Boole Cebri

George Boole (1815-1864) İngiliz Matematikçi

- B={0,1} kümesi üzerinde tanımlı
- İkili işlemler: VEYA, VE { + , • }
- Birli işlem: Tümleme (*complement*) { ' }
  Tümleme için diğer bir simge: $\overline{a}$

a • b

a \ b	0	1
0	0	0
1	0	1

a + b

a \ b	0	1
0	0	1
1	1	1

Tümleme

a	a' ($\overline{a}$)
0	1
1	0

- **Aksiyomlar:**

a , b ∈ B olmak üzere

1.Kapalılık (*Closure*): a + b ∈ B a • b ∈ B

2.Değişme (*Commutative*): a + b = b + a a • b = b • a

3.Birleşme (*Associative*): a + (b + c) = (a + b) + c a • (b • c) = (a • b) • c

4.Etkisiz eleman (*Identity*) : a + 0 = a a • 1 = a

5.Dağılma (*Distributive*): a + (b • c) = (a + b) • (a + c) a • (b + c) = (a • b) + (a • c)

6.Tümleme (*Inverse*): $a + \overline{a} = 1$ $a \bullet \overline{a} = 0$

- **İşlemler arasındaki öncelik** yüksekten öncelikten başlayarak şöyledir:
  1. Parantez, 2. Tümleme, 3. VE, 4. VEYA

- **İkilik (Düalite) Prensibi (*Duality principle*)**

Bir lojik ifadenin **düal**inin elde edilmesi: • yerine +, + yerine •, 0 yerine 1, 1 yerine 0, koyulur, ancak değişkenler değiştirilmez.

a + b + 0 ... ⇔ a • b • 1...

Örnek: a + a·b ifadesinin düali a·(a+b) ifadesidir.

İkilik (Düalite) Prensibi: Kanıtlanan her teorem düali için de geçerlidir.

Eğer bir denklemin (teoremin) doğru olduğu biliniyorsa onun düali de doğrudur.

Önceki yansıda yer alan aksiyomlarda düal ifadeler yan yana yazılmıştır.

Örnek:

Soğurma (*Absorption*) teoremi (sonraki sayfada verilecektir):

a + a·b = a kanıtlanırsa düali de doğrudur. a·(a+b) = a

Genelleştirilmiş düalite:

f (X1,X2,...,Xn,0,1,+,•) ⇔ f(X1,X2,...,Xn,1,0,•,+)

- Teoremlerin kanıtları arasında ilişki sağlar.
- Lojik ifadelerin dönüştürülmesini sağlayan bir yöntem değildir.
- Düal ifadeler birbirine eşit değildir.

Teoremler:

Burada gösterilen tüm teoremler Boole cebrinin tanımında yer alan işlemler ve aksiyomlar ile kanıtlanabilirler.

1. Yutma (*Annihilator*):
   a + 1 = 1 a • 0 = 0
2. Dönüşme (*Involution*):
   (a')' = a veya $\overline{\overline{a}} = a$
3. Sabit kuvvet (*Idempotency*):
   a+a+a+….+a = a a•a•a •… •a = a
4. Soğurma (*Absorption*):
   a + a·b = a (Kanıt 2.4'te) a·(a+b) = a
5. De Morgan Teoremi: *Augustus De Morgan (1806 – 1871)*
   $\overline{(a + b)} = \overline{a} \bullet \overline{b}$ $\overline{(a \bullet b)} = \overline{a} + \overline{b}$
5. Genel De Morgan Teoremi:
   $\overline{f(X1, X2 ,..., Xn, 0, 1, +, \bullet)} = f(\overline{X1}, \overline{X2}, ..., \overline{Xn}, 1, 0, \bullet, +)$
   - İkili işlemler (VE, VEYA) arasında ilişki sağlar: • ve + arasında

Teoremlerin Kanıtlanması:

a) Aksiyomlar ile

Örnek:

Teorem: $X \cdot Y + X \cdot \overline{Y} = X$

Kanıt:

Dağılma	$X \cdot Y + X \cdot \overline{Y}$	$= X \cdot (Y + \overline{Y})$
Tümleme	$X \cdot (Y + \overline{Y})$	$= X \cdot (1)$
Etkisiz	$X \cdot (1)$	$= X$ ✓

Örnek:

Teorem: X + X • Y = X Soğurma (*Absorption*)

Kanıt:

Etkisiz	X + X • Y	= X • 1 + X • Y
Dağılma		= X • (1 + Y)
Yutma		= X • (1)
Etkisiz		= X ✓

Teoremlerin Kanıtlanması: **b) Doğruluk Tablosu**

Tümleme (değil) (NOT) işleminin gösterilmesinde $\overline{A}$ simgesi de kullanılır.

De Morgan Teoreminin kanıtı:

$\overline{(X + Y)} = \overline{X} \bullet \overline{Y}$

X	Y	$\overline{X}$	$\overline{Y}$	$\overline{(X + Y)}$	$\overline{X} \bullet \overline{Y}$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

$\overline{(X \bullet Y)} = \overline{X} + \overline{Y}$

X	Y	$\overline{X}$	$\overline{Y}$	$\overline{(X \bullet Y)}$	$\overline{X} + \overline{Y}$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

Doğruluk tablolarında çok sayıda satır olsa da bunları bir bilgisayar programı yardımıyla kısa sürede sınamak mümkün olabilir.

Lojik ifadelerin aksiyom ve teoremler ile sadeleştirilmesi:

Bir lojik ifadenin minimize edilmesi;

- mümkün olduğu kadar az değişken ve işlem içeren,
- aynı girişler için orijinal ifade ile aynı çıkış değerlerini üreten,
- en kısa ifadeyi bulmak anlamına gelir.

Örnek:

Z = A' B C + A B' C + A B C' + A B C Orijinal ifade
= A' B C + A B' C + A B C' + A B C + A B C
= A' B C + A B' C + A B C' + A B C' + A B C
= (A' + A) B C + A B' C + A B C' + A B C
= (1) B C + A B' C + A B C' + A B C
= B C + A B' C + A B C' + A B C + A B C
= B C + A B' C + A B C + A B C' + A B C
= B C + A (B' + B) C + A B C' + A B C
= B C + A (1) C + A B C' + A B C
= B C + A C + A B (C' + C)
= B C + A C + A B (1)
= B C + A C + A B En sade (minimize edilmiş) ifade

Lojik İfadeler (*Expressions*)

Lojik ifade, değişkenlerin, sabitlerin ve işlemlerin kurallara uygun şekilde yazılmış sonlu kombinezonudur.

X= (x_1, x_2, ... x_n) ve her $x_i \in \{0,1\}$ olmak üzere ifade E(X) şeklinde gösterilir.

Örnekler:

$$\mathrm{E}(x_1, x_2, x_3, x_4) = x_2\overline{x_3} + x_1x_4 + \overline{x_1}x_2$$
$$\mathrm{E}(a, b, c) = a\bar{c} + ab$$
$$\mathrm{E}(a, b, c, d) = (a + b + \bar{c})(\bar{a} + d)(b + \bar{d})$$

Simge (*Literal*):

Bir lojik ifadede bir değişkenin kendisi veya tümleyeni şeklindeki her görüntüsüne **simge** (***literal***) denir.

Örneğin, $\mathrm{E}(a, b, c) = a\bar{c} + ab$ ifadesinin üç değişkeni (a, b, c) vardır ve ifade dört simgeden oluşmaktadır $(a\bar{c} + ab)$. İfadedeki iki simge aynıdır, çünkü a ifadede iki kere yer almaktadır.

E_1 ve E_2 lojik ifade ise, $\overline{E_1}$, $\overline{E_2}$, $E_1 + E_2$, $E_1 \cdot E_2$ gibi tüm kombinezonlar da birer lojik ifadedir.

Lojik İfadelerin Normal Biçimleri (Formları) :

Her lojik ifade iki özel biçimde (formda) yazılabilir.

1. **Lojik çarpımların lojik toplamı (ÇT)** Logical **s**um **o**f logical **p**roducts (**SOP**):
   Disjunctive normal form (DNF): $\Sigma\Pi$
   "VE"lerin "VEYA"lanması.
   Örnek: $b\bar{c} + ad + \bar{a}b$
2. **Lojik toplamların lojik çarpımı (TÇ)** Logical **p**roduct **o**f logical **s**ums (**POS**):
   Conjunctive normal form (CNF): $\Pi\Sigma$
   "VEYA"ların "VE"lenmesi
   Örnek: $(a + b + \bar{c})(a + d)(\bar{a} + b)$

Her lojik ifade ÇT ve TÇ biçiminde yazılabilir. Bir formda yazılan ifade diğer forma dönüştürülebilir ($\Sigma\Pi \leftrightarrow \Pi\Sigma$).

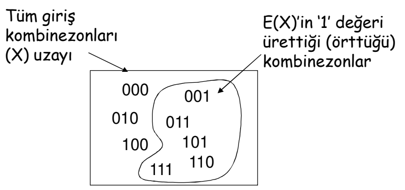
Bir lojik ifadenin değeri:

E(X) ifadesi X=(x_1, ... x_n) giriş vektörünün her değeri için B={0,1} kümesinden bir çıkış değeri üretir.

Bu değerler ifadenin doğruluk tablosunu oluşturur.

Örnek: E(X) = $x_1x_2 + x_3$ ifadesinin doğruluk tablosu

x_1	x_2	x_3	E(X)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Aksiyom ve teoremlerin ifadelere uygulanması

İkili değişkenler {0,1} için tanımlanmış olan aksiyom ve teoremler kapalılık özelliği nedeniyle ifadeler için de geçerlidirler.

Hatırlatma: Kapalılık (*closure*) aksiyomuna göre, E ifadesinin ürettiği değer ikili bir değerdir. E(X) ∈ B = {0,1}

Örnekler:

$$\mathrm{E(a, b, c)} = b\bar{c} + ad + \bar{a}b$$

Etkisiz Eleman: E(X) + 0 = E(X) E(X) • 1 = E(X)

$$\mathrm{E(a, b, c)} + 0$$
$$= (b\bar{c} + ad + \bar{a}b) + 0 = b\bar{c} + ad + \bar{a}b = \mathrm{E(a, b, c)}$$

$$\mathrm{E(a, b, c)} \cdot 1 = (b\bar{c} + ad + \bar{a}b) \cdot 1 = b\bar{c} + ad + \bar{a}b = \mathrm{E(a, b, c)}$$

Yutma: E(X) + 1 = 1 E(X) • 0 = 0

$$\mathrm{E(a, b, c)} + 1 = (b\bar{c} + ad + \bar{a}b) + 1 = 1$$

$$\mathrm{E(a, b, c)} \cdot 0 = (b\bar{c} + ad + \bar{a}b) \cdot 0 = 0$$

İkili (*binary*) değer vektörleri arasında sıra bağıntısı (*order relation*):

Lojik ifadelerin bazı özelliklerini ortaya koymak için iki sıra bağıntısı "< " ve "≤ " aşağıdaki gibi tanımlanır:

1. B={0,1} kümesinin elemanları arasında "< " sıra bağıntısı tanımlanır: 0 < 1

- 0, 1'den "önce gelir" ya da "küçüktür" diye okunur.

2. Buna göre X vektörleri arasında da " ≤ " sıra bağıntısı şöyle tanımlanabilir.

Eğer X1 vektörünün tüm elemanları X2 vektörünün aynı sıradaki elemanlarından yukarıda tanımlandığı anlamda "küçükse" (önce geliyorsa) ya da eşitse X1 ≤ X2 sıralaması geçerlidir.

Örnek:

X1=1001 , X2 = 1101 ise

X1 ≤ X2 dir.

İki vektör arasında sıra bağıntısı ≤ olmayabilir.

Örnek: X1=0011 , X2 = 1001 ise

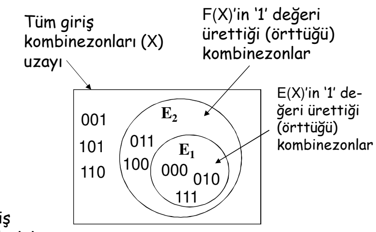
X1 ile X2 arasında sıra bağıntısı yoktur.
Ne X1 ≤ X2 ne de X2 ≤ X1 ilişkisi geçerli değildir.

İfadeler üzerinde tanımlı sıra bağıntısı:

E(X) ≤ F(X) yazılışı, X'in tüm kombinezonları için E'nin alacağı değerlerin aynı giriş kombinezonları için F'nin alacağı değerlere eşit ya da küçük olduğunu belirtir.

Örnek :

x_1	x_2	x_3	E(X)		F(X)
0	0	0	1	=	1
0	0	1	0	=	0
0	1	0	1	=	1
0	1	1	0	<	1
1	0	0	0	<	1
1	0	1	0	=	0
1	1	0	0	=	0
1	1	1	1	=	1



E(X)'in '1' değerini aldığı her giriş kombinezonu için F(X) de '1' değerini alıyor. E(X) ≤ F(X)

Bu özel bir durumdur. ≤ sıra bağlantısı tüm ifadeler arasında geçerli değildir (bkz. yansı 2.14).

E(X) ≤ F(X) ise
E(X), F(X)'i gerektirir, E(X)⇒F(X),
F(X), E(X)'i örter.

İfadeler arası soğurma özelliklerinin gösterilmesi için sıra bağıntınsın kullanılması:

Hatırlatma: Yansı 2.3'te ikili değerler için verilen teoremler kapalılık özelliği nedeniyle ifadeler için de geçerlidirler.

Sonuç olarak, ikili değerler için tanımlanmış olan soğurma (*absorption*) teoremi de (a + a·b = a ve a·(a+b) = a) kapalılık özelliği nedeniyle ifadeler için de geçerlidir.

Bununla beraber, ≤ sıra bağıntısı ifadelerin soğurma özelliklerini anlamayı kolaylaştırmaktadır.

Soğurma, ifadeleri sadeleştirmekte kullanılan önemli bir teorem olduğundan, ≤ sıra bağıntısını kullanarak bu teorem ifadeler üzerinde tekrar gösterilecektir.

Aşağıdaki iki durum ele alınacaktır.

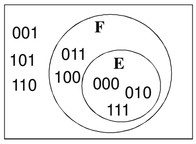
- A) Özel durum: E(X) ≤ F(X) (yansı 2.11'teki örnek gibi)
- B) Genel durum: Sıra bağıntısı ≤ E(X) ve F(X) ifadeleri arasında geçerli değildir.

İfadeler arası soğurma özelliklerinin gösterilmesi için sıra bağıntınsın kullanılması:

A) Özel durum: E(X) ≤ F(X)

Yansı 2.11'de verilen örneği dikkate alınız.

Soğurma özellikleri sağdaki diyagramda görülmektedir :



E(X) ≤ F(X) olduğundan,

1. E(X) + F(X) = F(X)
2. E(X) • F(X) = E(X)

Yansı 2.11'de gösterilen fonksiyonların ifadeleri:

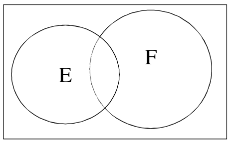
$\mathrm{E}(x_1,x_2,x_3) = \overline{x_1}\cdot\overline{x_3} + x_1\cdot x_2\cdot x_3$, $\mathrm{F}(x_1,x_2,x_3) = \overline{x_1}\cdot\overline{x_3} + \overline{x_2}\cdot\overline{x_3} + x_2\cdot x_3$

Soğurma özellikleri:

1. E(X) + F(X) = $\overline{x_1}\cdot\overline{x_3} + x_1\cdot x_2\cdot x_3 + \overline{x_1}\cdot\overline{x_3} + \overline{x_2}\cdot\overline{x_3} + x_2\cdot x_3$
   = $\overline{x_1}\cdot\overline{x_3} + \overline{x_2}\cdot\overline{x_3} + x_2\cdot x_3$ = F(X)

2. E(X) • F(X) = $(\overline{x_1}\cdot\overline{x_3} + x_1\cdot x_2\cdot x_3)\cdot(\overline{x_1}\cdot\overline{x_3} + \overline{x_2}\cdot\overline{x_3} + x_2\cdot x_3)$
   = $\overline{x_1}\cdot\overline{x_3} + +x_1\cdot x_2\cdot x_3$= E(X)

B) Genel durum: E(X) ve F(X) arasında sıra bağıntısı (≤) geçerli değildir.



E ve F lojik ifadeler olmak üzere, aşağıdaki eşitsizlikleri her zaman geçerlidir (soldaki diyagramı inceleyiniz):

E·F ≤ E ≤ E+F ve

E·F ≤ F ≤ E+F

Soğurma özellikleri:

$\boxed{E + E\cdot F = E}$ ve düali $\boxed{E(E+F) = E}$

Kanıt: E(E+F) = EE+EF = E+EF = E(1+F) = E

$\boxed{E + \bar{E}\cdot F = E + F}$ ve düali $\boxed{E(\bar{E}+F) = E\cdot F}$

Kanıt: $E + \bar{E}\cdot F = (E+\bar{E})(E+F) = 1(E+F) = E+F$

Bu özellikler lojik ifadelerin sadeleştirilmesinde kullanılır.

Örnek (genel durum):

E(a,b,c,d) = abc′ , F(a,b,c,d) = bd

E·F = abc′d E+F = abc′ + bd

a b c d	E	F	E·F	E+F
0 0 0 0	0	0	0	0
0 0 0 1	0	0	0	0
0 0 1 0	0	0	0	0
0 0 1 1	0	0	0	0
0 1 0 0	0	0	0	0
0 1 0 1	0	1	0	1
0 1 1 0	0	0	0	0
0 1 1 1	0	1	0	1
1 0 0 0	0	0	0	0
1 0 0 1	0	0	0	0
1 0 1 0	0	0	0	0
1 0 1 1	0	0	0	0
1 1 0 0	1	0	0	1
1 1 0 1	1	1	1	1
1 1 1 0	0	0	0	0
1 1 1 1	0	1	0	1

E ile F arasında sıra bağıntısı yoktur.

Önceki yansıdan bildiğimiz gibi ve doğruluk tablosundan da görüldüğü gibi aşağıdaki eşitsizlikler geçerlidir:

E·F < E ve E·F < F.

Bu nednele

E·F + E = E

~~abc′d~~ + abc′ = abc′

ve

E·F + F = F

~~abc′d~~ + bd = bd

E < E+F ve F < E+F.

Bu nednele

E·(E + F) = E

abc′ (~~abc′ + bd~~) = abc′

ve

F·(E + F) = F

bd (~~abc′ + bd~~) = bd

Konsensüs Teoremi (ÇT Formu)

E_1 ve E_2 içinde x_1 simgesi olmayan iki ifade olsun.

Bu ifadelerden birini x_1, diğerini de x_1'in tümleyeni ($\overline{x_1}$) ile "VE"leyerek yeni bir ifade oluşturabiliriz.

$$E = \boldsymbol{x_1}E_1 + \overline{\boldsymbol{x_1}}E_2$$

Burada, x_1 **iki biçimli** (*biform*) değişken olarak adlandırılır, çünkü ifade de hem kendisi ($\boldsymbol{x_1}$) hem de tümleyeni ($\overline{\boldsymbol{x_1}}$) yer almaktadır.

Örnekler: $\boldsymbol{x_1}(x_2+\overline{x_3}) + \overline{\boldsymbol{x_1}}(x_3+x_4)$,
$\boldsymbol{x_1}x_2\overline{x_3} + \overline{\boldsymbol{x_1}}x_4 + x_5$

- İki biçimli bir değişken içeren iki ifadenin **konsensüs terimini** (ÇT formu) oluşturmak için bu iki ifade birbiriyle çarpılır ("VE"lenir), seçilen değişken ve tümleyeni dışarıda bırakılır.
- **E_1E_2** çarpımı, $x_1E_1 + \overline{x_1}E_2$ ifadesinin x değişkenine göre **konsensüs terimidir**.

Örnek: $\boldsymbol{abc} + \boldsymbol{\bar{a}cd}$ ifadesinin *a*'ya göre konsensüsü $bccd = \boldsymbol{bcd}$.

Teorem: Konsensüs terimi asıl ifadenin yanında fazlalıktır; ifade tarafından yutulur.

$$\boxed{x_1E_1 + \overline{x_1}E_2 + \boldsymbol{E_1E_2} = x_1E_1 + \overline{x_1}E_2}$$

Konsensüs Teoremi (TÇ Formu)

İkilik prensibine (*duality principle*) göre, konsensüs teoremi TÇ formunda yazılan ifadeler için de geçerlidir

E_1 ve E_2 içinde x_1 simgesi olmayan iki ifade olsun:

Bu ifadelerden birini x_1, diğerini de x_1'in tümleyeni ($\overline{x_1}$) ile "VEYA"layerek yeni bir ifade oluşturabiliriz.

$$E = (\boldsymbol{x_1}+E_1)(\overline{\boldsymbol{x_1}}+E_2)$$

Burada, x_1 **iki biçimli** (*biform*) bir değişkendir.

Örnekler: $(\boldsymbol{x_1} + x_2 + \overline{x_3})(\overline{\boldsymbol{x_1}} + x_3 + x_4)$,
$(\boldsymbol{x_1} + x_2\overline{x_3})(\overline{\boldsymbol{x_1}} + x_3\, x_4)$

- İki biçimli bir değişken içeren iki ifadenin **konsensüs terimini** (TÇ formu) oluşturmak için bu iki ifade birbiriyle toplanır ("VEYA"lanır), seçilen değişken ve tümleyeni dışarıda bırakılır.
- **$E_1 + E_2$** ifadesi, $(x_1+E_1)(\overline{x_1}+E_2)$ ifadesinin x değişkenine göre **konsensüs terimidir**.

Örnek: $(\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c})(\boldsymbol{\bar{a}}+\boldsymbol{c}+\boldsymbol{d})$ ifadesinin konsensüsü: $b + c + c + d = \boldsymbol{b} + \boldsymbol{c} + \boldsymbol{d}$.

Teorem: Konsensüs terimi asıl ifadenin yanında fazlalıktır; ifade tarafından yutulur.

$$\boxed{(x_1+E_1)(\overline{x_1}+E_2)(\boldsymbol{E_1} + \boldsymbol{E_2}) = (x_1+E_1)(\overline{x_1}+E_2)}$$

Örnek: Konsensüs teoremi ile lojik ifadelerin indirgenmesi

F(A, B, C) = A'B'C + A'BC + AB'C + ABC + ABC'

C ye göre konsensüs eklendi

= A'B'C + A'BC + AB'C + ABC + ABC' + AB

= A'B'C + A'BC + AB'C + ~~ABC~~ + ~~ABC'~~ + AB Soğurma, yutma (*Absorption*)

= A'B'C + A'BC + AB'C + AB

B ye göre konsensüs eklendi

= A'B'C + A'BC + A'C + AB'C + AB

= A'~~B'~~C + A'~~B~~C + A'C + AB'C + AB Soğurma, yutma (*Absorption*)

= A'C + AB'C + AB

B ye göre konsensüs eklendi

= A'C + A~~B'~~C + AB + AC Soğurma (*Absorption*)

= A'C + AB + AC

A ya göre konsensüs eklendi

= ~~A'C~~ + AB + ~~AC~~ + C Soğurma (*Absorption*)

= AB + C

Lojik Fonksiyonlar (*Boolean Functions*)

Lojik fonksiyonlar B^n kümesi (n elemanlı 2'li kodların kümesi) üzerinde tanımlanırlar ve üçe ayrılırlar:

1. Yalın fonksiyonlar: *Çok girişli bir çıkışlı*

$\forall X^0 \in B^n$; $\exists! \, y^0 \in B$; $y=f(X)$

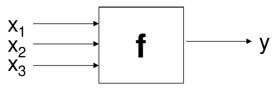
B^n kümesinden değer alan X^0 kombinezonuna f fonksiyonu uygulandığında B kümesinden değer alan bir y^0 değeri elde edilir ve bu değer tektir.

Örnek:

y = f(X)

$X \in B^3$

$y \in B$

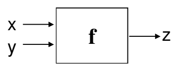


y = f(X) fonksiyonuna ilişkin doğruluk tablosu:

x_1 x_2 x_3	y
0 0 0	1
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

Yalın Lojik Fonksiyonlar (devam):

n girişli $2^{(2^n)}$ adet yalın lojik fonksiyon vardır.
İki girişli 16 adet yalın lojik fonksiyon vardır:



2 girişli 16 adet yalın lojik fonksiyon (F0–F15)

Girişler X	Y	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Fonksiyonlar: 0 ; X VE Y ; X ; Y ; X YA DA Y ; X VEYA Y ; X TVEYA Y (X VEYA Y)' ; X = Y ; Y' ; X' ; X TVE Y (X VE Y)' ; 1

2. Genel fonksiyonlar: *Çok girişli, çok çıkışlı*

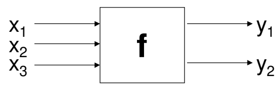
$Y = f(X): B^n \rightarrow B^m$, $X=(x_1, ... x_n)$, $Y=(y_1, ... y_m)$,

Örnek:

Y = f(X)

$X \in B^3$

$Y \in B^2$



Y = f(X) fonksiyonuna ilişkin doğruluk tablosu:

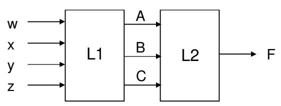
x_1 x_2 x_3	y_1 y_2
0 0 0	1 1
0 0 1	1 0
0 1 0	0 0
0 1 1	0 0
1 0 0	1 1
1 0 1	0 1
1 1 0	0 1
1 1 1	1 0

3. Tümüyle tanımlanmamış fonksiyonlar (*Incompletely specified functions*):

Büyük sayısal sistemler daha küçük alt lojik devrelerden oluşurlar.

Örnek:

Aşağıdaki örnekte L1 lojik devresinin çıkışları L2 lojik devresinin girişlerini sürmektedir.



- L1 devresinin A, B ve C çıkışları için tüm olası ikili değer kombinasyonlarını üretmediğini varsayalım.
- Örneğin, w, x, y ve z girişlerine uygulanabilecek hiçbir değer için A, B ve C çıkışlarının 001 ve 110 değerlerini almadığını varsayalım.
- Diğer bir deyişle, L1 hiçbir zaman 001 ve 110 çıkış değerlerini üretmemektedir.
- Bu durumda, L2 devresi tasarlanırken ABC = 001 ve 110 giriş değerleri için F çıkışının alacağı değerleri belirlemeye (dikkate almaya) gerek yoktur, çünkü 001 ve 110 değerleri hiçbir zaman L2 devresinin girişlerine uygulanmayacaktır.

3. Tümüyle tanımlanmamış fonksiyonlar (devamı):

- Örneğin, F'nin doğruluk tablosu aşağıdaki gibi olabilir:

A	B	C	F
0	0	0	1
0	0	1	X (belirsiz)
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	X (belirsiz)
1	1	1	1

- Tablodaki X'ler, ABC = 001 ve 110 giriş değerleri için F'nin üreteceği çıkış değerinin önemli olmadığını gösterir. Bu çıkış değerleri **belirsizdir**.
- Bu nedenle F fonksiyonu **tümüyle tanımlanmamıştır** (*incompletely specified*).
- A'B'C ve ABC' ifadeleri **önemsiz** (*don't care*) olarak adlandırılır çünkü bu ifadelerin fonksiyonda yer alıp almaması önemli değildir.

3. Tümüyle tanımlanmamış fonksiyonlar (devamı):

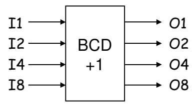
- Fonksiyonu devre ile gerçekleştirmek için belirsiz çıkışlara bir değer atamak gereklidir.
- Belirsiz çıkışın değerine karar verirken fonksiyonu basitleştirmeyi sağlayan değerler seçilir. Örneğin yansı 2.23'teki fonksiyonu inceleyelim:
  - Her iki belirsiz X'e 0 değeri atanırsa
    F = A'B'C' + A'BC + ABC = A'B'C' + BC
  - Birinci X'e 0, ikinci X'e 1 değeri atanırsa
    F = A'B'C' + A'BC + ABC'+ ABC = A'B'C' + BC + AB
  - Birinci X'e 1, ikinci X'e 0 değeri atanırsa
    F = A'B'C' + A'B'C + A'BC + ABC = A'B' + BC
  - Her iki belirsiz X'e 1 değeri atanırsa
    F = A'B'C' + A'B'C + A'BC + ABC'+ ABC = A'B' + BC + AB

Üçüncü seçenek en basit ifadenin oluşmasını sağlamıştır.

- 4. Bölümde belirsiz değerlerin seçilmesi ve tümüyle tanımlanmamış fonksiyonların sadeleştirilmesi ayrıntılı olarak ele alınacaktır.
- Tümüyle tanımlanmamış fonksiyonlar aşağıdaki durumlarda oluşurlar:
  - Bazı giriş değerlerinin oluşması mümkün değildir.
  - Tüm giriş değerleri oluşsa bile bazı değerler için o fonksiyonun üreteceği çıkış değerleri önemli değildir.

3. Tümüyle tanımlanmamış fonksiyonlar (devamı):

Örnek: BCD sayıları 1 arttıran fonksiyon:
Yansı 1.9'da gösterilen BCD sayıları artırmak için bir genel fonksiyon tasarlanacaktır.



Her bir BCD sayı 4 bit uzunluğunda olduğundan bu fonksiyonun 4 bit girişi ve 4 bit çıkışı olacaktır.

BCD sayılar 0000-1001 arasındaki ikili kod sözcükleri ile temsil edildiklerinden, bu fonksiyonun girişlerine 1010-1111 arasındaki değerler hiçbir zaman uygulanmayacaktır.

Bu değerler fonksiyonun girişine uygulansa bile fonksiyonun üreteceği çıkış değerleri önemli değildir.

I8	I4	I2	I1	O8	O4	O2	O1
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Bu girişler için devrenin (fonksiyonun) çıkışlarının alacağı değer belirsizdir.
Belirsiz değerleri göstermek için X yerine Φ sembolü de kullanılır.

Lojik Fonksiyonların Gösterilişi

Aynı lojik fonksiyon farklı yöntemler ile gösterilebilir.

Bu fonksiyona ilişkin devre tasarlanırken bu gösterilimlerden uygun olanı kullanılır.

Doğruluk Tablosu (*Truth Table*) İle Gösterilim

Tüm giriş kombinezonları için çıkışın (veya çıkışların) alacağı değerler tablo halinde yazılır.

Giriş değişkenleri ikili sayma sırasına göre sıralanırlar (0, 1, 2, ...).

(Bkz. Örnek tablolar 2.20 - 2.22)

Sayısal (*Indexed*) Gösterilim

Giriş kombinezonları 2'li sayılarla kodlandığına göre her kombinezona 10 tabanında bir numara verilir.

Fonksiyon hangi giriş kombinezonları için lojik "1" değeri (ya da lojik "0", "Φ") üretiyorsa o kombinezonların numaraları listelenir.

Örnek: Tümüyle tanımlanmış, yalın bir fonksiyonun sayısal gösterilimi:

Doğruluk Tablosu:

Satır No	Giriş x_1	Giriş x_2	Çıkış y
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	0

Sayısal (*indexed*) gösterim:

$y = f(x_1,x_2)= \cup_1(0,2)$

$\cup$: Birleşme (*union*) veya "kümesidir"
$\cup_1$, "1" üreten girişler kümesidir.

Değişkenlerin sırası önemlidir. Doğruluk tablosundaki sıraya dikkat edilmelidir. Aksi durumda kombinezon numaraları değişecektir.

Aynı fonksiyon. Sadece değişkenlerin sırası değiştirilmiştir (x_2,x_1).

Satır No	Giriş x_2	Giriş x_1	Çıkış y
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	0

$y = f(\mathbf{x_2},\mathbf{x_1}) = \cup_1(0,1)$

Aynı fonksiyon lojik 0 üreten kombinezonlar ile de gösterilebilir.

$y = f(x_1,x_2) = \cup_0(1,3)$

Örnek: Tümüyle tanımlanmış, genel bir fonksiyonun gösterilimi:
Her çıkış için sayısal gösterilim uygulanır.

Doğruluk Tablosu:

No	x_1	x_2	y_1	y_2
0	0	0	1	1
1	0	1	0	1
2	1	0	1	0
3	1	1	0	0

Sayısal (*indexed*) gösterim:

$y_1=f(x_1,x_2)= \cup_1(0,2)$

$y_2=f(x_1,x_2)= \cup_1(0,1)$

Aynı fonksiyon lojik 0 üreten kombinezonlar ile de gösterilebilir.

$y_1=f(x_1,x_2)= \cup_0(1,3)$
$y_2=f(x_1,x_2)= \cup_0(2,3)$

Örnek: Tümüyle tanımlanmamış, genel bir fonksiyonun gösterilimi:

Bu durumda sadece lojik "1" veya lojik "0" üreten çıkışları göstermek yeterli değildir.
Üç gruptan ("0" üreten, "1" üreten, belirsiz) en az ikisini yazmak gerekir.

Doğruluk Tablosu:

No	x_1	x_2	y_1	y_2
0	0	0	1	1
1	0	1	0	Φ
2	1	0	Φ	0
3	1	1	0	Φ

Sayısal (*indexed*) gösterim:

$y_1 = f(x_1,x_2) = \cup_1(0) + \cup_0(1,3)$
veya $y_1 = f(x_1,x_2) = \cup_1(0) + \cup_\Phi(2)$
veya $y_1 = f(x_1,x_2) = \cup_0(1,3) + \cup_\Phi(2)$

$y_2 = f(x_1,x_2) = \cup_1(0) + \cup_0(2)$
veya $y_2 = f(x_1,x_2) = \cup_1(0) + \cup_\Phi(1,3)$
veya $y_2 = f(x_1,x_2) = \cup_0(2) + \cup_\Phi(1,3)$

Grafik Gösterilim

Bir lojik fonksiyonun girişi kombinezonları B^n kümesinin elemanları olduklarına göre n boyutlu uzaydaki bir çok boyutlu küpün (*hypercube*) köşelerini oluştururlar.

Örnek: B^3 = {000, 001, …, 111}

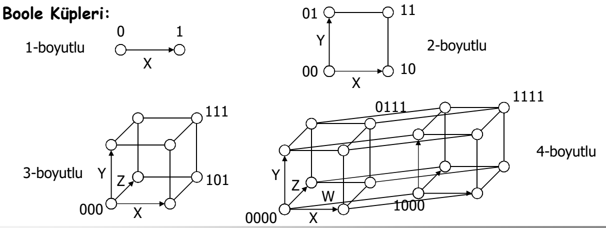
Bu değişkenleri bir n boyutlu küpün köşeleri olarak temsil edebiliriz.

Fonksiyonun (lojik 1) üreten kombinezonları küp üzerinde işaretlenir (boyanır).

Fonksiyonun giriş sayısı küpün boyutunu belirler.

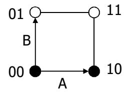
n giriş → n boyutlu küp

Boole Küpleri:



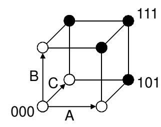
Örnek:
F(A,B)

A	B	F
0	0	1
0	1	0
1	0	1
1	1	0



Örnek:
F(A,B,C)

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Giriş sayısı arttıkça çizimin zorlaşması nedeniyle, Boole küpleri lojik fonksiyonların gösterilmesi için pratikte kullanılan bir yöntem değildir.

Bunula beraber, grafik gösterilim, lojik fonksiyonların bazı özelliklerinin (örneğin kombinezonların bitişikliği) görsel olarak anlaşılmasını ve bundan sonraki konuların anlatılmasını kolaylaştırmaktadır.

Karnaugh Diyagramları (*Karnaugh Maps*)

Maurice Karnaugh (1924-), ABD, fizikçi

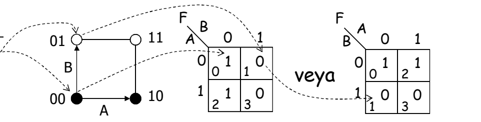
Karnaugh diyagramları lojik fonksiyonları göstermek ve basitleştirmek için kullanılan görsel araçlardır.

Lojik değişkenler ve ürettikleri çıkış değerleri bir doğruluk tablosundan matris şeklindeki Karnaugh diyagramına taşınabilir.

Örnek: F(A,B)

Doğruluk tablosu: Bool Küpü: Karnaugh diyagramları:

Satır No	A	B	F
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	0



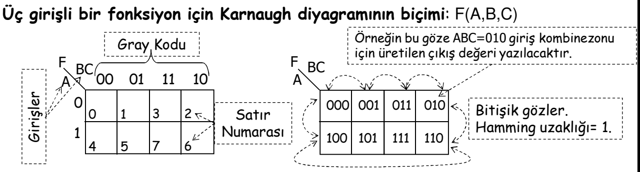
A: satır, B: sütun A: sütun, B: satır

Farklı değişkenler satırlara veya sütunlara yerleştirilebilir.

Karnaugh Diyagramları (*Karnaugh Maps*) (devamı)

Tabloların satır ve sütunları **Gray** koduna göre düzenlenir. Yan yana (ve alt alta) gözlere ait kombinezonların bitişik olması sağlanır.

Üç girişli bir fonksiyon için Karnaugh diyagramının biçimi: F(A,B,C)

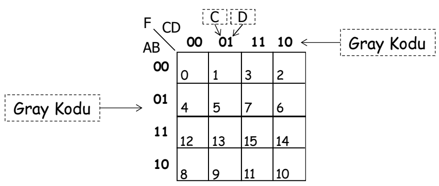


Örnek:

Num	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

A \ BC	00	01	11	10
0	0 0	1 0	3 1	2 0
1	4 0	5 1	7 1	6 1

4 girişli bir fonksiyona ilişkin Karnaugh diyagramının biçimi: F(A,B,C,D)



Örnek:

Aşağıdaki fonksiyonun Karnaugh diyagramını çiziniz.

F(A,B,C,D) = $\cup_1$ (0,2,5,8,9,10,11,12,13,14,15)

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	1	1
10	1	1	1	1

Karnaugh diyagramları dersin ilerleyen bölümlerinde fonksiyonların indirgenmesinde kullanılacaktır.

Cebirsel Gösterilim (İfadeler) ve Kanonik Açılımlar

Gerçek dünyadaki bir problemin çözümü doğruluk tablosu ile ifade edilebilir.

Örneğin; giriş değişkeni **A** bir aracın kapısının açık olduğunu, **B** anahtarın yuvaya takılı olduğunu ifade ederse, alarmın çalıp çalmadığı gösteren **Z** çıkışını (Z=1 ise alarm çalıyor) içeren doğruluk tablosu aşağıdaki gibi oluşturulabilir.

Num.	A	B	Z
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

Gerçek dünyadaki problemler çok daha fazla girişe sahip olduklarından doğruluk tabloları da daha karmaşıktır.

Bu problemlerin çözümlerini basitleştirmek ve ilgili devreleri lojik kaplılar ile gerçeklemek için fonksiyonların **cebirsel ifadelerini** (*expression*) bulmak gerekir.

Lojik fonksiyonların ifadeleri doğruluk tablolarından **kanonik açılımlar** şeklinde elde edilir.

İki tür kanonik açılım vardır:

- **1. kanonik açılım** : Çarpımların toplamı (ÇT) ($\Sigma\Pi$) Örnek: abc + ab'c
  "1" çıkışı üreten giriş kombinezonlarının çarpımlarının toplamından oluşur.
- **2. kanonik açılım** : Toplamların çarpımı (TÇ) ($\Pi\Sigma$) Örnek: (a+b+c') (a+b'+c')
  "0" çıkışı üreten giriş kombinezonlarının toplamlarının çarpımından oluşur..

Birinci Kanonik Açılım (1. Standart Biçim): Çarpımların Toplamı (ÇT)
(***1st Canonical (Standard) Form:*** *Sum of Products*)

- Birinci kanonik açılım, fonksiyonun "doğru" (1 çıkışı üreten) noktalarına ilişkin çarpımların (**minterim**) toplamından oluşur
- **Minterim (*minterm*)** n Değişkenli bir fonksiyonda n değişkenin <u>hepsini</u> sadece <u>bir defa</u> (ya kendisi ya da tümleyeni şeklinde) içeren çarpım ifadelerine **minterim** denir.
- Örneğin 3 değişkenli (a, b, c) bir fonksiyonun 8 adet minterimi vardır:
  a'b'c', a'b'c, a'bc', a'bc, ab'c', ab'c, abc', abc
- Her minterim doğruluk tablosunda sadece bir "doğru" (çıkış=1 olan) satırı örter.
  Örneğin; a'b'c' minterimi sadece abc=000 giriş kombinezonu için "1" değeri üretir.
  Diğer bütün giriş kombinezonları için a'b'c' minterimi "0" değeri üretir.
- Fonksiyonun 1. kanonik açılımı minterimlerin toplamından oluşur.

Minterimlerin bulunması:

Doğruluk Tablosu → Çarpımların Toplamı (ÇT) şeklinde ifade

- Doğruluk tablosunda çıkışın "1" olduğu satırlar seçilir.
- Bu satırlarda girişlerin "1" olduğu yerlere değişkenlerin kendileri (örneğin A, B, C) ve "0" olduğu yerlere tümleyenleri (A', B', C') yazılarak çarpımlar (minterimler) oluşturulur.

Örnek:

"Doğru" değer (1) üreten kombinezonlar: F = 001 011 101 110 111
Minterimlerin Toplamı: F = A'B'C + A'BC + AB'C + ABC' + ABC

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Fonksiyonun tümleyeninin 1. kanonik açılımı:

Tümleyen fonksiyonun 1. kanonik açılımı "yanlış" (çıkış=0) noktalardan hareket edilerek bulunur.

Örnek:

Önceki örnekte verilen F fonksiyonunun tümleyeninin 1. kanonik açılımını yazınız.

$\overline{\boldsymbol{F}}$ fonksiyonunun 1. kanonik açılımı:

$$\overline{F(A,B,C)} = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$$

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Birinci kanonik biçimdeki ifadenin sadeleştirilmesi:

Kanonik açılım çoğunlukla fonksiyonun en basit cebirsel ifadesi değildir.
Çoğunlukla kanonik açılımlar yalınlaştırılabilir (basitleştirilebilir).

Basitleştirme:

F(A, B, C) = A'B'C + A'BC + AB'C + ABC + ABC'
= (A'B' + A'B + AB' + AB)C + ABC'
= ((A' + A)(B' + B))C + ABC'
= C + ABC'
= ABC' + C
= AB + C

- Bir lojik (*Boolean*) fonksiyon birden fazla lojik ifade ile gösterilebilir. Bu ifadelerin hepsi aynı giriş değerleri için aynı çıkış değerlerini üretirler.
- Birinci kanonik (standart) biçimdeki ifade de yer alan her minterim, doğruluk tablosunda çıkışı 1 olan bir satıra karşı düşer.
- Bu nedenle bir lojik fonksiyonun birinci kanonik (standart) biçimdeki ifadesi <u>tektir</u> ve ona özeldir.
- Diğer bir söyleyişle, bir lojik fonksiyonun birinci kanonik (standart) biçimde sadece bir ifadesi vardır.

Minterimlerin numaralanması:

Minterimler giriş kombinezonlarının sıraları dikkate alınarak numaralandırılırlar.
Minterimin doğruluk tablosunda karşı geldiği satırın on tabanındaki numarası kullanılır.
Örneğin, ab'c (101) minterimine sıra numarası olarak 5 atanır ve m5 olarak adlandırılır.

Girişler:

A	B	C	minterimler	
0	0	0	A'B'C'	m0
0	0	1	A'B'C	m1
0	1	0	A'BC'	m2
0	1	1	A'BC	m3
1	0	0	AB'C'	m4
1	0	1	AB'C	m5
1	1	0	ABC'	m6
1	1	1	ABC	m7

3 değişkenli minterimlerin simgesel gösterilimi

Örnek:

Yansı 2.38'deki F fonksiyonun 1. kanonik açılımı:

F(A, B, C) = Σm(1,3,5,6,7)
= m1 + m3 + m5 + m6 + m7
= A'B'C + A'BC + AB'C + ABC' + ABC

F = $\Sigma_{A,B,C}$(1,3,5,6,7) (çarpımların toplamı)

İkinci Kanonik Açılım: Toplamların Çarpımı (TÇ)
(***2nd Canonical (Standard) Form:*** *Product of Sums*)

- İkinci kanonik açılım, fonksiyonun "yanlış" (0 çıkışı üreten) noktalarına ilişkin **maksterim** adı verilen özel toplamların çarpımlarından oluşur
- **Maksterim (*maxterm*):** n Değişkenli bir fonksiyonda n değişkenin <u>hepsini</u> sadece <u>bir defa</u> (ya kendisi ya da tümleyeni şeklinde) içeren toplama ifadelerine **maksterim** denir.
- Örneğin 3 değişkenli (a, b, c) bir fonksiyonun 8 adet maksterimi vardır:
  a+b+c, a+b+c', a+b'+c, a+b'+c', a'+b+c, a'+b+c', a'+b'+c, a'+b'+c'
- Her maksterim doğruluk tablosundaki **sade bir** giriş kombinezonu için 0 değerini alır.
  Örneğin; a+b+c maksterimi sadece abc=000 giriş kombinezonu için "0" değeri üretir.
  Diğer bütün giriş kombinezonları için a+b+c maksterimi "1" değeri üretir.
- Fonksiyonun 2. kanonik açılımı maksterimlerin çarpımlarından oluşur.

Maksterimlerin bulunması:

Doğruluk Tablosu → Toplamların Çarpımı (TÇ) şeklinde ifade

- Doğruluk tablosunda çıkışın "0" olduğu satırlar seçilir.
- Bu satırlarda girişlerin "0" olduğu yerlere değişkenlerin kendileri (örneğin A, B, C) ve "1" olduğu yerlere tümleyenleri (A', B', C') yazılarak toplamlar (maksterimler) oluşturulur.

Örnek:

"Yanlış" değer (0) üreten kombinezonlar: F = *000* *010* *100*

Maksterimlerin Çarpımı: F = (A + B + C) (A + B' + C) (A' + B + C)

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Bu örnekteki F fonksiyonu ile yansı 2.38'deki fonksiyonun doğruluk tabloları aynıdır. Bu nedenle bu örnekte F'nin birinci ve ikinci kanonik biçimdeki ifadeleri aynı doğruluk tablosunu oluştururlar.

Fonksiyonun tümleyeninin 2. kanonik açılımı:

Fonksiyonun tümleyeninin 2. kanonik açılımı benzer şekilde "doğru" (çıkış=1) noktalardan hareket edilerek yazılır:

Örnek:

Önceki örnekte verilen F fonksiyonunun tümleyeninin 2. kanonik açılımını yazınız.

$\bar{F}$ fonksiyonunun 2. kanonik açılımı:

F'(A,B,C) = (A + B + C') (A + B' + C') (A' + B + C') (A' + B' + C) (A' + B' + C')

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

İkinci Kanonik biçimdeki ifadenin sadeleştirilmesi:

Kanonik açılım çoğunlukla fonksiyonun en basit cebirsel ifadesi değildir.

Çoğunlukla kanonik açılımlar yalınlaştırılabilir (basitleştirilebilir).

Basitleştirme:

F(A, B, C) = (A+B+C) (A+B'+C) (A'+B+C)
= ((A+C)+(B·B')) (A'+B+C)
= (A+C) (A'+B+C)
= (A+C) (A'+B+C) (B+C) (konsensüs)
= (A + C) (B + C)

- Bir lojik (*Boolean*) fonksiyon birden fazla lojik ifade ile gösterilebilir. Bu ifadelerin hepsi aynı giriş değerleri için aynı çıkış değerlerini üretirler.
- İkinci kanonik (standart) biçimdeki ifade de yer alan her maksterim, doğruluk tablosunda çıkışı 0 olan bir satıra karşı düşer.
- Bu nedenle bir lojik fonksiyonun ikinci kanonik (standart) biçimdeki ifadesi tektir ve ona özeldir.
- Diğer bir söyleyişle, bir lojik fonksiyonun ikinci kanonik (standart) biçimde sadece bir ifadesi vardır.

Maksterimlerin numaralanması:

Maksterimler giriş kombinezonlarının sıraları dikkate alınarak numaralandırılırlar.

Maksterimin doğruluk tablosunda karşı geldiği satırın on tabanındaki numarası kullanılır.

Örneğin, a'+b+c' (101) maksterimine sıra numarası olarak 5 atanır ve M5 olarak adlandırılır.

Girişler:

A	B	C	maksterimler	
0	0	0	A+B+C	M0
0	0	1	A+B+C'	M1
0	1	0	A+B'+C	M2
0	1	1	A+B'+C'	M3
1	0	0	A'+B+C	M4
1	0	1	A'+B+C'	M5
1	1	0	A'+B'+C	M6
1	1	1	A'+B'+C'	M7

3 değişkenli maksterimlerin simgesel gösterilimi

Örnek: 2.43'teki F nin ikinci kanonik açılımı:

F(A, B, C) = ΠM(0,2,4)
= M0 • M2 • M4
= (A + B + C) (A + B' + C) (A' + B + C)

F = $\Pi_{A,B,C}$(0,2,4) şeklinde de yazılabilir.

Kanonik Biçimler Arasında Dönüşüm

- 1. kanonik açılımdan 2. kanonik açılıma (minterimden maksterime) dönüşüm:
  1. kanonik açılımda yer almayan minterimlerin indisleri maksterim olarak seçilir.
  Örnek: F(A,B,C) = Σm(1,3,5,6,7) = ΠM(0,2,4)
  F(A,B,C) = m1 + m3 + m5 + m6 + m7 = M0 · M2 · M4
  $F(A,B,C) = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC = (A+B+C)(A+\bar{B}+C)(\bar{A}+B+C)$
- 2. kanonik açılımdan 1. kanonik açılıma (maksterimden minterime) dönüşüm
  2. kanonik açılımda yer almayan maksterimlerin indisleri minterim olarak seçilir.
  Örnek: F(A,B,C) = ΠM(0,2,4) = Σm(1,3,5,6,7)
- Minterimlerin toplamı şeklindeki ifadenin tümleyeninin bulunması
  Birinci kanonik biçimde yer almayan minterimler seçilir
  Örnek: F(A,B,C) = Σm(1,3,5,6,7) F'(A,B,C) = Σm(0,2,4)
- Maksterimlerin çarpımı şeklindeki ifadenin tümleyeninin ifadenin bulunması
  İkinci kanonik biçimde yer almayan maksterimler seçilir
  F(A,B,C) = ΠM(0,2,4) F'(A,B,C) = ΠM(1,3,5,6,7)

Kanonik Açılımlar ve De Morgan Teoremi

- Bir fonksiyonun tümleyeninin 1. kanonik biçimdeki ifadesine De Morgan teoremi uygulanırsa fonksiyonun kendisinin 2. kanonik biçimdeki ifadesi elde edilir.

Örnek:

Yansı 2.39'dai fonksiyonun tümleyeninin ÇT şeklindeki ifadesi:

$\bar{F} = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C}$

De Morgan:

$\bar{\bar{F}} = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C}}$

$F = (A+B+C)(A+\bar{B}+C)(\bar{A}+B+C)$ 2. kanonik biçimdeki (TÇ) ifade (2.43).

- Bir fonksiyonun tümleyeninin 2. kanonik biçimdeki ifadesine De Morgan teoremi uygulanırsa fonksiyonun kendisinin 1. kanonik biçimdeki ifadesi elde edilir.

Örnek:

Yansı 2.44'teki fonksiyonun tümleyeninin TÇ şeklindeki ifadesi:

$\bar{F} = (A+B+\bar{C})(A+\bar{B}+\bar{C})(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$

De Morgan:

$\bar{\bar{F}} = \overline{(A+B+\bar{C})(A+\bar{B}+\bar{C})(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})}$

$F = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$ 1. kanonik biçimdeki (ÇT) ifade (2.38).