



دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

# عامل‌های منطقی

«هوش مصنوعی: یک رهیافت نوین»، فصل ۷

ارائه‌دهنده: سیده فاطمه موسوی

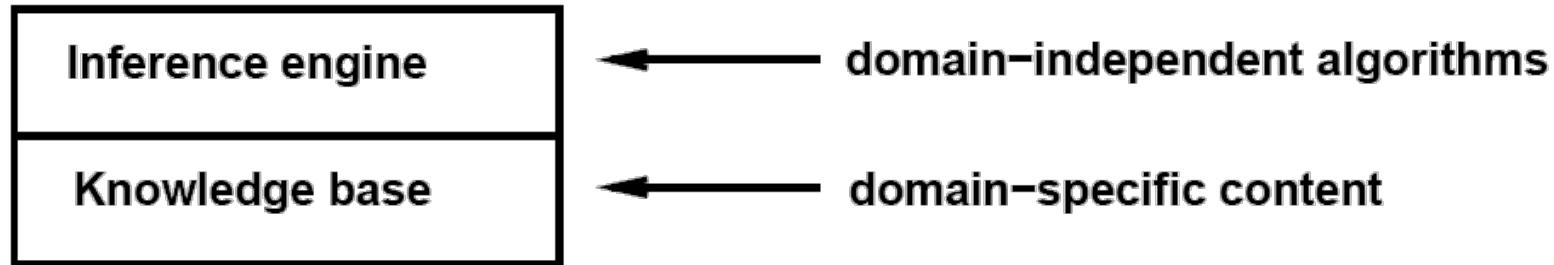
نیمسال اول ۱۳۹۸-۱۳۹۹

# رؤس مطالب

---

- عامل‌های مبنی بر دانش
- محیط و امپوس
- منطق- مدل‌ها و ایجاد
- منطق گزاره‌ای (بولین)
- همارزی، اعتبار و ارضاعیت
- قوانین استنتاج و اثبات تئوری
- رزولوشن
- زنجیره استنتاج رو به جلو
- زنجیره استنتاج رو به عقب

# عامل‌های مبتنی بر دانش



- جزء اصلی یک عامل مبتنی بر دانش، پایگاه دانش (Knowledge Base) است.
- پایگاه دانش = مجموعه ای از جملات (حقایق و قوانین) در زبان بازنمایی دانش
- به جملاتی که حقیقتی را در مورد دنیای عامل اظهار می‌کند axiom یا حقیقت می‌گویند.
- پایگاه دانش می‌تواند شامل دانش پیش‌زمینه (background) باشد یا شامل دانشی باشد که عامل به مرور یاد می‌گیرد.

# عامل‌های مبتنی بر دانش ...

---

- بر روی پایگاه دانش دو عمل اصلی انجام می‌شود:
- TELL: جمله‌ای را به پایگاه دانش اضافه می‌کند.
- ASK: اطلاعاتی را از KB استخراج می‌کند. (پرس‌وجو از دانسته‌ها)
- هر دو عمل فوق می‌توانند شامل استنتاج (Inference) باشند.
- یعنی به دست آوردن جملات جدید از قدیم
- استنتاج باید از این الزام اساسی پیروی کند که هرگاه سوالی از پایگاه دانش پرسیده شود، پاسخ باید از آن‌چه که قبلاً به پایگاه دانش گفته شده (TELL) نتیجه گردد.
- فرایند استنتاج نباید از خودش مفهوم جدیدی تولید نماید.

# عامل‌های مبتنی بر دانش ...

- طرح کلی برنامه عامل مبتنی بر دانش:

```
function KB-AGENT(percept) returns an action
  persistent: KB, a knowledge base
              t, a counter, initially 0, indicating time
  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  action  $\leftarrow$  ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t  $\leftarrow$  t + 1
  return action
```

# دو سطح بررسی عامل مبتنی بر دانش

---

- سطح دانش: عامل چه چیزی می‌داند، بدون توجه به چگونگی پیاده‌سازی
- سطح پیاده‌سازی: چه ساختمان‌داده‌هایی برای ذخیره دانش استفاده می‌کند و از چه الگوریتم‌هایی برای استنتاج استفاده می‌کند.

# دو روش ساخت یک عامل مبتنی بر دانش

---

- **روش اعلانی (declarative)**

- طراح عامل هر آنچه را عامل برای انجام وظایفش نیاز دارد یک به یک هر کدام به صورت یک جمله به عامل می‌گوید (TELL)
- در زمان اجرا عامل با انجام استدلال و استنتاج بر روی این دانش تصمیم می‌گیرد چه کاری انجام دهد.
- مزیت: امکان استفاده از دانش به شیوه‌هایی که طراح عامل پیش‌بینی نمی‌کرده است.

- **روش رویه‌ای (procedural)**

- طراح نحوه رسیدن به هر نتیجه خاص را به صورت یک رویه کد می‌کند.
- مزیت: سرعت اجرای بیشتر
- یک عامل موفق می‌تواند از ترکیب هر دو روش رویه‌ای و توصیفی ساخته شده باشد.

# دنیای وامپوس (Wumpus world)

Stench		Breeze	PIT
Wumpus	Breeze Stench Gold	PIT	Breeze
Stench		Breeze	
Player	Breeze	PIT	Breeze
START			

- معیار کارایی
  - طلا ۱۰۰۰ + ، مرگ -۱۰۰۰
  - به ازای هر قدم و ۱۰ - به ازای استفاده از تیر
- محیط
  - یک خانه مشبک  $4 \times 4$
  - عامل همواره از مربع [1,1] و جهت روبروی راست شروع می‌کند.
  - مکان طلا و وامپوس به‌طور تصادفی و در خانه‌ای غیر از خانه اول می‌باشد.
  - هر خانه به جز خانه اول با احتمال 0.2 گودال است.

# دنیای وامپوس (Wumpus world)

4

3

2

1

 Stench		 Breeze	PIT
	 Stench Gold	PIT	 Breeze
 Stench		 Breeze	
 START	 Breeze	PIT	 Breeze

- اقدام‌ها:

- حرکت مستقیم،  $90^{\circ}$  درجه چرخش به راست یا چرخش به چپ.
- حرکت به خانه‌ی دارای وامپوس و یا گودال منجر به مرگ می‌شود.
- حرکت رو به جلو زمانی که دیوار وجود دارد، بی‌اثر است.
- اقدام Grab، برای برداشتن طلا در همان خانه است.
- اقدام Shoot، تیر را در جهتی که عامل ایستاده است شلیک می‌کند.
- تیر تا زمانی که به وامپوس یا دیوار بخورد به راه خود ادامه می‌دهد.
- عامل فقط یک تیر دارد، بنابراین فقط اولین اقدام Shoot اثرگذار است.
- اقدام Climb برای زمانی که عامل از مربع  $[1,1]$  بیرون می‌آید.

# دنیای وامپوس (Wumpus world)

Stench		Breeze	PIT
Wumpus	Breeze Stench Gold	PIT	Breeze
Stench		Breeze	
Player	Breeze	PIT	Breeze
START			

- حس گرها:
- در خانه‌ای که وامپوس است و خانه‌های همسایه غیرقطری، عامل بوی بد دریافت می‌کند.
- در خانه‌های همسایه غیرقطری گودال، عامل نسیم خنکی را حس می‌کند.
- در خانه‌ای که طلا وجود دارد، عامل درخشش طلا را می‌بیند.
- هنگامی که عامل به دیوار بر می‌خورد، ضربه‌ای حس خواهد کرد.
- هنگامی که وامپوس کشته می‌شود، فریادی اندوهناک می‌کشد که صدایش در همه جای غار شنیده می‌شود.
- ادراکات با یک بردار پنج تایی بو، نسیم، درخشش، ضربه و جیغ نشان داده می‌شود.

[Stench, Breeze, None, None, None]

# مشخصات دنیای وامپوس

4			
3		  	
2			
1	 START		
	1	2	3
	4		

- مشاهده‌پذیر یا نیمه مشاهده‌پذیر؟
- قطعی یا غیرقطعی؟
- اپیزودیک یا مرحله‌ای؟
- ایستا یا پویا؟
- گستته یا پیوسته؟
- تک عاملی یا چند عاملی؟

# مثال دنیای وامپوس

A	= Agent
B	= Breeze
G	= Glitter, Gold
OK	= Safe square
P	= Pit
S	= Stench
V	= Visited
W	= Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1 A OK	2,1 OK	3,1	4,1

[None, None, None, None, None]

# مثال دنیای وامپوس

A	= Agent
B	= Breeze
G	= Glitter, Gold
OK	= Safe square
P	= Pit
S	= Stench
V	= Visited
W	= Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2 P?	3,2	4,2
OK			
1,1 V OK	2,1 A B OK	3,1 P?	4,1

[None, Breeze, None, None, None]

# مثال دنیای وامپوس

A	= Agent
B	= Breeze
G	= Glitter, Gold
OK	= Safe square
P	= Pit
S	= Stench
V	= Visited
W	= Wumpus

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

[Stench, None, None, None, None]

# مثال دنیای وامپوس

<b>A</b>	= <i>Agent</i>
<b>B</b>	= <i>Breeze</i>
<b>G</b>	= <i>Glitter, Gold</i>
<b>OK</b>	= <i>Safe square</i>
<b>P</b>	= <i>Pit</i>
<b>S</b>	= <i>Stench</i>
<b>V</b>	= <i>Visited</i>
<b>W</b>	= <i>Wumpus</i>

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

[Stench, Breeze, Glitter, None, None]

- منطق، یک زبان رسمی برای بازنمایی دانش است به طوری که بتوان از آن نتیجه‌گیری نمود.
- **نحو (Syntax)** ساختار جملات زبان را تعریف می‌کند.
- تمامی جملات خوش فرم(Well-formed) را مشخص می‌کند.
- **معنا (Semantic)** معنای جملات را تعریف می‌کند. درستی یک جمله را نسبت به هر دنیای ممکن تعریف می‌کند.
- مثال: زبان ریاضی
  - $x + 2 \geq y$  جمله است.
  - $x^2 + y \geq$  جمله نیست.
- جمله  $x + 2 \geq y$  در دنیایی با  $x = 7, y = 1$  درست و در دنیایی با  $x = 6$  نادرست است.

# مدل (Model)

---

- منطق دانان به هر دنیای ممکن یک مدل (مفاهیم انتزاعی ریاضی) می‌گویند.
- توجه: درستی یک جمله را نسبت به یک مدل می‌سنجیم، یعنی تعیین می‌کنیم آیا آن جمله به ازای مقادیری که در آن مدل به سمبول‌ها داده شده است درست است یا خیر.
- مثال: دنیایی با  $x$  مرد و  $y$  زن و جمله  $x+y=4$  را درنظر بگیرید.
- مدل‌های ممکن: همه انتساب‌های عدد حقیقی به  $x$  و  $y$  هستند.
- هر یک از این انتساب‌ها ارزش درستی جمله را تعیین می‌کند.
- جمله برای مدل‌هایی درست است که در آن مجموع تعداد افراد ۴ گردد مثلاً  $\{x=1, y=3\}$ .

# مدل (Model) ...

---

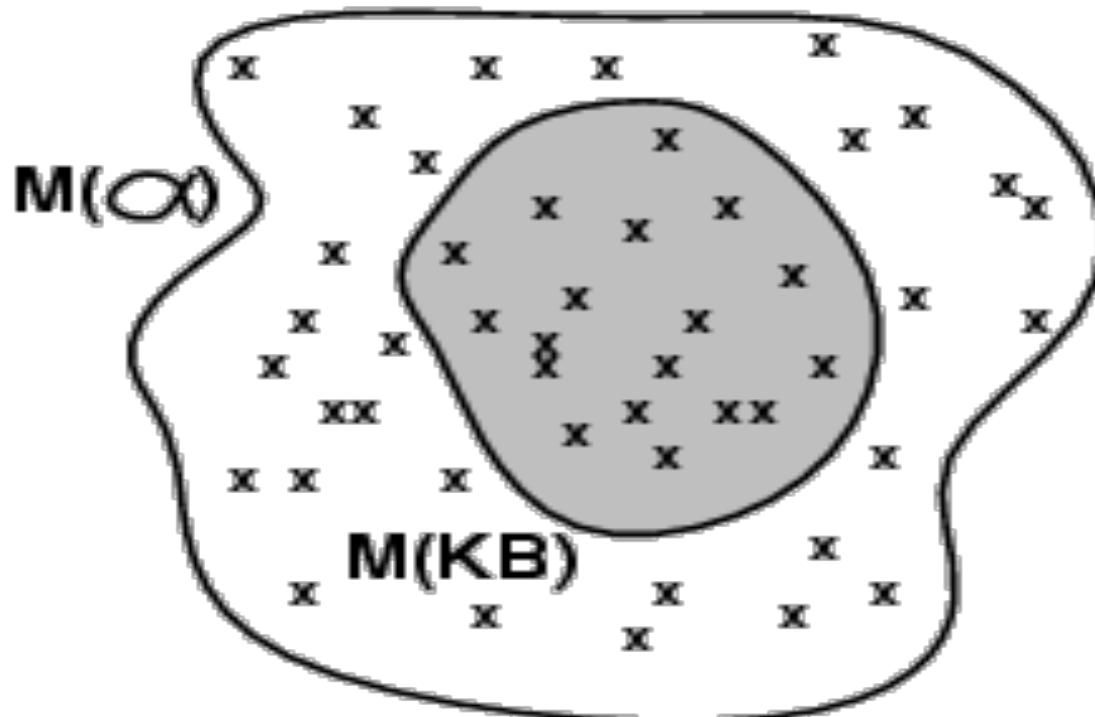
- "مدلی از  $\alpha$  است" یعنی "جمله  $\alpha$  در مدل  $m$  درست است". به عبارت دیگر یعنی جمله  $\alpha$  به ازای مقادیری که مدل  $m$  به سمبول‌ها می‌دهد برابر  $\text{true}$  می‌شود.
- If a sentence  $\alpha$  is true in model  $m$ , we say that  **$m$  satisfies  $\alpha$**  or sometimes  **$m$  is a model of  $\alpha$** .
- یک جمله می‌تواند مدل‌های زیادی داشته باشد. مجموعه تمام مدل‌های جمله  $\alpha$  را با  $M(\alpha)$  نشان می‌دهیم.
  - به عبارت دیگر  $M(\alpha)$  مجموعه تمام دنیاهایی است که  $\alpha$  در آن‌ها درست است.
  - $m \in M(\alpha)$  if  $\alpha$  is true in model  $m$

# استدلال منطقی: ایجاد (Entailment)

---

- ایجاد منطقی میان جملات یعنی یک جمله به طور منطقی از جمله دیگر استنباط شود.
- می‌گوییم جمله  $\alpha$  جمله  $\beta$  را ایجاد می‌کند  $\alpha \models \beta$  اگر و تنها اگر در هر مدلی که  $\alpha$  در آن درست باشد،  $\beta$  نیز در آن درست باشد.
- $\alpha \models \beta$  if and only if  $M(\alpha) \subseteq M(\beta)$ 
  - مثال:  $(x=0) \models (xy=0)$
  - یعنی در هر مدلی که  $x$  برابر با صفر است  $xy$  نیز صفر است.

# استدلال منطقی: ایجاد (Entailment)



- پایگاه دانش را می‌توان به صورت مجموعه‌ای از جملات و یا یک جمله که بیان‌کننده‌ی تمامی جملات است در نظر گرفت.

- پایگاه دانش در مدل‌هایی false است که با آن‌چه که عامل می‌داند در تناقض باشد.

- رابطه‌ی زیر برقرار است:  
$$KB \models \alpha \text{ iff } M(KB) \subseteq M(\alpha)$$

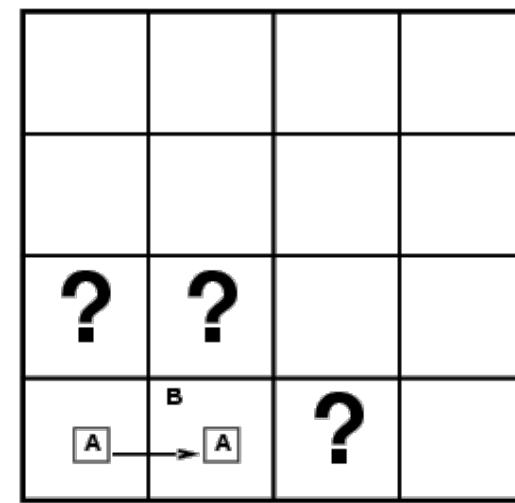
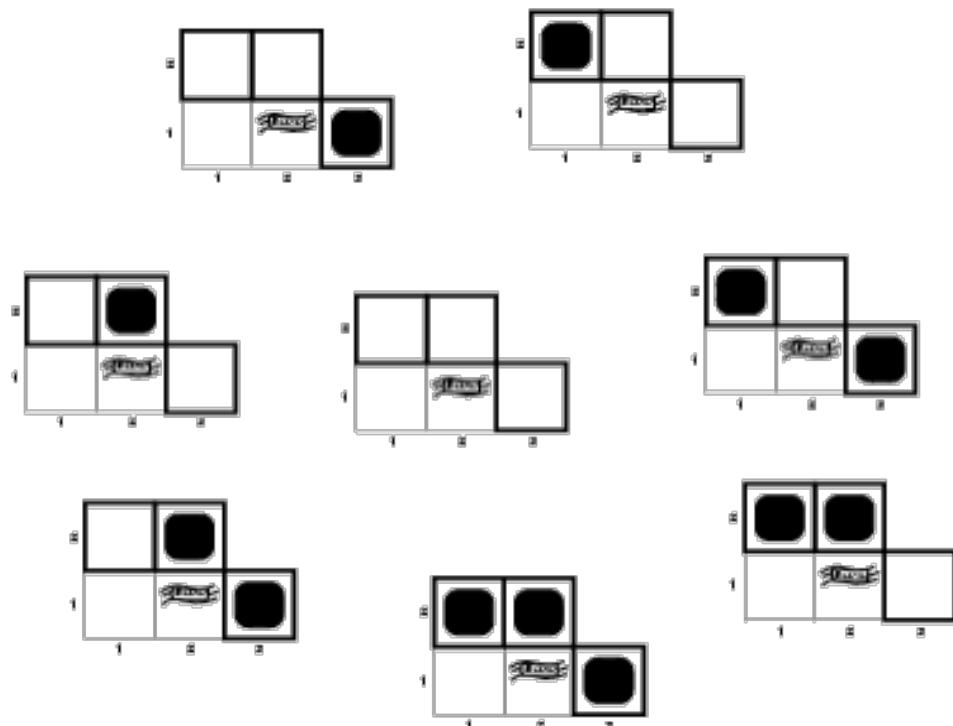
- مثال:

$KB = "A \text{ is red}" \text{ and } "B \text{ is blue}"$

$\alpha: "A \text{ is red}"$

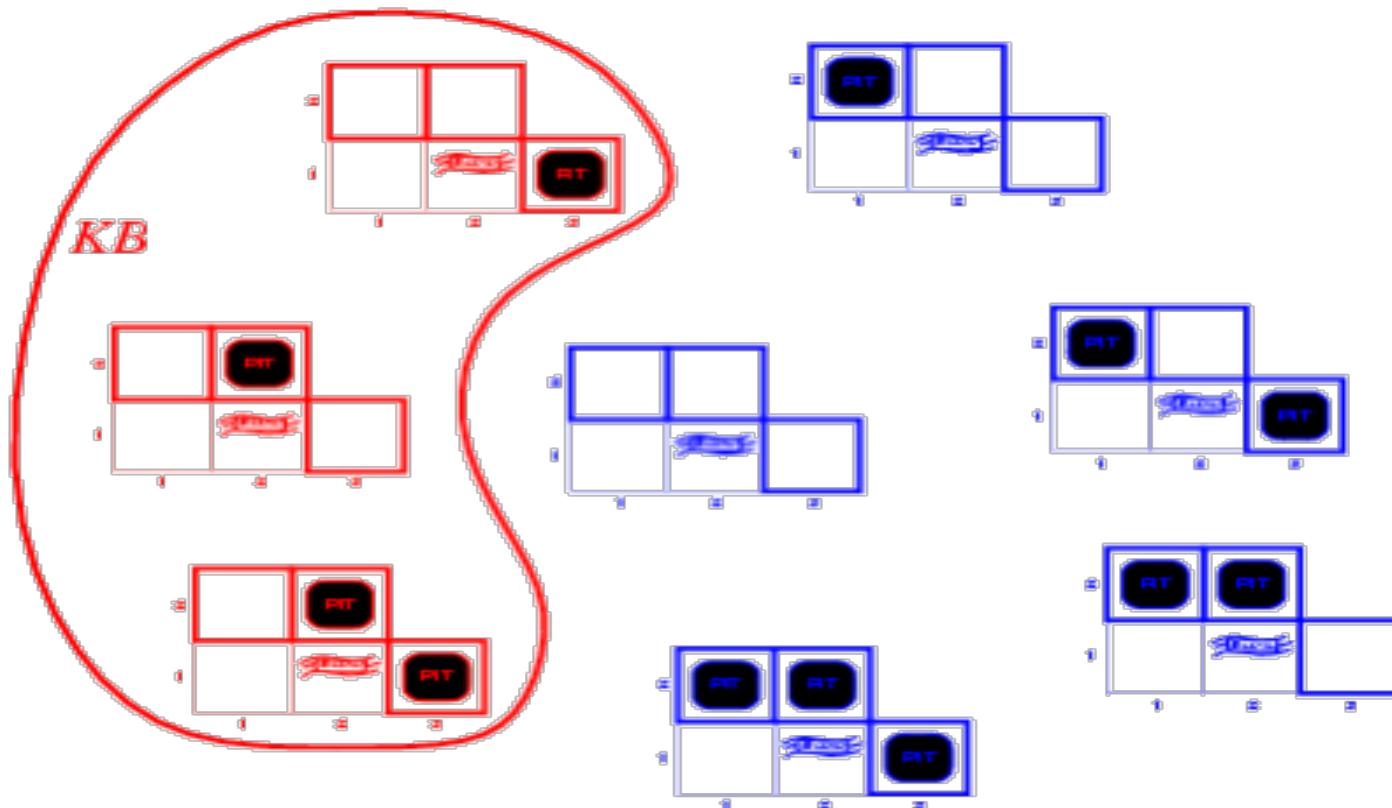
# ایجاد در دنیای وامپوس

- مدل‌های ممکن در ?ها را برای چاله‌ها در نظر بگیرید.
- سه انتخاب بولین: هشت مدل مختلف



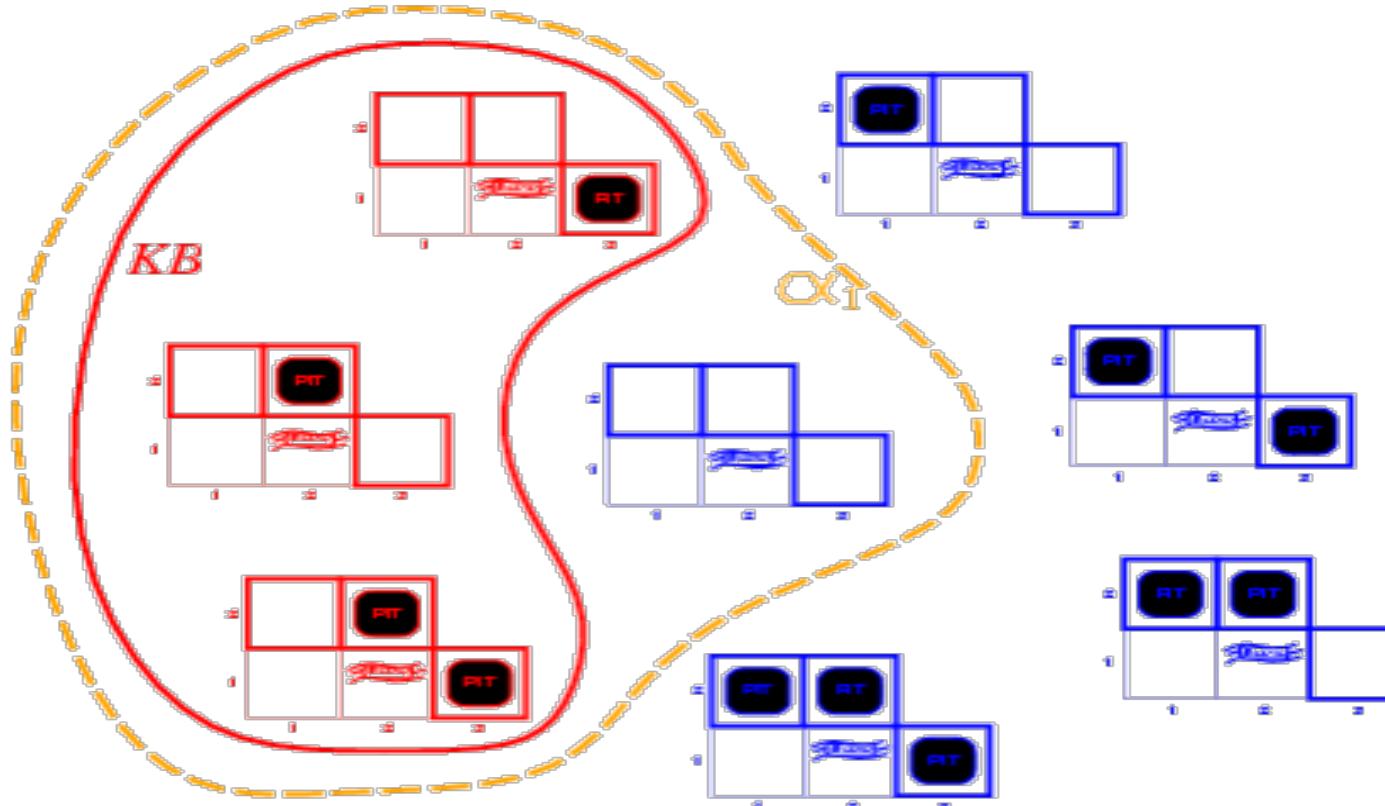
# ایجاد در دنیای وامپوس ...

- قبل از ادراک: KB شامل قوانین دنیای وامپوس است.
- ادراک: پس از آن که هیچ چیز در  $[1,1]$  درک نکرد به  $[2,1]$  می‌رود و نسیم احساس می‌کند.



$KB$  = wumpus-world rules + observations

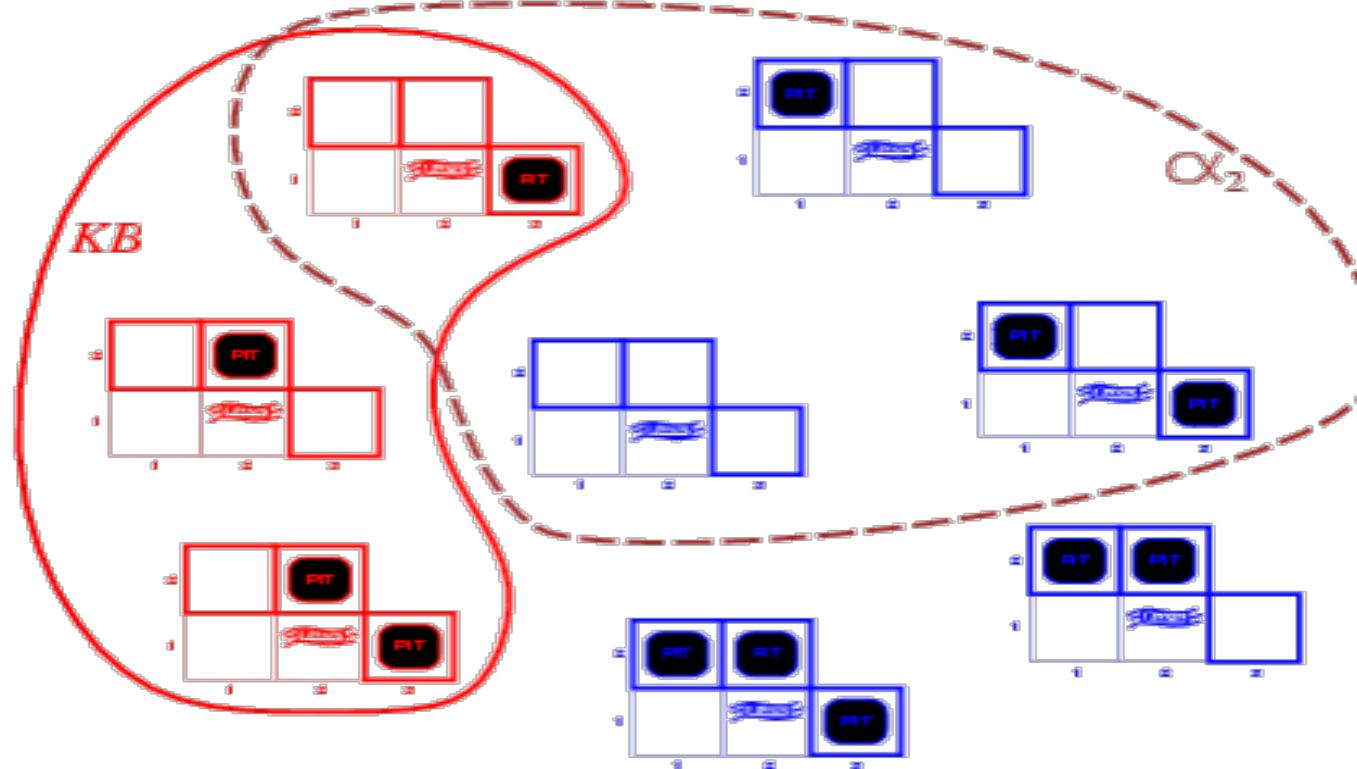
# ایجاد در دنیای وامپوس ...



$KB$  = wumpus-world rules + observations

$\alpha_1$  = “[1,2] is safe”  $\rightarrow M(KB) \subseteq M(\alpha_1) \rightarrow KB \models \alpha_1$   
(proved by model checking)

# ایجاد در دنیای وامپوس ...



$KB = \text{wumpus-world rules} + \text{observations}$

$\alpha_2 = "[2,2] \text{ is safe"} \rightarrow M(KB) \not\subseteq M(\alpha_2) \rightarrow KB \not\models \alpha_2$

# استنتاج (Inference)

- هدف از استنتاج این است که برای جمله‌ای مانند  $\alpha$  تعیین کنیم آیا  $KB \models \alpha$  برقرار است یا خیر.
- آیا یک جمله خاص مانند  $\alpha$  را می‌توان از  $KB$  نتیجه گرفت یا خیر.
- برای استنتاج الگوریتم‌های مختلفی می‌تواند وجود داشته باشد. اگر الگوریتم استنتاج  $\vdash$  بتواند جمله  $\alpha$  را از  $KB$  نتیجه بگیرد می‌نویسیم:  $KB \vdash_i \alpha$
- جمله  $\alpha$  از  $KB$  بوسیله رویه  $\vdash$  قابل استخراج می‌باشد.
- برای مثال الگوریتمی که در مثال قبل به کار بردیم را **وارسی مدل** (Model checking) می‌گویند. زیرا تمامی مدل‌های ممکن را برشماری می‌کند تا وارسی کند که آیا جمله در تمام مدل‌های درست  $KB$  درست است یا خیر ( $M(KB) \subseteq M(\alpha)$ ).

# استنتاج ... (Inference)

---

- دو ویژگی مهم الگوریتم‌های استنتاجی صحت و کامل بودن آنها است.
- الگوریتم استنتاجی که فقط جملات ایجابی را به دست آورد صحیح است.

## Soundness:

$i$  is sound if whenever  $KB \vdash_i \alpha$ , it is also true that  $KB \models \alpha$

- الگوریتم استنتاجی که بتواند هر جمله ایجاب شدنی را به دست آورد **کامل** است.

## Completeness:

$i$  is complete if whenever  $KB \models \alpha$ , it is also true that  $KB \vdash_i \alpha$

# منطق گزاره‌ای یا منطق بولی (Propositional Logic)

---

منطق گزاره‌ای ساده‌ترین منطق برای تشریح ایده‌های اساسی در مورد منطق و استدلال است.

# نحو منطق گزاره‌ای

- جملات اتمیک (Atomic): عناصر نحوی غیر قابل تجزیه هستند و از یک نماد گزاره‌ای تشکیل شده‌اند.
  - نمادهای گزاره‌ای را با حروف بزرگ P، Q و ... نشان می‌دهیم.
  - مثال:  $W_{1,3}$  یعنی وامپوس در خانه [1,3] است.
- هر کدام یک جمله هستند و به آن‌ها ثابت‌های گزاره‌ای می‌گویند.
  - True و False گزاره‌ای همواره درست و گزاره‌ای همواره غلط است.
- جملات مرکب (Complex): ترکیبی از جملات ساده‌تر بوسیله‌ی رابطه‌ای منطقی هستند.
  - رابطه‌ای منطقی: or، and، not و ...

# نحو منطق گزاره‌ای ...

- جملات مرکب:

- If  $S$  is a sentence,  $\neg S$  is a sentence (negation نقیض یا )
- If  $S_1$  and  $S_2$  are sentences,  $S_1 \wedge S_2$  is a sentence (conjunction عطفی یا )
- If  $S_1$  and  $S_2$  are sentences,  $S_1 \vee S_2$  is a sentence (disjunction فصلی یا )
- If  $S_1$  and  $S_2$  are sentences,  $S_1 \Rightarrow S_2$  is a sentence (implication شرطی یا )
- If  $S_1$  and  $S_2$  are sentences,  $S_1 \Leftrightarrow S_2$  is a sentence (biconditional دوشرطی یا )

- مثال:

$$(W_{1,3} \wedge P_{3,1}) \vee W_{2,2}$$

$$(W_{1,3} \wedge P_{3,1}) \Rightarrow \neg W_{2,2}$$

مقدم یا فرض (Antecedent)

تالی یا نتیجه (Consequent)

# دستور زبان BNF جملات در منطق گزاره‌ای

---

*Sentence* → Atomic Sentence | Complex Sentence

Atomic Sentence → True | False | P | Q | R | ...

Complex Sentence → (Sentence)

|  $\neg$  Sentence

| (Sentence  $\wedge$  Sentence)

| (Sentence  $\vee$  Sentence)

| (Sentence  $\Rightarrow$  Sentence)

| (Sentence  $\Leftrightarrow$  Sentence)

برای بدون ابهام کردن نحو از پرانتزگذاری و ترتیب اولویت رابطها استفاده می‌شود:

*Precedence*:  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$

# معانی منطق گزاره‌ای

- معانی، قواعد تعیین درستی یک جمله نسبت به یک مدل خاص هستند.
- یک مدل در منطق گزاره‌ای مقدار درستی (false یا true) را برای هر نماد گزاره‌ای مشخص می‌کند.
- مثال: برای نمادهای گزاره‌ای  $P_{1,2}$ ,  $P_{2,2}$  و  $P_{3,1}$  یک مدل ممکن می‌تواند به صورت زیر باشد:  
$$M_1 = \{P_{1,2} = \text{false}, P_{2,2} = \text{false}, P_{3,1} = \text{true}\}$$
- تعیین درستی یک جمله به صورت بازگشته انجام می‌شود. (بررسی درستی یک جمله مرکب به بررسی درستی جملات ساده‌تر تبدیل می‌شود.)
- می‌توان برای هر رابط (truth table) قواعدی به صورت جدول درستی (connective) خلاصه نمود که درستی هر جمله مرکب را به ازای تمام انتساب‌های ممکن مقادیر درستی به مولفه‌هایش تعیین می‌کند.

# معانی منطق گزاره‌ای - جدول درستی

$P$	$Q$	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

• قوانین درستی جملات ترکیبی در مدل  $m$ :

- $\neg P$  is true iff  $P$  is false in  $m$
- $P \wedge Q$  is true iff  $P$  is true and  $Q$  is true in  $m$
- $P \vee Q$  is true iff  $P$  is true or  $Q$  is true in  $m$
- $P \Rightarrow Q$  is true unless  $P$  is true and  $Q$  is false in  $m$
- $P \Leftrightarrow Q$  is true iff  $P$  and  $Q$  are both true or both false in  $m$

## تعريف پایگاه دانش

- اگر KB شامل گزاره‌های  $S_1$ ,  $S_2$ , ..., و  $S_n$  باشد می‌توان KB را معادل با ترکیب عطفی این گزاره‌ها در نظر گرفت.
- یعنی پایگاه دانش ما باید همیشه شامل جملات یا حقیقت‌های صحیح باشند.

$$KB = S_1 \wedge S_2 \wedge \dots \wedge S_n$$

# جملات دنیای وامپوس – پایگاه دانش ساده

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2 P?	3,2	4,2
OK			

1,1	2,1 V OK	3,1 B OK	4,1
-----	----------------	----------------	-----

- نمادهای زیر برای هر مکان  $[x,y]$  تعریف می‌شوند:

- $P_{x,y}$  is true if there is a pit in  $[x,y]$
- $W_{x,y}$  is true if there is a wumpus in  $[x,y]$ , dead or alive
- $B_{x,y}$  is true if the agent perceives a breeze in  $[x,y]$
- $S_{x,y}$  is true if the agent perceives a stench in  $[x,y]$

- قوانین کلی

- $R_1: \neg P_{1,1}$
- $R_2: B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
- $R_3: B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

- ادراکات

- $R_4: \neg B_{1,1}$
- $R_5: B_{2,1}$

# یک رویه استنتاج ساده

- هدف استنتاج منطقی: آیا برای جمله‌ای مانند  $\alpha$ , می‌توان گفت:  $KB \models \alpha$
- ساده‌ترین راه حل: پیاده‌سازی مستقیم تعریف ایجاد است یعنی برشماری مدل‌ها و وارسی این که آیا  $\alpha$  در هر مدلی که  $KB$  در آن درست است، درست می‌باشد.
- این کار از طریق بررسی سطرهای مختلف جدول درستی انجام می‌شود.
- مثال: پایگاه دانش زیر را در نظر بگیرید.

KB

$$R_1: \neg P_{1,1}$$

$$R_2: B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

$$R_3: B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$$

$$R_4: \neg B_{1,1}$$

$$R_5: B_{2,1}$$

$$\text{؟}KB \models \neg P_{1,2}$$

$$\text{؟}KB \models P_{2,2}$$

# استنتاج بواسیلهٔ جدول درستی - مثال

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$	$KB$
false	true	true	true	true	false	false						
false	false	false	false	false	false	true	true	true	false	true	false	false
:	:	:	:	:	:	:	:	:	:	:	:	:
false	true	false	false	false	false	false	true	true	false	true	true	false
false	true	false	false	false	false	true	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	false	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	true	true	true	true	true	true	<u>true</u>
false	true	false	false	true	false	false	true	false	false	true	true	false
:	:	:	:	:	:	:	:	:	:	:	:	:
true	false	true	true	false	true	false						

• آیا در هر سطری که  $KB \models \neg P_{1,2}$  ★ آیا  $\neg P_{1,2}$  درست است.

• آیا  $KB \models P_{2,2}$  ★ آیا  $P_{2,2}$  درست است.

# استنتاج بواسیلهٔ جدول درستی (یا وارسی مدل)

**function** TT-ENTAILS?( $KB, \alpha$ ) **returns** true or false

**inputs:**  $KB$ , the knowledge base, a sentence in propositional logic  
 $\alpha$ , the query, a sentence in propositional logic

$symbols \leftarrow$  a list of the proposition symbols in  $KB$  and  $\alpha$

**return** TT-CHECK-ALL( $KB, \alpha, symbols, \{ \}$ )

**function** TT-CHECK-ALL( $KB, \alpha, symbols, model$ ) **returns** true or false

**if** EMPTY?( $symbols$ ) **then**

**if** PL-TRUE?( $KB, model$ ) **then return** PL-TRUE?( $\alpha, model$ )

**else return** true // when  $KB$  is false, always return true

**else do**

$P \leftarrow$  FIRST( $symbols$ )

$rest \leftarrow$  REST( $symbols$ )

**return** (TT-CHECK-ALL( $KB, \alpha, rest, model \cup \{P = true\}$ )

**and**

        TT-CHECK-ALL( $KB, \alpha, rest, model \cup \{P = false\}$ ))

# استنتاج بوسیله جدول درستی – ویژگی‌ها

- الگوریتم استنتاج TT-ENTAILS یک برشماری بازگشتی از فضای محدود انتساب‌ها را برای متغیرها اجرا می‌کند.
- آیا این الگوریتم صحیح است؟
- بله زیرا به‌طور مستقیم تعریف ایجاب را پیاده‌سازی می‌کند.
- آیا این الگوریتم کامل است؟
- بله زیرا تعداد محدودی مدل برای آزمایش وجود دارد.
- پیچیدگی زمانی؟ ( $O(2^n)$ )
- اگر جمله  $\alpha$  و  $KB$  در مجموع  $n$  متغیر (سمبل) داشته باشد، تعداد مدل‌ها (یا همان تعداد سطرهای جدول درستی) برابر با  $2^n$  خواهد بود.
- پیچیدگی فضایی؟ ( $O(n)$ )
- زیرا از نوع اول عمق است.

# اثبات قضیه گزارهای

---

PROPOSITIONAL THEOREM PROVING

# استنتاج از طریق اثبات قضیه

---

- الگوریتم استنتاج در این روش سعی می‌کند جمله را اثبات کند.
- منظور از اثبات یک جمله، به کارگیری دنباله‌ای از قوانین استنتاج برای رسیدن از KB به آن جمله است.
- جستجوی برهان‌ها می‌تواند کارآمدتر از وارسی مدل باشد.
- می‌تواند از گزاره‌های نامرتبط صرف نظر کند.
- هنگامی که تعداد مدل‌ها زیاد است اما طول برهان‌ها کوتاه است، ایجاد با اثبات قضیه مفید است.

# همارزی منطقی

---

- دو جمله  $\alpha$  و  $\beta$  از نظر منطقی همارز هستند اگر و تنها اگر در مجموعه یکسانی از مدل‌ها درست باشند.
- با استفاده از همارزی‌های منطقی ما می‌توانیم به قوانین استنتاج زیادی دست یابیم.  
 $\alpha \equiv \beta$  if and only if  $\alpha \models \beta$  and  $\beta \models \alpha$

# همارزی منطقی ...

- $(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$  commutativity of  $\wedge$
- $(\alpha \vee \beta) \equiv (\beta \vee \alpha)$  commutativity of  $\vee$
- $((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$  associativity of  $\wedge$
- $((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$  associativity of  $\vee$
- $\neg(\neg\alpha) \equiv \alpha$  double-negation elimination
- $(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$  contraposition
- $(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$  implication elimination
- $(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$  biconditional elimination
- $\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$  De Morgan
- $\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$  De Morgan
- $(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$  distributivity of  $\wedge$  over  $\vee$
- $(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$  distributivity of  $\vee$  over  $\wedge$

# اعتبار (Validity)

---

- یک جمله معتبر (valid) است اگر در تمام مدل‌ها درست باشد.
- به عبارت دیگر یک جمله معتبر است اگر به ازای تمام سطرهای جدول درستی، آن جمله درست باشد.
- مثال:  $P \vee \neg P$
- نام دیگر: تاتولوژی (tautology)، بدیهیات (deduction theorem)
- قضیه قیاس (deduction theorem)

$\alpha \models \beta$  if and only if  $\alpha \Rightarrow \beta$  is valid

# ارضایذیری (Satisfiability)

- یک جمله ارضایذیر (satisfiable) است اگر در بعضی از مدل‌ها درست باشد.
  - به عبارت دیگر اگر حداقل به ازای یکی از سطرهای جدول درستی برابر با true باشد.
  - مثال: جمله  $P \vee Q$  ارضاشدنی است چون سه مدل وجود دارد که در آن‌ها درست است.
  - یک جمله ارضانایذیر است، هرگاه در هیچ مدلی درست نباشد.
  - به عبارت دیگر اگر به ازای تمام سطرهای جدول درستی برابر با false باشد.
  - مثال:  $\neg P \wedge P$
  - جمله  $\alpha$  معتبر (همیشه درست) است اگر و تنها اگر  $\neg \alpha$  ارضانشدنی باشد.
  - رابطه ایجاب و ارضانایذیری (برهان خلف)
- $\alpha \models \beta$  if and only if  $\alpha \wedge \neg \beta$  is unsatisfiable

# چند قانون استنتاج

- قوانین استنتاج می‌توانند برای حصول زنجیره‌ای از نتایج که به هدف مطلوب منجر می‌شوند، به کار برد.  
کار برد.

$$\frac{\alpha \Rightarrow \beta, \alpha}{\beta} \text{ :Modus Ponens}$$

- مثال:

$$\frac{(WumpusAhead \wedge WumpusAlive) \Rightarrow Shoot, (WumpusAhead \wedge WumpusAlive)}{Shoot}$$

$$\frac{\alpha \wedge \beta}{\alpha} \text{ :(And Elimination)}$$

- مثال:

$$\frac{(WumpusAhead \wedge WumpusAlive)}{WumpusAlive}$$

# چند قانون استنتاج ...

- تمامی همارزی‌های منطقی می‌توانند به عنوان قواعد استنتاجی به کار برد شوند.
- همارزی حذف دوشرطی (Biconditional Elimination)

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

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

# استفاده از قوانین استنتاج و همارزی در دنیای وامپوس

- آیا می‌توان با استفاده از قوانین موجود در پایگاه دانش زیر اثبات کرد در  $[1,2]$  گودالی نیست؟

$$R_1: \neg P_{1,1}$$

$$R_2: B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

$$R_3: B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$$

$$R_4: \neg B_{1,1}$$

$$R_5: B_{2,1}$$

$$\neg P_{1,2}$$

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

اعمال حذف دوشرطی بر روی  $R_2$

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

اعمال حذف And بر روی  $R_6$

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

$(P \Rightarrow Q) \equiv (\neg P \Rightarrow \neg Q)$

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

قانون Modus Ponens با قوانین  $R_4$  و  $R_8$

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

قانون De Morgan

# تست ۱

(مهندسی کامپیوتر دولتی ۹۵)

کدام یک از جملات زیر نادرست است؟

- ۱) اگر  $\gamma \Rightarrow \alpha \wedge \beta$  همیشه درست باشد، آن‌گاه حداقل یکی از دو جمله  $\gamma$  و  $\alpha \wedge \beta$  همیشه درست است. ✓
- ۲) اگر  $\gamma \Rightarrow \alpha \vee \beta$  همیشه درست باشد، آن‌گاه هر دو جمله  $\gamma$  و  $\alpha \vee \beta$  همیشه درست هستند.
- ۳) یک جمله همیشه درست اگر و فقط اگر  $\text{True} \Rightarrow \alpha$  همیشه درست باشد.
- ۴) یک جمله  $\alpha \Rightarrow \beta$  همیشه درست است اگر و فقط اگر  $\alpha \wedge \neg \beta$  غیرقابل ارضاء (unsatisfiable) باشد.

# یافتن اثبات به صورت یک مسئله جستجو

---

- **INITIAL STATE:** the initial knowledge base.
- **ACTIONS:** the set of actions consists of all the inference rules applied to all the sentences that match the top half of the inference rule.
- **RESULT:** the result of an action is to add the sentence in the bottom half of the inference rule.
- **GOAL:** the goal is a state that contains the sentence we are trying to prove.

# خاصیت یکنواختی (Monotonicity)

---

- با اضافه شدن جمله‌ای جدید به  $KB$ ، نتایجی که قبلاً از جملات قبلی  $KB$  گرفته شده‌اند معتبر باقی می‌مانند.

if  $KB \models \alpha$  then  $KB \wedge \beta \models \alpha$

- به عبارت دیگر با اضافه شدن جمله جدید به  $KB$  ممکن نیست یکی از جملات قبلی  $KB$  نامعتبر شود و از  $KB$  حذف شود، بلکه برعکس، اندازه  $KB$  رشد می‌کند.
- قواعد استنتاج می‌توانند تا هر زمان که مقدمه‌ای مناسب در پایگاه دانش یافت می‌شوند، به کار برده شوند.

# رزولوشن (Resolution)

---

- قواعد استنتاج گفته شده صحیح هستند.
- با این وجود، آیا هر مجموعه‌ای از قوانین برای استنتاج، کامل است؟
- خیر، در صورتی که قواعد استنتاج ناکافی باشد هدف دست‌نیافتنی خواهد بود.
- برای مثال اگر قانون حذف دوشرطی را در مثال دنیای وامپوس نادیده بگیریم، اثبات انجام نخواهد شد.
- **قوانین رزولوشن** را می‌توان به تنهایی در هر یک از الگوریتم‌های جستجوی کامل استفاده کرد و یک الگوریتم جستجوی کامل به دست آورد.

# رزولوشن ...

- قانون رزولوشن واحد (Unit): اگر  $l_i$  و  $m$  لیترال‌های نقیض هم باشند آن‌گاه

$$\frac{l_1 \vee \dots \vee l_k, m}{l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \dots \vee l_k}$$

- قانون رزولوشن کامل (Full): اگر  $l_i$  و  $m_j$  لیترال‌های نقیض هم باشند آن‌گاه

$$\frac{l_1 \vee \dots \vee l_k, \quad m_1 \vee \dots \vee m_n}{l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n}$$

- آیا قانون رزولوشن صحیح است؟ چرا؟

# کاربرد رزولوشن در اثبات

- یک اثبات‌کننده قضیه‌ی مبتنی بر رزولوشن می‌تواند برای هر جمله‌ی  $\alpha$  و  $\beta$  در منطق گزاره‌ای تصمیم بگیرد که آیا  $\alpha \models \beta$  یا خیر.
  - برای این منظور باید به دو بحث بپردازیم:
    - شکل نرمال عطفی (Conjunctive Normal Form - CNF)
    - ارائه یک الگوریتم مبتنی بر رابطه ایجاب و ارضانایپذیری (برهان خلف)
- $\alpha \models \beta$  if and only if  $\alpha \wedge \neg \beta$  is unsatisfiable

# شکل نرمال عطفی (CNF)

- جمله‌ای در CNF است که به صورت ترکیب عطفی یک یا چند عبارت ساده نوشته شده باشد به‌طوری‌که هر یک از این عبارات ساده، یک نماد یا ترکیب فصلی چند نماد باشد.

$CNF Sentence \rightarrow Clause_1 \wedge \dots \wedge Clause_n$

$Clause \rightarrow Literal_1 \vee \dots \vee Literal_m$

$Literal \rightarrow Symbol \mid \neg Symbol$

$Symbol \rightarrow P \mid Q \mid R \mid \dots$

- برای مثال:

$$(P_1 \vee P_2) \wedge (\neg P_3 \vee P_4 \vee P_5) \wedge P_6$$

# مراحل تبدیل یک گزاره به شکل نرمال عطفی (CNF)

---

۱- حذف  $\alpha \leftrightarrow \beta$  با استفاده از همارزی  $\alpha \leftrightarrow \beta \equiv (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$

۲- حذف  $\alpha \Rightarrow \beta$  با استفاده از همارزی  $\alpha \Rightarrow \beta \equiv (\neg \alpha \vee \beta)$

۳- انتقال  $\neg$  به داخل پرانتز با استفاده از همارزی‌های زیر

$$\neg(\neg \alpha) \equiv \alpha$$

$$\neg(\alpha \vee \beta) \equiv \neg \alpha \wedge \neg \beta$$

$$\neg(\alpha \wedge \beta) \equiv \neg \alpha \vee \neg \beta$$

۴- استفاده از توزیع‌پذیری  $\wedge$  نسبت به  $\vee$

• مثال: جمله  $A \leftrightarrow (B \vee C)$  را به شکل CNF درآورید.

# الگوریتم رزولوشن

- $\neg \alpha$  را به فرم CNF نوشته و هر دو بندی که شامل لیترال‌های مکمل باشند را در هم resolve کرده تا در صورت وجود بند جدیدی حاصل شود. این فرایند تا زمان رخ دادن یکی از حالات زیر ادامه می‌یابد:
  - هیچ بند جدیدی را نتوان با انجام رزولوشن ایجاد کرد. در این حالت  $\alpha$  را نمی‌توان استنباط کرد.
  - با انجام قاعده رزولوشن به عبارت تهی برسیم. در این صورت عبارت  $\neg \alpha \wedge KB$  برابر با False است. یعنی  $\alpha$  را نمی‌توان ایجاب کرد.
- توجه: بند تهی با False همارز است. بند تهی تنها از حل دو بند واحد مکمل مانند  $P$  و  $\neg P$  ناشی می‌شود.

# الگوریتم رزولوشن - مثال

- فرض کنید پایگاه دانش به شکل زیر باشد:
- در  $[1,1]$  هیچ نسیمی وجود ندارد.
- اگر عامل در  $[1,1]$  باشد و نسیمی حس کند آن‌گاه ممکن است گودالی در همسایه‌های غیرقطری آن وجود داشته باشد.

$$KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$$

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1	2,1	3,1	4,1

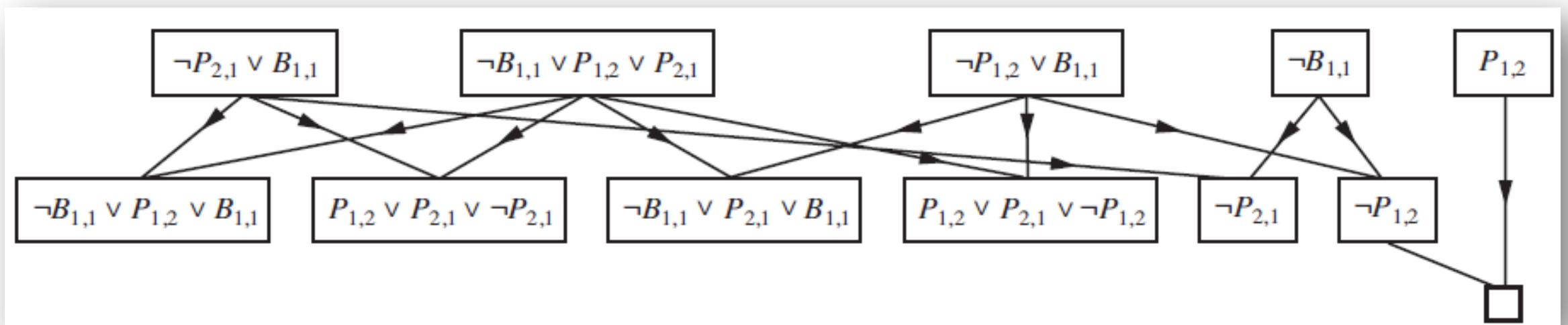
- آیا می‌توان نتیجه گرفت در  $[1,2]$  گودال نیست؟

$$\alpha = \neg P_{1,2}$$

# الگوریتم رزولوشن – مثال ...

- تبدیل به شکل نرمال عطفی

$$KB = (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1}) \wedge (\neg B_{1,1}) \wedge P_{1,2}$$



# الگوریتم رزولوشن ...

**function** PL-RESOLUTION( $KB, \alpha$ ) **returns** *true* or *false*

**inputs:**  $KB$ , the knowledge base, a sentence in propositional logic  
 $\alpha$ , the query, a sentence in propositional logic

*clauses*  $\leftarrow$  the set of clauses in the CNF representation of  $KB \wedge \neg\alpha$

*new*  $\leftarrow \{ \}$

**loop do**

**for each** pair of clauses  $C_i, C_j$  **in** *clauses* **do**

*resolvents*  $\leftarrow$  PL-RESOLVE( $C_i, C_j$ )

**if** *resolvents* contains the empty clause **then return** *true*

*new*  $\leftarrow$  *new*  $\cup$  *resolvents*

**if** *new*  $\subseteq$  *clauses* **then return** *false*

*clauses*  $\leftarrow$  *clauses*  $\cup$  *new*

یک الگوریتم استنتاج صحيح و كامل است.

## تست ۲

پایگاه دانش زیر مفروض است. کدام یک از گزینه‌های زیر با استفاده از روش رزولوشن از این پایگاه دانش قابل استنتاج است؟  
(مهندسی کامپیوتر دولتی ۹۲)

S (۱) ✓

W (۲)

V (۳)

T (۴)

P

$V \vee T$

$\neg P \vee U$

$R \vee \neg Q$

$V \Rightarrow W$

$P \Rightarrow Q$

$S \Rightarrow (U \vee T)$

$(P \wedge R) \Rightarrow S$

# شکل نرمال هورن (HNF)

- یک بند هورن، ترکیب فصلی لیترال‌هایی است که حداقل یکی از آن‌ها مثبت است.
- یک بند معین، ترکیب فصلی لیترال‌هایی که فقط یک لیترال آن مثبت است.
- به بندهای هورن که لیترال مثبت ندارند، بند هدف می‌گویند.

$\neg A \vee \neg B \vee \neg C$       **بند هدف**

$A \vee \neg B$       **بند معین**

$\neg A \vee \neg B \vee C$       **بند معین**

$A$       **بند معین**

- بندهای هورن نسبت به رزولوشن بسته‌اند. یعنی اگر قانون رزولوشن را بر روی دو بند هورن اعمال کنیم، یک بند هورن خواهیم داشت.

# شکل نرمال هورن (... (HNF

---

Clause  $\rightarrow$  Literal<sub>1</sub>  $\vee \dots \vee$  Literal<sub>*m*</sub>

Literal  $\rightarrow$  Symbol |  $\neg$ Symbol

Symbol  $\rightarrow$  P | Q | R | ...

HornClauseForm  $\rightarrow$  DefiniteClauseForm | GoalClauseForm

DefiniteClauseForm  $\rightarrow$  (Symbol<sub>1</sub>  $\wedge \dots \wedge$  Symbol<sub>*l*</sub>)  $\Rightarrow$  Symbol

GoalClauseForm  $\rightarrow$  (Symbol<sub>1</sub>  $\wedge \dots \wedge$  Symbol<sub>*l*</sub>)  $\Rightarrow$  False

- هر پایگاه دانش را می‌توان به شکل ترکیب عطفی جملات به فرم HNF نوشت.

# دلایل استفاده از بند معین در پایگاه دانش

۱- هر بند معین می‌تواند به صورت یک استلزم که مقدم آن ترکیب عطفی لیترال‌های مثبت و نتیجه‌ی آن یک لیترال مثبت است، نوشته شود.

A

$\neg A \vee \neg B \vee C$

True  $\Rightarrow A$

$A \wedge B \Rightarrow C$

۲- استنتاج در بندهای معین می‌تواند از طریق الگوریتم‌های زنجیرهای روبه‌جلو و زنجیرهای پس‌رو انجام شود.

۳- زمان تصمیم‌گیری در مورد ایجاد در بندهای معین می‌تواند بر حسب اندازه‌ی پایگاه دانش به صورت خطی باشد.

# الگوریتم زنجیره‌ای رو به جلو

یک پایگاه دانش  $KB$  به صورت ترکیب عطفی عبارات معین و لیترال  $\alpha$  داده شده است می‌خواهیم نشان دهیم آیا می‌توان  $\alpha$  را از  $KB$  ایجاد کرد؟

- ۱- تمام حقایق را به عنوان مجموعه لیترال‌های درست (agenda) در نظر می‌گیرد.
- ۲- به ازای هر قانون شرطی که تمام لیترال‌های موجود در مقدم آن شرط در مجموعه inferred باشد، لیترال موجود در تالی آن شرط به agenda اضافه می‌شود.
- ۳- مرحله ۲ تکرار می‌شود تا یکی از دو مورد زیر رخ دهد:
  - لیترال  $\alpha$  به agenda اضافه شود در این صورت  $\alpha$  از  $KB$  ایجاد می‌شود.
  - نتوان لیترال جدیدی به agenda اضافه کرد. در این صورت  $KB$  مستلزم  $\alpha$  نیست.

# الگوریتم زنجیره‌ای روبره جلو ...

**function** PL-FC-ENTAILS?( $KB, q$ ) **returns** *true* or *false*

**inputs:**  $KB$ , the knowledge base, a set of propositional definite clauses  
 $q$ , the query, a proposition symbol

$count \leftarrow$  a table, where  $count[c]$  is the number of symbols in  $c$ 's premise

$inferred \leftarrow$  a table, where  $inferred[s]$  is initially *false* for all symbols

$agenda \leftarrow$  a queue of symbols, initially symbols known to be true in  $KB$

**while**  $agenda$  is not empty **do**

$p \leftarrow \text{POP}(agenda)$

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

**if**  $inferred[p] = \text{false}$  **then**

$inferred[p] \leftarrow \text{true}$

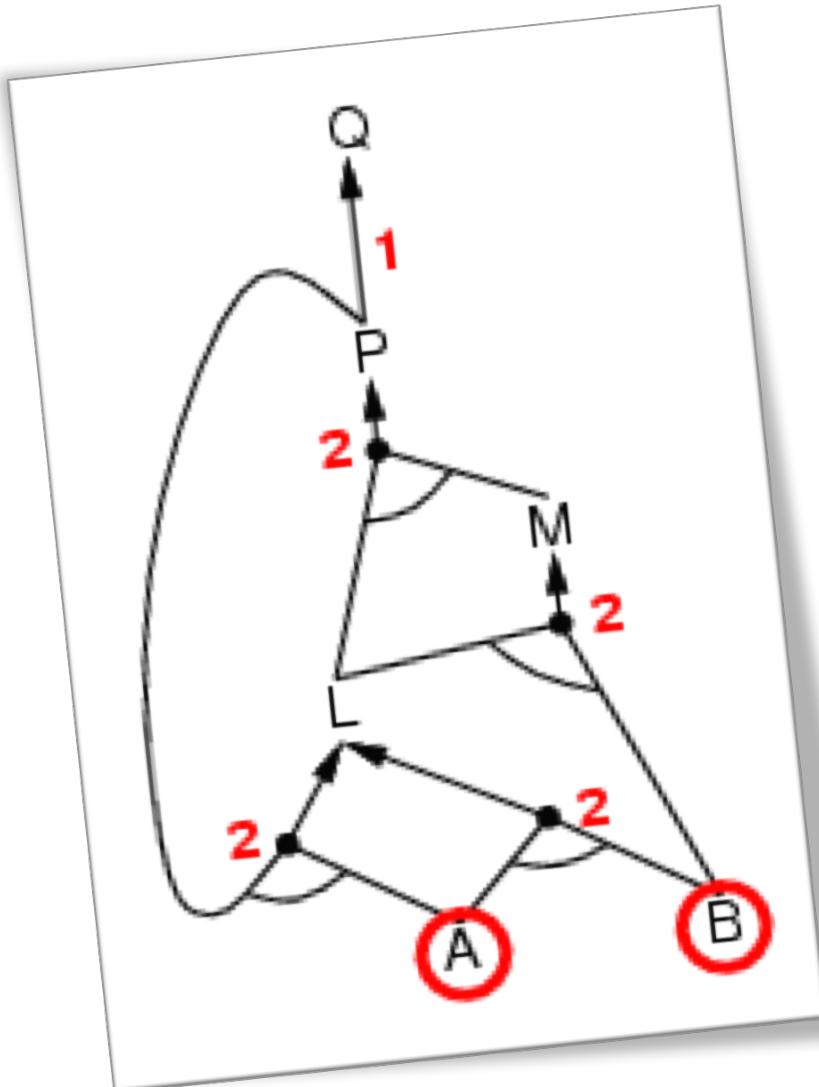
**for each** clause  $c$  in  $KB$  where  $p$  is in  $c.\text{PREMISE}$  **do**

            decrement  $count[c]$

**if**  $count[c] = 0$  **then** add  $c.\text{CONCLUSION}$  to  $agenda$

**return** *false*

# استنتاج روبره جلو - مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

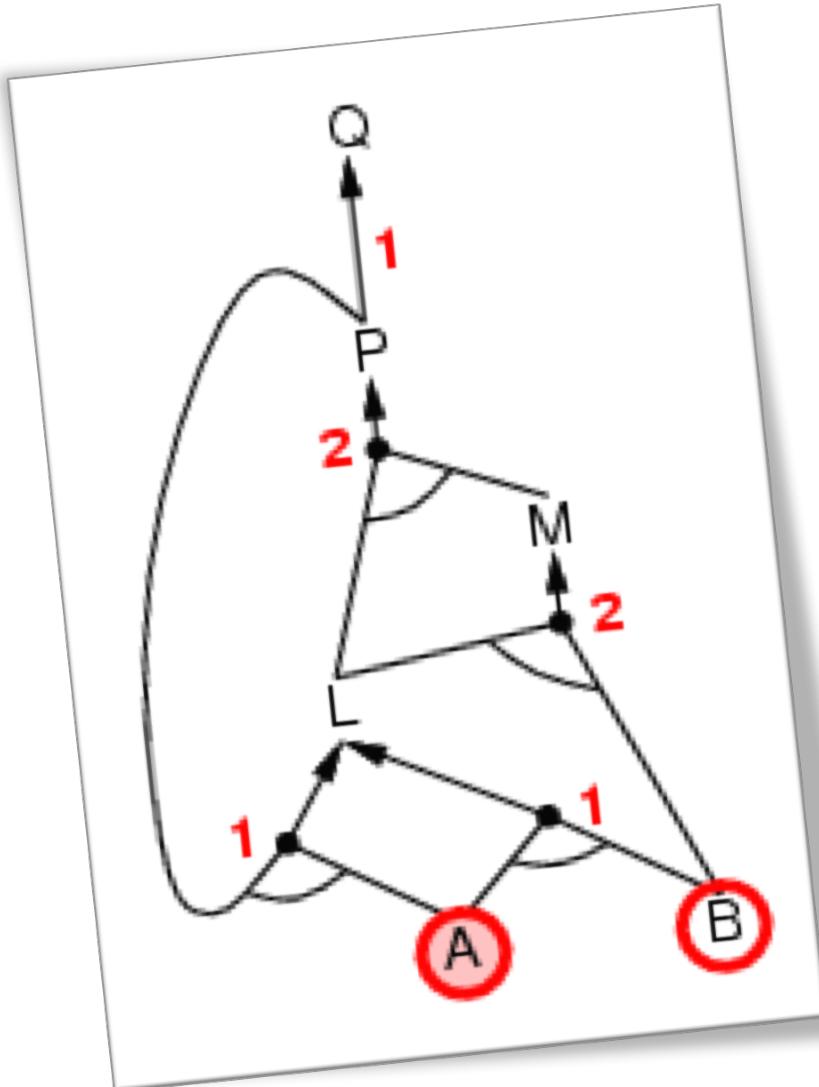
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره جلو - مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

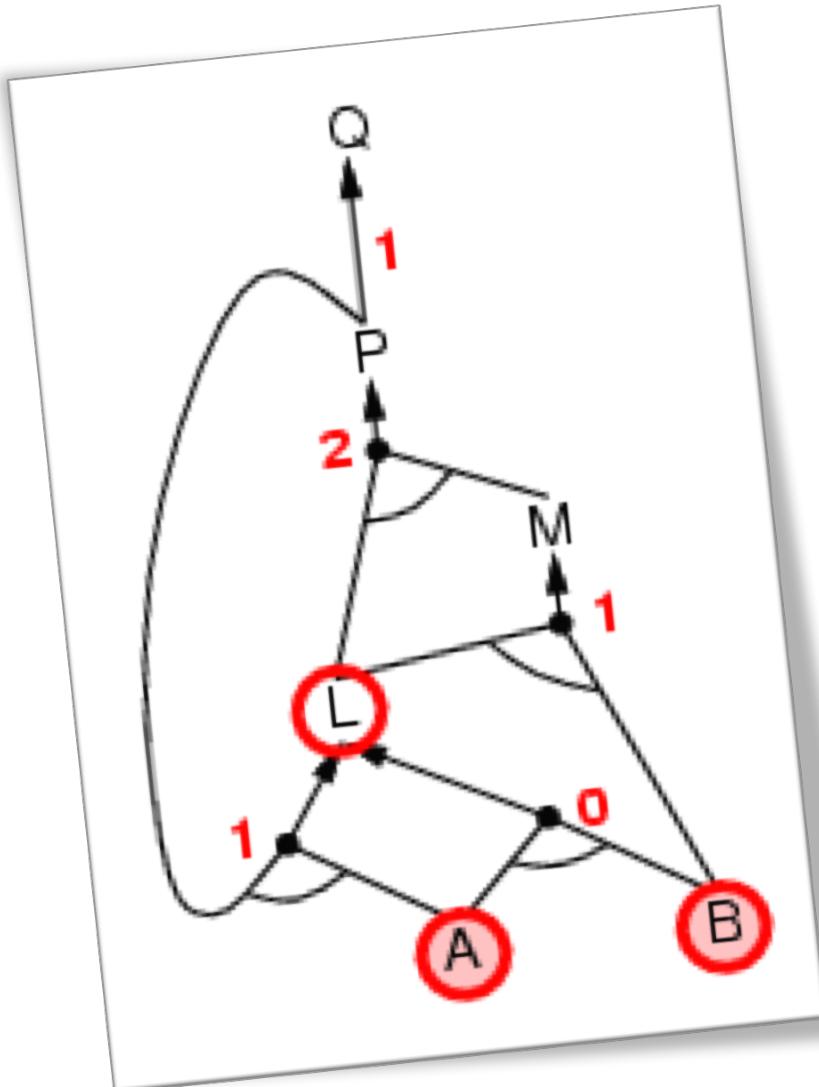
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبه جلو – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

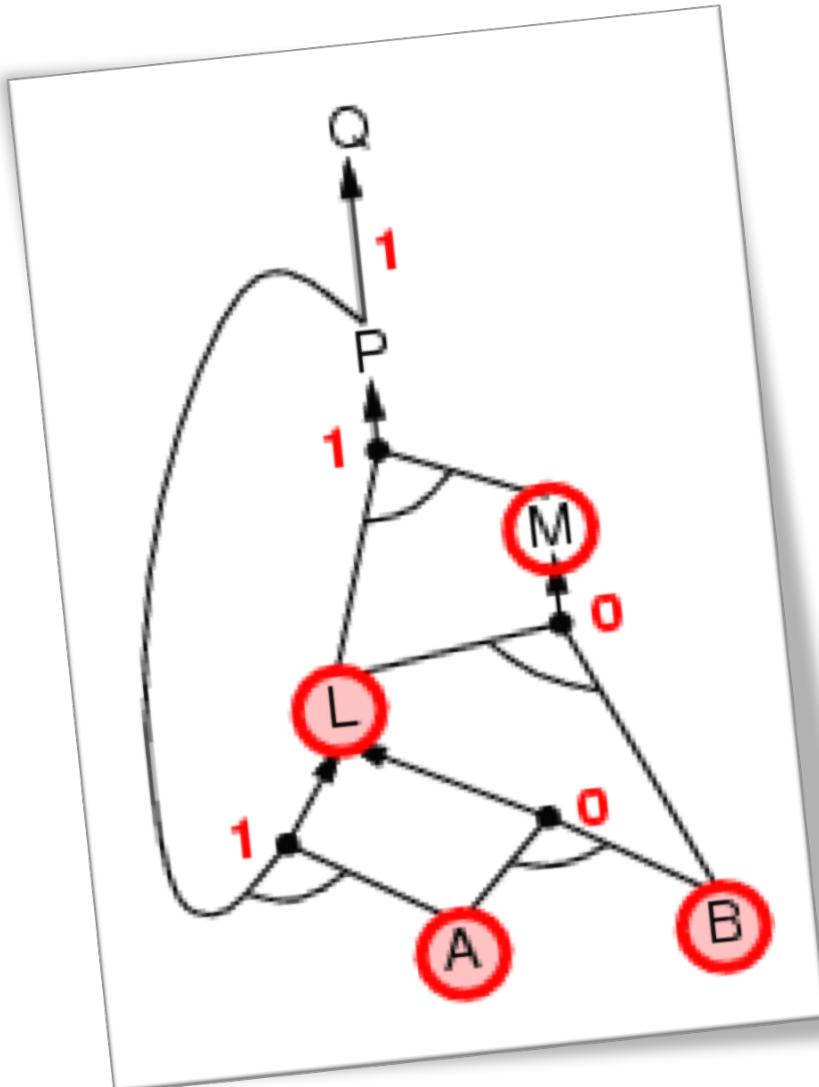
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره جلو - مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

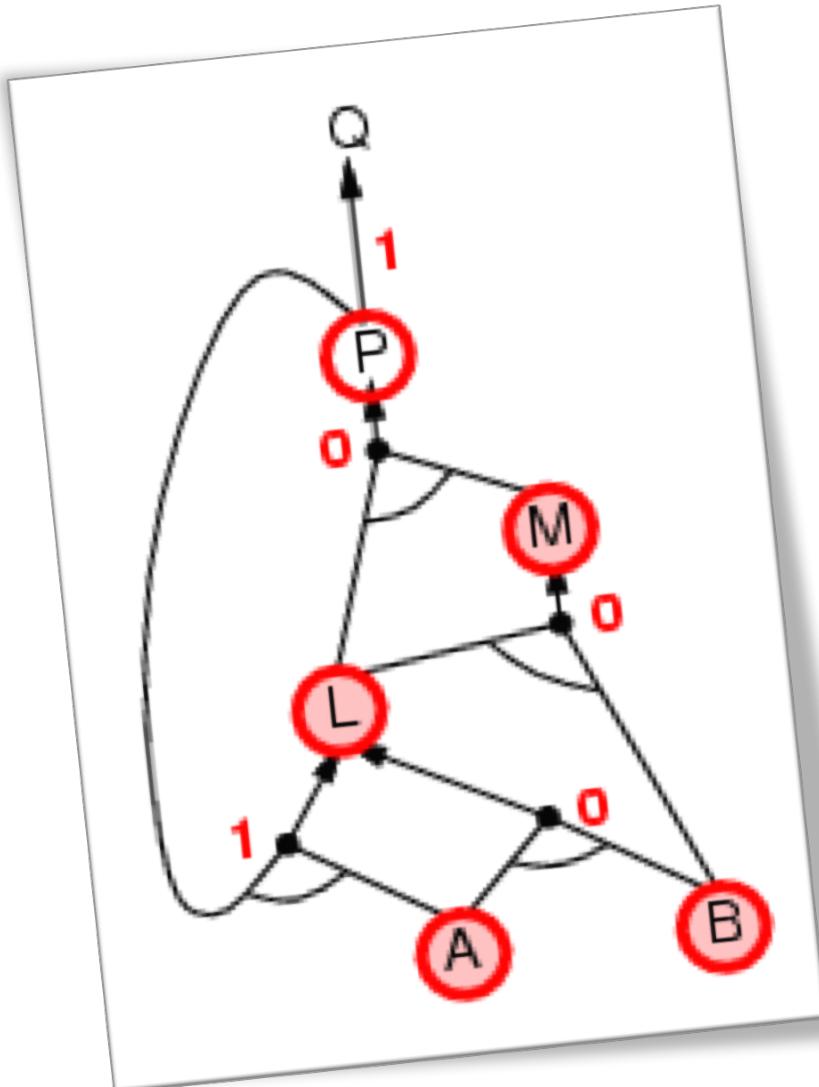
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B

# استنتاج روبره جلو - مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

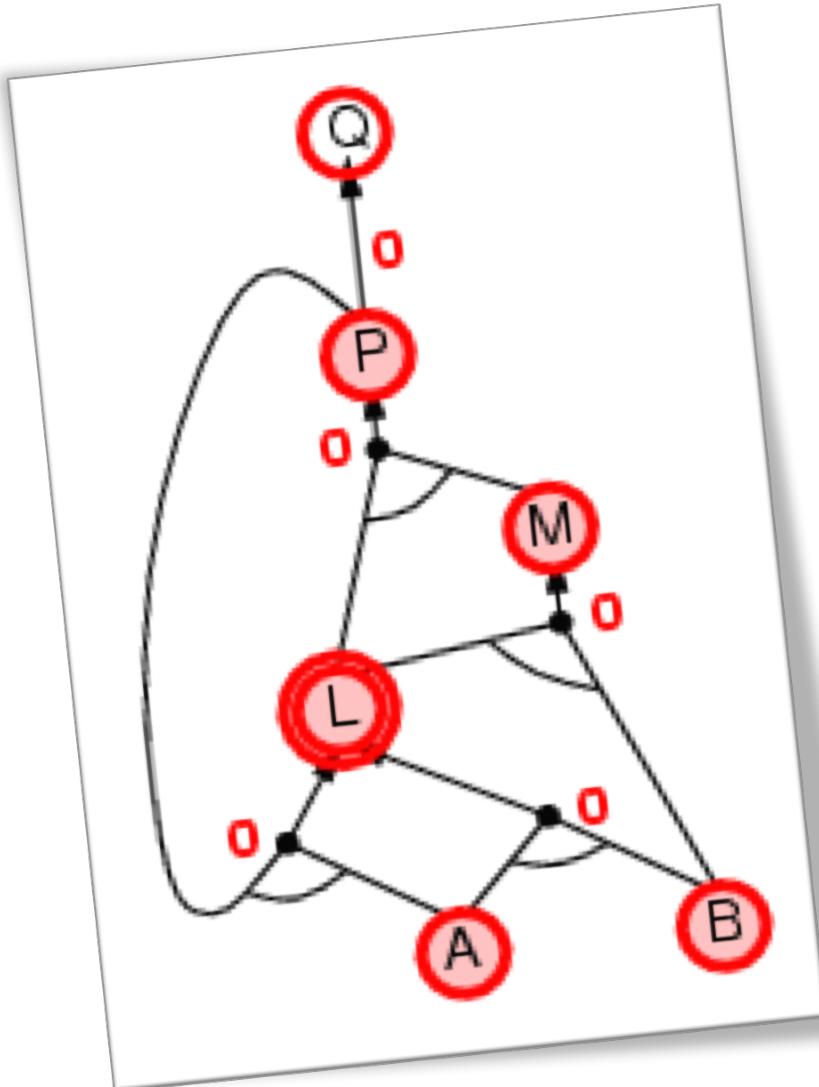
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B

# استنتاج روبره جلو – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

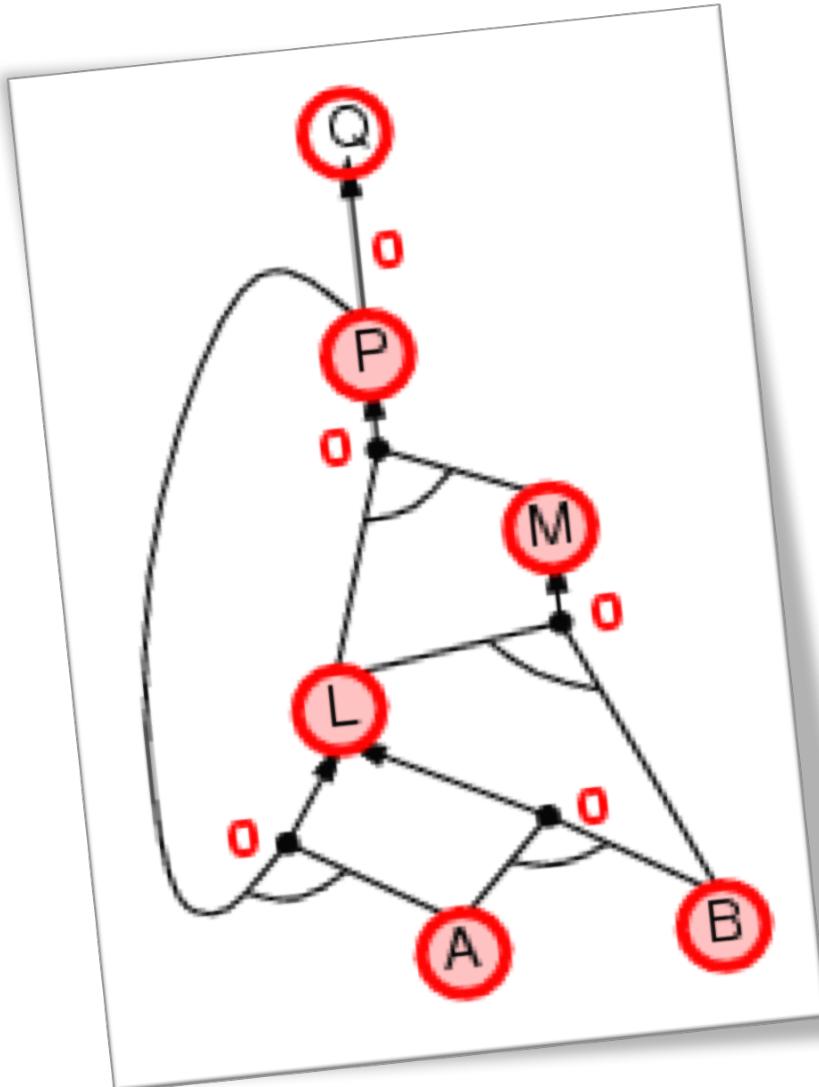
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره جلو – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

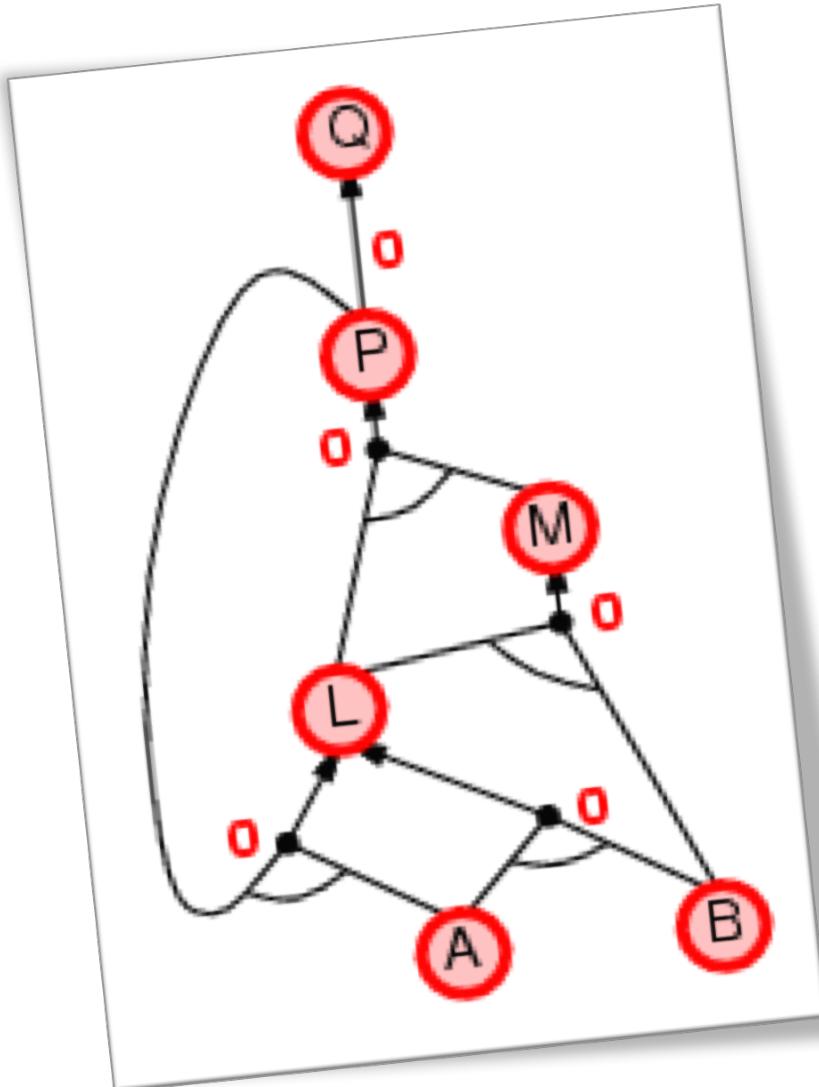
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره جلو – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B

# ویژگی‌های الگوریتم زنجیره‌ای روبه‌جلو

---

- زمان؟
- زمان خطی نسبت به اندازه پایگاه دانش
- صحت؟
- اساس کار الگوریتم زنجیره‌ای روبه‌جلو، استفاده از Modus Ponens است که صحیح می‌باشد.
- کامل؟
- بله، اثبات در اسلاید بعد ☺
- زنجیره‌سازی روبه‌جلو، یک نوع استدلال مبتنی بر داده است و می‌توان از آن برای استخراج دانش از روی ادراکات عامل استفاده کرد.

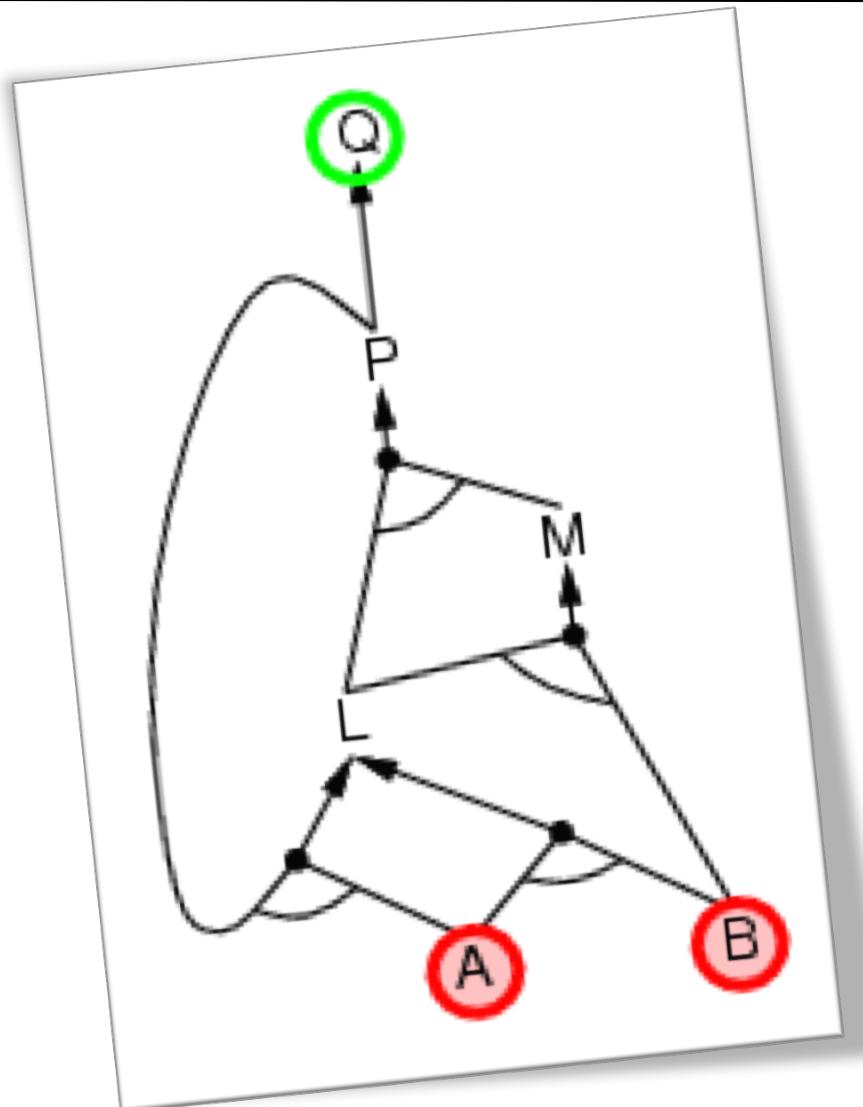
# اثبات کامل بودن الگوریتم زنجیرهای رو به جلو

- حالت پایانی inferred را در نظر بگیرید. (یعنی نقطه ثابتی (fixed point) که هیچ جمله اتمیک جدیدی به دست نمی‌آید.)
- حالت پایانی را به عنوان یک مدل  $m$  در نظر بگیرید که در آن به نمادهای استنتاج شده true و false تخصیص داده شده است.
- هر بند معین در  $m$  در KB نیز درست است
- اگر  $b \Rightarrow a_1 \wedge \dots \wedge a_n \Rightarrow b$  غلط باشد آن‌گاه  $a_1 \wedge \dots \wedge a_n$  درست و  $b$  غلط است. این موضوع با رسیدن به یک نقطه ثابت در تناقض است.
- بنابراین  $m$  یک مدل از KB است.
- اگر  $\text{KB} \models q$ , جمله اتمیک  $q$  در هر مدل از KB شامل  $m$  درست است.

# الگوریتم زنجیره‌ای روبه عقب

- ایده: از query شروع کن و به سمت عقب حرکت کن.
  - if q is known to be true then
    - no work is needed
  - else
    - find those rule concluding q
    - if all premises of one of them can be proved (by backward chaining) then q is true
- اجتناب از حلقه‌ها: بررسی آن که آیا زیرهدف جدید قبلاً بر روی پشته هدف بوده است.
- اجتناب از کار تکراری: بررسی آن که آیا قبلاً درستی زیرهدف جدید اثبات شده یا با شکست روبه رو شده است.

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

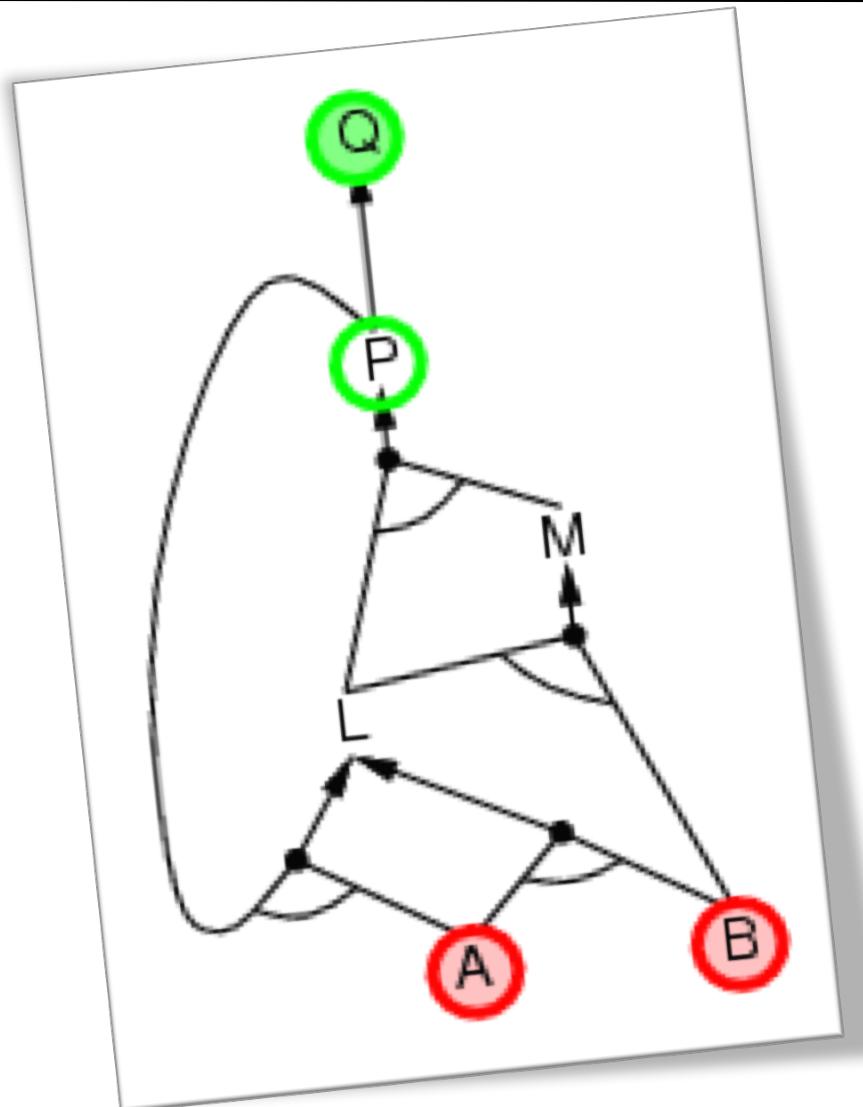
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

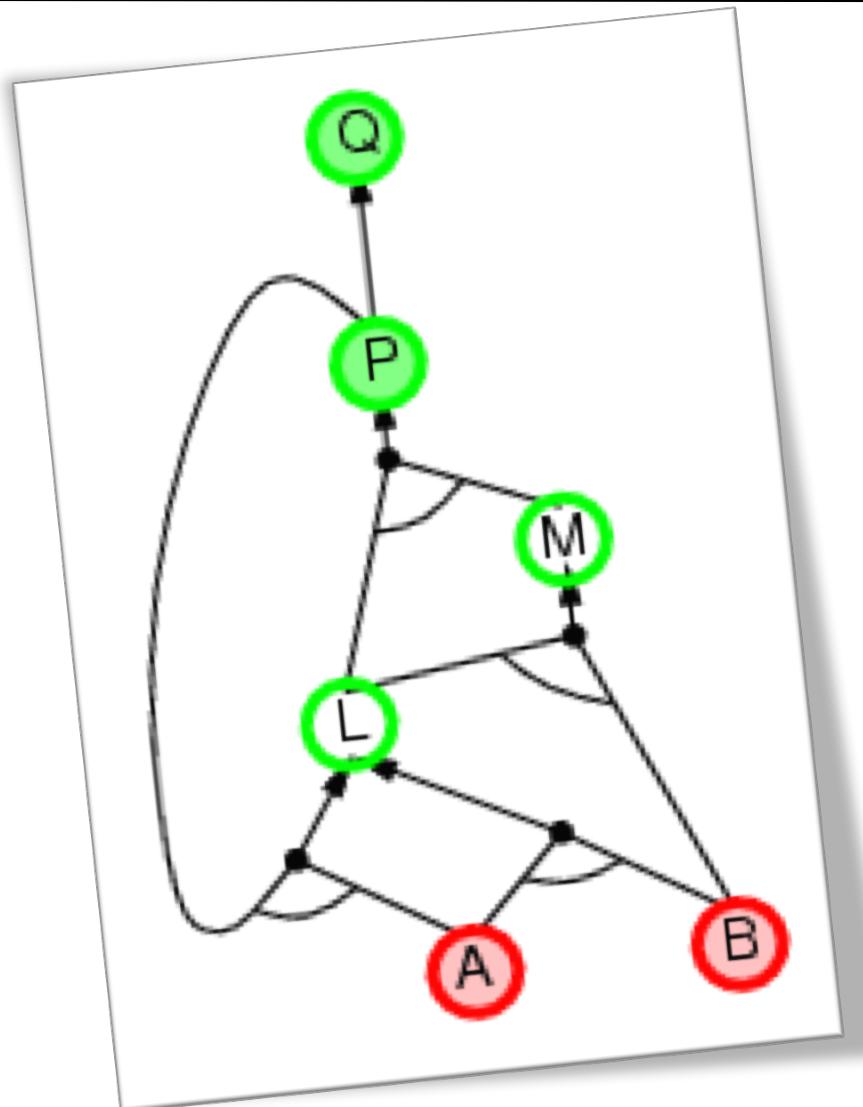
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

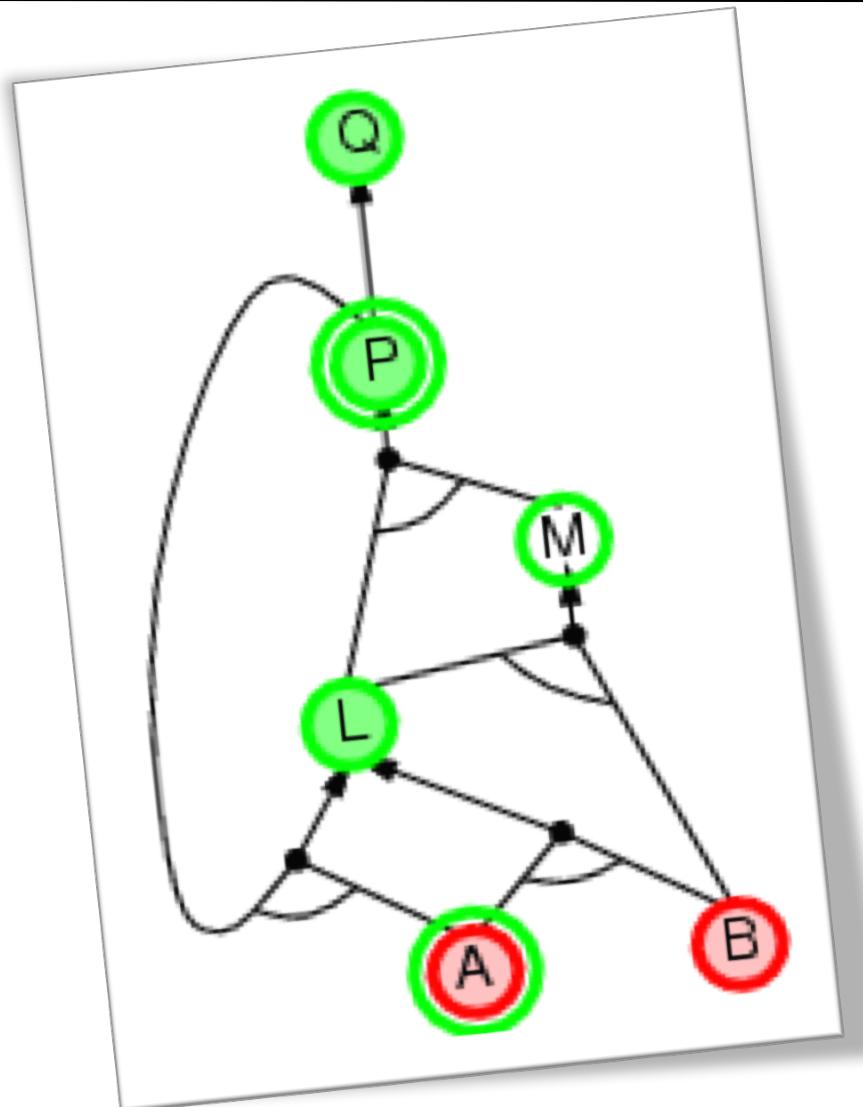
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

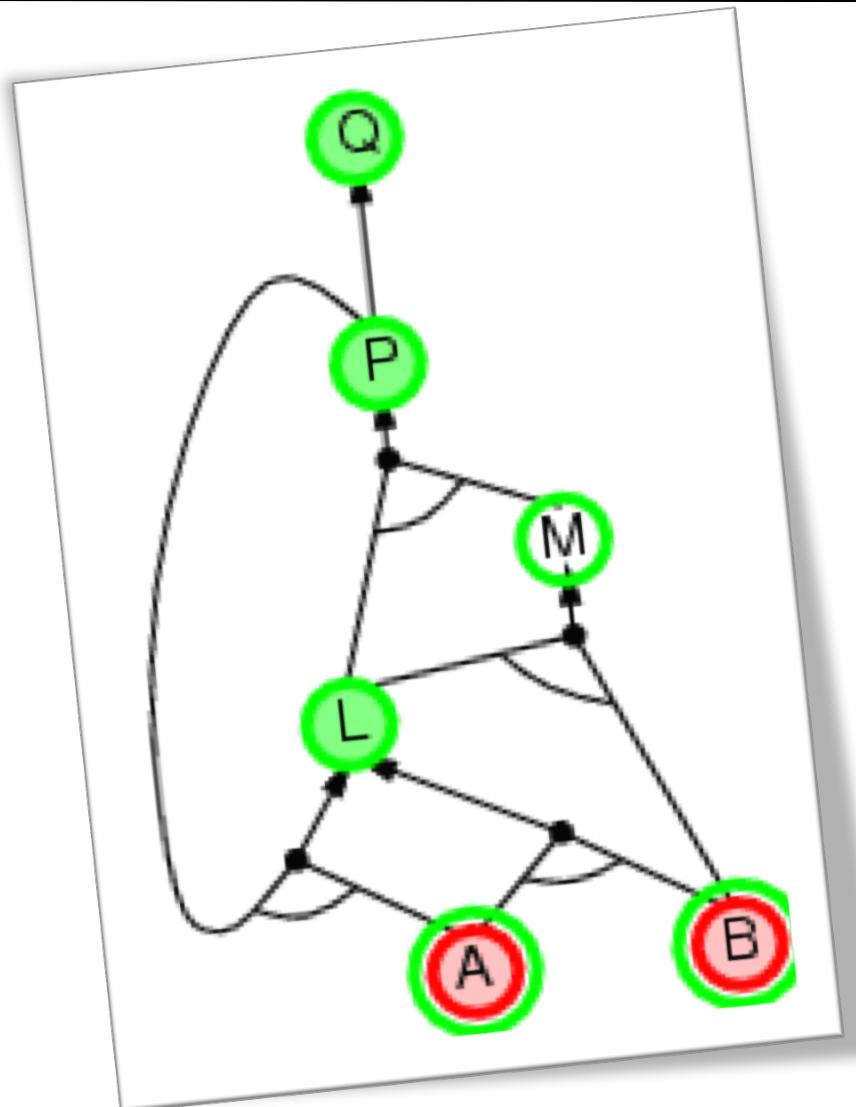
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

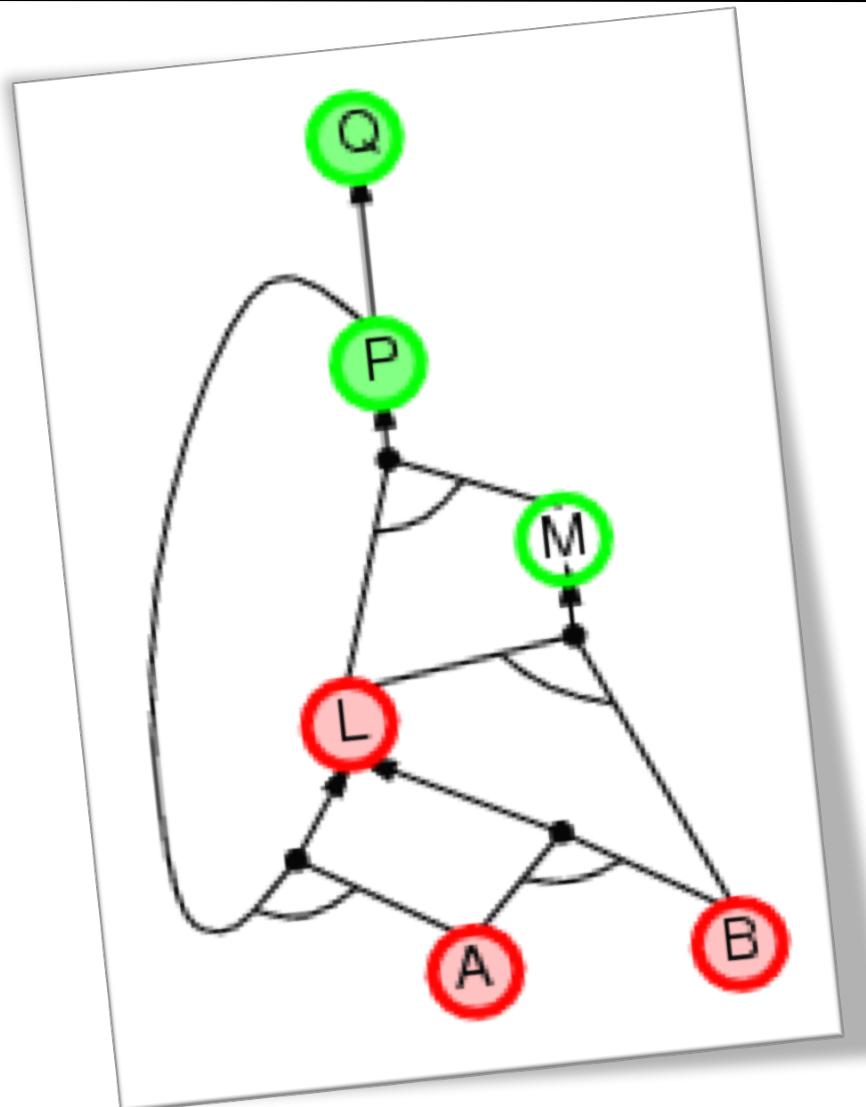
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

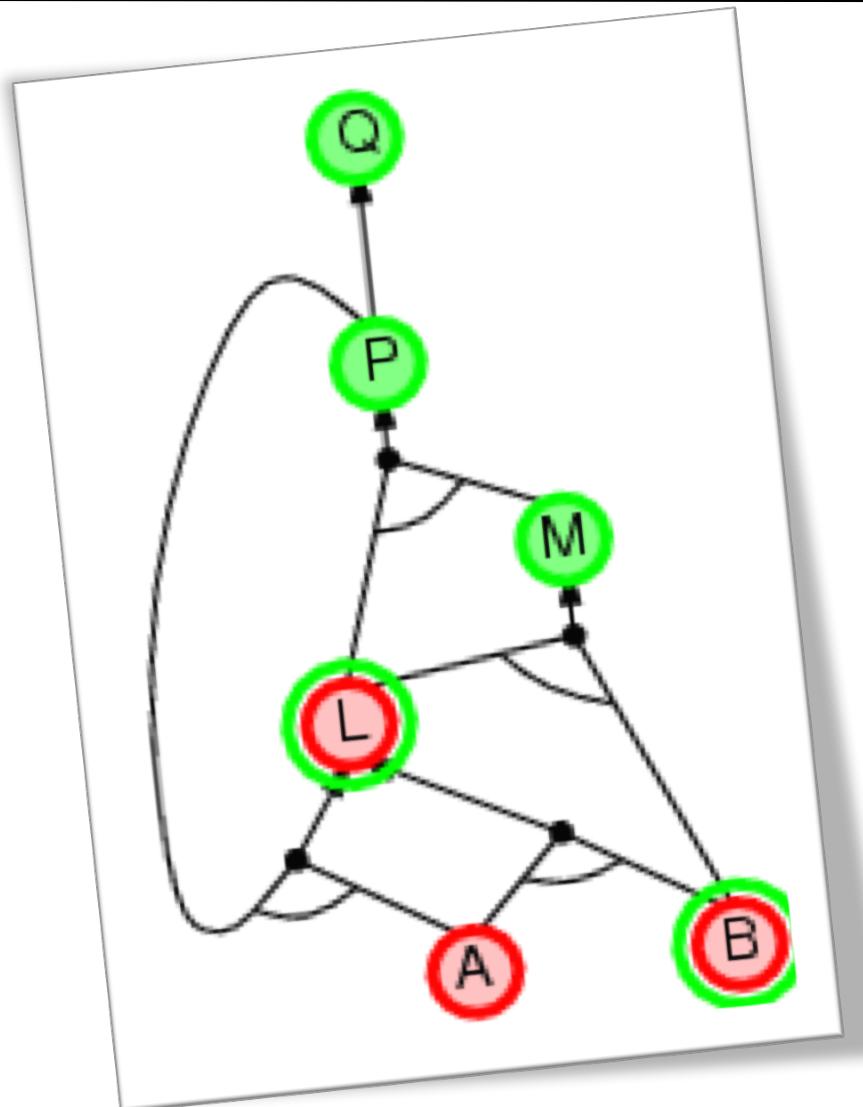
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

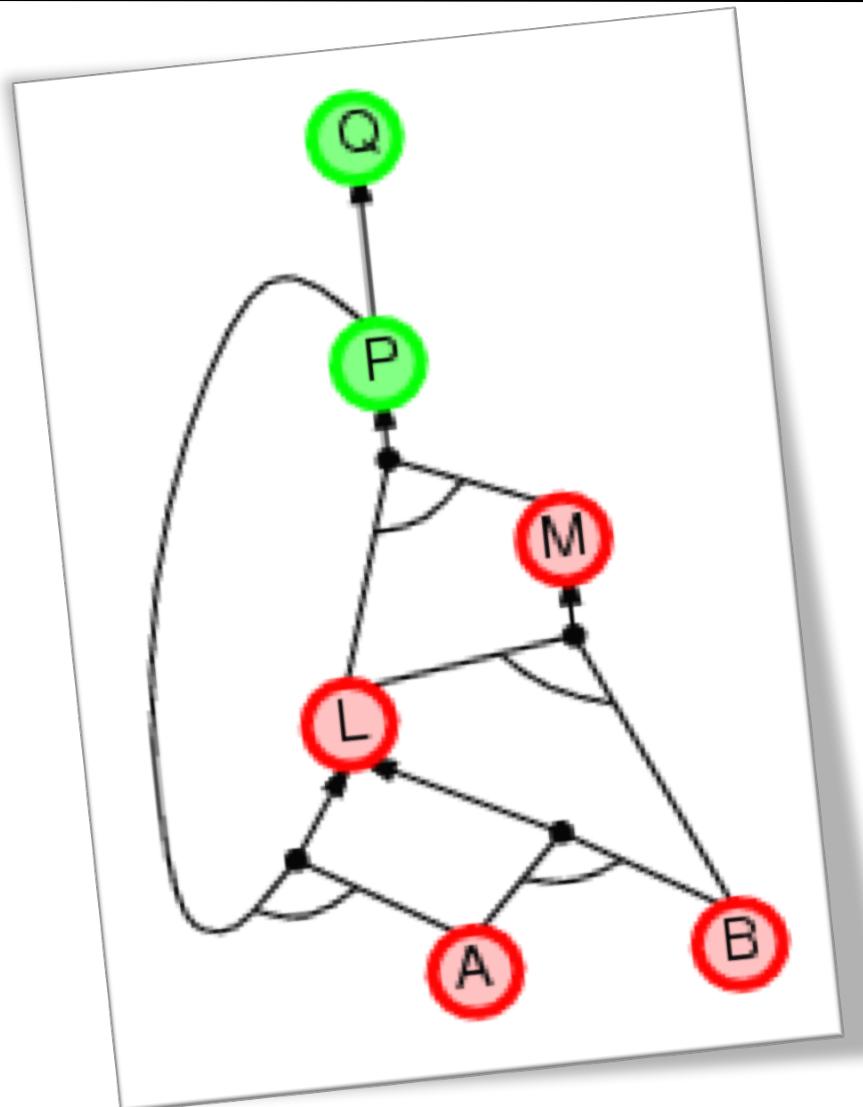
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

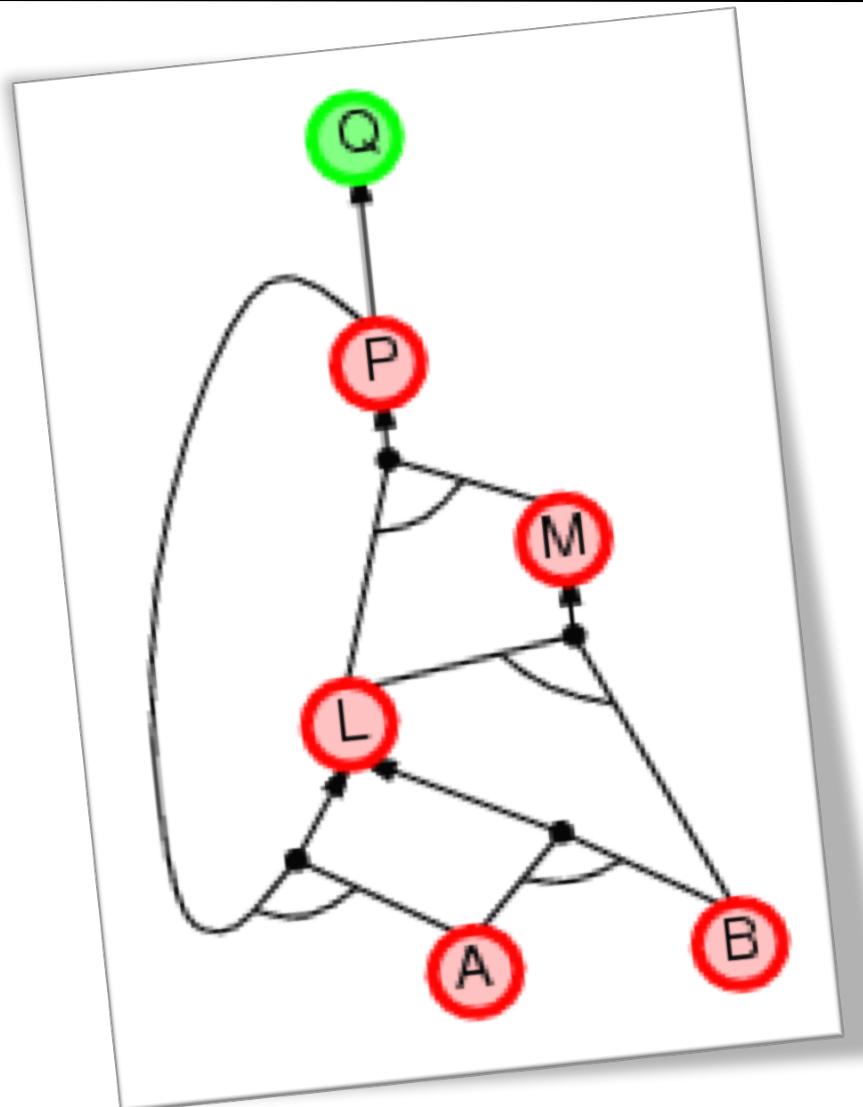
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

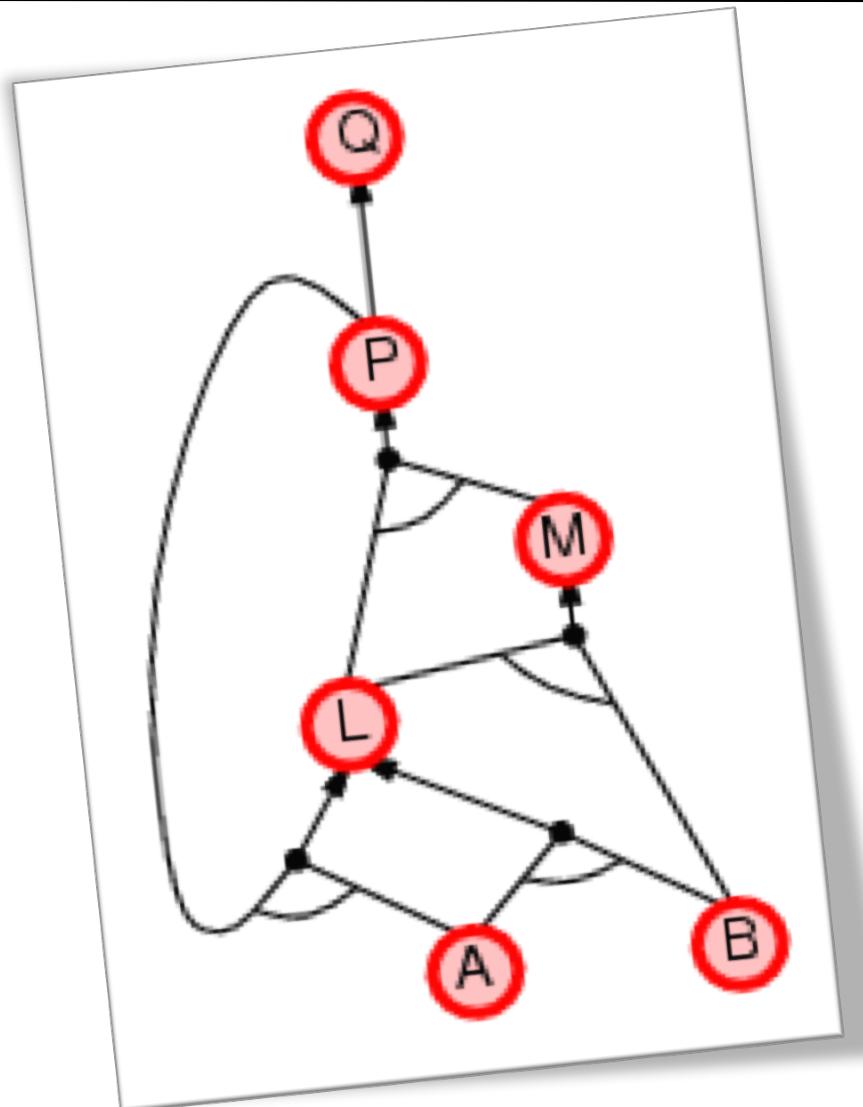
$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# استنتاج روبره عقب – مثال



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# ویژگی‌های الگوریتم زنجیره‌ای روبه عقب

---

- الگوریتم زنجیره‌ای روبه عقب یک الگوریتم مبتنی بر هدف است.
- زمان اجرای زنجیره‌سازی روبه عقب مانند زنجیره‌سازی روبه‌جلو، بر حسب اندازه پایگاه دانش خطی است. اما چون فقط واقعیت‌های مرتبط با هدف را بررسی می‌کند، اغلب سریع‌تر از آن عمل می‌کند.

# استنتاج گزارهای مؤثر

---

# استنتاج گزارهای مؤثر

- هدف بررسی دو خانواده از الگوریتم‌های مؤثر در استنتاج گزارهای مبتنی بر وارسی مدل است.
- الگوریتم‌های کامل جستجوی عقبگرد
  - الگوریتم DPLL (Davis, Putnam, Logemann, Loveland)
  - الگوریتم ناکامل جستجوی محلی
  - الگوریتم WalkSAT
- این الگوریتم‌ها برای تعیین ارضایی‌یاری یک جمله استفاده می‌شوند.
- برای تعیین آن که  $\alpha \models_{KB} \neg\alpha$  یا خیر می‌توان ناسازگار بودن  $\alpha \neg\alpha$  را بررسی کرد.

- ورودی DPLL یک جمله به شکل نرمال عطفی (CNF) یعنی مجموعه‌ای از بندها (clauses) است.
- اساسا مشابه با TT-Entail? و Backtracking-search یک برشماری بازگشتی و اول عمق از مدل‌های ممکن انجام می‌دهد.
- بهبود آن نسبت به این روش‌ها به دلیل استفاده از سه هیوریستیک زیر برای تعیین امیدبخش یا غیر امیدبخش بودن گره‌ها در فضای جستجو است:
  - خاتمه زود هنگام (*Early termination*)
  - هیوریستیک نماد محض (*Pure symbol heuristic*)
  - هیوریستیک بند واحد (*Unit clause heuristic*)

# الگوریتم DPLL - هیوریستک‌ها

---

- خاتمه زود هنگام
- یک بند  $true$  است اگر حداقل یکی از لیترال‌های آن  $true$  باشد.
- درست است  $(A \vee B) \wedge (A \vee C)$  در صورت درست بودن  $A$  صرف نظر از مقادیر  $B$  و  $C$  درست است
- یک جمله  $false$  است اگر حداقل یکی از بندهای آن  $false$  باشد.
- خاتمه زودهنگام از آزمایش کل زیردرخت‌ها در فضای جستجو جلوگیری می‌کند.

# الگوریتم DPLL - هیوریستک‌ها

## • هیوریستیک نماد محض

- نماد محض: نمادی که در تمام بندها با یک علامت ظاهر شود.
- در سه بند  $(A \vee \neg B)$ ,  $(\neg B \vee \neg C)$ ,  $(C \vee A)$  نمادهای  $A$  و  $B$  محض و  $C$  غیر محض است.
- اگر جمله‌ای که به فرم CNF است ارضاشدنی باشد حتماً برای آن مدلی وجود دارد که در آن مقدار نمادهای محض مثبت برابر true و مقدار نمادهای محض منفی برابر false است.
- در تعیین نماد محض از بندهایی که قبلاً با مقدار true شناخته شده‌اند صرف نظر می‌شود.
- در عبارت بالا با تعیین  $B=false$  بندهای  $(A \vee \neg B \vee \neg C)$  و  $(\neg B \vee \neg C)$  پیدا می‌کند و در نتیجه  $C$  نماد محض خواهد بود.

# الگوریتم DPLL - هیوریستک‌ها

- هیوریستیک بند واحد
- بند واحد: تنها شامل یک لیترال می‌باشد. یا بندی است که تمام لیترال‌های آن غیر از یک لیترال، تا کنون مقدار False گرفته باشد.
- در این صورت، تنها لیترال موجود در یک بند واحد باید True باشد تا بند نیز گردد.
- انتساب به یک بند واحد می‌تواند بند واحد دیگری تولید کند به این کار انتشار واحد می‌گویند. (Unit propagation)
- هنگامی که به C مقدار false داده شود بند  $(C \vee A)$  یک بند واحد می‌شود.
- هیوریستیک‌های دیگری نیز وجود دارد (صفحه ۲۶۱ کتاب مرجع ویرایش ۳).
- هیوریستیک درجه: انتخاب متغیری که در تعداد بندهای باقی‌مانده‌ی بیشتری ظاهر شده باشد.
- آنالیز مؤلفه‌های مجزا: مجموعه بندهای مجزا یعنی آن‌هایی که هیچ متغیر مشترکی ندارند.

**function** DPLL-SATISFIABLE?(*s*) **returns** *true* or *false*

**inputs:** *s*, a sentence in propositional logic

*clauses*  $\leftarrow$  the set of clauses in the CNF representation of *s*

*symbols*  $\leftarrow$  a list of the proposition symbols in *s*

**return** DPLL(*clauses*, *symbols*, { })

**function** DPLL(*clauses*, *symbols*, *model*) **returns** *true* or *false*

**if** every clause in *clauses* is true in *model* **then return** *true*

**if** some clause in *clauses* is false in *model* **then return** *false*

*P, value*  $\leftarrow$  FIND-PURE-SYMBOL(*symbols*, *clauses*, *model*)

**if** *P* is non-null **then return** DPLL(*clauses*, *symbols* – *P*, *model*  $\cup$  {*P*=*value*})

*P, value*  $\leftarrow$  FIND-UNIT-CLAUSE(*clauses*, *model*)

**if** *P* is non-null **then return** DPLL(*clauses*, *symbols* – *P*, *model*  $\cup$  {*P*=*value*})

*P*  $\leftarrow$  FIRST(*symbols*); *rest*  $\leftarrow$  REST(*symbols*)

**return** DPLL(*clauses*, *rest*, *model*  $\cup$  {*P*=true}) **or**

DPLL(*clauses*, *rest*, *model*  $\cup$  {*P*=false}))

می خواهیم مسئله ارضآپذیری زیر را با استفاده از الگوریتم DPLL حل کنیم. اگر انتساب مقدار «صفر» به متغیرها، بر انتساب مقدار «یک» به آنها اولویت داشته باشد، از کدام یک از دو تکنیک Pure Literal (PL) و Unit Clause (UC) در حل این مسئله خاص استفاده خواهد شد؟  
(مهندسی کامپیوتر دولتی ۹۴)

۱) فقط از UC استفاده می شود. ✓

۲) فقط از PL استفاده می شود.

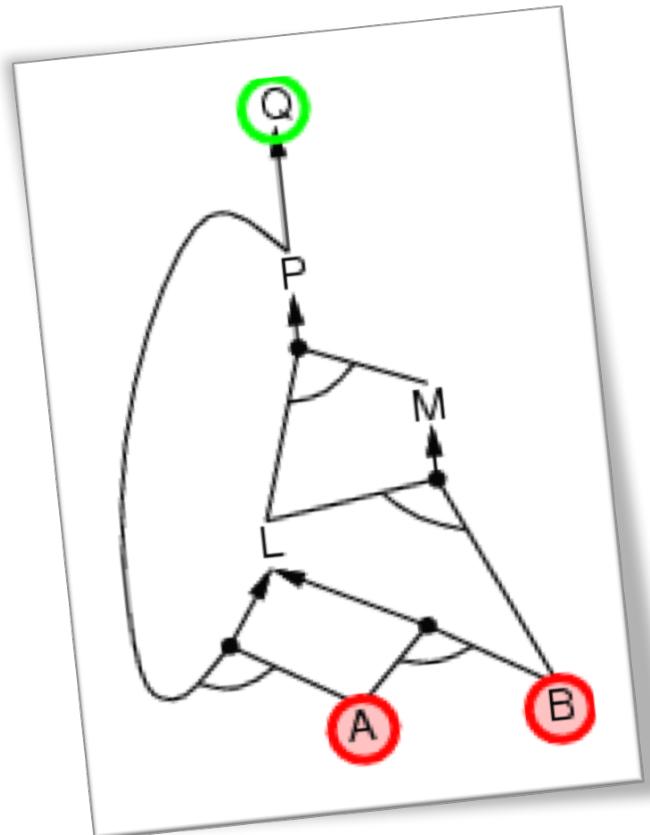
۳) هم از UC هم از PL استفاده می شود.

۴) از UC و یا PL استفاده نمی شود.

$$\{\neg A \vee B \vee C\}, \{A \vee \neg B \vee C\}, \{A \vee B \vee \neg C\}, \{A \vee B \vee C\}$$

# الگوریتم DPLL و زنجیره‌سازی رو به جلو

- تمرین: بررسی کنید که اگر عبارت CNF فقط شامل جملات هورن باشد آن‌گاه دقیقاً مانند زنجیره پیش‌رو عمل می‌کند.
- برای مثال رفتار DPLL در این پایگاه دانش متناظر با رفتار FC است.



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

# الگوریتم جستجوی محلی WalkSat

---

- در هر تکرار یک بند ارضا نشده را انتخاب می‌کند و یک نماد آن را برمی‌گزیند تا مقدار آن را عکس کند.
- تابع ارزیابی: هیوریستیک حداقل تناقض (min-conflict) یعنی تعداد بندهایی که در مدل فعلی True نمی‌شود.
- برای انتخاب یک نماد به صورت تصادفی از یکی از دو راه زیر استفاده می‌کند:
  - یک گام حداقل تناقض که تعداد بندهای ارضا نشده در حالت جدید را حداقل می‌کند.
  - یک گام راه رفتن تصادفی که نمادها را به صورت تصادفی انتخاب می‌کند. این روش سعی می‌کند از گیر افتادن در نقطه بهینه محلی جلوگیری کند.

# الگوریتم جستجوی محلی WalkSat

**function** WALKSAT(*clauses*, *p*, *max-flips*) **returns** a satisfying model or *failure*

**inputs:** *clauses*, a set of clauses in propositional logic

*p*, the probability of choosing to do a “random walk” move, typically around 0.5

*max-flips*, number of flips allowed before giving up

*model*  $\leftarrow$  a random assignment of *true/false* to the symbols in *clauses*

**for** *i* = 1 **to** *max-flips* **do**

**if** *model* satisfies *clauses* **then return** *model*

*clause*  $\leftarrow$  a randomly selected clause from *clauses* that is false in *model*

**with probability** *p* flip the value in *model* of a randomly selected symbol from *clause*

**else** flip whichever symbol in *clause* maximizes the number of satisfied clauses

**return** *failure*

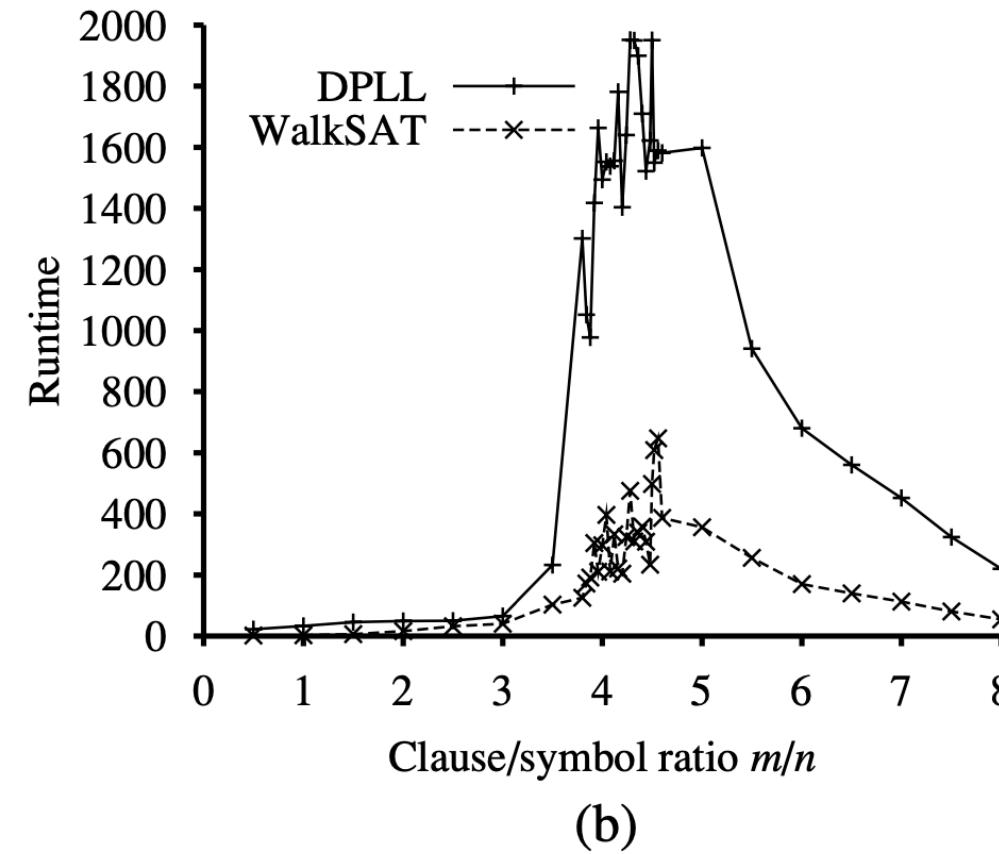
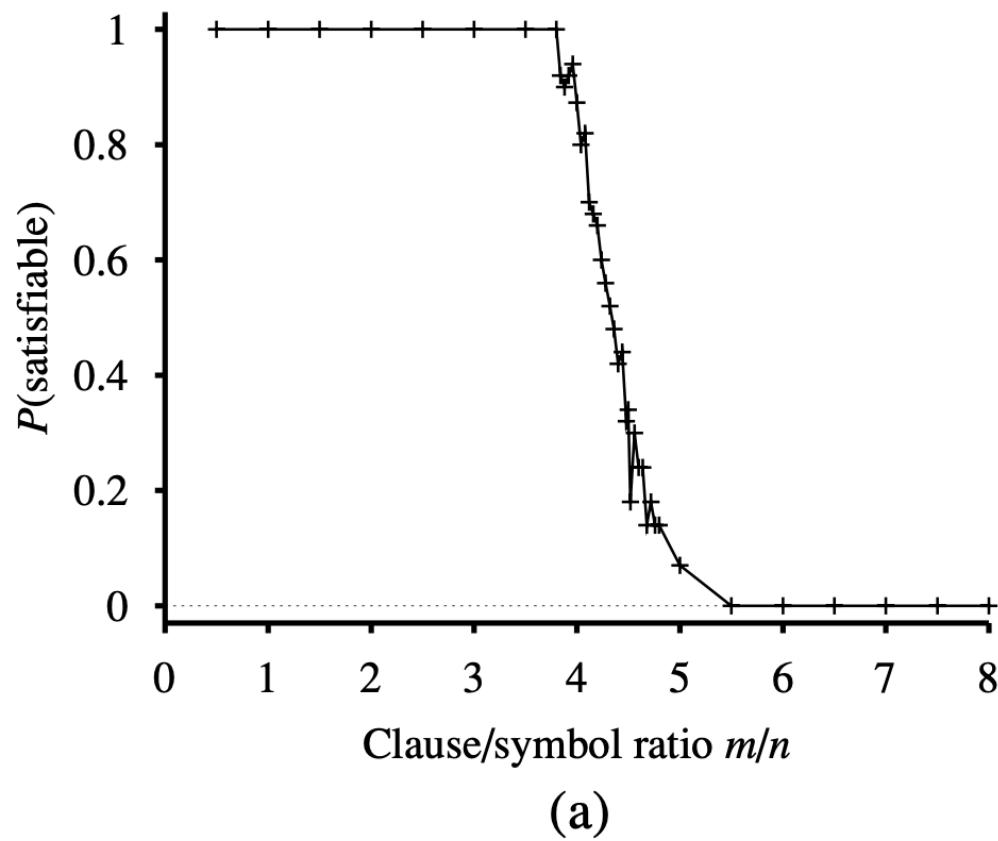
# الگوریتم جستجوی محلی WalkSat

---

- با توجه به نحوه انتخابش، تعادلی مناسب میان میزان حریصانه بودن و تصادفی بودن برقرار می کند.
- اگر الگوریتم مدلی را بازگرداند جمله ورودی ارضیابذیر است ولی اگر failure را بازگرداند نمی توانیم تعیین کنیم که آن جمله ارضانایپذیر است.
- اگر جواب وجود داشته باشد و max-flips مقدار بینهایت داشته باشد و  $p > 0$  الگوریتم یک مدل را برمی گردد.
- اگر جمله ناسازگار باشد و max-flips مقدار بینهایت داشته باشد الگوریتم هیچگاه پایان نمی پذیرد.
- باید یک مقدار محدود برای max-flips مشخص کرد.

# میزان سختی مسائل SAT

- جملات 3-CNF( $m,n$ ) (هر عبارت ۳ لیترال دارد.)
- $n=50$  تعداد متغیرها و  $m$  تعداد عبارات



# Wumpus world example: propositional symbols

---

- ▶ Atemporal variables that may be needed
  - ▶  $P_{i,j}$  is true if there is a pit in  $[i,j]$ .
  - ▶  $W_{i,j}$  is true if there is a wumpus in  $[i,j]$ , dead or alive.
  - ▶  $B_{i,j}$  is true if the agent perceives a breeze in  $[i,j]$ .
  - ▶  $S_{i,j}$  is true if the agent perceives a stench in  $[i,j]$ .
- ▶ Some of the temporal variables that may be needed
  - ▶ Variables for percepts and actions
  - ▶  $L_{i,j}^t, FacingEast^t, FacingWest^t, FacingNorth^t, FacingSouth^t$
  - ▶  $HaveArrow^t$
  - ▶  $WumpusAlive^t$

# Wumpus world example: axioms on atemporal aspect of the world

---

- ▶ **Axioms:** general knowledge about how the world works
- ▶ General rules on atemporal variables

$$\neg P_{1,1}$$

$$\neg W_{1,1}$$

$$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y}) \quad \text{A distinct rule is needed for each square}$$

$$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$$

$$W_{1,1} \vee W_{1,2} \vee \dots \vee W_{4,4}$$

$$\neg W_{1,1} \vee \neg W_{1,2}$$

$$\neg W_{1,1} \vee \neg W_{1,3}$$

...

# Wumpus world example: perceptions & actions

---

- ▶ **Perceptions** are converted to facts and are Telled to KB
  - ▶  $MAKE\_PERCEPT\_SENTENCE([Breeze, Stench, None, None, None], t)$
  - ▶  $Breeze^t, Stench^t$  are added to KB
  
- ▶ Selected **action** is converted to a fact and is Telled to KB
  - ▶  $MAKE\_ACTION\_SENTENCE(Forward, t)$
  - ▶  $Forward^t$  is added to KB

# Wumpus world example: transition model

---

- ▶ Changing aspect of world (**Temporal variables**)
  - ▶ They are also called as fluents or state variables
- ▶ Initial KB includes the initial status of some of the temporal variables:
  - ▶  $L_{1,1}^0, FacingEast^0, HaveArrow^0, WumpusAlive^0$
- ▶ Percepts are connected to fluents describing the properties of squares
  - ▶  $L_{x,y}^t \Rightarrow (Breeze^t \Leftrightarrow B_{x,y})$
  - ▶  $L_{x,y}^t \Rightarrow (Stench^t \Leftrightarrow S_{x,y})$
  - ▶ ...
- ▶ Transition model: fluent can change as the results of agent's actions

# Wumpus world example: transition model Frame problem

---

- ▶ Transition model:

$$L_{1,1}^0 \wedge FacingEast^0 \wedge Forward^0 \Rightarrow (L_{2,1}^1 \wedge \neg L_{1,1}^1)$$

$$L_{1,1}^0 \wedge FacingEast^0 \wedge TurnLeft^0 \Rightarrow (L_{1,1}^1 \wedge FacingNorth^1 \wedge \neg FacingEast^1)$$

...

- ▶ After doing *Forward* action at time 0:

- ▶  $ASK(KB, HaveArrow^1) = false$

- ▶ Why?

- ▶ **Frame problem:** representing the effects of actions without having to represent explicitly a large number of intuitively obvious non-effects

# Wumpus world example

Solution to frame problem: successor-state axioms

---

- ▶ A Solution to frame problem: Successor-state axioms

Instead of writing axioms about actions, we write axioms about fluents

- ▶  $F^{t+1} \Leftrightarrow ActionCausesF^t \vee (F^t \wedge \neg ActionCausesNotF^t)$

$$HaveArrow^{t+1} \Leftrightarrow (HaveArrow^t \wedge \neg Shoot^t)$$

$$\begin{aligned} L_{1,1}^{t+1} &\Leftrightarrow (L_{1,1}^t \wedge (\neg Forward^t \vee Bump^{t+1})) \\ &\vee (L_{1,2}^t \wedge (FacingSouth^t \wedge Forward^t)) \\ &\vee (L_{2,1}^t \wedge (FacingWest^t \wedge Forward^t)) \end{aligned}$$

# A hybrid agent for the wumpus world

## ► Basic concepts:

### ► Safeness of $[x, y]$ square:

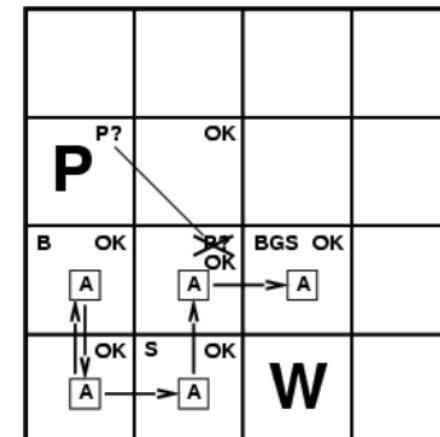
$$\triangleright OK_{x,y}^t \Leftrightarrow \neg P_{x,y} \wedge \neg(W_{x,y} \wedge WumpusAlive^t)$$

### ► Set of safe squares:

$$\triangleright safe \leftarrow \{[x, y]: ASK(KB, OK_{x,y}^t) = true\}$$

### ► Set of unvisited squares:

$$\triangleright unvisited \leftarrow \{[x, y]: ASK(KB, L_{x,y}^{t'}) = false \text{ for all } t' \leq t\}$$



**function** HYBRID\_WUMPUS\_AGENT(*percept*) **returns** an action sequence

**inputs:** *percept*, a list, [*breeze*, *stench*, *glitter*, *bump*, *scream*]

**persistent:** *KB*, initially the atemporal “wumpus physics”

*t*, a time counter, initially 0

*plan*, an action sequence, initially empty

TELL(*KB*, MAKE\_PERCEPT\_SENTENCE(*percept*, *t*))

TELL the *KB* the temporal “physics” sentences for time *t*

*safe*  $\leftarrow \{[x,y] : \text{ASK}(KB, OK_{x,y}^t) = \text{true}\}$

**if** *ASK(KB, Glitter<sup>t</sup>) = true* **then**

*plan*  $\leftarrow [Grab] + \text{PLAN\_ROUTE}(\text{current}, \{[1,1]\}, \text{safe}) + [Climb]$

**if** *plan* = {} **then**

*unvisited*  $\leftarrow \{[x,y] : \text{ASK}(KB, L_{x,y}^{t'}) = \text{false} \text{ for all } t' \leq t\}$

*plan*  $\leftarrow \text{PLAN\_ROUTE}(\text{current}, \text{unvisited} \cap \text{safe}, \text{safe})$

**if** *plan* = {} and *ASK(KB, HaveArrow<sup>t</sup>) = true* **then**

*possible\_wumpus*  $\leftarrow \{[x,y] : \text{ASK}(KB, \neg W_{x,y}) = \text{false}\}$

*plan*  $\leftarrow \text{PLAN\_SHOT}(\text{current}, \text{possible_wumpus}, \text{safe})$

**if** *plan* = {} **then**

*not\_unsafe*  $\leftarrow \{[x,y] : \text{ASK}(KB, \neg OK_{x,y}^t) = \text{false}\}$

*plan*  $\leftarrow \text{PLAN\_ROUTE}(\text{current}, \text{unvisited} \cap \text{not_unsafe}, \text{safe})$

**if** *plan* = {} **then**

*plan*  $\leftarrow \text{PLAN\_ROUTE}(\text{current}, \{[1,1]\}, \text{safe}) + [Climb]$

*action*  $\leftarrow \text{POP}(\text{plan})$

TELL(*KB*, MAKE\_ACTION\_SENTENCE(*action*, *t*))

*t*  $\leftarrow t + 1$

**return** *action*

# Hybrid agent (phases listed based on priority)

---

- ▶ Simple-reflex (condition-action rule):

**if**  $\text{ASK}(KB, \text{Glitter}^t) = \text{true}$  **then**

$plan \leftarrow [\text{Grab}] + \text{PLAN\_ROUTE}(\text{current}, \{[1,1]\}, \text{safe}) + [\text{Climb}]$

- ▶ Goal-based & reasoning:

**if**  $plan = \{\}$  **then**

$unvisited \leftarrow \{[x, y] : \text{ASK}(KB, L_{x,y}^{t'}) = \text{false} \text{ for all } t' \leq t\}$

$plan \leftarrow \text{PLAN\_ROUTE}(\text{current}, unvisited \cap \text{safe}, \text{safe})$

**if**  $plan = \{\}$  and  $\text{ASK}(KB, \text{HaveArrow}^t) = \text{true}$  **then**

$possible\_wumpus \leftarrow \{[x, y] : \text{ASK}(KB, \neg W_{x,y}) = \text{false}\}$

$plan \leftarrow \text{PLAN\_SHOT}(\text{current}, possible\_wumpus, \text{safe})$

# Hybrid agent (phases listed based on priority)

---

## ▶ Exploration & reasoning

**if**  $plan = \{\}$  **then**

$not\_unsafe \leftarrow \{[x,y]: ASK(KB, \neg OK_{x,y}^t) = false\}$

$plan \leftarrow PLAN\_ROUTE(current, unvisited \cap not\_unsafe, safe)$

## ▶ Goal-based:

**if**  $plan = \{\}$  **then**

$plan \leftarrow PLAN\_ROUTE(current, \{[1,1]\}, safe) + [Climb]$

# Hybrid agent: Using A\* to make a plan

**function** *PLAN\_ROUTE*(*current, goal, allowed*) **returns** an action sequence

**inputs:** *current*, the agent's current position

*goals*, a set of squares; try to plan a route to one of them

*allowed*, a set of squares that can form part of the route

*problem*  $\leftarrow$  *ROUTE\_PROBLEM*(*current, goals, allowed*)

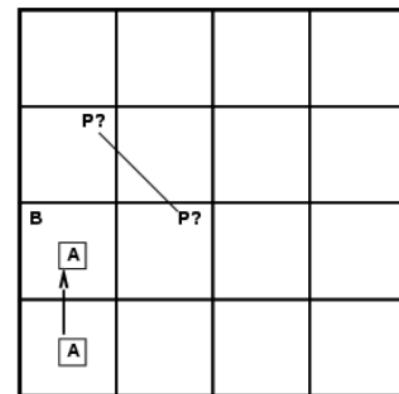
**return** *A\* GRAPH SEARCH*(*problem*)

# Logical state estimation

- ▶ **Problem:** Computational time of ASK grows by the length of the agent's life
  - ▶ We must save the results of inference to use them in the next time steps
- ▶ **Solution:** Belief state instead of the whole past history of percepts and actions
  - ▶ Belief state: the agent's knowledge about the state of the world (the set of all possible current states of the world)

$$B_{1,2} \wedge (P_{1,3} \vee P_{2,2})$$

$$\wedge L_{1,2}^1 \wedge FacingNorth^1 \wedge HaveArrow^1 \wedge WumpusAlive^1$$



- ▶ State estimation: process of updating the belief state when doing actions and receiving new percepts

# Approximate state estimation

---

- ▶ **Problem:** The exact belief state estimation may require logical formulas whose size is exponential in the number of symbols.
  - ▶  $n$  fluent symbols,  $2^n$  possible physical states,  $2^{2^n}$  possible belief states
- ▶ **Solution:** approximation
  - ▶ e.g., 1-CNF belief state: given the belief state at  $t - 1$ , agent tries to prove  $X^t$  or  $\neg X^t$  for each  $X^t$ )
- ▶ Approximate belief state estimation?  
compact representation (open area for research)
- ▶ Relation to model-based agents

# Limitations of propositional logic

---

- ▶ Wumpus world
  - ▶ Distinct rules “for each square [x,y]” and “for each time t”
  - ▶  $L_{x,y}^t \wedge FacingEast^t \wedge TurnLeft^t \Rightarrow (L_{x,y}^{t+1} \wedge FacingUp^{t+1} \wedge \neg FacingEast^{t+1})$
- ▶ Many rules are needed to describe the world axioms
  - ▶ Rather impractical for bigger world
- ▶ We can not express general knowledge about “physics” of the world directly in propositional language
  - ▶ More expressive language is needed.
- ▶ Qualification problem: specifying all the exceptions
  - ▶ Probability theory allows us to summarize all the exceptions (without explicitly naming them)