

## مقدمه

- تقلیل استنتاج مرتبه-اول به استنتاج گزاره ای
- یکسان سازی (Unification)
- MP تعمیم یافته (GMP)
- زنجیره استنتاج روبه جلو
- زنجیره استنتاج رو به عقب
- رزولوشن

N. Razavi- AI course - 2006

2

## استنتاج در منطق مرتبه اول

فصل نهم

سید ناصر رضوی

E-mail: [razavi@Comp.iust.ac.ir](mailto:razavi@Comp.iust.ac.ir)

۱۳۸۴

## نمونه سازی عمومی (UI)

- هر نمونه از یک جمله دارای سور عمومی، توسط آن جمله قابل استلزام است.

$$\frac{\forall v \alpha}{\text{Subst}(\{v/g\}, \alpha)}$$

- مثال:  $\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$

$$\text{King}(\text{John}) \wedge \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John})$$

$$\text{King}(\text{Richard}) \wedge \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard})$$

$$\text{King}(\text{Father}(\text{John})) \wedge \text{Greedy}(\text{Father}(\text{John})) \Rightarrow \text{Evil}(\text{Father}(\text{John}))$$

•  
•  
•

N. Razavi- AI course - 2006

3

## نمونه سازی وجودی (EI)

- برای هر جمله  $\alpha$ ، متغیر  $V$  و سیمبول ثابت  $k$  که در جای دیگری از پایگاه دانش ظاهر نشده باشد:

$$\frac{\exists v \alpha}{\text{Subst}(\{v/k\}, \alpha)}$$

- مثال:  $\exists x \text{ Crown}(x) \wedge \text{OnHead}(x, \text{John})$

$$\text{Crown}(C_1) \wedge \text{OnHead}(C_1, \text{John})$$

- به شرطی که  $C_1$  یک سیمبول جدید باشد، که به آن ثابت اسکولم می گویند.

N. Razavi- AI course - 2006

4

## تقلیل به استنتاج گزاره ای (ادامه)

- هر پایگاه دانش در FOL می تواند به گونه ای گزاره ای سازی شود که استلزام را حفظ کند
- یک جمله توسط KB جدید قابل استلزام است اگر و فقط اگر توسط KB اصلی قابل استلزام باشد
- ایده: پایگاه دانش و کوئری را گزاره ای سازی کن، رزولوشن را اعمال و نتیجه را برگردان
- مشکل: در مورد سیمبول های تابعی تعداد نامحدودی ترم زمینی وجود دارد، مثلاً:

*Father(Father(Father(John)))*

## تقلیل به استنتاج گزاره ای

- فرض کنید که KB فقط شامل جملات زیر باشد:  
 $\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$   
 $\text{King}(\text{John})$   
 $\text{Greedy}(\text{John})$   
 $\text{Brother}(\text{Richard}, \text{John})$
- با نمونه سازی جمله دارای سور عمومی به **تمام طرق ممکن**، داریم:  
 $\text{King}(\text{John}) \wedge \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John})$   
 $\text{King}(\text{Richard}) \wedge \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard})$   
 $\text{King}(\text{John})$   
 $\text{Greedy}(\text{John})$   
 $\text{Brother}(\text{Richard}, \text{John})$
- KB جدید **گزاره ای سازی** شده است. سیمبول های گزاره ای عبارتند از:  
 $\text{King}(\text{John}), \text{Greedy}(\text{John}), \text{Evil}(\text{John}), \text{King}(\text{Richard}), \text{etc}$

## مشکلات گزاره ای سازی نمودن

- به نظر می رسد که گزاره ای سازی کردن جملات نامربوط زیادی تولید می کند. برای مثال از جملات زیر:

$\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$

$\text{King}(\text{John})$

$\forall y \text{ Greedy}(y)$

$\text{Brother}(\text{Richard}, \text{John})$

- بدیهی است که  $\text{Evil}(\text{John})$ . اما گزاره ای سازی کردن حقایق زیادی مانند  $\text{Greedy}(\text{Richard})$  تولید می کند که نامربوط می باشند

## تقلیل به استنتاج گزاره ای (ادامه)

- تئوری Herbrand (۱۹۳۰): اگر جمله  $\alpha$  توسط پایگاه دانش FOL قابل استلزام باشد، آنگاه این جمله توسط زیرمجموعه محدودی از پایگاه دانش گزاره ای سازی شده قابل استلزام است
- ایده:
- For  $n = 0$  to  $\infty$  do  
 create a propositional KB by instantiating with depth- $n$  terms  
 see if  $\alpha$  is entailed by this KB
- مشکل: اگر KB مستلزم  $\alpha$  باشد کار می کند، در غیر اینصورت در حلقه بی نهایت می افتد
- تئوری تورینگ و چرچ (۱۹۳۶): استلزام برای FOL **نیمه تصمیم پذیر** است
- الگوریتم هایی وجود دارند که برای هر جمله قابل استلزام پاسخ بله را تولید می کنند، ولی الگوریتمی وجود ندارد که برای هر جمله ای که قابل استلزام نباشد پاسخ خیر تولید کند.

## یکسان سازی (Unification)

- اگر بتوانیم یک جانشینی مانند  $\theta$  بیابیم به طوری که  $\text{King}(x)$  و  $\text{Greedy}(x)$  با  $\text{King}(\text{John})$  و  $\text{Greedy}(y)$  تطبیق یابند، آنگاه استنتاج بلافاصله بدست می آید.

$$\theta = \{x/\text{John}, y/\text{John}\}$$

- $\text{Unify}(\alpha, \beta) = \theta$  if  $\alpha\theta = \beta\theta$

p	q	$\theta$
Knows(John,x)	Knows(John,Jane)	
Knows(John,x)	Knows(y,OJ)	
Knows(John,x)	Knows(y,Mother(y))	
Knows(John,x)	Knows(x,OJ)	

## یکسان سازی (Unification)

- اگر بتوانیم یک جانشینی مانند  $\theta$  بیابیم به طوری که  $\text{King}(x)$  و  $\text{Greedy}(x)$  با  $\text{King}(\text{John})$  و  $\text{Greedy}(y)$  تطبیق یابند، آنگاه استنتاج بلافاصله بدست می آید.

$$\theta = \{x/\text{John}, y/\text{John}\}$$

- $\text{Unify}(\alpha, \beta) = \theta$  if  $\alpha\theta = \beta\theta$

p	q	$\theta$
Knows(John,x)	Knows(John,Jane)	$\{x/\text{Jane}\}$
Knows(John,x)	Knows(y,OJ)	
Knows(John,x)	Knows(y,Mother(y))	
Knows(John,x)	Knows(x,OJ)	

## یکسان سازی (Unification)

- اگر بتوانیم یک جانشینی مانند  $\theta$  بیابیم به طوری که  $\text{King}(x)$  و  $\text{Greedy}(x)$  با  $\text{King}(\text{John})$  و  $\text{Greedy}(y)$  تطبیق یابند، آنگاه استنتاج بلافاصله بدست می آید.

$$\theta = \{x/\text{John}, y/\text{John}\}$$

- $\text{Unify}(\alpha, \beta) = \theta$  if  $\alpha\theta = \beta\theta$

p	q	$\theta$
Knows(John,x)	Knows(John,Jane)	$\{x/\text{Jane}\}$
Knows(John,x)	Knows(y,OJ)	$\{x/\text{OJ}, y/\text{John}\}$
Knows(John,x)	Knows(y,Mother(y))	
Knows(John,x)	Knows(x,OJ)	

## یکسان سازی (Unification)

- اگر بتوانیم یک جانشینی مانند  $\theta$  بیابیم به طوری که  $\text{King}(x)$  و  $\text{Greedy}(x)$  با  $\text{King}(\text{John})$  و  $\text{Greedy}(y)$  تطبیق یابند، آنگاه استنتاج بلافاصله بدست می آید.

$$\theta = \{x/\text{John}, y/\text{John}\}$$

- $\text{Unify}(\alpha, \beta) = \theta$  if  $\alpha\theta = \beta\theta$

p	q	$\theta$
Knows(John,x)	Knows(John,Jane)	$\{x/\text{Jane}\}$
Knows(John,x)	Knows(y,OJ)	$\{x/\text{OJ}, y/\text{John}\}$
Knows(John,x)	Knows(y,Mother(y))	$\{y/\text{John}, x/\text{Mother}(\text{John})\}$
Knows(John,x)	Knows(x,OJ)	

## یکسان سازی (Unification)

- اگر بتوانیم یک جانشینی مانند  $\theta$  بیابیم به طوری که  $\text{King}(x)$  و  $\text{Greedy}(x)$  با  $\text{King}(\text{John})$  و  $\text{Greedy}(y)$  تطبیق یابند، آنگاه استنتاج بلافاصله بدست می آید.  
 $\theta = \{x/\text{John}, y/\text{John}\}$

- $\text{Unify}(\alpha, \beta) = \theta$  if  $\alpha\theta = \beta\theta$

p	q	$\theta$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(\text{John}, \text{Jane})$	$\{x/\text{Jane}\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(y, \text{OJ})$	$\{x/\text{OJ}, y/\text{John}\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(y, \text{Mother}(y))$	$\{y/\text{John}, x/\text{Mother}(\text{John})\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(x, \text{OJ})$	$\{\text{fail}\}$
استاندارد سازی متغیرها: مثلاً $\text{Knows}(z, \text{OJ})$ N. Razavi- AI course - 2006		

13

## یکسان سازی

- برای یکسازی سازی  $\text{Knows}(\text{John}, x)$  و  $\text{Knows}(y, z)$  دو جانشینی وجود دارد یا

- $\theta = \{y/\text{John}, x/z\}$
- $\theta = \{y/\text{John}, x/\text{John}, z/\text{John}\}$

- اولین یکسان ساز عمومی تر از دومی می باشد
- فقط یک عمومی ترین یکسان ساز (MGU) وجود دارد که منحصر به فرد می باشد.

$$\text{MGU} = \{y/\text{John}, x/z\}$$

N. Razavi- AI course - 2006

14

## الگوریتم یکسان سازی

```

function UNIFY( $x, y, \theta$ ) returns a substitution to make  $x$  and  $y$  identical
inputs:  $x$ , a variable, constant, list, or compound
        $y$ , a variable, constant, list, or compound
        $\theta$ , the substitution built up so far

if  $\theta = \text{failure}$  then return failure
else if  $x = y$  then return  $\theta$ 
else if VARIABLE?( $x$ ) then return UNIFY-VAR( $x, y, \theta$ )
else if VARIABLE?( $y$ ) then return UNIFY-VAR( $y, x, \theta$ )
else if COMPOUND?( $x$ ) and COMPOUND?( $y$ ) then
    return UNIFY(ARGS[ $x$ ], ARGS[ $y$ ], UNIFY(OP[ $x$ ], OP[ $y$ ],  $\theta$ ))
else if LIST?( $x$ ) and LIST?( $y$ ) then
    return UNIFY(REST[ $x$ ], REST[ $y$ ], UNIFY(FIRST[ $x$ ], FIRST[ $y$ ],  $\theta$ ))
else return failure
    
```

N. Razavi- AI course - 2006

15

## الگوریتم یکسان سازی

```

function UNIFY-VAR( $var, x, \theta$ ) returns a substitution
inputs:  $var$ , a variable
        $x$ , any expression
        $\theta$ , the substitution built up so far

if  $\{var/val\} \in \theta$  then return UNIFY( $val, x, \theta$ )
else if  $\{x/val\} \in \theta$  then return UNIFY( $var, val, \theta$ )
else if OCCUR-CHECK?( $var, x$ ) then return failure
else return add  $\{var/x\}$  to  $\theta$ 
    
```

N. Razavi- AI course - 2006

16

## Generalized Modus Ponens (GMP)

$$\frac{p_1', p_2', \dots, p_n', (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{q\theta} \quad \text{where } p_i'\theta = p_i \theta \text{ for all } i$$

$p_1'$  is *King(John)*       $p_1$  is *King(x)*  
 $p_2'$  is *Greedy(y)*       $p_2$  is *Greedy(x)*  
 $\theta$  is  $\{x/\text{John}, y/\text{John}\}$        $q$  is *Evil(x)*  
 $q\theta$  is *Evil(John)*

- GMP با پایگاه دانشی از فراکردهای معین (دقیقاً یک لیترال مثبت) کار می کند
- فرض می شود که تمام متغیرها دارای سور عمومی هستند.

## صحت GMP

- باید نشان دهیم:

$$p_1', \dots, p_n', (p_1 \wedge \dots \wedge p_n \Rightarrow q) \models q\theta$$

به شرطی که برای هر  $i$  داشته باشیم:  $p_i'\theta = p_i\theta$

- لم: برای هر جمله  $p$  توسط UI داریم:  $p \models p\theta$

1.  $(p_1 \wedge \dots \wedge p_n \Rightarrow q) \models (p_1 \wedge \dots \wedge p_n \Rightarrow q)\theta = (p_1\theta \wedge \dots \wedge p_n\theta \Rightarrow q\theta)$
2.  $p_1', \dots, p_n' \models p_1' \wedge \dots \wedge p_n' \models p_1'\theta \wedge \dots \wedge p_n'\theta$
3. From 1 and 2,  $q\theta$  follows by ordinary Modus Ponens

## پایگاه دانش نمونه

- The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.
- 
- Prove that Colonel West is a criminal

## پایگاه دانش نمونه (ادامه)

... it is a crime for an American to sell weapons to hostile nations:

$$\text{American}(x) \wedge \text{Weapon}(y) \wedge \text{Sells}(x,y,z) \wedge \text{Hostile}(z) \Rightarrow \text{Criminal}(x)$$

Nono ... has some missiles, i.e.,  $\exists x \text{ Owns}(\text{Nono}, x) \wedge \text{Missile}(x)$ :

$$\text{Owns}(\text{Nono}, M1) \text{ and } \text{Missile}(M1)$$

... all of its missiles were sold to it by Colonel West

$$\text{Missile}(x) \wedge \text{Owns}(\text{Nono}, x) \Rightarrow \text{Sells}(\text{West}, x, \text{Nono})$$

Missiles are weapons:

$$\text{Missile}(x) \Rightarrow \text{Weapon}(x)$$

An enemy of America counts as "hostile":

$$\text{Enemy}(x, \text{America}) \Rightarrow \text{Hostile}(x)$$

West, who is American ...

$$\text{American}(\text{West})$$

The country Nono, an enemy of America

## الگوریتم استنتاج رو به جلو

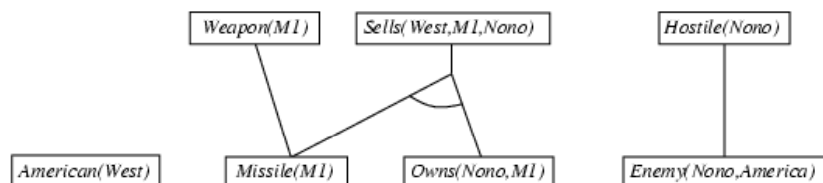
```

function FOL-FC-Ask( $KB, \alpha$ ) returns a substitution or false
  repeat until new is empty
     $new \leftarrow \{\}$ 
    for each sentence  $r$  in  $KB$  do
       $(p_1 \wedge \dots \wedge p_n \Rightarrow q) \leftarrow \text{STANDARDIZE-APART}(r)$ 
      for each  $\theta$  such that  $(p_1 \wedge \dots \wedge p_n)\theta = (p'_1 \wedge \dots \wedge p'_n)\theta$ 
        for some  $p'_1, \dots, p'_n$  in  $KB$ 
           $q' \leftarrow \text{SUBST}(\theta, q)$ 
          if  $q'$  is not a renaming of a sentence already in  $KB$  or  $new$  then do
            add  $q'$  to  $new$ 
             $\phi \leftarrow \text{UNIFY}(q', \alpha)$ 
            if  $\phi$  is not fail then return  $\phi$ 
    add  $new$  to  $KB$ 
  return false
    
```

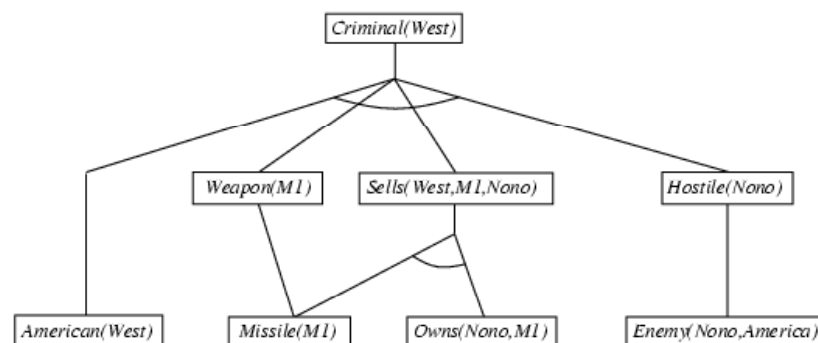
## اثبات توسط استنتاج رو به جلو

$American(West)$ 
 $Missile(M1)$ 
 $Owns(Nono, M1)$ 
 $Enemy(Nono, America)$

## اثبات توسط استنتاج رو به جلو



## اثبات توسط استنتاج رو به جلو



## کارایی استنتاج رو به جلو

- استنتاج رو به جلوی افزایشی: در تکرار  $k$  ام نیاز به تطبیق قانونی که هیچ کدام از شرایطش در تکرار  $k-1$  اضافه نشده اند، نیست
- خود انطباق می تواند پرهزینه باشد:  
– شاخص بندی پایگاه داده اجازه می دهد که حقایق شناخته شده در زمان  $O(1)$  بازیابی شوند.
- استنتاج رو به جلو به طور گسترده ای در پایگاه های داده استنتاجی به کار می رود

## خصوصیات استنتاج رو به جلو

- برای فراکردهای معین FOL، کامل و صحیح است
- Datalog = فراکردهای معین FOL بدون سیمبول تابعی
- FC برای Datalog با تعداد تکرارهای محدودی متوقف می شود.
- در حالت کلی، اگر پایگاه دانش مستلزم  $\alpha$  نباشد، ممکن است متوقف نشود
- این اجتناب ناپذیر است: استلزام با فراکردهای معین نیمه تصمیم پذیر است

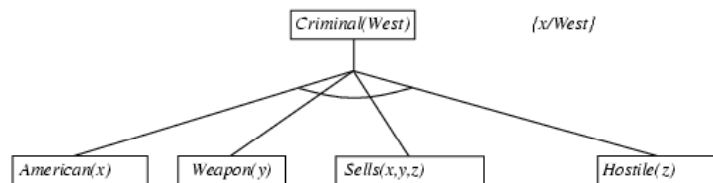
## مثال استنتاج رو به عقب

*Criminal(West)*

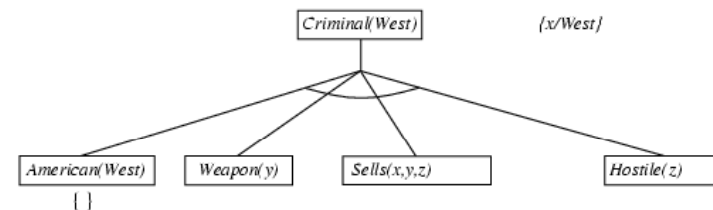
```
function FOL-BC-ASK( $KB, goals, \theta$ ) returns a set of substitutions
  inputs:  $KB$ , a knowledge base
          $goals$ , a list of conjuncts forming a query
          $\theta$ , the current substitution, initially the empty substitution  $\{ \}$ 
  local variables:  $ans$ , a set of substitutions, initially empty
  if  $goals$  is empty then return  $\{ \theta \}$ 
   $q' \leftarrow SUBST(\theta, FIRST(goals))$ 
  for each  $r$  in  $KB$  where  $STANDARDIZE-APART(r) = (p_1 \wedge \dots \wedge p_n \Rightarrow q)$ 
    and  $\theta' \leftarrow UNIFY(q, q')$  succeeds
     $ans \leftarrow FOL-BC-ASK(KB, [p_1, \dots, p_n | REST(goals)], COMPOSE(\theta, \theta')) \cup ans$ 
  return  $ans$ 
```

$SUBST(COMPOSE(\theta_1, \theta_2), p) = SUBST(\theta_2, SUBST(\theta_1, p))$

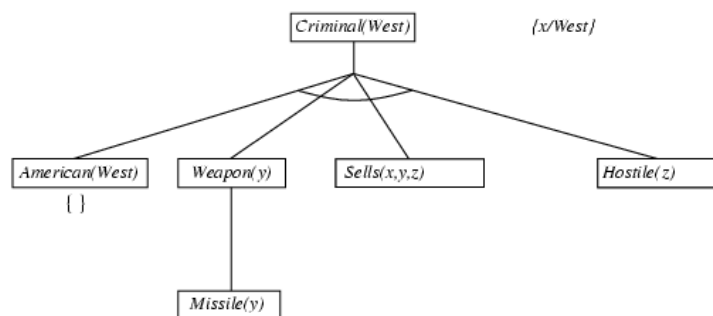
## مثال استنتاج رو به عقب



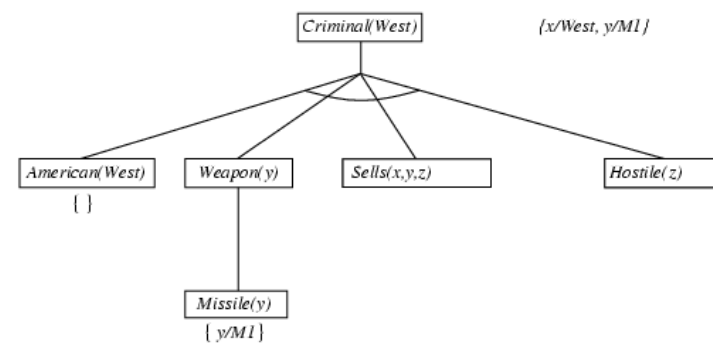
## مثال استنتاج رو به عقب



## مثال استنتاج رو به عقب

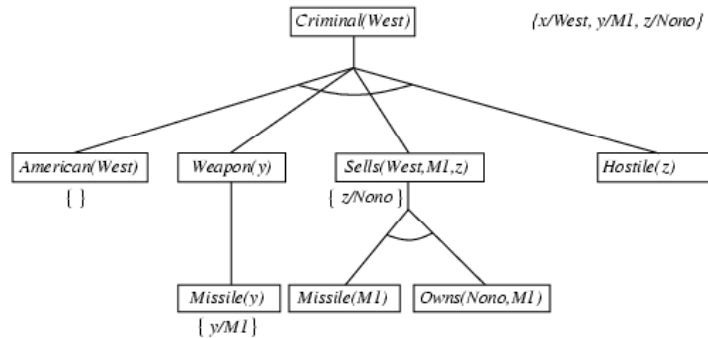


## مثال استنتاج رو به عقب

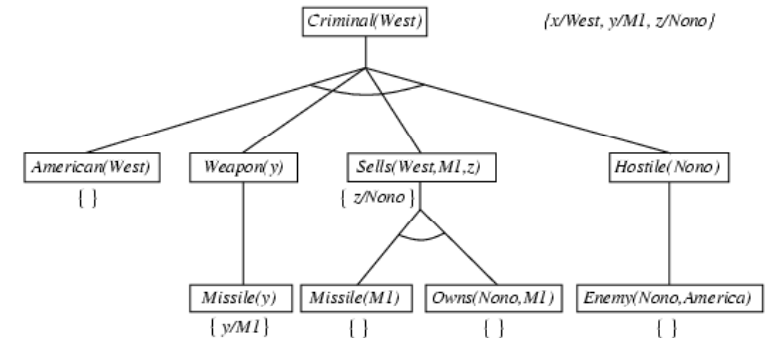




## مثال استنتاج رو به عقب



## مثال استنتاج رو به عقب



## خصوصیات استنتاج رو به عقب

- اثبات: جستجوی اول-عمق بازگشتی - فضای حالت بر حسب اثبات خطی
- ناکامل به دلیل وجود حلقه ها
- - راه حل: مقایسه هدف فعلی با تمام اهداف موجود در پشته
- ناکارآ به دلیل زیرهدف های تکراری (چه موفقیت و چه شکست)
- - راه حل: ذخیره نتایج قبلی (با مصرف حافظه بیشتر) - نوعی memoization
- به طور گسترده ای در **برنامه نویسی منطقی** استفاده می شود

## برنامه نویسی منطقی: پرولوگ

- Algorithm = Logic + Control
- اصول: عقبگرد با فراکردهای هورن- به طور گسترده ای در اروپا و ژاپن استفاده شده است ( در پروژه نسل پنجم کامپیوترها)
- برنامه = مجموعه ای از فراکردها
- فراکرد =

head :- literal<sub>1</sub>, ... literal<sub>n</sub>.

- مثال:

criminal(X) :- american(X), weapon(Y), sells(X,Y,Z), hostile(Z).

## پرولوگ

- اضافه کردن یک لیست به انتهای لیست دیگر برای تولید یک لیست جدید (عمل Append)

`append([ ], Y, Y) .`

`append([X|L], Y, [X|Z]) :- append(L, Y, Z) .`

- کوئری:

`append(A, B, [1, 2]) ?`

- پاسخ ها:

`A = [ ]      B = [ 1, 2 ]`

`A = [ 1 ]      B = [ 2 ]`

`A = [ 1, 2 ]      B = [ ]`

N. Razavi- AI course - 2006

38

## خصوصیات پرولوگ

- استنتاج روبه عقب، اول-عمق و از چپ به راست
- عملگرهایی برای محاسبات ریاضی مثلاً: `X is Y * Z + 3`
- مسندهایی با عوارض جانبی (مثلاً ورودی و خروجی)
- فرض دنیای بسته (> نقیض به عنوان شکست <)
- مثال: با داشتن

`alive(X) :- not dead(X).`

`alive(Joe)` موفق می شود اگر `not dead(Joe)` شکست بخورد

N. Razavi- AI course - 2006

37

## تبدیل به CNF

- Everyone who loves all animals is loved by someone:  
 $\forall x [\forall y \text{ Animal}(y) \Rightarrow \text{Loves}(x,y)] \Rightarrow [\exists y \text{ Loves}(y,x)]$

- 1. Eliminate biconditionals and implications

$\forall x [\neg \forall y \neg \text{Animal}(y) \vee \text{Loves}(x,y)] \vee [\exists y \text{ Loves}(y,x)]$

- 2. Move  $\neg$  inwards:  $\neg \forall x p \equiv \exists x \neg p$ ,  $\neg \exists x p \equiv \forall x \neg p$

$\forall x [\exists y \neg (\neg \text{Animal}(y) \vee \text{Loves}(x,y))] \vee [\exists y \text{ Loves}(y,x)]$

$\forall x [\exists y \neg \neg \text{Animal}(y) \wedge \neg \text{Loves}(x,y)] \vee [\exists y \text{ Loves}(y,x)]$

N. Razavi- AI course - 2006

40

## رزولوشن: خلاصه

- رزولوشن در منطق مرتبه اول

$l_1 \vee \dots \vee l_k, \quad m_1 \vee \dots \vee m_n$

$(l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n) \theta$

که  $\text{Unify}(l_i, \neg m_j) = \theta$

- فرض می شود که دو فراکرد استاندارد شده اند به گونه ای که شامل متغیر یکسانی نباشند
- مثال:

$$\frac{\neg \text{Rich}(x) \vee \text{Unhappy}(x)}{\text{Rich}(\text{Ken})}$$
  
$$\text{Unhappy}(\text{Ken})$$

with  $\theta = \{x/\text{Ken}\}$

N. Razavi- AI course - 2006

39

## تبدیل به CNF (ادامه)

3. Standardize variables: each quantifier should use a different one

4.  $\forall x [\exists y \text{ Animal}(y) \wedge \neg \text{Loves}(x,y)] \vee [\exists z \text{ Loves}(z,x)]$

4. Skolemize: a more general form of existential instantiation.

Each existential variable is replaced by a **Skolem function** of the enclosing universally quantified variables:

$\forall x [\text{Animal}(F(x)) \wedge \neg \text{Loves}(x,F(x))] \vee \text{Loves}(G(x),x)$

5. Drop universal quantifiers:

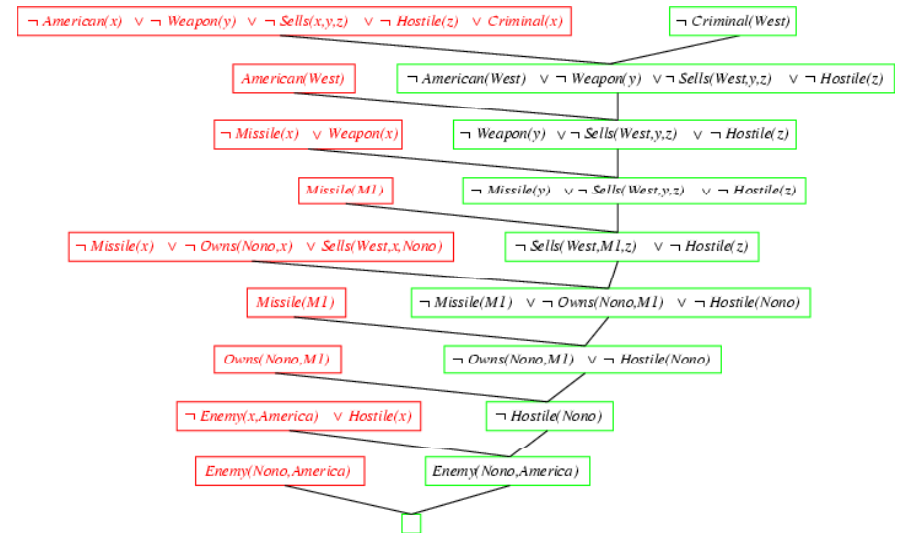
6.  $[\text{Animal}(F(x)) \wedge \neg \text{Loves}(x,F(x))] \vee \text{Loves}(G(x),x)$

N. Razavi- AI course - 2006

41

6. Distribute  $\vee$  over  $\wedge$  :

## اثبات بوسیله رزولوشن: مثال



N. Razavi- AI course - 2006

42