

Yapay Zeka

Ders 13 – Bölüm 1

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

Birinci-derece mantık ile çıkarım

- Çözülme (İng: Resolution)
 - Çözülme bütün bir çıkarım algoritmasıdır.
 - Çelişki bulma yöntemi ile çalışır
 - $BT \models \alpha$ olduğunu göstermek için $BT \wedge \neg\alpha$ olamayacağını ispatlar.
 - Çözülmenin çalışması için BT ve $\neg\alpha$ birleşme normal formu (CNF) olmalıdır.
 - Çözülme ile iki koşul birleşerek yeni bir tane oluşturur.



- Boş koşul oluşuncaya kadar çıkarım devam eder.
 - Boş koşul bir çelişki anlamına gelir.

Birinci-derece mantık ile çıkarım

- Çözülme çıkarım kuralı
 - Basit önerimsel versiyonunu hatırlayalım

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

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

- Tam birinci-derece versiyonu

$$\frac{\begin{array}{c} p_1 \vee \dots p_j \dots \vee p_m, \\ q_1 \vee \dots q_k \dots \vee q_n \end{array}}{(p_1 \vee \dots p_{j-1} \vee p_{j+1} \dots p_m \vee q_1 \dots q_{k-1} \vee q_{k+1} \dots \vee q_n)\sigma}$$

- $p_j \sigma = \neg q_k \sigma$ Olduğunda
- Örnek: $\neg \text{Zengin}(x) \vee \text{Mutsuz}(x)$

Zengin(Ali)

Mutsuz(Ali)

- $\sigma = \{x/\text{Ali}\}$

Birinci-derece mantık ile çıkarım

- Birleştirim normal formu
 - İfade = (olumsuz olabilir) atomik cümle, örneğin $\neg \text{Zengin}(\text{Ali})$
 - Koşul = İfadelerin ayrışması, örneğin $\neg \text{Zengin}(\text{Ali}) \vee \text{Mutsuz}(\text{Ali})$
 - BT, koşulların birleşimidir.
 - Her birinci-derece mantık BT CNF haline aşağı belirtilen adımların uygulanması ile dönüştürülebilir.
 - $P \rightarrow Q$ kurallarını $\neg P \vee Q$ haline dönüştür.
 - \neg içeri hareket ettir. Örneğin $\neg \forall x P, \exists x \neg P$ olur.
 - Kullanılan değişken isimlerini değiştir. Örneğin $\forall x P \vee \exists x Q, \forall x P \vee \exists y Q$ haline gelir.
 - Niceleyicileri sola sırasıyla hareket ettir. Örneğin $\forall x P \vee \exists y Q, \forall x \exists y P \vee Q$
 - Skolem işlemi ile \exists sembollerinden kurtul
 - Evrensel niceleyicileri at
 - \wedge işlemi \vee üzerine dağıt. Örneğin $(P \wedge Q) \vee R, (P \vee R) \wedge (Q \vee R)$ olur

Birinci-derece mantık ile çıkarım

- Birleştirim normal formu
 - Skolem formu
 - $\exists x \text{ Zengin}(x)$ yerine $\text{Zengin}(G1)$ yazılır.
 - $G1$ yeni bir “skolem sabit” olarak tanımlanır.
 - \forall içinde \exists olduğunda “skolem fonksiyonu” kullanılır.
 - Her insanın bir kalbi vardır.
 - $\forall x \text{ İnsan}(x) \rightarrow \exists y \text{ Kalp}(y) \wedge \text{Sahip}(x,y)$ yerine
 - $\forall x \text{ İnsan}(x) \rightarrow \text{Kalp}(H(x)) \wedge \text{Sahip}(x,H(x))$
 - Skolem fonksiyonun argümanları evrensel nicelenmiş değişkenleri kapsar.

Birinci-derece mantık ile çıkarım

- Birinci-derece mantık cümlesini koşul formuna çevirme
 - $(\forall x)(P(x) \Rightarrow ((\forall y)(P(y) \Rightarrow P(f(x,y))) \wedge \neg(\forall y)(Q(x,y) \Rightarrow P(y))))$
 - \rightarrow eleme
 - $(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge \neg(\forall y)(\neg Q(x,y) \vee P(y))))$
 - Olumsuzun etki alanını azalt
 - $(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge (\exists y)(Q(x,y) \wedge \neg P(y))))$
 - Değişkenleri standartlaştır
 - $(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge (\exists z)(Q(x,z) \wedge \neg P(z))))$
 - \exists eleme
 - $(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge (Q(x,g(x)) \wedge \neg P(g(x)))))$
 - \forall kaldır
 - $(\neg P(x) \vee ((\neg P(y) \vee P(f(x,y))) \wedge (Q(x,g(x)) \wedge \neg P(g(x)))))$

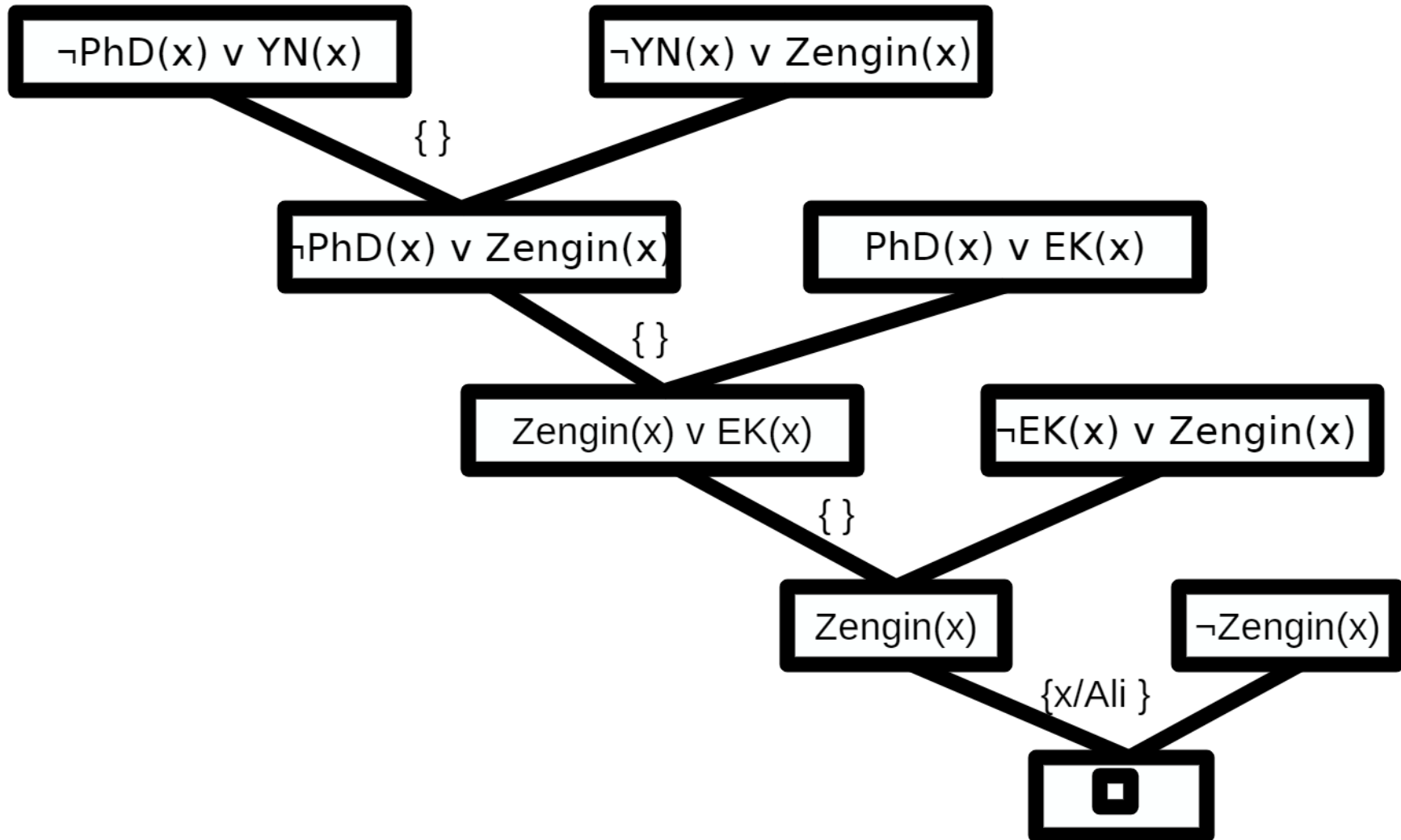
Birinci-derece mantık ile çıkarım

- Birinci-derece mantık cümlesini koşul cümlesine çevirme
 - $(\neg P(x) \vee ((\neg P(y) \vee P(f(x,y))) \wedge (Q(x,g(x)) \wedge \neg P(g(x)))))$
 - Ayrılmaların birleşimine çevir
 - $(\neg P(x) \vee \neg P(y) \vee P(f(x,y))) \wedge (\neg P(x) \vee Q(x,g(x))) \wedge (\neg P(x) \vee \neg P(g(x)))$
 - Ayrı koşullar oluştur
 - $\neg P(x) \vee \neg P(y) \vee P(f(x,y))$
 $\neg P(x) \vee Q(x,g(x))$
 $\neg P(x) \vee \neg P(g(x))$
 - Değişkenleri standartlaştır
 - $\neg P(x) \vee \neg P(y) \vee P(f(x,y))$
 $\neg P(z) \vee Q(z,g(z))$
 $\neg P(w) \vee \neg P(g(w))$

Birinci-derece mantık ile çıkarım

- Çözülme ile ispat
 - α cümlesini ispatlamak için
 - Olumsuz yap
 - CNF haline getir
 - CNF durumundaki BT'ye ekle
 - Çelişki bul
 - Örnek: Zengin(Ali) cümlesini ispatlamak için \neg Zengin(Ali) cümlesini CNF BT'ye ekle
 - $\neg \text{PhD}(x) \vee \text{YüksekNitelikli}(x)$
 - $\text{PhD}(x) \vee \text{ErkenKazanç}(x)$
 - $\neg \text{YüksekNitelikli}(x) \vee \text{Zengin}(x)$
 - $\neg \text{ErkenKazanç}(x) \vee \text{Zengin}(x)$

Birinci-derece mantık ile çıkarım



Birinci-derece mantık ile çıkarım

- Mantık programlama (İng: Logic programming)
 - Mantıksal BT üzerinde otomatik çıkarım yapma

Mantık Programlama

1. Problemi tanımla
2. Bilgi topla
3. Çay molası
4. Bilgiyi BT'de tanımla
5. Problem örneklerini gerçek olarak tanımla
6. Sorgu gönder
7. Yanlış gerçekleri bul

Normal Programlama

- Problemi tanımla
- Bilgi topla
- Çözümü oluştur
- Çözümü programla
- Problem örneklerini veri olarak tanımla
- Programa veri gönder
- Fonksiyon hatalarını bul ve düzelt

Birinci-derece mantık ile çıkarım

- Prolog ile mantık programlama
 - Temel: Horn cümleleri ve zil-ıslık ile geri zincirleme
 - Avrupa ve Japonya'da birçok kullanıcısı vardır.
 - Program = koşullar kümesi = baş :- ifade₁, ... ifade_n
 - Birleştirme açık kodlama ile verimli şekilde yapılır
 - Direkt eşleme ile verimli şekilde bilgi elde edilir.
 - Derinlik-öncelikli, soldan sağa geri zincirleme
 - Artimetik için önceden tanımlı önerimler içerir.

