1. Önerme Mantığı ve İspatlar

Mantık önermelerin doğruluğunu kanıtlamak için kullanılır. Önermenin ne olduğu ile ilgilenmek yerine bazı kurallar koyar ve böylece önermenin genel formunun geçerli olup olmadığını yargılar. Mantığın bize sağladığı kurallar, belirtilen aşamalardan çıkan sonucun tutarlı olup olmadığını veya sonucun doğruluğunun ispatlanması aşamasındaki basamaklarda hatalı bir kısmın bulunup bulunmadığını değerlendirmemizi sağlar.

1.1 Önermeler ve Doğruluk Tabloları

Önerme, doğru veya yanlış değerinden sadece ve sadece birini alabilen ifadedir. Fakat aynı anda iki değeri birden alamaz. Örneğin aşağıdaki ifadeler birer önermedir.

1. Bu gül beyazdır.
2. Üçgenin dörtkenarı vardır.
3. $3 + 2 = 6$.
4. $6 < 24$
5. Yarın benim doğum günümdür.

Aynı önermenin nerede, ne zaman ve kim tarafından söylendiğine bağlı olarak bazen doğru bazen yanlış olabileceğine dikkat ediniz. Yarın doğum günü olan biri için 5. önerme doğru iken, başka biri tarafından ifade edildiğinde yanlış olacaktır. Hatta bugün herhangi biri için doğru olan bir önerme başka bir gün için yanlış olabilir.

Ünlemler, sorular ve istekler doğru veya yanlış diye ifade edilemediklerinden birer önerme değildirler. Bu nedenle aşağıdakiler önerme değildir.

6. Çimlerden uzak durun.
7. Çok yaşa kraliçe!
8. Jane'in partisine gittin mi?
9. Öyle söyleme.

Bir önermenin doğruluğu (T) veya yanlışlığı (F) önermenin **doğruluk değeri** şeklinde adlandırılır. 4. önerme doğru (T) doğruluk değerini taşırken, 2 ve 3. önermeler yanlış (F) doğruluk değerini taşır. 1 ve 5 numaralı önermenin doğruluk değerleri ifade edildikleri duruma bağlıdır.

Geleneksel olarak önermeler p,q,r… harfleri kullanılarak sembolize edilirler. Örneğin p: Manchester İskoçya'dadır, q: Dinozorlar hala yaşamaktadır.

1.2 Mantıksal Bağlılıklar ve Doğruluk Tabloları

Bundan önceki konudaki 1-5 numaralı ifadeler basit birer ifade oluşturduklarından **basit önerme**lerdir. Bu bölümde basit önermelerin nasıl bağlanarak **bileşik önerme**ler şeklinde adlandırılan daha karışık önermeler oluşturulacağı anlatılacaktır. Önerme çiftlerini bağlamaya yarayan araçlara mantıksal bağlayıcılar denir ve herhangi bir bileşik önermenin doğruluk değeri tamamen (a) kendisini oluşturan basit önermelerin doğruluk değerleri (b) bunları bağlayan özel bağlayıcı veya bağlayıcılar tarafından belirlenir.

En çok kullanılan bağlaçlara geçmeden önce, basit bir önerme üzerinde gerçekleştirilebilen bir işleme bakalım. Bu işleme **tersini alma** denir ve önermenin doğruluk değerini tersine çevirme etkisi yapar. Tersini alma sonucunda önerme eğer doğru ise yanlış, yanlış ise doğru değerini alır. Bu işlemi bir tablo ile özetleyebiliriz. Eğer p bir önermeyi sembolize ediyorsa, $\overline{p}$ (~p,-p veya $\neg$p) p'nin tersini temsil eder. Aşağıdaki tablo p ve $\overline{p}$ 'nün doğruluk değerleri arasındaki ilişkiyi gösterir.

p	$\overline{p}$
T	F
F	T

Soldaki sütun p için tüm olası doğruluk değerlerini verirken sağ sütun p' nin tersi $\overline{p}$ için karşılık gelen doğruluk değerlerini verir. Bu şekilde, önermelerin doğruluk değerlerini özetleyen tabloya **doğruluk tablosu** denir.

Bir önermenin tersini ifade etmenin çeşitli yolları vardır. "Bütün köpekler vahşidir" önermesini düşünürsek, bu önermenin tersi şunlar olabilir:

Bütün köpeklerin vahşi olması söz konusu değildir.
Köpeklerin hepsi vahşi değildir.
Bazı köpekler vahşi değildir.

Dikkate edilirse "Hiçbir köpek vahşi değildir" önermesi "Bütün köpekler vahşidir" önermesinin tersi değildir. Tersini alma işleminde, ilk ifadenin doğru olduğu her durumda ikinci ifade yanlış olmalı ya da tam tersi olmalıdır. "Bütün köpekler vahşidir" önermesi sadece bir köpek bile vahşi olduğunda yanlıştır ancak "Hiçbir köpek vahşi değildir" önermesi bu durumda doğru değildir.

Mantıksal bağlayıcılar önerme çiftlerini bağlamaya yararlar. Burada çok kullanılan beş mantıksal bağlayıcıdan bahsedilecektir: kesişim, dahili birleşim, harici birleşim, koşullu önerme ve iki yönlü koşullu önerme.

1.2.1 Kesişim (Conjunction)

İki basit önerme aralarına 've' kelimesi koyarak bağlanabilir. Bunun sonucunda oluşan bileşke önermeye iki basit önerme bileşeninin kesişimi denir. Eğer p ve q iki basit önerme ise $p \wedge q$ (veya p.q) p ve q 'nun birleşimini temsil eder.

p: Güneş Parlıyor.
q: Köpekler havlar.
$p \wedge q$: Güneş parlıyor ve köpekler havlar.

Alttaki doğruluk tablosu p ve q 'nun tüm olası doğruluk değerleri için $p \wedge q$ 'nun doğruluk değerlerini gösterir.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Yukarıdaki tablodan da görülebildiği gibi $p \wedge q$ sadece p ve q 'nun her ikisinin de doğru olduğu zaman doğrudur.

1.2.2 Birleşim (Disjunction)

Veya kelimesi iki basit önermeyi birleştirmek için kullanılabilir. Oluşan bileşke önerme iki basit önermenin birleşimi olarak adlandırılır. Mantıkta iki çeşit birleşim vardır: dahili ve harici. Gerçek hayatta kullandığımız veya kelimesi bazen kafa karıştırıcı olabilir.

p ve q birer önerme ise p v q, p ve q ‘nun dahili birleşimini temsil eder. Bu bileşke önerme bileşenlerinden herhangi birisi veya her ikisinin doğru olması durumunda doğru aksi halde yanlıştır. $p \vee q$ için doğruluk tablosu aşağıdadır.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p ve q’ nun harici birleşimi ise $p \veebar q$ şeklinde gösterilir. Bu bileşke önerme sadece bir bileşenin doğru olması durumunda doğrudur. $p \veebar q$ ‘nun doğruluk tablosu aşağıdaki gibidir:

p	q	$p \veebar q$
T	T	F
T	F	T
F	T	T
F	F	F

İki basit önerme “veya” kullanılarak bağlanırken hangi tip birleşimin kullanılacağı cümlenin genel durumundan anlaşılır. Örneğin, ‘Yarın yüzmeye gideceğim veya golf oynayacağım’ cümlesi iki işin birden yapılmayacağı anlamı taşıdığından harici birleşimdir. Diğer taraftan, ‘Adaylar 25 yaşın üzerinde veya en az 3 yıllık tecrübeye sahip olmalıdır’ cümlesinde iki koşuldan birini sağlayan adaylar dikkate alınacakmış izlenimi verdiğinden dahili birleşimidir.

1.2.3 Koşullu Önermeler

Koşullu önerme bağlayıcısı → işareti ile sembolize edilir. Koşullu önermenin normal dildeki karşılığı örnekte de görüleceği gibi ‘Eğer …’ dir.

p: Kahvaltı yaparım.
q: Öğlen yemeği yemem.
$p \rightarrow q$: Eğer kahvaltı yaparsam, öğlen yemeği yemem.

Yukarıdaki örnekteki $p \rightarrow q$ için diğer alternatifler:

Sadece eğer öğlen yemeği yemezsem kahvaltı yaparım.
Kahvaltı yapmam öğlen yemeği yemeyeceğim anlamına gelir.
Ne zaman kahvaltı yapsam öğlen yemeği yemem.

Aşağıdaki tablo $p \rightarrow q$’ nun doğruluk tablosudur.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Dikkat edilirse 'p ise q' önermesi sadece p' nin doğru q' nun yanlış olması durumunda yanlıştır.(örneğin doğru bir ifade yanlış bir ifade anlamına gelemez.) Eğer p yanlış ise bileşke önerme q'nun doğruluk değeri ne olursa olsun doğrudur. Şu önermeye bakalım: 'Eğer derslerimi geçersem çok sevineceğim'. Bu ifade eğer sınavlarımı geçemezsem ne yapacağım hakkında hiçbir şey söylemiyor. Belki sevinirim, belki sevinmem ama hiçbir durumda söylenen ifade yanlış değildir. Önermenin yanlış olabileceği tek durum sınavlarımı geçip sevinmediğim durumdur.

Koşullu önermelerde, p önermesi önceki ve q önermesi sonraki olarak adlandırılır. p önermesi q için **yeterli** koşul, q ise p için **gerekli** koşuldur.

1.2.4 Çift Yönlü Koşullu Önermeler

Çift yönlü koşullu bağlayıcı ↔ ile gösterilir ve 'sadece ve sadece …. ise …….' şeklinde ifade edilir. Önceki örneğe tekrar dönersek:

p: Kahvaltı yaparım.
q: Öğlen yemeği yemem.
p ↔ q : Sadece ve sadece öğlen yemeği yemezsem kahvaltı yaparım.(alternatif olarak sadece ve sadece kahvaltı yaparsam öğlen yemeği yemem.)

p ↔ q 'nin doğruluk tablosu şu şekildedir:

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Dikkat edilirse p ↔ q nun doğru olabilmesi için p ve q nun her ikisinin de aynı doğruluk değerine sahip olması gerekir.

Örnek 1.1: p, 'Bugün Pazartesi' ve q 'İstanbul'a gideceğim' önermeleri olsun. Buna göre aşağıdaki önermeleri sembollerle ifade ediniz.

(i) Eğer bugün Pazartesi ise İstanbul'a gitmeyeceğim.
(ii) Bugün Pazartesi veya İstanbul'a gideceğim fakat ikisi birden değil.
(iii) Bugün İstanbul'a gideceğim ve bugün Pazartesi değil.
(iv) Sadece ve sadece bugün Pazartesi değilse İstanbul'a gideceğim.

Çözüm 1.1:

(i) $p \rightarrow \overline{q}$
(ii) $p \veebar q$
(iii) $q \wedge \overline{p}$
(iv) $\overline{p} \leftrightarrow q$.

Örnek 1.2: Aşağıdaki bileşik önermeler için doğruluk tabloları oluşturunuz.

(i) $\overline{p} \vee q$

(ii) $\overline{p} \wedge \overline{q}$

(iii) $\overline{q} \rightarrow p$

(iv) $\overline{p} \leftrightarrow \overline{q}$

Çözüm 1.2:

(i)

p	q	$\overline{p}$	$\overline{p} \vee q$
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

(ii)

p	q	$\overline{p}$	$\overline{q}$	$\overline{p} \wedge \overline{q}$
T	T	F	F	F
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

(iii)

p	q	$\overline{q}$	$\overline{q} \rightarrow p$
T	T	F	T
T	F	T	T
F	T	F	T
F	F	T	F

(iv)

p	q	$\overline{p}$	$\overline{q}$	$\overline{p} \leftrightarrow \overline{q}$
T	T	F	F	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

1.3 Tutolojiler ve Çelişkiler

Bileşenlerinin doğruluk değeri ne olursa olsun her zaman doğru olan birçok bileşik önerme mevcuttur. Benzer şekilde bileşenlerinden bağımsız olarak her zaman yanlış olanlar da vardır. Her iki durumda da bu özellik bileşke önermenin yapısının sonucudur.

Tutoloji, basit bileşenlerinin doğruluk değeri ne olursa olsun doğru olan bileşke önermedir. Örnek : insanlar erkektir veya kadındır önermesi her zaman doğrudur. O nedenle bu önerme bir tutolojidir.

Çelişki ise, basit bileşenlerinin doğruluk değeri ne olursa olsun yanlış olan bileşke önermedir.

Tutoloji *t* ile, çelişki ise *f* ile gösterilir.

Örnek 1.3 : $p \vee \overline{p}$ ‘nin tutoloji olduğunu gösteriniz.

Çözüm1.3 : Eğer $p \vee \overline{p}$ ‘in doğruluk tablosunu yaparsak:

p	$\overline{p}$	$p \vee \overline{p}$
T	F	T
F	T	T

Dikkat edilirse $p \vee \overline{p}$ her zaman doğru değerini verir (p yerine hangi önerme konulursa konulsun) ve bu sebeple tutolojidir.

Örnek 1.4 : $(p \wedge q) \vee \left(\overline{p \wedge q}\right)$ ‘nin tutoloji olduğunu gösteriniz.

Çözüm 1.4 : Verilen önermenin doğruluk tablosu aşağıdaki gibidir:

p	q	$p \wedge q$	$\overline{p \wedge q}$	$(p \wedge q) \vee \left(\overline{p \wedge q}\right)$
T	T	T	F	T
T	F	F	T	T
F	T	F	T	T
F	F	F	T	T

Doğruluk tablosunun en son sütunu sadece T doğruluk değerini gösterir ve bu nedenle $(p \wedge q) \vee \left(\overline{p \wedge q}\right)$ ifadesi bir tutolojidir.

Son örnekte, ilk örnekten elde ettiğimiz ‘herhangi bir önermenin tersinin dahili birleşimi bir tutolojidir’ sonucunu kullanabilirdik. İkinci örnekte elimizde $p \wedge q$ önermesi ve tersi $\overline{p \wedge q}$ var. Bu nedenle önceki sonuca göre $(p \wedge q) \vee \left(\overline{p \wedge q}\right)$ bir tutolojidir. $(p \wedge q) \vee \left(\overline{p \wedge q}\right)$ önermesi, $p \vee \overline{p}$ önermesinin **yedek örneği**dir denir. Açıkça görülüyor ki, bir tutolojinin yedek örneği kendi başına bir tutolojidir ve dolayısıyla bir önermenin tutoloji olduğunu göstermenin bir yolu da bu önermenin tutoloji olduğu bilinen başka bir önermenin yedek örneği olduğunu göstermektir. Tıpkı tutolojilerde olduğu gibi bir çelişkinin de yedek örneği bir çelişkidir.

1.4 Mantıksal Eşdeğerlilik ve Mantıksal Anlam

İki önerme, kendilerini oluşturan bileşenlerinin tüm doğruluk değeri kümesi için aynı doğruluk değerine sahipse bu iki önerme **mantıksal eşdeğer**dir denir. P ve Q iki bileşik önerme olsun, P ve Q mantıksal eşdeğerse P ≡ Q şeklinde gösterilir.Tutolojiler ve çelişkilerde olduğu gibi mantıksal eşdeğerlilik de P ve Q’ nun yapılarının sonucudur.

Örnek 1.5 : $\overline{p} \vee \overline{q}$ ve $\overline{p \wedge q}$ ’ nun mantıksal eşdeğer olduğunu gösteriniz.

Çözüm 1.5 : $\overline{p} \vee \overline{q}$ ve $\overline{p \wedge q}$ için doğruluk tablosunu çizelim.

p	q	$\overline{p}$	$\overline{q}$	$\overline{p} \vee \overline{q}$	$p \wedge q$	$\overline{p \wedge q}$
T	T	F	F	F	T	F
T	F	F	T	T	F	T
F	T	T	F	T	F	T
F	F	T	T	T	F	T

$\overline{p} \vee \overline{q}$ ve $\overline{p \wedge q}$ için hesaplanan sütunlardaki doğruluk değerleri karşılaştırılırsa aynı olduklarını görülür. Bu nedenle bu iki önerme mantıksal eşdeğerdir denilebilir.

Eğer iki bileşke önerme mantıksal eşdeğerse, bu iki önermenin çift yönlü koşullu bağlayıcı ile bağlanmasıyla oluşan önerme bir tutoloji olmalıdır.($P \equiv Q$ ise $P \leftrightarrow Q$ tutoloji olmalı) Bunun nedeni, iki mantıksal eşdeğer önermenin ikisi de aynı anda ya doğrudur ya yanlıştır. Her iki durumda da çift yönlü koşullu önerme doğrudur. Bu durumun tersi de yani $P \leftrightarrow Q$ bir tutoloji ise $P \equiv Q$. Bunun nedeni şu gerçeğe dayanır: Çift yönlü koşullu önerme $P \leftrightarrow Q$ sadece P ve Q' nun her ikisinin de aynı doğruluk değerine sahip olduğu zaman doğrudur.

İki önerme arasında oluşabilecek bir diğer yapıya-bağımlı ilişki de mantıksal anlamdır. Eğer bir P önermesi her doğru olduğunda Q önermesi de doğru oluyorsa, P önermesi mantıksal olarak Q önermesi anlamına gelir. Ancak bunun tersi doğru değildir yani Q, P yanlış olduğunda da doğru olabilir. Mantıksal anlam $\vdash$ ile sembolize edilir. $P \vdash Q$, P mantıksal olarak Q anlamına gelir demektir.

Örnek 1.6: $q \vdash (p \vee q)$ olduğunu gösteriniz.

Çözüm 1.6: q'nun her doğru olduğu anda $(p \vee q)$ nun da doğru olduğunu göstermek gerekir. Doğruluk tablosunu yaparsak:

p	**q**	**$p \vee q$**
T	T	T
T	F	T
F	T	T
F	F	F

q'nun doğru olduğu her durumda (1 ve 3. satırlar) $p \vee q$ da doğrudur. $p \vee q$, q yanlış olduğunda da doğrudur (2. satır) fakat bunun q, $p \vee q$ ile mantıksal anlamdır ifadesinin sağlanmasıyla bir alakası yoktur.

'$P \vdash Q$' ile '$P \rightarrow Q$ bir tutolojidir' ifadeleri benzer ifadelerdir. $P \vdash Q$ ise P doğru iken Q hiçbir durumda yanlış değildir. Bu da sadece $P \rightarrow Q$' nun yanlış olduğu durumda mümkün olduğundan $P \rightarrow Q$ bir tutoloji olmalıdır.

1.5 Önermeler Cebri

Aşağıdaki liste bir önceki konudaki teknikler kullanılarak ispatlanabilecek mantıksal eşitlikleri içerir. Bu kurallar p, q ve r gibi basit önermeler ve bunların yerine konabilecek yedek örneklerin tamamı için geçerlidir.

Aynılık (Tek Kuvvet) Özelliği(idempotence)

$p \wedge p \equiv p$
$p \vee p \equiv p.$

Değişme Özelliği(Commutativity)

$p \wedge q \equiv q \wedge p$
$p \vee q \equiv q \vee p$
$p \underline{\vee} q \equiv q \underline{\vee} p$
$p \leftrightarrow q \equiv q \leftrightarrow p.$

Birleşme Özelliği(Associativity)

$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
$(p \vee q) \vee r \equiv p \vee (q \vee r)$
$(p \veebar q) \veebar r \equiv p \veebar (q \veebar r)$
$(p \leftrightarrow q) \leftrightarrow r \equiv p \leftrightarrow (q \leftrightarrow r)$.

Yutan Eleman(absorbtion)

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

Dağılma Özelliği(Distributivity)

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

Çift ters Özelliği((Involution)

$\overline{\overline{p}} \equiv p$.

De Morgan Kuralları

$\overline{p \vee q} \equiv \overline{p} \wedge \overline{q}$
$\overline{p \wedge q} \equiv \overline{p} \wedge \overline{q}$.

Özdeşlik Özelliği(identity)

$p \vee f \equiv p$
$p \wedge t \equiv p$
$p \vee t \equiv t$
$p \wedge f \equiv f$.

Tamamlama Özelliği(Complement)

$p \vee \overline{p} \equiv t$
$p \wedge \overline{p} \equiv f$
$\overline{f} \equiv t$
$\overline{t} \equiv f$

1.5.1 Eşlik Kuralı (Duality Principle)

Sadece $\vee$ ve $\wedge$ bağlayıcılarını içeren herhangi bir P önermesi verilmiş ise, bu önermenin eşi $\vee$ yerine $\wedge$, $\wedge$ yerine $\vee$, *t* yerine *f* ve *f* yerine de *t* koyarak elde edilir. Örneğin, $(p \wedge q) \vee \overline{p}$' nin eşi $(p \vee q) \wedge \overline{p}$ olmalıdır.

Dikkat edilirse kesişim ve dahili birleşim dışındaki bağlayıcılarla bağlanmış bileşik önermelerin eşinin nasıl elde edildiğinden bahsedilmedi. Bunun önemi yoktur çünkü diğer bağlayıcılara sahip önermelerin hepsi sadece tersini alma ve kesişim bağlayıcılarını içeren mantıksal eşdeğer formunda yazılabilir. Eşlik kuralına göre eğer iki önerme mantıksal eşdeğerse, eşleri de mantıksal eşdeğerdir.

1.5.2 Yerine Koyma Kuralı

Diyelim ki, elimizde mantıksal eşdeğer P_1 ve P_2 önermeleri ile P_1 ‘i içeren Q bileşik önermesi var. Yerine koyma kuralına göre P_1 yerine P_2 koyarsak sonuçta oluşan önerme Q ile mantıksal eşdeğerdir. Bu sebeple mantıksal eşdeğer önermeleri birbirinin yerine koymak sonuçta oluşan önermenin doğruluk değerini değiştirmez. Bunun ispatı şu şekilde yapılabilir: Doğruluk tablosunda P_1 sütunu yerine P_2 sütununu koyarsak sonuç değişmez zira P_1 ve P_2 ‘nin doğruluk tabloları aynıdır.

Yerine koyma kuralı ve önermeler cebri kuralları doğruluk tabloları çizmeden önermeler arasında mantıksal eşitlikler kurabilmemizi sağlar.

Örnek 1.7: $(\overline{p} \wedge q) \vee \left(\overline{p \vee q}\right) \equiv \overline{p}$ olduğunu ispatlayınız.

Çözüm 1.7: $(\overline{p} \wedge q) \vee \left(\overline{p \vee q}\right) \equiv (\overline{p} \wedge q) \vee (\overline{p} \wedge \overline{q})$ (De Morgan Kuralı)

$\equiv \overline{p} \wedge \left(q \vee \overline{q}\right)$ (Dağılma özelliği)

$\equiv \overline{p} \wedge t$

$\equiv \overline{p}$

1.5.3 Koşullu önermler ile ilgili diğer özellikler

Verilen $p \rightarrow q$ koşullu önermesi için;

a) $p \rightarrow q$ ‘nün karşıtı(converse) $q \rightarrow p$

b) $p \rightarrow q$ ‘nün tersi(inverse) $\overline{p} \rightarrow \overline{q}$

a) $p \rightarrow q$ ‘nün ters pozitifi(contrapositive) $\overline{q} \rightarrow \overline{p}$

Doğruluk Tablosu

p	q	$p \rightarrow q$	$q \rightarrow p$	$\overline{p} \rightarrow \overline{q}$	$\overline{q} \rightarrow \overline{p}$
T	T	T	T	T	T
T	F	F	T	T	F
F	T	T	F	F	T
F	F	T	T	T	T

Tablodan $p \rightarrow q$ ‘nün ters pozitifi olan, $\overline{q} \rightarrow \overline{p}$ ‘nin mantıksal eşdeğer oldukları görülmektedir.

$$p \rightarrow q \equiv \overline{q} \rightarrow \overline{p}$$

Koşullu önermenin karşıtı veya tersi kendisi ile mantıksal eşdeğer değildir. Hâlbuki karşıt ve zıttı birbiriyle mantıksal eşdeğerdir.

Örnek: **p** : bu gün salı **q**: bu gün bir sınavım var

$p \to q$: eğer bugün salı ise bugün bir sınavım var

a) $p \to q$ 'nün karşıtı(converse) $q \to p$: Eğer bugün sınavım var ise bugün salı.

b) $p \to q$ 'nün tersi(inverse) $\overline{p} \to \overline{q}$: Eğer bu gün salı değil ise bugün sınavım yok

a) $p \to q$ 'nün ters pozitifi(contrapositive) $\overline{q} \to \overline{p}$: Eğer bugün sınavım yok ise bugün salı değil.

Tez(Argument): birbirini oluşturan önemeleri dayanak olarak alan önemeler kümesine denir ve sonunda bir sonuca ulaşır. Dayanak noktalarındaki önermeler bağlaç ile birbirine bağlanırlar ve sonunda mantıksal bir sonuca ulaşırlar. Aksi halde tez geçersizdir.

Eğer dayanak noktasındaki önermeler $P_1, P_2, \ldots, P_n$ ve sonucu Q ise tez,

Eğer $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \vdash Q)$ veya $(P_1 \wedge P_2 \wedge \ldots \wedge P_n \to Q)$ bir tutolojidir. $P_1, P_2, \ldots, P_n$ doğru olduğunda , Q doğru olmalıdır.

1.5.4 Yüklem mantığı(Predicate Logic)

Yüklem, bir veya birkaç nesnenin veya bireylerin özelliklerini açıklar.

…. kırmızı,

….. nın uzun dişleri var

……Başı üzerinde durmaktan hoşlanır . gibi.

Yüklemi ifade etmek için büyük harf ile semboller kullanırız.

M: kırmızıdır

B: uzun dişleri var

G: başının üzerinde durmaktan hoşlanır

Küçük harf semboller ise bireyleri ifade etmekte kullanılır.

a : bu gül

b: Ahmet

Basit önerme aşağıdaki gibi ifade edilebilir.

M(a) : Bu gül kırmzıdır

M(b) :Ahmet kırmızıdır

G(b) : Ahmet başı üzerinde durmaktan hoşlanır.

Eğer M , kırmızıdır yüklemi olarak ifade edilirse M 'yi M(x) olarak ifade ederiz ve "x kırmızıdır" anlamına gelir. Burada x değişkeni, herhangibir nesne veya birey adı ile yer değiştirilebilir. Bu nedenle M(x) önermesel fonksiyon olarak adlandırılır. Önermesel fonksiyonun tersi ,

Eğer M(x) : "x kırmızıdır" ise $\overline{M(x)}$: "x kırmızı değildir" anlamına gelir.

Evrensel Niteleyici :"Bütün sıçanlar gridir" önermesini düşünelim. Bunu ilk yolu bütün sıçanlar için ; eğer x bir sıçan ise x gridir . önermesi ifade edilebilir. Bu bize yeni bir gösterim tanımlamayı getirir.

R(x) : x bir sıçandır , G(x) : x gridir. Her x için ‘i $\forall$ x olarak ifade edip

$(\forall x)[R(x) \rightarrow G(x)]$ şeklinde yazılır. Burada $\forall$ evrensel niteleyici olarak adlandırılır.

Varlık Niteleyici :Eğer aynı önermede “Enaz bir adet x” vardır ‘ı ∃x şeklinde ifade ederek, “Bazı sıçanlar gridir” önermesini ; vardır şeklinde yazarız.

$(\exists x)[R(x) \rightarrow G(x)]$ olarak yazabiliriz burada ∃ ye varlık niteleyici denir ve en az bir x vardır veya bazı x’ler için şeklinde söylenir.

Yüklem mantığında Düşünceler

Yüklem mantığında bir tezin geçerliliği sağlanır. Bütün yüklemler doğruluğunun sağlandığı durumda sonuçta doğrudur. Aşağıdaki dört kural yüklem mantığında geçerlidir.

1. Evrensel tanım : Eğer önerme $(\forall x)F(x)$ doğru ise F(a) ‘da evrendeki her a için doğrudur.

2. Evrensel Genelleştirme: Eğer önerme F(a), evrendeki her a için doğru ise $(\forall x)F(x)$ ‘da doğrudur.

3. Varlık tanımı : Eğer önerme $(\exists x)F(x)$ doğru ise, evrende, F(a)‘yı doğru yapan bir a vardır.

4. Varlık genelleştirmesi : Eğer önerme evrendeki bazı a’lar için F(a) doğru oluyorsa‘ $(\exists x)F(x)$ doğrudur.

Örnek : Yeşil gözlü olan herkese güvenilmez. Ali’nin yeşil gözleri var. Öyleyse Ali’ye güvenilmez. Tezinin geçerliliğini gösterelim.

Eğer G(x) : x’in yeşil gözleri var ve T(x) : x güvenilir ve a , ali'yi gösterirse;

$(\forall x)[G(x) \rightarrow \overline{T(x)}]$ ve $G(a) \rightarrow \overline{T(a)}$ şeklindedir.

1.6 Matematiksel İspat

Matematiksel ispatın popüler görünümü genellikle sembollerle yazılan birtakım adımların ard arda sıralanması şeklindedir. Her bir adım mantıksal olarak ispatın bir önceki adımını takip eder ve son satır ispatlanacak ifadedir. Bu imaja bağlı olarak ortak kanı, ispatın matematiksel doğruluğun mutlak ve sıkı bir testi olmasıdır. Fakat sürpriz bir biçimde, kendi aralarında ortak bir kanı olmamasına rağmen, bu görüş çoğu profesyonel matematikçinin görüşü değildir. Çoğu ispatın sosyolojik boyutunun olduğu görüşünü savunur ve ispatı, fikirlerin açıklanması ve iletimi için bir şart olarak kabul eder.

Aslında her iki görüş de doğrudur. İspat kelimesi geniş bir yelpazeyi kapsar. Bu yelpazenin bir ucunda birinci bölümdeki mantıksal işaretlerle ifade edilen çok resmi ispatlar bulunur. Her bir adım bir önceki adımı mantık kuralları çerçevesinde takip eder. Aslında, ispat yapmak için sadece semboller kullanmak mümkündür fakat bu tabi ki takip etmesi zor bir olaydır. Daha az resmi ispatlar ise kelimelerin, sembollerin ve diyagramların karışımıyla gerçekleştirilir. Matematik ile ilgili kitaplardaki ispatlar genellikle az resmi ispatlardır.

Matematikte onay verilmeyen bir şey varsa o da gözlemlere dayanarak sonuca gitmektir. Öte yandan, birçok kez bir çift sayının karesini aldığımızda sonucun bir çift sayı olduğunu gözlememize rağmen bu çift sayıların karesinin çift sayı olduğunu kanıtlamaz. Ancak buna inancımızı kuvvetlendirir ve bu gözleme geçerli bir kanıt aramaya teşvik eder. Gözlemlere dayanarak çeşitli gerçekler hakkında yargılarda bulunmaya tümevarım denir. Mantıksal çıkarımlarla sonuca varılan yargıya ise tümdengelim denir.

1.6.1 Aksiyomlar ve Aksiyom Sistemleri

Matematiksel bir teori, örneğin küme teorisi, sayı teorisi veya grup teorisi değişik bileşenler içerir. Bunların en önemlileri:

1. Tanımlanmamış terimler.
2. Aksiyomlar.
3. Tanımlar.
4. Teoremler.
5. İspatlar.

3, 4 ve 5. maddelerde sıralananlar hakkında herkes bilgi sahibi olabilir. Matematikte tanımlanmamış terimlere ihtiyaç duymamız sürpriz gelebilir fakat biraz düşünülürse bunun gerekliliği anlaşılabilir.

Diyelim ki, küme teorisi üzerine eksiksiz bir çalışma yapmak istiyoruz. Açıktır ki başlangıç noktası kümenin ne olduğunu anlatmaktır: Tanım 1: Küme …..- Yani? Problem şu ki, kümeyi tarif etmek için başka bir terime ihtiyacımız var (örneğin topluluk) ancak bu sefer de diğer terim tanımlanmamış durumdadır. Diğer terimi tanımlayabilmek için yine başka bir tanımlanmamış terime ihtiyacımız var ve bu böyle devam eder. Açık ki, sonsuz bir tanım dizisinden uzak durmamız gerekiyor. Bu da bizi bazı terimleri tanımlanmamış bırakmaya zorlar. Tabi ki, hala aklımızdakini sezgisel biçimde ifade edebiliriz ancak bu sezgisel tanımlama teorimizin bir parçası olmak zorunda değildir.

Listedeki 2 numaralı eleman olan aksiyomların da biraz açıklanmaya ihtiyacı var. Matematiksel bir teorideki bütün terimleri tanımlayamadığımız gibi aynı sebeple teorideki her ifadeyi de kanıtlayamayız. Bir yerden başlamak için kanıtlanmayacak bazı ifadelere ihtiyaç vardır. Bu ifadelere **aksiyom** denir. Aksiyomlar teorinin temel özelliklerini temsil ederler.

Bilmek gerekir ki, aksiyomların doğruluğundan veya yanlışlığından söz edilmez; onlar sadece teorinin ilerleyebilmesini sağlayan tanımlanmamış terimler hakkındaki ifadelerdir. Öte yandan kendi aralarında tutarlı olmalıdırlar ve aynı anda hepsinin doğru olma imkanı olmalıdır. Kendi aralarında çelişen aksiyomlar kabul edilemez. Matematiksel bir ifadeyi uygulamaya gelince, tanımlanmamış terimler yorumlanırlar ve aksiyomlar doğru veya yanlış şeklinde önermeler haline gelir.

Bir aksiyomatik teori tanımlar yaparak ve teorem ispatlanarak gelişir. Tanımlar, tanımlanmamış terimlerin yanlış şeylerle ilişkilendirilmemeleri için yapılırlar. Teorem ise birinci bölümde anlatılan mantıksal yargıları kullanan aksiyomları takip eden, sistemdeki çeşitli terimler hakkındaki ifadelerdir. Teorem orijinal aksiyomlardan gittikçe uzaklaşarak yayılır fakat sonuçta onlar üzerine inşa edilir. Teoremler ve ispatları saf matematiğin kalbini oluşturur.

1.7 İspat Yöntemleri

Görüldüğü gibi, resmi matematik, aksiyomatik yöntem üzerine inşa edilmiştir. Tanımlanmamış terimler ve aksiyomlar ile başlar, mantık kurallarını kullanarak teoremleri ispatlayarak gelişir. Bu bölümde ispatın temel özellikleri ve bazı ispat yöntemlerinin genel yapısından bahsedilecektir.

Diyelim ki, $A_1, A_2, \ldots A_n$ bir aksiyom sistemi veri̇ldi. Teorem, aksiyomların birleşimi ile mantıksal olarak anlamlandırılan sistem terimleri hakkındaki ifadelerdir. Bu sebeple, sistem içindeki bir teoremi resmi olarak bir T önermesi şeklinde öyle ki;

$$(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \vdash T.$$

Hatırlarsak P $\vdash$ Q, P' nin doğru olduğu her durumda doğrudur. Aksiyom sisteminin herhangi bir modelinde aksiyomlar doğru önermeler şeklinde yorumlara sahiptir böylece her teorem doğru önerme şeklinde bir yoruma sahiptir. Bu nedenle teoremler, aksiyom sistemindeki her modelde doğru olan önermeleridir.

O halde bir teoremin ispatını ne oluşturur? Gayri resmi olarak ispat, sonucu teorem olan mantıklı düşüncelerdir. Bir teorem bir kez ispat edildiğinde diğer teoremlerin ispatı için diğer aksiyomlar ile birlikte kullanılabilir. Bundan dolayı, A_i (i=1,2, …,n) aksiyomlar; T_j (j=1,2,…m) ispatlanmış teoremler olmak üzere T teoremini ispat etmek için

$$(A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge T_1 \wedge T_2 \wedge \ldots \wedge T_m) \vdash T$$

olduğunu göstermek gerekir. Bunu aksiyomların doğruluğunu varsayarak ve bunun T' nin doğruluğunu garantilediğini göstererek yaparız.

1.7.1 Koşullu Önermelerin Doğrudan İspatı

Birçok matematiksel varsayım koşullu önerme (P→Q) biçiminde ifade edilebilir. Bu sebeple bunların ispatları, A_i ve T_j aksiyomlar ve teoremler olmak üzere

$$(A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge T_1 \wedge T_2 \wedge \ldots \wedge T_m) \vdash (P\rightarrow Q)$$

olduğunu göstermeyi içerir. Bu

$$(A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge T_1 \wedge T_2 \wedge \ldots \wedge T_m) \rightarrow (P\rightarrow Q)$$

ifadesinin bir tutoloji olduğunu ve $R \rightarrow (P\rightarrow Q)$ ve $(R \wedge P) \rightarrow Q$ mantıksal eşdeğerliliğini kullanarak

$$(A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge T_1 \wedge T_2 \wedge \ldots \wedge T_m \wedge P) \rightarrow Q$$

ifadesinin bir tutoloji olduğunu veya

$$(A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge T_1 \wedge T_2 \wedge \ldots \wedge T_m \wedge P) \vdash Q$$

olduğunu göstermeye denktir. O halde, P→Q şeklindeki teoremlerin direkt ispatı için aksiyomların doğruluğunu ve bundan dolayı tüm ispatlanmış teoremlerin doğruluğunu varsayarız.

Örnek 1.8: Tüm n tamsayıları için, n çift ise n^2'nin de çift olduğunu kanıtlayınız.

İspat: n çift bir tamsayı olsun. Bu halde 2, n'in çarpanlarından biridir ve n, m bir tamsayı olmak üzere n=2m şeklinde ifade edilebilir. Buradan yola çıkarak $n^2=(2m)^2=4m^2$ olur. $4m^2$ ifadesi $2m^2$ tamsayı olmak üzere $2(2m^2)$ şeklinde yazılabilir. Bu sebeple n^2 çifttir.

Dikkat edilirse birçok adımda sebepler göz ardı edilmiştir. Örneğin, $(2m)^2=4m^2$ eşitliğinin doğruluğu için herhangi bir sebep belirtilmedi. Bunun nedeni bu adımın çok açık olması.Öte yandan ciddi bir ispatta eksik detaylar sağlanmalıdır.

1.7.2 Koşullu Önermelerin Ters Pozitif(contrapositive) Kullanarak İspatı

Hatırlarsak ters pozitif $\overline{Q} \rightarrow \overline{P}$, P→Q koşullu önermesi ile mantıksal eşdeğerdir. Bu nedenle, ters pozitifin doğruluğunu sağlarsak koşullu önermenin de doğru olduğu sonucuna varabiliriz. Bu da $\overline{Q} \rightarrow \overline{P}$ 'nun kendisi de koşullu bir önerme olduğundan direkt ispatını kullanabilmemize rağmen P→Q 'nun dolaylı ispatını oluşturur.

Örnek 1.9: Ters pozitifini sağlayarak, her n tamsayısı için n^2 çift ise n de çifttir ifadesini ispatlayınız.

İspat: İspatlanacak ifade P(n) 'n^2 çifttir', Q(n) 'n çifttir' ve n seçilmiş bir tamsayı olmak üzere, P(n) → Q(n)' dir. Ters pozitif ise ~Q(n) →~P(n): n tek ise n^2 tektir. Bu ifadeyi 'n tektir' in doğru olduğunu varsayarak ve n^2 'nin tek olduğunu göstererek kanıtlayabiliriz.

n tek bir tamsayı olsun.

$n=2m+1$ m tamsayı

$\Rightarrow$ $n^2=(2m+1)^2$

$=4m^2+4m+1$

$=2(2m^2+2m)+1$ $(2m^2+ 2m)$ tamsayı

$\Rightarrow$ n^2 tektir.

Örnek 1.10: m ve n birer pozitif tamsayı ve mn=100 ise, $m \leq 10$ veya $n \leq 10$ olduğunu ispatlayınız.

İspat: P(m,n), 'm ve n, mn=100 olan iki rastgele pozitif tamsayı' ve Q(m,n), '$m \leq 10$' ve '$n \leq 10$' önermelerinin dahili birleşimi olmak üzere P(m,n) →Q(m,n) olduğunu göstermek gerekir. De Morgan kuralından $\overline{(p \vee q)} \equiv \overline{p} \wedge \overline{q}$ böylece Q(m,n)'nin tersi 'm>10' ve 'n>10' dur. Ters pozitif ~Q(m,n) →~P(m,n), bu nedenle 'm ve n rastgele tamsayılar olmak üzere m>10 ve n>10 ise mn≠100'.

m ve n pozitif tamsayılar olsun. Böylece,

m>10 ve n>10

$\Rightarrow$ mn>100

$\Rightarrow$ mn≠100

Böylece teorem ispatlanmış olur.

1.7.3 Çelişki(contradiciton) ile İspat

Bir doğruluk tablosu kullanarak f bir çelişki olmak üzere P ve $\overline{P} \rightarrow f$ ' nin mantıksal eşdeğerliklerini kolayca sağlayabiliriz. Bu sebeple T teoremini ispatlamak için bunun yerine $\overline{T} \rightarrow f$ koşullu önermesini ispat edebiliriz. Bu da aksiyomların ve teoremlerin ve de $\overline{T}$' nün doğruluğunu (T' nin yanlışlığını) varsayarak gerçekleştirilebilir. Daha sonra bunun daima yanlış olan bir önerme yani bir çelişki anlamına geldiğini gösterebiliriz. Çoğunlukla, çelişki bir önerme ve tersinin kesişimi $Q \wedge \overline{Q}$ şeklindedir. $\overline{T} \rightarrow f$ 'nin doğru olduğu sonucuna varabiliriz ve bu nedenle T teoremi doğrudur.

Örnek 1.11: $\sqrt{2}$ 'nin rasyonel olmadığını ispatlayınız.(p ve q tamsayı ve q≠0 olmak üzere p/q biçiminde yazılabilen tamsayılara rasyonel sayı denir.)

İspat: Bu teoremin ispatı çelişki ile ispatlamanın bilinen bir örneğidir. $\sqrt{2}$ nin rasyonel olduğunu varsayarak bunun bir çelişkiye neden olduğunu göstermemiz gerekir.

Diyelim ki, $\sqrt{2}$ rasyonel bir sayı ve m ile n tamsayı ve $n\neq 0$ olmak üzere $\sqrt{2}$ =m/n. m/n kesrinin en sadeleşmiş halinde yani m ve n'nin ortak çarpanlarının olmadığını varsayabiliriz. Eğer ortak çarpanları varsa sadeleştiririz. Şimdi,

$$\sqrt{2} = m/n$$

$\Rightarrow$ $2=m^2/n^2$

$\Rightarrow$ $2n^2=m^2$

$\Rightarrow$ m^2 çifttir.

$\Rightarrow$ m çifttir. (Örnek 1.9)

$\Rightarrow$ $m=2p$ herhangi bir p tamsayısı için

$\Rightarrow$ $m^2=4p^2$.

Bu sonucu $2n^2=4p^2$ eşitliğinde yerine koyarsak,

$$2n^2=4p^2$$

$\Rightarrow$ $n^2=2p^2$

$\Rightarrow$ n^2 çifttir

$\Rightarrow$ n çifttir.

Böylece hem m hem de n'nin çift olduğunu yani 2'nin ortak çarpan olduğunu göstermiş olduk. Ancak m ve n hiçbir ortak çarpana sahip değildi çünkü böyle bir çarpan en başta sadeleştirildi. Bu nedenle bir önerme ve tersinin birleşimini yani bir çelişkiyi ortaya çıkardık ve bu da teoremi ispatlamaktadır.

1.7.4 Çift Yönlü koşullu Önermelerin İspatı

Çift yönlü bir önermeyi $P\leftrightarrow Q$, ispat etmek için genellikle $P\leftrightarrow Q$ ve $[(P\rightarrow Q)\wedge (Q\rightarrow P)]$' nin mantıksal eşdeğerliliğine başvururuz. Bu nedenle çift yönlü koşullu önermelerin ispatı iki ayrı bölüm içerir: biri $P\rightarrow Q$'yu diğeri $Q\rightarrow P$' yi ispatlamak.

Örnek 1.12: Herhangi x ve y tamsayıları için xy çarpımının, sadece ve sadece 'x çiftse' veya 'y çiftse' çift olduğunu ispatlayınız.

İspat: Önce direkt ispat kullanarak x çiftse veya y çiftse xy' nin çift olduğunu kanıtlarız.

x çift olsun. Örneğin n bir tamsayı olmak üzere x=2n. O halde xy=2ny yani xy çifttir. Eğer y çift olsaydı benzer bir argüman xy' nin çift olduğunu gösterebilir.

Şimdi tersini ispatlayalım: Eğer xy çift ise x çifttir veya y çifttir. Bunun için ters pozitifin direkt ispatını kullanabiliriz: x ve y tek ise xy de tektir.

O halde x ve y tek olsun.

$\Rightarrow$ $x=2n+1$, $y=2m+1$ m ve n tamsayı olmak üzere

Öyleyse $xy=(2n+1)\quad(2m+1)$

$=4mn+2n+2m+1$

$=2(2mn+n+m)+1$

$\Rightarrow$ xy tektir. Bu da ispat demektir.

1.7.5 Aksine Örneklerin Kullanımı

Birçok matematiksel konjektür, 'tüm A lar B dir' veya 'A özelliğine sahip tüm nesneler B özelliğine sahiptir' biçimindedir. Bu ifade şu şekilde de yazılabilir: A(x) 'x, A dır(A özelliğine sahiptir)' ve B(x) 'x B dir (B özelliğine sahiptir)' olmak üzere $(\forall x)[A(x) \rightarrow B(x)]$.

Önerme şu şekilde de yazılabilir: x A evreni ile sınırlandırılmış olmak üzere $(\forall x)[B(x)]$. Daha önce söylenildiği gibi B özelliğine sahip olmayan bir x bulamamak teoremin ispatını oluşturmaz. Öte yandan B özelliğine sahip birçok x bulmak da bu özelliğe sahip olmayan x bulamayacağımızı garanti etmez. Ancak, evren sonlu bir evrense ve yeterli zaman varsa bütün elemanları kontrol edip özelliğin olup olmadığı sorusunun cevabını bulabiliriz. Eğer tüm elemanlarda bu özellik varsa teorem ispatlanmış olur. Bu yönteme **tüketme ile ispat** denir çünkü x' in bütün olasılıkları tüketilir.

Diğer yandan, $(\forall x)[B(x)]$ biçiminde bir konjektürün yanlış olduğunu ispat etmek için evrendeki sadece bir üyenin B özelliğine sahip olmadığını bulmamız gerekir. Bu aksine örnekle ispatın esasıdır.

1.8 Matematiksel İndüksiyon

Aslında matematiksel indüksiyon diye bilinen ispat yöntemi tümevarımsal bir ispat değildir. Olmamasının nedeni kabul edilen ispatlar sadece tümdengelimsel yargılar barındırır. İndüksiyonun doğruya yakın olan bilgiyi sağlama görevi vardır. Herhangi bir ispatla ilgili problem, onu ispatlamadan önce sonucu bilmemiz gerektiğidir.

Birçok matematiksel konjektür pozitif tamsayıların özellikleri ile ilgilidir. Örneğin şu problem: ilk n tek tamsayının toplamı için bir formül bulun. Başlama noktası olarak n' in küçük bir değerleri için toplamı yazmak ve bunun bize olası konjektür hakkında bir fikir verip vermediğini gözlemektir.

n=1 için, toplam 1' dir.
n=2 için, toplam 1+3=4' tür.
n=3 için, toplam 1+3+5=9' dur.
n=4 için, toplam 1+3+5+7=16' dır.

Bu aşamada n' in her değeri için toplamın n^2 olduğunu fark ederiz. Birkaç tane daha deneyip daha da emin olmak isteriz.

n=5 için, toplam 16+9=25' dur.
n=6 için, toplam 25+11=36' dır.

Tümevarımsal yargı bizi ilk n tek tamsayının toplamı n^2'dir konjektürüne götürür. Bunun tüm pozitif n tamsayıları için doğru olduğunu tümdengelime dayanarak ispatlamalıyız.

Matematiksel indüksiyon sonucun tüm pozitif tamsayılar için geçerli olduğunu ispatlamak için uygundur ve şu adımları içerir:

(a) Konjektürün n=1 için geçerli olduğunu ispatla
(b) Her k≥1 için, eğer sonuç n=k için sağlanıyorsa n=k+1 için de sağlanmalıdır. Bu adım tümevarımsal adım olarak bilinir.

(b) şıkkındaki koşullu önermeyi ispatlamak için bir önceki konuda anlatılan teknikler kullanılır. Öte yandan, tümevarımsal adım genellikle direkt ispat kullanılarak sağlanır. Sonucun n=k için sağlandığını varsayarız. (Bu varsayım bazen tümevarımsal hipotez şeklinde adlandırılır.) Bundan n=k+1 için de sağlandığı sonucunu çıkarırız. n=1 için sağlandığına göre tümevarımsal adım bizi n=2, n=3 vs. için de sağladığı sonucuna götürür. Matematiksel indüksiyonun prensibi, sonucun tüm n pozitif tamsayıları için sağlandığını gösterir.

Örnek 1.13: İlk n tane pozitif tek tamsayının toplamının n^2 olduğunu ispatlayınız.

İspat: İspatlamak istediğimiz şey: $\underbrace{1 + 3 + 5 + \ldots}_{\longleftarrow \text{n terms} \longrightarrow} = n^2$.

Dizideki son eleman 2n-1 'dir ve bu nedenle konjektürümüzü şu şekilde yazabiliriz:

$$1 + 3 + 5 + \ldots + (2n-1) = n^2.$$

Daha sonra aşağıdaki adımları izleriz.

(a) Konjektürün n=1 için doğru olduğunu ispatla.

n=1 için, $1=n^2$. O halde n=1 için konjektür doğrudur.

(b) k≥1 olmak üzere konjektürün n=k için doğru olduğunu varsay ve bunun n=k+1 için konjektürün doğruluğuna yol açtığını göster.

Varsayalım ki, $1 + 3 + 5 + \ldots + (2k-1) = k^2$. Bir sonraki tamsayı olan 2k+1'i eşitliğin iki tarafına eklersek,

$$\begin{aligned} 1 + 3 + 5 + \ldots + (2k-1) + (2k+1) &= k^2+(2k+1) \\ &=(k+1)^2. \end{aligned}$$

Bu eşitliğin sol tarafı ilk k+1 tane tek tamsayının toplamıdır ve tümevarımsal hipotezi kullanarak gösterdik ki, bu toplam $(k+1)^2$'dir. Böylece konjektürün eğer n=k için sağlanıyorsa, n=k+1 için de sağlandığını göstermiş olduk. Ayrıca n=1 için de sağlandığını gösterdik ve matematiksel indüksiyon kuralına dayanarak teorem tüm pozitif n tamsayıları için sağlanır diyebiliriz.

1.8.1 Matematiksel İndüksiyon Prensibinin değişimleri

Tümevarımsal prensip üzerinde değişik modifikasyonlar yapılabilir. Örneğin, S(n) önermesinin sabit bir N tamsayısından büyük veya eşit tüm tamsayılar için sağlandığını ispat etmek isteyelim. Tümevarım prensibi üzerinde bazı modifikasyonlar yaparsak şunu elde ederiz:

(a) S(N)' in doğru olduğunu ispatla.
(b) k≥N' yi sağlayan her tamsayı için, eğer S(k) doğru ise S(k+1) de doğrudur.

Bu tümevarım ile ispatın standart metodudur sadece 1 yerine N ile başlanmıştır.

Tümevarımsal ispatın daha önemli bir modifikasyonu 'indüksiyonun ikinci prensibi' ile sağlanır.

Bunun önemi şudur: Tümevarımsal adıma geldiğimizde S(n)' nin sadece k yerine, k' dan küçük ve eşit tüm pozitif r tamsayıları için doğru olduğunu varsayarız.

İndüksiyonun İkinci Prensibi

S(n) doğal n sayısı ile ilgili bir ifade ve q sabit bir doğal sayı olsun. S(n) 'in tüm n≥ q için doğruluğunun indüksiyon ile ispatı için adımlar;

(a) Temel adım :S(q) nin doğruluğunu ispatla ve,
(b) eğer k≥q için, S(q), S(q+1), S(q+2),…., S(k) doğru ise (tüm q≤k için S(q)' nin doğruluğu S(k+1)'in doğruluğu anlamına gelir.

İndüksiyonun bu ikinci prensibi ilk başta ilkinden daha genel gibi gözükür çünkü S(k+1)'in doğru olduğu sonucuna varmak için daha fazla varsayımda bulunuruz. Ancak, 'S(q), q≤k' yı sağlayan tüm pozitif tamsayılar için doğrudur' önermesine T(n) dersek, ikinci prensibin iki kısmı:

(a) T(q) doğrudur, ve
(b) k≥q için T(q)'nin doğruluğu T(k+1)'in doğruluğu anlamına gelir.

Bu durumda indüksiyonun ikinci adımında öncekine göre daha fazla bilgi gerekir. Buna indüksiyonun kuvvetli prensibi denir. Bu şekle **tam indüksiyon** denir.

Örnek: birden büyük olan herhangi bir doğal sayının asal sayıların çarpımı şeklinde gösterilebileceğini ispatlayın.

S(n), n, birden büyük doğal sayı ise n'in asal sayıların çarpımı olduğunu indüksiyon ile tüm n'ler için gösterelim.

a)Temel adım. S(2) için doğru. 2 asal sayıların çarpımı şeklinde gösterilebilir

b) İndüksiyon adımı: S(2), S(3), …… ,S(k) nın doğruluğu S(k+1)'in doğruluğunu kanıtlar. Şimdi eğer k+1 asal sayı ise doğrudur , eğer k+1 asal sayı değil ise m,n <k olmak üzere k=m.n şeklinde gösterilebilir. İndüksiyon adımı ile m ve n 'nin her ikiside asal sayıların çarpımı olarak gösterilebilir.. Böylece k+1 asal sayıların çarpımı olarak gösterilebilir.

1.8.2 Tümevarımsal Tanımlar (Kümelerin ve fonksiyonların, yinelemeli(rekürsif) tanımları)

Tümevarımsal prensibin kullanımı sadece pozitif tamsayılar hakkındaki önermelerin ispatı ile sınırlandırılmamıştır; matematiksel nesnelerin ve özelliklerin tanımı için de kullanılırlar. Bazı durumlarda nesnelerin açık olarak tanımlanması zordur. Bu durumlarda nesneler kendileri cinsinden tanımlanırlar. Böyle tanımlamaya yinelemeli(rekürsif) tanımamla denir. Yinelemeli tanım, seri, fonksiyon ve kümelerin tanımında kullanılabilir. Örnek olarak, $a_n = 2^n$ (n=0,1,2, ……) olarak verilen 2'nin kuvvetleri dizisi verilsin. Bu diziyi ilk terimi $a_0 = 1$ ve sonraki elemanların öncekiler cinsinden tanımı için bir kural vererek $a_{n+1} = 2.a_n$ (n=0,1,2, ……) şeklinde tanımlanır.

Kümelerin tümevarımla tanımlanması bazı problemlerin çözümünü kolaylaştırır. Bu tanıma indüktif veya yinelemeli(recursive) tanımlama denir. Bir kümenin yinelemeli tanımı üç adımdan oluşur.

1. Temel adım. Tanımlanacak kümenin belirli elemanı kümeye ait olduğu ifade edilir.
2. Indüktif(yinelemeli) adım. Bu adımda kümenin içindeki mevcut elemanları kullanarak

kümenin daha fazla eleman bulundurabileceğini söyler.

3. Kapalı parça. Kümenin içinde 1ve 2 adımda tanımlanan elemanlar olduğunu söyler.

Örnek: 5 ile bölünebilen tamsayılarda oluşan A kümesinin tanımı aşağıdaki adımlardan oluşur.

i. 5 sayısı A'nın bir elemanıdır.

ii. Eğer n , A'nın elemanı ise, n+5'de A'nın elemanıdır.

iii. A'daki bir nesne ancak ve ancak (a) ve (b) adımlarının tekrarlanmasıyla elde edilebilir.

Fonksiyonların yinelemeli tanımı: Eğer bir fonksiyon f(n) ondan önce gelen elemanlar f(i) ler cinsinden tanımlanıyorsa buna yinelemeli(rekürsif) tanım denir. $f(0)$,$f(1)$,$f(2)$,…, $f(k)$'ya da başlangıç değerleri denir. Bir başka ifade ile;

a) Fonksiyonun sıfırdaki değerini ata.

b) Fonksiyonun değerini bir tamsayı olarak hesaplayan ve kendisinden küçük sayılar cinsinden ifade eden bir kural tanımla.

Örnek : F(n) =n! Faktöriyel fonksiyonunu yinelemeli olarak tanımlayalım.

a) fonksiyonun sıfırdaki değeri F(0) = 1

b) F(n+1) 'i F(n) cinsinden hesaplayan kural , (n+1)! 'in n!'den hesaplanabilmesi (n+1) ile çarpılarak olacaktır. Bu durumda kural:

$F(n+1)=(n+1).F(n)$ şeklinde olacaktır.

Aşağıdaki Fibonacci sayıları dizisini ele alırsak:

1, 1, 2, 3, 5, 8, 13, 21,…

Dizideki her bir sayı kendinden önceki iki sayının toplamıdır. f_n n. Fibonacci sayısını temsil ediyorsa, diziyi şu şekilde tanımlayabiliriz:

$f_1 = 1, f_2 = 1$ ve n≥3 için, $f_n = f_{n-1} + f_{n-2}$

Fark edileceği gibi tümevarımsal tanım indüksiyon prensiplerine uymaz. Tümevarımsal tanıma başlamak için, ilk iki Fibonacci sayısını tanımlamamız gerekir, sadece ilkini değil. Aşağıda pozitif n tamsayısına dayanan A_n matematiksel nesne ve özelliğine ait tümevarımsal tanımın genel formu gösterilmiştir.

Tüm pozitif tamsayılar için A_n 'i tanımlamak için:

(a) k=1,2,…r için ayrı ayrı A_k'yi tanımla

(b) k>r için A_k'yi $A_1, \ldots A_{k-1}$ biçiminde tanımla

Bazı nesneleri veya tümevarımsal olarak tanımlanmış bazı özellikleri içeren önermeleri ispatlamak için matematiksel indüksiyonu kullanmak doğaldır.

1.9 Alıştırmalar

1- Aşağıdaki mantıksal eşdeğerlilikleri sağlayınız.

(i) $(p \leftrightarrow q) \equiv (\overline{p \wedge \overline{q}}) \wedge (\overline{q \wedge \overline{p}})$

(ii) $(p \veebar q) \equiv \overline{(\overline{p \wedge \overline{q}}) \wedge (\overline{q \wedge \overline{p}})}$

2- Aşağıdaki argümanların doğruluğunu test ediniz.

(i) Okulu bırakırsam bankada işe başlayacağım. Okulu bırakmıyorum o halde bankada işe başlamayacağım.

(ii) James polis veya futbolcudur. Eğer polisse tabancası vardır. James'in tabancası yoktur o halde James futbolcudur.

3- n>0 olmak üzere n^3+2n 'in 3 ile bölünebildiğini tümevarım ile ispatlayınız.

4- Herhangi üç ardışık tamsayının çarpımının 6 ile bölünebildiğini ispatlayınız.

5- İlk n pozitif tamsayının karelerinin toplamının $\frac{n\,(n+1)\,(2n+1)}{6}$ olduğunu ispatlayınız.

2 Küme Teorisi

2.1 Kümeler ve Üyeler

Küme notasyonu matematikteki temel konseptlerden biridir. Bir kümenin kusursuz bir tanımı burada verilmeyecektir zira küme teorisine göre küme çoğunlukla tanımsızdır. Ancak bu terimle ne demek istediğimizi açıklayabiliriz: hangi tip olursa olsun objeler topluluğu küme olarak düşünülür. Objeler her şey olabilir ve bunlara kümenin elemanları denir. Bir kümedeki elemanların ortak özelliği olmasına gerek yoktur (aslında en bariz ortak noktaları aynı küme içinde bulunmalarıdır). Benzer şekilde eleman sayısında da belli bir kısıtlama yoktur; sonsuz sayıda, sonlu sayıda veya hiç eleman olmayabilir. Diğer yandan tek bir sınırlama vardır: verilen bir küme ve obje ile objenin kümenin elemanı olup olmadığına karar verebilmemiz gerekir.

Örnek 2.1:

1. Bir küme Picasso' yu, Eyfel Kulesini ve π sayısını içerecek şekilde tanımlanmış olabilir. Bu (biraz garip olsa da) sonlu bir kümedir.

2. Tüm pozitif çift tamsayıları içeren küme açıkça sonsuz bir kümedir.

3. Gelmiş geçmiş en iyi 10 şarkıyı içeren kümeyi düşünelim. Eğer en iyinin tanımını vermezsek bu küme geçerli bir küme olmaz. Kime göre en iyi? Bu tanım bir elemanın bir kümenin elemanı olup olmadığına karar verebilmemiz koşuluna uymaz.

2.1.1 Notasyon

Genellikle kümeleri ifade etmek için büyük harfler, elemanları ifade etmek için küçük harfler kullanılır. $\in$ sembolü '-e ait' veya '-nin elemanıdır' anlamına gelir. Bu nedenle

$\alpha \in A$ 'nın anlamı α elemanı A kümesine aittir ve

$\alpha \notin A$ 'nın anlamı $\sim(\alpha \in A)$ veya α A'ya ait değildir.

2.1.2 Kümeleri Tanımlamak

Kümeler değişik biçimlerde tanımlanabilir. En basiti elemanları köşeli parantezler {} arasına listelemektir. Örnek 2.1' deki iki kümeyi bir daha yazarsak:

A={Picasso, Eyfel Kulesi, π }

B={2,4,6,8…..}

İkinci kümede bütün elemanları listelemeyiz. Bu yüzden '…' kullanarak listenin daha böyle devam ettiğini belirtiriz. Diğer küme gösterim örnekleri şunlardır:

Sabit bir pozitif n tamsayısı için, $C_n=\{1,2,\dots n\}$, ilk n pozitif tamsayının kümesidir. Yine sonlu sayıda olmasına rağmen arada birçok elemanın var olduğunu göstermek için '…' kullandık.

D={}, **boş küme**dir yani hiçbir elemanı yoktur. Bu küme genellikle $\varnothing$ ile gösterilir.

Bir kümenin elemanlarını listelemek küçük veya belli bir kalıba sahip elemanlı kümeler haricinde pek pratik değildir. Alternatif bir yol küme elemanlarını bir özellik ile tanımlamaktır. Daha açık bir ifadeyle, P(x) tek değişkenli bir önermesel fonksiyon ise elemanları, α için P(α)'nın doğru bir önerme olduğu tüm α objeleri olan kümeyi oluşturabiliriz.

Bu şekilde tanımlanan bir küme; A={x:P(x)} şeklinde ifade edilir. (Bu şu şekilde okunur: P(x)' i sağlayan tüm x'lerin kümesi)

2.1.3 Kümelerin Eşitliği

İki küme sadece ve sadece aynı elemanları içeriyorsa eşit olarak tanımlanır; şöyle ki, eğer $(\forall x)\left[x \in A \leftrightarrow x \in B\right]$ doğru ise A=B' dir yada tersi. Listelenen elemanların sırası önemsizdir.

Şunu da unutmamak gerekir ki; sadece bir boş küme vardır veya tüm boş kümeler eşittir. Çünkü tüm boş kümeler aynı elemanı içerir yani hiçbir elemanı.

Ayrıca, eğer P(x) ve Q(x) aynı x objeleri için doğru olan önermesel fonksiyonlar ise tanımladıkları kümeler eşittir.

$$\{x:P(x)\}=\{x:Q(x)\}.$$

Tanım: Eğer A sonlu bir küme ise kardinal itesi, |A|, içerdiği (farklı) elemanların sayısıdır.

Eğer A sonsuz sayıda elemana sahipse, sonsuz kardinalitesi vardır deriz ve şu şekilde ifade ederiz: |A|=∞.

A' nın kardinalitesi için kullanılan diğer notasyonlar n(A), #(A) ve $\overline{\overline{A}}$.

Örnek 2.2:

4. $|\varnothing|=0$ çünkü $\varnothing$' nın hiç elemanı yoktur.
5. |{ π ,2,Einstein}|=3.
6. Eğer X={0,1,……..,n} ise |X|=n+1.
7. |{2,4,6,8,…..}| = ∞

Kardinalite basit bir konsept gibi görünse de verilen bir kümenin kardinalitesini hesaplamak bazen pratikte zor olabilir. Bu durum genellikle verilen kümenin elemanlarından bazıları kendileri birer küme olduğunda gerçekleşir. Küme elemanlarının kendi başlarına bir küme olması geçerli bir yapıdır.

Örneğin, X={{1,2}} olsun. Bu durumda X sadece tek bir eleman içerir yani {1,2} kümesini ve |X|=1' dir. Kardinalitesi 2 olan {1,2} kümesi ile tek elemanı {1,2} kümesi olan X kümesini ayırt etmek son derece önemlidir. Benzer şekilde $\varnothing$ ve $\{\varnothing\}$ kümeleri de farklıdır zira $|\{\varnothing\}|=1$' dir.

Örnek 2.3: |{1,2,{1,2}}|=3,

$|\{\varnothing,\{1,2\}\}|=2$,

$|\{\varnothing,\{\varnothing\}\}|=2$,

$|\{\varnothing,\{\varnothing\},\{1,2\}\}|=3$,

$|\{\varnothing,\{\varnothing,\{\varnothing\}\}\}|=2$.

2.2 Alt Kümeler

Tanım: A' nın tüm elemanları aynı zamanda B' nin de elemanları ise A kümesi B kümesinin alt kümesidir denir ve $A\subseteq B$ şeklinde gösterilir. Sembolik olarak, $(\forall x)[x \in A \rightarrow x \in B]$ ise $A\subseteq B$' dir.

Eğer A B' nin alt kümesi ise B, A 'nın süper kümesidir (superset) veya kapsar deriz ve $B\supseteq A$ yazarız. $A\subset B$ notasyonu 'A, B 'nin tam alt kümesidir' ifadesi için kullanılır. Bu nedenle sadece ve sadece $A\subseteq B$ ve $A\neq B$ ise $A\subset B$' dir. Ayrıca tüm A kümeleri için $\varnothing\subseteq A$' dir.

İki kümenin eşit olduğunu kanıtlamak için her birinin diğerinin alt kümesi olduğunu göstermek yeterlidir. Esasen bu, aşağıdaki bileşik önemelerin mantıksal eşdeğerliliğinden kaynaklanır.

$$[(p \rightarrow q) \wedge (q \rightarrow p)] \equiv (p \leftrightarrow q).$$

Alt kümenin tanımı ile $A\subseteq B$' nin anlamı $(\forall x)[x \in A \rightarrow x \in B]$ doğrudur ve $B\subseteq A$ 'nin anlamı $(\forall x)[x \in B \rightarrow x \in A]$ doğrudur, bu durumda $A\subseteq B$ ve $B\subseteq A$ sadece ve sadece $(\forall x)[x \in A \rightarrow x \in B] \wedge (\forall x)[x \in B \rightarrow x \in A]$ ise doğrudur. $(\forall x)[x \in A \rightarrow x \in B]$' nin ve $(\forall x)[x \in B \rightarrow x \in A]$' nin doğruluğu $(\forall x)[x \in A \leftrightarrow x \in B]$'nin doğruluğunu garantiler ve tam tersi. Bu sebeple, A=B olduğu zaman $A\subseteq B$ ve $B\subseteq A$ ifadelerinin ikisi de doğrudur. Özet olarak:

Teorem 2.1: İki küme; A ve B sadece ve sadece $A\subseteq B$ ve $B\subseteq A$ ise eşittir.

Örnek 2.4: A={{1},{2},{1,2}} ve B {1,2}' nin boş olmayan tüm alt kümeleri olsun. A=B olduğunu gösteriniz.

Çözüm: $A\subseteq B$ 'dir çünkü A' nın 3 elemanının her biri {1,2}'nin boş olmayan alt kümesidir ve bu nedenle B' nin bir elemanıdır.

$B\subseteq A$' dır çünkü {1,2}' nin boş olmayan tüm alt kümeleri A' da yer alır. Yukarıdaki teoremi kullanarak A=B sonucuna varabiliriz.

Küme konsepti çok geniş olduğundan çoğunlukla belirli konteks için gerekli olan kümelere önem verilir. Mevcut görevi veya çalışmayı ilgilendiren kümeleri içine alan evrensel bir küme tanımlamak uygundur. Evrensel kümenin dışında kalan her şey göz ardı edilir. Evrensel küme her zaman için sabit olan bir şey değildir- kontekse göre değişir.

Evrensel küme olarak kullanılan bazı özel sayı kümeleri aşağıdadır.

N= {0,1,2,3,….} doğal sayılar kümesi.

Z= {…, -2,-1,0,1,2,…} tam sayılar kümesi.

Q= {p/q:p,q$\in$Z ve q$\neq$0} rasyonel sayılar kümesi.

$\Re$ = reel sayılar kümesi; reel sayılar sayı doğrusu üzerindeki noktalar veya ondalık şeklinde yazılan sayılar şeklinde düşünülebilir.

$C=\{x+iy: x,y \in \Re$ ve $i^2=-1\}$ kompleks sayılar kümesi.

Açıkça görüldüğü gibi bu kümeler arasında şu alt küme ilişkileri vardır:

$$N \subseteq Z \subseteq Q \subseteq \Re \subseteq C.$$

Ayrıca Z^+, Q^+ ve R^+ sırasıyla pozitif tamsayıları, rasyonel sayıları ve reel sayıları ifade etmek için kullanılır. Dikkat edilirse N, Z^+ 'ya eşit değildir zira 0 ilkine dahil olmasına rağmen ikincisine değildir. Ek olarak, bazen çift ve tek sayıları ifade etmek için E ve O'yu kullanırız:

$$E=\{2n: n \in Z\}=\{\ldots,-4,-2,0,2,4,\ldots\}$$

$$O=\{2n+1: n \in Z\}=\{\ldots,-3,-1,1,3,5,\ldots\}.$$

Eğer evrensel bir küme $\{x:P(x)\}$ notasyonu ile tanımlanmış ise bunun anlamı P(x)'i sağlayan evrensel kümedeki tüm x' lerin kümesidir. Bu nedenle eğer mevcut evrensel kümemiz Z ise $X=\{x:2x^2+3x-2=0\}$, $\{-2\}$ kümesidir fakat *U,* Q veya R ise $X=\{-2,1/2\}$. İlk durumda sınırlandırmayı daha belirgin yapabilirdik ve şekilde yazabilirdik:

$$X=\{x: x \in Z \text{ ve } 2x^2+3x-2=0\} \text{ veya } X=\{x \in Z : 2x^2+3x-2=0\}.$$

2.3 Kümeler Üzerinde İşlemler

Venn şeması kümelerin yararlı bir görsel gösterimidir. Böyle bir şemada kümeler, düzlemdeki bölgeler olarak temsil edilir ve verilen kümeye ait elemanlar kendisini temsil eden bölgenin içine yerleştirilir. Bazen tüm kümeler evrensel kümeyi temsil eden bir kutuya yerleştirilir. Eğer bir eleman iki kümenin birden elemanı ise iki küme iç içe çizilir ve bu elemanlar iç içe geçmiş kısma konur.

Verilen A ve B kümeleri ile aşağıdaki gibi yeni iki küme tanımlayabiliriz.

A ve B' nin **kesişimi**, A ve B' nin her ikisine birden ait olan tüm elemanların kümesidir ve $A \cap B$ şeklinde gösterilir.

A ve B' nin **birleşimi**, A' ya, B' ye veya her ikisine ait olan tüm elemanların kümesidir ve $A \cup B$ şeklinde gösterilir.

Sembolik olarak;

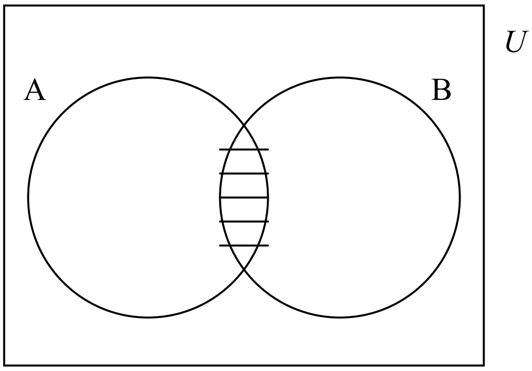
$$A \cap B=\{x: x \in A \text{ ve } x \in B\}$$
$$A \cup B=\{x: x \in A \text{ veya } x \in B \text{ veya her ikisi birden}\}.$$

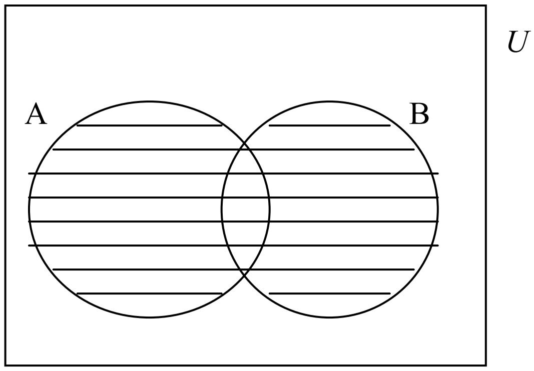
Kümelerin kesişimi ile önermelerin kesişimi arasında açık bir bağlantı vardır tıpkı kümelerin birleşimi ve önermelerin dahili birleşimi arasında olduğu gibi. Eğer A ve B, sırasıyla P(x) ve Q(x) önermesel fonksiyonları ile tanımlanmışlar ise;

$$A \cap B=\{x: P(x) \wedge Q(x)\} \text{ ve}$$

$$A \cup B=\{x: P(x) \vee Q(x)\}.$$

Bu kümeler en iyi aşağıdaki Venn şemaları ile gösterilebilir. Taralı bölgeler kesişim ve birleşimi gösterir.





Şekil 2.1: $A \cap B$ **Şekil 2.2:** $A \cup B$

Kesişim ve birleşimin tanımlarını ikiden fazla kümeye genişletebiliriz. A_1, A_2, … A_n küme olsun.

Bunların kesişimi:

$$\bigcap_{r=1}^{n} A_r = A_1 \cap A_2 \cap \ldots \cap A_n$$

$$= \{x: x \in A_1 \text{ ve } x \in A_2 \text{ ve } \ldots \text{ ve } x \in A_n\}$$

=\{x: x, r=1,2,…,n olmak üzere her bir A_r kümesine aittir.\}

Birleşimi ise;

$$\bigcup_{r=1}^{n} A_r = A_1 \cup A_2 \cup \ldots \cup A_n \quad = \{x: x \in A_1 \text{ veya } x \in A_2 \text{ veya } \ldots \text{ veya } x \in A_n\}$$

=\{x: x, r=1,2,…,n olmak üzere en az bir A_r kümesine aittir.\}

A ve B kümeleri ortak elemana sahip değilse **ayrık**tır denir yani $A \cap B = \emptyset$. Venn şemasında bu iç içe geçmemiş kümeler şeklinde gösterilir.

Verilen bir A kümesinin **tümleyeni**, A ‘ya ait olmayan fakat *U*’da yer alan tüm elemanlardır. A’nın tümleyeni $\overline{A}$ (veya A’) şeklinde gösterilir. Tümleyen ile tersini alma arasında açık bir ilişki vardır; eğer A=\{x: P(x)\} ise $\overline{A}$=\{x: ~P(x)\}’ tir.

Bir kümenin tümleyeni ile bağlantılı olarak A ve B kümelerinin **farkı** A-B veya A\B şeklinde gösterilir ve bu küme A’nın B’de yer almayan tüm elemanlarını içerir:

$$A-B=\{x: x \in A \text{ ve } x \notin B\}.$$

A’nın tümleyeni A’=*U*-A’dır.

Örnek 2.5: $U=\{1,2,3,\ldots,10\}=\{n: n \in Z^+ \text{ ve } n \le 10\}$, $A=\{n \in U : 1 \le n < 7\}$, $B=\{n \in U$: n 3’ün katları\} olsun. O halde; A=\{1,2,3,4,5,6\} ve B=\{3,6,9\}. Bu nedenle:

$$A \cap B = \{3,6\}$$

$A \cup B = \{1,2,3,4,5,6,9\}$

$A - B = \{1,2,4,5\}$

$B - A = \{9\}$

$\overline{A} = \{7,8,9,10\}$

$\overline{B} = \{1,2,4,5,7,8,10\}$

$\overline{A \cup B} = \{7,8,10\} = \overline{A} \cap \overline{B}$

$\overline{A \cap B} = \{1,2,4,5,7,8,9,10\} = \overline{A} \cup \overline{B}$

2.4 Sayma Teknikleri

Bazı kompleks matematiksel sonuçlar sayma argümanlarının ispatlarına bağlıdır: çeşitli kümelerin eleman sayılarını saymak, belli bir sonucun kaç değişik yolla elde edilebileceğini saymak gibi. Sayma kısmen kolay bir olay gibi görünse de, pratikte çok kompleks olabilir. Matematikçiler sayma problemleri için birçok teknik ve sonuç üretmişlerdir ve konuya sayma teorisi adını vermişlerdir.

Saymanın en basit sonuçlarından biri şudur: iki ayrık A ve B kümesinin toplam eleman sayısını bulmak için A'nın elemanlarını, B'nin elemanlarını sayıp toplarız.

Sayma Prensibi 1: Eğer A ve B ayrık iki küme ise $|A \cup B| = |A| + |B|$.

Çoğu uygulama doğal olarak ikiden fazla küme içerir. Yukarıdaki prensip aşağıdaki şekilde genelleştirilir.

Sayma Prensibi 2: Eğer $A_1, A_2, \ldots, A_n$ küme ise ve bu kümelerin hiçbir çifti ortak bir elemana sahip değilse $|A_1 \cup A_2 \cup \ldots \cup A_n| = |A_1|+|A_2|+\ldots+|A_n|$.

Bazen, elemanları sayılacak kümeler yukarıdaki sayma prensiplerinin katı kuralını-herhangi bir çiftin ayrık olması- sağlamayabilir. Öte yandan, bu durumda kümeyi sayma prensiplerinin koşullarını sağlayacak alt kümeler bölmek mümkündür. Bu şekilde ispatlanabilecek en basit sonuç şudur:

Teorem 2.2(Ekleme(inclusion)-Çıkarma(exclusion) Prensibi): Eğer A ve B sonlu kümeler ise $|A \cup B|=|A|+|B|-|A \cap B|$.

İspat: $A \cup B$' yi sayma prensibi 2'yi sağlayan alt kümelerine böleriz: A-B, $A \cap B$ ve B-A.

Sayma prensibi 2' den,

$$|A \cup B| = |A-B|+|A \cap B|+|B-A|. \qquad (1)$$

A ve B kümelerinin kendileri sırasıyla A-B, $A \cap B$ ve B-A, $A \cap B$ şeklinde ayrık alt kümelere bölünebilir. Böylece;

$$|A|=|A-B|+|A \cap B| \qquad (2)$$

$$|B|=|B-A|+|A \cap B|. \qquad (3)$$

Bu durumda (1), (2) ve (3) eşitliklerini birleştirerek istenilen sonucu elde etmek çok kolay bir işlemdir. Ekleme-çıkarma prensibi bu şekilde adlandırılır çünkü $A \cup B$' nin elemanlarını saymak için A'nın elemanlarını ve B'nin elemanlarını ekledik ve böylece $A \cap B$' nin elemanlarını iki kere eklemiş olduk. $A \cup B$' nin doğru eleman sayısını elde etmek için $A \cap B$'yi bir kere çıkarmak gerekir.

İki kümeden fazla durumlar için benzer sayma teknikleri vardır. Üç küme için sonuç aşağıdaki teoremdeki gibi bulunur.

Teorem 2.3: A, B ve C sonlu kümeler ise

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|.$$

2.5 Kümeler Cebri

Açıktır ki, kesişim, birleşim ve tümleyen işlemleri birbiriyle ilişkilidir. Örneğin;

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

Aşağıdaki kurallar tüm A, B ve C kümeleri için geçerlidir.

Aynılık (Tek Kuvvet) Özelliği

$A \cap A = A$
$A \cup A = A.$

Değişme Özelliği

$A \cap B = B \cap A$
$A \cup B = B \cup A$

Birleşme Özelliği

$A \cap (B \cap C) = (A \cap B) \cap C$
$A \cup (B \cup C) = (A \cup B) \cup C$

Yutan Eleman

$A \cap (A \cup B) = A$
$A \cup (A \cap B) = A.$

Dağılma Özelliği

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Çift ters Özelliği((Double negation)

$\overline{\overline{A}} = A.$

De Morgan Kuralları

$\left(\overline{A \cup B}\right) = \overline{A} \cap \overline{B}$
$\left(\overline{A \cap B}\right) = \overline{A} \cup \overline{B}.$

Özdeşlik Özelliği

$A \cup \varnothing = A$
$A \cap U = A$
$A \cup U = U$
$A \cap \varnothing = \varnothing$.

Tamlama Özelliği

$A \cup \overline{A} = U$
$A \cap \overline{A} = \varnothing$
$\overline{\varnothing} = U$
$\overline{U} = \varnothing$.

Bu kurallar uygun önermeler arasındaki mantıksal eşitliklerden de türetilebilmesine rağmen en iyi Venn şemaları ile gösterilir.

2.5.1 Eşlik Kuralı (Duality Principle)

$\wedge,\vee$ ve tersini alma bağlayıcılarını içeren bileşik önermelerin eşli önermeye sahip olduğu gibi $\cap,\cup$ ve tümleme içeren kümeler hakkındaki ifadeler de eşlidir. Böyle bir ifadenin eşi orijinal ifadedeki tüm $\cap$' lerin $\cup$ ile; tüm $\varnothing$' lerin U ile değiştirilmesi ile elde edilir. Örneğin;

$(A \cap \varnothing) \cup (B \cap U) \cup \overline{B} = U$ 'in eşi $(A \cup U) \cap (B \cup \varnothing) \cap \overline{B} = \varnothing$.

Kümeler cebrinin her bir kuralının eşi de ayrıca bir kuraldır. Bunun sonucu olarak kümeler için aşağıdaki eşlik kuralı ortaya çıkmıştır.

Kümeler için eşlik kuralı: Eğer kümeler ile ilgili bir ifade tüm kümeler için doğruysa bunun eş ifadesinin de tüm kümeler için doğru olması gerekir.

2.6 Kümelerin Aileleri

Kümelerin ailesi veya kümlerin toplanması terimiyle, kümelerin kümesi kastedilmekte ise de her iki terimde sıklıkla kullanılmaktadır.

$I=\{1,2,\ldots,n\}$ verilsin ve $\forall i \in I$ *için,* A_i kümesi aşağıdaki şekilde tanımlanır.

$\{A_i : i \in I\} = \{A_1, A_2, \quad, A_n\}$

I kümesine gösterge kümesi denir ve Ai 'leri birleşme için göstergeler. Eğer, $I=\{1,2,n\}= Z^+$ ise, $\{A_i : i \in I\} = \{A_1, A_2, A_3, \ldots\ldots\}$ dır.

Bu notasyonu kullanarak, kümelerin keyfi bir ailesine kesişim ve birleşimi aşağıdaki şekilde tanımlanır.

$\boldsymbol{F} = \{A_i : i \in I\}$ kümelerin bir ailesi olarak verilsin burada, I herhangi bir gösterge kümesidir.

F ailesinin Kesişim ve birleşimi : $\bigcap_{i \in I} A_i = \{x : x \in A_i$ bütün $i \in I$ için $\}$; $\bigcup_{i \in I} A_i = \{x : x \in A_i$ bazı $i \in I$ için $\}$

Örnek: I= Z^+ = {1,2,3,…..,} ve her bir i∈ Z^+ için A_i = {i} . Böylece, A_1= {1}, A_2= {2}, ...

Bu nedenle: $\bigcap_{i \in Z^+} A_i = \varnothing$; ve $\bigcup_{i \in Z^+} A_i$ = {1,2,3,…..,} = $\mathbf{Z}^+$ dir.

2.6.1 Kuvvet Kümesi

Verilen bir A kümesinin bütün alt kümelerinin kümesine A'nın kuvvet kümesi denir ve P(A) ile gösterilir. P(A) = {B : B ⊆ A} dir

Örnek1 : A={ a,b } ise P(A)= { ∅,{a},{b},{a,b}} dir.

Örnek2 : A herhangi bir küme olsun ve kümelerin sırası A, P(A), P(P(A)), P(P(PA))), ... dır.

P*(A) bu ailede A'nın tüm elemanlarının kümesinin ailesini temsil eder.

P*(A) ={x: x ∈ A veya x ⊆ y burada , y∈ P*(A)} dır ve P*(A) sonsuz bir kümedir.

2.6.2 Bir Kümenin Bölmelenmesi

A bir küme olsun. A'nın bölmelenmesi, A'nın boş olmayan alt kümeleri { S_i : i∈I } dir öyleki;

i) $\bigcup_{i \in I} S_i = A$, ve

ii) $S_i \cap S_j = \varnothing$ eğer, bütün i,j ∈ I için i ≠j ise) dir

Örnek 1: a={1,2,3,4,5,6} ise A'nın bölmelenmesi {{1},{2,3},{4,5,6}} dır.

Örnek 2: herbir α gerçel sayısı için L_α , (α,0) noktasından geçen düşey çizgi üzerindeki noktaların kümesi olsun:

L_α = x = α ve y gerçel birsayıdır} = {(α,y) : y ∈ ***R***} dir

{ L_α : α ∈ ***R*** } kümelerinin ailesi düzlemi bölmeler : L_α çizgileri üzerindeki herbir nokta ve herhangi iki çizgi birbirinden ayrıdır.

2.7 Kartezyen Çarpım

Bir kümenin elemanlarının hangi sıra ile listelendiği önemsizdir. Öte yandan bazı durumlarda sıra çok önemlidir. Örneğin, koordinat geometride (1,2) noktası ile (2,1) noktası faklıdır.

Sıranın önemli olduğu durumlarla başa çıkmak için x ve y objelerinin **sıralı ikili** (x,y)'yi tanımlarız.

(x,y) = (x',y') sadece ve sadece x=x' ve y=y' ise.

Bu tanımla birlikte açıktır ki (x,y) ile (y,x) faklıdır(x=y değilse) ve sıra önemlidir.

Şu an ileriki bölümlerde temel teşkil edecek iki kümenin kartezyen çarpımı kavramını tanımlayabilecek durumdayız.

Tanım: X ve Y kümelerinin kartezyen çarpımı X x Y, x X' e y de Y' ye ait olmak üzere tüm (x,y) sıralı ikililerinin kümesidir.

$$X \times Y = \{(x,y): x \in X \text{ ve } y \in Y\}.$$

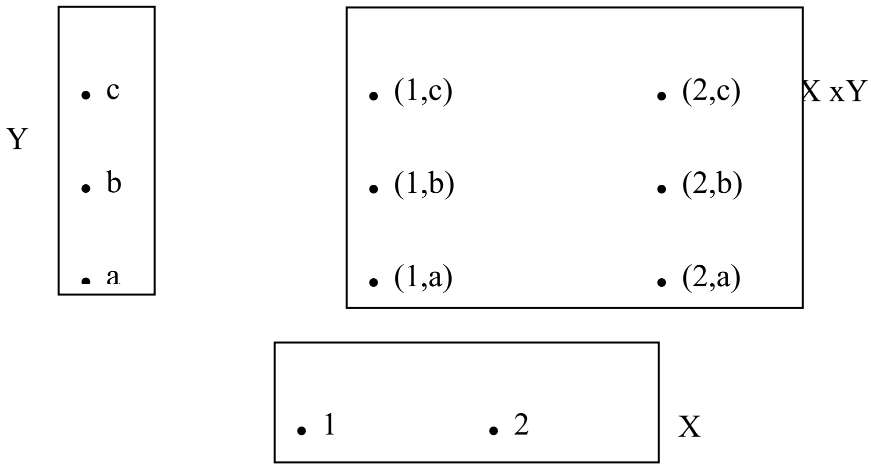
X=Y olması durumunda X x X, X^2 ile gösterilir ve 'X iki' şeklinde okunur, 'X kare' şeklinde değil.

Eğer X veya Y (veya her ikisi de) boş küme ise X x Y de boş kümedir. X ve Y' nin her ikisi de boş olmayan kümeler ise sadece ve sadece X=Y ise X x Y= Y x X 'dir.

Örnek 2.6: Eğer X={1,2} ve Y={a,b,c} ise

$$X \times Y = \{(1,a), (1,b), (1,c),(2,a),(2,b),(2,c)\}.$$

X, Y ve X x Y' nin elemanları basit bir Venn şeması ile sistematik olarak gösterilebilir. Bu Venn şeması Şekil 2.3' teki gibidir.



Şekil 2.3

Şekil 2.3 gibi diyagramlar ve R^2= R x R düzleminin koordinat geometri resimleri Kartezyen çarpımının yararlı gösterimleridir. Farklı bir gösterimde ise X ve Y kümeleri Venn diyagramlarındaki gibi iki boyutlu bölgeler yerine tek boyutlu bölgeler olarak çizilir. X ve Y doğru segmentleri olarak çizilir ve elemanları bu doğru segmentinin üzerine yerleştirilir. Uygun olan X'i temsil eden doğrunun yatay olarak çizilmesi ve doğruların birbirine dik olmasıdır. Kartezyen çarpım X 'in üzerinde,Y' nin sağında bulunan dikdörtgensel bölgedir ve (x,y) sıralı ikilileri bu dikdörtgenin içine noktalar dikey olarak x' in üzerine, yatay olarak y' nin sağına gelecek şekilde yerleştirilir.

(x,y) sıralı ikilisi aşağıdaki özellik yardımıyla sıralı n-elemanlı (ordered n-tuple) şekle genelleştirilebilir.

$$(x_1,x_2,\ldots,x_n)=(x_1',x_2',\ldots,x_n') \quad \text{sadece ve sadece } x_1= x_1', x_2= x_2',\ldots, x_n= x_n' \text{ ise.}$$

n tane kümenin kartezyen çarpımı iki kümedeki durumun doğal genelleştirilmesidir.

Tanım: $X_1, X_2, \ldots, X_n$ kümelerinin kartezyen çarpımı $X_1 \times X_2 \times \ldots, \times X_n$ 'dir.

$$= \{(x_1,x_2,\ldots,x_n): x_1 \in X_1 \text{ ve } x_2 \in X_2 \text{ ve } \ldots \text{ ve } x_n \in X_n\}$$
$$= \quad (x_1,x_2,\ldots,x_n): \quad x_i \in X_i \quad i=1,2,\ldots,n \text{ olmak üzere}\}.$$

Örnek 2.7: $A=\{1,2\}$, $B=\{a,b\}$ ve $C=\{\alpha, \beta\}$ ise
$A \times B \times C = \{(1,a,\alpha), (1,a,\beta), (1,b,\alpha), (1,b,\beta), (2,a,\alpha), (2,a,\beta), (2,b,\alpha), (2,b,\beta)\}$.

Üç kümeden oluşan kartezyen çarpımları göstermek kolay değildir fakat üç boyutlu bölgeler ile gösterilebilecekleri açıktır.

X ve Y sonlu kümeler olmak üzere $|X|=n$ ve $|Y|=m$ ise açıktır ki kartezyen çarpım X x Y mn elemana sahiptir. Öyle ki;

$$|X \times Y| = |X| \cdot |Y|.$$

Bu sonuç aşağıdaki gibi n tane küme için genelleştirilebilir.

Teorem 2.4: $X_1, X_2, \ldots, X_n$ sonlu kümeler ise $|X_1 \times X_2 \times X_n| = |X_1| \cdot |X_2| \cdot \ldots \cdot |X_n|$.

Kartezyen çarpım işleminin kesişim ve birleşim gibi diğer küme teoremi işlemleri ile nasıl davranacağına geçmeden önce aşağıdaki örneğe bakalım.

Örnek 2.8: $A=\{a,b,c,d\}$, $X=\{x,y,z\}$, $Y=\{y,z,t\}$ olsun. Bu durumda

$X \cap Y=\{y,z\}$ olur ve
$A \times (X \cap Y)=\{(a,y),(a,z),(b,y),(b,z), (c,y),(c,z), (d,y),(d,z)\}$'dır. Şimdi,
$A \times X=\{(a,x),(a,y),(a,z), (b,x),(b,y),(b,z),(c,x),(c,y),(c,z), (d,x),(d,y),(d,z)\}$ ve
$A \times Y=\{(a,y),(a,z),(a,t), (b,y),(b,z),(b,t),(c,y),(c,z),(c,t), (d,y),(d,z),(d,t)\}$ olur. Bu nedenle,
$(A \times X) \cap (A \times Y)=\{ (a,y),(a,z), (b,y),(b,z), (c,y),(c,z), (d,y),(d,z)\}$ olur.

O halde bu örnekteki kümeler için;

$A \times (X \cap Y)=(A \times X) \cap (A \times Y)$ olduğuna göre bu özelliğin diğer A, X ve Y kümeleri için de doğru olup olmadığına bakabiliriz.

Aslında yukarıdaki örnekte elde ettiğimiz sonuçlar tüm A, X ve Y kümeleri için geçerlidir. Aşağıdaki teoremde Kartezyen çarpımın kesişim ve birleşim işlemlerinde nasıl davrandığını belirten özellikler listelenmiştir.

Teorem 2.5 (i) Tüm A, X ve Y kümeleri için
$A \times (X \cap Y) = (A \times X) \cap (A \times Y)$ ve
$(X \cap Y) \times A = (X \times A) \cap (Y \times A)$. (Bunun anlamı Kartezyen çarpım kesişim üzerine dağılabilir.)
(ii) Tüm A, X ve Y kümeleri için
$A \times (X \cup Y) = (A \times X) \cup (A \times Y)$ ve
$(X \cup Y) \times A = (X \times A) \cup (Y \times A)$. (Bunun anlamı Kartezyen çarpım birleşim üzerine dağılabilir.)

İspat: (i). kısmın ispatı şu şekildedir.

$(a,x) \in A \times (X \cap Y)$ olsun. Kartezyen çarpımın tanımından bunun anlamı $a \in A$ ve $x \in (X \cap Y)$' dır. Bu sonuçla, $x \in X$' tir, öyleyse (a,x) A x X' e aittir; $x \in Y$' tir, öyleyse (a,x) A x Y' e aittir. Bu nedenle, $(a,x) \in (A \times X) \cap (A \times Y)$' dir ki bu da $A \times (X \cap Y) \subseteq (A \times X) \cap (A \times Y)$ olduğunu ispatlar.

Alt küme ilişkisini diğer taraftan ispatlamak için; $(a,x) \in (A \times X) \cap (A \times Y)$ olsun.

Bu durumda $(a,x) \in (A \times X)$ 'tir öyleyse $a \in A$ ve $x \in X$; ayrıca $(a,x) \in (A \times Y)$'tir öyleyse $a \in A$ ve $x \in Y$' dir. Bu nedenle $a \in A$ ve $x \in (X \cap Y)$' dir ve bunun anlamı (a,x) sıralı ikilisi $A \times (X \cap Y)$ kartezyen çarpımına aittir. Bundan dolayı $(A \times X) \cap (A \times Y) \subseteq A \times (X \cap Y)$ olmalıdır.

Şu halde $A \times (X \cap Y)$ ve $(A \times X) \cap (A \times Y)$ kümelerinin eşit olduğu sonucu sağlanmış olur zira her iki küme de birbirinin alt kümesidir.

Son olarak Kartezyen çarpımın alt küme ilişkilerinde nasıl davranacağına ilişkin bir teorem yazabiliriz.

Teorem 2.6: (i) Tüm A, B ve X kümeleri için $A \subseteq B$,$(A \times X) \subseteq (B \times X)$ anlamına gelir.

(ii) Eğer X boş olmayan bir küme ise $(A \times X) \subseteq (B \times X)$, $A \subseteq B$ anlamına gelir.

2.8 Alıştırmalar

1- A ve B' nin ortak elemanı yoksa ve $C=\{x: x \in A \wedge x \in B\}$ ise C' nin boş küme olduğunu kanıtlayınız.

2- $\{x: 2x^2+5x-3=0\} \subseteq \{x: 2x^2+7x+2=3/x\}$ olduğunu ispatlayınız.

3- Aşağıdaki ifadelerin doğruluğunu göstermek için Venn şemalarını çiziniz.

(i) $(\overline{A-B})=B \cup \overline{A}$
(ii) $(A-B) \cup (B-A)=(A \cup B)-(A \cap B)$

4- $[0,1]=\{x \in IR: 0 \leq x \leq 1\}$ $(0,1)=\{x \in IR: 0<x<1\}$

$[0,1)=\{x \in IR: 0 \leq x<1\}$ $(0,1]=\{x \in IR: 0<x \leq 1\}$

olsun. Bu durumda aşağıdaki kümeleri geometrik olarak tanımlayınız.

(i) $[0,1] \times [0,1)$
(ii) $[0,1) \times (0,1]$

5- X x Y = X x Z ise Y=Z olmak zorunda mıdır? Açıklayınız.

3 Bağıntılar ve Fonksiyonlar

Bağıntı notasyonu kümeler gibi çok genel bir notasyondur. Bu konu matematiğin anahtar konularından biridir ve başka birçok konuda da kullanılır. Üç özel tip bağıntı çok önemlidir: fonksiyonlar, eşitlik bağıntıları ve sıra bağıntıları.

3.1 Bağıntılar ve Gösterimleri

İkili yüklemler yani "…daha ağırdır" şeklindeki cümleleri önermesel fonksiyona çevirmek için iki tane değişkene ihtiyaç vardır. Örneğin, H '…daha ağırdır' anlamına geliyorsa H(x,y) "x, y' den daha ağırdır" önermesel fonksiyonunu ifade eder. İki değişkenli önermesel fonksiyonları iki değişkeni arasındaki ilişkiyi tanımlamak gibi düşünebiliriz. a ve b objeleri verilmiş ise, H(a,b) sadece ve sadece objeleri uygun biçimde ilişkilendirilmiş ise doğrudur.

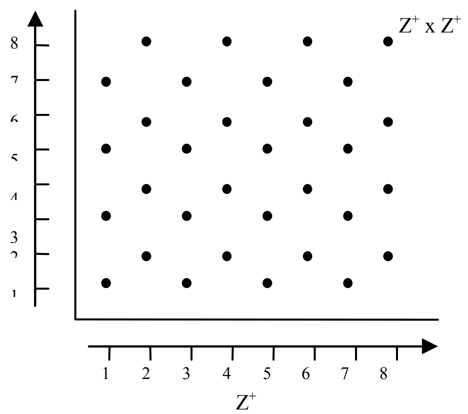
Hatırlanması gereken ilk şey, iki değişkenli önermesel fonksiyon F(x,y)'de değişkenlerin sırasının önemli olduğudur. Özel a ve b objeleri için F(a,b) ile F(b,a) farklı doğruluk değerine sahip olabilir. Önemli olan bir başka şey de x ve y değişkenlerinin farklı tip objeler olabileceğidir. Örneğin, C(x,y) önermesel fonksiyonunu düşünelim: x, y'nin başkentidir. Burada x bir şehir ismi fakat y bir ülke ismidir. O halde, C(a,b)'yi sağlayan (a,b) sıralı ikililerinin kümesi, A={şehirler} B={ülkeler} olmak üzere A x B kartezyen çarpımının alt kümesidir.

Aşağıdaki bağıntı tanımı şaşırtıcı şekilde basit ve geneldir. Bazıları buna ikili bağıntı da der çünkü iki objeyi ilişkilendirir.

Tanım: A ve B iki küme olsun. A'dan B 'ye bir bağıntı A x B kartezyen çarpımının bir alt kümesidir.

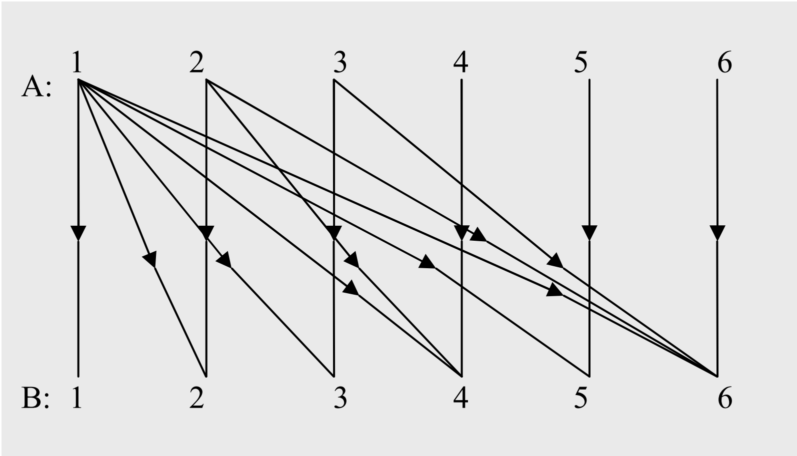
Tanıma bakıldığında ilk göze çarpan bağıntının bir küme olduğudur(sıralı ikililerden oluşan bir küme). $\boldsymbol{R}$, A'dan B'ye bir bağıntı ise eğer $(a,b) \in \boldsymbol{R}$ ise $a \in A$, $b \in B$ ile ilişkilidir deriz. Bu sebeple, $\boldsymbol{R}$ bağıntısının kendisi basitçe tüm ilişkili eleman çiftlerinin kümesidir. Genellikle kullanılan notasyon 'a, b ile ilişkilidir' için a $\boldsymbol{R}$ b' dir.

Bağıntıları görsel olarak ifade etmenin çeşitli yolları vardır, özellikle sonlu kümeler arasındaki bağıntıları. Şekil 3.1'de $\boldsymbol{R}$'nin elemanları A x B kartezyen çarpımının koordinat çizelgesi diyagramı üzerinde işaretlenmiştir. Bu tip diyagramlar $\boldsymbol{R}$'nin A x B'nin alt kümesi olduğunu açıkça gösterir fakat bağıntının diğer özelliklerini göstermede iyi değillerdir.



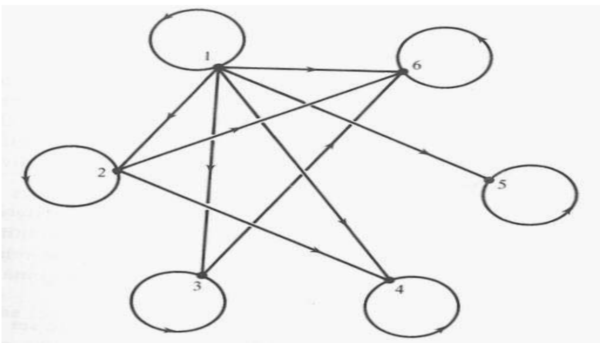
Şekil 3.1

Sonlu kümeler için başka bir alternatif A ve B’ nin elemanlarını üst üste yatay şekilde sıralamak ve a ***R*** b olduğunda $a \in A$’ dan $b \in B$ ‘ye bir ok çizmektir. Şekil 4.2’ de bu çeşit bir diyagram gösterilmiştir.



Şekil 3.2: Bağıntının diyagram olarak gösterilimi

Şekil 3.2’deki gösterim büyük kümeler için karmaşık hale gelebilir. Öte yandan bir küme üzerindeki bağıntılarda (yani A=B olanlarda), şekil daha basitleştirilebilir. A’daki elemanları iki kere listelemek yerine bu elemanları düzlemde birer nokta olarak temsil edebiliriz. Aynı şekilde a’dan b’ye yönlü bir ok sadece ve sadece a ***R*** b olması durumunda çizilir. Sonuçta ortaya çıkan diyagrama (şekil 3.3) **digraph** denir.



$$\begin{bmatrix} 1 & 0 & . & 1 \\ 1 & 1 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & 0 \end{bmatrix}$$

Şekil 3.3: Bağıntının digraf ve matris gösterilimi

Diğer üçüncü bir bağıntı gösterim biçimi de ikili matristir. $A=\{a_1, a_{2,\ldots}a_n\}$ ve $B=\{b_1, b_{2,} \ldots b_m\}$ sonlu kümeler ve R, A’dan B’ye bağıntı olsun. R’nin ikili matrisi n satırlı ve m sütunlu 0 ve 1’lerden oluşan dizi şeklindedir.

3.2 Bağıntıların Özellikleri

Bağıntıların önemi, ek özellikleri sağlayan özel bir takım bağıntılar yüzündendir. Bu özel bağıntıların ikisi eşlik bağıntıları ve sıra bağıntılarıdır ki bunların ikisi de kümeler üzerindeki bağıntılardır.

Tanım: ***R***, A kümesi üzerinde bir bağıntı olsun. O halde ***R***;

(i) Tüm $a \in A$ için, sadece ve sadece a ***R*** a ise **yansıyandır.** (reflexive).

(ii) Tüm a,b $\in$ A için, sadece ve sadece a ***R*** b, b ***R*** a anlamına geliyorsa **simetriktir.**

(iii) Tüm a,b $\in$ A için sadece ve sadece a ***R*** b ve b ***R*** a, a=b anlamına geliyorsa **ters simetriktir.**

(iv) Tüm a,b,c $\in$ A için sadece ve sadece a ***R*** b ve b ***R*** c, a ***R*** c anlamına geliyorsa **geçişlidir**(transitive).

Örnek 3.1: A={a,b,c,d} ve ***R***={(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(b,d),(d,d)} olsun.

R bağıntısı yukarıdaki tanımdaki hiçbir özelliği sağlamaz.

R yansıyan değildir çünkü c ***R*** c değildir; bu nedenle tüm x $\in$ A için x ***R*** x doğru değildir.

R simetrik değildir zira örneğin a ***R*** c'dir fakat c ***R*** a değildir.

R ters simetrik değildir zira a ***R*** b ve b ***R*** a 'dır fakat a=b değildir.

R geçişli değildir çünkü a ***R*** b ve b ***R*** d 'dir fakat a ***R*** d değildir.

Verilen digraphlar veya ikili matrisler ile bağıntı özelliklerinin anlaşılması mümkündür. Eğer bağıntı yansıyan bağıntı ise ***R***' nin digraphının her noktasından kendisine bir yönlü ok vardır. İkili matrisinde ise diyagonal elemanların hepsi 1' dir.

Eğer ***R*** simetrik ise digraphtaki okların tamamı iki-yönlüdür. Ters simetrik ise okların hiçbiri iki yönlü değildir. Öte yandan geçişli bağıntıların digraphlarından veya ikili matrislerinden özellik tanımlamak zordur.

3.3 Kesişimler ve Bağıntıların Birleşimi

A ve B arasındaki ***R*** bağıntısı A x B kartezyen çarpımının alt kümesi olduğuna göre bağıntıların kesişim ve birleşimini tanımlayabiliriz.

R ve ***S***, A kümesinden B kümesine iki bağıntı olsun. Hem ***R*** $\cap$***S*** hem de ***R***$\cup$***S*** A x B 'nin alt kümesidir. O halde, A'dan B'ye iki bağıntının hem kesişimi hem de birleşimi de aynı zamanda A'dan B'ye bağıntılardır.

R ve ***S*** bağıntılarının farklı küme çiftleri arasında olması durumu ise biraz daha karışıktır. ***R***'nin A'dan B'ye; ***S***'nin ise C'den D'ye bağıntı olduğunu düşünelim. ***R*** ve ***S*** sıralı ikililerden oluşan kümeler olduğundan ***R***$\cap$***S*** ve ***R***$\cup$***S*** birer bağıntıdır fakat hangi kümelerin ***R***$\cap$***S*** ile hangilerinin ***R***$\cup$***S*** ile ilişkili olduğu çok açık değildir.

Doğal olarak burada, ***R*** ve ***S***, A'dan B'ye bağıntı olduğuna göre kesişim veya birleşimleri bu bağıntıların özelliklerini miras alır mı sorusu akla gelir. Bağıntıların dört özelliğine ***R*** ve ***S***'nin aynı A kümesi üzerinde bağıntılar olduğunu varsayarak bakalım:

Yansıma özelliğine bakarsak: Hem ***R*** hem de ***S*** yansıyan ise tüm a$\in$ A için (a,a)$\in$***R*** ve (a,a)$\in$***S*** olmalıdır. Bu nedenle, (a,a) tüm a$\in$ A için ***R***$\cap$***S*** ve ***R***$\cup$***S***'ye aittir, öyleyse ***R*** ve ***S*** 'nin kesişimi ve birleşimi de yansıyandır.

İkinci olarak ***R*** ve ***S***'nin simetrik olduğunu düşünelim. a,b$\in$ A öyle ki (a,b)$\in$ ***R***$\cap$***S*** olsun. O halde, a ***R*** b ve a ***S*** b' dir. ***R*** ve ***S*** simetrik olduğundan b ***R*** a ve b ***S*** a ' dır ve bunun anlamı da (b,a) $\in$ ***R***$\cap$***S***' dir. O halde ***R***$\cap$***S*** de simetriktir. Aynı durum ***R***$\cup$***S*** için de geçerlidir.

Anti-simetriklik durumu biraz daha karmaşıktır. ***R***$\cap$***S*** 'nin ters simetrik olduğu yukarıdaki argümanlar ile gösterilebilir fakat birleşim her zaman ters simetrik olmayabilir. Tersine örnekle bunu gösterebiliriz. A={a,b} ve ***R***={(a,b)} ve ***S***={(b,a)}olsun. ***R*** ve ***S*** bağıntıları ters simetrik olduğu açıktır. Fakat $\boldsymbol{R} \cup \boldsymbol{S}$={(a,b), (b,a)}ters simetrik değildir çünkü a b ile, b de a ile ilişkilidir fakat a ve b eşit değildir.

Geçişlilik durumu ise ters simetriye benzer. Geçişli iki bağıntının kesişimi de geçişlidir. Ancak birleşimi geçişli olmayabilir. Aşağıdaki teorem bu özellikleri özetlemektedir.

Teorem 3.1: ***R*** ve ***S*** aynı A kümesi üzerinde iki bağıntı olsun.

a. Hem ***R*** hem de ***S*** yansıyan ise $\boldsymbol{R} \cap \boldsymbol{S}$ ve $\boldsymbol{R} \cup \boldsymbol{S}$ de yansıyandır.

b. ***R*** ve ***S*** simetrik ise $\boldsymbol{R} \cap \boldsymbol{S}$ ve $\boldsymbol{R} \cup \boldsymbol{S}$ de simetriktir.

c. ***R*** ve ***S*** ters simetrik ise $\boldsymbol{R} \cap \boldsymbol{S}$ de ters simetriktir fakat $\boldsymbol{R} \cup \boldsymbol{S}$ ters simetrik olmayabilir.

d. ***R*** ve ***S*** geçişli ise $\boldsymbol{R} \cap \boldsymbol{S}$ ***d***e geçişlidir fakat $\boldsymbol{R} \cup \boldsymbol{S}$ geçişli olmayabilir.

3.4 Eşdeğerlik Bağıntısı ve Bölmelemeler

Yaşayan insanlar kümesi üzerinde x ***R*** y sadece ve sadece x, y ülkesinde yaşıyorsa şeklinde tanımlanan bir bağıntıyı düşünelim. Her insanın sadece bir ülkede yaşadığını varsayarsak bağıntı şu üç özelliği sağlar:

x, x ile aynı ülkede yaşamaktadır, o halde yansıyandır.

x, y ile aynı ülkede yaşıyorsa; y de x ile aynı ülkede yaşıyor demektir, o halde simetriktir.

x, y ile aynı ülkede; z de y ile aynı ülkede yaşıyorsa x, z ile aynı ülkede yaşıyor demektir, o halde ***R*** geçişlidir.

Tanım: A kümesi üzerindeki ***R*** bağıntısı yansıyan, simetrik ve geçişli ise bu bağıntı **eşdeğerlik bağıntısı**dır.

Örnek 3.2: A=IR(reel sayılar kümesi) olsun ve A üzerinde

x ***R*** y sadece ve sadece $x^2=y^2$ ise

şeklinde bir bağıntı tanımlayalım. O halde;

R yansıyandır zira tüm x reel sayıları için $x^2= x^2$ ‘dir.

R simetriktir zira $x^2=y^2$, $y^2=x^2$ anlamına gelir.

R geçişlidir çünkü $x^2=y^2$ ve $y^2=z^2$ ise $x^2=z^2$ ‘dir.

O halde ***R*** eşdeğerlik bağıntısıdır.

Tanım: ***R***, A kümesi üzerinde bir eşdeğerlik bağıntısı ve $x \in A$ olsun. x’ in **eşdeğerlik sınıfı** [x] ile gösterilir ve A üzerinde x ile ilişkili tüm elemanların kümesidir öyle ki [x]={$y \in A$: x ***R*** y}.

Eğer iki eleman ilişkili ise eşdeğerlik sınıfları eşittir. Bunu göstermek için diyelim ki ***R***, A üzerinde bir eşdeğerlik bağıntısı ve A’ nın x ve y elemanları için x ***R*** y olsun. [x]=[y] olduğunu göstermek istiyoruz. $z \in [x]$ dersek x ***R*** z olur. x ***R*** y ve ***R*** simetrik ise aynı zamanda y ***R*** x olduğunu biliyoruz. O halde, y ***R*** x ve x ***R*** z ise geçişlilik özelliğinden y ***R*** z yani $z \in [y]$’ dir. Bu

[x] $\subseteq$[y] olduğunu gösterir. [y] $\subseteq$[x] 'in ispatı da benzer şekilde yapılabilir. Öyleyse, [x]=[y] sonucuna varılabilir.

Teorem 3.2: ***R***, bir A kümesi üzerinde bir eşdeğerlik bağıntısı ve x,y $\in$A olsun. Bu durumda sadece ve sadece x ***R*** y ise [x]=[y]' dir.

Bir küme üzerindeki eşdeğerlik bağıntısının eşdeğerlik sınıfları topluluğu, o kümenin bir bölmelemesini oluşturur.

Teorem 3.3: ***R***, boş olmayan bir A kümesi üzerinde bir eşdeğerlik bağıntısı olsun. Birbirinden farklı R-eşdeğerlik sınıfları topluluğuna A'nın bölmelemesi denir.

Örnek 3.3: ***R***, reel sayılar üzerinde; tamsayı(x) x'ten küçük veya eşit en büyük tamsayı olmak üzere x ***R*** y sadece ve sadece tamsayı(x)=tamsayı(y) ise şeklinde tanımlanmış bir bağıntı olsun.

R' nin reel sayılar kümesi üzerinde bir eşdeğerlik bağıntısı olduğunu kontrol etmek oldukça basittir. Örneğin; ½ $\in$ R: tamsayı(½)=0, öyleyse eşdeğerlik sınıfı

$$[½] = \{x\in \text{R: tamsayı}(x)=0\}$$

$$= \{x\in \text{R}: 0\leq x<1\}.$$

Bu kümeye yarı-açık aralık denir ve [0,1) şeklinde ifade edilir.

Bir küme üzerinde verilen eşdeğerlik bağıntısından eşdeğerlik sınıfları ile bölmeleme tanımlayabileceğimiz gibi bir küme üzerinde verilen bir bölmelemeden de eşdeğerlik sınıfları bölmelemeyi oluşturan orijinal alt kümeler olacak şeklinde bir eşdeğerlik bağıntısı tanımlayabiliriz.

Teorem 3.4: $\{S_i: i\in I\}$ bir A kümesinin bölmelemesi olsun. O halde, $i\in I$ için, x ***R*** y sadece ve sadece $x,y\in S_i$ ise eşdeğerlik sınıfları bölmelemedeki S_i kümeleri olan A üzerinde bir eşdeğerlik bağıntısı tanımlar.

Modulo Aritmetik : n pozitif bir tamsayı olarak verilsin. **Z** tamsayılar kümesi üzerinde ,modulo n bağıntısı aşağıdaki şekilde tanımlanır.

$a\equiv_n b$ ancak ve ancak eğer bazı $k\in\mathbf{Z}$ için a-b = k.n ise

$a\equiv_n b$ için alternatif tanım $a\equiv b$ mod n şeklindedir.

Örnek : mod5 'de n=5 dir. $a\equiv_5 b$ yi kısa olsun diye $a\equiv b$ şeklinde yazarız. Bu durumda ancak ve ancak a-b =5k ise $a\equiv_5 b$ dir. k gibi bir tamsayı vardır öyleki, a=5k+b dir. Bu yüzden

[p] =$\{q\in\mathbf{Z}$ **:** q=5k +p, bazı $k\in$Z için$\}$ Eşdeğerlik sınıfları sonsuzdur bazıları:

[0] ={….., -10,-5,0,5,10,15,…..}

[1] ={….., -9,-4,1,6,11,16,…..}

[2] ={….., -8,-3,2,7,12,17,…..}

[3] ={….., -7,-2,3,8,13,18,…..}

[4] ={….., -6,-1,4,7,12,19,…..} Bunlar beş adet farklı eşdeğerlik sınıfıdır.

3.5 Sıra Bağıntıları

Birçok küme doğal olarak sıralanmış elemanlara sahiptir. Örneğin büyüklüğe göre sıralanmış reel sayılar kümesi. Benzer şekilde bir küme topluluğu eleman sayısına göre sıralanabilir. Örneğin, $A \subseteq B$ ise A, B' den küçüktür deriz.

Eşdeğerlik bağıntılarından farklı olarak birçok farklı tip sıra bağıntısı vardır. En genel sıra bağıntısı 'parçalı sıra' bağıntısıdır.

Tanım: Bir kümedeki **parçalı sıra**, yansıyan, ters simetrik ve geçişli olan bir bağıntıdır.

Bir kümede parçalı sıra varsa bu kümeye **parçalı sıralı küme** denir.

Örnek 3.4: Reel sayılar kümesi üzerinde x ***R*** y sadece ve sadece $x \leq y$ ise şeklinde tanımlanan ***R*** bağıntısı parçalı sıradır.

Öte yandan, x ***S*** y sadece ve sadece x<y ise şeklinde tanımlanan ***S*** bağıntısı parçalı sıra değildir çünkü yansıyan değildir.

Teorem 3.5: ***R***, A kümesi üzerinde parçalı bir sıra ve B de A'nın herhangi bir alt kümesi olsun. Bu durumda, $\boldsymbol{S}=\boldsymbol{R} \cap (B \times B)$ B üzerinde bir parçalı sıradır.

Topyekûn sıra: (Doğrusal sıra) : Kümenin her hangi iki elemanı arasında sıralama yapılabilirse topyekûn sıra bağıntısı vardır. (Doğal sayılarda büyüklük, küçüklük bağıntısı)

Sözlük sırası : S ve T topyekûn sıralı kümeler ise SXT (kartezyen çarpım) kümesinde sözlük sırası:

a, $a' \in$ S; b, $b' \in$ T olmak üzere;

$(a,b) < (a',b') \Rightarrow a < a'$ yada $a=a'$, $b<b'$ dür.

Örnek: A= (1,2,3,4,6,8,12) kümesinde bölünebilirlik bağıntısıyla kısmi bir sıralama yapılırsa, bağıntı matrisi Tablo 1.4'deki şekilde olacaktır.

	1	2	3	4	6	8	12
1	1	1	1	1	1	1	1
2	0	1	0	1	1	1	1
3	0	0	1	0	1	0	1
4	0	0	0	1	0	1	1
6	0	0	0	0	1	0	1
8	0	0	0	0	0	1	0
12	0	0	0	0	0	0	1

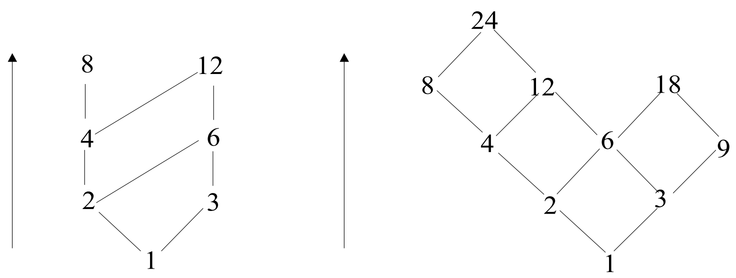
Tablo 3.1.

Halef-Selef(Predecessor-Successor, ilk öndegelen- ilk izleyen) Bağıntısı:

b, a'nın halefi ise a<c<b olamaz. Yani a ile b arasında sırlanabilen bir c elemanı bulmak mümkün değildir, yani a << b'dir.

Bu durumda kısmi sıralı küme için yeni bir graf tanımı(hasse diyagramı) yapılarak çizilir.

Hasse Diyagramı: a<<b şeklindeki çiftleri birleştiren ve en önde gelenin en alta konulduğu graftır.Örnek: Şekil 3.4..



Şekil 3.4.

3.5.1 En büyük ve en küçük eleman

Teorem 3.5' e göre reel sayıların her hangi bir alt kümesi $\leq$ bağıntısı ile parçalı sıralıdır. Bu şekilde sıralanmış bazı reel sayı kümeleri en büyük veya en küçük elemana sahip olabilir, bazıları da olmayabilir. Örneğin, tam sayılar kümesinin en büyük veya en küçük elemanı yokken, pozitif tamsayıların en küçük elemanı 1' dir fakat en büyük elemanı yoktur.

En büyük veya en küçük eleman bir tane olmayabilir. Örneğin, {a,b,c} kümesinin öz alt kümelerini eleman sayısına göre sıralarsak, en küçük eleman $\varnothing$ iken en büyük eleman üç tanedir çünkü üç tane iki elemanlı alt küme vardır.

Tanım: ***R***, A kümesi üzerinde bir parçalı sıra olsun. A' nın **en büyük elemanı,** tüm $a \in A$ için a ***R*** α olmak üzere α elemanıdır.

Benzer şekilde, A' nın **en küçük elemanı,** tüm $a \in A$ için β ***R*** a olmak üzere β elemanıdır.

Yeniden {a,b,c}' nin öz alt kümeleri örneğine dönersek iki elemanlı her bir alt küme en büyük eleman olacaktır. O halde bu düşünceyi maksimal eleman tınımı ile formülize edebiliriz.

Tanım: A, ***R*** sıra bağıntılı bir parçalı sıralı küme olsun. Tüm $a \in A$ için x ***R*** a x=a anlamına geliyorsa A' daki x elemanı **maksimal**dir.

Benzer şekilde, tüm $a \in A$ için a ***R*** y a=y anlamına geliyorsa y elemanı **minimal**dir.

3.6 n-ögeli(n-tuple) bağıntılar ve uygulamaları

3.6.1 n-ögeli(n-tuple) bağıntılar

Birden fazla kümeler arasındaki bağıntılar sık sık karşımıza çıkar. Örneğin, öğrenci adı, öğrencini bölümü, öğrencinin başarı notunu içeren kümeler arasında bir bağıntı vardır. Bu bölümde birden fazla kümeler arasında olan ve n-ögeli(n-tuple) bağıntılar olarak adlandırılan bağıntılar açıklanacaktır.

Tanım: A_1, A_2, ,……, A_n: kümesi verilsin. Bu kümeler üzerindeki bir n-ögeli bağıntı, $A_1XA_2X……XA_n$ kartezyen çarpımının bir alt kümesidir. A_1, A_2, ,……, A_n kümelerine bağıntının alanı(domain) ve n'e derecesi denir.

Örnek: 5 ögeli(B, N, A, D,N) bir R bağıntısı B: Bölümü, N :Numarası, A: Öğrencini adı, D : Dersin adı, N: Notu 'nu göstermek üzere veriliyor. Örneğin, Bilgisayar mühendisliği bölümü 01104115 numaralı Ali nin Ayrık Matematik dersi notu BA'nın anlamı (Bilgisayar,01104115, Ahmet, Ayrık matematik,BA) R bağıntısına aittir. Bu durumda derecesi 5 olan R bağıntısının alanı olan kümeler, Bölümler,Öğrenci numaraları, Öğrenci adları, Dersler ve Notlar şeklinde olacaktır.

Uygulama: İlişkisel Veritabanları

Bilgiyi saklamak ve işlemek için tasarlanmış bilgisayar sistemine veritabanı sistemi denir. Saklanmış verilerin işlenmesinin kontrol eden yazılıma da veritabanı yönetim sistemi (database management system) veya DBMS denir.

Tüm veritabanı yönetim sistemleri, verinin özel bir tip yapıya sahip olduğunu ve DBMS' in saklı veriyi, verinin kendi teorik modeline göre işlediğini varsayar. Bu yüzden birçok değişik tip DBMS bulunur: ilişkisel, ağ ve hiyerarşik. Bu bölümde matematiksel bağıntıları esas alan ilişkisel veritabanı sistemlerinden bahsedilecektir.

Bir veri birçok kısımdan oluşur. Örneğin, adres defterindeki bir kayıt isime, adrese, telefon numarasına göre sınıflandırılabilir. Verinin her bir parçası 'attribute (nitelik)' olarak adlandırılır. Verilerin her zaman belli bir nitelik kümesine sahip olduğunu varsayarız ve bu nitelik kümesine kayıt tipi(record type) adı verilir. Bir kayıt dosyası (record file), verilen kayıt tipine ait verilerin toplamıdır. Tüm verilerin aynı tip olduğu kayıt dosyalarına **birincil normal form**dadır **(first normal form**) denir. İlişkisel veritabanlarının temel kuralı tüm kayıt dosyalarının birincil normal formda olmasıdır.

Tanım: Veri **attribute** adı denilen bileşenlerine ayrılır. Bir **kayıt tipi** bir attribute'lar(veya **field**lar) kümesidir. Bir **kayıt örneği** (record instance), belli bir kayıt tipinin gerçek verisidir ve **kayıt dosyası** aynı kayıt tipinden olan kayıt örneklerinin kümesidir.

Örnek 3.5: GYTE isimli bir yardım derneği kendisine yapılan bağışları yapan kişileri, isimlerini, adreslerini, telefon numaralarını ve bağışla ilgili diğer detayların bilgilerini tutmak istediğini varsayalım.

Öncelikle bu dernek; bağışlayanın_adı, bağışlayanın_adresi, bağışlayanın_telefonu, bağış_miktarı ve bağış_tarihi şeklinde adlandırabileceğimiz attribute' ları belirler. Bu beş attribute kayıt tipini tanımlar. Tablo 3.2' de bazı kayıt örnekleri gösterilmiştir.

bağışlayanı_adı	***bağışlayanın_adresi***	***bağışlayanın_telefonu***	***bağış_miktarı***	***bağış_tarihi***
Kaya, R	Çayırova, Kocaeli	262 614-3939	100	Ocak 1997
Kaya, R	Çayırova, Kocaeli	262 614-3939	150	Mart 1999
Beyaz, S	Gebze, Kocaeli	262 578-4108	300	Ekim 1998
Verir,S	Pendik,İstanbul	216 467-1297	250	Kasım 2000
Verir, S	Pendik,İstanbul	216 467-1297	500	Aralık 1999

Tablo 3.2

Bağış yapanın açık adresi sadece bir attribute ile etiketlendiğinden bu kayıt dosyasından coğrafik bilgiyi elde etmek kolay olmayabilir. Örneğin dernek, İstanbul'dan bağış yapanları bulmak isterse şehir adı tek başına bir attribute olarak istenmediğinden çok zor olacaktır. bağışlayanın_adresi isimli tek bir attribute cadde ve şehir olarak ikiye ayrılsaydı şirketin işi çok daha kolay olurdu.

Bu örnek, attribute tanımlamak için önemli bir noktayı göstermiştir. Bir kayıt örneğindeki potansiyel yararlı bilgi parçalarının her biri bir attribute ile belirtilmelidir. Bu mecburi bir kural değildir zira 'potansiyel yararlı bilgi parçası' verinin kullanıldığı yere göre değişir.

Yukarıdaki örnekte eğer coğrafik konumun bir önemi yoksa adresleri tek bir attribute olarak belirtmek daha mantıklıdır.

İlişkisel veritabanı modelinde kayıt dosyası bir tablo olarak gösterilir. Tablonun sütunları attribute isimlerini, satırları ise her bir kayıt örneğini oluşturur.

Bir kayıt tipinin A_1, A_2,…,A_n şeklinde n tane attribute' ten oluştuğunu düşünelim. Bu durumda herhangi bir A_i attribute' u için bir veri girişleri kümesi olacaktır. X_i' ye de A_i attribute' u ile elde edilen değerler kümesi diyelim. X_i kümeleri zamana bağımlıdır ve kayıt dosyasına yeni girişler oldukça veya kayıt silindikçe değişir.

Bu notasyona göre; verilen bir kayıt örneği her bir x_i, Xi kümesine ait olmak üzere n-tuple' dır(x_1, x_2,…x_n). Bunun anlamı tüm kayıt örnekleri n tane aynı tip bilgi parçasından oluşur. $x_i \in X_i$ olmak üzere tüm n-tupple' ların (x_1, x_2,…x_n) kümesi X_1 x X_2 x…x X_n kartezyen çarpımıdır. Bu yüzden R kayıt dosyası kartezyen çarpımın alt kümesidir ($R \subseteq (X_1 \text{ x } X_2 \text{ x} \ldots \text{x } X_n)$).

Örnek 3.6: A_1…A_5 sırasıyla bağışlayanın_adı, bağışlayanın_adresi, bağışlayanın_telefonu, bağış_miktarı ve bağış_tarihi olsun. Her bir A_i attribute' u için bu attribute'a karşılık gelen X_i kümesi olduğunu varsayarız. O halde, bir kayıt örneği $x_i \in X_i$ olmak üzere 5 ögeli (5-tuple)' dır.

Bu kayıt tipine göre önemli sayıda bilgi yinelemesi olur. Örneğin, bağış yapanın ismi, adresi ve telefonu her bağış yaptığında tekrar kaydedilir. Bu bilginin tutulduğu yerden kayıplara yol açacağı gibi kayıt dosyasının güncellenmesini de zorlaştırır. Mesela, iki bağış yapmış Bay Kaya adres değiştirdi diyelim. Bu durumda, kayıt dosyasını güncelleştirmek için iki kayıtta da adresi değiştirmek gerekecektir.

Bu sebeplerle veriyi aşağıdaki gibi iki ayrı kayıt dosyasına bölmek daha mantıklıdır.

A_1, A_2, A_3: bağışlayanın_adı, bağışlayanın_adresi, bağışlayanın_telefonu

A_1, A_4, A_5: bağışlayanın_adı, bağış_miktarı, bağış_tarihi

Bu durumda orijinal veritabanındaki yineleme probleminden kurtulmuş oluruz ve daha kolay güncelleme yapabiliriz. Mevcut durumda veritabanı iki ilişkili kayıt dosyası içerir; birisi X_1 x X_2 x X_3 'ün, diğeri X_1 x X_4 x X_5 'ün alt kümesidir. Tabii ki, iki kayıt dosyasını bağışlayanın_adı attribute' u bağlar.

Tablo 3.3 ve tablo 3.4, tablo 3.2' deki bilginin nasıl iki kayıt dosyasına ayrıldığını göstermektedir.

bağışlayanın_adı	***bağışlayanın_adresi***	***bağışlayanın_telefonu***
Kaya, R	Çayırova, Kocaeli	262 614-3939
Verir,S	Pendik,İstanbul	216 467-1297
Beyaz, S	Gebze, Kocaeli	262 578-4108

Tablo 3.3

bağışlayanın_adı	***bağış_miktarı***	***bağış_tarihi***
Kaya, R	100	Ocak 1997
Kaya, R	150	Mart 1999
Beyaz, S	300	Ekim 1998
Verir,S	250	Kasım 2000
Verir, S	500	Aralık 1999

Tablo 3.4

Tanım: A_1, A_2,…, A_n attribute' ler topluluğu olsun ve her bir A_i' ye ilişkin bir X_i veri kümesi olduğunu düşünelim. **İlişkisel veritabanı** her biri bazı X_i kümeleri arasındaki bağıntılar topluluğudur. Her bir bağıntı bir **kayıt dosyası**dır.

Kayıt dosyasındaki kayıt örneklerine anahtar (key) ile erişilir. Key, tek bir kayıt örneğini belirten attribute' lar kümesidir, fakat bu kümenin hiçbir öz alt kümesi tek bir kayıt örneğini belirtme özelliğine sahip değildir.

Pratikte birçok olası anahtar seçme imkânı vardır. Key olarak kullanılabilecek attribute'lar kümesine **candidate key** denir. Bunlardan biri gerçek key olarak seçilir ve buna **primary (birincil) key** denir.

Örneğimizde, {bağışlayanın_adı} herhangi iki bağış yapanın adının aynı olmaması durumunda Tablo 1 için bir candidate keydir. Bu durumda her bir kayıt örneği bağış yapanın adı ile belirtilebilir. Öte yandan iki farklı bağış yapan kişinin aynı adı taşıması durumunda {bağışlayanın_adı} key olmaz bunun yerine {bağışlayanın_adı, bağışlayanın_telefonu} attribute kümesi key olarak kullanılabilir.

İlişkisel veritabanları üzerinde beş çeşit işlem yapılabilir.

3.6.2 Selection (Seçme)

Selection işlemi kayıt dosyasından verilen kriter kümesini sağlayan kayıt örneklerini listeler. Örneğin, X şehrinde yaşayan müşterilerin tüm isim ve adres kayıtlarını listelemek bir selection örneğidir.

Selection işlemini yeni kayıt dosyaları tanımlamak yani veri tabanındaki kayıt dosyalarının alt kümeleri şeklinde düşünebiliriz. Bu yeni kayıt dosyaları muhtemelen geçicidir ve veritabanını oluşturan kayıt dosyaları kümesine eklenmezler. Aynı zamanda selection kayıt dosyasının tablo gösterimi şeklinde de tanımlanabilir. Bu yeni kayıt dosyaları gerekli attribute'lara sahip satırları çekerek elde edilir.

Örneğin; GYTE veritabanında 'Ocak 1999'dan sonraki tüm bağışları seçmek' istediğimizde tablo 3.3 'te gösterilen kayıt dosyasından ikinci, dördüncü ve beşinci satırlar elde edilecektir.

3.6.3 İzdüşüm (Projection)

Selection tablodaki belli satırları geri döndürürken projection işlemi sütunları döndürür. Sütunlar attribute' lara karşılık geldiğinden sonuçta ortaya çıkan kayıt dosyası orijinalden daha az sayıda attribute'lu kayıt tipine sahiptir.

Projection işleminin resmi tanımı şöyledir: R, (A_1, …, A_p) tipinde bir kayıt dosyası ve $q \leq p$ ve her bir B_i aynı zamanda R' nin attribute' u olmak üzere (B_1,…, B_q) kayıt tipi olsun. Yani, her bir B bir j için A_j' ye eşit olsun. Projection, kayıt örnekleri R' nin her bir kayıt örneklerinin B_i attribute' larından oluşan (B_1, …, B_q) tipinde yeni kayıt dosyası tanımlar.

3.6.4 Doğal Birleşim (Natural Join)

GYTE veritabanının örnek 3.6 'daki gibi ikiye ayrıldığını düşünelim. Bu durumda bağış yapanların isimlerini, telefon numaralarının ve bağış miktarlarını nasıl alabiliriz? Buradaki problem bağış yapanın telefon numarası ile bağış miktarlarının farklı kayıt dosyalarında olmalarıdır. O halde kayıt dosyalarını birleştirerek üç attribute 'u da içeren yeni bir kayıt dosyası üretmemiz gerekir. İki dosyada ayrıca bağışlayanın_adresi ve bağış_tarihi de bulunur ve sonuçta oluşacak birleşmiş tabloda bu attributeler de bulunacaktır. Ancak bu bir sorun değildir zira projection ile bu dosyadan gerekli kayıt tipleri çekilebilir.

Natural join işleminin matematiksel temeli şöyledir: R ve S, ($A_1,\ldots, A_p, B_1,\ldots, B_q$) ve ($A_1,\ldots, A_p, C_1,\ldots, C_r$) tipinde kayıt dosyaları olsun. R ve S' nin doğal birleşimi ($A_1,\ldots, A_p, B_1,\ldots, B_q, C_1,\ldots, C_r$) tipinde yeni bir kayıt dosyasıdır. Doğal birleşimim oluşturan kayıt örneklerinin hepsi ($x_1,\ldots, x_p, y_1,\ldots y_q$) $\in$ R ve ($x_1,\ldots, x_p, z_1,\ldots z_r$) $\in$S özelliğine sahip (p+q+r)-tuple ($x_1,\ldots, x_p, y_1,\ldots y_q, z_1,\ldots,z_r$) 'dır.

3.6.5 Birleşim ve Fark (Union and Difference)

Verilen iki aynı kayıt tipinde R ve S kayıt dosyasının birleşimi ve farkı, bildiğimiz küme teorisindeki birleşim ve fark işlemlerine karşılık gelir. Bu yüzden $R\cup S$, ve R ve S 'deki kayıt örneklerinin tamamını (listeyi tekrarlamadan) içeren kayıt dosyasıdır. R-S ise R de bulunan fakat S' de bulunmayan kayıt örneklerini içeren kayıt dosyasıdır.

3.7 Fonksiyonlar ve Tanımları

x^2+5x-8, $1/(x+3)^3$, cos(x), log(x) vs. gibi ifadeler genellikle ***f***(x) ile gösterilir ve "x' in fonksiyonu" olarak adlandırılır. İfadenin kendisinden daha önemli olan verilen herhangi bir x değeri için fonksiyonun değerini hesaplamak için bir kural tanımlamasıdır. İki farklı ifade ***f***(x) ve g(x), tüm x reel sayıları için aynı değerleri verebilir ve biz bu iki ifadenin aynı fonksiyonu tanımladığını söyleyebiliriz. Örneğin $f(x)=x^2+4x-5$ ve $g(x)=(x+2)^2-9$.

Tanım: A ve B iki küme olsun. A' dan B' ye bir ***f*** fonksiyonu; ***f***: A→B şeklinde yazılır ve her bir $a\in A$ 'yı tek bir $f(a)\in B$ elemanı ile eşleştiren bir kuraldır.

Örnek 3.7: x^2+4x-5 ifadesi tek başına bir fonksiyon değildir zira tanımıza göre A ve B kümeleri belirtilmemiştir. Öte yandan, bu ifade şu şekilde tanımlanabilir: ***f***: R → R olmak üzere $f(x)=x^2+4x-5$.

Tanım: A ve B küme olsun. ***f***, A' dan B' ye bir fonksiyon ***f***: A→B şeklinde yazılır ve $f\subseteq(AxB)$ 'nin alt kümesidir ve şu kuralı sağlar:

Her bir $a\in A$ için $(a,b) \in f$ olmak üzere tek bir $b\in B$ vardır.

A kümesi ***f***' nin tanım kümesi ve B kümesi de ***f***' nin değer kümesi denir. $(a,b) \in f$ ise $b\in B$ elemanı $a\in A$ elemanının görüntüsüdür denir ve $b=f(a)$ veya ***f***: a→b şeklinde yazılır.

Tanım: ***f***: A→B ve g: A'→B' fonksiyonları

(i) A=A'

(ii) B=B'

(iii) $f(a)=g(a)$ (A=A' 'ne ait tüm a elemanları için)

ise eşittir.

Bir fonksiyonun grafiği R^2= R x R düzleminde $y=f(x)$' i sağlayan (x,y) noktalarını içeren eğridir. Ancak unutulmaması gereken nokta x-y düzlemindeki her eğri her hangi bir ***f***: A→R ($A\subseteq R$) fonksiyonun grafiği değildir. Örneğin, merkezi orijin (0,0), yarıçapı 1 olan çemberin denklemi

$x^2+y^2=1$' dir. -1 ve 1 arasındaki her bir x değeri için iki tane y değeri vardır.

(x,y)-düzleminde verilen bir eğrinin bir fonksiyonun grafiği olup olmadığını anlamak kolaydır. x=a dikey doğrusu sadece ve sadece eğriyi tek bir yerde kesiyorsa, verilen $a\in A$ için $y=f(a)$ olacak şekilde tek bir $y\in$ R vardır.

Bir fonksiyonun tanımında karışıklığa sebep olan iki özellik vardır. Birincisi, tanım kümesinin iki veya daha fazla elemanı değer kümesinde aynı görüntüye sahipse. İkincisi ise, değer kümesindeki tüm elemanların, tanım kümesindeki bir elemanın görüntüsü olmak zorunda olmadığıdır.

Tanım: ***f***: A→B bir fonksiyon olsun. ***f*' nin görüntüsü** (aralığı)

$$\mathrm{im}(f)=\{b\in B: (a,b)\in f,\ a\in A \text{ için}\} \text{ kümesidir.}$$

Dikkat edilirse im(***f***) değer kümesi B' nin alt kümesidir ve $a\in A$ elemanının görüntüsü ***f***(a) ile karıştırılmamalıdır. Bir elemanın görüntüsü bir elemandır fakat bir fonksiyonun görüntüsü bir kümedir ; bir başka deyişle tanım kümesindeki elemanların görüntülerinin tamamını içeren kümedir.

$$\mathrm{im}(f)=\{f(a): a\in A\}.$$

Örnek 3.8: ***f***: R→R olmak üzere $f(x)=\dfrac{3x}{x^2+1}$ fonksiyonunun görüntüsünü bulunuz.

Çözüm: Tanıma göre $y\in \mathrm{im}(f)$ sadece ve sadece $x\in$ R için $y=\dfrac{3x}{x^2+1}$ ise.

Bu eşitlik şuna eşittir: $yx^2+y=3x$ veya

$$yx^2-3x+y=0.$$

Bu durumda $x=\dfrac{3\pm\sqrt{9-4y^2}}{2y}$ olur.

Bu nedenle gerçek çözüm $y\neq 0$ ve $9-4y^2\geq 0$ olmalıdır.

Böylece $y^2\leq 9/4$ yani $-3/2\leq y\leq 3/2$ (ve $y\neq 0$) elde edilir.

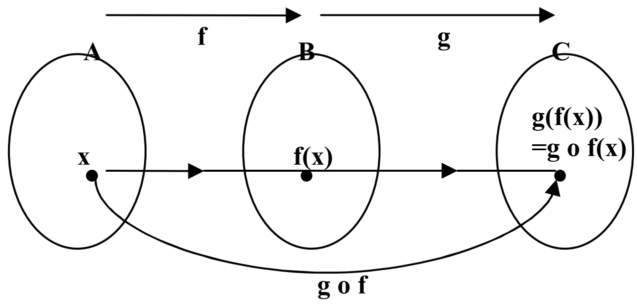
Bu durumda $-3/2\leq y\leq 3/2$, $y\neq 0$ sağlandığında $y=f(x)$ olacak şekilde bir x reel sayısı bulunabilir. y=0 özel bir durumdur fakat açıkça $f(0)=0$' dır o halde, $0\in \mathrm{im}(f)$ ' dir.

Böylece, $\mathrm{im}(f)=[-3/2, 3/2]=\{y\in \mathrm{R}: -3/2\leq y\leq 3/2\}$.

f: R →R gibi bir fonksiyonun grafiği verilmişse bu fonksiyonun görüntüsü kolayca bulunabilir. A' nın her bir elemanının görüntüsü ***f***(a); a' dan grafiği kesene kadar dikey doğru çizerek ve sonra kesişim noktasından da y-eksenine yatay bir doğru çizerek bulunabilir.

3.7.1 Bileşik Fonksiyonlar, Birebir(injective) ve Örten(Surjektive) fonksiyonlar

f: A→B ve ***g***: B→C iki fonksiyon olsun. x, A'nın elemanı ise y=***f***(x) B' ye aittir. Bu nedenle ***g***(y)=***g***(***f***(x)) C' nin elemanıdır. A 'dan C' ye bir fonksiyon tanımlamak için ***f*** ve ***g*** 'nin bileşkesi dediğimiz ve $g \circ f$ ile gösterdiğimiz $x \mapsto g(f(x))$ ortaklığını kullanabiliriz. $g \circ f$ bileşik fonksiyonu Şekil 1.15'deki gibi gösterilebilir.



Şekil 1.15. Bileşik fonksiyon

Tanıma göre $g \circ f$ fonksiyonu z= $g \circ f$(x) olacak şekilde tüm (x,z) elemanlarını içeren A x C kartezyen çarpımının alt kümesi olmalıdır. y=f(x)$\in$B dersek (x,y) $\in f$ ve (y,z) $\in g$ 'dir. Bu nedenle, bileşik fonksiyonları şu şekilde tanımlarız.

Tanım: f: A→B ve g: B→C iki fonksiyon olsun. **Bileşik fonksiyon** $g \circ f$: A→C:

$$g \circ f = \{(x,z) \in A \times C: (x,y) \in f \text{ ve } (y,z) \in g \ (y \in B \text{ için})\}$$

İki rastgele fonksiyonun bileşkesi $g \circ f$ olmayabilir. Yukarıdaki tanıma göre g' nin tanım kümesi, f' in değer kümesine eşittir. Ancak bu kati bir kural değildir. $g \circ f$ tanımını biraz genişletirsek:

f: A→B ve g: B'→C iki fonksiyon ve a$\in$A olsun. $g(f(a))$' nın tanımlı olabilmesi için f(a)'nın g' nin tanım kümesi olan B' kümesine ait olması gerekir. Bu durumda $g \circ f$ 'i tanımlamak için $g(f(a))$' nın tüm a$\in$A için tanımlı olması şarttır. Böylece $g \circ f$ sadece ve sadece f' in görüntüsü g' nin tanım kümesinin alt kümesi ise tanımlıdır. Tabii ki, bu şart yukarıdaki tanımda olduğu gibi B=B' ise sağlanır.

Örnek 3.9: f ve g R→R ve $f(x)=x+2$, $g=1/(x^2+1)$ şeklinde tanımlı olsun. Bu durumda,

$$\begin{aligned} g \circ f(x) &= g(f(x)) \\ &= g(x+2) \\ &= \frac{1}{(x+2)^2+1} \\ &= \frac{1}{x^2+4x+5} \end{aligned}$$

Benzer şekilde;

$$\begin{aligned} f \circ g(x) &= f(g(x)) \\ &= f(1/(x^2+1)) \\ &= \frac{1}{x^2+1}+2 \\ &= \frac{2x^2+3}{x^2+1} \end{aligned}$$

Bu örnek gösteriyor ki, genellikle $f \circ g \neq g \circ f$.

Teorem 3.6: $\boldsymbol{f}$: A→B ve $\boldsymbol{g}$: B→C iki fonksiyon olsun. Bu durumda im($\boldsymbol{g} \circ \boldsymbol{f}$) $\subseteq$ im($\boldsymbol{g}$).

İspat: $c \in$ im($\boldsymbol{g} \circ \boldsymbol{f}$) olsun. O halde, ($\boldsymbol{g} \circ \boldsymbol{f}$)(a)=$\boldsymbol{g}$($\boldsymbol{f}$(a))=c olacak şekilde $a \in A$ mevcuttur. Bu durumda, b=$\boldsymbol{f}$(a) $\in$ B dersek $\boldsymbol{g}$(b)=c' dir ve bu nedenle $c \in$ im($\boldsymbol{g}$)' dir. Bu sebeple, im($\boldsymbol{g} \circ \boldsymbol{f}$) $\subseteq$ im($\boldsymbol{g}$).

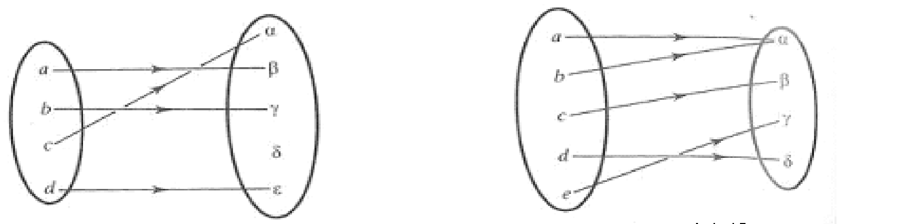
Önceki bölümlerden hatırlayacağımız gibi bir $\boldsymbol{f}$: A→B fonksiyonu

(i) tanım kümesinin farklı elemanları aynı görüntüye sahip olabilir.

(ii) değer kümesinin bazı elemanları tanım kümesinin herhangi bir elemanının görüntüsü olmayabilir.

Örneğin, $\boldsymbol{f}$: R→R, $\boldsymbol{f}$(x)=x^2 fonksiyonunda bu iki olasılık da mümkündür. Hem 2 hem de -2 aynı görüntüye sahip olduğu gibi herhangi bir negatif reel sayı $\boldsymbol{f}$' in görüntüsüne dahil değildir.(tüm x reel sayıları için $x^2 \geq 0$).

Yukarıdaki maddelerden ilkinin mümkün olmadığı fonksiyonlara birebir (injective), ikincisinin mümkün olmadığı fonksiyonlara da örten (surjective) denir. Bu iki durum 1.16' te gösterilmiştir. Şekil 1.14 (a) 'daki $\boldsymbol{f}$:{a,b,c,d}→{$\alpha, \beta, \gamma, \delta, \varepsilon$} fonksiyonu injective'dir fakat surjective değildir. Diğer yandan, şekil 1.14 (b) 'deki g:{a,b,c,d,e}→{$\alpha, \beta, \gamma, \delta$} fonksiyonu surjective'dir fakat injective değildir.



(a) injektif
Tanım kümesinin farklı elemanları farklı görüntüye sahip

(b) sürjektif
Değer kümesinin tüm elemanları tanım kümesinin bir elemanının görüntüsü

Şekil 1.16.

Tanım: $\boldsymbol{f}$: A→B bir fonksiyon olsun.

(i) Tüm a, a' $\in$ A elemanları için aşağıdaki durum sağlanıyorsa $\boldsymbol{f}$ birebir(injective)dir veya bir injeksiyon(birebir fonksiyon)dur deriz:

Eğer (a,b), (a',b') $\in \boldsymbol{f}$ ve a $\neq$ a' ise b $\neq$ b'.

(ii) Eğer her $b \in B$ için (a,b) $\in$ $\boldsymbol{f}$ olacak şekilde $a \in A$ mevcut ise $\boldsymbol{f}$ örten(surjective)dir veya bir örten fonksiyon(surjeksiyon)dur deriz.

Örnek 3.10: $\boldsymbol{f}$: R→R, $\boldsymbol{f}$(x)=3x-7 olsun. f' in hem birebir hem de örten fonksiyon olduğunu gösteriniz.

Çözüm: $\boldsymbol{f}$' in inbirebir olduğunu göstermek için tüm x ve y reel sayıları için $\boldsymbol{f}$(x)=$\boldsymbol{f}$(y) 'nin x=y anlamına geldiğini ispatlamamız gerekir.

$\boldsymbol{f}$(x) = $\boldsymbol{f}$(y)

3x-7 = 3y-7

3x = 3y

x = y. O halde f birebir dir.

$\boldsymbol{f}$' in örten olduğunu göstermek için, y 'nin R değer kümesinin herhangi bir elemanı olduğunu düşünelim. $\boldsymbol{f}$(x)=y olacak şekilde $x \in$ R bulmamız gerekir. x=(y+7)/3 olsun. O halde, $x \in$ R ve

$f(x) = f((y+7)/3)$

$= 3.\ \frac{y+7}{3} - 7$

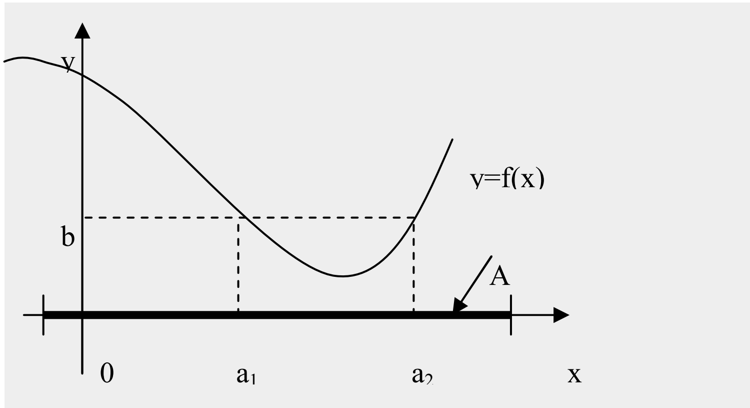
$= y+7-7$

$= y$. O halde f örtendir.

Bu ispat herhangi bir **doğrusal fonksiyon**un f: R→R, $f(x)=ax+b$ hem birebir hem de örten olduğunu göstermek için kullanılabilir.

A ve B R' nin alt kümeleri olmak üzere f: A→B fonksiyonu olsun. Bir fonksiyonun grafiğinden birebir veya örten olup olmadığını anlayabiliriz.

f ' in birebir olmadığını farz edelim. O halde, A' da $f(a_1)=f(a_2)=b$ olacak şekilde iki tane a_1 ve a_2 elemanı vardır. Bunun anlamı b 'den çizilen yatay doğru x-eksenini $x=a_1$ ve $x=a_2$ 'de keser. Bu durum şekil 1.17'de gösterilmiştir.



Şekil 1.17.

Öte yandan eğer f birebir ise bu durum hiçbir zaman gerçekleşmez. Yani yatay doğru grafiği birden fazla yerden kesmez.

Örten özelliği ise şu şekildedir: im(f)=B olmak üzere f, sadece ve sadece B' nin bir noktasından geçen her yatay doğru grafiği en az bir kere kesiyorsa örtendir.

Teorem 3.7: f: A→B ve g: B→C iki fonksiyon olsun.

(i) Eğer f ve g her ikisi birden birebir ise $g \circ f$ de birebir'dir.

(ii) Eğer f ve g her ikisi birden örten ise $g \circ f$ de örten'dir.

İspat: (i) f ve g' nin birebir fonk. olduğunu düşünelim. a, a'$\in$A, b=f(a) ve b'=f(a') olsun. Bu durumda,

$g \circ f(a) = gf(a')$

$\Rightarrow \quad g(f(a)) = g(f(a'))$

$\Rightarrow \quad g(b) = g(b')$

$\Rightarrow \quad b = b'$ (zira g birebirdir.)

$\Rightarrow \quad f(a) = f(a')$ (çünkü $f(a)=b$, $f(a')=b'$.)

$\Rightarrow$ $a = a'$ (zira f birebirdir.)

Böylece $g \circ f$ bir birebir fonk.dur.

Teorem 3.8: f: A→B ve g: B→C iki fonksiyon olsun.

(i) $g \circ f$ bileşik fonksiyonu birebir ise f de birebirdir.

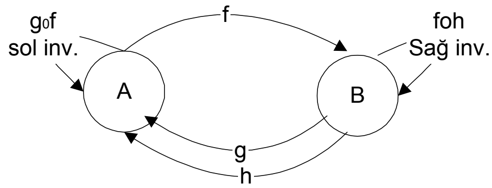
(ii) $g \circ f$ bileşik fonksiyonu örten ise g de örtendir.

Tanım: id_A: A→A fonksiyon, id_A:{(x,x):x∈A}, $id_A(x)=x$ x∈A dır.

Teorem 3.9: **(i)** f: A→B fonksiyonu sadece ve sadece $g \circ f = id_A$: A→A (A'nın özdeşlik fonksiyonu) olacak şekilde bir g: B→A varsa birebir'dir.

(ii) f: A→B fonksiyonu sadece ve sadece $f \circ h = id_B$: B→B (B'nın özdeşlik fonksiyonu) olacak şekilde bir h: B→A varsa birebir'dir.

Tanım: f: A→B herhangi bir fonksiyon olsun. $g \circ f = id_A$ olacak şekilde bir g: B→A fonksiyonu f için bir **sol inverse**, benzer şekilde $f \circ h = id_B$ olacak şekilde bir h: B→A fonksiyonu f için bir **sağ inverse** denir.



Şekil 3.8.

3.7.2 Ters Fonksiyonlar

Hem birebir hem de örten fonksiyonlar ilginç ve önemli özelliklere sahiptir.

Tanım: f: A→B fonksiyonu hem birebir hem de örten ise birebir ve örten(bijective)dir veya bijeksiyondur.

Teorem 3.10: A ve B R' nin alt kümeleri olmak üzere f: A→B fonksiyon olsun. O halde f, sadece ve sadece B' nin bir noktasından çizilen her doğru f' nin grafiğini tam olarak bir yerde kesiyorsa birebir ve örten'dir.

Teorem 3.11: **(i)** İki birebir ve örten fonksiyonun bileşkesi yine birebir ve örten fonk.dur.

(ii) f: A→B fonksiyonu sadece ve sadece hem sol hem de sağ inverse' e sahipse birebir ve örten fonk.dur.

(iii) A ve B sonlu kümeler olmak üzere f: A→B bijeksiyon ise $|A| = |B|$.

Dikkat edilirse (i) şıkkının tersi yanlıştır. Eğer bir bileşik fonksiyon $g \circ f$ birebir ve örten ise hem f hem de g birebir ve örten olmak zorunda değildir. Eğer A ve B aynı kardinaliteye sahip sonlu kümeler ise A' dan B' ye bir birebir ve örten fonk. vardır.

Şimdi şu soruya bir göz atalım. f: A→B şeklinde verilmiş bir fonksiyon olsun. g={(b,a):(a,b) ∈f} hangi durumlarda bir fonksiyon tanımlar? Burada g' yi f' in diyagramında okları tersine çevirmek gibi düşünebiliriz: Eğer b=f(a) ise a=g(b) 'dir.

Genel duruma bakacak olursak, f: A→B bir fonksiyon ve g={(b,a):(a,b)∈f} şeklinde tanımlanmış olsun. O halde g, her bir b∈B için (b,a) ∈g veya (a,b)∈f olacak şekilde tek bir a∈A varsa fonksiyondur. B' nin her bir elemanı için gerekli özellikleri sağlayan a' ların varlığı aynı zamanda f' in örten olması için aranan şartlardır. Bunun da ötesinde, (a,b)∈f olacak şekilde bir $a \in A$ elemanı sadece ve sadece ***f*** birebir ise tektir.

Teorem 3.12: ***f***:A→B bir fonksiyon olsun. ***g***={(b,a)B x A: (a,b) ∈***f***} bağıntısı sadece ve sadece ***f*** birebir ve örten ise B' den A' ya bir fonksiyondur.

Tanım: ***f***:A→B bir birebir ve örten fonk. ise ***g***: B→A, '***g***(b)=a sadece ve sadece ***f***(a)=b ise' şeklinde tanımlanan fonksiyona ***f***' in ters fonksiyonu denir ve f^{-1} şeklinde gösterilir.

Teorem 3.13: f:A→B bir bijeksiyon ise f^{-1}:B→A f için hem sol hem de sağ inverse' tir.

Örnek 3.11: ***f***: R -{1}→ R -{2}, ***f***(x)=2x/(x-1) fonksiyonunun birebir ve örten olduğunu gösterin ve tersini bulun.

Çözüm: Eğer f^{-1} 'i bulabilirsek ***f*** birebir ve örten olmalıdır. f^{-1}' i bulabilmek için tanımı kullanırız: y=***f***(x) ise x=f^{-1}(y). O halde,

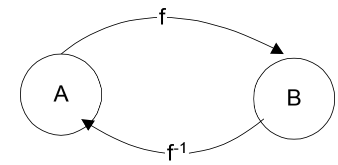
y=2x/(x-1)

y(x-1)=2x

yx-2x=y

x(y-2)=y

x=y/(y-2)



Şekil 3.9.

Bu sebeple şu fonksiyonu tanımlayabiliriz:

g: R -{2}→ R -{1}, ***g***(y)=y/(y-2).

3.8 Alıştırmalar

1- $A=Z^+ \times Z^+$ ve R, A üzerinde '(a,b)***R***(c,d) sadece ve sadece a+d=b+c ise' şeklinde tanımlanan bir bağıntı olsun. ***R*** bağıntısının yansıyan, simetrik ve geçişli olduğunu fakat ters simetrik olmadığını gösteriniz.

2- ***R***, A' dan B 'ye ve ***S***, B' den C 'ye birer bağıntı olsun. ***R*** ve ***S*** 'nin bileşkesi A' dan C 'ye $S \circ R$ bağıntısıdır ve 'a($S \circ R$)c sadece ve sadece a***R***b ve b***S***c olacak şekilde bir b∈B elemanı var ise' şeklinde tanımlanmıştır.

Bu tanıma göre, ***R***, Z^+ üzerinde tanımlı bir bağıntı olsun.

n ***R*** m sadece ve sadece $m=n^2$ olduğuna göre; Z^+ üzerinde $R^2= R \circ R$ bağıntısını tanımlayınız.

3- A={1,2,3,4} olsun.

a. A üzerinde kaç tane eşdeğerlik bağıntısı vardır?

b. A üzerinde (1,2)***R*** özelliğine sahip kaç tane ***R*** eşdeğerlik bağıntısı vardır?

4- Bir ***R*** bağıntısı R^2 üzerinde şu şekilde tanımlanmıştır.

$(x_1,y_1)R(x_2,y_2)$ sadece ve sadece $x_1<x_2$ veya hem $x_1=x_2$ hem de $y_1 y_2$ ise.

R 'nin R^2 üzerinde bir parçalı sıra olduğunu gösteriniz.

5- Aşağıdaki üç tabloda öğrenciler, dersler ve öğrencilerin derslerde aldıkları notlar ile ilgili bilgiler yer almaktadır.

student_id	first_name	last_name	Level	concentration
100	Lynn	Icks	Senior	Independent
101	Bernard	Mac	Junior	CS- Theatre Arts
102	Mike	Soft	Freshman	Business Econ
103	June	Icks	Junior	CS-AM
104	Alan	Turing	Grad	Math

Tablo 3.5- STUDENTS

Course_id	course_name	Professor	semester	year
1	CS22: Discrete Math.	Franco	Spring	2003
2	CS22: Discrete Math.	Herlihy	Spring	2002
3	CS123: Purty Pictures	Avd	Fall	2003
4	EC187: Game Theory	Dal Bo	Fall	2002
5	CS51: Turing's Factory	Savage	Fall	2003
6	MA ∞	Unknown	Fall	2001

Tablo 3.6 – COURSES

student_id	course_id	student_grade
100	2	A
101	1	S
101	2	NC
102	4	A
102	1	B
103	5	A
104	5	A
104	6	A

Tablo 3.7 – COURSE_GRADES

Buna göre;

a. A almış Freshman seviyesindeki öğrencileri seçiniz.

b. Courses tablosunda (course_name, semester) kayıt tipine göre projection işlemini gerçekleştirin.

c. Üç tablo üzerinde natural join işlemini gerçekleştirin ve (student_name,course_name,grade) kayıt tipine göre projection yazın.

6- Aşağıdaki fonksiyonların görüntülerini bulunuz.

a. ***f***:R→R, $x \mapsto (x+2)^2$

b. ***f***:R→R, $x \mapsto x^4$

7- ***gf*** bileşke fonksiyonunu tanımlayınız.

f:R→R, $f(x)= \begin{cases} x^2 + x, x \geq 0 \\ 1/x \ , \quad x < 0 \end{cases}$

g:R→R, $g(x)= \begin{cases} \sqrt{x+1}, x \geq 0 \\ 1/x \ , \quad x < 0 \end{cases}$

8- ***f***: A→B ve ***g***: B→C iki fonksiyon olmak üzere, ‘Hem ***f*** hem de ***g*** surjective ise $g \circ f$ bileşkesi de surjective’dir’ şeklindeki teoremi ispatlayınız.

9- ***f***(x)= fonksiyonunun tersini (***g***(x)) bulup $g \circ f$ ve $f \circ g$ bileşke fonksiyonlarını yazarak sonucun x olduğunu ispatlayınız.

10- ***f*** ve ***g*** $R \rightarrow R$ ve $k \in R$ olmak üzere birer fonksiyon olsun. ***f***+***g***, ***f******g*** ve k***f***: $R \rightarrow R$ fonksiyonları sırasıyla şu şekilde tanımlanmaktadır:

$(f+g)(x)=f(x)+g(x)$

$(f*g)(x)=f(x)*g(x)$

$(kf)(x)=k.f(x).$

(i) Eğer $k \neq 0$ ise kf ‘in sadece ve sadece f bijeksiyon ise bijeksiyon olduğunu ispatlayınız.

(ii) Ne ***f***+***g*** ’nin ne de ***f******g*** ‘nin bijeksiyon olmadığı ***f*** ve ***g*** bijeksiyonları tanımlayınız.

8 Graf Teorisi

8.1 Graflar ve Tanımlar

Graf teorisinin uygulamaları modern hayatın karmaşık ve geniş kapsamlı birçok probleminin çözümü için kullanılmaktadır. Bu uygulamalar; ekonomi, yönetim bilimi, satış pazarlama, bilgi iletimi, taşıma planlaması gibi alanları kapsamaktadır. Graf teorisi problemleri tanımlama ve yapısal olarak ilişkileri belirlemekte de faydalıdır.

Bir grafın ne olduğunu açıklamadan önce belki de ne olmadığını söylemek daha iyi olabilir. Bu bölümde kullanılan graf bir fonksiyonun grafiği değildir. O halde graf nedir? Basitçe bir graf düğüm olarak adlandırılan noktalar ve her biri bu noktaları veya sadece noktanın kendisini birleştiren ve ayrıt olarak adlandırılan çizgiler topluluğudur. Örnek olarak şehirleri düğüm(vertice) ve onları bağlayan yolları ayrıt(edge) olarak gösteren yol haritaları verilebilir.

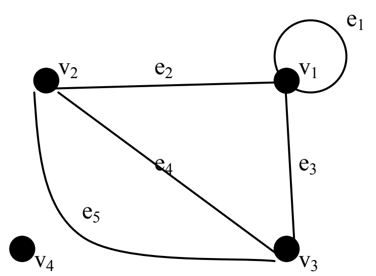
Bir grafı tanımlamak için öncelikle düğümlerin ve ayrıtların kümesini tanımlamak gerekir. Daha sonra hangi ayrıtların hangi düğümleri bağlandığı belirtilmelidir. Bir ayrıt her iki ucunda da bir düğüm olacak şekilde tanımlandığından graftaki tüm ayrıtların uç noktalarını bir düğüm ile ilişkilendirmek gerekir. Bu nedenle, her bir e ayrıt'ı için $\{v_1, v_2\}$ kümesi tanımlarız. Bunun anlamı e ayrıt'ının v_1 ve v_2 düğümlerini **bağladığı**dır. $v_1 = v_2$ olabilir. $\{v_1, v_2\}$kümesi $\delta(e)$ ile gösterilir ve düğümler kümesinin bir alt kümesidir.

Tanım: Bir **yönsüz (undirected) graf G** şunlardan oluşur: **G**(V,E)

(i) boş olmayan sonlu bir V düğümler kümesi

(ii) sonlu bir E ayrıtlar kümesi ve

(iii) bir $\delta:E\rightarrow P(V)$ fonksiyonu öyle ki her bir e ayrıtı için $\delta(e)$ V 'nin bir veya iki elemanlı bir alt kümesidir.



Şekil 7.1

Şekil 7.1 ' deki G grafına bakalım. Açıktır ki, G grafının düğüm kümesi $\{v_1, v_2, v_3, v_4\}$ ve ayrıt kümesi $\{e_1, e_2, e_3, e_4, e_5\}$. $\delta:E\rightarrow P(V)$ fonksiyonu şöyle tanımlanmaktadır:

δ: $e_1\{v_1\}$

δ: $e_2\{v_1,v_2\}$

δ: $e_3\{v_1,v_3\}$

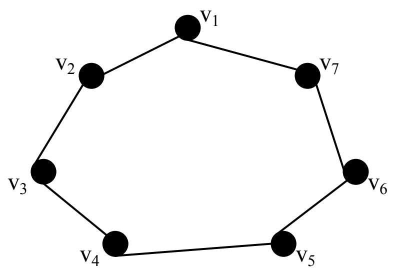
δ: $e_4\{v_2,v_3\}$

δ: $e_5\{v_2,v_3\}$

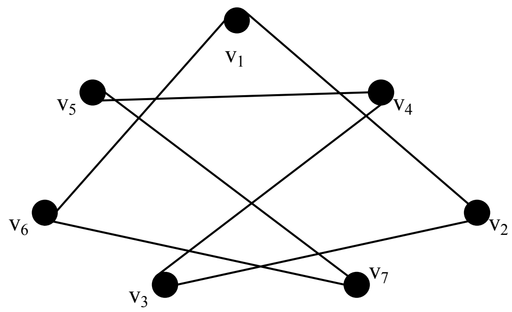
Bu basitçe, e_1' in v_1 düğümünü kendisine, e_2'nin v_1 ve v_2 düğümlerini vs. bağladığını gösterir.

Yukarıda görüldüğü gibi bir ayrıt bir düğümü yine kendisine bağlayabileceği (loop) gibi, bir düğüm hiçbir ayrıt ile bağlanmamış olabilir (v_4' te olduğu gibi). Ayrıca iki düğüm birden fazla ayrıt ile de (multiple ayrıt) bağlanmış olabilir.

Dikkat edilmesi gereken bir nokta bir graf ile onu temsil eden diyagram aynı değildir. Daha önce de söylediğimiz gibi bir graf bir fonksiyon ile birlikte iki kümeden oluşur. Şekil 7.1 kendi başına bir graf değildir sadece bir grafın gösterimidir. Verilen bir graf birbirinden çok farklı görünen iki graf ile gösterilebilir. Örneğin şekil 7.2 'deki iki diyagram çok farklı görünmelerine rağmen aynı G grafını temsil ederler. Şekil 7.2(a) 'daki graf 7 düğümlü çember (Wheel) graf olarak adlandırılır ve W_7 şeklinde gösterilir. Tüm n pozitif tamsayıları için n düğümlü ve n ayrıtlı bir W_n çember grafı vardır.



Şekil 7.2 (a)

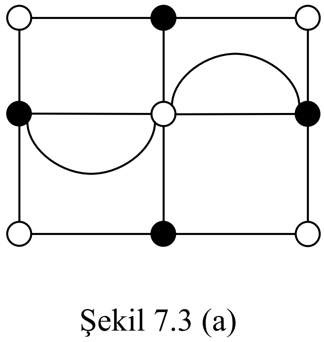


Şekil 7.2 (b)

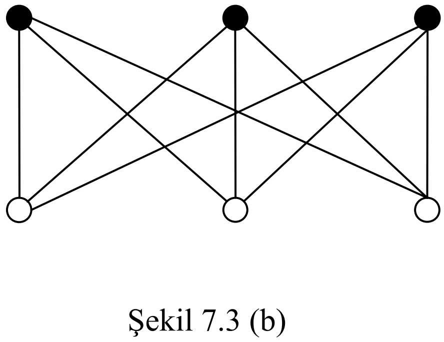
Eğer bir graf bir düğümü yine kendisine bağlayan (çevrim içermeyen) bir ayrıt ve aynı iki düğümü birden fazla bağlayan ayrıtlara sahip değilse **basit** (simple) graftır. Eğer grafta her bir ayrıta bir gerçel sayı atanmış ise graf bir ağ dır veya ağırlıklı graftır.

Bir yönlü graf (veya digraf) G=(V,E), burada V sonlu düğümü kümesi ve E 'de sonlu yaylar kümesidir. Öyle ki, E'deki her bir e yayı v ve w sıralı düğümlerini birleştirir. e=(v,w) demek, e, v'den w'ye bir yaydır veya e v'den çıkar w'ye girer demektir. **Karma**(mixed) G=(V,E) grafta ise, E'nin en az bir elemanı kenar ve e'nin en az bir elemanı yaydır.

İki parçalı grafta, düğümler kümesi, iki ayrı kümeye parçalanabilir, öyle ki V_1'deki bir düğüm ve V_2' deki bir düğüm arasındaki her bir ayrıt $G=(V_1,V_2;E)$ ile gösterilir. Eğer n düğümlü bir basit graf'ta her bir düğüm çifti arasında bir ayrıt var ise buna tam graf denir ve K_n ile gösterilir. Bir komple iki parçalı graf tamamen $|V_1|$ ve $|V_2|$ ile belirtilir. n ve m düğümlü komple iki parçalı graf $K_{n,m}$ ile gösterilir ve $|V_1|=n$ ve $|V_2|=m$ 'dir. Şekil 7.3 iki tane iki parçalı graf örneğidir. Her iki şekilde de V_1 'in düğümleri içi dolu noktalar ile, V_2 **'**nin düğümleri ise içi boş noktalar ile gösterilmiştir. (b)' deki graf komple iki parçalı $K_{3,3}$ grafıdır.



Şekil 7.3 (a)



Şekil 7.3 (b)

Tanım: (i) v ve w düğüm çiftini bağlayan bir e ayrıt' ı varsa bu iki düğüm komşudur (adjacent). Bu durumda hem v hem de w e' ye değer deriz ve ayrıca e de v ve w 'ya değer deriz.

(ii) e_1, e_2, … e_n ayrıtları en az bir ortak düğüme sahipse komşudur.

(iii) Bir v düğümünün derecesi (**degree**) $\sigma(v)$, v düğümüne bağlı olan ayrıtların sayısıdır. (Aksi belirtilmediği sürece v' yi kendisine bağlayan ayrıt v'nin derecesini iki arttırır.) Tüm düğümleri aynı r derecesine sahip grafa r dereceli düzenli (regular) graf veya r-derece denir.

(iv) Bir boş graf (**null**) veya tamamen bağlı olmayan (**totally disconnected**) graf ayrıt kümesi boş olan graftır.

(v) Bir tam (**complete**) graf ayrı düğüm çiftlerinin tümü bir ayrıt ile bağlı olan basit graftır ve n düğüm sayısı olmak üzere K_n şeklinde gösterilir.

(vi) Düğüm kümesi, tüm ayrıtların V_1 'in bir düğümünü V_2' nin bir düğümüne bağladığı $\{V_1, V_2\}$ şeklinde bir bölmelemeye sahip olan grafa iki parçalı (**biparite**) graf denir.

(vii) Bir komple iki parçalı (**complete bipartite**) graf V_1 'in tüm düğümlerini V_2' nin tüm düğümlerine tek bir ayrıt ile bağlayan iki parçalı graftır.

Örnek 7.1: G , V düğüm setinin $\{V_1, V_2\}$ bölmelemesine sahip olduğu bir iki parçalı graf olsun. Dikkat edilirse G basit graf olmak zorunda değildir. Gereken tek şey her bir ayrıt V_1 'in bir düğümü ile V_2' nin bir düğümünü bağlamalıdır. $v_1 \in V_1$ ve $v_2 \in V_2$ dersek, bunları bağlayan birden fazla ayrıt olabilir veya hiç ayrıt olmayabilir. Açıkça görülüyor ki, G'de çevrim yoktur.

G grafının farklı gözüken diyagramlar ile gösterilebileceğini belirtmiştik. Bir grafı göstermenin başka bir yolu da ileride tanımlayacağımız komşuluk matrisi (adjacency matrix) yardımıyla olur.

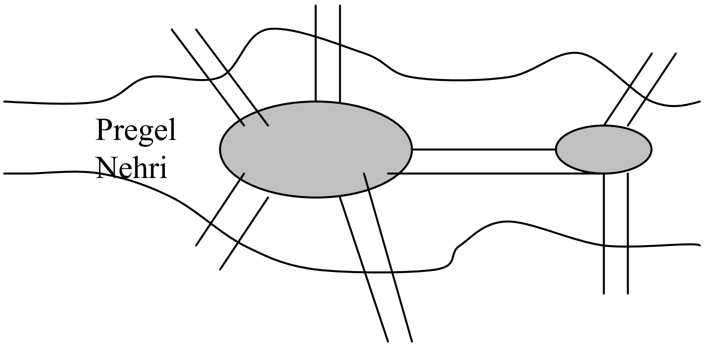
Tanım: Eğer, **V'** $\subseteq$ V(düğüm kümesi), E' $\subseteq$ E(Ayrıt kümesi) ve $\delta_{G'}(e)=\delta_G(e)$ ise, bir G' =(V',E') grafı G =(V,E) grafının alt grafıdır(subgraph) ve G' $\leq$ G şeklinde gösterilir.

G' grafının tüm e ayrıtları için $\delta_{G'}(e)=\delta_G(e)$ durumu G' alt grafının ayrıtlarının G de olduğu gibi aynı düğümleri bağlaması gerektiği anlamına gelir. Eğer G nin diyagramından bazı düğümleri veya ayrıtları silerek G' nin diyagramını elde edebiliyorsak G', G 'nin alt grafıdır. Tabii ki, bir düğümü siliyorsak ona çakışık olan tüm ayrıtları da silmeliyiz.

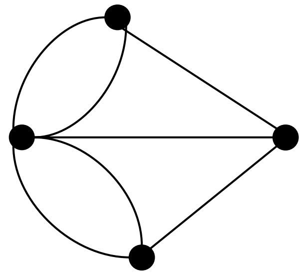
Örnek: Könnisberg Köprüsü problemi

Euler 1736 yılında yazdığı makale ile graf teorisinin doğmasına sebep olmuştur. Bu makale aşağıda tanımlanan Könnigsberg Bridge Problemlemini çözebilen bir teoriyi içeriyordu. Pregel nehri Königsberg kasabasının içinden akmaktadır. Nehrin ortasında şekil 7.4 (a) daki gibi nehrin kıyılarına ve birbirine köprüler ile bağlı iki ada bulunmaktadır. Königsberg kasabasının vatandaşları için problem kıyıların veya adaların birinden başlayıp tüm köprülerden sadece bir kez geçerek başladığımız yere yürüyebilir miyiz?

Euler öncelikle şekil 7.4 (b) ' deki gibi Königsberg coğrafyasının gerekli özelliklerini bir graf ile gösterdi. Her bir nehir kıyısı ve adalar bir düğüm ile köprüler de ayrıtlar ile temsil edildi. Graf teorisi terimleri ile problem şu hale geldi: grafın tüm ayrıtlarını içeren kapalı bir yol var mıdır?



Şekil 7.4 (a)



Şekil 7.4 (b)

Komşuluk ve Çakışım Matrisleri

Tanım: G düğüm kümesi $\{v_1, v_2,..,v_n\}$ olan bir graf olsun. G 'nın komşuluk matrisi; a_{ij}, v_i ve v_j 'yi bağlayan ayrı ayrıtların sayısı olmak üzere n x n A=A(G) matrisidir.

Komşuluk matrisi v_i ve v_j 'yi(düğüm) bağlayan ayrıtların sayısı, v_j ve v_i 'yi bağlayan ayrıtların sayısıyla aynı olduğundan simetrik olmalıdır. v_i düğümünün derecesi komşuluk matrisinden kolayca belirlenebilir. v_i de bir loop yoksa bu düğümün derecesi matrisin i. sütunundaki değerlerin toplamıdır. Her bir loop dereceyi iki kere etkilediğinden i. sütundaki değerleri toplarken a_{ii} diyagonal elemanın iki katı alınır.

Örnek 7.2: Aşağıdaki A komşuluk matrisi şekil 7.1 'de gösterilen grafa aittir.

$$A=\begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}.$$

$V=\{v_1, v_2, v_3, v_4\}$ ile A' nın satırları ve sütunları düğümleri temsil eder.

Grafın iki özelliği matrise bakılarak hemen görülebilir. Öncelikle, diyagonale bakıldığında bir tek çevrim(loop) vardır- v_1 ' den kendisine. İkincisi, son satır veya sütundaki 0' lar v_4 'ün bir tecrit edilmiş düğüm yani başka hiçbir düğüme bağlı olmayan (kendisi dahil) bir düğüm olduğunu gösterir.

Düğümlerin dereceleri matristen kolayca hesaplanabilir:

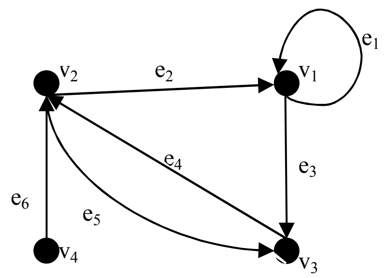
$\sigma(v_1) = 2.1+1+1=4$

$\sigma(v_2) = 1+2=3$

$\sigma(v_3) = 1+2=3$

$\sigma(v_4) = 0.$

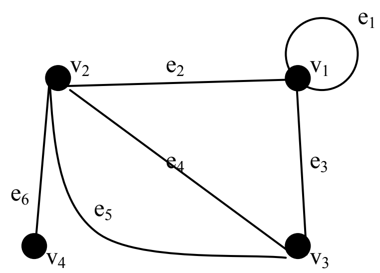
n ayrıtlı bir **yönlü grafın komşuluk matrisi** de nxn lik bir matristir $A=(a_{ij})$. Eğer i. düğümden den j. düğüme bir yay var ise $a_{ij}=1$ diğer durumda ise 0 dır.(Şekil 7.5)



1	0	1	0
1	0	1	0
0	1	0	0
0	1	0	0

Şekil 7. 5: Yönlü Graf ve Komşuluk Matrisi

Graf ve yönlü graflarda diğer önemli bir matris ise çakışım matrisidir. Komşuluk matrisinin tersine çakışım matrisinde çoklu ayrıt ve paralel yaylar gösterilebilir. V={1,2,,…,n} ve E={e_1,e_2,….,e_m} olmak üzere G=(V,E) grafı verilsin. G grafının çakışım matrisi, nxm boyutlu olan ve her bir satırın bir düğüme ve her bir kolonun bir ayrıta karşılık geldiği bir B=(b_{ik}) matrisidir, öyle ki eğer e_k, i ve j. düğümler arasındaki bir ayrıt ise k. kolonun elemanlarından b_{ik}=b_{jk}=1, diğerleri sıfırdır. Çevrim olan ayrıtın kolonunda sadece bir tek 1 vardır. Eğer grafta çevrim yok ise düğümlerin derecelerinin toplamı ayrıt sayısının iki katına eşittir. Çünkü bu özellikteki ayrıtlar iki düğümü birbirine bağlarlar.(Şekil 7.6.)



	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1	1	1	0	0	0
v_2	0	1	0	1	1	1
v_3	0	0	1	1	1	0
v_4	0	0	0	0	0	1

Şekil 7.6 Çakışım matrisi

Teorem 7.1: Eğer G, çevrim içermeyen ve m ayrıtlı bir çoklu graf ise, G'nin bütün düğümlerinin derecelerinin toplamı 2m'dir.

Bir yönlü grafın(çevrim içermeyen) çakışım matrisi, eğer e_k i den j'ye bir yay ise, k. kolondaki b_{ik}=-1 ve b_{jk}=1 diğer elemanlar sıfırdır.

Graflarda Bağlılık(Connectedness)

Bazı graflar tek bir parça halinde iken diğerleri çeşitli parçalardan oluşuyor olabilir. Bu fikri daha belirgin hale getirmek için yolları kullanabiliriz. Eğer bir graf çeşitli şehirleri bağlayan yol ağını temsil ediyorsa aklımıza şu soru gelebilir: her yoldan bir kere geçerek ve her şehre sadece bir kez uğrayarak aynı şehirden başlayıp aynı şehirde biten bir yolculuk yapılabilir mi?

Tanım: (i) G grafında n uzunluğunda ayrıt dizisi; i =1, 2, … n-1 için e_i ve e_{i+1} komşu olmak üzere e_1, e_2,…e_n ayrıtlarının dizisidir. Ayrıt dizisi, δ(e_i)={v_{i-1}, v_i} olmak üzere v_0,v_1,v_2,…v_{n-1},v_n düğüm dizisini belirler. v_0'a ilk düğüm, v_n 'e son düğüm denir.

(ii) Bir **yol (path)** tüm ayrıtları birbirinden ayrı (distinct) olan ayrıt dizisidir. Buna ek olarak eğer tüm düğümler de birbirinden ayrı ise bu yol basit (simple) yoldur.

Diğer bir tanım(**Yol**): Bir grafta, v_1 ve v_r düğümü arasındaki bir yol, düğümlerin v_1,e_1,v_2,e_2,v_3,e_3,…,e_r,v_r şeklindeki ayrıtların sonlu bir dizisidir. Burada, e_k, v_{k-1} ve v_k düğümleri arasındaki ayrıttır.

(iii) v_0=v_n ise ayrıt dizisi kapalıdır(closed). En az bir ayrıt içeren basit kapalı bir yol devre (circuit) olarak adlandırılır.

Bir ayrıt dizisi grafın diyagramında kalemi kâğıdın üzerinden kaldırmadan çizebileceğimiz herhangi sonlu ayrıt dizisidir. Ayrıtlar tekrar edilebilir veya çevrimler tekrarlanabilir. Ayrıt dizileri çok genel olduklarından kullanıma uygun değillerdir ve bu yüzden yollar tanımlanmıştır. Bir yolda aynı ayrıttan birden fazla geçmeye izin verilmez. Buna ek olarak eğer aynı düğümü birden fazla ziyaret etmiyorsak bu yol basit yoldur. Ayrıt dizisi veya yol, bir yerden başlayıp aynı yerde bitiyorsa kapalıdır.

Herhangi bir graf doğal olarak bileşen (component) adı verilen belli sayıda bağlı alt graflara bölünebilir. Bileşenler maksimal bağlı alt graflar olarak tanımlanabilirler. Bunun anlamı G_1, G 'nin bağlı bir alt grafı ise ve kendisi G 'nin başka herhangi bir bağlı alt grafının alt grafı değilse, G 'nın bileşenidir. Bu ikinci durum maksimal terimi ile anlatmak istediğimiz şeydir yani Σ, $G_1 \leq \Sigma$ olacak şekilde bir bağlı alt grafsa $\Sigma = G_1$, böylece G 'in G_1 'den daha büyük bağlı bir alt grafı yoktur

G şekil 7.1 'de gösterilen graf olsun. Bu grafın komşuluk matrisi:

$$A=\begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}. \text{ Dir.}$$

A' nın (i,j). elemanı v_i ve v_j düğümlerini bağlayan ayrıtların sayısıdır. Bunu bu iki düğümü bağlayan 1 uzunluğunda ayrıt dizilerinin sayısı şeklinde düşünebiliriz. Bu durumda, komşuluk matrisinin karesi:

$$A^2=\begin{vmatrix} 3 & 3 & 3 & 0 \\ 3 & 5 & 1 & 0 \\ 3 & 1 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}.\text{dir.}$$

A^2' de (i,j). eleman v_i ve v_j 'yi bağlayan 2 uzunluğunda ayrıt dizilerinin sayısını temsil eder. Örneğin, (2,2). eleman 5' tir ve v_2 'yi kendisine bağlayan 2 uzunluğunda 5 tane ayrıt dizisi vardır: e_2, e_2; e_4, e_4; e_5, e_5; e_4, e_5; e_5, e_4.

Bunun niçin böyle olduğunu görmek zor değildir. A^2' nin (i,j). elemanı A' nın i. satırı ile j. sütununun çarpılması ile elde edilir.

$$a_{ij}= \sum_{k=1}^{n} a_{ik} a_{kj} .$$

Toplamdaki r. terim $a_{ir}a_{rj}$, v_i ve v_r 'yi bağlayan ayrıtların sayısı ile v_r ve v_j 'yi bağlayan ayrıtların sayısının çarpımıdır. Bir başka ifade ile v_i ve v_j 'yi v_r aracılığı ile bağlayan 2 uzunluğundaki ayrıt dizilerinin sayısıdır. Tüm k değerleri için ortaya çıkanların toplanması v_i ve v_j 'yi bağlayan 2 uzunluğunda ayrıt dizilerinin sayısını verir.

Benzer şekilde A^3 'te (i,j). eleman v_i ve v_j 'yi bağlayan 3 uzunluğundaki ayrıt dizilerinin sayısıdır. Bu graf için

$$A^3=\begin{vmatrix} 9 & 9 & 9 & 0 \\ 9 & 5 & 13 & 0 \\ 9 & 13 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}.$$

Teorem 7.2: G, düğüm kümesi $\{v_1, v_2, \ldots, v_n\}$ ve komşuluk matrisi A olan bir graf olsun. A^n ‘ in (i,j). elemanı v_i ve v_j ‘ yi bağlayan n uzunluğunda ayrıt dizilerinin sayısıdır.

Yönlü Graflarda yol

Bir yönlü grafta v düğümünden w düğümüne olan yönlü bir yol, düğümlerin ve yayların $v_1,a_1,v_2,a_2,v_3,a_3,\ldots ,v_r,a_r,v_{r+1}$ şeklindeki sonlu bir dizisidir. Burada, ilk düğüm v ve son düğüm w ve a_i, v_i den v_{i+1}. düğüme olan yaydır. Eğer v’den w’ye bir yönlü yol var ise, v, w’ye bağlıdır ve w, v’den bağlıdır. Bir düğümden kendine olan bir yönlü yol, bir kapalı yönlü yoldur. Eğer düğümlerin çifti birbirine bağlı ise bu düğümlere **kuvvetli bağlı** çift denir. Bir grafta her bir düğüm çifti kuvvetli bağlı ise bu graf **kuvvetli bağlı**dır. Aksi halde **zayıf bağlı** graftır.

Eğer {v,w} kuvvetli bağlı çift ise, vRw ile, V düğüm kümesini iki ayrık alt küme sınıfına ayıran bir eşdeğerlik bağıntısı tanımlanır. Bu alt kümelerin her birine yönlü grafın bir kuvvetli parçası denir.

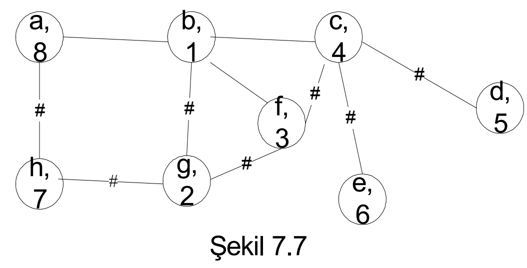
Teorem 7.3 : Eğer A bir yönlü grafın komşuluk matrisi ise, A’nın k. kuvvetinin($k\geq1$) (i,j) elemanı, i den j’ye olan k- yönlü yolun sayısını verir.

Grafların Bağlılık Testi

Tanım: Bir grafta eğer birbirinden ayrı düğümlerini bağlayan bir yol varsa bağlıdır(connected).

Verilen bir grafın bağlı olup olmadığının sorulması doğaldır. Elbette grafın şemasından bağlı olup olmadığının görülmesi kolaydır. Ancak büyük graflarda bu yöntem makul değildir. Grafın bilgisayara girilmesi durumunda, bağlılık testi için bir algoritma gereklidir. Böyle bir algoritma, grafın düğümlerinin yeniden etiketlendiği ilk derinlik arama(dept-first search) tekniğidir.

G grafının düğümleri $v_1,v_2,\ldots,v_n$ olsun.
Keyfi bir nokta seç ve onu 1 olarak etiketle.
1’e komşu etiketsiz bir düğüm seç ve onu 2 olarak etiketle.
{1,2} yi kullanılan ayrıt olarak işaretle ve tekrar kullanma.
Benzer şekilde v_i düğümünü k ile etiketle. Bu düğüme komşu olan ve etiketsiz olan bütün düğümleri ara ve birisini seçerek (k+1) olarak etiketle.
{k,k+1} i kullanılmış kenar olarak işaretle. Şimdi k’nın bütün komşu düğümleri etiketlenmiş olabilir.
Eğer öyle ise, (k-1). düğüme git ve onun etiketsiz komşu düğümlerini ara. Eğer böyle bir düğüm var ise onu (k+1) olarak etiketle ve {k-1,k+1} kenarı kullanılmış kenar olarak işaretle.
İşleme bütün düğümler işaretleninceye kadar devam et veya en az bir etiketlenmemiş düğüm ile 1. düğüme dön.
Örnek: Şekil 7.7’de 8 düğümlü {a,b,c,d,e,f,g,h} olan grafta, b seçilerek 1 olarak etiketlemiş ve {1,2} kullanılmış ayrıt olarak işaretlenerek devam edilmiş ve diğer düğümler etiketlenmiştir.



Şekil 7.7

İlk durumda graf bağlıdır ve tam olarak (n-1) kullanılmış ayrıt vardır. Periyodik olmayan bir graf n düğüm içerir ve bu (n-1) ayrıta grafın ilk derinlik arama uzaklık ağacı(first-depth search spanning tree) denir. Eğer DFS tekniği ile tüm n adet düğüm etiketlenmemiş ise graf bağlı değildir sonucuna varırız. Bu algoritmanın en kötü durumda karmaşıklığı, eğer m ayrıt var ise en fazla 2m araştırma yapılacak ve n adet etiketlenecek düğüm olacaktır. Böylece karmaşıklık en kötü durumda n+2m olacaktır. m'nin en büyük değeri n(n-1)/2 olduğundan(bütün düğüm çiftleri arasında bir ayrıt olduğu durum) en kötü durumda karmaşıklık $O(n^2)$ olacaktır.

8.2 Yollar ve Devreler

Euler Yolları (Eulerian Paths)

Tanım: G grafındaki bir Euler Yolu, G' nın tüm ayrıtlarını kenar olarak bir kere içeren kapalı bir yoldur. Kapalı bir Euler yolu bir Euler devresidir. Bir graf içinde en az bir Euler Yolu barındırıyorsa bu graf Euler grafıdır.

Euler devresi fikri meşhur Königsberg Köprüsü Probleminden ortaya çıkmıştır. Pregel nehri Königsberg kasabasının içinden akmaktadır. Nehrin ortasında şekil 7.4 (a) daki gibi nehrin kıyılarına ve birbirine köprüler ile bağlı iki ada bulunmaktadır. Königsberg kasabasının vatandaşları için problem, kıyıların veya adaların birinden başlayıp tüm köprülerden sadece bir kez geçerek başladığımız yere yürüyebilir miyiz?

Euler, öncelikle şekil 7.4 (b) ' deki gibi Königsberg coğrafyasının gerekli özelliklerini bir graf ile gösterdi. Her bir nehir kıyısı ve adalar bir düğüm ile köprüler de ayrıtlar ile temsil edildi. Graf teorisi terimleri ile problem şu hale geldi: grafın tüm ayrıtlarını içeren kapalı bir yol var mıdır?

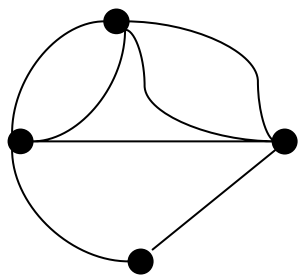
Bir yolda hiçbir ayrıt dan birden fazla geçilemeyeceğinden Euler yolu tüm ayrıtları sadece bir kez içerir fakat düğümlerden birden fazla geçilebilir.

Bağlı bir G grafında Euler yolu olup olmadığını belirlemek için gereken durumu tanımlamak çok kolaydır: bütün düğümlerin derecesi çift olmalıdır.Bunu görmek için G bağlı ve Euler yoluna sahip olsun. G bağlı olduğundan Euler yolunun düğüm dizisi bütün düğümleri içerir. Yol ne zaman bir düğümden geçse bu derecesine iki katkı yapar. Tüm ayrıtlar yolda bir kere bulunduğundan her düğüm çift dereceye sahip olmalıdır.

Königsberg'dekiler aradıkları yolu bulamamakta haklıdırlar zira böyle bir yol yoktur. Problemi temsil eden şekil 7.4 (b)' deki graf bağlıdır fakat gerekli koşulu sağlamaz. Aslında tüm düğümlerin derecesi tektir. Aşağıdaki teorem bu soruyu sabitler.

Teorem 7.4: Çevrim içermeyen bağlı bir G grafı sadece ve sadece tüm düğümleri çift dereceli ise Euler grafıdır.

Örnek: Şekil'deki grafta tüm düğümlerin dereceleri çift olduğu için bu bir Euler grafıdır.



Bir grafta Euler devresi bulmak için kolay bir yol Fleury'in algoritması olarak bilinir. Bu yöntemde, herhangi bir düğümden başlanır ve geçilen bir ayrıt silinir. Aynı zamanda, yardım etseniz bile bir köprü asla geçilmez. Eğer geçilen ayrıtlar silinerek başlanılan noktaya ulaşılabiliyor ise, devre Euler devresidir ve graf Euler dir.

Teorem 7.5: Bağlı ve çevrim içermeyen Euler olmayan bir G grafında ancak ve ancak tam olarak iki tek dereceli düğüm var ise bir Euler yolu vardır.

İspat: Eğer G, u'dan v'ye bir Euler yoluna sahip ise, u ve v'nin her ikisi de tek dereceli ve bu yolda her düğümden geçip her ayrıt bir kez ziyaret edildiği için diğer düğümlerin her biri çift dereceli olmalıdır. Diğer taraftan, G nin u ve v olan iki tek dereceli düğüm ile bağlandığını kabul edelim. Bu u ve v düğümleri ya komşudur veya değildir. İlk durumda ikisi arasında bir e ayrıtı olsun. Her bir düğümü çift dereceli olan G' grafını elde etmek için e'yi silelim.Eğer G' bağlı ise u'dan başlayıp v'ye kavuşan bir Euler yolu elde edilir. Eğer G' iki parça ise, birisi u'yı içeren G_1 diğeri de v'yi içeren G_2'dir. Elbette her ikisi de Euler grafıdır. Dolayısı ile bir Euler yolu bulunabilir.

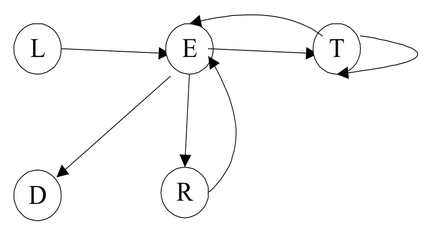
Teorem 7.6: Bir zayıf bağlı yönlü graf ancak ve ancak her bir düğümün giriş ve çıkış dereceleri aynı ise bir yönlü Euler devresi içerir.

Kodlama ve de Brujin yönlü grafları

Euler yol ve devrelerinin, bilgisayar bilimleri, yöneylem araştırması, kriptografi ve taşıma problemleri gibi ilginç ve yararlı uygulamaları vardır. Bu bölümde birkaç örnek verilecektir.

Çinli postacı problemi, keyfi bağlı bir ağın Euler grafına genişletildiği bir optimizasyon problemidir. Problemde; Bir posta taşıyıcısı postaneden çıkar, bölgesindeki her bir bloğa postaları dağıtır ve ofisine geri döner. Eğer yolu üzerindeki her bir cadde köşesini bir düğüm ve iki köşe arasındaki yolu bir ayrıt olarak alırsak, bu problemin modeli olan bir graf elde ederiz. Eğer graf bir Euler Grafı ise postacı her bir caddeyi bir kere geçmelidir. Eğer Euler grafı değil ise, postacı bazı caddeleri tekrarlayacaktır. Bir optimizasyon problemi bu tekrarlanan caddeleri toplam gidilen yolu en aza indirecek şekilde konumlandırmaktır.

Diğer bir problemde Euler grafının kodlama teorisindeki uygulamasıdır. m harf uzunluğunda ve n farklı harfi olan bir kelime bir zayıf bağlı, n düğümlü, m-1 yaylı G grafı olarak ilişkilendirilebilir. Burada, eğer kelimenin ilk harfi ve son harfi farklı ise yönlü bir Euler yolu, aynı ise yönlü bir Euler devresinin oluştuğu görülür. Uygulama olarak LETTERED kelimesi ile oluşturulan yönlü Euler grafında m=8 ve n=5'dir. Şekil 7.8'de gösterilen yönlü grafta,ilk harf L 'den son harf D'ye gitmek için 5 düğüm ve 8 yaydan geçilmelidir.



Şekil 7.8

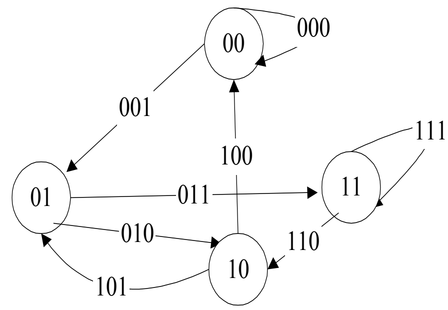
$A_1,A_2,\ldots A_n$ gibi n farklı harfli bir kelimde, $f(A_i)$, A_i'nin kelimdeki frekansı olsun. Buradan n harfin frekanslarının toplamı m dir. Yönlü grafta, A_i'den A_j'ye olan yay sayısını gösteren m_{ij}, A_j 'nin A_i'nin hemen arkasından gelme sayısını göstersin. Örneğim MATHEMATICS kelimesinde, $m_{AT}=2$, $m_{TA}=0$, $m_{TH}=1$ ve böyle devam eder. Buradan $M=(m_{ij})$, nxn boyutlu bir matris olarak tanımlanır. M'de i. Satırın satır toplamı A_i düğümünün çıkış derecesi, j. kolonun kolon toplamı A_j'nin giriş derecesidir.

Böylece, n farklı harfli bir kelimeye karşılık olarak n pozitif tamsayının frekansı ve elemanları pozitif olan nxn'lik bir matrisimiz vardır. Örneğin "LETTERED" kelimesinin frekans kümesi, {1,3,1,1,2} ve 5x5 lik matris yanda gösterilmiştir.

$$
M = \begin{array}{cccccccc}
 & D & E & L & R & T & \\
D & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 1 & 0 & 0 & 1 & 1 & 3 \\
L & 0 & 1 & 0 & 0 & 0 & 1 \\
R & 0 & 1 & 0 & 0 & 0 & 1 \\
T & 0 & 1 & 0 & 0 & 1 & 2 \\
 & 1 & 3 & 0 & 1 & 2 &
\end{array}
$$

Euler graflarının diğer bir uygulaması da de Brujin graflarıdır. Uzunluğu n-1 olan 2^{n-1} ikili kelime var. Burada 2^{n-1} düğümlü bir yönlü graf oluşturacağız. Her bir n-1 uzunluğundaki kelime bir düğümde olsun. Her bir $v=a_1a_2\ldots a_n$ şeklindeki düğümden , $a_2a_3\ldots a_n0$ ve $a_2a_3\ldots a_n1$ şeklinde iki n harfli kelimeyi temsil edecek şekilde v_1 ve v_2 yaylarını çiz. Böylece, n uzunluğundaki kelimeleri temsil eden 2^n yaylı yönlü graf çizilmiş olur. Bu graf G(2,n) zayıf bağlı de Brujin grafıdır ve her bir düğümün giriş ve çıkış derecesi eşit olduğundan Euler grafıdır. G(2,3) yönlü grafı Şekil 7.9'da gösterilmiştir.

Daha genel olarak, p harfli alfabe için G(p,n) giriş ve çıkış derecesi p olan ve p^{n-1} düğüm p^n yay içeren bir de Brujin yönlü grafıdır. Böylece G(p,n) bir Euler grafıdır. Bu yönlü grafta bir Euler yolu, bir dizide p^n yay içerir. Bu kelimelerin ilk harfleri ile bir dizi oluşturmak istenirse, böyle bir dizi, $a_1a_2 \ldots a_r$ dir, burada $r=p^n$ dir. Buradan uzunluğu n olan r farklı kelime $a_ia_{i+1} \ldots a_{i+n-1}$ şeklindedir, burada alt indiste belirtilen toplama işlemi modulo r şeklindedir. Örneğin p=2, n=3 ise a_9 a_1 ile aynı olacaktır. Örneğin, Şekil 7.9'daki yönlü Euler grafında, 00'dan başlayan devre, 000,001,011,111,110,101,010,100 olan sekiz yay dizisini içerir. Bunların ilk harfleri ile oluşturulan dizi, 00011101 şeklinde ve buradan oluşturulacak $a_ia_{i+1}a_{i+2}$ şeklindeki üç harflik dizi de $a_7a_8a_9=a_7a_8a_1=010$ olacaktır.



Şekil 7.9

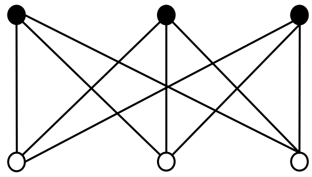
Buradan de Brujin dizisini biçimsel olarak, p ve n iki pozitif tamsayı için tanımlarsak; Eğer S, p harf içeren bir alfabe ise, r ($r=p^n$) harfin $a_1a_2 \ldots a_r$ dizisine B(p,n) ile gösterilen de Brujin Dizisi denir. S'den n uzunluğundaki her bir kelime, $a_ia_{i+1} \ldots a_{i+n-1}$ (i=1,2,,…,r) olarak gerçeklenebilir. Burada alt indisteki toplama işlemi modulo r olarak gerçeklenir.

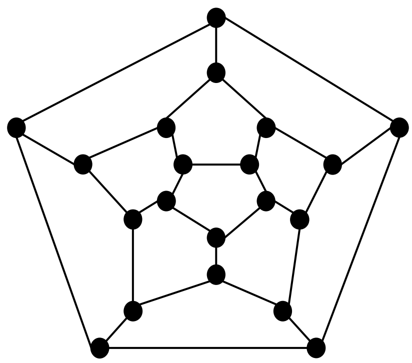
Hamilton Devreleri (Hamiltonian Circuits)

Benzer bir problem de herhangi bir ayrıtdan birden fazla geçmemek kaydıyla her bir düğümü sadece bir kez ziyaret edip başladığımız yere geri dönebilir miyiz? Şeklinde sorulabilir. Bu problem Hamilton tarafından irdelenmiştir ve ismi bu yollar ile birlikte anılmaktadır.

Tanım: Eğer bir grafta her bir düğümden sadece bir kere geçilen bir yol varsa iki düğüm arasındaki yola, **Hamilton yolu** denir. Bir graftaki Hamilton devresi tüm düğümlerden bir kez geçen bir devredir. Bir graf içinde bir Hamilton devresi barındırıyorsa Hamilton grafıdır. Her bir düğümden tam olarak bir kere geçen ve tüm ayrıtların farklı olduğu bir kapalı yol Hamilton devresidir.Bir graf Hamilton devresi içeriyorsa bu bir Hamilton grafıdır. Bir yönlü grafta, bir düğümden diğerine geçen yönlü yol eğer her düğümden bir kere geçerse bu yönlü Hamilton yoludur. Bir kapalı yönlü Hamilton yolu bir yönlü Hamilton devresidir.

Örnek 7.3: Şekil 7.10 'da iki tane Hamilton devresi vardır.


Şekil 7.10 (a)


Şekil 7.10 (b)

Euler grafları basit bir karaktere sahipken aynı durum Hamilton grafları için doğru değildir.

Aslında bir asırdan beri üzerinde çalışıldığı halde Hamilton graflarının karakteri hakkında her şey bilinmemektedir (Karakter ile bir grafın Hamilton olması için gerek ve yeter koşul kastedilmiştir). Bu graf teorisinin çözülememiş büyük problemlerinden biridir. Açık bir gerek koşul grafın bağlı olmasıdır. Ayrıca çeşitli yeter koşullar da bilinmektedir.

Bununla birlikte Hamilton grafı için aşağıdaki teoremler verilebilir.

Teorem 7.7: Bir n düğümlü($n \geq 3$) basit grafta, eğer komşu olmayan düğümlerin her çiftinin derecesi toplamı en az n ise bu bir Hamilton grafıdır.

Sonuç: Eğer, G n ($n \geq 3$) düğümlü basit bağlı bir graf ise ve tüm v düğümleri için derecesi $\sigma(v) \geq n/2$ ise G Hamilton'dur.

Dereceler ile ilgili koşul G 'nin Hamilton olması için gerek koşul değildir o halde, bu koşulu sağlamayan bir graf da Hamilton olabilir. Şekil 7.10 (b) 'ye bakarak bunu görebiliriz. Grafın 15 düğümü vardır, her düğümün derecesi 3 'tür fakat hala Hamilton grafıdır.

Teorem 7.8: Bir n düğümlü basit grafta, eğer komşu olmayan düğümlerin her çiftinin derecesi toplamı en az n-1 ise bu graf bir Hamilton yolu içerir.

Sonuç: Bir n düğümlü basit grafta, eğer her bir düğümün derecesi en az (n-1)/2 ise bu graf bir Hamilton yolu içerir.

Hamilton devresinin Uygulaması: Hamilton yolu ve devrelerinin ilginç uygulamaları vardır. Buna bir örnek aşağıda verilmiştir.

Örnek(Satıcı Seyahat Problemi): Bir ülkedeki şehirler düğümleri gösterecek, uçak seferi olan şehirler arasındaki bağlantılarda ayrıtlar olmak üzere bir graf oluşturulsun. Bir satıcı her bir şehre bir kere uğrayıp tekrar başladığı yere dönmek üzere bir yolculuk programı yapmak istiyor. Böyle herhangi bir tur bir Hamilton devresidir. Böyle bir devrenin var olduğu kabul edilirse, toplam maliyeti en az olan bir yol bulmak bir optimizasyon problemidir.

Örnek(Planlama) : Bir makine atölyesinde n adet makine bulunsun. Bir iş keyfi bir sırada olmayacak şekilde bu makineler arasında yapılacaktır. Her bir makine bir yönlü grafta bir düğümü temsil etsin. Her bir düğümden diğerine bir yay çiz. Bu yönlü grafta herhangi bir yönlü Hamilton yolunun bulunması bir planlamadır. Eğer, bir işin i.makineden j. Makineye giderken gerekli olan düzenleme zamanı c_{ij} ise, en az zamana sahip bir planlamayı bulmak yine bir optimizasyon problemidir.

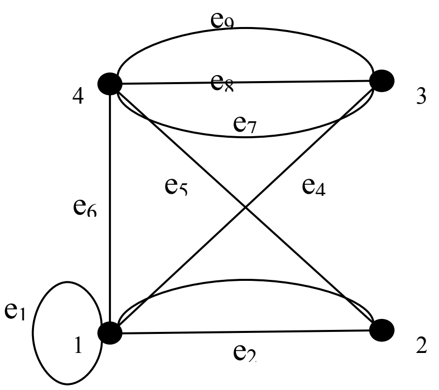
8.3 Grafların İzomorfizmi

Aşağıdaki gibi tanımlanan G ve Σ graflarını düşünelim. G' nin düğüm kümesi {1,2,3,4}, komşuluk matrisi A ve Σ 'nin düğüm kümesi {a,b,c,d}, komşuluk matrisi B olsun.

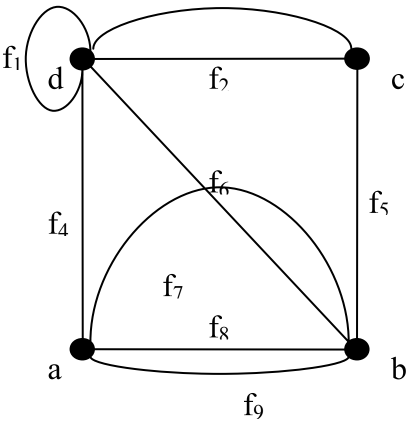
Aşağıdaki gibi tanımlanan G ve Σ graflarını düşünelim. G' nin düğüm kümesi {1,2,3,4}, komşuluk matrisi A ve Σ 'nin düğüm kümesi {a,b,c,d}, komşuluk matrisi B olsun.

$$A = \begin{vmatrix} 1 & 2 & 1 & 1 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 3 \\ 1 & 1 & 3 & 0 \end{vmatrix} \qquad B = \begin{vmatrix} 0 & 3 & 0 & 1 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 1 & 1 & 2 & 1 \end{vmatrix}$$

G ve Σ graflarını temsil eden diyagramlar şekil 7.11 ‘da gösterilmiştir.



Şekil 7.11 (a)



Şekil 7.11 (b)

Biraz dikkatli bakılırsa şekil 7.11‘ de gösterilen grafların aynı olduğu görülebilir. Σ grafındaki a,b,c,d düğümlerini 3,4,2,1 şeklinde; ve f_i ayrıtlarını i=1,…,8 için e_i ile tekrar etiketlersek şekil 7.11 ‘deki iki diyagrama aynı grafın farklı gösterimleri şeklinde bakabiliriz. Tabii ki, G ve Σ grafları birebir aynı değildir. Örneğin farklı düğüm kümelerine sahiptirler. Öte yandan aynı yapıya sahiptirler. G ve Σ grafları izomorfiktir graflar diyebiliriz.

Σ ‘nın düğümlerini yeniden etiketleyerek G ve Σ ‘nın düğüm kümeleri arasında bir bijeksiyon tanımlamış oluruz.

Tanım: G ve Σ iki graf olsun. G ‘den Σ’ a bir izomorfizm (Θ, Φ) bir bijeksiyon çiftinden oluşur.

$$\Theta: V_G \rightarrow V_\Sigma \text{ ve } \Phi: E_G \rightarrow E_\Sigma$$

öyle ki G’ nın tüm e ayrıtları için eğer $\delta_G(e) = \{v, w\}$ ise $\delta_\Sigma(\Phi(e))=\{ \Theta(v), \Theta(w)\}$.

İki graf, bir graftan diğerine bir izomorfizm varsa izomorfiktir denir ve G Σ şeklinde gösteririz.

$\delta_G(e) = \{v, w\}$ ise $\delta_\Sigma(\Phi(e))=\{ \Theta(v), \Theta(w)\}$ olması şartının anlamı iki grafın ayrıtları ve düğümleri arasındaki uyuşmanın doğru şekilde sağlandığından emin olmak içindir.

Basit bir G grafı için G’ dan Σ ‘ya bir izomorfizm tanımlamak için sadece uygun $\Theta: V_G \rightarrow V_\Sigma$ düğüm bijeksiyonunu belirlemek gerekir. Bunun nedeni herhangi düğüm çiftini birleştiren en az bir tane ayrıt vardır o halde, bir kez Θ tanımlandığında gerekli özellikleri sağlayan sadece bir tane $\Phi: E_G \rightarrow E_\Sigma$ fonksiyonu vardır.

İzomorfik grafların aynı yapıya sahip olmaları gerektiğinden birine ait graf teorisine dahil herhangi bir özellik diğerinde de bulunmalıdır. Bu özelliklerin bir kısmı aşağıdaki teoremde sıralanmıştır.

Teorem 7.8: (Θ, Φ) G’ dan Σ ‘ya bir izomorfizm olsun. Bu durumda;
(i) G ve Σ aynı sayıda düğüme sahiptir;

(ii) G ve Σ aynı sayıda ayrıt’ e sahiptir;

(iii) G ve Σ aynı sayıda bileşene sahiptir;

(iv) birbirine karşılık gelen düğümler aynı dereceye sahiptir;

(v) G basitse, Σ da öyledir;

(vi) G Euler grafı ise Σ da Eulerdir.

(vii) G Hamilton grafı ise Σ da Hamiltondur.

İzomorfizm Prensibi

İki grafın izomorfik olduğunu göstermek için birinden diğerine bir izomorfizm bulunmalıdır; iki grafın izomorfik olmadığını göstermek için ise bir grafın sahip olduğu ama diğerinin sahip olmadığı bir graf teorisine dahil bir özellik bulunmalıdır.

8.4 Düğüm Boyama, Ağaçlar ve Düzlemsel Graflar

Eğer bir grafta, iki komşu düğüm aynı renkte olmayacak şekilde, her bir düğüme bir renk verilirse graf boyalıdır denir. Eğer böyle bir boyama en çok k renk kullanılarak mümkün olursa, graf k-renklidir. Böyle k-renkli bir G grafında en küçük k değeri G'nin kromatik sayısıdır.

Bir grafta ancak ve ancak hiç ayrıt yok ise kromatik sayısı birdir. n düğümlü bir tam grafın kromatik sayısı n, iki parçalı grafın kromatik sayısı ise 2 dir. Bir ağacın kromatik sayısı 2 dir. p düğümlü bir devre ancak ve ancak, p çift ise 2-renkli, benzer şekilde eğer bir G grafı tek devre(devredeki ayrıt sayısı tek) içeriyorsa, G grafı 2-renkli değildir. Eğer bir grafta hiç tek devre yoksa graf, 2-renklidir.

Graflarda düğüm boyama için değişik algoritmalar geliştirilmiştir. İki örnek algoritma aşağıda verilmiştir.

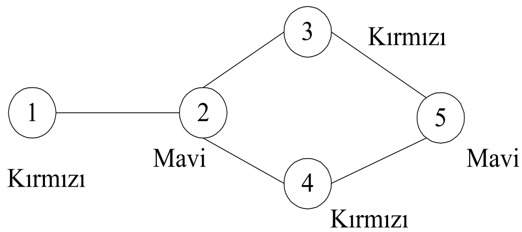
Graf Boyama için açgözlü bir algoritma

1. Bir düğümü al ve kullanılmayan bir rengi ver

2.Komşu olan düğümlere farklı komşu olmayanlara mümkün olduğunca aynı renk ver.

3. İşlemleri bütün düğümler için tekrarla.

Şekil 7.12'de gösterilen graf için boyama örneği verilmiştir. İlk 1 nolu düğümden başlayıp kırmızı renk, ona komşu olmayan 3 ve 4 nolu düğümler yine kırmızı, birbirine komşu olmayan 2 ve 4 nolu düğümlere ise mavi renk verildi. Böylece sadece iki renk kullanılmış olur.



Şekil 7.12

Diğer bir algoritma: Yukarıdaki basit algoritmaya biraz sezgisel yaklaşım ekleyerek geliştirelim.

1. Düğümleri($v_1,v_2,\ldots,v_n$) derecelerine göre azalan sırada sırala $\sigma(v_1)\geq \sigma(v_2)\geq\ldots\geq \sigma(v_n)$
2. Renk 1'i v_1'e ve listede v_1'e komşu olmayan sonraki düğüme(eğer var ise) ver
3. Renk 2'yi listede boyanmayan ve renk 2 ile boyanmış düğümlere komşu olmayan düğümlere ver.
4. Eğer boyanmayan düğüm kalmış ise, renk 2 yi ver.
5. Bu işleme bütün düğümler boyanıncaya kadar devam et.

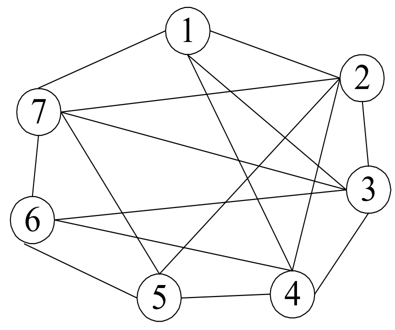
Graf boyama fikri birçok planlama probleminin çözümünde faydalıdır.

Örnek: Bir üniversitede final sınavlarının çakışmayacak şekilde planlamasının yapılması:

Çözüm: Bu planlama problemi, düğümler dersleri ve düğümler arasındaki ayrıtlar eğer derslerde ortak öğrenci var ise onu temsil eden bir graf modeli oluşturulur. Final sınavı için her zaman dilimi farklı bir renk ile gösterilir. Sınav planlaması oluşturulan grafın boyanması problemine dönüşür. Örneğin, 7 adet planlanacak sınav olsun. Dersler 1-7 arasında numaralandırılmış olsun. 1 ve 2, 1 ve 3, 1 ve 4, 2 ve 3, 2 ve 4, 2 ve 5, 2 ve 7, 3 ve 4, 3 ve 6 , 3 ve 7, 4 ve 5, 4 ve 6, 5 ve 6, 5 ve 7, 6 ve 7 nolu dersler ortak öğrenci bulundursunlar. Şekil 7.13'de gösterilen grafta, planlama problemi düğüm boyama problemine dönüşmüş olur.

Bu grafın kromatik sayısı 4 olduğundan sınavlar için 4 zaman dilimi gereklidir. Graftaki renkler, 1ve 6 kırmızı, 2 Mavi ,3 ve 5 Yeşil, 4 ve 7 kahverengi olarak boyanır. Bu zaman dilimlerindeki sınavlar aşağıdaki tabloda gösterilmiştir.

Zaman Dilimi	Dersler
I	1,6
II	2
III	3,5
IV	4,7

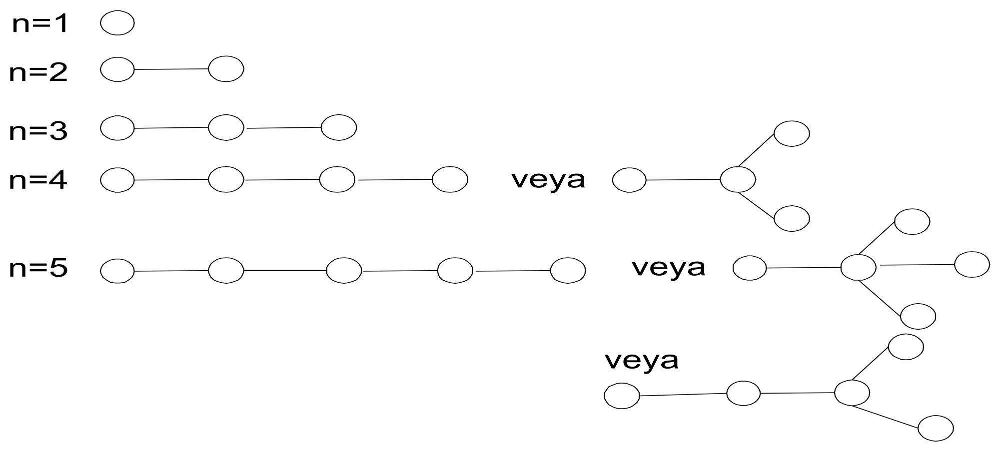


Şekil 7.13

Ağaçlar

Tanım: İçinde devre (circuit) içermeyen bağlı graflara **ağaç (tree)** denir.

Tanımdan da açıkça görüldüğü gibi bir ağaçta loop veya çoklu ayrıt yoktur. Herhangi bir loop kendi başına bir devredir ve e_i ve e_j aynı düğüm çiftinin bağlıyorsa e_i, e_j dizisi de bir devredir. Bazı ağaç örnekleri Şekli 7.14'de gösterilmiştir..



Şekil 7.14. Düğüm sayısına göre farklı ağaçlar.

Teorem 7.9.: n düğümlü bir G grafında aşağıdakiler eşdeğerdir.

i. G bir ağaçtır.
ii. G'de her düğüm çifti arasında, sadece bir yol vardır.
iii. G bağlıdır ve G'deki her bir ayrıt bir köprüdür.(köprü silinince graf bağlı olmaktan çıkar)
iv. G bağlıdır ve (n-1) ayrıtı vardır.
v. G çevrimsizdir ve (n-1) ayrıtı vardır.
vi. G çevrimsizdir ve, G'de komşu olmayan iki keyfi düğüm bir ayrıt ile birleştirildiği zaman sonuçtaki genişleyen G' grafı tek bir tur içerir.

vii. G bağlıdır, ve eğer G'de herhangi komşu olmayan iki keyfi düğüm bir ayrıt ile birleştirilirse, elde edilen yeni grafın tek bir çevrimi vardır.

Genel olarak, ağaçlar ile ilgili algoritmalar üç türlüdür.

- Verilen bir ağaçta arama ve etiketleme algoritmaları
- Farklı türlerde ağaç oluşturmak için algoritmalar.
- Özel bir türdeki ağaçları saymak için algoritmalar.

Ağaçlar ile ilgili Tanımlar ve özellikleri

- Bir ağaç, çevrim içermeyen bir bağlı yönsüz graftır.
- Bir yönsüz graf ancak ve ancak, herhangi iki düğümü arasında tek bir basit yol var ise bir ağaçtır.
- Bir köklü ağaç, bir ağaçtan bir düğümün kök olarak belirlenmesi ve herbir ayrıt kökten yönlendirilerek elde edilen bir yönlü graftır.
- Bir köklü T ağacında, (u,v) bir yönlü ayrıt olsun,
  - ✓ u, v'nin ebeveyni ve v'de u'nun çocuğudur,
  - ✓ aynı ebeveyne sahip çocuklara kardeş denir;
  - ✓ bir v düğümünün kök haricindeki ataları, kökten v'ye kadar olan yol üzerindeki düğümlerdir,
  - ✓ v düğümünün torunları v'yi ata olarak gören düğümlerdir;
  - ✓ bir yaprak, hiç çocuğu olmayan bir düğümdür,
  - ✓ çocuğu olan düğümlere iç düğümler denir;
  - ✓ torunlarıyla, birlikte bir v düğümü ve bu torunlara komşu bütün ayrıtlar bir alt graf oluşturur.
- Her iç düğümü ≤ *m* çocuğa sahip olan bir köklü ağaca m-ilişkili ağaç denir, eğer m=2 ise ikili ağaçtır.
- Bir köklü ağaçta, bir v düğümünün seviyesi, kökten v'ye olan tek yolun uzunluğudur.
- Bir köklü ağacın yüksekliği, düğümlerin seviylerinin en büyüğüdür.
- Yüksekliği h olan bir köklü m-ilişkili ağaç, eğer bütün yapraklar *h* veya *h-1* seviyesinde ise dengeli ağaçtır.
- Bir sıralı köklü ağaçta, herbir iç düğümün çocukları sıralıdır. Eğer bir düğümün iki çocuğu varsa, ilk çocuğa sol alt ağaç, ve sağ çocuğa sağ alt ağaç denir.
- Ağaçlar; doymuş hidrokarbonları, Kuruluşları, Dosya kataloglarını, parallel işlem için ağ iç bağlantılarını modellemek için kullanılabilir.
- ***Ağaçların Özellikleri***
  - ▪ n düğümlü bir ağacın tam olarak n-1 ayrıtı vardır.
  - ▪ i adet iç düğümü olan bir tam m ilişkili ağaçta n=m.i +1 düğüm bulunur.
  - ▪ Bir tam m-ilişkili;
    - ❑ *n düğümlü* ağacın, $i = (n-1)/m$ iç düğümü ve $l = [(m-1)n + 1]/m$ yaprağı vardır.
    - ❑ i iç düğümlü ağacın $n = m \cdot i + 1$ düğümü ve $l = (m-1)i + 1$ yaprağı vardır.
    - ❑ l yapraklı ağacın, $n = (m \cdot l-1)/(m-1)$ düğümü ve $i = (l-1)/(m-1)$ iç düğümü vardır.
  - ▪ Yüksekliği h olan m-ilişkili bir ağaçta en çok m^h yaprak vardır.

- Eğer yüksekliği h olan bir m-ilişikili ağacın l yaprağı var ise, $h \geq \log_m l$ dir. Eğer m-ilişikili ağaç tam ve dengeli ise, $h = \log_m l$ ‘dir.

Ağaçların Uygulamaları

- ***İkili Arama Ağacı:*** Bir sıralı köklü ikili ağaçta herbir düğüme; sol alt ağacındaki düğümlerdeki anahtarlardan büyük ve sağ alt ağacındaki düğümlerde bulunan anahtarlardan küçük bir anahtar atanır.(*İkili Arama Ağacı Algoritması.)*
- ***Karar Ağacı*****:** Herbir iç düğümün bir karara karşılık geldiği bir köklü ağaçta, kararın herbir olası sonucu için bu düğümlerde bir alt ağaç bulunur.(Örnek, Sahte jetonların bulunması)
- ***Önek Kodları*****:** Farklı uzunluktaki bit dizilerini kodlamaya dayalı kodlar, bir harf için bit dizisinin diğerbir harfin ön ekinde olmaması özelliği ile harfleri kodlamakta kullanılır.

  Huffman Kodlama Algoritması

  Bir ikili ağacı verilen $w_1 \leq w_2 \leq \ldots \leq w_n$ ağırlıklar ile aşağıdaki şekilde yinelemeli olarak oluştur:

  1. En küçük iki ağırlığında köklü alt ağaçlı şekilde bir ağaç oluştur. Onların birleştirilmiş ağırlıkları, diğer dalların oluşturulması için ağırlıkların kullanılabileceği bu alt ağacın kökünün ağırlığı olur.
  2. Bütün ağırlıklar birleştirilene kadar adım 1’i tekrarla.
  3. Herbir iç düğümün 2 dalı 0 ve 1 olarak etiketlenir. Herbir harf, ikili ağaçtan elde edildiği şekilde etiketlerin yolunu alır.

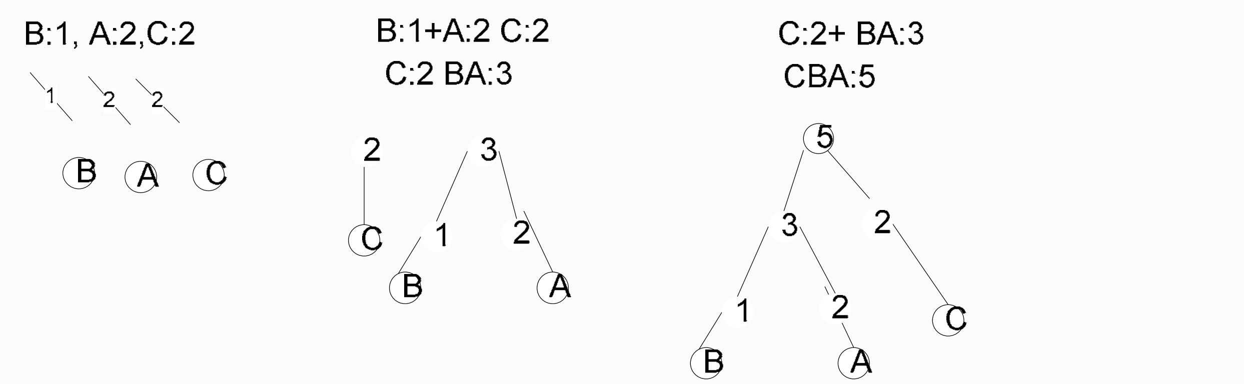
Örnek: Ağacın oluşturulması

İlk önce karakterlerin frekansları (kullanım sıklıkları) hesaplanmalıdır.

Örneğin, elimizdeki veri "**BAACC**" olsun,
B: 1
A: 2
C: 2

En küçük iki frekans toplanır ve frekans tablosu yeniden düzenlenir,

Tek bir ağaç oluşturulana kadar sürekli en küçük frekanslar toplanır,



Şekil 7.15. .Huffman kodlama algoritması örneği

Ağaçların graf teorisinde önemli olmasının bir nedeni tüm bağlı grafların bir ağaç içermesindendir. Buna spanning tree denir ve bütün düğümleri bağlar.

Tanım: G, düğüm kümesi V olan bir bağlı graf olsun. G 'deki bir spanning tree yine ağaç olan bir alt graftır ve düğüm kümesi V 'dir.

Teorem 7.10: Bütün bağlı graflar bir spanning tree içerir.

İspat: G bağlı bir graf olmak üzere ; G bir devre içermiyorsa G' nın kendisi bir spanning tree olduğundan kanıtlanacak bir şey yoktur.

G, bir devre içeriyor diyelim. Devreden bir ayrıt çıkarırsak elimizde hala bağlı bir graf kalır. Eğer yeni graf bir devre içeriyorsa devreden tekrar bir ayrıt çıkarırız. Bu işlemi sonuç grafı T bir devre içermeyinceye kadar devam ettiririz. Hiçbir düğümü kaldırmadığımıza göre T, G ile aynı düğüm kümesine sahip olacaktır ve yukarıdaki işlemin her aşamasında bağlı bir graf elde ederiz. Bu nedenle, T 'nin kendisi bağlıdır; G için bir spanning tree' dir.

Düzlem Graflar

Tanım: Düğümleri düzlemde noktalar ve ayrıtları sadece grafın düğümlerinde kesişen doğrular veya yaylar olan grafa düzlem grafı denir.

Bir graf, eğer bir düzlem grafıyla izomorfik ise örneğin düzlemde hiçbir ayrıt'ı kesişmeden bir diyagram ile temsil edilebiliyorsa düzlemsel (planar) graftır.

Euler' in Formülü

G bağlı düzlemsel bir graf olsun. Düzlemde çizilen G 'nın diyagramı 'yüz' (face) adını verdiğimiz bölgelere ayırır.

Bağlı düzlemsel bir grafın düğümlerinin, ayrıtlarının ve yüzlerinin sayısı arasında bir ilişki kurmak için basit bir formül vardır. Aşağıdaki tablo bu formülü görmek için faydalı olabilir.

Graf	*Düğüm Sayısı*	*Ayrıt Sayısı*	*Yüz Sayısı*
Şekil 7.2 (a)	7	7	2
Şekil 7.3 (a)	9	14	7
Şekil 7.4 (b)	4	7	5
Şekil 7.11 (b)	4	9	7
Herhangi ağaç	n	n-1	1

Bütün bu graflar bağlıdır ve düzlemseldir ve $|F|$, $|E|$, $|V|$ sırasıyla yüzlerin, ayrıtların ve düğümlerin sayısı olmak üzere

$$|F| = |E| - |V| + 2$$

ilişkisini sağlarlar. Bu ilişki tüm bağlı düzlemsel graflar için sağlanır ve Euler' in formülü olarak bilinir.

Teorem 7.11: G, $|V|$ düğümlü, $|E|$ ayrıtlı ve düzlemi $|F|$ yüze veya bölgeye ayıran herhangi bir bağlı düzlemsel graf olsun. Bu durumda, $|F|=|E| -|V| + 2$ olur.

İspat: G' nın ayrıt sayısına tümevarım yöntemi uygulayarak ispat yapılabilir. $|E|=0$ ise $|V|=1$ (G bağlıdır o halde iki veya daha fazla düğüm olamaz) ve tek bir yüz vardır yani $|F|=1$. Bu nedenle bu durum için teorem doğrudur.

Şimdi, teoremin n ayrıtdan az graflar için de sağlandığını düşünelim. G, n ayrıtlı bağlı düzlemsel graf olsun; yani |E|=n. G bir ağaç ise |V|=n+1 (teorem 7.10) ve |F|=1 o halde, teorem bu durumda da sağlanır. Eğer G bir ağaç değilse G‘deki herhangi bir devreyi seç ve bir ayrıt'ını sil. Sonuçtaki graf G' bağlıdır, düzlemseldir, n-1 ayrıt' i, |V| düğümü, ve |F|-1 yüzü vardır. Tümevarımsal hipoteze dayanarak Euler' in formülü G' için sağlanır.

$$|F|-1 = (|E|-1) - |V| + 2 \quad \text{o halde,}$$

$$|F| = |E| - |V| + 2.$$

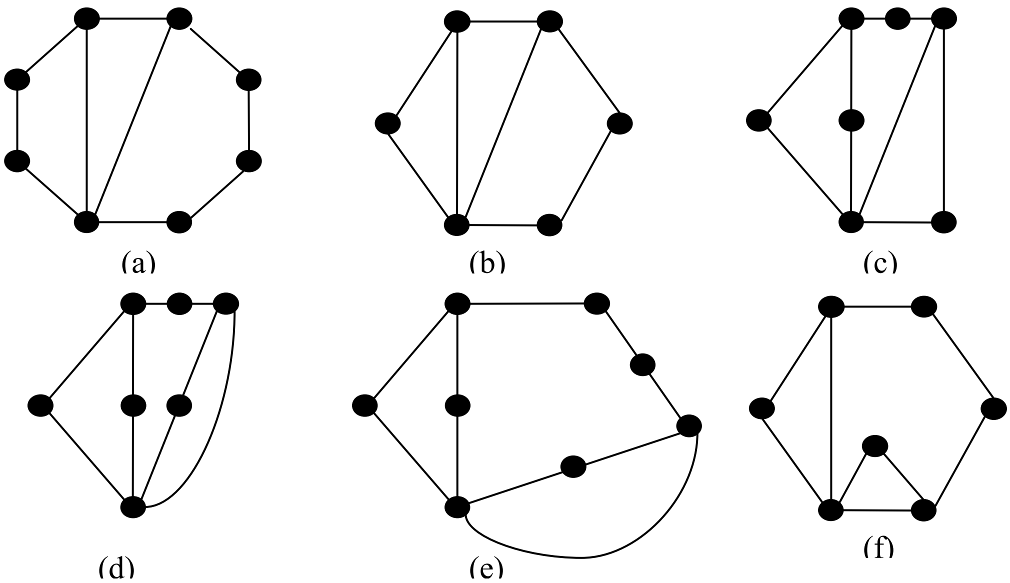
Teorem 7.12: n düğümlü($n \geq 3$) basit düzlemsel bağlı bir grafta, en fazla (3n-6) ayrıt vardır.

İspat : Eğer n=3 ise ayrıt sayısı en fazla 3'tür. n 3'e eşit veya daha büyük olsun. Düzlemsel grafı $F_1,F_2,\ldots,F_p$ olarak çizelim. F_i ile tanımlanan yüzün ayrıt sayısı r_i olsun. Her bir i için r_i en az üç olur. Böylece $3p \leq (r_1 + r_2 + \ldots + r_p)$ dir. Şimdi, sınırlardaki ayrıtları sayarsak her bir ayrıt ençok iki kere sayılır. Böylece eşitsizliğin sağ tarafı en fazla 2m olur. Burada m garaftaki ayrıt sayısıdır. Böylece, 3p en çok 2m dir. Fakat teorem 7.11'den p(|F|)= 2-n(|V|) +m(|E|) dir. Bu teorem bazı meşhur grafların düzlemsel olmadığını göstermek için kullanılır.

Kuratowski 'nin Teoremi

Tanım: Eğer bir graf, diğer bir grafın ayrıtlarına derecesi 2 olan düğümler ekleyerek veya çıkararak elde edilebiliyorsa bu iki graf homomorfiktir (izomorfik kopyasıdır).

Örnek 7.4: Şekil 7.16 'da gösterilen grafların hepsi homomorfiktir. (a)' daki graftan (b)' dekini elde etmek için 2 düğüm sileriz ve (b)' dekinden (c)' deki grafı elde etmek için bir düğüm sileriz ve iki düğüm ekleriz. (d)' den (e)' yi elde etmek için bir düğüm ekleriz. (e) ve (f)' deki graflar izomorfiktir- herhangi bir düğümün eklenmesine veya çıkarılmasına gerek yoktur.



Şekil 7.16

Teorem 7.13: Bir düzlemsel grafın kromatik sayısı dördü geçemez.

Alıştırmalar

1- Düğüm kümesi {1,2,3,4,5} ve ayrıt kümesi E={{1,2},{1,3},{1,5},{2,3},{3,4},{3,5}, {4,5}} olan grafı çiziniz. Bu grafın komşuluk matrisini bulunuz.

2- Γ ve Σ graflarının düğüm kümeleri ve komşuluk matrisleri sırasıyla şöyledir: $V_\Gamma = \{v_1, v_2, v_3, v_4, v_5\}$ ve $V_\Sigma = \{v_1, v_2, v_4, v_5\}$

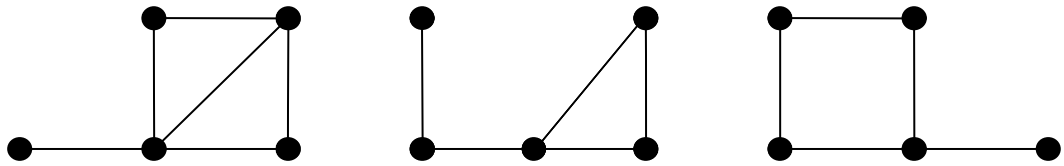
ve

Γ ve Σ graflarını çiziniz ve bu iki graf arasında nasıl bir ilişki vardır açıklayınız.

3- a) Hamilton grafı olan fakat Euler grafı olmayan dört düğümlü bağlı bir graf çiziniz.

b) Ne Hamilton ne de Euler grafı olmayan dört vertexli bağlı bir graf çiziniz.

4- Aşağıdaki 3 şekilden herhangi ikisinin izomorfik olmadığını ve hangi iki tanesinin homomorfik olduğunu gösteriniz.



(a) (b) (c)

9 Yineleme (Recurrence) Bağıntıları

$a_0,a_1,a_2,a_3,\ldots$ şeklindeki sekansları ele aldığımızda, a_r, belirli kombinasyonel problemlerde r girişine bağlı olan çözümdür. Bazı durumlarda a_r, sekansın önceki elemanlarına bağlı olarak ifade edilebilir. Örneğin, 4,7,10,13,16,…. Şeklindeki, dizide, $a_0=4$ ve ortak fark 3 'tür. Dolayısı ile, sıranın r. terimi a_r kendinden önceki (r-1) terime bağlı olarak $a_r= a_{r-1}+3$ şeklinde ifade edilebilir. Bu şekilde ifade edilen bağıntılara yineleme(recurrence) bağıntıları denir. $a_0=4$ ise başlangıç koşuludur. Başlangıç koşulu esas alınarak herhangi bir terim ardışık olarak hesaplanabilir. Diğer bir yol ise yineleme bağıntısını çözerek r. terimin bulunmasıdır. Bu örnekte $a_r=4+3r$ olarak bulunur. Diğer terimler bu çözümden hesaplanabilir.

Yineleme bağıntıları, fark(difference) ve diferansiyel denklemleri

Gerçel sayılardan oluşan bir $\{a_n\}$ dizisinde, ilk fark, $d\{a_n\}$, a_n-a_{n-1}, ikinci fark $d^2\{a_n\}$ ise, $d\{a_n\}-d\{a_{n-1}\}$ dir bu ise $a_n-2a_{n-1}+a_{n-2}$ dir Daha genel olarak, k. fark $d^k\{a_n\}$, $d^{k-1}\{a_n\}-d^{k-1}\{a_{n-1}\}$ dir. Bir fark denklemi a_n ve onun farklarını içeren denklemdir. Örnek, $3d^2(a_n) +2d(a_n)+7a_n$, ikinci dereceden homojen bir fark denklemidir. Herbir $a_i(i=0,1,2,\ldots,n-1)$ a_n'in terimleriyle ifade edilebilir çünkü $a_{n-1}=a_n-d(a_n)$, $a_{n-2}=a_{n-1}-d(a_{n-1})$ …. Olduğundan herbir yineleme bağıntısı bir fark denklemi olarak ifade edilebilir.

Örnek olarak, $3d^2(a_n) +2d(a_n)+7a_n$ fark denklemi, $12a_n=8a_{n-1}-3a_{n-2}$ şeklindeki yineleme bağıntısı olarak ifade edilebilir. Böylece bazı yazarlar fark denklemleri ve yineleme bağıntılarını değiştirerek kullanırlar. Yineleme bağıntılarının çözümünde, fark denklemlerinin çözüm yöntemleri kullanılır. Bu yöntemler ise, diferansiyel denklem sistemlerinin çözüm yöntemlerine benzerdir. Gerçekte, fark(Difererans) denklemleri diferansiyel denklemlerin sayısal ortamdaki ifade edilmesi şeklindedir. Bu noktada kısaca diferansiyel denklemleri hatırlamakta fayda vardır.

Diferansiyel denklemler, bilinmeyen y = y(x) fonksiyonunun türevlerini içeren bir eşitliktir. Bu eşitlikte türevlerle beraber y = y(x) fonksiyonunun kendisi x in bilinen fonksiyonları ve sabitler de bulunabilir. Türevler denildiğinde I. mertebeden, II. mertebeden,…. türevler kastediliyorlar. Denklemdeki en yüksek mertebeden türevin mertebesine **diferansiyel denklemin mertebesi** denir. Örneğin,
$y' = \sin x$, $y' - y = 0$, $xy' + x2y = 3$ denklemleri I. mertebeden,
$y'' + 4y = 0$, $y'' + 3y' + 5y = 0$ denklemleri ise II. mertebeden denklemlerdir.
Not: Yukarıdaki denklemlerde y, y', y'' fonksiyonları x değişkeninin fonksiyonlarıdır.
Genellikle, denklem yazılımında y, y', y'', . . . altındaki x değişkeni yazılmıyor.

Örneğin, $y'(x) - y(x) = 0$ yerine kısaca $y' - y = 0$ yazılır.

Diferansiyel denklemlerin fark denklemleriyle olan ilişkisini açıklamak için ise, Hesap bilimlerinden bildiğiniz gibi, y' sürekli bir y(x) fonksiyonunun türevidir. x ayrık olduğu zaman $y'(x)= y(x+1)-y(x)$'dir. Bu, $d\{y\}$ fark operatörü ile aynıdır. Bu ifade, türev ile benzer olan "fark sekanslarını" oluşturur

Daha yüksek mertebeden türevler olduğu gibi daha yüksek mertebeden fark sekansları vardır. $y''(x) = y'(x+1)-y'(x)$ türevi, $y(x+2)-2y(x+1)+y(x)$'e genişletilebilir..

Örnek: $y'-y=0$ diferansiyel denklemini fark denklemi olarak ifade edersek;
$y'(x)= y(x+1)-y(x)$ ve $y=y(x)$ dir. Sonuçta;
$y(x+1)-y(x)-y(x)=0$
$y(x+1)- 2y(x)=0$ (benzer şekilde $a_{n+1}=2a_n$ dir.)

Örnek: n farklı elemanı bir satıra dizme yollarının sayısını (a_n) hesaplamak için gerekli ifadeyi yineleme bağıntısı olarak bulun.

Çözüm: Seçilen bir elemanı ilk konuma yerleştirmek için n adet yol vardır. Bir elemanı ilk konuma yerleştirdikten sonra, kalan n-1elemanı yerleştirme şekli a_{n-1} dir. Böylece yineleme bağıntısı $a_n=na_{n-1}$ olarak ifade edilir. (burada başlangıç koşulu $a_1=1$'dir.)

Yineleme bağıntıları diferansiyel denklemlerde olduğu gibi homejen ve homojen olmayan olarak iki grupta toplanır. Burada doğrusal ve sabit katsayılı yineleme bağıntılarının çözümü üzerinde durulacaktır.

Tanım: Eğer $c_i(i=1,2,\ldots..,r)$ sabitler ise, $a_n = c_1a_{n-1}+c_2a_{n-2} +\ldots..+c_ra_{n-r} +f(n)$ ye r. dereceden sabit katsayılı doğrusal yineleme bağıntısı denir. Eğer f(n)=0 ise yineleme bağıntısı homojen, değil ise homojen olmayan yineleme bağıntısı denir. Eğer g(n) , $a_n=g(n)$ (n=0,1,2,….) şeklinde olan bir fonksiyon ise, g(n) yineleme bağıntısının bir çözümüdür.

Homojen Yineleme Bağıntılarının Çözümü

Teorem :(Süper pozisyon prensibi): Eğer $g_i(n)$ (i=1,2,…,k),
$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ra_{n-r} + f_i(n)$ şeklindeki bir yineleme bağıntısının çözümleri ise;
$A_1g_1(n)+A_2g_2(n)+\ldots+ A_kg_k(n)$ şeklindeki k çözümün kombinasyonu;
$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ra_{n-r} + A_1f_1(n) + A_2f_2(n) +\ldots A_kf_k(n)$ şeklindeki bir yineleme bağıntısının çözümüdür. Burada, A_i(i=1,2,…k) gerçel sayılardır. Herhangi bir homojen yineleme bağıntısının çözümlerinin doğrusal kombinezonu, homojen yineleme bağıntısının yine bir çözümüdür.

İspat: $h(n)= A_1g_1(n) + A_2g_2(n) +\ldots A_kg_k(n)$ olsun.
$g_i(n)$, $a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ra_{n-r} + f_i(n)$ 'in çözümü olduğu için;
$g_i(n)= c_1g_i(n-1) + c_2g_i(n-2) +\ldots c_rg_i(n-r) + f_i(n)$ yazılabilir.
Bu nedenle; $h(n)= c_1h(n-1) + c_2h(n-2) +\ldots c_rh(n-r) + A_1f_1(n) + A_2f_2(n)+\ldots+ + A_kf_k(n)$ iddiamızı ispatlar.

Sabit katsayılı homojen doğrusal yineleme bağıntılarını çözmek için basit yöntem vardır. Bu yöntem;
r bir sabit olmak üzere, $a_r=x^r$;$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ra_{n-r}$ 'nin bir çözümü kabul edilir ve kabul edilen çözüm bağıntıda yerine koyulursa;
$x^n = c_1x^{n-1} + c_2x^{n-2} + \ldots + c_rx^{n-r}$. elde edilir.
Bu denklemi r^{n-r} 'ye böler ve sağ tarafı sola geçirirsek;
$x^r - c_1x^{r-1} - c_2x^{r-2} - \ldots - c_{r-1}x – c_r = 0$ bulunur ve derecesi r olan ve genelde r adet kökü olan bu polinoma yineleme bağıntısının karakteristik denklemi denir. Bu denklemin kökü birden fazla veya karmaşık sayı olabilir.
Eğer x_i(i=1,2,…,r) karakteristik denklemin r adet kökü ise, $a_n=(x_i)^n$ homojen yineleme bağıntısının bir çözümüdür ve önceki önermede olduğu gibi böyle çözümlerin doğrusal kombinezonuda bağıntının bir çözümüdür.
Örnek olarak, $a_n=5a_{n-1} -6a_{n-2}$ bağıntısının karakteristik denklemi $x^2-5x+6=0$ dır ve kökleri $x_1=2$ ve $x_2=3$ dür. Böylece;
$A_n= A(2)^n +B(3)^n$ A ve B'sabitlerinin herhangi bir seçimi için yineleme bağıntısının bir çözümüdür. Diğer bir deyişle, herbir r kök, x_i (i=1,2,..,r) gerçel ve farklı ise, herbir genel çözüm bu $(x_i)^n$ çözümlerinin doğrusal bir kombinasyonudur.
Teorem: r. dereceden bir doğrusal homojen yineleme bağıntısının karakteristik denkleminin r kökü x_i (i=1,2,..,r) gerçel ve farklı ise, herbir genel çözüm bu $(x_i)^n$ çözümlerinin doğrusal bir

kombinasyonudur. Bununla birlikte, yineleme bağıntısının r ardışık başlangıç değeri $a_k, a_{k+1}, \ldots, a_{k+r-1}$ biliniyorsa, bu r adet başlangıç değerin çözüme uygulanmasıyla r keyfi sabit hesaplanır ve bu çözüm tektir.

Örnek: $a_n-9a_{n-2}=0$ bağıntısını, $a_0=6$, $a_1=12$ için çözün.

Çözüm: Karakteristik denklem $x^2-9= 0$ dır ve denklemin kökleri $x_1=3$ ve $x_2=-3$ dür. Buradan bağıntının genel çözümü; A ve B keyfi sabitler olmak üzere;
$a_n= A(3)^n +B(-3)^n$
Şimdi verilen başlangıç koşullarına bakarak A ve B sabitlerini bulalım.
$A+B=6$; $a_0=6$ için ve;
$3A-3B=12$; $a_1=12$ için
Bu iki denklemin çözümünden, A=5 ve B=1 bulunur. Buradan genel çözüm,
$a_n= 5(3)^n + (-3)^n$

Eğer karakteristik denklemin kökleri tekrarlanan çoklu kök ise bu durumda aşağıdaki örneği inceleyelim,

Örnek :
$a_n = 4a_{n-1}-4a_{n-2}$ yineleme bağıntısının karakteristik denklemi, $(x-2)^2=0$ dır ve denklemin kökleri $x_1=2$ ve $x_2=2$ dir. Bu durumda $A(2)^n$ bir çözümdür. Diğer çözüm ise elbette $Bn(2)^n$ şeklinde ve genel çözüm ise, $A(2)^n + Bn(2)^n$ şeklinde olacaktır. Yine benzer şekilde başlangıç koşullarının çözüme uygulanmasıyla A ve B sabitlerinin değeri hesaplanır.

Teorem: (a) Bir yineleme bağıntısının karakteristik denklemin bir çarpanı (t kök ve s de katlılık olmak üzere) $(x-t)^s$, olsun. Buradan,
$u= (t)^n(A_1+A_2n+A_3n^2+\ldots.+A_sn^{s-1})$ yineleme bağıntısının bir çözümüdür. Burada, $A_j(j=1,2,,\ldots,s)$ keyfi sabitlerdir. Bu çözüm, bağıntının r'ye göre temel çözümüdür.
(b) Yineleme bağıntısının kökleri t_k (k=1,2,…,q, burada s_k, t_k'nın katlılığıdır) ve u_k, bağıntının t_k köküne göre temel çözümü olsun. Buradan, yineleme bağıntısının her çözümü, bu q temel çözümün toplamıdır.

Örnek: Karakteristik denklemi $(x-2)^3(x+3)(x-4)^2$ şeklinde olan bir yineleme bağıntısının genel çözümünü bulun.

Çözüm : Denklemin kökleri 2,2,2,-3,4 ve 4 'dür. Tekrarlanan kök 2 için temel çözüm ;
$u_1= 2^n(A_1+A_2n+A_3n^2)$, kök -3 için temel çözüm, $u_2= A_4(-3)^n$ ve tekralanan kök 4 için temel çözüm $u_3= 4^n(A_5+ A_6n)$ dir. Böylece genel çözüm $u_1 + u_2 + u_3$ şeklindedir.

Homojen Olmayan Yineleme Bağıntıları.

Bu bölümde, $a_n = h_n +f(n)$ tipindeki doğrusal yineleme bağıntılarının çözümü üzerinde durulacaktır. Burada, $h_n= c_1a_{n-1}+c_2a_{n-2} +\ldots..+c_ra_{n-r}$,ve f(n) n'in bir fonksiyonudur. Verilen homojen olmayan bağıntının homojen parçası $a_n=h_n$ dir. Eğer verilen bağıntının homojen parçasının bir çözümü $a_n=u_n$ ve homojen olmayan bağıntının bir çözümü, $a_n=v_n$ ise süper pozisyon prensibi gereği, $a_n=u_n + v_n$ de aynı homojen olmayan bağıntının bir çözümüdür. Eğer, u_n 'in r keyfi sabiti var ise , $u_n + v_n$'in de r keyfi sabiti vardır. Eğer homojen olmayan bağıntının r ardışık başlangıç koşulu biliniyor ise, bu başlangıç koşulları, tek bir çözüm veren r değişkenli r denklem tanımlamak için kullanılır. Diğer bir deyimle, eğer homojen olmayan yinelemeli bağıntının homojen kısmının bir genel çözümü u_n ise ve eğer v_n de homojen olmayan bağıntının bir kısmi çözümü ise, $u_n + v_n$ aynı homojen olmayan bağıntının bir genel çözümüdür.

Örnek: $a_n=5a_{n-1}-6a_{n-2}+6(4)^n$ bağıntısının genel çözümün bulun.
Çözüm: Bağıntının homojen kısmının karakteristik denklemi $x^2-5x+6=(x-2)(x-3)$ dür. Buradan $x_1=2$,$x_2=3$ bulunur. Dolayısı ile homojen parçanın çözümü;
$u_n= A(2)^n +B(3)^n$ dir.
Kısmi çözüm için ise;
$f(n)= (4)^n$ olduğundan $vn=A.f(n)= A.4^n$ bir çözüm kabul edilsin ve bu çözüm bağıntıda yerine koyulursa;
$A(4)^n =5.A(4)^{n-1} -6.A(4)^{n-2} +6(4)^n$
Buradan A=48 ve $v_n=48(4)^n$ bulunur. Sonuçta; $a_n=u_n + v_n =A(2)^n +B(3)^n +48(4)^n$ bulunur. A ve B keyfi sabitleri, ardışık başlangıç koşulları kullanılarak bulunur.

Homojen yineleme bağıntısının çözümünün tersine, homojen olmayan bağıntıların kısmi çözümü için genel bir yöntem yoktur. Bununla birlikte iki özel durumda:

i. Eğer $f(n)= c(q)^n$ ise(burada c bilinen bir sabit) ve eğer q karakteristik denklemin kökü değil ise, $A(q)^n$ kısmi çözüm olarak seçilir. Burada A, homojen olmayan eşitlikte a_n yerine $A(q)^n$ koyularak hesaplanabilecek bir sabittir. Eğer q karakteristik denklemin k katlı bir kökü ise, bu durumda $A(n)^k(q)^n$ kısmi çözüm olarak seçilir
ii. Eğer, $f(n)= c(n)^k$ ise ve eğer karakteristik denklemin kökü 1 değil ise, kısmi çözüm için $A_0+A_1n +A_2n^2+\ldots+A_kn^k$ şeklinde derecesi k olan n'e bağlı bir polinom seçilir. Eğer 1, karakteristik denklemin t katlı kökü, ise, kısmi çözüm için $A_0n^t + A_1n^{t+1} +A_2n^{t+2}+\ldots+A_kn^{t+k}$ şeklindeki polinom seçilir.

Örnek: Homojen olmayan bir yineleme bağıntısının karakteristik denklemi, $(x-1)^2(x-2)((x-3)^2=0$ dır. Aşağıdaki f(n) değerlerine göre kısmi çözümleri bulun.
(a) $f(n)=4n^3 +5n$
(b) $f(n)=4^n$
(c) $f(n)=3^n$

Çözüm: Karakteristik denklemin kökleri, iki katlı kök 1, bir katlı kök 2 ve iki katlı kök 3 dür. Homojen parçanın genel çözümü u_n ve v_n de kısmi çözümler olmak üzere;
$u_n= c_1 + c_2.n + c_3.2^n +c_4.3^n + c_5.n.3^n$ dir.

(a) $v_n = An^2 +Bn^3 + Cn^4 + Dn^5$
(b) $v_n = A.4^n$
(c) $v_n = A.n^2\ 3^n$.

Yineleme Bağıntılarının iterasyon ile çözümü

Yineleme bağıntıları iterasyon yöntem ile de çözülebilir. Bu çözüm şekli için;

Örnek: $a_n=k.a_{n-1}+f(n)$ şeklindeki bir bağıntının çözümünü ele alalım.
Çözüm: Önce açıklandığı gibi, u_n homojen parçanın çözümü, v_n ise kısmi çözüm olmak üzere $a_n=u_n + v_n$ dir.
Durum(1): k=1, c keyfi bir sabit olmak üzere,u_n=c dir , böylece, $a_n=c + v_n$ dir. Burada v_n'in özelliği f(n)'e bağlı ve u_n de bir sabittir. Bununla birlikte;

$a_n=a_{n-1}+f(n)$
$a_{n-1}=a_{n-2}+f(n-1)$
.
.
$a_2=a_1+f(2)$
$a_1=a_0+f(1)$ dir. Bu n adet eşitlik toplanırsa;

$a_n=a_0+ f(1) + f(2) +…+ f(n)$ elde edilir.Böylece,
$f(1) + f(2) +…+ f(n)=a_n-a_0 =c+v_n-a_0$ elde edilir.
Durum(2): k, 1'e eşit değil ise, $u_n=c\, k^n$'dir. daha önce anlatıldığı gibi v_n, f(n) ve u_n 'e bağlıdır.

Örnek: $a_n= k.a_{n-1}+bn$ şeklindeki yineleme bağıntısının çözümünü iterasyon ile bulun.
$a_n= k.a_{n-1}+bn$
$k/\ a_{n-1}= k.a_{n-2}+b(n-1)$ (eşitlik k ile çarpılır)
$k^2/a_{n-2}= k.a_{n-2}+b(n-2)$ (eşitlik k^2 ile çarpılır)
.
.
$k^{n-2}/a_2= k.a_1+b(n-(n-2))$ (eşitlik k^{n-2}/ ile çarpılır)
$k^{n-1}/a_1= k.a_0+b(n-(n-1))$ (eşitlik k^{n-1}/ ile çarpılır)Bu eşitlikler toplanırsa;

$$a_n = a_0.k^n +bn + kb(n-1) + k^2b(n-2)+\ldots+ k^{n-1}b(n-(n-1))$$
$$= a_0.k^n +b[n(1+k +k^2+\ldots+k^{n-1})- k(1+2k +3k^2+\ldots+(n-1)k^{n-2})] \text{ dir.(2. seri 1.nin türevi)}$$
$$= a_0.k^n+b[n(\frac{k^n-1}{k-1})- k(\frac{n.k^{n-1}(k-1)+1-k^n}{(k-1)^2})] \text{ dir.}$$

$a_0= 1$, k=2 ve b=1 için çözüm $a_n= 3.2^n -2-n$ dir.
Not: Aynı problemi önceki yöntem ile çözünüz

9.1 Alıştırmalar:

1. n elemanlı bir kümenin tüm alt kümlerinin sayısını bulmak için gerekli yineleme bağıntısını tanımlayın.
2. Bir satırdaki n farklı elemanın dizilişinin sayısını veren yineleme bağıntısını tanımlayın.
3. Bir bankanın yıllık faiz oranının %r olduğunu kabul edelim.Eğen a_n , n yıl sonraki para miktarı ise, a_n için basit ve bileşik faize göre yineleme bağıntısını bulun.
4. f(n)= 2f(n-1) yineleme bağıntısını f(0)=1 için çözün.
5. f(n)=3f(n-1) + 4f(n-2) yineleme bağıntısını f(0)=1 f(1)=2 için çözün.
6. $F(n) = 4f(n-1) +5(3)^n$ yineleme bağıntısını f(0)= 1 için çözün.
7. F(n)=4f(n-1) -4f(n-2) +n yineleme bağıntısını f(0)=1 ve f(1)=2 için çözün

10 Algoritmalar ve Sonlu Durumlu Makinalar

10.1 Algoritmalar ve Karmaşıklık

Ayrık matematikte karşılaşılan birçok problem sınıfı mevcuttur. Örneğin verilen tamsayı grubu içindeki en büyük olanının bulunması, verilen bir kümenin bütün alt kümelerinin listelenmesi, verilen bir tamsayı kümesinin artan sırada sıraya dizilmesi, verilen bir ağ'da iki kenar arasındaki en kısa yol'un bulunması gibi. Böyle problemler ile karşılaşıldığında, ilk olarak problemin matematiksel yapısını içeren bir modelinin bulunması gerekir. Böyle modellerde, permutasyonlar, bağıntılar, graflar, ağaçlar ve sonlu durumlu makinalar gibi ayrık yapılar sıkça kullanılırlar.

Uygun matematiksel modelin kurulması çözümün ilk adımıdır. Modeli kullanarak çözümü gerçekleştiren bir yöntem gerekli olacaktır. Cevabı bulmak için sıralı adımları takip eden bir yordam olacaktır. Böyle sıralı işlem adımlarına Algoritma denir.

*Tanım: **Algoritma;** bir hesaplamayı gerçekleştirmek veya bir problemi çözmek için kesin işlemlerin sonlu bir kümesine algoritma denir*

Algoritma kelimesi Arap matematikçisi olan ve 9. yüzyılda yaşamış olan Al-Khowarizmi 'nin isminden türetilmiştir.

Örnek: Sonlu sayıdaki tamsayılar kümesinin en büyük elemanının bulunması için bir algoritma kurulmaya çalışılırsa:

Problemin uygulaması çeşitli olabilir. Örneğin üniversite öğrencileri arasında derecesi en yüksek olanın belirlenmesi, bir spor organizasyonunda en yüksek dereceli olan sporcunun belirlenmesi gibi.

Problemin birçok çözümü olabilir. Bir çözüm aşağıda verilmiştir.

1. Sayılarin ilkini geçici en büyük olarak belirle.
2. Bir sonraki sayı ile geçici en büyük tamsayıyı karşılaştır. Eğer yeni sayı geçiçi enbüyükten büyük ise yeni sayıyı geçici enbüyük olarak belirle.
3. Eğer daha sayı var ise bir önceki adımı tekrarla.
4. Dizide başka eleman kalmamış ise dur. Bu noktada en büyük sayı geçici enbüyük olarak belirlenmiş olan sayı olacaktır.

Algoritma bir bilgisayar dili yardımı ile gösterilebilir. Ancak bunu anlamak çoğu zaman zor olur. Bunun yerine ortak olarak kullanılan ***pseudokod*** ile ifade edilir. Bu kod ingilizce olarak ifade edilen işlem adımlarını gösterir.

Psudokod temel bilgileri: Bu bölümde psudokod ile ilgili temel bilgiler(Algoritmaları anlatmak için kullanılan) kısa olarak verilecektir**.**
Ayıraçlar : begin ,end , ; dir.

```
Begin
   Program
End
```

Bildiriler : procedure, begin, integer,boolean, real, array, string
İşlemler

End.
Procedure’ların arasında arşiv fonksiyonlarını belirtmeye gerek yok.

Atama : =, :=
örnek b:=2 , c:=3, a:=b+c gibi
Psodokod aşağıdaki gibi blok yapısındadır.

```
begin
        real temp;
        temp:=a;
        a:=b;
        b:=temp;
end
```

Kontrol yapıları :

if p **then** s_1 **else** s_2; { eğer p önermesi doğru ise s_1 ‘i değil ise s_2’yi yap.}

```
if a>b then
        begin
        ...
        end
else
        begin
        .......
        end
```

```
for j :=1 to n do
        begin
        .....
        end {j nin değerini 1denbaşlatarak arttır. j=n oluncays kadar begin end bloğunu icra et}
```

while p **do** s {p önermesi doğru olduğu sürece s’i icra et. İşlem sonunda p’nin değişmesi gerekir}

```
örnek   j:=1;
        toplam:=j;
        while j <10 do
        begin
                j:=j+1;
                toplam:= toplam+j;
        end
```

do s **until** p {p önermesi sağlanıncaya kadar s işlemini yap. s işlemi p önermesine etkili olmalıdır}
örnek :

```
j:=1;
toplam:=j;
do
begin
        j:= j+1;
        toplam:= toplam+1;
end
until j = 10;
```

Örnek . Enbüyük tam sayıyı bulma algoritması

Procedure *enbuyuk*($a_1, a_2, ..., a_n$: integers)
enbuyuk := a_1
for *i* := 2 **to** *n*
    if *enbuyuk* < a_i **then** *enbuyuk* :=a_i
{işlem sonunda enbuyuk tamsayı bulunmuş olur}

Başka bir algoritma da bu tamsayıları büyükten küçüğe doğru azalan şekilde sıralayıp ilk elemenı enbuyuk olarak almak olabilir.

Algoritmalarda aşağıdaki özellikler bulunur ve bu özellikleri akıldan çıkartmamak gereklidir.

Giriş: Belirlenen veri kümesinden algoritma giriş değerleri alır.

Çıkış: Algoritma herbir giriş kümesinde çıkış değerleri üretir. Bu değerler problemin çözümüdür.

Açıklık : Algoritmanın adımları açık olarak tanımlanmalıdır.

Doğruluk: Algoritma herbir giriş kümesi içn doğru çıkış üretmelidir.

Sonluluk : Algoritma, herbir giriş kümesi için amaçlanan çıkışı, sonlu işlem adımı(büyük olabilir) sonunda üretmelidir

Verimlilik : Algoritmanın herbir adımı tam ve sonlu bir zaman diilimnde gerçeklenmelidir.

Genellik : Yordam formdaki her probleme uygulanabilecek şekilde genel olmalıdır.

Enbüyük tamsayıyı bulma algoritması bu açıdan değerlendirilirse;
Giriş : Sonlu sayıda tamsayı kümesi
Çıkış : kümedeki en büyük tamsayı
Açık olarak adımlar tanımlanmıştır, ve doğru sonuç üretir.
Algoritma sonlu işlem adımı kullanır(n ad)
Algoritma herbir adımda bir karşılaştırma işlemi yapar.(verimlilik)
Algoritma bu tür kümelerdeki enbüyük tamsayıyı bulacak şekilde geneldir.

Arama Algoritmaları:

Sıralı listedeki bir elemanın yerinin bulunması çok değişik olarak karşılaşılan bir problemdir. Örneğin sözlükten bir kelime aranması gibi problemlere arama problemleri denir.

Genel arama problemi aşağıdaki şekilde açıklanabilir.: Farklı elemanları $a_1, a_2, ..., a_n$ olan bir listede bir *x* elemanın yerinin öğrenilmesi veya listede olup olmadığının öğrenilmesi şeklinde olabilir. Bu arama probleminin çözümü, x elemanına eşit olan a_i elemanın yerinin bulunmasıdır. (eğer $x = a_i$ ise x i. Elemandır.)

Problemin çözümü için ilk algoritma doğrusal veya ardışıl aramadır. Doğrusal arama algoritması, x ve a_1'i karşılaştırarak işleme başlar. Eğer $x = a_1$ ise aranan eleman 1. elemandır. $x \neq a_1$ ise, x ile a_2 karşılaştırılır. Eğer $x = a_2$ ise çözüm a_2'nin konumudur. Eğer $x \neq a_2$ ise, x , a_3 ile karşılaştırılır.Bu işlem bir uyuşma bulununcaya kadar devam eder.Uyuşma olmadıkça işlem devam eder. Eğer bir uyuşma bulunamaz ise sonuç sıfır olarak elde edilir. Doğrusal arama algoritmasının pseudokod'u aşağıda verilmiştir.

Procedure dogrusalarama(x: integer, $a_1, a_2, ..., a_n$: farklı tamsayılar)
i := 1;
while($i \leq n$ and $x \neq a_i$)
    i := i+1;
if $i \leq n$ **then** konum := i
else konum := 0; {konum, x'e eşit olan terimin indisidir.Eğer x bulunamamış ise değeri sıfırdır}

Şimdi başka bir algoritma düşüneceğiz.

Bu algoritmada verilen veriler artan şekilde sıralanmıştır. Veriler tamsayı ise en küçükten en büyüğe doğru sıralanmış, eğer kelime iseler alfabetik olarak sıralı şekildedir. Böyle bir veri kümesinde bir eleman aranması için kullanılacak algoritma, ikili aramadır. İkili arama algoritmasının mantığı sıralı veri kümesi ortadan iki kümeye ayrılarak bulunması istenen veri bu alt kümelerden hangisinin içerisinde olabileceğine bakılır. Arama işlemi alt kümelerde tekrarlanarak bulunması gerekli olan veri bulunmaya çalışılır. Aşağıdaki örnek ikili aramayı gösterir.

Örnek: 1,2,3,5,6,7,8,10,12,13,15,16,18,19,20,22 sıralı dizisi içerisinde 19 tamsayısı aransın.

Dizide 16 eleman bulunduğundan 1,2,3,5,6,7,8,10 12,13,15,16,18,19,20,22 şeklinde 8'li iki alt kümeye ayrılır. 19 tamsayısı birinci alt kümenin enbüyük elemanı ile karşılaştırılır. 10 < 19 olduğundan aranan sayı ikinci alt kümededir. Bundan sonra ikinci alt küme 12,13,15,16 18,19,20,22 olmak üzere 4 elemanlı iki alt kümeye ayrılır.

10 < 19 olduğundan aranan tamsayı sağ alt kümede olabilecektir. Bu nedenle sağ alt küme yine 18,19 ve 20,22 olmak üzere iki elemanlı iki alt kümeye ayrılır.

Şimdi 19 tamsayısı, son ikili kümenin en büyük elemanından büyük olmadığı için arama ilk kümenin 13. ve 14. elemanını içeren kümeyle sınırlanır. Böylece son kümede 18 ve 19 tamsayılı ve birer elemanlı iki alt kümeye ayrılır. 18 < 19 olduğundan arama 19 tamsayısından oluşan son kümeye sınırlanır. ve kümenin 14. elemanı olarak bulunur.

Algoritmanın pseudokod'u aşağıda verilmiştir.

Procedure ikiliarama(x: integer, $a_1, a_2,............, a_n$: artan tamsayılar)
i := 1; { i ,arama aralığının sol bitiş noktasını gösterir}
j := n; { j ,arama aralığının sağ bitiş noktasını gösterir}

while i < j **do**
begin
    m := [(i+j)/2];
    if x > a_m **then** i:= m+1;
     else j := m;
end
if x = a_i **then** konum := i
else konum := 0;
{ konum, x'e eşit olan terimin indisidir. Veya eğer x bulunamamış ise değeri sıfırdır}

Algoritmaların karmaşıklığı

Algoritmaların özellikleri içerisinde verimlilik olması gerektiği açıklanmıştı. Algoritmanın verimliliği ne demektir? Bunun analizi nasıl yapılır? Verimliliğin bir ölçütü, algoritmanın belirli bir giriş verisine karşın, problemin çözümü için bilgisayarın harcadığı zamanın ölçülmesidir. Diğer bir ölçü ise belirli giriş verisine karşı bilgisayarın kullandığı bellek miktarıdır. Böyle sorular algoritmanın bir hesaplama karmaşıklığının geliştirilmesini gerektirir. Problemi çözmek için algoritmanın harcadığı zamanın analizi zaman karmaşıklığı'nı, gerekli belleğin analizi ise yer(space) karmaşıklığının hesabını gerektirir.

Yer karmaşıklığı probleminin çözümü, algoritmayı gerçeklerken kullanılan veri yapıları ile bağlantılıdır. Ancak bu konular içerisinde yer karmaşıklığından bahsedilmeyecektir.

Algoritmanın zaman karmaşıklığı ise, belirli miktardaki giriş verisine karşılık, yapılan karşılaştırma, tamsayı toplama, tamsayı çıkartma, tamsayı çarpma ve bölme işlemleri ile diğer basit işlemlerin sayısı olarak hesaplanır.

Örnek(Zaman karmaşıklığı için) : Bir A dizisinin en küçük elemanını bulan algoritmanın zaman karmaşıklığının hesabı:

Procedure *enkucuk*(*A real array,*, *enkucuk*:*real*)
enkucuk := A[1];
for *i* := 2 **to** *n*
    begin
      if A[i] < *enkucuk* **then** *enkucuk :*= A[i] { en kötü durumda ***n-1*** defa icra edilir.}
    end
{işlem sonunda enküçük sayı bulunmuş olur}

Bu algortimanın zaman karmaşıklığı en kötü durumda dizinin büyüklüğü mertebesindedir. Bu yordamın işlem sayısını hesaplamaya çalışalım. Mertebesi n-1dir.

Karşılaştırma işlemlerinin sayısı : n-1

Atama işlemlerinin sayısı: n-1 ; Algoritmanın karmaşıklığı(zaman) O(n) dir.

Karmaşıklığı ifade etmek için O(n) notasyonu kullanılır. O mertebe işareti , (n) in sonlu bir çarpanla çarpımından daha küçüktür.

İşlemlerin sayısı ≤ kn { **k** : sınırlı sabit ,**n** : problemin büyüklüğü olarak tanımlanır}

Örnekler,

En küçük sayıyı bulma algoritması : n-1 O(n)

Hem en küçük hemde en büyüğü bulma : n-1 + n-2 = 2n-3 :O(n)

Sıralama yapan algoritma(En küçükten büyüğe): (n-1) +(n-2) +(n-3) +...= :O(n^2)

Algoritma	Zaman Karmaşıklığı	Çözülebilen En büyük problem			Örnek algoritma
		1 sn.	1 dk.	1 saat	
A1	n	1000	$6x10^4$	$3.6x10^6$	En küçüğü bulma
A2	nlogn	140	4893	$2x10^5$	Sıralama(Quicksort)
A3	n^2	31	244	1897	Geleneksel sıralama
A4	n^3	10	39	153	Matris Çarpımı
A5	2^n	9	15	21	Torba doldurma problemi

Burada ilginç nokta, mertebe arttıkça çözebildiğimiz problem sayısı çok çabuk düşer.

Örnek: İkili Arama algoritmasının karmaşıklık hesabının yapılması:

Çözüm : Basitlik için a_1, a_2,..........., a_n listesinde n = 2^k eleman olduğunu varsayalım.(k > 0) Burada k =logn olacaktır.Eğer kümedeki elemanların sayısı 2'nin katı şeklinde değil ise, liste

2^{k+1} elemanlı daha büyük bir liste olacaktır(burada . 2^k <n< 2^{k+1} dir.)

Aranan sayı bulununcaya kadar, i ve j sayıları birbirine yaklaşır. İlk adımda liste 2^{k-1} e sınırlanır. İkinci adımda liste 2^{k-2} 'ye sınırlanır. En sonunda liste 2^1 = 2 elemanlı olarak kalır. Listede tek eleman kalınca karşılaştırma başka eleman olmadığını gösterir işlem biter. İkili arama algoritmasını icra etmek için toplam 2k+2= 2logn+2 karşılaştırma yapılır Buradan ikili arama algoritmasının karmaşıklığının en kötü durumda O(logn) olduğu söylenebilir. Diğer bir karmaşıklık analizi ortalama durum analizidir. En kötü durum analizinden daha karmaşık olan bu analiz doğrusal arama algoritmasının karmaşıklık hesabında kullanılmıştır.

10.2Sonlu Durumlu Makinalar ve Turing Makinaları

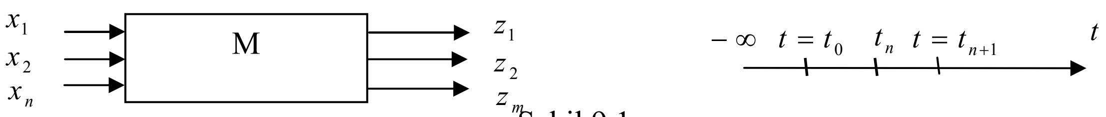
10.2.1 Sonlu durumlu Makina:

(a) Bir başlangıç durumu olan ve Sonlu sayıda duruma sahip {Q = $q_0,q_1,.......q_n$} olan,

(b) : Giriş{G=$x_1,x_2,......,x_n$}, ve Çıkış{Ç=$z_1,z_2,......,z_m$}, olmak üzere sonlu Alfabe(A) vardır.

(c) : Bu parametreler ile bir geçiş fonksiyonu tanımlanır: { QxG → ÇxQ }

Z(t+1) çıkışı temelde x(t)'ye bağlıdır Q(t)'ye , o andaki durum veya makinaların başından geçen olaylar(History) denir.



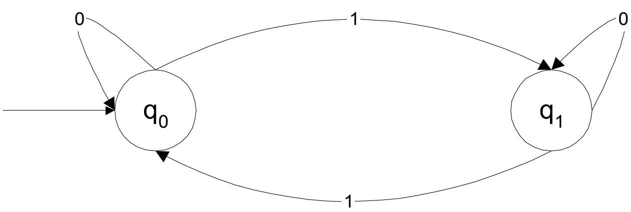
Şekil 9.1

10.2.2 Akseptör(Sonlu) : Bir sonlu akseptör aşağıdakilerden oluşur.

(a) Başlangıç durumu q_0, Son durumlar alt kümesi olmak üzere bir (sonlu) durum kümesi:
(b) : Bir A alfabesi(sonlu)
(c) : g : QxA → Q fonksiyonu
Örnek:
Örnek:

g/ç	q_0	q_1
0	q_0	q_1
1	q_1	q_0

Q ={ q_0, q_1 } ; A{0,1}

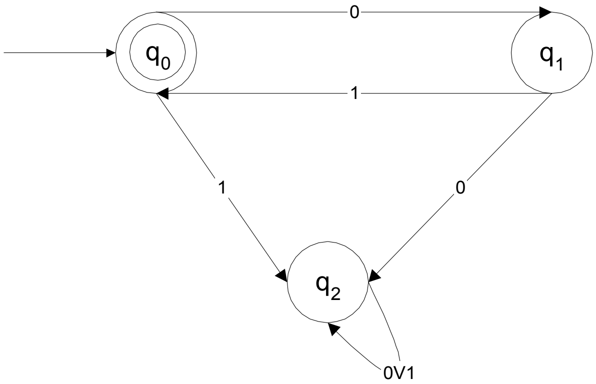


Şekil 9.2

Çıkışta bir işaret yok. q_1'i son durum alsak, tek sayıda 1 vererek bunu yine q_1'e getirmek mümkün. Bu akseptör tek sayıda bir bulunan bir katarı kabul eder.(10101110001'i kabul etmez, 6 ad .1 var)

Sonuç : Sonlu akseptör verilen bir katarın verilen bir gramere uygun olup olmadığını kontrol eder. Uygunluk son duruma erişip erişmeme ile anlaşılıyor.

Örnek : A{0,1} , Q{ q_0,q_1,q_2} , q_2 : dipsiz kuyu giren çıkamaz}

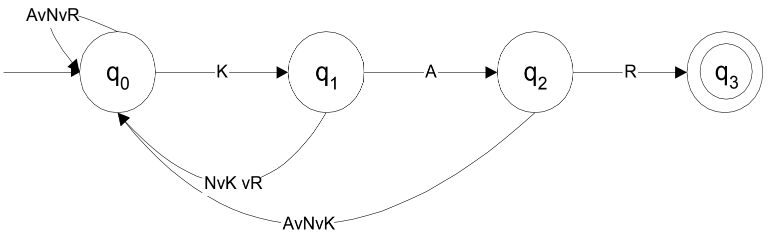


Şekil 9.3

Bu akseptör boş katar, 01,0101, 0101001 gibi katarları kabul eder. Türkçedeki bazı heceler sesli sessiz harflarden oluşur. Bunlar düzenlenebilir.

Örnek : a 0 ; at 01 ; yat 101 , dört 1011 gibi

Örnek : ANKARA'da KAR varmı? Akseptörü



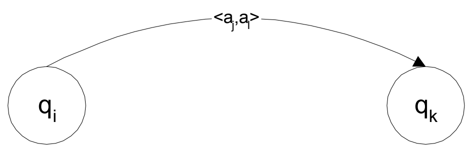
Şekil 9.4.

KAR bulduğu zaman son durumuna gelecektir. Buna Karakter uyuşturma{string matching} denilir. Uzun bir metnin içerisinde belirli bir harf dizisi varmı onu arıyoruz.

KAR Değilde başka bir şey aranırsa, mesela, aynı katar tekrarlanıyor, öyleki tekrardan sonra en bşa değil de daha ileri bir duruma geçilecek. Örnek KARAKAYA, Alt katarlar tekrarlanıyorsa akseptörü çizmek bir hayli zordur.

10.2.3 Sonlu Dönüştürücüler(Transduser) :

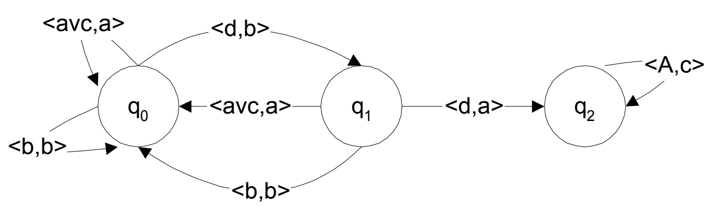
Bir Q kümesi (başı q_0)
Bir A alfabesi
g : $Q x A \rightarrow Q x A$
Bu makinalar yeni bir katarı alır, bundan yeni bir katar üretir
$q_i a_j \rightarrow q_k a_l$; $q_i \in Q$; $a_j \in A$



Şekil 9.5.

(q_i,a_j,q_k,a_l) dörtlüsü g fonksiyonunu tanımlar (q_i : durum , a_j , alfabe, q_k : donraki durum a_l: çıkış)

Örnek : A= {a,b,c,d} alfabesi üzerinde aşağıdaki dönüşüm işlemi yapılacaktır. Arka arkaya 2 d görülünceye kadar a ve b karakterleri aynı şekilde kopyalanacak, c'ler a'ya ; d'ler b'ye dönüştürülecektir. Arka arkaya gelen 2 d'den ikincisi a'ya dönüştürülecek. Daha sonra gelen karakterlerin yerine c koyulacaktır.



Şekil 9.6

Böylece dönüştürücünün durum geçiş diyagramı yukarıdaki gibi olacaktır. Bunlarda bellek yoktur. Eğer bellek eklenirse Turing makinaları elde edilir.

10.2.4 Turing Makinaları :

Bu makinalarda şerit şeklinde bellek vardır. Herbir bellek gözünde alfabenin sembollerinden biri olacaktır. Yine,

Bir Q kümesi (başı q_0)
Bir A alfabesi (b boşluk dahil)
g : $QxA \rightarrow QxA$ {R,L} kümesi bu şeridi okuyup kafanın sağamı, yoksa solamı hareket ettiğini belirtiyor.
(q_i,a_j,q_k,a_l,Y) ile tanımlanır

q_i : Makinanın durumu

a_j : kafanın şeritten okuduğu sembol

q_k : Makinanın yeni durumu

a_l : Kafanın şerite yazdığı yeni karakter

Y : R veya L olarak sağa yada sola doğru kafanın hareketi(bir göz hareket edecek). Böyle bir bellek özelliği olan ilkel makine turing makinası olarak bilinir. Turing makinası programı belirli bir işlemi yapan 5'lilerden oluşur

Örnek: m,n tamsayıları birli sistemde şerit üzerinde temsil edilmiştir. Sıfırı belirtmek için m tamsayısı m+1 adet 1 ile temsil edilmiştir

Birli sistemde 1 ve 4'ü'i temsil etmek için aşağıdaki gösterilim kullanılır.

b	1	1	b	1	1	1	1	1	b		

Yazacağımız program m ile n'i toplayacaktır

b	1	1	1	b 1	1	1	1	1	1	1	B		

Burada kafanın konumunun nerede olduğu önemlidir. Kafa en soldaki 1'in üzerindedir

q_i	s_j	q_k	s_l	Y
0	1	0	1	R
0	b	1	1	L
1	1	1	1	L
1	b	2	b	R
2	1	3	b	R
3	1	4	b	R

Bunu yukarıdaki programla toplayabiliriz.

b	1	1	b	1	1	1	1	1	b		

Turing makinaları ile başka işlemler de yapılabilir.

Örnek : A = {0,1} ,başta ve sonda boşluk bulunsun.

b	0	1	1	0	0	1	b		

Bu sayıyı tek yada çift pariteli yapmak için gerekli karakteri en sağına ilave eden program.

b	0	1	1	0	0	1	b		

1'leri sayıp tek ise sona 1 koyar, çift sayıda 1 varsa boşluğu sıfır yapar

	b	b	0	1	1	0	0	1	b	b	

Başka bir örnekte ikili sistemdeki sayının değerini birli sistemde yazmaktır. Mesela 15'i ikili sistemde okuyup 16 tane bir koymak olabilir.

10.3Alıştırmalar

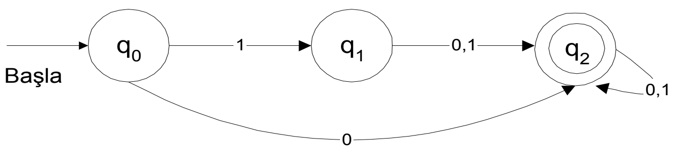
1. n uzunluğundaki bir listede bulunan sayılardan tamsayı(ondalık kısmı sıfır) olanların toplamını bulan bir algoritmayı psudokod ile yazınız.

2. Sadece atama deyimleri kullanarak x ve y değişkenlerinin değerlerini yer değiştiren algoritmayı psudokod ile yazınız.

3. Herbir işlemin 10^{-9} sn aldığı bir işlemde f(n) in aşağıdaki karmaşıklık değerleri olduğu algoritmalarda bir saniye içinde ne kadar büyüklükte problem çözülebileceğini hesaplayın.

   a)logn   b)n   c) n^2 d) 2^n

4. Aşağıdaki akseptör'ün hangi dizileri kabul ettiğini bulun..



5. {1,01,11} dizilerini kabul eden bir akseptör çizin.

Kaynaklar

1. Rowan Garnier, John Taylor, "Discrete mathematics for new technology ", Adam Hilger Publishing, 1992.
2. Sait Akkas, "Soyut matematik ", Gazi Üniversitesi, 1984.
3. Kenneth H .Rosen, éDiscrete Mathematics and its Applications", Mc Graw Hill ,1999.