

# Yapay Zeka

Ders 12 - Bölüm 2

Doç. Dr. Mehmet Dinçer Erbaş  
Bolu Abant İzzet Baysal Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü

# Birinci-derece mantık ile çıkarım

- Önerimsel mantık ile çıkarım yöntemlerini hatırlayalım.
  - Modus ponens: Bir koşul cümlesinin önerim kısmı doğru ise sonuca erişebiliriz.

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

- Ve-Eleme: Birleşme ile bağlanmış önermelere erişebiliriz.

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n}{\alpha_i}$$

- Ve-Ekleme: Önermeleri birleştirilebiliriz.

$$\frac{\alpha_1, \alpha_2, \dots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n}$$

- Veya-Ekleme: Bir önermeden, bu önermenin başka önermelerle ayrılmasına erişebiliriz.

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \dots \vee \alpha_n}$$

# Birinci-derece mantık ile çıkarım

- Önerimsel mantık ile çıkarım yöntemlerini hatırlayalım
  - İkili-olumsuz eleme: iki kez negatif yapılmış bir cümleden, cümlenin pozitif halini elde edebiliriz

$$\frac{\neg\neg\alpha}{\alpha}$$

- Birim çözülme

$$\frac{\alpha \vee \beta, \quad \neg\beta}{\alpha}$$

- Çözülme

$$\frac{\alpha \vee \beta, \quad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

$$\frac{\neg\alpha \Rightarrow \beta, \quad \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

# Birinci-derece mantık ile çıkarım

- Taban terim (İng: ground term)
  - Herhangi bir değişken içermeyen terimdir.
  - Bir sabit simbol veya taban terime bir fonksiyon uygulanması ile oluşabilir.
- $\{X/a\}$ : değiştirme / bağlantı listesi
  - “X değişkenine a değeri veriliyor” anlamına gelir.

# Birinci-derece mantık ile çıkarım

- İspatlar
  - Geçerli çıkarım: BT  $\models \alpha$  olan  $\alpha$  bulma.
  - İspat işlemi bir aramadır.
    - Uygulanan işlemler çıkarım kurallarından oluşur.
  - Örnek: Modus Ponens:

$$\frac{\alpha, \quad \alpha \Rightarrow \beta}{\beta} \qquad \frac{At(Joe, UCB) \quad At(Joe, UCB) \Rightarrow OK(Joe)}{OK(Joe)}$$

- Örnek: Ve-Ekleme

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta} \qquad \frac{OK(Joe) \quad CSMajor(Joe)}{OK(Joe) \wedge CSMajor(Joe)}$$

# Birinci-derece mantık ile çıkarım

- Birinci-derece mantık için üç yeni çıkarım kuralı tanımlayacağız.
  - Evrensel eleme: Her  $\alpha$  cümlesi,  $x$  değişkeni ve  $\tau$  taban terimi için

$$\frac{\forall x \quad \alpha}{\alpha\{x/\tau\}}$$

- Örnek:  $\forall x$  Sever(x, Şeker) ve  $\{x/Ali\}$  kullanarak, Sever(Ali,Şeker) çıkarımı yapabiliriz.
- Varlık eleme: Her  $\alpha$  cümlesi, değişken  $x$  ve BT'de bulunmayan k sabit simbolü için

$$\frac{\exists x \quad \alpha}{\alpha\{x/k\}}$$

- Örnek:  $\exists x$  Öldür(x,Maktül) kullanarak, eğer "Katil" simbolü BT'de yok ise Öldür(Katil, Maktül) cümlesine ulaşabiliriz

# Birinci-derece mantık ile çıkarım

- Birinci-derece mantık için üç yeni çıkarım kuralı tanımlayacağız
  - Varlık ekleme: Her  $\alpha$  cümlesi,  $\alpha$  içinde geçmeyen  $x$  değişkeni ve  $\alpha$  içinde geçen  $g$  taban terimi için

$$\frac{\alpha}{\exists x \quad \alpha\{g/x\}}$$

- Örnek:  $\text{Sever}(\text{Ali}, \text{Şeker})$  cümlesinden  $\exists x \text{ Sever}(x, \text{Şeker})$  cümlesine ulabiliz.

# Birinci-derece mantık ile çıkarım

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>• Bob bir bufalodur</li><li>• Pat bir domuzdur</li><li>• Bufolalar domuzlardan hızlı koşar</li></ul> | <ol style="list-style-type: none"><li>1. Bufalo(Bob)</li><li>2. Domuz(Pat)</li><li>3. <math>\forall x,y \text{Bufalo}(x) \wedge \text{Domuz}(y) \rightarrow \text{Hızlı}(x,y)</math></li></ol> |
| <ul style="list-style-type: none"><li>• Bob Pat'ten hızlı koşar</li></ul>  |  |

# Birinci-derece mantık ile çıkarım

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>• Bob bir bufalodur</li><li>• Pat bir domuzdur</li><li>• Bufolalar domuzlardan hızlı koşar</li></ul> | <ol style="list-style-type: none"><li>1. Bufalo(Bob)</li><li>2. Domuz(Pat)</li><li>3. <math>\forall x,y \text{Bufalo}(x) \wedge \text{Domuz}(y) \rightarrow \text{Hızlı}(x,y)</math></li></ol> |
| <ul style="list-style-type: none"><li>• Bob Pat'ten hızlı koşar</li></ul>  |  |
| <ul style="list-style-type: none"><li>• Ve-Ekleme (1 ve 2)</li></ul>   | <ol style="list-style-type: none"><li>4. Bufalo(Bob) <math>\wedge</math> Domuz(Pat)</li></ol>  |

# Birinci-derece mantık ile çıkarım

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>• Bob bir bufalodur</li><li>• Pat bir domuzdur</li><li>• Bufolalar domuzlardan hızlı koşar</li></ul>         | <ol style="list-style-type: none"><li>1. Bufalo(Bob)</li><li>2. Domuz(Pat)</li><li>3. <math>\forall x,y \text{Bufalo}(x) \wedge \text{Domuz}(y) \rightarrow \text{Hızlı}(x,y)</math></li></ol> |
| <ul style="list-style-type: none"><li>• Bob Pat'ten hızlı koşar</li></ul>  |  |
| <ul style="list-style-type: none"><li>• Ve-Ekleme (1 ve 2)</li><li>• Evrensel-Eleme (3),<br/><math>\{x/\text{Bob}, y/\text{Pat}\}</math></li></ul> | <ol style="list-style-type: none"><li>4. Bufalo(Bob) <math>\wedge</math> Domuz(Pat)</li><li>5. Bufalo(Bob) <math>\wedge</math> Domuz(Pat) <math>\rightarrow</math> Hızlı(Bob,Pat)</li></ol>    |

# Birinci-derece mantık ile çıkarım

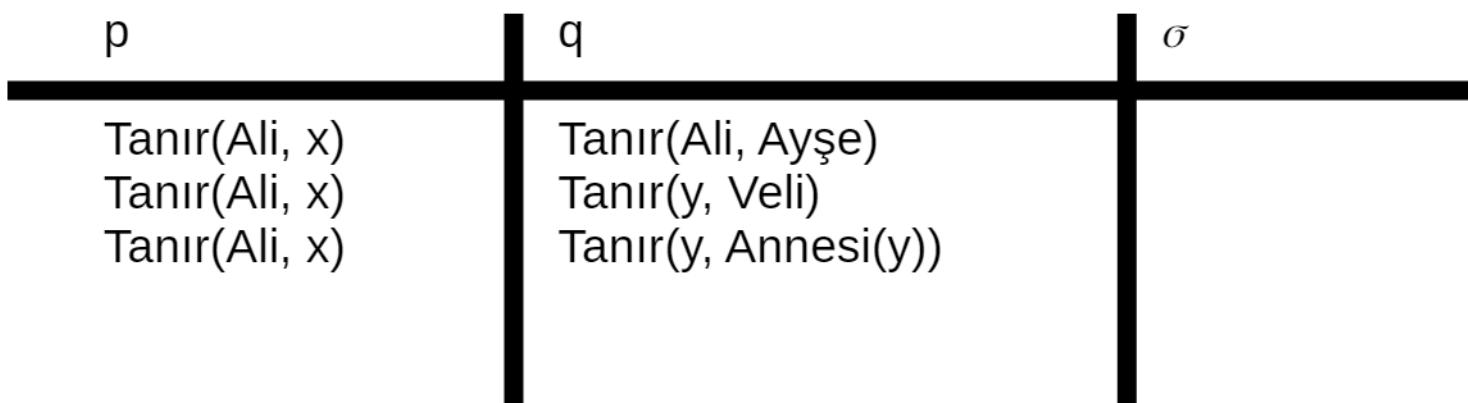
- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>• Bob bir bufalodur</li><li>• Pat bir domuzdur</li><li>• Bufolalar domuzlardan hızlı koşar</li></ul>   | <ul style="list-style-type: none"><li>1. Bufalo(Bob)</li><li>2. Domuz(Pat)</li><li>3. <math>\forall x,y \text{Bufalo}(x) \wedge \text{Domuz}(y) \rightarrow \text{Hızlı}(x,y)</math></li></ul>                        |
| <ul style="list-style-type: none"><li>• Bob Pat'ten hızlı koşar</li></ul>  |   |
| <ul style="list-style-type: none"><li>• Ve-Ekleme (1 ve 2)</li><li>• Evrensel-Eleme (3),<br/><math>\{x/\text{Bob}, y/\text{Pat}\}</math></li><li>• Modus Ponens (4 ve 5)</li></ul> | <ul style="list-style-type: none"><li>4. Bufalo(Bob) <math>\wedge</math> Domuz(Pat)</li><li>5. Bufalo(Bob) <math>\wedge</math> Domuz(Pat) <math>\rightarrow</math> Hızlı(Bob,Pat)</li><li>6. Hızlı(Bob,Pat)</li></ul> |

# Birinci-derece mantık ile çıkarım

- Önerimsel mantıktan gelen kuralları uygulamak arama yapmaya benzer.
  - Uyguladığımız eylemler belirttiğimiz çıkışım kurallarıdır.
  - Durumlar yeni oluşan cümlelerdir.
- Ancak bu şekilde yaptığımız arama oldukça verimsizdir.
  - Örneğin Evrensel eleme operasyonunu birçok taban terimine uygulayabiliriz.
  - Bu durum dallanma faktörünü büyütür.
- Bu sebeple farklı çıkışım yöntemleri geliştirilmiştir.

# Birinci-derece mantık ile çıkarım

- Birleştirim
  - Bir  $\sigma$  birleştirimi iki atomik cümle,  $p$  ve  $q$ , cümlelerine uygulandığında  $p_\sigma = q_\sigma$  olur.
  - Çıkarım kuralları farklı mantık ifadelerini aynı yapan birleştirimleri bulmalıdır.
    - Amacımız  $\sigma$  değişimini bulmak



# Birinci-derece mantık ile çıkarım

- Birleştirim
  - Bir  $\sigma$  birleştirimi iki atomik cümle,  $p$  ve  $q$ , cümlelerine uygulandığında  $p_\sigma = q_\sigma$  olur.
  - Çıkarım kuralları farklı mantık ifadelerini aynı yapan birleştirimleri bulmalıdır.
    - Amacımız  $\sigma$  değişimini bulmak

$p$	$q$	$\sigma$
Tanır(Ali, x)	Tanır(Ali, Ayşe)	{x/Ayşe}
Tanır(Ali, x)	Tanır(y, Veli)	{x/Veli, y/Ali}
Tanır(Ali, x)	Tanır(y, Annesi(y))	{y/Ali, x/Annesi(Ali)}

# Birinci-derece mantık ile çıkarım

- Birleştirim
  - Kuralın önkoşulları ile bilinen gerçekler birleştirilir, birleştirici çıkarıma uygulanır.
    - Örnek:  $q$  ile  $\text{Tanır}(\text{Ali}, x) \rightarrow \text{Sever}(\text{Ali}, x)$  biliniyorsa  
Şu sonuçlara varabiliriz       $\text{Sever}(\text{Ali}, \text{Ayşe})$   
     $\text{Sever}(\text{Ali}, \text{Veli})$   
     $\text{Sever}(\text{Ali}, \text{Annesi}(\text{Ali}))$

# Birinci-derece mantık ile çıkarım

- Birleştirim
  - Başka bir örnek

p	q	$\sigma$
Öğrenci(x)	Öğrenci(Ali)	{x/Ali}
Satar(Ali, x)	Satar(x, Saat)	{x/Saat, x/Ali} Bu değiştirme doğru mu?

# Birinci-derece mantık ile çıkarım

- Birleştirim
  - Başka bir örnek

p	q	$\sigma$
Öğrenci(x)	Öğrenci(Ali)	{x/Ali}
Satar(Ali, x)	Satar(y, Saat)	{x/Saat, y/Ali}

Bu işleme değişken standartlaştırma ismi verilir.

# Birinci-derece mantık ile çıkarım

- Birleştirim
  - $\text{birleştir}(P(a,X), P(a,b)) \quad \sigma = \{X/b\}$
  - $\text{birleştir}(P(a,X), P(Y,b)) \quad \sigma = \{X/b, Y/a\}$
  - $\text{birleştir}(P(a,X), P(Y,f(a))) \quad \sigma = \{X/f(a), Y/a\}$
  - $\text{birleştir}(P(a,X), P(X,b)) \quad \sigma = \text{başarısız}$
  - Not: Ancak  $P(a,X)$  ile  $P(X,b)$  birbirinden bağımsız ise,  $X$ 'lerden birini  $Y$  ile değiştirmek birleştirme yapılabilir.

# Birinci-derece mantık ile çıkarım

- Genelleşmiş Modus Ponens

$$\frac{p_1', \ p_2', \ \dots, \ p_n', \ (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{q\sigma} \quad p_i' \sigma = p_i \sigma \text{ her } i \text{ için}$$

- Örnek:  $p1' = \text{Hızlı(Ali, Ayşe)}$

$p2' = \text{Hızlı(Ayşe, Veli)}$

$p1 \wedge p2 \rightarrow q = \text{Hızlı}(x,y) \wedge \text{Hızlı}(y,z) \rightarrow \text{Hızlı}(x,z)$

$\sigma = \{x/\text{Ali}, y/\text{Ayşe}, z/\text{Veli}\}$

$q\sigma = \text{Hızlı(Ali, Veli)}$

- GMP örnekte belirli cümlelerle (tam bir tane olumlu terim) birlikte kullanılmıştır.

– Ön kısım ya bir atomik cümledir yada atomik cümlelerin birleşiminden oluşmuştur.

– Avraca her deðiskenenin evrensel niceleñindiði var savılmıştır.

# Birinci-derece mantık ile çıkarım

- Genelleşmiş Modus Ponens
  - Örnek: İnsan(Ali), Zengin(Ali), ( $\text{İnsan}(x) \wedge \text{Zengin}(x) \rightarrow \text{Popular}(x)$ )
  - Popular(Ali)
    - Değiştirme ile  $Q = \{x/\text{Ali}\}$
  - GMP mantıklıdır çünkü
    - Mمungkin olduğunda ileri doğru adımlar atar.
    - Mantıklı adımlar atar: Yapılan değiştirmeler yardımcı olur.  
( $\{x/\text{Ali}, y/\text{Ayşe}, z/\text{Veli}\}$ )
    - Ayrıca GMP önderleme aşaması içerir.
      - BT'deki cümleler horn cümlesiine çevrilir.
    - Hatırlatma: Horn tipi cümleler horn ifadelerinin (en fazla bir tane olumlu önerim) birleşmesinden oluşur.
      - Örnek:  $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D) \equiv (B \rightarrow A) \wedge ((C \wedge D) \rightarrow B)$

# Birinci-derece mantık ile çıkarım

# Birinci-derece mantık ile çıkışım

- İleri zincirleme (İng: forward chaining)
  - Yeni bir p gerçeği BT'ye eklendiğinde
    - p'nin önkoşul kısmı ile birleştiği her kural için
      - Eğer kuralın diğer önkoşulları biliniyor ise
      - Kuralın sonuç kısmı BT'ye eklenir ve zincir devam eder.
  - İleri zincirleme veri-odaklıdır.
    - Algılardan özellikler ve kategoriler elde edilir.
  - İleri zincirleme sonsuz döngülere girebilir.
    - Örnek: BT şu kuralı içersin,  $\forall x \text{ TamSayı}(x) \rightarrow \text{TamSayı}(+(x,1))$ , sonra da  $\text{TamSayı}(0)$  gerçeğini BT'ye ekleyelim.

# Birinci-derece mantık ile çıkarım

- İleri zincirleme: Örnek
  - 1, 2, 3, 4, 5, 6, 7 numaralı gerçekleri BT'ye ekleyelim.
  - [] içindeki değer birleşen terimi gösterir,  $\checkmark$  kural gerçekleşmesini gösterir.
    1. Bufalo(x)  $\wedge$  Domuz(y)  $\rightarrow$  Hızlı(x,y)
    2. Domuz(y)  $\wedge$  SümüklüBöcek(z)  $\rightarrow$  Hızlı(y,z)
    3. Hızlı(x,y)  $\wedge$  Hızlı(y,z)  $\rightarrow$  Hızlı(x,z)
    4. Bufalo(Bob) [1a, x]
    5. Domuz(Pat) [2a,x][1b, $\checkmark$ ]  
 $\rightarrow$  6. Hızlı(Bob,Pat) [3a,x][3b,x]
    7. SümüklüBöcek(Steve) [2b, $\checkmark$ ]  
 $\rightarrow$  8. Hızlı(Pat,Steve) [3a,x][3b, $\checkmark$ ]  
 $\rightarrow$  9. Hızlı(Bob,Steve) [3a,x][3b,x]

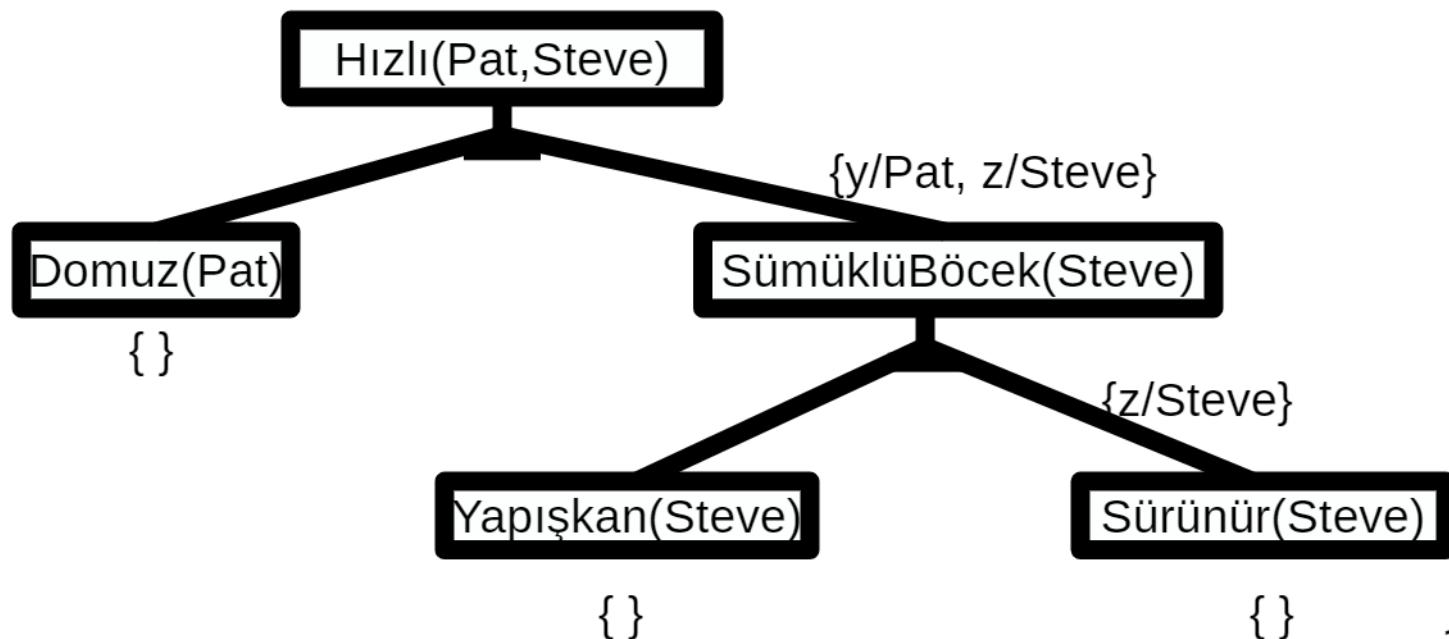
# Birinci-derece mantık ile çıkışım

- Geri zincirleme
  - Soru  $q$  sorulduğunda
    - Eğer  $q$  ile eşleşen bir  $q'$  gerçeği biliniyorsa, birleştirici döner.
    - Sonuç  $q'$  kısmı  $q$  ile eşleşen her kural için
      - İlgili kuralın her önkoşulunu geri zincirleme ile ispatlamayı dene
  - Birleştiricilerin hatırlanması gereklidir.
  - Ayrıca sonsuz döngüler kontrol edilmelidir.
  - İki farklı geri zincirleme versiyonu bulunmaktadır.
    - Her çözümü bulanlar ve herhangi bir çözümü bulanlar
  - Geri zincirleme mantık programlamanın temelini oluşturur.
    - Örnek: Prolog

# Birinci-derece mantık ile çıkarım

- Geri zincirleme: Örnek

1. Domuz(y)  $\wedge$  SümüklüBöcek(z)  $\rightarrow$  Hızlı(y,z)
2. Yapışkan(z)  $\wedge$  Sürünür(z)  $\rightarrow$  SümüklüBöcek(z)
3. Domuz(Pat)
4. Yapışkan(Steve)
5. Sürünür(Steve)



# Birinci-derece mantık ile çıkarım

- Daha önce belirttiğimiz üzere, GMP cümlelerin horn formunda olmasını gerektiriyor
  - Atomik veya birleşmelerden oluşan bir koşul, bir veya daha fazla önkoşula ve bir atomik cümle sonuca sahip
- Ancak bazı cümleler Horn formunda söylenenemez
  - Örnek:  $\forall x \neg \text{DerstenSıkılmış}(x)$
  - Olumsuzluk nedeniyle horn formunda göstermek mümkün değildir.
- Bu sebeple, BT'nında bulunan bazı cümleler çıkarım sırasında kullanılamaz.
  - Modus ponens ile çıkarım bütün değildir.

# Birinci-derece mantık ile çıkarım

- Bütünlük
  - Bütünlük teoremine göre bütün çıkışım kuralları bulmak mümkündür.
    - Teoreme göre, Bir cümle kümesi tarafından gerektirilen her cümleyi ispat edebiliriz.
  - Çözülme algoritması bütün bir çıkışım algoritmasıdır.
  - Bu önemli bir önermedir, çünkü iç içe geçmiş bir çok fonksiyon ile evrensel nicelendirme birleştiğinde sınırsız bir arama uzayı oluşturabilir.
  - Ancak birinci-derece mantıktaki gerektirme yarımkarar verilebilirdir.
    - Yani bir cümle gerektiriliyorsa bu ispatlanabilir.
    - Ancak gerektirilmiyorsa bu durum ispatlanamayabilir.

# Birinci-derece mantık ile çıkarım

- Bütünlük
  - İleri ve geri zincirleme yöntemleri Horn formundaki BT için bütündür, ancak genel birinci-derece mantık için bütün değildir.
    - Örnek:
      - $\text{PhD}(x) \rightarrow \text{YüksekNitelikli}(x)$
      - $\neg\text{PhD}(x) \rightarrow \text{ErkenKazanç}(x)$
      - $\text{YüksekNitelikli}(x) \rightarrow \text{Zengin}(x)$
      - $\text{ErkenKazanç}(x) \rightarrow \text{Zengin}(x)$
    - Bu BT'den  $\text{Zengin}(\text{ben})$  cümlesine erişebilmeliyiz.
    - Ancak İleri zincirleme veya geri zincirleme bunu başaramaz.

