Cebirsel Yapılar

Ders 6

6-1

İkili İşlemler ve Özellikleri

- Kümelerdeki birleşim ve kesişim işlemleri iki kümeyi birleştirerek üçüncü bir küme ortaya çıkarırken,

  $$g \circ f$$

  bileşke fonksiyonu ***f*** ve ***g*** fonksiyonlarından farklı bir fonksiyon ortaya çıkarmaktadır.
- Diğer örnekler ise iyi bildiğimiz aritmetik işlemler yani toplama, çıkarma, çarpma ve bölme olabilir.
- Bu örneklerin ortak noktası özel bir kümenin elemanlarını birleştirecek kurallar tanımlıyor olmalarıdır.
- Bu konudaki amacımız için ise bu kuralların iki elemanı birleştirirken ortaya çıkacak sonucun da kümenin elemanı olmasını sağlaması gerekmektedir.
- Bu kriterleri sağlayan kurala *ikili işlem* denir.

6-2

İkili İşlemler ve Özellikleri

- Yukarıda verdiğimiz örneklerin ikili işlem olup olmadıkları sorudaki kümeye göre değişir.
- Örneğin, pozitif tamsayılar kümesinde toplama işlemi ikili işlem iken çıkarma işlemi değildir. Çünkü verilen iki pozitif tamsayının çıkarılması sonucunda negatif bir tamsayı elde edilebilir.
- Öte yandan çıkarma işlemi tüm tamsayılar için bir ikili işlemdir.
- Bazı ikili işlemler için elemanların birleştirilme sıraları önemli iken bazıları için önemli olmayabilir.
- Örneğin, toplama işleminde m+n=n+m 'dir fakat m-n ile n-m aynı değildir.
- Bu nedenle, ikili işlem bir kümenin herhangi eleman çiftleri üzerinde değil sıralı ikilileri üzerinde etkilidir diyebiliriz.
- Özet olarak, bir ikili işlem için iki şey gereklidir: bir küme ve bu kümenin elemanlarının herhangi bir sıralı ikilisini birleştirerek sonucun yine kümenin bir elemanı olmasını sağlayan bir kural.

6-3

İkili İşlemler ve Özellikleri

- **Tanım:** Boş olmayan S kümesi üzerinde bir * **ikili işlemi** herhangi iki $x,y \in S$ elemanı birleştirerek $z \in S$ elemanını veren bir kuraldır ve z=x*y şeklinde gösterilir.
- Ayrıca ikili işlemler simge olarak, $\otimes$, °, ×, $\cap$ işaretleri ile de gösterilebilir.
- Tanımdan da anlaşılacağı gibi bir ikili işlem S' nin bir elemanını S' ye ait olan x ve y elemanlarının tüm (x,y) sıralı ikililerine atayan bir fonksiyondur. Bu sıralı ikililerin kümesi tabii ki S × S kartezyen çarpımıdır. Bu da bizi aşağıdaki tanıma götürür:
- **Tanım:** Boş olmayan S kümesi üzerinde bir **ikili işlem**

  $$f: S \times S \rightarrow S$$

  şeklinde bir fonksiyondur. x ve y S kümesinin elemanları ise f(x,y)' yi x*y şeklinde gösteririz.
- x*y' nin S kümesine ait olması gerektiği koşulu ikili işlemin **kapalılık** özelliğidir ve bu koşul sağlanırsa S, * işlemine göre kapalıdır deriz.

6-4

İkili İşlemler ve Özellikleri

- **Örnek 4.1:** Sonlu bir küme üzerindeki bir ikili işlemin sonucu bir tablo ile de gösterilebilir. Örneğin S={a,b,c,d} kümesi üzerinde bir * ikili işlemi tanımlayalım:

*	**a**	**b**	**c**	**d**
a	a	b	c	d
b	d	c	a	b
c	c	b	a	a
d	d	b	c	a

- Tablo 4.1'i yorumlamadaki mantık şu şekildedir: Örneğin b*d işleminin sonucu b ile etiketlendirilmiş satır ve d ile etiketlenmiş sütunun kesişimi ile elde edilir. O halde, b*d=b 'dir. Benzer şekilde c*d=a, d*c=c, c*c=a 'dır.
- Şimdi de birleşme özelliğinin sağlanıp sağlanmadığını inceleyelim. Bu ifadeyi (a*b)*c yani önce a ve b 'yi sonra sonucu c ile birleştirmek şeklinde yorumlayabiliriz. Yada a*(b*c) yani a ' yı b*c işleminin sonucu ile birleştirmek şeklinde de yorumlayabilirdik.
- Bazı ikili işlemler için örneğin reel sayılar üzerindeki çıkarma işleminde iki yorum iki farklı sonuç verir. Birleşme özelliğine sahip değildir.
- Bazıları için ise hiçbir şey değişmez. Bu tip ikili işlemler 'birleşme' özelliğine sahiptir denir.

6-5

İkili İşlemler ve Özellikleri

Tanım: Bir S kümesi üzerindeki * ikili işlemi, tüm x,y,z $\in$S için (x*y)*z=x*(y*z) ise **birleşme (**associative**)** özelliğine sahiptir.

- Birleşme özelliğine sahip olmayan bir ikili işlemden ikiden fazla terim içeren ifadeler için ilk önce hangi elemanların işleme tabi tutulacağını belirtmek amacıyla parantez kullanılmalıdır.
- Hatırlarsak ikilli işlemi (x,y) sıralı ikilisi üzerinde tanımlamıştık. Tanımın haricinde, eğer x ve y bir kümenin elemanları ise x*y ve y*x de kümenin elemanları olmalıdır. Fakat bu işlemlerin sonucu aynı olmayabilir. Toplama işlemi gibi x*y=y*x olan bazı ikili işlemler için değişme özelliğine sahiptir denir.

Tanım: Bir S kümesi üzerindeki * ikili işlemi, tüm x,y$\in$S için x*y=y*x ise **değişme (**commutative**)** özelliğine sahiptir.

- Belli ikili işlemlerde, kümenin herhangi bir elemanı ile birleştirildiğinde bu elemanı değiştiremeyen bir eleman vardır.
- Örneğin reel sayılardaki toplama işleminde sıfır bu özelliğe sahiptir:

  x+0 = 0+x = x.
- Eğer varsa böyle bir elemana **etkisiz eleman** denir.

6-6

İkili İşlemler ve Özellikleri

- **Tanım:** * bir S kümesi üzerinde bir ikili işlem olsun. Tüm $x \in S$ için $x*e=e*x=x$ özelliğine sahip bir $e \in S$ elemanı * işlemi için **etkisiz eleman** olarak adlandırılır.

- Dikkat edilirse, e etkisiz eleman olmak üzere, S kümesindeki tüm x elemanları için $x*e=x$ ve $e*x=x$ eşitliklerinin her ikisi de sağlanması gerekmektedir.
- Tamsayılardaki çıkarma işleminde 0 etkisiz eleman değildir. Çünkü $x-0=x$ fakat $0-x=-x$' tir.
- Aşağıdaki son özellik ise sadece etkisiz elemana sahip ikili işlemlerle ilgilidir.

- **Tanım:** * bir S kümesi üzerinde bir ikili işlem olsun ve bir $e \in S$ etkisiz elemanı olduğunu varsayalım. x, S'in bir elemanıdır dersek, x' in **tersi** bir $y \in S$ elemanıdır öyle ki,

$$x*y=y*x=e.$$

  Bir elemanın tersi tektir ve $y=x^{-1}$ şeklinde yazılır.
- Aynı zamanda $x=y^{-1}$.
- x^{-1} bazen $1/x$ ile karıştırılabilir. $1/x$, sadece eğer S kümesi sıfır hariç reel sayılar kümesi ve * çarpma ikili işlemi ise x' in tersidir.

6-7

İkili İşlemler ve Özellikleri

- **Örnek 4.2:** Tablo 4.1 de verilen ikili işlem birleşim özelliğine sahip değildir zira

$$(b*d)*a = b*a = d \text{ iken}$$
$$b*(d*a) = b*d = b \text{ 'dir.}$$

- Benzer şekilde bu ikili işlemin değişme özelliği de yoktur Çünkü

$$b*a \neq a*b \text{ 'dir.}$$

- Tabloya bakıldığında etkisiz elemanın da olmadığı kolayca görülebilir.

*	**a**	**b**	**c**	**d**
a	a	b	c	d
b	d	c	a	b
c	c	b	a	a
d	d	b	c	a

- İkili işlemde etkisiz eleman olmayabilir fakat eğer varsa bu eleman tektir.

6-8

İkili İşlemler ve Özellikleri

Teorem 4.1: *, S kümesi üzerinde bir ikili işlem olsun. Eğer bir etkisiz eleman varsa bu eleman tektir.

İspat: e_1 ve e_2 S kümesinde * işlemi altında etkisiz elemanlar olsun. e_2 bir etkisiz eleman olduğuna göre,

$$e_1 * e_2 = e_2 * e_1 = e_1.$$

Fakat e_1 de bir etkisiz eleman olduğuna göre

$$e_2 * e_1 = e_1 * e_2 = e_2.$$

Bu durumda açıkça görülüyor ki $e_1 = e_2$. O halde, etkisiz eleman tektir.

İkili İşlemler ve Özellikleri

Teorem 4.2: *, S kümesi üzerinde birleşme özelliğine sahip bir ikili işlem ve *e* bu işlem altında bir etkisiz eleman olsun. Bu durumda bir elemanın tersi eğer varsa tektir.

İspat: $x \in S$ elemanın tersinin y ve z olduğunu düşünelim. O halde,

$$y * x = x * y = e$$
$$z * x = x * z = e.$$

Bu durumda,

$$\begin{aligned} y &= y*e \\ &= y*(x*z) \\ &= (y*x)*z \text{ (birleşme kuralı)} \\ &= e*z \\ &= z. \end{aligned}$$

Böylece x' in tersinin tek olduğunu ispatlamış oluruz.

- Dikkat edilirse bu teoremin ispatında ikili işlemin birleşme özelliğine sahip olması gerekli şarttır.
- Eğer ikili işlem birleşme özelliğine sahip değilse ters elemanın tek olması garanti değildir.

Cebirsel Yapılar

- Bir **cebirsel yapı,** bir veya daha fazla küme ile birlikte bir şekilde küme elemanlarını birleştirebilen bir veya daha fazla işlemden oluşur.
- Belli bir cebrik yapı için önemli olan çoğu özelliğinin içerdiği işlemlerden tahmin edilebilmesidir.
- Bunun anlamı ortak özelliklere sahip cebrik yapıların sınıflara (ailelere) ayrılabileceğidir.
- Verilen bir cebrik yapının hangi belli yapı ailesine ait olduğunu bulabilmek, bu ailenin tüm elemanlarının hangi karakteristik özelliklere sahip olduğu sonucuna varabilmemize imkan verir.
- Yani eğer belli bir yapının 'grup' olduğunu anlayabiliyorsak bu yapının grupların tüm karakteristik özelliklerine sahip olduğunu varsayabiliriz.
- Burada inceleyeceğimiz cebrik yapılar tek bir S kümesi ile birlikte bu kümenin elemanlarını birleştiren tek bir ikili işlemden oluşur.
- Bu tip bir yapıyı iki gerekli kısımdan oluştuğunu belirtmek için (S, *) şeklinde (bir küme ve bu küme üzerinde bir ikili işlem) gösterebiliriz.

Cebirsel Yapılar

Yarı-Gruplar

- İlk cebrik yapı sınıfımız için ikili işlemin sadece birleşme özelliğine sahip olması gerekir. Bu özelliğe sahip cebrik yapılara 'yarı-grup' denir.

Tanım: S, boş olmayan bir küme ve *, S üzerinde tanımlı bir ikili işlem olsun. (S,*) yapısı S üzerinde * işlemi **birleşme** özelliğine sahipse **yarı-grup** tur.

- Eğer işlem hem **birleşme** hem de **değişme** özelliğine sahipse (S, *) yapısı **değişken yarı-grup (Abelyen yarı grup)** adını alır.

Örnek: A sembollerden oluşan boş olmayan bir küme olsun. Böyle bir kümeye **alfabe** denir. Bazı alfabe örnekleri şunlardır:

(a) $A=\{\alpha, \beta, \gamma, \delta, \varphi, \pi\}$

(b) $A=\{a, b, c, d, \ldots, x, y, z\}$

(c) $A=\{\times, +, -, \div, /, \$, \%, \&\}$.

- Bir A alfabesi verilmişse, bu alfabeden sonlu sıralı semboller dizisi tanımlayabiliriz ve buna **kelime (string)** deriz. Kelimenin uzunluğu içerdiği sembol sayısı kadardır.

- O halde, A bir alfabe diyelim ve A üzerindeki tüm kelimelerin kümesi A^* kümesini düşünelim. A^* kümesindeki elemanlar üzerinde bir **ekleme** (concatenation) işlemi tanımlayalım. x ve y, A^* kümesinin iki elemanı ise x ve y 'nin ekleme işlemi $x*y$ şeklinde gösterilir ve x ile y kelimelerinin yan yana yazılmaları ile elde edilir. Örneğin $A=\{a, b, c, d\}$ ise

$$abd*cabc = abdcabc$$

$$baaa*ccbabb = baaaccbabb.$$

- Verilen bir A alfabesi için A^* üzerindeki ekleme işlemi bir ikili işlemdir ve tanımdan da açıkça anlaşılacağı gibi bu işlem birleşme özelliğine sahiptir. Bu nedenle, $(A^*, *)$ yapısı bir yarıgruptur.

6-13

Monoidler

- Yarı-grupların ikili işlemlerindeki tek kısıtlama, çok fazla ilginç özelliğin ortaya çıkmasına yetecek yapıyı vermez. Bu nedenle sıradaki cebrik yapı ailesinde birleşme özelliğine bir şart daha ekleyeceğiz- etkisiz elemanın varlığı. Bu iki özelliğe sahip cebrik yapıya **monoid** denir.

Tanım: Bir **monoid** etkisiz elemana sahip bir $(S, *)$ yarı-gruptur.

- Eğer *, aynı zamanda değişme özelliğine de sahipse monoid, değişken monoid (Abelyen monoid) diye adlandırılır.

Örnek : Önceki örnekte kelimeler üzerindeki ekleme işlemini tanımlamıştık. A^* kümesine boş kelimeyi (empty string) yani hiçbir sembol içermeyen kelimeyi eklediğimizi düşünelim. Boş kelimeyi λ ile gösterirsek, tüm $x \in A^* \cup \{\lambda\}$ için

$$x*\lambda = \lambda *x = x \text{ 'dir.}$$

O halde, $(A^* \cup \{\lambda\}, *)$ yapısı bir monoid olur.

6-14

Gruplar

- Cebrik yapıların tek bir işlem içeren en önemli ve ilginç örneklerinin çoğu, monoidleri tanımlayan iki şarta ek olarak bir üçüncü şartı daha sağlar. Bu şart da kümenin her bir elemanın işleme göre tersinin olduğudur. Bu şartı monoidlerin şartlarına eklersek 'grup' olarak bilinen cebrik yapıyı tanımlamış oluruz.

Tanım: Her bir elemanın tersinin olduğu monoide (S, *) **grup** denir. Yani (S, *) çifti şu üç şartı sağlar:

(G_1) *, S üzerinde birleşme özelliğine sahiptir.

(G_2) bir etkisiz eleman mevcuttur.

(G_3) S' in her bir elemanının tersi mevcuttur.

- Hatırlarsak, birleşme özelliğine sahip bir ikili işlemde ters elemanın tek olduğunu ispatlamıştık.

Örnek 4.5: Bir önceki örnekte tanımladığımız ($A^* \cup \{\lambda\}$, *) monoid bir grup değildir. Boş olmayan bir x kelimesi için λ boş kelime olmak üzere,

$$x*y = y*x = \lambda$$

şartının sağlayan başka bir y kelimesi bulamayız. Bu nedenle, $A^* \cup \{\lambda\}$ kümesinde λ haricinde hiçbir elemanın ekleme işlemi altında bir tersi yoktur.

Gruplar ve Yarı-Gruplar

- Bu konuda daha önce bahsettiğimiz üç cebrik yapı içerisinde en önemli olan grup yapısından bahsedilecektir.
- Bu kısımda ve bundan sonraki kısımlarda belirtilmemiş ikili işlemler içeren ifadeler yazarken * simgesini göz ardı edeceğiz. Sadece yanlış anlamalara imkan verecek iki ikili işlemi birbirinden ayırt etmek için bu sembolü kullanacağız.
- Örneğin x*y yerine xy yazacağız (ancak çarpma işlemi ile karıştırmamalıyız).
- Ayrıca aşağıdaki gibi x' in üslerini tanımlayacağız.
  $n \in Z^+$ olmak üzere $x^n = x*x*\ldots*x$ (n tane) ve
  $n \in Z^-$ olmak üzere $x^n = (x^{-1})^{|n|} = x^{-1}* x^{-1}*x^{-1}*\ldots* x^{-1}$ (n tane)
- Ayrıca etkisiz elemanı da şu şekilde tanımlarız: $x^0=e$ ($xe=ex=x$).
- Herhangi bir (G,*) grubun en belirgin özelliği büyüklüğü yani grubun temelini oluşturan G kümesinin eleman sayısıdır. Buna (G,*) grubunun mertebesi (order) denir.

Tanım: (G,*) grubunun mertebesi G kümesinin kardinalitesidir ve |G| şeklinde gösterilir.

Teorem 5.1: $(G,*)$ bir grup ise sol ve sağ sadeleşme kuralları uygulanabilir yani, $a,x,y \in G$ ise

1. $ax=ay$ ifadesinin anlamı $x=y$ (sol sadeleşme kuralı)
2. $xa=ya$ ifadesinin anlamı $x=y$ (sağ sadeleşme kuralı)

Teorem 5.2: $(G,*)$ bir grupsa ve $a, b \in G$ ise;

(a) $ax=b$ denkleminin $x=a^{-1}b$ şeklinde tek bir çözümü vardır ve

(b) $ya=b$ denkleminin $y=b\,a^{-1}$ şeklinde tek bir çözümü vardır.

İspat: $ax=b$ olsun. Bu denklemin her iki tarafını a^{-1} ile çarparsak:

$$a^{-1}(ax) = a^{-1}b$$
$$(a^{-1}a)x = a^{-1}b$$
$$ex = a^{-1}b$$
$$x = a^{-1}b.$$

Böylece $x=a^{-1}b$ denklemin bir çözümüdür. Bu çözümün tek çözüm olduğunu göstermemiz gerekir. x_1 ve x_2 her ikisi de $ax=b$ 'nin çözümü olsun. O halde,

$$ax_1 = ax_2$$
$$x_1 = x_2$$

Böylece $x=a^{-1}b$ tek çözüm olduğu görülür.

- Bu iki teoremin yararlı bir sonucu sonlu sayıda elemana sahip bir grubun Cayley tablosuna yerleştirilmesidir. İkinci teorem her bir elemanın her satır ve sütunda tam olarak bir kere bulunmasını garantiler.
- **Teorem 5.3:** Eğer $(G,*)$ sonlu bir grupsa, bu grubun Cayley tablosunda G' nin her elemanı her bir satır ve sütunda sadece bir kez yer alır.

6-17

Periyodik (Cyclic) Grupları

- Tablo 5.1'deki Cayley tablosu ile tanımlanmış grubu ele alalım:

*	e	a	b	c
e	e	a	b	c
a	a	b	c	e
b	b	c	e	a
c	c	e	a	b

- Her bir elemanı n bir tamsayı olmak üzere a^n biçiminde yazabileceğimizden bu grup için $a^1=a$, $a^2=b$, $a^3=c$ ve $a^4=e$ 'dir. Verilen herhangi bir eleman için bu gösterim aynı değildir.
- Örneğin, $b = a^2 = a^6 = a^{-2}$ vs. yazabiliriz. Aslında kümenin her bir elemanını a' nın kuvvetleri biçiminde göstermek için sonsuz sayıda yol vardır.
- $\{e,a,b,c\}$ 'nin her elemanı a^n biçiminde yazılabilir ve bu duruma a grubun bir üretecidir (generator) denir.
- Doğal olarak aklımıza diğer başka elemanlar da grubun üreteci midir sorusu aklımıza gelir. c elemanının da bir üreteç olduğunu fakat n çift ise $b^n=e$ ve n tek ise $b^n=b$ olduğundan b 'nin bir üreteç olmadığını söyleyebiliriz.
- En az bir tane üretece sahip gruplara periyodik veya dairesel (Cyclic) denir.

Tanım: $(G,*)$ grubu, n bir tamsayı olmak üzere her bir $g \in G$ için bir $g = a^n \in G$ elemanı mevcutsa periyodik veya dairesel (cyclic) gruptur denir. $(G,*)$ grubu a tarafından üretilmiştir denir ve a $(G,*)$ grubunun üreticidir.

6-18

Örnek 5.1: (Z,+) grubunun periyodic (cyclic) olduğunu ve üretecinin 1 olduğunu gösteriniz.

Çözüm: Etkisiz eleman 0, ve 1 elemanının tersi -1 'dir.

$n \in Z$ elemanı için $n>0$ olmak üzere;

$n = 1+1+\cdots+1$ (n tane)

$= n \cdot 1$

$n<0$ ise;

$n = (-1)+(-1)+\cdots+(-1)$ (n tane)

$= |n|\,(-1)$

$= n \cdot 1.$

$n=0$ ise;

$n = 0 \cdot 1 = n \cdot 1$

Böylece (Z,+) halka grubudur ve 1 bir üreteçtir.

6-19

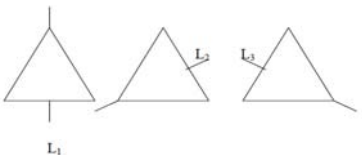
Dihedral Gruplar

- Yandaki şekilde köşeleri 1,2, ve 3 ile numaralandırılmış eşkenar üçgene göz atalım.

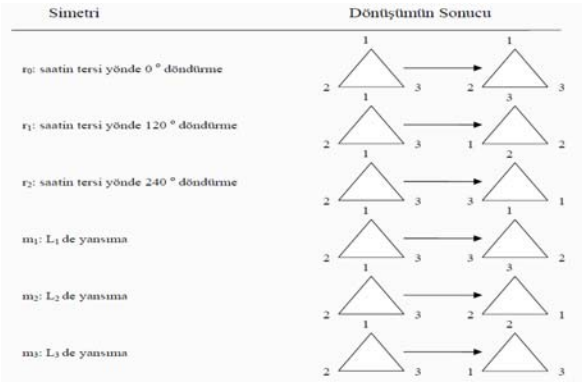
- Şimdi bu üçgenin köşelerinin yerlerinin değişimine yol açacak olası dönüşümlerini düşünelim. Örneğin, bu üçgen saat yönünün tersine merkezinden 120^o döndürülürse,

- Üçgenin en tepe noktası (1) ile bu noktanın karşısındaki kenarın orta noktasını birleştiren doğrudan yansıması ise;

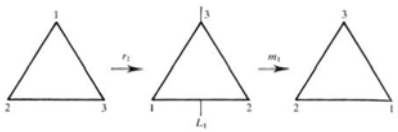
- Bütün bu dönüşümlerin kümesine eşkenar üçgenin simetrileri kümesi denir. Bu tip, üç tane rotasyonları içeren, üç tane de aşağıdaki L_1, L_2 ve L_3 doğrularında yansımaları içeren altı tane simetri vardır.



6-20



- T={r_0, r_1, r_2, m_1, m_2, m_3} kümesini ve * işlemini düşünelim ve a*b=ab 'nin anlamı 'a dönüşümünden sonra b dönüşümünü uygula' olsun. Bu nedenle, r_1*m_1 'in anlamı 'üçgeni saat yönünün tersine 120° çevir ve sonucu L_1 üzerinde yansıt'.



- Ortaya çıkan sonuç tek bir m_2 dönüşümüne eşittir ve $r_1m_1=m_2$ yazılabilir. * işlemi değişme özelliğine sahip değildir çünkü $m_1r_1=m_3$ 'tür.

6-21

- T kümesinin * işlemi altındaki Cayley tablosu Tabloda gösterilmiştir

*	r_0	r_1	r_2	m_1	m_2	m_3
r_0	r_0	r_1	r_2	m_1	m_2	m_3
r_1	r_1	r_2	r_0	m_2	m_3	m_1
r_2	r_2	r_0	r_1	m_3	m_1	m_2
m_1	m_1	m_3	m_2	r_0	r_2	r_1
m_2	m_2	m_1	m_3	r_1	r_0	r_2
m_3	m_3	m_2	m_1	r_2	r_1	r_0

- Açıktır ki *, T üzerinde bir ikili işlemdir ve (T,*) 'in değişme özelliğine sahip olmayan bir grup olduğunu gösterebiliriz.
- Etkisiz eleman r_0 'dır ve her bir elemanın tersi vardır.
- Öte yandan her dönüşüm bir fonksiyon olarak düşünülürse * işleminin birleşme özelliğine sahip olduğu kolayca gösterilebilir.
- (T,*) grubu genellikle D_3 şeklinde ifade edilir ve eşkenar üçgenin simetri grupları veya 3. dereceden dihedral grup şeklinde isimlendirilir.
- Benzer simetri grupları tüm düzgün çokgenler için de geçerlidir. n. dereceden dihedral grup n kenarlı düzgün çokgenin simetri grubudur. 2n tane elemanı vardır ve D_n şeklinde gösterilir.

6-22

Permutasyon Grupları

Tanım: S boş olmayan bir küme olsun. S' in bir permutasyonu S' ten S' e bir bijeksiyondur.

- Belli bir bijeksiyonu tanımlamak için kullanılan yol genellikle S' in tüm elemanlarının eşleşmelerinin etkilerini göstermektir.
- Örneğin, S={1,2,3,4} ise şu şekilde bir p_1 bijeksiyonu tanımlayabiliriz.

$$p_1(1)=2 \quad p_1(2)=4 \quad p_1(3)=3 \quad p_1(4)=1.$$

p_1'i göstermenin daha uygun yolu, ilk satırı S 'in elemanlarından ve ikinci satırı bunlara karşılık gelen görüntülerinden oluşan bir dizi kullanmaktır. p_1 bijeksiyonu için şunu yazabiliriz:

$$p_1=\begin{vmatrix} 1 & 2 & 3 & 4 \\ p_1(1) & p_1(2) & p_1(3) & p_1(4) \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{vmatrix}.$$

- İlk satırda listelenen S' in elemanlarının sırası önemli değildir. Önemli olan her bir elemanın altındakinin uygun bijeksiyondaki görüntüsünün olmasıdır.Yani p_1' i şu şeklide de yazabiliriz:

$$p_1=\begin{vmatrix} 2 & 1 & 4 & 3 \\ 4 & 2 & 1 & 3 \end{vmatrix}$$

6-23

Permutasyon Grupları

Şimdi de A={1,2,3} kümesini düşünelim ve S_3 A' nın tüm permutasyonlarının kümesi olsun.(S_3 notasyonunu kullanmamızın sebebi kümenin 3 elemanlı olduğunu belirtmektir). S_3'ün aşağıdaki gibi 6 elemanlı p_1, p_2, …, p_6 olduğunu tahmin etmek zor değildir.

$$p_1=\begin{vmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{vmatrix} \quad p_2=\begin{vmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{vmatrix} \quad p_3=\begin{vmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{vmatrix}$$

$$p_4=\begin{vmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{vmatrix} \quad p_5=\begin{vmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{vmatrix} \quad p_6=\begin{vmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{vmatrix}$$

S_3 üzerinde tanımlanabilecek doğal bir ikili işlem vardır ki bu fonksiyonların birleşmesidir. Bu nedenle, p_ip_j (p_i, $p_j \in S_3$) p_i ve p_j bijeksiyonlarının bileşkesi anlamına gelir.

İşlem açıkça görüldüğü gibi bir ikili işlemdir. Çünkü S üzerindeki bijeksiyonların bileşkesi yine S üzerinde bir bijeksiyondur. Örneğin p_3p_5 'i düşünelim. Dizi biçiminde şöyle yazabiliriz:

$$p_3p_5=\begin{vmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{vmatrix} \begin{vmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{vmatrix} \quad p_3p_5=\begin{vmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{vmatrix} \begin{vmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 1 & ? & ? \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{vmatrix}$$

Bu dizi p_4'ü temsil eden dizidir, o halde $p_3p_5=p_4$ yazabiliriz.

6-24

- $(S_3,*)$ için Cayley tablosu; aşağıdaki gibidir.

*	p_1	p_2	p_3	p_4	p_5	p_6
p_1	p_1	p_2	p_3	p_4	p_5	p_6
p_2	p_2	p_3	p_1	p_5	p_6	p_4
p_3	p_3	p_1	p_2	p_6	p_4	p_5
p_4	p_4	p_6	p_5	p_1	p_3	p_2
p_5	p_5	p_4	p_6	p_2	p_1	p_3
p_6	p_6	p_5	p_4	p_3	p_2	p_1

- $(S_3,*)$ yapısının değişme özelliğine sahip olmayan bir grup olduğunu kolayca doğrulayabiliriz. Birleşme özelliği ise fonksiyonların bileşkesinin birleşme özelliğinden gelir.
- Eğer S={1,2,…,n} ise bu durumda |S|=n ve permutasyonlar kümesi S_n
  $n(n-1)(n-2)\ldots1=n!$
  elemanlıdır.
- Bu nedenle S' ten S' e bijeksiyon tanımlarken S' in ilk elemanı, S'in herhangi bir elemanıyla; ikinci elemanı S'in diğer n-1 elemanında biriyle vs. eşleştirilebilir.
- Bu da bize toplamda n! olası bijeksiyon verir.
- Herhangi bir pozitif n tamsayısı için * bijeksiyonların bileşkesini ifade etmek üzere, (Sn,*) **n. dereceden simetrik grup** şeklinde adlandırılan bir gruptur.

6-25

Morfizm ve Grup Kodları

İsomorfizm (Isomorphism)

- Daha önceki konularda üç önemli grup ailesi (periyodik grupları, dihedrak gruplar ve permutasyon grupları) örneklerini incelemiştik. 3 elemanlı bir kümenin D_3 dihedral grubu ve S_3 permutasyon grubu için Cayley tablosu, aşağıdaki gibidir.

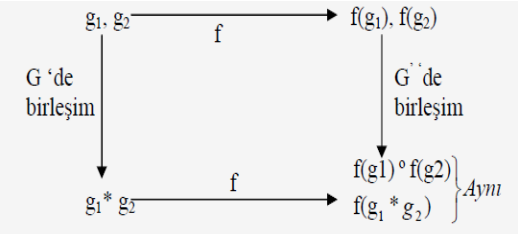
*	r_0	r_1	r_2	m_1	m_2	m_3
r_0	r_0	r_1	r_2	m_1	m_2	m_3
r_1	r_1	r_2	r_0	m_2	m_3	m_1
r_2	r_2	r_0	r_1	m_3	m_1	m_3
m_1	m_1	m_3	m_2	r_0	r_2	r_1
m_2	m_2	m_1	m_3	r_1	r_0	r_2
m_3	m_3	m_2	m_1	r_2	r_1	r_0

*	p_1	p_2	p_3	p_4	p_5	p_6
p_1	p_1	p_2	p_3	p_4	p_5	p_6
p_2	p_2	p_3	p_1	p_5	p_6	p_4
p_3	p_3	p_1	p_2	p_6	p_4	p_5
p_4	p_4	p_6	p_5	p_1	p_3	p_2
p_5	p_5	p_4	p_6	p_2	p_1	p_3
p_6	p_6	p_5	p_4	p_3	p_2	p_1

- Yukarıdaki tabloları karşılaştırırsak ilginç bir şekilde her iki tablonun isimlendirmeler dışında aynı olduğunu görürüz. İlk tabloda r_2 'nin olduğu yerlerde ikinci tabloda p_3; m_3 'ün olduğu yerlerde p_6 vardır vs. p_1, p_2, …, p_6 dönüşümleri yerine sırasıyla r_0, r_1, r_2, m_1, m_2, m_3 kullanırsak iki tablo aynı olur.
- İki sonlu grup bu şekilde ilişkilendirilmiş ise 'izomorfik' olarak adlandırılırlar.
- İzomorfik olmak grupların aynı olması demek değildir. Örneğimizde iki küme elemanları nasıl etiketlendirilirse etiketlendirilsin farklıdır ve ikili işlemleri aynı değildir.
- Öte yandan, izomorfik gruplar arasında çok yakın bir ilişki vardır öyle ki elemanları aynı olmasa da yapıları aynıdır ve bir şekilde bu ilişkiyi matematiksel olarak tanımlamamız gerekir.

6-26

- Cayley tablolarının isimlendirilme dışında aynı olması demek, D_3' ün elemanları ve S_3 'ün elemanları arasında bire-bir eşleşme olması demektir.
- Bu bire-bir eşleşme grup yapısını muhafaza etme özelliğine sahip bir bijective fonksiyondur.
- Bu tip bir fonksiyona *izomorfizm* denir.
- Daha ciddi bir ifadeyle: $(G,*)$ ve (G',o) şeklinde verilmiş iki grup varsa bir izomorfizm bijective bir $f:G\rightarrow G'$ fonksiyonudur öyle ki; $g_1 * g_2$ 'nin görüntüsü G' kümesinin elemanıdır ve o işleminin g_1 ve g_2 'nin görüntülerine uygulanması işleminin sonucudur.
- Aşağıdaki şekilde bu düşünceler özetlenmiştir.



- **Tanım:** $(G,*)$ grubundan (G',o) grubuna bir **izomorfizm** bir bijective $f:G\rightarrow G'$ fonksiyonudur ve tüm $g_1, g_2 \in G$ için

  $f(g_1*g_2)= f(g_1)$ o $f(g_2)$ 'dır.

- Eğer böyle bir fonksiyon varsa $(G,*)$ grubu (G',o) grubu ile **izomorfik**tir deriz ve $(G,*) \cong (G',o)$ şeklinde yazarız.

- D_3 'ten S_3' e bir izomorfizm

  f: $r_0\rightarrow p_1$, $r_1\rightarrow p_2$, $r_2\rightarrow p_3$, $m_1\rightarrow p_4$, $m_2\rightarrow p_5$, $m_3\rightarrow p_6$

  şeklinde tanımlanır.

- Daha genelleştirip n elemanlı bir kümenin tüm permutasyonlarının grubu n. derece bir dihedral ile izomorfik midir diye sorarsak cevap hayırdır.
- Çünkü, n>3 için bu grupların mertebeleri eşit değildir.
- O halde, $D_n\rightarrow S_n$ şeklinde bir bijeksiyon mevcut değildir. n derece bir dihedral için

  $|D_n|=2n$ iken $|S_n|=n!$ 'dir.

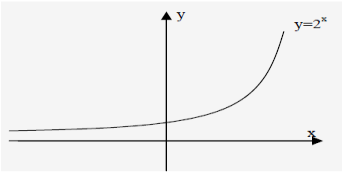
Örnek 5.2: (R, +) ve (R^+, ×) gruplarını düşünelim. **f**: R→R^+ ve **f**(x)=2^x fonksiyonunun (R,+)' dan (R^+, ×)' ye bir izomorfizm tanımladığını gösteriniz.

Çözüm: İki şeyi göstermemiz gerekir:

a. **f**' in bijeksiyon olduğunu,

b. x,y ∈ R ise **f**(x+y)=**f**(x) × **f**(y) olduğunu.

f 'in bijeksiyon olduğunu doğrulamanın en kolay yolu y=**f**(x) 'in grafiğini çizmektir. Bu grafik şekilde gösterilmiştir.



y- ekseninin pozitif kısmındaki tüm yatay doğrular grafiği tam olarak bir yerde kesiğine göre f bir bijeksiyondur. Ayrıca,

$$\mathbf{f}(x+y)=2^{x+y}$$
$$= 2^x.2^y$$
$$= \mathbf{f}(x) \times \mathbf{f}(y)$$

O halde f' in (R,+) 'dan (R^+, ×)' ya bir izomorfizm olduğunu göstermiş olduk.

6-29

- İki grubun izomorfik olup olmadığını belirlemek için birçok deneme yanılma yapmak gerekir ve bu da özellikle grupların mertebeleri büyükse çok zaman alır.
- Bu zaman izomorfizmin bilinen özellikleri kullanılarak azaltılabilir. Bu özelliklerin bir kısmı aşağıdaki teoremde listelenmiştir.

- **Teorem 5.3:** Eğer **f**:$G_1 \rightarrow G_2$ (G_1,*) ve (G_2,o) grupları arasında bir izomorfizm ise;
  1) e, (G_1,*) 'da bir etkisiz eleman ise **f**(e) (G_2,o) 'da bir etkisiz elemandır.
  2) (G_1, *), sadece ve sadece (G_2,o) değişme özelliğine sahip bir grupsa değişme özelliğine sahiptir.
  3) a^{-1} (G_1, *)' da a' nın tersi ise **f**(a^{-1}), **f**(a)' nın (G_2, o) 'da tersidir. Yani, **f**(a^{-1})=[**f**(a)] $^{-1}$.
  4) f^{-1}: $G_2 \rightarrow G_1$ ters fonksiyonu (G_2,o)'dan (G_1,*)' ya bir izomorfizm tanımlar.
  5) (G_1,*) sadece ve sadece (G_2,o) cyclic ise cyclictir.
  6) Eğer $a \in G_1$ ise |a|=|**f**(a)|.

- Bu özelliklerin izomorfik gruplara uygulanabileceğini ve bu özelliklerden birinin olmaması durumunda sorulan iki grubun izomorfik olamayacağını kanıtlamak çok zor değildir.
- Bu nedenle, iki grubun izomorfik olmadığını göstermek için bir grup için geçerli diğeri için geçersiz olan bir özellik bulmak gerekir.
- Öte yandan iki grubun izomorfik olduğunu göstermek için ortak özelliklere sahip olduklarını göstermek yeterli değildir.

6-30

- **İzomorfizm Prensibi**

  İki grubun izomorfik olduğunu göstermek için birinden diğerine bir izomorfizm bulunmalıdır; iki grubun izomorfik olmadığını göstermek için bir grubun sahip olduğu diğerinin sahip olmadığı bir grup özelliği bulunmalıdır.

Örnek 5.3: D_3 ve $(Z/6, +_6)$ grupları izomorfik midir?

Çözüm: Teorem 5.2' deki 2. maddeyi uygularsak bu iki grubun izomorfik olmadığını söyleyebiliriz. Çünkü $(Z/6, +_6)$ değişme özelliğine sahipken D_3 değildir.

6-31

Morfizmler

- $(G,*)$ ve (G',o) gruplarının izomorfik olması için bijective ve ayrıca grubun yapısını devam ettiren bir $\mathbf{f}:G\rightarrow G'$ fonksiyonu tanımlayabilmemiz gerekir.
- Bijective koşulunu kaldırırsak 'morfizm' olarak adlandırdığımız daha genel bir yapı-koruyan fonksiyon tanımlarız. Morfizmler sadece grup çiftleri arasında değil aynı zamanda herhangi iki cebrik yapı arasında da tanımlanabilir.
- **Tanım:** $(A,*)$ ve (B,o) şeklinde iki cebrik yapı verilmişse, $(A,*)$' dan (B,o)' ya bir morfizm $\mathbf{f}:A\rightarrow B$ fonksiyonudur öyle ki tüm $a_1, a_2\in A$ için;

  $$\mathbf{f}(a_1*a_2)= \mathbf{f}(a_1)\ o\ \mathbf{f}(a_2)$$

- Bir morfizm surjective olmak zorunda değildir o halde B 'de, A' daki herhangi bir elemanın görüntüsü olmayan elemanlar olabilir.
- Eğer surjective ise bu morfizm **epimorfizm** olarak adlandırılır.
- Benzer şekilde bir morfizm injective olmak zorunda değildir o halde B 'de, A' daki birden fazla elemanın görüntüsü olan elemanlar olabilir.
- İnjective bir morfizme **monomorfizm** denir.
- Bir izomorfizm hem surjective hem de injective olan bir morfizmdir. $(A,*)$ ve (B,o) cebrik yapıları arasındaki morfizmler ile ilgili önemli olan, * işlemi altında A'nın birçok özelliğinin, o işlemi altında görüntü kümesi $\mathbf{f}(A)$ 'da korunmasıdır.
- **Teorem 5.4:** $(A,*)$ ve (B, o) cebrik yapılar ve $\mathbf{f}:A\rightarrow B$ bir morfizm olsun.
  (a) $(A,*)$ bir yarı-grup ise $(\mathbf{f}(A),o)$ da yarı-grup tur.
  (b) $(A,*)$ bir monoid ise $(\mathbf{f}(A),o)$ da monoid dir.
  (c) $(A,*)$ bir grup ise $(\mathbf{f}(A),o)$ da grup tur.

6-32

Teorem 5.5: (A,*) ve (B, o) cebrik yapılar ve **f**:A→B, (A,*)' dan (B,o)' ya bir morfizm olsun. Bu durumda,
(1) e, (A,*) 'da bir etkisiz eleman ise **f**(e), (**f**(A),o) 'da bir etkisiz elemandır.
(2) (A,*) değişme özelliğine sahipse (**f**(A),o) da değişme özelliğine sahiptir.
(3) a^{-1} (A,*)' da a' nın tersi ise **f**(a^{-1}), **f**(a)' nın (**f**(A), o) 'da tersidir. Yani, **f**(a^{-1})=[**f**(a)] $^{-1}$.
(4) (A,*) bir periyodik grubu ise (**f**(A),o) da öyledir.

- (f(A),o) yapısı (A,*)' ın morfik görüntüsü olarak adlandırılır.
- Eğer morfizm surjective ise (f(A),o), yukarıdakilerde (B,o) ile yer değiştirilebilir.

Örnek 5.4: (Z,+) grubunu düşünelim. **f**:Z→Z, **f**(x)=2x şeklinde tanımlanmış olsun. **f** fonksiyonunun (Z,+)'dan (Z,+)'ya bir morfizm olduğunu ispatlayınız.

Çözüm: Burada tüm x,y∈Z için **f**(x+y)=**f**(x)+**f**(y) olduğunu göstermeliyiz.

f(x+y) = 2(x+y)
= 2x+2y
=**f**(x)+**f**(y).

Böylece **f**' in bir morfizm olduğunu kanıtlamış olduk.
Fonksiyon injectivedir fakat surjective değildir. Görüntü kümesi çift tamsayılar kümesi (sıfır dahil) E 'dir ve Teorem 5.4(c) 'nin sonucunu doğrulayabiliriz. E kümesi toplama işlemi altında bir gruptur.

6-33

Grup Kodları

- Modern teknolojinin birçok uygulaması bir noktadan diğerine veri iletimini içerir. Bu iki nokta, bir bilgisayarın hafızasının bir yerinden başka yerine olduğu gibi kısmen yakın mesafede de olabilir, uydu haberleşmelerindeki gibi binlerce kilometre uzakta da olabilir. Her iki durumda da sistemin gerekli özellikleri aynıdır. Verinin iletildiği bir iletişim kanalı vardır ve kanalın bir ucundan alınan veri ile diğer ucundaki veri gönderilen veri aynıdır.

- Amaçlarımız doğrultusunda konuyla ilgili tüm veriyi her biri 0 veya 1 olan basamaklardan oluşan stringler ({0,1} alfabesinden oluşan kelimeler şeklinde düşünebiliriz. Bu tip kelimelere ikili kelimeler (**binary words**) ve bunların basamaklarına **bit** denir.

- Öte yandan, her ne kadar iletim sistemimizin tamamen güvenli olmasını istesek de zaman zaman bazı hataların olması, dış kaynaklardan gürültü gelmesi kaçınılmazdır.
- Bunlar iletimde hataya sebep olur ve gönderilen kelimenin alınan kelime ile aynı olmamasına neden olur.
- Bu nedenle, alınan kelimenin hatalı olduğunu anlayabilmek ve mümkünse gönderilen asıl kelimeyi belirlemek çok önemlidir.
- Asıl kelime belirlenemese bile en azından hatanın tespit edilmesi verinin tekrar istenmesine olanak sağlar.

6-34

- İletim hataları konusunda şu varsayımları yapmamız yararlı olur:
  a) İletilen verinin bir veya daha fazla bitinde 1'in 0' a veya 0' ın 1' e dönüşmesi şeklinde oluşurlar
  b) 1' in 0' a dönüşümü ile 0' ın 1 'e dönüşümü ihtimali aynıdır.
  c) Ayrı bitlerde oluşan hatalar birbirinden bağımsız oluşur.
  d) İletilen kelimeyi oluşturan bitlerin her birinde hata olma oranı aynıdır.
  e) n<m için n tane hatanın olma olasılığı m tane hatanın olma olasılığından daha fazladır o halde, yanlış iletilen bir kelime için hata sayısının bir olma ihtimali en yüksektir.

6-35

- Şimdi hata bulma ve düzeltmenin gerekli özelliklerini açıklamak için birkaç örneğe bakalım.
- Bir iletim kanalından iletilen verilerin, uzunluğu 3 olan ikili kelimeler kümesinin tüm elemanları olduğunu düşünelim. Bu kümeyi B_3 ile gösteririz ve

$$B_3=\{000, 001, 010, 011, 100, 101, 110, 111\}.$$

- 010 kelimesinin iletildiğini ve 3. basamakta hata olduğunu, böylece 011 kelimesinin alındığını düşünelim.
- Bu hatayı tespit edebilmenin imkanı yoktur zira 011 de bu kümenin elemanıdır ve alma ihtimali olduğumuz kelimelerden biridir.
- Eğer hatayı tespit edemezsek bunu düzeltme imkanımız da yoktur.
- Bu örnek hatayı tespit edebilmek için gerekli bir özelliği vurgulamaktadır: yanlış iletilen kelime, almayı beklediğimiz kelimeler kümesine dahil olmamalıdır.

- Yukarıdaki örnekteki kelimeler bir birine çok yakındır. Herhangi bir hata kümenin başka bir elemanına dönüşür.
- Bunun yerine iletilen kelimelerin {111, 100, 001, 010} kümesinin elemanları olduğunu düşünelim. Bu durumda 010 kelimesinin 3. basamağında hata meydana gelirse, 011 kelimesi alınacaktır ve bu kelime almayı beklediğimiz bir kelime değildir.
- Öte yandan, hata meydana geldiğini bilsek de hatanın nerede olduğunu bulamayız.
- Tek hatanın olma ihtimalinin en yüksek olduğunu kabul edersek iletilen kelime büyük ihtimalle 111, 001 veya 010 'dur.
- Ancak unutmamak gerekir ki; çift hata tespit edilemez zira bir kelimenin herhangi iki basamağında meydana gelecek hata kümenin başka bir elemanına dönüşür.

6-36

- {111, 100, 001, 010} kümesinin elemanları hala birbirine çok yakındır. Sadece {000, 111} kümesinin elemanlarını ilettiğimizi düşünelim. Bu durumda bir veya iki basamaktaki hatalar tespit edilebilir ve tek bir hata olmuşsa düzeltilebilir.
- Örneğin, 011 kelimesini almışsak ona en yakın 111 kelimesini almamız gerektiğini anlarız. Öte yandan çift hatalar düzeltilemeyecektir.
- Eğer 000 kelimesinde çift hata meydana gelir ve 011 alınırsa, tek hata meydana geldiğini varsayıp esas gönderilenin 111 olduğunu zannedebiliriz.
- Bu kelime kümesi için iki tane hatayı tespit edebilir, fakat düzeltemeyiz.
- Bu örnekler hata tespitinin hata düzeltmekten daha kolay olduğunu gösterir.
- Öte yandan, ikisi de olası iletilecek kelimeler kümesindeki kelimelerin birbirinden farklılığına bağlıdır.

Tanım: x ve y, n uzunluğunda ikili kelimeler olsun. x ve y arasındaki **uzaklık** (**Hamming uzaklığı**) d(x,y) şeklinde gösterilir ve x ve y 'nin farklılaştığı basamak sayısına eşittir.

- Örneğin, x=001101 ve y=111110 ise iki kelime birinci, ikinci, beşinci ve altıncı bitlerde farklılaşır. Bu nedenle d(x,y)=4 'tür.

6-37

- n uzunluğundaki x,y ve z için aşağıdaki uzaklık özelliklerini göstermek çok zor değildir.
  (a) $d(x,y) \geq 0$
  (b) $d(x,y)=0$ sadece ve sadece x=y ise
  (c) $d(x,y)= d(y,x)$
  (d) $d(x,z) \leq d(x,y)+ d(y,z)$
- X kümesi üzerinde bu özelliklere sahip herhangi bir
  $d: X \times X \rightarrow R^{+} \cup \{0\}$
  fonksiyonuna **metrik** adı verilir.
- Bu nedenle uzaklık B^n kümesi üzerinde bir metriktir.
- Başarılı bir hata tespiti ve düzeltimi için olası iletilecek kelimeler kümesindeki farklı kelimeler arasındaki uzaklığın olabildiğince çok olması makbuldür.

6-38

- Pratikte hataların tespiti ve düzeltimi, kelimeleri iletimden önce kodlayarak gerçekleştirilir.
- Genellikle bu kelimenin sonuna bir veya daha fazla bit ekleyerek yapılır. Bunlara kontrol **basamağı** (check digits) denir ve alınan kelimenin bir kısmının veya tamamının doğruluğunu kontrol etmek için kullanılırlar.
- Böylece iletilen herhangi *n* uzunluğunda bir ikili kelime gönderilecek bilgiyi taşıyan bilgi biti denilen *m* basamak ile hata tespiti ve düzeltilmesine yarayan r=n-m kontrol basamağından oluşur.
- B^n, n uzunluğundaki tüm ikili kelimelerin kümesini temsil ediyor dersek, kodlama mekanizmasını E: $B^m \rightarrow B^n$ şeklinde bir fonksiyon olarak görebiliriz. Böyle bir fonksiyona **encoding fonksiyonu**, bu fonksiyonun görüntü kümesinin elemanlarına da **codewords** denir.
- Her codeword B^m içinde tek bir kelimeye karşılık gelmek zorunda olduğundan bir encoding fonksiyonu injective olmalıdır.
- Her bir encoding fonksiyonu için E(x)=y olmak üzere bir codeword 'ü ($y \in B^n$) $x \in B^m$ 'a eşleştiren bir D: $B^n \rightarrow B^m \cup$ {'hata'} decoding fonksiyonu vardır. m<n olduğundan, codeword' ler kümesi B^n' in öz alt kümesidir, o halde D' nin tanım kümesinin codeword olmayan elemanları da vardır.
- Eğer bu kategoriye düşen bir w1 kelimesi alınmışsa, D(w1)=D(w) öyle ki w, w1 codeword' üne en yakın yani en az sayıda bitte farklılaşan codeword' tür. Buna 'en yakın komşu kodlaması' denir.
- Eğer en yakın komşu tek değilse D(w1)='hata' diyebiliriz.

6-39

- Bir E: $B^m \rightarrow B^n$ encoding fonksiyonu ve bir D: $B^n \rightarrow B^m \cup$ {'hata'} decoding fonksiyonu içeren bir coding/encoding prosedürüne **(m,n) blok kodu** denir.
- En basit encoding fonksiyonu codeword' teki 1' lerin sayısını çift sayı yapacak şekilde seçilen bir biti kelimenin sonuna ekler. Böyle bir koda çift parite kontrol kodu (even parity check code) denir.
- Encoding fonksiyonu E: $B^m \rightarrow B^{m+1}$ şeklindedir ve örneğin eğer m=4 ise E(0011)=00110 ve E(1000)=10001.
- Çift parite kontrol kodu için tek kontrol basamağı kullanılırsa, bir hata tespit edilebilir çünkü bu durumda tek sayıda bir vardır.
- Öte yandan, hatalar düzeltilemez çünkü hatanın nerede olduğunu söylemek mümkün değildir.

- İçinde k' nın herhangi bir kombinasyonu veya daha az hata tespit edilebilecek bir kod, **k-hata tespiti** (k-error detecting);
- İçinde k' nın herhangi bir kombinasyonu veya daha az hata düzeltilebilecek bir kod, **k-hata düzeltimi** (k-error correcting) olarak adlandırılır.
- Çift parite kontrol kodu 1-hata tespiti ve 0-hata düzeltimidir.

6-40

- Hataların tespit edilebilmesi ve düzeltilebilmesinin codewordler arasındaki uzaklığa bağlı olduğunu görmüştük.
- Kontrol basamakları içeren kodlar için her bir codeword çifti arasındaki uzaklık aynı olmak zorunda değildir o halde, kodun hata bulma ve düzeltme kapasitesini belirleyen faktör codeword çiftleri arasındaki uzaklığın minimumudur.

- Bir kodun **minimum uzaklığı** ayrı codeword çiftleri arasındaki tüm uzaklıkların minimumu olarak tanımlanır.

Teorem 5.6: Bir kod sadece ve sadece minimum uzaklığı en az k+1 ise k-hata tespitidir.

İspat: Bir codeword' te herhangi sayıdaki hata, codeword başka bir codeword' e dönüşmemiş ise tespit edilebilir. Eğer codeword' ler arasındaki minimum uzaklık k+1 ise k+1' den daha az hata yeni bir codeword' e yol açmaz ve tespit edilir. Bundan dolayı k veya daha az hata tespit edilebilir ve böylece kod k-hata tespitidir.

- **Teorem 5.7:** Bir kod sadece ve sadece minimum uzaklığı en az 2k+1 ise k-hata düzeltimidir.

- **Örnek 5.5:** Aşağıdaki gibi tanımlanan $E:B^2 \rightarrow B^6$ encoding fonksiyonunu düşünelim.
  E(00)=001000 E(01)=010100
  E(10)=100010 E(11)=110001.
  Bu kodun kaç tane hata tespiti ve düzeltimi yapabileceğini bulunuz.

- **Çözüm:** Codeword çiftleri arasındaki uzaklıklar aşağıdaki gibidir.
  d(001000, 010100)=3 d(001000, 100010)=3
  d(001000, 110001)=4 d(010100, 100010)=4
  d(010100, 110001)=3 d(100010, 110001)=3
  Minimum uzaklık üçtür o halde, kod k+1=3 olmak üzere 2-hata tespitidir. 2k+1=3 olduğundan kod 1-hata düzeltimidir.

- Grupların kodlama teorisindeki önemini vurgulamak için n bit kelimeler kümesi B^n üzerinde bir ikili işlem tanımlamamız gerekir.

- **Tanım:** x ve y n uzunluğunda codeword' ler öyle ki, x' in i. basamağı x_i ve y 'nin i. basamağı y_i olsun. x ve y 'nin **toplamı** $x \oplus y$ şeklinde gösterilir ve $+_2$ toplam mod 2 olmak üzere, i. basamağı $x_i +_2 y_i$ olan n bit kelimedir.
-
- Bu nedenle iki codeword' ün toplamı gereken bitlere toplam mod 2 işlemi uygulanması ile elde edilir. Örneğin,

  $1011001 \oplus 1000111 = 0011110$ ve

  $111001 \oplus 110011 = 001010$.
- **Tanım:** x kelimesinin ağırlığı(weight) w(x) ile gösterilir ve içerdiği birlerin sayısıdır.
- Örneğin, w(101101)=4 ve w(011110111)=7.
  İki n bit ikili kelime x ve y arasındaki mesafe şöyle bulunur:

  $$d(x,y)=w(x \oplus y).$$
- Codeword 'ler kümesi $\oplus$ işlemi altında bir grup olan koda **grup kodu** denir. B^n kümesinin bu işlem altında bir grup olduğunu göstermek çok kolaydır. Öte yandan, bu çok yararlı bir codeword kümesi değildir.
- Çünkü, daha önce de bahsettiğimiz gibi kelimeler birbirine çok yakındır. Hata bulma ve düzeltme kapasitesinin minimum uzaklık ile ilgili olduğunu göstermiştik.

6-43

- Herhangi bir kod için minimum uzaklığı bulmak tüm olası codeword çiftleri arasındaki uzaklıkları bulmayı içerir ve codeword' lerin sayısı çok büyükse bu çok zahmetli bir iştir. Öte yandan, bir grup kodu için minimum uzaklığın tüm sıfır-olmayan codeword' lerin minimum ağırlığına eşit olduğunu gösterebiliriz.
- **Teorem 5.8:** Bir grup kodunun minimum uzaklığı, tüm sıfır-olmayan codeword' lerin minimum ağırlığına eşittir.
- Kontrol basamakları ekleyerek kelimeyi kodlayan bir $E:B^m \rightarrow B^n$ (n>m) encoding fonksiyonu içerisinde 1 ve 0 lar bulunan G gibi m×n tipindeki matrisler kullanılarak kolayca tanımlanabilir.
- Böyle matrisler kodlar için üreteç matrisi olarak adlandırılır. m bitlik bir kelimeyi encode etmek için kelimenin 1×m lik bir matris olduğunu görürüz ve bu satır matris ile G matrisini çarpar ve sonucun 2 modülüne göre değerlendiririz.
- Sistematik bir kod için codeword'ün ilk m bitinin encode edilecek kelimenin m biti ile aynı olmasını talep ederiz. Bundan dolayı G matrisinin ilk m sütunu birim matris I_m ile aynı olması gerekir.

6-44

Örnek:

- E:$B^3 \rightarrow B^6$, her $x \in G$ için $E(x)=xG$

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

şeklinde tanımlanan encoding fonksiyonunu göz önüne alalım.

$$E(011) = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

- 011 codeword 011101 olarak encode edilir. Diğer bir örnekte;

$$E(100) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

6-45

- En genel halde x_1, x_2, x_3 rakamlarından oluşan üç-bit kelime için

$$E(x_1x_2x_3) = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} x_1 & x_2 & x_3 & x_1 +_2 x_2 & x_2 +_2 x_3 & x_1 +_2 x_3 \end{pmatrix}$$

- $x_1x_2x_3$ kelimesi $\omega_1\,\omega_2\,\omega_3\,\omega_4\,\omega_5\,\omega_6$ olarak encode edilir ve

$$w_1 = x_1 \qquad w_2 = x_2 \qquad w_3 = x_3$$
$$w_4 = x_1 +_2 x_2 \qquad w_5 = x_2 +_2 x_3 \qquad w_6 = x_1 +_2 x_3$$

ve buradan aşağıdaki denklem sistemi elde edilir:

$$w_4 = w_1 +_2 w_2 \qquad w_5 = w_2 +_2 w_3 \qquad w_6 = w_1 +_2 w_3.$$

- Codewordün son üç basamağı farklı bilgi bit parçaları için parite kontrolü olarak hareket eder. Codeworddeki altı bitinin birisinde meydana gelebilecek hata bu denklemlerden hangisinin sağlanmadığı kontrol edilerek belirlenir.
- Örneğin, dördüncü bitteki bir hata için sadece dördüncü denklem sağlanmaz, birinci bitteki bir hata için birinci ve dördüncü denklemler sağlanmaz.

6-46

- $0 +_2 0=1 +_2 1=0$ olduğundan yukarıdaki denklem sistemi aşağıdaki gibi yazılabilir:

$$\begin{aligned} w_1 +_2 w_2 \quad &+_2 w_4 \quad = 0 \\ w_2 +_2 w_3 \quad &+_2 w_5 \quad = 0 \\ w_1 \quad +_2 w_3 \quad &+_2 w_6 = 0 \end{aligned}$$

Bu denklem sistemi matris formunda aşağıdaki hali alır;

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Buradaki matris $H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$

Parite kontrol matrisi olarak adlandırılır ve her doğru transfer edilen codeword iç

$H w^{\mathrm{T}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ denklemi sağlanır. Buradaki G üreteç matris ile H parite kontrol matrisi arasındaki ilişkiye dikkat edelim.

$F = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ Şeklinde tanımlana F matrisi için G matrisi (I_3 F) ve H matriside ($F^{\mathrm{T}} I_3$) olduğu görülür. $G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$

6-47

Örnek

- Üreteç matrisi $G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$

olan $E:B^4 \rightarrow B^7$ encoding fonksiyonunu göz önüne alalım. Örneğin

$$E(1101) = \begin{pmatrix} 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Buradaki $G=(I_4\ F)$ matrisi incelenecek olursa F matrisinin $F = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

olduğu kolayca belirlenir. Dolayısıyla, G matrisine karşılık gelen parite kontrol matrisi olan $H=(F^{\mathrm{T}}\ I_3)$ aşağıdaki şekilde bulunur:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

1101101 codeword'u için sonuç olarak aşağıdaki doğrulama gerçeklenir:

$$H(1\ 1\ 0\ 1\ 1\ 0\ 1)^{\mathrm{T}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

6-48

- **Teorem:** *G* üreteç matris olmak üzere *E(x)=xG* şeklinde verilen $E:B^m \rightarrow B^n$ bir encoding fonsiyonu olsun. $E(B^m)$ codewordler kümesi modülo 2 bit toplama işlemine göre bir gruptur.
- Verilen herhangi $x_1, x_2 \in B^m$ için

$$\begin{aligned} E(x_1 \oplus x_2) &= (x_1 \oplus x_2)G \\ &= x_1G \oplus x_2G \\ &= E(x_1) \oplus E(x_2). \end{aligned}$$

*E, $(B^m, \oplus)$'den $(B^n, \oplus)$*ye morfizmdir. $(B^m, \oplus)$ bir grup olduğundan $E(B^m, \oplus))$de bir grup oluşturur.

6-49

- Daha önceden bir kodun hata tespiti ve hata düzeltme kapasitesinin kodun minimum uzaklığına bağlı olduğunu görmüştük. Bir grup kode için minimum uzaklık; sıfırdan farklı codewordun minimum ağırlığıdır. Bu sonucu kullanarak; parite kontrol matrisinden kaç tane hatanın tespit edilebileceğinin veya düzeltilebileceğinin nasıl yapılacağını gösterebiliriz.
- r×n tipindeki H matrisinin sütunlarının $h_1, h_2, \ldots, h_n$ şeklinde gösterildiğini ve bu sütunların k tanesi için ilgili satıra karşılık gelen elemanların toplamının sıfır olduğunu kabul edelim. Bu sütundaki elemanları $h_{i_1}, h_{i_2}, \ldots, h_{i_k}$ ile gösterelim. n basamaklı *w* kelimesi H w^T=0 bağıntısını sağlar ve bir codewordtur öyle ki $i_1, i_2, \ldots, i_k$ basamaklarında birler ve diğer basamaklarında sıfır vardır.
- Bunun böyle olduğunu göstermek için aşağıdaki H matrisini göz önüne alalım:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} \end{pmatrix}$$

- Şimdi H'nin birinci, üçüncü ve dördüncü sütunlarının toplamının sıfır olduğunu kabul edelim.

$$\begin{aligned} h_{11} +_2 h_{13} +_2 h_{14} &= 0 \\ h_{21} +_2 h_{23} +_2 h_{24} &= 0 \\ h_{31} +_2 h_{33} +_2 h_{34} &= 0. \end{aligned}$$

6-50

- Şimdi w_1, w_2, w_3, w_4, w_5 basamaklarından oluşan ve $w_1 = w_3 = w_4 = 1$, $w_2 = w_5 = 0$ bağıntılarını sağlayan w kelimesini göz önüne alalım.

$$
\begin{aligned}
H\mathbf{w}^{\mathrm{T}} &= \begin{pmatrix} h_{11}w_1 + h_{12}w_2 + h_{13}w_3 + h_{14}w_4 + h_{15}w_5 \\ h_{21}w_1 + h_{22}w_2 + h_{23}w_3 + h_{24}w_4 + h_{25}w_5 \\ h_{31}w_1 + h_{32}w_2 + h_{33}w_3 + h_{34}w_4 + h_{35}w_5 \end{pmatrix} \\
&= \begin{pmatrix} h_{11}w_1 + h_{13}w_3 + h_{14}w_4 \\ h_{21}w_1 + h_{23}w_3 + h_{24}w_4 \\ h_{31}w_1 + h_{33}w_3 + h_{34}w_4 \end{pmatrix} \quad \text{since } w_2 = w_5 = 0 \\
&= \begin{pmatrix} h_{11} + h_{13} + h_{14} \\ h_{21} + h_{23} + h_{24} \\ h_{31} + h_{33} + h_{34} \end{pmatrix} \qquad \text{since } w_1 = w_3 = w_4 = 1 \\
&= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.
\end{aligned}
$$

- Dolayısıyla 10110 bir codewordtür.
- Tersine; bir codewordün sadece $i_1, i_2, \ldots, i_k$ basamaklarında birler varsa H matrisinin $i_1, i_2, \ldots, i_k$ sütunlarının toplamı sıfırdır.
- Bu sonuç, G üreteç matrisi veya denk olarak H parite kontrol matrisi ile tanımlanan kelimenin minimum ağırlığının belirlenmesine imkan sağlar.
- Sonuç basit olarak; H matrisinin toplamı sıfır olan sütunlarının minimum sayısına karşılık gelir. Bir kod grup code olduğundan minimum ağırlık minimum uzaklığa eşit olduğundan kodun hata tespit ve hata düzeltme kapasitelerini belirleyebiliriz.

Örnek

- E:$B^4 \rightarrow B^7$, her $x \in B^4$ için $E(x)=xG$ olan aşağıdaki G

$$
G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}
$$

üreteç matrisini tarafından tanımlanan E encoding fonksiyonu yardımıyla tanımlanan grup kodunu göz önüne alalım. G ye karşı gelen parite kontrol matrisi

$$
H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.
$$

şeklinde yazılır.

- Kodun minimum ağırlığını bulmak için H matrisinin toplamı sıfır olan sütunlarının minimum sayısını bulmalıyız. Toplamları sıfır olan iki sütun için elemanlarının her birinin tamamıyla aynı olması gerekir.
- Sıfırlanan hiçbir sütun olmadığında veya tamamıyla bir birinin eşiti olan sütun olmadığında minimum ağırlık en azından üçtür.
- Bu şu şekilde doğrulanabilir; 1, 2 ve 5. sütunların toplamı sıfır yapar (bunlar sadece toplamları sıfır olan sütunlar değildir). Bu kodun minimum ağırlığı üçtür ve codewordler arasındaki minimum uzaklık üçtür. Buradan bu kodun 2-hata tespiti ve 1-hata düzeltmeli olduğu sonucuna ulaşılır.

- Parite kontrol veya üreteç tarafından tanımlana bir sistematik kod verilmiş olsun. Alınan w kelimesinin decode edilmesi

$$H w^T=0$$

bağıntısının hesaplanmasını içerir. Bu değer w kelimesinin sendromu (syndrom) olarak adlandırılır. Eğer sendrom tamamı sıfır olan elemanlara sahipse, akla uygun olarak kelimenin doğru transfer edildiği sonucuna ulaşırız ve decodin aşaması ilk m bilgi basamaklarının seçimini içerir.

- Eğer sendromun bir vaya daha fazla elemanı sıfırdan farklı ise ne olur?
- Bu durumda en azından bir iletim hatasının meydana geldiğini biliriz.
- w_r ile alınan kelimeyi ve w_t ile de gönderilen kelimeyi gösterelim. w_r kelimesi w_t kelimesinin (modülo 2 ye göre) i. basamağına 1 eklenmesinden dolayı oluşacağından bu iki kelime farklılaşacaktır.
- e i. basamak hariç tüm basamakları sıfır olan ikili kelime (binary word) olarak tanımlanacak olursa, e hata deseni (error pattern) olarak adlandırılır ve $w_r=w_t \oplus e$ olur.

6-53

Tanım: n basamaklı w_t kelimesi gönderildiğinde n basamaklı bir w_r kelimesinin alındığını kabul edelim. Hata deseni $e_1,e_2, \ldots, e_n$ basamaklarına sahip e ikili kelimesidir. Burada

$$e_i = \begin{cases} 0 & w_r \text{ ve } w_t \text{'nin } i.\text{basamağı aynı ise} \\ 1 & w_r \text{ ve } w_t \text{'nin } i.\text{basamağı farklı ise} \end{cases}$$

w_t kelimesinin i. Basamağında meydana gelen iletim hatası için Hw_t^T sendrom aşağıdaki şekilde karşımıza çıkar.

$$\begin{aligned} H w_r^T &= H(w_t \oplus e)^T \\ &= H(w_t^T \oplus e^T) \\ &= H w_t^T \oplus H e^T \\ &= H e^T \qquad \text{(since } w_t \text{ is a codeword)} \\ &= i\text{th column of } H. \end{aligned}$$

Sonuç olarak, tek hata oluştuğunda, sendrom hatanın tam olarak hangi basamakta meydana geldiğini bize söyler.

6-54

Örnek

- E:$B^3 \rightarrow B^6$, parite kontrol matrisi

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

ile verilen encoding fonksiyonunu göz önüne alalım. 100001 kelimesinin alındığını kabul edelim. Gönderilen kelimenin hangi kelime olabileceğini araştıralım.

Önce gönderilen kelimenin sendromunu hesaplayalım

$$Hw^T = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Hw_t^T sıfır olmadığından 10001 bir codeword değildir ve doğru olarak transfer edilememiştir. Sonuç H matrisinin ikinci sütununu verdiğinden kelimenin ikinci basamağı yanlış iletilmiştir. Alınması gereken kelime 110001 olmalıydı. Dolayısıyla Kelime 110 olarak decode edilmelidir.

6-55

UYARI

- Sendromun sıfır olamadığını veya H nin bir sütunu ile aynı olmadığını kabul edelim.
- Bu durumda, hatanın birden daha fazla basamakta meydana geldiği sonucuna ulaşırız ve tek-hata-düzeltmeli kod ile alınan kelime akla uygun bir şekilde decode edilemeyecektir.
- Bu kısa kod teorisine giriş ile tek hata düzeltmeli kodlar üzerinde yoğunlaştık.
- Bu konunun incelenmesi, tek hatalı kelimeler ile karşılaşmanın çok hatalı kelimeler ile karşılaşmaktan daha sık karşılaşılabilecek durum olduğundan önemlidir.
- Uzay gemilerinden gönderilen kelimelerde olduğu gibi çok hatalı kelime transferinin az olmadığı durumlarda daha karmaşık hata tespit ve hata-düzeltme kapasitesine sahip kodlar kullanılmalıdır.
- Bu ise soyut cebir alanındaki teorilerin bu alana uygulanmasıyla yapılabilmektedir.

6-56