Ayrık Matematik
(Ayrık İşlemsel Yapılar)

Fırat İsmailoğlu, PhD

Hafta 2:
Genel Matematik Terimleri
ve
Lojik I

Hafta 2
Plan

1. Genel Matematik Terimleri: Toplama, Çıkarma sembolleri, Kumeler, Fonksiyonlar
2. Lojik
3. Ve baglaci
4. Veya baglaci
5. XOR baglaci
6. Totoloji
7. Mantiksal işlem sirasi
8. Mantik devreleri

Toplama ve Çarpma Sembolleri

Toplama Sembolü:

$x_1, x_2, \ldots, x_n$; n tane sayıdan olusan bir dizi olsun. Bu sayı dızısının toplamı:

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + \cdots + x_n.$$

(bitiş: n, başlangıç: $i=1$)

```
int i , toplam=0;
for(i = 1; i ≤ n; i + +){
    toplam=toplam+ x_i; }
```

ör. $\sum_{i=1}^{4} i^2 = 1^2 + 2^2 + 3^2 + 4^2$

ör. $\sum_{i=1}^{4} 5 = 5\sum_{i=1}^{4} = 5(1 + 2 + 3 + 4) = 50$

İçiçe Toplam (Nested Sum)

A matriksi şu şekilde olsun: $A = \begin{bmatrix} 7 & 5 & 6 & 5 \\ 5 & 5 & 1 & 7 \end{bmatrix}$

Bu matriksin elemanlarını toplayalım:

$$\sum_{i=1}^{2}\sum_{j=1}^{4} A_{i,j}$$

```
int i , j, toplam=0;
for(i = 1; i ≤ 2; i + +){
    for(j = 1; i ≤ 4; j + +){
        toplam=toplam+ A_{i,j}; }}
```

Toplama ve Çarpma Sembolleri

Meyveler kümesi şöyle olsun: $M = \{$  ,  ,  $\}$

Bu durumda

$\sum_{x \in M} x$ =  +  +  ⟵ x değişkeni M kümesini tarıyor!

Her zaman bir dizinin sıralanmış elemanları toplam ile toplanmaz, bir kümenin elemanlarini da toplayabiliriz.

Çarpma Sembolü:

$x_1, x_2, \ldots, x_n$; n tane sayıdan olusan bir dizi olsun. Bu sayı dızısının çarpımı:

$$\prod_{i=1}^{n} x_i = x_1 \cdot x_2 \cdot \ \ldots \cdot x_n$$

ör. $\prod_{i=1}^{5} i = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 5!$

```
int i , carpim=1;
for(i = 1; i ≤ 5; i + +){
    carpim=carpim · x_i; }
```

Kümeler

Küme: Sıralanmamış objeler topluluğudur. Büyük harfle gösterilirler.

ör. Asal sayılar kümesi: $P = \{2,3,5, \dots\}$, $\mathbb{R}$ reel sayilar kumesi, $\mathbb{Z}$ tam sayilar kumesi.

- Küme eleman sayısı (cardinality) $|\quad|$ ile gösterilir. $|M| = 3$.

Kümelerin Gösterimi:

i) Venn semasi ile:

ii) Liste ile: ör. $A = \{-1,3,4\}$

iii) Ortak özellikler yönetimi (set abstraction)

E evrensel küme olsun, yani olabilecek tüm elemanlarin kumesi olsun.

$P: E \to \{0,1\}$ fonksiyonu bir özellik fonksiyonu (predicate) olsun. $x \in E$ için eger $P(x) = 1$ ise x elamani P özelligini tasiyordur; $P(x) = 0$ ise P özelligini tasimiyordur. Su halde ortak ozellik gosterimi:

$$\{x \in E \mid P(x)\}$$

P özelliğini tasiyan x elamanlarinin kumesi

ör. $\{x \in \mathbb{Z} | \, x \text{ asal ve tek sayı}\}$ $\{x \in \mathbb{N} | \, x \leq 100\}$ $\{x \in \mathbb{N} | \, \sqrt{x} \in \mathbb{Z}\}$

x'in tasidigi ozellik

ör. $\{x^2 | \, x \in \{1,2,3,4\}\}$

Bir Kümenin Toplamı, Çarpımı, Minimumu ve Maximumu:

S bir kume olsun. S'nin toplami, carpimi, minimumu ve maksimumu sırasıyla:

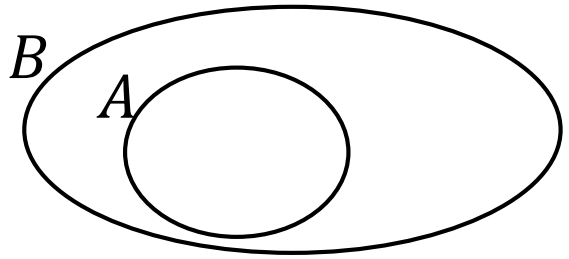
$$\sum_{x \in S} x \quad , \quad \prod_{x \in S} x \quad , \quad min_{x \in S} x \quad , \quad max_{x \in S} x$$

Alt Küme

Büme diğerinin bir alt kümesi ise o kümenin her bir elemani diğerinin de bir elemani olmak zorundandir.

Matematiksel olarak: $A \subseteq B \iff \forall\, x \in A$ için $x \in B$

Görsel olarak



Kümelerin Karsilastirilmasi

Sayilar gibi kumeleri de karsilastirabiliriz.

1. Esitlik: A ve B gibi iki kumenin birbirine esit olabilmesi $(A = \mathrm{B})$ icin:

i) $A \subseteq B$ $(\forall\, x \in A$ için $x \in B)$

ii) $B \subseteq A$ $(\forall\, x \in B$ için $x \in B)$

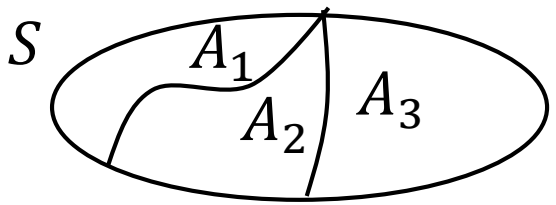
2. Büyüklük - Küçüklük: Reel sayilardaki buyukluk- kucukluk $(x \leq y)$, kümerlerde alt küme kavramı ile karşılık bulur. $A \subseteq B$ iken A kümesı B'den küçüktür gibi düşünebiliriz.

Bir Kümenin Parçalanması (Partition)

S, bir küme ve $A_1, A_2, \ldots, A_k$ boştan farkli kumler olsun. Eger bu kumeler asagidaki sartlari saglarlarsa $\{A_1, A_2, \ldots, A_k\}$ kümesi S'nin bir parcalanmasi olur.

i) $A_1 \cup A_2 \cup \cdots \cup A_k = S$

ii) herhangi $i, j \in \{1, \ldots, k\}$ için $A_i \cap A_j = \emptyset$ ise A_i ve A_j ayrık kümelerdir.



Programa Dillerinde Kümeler

Küme kavrami bazi programlama dillerinde (Python,Scheme, ..) 'list' (liste) kavrami ile karsilik bulur.

Örnegin $A = \{1,2,3\}$ kümesi Pyhton'da `A=[1,2,3]` listesiyle ifade edilir.

Diziler (Sequences)

Dizi: Sıralanmış objeler topluluğudur. Kümenin aksine parentezle gosterilirler:

$O = (\text{ilkokul}, \text{ortaokul}, \text{lise}).$

Bir dizinin uzunlugu 2 ise buna *sıralı ikili* denir. ör. $(4,1)$

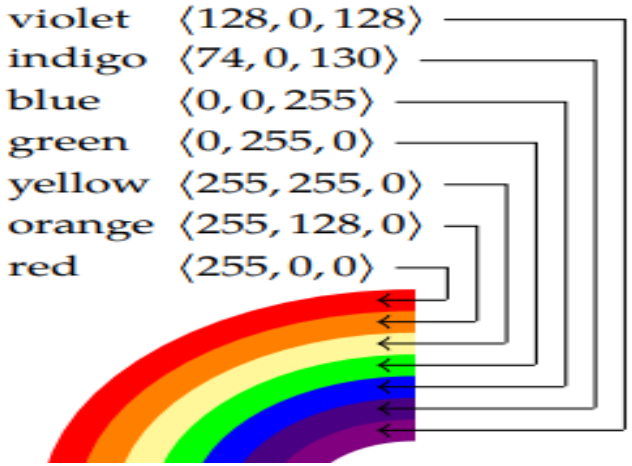
Kartezyen Çarpım

A ve B kümelerinin kartezyen carpimi $A \times B$ ile gosterilir ve şu kumeye esittir:

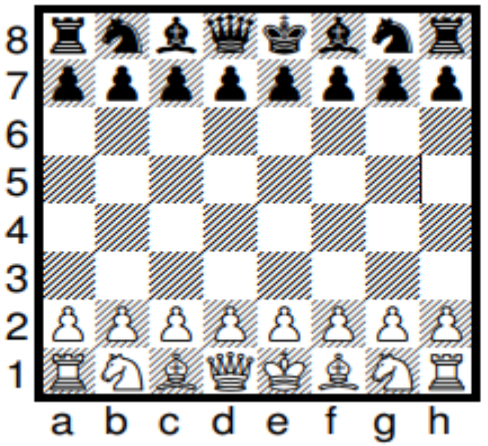
$$A \times B = \{(a, b) | a \in A, b \in B\}$$

bu kume ilk elemani A kumesine ait; ikinci elemani B kumesine ait tüm sıralı ikililerin kumesidir.

ör. RGB (red green blue) renk uzayı= $\{0,1,\ldots,255\} \times \{0,1,\ldots,255\} \times \{0,1,\ldots,255\}$ kartezyen carpimindan olusur. Burada bu ilk bilesen rengin ne kadar kirmizi oldugunu, ikinci bileşen ne kadar yeşil oldugunu, üçüncü bilesen ne kadar mavi oldugunu gosterir:



ör. Satranç tahtası: $\{a, b, c, d, e, f, g, h\} \times \{1,2,3,4,5,6,7,8\}$



satranç tahtasindaki her bir kare kartezyen çarpımının bir elemanina denk gelir.

Fonksiyonlar

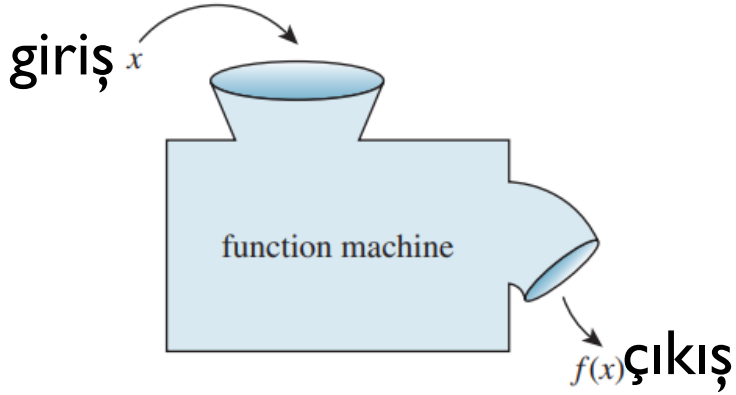
A ve B iki kume olsun. A'dan B'ye bir fonksiyon $\mathrm{f} : A \to B$ şeklinde yazilir. Bu fonksiyon iki önemli özellik taşır:

1) A'daki her elemani B'ye goturur (A'da boşta eleman kalmaz),

2) A' daki bir eleman B'de birden fazla elemana karşılık gelmez.

Burada A'ya tanim kumesı; (domain) B'ye değer kumesi (codomain) denir.

Fonksiyonlari bir makine gibi de dusunebiliriz:

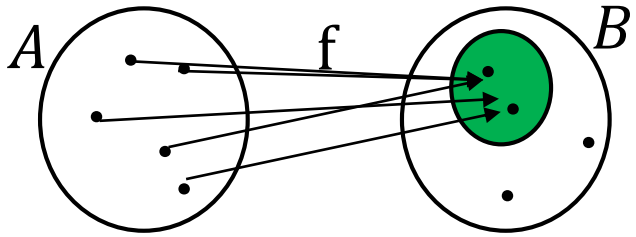


ör. $\mathrm{f}(x) = x^2$ formulu ile verilen f, $\mathbb{R}$'den $\mathbb{R}$'ye bir fonksiyondur.

- f, aldiği her sayiyi karesine gonderir/donusturur/eşler.
- $x \in \mathbb{R}$'in goruntusu f altinda x^2'dir.

Fonksiyonun görüntüsü (image or range of a function):

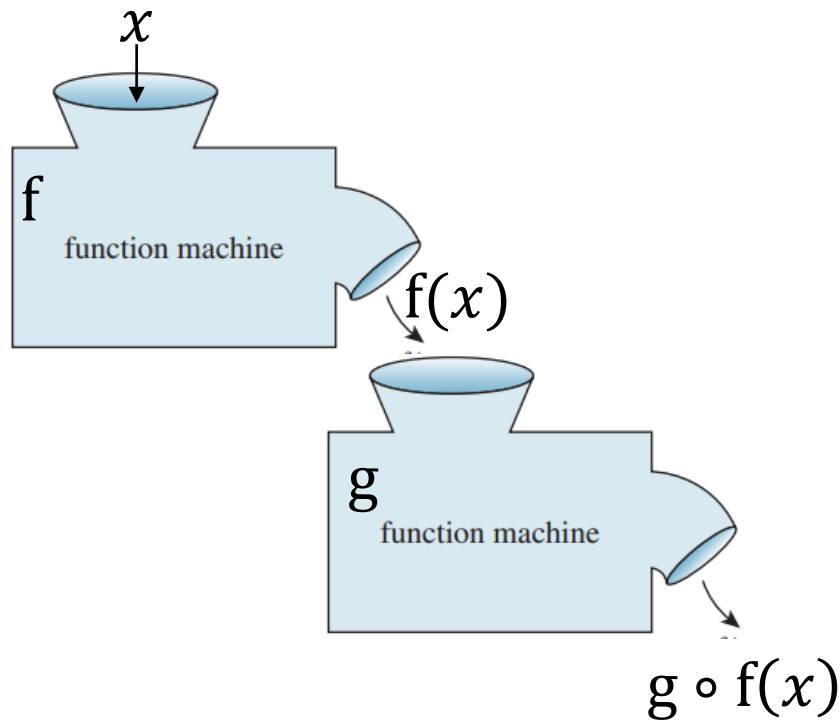
$f : A \to B$ fonksiyonu A kumesindeki her elamani B'den bir elemana esler. Fakat bu, B'deki butun elemanlarin kullanildigi (ortuldugu) anlamina gelmez. B'deki $f(x)$ $(x \in A)$ elamanlari f fonksiyonun goruntusunu olusturur.



Birleşik Fonksiyon

$f : A \to B$ fonksiyonu ve $g: B \to C$ ye fonskiyonlari verilsin.

Bu durumda $g \circ f: A \to C$ fonksiyonuna f ve g fonksiyonlarinin birleşik fonksiyonu denir.



or. $f : \mathbb{R} \to \mathbb{R}$ fonksiyonu $f(x) = x^2$ seklinde tanimlansin.

$g : \mathbb{R} \to \mathbb{R}$ fonksiyonu $g(x) = 2x + 1$ seklinde tanimlansin.

Bu durumda $g \circ f: \mathbb{R} \to \mathbb{R}$ bilesik fonskiyonu:

$$g \circ f\,(x) = g\big(f(x)\big) = g(x^2) = 2x^2 + 1.$$

Algoritmalar

Algoritma, bir girişi bir çıkışa dönüstüren adım adım prosedurdur: $x \xrightarrow{\text{algoritma}} f(x)$

Algoritma sıralanmış basit operasyonlardan olusan bir fonksiyondur.

Algoritmalar pseudo kod (sözde/kaba kod) ile verilir.

ör. Aşağıdaki pseudo kod girilen bir listenin en buyuk elemaninin indeksini çıktı olarak verir.

```
enBuyukIndeksBul(L):
Giriş: n elemanli bir liste L.
Çıkış: i indeksi oyleki L[i], L'deki en buyuk değer
1. enBuyukIndeks := 1
2. for i:=2 to n:
3.         if L[i] > L[enBuyukIndeks] then
4.           enBuyukIndeks :=i
5. return enBuyukIndeks
```

LOJİK (MANTIK)

Önerme (prepsotion): Doğru yada yanlış bir ifadeye (statement) önerme denir.

örnekler: 3+2=5 (doğru onerme).

49 bir asal sayidir (yanlis onerme).

Sivas Türkiye'nin batisinda değildir (dogru onerme).

O bir ogrencidir. onerme degil O'nun kim olduguna bagli degisir.

$x + y > 3$ onerme degil, cunku bellirsiz, kesin değil.

Hava soğuk mu? Soru cumlesi- onerme degil.

Önerme Çesitleri:

Onermeler parçalanabilir olma durumlarina gore ikiye ayrilir: atomik ve birlesik

1. Atomik Önerme: Daha basit ifadelere parçalanamayan önermelere atomik önerme denir:

ör. Sivasspor birinci ligdedir.

* Atomik onermelere bazen Boolyan (Boolean) değişken de denir.

2. Birleşik Önerme: Mantıksal bağlaçlarla birbirine baglanmis atomik önermelerin olusturudugu önermedir.

Mantıksal Bağlaçlar ($\backsim$, $\wedge$, $\vee$)

Mantıksal bağlaçlar atomik önermelerden komplike birleşik önermeler oluşturan yapışkanlardır. Bu bağlaçlar üç tanedir.

1. Değil Bağlacı ($\backsim$)

$\backsim p$, p'nin değili diye okunur. p doğru iken $\backsim p$ yanlis; p yanlış iken $\backsim p$ doğrudur.

Bir önermenin degili, bu önerme yanlis oldugunda ne anlama geldigini anlatir.

ör. p: bugün hava sıcak; $\sim p$: bugün hava sıcak değil.

Doğruluk Tablosu (Truth table)

p	$\backsim p$
D	Y
Y	D

Ve Bağlacı ($\wedge$)

$p \wedge q$; yalnızca p ve q aynı anda doğru olduğunda doğru olur, diğer bütün hallerde (p doğru q yanlış; q doğru p yanlış; p ve q yanlış) $p \wedge q$ yanlıştır.

ör. p: sen mecnunsun; q: ben leylayim

$\backsim p \wedge \backsim q$: ne sen mecnunsun nede ben leyla

p	q	$p \wedge q$
D	Y	Y
Y	D	Y
Y	Y	Y
D	D	D

$\longrightarrow$ hem p hem de q doğru olmak zorunda

Not: $p_1 \wedge p_2 \wedge \cdots \wedge p_k$ bu k tane önermeden tek bir tanesi yanliş olsun birleşik önerme yanlış olur.

Veya Bağlacı (V)

$p \vee q$; doğru olmasi için p veya q nun doğru olmasi yeterlidir. $p \vee q$; yalnızca p ve q ayni anda yanlis olduğunda yanlistir.

ör. p: elma bir meyvedir, q: kaplumbağalar uçar.

$p \vee q$: elma bir meyvedir veya kaplumbagalar ucar.

p	q	$p \vee q$
D	Y	D
Y	D	D
Y	Y	Y
D	D	D

→ yalnızca hem p hem de q nun yanlış olduğunda $p \vee q$ yanlıştır.

Not: $p_1 \vee p_2 \vee \cdots \vee p_k$ bu k tane önermeden tek bir tanesi doğru olmasi bırleşik önermenin dogru olmasina yeter.

Kodlamada Ve ve Veya (Kısa Devre Değerlendirme)

Bir çok programlama dilinde compiler bir if state'deki ve (∧) ile bağlanmış bir birleşik onermenin dogru olup olmadiğini degerlendirirken soldan başlayarak sağa dogru her bir onermemenin doğruluğuna bakar. Yanlış bir onermeye denk geldiginde sağa doğru ilerlemeyi bırakir ve if bloğunu calistirmaz.

ör. `if` $(2 > 3 \;\&\&\; x + y < 8)\{\quad\} \equiv$ `if` $(2 > 3\;)\{\quad\}$

Aynı şekilde veya (V) ile olusturulmus bir birlesik onerme degerlendirilirken compiler onermeleri soldan saga dogru degerlendirerek ilerler. Herhangi bir onermenin dogru oldugunda compiler artik daha fazla sağa dogru ilerlemez, birlesik onermeyi kabul eder ve if bloğunu calıştırır.

ör. `if` $(x == 0 \;||\; (x - 1)/x > 0)\;\{\quad\}$

`if` $(x > 20 \;||\; x <= 20 \;\&\&\; y < 0)\{\quad\} \equiv$ `if` $(y < 0)\{\quad\}$

Harici Veya (Exclusive OR (XOR))

Veya bağlacı ile bağlanan önermelerin ayni anda doğru olmasi bazen mumkun degildir.

ör. Haftasonu Mersin'de veya Sivas'ta olacagim.

p: Mersin'de olmak, q: Sivas'ta olmak.

Aynı anda hem Mersin'de hem de Sivas'ta olmak mumkun degildir. Oysa $p \vee q$ önermesi, p ve q dogru oldugunda dogru idi.

Bu noktada yeni bir mantıksal bağlaca ihtiyaç vardir. Bu bağlaca harici veya (exclusive or (XOR)) diyecegiz ve $\oplus$ ile gosterecegiz.

$p \oplus q$; p veya q dan yalniz biri dogru oldugunda dogru olur.

p	q	$p \oplus q$
D	Y	D
Y	D	D
Y	Y	Y
D	D	Y

→ yalnızca p veya q nun bir tanesi dogruoldugunda

→ $p \oplus q$ doğru olur.

Çıkarım (implication) (if – then)

$p \Rightarrow q$: p'nin doğruluğu q'nun dogru olmasina isaret ederse $p \Rightarrow q$ birlesik onermesi dogurudur.

ör. piyango kazanirsam araba alacagim.

p: piyangoyu kazanmak, q araba almak.

Burada p'ye öncül (premise) , q'ya sonuç (conclusion) denir.

Not burada piyangoyu kazanmassam araba almayacagim anlami cikmamali. Belki yine araba alacagim ama kesin degil; p'nin varligi q'yu garanti ediyor, ama p yoklugu q'nun yoklugunu garanti etmez: $p \Rightarrow q \not\equiv \backsim p \Rightarrow \backsim q$.

p	q	$p \Rightarrow q$
D	Y	Y
Y	D	D
Y	Y	D
D	D	D

→ $p \Rightarrow q$ yalnizca p doğru, q yanlış olduğunda yanlış olur diğer tüm durumlarda doğrudur.

→ kaplumbağalar uçar ise gökyüzü yeşildir.

Not: Bir onceki not su sekilde de aciklanabilinir. $p \Rightarrow q$ onermesinde p ile q arasinda bir nedensellik (causation) yoktur. p, q ya neden olmaz, q kendi basina da dogru olabilir.

ör. 90 kg'yi gecersem spor yapacagim.

p: 90 kg'yi gecmek; q: spor yapmak

Spor yapmak icin (q'nun olmasi) 90 kg'yi gecmek (p'nin olmasi) zorunda degilim.

ör. *Yazacaginiz sifre en az 8 karakater uzunlugunda olursa ve daha once kullanmadiginiz bir sifre olursa gecerli olur.*

p: yazilan sifre en az 8 karakterli , q: yazilan sifre daha once kullanilmis, r: sifre gecerli

$$p \wedge \sim q \Rightarrow r$$

ör. p: oy kullanmak, q: Türk vatandaşi olmak

$p \Rightarrow q$ Oy kullaniyorsan Türk vatandaşısın.

oy kullanman Türk vatandaşi olamanı ifade eder/ışaret eder/gerektirir.

Ancak Ve Ancak (if and only if)(gerek ve yeter şart)

$p \Leftrightarrow q$, p ve q ayni anda dogruysa yada p ve q ayni anda yanlışsa doğru olur.

ör. p: ıslanmam, q: yağmurun yağması

$p \Leftrightarrow q$: ıslanmam için ancak ve ancak yağmurun yağması gereklidir.

p	q	$p \Leftrightarrow q$
D	Y	Y
Y	D	Y
Y	Y	D
D	D	D

→ p ve q ayni anda yanlış veya

→ p ve q ayni anda doğru olmali.

Not: $p \Leftrightarrow q$; $p \Rightarrow q$ ve $q \Rightarrow p$ şeklinde iki parçaya ayrilabilinir: $p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$

$p \Rightarrow q$: ıslanmışsam yagmur yağmıştır

$q \Rightarrow p$: yağmur yağmışsa islanmışımdır.

Totoloji

Eğer bir birleşik önerme kendini oluşturan önermeler (p ve q lar) ne olursa olsun (doğru yada yanlış) her zaman doğru oluyorsa (doğruluk tablosunda yalnizca D varsa) bu önermeye *totoloji* denir.

* En önemli totolojilerden biri $p \vee \sim p$ dir. p ne olursa olsun $p \vee \sim p$ hep doğrudur.

p	$\sim p$	$p \vee \sim p$
D	Y	D
Y	D	D

$\longrightarrow$ yalnizca D var.

ör. p: yağmur yağıyor, $\sim p$ yağmur yağmiyor; $p \vee \sim p$ yagmur yagiyor veya yagmur yagmiyor.

* Bir baska önemli totoloji Modus Ponens'tir: $p \wedge (p \Rightarrow q) \Rightarrow q$

p	q	$p \Rightarrow q$	$p \wedge (p \Rightarrow q)$	$p \wedge (p \Rightarrow q) \Rightarrow q$
D	D	D	D	D
D	Y	Y	Y	D
Y	D	D	Y	D
Y	Y	D	Y	D

ör. p: 90 kg'yi gecmek, q: spora başlamak.

$p \wedge (p \Rightarrow q) \Rightarrow q$:

"90 kg'yi gectim" ve "90 kg'yi gecersem spora baslayacagim" o halde spora başlayacagim.

Çelişki (Tezat) (Contradiction)

Çelişki, totolojinin tersidir. Her zaman yanlış olan birleşik önermeye denir.

ör. $(p \Leftrightarrow q) \wedge (p \oplus q)$

p	q	$p \Leftrightarrow q$	$p \oplus q$	$(p \Leftrightarrow q) \wedge (p \oplus q)$
D	D	D	Y	Y
D	Y	Y	D	Y
Y	D	Y	D	Y
Y	Y	D	Y	Y

ör.

Mantıksal Bağlaçların İşlem Sırası

Mantıksal bağlaçlar arasındaki işlemde öncelik sırası şöyledir:

değil	$\sim p$	en yüksek öncelik ↓
ve	$p \wedge q$	
veya	$p \vee q$	
XOR	$p \oplus q$	
çıkarım	$p \Rightarrow q$	
ancak ve ancak	$p \Leftrightarrow q$	en düşük öncelik

ör. $p \vee q \Leftrightarrow p \equiv (p \vee q) \Leftrightarrow p$

ör. $p \wedge \sim q \Rightarrow r \Leftrightarrow s \equiv \big((p \wedge (\sim q)) \Rightarrow r\big) \Leftrightarrow s$

ör. $p \Rightarrow q \Rightarrow r \wedge s \equiv (p \Rightarrow q) \Rightarrow (r \wedge s)$

Mantıksal Denklik (Logical Equivalance)

İki önermenin mantıksal olarak denk olabilmesi için bu önermelerin doğruluk tablolarinin aynı olması gerekir.

Başka bir ifadeyle Φ ve Ω önermeleri birbirine mantıksal olarak denk ise $\Phi \Leftrightarrow \Omega$ her zaman doğru olur (totoloji) olur.

ör. De Morgan Kuralları 1. $\sim(p \wedge q) \equiv \sim p \vee \sim q$ 2. $\sim(p \vee q) \equiv \sim p \wedge \sim q$

p	q	$\sim p$	$\sim q$	$p \vee q$	$\sim(p \vee q)$	$\sim p \wedge \sim q$	$p \wedge q$	$\sim(p \wedge q)$	$\sim p \vee \sim q$
D	D	Y	Y	D	Y	Y	D	Y	Y
D	Y	Y	D	D	Y	Y	Y	D	D
Y	D	D	Y	D	Y	Y	Y	D	D
Y	Y	D	D	Y	D	D	Y	D	D

Mantıksal Denklik

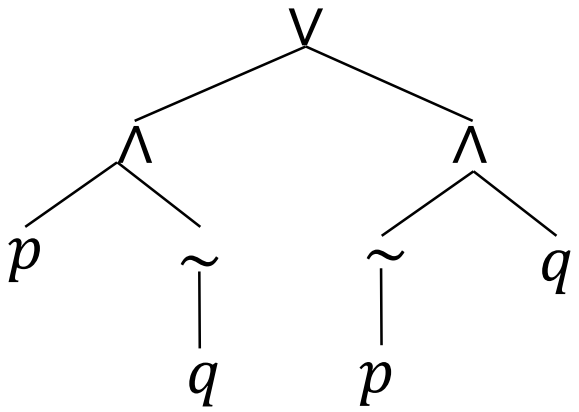
ör. $p \Rightarrow q$ önermesinin karşıtı: (converse) $q \Rightarrow p$ ve tersi (inverse) : $\sim p \Rightarrow \sim q$ de mantıksal olarak biribirine denk önermelerdir.

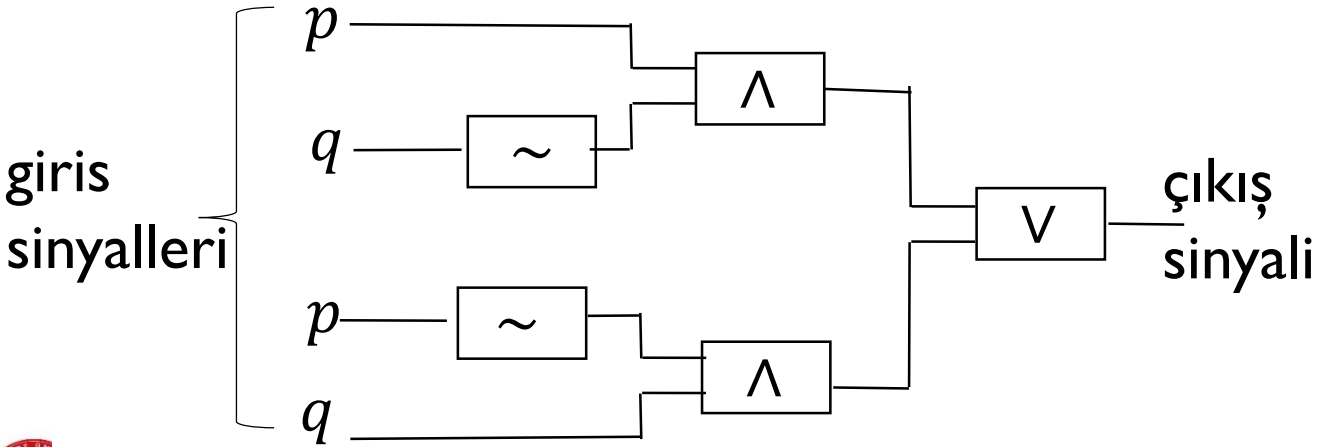
p	q	$p \Rightarrow q$	$q \Rightarrow p$	$\sim p \Rightarrow \sim q$
D	D	D	D	D
D	Y	Y	D	D
Y	D	D	Y	Y
Y	Y	D	D	D

Mantık Devreleri (Logic Circuits)

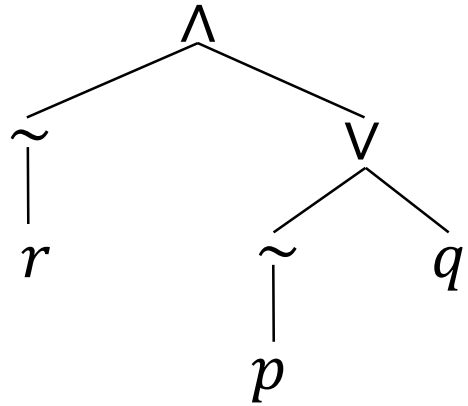
Birleşik önermeleri elektrik devresi olarak ifade edebiliriz. Bunun için verilen önermeyi öncelikle ağaç diyagramına çevirmeliyiz.

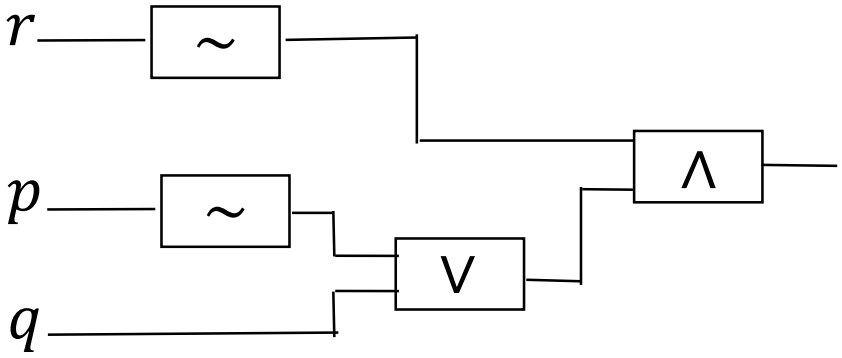
ör. $(p \wedge \sim q) \vee (\sim p \wedge q)$





ör. $\sim r \wedge (\sim p \vee q)$





Mantık Devreleri (Logic Circuits)

Not: değil, ve, veya operatörleri bazi kaynaklarda aşağıdaki gibi de gösterilir.

Inverter

OR gate

AND gate