

Yapay Zeka

Ders 11 - Bölüm 1

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

Mantıklı etmenler

- Çıkarım ve ispatlar

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

$$\frac{\alpha \wedge \beta}{\alpha}$$

$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}$$

$$\frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$

Mantıklı etmenler

- Çıkarım ve ispatlar: R_5 kuralına kadar olan kurallar 24. slaytta.

$$R_6 : (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$$

$$R_7 : ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$$

$$R_8 : (\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1}))$$

$$R_9 : \neg(P_{1,2} \vee P_{2,1})$$

$$R_{10} : \neg P_{1,2} \wedge \neg P_{2,1}$$

Mantıklı etmenler

- Bir önceki slaytta gösterdiğimiz ispatı el ile yaptık.
- Ancak bu şekilde çözüme ulaşan, daha önce öğrendiğimiz arama algoritmalarını kullanan bir yöntem geliştirebiliriz.
 - Başlangıç durumu: baştaki bilgi tabanı
 - Aksiyonlar: Gördüğümüz çıkarım kurallarının üstte kalan kısımlarını cümlelere uygularız.
 - Sonuç: Çıkarım kurallarının altta kalan kısmı ile oluşan cümleleri bilgi tabanına ekleriz.
 - Hedef: Hedef durumumuz başta ispatlamak istediğimiz cümledir.

Mantıklı etmenler

- Çözülme ile ispat
 - Çözülme (İng: resolution) oldukça sıkılıkla kullanılan bir çıkarım yöntemidir.
 - Aşağıdaki iki gerçeği bilgi tabanımıza ekleyelim

$$R_{11} : \neg B_{1,2} .$$

$$R_{12} : B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$$

- Daha önce R_{10} bulunurken yaptığımız gibi [2,2] ve [1,3] odalarında uçurum olmadığı sonucuna ulaşabiliriz.

$$R_{13} : \neg P_{2,2}$$

$$R_{14} : \neg P_{1,3}$$

- [1,1], [2,2] veya [3,1] odalarından birinde uçurum olmalı

$$R_{15} : P_{1,1} \vee P_{2,2} \vee P_{3,1}$$

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Mantıklı etmenler

- Çözülme ile ispat
 - Çözülme kuralına göre, [2,2] odasında uçurum olmadığına göre

$$R_{16} : P_{1,1} \vee P_{3,1}$$

- Daha önce [1,1] odasında uçurum olmadığını biliyorduk. Öyleyse tekrar çözülme kuralını uygularsak

$$R_{17} : P_{3,1}$$

- Çözülme kuralı genel olarak şunu söyler:
 - Aşağıdaki cümlede m ile ℓ_i birbirini tamamalyıcı ifadeler ise (birbirlerinin zıtları ise.

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \quad m}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k}$$

Mantıklı etmenler

- Çözülme ile ispat
 - Çözülme kuralını uygulayabilmemiz için cümlenin birleşmeler ile ayrılmış durumda olması gereklidir.
 - Bu durumu birleşme normal formu (İng: conjunctive normal form) denir.
 - Herhangi bir ifadeyi birleşme normal hal yapabiliriz.
 - Örnek: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
 - $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
 - $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$
 - $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$
 - $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Mantıklı etmenler

- Bir çözülme algoritması
 - Çözülme mantığıyla çıkarım yapan algoritmalar çelişki ile ispat yöntemini (İng: proof by contradiction) kullanırlar.
 - $BT \models \alpha$ doğru olduğunu göstermek için $(BT \wedge \neg\alpha)$ 'nın imkansız olduğunu (tatmin edilemez olduğunu) gösterirler.

fonksiyon PL-ÇÖZÜLME(BT, α) **dönüş** *true* veya *false*

girdiler: BT , bilgi tabanı, önerimsel mantık cümlesi
 α , soru, önerimsel mantık cümlesi

ifadeler $\leftarrow BT \wedge \neg\alpha$ içerisindeki BNF durumundaki ifadeler kümesi
yeni $\leftarrow \{ \}$

loop do
for each ifade çifti C_i, C_j **in** ifadeler **do**

çözüçüler \leftarrow PL-ÇÖZ(C_i, C_j)

if çözüçüler boş ifadeyi içeriyorsa **then return true**

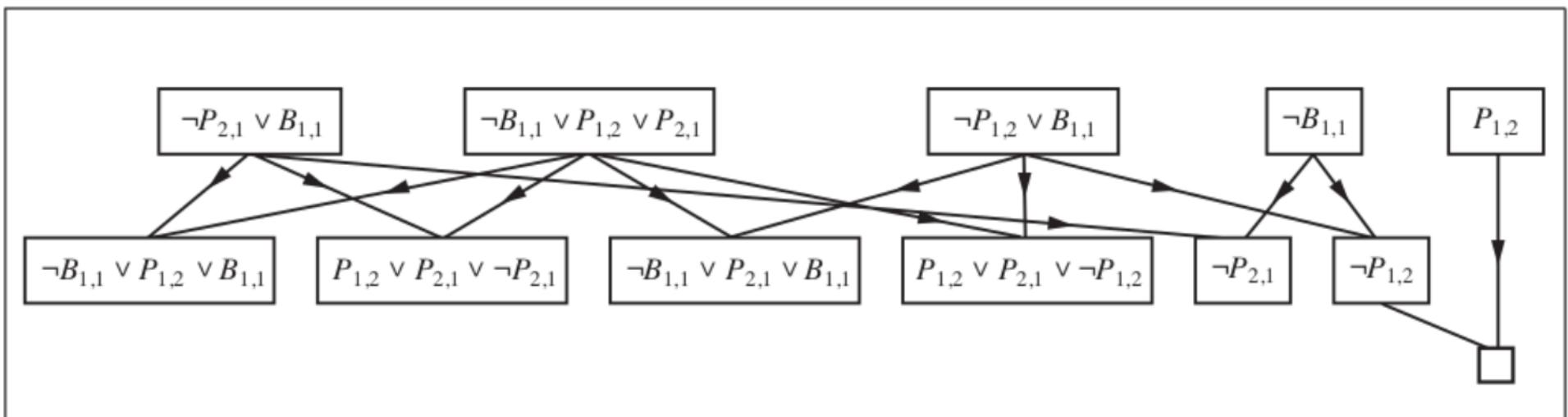
yeni \leftarrow *yeni* \cup çözüçüler

if *yeni* \subseteq ifadeler **then return false**

ifadeler \leftarrow *ifadeler* \cup *yeni*

Mantıklı etmenler

- Bir çözülme algoritması



- Algoritma çalıştığında boş ifadeye ulaşıyoruz, başta belirttiğimiz ifade BT tarafından gerektirilmektedir anlamına gelir.

- $BT \models \alpha$
- Çözülme algoritması bütündür, yani gerektirilen her cümleye ulaşır.

Mantıklı etmenler

- İleri ve geri zincirleme
 - Birçok durumda çözülme kullanmasına gerek duyulmadan cümlelerin doğruluğu sınanabilir.
 - Horn cümlesi (İng: horn clause) sınırlı bir cümle çeşididir.
 - Bir horn cümlesi en fazla bir tanesi olumlu olabilen terimlerin ayrılımasından oluşur.
 - Belirli cümleler: Tam olarak bir tane olumlu terim içeren ayrılmalar.
 - Örnek: $\neg L_{1,1} \vee \neg Breeze \vee B_{1,1}$
 - Hedef cümleler: Tamamı olumsuz terimlerden oluşan ayrılmalar.
 - Örnek: $\neg B_{1,1} \vee \neg L_{1,1} \vee \neg Breeze$
 - İki horn cümlesi üzerinde çözülme uygularsak, elde ettiğimiz cümle de horn cümlesidir.
 - Yani horn cümleleri çözülme altında kapalıdır.

Mantıklı etmenler

- İleri ve geri zincirleme
 - Her belirli cümle önermesi olumlu terimlerin birleşmesi ve sonucu bir olumlu terim olan koşul cümlesi olarak yazılabilir.
 - Örnek: $\neg L_{1,1} \vee \neg Breeze \vee B_{1,1}$ cümlesi $(L_{1,1} \wedge Breeze) \Rightarrow B_{1,1}$ şeklinde yazılabilir.
 - Koşul şeklinde yazılan cümleleri anlamak daha kolaydır.
 - Önerme kısmı gövde, sonuç kısmı ise baş olarak adlandırılır.
 - Horn cümleleri ile çıkarım ileri ve geri zincirleme yöntemi ile yapılabilir.
 - Horn cümleleri ile çıkarım yapmak bilgi tabanının büyüklüğü ile doğrusal zamanda yapılabilir.

Mantıklı etmenler

- İleri ve geri zincirleme

fonksiyon PL-İZ-GEREKTİRİR?(*BT*, α) **dönüş** *true* veya *false*

girdiler: *BT*, bilgi tabanı, önerimsel mantık cümlesi
 q , soru, bir önerimsel sembol

sayac \leftarrow bir tablo, *sayac*[*c*] *c*'nin önerme kısmındaki sembol sayısıdır.

çıkarılmış \leftarrow bir tablo, *çıkarılmış*[*s*] başlangıçta her sembol için *false*

ajanda \leftarrow bir sembol sırası, başlangıçta *BT* içinde doğru olduğu bilinen sembollerin içindedir.

while *ajanda* boş değil ise **do**

p \leftarrow POP(*ajanda*)

if *p* = *q* **then return** *true*

if *çıkarılmış*[*p*] = *false* **then**

çıkarılmış[*p*] \leftarrow *true*

for each *p*'nin *c*.ÖNERME içinde olan *BT* içerisindeki *c* cümleleri için **do**

 azalt *count*[*c*]

if *count*[*c*] = 0 **then add** *c*.SONUÇ **to** *ajanda*

return *false*

Mantıklı etmenler

- İleri ve geri zincirleme

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

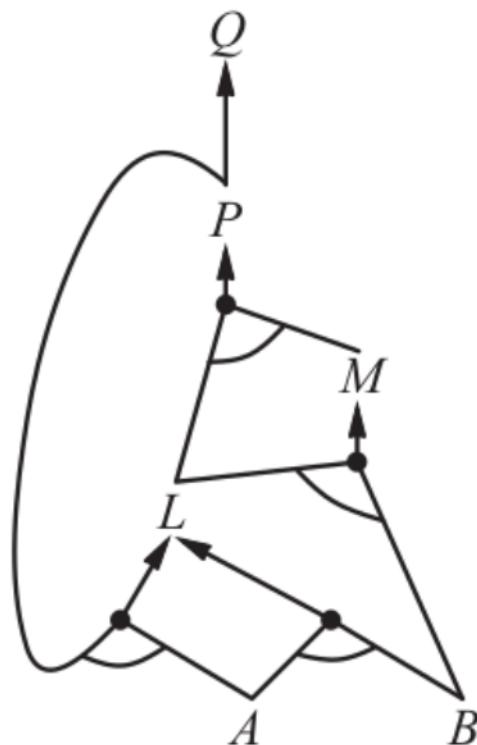
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

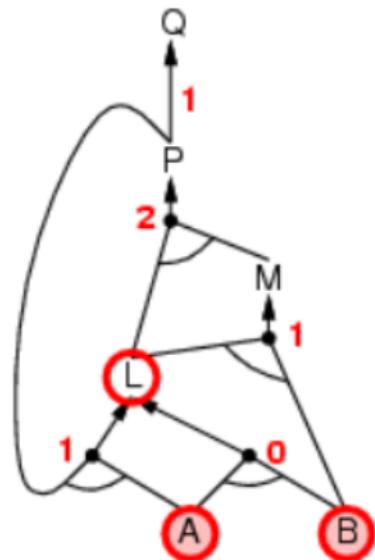
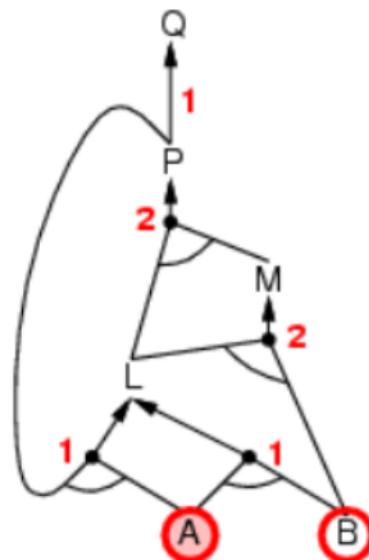
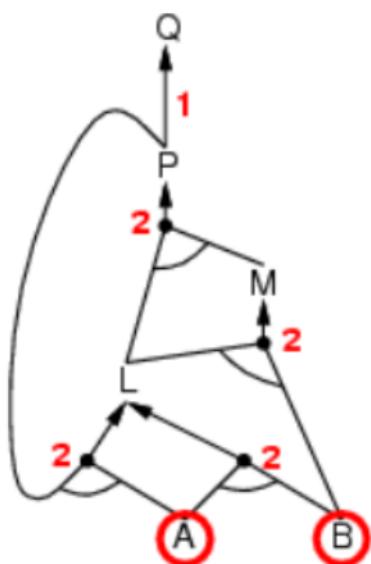
$$A$$

$$B$$



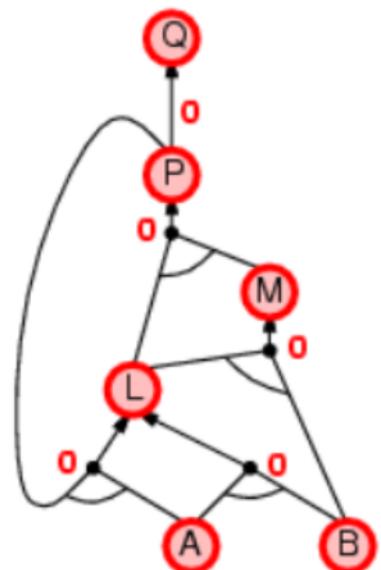
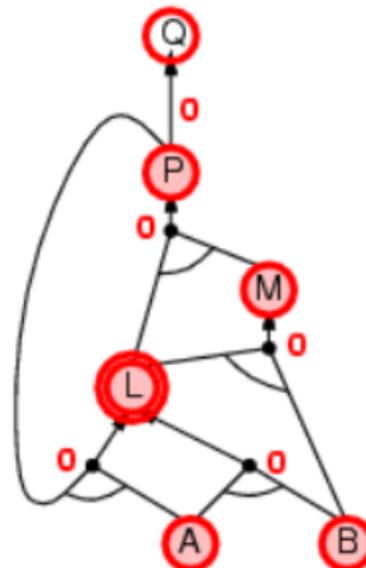
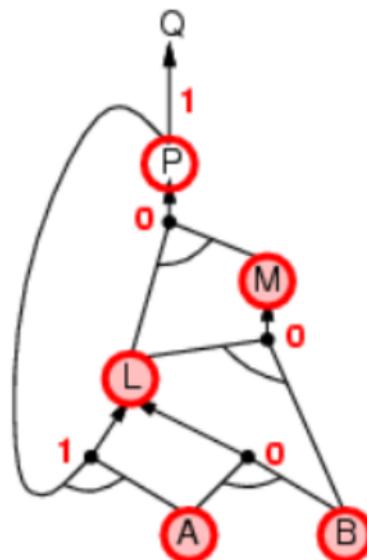
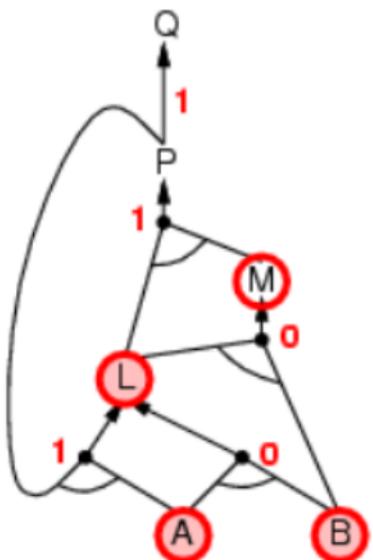
Mantıklı etmenler

- İleri ve geri zincirleme



Mantıklı etmenler

- İleri ve geri zincirleme

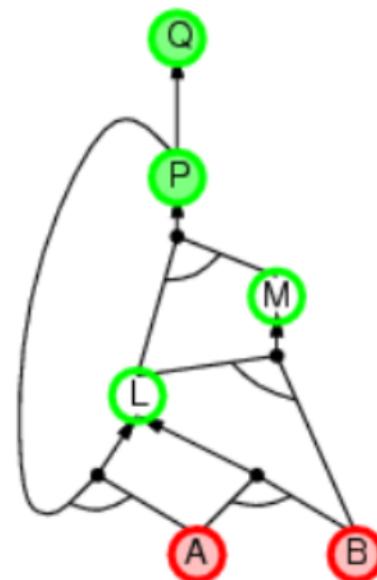
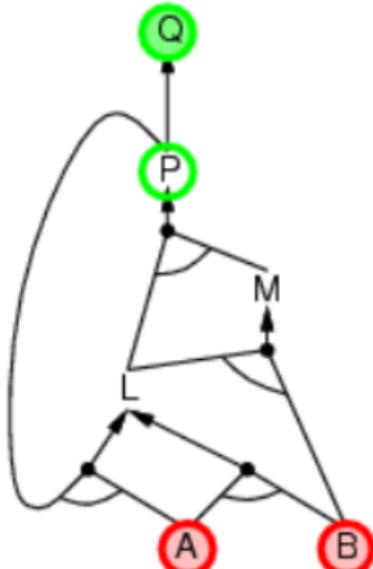
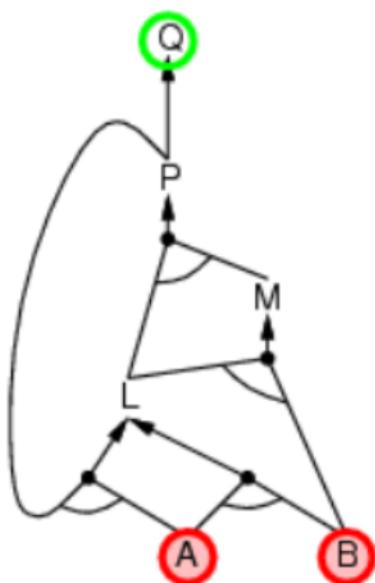


Mantıklı etmenler

- İleri ve geri zincirleme
 - İleri zincirleme yöntemi geçerlidir.
 - Her çıkarım Modus Ponens kuralının uygulanması ile oluşur.
 - İleri zincirleme yöntemi bütündür.
 - Her gerektirilen atomik cümleye ulaşılacaktır.
 - İleri zincirleme veri güdümlü muhakemeye örnektir.
 - Bu yöntem ile çıkarım yapma aktivitesi eldeki veriler ile başlar.
 - Geri zincirleme yöntemi ise verilen soru, q , ile başlar.
 - q cümlesinin doğru olduğu biliniyorsa tamamdır, yoksa q cümlesinin öncüllerine bakılır.
 - Bu şekilde geriye doğru hareket edilir.
 - Geri zincirleme hedef-odaklı muhakemeye örnektir.

Mantıklı etmenler

- İleri ve geri zincirleme



Mantıklı etmenler

- İleri ve geri zincirleme

