

KANITLAR

KESİKLİ MATEMATİKSEL YAPILAR

2012-13

Kavramlar

- **Aksiyomlar**, daima doğru kabul edilen önermedirler.
- **Tanımlamalar**, var olanlardan yeni kavramlar türetmek için kullanılırlar.
- **Teorem**, doğruluğu gösterilmiş olan bir önermedir.
- **Lemma**, çok ilginç olmayan ve ama bir başka teorem için kullanılan bir teoremdir.
- **Sonuç**, bir başka teoremden kolayca elde edilebilen bir teoremdir.
- **Kanıt**, bir teoremin doğruluğunun gösterilmesidir
- **Mantık**, bir kanıtın analizi için kullanılan bir araçtır.

Kavramlar

- **Aksiyomlar:**

Verilen iki farklı noktadan, tek bir doğru geçer.

- **Tanımlamalar:**

Bir x gerçel sayısının mutlak değeri, eğer x pozitifse x olarak o 'sa o ya da negatifse $-x$ olarak tanımlanır.

- **Teorem:**

Bir üçgenin iki kenarı eşitse, bu durumda karşı kenarlardaki açılarda eşittir.

- **Lemma:**

Eğer n bir pozitif tamsayıysa, bu durumda $n-1$ 'de ya pozitif tamsayıdır ya da $n-1=0$ 'dır.

Teoremler

- Teoremler genellikle
“Her $x_1, x_2, \dots, x_n$ için
 $p(x_1, x_2, \dots, x_n)$
ise, bu durumda
 $q(x_1, x_2, \dots, x_n)$
‘dir’
biçimindedirler.

Kanıtlar

- P önermesi doğru kabul edilip Q önermesinin doğruluğu gösterilirse buna *dolaysız kanıt* denir. Yani bu

$$P \rightarrow Q$$

olduğunun gösterilmesidir

- P önermesini doğru ve Q önermesini yanlış kabul edip çelişki elde etmeye *çelişkiye vararak kanıt yöntemi* denir.
- Çelişki $r \wedge r^/$ biçiminde bir önermedir. Çelişkiye vararak kanıt yöntemine *dolaylı kanıt* da denir.

Karşıt Örnek

- $\forall xP(x)$
ifadesinin doğru olmadığını göstermek için, $P(x)$ önermesini yanlış yapan tanım kümesinden bir x ögesi bulmalıyız.
- Böyle bir x ögesine *karşıt örnek* denir.

Hipotezlere Göre Kanıt

- Eğer orjinal hipotez farklı hipotezlere bölümlenebiliyorsa buna *hipotezlere göre kanıt* denir.
- Eğer $p \rightarrow q$ gösterilecekse ve p deyimi $p_1 \vee p_2 \vee \dots \vee p_n$ deyimine denkse. Bu durumda

$$(p_1 \vee p_2 \vee \dots \vee p_n) \rightarrow q$$

yerine

$$(p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge \dots \wedge (p_n \rightarrow q)$$

olduğu gösterilir.

Denkliğe Göre Kanıt

- Bazı teoremler

“p gerekli ve yeterli koşul q”
şeklindedir.

$$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

olduğundan

“eğer p ise, q”

ve

“eğer q ise, p”

ifadelerini göstermek yeterlidir.

Olmayana Ergi Yoluyla Kanıt

- Kabul edelim ki, $p \rightarrow q$ ifadesinin kanıtlamak yerine ona dengi olan karşıt tersini alarak

$$q' \rightarrow p'$$

ifadesinin kanıtlayabiliriz. Buna *olmayana ergi yoluyla kanıt* (contrapositive) denir.

Varlık Kanıtı

- $\exists xP(x)$
‘ in kanıtlanmasına *varlık kanıtı* denir.

Kararlılık

- **Kararlılık** 1965 yılında J.A. Robinson tarafından önerilen bir kanıt tekniğidir. Tek bir kurala dayanmaktadır.
“eğer her iki $p \vee q$ ve $p \wedge r$ önermeleri doğruysa, bu durumda $q \vee r$ önermesi de doğrudur”
- Kararlılık ile kanıtta, hipotez ve sonuç bir **cümlecik** (clauses) olarak yazılır.
- Bir cümlecik bir değişken ya da değişkenin olumsuzunun “ya da” ‘lar ile bağlanmasından oluşur.

Kararlılık

● ÖRNEK 2.2.9:

“ Eğer ben çalışırsam ya da zekiysem, dersimden geçerim. Eğer ben dersimden geçerse, bir sonraki dersimi alabilirim. O halde ben bir sonraki dersimi alamazsam, ben zeki değilim ”

● Burada önermelerimizi şöyle alabiliriz:

s = “ben çalışırım”

g = “ben zekiyim”

p = “dersimden geçerim”

n = “bir sonraki dersi alırım”

Kararlılık

● ÖRNEK 2.2.10:

“Eğer ben çalışırsam ya da zekiysem, dersimden geçerim. Bir sonraki dersimi almaya gerek yoktur. Eğer dersimden geçerse, bir sonraki dersimi almalıyım. O halde çalışmama gerek yoktur.”

s = “ben çalışırım”

g = “ben zekiyim”

p = “dersimden geçerim”

n = “bir sonraki dersi alırım”

Tümevarım

- Kabul edelim ki, her bir n pozitif tamsayısı için bir $S(n)$ önermesi var olsun.

Ayrıca, yine kabul edelim ki,

- $S(1)$ doğru; (2.4)
- Her $i < n+1$ için $S(i)$ doğru olduğunda, $S(n+1)$ doğru olsun. (2.5)

Bu durumda her pozitif n tamsayısı için $S(n)$ doğrudur.

- Burada (2.4) 'e *temel adım* ve (2.5) 'e de *tümevarım adımı* denir.

Tümevarım

```
FUNCTION SQ (A)
1.    C ← 0
2.    D ← 0
3.    WHILE (D ≠ A)
      a.    C ← C+A
      b.    D ← D+1
4.    RETURN (C)
END OF FUNCTION SQ
```

Tümevarım

```
SUBROUTINE EXP (N, M; R)
1.    R ← 1
2.    WHILE (M > 0)
      a.    R ← R × N
      b.    M ← M - 1
3.    RETURN
END OF SUBROUTINE EXP
```

Tümevarım

```
FUNCTION GCD (X, Y)
1. WHILE (X≠Y)
    a. IF (X>Y) THEN
        i.  $X \leftarrow X - Y$ 
    b. ELSE
        i.  $Y \leftarrow Y - X$ 
2. RETURN (X)
END OF FUNCTION GCD
```