Kesikli Matematik

Mantık ve İspatlar

Dr. Öğr. Üyesi Mehmet Ali ALTUNCU
Bilgisayar Mühendisliği

Önerme Mantığının Sınırlamaları

- Önermeler mantığı matematik ve doğal dildeki bütün ifadelerin anlamını yeterince ifade edemez.

Örnek:

- “Bilgisayar Mühendisliği bölümündeki bir bilgisayar saldırı altında.” ifadesinden;
- “Üniversite ağında saldırı altında bir bilgisayar var.”

gerçeği sonucuna varmak için önermeler mantığının kurallarını kullanamayız.

ya da

- “Ali Bilgisayar Mühendisliği bölümü öğrencisidir.”

Ali : **Nesne**

Bilgisayar Mühendisliği bölümü öğrencisi: **Özellik**

- Nesneler ve özellikler ifadede gizlidir, bunlar hakkında akıl yürütmek mümkün değildir.

Önerme Mantığının Sınırlamaları

(1) Birçok nesne için tekrarlanması gereken ifadeler

Örnek:

- Ali Bilgisayar Mühendisliği mezunuysa, kesikli matematik dersini geçmiştir.

Çeviri:

- Ali Bilgisayar Mühendisliği mezunu ⟶ Ali kesikli matematik dersini geçmiştir.

Diğer Bilgisayar Mühendisliği mezunları için de benzer ifadeler yazılabilir:

- Ayşe Bilgisayar Mühendisliği mezunu ⟶ Ayşe kesikli matematik dersini geçmiştir.
- Mehmet Bilgisayar Mühendisliği mezunu ⟶ Mehmet kesikli matematik dersini geçmiştir.

• **Çözüm:** Değişkenlerle ifadeler oluşturulur.

– x Bilgisayar Mühendisliği mezunuysa, x kesikli matematik dersini geçmiştir.

– x Bilgisayar Mühendisliği mezunudur ⟶ x kesikli matematik dersini geçmiştir.

Önerme Mantığının Sınırlamaları

(2) Nesneler grubunun özelliğini tanımlayan ifadeler

Örnek:

- Tüm yeni arabalar kayıtlı olmalıdır.
- Bilgisayar Mühendisliği mezunlarından bazıları onur derecesiyle mezun oluyor.

● **Çözüm:** Niceleyicilerle ifadeler yapılır.

– **Evrensel niceleyici:** Özellik grubun tüm üyeleri tarafından karşılanır.

– **Varoluşsal niceleyici:** Grubun en az bir üyesi özelliği karşılar.

Yüklem Mantığı

Önerme mantığının sınırlamalarını giderir.

- Nesneleri ve özelliklerini açıkça modeller.
- Değişkenlerle ifadeler oluşturmaya ve bunları niceleştirmeye izin verir.

Yüklem mantığının temel yapı taşları:

- Sabit - Belirli bir nesneyi modeller (**Örnekler:** “John”, “Fransa”, “7”)
- Değişken - Belirli türdeki nesneyi temsil eder. (**Örnekler:** x, y)
- Yüklem - bir, iki veya daha fazla değişken veya sabit.
    - Nesneler arasındaki özellikleri veya ilişkileri temsil eder.
    - Örnekler: Kırmızı (araba23), öğrenci(x), evli(John,Ann)

Yüklemler

- Yüklemler, nesneler arasındaki özellikleri veya ilişkileri temsil eder.
- Bir P(x) yüklemi, özelliğin x için geçerli olup olmamasına bağlı olarak her x'e doğru veya yanlış bir değer atar.

Örnek:

- P(x), “x bir öğrencidir.” ifadesini belirtsin.
  - P(Ali) .... **T** (Ali bir öğrenciyse)
  - P(Ayşe) .... **T** (Ayşe bir öğrenciyse)
  - P(Mehmet) ..... **F** (Mehmet öğrenci değilse)

Yüklemler

“x bir asal sayıdır.” ifadesini temsil eden bir P(x) yüklemi varsayalım:

Aşağıdakilerin doğruluk değerleri nelerdir:

P(2) **T**

P(3) **T**

P(4) **F**

P(5) **T**

P(6) **F**

P(7) **T**

- P(2), P(3), P(4), P(5), P(6), P(7) önermedir.
- **P(x) bir önerme midir? Hayır (Birçok olası ikame mümkündür.)**

Yüklemler

- Yüklemler, nesneler arasındaki ilişkileri temsil eden daha fazla argümana sahip olabilir.

Örnek:

- **Daha yaşlı (John, Peter):** 'John, Peter'dan daha yaşlıdır.' anlamına gelir. **(Önermedir.)**
- **Daha yaşlı (x, y):** 'x, y'den daha yaşlıdır.' **(Önerme değildir.)**

Q(x,y), 'x+5 >y' yi ifade etsin.

- Q(x,y) bir önerme midir? **Hayır**
- Q(3,7) bir önerme midir? **Evet**
- Q(3,y) bir önerme midir? **Hayır**
- Q(3,7)'nin doğruluk değeri : **T**
- Q(1,6)'nin doğruluk değeri : **F**

Yüklem mantığında bileşik ifadeler

- Bileşik ifadeler mantıksal bağlaçlar aracılığıyla elde edilir

Örnekler:

- Öğrenci(Ali) $\land$ Öğrenci(Ayşe)
- **Çeviri:** "Hem Ali hem de Ayşe öğrencidir." **(Önerme)**

- Ülke(Sienna) $\lor$ Nehir(Sienna)
- **Çeviri:** “Sienna bir ülke veya bir nehirdir.” **(Önerme)**

- Bilgisayar Mühendisliği Bölümü (x) $\rightarrow$ Öğrenci(x)
- **Çeviri:** “x bir Mühendisliği Bölümü’nde ise, o zaman x bir öğrencidir.” **(Önerme değil)**

Yüklemler

- P(x) ifadesi, uygulanabileceği daha fazla nesne olduğundan bir önerme değildir.
- Bu, önermeler mantığında olduğu gibidir.

- **Ama fark şu:**
- Yüklem mantığı, nesne grupları hakkında ifadeler yapmamızı sağlar.
- Bunu yapmak için özel nicel ifadeler kullanılır.

Sayısal (Nicel) ifadeler

- İki tür nicel ifade vardır.

Evrensel Niceleyiciler

- **Örnek:** ' Tüm Bilgisayar Mühendisliği mezunları Ayrık Matematik dersini geçmek zorundadır.”
- Bu ifade tüm mezunlar için geçerlidir.

Varoluşsal Niceleyiciler

- **Örnek:** 'Bazı Bilgisayar Mühendisliği öğrencileri onur derecesiyle mezun oluyor.'
- Bu ifade bazı insanlar için doğrudur.

Evrensel Niceleyiciler

Tanım: P(x)'in evrensel nicelemesi şu önermedir:

- "P(x), tanım bölgesindeki tüm x değerleri için doğrudur." **∀x P(x)** gösterimi, P(x)'in evrensel niceliğini belirtir ve “her (tüm) x P(x) için” olduğu gibi ifade edilir.

Örnek:

- P(x) 'x > x - 1' i göstersin. ∀x P(x)'in doğruluk değeri nedir? (x'in tanım bölgesinin gerçek sayılar olduğunu varsayalım.)
- **Cevap:** Her x sayısı kendisinin 1 eksiğinden büyük olduğundan, ∀x P(x) doğrudur.
- Birçok olası ikame olduğundan **P(x) önerme değildir**. **∀x P(x)** ise tüm x değeri için doğru olduğundan **önermedir**.

Evrensel Niceleyiciler

- Bilgisayar Mühendisliği Bölümü (x) → Öğrenci(x)
- **Çeviri:** “x bir Mühendisliği Bölümü’nde ise, o zaman x bir öğrencidir.” **(Önerme değil)**

- **∀x** Bilgisayar Mühendisliği Bölümü (x) → Öğrenci(x)
- **Çeviri:** “(Bütün insanlar için öyledir) Eğer bir kişi Bilgisayar Mühendisliği Bölümü’nde ise, o zaman öğrencidir.” **(Önerme)**

Varoluşsal Niceleyiciler

Tanım: P(x)'in varoluşsal nicelemesi şu önermedir:

- "Tanım bölgesinde öyle bir öğe vardır ki, P(x) doğrudur" önermesidir. **∃ x P(x)** gösterimi, P(x)'in varoluşsal niceliğini belirtir ve “P(x) doğru olacak şekilde en az bir x vardır” gibi ifade edilir.

Örnek:

- T(x) 'x > 5' i göstersin. ∃ x T(x)'in doğruluk değeri nedir? (x'in tanım bölgesinin gerçek sayılar olduğunu varsayalım.)
- **Cevap:** x=10 için, 10>5 olduğundan **∃ x T(x) doğrudur**.

Varoluşsal Niceleyiciler

Örnek (devam):

- Q(x) 'x=x+2' yi göstersin. ∃x P(x)'in doğruluk değeri nedir? (x'in tanım bölgesinin gerçek sayılar olduğunu varsayalım.)
- **Cevap:** Hiçbir gerçek sayı kendisinden 2 büyük olmadığı için **∃x P(x) yanlıştır**.

- Kocaeli Üniversitesi-mezun (x) ∧ Onur-öğrenci(x)
- **Çeviri:** "x bir Kocaeli Üniversitesi mezunudur ve x bir onur öğrencisidir." **(Önerme değil)**

- **∃x** Kocaeli Üniversitesi-mezun (x) ∧ Onur-öğrenci(x)
- **Çeviri:** "Kocaeli Üniversitesi mezunu ve onur öğrencisi olan en az bir kişi vardır." **(Önerme)**

Niceleyiciler

Örnek (devam):

Niceleyici	Doğru Olduğunda	Yanlış Olduğunda
$\forall x \, P(x)$	P(x) her bir x için geçerlidir.	P(x)'in yanlış olduğu bir x vardır.
$\exists x \, P(x)$	P(x)'in doğru olduğu bir x vardır.	P(x) her bir x için yanlıştır.

- Tanım bölgesindeki elemanların x_1, x_2, ..., x_n olarak sıralanabileceğini varsayalım:
- $\forall x \, P(x)$, $P(x_1) \land P(x_2) \land \dots \land P(x_n)$ doğru olduğunda doğrudur.
- $\exists x \, P(x)$, $P(x_1) \lor P(x_2) \lor \dots \lor P(x_n)$ doğru olduğunda doğrudur.

Niceleyicilerle Çeviri

Cümle: “Tüm Kocaeli Üniversitesi öğrencileri akıllıdır.”

- **Varsayım:** x'in tanım alanı Kocaeli Üniversitesi öğrencileri olsun.
- **Çeviri:** $\forall x$ Akıllı (x)

- **Varsayım:** x'in tanım alanı tüm öğrenciler olsun.
- **Çeviri:** $\forall x$ (x, Kocaeli Üniversitesi) $\rightarrow$ Akıllı (x)

- **Varsayım:** x'in tanım alanı tüm insanlar olsun.
- **Çeviri:** $\forall x$ Öğrenci (x) $\land$ (x, Kocaeli Üniversitesi) $\rightarrow$ Akıllı (x)

Niceleyicilerle Çeviri

Cümle: “Kocaeli Üniversitesi’ndeki biri akıllıdır.”

- **Varsayım:** x'in tanım alanı Kocaeli Üniversitesi’nin tüm birimleri olsun.
- **Çeviri:** $\exists$ x Akıllı (x)

- **Varsayım:** x'in tanım alanı tüm insanlar olsun.
- **Çeviri:** $\exists$ x (x, Kocaeli Üniversitesi) $\land$ Akıllı (x)

Niceleyicilerle Çeviri

- **S(x) ve P(x)** olmak üzere iki yüklem varsayalım.

Evrensel niceleyiciler genellikle çıkarımlarla bağlanır.

- Tüm S(x), P(x) ise;
  - $\forall x\, (S(x) \longrightarrow P(x))$
- Hiçbir S(x), P(x) değilse;
  - $\forall x\, (S(x) \longrightarrow \neg P(x))$

Varoluşsal niceleyiciler genellikle genellikle bağlaçlarla bağlanır.

- Bazı S(x), P(x) ise;
  - $\exists x\, (S(x) \land P(x))$
- Bazı S(x), P(x) değilse;
  - $\exists x\, (S(x) \land \neg P(x))$

İç İçe Niceleyiciler

- Yüklem mantığındaki bir ifadenin anlamını yakalamak için birden fazla niceleyici gerekli olabilir.

Örnek:

- Her reel sayının kendisine karşılık gelen bir negatifi vardır.
- **Çeviri:**
  - **Varsayım:** Reel bir sayı x ve negatifi y ile gösterilsin.
  - Bir P(x,y) yüklemi şunu ifade eder: “x + y =0”
  - O zaman şunu yazabiliriz: **$\forall x \; \exists y \; P(x,y)$**

İç İçe Niceleyiciler

Örnek (devam):

- Bazıları herkesi sever.
- **Çeviri:**
  - **Varsayım:** x ve y insanları belirtsin.
  - L(x,y) yüklemi şunu ifade eder: x, y'yi sever.
  - O zaman şunu yazabiliriz: $\exists x \, \forall y \, L(x,y)$

İç İçe Niceleyiciler

- Niceleyiciler farklı türdeyse, iç içe niceleyicilerin sırası önemlidir.
- **$\forall x\ \exists y\ L(x,y)$ ile $\exists x\ \forall y\ L(x,y)$ aynı değildir.**

Örnek: L(x,y) yüklemi şunu ifade ettiğini varsayalım: “x, y'yi sever.”

- **$\forall x\ \exists y\ L(x,y)$:** Herkes birini sever.
- **$\exists x\ \forall y\ L(x,y)$:** Bazıları herkesi sever.

Matematiksel İndüksiyon (Tümevarım)

- Matematiksel tümevarım, tüm pozitif tamsayılar (n) için meşru bir ispat yöntemidir.

Kural: P_n pozitif bir tam sayı olan n'yi içeren bir ifade olsun.

- P_1 doğru ise,
- P_k gerçeği, her pozitif k için P_{k+1} gerçeğini sağlıyorsa,

o zaman tüm n pozitif tamsayıları için P_n doğru olmalıdır.

Matematiksel İndüksiyon

Örnek: $P_k : S_k = 3(2k+1) / (k-1)$ için P_{k+1} ‘i bulalım.

Çözüm: $P_{k+1} : S_{k+1} = 3(2(k+1)+1) / (k+1-1)$

$= 3(2k+2+1) / k$

$= 3(2k+3) / k$

Matematiksel İndüksiyon

Örnek: Her pozitif tamsayı için;

$S_n = 2 + 4 + 6 + 8 + \ldots + 2n = n(n + 1)$ ifadesini matematiksel indüksiyon ile ispatlayınız.

Çözüm:

1) **n=1 için;** $S_1 = 1(1 + 1) = 2$ **(T)**
2) Formülün bazı k tam sayıları için geçerli olduğunu varsayalım. Formülün bir sonraki tamsayı olan k + 1 için geçerli olduğunu kanıtlamak için bu varsayımı kullanıp, $S_{k+1} = (k + 1)(k + 2)$ formülünün doğru olduğunu gösterelim.

Matematiksel İndüksiyon

Örnek: Her pozitif tamsayı için;

$S_n = 2 + 4 + 6 + 8 + \ldots + 2n = n(n + 1)$ ifadesini matematiksel indüksiyon ile ispatlayalım.

Çözüm (devam):

$S_k = 2 + 4 + 6 + 8 + \ldots + 2k = k(k + 1)$ **(Varsayım)**

$S_{k+1} = 2 + 4 + 6 + 8 + \ldots + 2k + 2(k+1)$

$= S_k + (2k+2)$

$= k(k+1) + (2k+2)$

$= k^2+3k+2$

$= (k+1)(k+2)=(k+1)((k+1)+1)$ $\Rightarrow$ **$S_n = n(n + 1)$, n'nin tüm pozitif tamsayı değerleri için geçerlidir.**

Tam Sayıların Kuvvet Toplamları

1. $$\sum_{i=1}^{n} i = 1+2+3+4+\cdots+n = \frac{n(n+1)}{2}$$

2. $$\sum_{i=1}^{n} i^2 = 1^2+2^2+3^2+4^2+\cdots+n^2 = \frac{n(n+1)(2n+1)}{6}$$

3. $$\sum_{i=1}^{n} i^3 = 1^3+2^3+3^3+4^3+\cdots+n^3 = \frac{n^2(n+1)^2}{4}$$

4. $$\sum_{i=1}^{n} i^4 = 1^4+2^4+3^4+4^4+\cdots+n^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$$

5. $$\sum_{i=1}^{n} i^5 = 1^5+2^5+3^5+4^5+\cdots+n^5 = \frac{n^2(n+1)^2(2n^2+2n-1)}{12}$$

Tam Sayıların Kuvvet Toplamları

Örnek: $$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + 4^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

ifadesini tüm pozitif n tamsayıları için matematiksel indüksiyon ile ispatlayalım.

Çözüm:

$S_1 = 1(1+1)(2+1)/6 = 1$ **(T)**

$S_k = 1^2 + 2^2 + 3^2 + 4^2 + \ldots + k^2 = k(k+1)(2k+1)/6$ **(Varsayım)**

$S_{k+1} = 1^2 + 2^2 + 3^2 + 4^2 + \ldots + k^2 + (k+1)^2$

$= S_k + (k+1)^2$

$= S_k + (k^2+2k+1) = k(k+1)(2k+1)/6 + (k^2+2k+1)$

Tam Sayıların Kuvvet Toplamları

Örnek: $\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + 4^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$

ifadesini tüm pozitif n tamsayıları için matematiksel indüksiyon ile ispatlayalım.

Çözüm:

$S_{k+1} = (2k^3+3k^2+k) / 6 + (6k^2+12k+6) / 6$

$= (2k^3+9k^2+13k+6) / 6$

$= (k^2+3k+2)(2k+3) / 6$

$= (k+1)(k+2)(2k+3) / 6 = (k+1)[(k+1)+1][(2(k+1)+1] / 6$

- **Formül, n'nin tüm pozitif tamsayı değerleri için geçerlidir.**

Kaynaklar

- **Kenneth Rosen,** “Discrete Mathematics and Its Applications”, 7th Edition , McGraw Hill Publishing Co., 2012.
- **Milos Hauskrecht,** “Discrete Mathematics for Computer Science”, University of Pittsburgh, Ders Notları.