sigma^1 = \{0,1\}$ ,  $\Sigma^2 = \{00,01,10,11\}$ , ...

**Not:** Tüm  $\Sigma$  alfabeleri için  $\Sigma^0 = \{\varepsilon\}$ . Yani yalnızca boş kelimeyi içerir.

$\Sigma^*$ :  $\Sigma$  alfabesinin harfleri kullanılarak oluşturulabilecek bütün kelimelerin kümesidir:

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

ör.  $\Sigma = \{0,1\}$  için  $\Sigma^* = \{\varepsilon, 0,1,00,11,01,10,000,111,010, \dots\}$



## V. Otomata için Genel Kavramlar

**Dil:** Bir  $\Sigma$  alfabeti üzerine bir dil  $\Sigma^*$  'ın bir alt kümesidir.  $L$  ile gösterilir.

**ör.**  $L = \{0,00,001,0000,00001, \dots\} \rightarrow \Sigma = \{0,1\}$  alfabeti üzerine 0 ile başlayan kelimelerin dili.


**ör.**  $L = \{\varepsilon, 01,10,0011,0101,1001, \dots\} \rightarrow \Sigma = \{0,1\}$  alfabeti üzerine içerdiği toplam 0 sayısı içerdiği toplam 1 sayısına eşit olan kelimelerin dili.

**ör.** Türkçe dili  $\Sigma = \{a, b, c, \dots, y, z\}$  üzerine kelimelerin kümesi.

**Dillerin bitleştirilmesi** (concatenation of languages) : Kelimelerin bitleştirilmesi kavramı diller için de genişletilebilir.  $L_1$  ve  $L_2$  iki dil iken bunların bitleştirilmesi

$$L_1L_2 = \{xy \mid x \in L_1, y \in L_2\}$$

**ör.**  $L_1 = \{0, 01\}, L_2 = \{\varepsilon, b, bb\}$  iken  $L_1L_2 = \{0,0b, 0bb, 01,01b, 01bb\}$


$$0\varepsilon = 0$$

