

BOOLE İŞLEVLERİNİN SADELEŞTİRMESİNDE KARNAUGH HARİTASI TEKNİĞİ

Mantık devrelerinin tasarımında, bir devrenin iyi niteliklerinin değerlendirilmesi sırasında birçok etken göz önüne alınmalıdır. Bu etkenlerden biri ve en önemlisi maliyettir. Bir devrenin maliyeti, onu oluşturan öğelerin maliyeti, tasarımın maliyeti ve devrenin kurulumunun maliyeti ile devrenin bakımının bir işlevidir. Ek olarak, devrenin güvenilirliği toplam niteliğinin değerlendirilmesinde göz önüne alınmalıdır. Güvenilirlik, yüksek ölçüde güvenilir öğelerin kullanılmasıyla ya da içinde daha az güvenilir çok sayıda öğenin yer aldığı yedeklilik teknikleriyle sağlanabilir. Bir mantık devresinin niteliksel değerlendirilmesinde göz önüne alınacak üçüncü bir etken, girişlerindeki değişimlere yanıt vermek üzere harcadığı süredir. Bu üç etken, devrenin değerlendirilmesi sırasında göz önüne alınması gereken maddelerin birer durumlarını ve olasılıklarını kapsayan bir liste oluşturmadıkları gibi bağımsız da değildirler.

Yukarıdaki etkenlerin tümü önemli olsa bile, tümünü başaran tek bir basit tasarım yöntemi mevcut değildir. Bununla birlikte, bu etkenlerin belirli yönlerinin göze çarpan özellikler olduğu düşünülürse, en uygun mantık devrelerinin tasarımına biçimsel bir yaklaşım geliştirilebilir.

Verilen bir devre teknolojisi için, bir devrenin toplam yanıt süresinin en ufak olduğunu varsayalım. Bu da, tüm kapılar yayılım gecikmesi ortaya çıkardıklarından dolayı bir sinyalin geçmesi gerektiği mantık seviyelerinin sayısının azaltılmasıyla sağlanır. Her birleşik devrenin kanonik (doğal) bir formülle betimlenebildiği hatırlanırsa, herhangi bir mantık devresini çift hatlı mantık varsayımıyla en fazla iki seviyede kurmak mümkündür. Genellikle bir fiziksel gerçekleştirmede bir değişkenin tümleyeninin ve kendisinin girişler olarak her zaman mevcut olacağı söylenebilir. Bu kanonik ifadeye Boole cebri teoremlerinin uygulanmasıyla, çeşitli iki seviyeli devreler cebirsel biçimde temsil edilebilir. Bu nedenle, bir devrenin yayılım gecikmesi süresini en azda tutmak üzere, ilgi normal formüllü gösterimli devrelerle sınırlandırılmalıdır.

Ayrıca, bir mantık devresinin niteliksel değerlendirilmesini etkileyen tek diğer etkenin öge maliyeti olduğunu varsayalım. Genel olarak, herhangi bir Boole işlevi için her biri normal bir formülle temsil edilen çok sayıda iki seviyeli gerçekleştirme mümkündür. Bu yüzden devre gerçekleştirmesinde en düşük öge maliyetli normal formülü belirlemek istenir. Öge maliyetinin basit bir ölçümü gerçekleştirmede kullanılacak olan kapıların sayısıdır. Normal biçimdeki cebirsel ifadeler türünden, kapıların sayısı ifadedeki birden fazla harf-değişkenli terimlerin sayısından bir büyüktür. Ancak, eğer yalnızca tek bir terim varsa, kapıların sayısı basitçe bir tanedir. Öge maliyetinin ikinci bir ölçümü, devredeki kapı girişlerinin toplam sayısının sayılmasıdır. Bu da yine normal biçimdeki cebirsel ifadelerle ilişkilendirilebilir. Kapı girişlerinin sayısı, ifadedeki harf-değişkenlerin sayısı + birden fazla harf değişken içeren terimlerin sayısına eşittir. Ancak eğer yalnızca tek bir terim varsa, kapı girişlerinin sayısı basitçe harf değişkenlerin sayısına eşittir.

Yukarıdaki iki ölçütün her birinin bir Boole ifadesine uygulanmasıyla, karmaşıklığının bir ölçüsünün elde edilmesi mümkündür. Bu sayısal nicelik, ifadenin maliyeti diye adlandırılır. Yukarıdaki irdelemelerden ayrı düşünilemeyen şey, yalnızca tek çıkışlı birleşik devrelerin gerçekleştirilmekte olduğudur. Çok çıkışlı birleşik bir devre, bir Boole eşitlikleri takımı tarafından betimlenir. Böyle bir durumda, maliyet ölçütünün biraz değiştirilmesi gerekir.

Daha önceden incelediğimiz cebirsel yöntemler, kuramsal olarak bize herhangi bir işlevi sadeleştirme olanağını sağlamaktadır. Ancak bu yaklaşımda bazı sorunlar bulunmaktadır:

- a. Sadeleştirme sırasında izlenecek belirli bir yöntem yoktur.
- b. Yaklaşım, büyük ölçüde deneyimlere dayanmaktadır.
- c. Bir işlev işlemlendiğinde, sonucun en sadeleşmiş ifade olup olmadığından emin olunamaz. Yapacak başka bir sadeleştirme kalmadığında dahi, en sade ifadeye ulaşılamamış olunabilir.
- d. Dört ya da beş değişkenden daha fazla değişken içeren işlevlerde cebirsel sadeleştirme oldukça zorlaşır.
- e. Eşitlikler yeniden yazılırken, kopyalama hataları yapmak çok kolaydır.

Bütün bunlar nedeniyle, işlevlerin sadeleştirilmesi sırasında daha algoritmik tekniklerin kullanılması gerekmektedir. Bu teknikler,

- a. Karnaugh haritası
 - b. Yinelemeli uzlaşma
 - c. Quine-McCluskey listeleme yöntemi
- şeklinde dir.

Burada yalnızca dört değişken dahil olmak üzere Karnaugh haritası (kısaca, K-haritası) tekniği incelenen ektir.

Boole işlevi uygulanan sayısal mantık devrelerinin karmaşıklığı, doğrudan doğruya işlevin uygulandığı cebirsel ifadenin karmaşıklığından kaynaklanmaktadır. Bir işlevin doğruluk tablosu ile gösterilmesi benzersiz olmasına rağmen, cebirsel olarak birçok farklı biçimde ifade edilebilmektedir. Daha önce incelenen yöntemlerle Boole işlevleri cebirsel olarak sadeleştirilebilir. Ancak bu sadeleştirme işlemi sıkıcı ve acemicedir, çünkü işlem sürecindeki birbirini izleyen her adımı tahmin etmek için kullanılan özel kurallardan yoksundur. Harita yöntemi, Boole işlevlerinin sadeleştirilmesinde daha basit ve dolaysız bir yöntemdir. Bu yöntem, bir doğruluk tablosunun şemasal bir gösterimi olarak da değerlendirilebilir. İlk kez E. W. Veitch tarafından önerilen ve daha sonra M. Karnaugh tarafından biraz değiştirilen harita yöntemi, "Veitch şeması" ya da "Karnaugh haritası" olarak da adlandırılır.

Harita karelerden oluşan bir şemadır. Her bir kare bir miniterimi (ya da maksiterimi) göstermektedir. Dolayısıyla, n değişkenli bir haritada her biri tek bir miniterim olan 2^n tane kare bulunmaktadır. Kare ayrıca "hücre" olarak da adlandırılır. Haritanın her bir hücreindeki miniterim, hücrenin apsis ve ordinatlarında gösterilen değişkenlerin çarpımıdır.

Boole işlevleri miniterimlerin toplamı olarak açıklanabileceği için, işlev haritada grafiksel olarak işlevin miniterimleri içerdiği karelerle çevrili alanlarla gösterilebilir. Harita aslında, bir işlevin standart bir biçimde ifade edilebileceği olası tüm yolları içeren görsel bir şemayı temsil eder. Uygulayıcı, aynı işlev için farklı cebirsel ifadeler geliştirebilir ve bunların içinden en sadesini seçebilir.

Harita ve uygulamalarına geçmeden, birkaç önemli tanımın verilmesi gerekmektedir.

Mod-2 toplama ($\oplus$):

x	y	$\oplus$
0	0	0
0	1	1
1	0	1
1	1	0

Hamming uzaklığı (H): $A = a_1a_2.....a_k$ ve $B = b_1b_2.....b_k$ biçiminde ikili sayılar olsun. Bu sayılar arasındaki Hamming uzaklığı,

$$H = \sum_{i=1}^k a_i \oplus b_i$$

$A = 10101$

$B = 10011$

$\oplus$

Örnek 1:

$10011 \rightarrow H = 1 + 0 + 0 + 1 + 1 = 3$

$\rightarrow H = 1 + 0 + 0 + 1 + 1 = 3$

$A = 101010$

$B = 101110$

$\oplus$

$000100 \rightarrow H = 0 + 0 + 0 + 1 + 0 + 0 = 1$

$\rightarrow H = 0 + 0 + 0 + 1 + 0 + 0 = 1$

Bitişik (komşu) hücreler: Aralarındaki Hamming uzaklığının (1)'e eşit olduğu hücrelere denir. Diğer bir deyişle, aralarında yalnızca bir bit fark bulunan hücrelerdir. n değişkenli bir haritada, her bir hücre n sayıda komşu hücreye sahiptir.

Alt küp: Bir haritada, her biri m sayıda hücreye komşu olan 2^m tane hücre kümesidir. Değişken sayısı n ve alt küpün derecesi m olmak üzere bir alt küp, içinde $n-m$ sayıda harf-değişken içeren bir çarpım terimi ile açıklanır. Daha basit bir deyişle, kullanımda kolaylık sağlaması açısından bir alt küp içinde 1 içeren 2^m komşu hücre topluluğudur.

BİR DEĞİŞKENLİ HARİTA

Burada ilgilenilen değişken x olsun. Tek değişken için iki miniterim mevcuttur.

$x = 0$	$m_0 = x'$
$x = 1$	$m_1 = x$

Haritalarda yalnızca değişkenlerin (1)'e eşit olduğu koordinatların gösterilmesi, karışıklığa yol açmaması nedeniyle tercih edilir. Buna göre, bir değişkenli bir harita aşağıdaki şekilde gösterilecektir.

	m_0
x	m_1

Kural olarak, n değişken için 2^{2^n} farklı anahtarlama (Boole) işlevi elde edilebilir. Buna göre bir değişkenli bir haritada dört farklı işlev üretilecektir. Bu işlevler sırasıyla aşağıda gösterilmiştir:

x	$f(x)$
0	0
1	0

$f(x) = 0$

x	$f(x)$
0	1
1	1

$f(x) = 1$

x	$f(x)$
0	0
1	1

$f(x) = x$

x	$f(x)$
0	1
1	0

$f(x) = x'$

x	

$f(x) = 0$

	1
x	1

$f(x) = 1$

x	1

$f(x) = x$

	1
x	

$f(x) = x'$

İKİ DEĞİŞKENLİ HARİTA

Burada ilgilenilen değişkenler x ve y olsun. İki değişken için dört miniterim mevcuttur ve onaltı farklı işlev üretilebilir. İki değişkenli bir harita aşağıdaki biçimdedir:

	y		
	$y' (y = 0)$	$y (y = 1)$	
$x' (x = 0)$	m_0 (00) $x'y'$	m_1 (01) $x'y$	
$x (x = 1)$	m_2 (10) xy'	m_3 (11) xy	

$\equiv$

	y	
x	0	1
	2	3

y

İkinci haritada, miniterimler onlu sayı karşılığı olan değerlerle gösterilmiştir. Haritadan görüleceği gibi, iki değişken için dört miniterim bulunduğundan, haritada her miniterim için bir tane olmak üzere dört hücre bulunur. Her satır ve sütun için işaretlenen (0)'lar ve (1)'ler, sırasıyla x ile y değişkenlerinin değerini gösterir. x değişkeni satır 0'da tümleyen, satır 1'de ise normal olarak gözükmemektedir. Benzer şekilde, y değişkeni de sütun 0'da tümleyen, sütun 1'de ise normal olarak gözükmemektedir.

x <table><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table> y $f(x,y) = 0$					x <table><tr><td>1</td><td></td></tr><tr><td></td><td></td></tr></table> y $f(x,y) = x'y' \text{ (NOR)}$	1				x <table><tr><td></td><td>1</td></tr><tr><td></td><td></td></tr></table> y $f(x,y) = x'y$		1			x <table><tr><td></td><td></td></tr><tr><td></td><td>1</td></tr></table> y $f(x,y) = xy \text{ (AND)}$				1
1																			
	1																		
	1																		
x <table><tr><td></td><td></td></tr><tr><td>1</td><td></td></tr></table> y $f(x,y) = xy'$			1		x <table><tr><td>1</td><td>1</td></tr><tr><td></td><td></td></tr></table> y $f(x,y) = x'$	1	1			x <table><tr><td></td><td></td></tr><tr><td>1</td><td>1</td></tr></table> y $f(x,y) = x$			1	1	x <table><tr><td>1</td><td></td></tr><tr><td></td><td>1</td></tr></table> y $f(x,y) = x'y' + xy \text{ (XNOR)}$	1			1
1																			
1	1																		
1	1																		
1																			
	1																		
x <table><tr><td>1</td><td></td></tr><tr><td>1</td><td></td></tr></table> y $f(x,y) = y'$	1		1		x <table><tr><td>1</td><td></td></tr><tr><td>1</td><td></td></tr></table> y $f(x,y) = y$	1		1		x <table><tr><td></td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table> y $f(x,y) = x + y \text{ (OR)}$		1	1	1	x <table><tr><td>1</td><td></td></tr><tr><td>1</td><td>1</td></tr></table> y $f(x,y) = x + y'$	1		1	1
1																			
1																			
1																			
1																			
	1																		
1	1																		
1																			
1	1																		
x <table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td></td></tr></table> y $f(x,y) = x' + y' \text{ (NAND)}$	1	1	1		x <table><tr><td></td><td>1</td></tr><tr><td>1</td><td></td></tr></table> y $f(x,y) = x'y + xy' \text{ (XOR)}$		1	1		x <table><tr><td>1</td><td>1</td></tr><tr><td></td><td>1</td></tr></table> y $f(x,y) = x' + y$	1	1		1	x <table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table> y $f(x,y) = 1$	1	1	1	1
1	1																		
1																			
	1																		
1																			
1	1																		
	1																		
1	1																		
1	1																		

ÜÇ DEĞİŞKENLİ HARİTA

Byrada ilgililen değışkenler x , y ve z olsun. Üç değışken için sekiz miniterim mevcuttur. Üç değışken için bir harita aşağıdaki biçimdedir:

	$y'z'$	yz'	yz'	yz'
$x' (x = 0)$	m_0 (000) $x'y'z'$	m_1 (001) $x'y'z$	m_3 (011) $x'yz$	m_2 (010) $x'yz'$
$x (x = 1)$	m_4 (100) xyz'	m_5 (101) xyz	m_7 (111) xyz	m_6 (110) xyz'

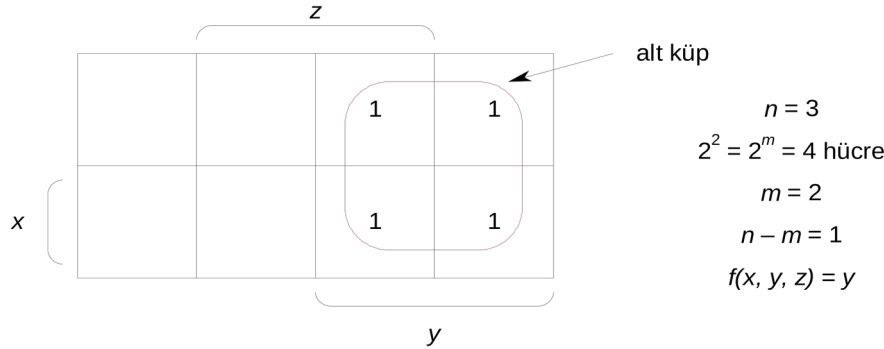
≡

x {	0	1	3	2
	4	5	7	6
	y			

Bu haritada miniterimler normal ikili sırada verilmemiştir. Sıranın özelliği, verilen sırada yalnızca bir bitin 1'den 0'a ya da 0'dan 1'e değişmiş olmasıdır. Örneğin, m_6 olarak belirtilen hücre, satır 1 ve sütun 10'a karşılıktır. Bu iki değer birleştiği zaman, onlu 6 sayısının eşdeğeri olan 110 ikili sayısı elde edilir. her bir değişkenin 1 olduğu dört hücre ve 0 olduğu dört hücre olduğuna dikkat edilmelidir.

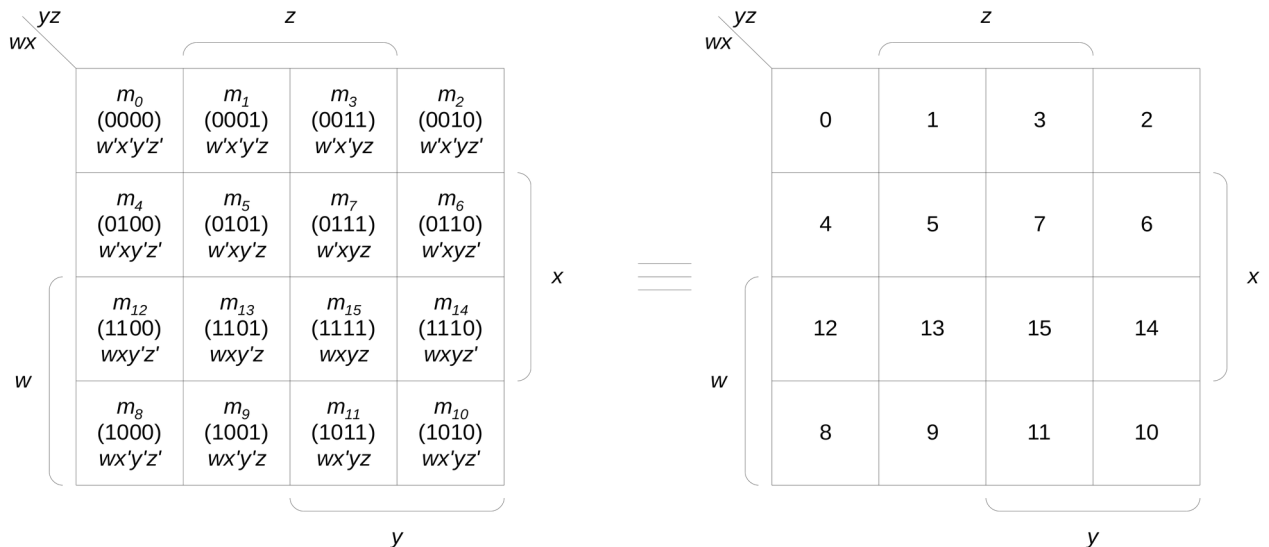
Üç değişkenli bu haritada m_5 ve m_7 komşu hücrelerde bulunmaktadır. y değişkeni m_5 'de tümleyen, m_7 'de ise normal durumundadır. Buna karşılık, x ve z değişkenleri bu iki hücrede de aynıdır. Boole cebri önermelerinden, komşu hücrelerdeki iki miniterimin toplamının, yalnızca iki harf-değişken içeren tek bir VE terimine sadeleştirilebileceği sonucu ortaya çıkar. Yani, $m_5 + m_7 = xy'z + xyz = xz(y + y') = xz$ olur. Dolayısıyla, bu iki hücre arasındaki fark y değişkenidir. Bu da iki terimin toplamı alındığında iptal edilebilir. Bu nedenle, komşu hücrelerdeki iki miniterimin VEYA'lanması, farklı olan değişkenin atılmasına neden olur. Yine bu haritada, m_3 'ün komşu hücreleri m_1 , m_2 ve m_7 'dir.

Örnek 2: $f(x, y, z) = \sum m(2, 3, 6, 7)$



DÖRT DEĞİŞKENLİ HARİTA

Burada ilgilenilen değişkenler w , x , y ve z olsun. Dört değişken için onaltı miniterim mevcuttur. Dört değişkenli bir harita aşağıdaki biçimdedir:

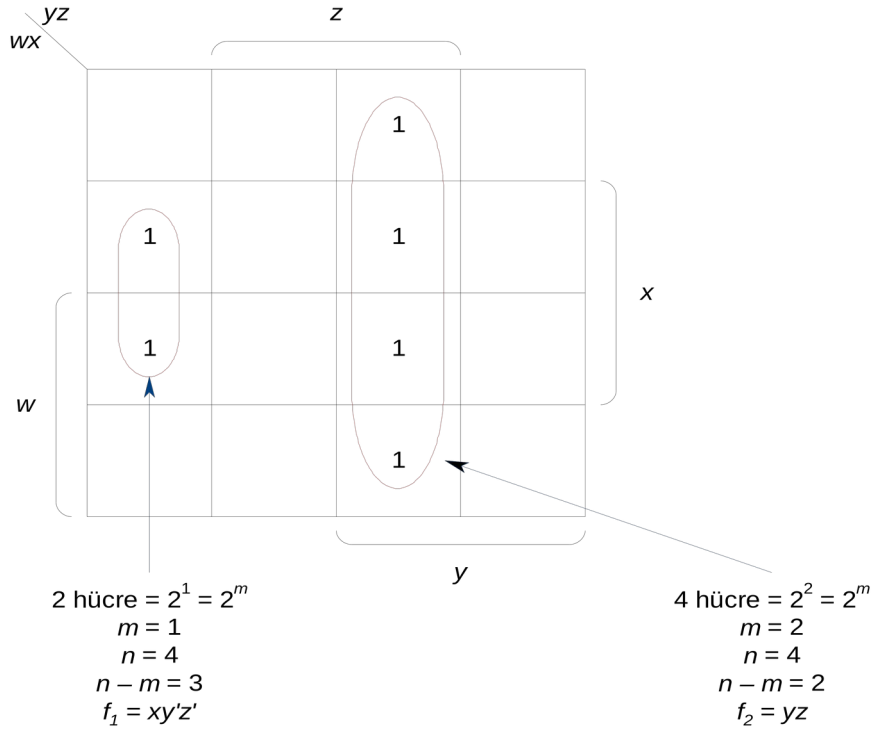


Dört değişkenli Boole işlevlerinin harita yöntemi ile sadeleştirilmesi, üç değişkenli işlevlerinkine benzemektedir. Burada, örneğin m_{15} 'in komşu hücreleri m_7 , m_{11} , m_{13} ve

m_{14} 'dür. Buna ek olarak, haritanın hem üst ve alt kenarlarının, hem de sağ ve sol kenarlarının komşu hücreler oluşturacak şekilde birbirine dokunacak tarzda bir yüzey üzerinde bulunduğu varsayılacaktır. Örneğin, m_4 ve m_6 , m_1 ve m_9 komşu hücreler oluşturur. Sadeleştirme işleminde kullanılan komşu hücreler birleşimi, dört değişkenli haritanın incelenmesiyle kolayca belirlenir.

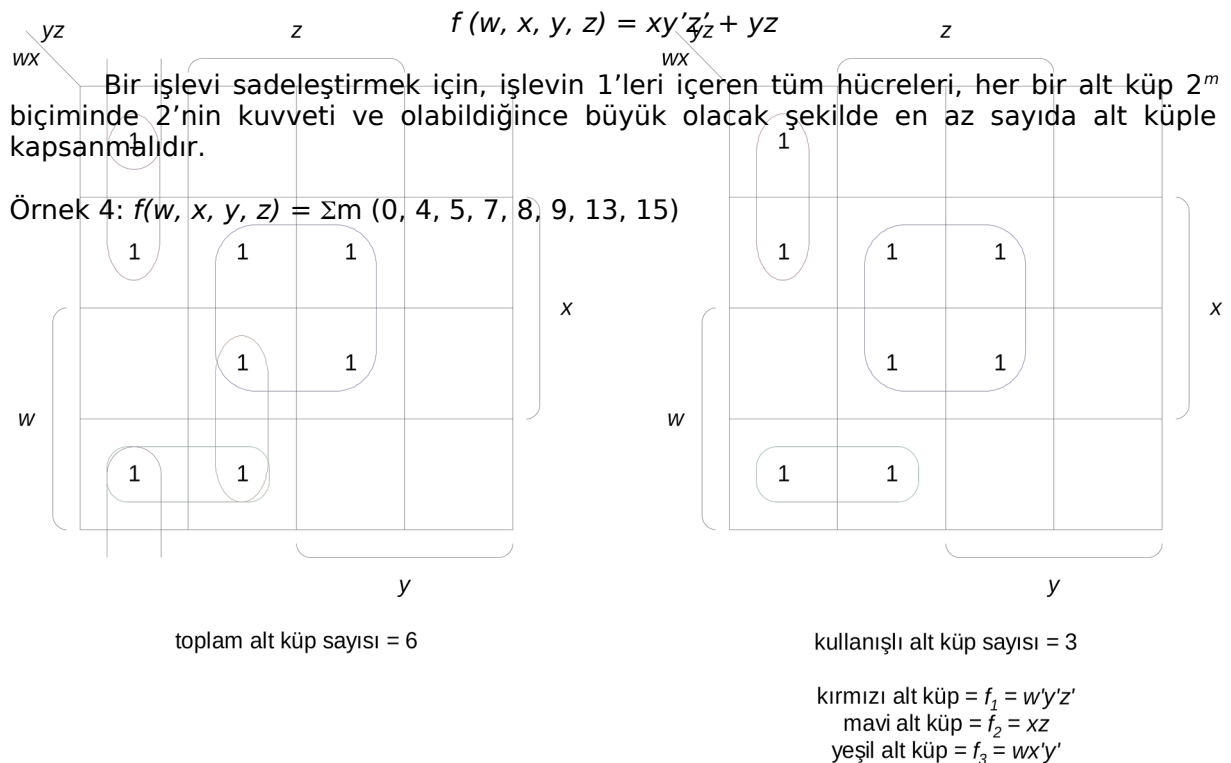
Dört değişkenli bir haritada, bir hücre bir miniterimi temsil eder ve dört harf-değişkenli bir çarpım terimi verir. İki komşu hücre üç harf-değişkenli, dört komşu hücre iki harf-değişkenli, sekiz komşu hücre de bir harf değişkenli bir çarpım terimi verir. Onaltı komşu hücre ise, 1'e eşit olan bir işlevi temsil eder. Diğer hücre birleşimleri işlevi sadeleştiremez.

Örnek 3: $f(w, x, y, z) = \sum m(3, 4, 7, 11, 12, 15)$



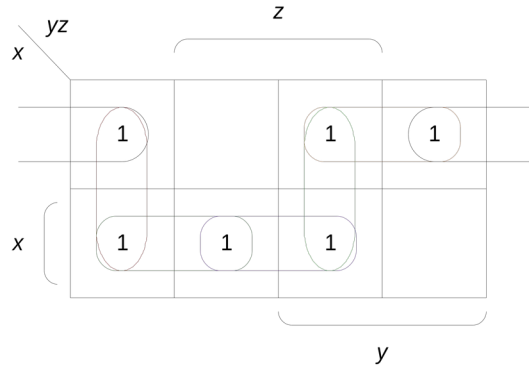
Bir işlevi sadeleştirmek için, işlevin 1'leri içeren tüm hücreleri, her bir alt küp 2^m biçiminde 2'nin kuvveti ve olabildiğince büyük olacak şekilde en az sayıda alt küple kapsanmalıdır.

Örnek 4: $f(w, x, y, z) = \sum m(0, 4, 5, 7, 8, 9, 13, 15)$

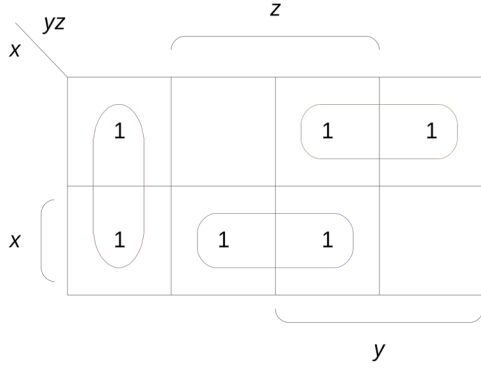


Harita ile sadeleştirmede, en sade ifade yegane ifade olmayabilir.

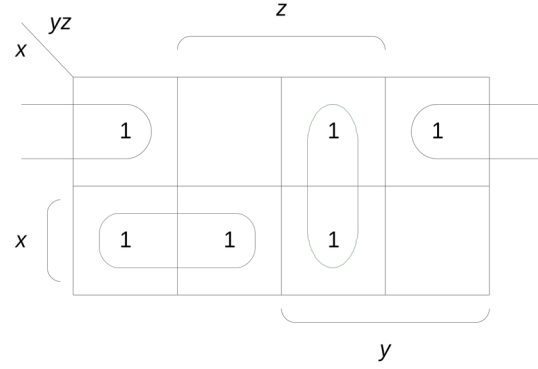
Örnek 5: $f(x, y, z) = \sum m(0, 2, 3, 4, 5, 7)$



toplam alt küp sayısı = 6



$$f(x, y, z) = y'z' + xz + x'y$$



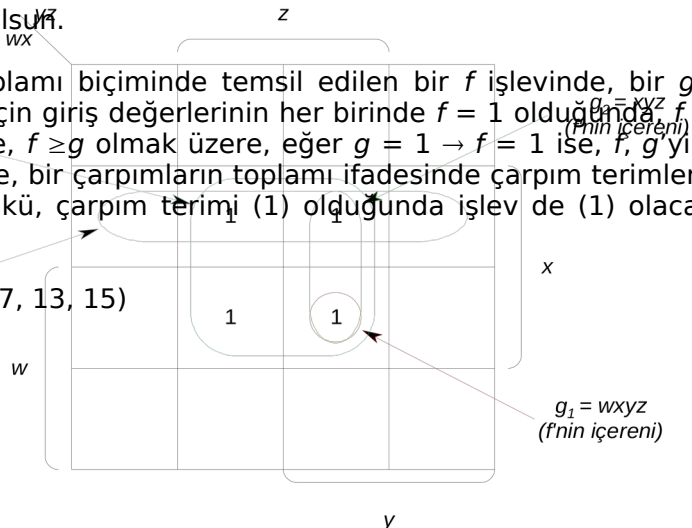
$$f(x, y, z) = x'z' + xy' + yz$$

ASAL İÇERENLER

f işlevi bir Boole işlevi olsun.

İçeren: Çarpımların toplamı biçiminde temsil edilen bir f işlevinde, bir g çarpım terimi ancak ve ancak $g = 1$ için giriş değerlerinin her birinde $f = 1$ olduğunda, f işlevinin bir içerenidir. Diğer bir deyişle, $f \geq g$ olmak üzere, eğer $g = 1 \rightarrow f = 1$ ise, f, g 'yi içeriyor demektir. Basit bir açıklama ile, bir çarpımların toplamı ifadesinde çarpım terimlerinin her biri, işlevin bir içerenidir. Çünkü, çarpım terimi (1) olduğunda işlev de (1) olacağından, işlevi içermektedir.

Örnek 6: $f(w, x, y, z) = \sum m(5, 7, 13, 15)$
(f 'nin içereni değil)



Asal İçeren: f 'nin bir içereni, eğer f 'nin bir içerenini oluşturan daha büyük bir miniterim öbeği (alt küp) tarafından kapsanmıyorsa, f 'nin bir asal içerenidir. Diğer bir deyişle, f 'nin herhangi bir asal içereni f tarafından kapsanan bir çarpım terimi olup, bu asal içereinden herhangi bir harf-değişkenin atılması, f tarafından kapsanmayan yeni bir çarpım terimi oluşturur.

Örnek 7: Örnek 6'da verilen ifadede,

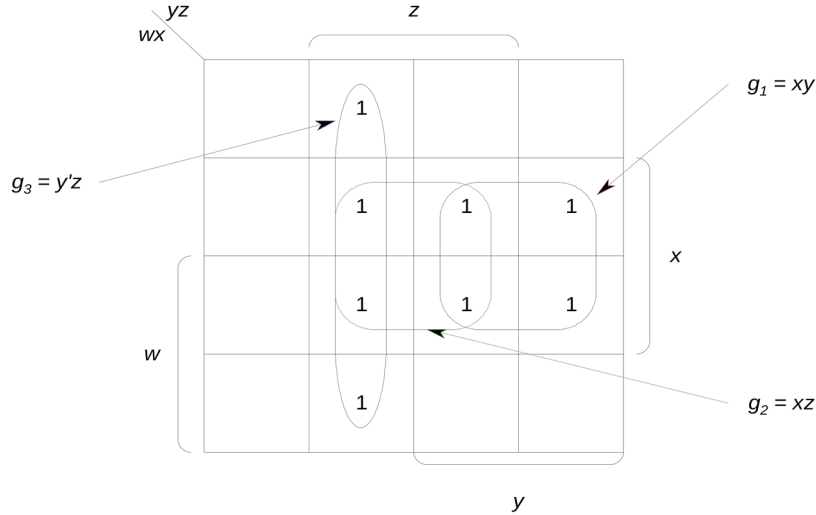
$w'xz$ bir asal içeren değildir, çünkü xz , $w'xz$ kısmını kapsamaktadır.

xyz bir asal içeren değildir, çünkü xz , xyz kısmını kapsamaktadır.

xz bir asal içerenidir, çünkü daha büyük bir alt küp tarafından kapsanmamaktadır.

Öz Asal İçeren: f 'nin bir asal içereni, eğer f 'nin diğer bir asal içereni tarafından kapsanmayan bir miniterim (1-hücresi) kapsıyorsa, f 'nin bir öz asal içerenidir. "Öz" ibaresinin yer almasının nedeni, en küçük (sadeleşmiş) çarpımların toplamı ifadesinde bu asal içerenin kullanılması zorunluluğundandır.

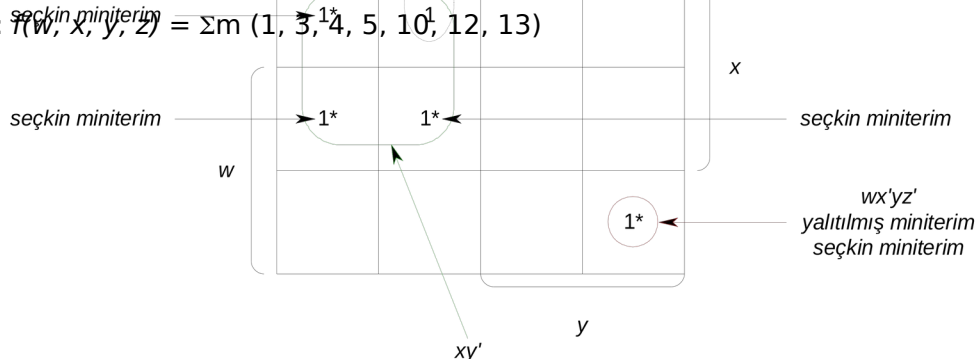
Örnek 8: $f(w, x, y, z) = \sum m(1, 5, 6, 7, 9, 13, 14, 15)$



Verilen haritada, g_1, g_2 ve g_3 asal içerenlerdir; g_1 ve g_3 öz asal içerenlerdir, g_2 ise öz asal içeren değildir.

Seçkin miniterim: f 'nin herhangi bir yalıtılmış miniterimi, ya da f 'nin herhangi diğer asal içerenleri tarafından kapsanmayan herhangi bir öz asal içeren miniterimi, f 'nin bir seçkin miniterimidir ve (*) işareti ile gösterilir.

Örnek 9: $f(w, x, y, z) = \sum m(1, 3, 4, 5, 10, 12, 13)$



Verilen haritada,

İçerenler: $wx'yz'$, $w'x'z$, $w'y'z$, $w'xy'$, wxy' , $xy'z'$, $xy'z$, tüm miniterimler

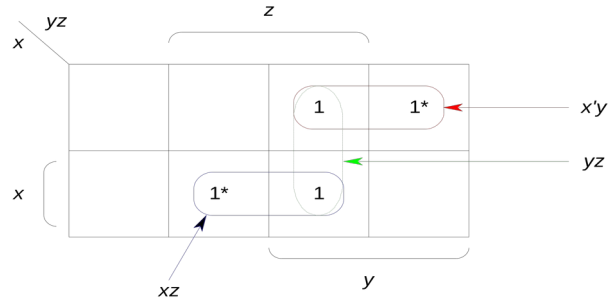
Asal içerenler: $wx'yz'$, $w'x'z$, $w'y'z$, xy'

Öz asal içerenler: $wx'yz'$, $w'x'z$, xy'

Dolayısıyla en küçük çarpımların toplamı ifadesi, $f(w, x, y, z) = wx'yz' + w'x'z + xy'$ olur.

Herhangi bir Boole işlevinde, asal içerenler kümesi yegânedir, çünkü yegâne bir miniterim kümesinden türetilmiştir. Burada sorulması gereken, işlevi en ufak biçimde temsil etmek için asal içerenlerin tümüne gerek duyulup duyulmayacağıdır. Bir işlevin tüm miniterimleri, asal içerenler kümesinin uygun bazı alt kümeleriyle kapsanıyorsa, en son ifade bu alt kümede bulunmayanlar da dahil, işlev için gereğinden daha uzun bir ifade ile sonuçlanır. Uzlaşma teoremi ile verilenler buna örnek teşkil eder.

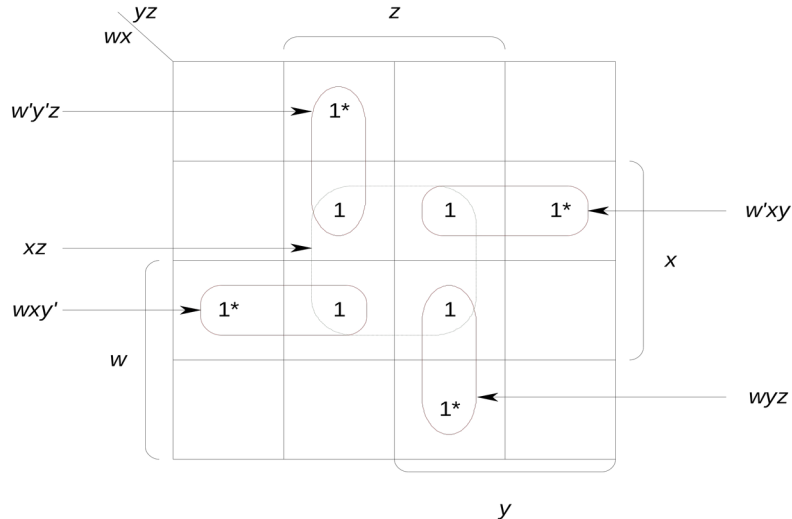
Örnek 10: $f(x, y, z) = \sum m(2, 3, 5, 7)$



Verilen örnekte, $x'y$, yz ve xz asal içerenlerdir. Ancak, $x'y$ ile xz öz asal içerenlerdir. yz asal içereni gerekli olmayan asal içerenlerdir, bu nedenle en küçük ifadeye yer almaz. Aynı durum uzlaşma teoremi ile de gösterilebilir.

$f(x, y, z) = x'y + yz + xz$ olup, yz uzlaşma terimidir. Bu nedenle ifadeden çıkartılabilir. Dolayısıyla, en küçük çarpımların toplamı, $f(x, y, z) = x'y + xz$ olacaktır.

Örnek 11: $f(w, x, y, z) = \sum m(1, 5, 6, 7, 11, 12, 13, 15)$



Verilen örnekte, $w'y'z$, wxy' , $w'xy$, wyz ve xz asal içerenlerdir. Ancak xz asal içereni gerekli olmayan asal içerebilir, bu nedenle en küçük ifadede yer almaz.

Dolayısıyla en küçük ifade, $f(w, x, y, z) = w'y'z + wxy' + w'xy + wyz$ olur.

Verilen tüm açıklamalar çerçevesinde, en küçük bir çarpımların toplamı ifadesinin belirlenmesinde izlenmesi gereken yöntem aşağıdaki şekildedir:

1. İşlevin tüm asal içerenlerini ve öz asal içerenlerini bulup listeleyin.
2. En küçük çarpımların toplamına tüm öz asal içerenleri ekleyin.
3. 1-hücreleri öz asal içerenler tarafından kapsanan asal içerenleri listeden kaldırın.
4. İşlevin tüm 1-hücreleri kapsanıyorsa bulunan ifade yegânedir.
5. Aksi takdirde işlevi tümüyle kapsayacak tarzda ek asal içerenleri seçin.
6. İşlevin en küçük çarpımları toplamının her zaman yegâne olmayacağını unutmayın.

DİKKATE ALINMAYANLAR

Şimdiye kadar karşılaşılan mantıksal tasarım problemlerinin tümünde, belirli bir çıkış değeri kendisiyle ilişkili olan olası giriş değerleriyle bağlantılı idi. Ender olarak, belirli giriş birleşimleri olmadan da mevcut olabilen sistemler de olabilir. Böyle durumlarda, çıkış doğru ya da yanlış olarak tanımlanabilir. Bütün bunlardan sonra, eğer belirli bir giriş olanaklı değilse bununla ilişkili çıkış da anlamlı olmayacaktır. Ya da acaba öyle midir? Burada bu anlamsız çıkışların nasıl kullanılabilir bir şekle getirilebileceği açıklanacaktır.

Haritadaki 1'ler ve 0'lar, işlevi sırasıyla 1'e ya da 0'a eşitleyen değişkenler birleşimini gösterir. Birleşimler genellikle işlevin 1 olduğu durumları veren bir doğruluk tablosundan elde edilir. Diğer bütün durumlarda işlevin 0'a eşit olduğu varsayılır. Bu varsayım her zaman doğru değildir. çünkü bazı giriş değişkenleri birleşimleri hiçbir zaman görülmez. Örneğin, dört bitlik onlu bir kodda (doğal BCD kodu) kullanılmayan altı birleşim vardır ve bunlar onlu 10, 11, 12, 13, 14 ve 15 sayılarına karşılık gelir. Bu kodun kullanıldığı sayısal bir devre, sistem gerektiği gibi çalıştığı sürece, kullanılmayan bu birleşimlerin hiçbir zaman olmayacağı varsayımı ile çalışır. Sonuç olarak, değişkenlerin bu birleşimlerinde işlev çıkışının ne olacağına aldırış edilmez. İşlevin daha ileri düzeyde sadeleştirilmesi için, harita üzerinde dikkate alınmayan bu durumlar kullanılabilir.

Bu dikkate alınmaz birleşiminin harita üzerinde 1 ile işaretlenemeyeceği unutulmamalıdır. Çünkü, bu söz konusu giriş birleşimi için işlevin her zaman 1 olmasını gerektirecektir. Benzer şekilde, hücreye 0 konulması, işlevin 0 olmasını gerektirir. Bu nedenle, dikkate alınmaz bu durumları 1'lerden ve 0'lardan ayırt etmek üzere "d" harfi kullanılır.

Dikkate alınmaz çıkış değerleri önemli olmadığından, bunların 1 ya da 0 ile doldurulabilir olması bir üstünlüktür. Dolayısı ile, haritada işlevi sadeleştirmek için komşu hücreler seçilirken, "d"lerin 0 ya da 1 olduğu varsayılır. Burada önemli olan, hangi "d"nin en sadeleşmiş ifadeyi verdiğidir. Ayrıca daha geniş bir alt küpün kapsanmasına katkıda bulunmuyorsa, "d" hiç kullanılmayabilir. Her durum için seçim, yalnızca ulaşılabilecek en son sadeleştirmeye bağlıdır. Kısaca özetlenirse, "d"lerin seçilmiş bir alt küpte içerilmiş olması gerekmez. Yalnız 1'leri içeren bir alt küp genişletilmek istendiğinde, "d"lerden de gerektiği kadarı içerilmelidir. Yine burada, Karnaugh haritası sadeleştirme yöntemlerine ve kurallarına tümüyle uyulması gerekmektedir.

Örnek 12: Üç katlı bir binada bir asansör kapısını denetleyecek olan bir mantık devresi tasarlanacaktır. Devrede, asansör motoruna ait M , 1. kat algılayıcısına ait KA_1 , 2. kat

algılayıcısına ait KA_2 ve 3. kat algılayıcısına ait KA_3 olmak üzere dört giriş ve kapaıya ait bir K çıkışı bulunmaktadır. Sistem alıřması ařağıdaki řekilde olacaktır:

- M sinyali, asansör motorunun hareket ettiğini ($M = 1$) ya da durduğunu ($M = 0$) göstermektedir.
- KA_1 , KA_2 ve KA_3 sinyalleri kat belirleme sinyalleri olup, normalde 0 durumundadır. Ancak, asansör istenen katın seviyesiyle aynı hizaya geldiklerinde 1 durumuna geçerler. Örneğın, asansör 2. kata geldiğinde $KA_2 = 1$ ve $KA_1 = KA_3 = 0$ olacaktır.
- Devrenin K çıkışı, asansör kapısının açılmasını denetleyen bir sinyal olup, normalde 0 durumundadır. Asansör istenen kata geldiğinde kapısının açılması için $K = 1$ olur.

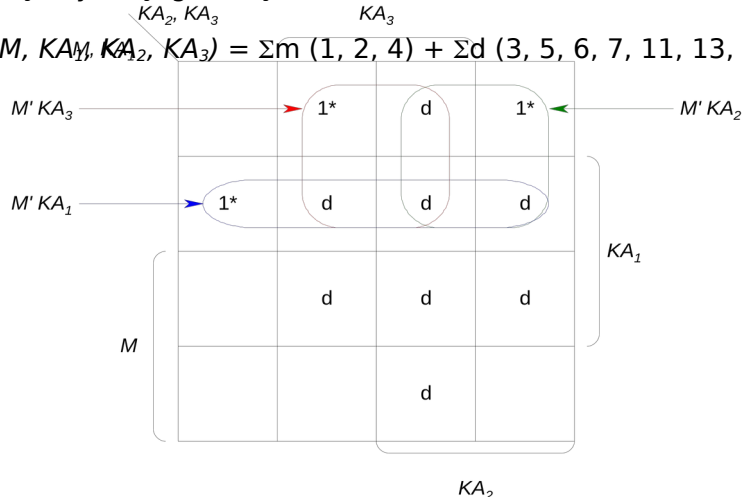
Doğruluk tablosu oluştururken ařağıdakiler göz önüne alınmalıdır:

- Asansör aynı anda birden fazla katla aynı hizaya gelemeyeceğı için belirli bir anda kat algılama sinyallerinden ancak bir tanesi 1 olabilir (ama algılayıcılar bozulmuşsa, hepsi birden 1 olabilir !). Dolayısıyla, birden fazla kat algılama sinyalinin 1 olduğı durumlar olanaksız giriş koşulları olup, “d” ile işaretlenecektir. Bu koşullar sekiz satırda ayrı ayrı gösterilmektedir.
- Diğerk sekiz duruma bakıldığında, M sinyalinin 1 olması asansörün hareket etmekte olduğunu gösterdiğinden $K = 0$ olmalıdır. Aksi durum, can güvenliğı açısından bir tehlike oluşturur. $M = 0$ olduğunda, yani asansör durduğunda, kat algılayıcılarının herhangi birinden gelen sinyal 1 ise, $K = 1$ olacaktır. $M = 0$ ve tüm kat algılayıcı sinyal bilgileri 0 iken, asansörün durduğu, ancak katlardan herhangi birinin hizasında olmadığı anlaşılmaktadır. Bu nedenle burada da yine $K = 0$ olmalıdır., çünkü yine can güvenliğı açısından bir tehlike söz konusudur.

M	KA_1	KA_2	KA_3	K
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	d
0	1	0	0	1
0	1	0	1	d
0	1	1	0	d
0	1	1	1	d
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	d
1	1	0	0	0
1	1	0	1	d
1	1	1	0	d
1	1	1	1	d

Sistemi betimleyen doğal (kanonik) arpımların toplamı ifadesi, dikkate alınmayanlar da eklenmiş olmak koşuluyla ařağıdaki řekildedir:

$$K(M, KA_1, KA_2, KA_3) = \sum m(1, 2, 4) + \sum d(3, 5, 6, 7, 11, 13, 14, 15)$$



Haritada “d” ler ile gösterilen miniterimler, asla seçkin miniterim olarak işaretlenemez.
Buna göre en küçük çarpımların toplamı ifadesi aşağıdaki şekilde olur.

$$K = M' KA_1 + M' KA_2 + M' KA_3 = M' (KA_1 + KA_2 + KA_3)$$

Zaten bulunan en sade ifadeye bakıldığında, kapının açılması için motorun durması ve asansörün katlardan herhangi birine gelmiş olması gerektiği rahatça anlaşılmaktadır. En sade ifadenin ilkinin gerçekleştirmek için üç VE kapısı, bir VEYA kapısı ve bir DEĞİL kapısı gerekli iken, ikincisinin gerçekleştirilmesi için bir VEYA kapısı, bir VE kapısı ve bir DEĞİL kapısı gerekmektedir.