Mantık

Önerme Mantığı

Önerme, Doğru veya Yanlış olan bir gerçeği bildiren bir cümledir.

Aşağıdaki ifadelerin tümü önermedir:

Ankara, Türkiye'nin başkentidir.

1+1, 2'dir.

1+1, 3'tür.

Jawaharlal Nehru, Hindistan'ın ilk başbakanıdır.

Dün yağmur yağdı.

Eğer x bir tamsayı ise, o zaman x^2 de bir tamsayıdır

Aşağıdaki ifadeler önerme değildir:

Lütfen sabah 11'de rapor verin

Adın ne?

$x^2 = 13$

Bu kurabiye ne kadar?

Bu cümle yanlıştır.

Önerme mantığı, yeni önermeler türetmek amacıyla önermeleri birleştirmek için bir dizi biçimsel kuraldan oluşur. Önermeleri temsil etmek ve her önerme kuralını fonksiyon biçiminde tanımlamak ve boole değişkenlerini ifade etmek için p, q ve r benzeri harfler kullanılır. İfadeler veya önerme değişkenleri, bileşik ifadeler adı verilen tek bir ifade oluşturmak için Tablo'da verilen mantıksal bağlantılar (operatörler) aracılığıyla birleştirilir.

Symbol	Connective	Name
$\sim$	Not	Negation
$\wedge$	And	Conjunction
$\vee$	Or	Disjunction
$\longrightarrow$	Implies or if...then	Implication or conditional
$\longleftrightarrow$	If and only if	Equivalence or biconditional

Doğruluk tablosu, bileşenlerinin doğruluk değerlerini kullanarak bileşik önermelerin doğruluk değerlerini gösterme yöntemidir. Tipik olarak olası doğruluk değerlerini temsil eden satırlarla ve önermeleri temsil eden sütunlarla oluşturulur.

Olumsuzlama (Negation)

Olumsuzlama, karşıt doğruluk değerine sahip bir ifadedir. ¬p ile gösterilen bir p önermesinin yadsınması, "P öyle değil ki" önermesidir.

Örneğin, "Bugün Cuma" önermesinin olumsuzlanması. "Bugün Cuma öyle değil." olurdu veya daha kısaca "Bugün Cuma değil."

p	$\neg p$
True	False
False	True

VE Bağlacı (Conjunction)

"Ben bir kayayım ve ben bir ada'yım."

p ve q önerme olsun. Matematikte p∧q ile gösterilen p ve q bağlacı, hem p hem de q Doğru olduğunda Doğru, aksi takdirde Yanlış'tır.

$$| 1 \land 0 = 0 \mid 0 \land 1 = 0 \mid 1 \land 1 = 1 \mid 0 \land 0 = 0 |$$

p	q	$p \land q$
True	True	True
True	False	False
False	True	False
False	False	False

Python Kodu

```
x = (1 and 0)
print(x)
# >> 0
x = (1 and 1)
print(x)
# >> 1
x = (False and True)
print(x)
# >> False
x = (True and True)
print(x)
# >> True
```

VEYA Bağlacı (Disjunction)

"Çok çalıştı ya da son derece zeki."

p ve q önerme olsun. Matematikte p∨q ile gösterilen p ve q'nun ayrışımı, p ve q'dan en az biri Doğru olduğunda Doğru, aksi halde Yanlış'tır. Bir VEYA ifadesinin doğru olması için ya "a" ya da "b" doğru olmalıdır.

$$| \ 1 \vee 0 = 1 \ | \ 0 \vee 1 = 1 \ | \ 1 \vee 1 = 1 \ | \ 0 \vee 0 = 0 \ |$$

p	q	$p \vee q$
True	True	True
True	False	True
False	True	True
False	False	False

Python Kodu

```
x = (0 or 0)
print(x)
# >> 0
x = (1 or 0)
```

```
print(x)
# >> 1
x = (1 or 1)
print(x)
# >> 1
x = (False or False)
print(x)
# >> False
x = (False or True)
print(x)
# >> True
x = (True or True)
print(x)
# >> True
```

Özel VEYA (Exclusive Disjunction)

"Ya 2 Advil ya da 2 Tylenol alın."

p ve q önerme olsun. Matematikte p⨁q (⨁ veya ⊻) ile gösterilen p ve q'nun (XOR olarak da bilinir) özel ayrımı, tam olarak p ve q'dan biri Doğru olduğunda Doğru, aksi halde Yanlış'tır. Özel veya şu anlama gelir: “a” veya “b” ama “a” ve “b” değil. Asla ikisi aynı anda doğru veya yanlış olamaz.

| 1 ⊻ 0 = 1| 0 ⊻ 1= 1|1 ⊻ 1= 0| 0 ⊻ 0= 0 |

p	q	$p \oplus q$
True	True	False
True	False	True
False	True	True
False	False	False

Python Kodu

```
# Python'da XOR bildirimlerini tanımlamak için ^ sembolü kullanılır.
x = (0 ^ 0)
print(x)
# >> 0
```

```
x = (1 ^ 1)
print(x)
# >> 0
x = (1 ^ 0)
print(x)
# >> 1
x = (0 ^ 1)
print(x)
# >> 1
```

Çıkarım-İma / eğer-öyleyse (⟶): (Implication)

"Final sınavından 100 alırsanız, sınıfta/dersten A alırsınız."

p ve q önerme olsun. Matematikte p⟹q ile gösterilen p ve q'nun anlamı, "eğer p ise sonra q ise" ifadesinin kısa yoludur. Bu nedenle, p Doğru olduğunda, çıkarımın q'nun Doğru olmasını gerektirir. p Doğru değilse, q herhangi bir değer olabilir. Başka bir deyişle, p Doğru ve q Yanlış olduğunda çıkarım başarısız olur (Yanlıştır). Bunun "p ancak ve ancak q ise"den farklı olduğuna dikkat edin.

$$| \, 0 \rightarrow 0 = 1 \, | \, 0 \rightarrow 1 = 1 \, | \, 1 \rightarrow 1 = 1 \, | \, 1 \rightarrow 0 = 0 \, |$$

p	q	$p \implies q$
True	True	True
True	False	False
False	True	True
False	False	True

Uygulama, yalnızca koşullar yerine getirildiğinde ve sonuçlar yerine getirilmediğinde başarısız olan bir "sözleşme" olarak kabul edilebilir.

Pyhon Kodu

```
def conditional(x, z):
  if x == True:
    return z
  return True

x = True
z = False
```

```
print(conditional(x, z))
```

3.1.6. Bir Çıkarımın Converse, Contrapozitif ve Tersi (Converse, Contrapositive and Inverse of an Implication)

p⟹q çıkarımından yeni bileşik önermeler oluşturabiliriz. Onlar

(converse of implication)=**tersi/yer değiştirme** : q⟹p

(contrapositive of imlication)=**Zıt anlamlı** : ¬q⟹¬p

(inverse of implication)=**ters** ¬p⟹¬q

Bu yeni önermelerin doğruluk değerleri aşağıdaki tabloda gösterilmiştir.

p	q	$p \implies q$ **(conditional)**	$q \implies p$ **(converse)**	$\neg q \implies \neg p$ **(contrapositive)**	$\neg p \implies \neg q$ **(inverse)**
True	True	True	True	True	True
True	False	False	True	False	True
False	True	True	False	True	False
False	False	True	True	True	True

Önerme denklikleri bölümünde, doğruluk tablosunun koşullu p⟹q ve zıt pozitif ¬q⟹¬p'nin mantıksal olarak eşdeğer olduğunu ve ters q⟹p ve ters ¬p⟹¬q'nun neden mantıksal olarak eşdeğer olduğunu gösterdiğini açıklayacağız.

Bu fikirleri bir örnekle açıklayalım.

Örnek 6 - Koşullu, Converse, Contrapozitif ve Ters.

a) "Bir n tamsayısı 4'e bölünebiliyorsa, 2'ye de bölünebilir" ifadesini bir koşullu kullanarak çevirin.

b) Tersini/yerdeğiştirmesini, kontrapozitifini ve ters'ini oluşturun.

Çözüm

p: Bir tamsayı n, 4'e bölünebilir

q: Bir tamsayı n, 2'ye bölünebilir

önermelerini göstersin.

"Bir n tamsayısı 4'e bölünebiliyorsa, 2'ye de bölünebilir" cümlesi p⟹q olarak çevrilir.

Bunun tersi $q \Longrightarrow p$'dir ve şu şekilde çevrilebilir: "Eğer bir n tamsayısı 2'ye bölünebiliyorsa, o zaman 4'e de bölünebilir."

Çelişkili/contrapositive olan $\neg q \Longrightarrow \neg p$'dir ve şu şekilde çevrilebilir: "Eğer bir n tamsayısı 2'ye bölünemiyorsa, o zaman 4'e de bölünemez."

Tersi $\neg p \Longrightarrow \neg q$ şeklindedir ve şu şekilde çevrilebilir: "Eğer bir n tamsayısı 4'e bölünemiyorsa, o zaman 2'ye de bölünemez."

3.1.7. Eğer ve Sadece Eğer (↔) ise | iki koşullu (Bi-Implication, Bi-condition)

"Dışarıda yağmur yağıyor, ancak ve ancak hava bulutluysa."

p ve q önerme olsun. Matematikte $p \Leftrightarrow q$ ile gösterilen p ve q'nun iki anlamı, "p ancak ve ancak q ise" ifadesinin kısaltmasıdır. İki koşullu, her iki önermenin de aynı değere sahip olup olmadığını kontrol ederek "eğer ve ancak" koşulunu temsil eden bir bağlaçtır. Bu nedenle, bi-implication, yalnızca p True olduğunda q'nun True olmasını gerektirir. Başka bir deyişle, p Doğru ve q Yanlış veya p Yanlış ve q Doğru olduğunda çift yönlü ima başarısız olur (Yanlıştır). İki koşullu ifadeler ↔ sembolü ile gösterilir ve $A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$ ifade edilir.

$| \, 0 \leftrightarrow 0 = 1 \, | \, 0 \leftrightarrow 1 = 0 \, | \, 1 \leftrightarrow 1 = 1 \, | \, 1 \leftrightarrow 0 = 0 \, |$

p	q	$p \iff q$
True	True	True
True	False	False
False	True	False
False	False	True

İki yönlü ima ve tek yönlü ima ile zıtlık oluşturmak önemlidir. "Final sınavından 100 alırsanız, sınıfta/dersten A alırsınız" çıkarım örneğini düşünün. Bu, finalde 100 aldığınızda sınıfta/dersten de A alacağınız anlamına gelir.

İki koşullu bir sonuç olarak, "Yalnızca ve ancak sınıfta/dersten A alırsanız final sınavından 100 alırsınız" önermesini göz önüne alalım. Bu önerme, finalde 100 alarak sınıfta A kazanabileceğiniz iki yönlü bir sözleşme anlamı taşır. Ancak finalde 100 almazsanız A kazanamazsınız.

Python Kodu

```
def main(a, b):

  def conditional(a, b):
    if a == True:
      return b
    return True

  A = conditional(a, b)
  B = conditional(b, a)
  if A == B:
    return True
  else:
    return False

a, b = True, False
print(main(a, b))

a, b = True, True
print(main(a, b))

a, b = False, True
print(main(a, b))

a, b = False, False
print(main(a, b))
```

Örnek 8

Aşağıdaki kod, bileşik önermenin doğruluk tablosunu ortaya koymaktadır:

$(p \wedge q) \vee \neg q$

Hatırlayın: $\neg q$, q değil için matematiksel kestirme yoldur.

Çalışma örneği

Bileşik önermenin doğruluk değerini ortaya çıkarmak için doğruluk tablosu oluşturun:

$(p \vee \neg q) \wedge \neg p$

Kendi doğruluk tablonuzu oluştururken, basit önermelerin her biri için olası tüm doğruluk değerlerine sahip olduğunuzdan emin olmak konusunda sistematik olmak çok önemlidir. Her basit önermenin iki olası doğruluk değeri vardır, bu nedenle tablodaki satır sayısı 2n olmalıdır, burada n önerme sayısıdır. Ayrıca karmaşık önermeleri daha küçük parçalara ayırmayı da düşünün.

Örnek 9

Bileşik önerme için bir doğruluk tablosu oluşturun:

$(p \wedge q) \Longrightarrow (p \wedge r)$ p, q, r'nin tüm değerleri için.

Çözüm

8 satırı olmalıdır - çünkü üç basit önerme vardır ve her birinin iki olası doğruluk değeri vardır.

p	q	r	$p \wedge q$	$p \wedge r$	$(p \wedge q) \implies (p \wedge r)$
T	T	T	T	T	T
T	T	F	T	F	F
T	F	T	F	T	T
T	F	F	F	F	T
F	T	T	F	F	T
F	T	F	F	F	T
F	F	T	F	F	T
F	F	F	F	F	T

Mantıksal Çeviriler

Uzun zaman önce filozoflar, düşüncelerimizi sembollere dönüştürebileceğimizi ve akıl yürütme çizgilerini daha kolay takip edebileceğimizi keşfettiler. Bu, modern teknolojik toplumumuzun nihai gelişiminde ve dijital bilgisayar kullanımımızda önemli bir adımdı. Bilgisayarların çalışabilmesi için önce düşüncelerimizi onlara vermemiz gerekir.

AMA, İngilizce dili zordur ve aynı mantıksal ifadeleri temsil etmek için birçok farklı ifade kullanırız. İngilizce cümlelerden sembollere ve geri ifadelere çevirmek çok pratik gerektiren bir beceridir.

3.2. Önerme Denklikleri

Her durumda aynı doğruluk değerlerine sahiplerse, iki önerme mantıksal olarak eşdeğer (veya basitçe eşdeğer) olarak kabul edilir. Bunu iki önerme için bir doğruluk tablosu oluşturarak ve karşılaştırarak görmek genellikle en kolayıdır.

Örnek 10

$\neg p \vee q$ ve $p \Longrightarrow q$ önermelerini göz önünde bulundurun.

p	q	$\neg p \vee q$	$p \implies q$
True	True	True	True
True	False	False	False
False	True	True	True
False	False	True	True

İki bileşik önerme için tüm satırlardaki doğruluk tablosu aynı olduğundan, bunlar eşdeğerdir.

Örnek 11

Üç bileşik önerme düşünün:

$(p \wedge q) \Longrightarrow r$

$(p \Longrightarrow q) \wedge (p \Longrightarrow r)$

$p \Longrightarrow (q \wedge r)$

Doğruluk tablolarını oluşturun (siz oluşturun).

3.2.1. De Morgan'ın Kanunları

İki önemli mantıksal eşdeğerlik De Morgan Yasasıdır. Bunlar, olumsuzlamayı ve veya operatörleri arasında nasıl "dağıttığımızı" açıklar.

De Morgan Kuralı

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$

Mantıksal olarak eşdeğer olan iki ifadeyi belirtmek için ≡ sembolünü kullanırız.

Örnek 12

Yukarıda ima önermesi için verilen doğruluk tablosunu hatırlayın.

p	q	$p \implies q$	$\neg(p \implies q)$
True	True	True	False
True	False	False	True
False	True	True	False
False	False	True	False

3.2.2. Totolojiler, Çelişkiler ve Olasılıklar/beklenmedik durum

Bir önerme, doğruluk değeri her zaman Doğru ise bir totolojidir.

Bir önerme, doğruluk değeri her zaman Yanlış ise bir çelişkidir.

Ne totoloji ne de çelişki olmayan bir önermenin beklenmedik durum (contingency) olduğu söylenir.

Örnek 13 - Totoloji ve Çelişki

$p \vee \neg p$ bir totoloji örneğidir.

$p \wedge \neg p$ bir çelişki örneğidir.

Bu doğruluk tablosunda görülebilir.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
True	False	True	False
False	True	True	False

p∨¬p için doğruluk değerlerinin tümünün Doğru ve p∧¬p'nin tümünün Yanlış olduğuna dikkat edin.

3.4. **Mantık Uygulamaları**

Bu bölümde, mantığın bilgi teknolojisi ve bilgisayar bilimi için iki uygulamasını ele alacağız. Birincisi, bitsel işlemleri, ikincisi ise mantık devrelerini tasarlamayı ve analiz etmeyi içerir.

3.4.1. **bit düzeyinde işlemler**

Bit düzeyinde işlem, işlenenin/işlenenlerin ayrı bitleri (0'lar veya 1'ler) üzerinde çalışan ve özetlenen bir Boole işlemidir.

Bitsel İşlemler

"&" ile gösterilen bit düzeyinde AND , ve ∧'yi her işlenenin karşılık gelen bitlerine uygular.

"|" ile gösterilen bit düzeyinde OR, veya ∨'yi her işlenenin karşılık gelen bitlerine uygular.

"∧" ile gösterilen bit düzeyinde XOR, her işlenenin karşılık gelen bitlerine ayırıcı veya ⊕ uygular.

"!" ile gösterilen bitsel NOT , olumsuzlamayı ¬ (0⟷1 döndürür) her işlenenin karşılık gelen bitlerine uygular.

Bitsel boole operatörleri için doğruluk tabloları aşağıda özetlenmiştir.

p	q	AND &	OR |	XOR ^	IF ⇒	IFF ⇔
1	1	1	1	0	1	1
1	0	0	1	1	0	0
0	1	0	1	1	1	0
0	0	0	0	0	1	1

Örnek 24 - **Bitsel İşlemler**

Aşağıdaki ikili sayılar için bit düzeyinde VE, VEYA, XOR'u bulun,

A=111101

B=001111

Çözüm

Sonuçların alt satırda not edildiği Boole operatörleri için doğruluk tablolarını kullanarak,

Bitwise AND	Bitwise OR	Bitwise XOR
111101	111101	111101
001111	001111	001111
001101	**111111**	**110010**

3.4.2. **Mantık Devreleri**

Mantık devreleri, bir bilgisayar işlemcisinin aritmetik ve mantık birimlerinin tasarımında önemlidir. İkili olarak iki adet 8 bitlik sayı ekleme problemini düşünün.

İkili 0+0=0 ve 1+0=0+1=1'de, ancak ondalık toplamada olduğu gibi, ikili 1+1=2'de, bu ikilide toplamı 0 ve 1'in taşıması olacaktır. Soldaki bir sonraki önemli sütun.

Daha sonra, örneğin A ve B gibi iki ikili basamaktan oluşan belirli bir sütun eklemeyi düşünmek, girdi olarak A, B rakamlarını içerir ve önceki sütundan gelen, C_{in} dir.

Çıktı, S toplamı olacak ve C_{out} denir, bir sonraki sütuna yapılacak.

Bunlar, ikili toplayıcı denilen şeyin temel bileşenleridir.

Şekil 10. İkili bir toplayıcı

A,B,C_{in} dijital girişlerine ve S ve C_{out} dijital çıkışlarına dayalı ikili ekleme için mantık tablosu aşağıdaki özetlenmiştir.

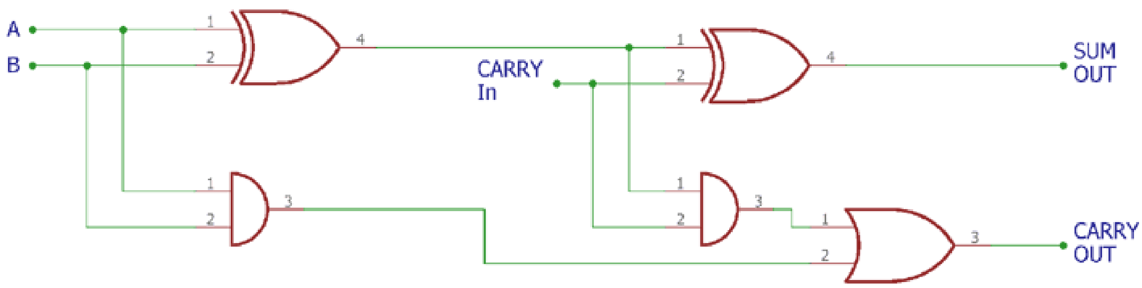
Tablo 2. İkili toplayıcı için doğruluk tablosu

A	B	C_{in}	**S**	**C$_{out}$**
1	1	1	**1**	**1**
1	1	0	**0**	**1**
1	0	1	**0**	**1**
1	0	0	**1**	**0**
0	1	1	**0**	**1**
0	1	0	**1**	**0**
0	0	1	**1**	**0**
0	0	0	**0**	**0**

S ve C_{out} çıkışlarının mantığının aşağıdaki önermelerle verildiği gösterilebilir.

$$C_{out} = (A \wedge B) \vee (B \wedge C_{in}) \vee (A \wedge C_{in})$$

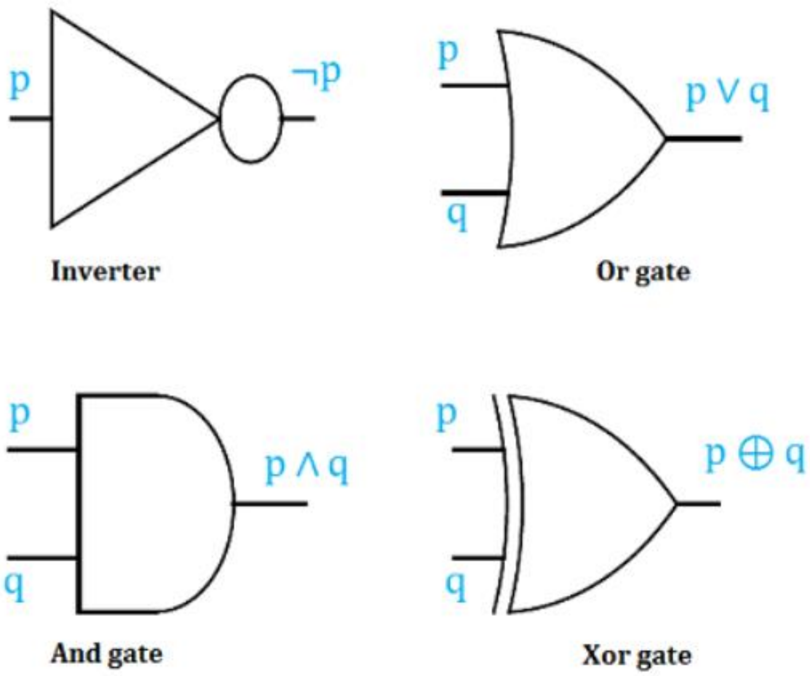
$$S = (\sim A \wedge \sim B \wedge C_{in}) \vee (\sim A \wedge B \wedge \sim C_{in}) \vee (A \wedge \sim B \wedge \sim C_{in}) \vee (A \wedge B \wedge C_{in})$$



Full-Adder Circuit

Girişlere (A,B,C_{in}) dayalı olarak bu mantıksal çıkışların uygulanması, mantık kapıları adı verilen elektronik devrelerin kullanılmasıyla olur.

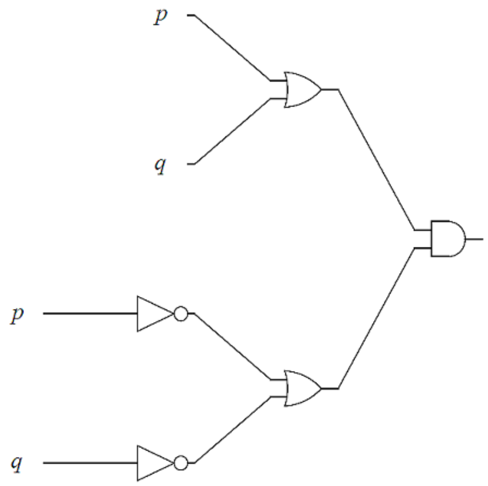
Temel mantık kapıları, Çevirici veya Değil kapısı, Ve kapısı, Or kapısı ve Xor kapısıdır. Her biri için grafik gösterimi aşağıda gösterilmiştir.



Bu bölümü, önce mantık devrelerini giriş değişkenleri açısından çıkışlarını vermek üzere analiz ederek ve ardından mantıksal ifadelere dayalı mantık devreleri oluşturarak sonlandırıyoruz.

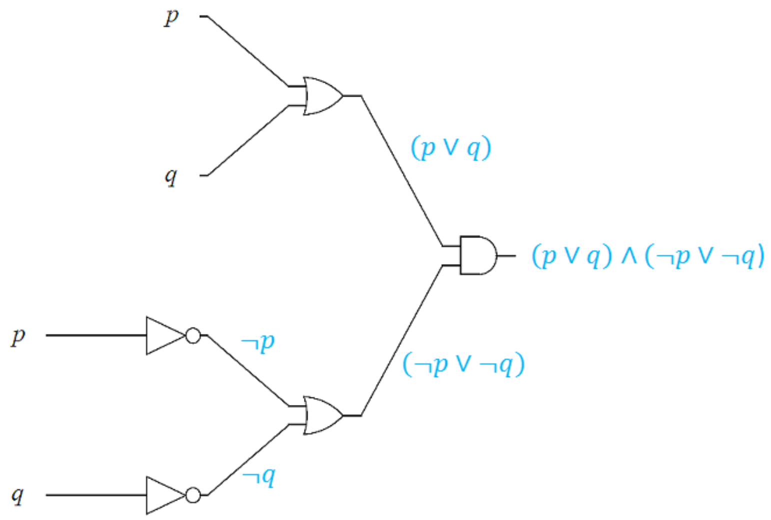
Örnek 25 - Bir mantık devresinin terim cinsinden çıkışı Giriş

Aşağıdaki mantık devresinin çıkışını giriş değişkenleri, p, q ve r cinsinden belirleyin.



Çözüm

Soldan sağa doğru ilerleyerek, ilk olarak temel kapı çıkışlarını kullanarak en soldaki kapıların çıkışını belirleyin.



Mantık devresinin çıkışı $(p \vee q) \wedge (\neg p \vee \neg q)$

Sonraki iki örnekte, mantıksal önermelere dayalı mantık devreleri tasarlıyoruz. Buradaki fikir, sağdan sola işlem sırasını kullanarak geriye doğru çalışmaktır.

Örnek 26 - Bir Mantık Devresi Tasarlayın (p∨¬ q)∧¬p için bir mantık devresi tasarlayın.

Çözüm

Sağdan sola doğru geriye doğru çalışarak aşağıdaki kapı sırasına sahibiz

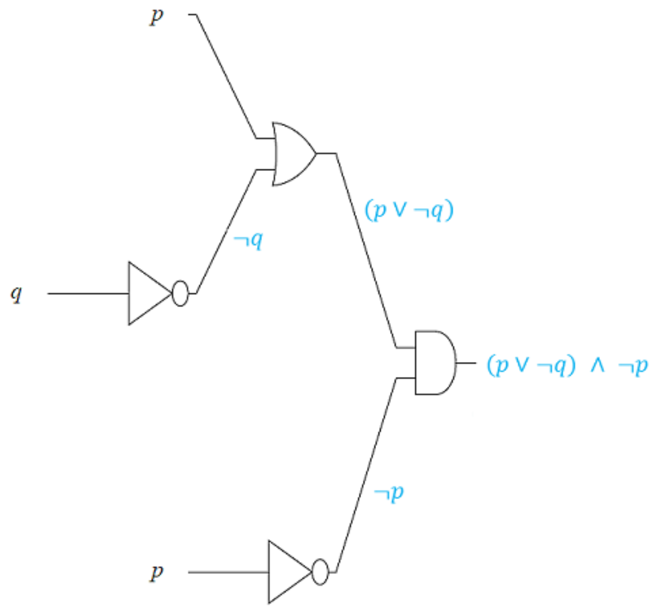
1) Bir AND geçidi (p∨¬ q)∧—¬p.

2) AND geçidinin girişleri (p∨¬ q) ve ¬p'dir.

3) Bu girişler, ¬— p ve bir VEYA geçidi (p∨—¬ q) için bir INVERTER çıkışından gelir.

4) VEYA geçidinin (p∨—¬ q) p olmak üzere iki girişi ve bir İNVERTERİN ¬—q çıkışı vardır.

Bunları şimdi soldan sağa sıralayarak aşağıdaki mantık devresini elde ederiz.



Örnek 27 - Bir Mantık Devresi Tasarlayın

r ∧ (p∨ (r∧¬q)) için bir mantık devresi tasarlayın.

Çözüm

Sağdan sola doğru geriye doğru çalışarak aşağıdaki kapı sırasına sahibiz

1) Bir VE kapısı r∧—(p∨(r∧¬q))).

2) AND geçidinin girişleri r ve p∨(r∧¬q)'dir.

3) Giriş, p∨(r∧¬q), p∨—(r∧¬q için bir VEYA geçidinin çıkışından gelir.

4) VEYA geçidinin girişleri, p∨—(r∧¬q), p ve (r∧¬q), bir AND geçididir.

5) AND , geçidinin r∧—¬q girişleri r'dir ve bir INVERTER ¬—q'nun çıkışıdır.

Bunları şimdi soldan sağa sıralayarak aşağıdaki mantık devresini elde ederiz.