هوش مصنوعی

فصل ششم

عامل های منطقی

فهرست

- عاملهای مبتنی بر دانش
- منطق
- منطق گزاره ای
- الگوهای استدلال در منطق گزاره ای
- الگوریتم resolution
- زنجیر پیشرو و عقبگرد

عاملهای مبتنی بر دانش

- مؤلفه اصلی عامل مبتنی بر دانش، **پایگاه دانش** آن است
  - پایگاه دانش: مجموعه ای از جملات
  - جمله: **زبان نمایش دانش** و بیان ادعاهایی در مورد جهان

- برای اضافه کردن جملات به پایگاه دانش و درخواست دانسته ها

  - TELLو ASK

- پیروی:انجام فرایند استنتاج تحت مقررات خاص

```
function KB-AGENT(percept) returns an action
  static: KB, a knowledge base
          t, a counter, initially 0, indicating time

  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  action ← ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t ← t + 1
  return action
```

Figure 7.1 A generic knowledge-based agent.

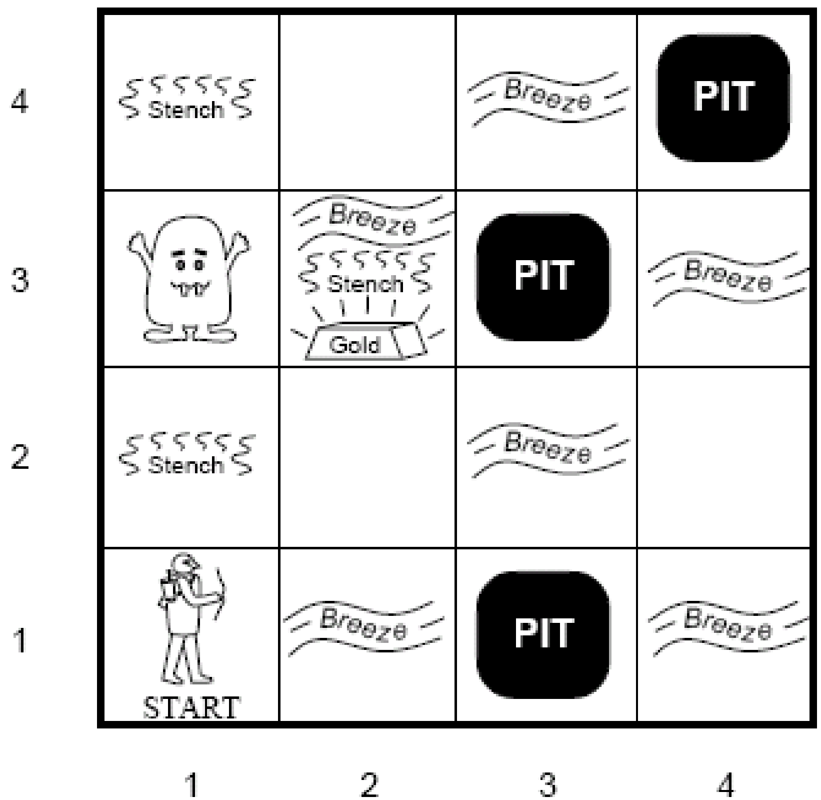
عاملهای منطقی

عاملهای مبتنی بر دانش

- عامل مبتنی بر دانش باید بتواند:
  - نمایش حالات و فعالیتها
  - ترکیب ادراکات جدید
  - بروز کردن تصور داخلی خود از جهان
  - استنباط خصوصیات مخفی جهان
  - استنتاج فعالیتهای مناسب
- عامل پایگاه دانش خیلی شبیه به عاملهایی با حالت درونی است
- عاملها در سه سطح متفاوت تعریف میشوند:
  - سطح دانش: عامل چه چیزی میداند و اهداف آن کدامند؟
  - سطح منطقی : دانش به صورت جملات رمزگذاری می شود.
  - سطح پیاده سازی: ساختمان داده اطلاعات پایگاه دانش و چگونگی دستکاری آنها

عاملهای منطقی

جهان WUMPUS



معیار کارایی:

- ۱۰۰۰+ انتخاب طلا، ۱۰۰۰- افتادن در گودال یا خورده شدن، ۱- هر مرحله، ۱۰- برای استفاده از تیر

محیط:

- بوی تعفن در مربعهای همجوار WUMPUS
- نسیم در مربعهای همجوار گودال
- درخشش در مربع حاوی طلا
- کشته شدن WUMPUS با شلیک در صورت مقابله
- تیر فقط مستقیم عمل میکند
- برداشتن و انداختن طلا

حسگرها:

- بوی تعفن، نسیم، تابش، ضربه، جیغ زدن

محرکها:

- گردش به چپ، گردش به راست، جلو رفتن، برداشتن، انداختن، شلیک کردن

عاملهای منطقی

توصیف جهان WUMPUS

قابل مشاهده کامل: خیر، فقط ادراک محلی

قطعی: بله، نتیجه دقیقا مشخص است

رویدادی: خیر، ترتیبی از فعالیتهاست

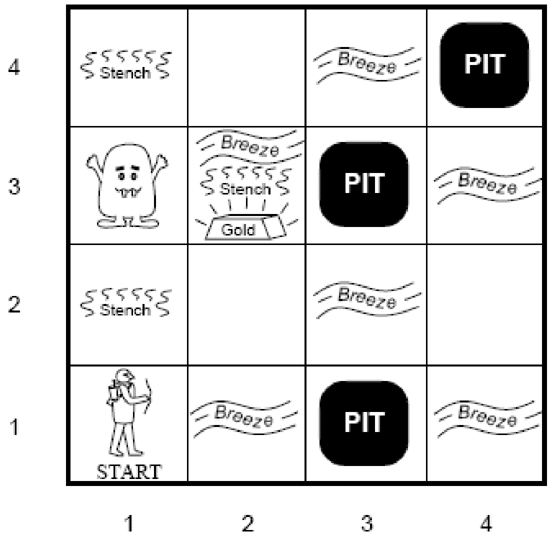
ایستا: بله، WUMPUS و گودالها حرکت ندارند

گسسته: بله

تک عامله: بله، WUMPUS در اصل یک خصوصیت طبیعی است

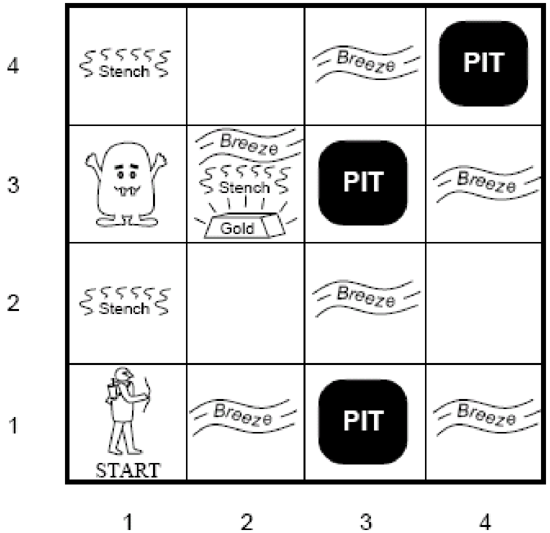
Move to A1,1

- R1 : A1,1 : ok
- ---------------------------
- R2 : ~ B1,1
- R3 : ~ S1,1
- --------------------------
- R4 : ~P2,1
- R5 : ~P1,2
- R6 : ~W2,1
- R7 : ~W1,2
- R8 : A2,1 ok
- R9 : A1,2 ok
- R10 : B1,1 ⇔ P1,2 ∨ P2,1



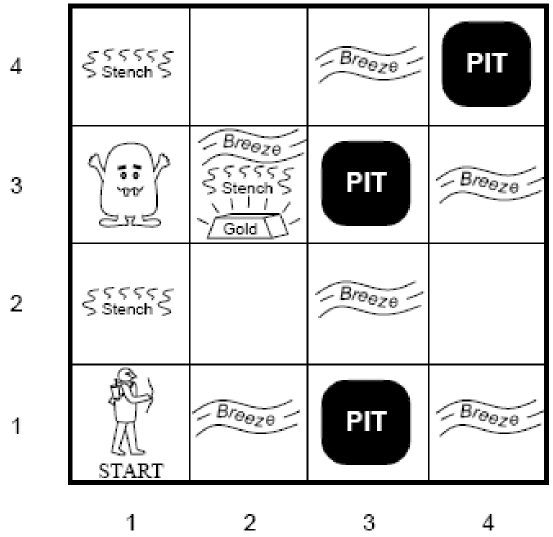
move to A2,1

- R11 : B2,1
- R12 : ~ S2,1
- --------------------------
- R13 : P2,2 ∨ P3,1
- R14 : ~W2,2
- R15 : ~W3,1
- R16 : B2,1⇔P1,1 ∨ P3,1 ∨ P2,2



back to A1,1 then move to A1,2

- R17 : ~B1,2
- R18 : S1,2
- ---------------------------
- R19 : W1,3 (R15 && R18)
- R20 : ~P2,2 (R17)
- R21 : P3,1 (R20 && R13)
- R22 : A2,2 OK
- R23 : $B1,2 \Leftrightarrow P1,1 \lor P1.3 \lor P2,2$



عاملهای منطقی

منطق

- یک زبان رسمی:
  - **ترکیب(نحو):** چه کلمه بندی صحیح است.(خوش فرم)
  - **معناشناسی:** یک کلمه بندی صحیح چه معنایی دارد
  - در منطق، معنای زبان، درستی هر جمله را در برابر هر جهان ممکن تعریف میکند

- مثال، در زبان ریاضیات
  - X+2 >= y یک جمله اما x2+y جمله نیست
  - X+2 >= y در جهان درست است اگر x=7 و y =1
  - X+2 >= y در جهان غلط است اگر x=0 و y =6

عاملهای منطقی

استلزام

- استلزام منطقی بین جملات این است که جمله ای بطور منطقی از جمله دیگر پیروی میکند

$$a \models b$$

  - جمله a استلزام جمله b است
  - جمله a جمله b را ایجاد میکند
  - اگر و فقط اگر، در هر مدلی که a درست است، b نیز درست است
  - اگر a درست باشد، b نیز درست است
  - درستی b در درستی a نهفته است

- مثال: جمله x+y=4 مستلزم جمله 4=x+y است

عاملهای منطقی

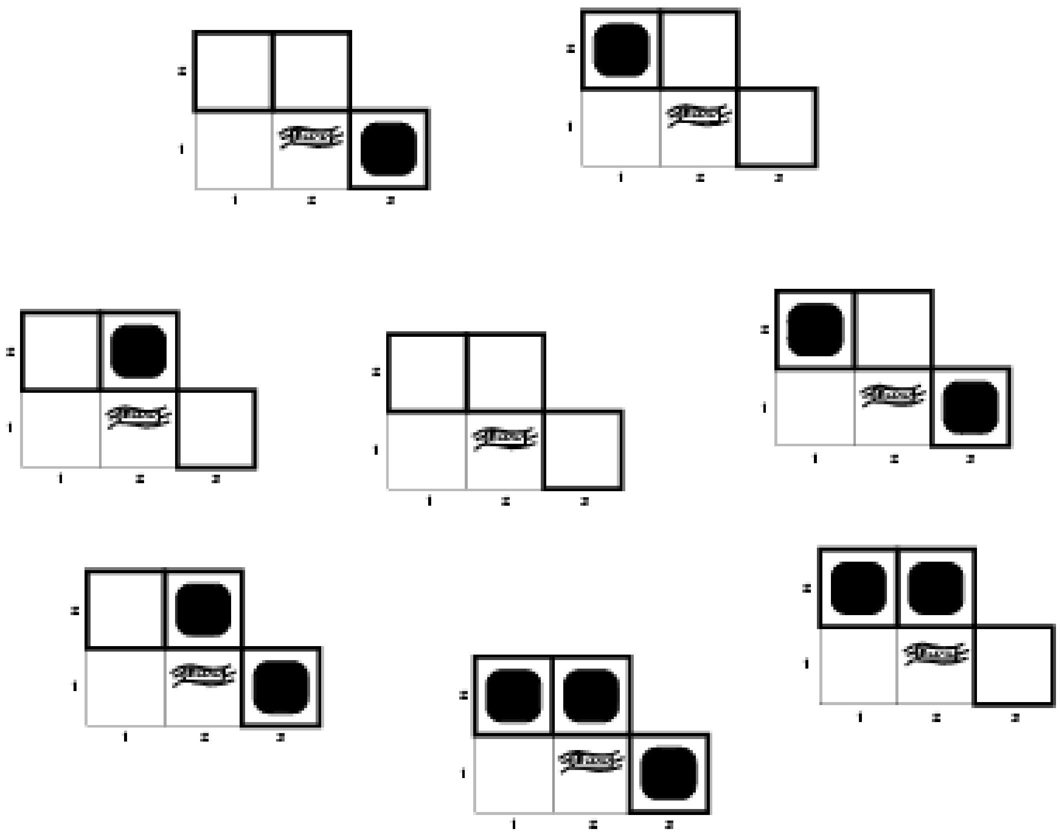
استلزام

در اغلب موارد پایگاه معرفت به عنوان یک عبارت در نظر گرفته می شود و ما درباره پایگاه معرفتی که یک جمله را مستلزم می سازد گفتگو می کنیم یعنی:

$$KB \models \alpha$$

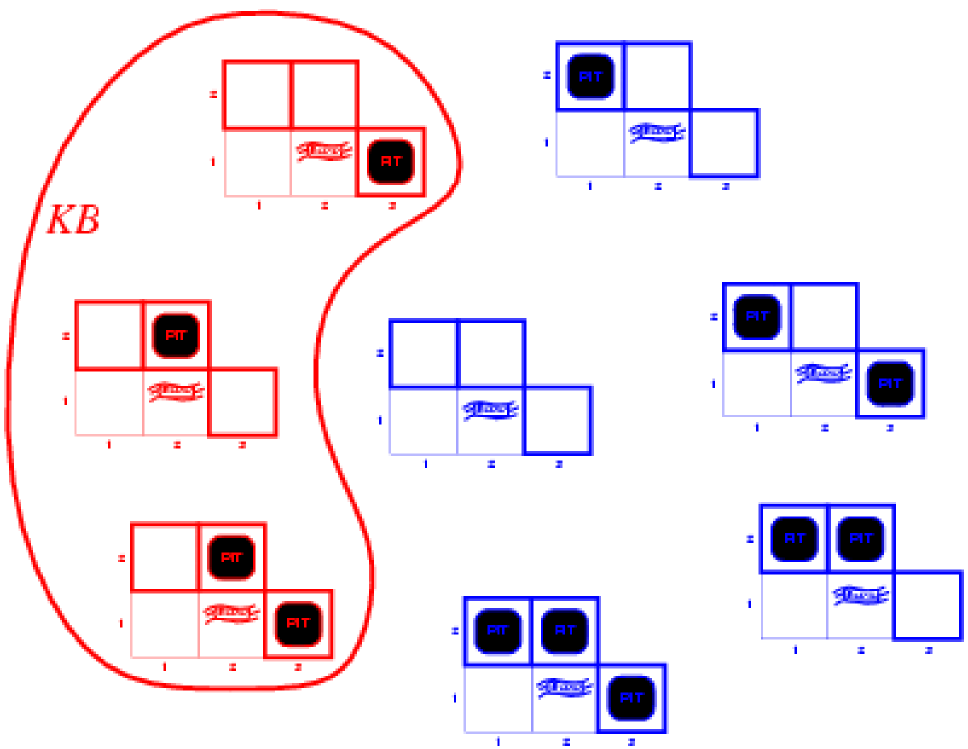
عاملهای منطقی

مدلهای Wumpus



عاملهای منطقی

مدلهای Wumpus



KB

=

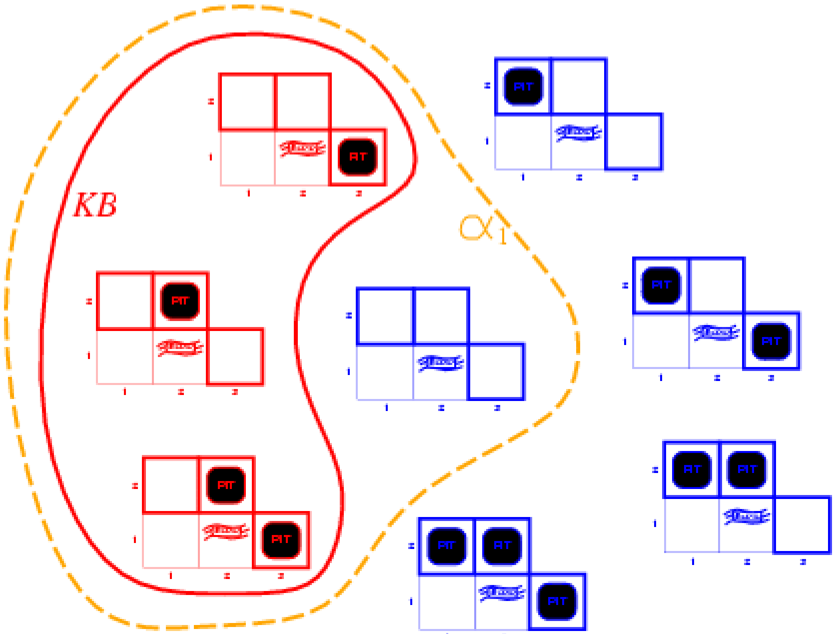
قوانین دنیای Wumpus

+

مشاهدات

عاملهای منطقی

مدلهای Wumpus

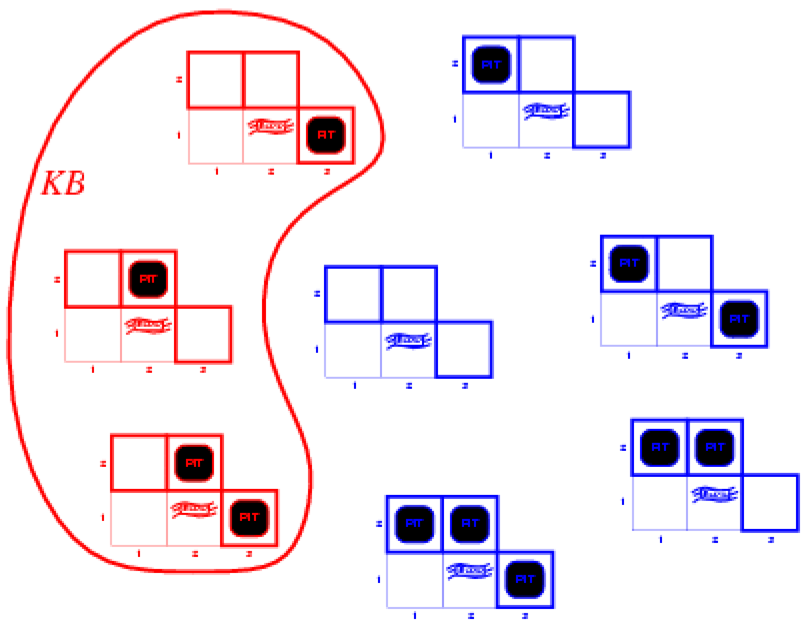


KB = wumpus مشاهدات + دنیای

α1 = "[1,2] امن است", *KB* ╞ α1

عاملهای منطقی

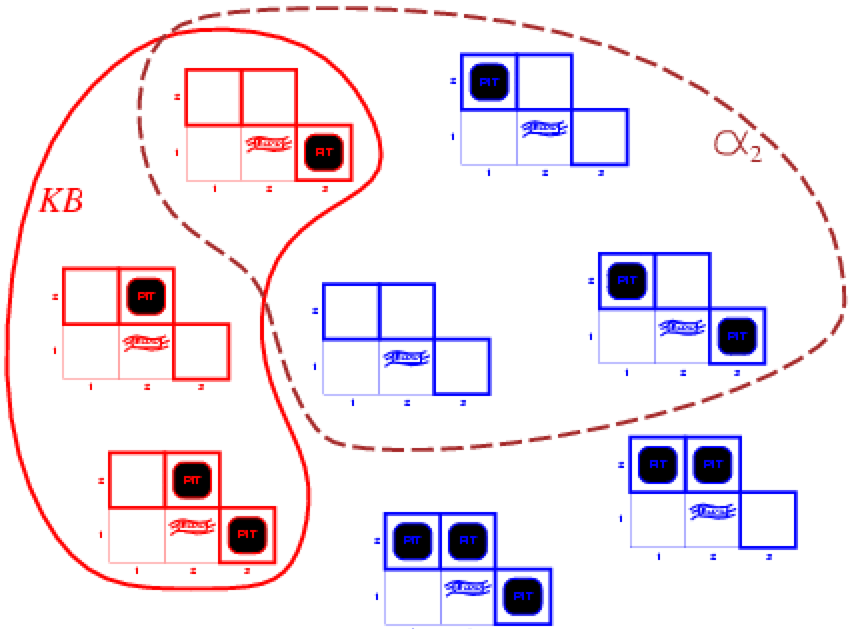
مدلهای Wumpus



KB = wumpus مشاهدات + دنیای

عاملهای منطقی

مدلهای Wumpus



KB = مشاهدات + دنیای wumpus

α_2 = "[2,2] امن است", $KB \not\models \alpha_2$

عاملهای منطقی

منطق گزاره ای

- نحو منطق گزاره ای، جملات مجاز را تعریف میکند
- جملات اتمیک(عناصر غیر قابل تعمیم) تشکیل شده از یک نماد گزاره
- هر یک از این نمادها به گزاره ای درست یا نادرست اختصاص دارد
  - نمادها از حروف بزرگ مثل P،Q،R استفاده میکنند
- لیترال : جملات اتمی مثبت یا منفی
- جملات پیچیده با استفاده از رابطهای منطقی، از جملات ساده تر ساخته میشوند
  - ┐ (not) جمله ای مثل $W_{1,3}$ ┐ نقیض $W_{1,3}$ است
  - لیترال یک جمله اتمیک(لیترال مثبت)، یا یک جمله اتمیک منفی(لیترال منفی) است
  - ^ (and) مثل $P_{1,3}$ ^ $W_{1,3}$ ترکیب عطفی نام دارد.هر بخش آن یک عطف نامیده میشود
  - v (or) مثل $(W_{1,3}$ ^ $P_{3,1})$ v $W_{2,2}$ ترکیب فصلی مربوط به فصل های $W_{2,2}$ و $W_{1,3}$ ^ $P_{3,1}$
  - => (استلزام): (W1,3 ^ P3,1) v ┐ W2,2 استلزام یا شرطی نامیده میشود. مقدمه یا مقدم آن $W_{1,3}$ ^ $P_{3,1}$ و نتیجه یا تالی آن $W_{2,2}$ ┐ است
  - ⇔ جمله $W_{2,2}$ ⇔ $W_{1,3}$ دو شرطی نام دارد

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathbf{True} \mid \mathbf{False} \mid \mathit{Symbol} \\
\mathit{Symbol} & \rightarrow & \mathbf{P} \mid \mathbf{Q} \mid \mathbf{R} \mid \ldots \\
\mathit{ComplexSentence} & \rightarrow & \neg\, \mathit{Sentence} \\
& \mid & (\, \mathit{Sentence} \wedge \mathit{Sentence}\,) \\
& \mid & (\, \mathit{Sentence} \vee \mathit{Sentence}\,) \\
& \mid & (\, \mathit{Sentence} \;\Rightarrow\; \mathit{Sentence}\,) \\
& \mid & (\, \mathit{Sentence} \;\Leftrightarrow\; \mathit{Sentence}\,)
\end{array}
$$

معانی (Semantic)

- معانی قوانینی برای تعیین صحت جمله نسبت به یک مدل خاص را تعیین می کنند.
- در منطق گزاره ها، مدل مقادیر درستی (درست یا نادرست) را برای هر نماد گزاره ای مشخص می کند.
  - برای مثال، اگر جمله ای در پایگاه معرفت از نمادهای گزاره ای استفاده کند، آنگاه یک مدل ممکن

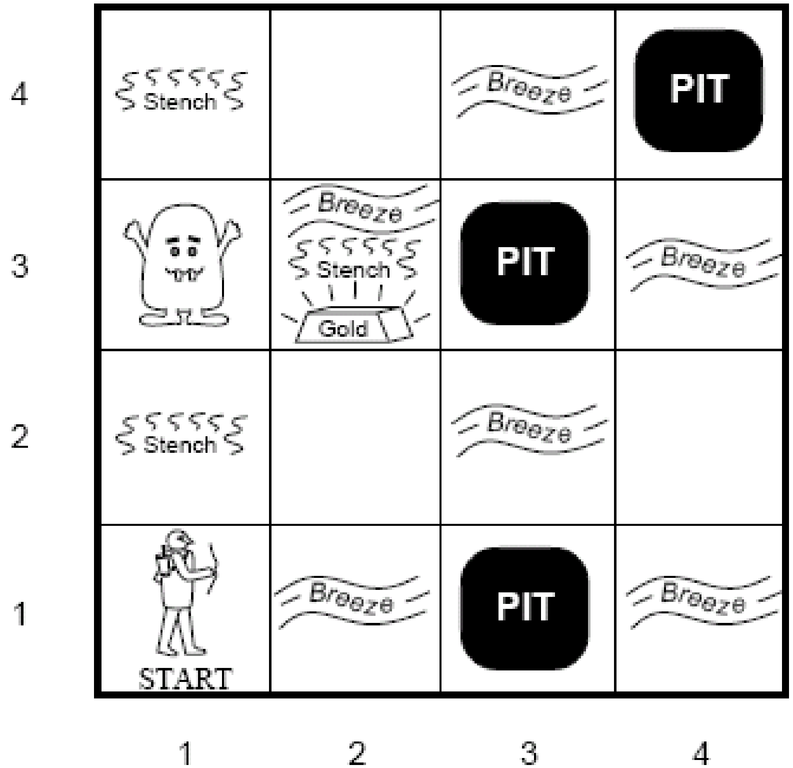
$$m_1 = \{P_{1,2} = false, P_{2,2} = false, P_{3,1} = true\}$$

- معنی در منطق گزاره ها باید مشخص کند که چگونه مقدار درستی هر جمله ای را با مدل داده شده محاسبه کنیم.

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

عاملهای منطقی

منطق گزاره ای در دنیای Wumpus



در $B_{1,1}$ نسیمی وجود دارد

$R2: B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

در [1,1] گودالی وجود ندارد

$R1: \neg P_{1,1}$

- چاهی در [1,1] نیست.

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

- خانه ای نسیم دارد اگر و تنها اگر در همسایگی آن چاهی باشد، این برای هر خانه ای صادق است، الان تنها خانه های مربوط را شامل می کنیم:

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

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

- جملات قبلی برای تمامی دنیاهای هیولا صحت دارند. حال درک نسیم برای دو خانه ملاقات شده اول در دنیای خاص عامل آن را به موقعیت شکل ۶-۳-b می رساند:

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

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

- پایگاه معرفت هم اکنون شامل جملات R_1 تا R_5 است. می توان آنها را با یک جمله واحد نشان داد زیرا ادعا می کند تمامی جملات مجزا صحیح هستند.

$$R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$$

استنتاج (از طریق پیاده سازی مستقیم تعریف استلزام)

هدف استنتاج منطقی تصمیم گیری درباره $KB|=\propto$ برای جمله ای همانند $\propto$ است.

مدل ها را بررسی کن و کنترل کن $\propto$ در هر مدلی که KB در آن درست است، درست یا نه.

برای منطق گزاره ها، مدل ها جایگذاری های درست یا نادرست روی هر نماد گزاره ای هستند. اگر به مثال دنیای هیولا برگردیم، نمادهای گزاره ای مربوط $B_{1,1}, B_{2,1}, P_{1,1}, P_{1,2}, P_{2,1}, P_{2,2}, P_{3,1}$ هستند. با هفت نماد، $2^7=128$ مدل ممکن داریم که در سه تای آنها KB صحیح است (شکل ۶-۷). در این سه مدل، $\sim P_{1,2}$ صحیح است، پس چاهی در [۲و۱] نیست. از سوی دیگر، $P_{2,2}$ نیست.

استنتاج (از طریق پیاده سازی مستقیم تعریف استلزام)

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*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*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

استنتاج از طریق پیاده سازی مستقیم تعریف استلزام

R1: $A \vee B$

R2: $\sim A$

B

$KB = R1 \wedge R2$

$KB \models B$

A	B	R1	R2	KB
0	0	0	1	0
0	1	1	1	1
1	0	1	0	0
1	1	1	0	0

اعتبار

- جمله معتبر است اگر و تنها اگر در تمامی مدل ها درست باشد.

برای مثال، جمله $P \vee \sim P$ معتبر است. جملات معتبر اغلب **تاتولوژی** (tautology) نام دارند،

- بنابر تعریف استلزام می توانیم قضیه قیاس را بدست آوریم:
- $KB|=\alpha$ اگر و فقط اگر جمله $KB \rightarrow \alpha$ معتبر باشد.

استنتاج از طریق اعتبار

R1: $A \vee B$

R2: $\sim A$

B

$KB = R1 \wedge R2$

$KB \models B$

A	B	R1	R2	KB	VALIDITY KB→B
0	0	0	1	0	1
0	1	1	1	1	1
1	0	1	0	0	1
1	1	1	0	0	1

عاملهای منطقی

منطق گزاره ای

$$
\begin{array}{rcll}
(\alpha \wedge \beta) & \equiv & (\beta \wedge \alpha) & \text{commutativity of } \wedge \\
(\alpha \vee \beta) & \equiv & (\beta \vee \alpha) & \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) & \equiv & (\alpha \wedge (\beta \wedge \gamma)) & \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) & \equiv & (\alpha \vee (\beta \vee \gamma)) & \text{associativity of } \vee \\
\neg(\neg\alpha) & \equiv & \alpha & \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\beta \Rightarrow \neg\alpha) & \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\alpha \vee \beta) & \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv & ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) & \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) & \equiv & (\neg\alpha \vee \neg\beta) & \text{de Morgan} \\
\neg(\alpha \vee \beta) & \equiv & (\neg\alpha \wedge \neg\beta) & \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) & \equiv & ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) & \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) & \equiv & ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) & \text{distributivity of } \vee \text{ over } \wedge
\end{array}
$$

ارضاء پذيري

- جمله ارضاء شدنی است اگر در برخی مدل ها درست باشد.
- ارضاء پذیری از طریق امتحان کردن مدل های ممکن تا یافتن یک جمله ارضاء شده کنترل می شود.
- استنتاج از طریق تعریف ارضا پذیری بصورت زیر انجام می شود:

$KB|=\alpha$ اگر و فقط اگر جمله $KB\wedge\sim\alpha$ ارضا ناپذیر باشد.

استنتاج از طریق ارضاپذیری

R1: $A \lor B$

R2: $\sim A$

B

$KB = R1 \land R2$

$KB \models B$

A	B	R1	R2	KB	VALIDITY $KB \rightarrow B$	SATISFIABILITY $KB \land \sim B$
0	0	0	1	0	1	0
0	1	1	1	1	1	0
1	0	1	0	0	1	0
1	1	1	0	0	1	0

الگوهای استدلال در منطق گزاره ای

- قوانین استنتاج: الگوهایی استاندارد که زنجیره ای از نتایج را برای رسیدن به هدف ایجاد میکند

- **قیاس استثنایی: قانون انتظار : Modus Ponens**

R1 : A → B

R2: A

B

(WumpusAhead ^ WumpusAlive)

و

(WumpusAhead ^ WumpusAlive) => Shoot

Shoot

عاملهای منطقی

$\alpha \wedge \beta$

α

β

حذف and: هر عطف را میتوان از ترکیب عطفی استنتاج کرد

مثال: WumpusAlive را میتوان از جمله زیر استنتاج کرد
(WumpusAhead ^ WumpusAlive)

خاصیت یکنواختی

مجموعه ای از جملات استلزامی که فقط میتواند در صورت اضافه شدن اطلاعات به پایگاه دانش رشد کند.

$KB \models A \rightarrow (KB \wedge B) \models A$

- قانون هم ارزی: تمامی هم ارزی های منطقی شکل ۶-۸ می توانند به عنوان قوانین استنتاجی استفاده شوند. برای مثال، هم ارزی حذف دو شرطی دو قانون استنتاجی زیر را می دهد:

$$\frac{\propto \Leftrightarrow \beta}{(\propto \Rightarrow \beta) \wedge (\beta \Rightarrow \propto)}, \frac{(\propto \Rightarrow \beta) \wedge (\beta \Rightarrow \propto)}{\propto \Leftrightarrow \beta}$$

- قانون OR introduction

اگر در رابطه زیر یک $\propto_i$ درست باشه پس $\propto_1 \vee \propto_2 \vee \propto_3 \ldots \vee \propto_n$ نیز درست است.

- قانون $\sim\sim\propto$

$$\frac{\sim\sim\propto}{\propto}$$

قانون resolution

- قانون resolution واحد، یک عبارت و یک لیترال را گرفته، عبارت دیگری تولید میکند

R1 : A v B

R2 : ~A

B

$L1 \vee L2 \vee .....Lk$

$\sim Li$

$L1 \vee L2...Li\text{-}1 \vee Li\text{+}1 \vee ...Lk$

- چاهی در [1,1] نیست.

$R_1: \sim \mathrm{P}_{1,1}$

- خانه ای نسیم دارد اگر و تنها اگر در همسایگی آن چاهی باشد، این برای هر خانه ای صادق است. الان تنها خانه های مربوط را شامل می کنیم:

$$R_2: \mathrm{B}_{1,1} \Leftrightarrow \left(\mathrm{P}_{1,2} \vee \mathrm{P}_{2,1}\right)$$
$$R_3: \mathrm{B}_{2,1} \Leftrightarrow \left(\mathrm{P}_{1,1} \vee \mathrm{P}_{2,2} \vee \mathrm{P}_{3,1}\right)$$

- جملات قبلی برای تمامی دنیاهای هیولا صحت دارند. حال درک نسیم برای دو خانه ملاقات شده اول در دنیای خاص عامل آن را به موقعیت شکل ۶-۳-b می رساند:

$$R_4: \sim \mathrm{B}_{1,1}$$
$$R_5: \mathrm{B}_{2,1}$$

پایگاه معرفت هم اکنون شامل جملات R_1 تا R_5 است. می توان آنها را با یک جمله واحد $R_1 \wedge \mathrm{R}_2 \wedge \mathrm{R}_3 \wedge \mathrm{R}_4 \wedge \mathrm{R}_5$ نشان داد زیرا ادعا می کند تمامی جملات مجزا صحیح هستند.

$KB \mid = \sim P_{1,2}$

ابتدا حذف دو شرطی را روی R_2 اعمال می کنیم.

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

سپس حذف AND را روی R6 اعمال می کنیم.

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

هم ارزی منطقی برای ضد مثبت می دهد:

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

حال Modus Ponens را روی R_8 , R_4 اعمال کرده تا:

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

در نهایت قانون دمورگان را اعمال می کنیم:

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

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

یعنی نه در [1,2] و نه در [2,1] چاهی نیست. این عملیات اثبات نام دارد.

شکل ۶-۴-a را در نظر بگیرید. عامل از [2,1] به [1,1] برگشته و سپس به [1,2] می رود که بوی تعفن احساس می کند و نسیمی در کار نیست. این واقعیات را به پایگاه معرفت اضافه می کنیم.

$R_{11}: \sim B_{1,2}$

$R_{12}: B_{1,2} \Leftrightarrow \left(\mathrm{P}_{1,1} \vee \mathrm{P}_{2,2} \vee P_{1,3}\right)$

ثابت کنید : $\sim P_{2,2}$, $\sim P_{1,3}$

$R_{11}: \sim \mathrm{B}_{1,2}$

قانون ترکیب دو شرطی :

$$R_{12}: [B_{1,2} \Longrightarrow (\mathrm{P}_{1,1} \vee \mathrm{P}_{2,2} \vee \mathrm{P}_{1,3})] \wedge [(\mathrm{P}_{1,1} \vee \mathrm{P}_{2,2} \vee \mathrm{P}_{1,3}) \Longrightarrow \mathrm{B}_{1,2}]$$

هم ارزی منطقی برای ضد مثبت:

$$\sim B_{1,2}$$

$$\sim \mathrm{B}_{1,2} \Rightarrow \sim (\mathrm{P}_{1,1} \vee \mathrm{P}_{2,2} \vee \mathrm{P}_{1,3})$$

طبق قانون Modus ponens نتیجه زیر گرفته می شود:

$$\sim (\mathrm{P}_{1,1} \vee \mathrm{P}_{2,2} \vee \mathrm{P}_{1,3})$$

دمورگان:

$$\sim \mathrm{P}_{1,1} \wedge \sim \mathrm{P}_{2,2} \wedge \sim \mathrm{P}_{1,3}$$

حذف AND:

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

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

به عنوان مثال دیگری ثابت کنید : $P_{3,1}$

با اعمال حذف شرط دو طرفی به R3 به همراه modus ponens به R5 به این واقعیت می رسیم که چاهی در [1,1] یا [۲و۲] یا [۱و۳] وجود دارد:

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

$R_5: B_{2,1}$

قانون ترکیب دو شرطی :

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

حذف AND و Modus Ponens:

$$\frac{\begin{array}{l} B_{2,1} \Rightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1}) \\ B_{2,1} \end{array}}{P_{1,1} \vee P_{2,2} \vee P_{3,1}}$$

حال اولین کاربرد قانون رزولوشن پیش می آید: لیترال $\sim P_{2,2}$ در R_{13} با لیترال $P_{2,2}$ در R_{15} حل شده و می دهد.

$$\frac{\begin{array}{l} R_{15}: (P_{1,1} \vee P_{2,2} \vee P_{3,1}) \\ R_{13}: \sim P_{2,2} \end{array}}{P_{1,1} \vee P_{3,1}}$$

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

به زبان فارسی اگر چاهی در یکی از خانه های [۱و۱]، [۲و۲] و[۱و۳] باشد و در [۲و۲] نباشد پس در

حل شده تا داشته باشیم: R_{16} در $P_{1,1}$ بالیترال $\sim R_{1,1}$: R_1[۱و۱] یا [۱و۳] داشت. مشابهاً

$$\frac{\begin{array}{l} R_{16}: P_{1,1} \vee P_{3,1} \\ R_1 : \sim P_{1,1} \end{array}}{P_{3,1}}$$

R17 :$P_{3,1}$ به زبان فارسی اگر چاه در [۱و۱] یا [۱و۳] باشد و در [۱و۱] نباشد، پس در [۱و۳] است. دو

مرحله استنتاجی آخر مثال هایی از رزولوشن واحد قوانین استنتاجی هستند.

فرم نرمال عطفی Conjunctive Normal Form (CNF)

قانون رزولوشن تنها قابل اعمال به ترکیب فصلی لیترالهاست و هر جمله منطق گزاره ها نیز هم ارز منطقی با یک ترکیب عطفی از ترکیبات فصلی لیترالهاست.

جمله ای که به صورت ترکیب عطفی از ترکیبات فصلی لیترالها بیان شوم فرم CNF دارد.

$(l_{1,1} \vee ... \vee l_{1,k}) \wedge .... \wedge (l_{n,1} \vee ... \vee l_{n,k})$

به هر ترکیب فصلی از لیترالها یک کلاز (clause) می گویند. هر قانون را می توان به صورت CNF

نوشت. قانون R_2 به فرم CNF به صورت زیر است :

عاملهای منطقی

الگوریتم resolution

- **شکل نرمال عطفی(CNF):** جمله ای که بصورت ترکیب عطفی از ترکیبات فصلی لیترالها بیان میشود.در هر عبارت موجود در جمله k-CNF دقیقا k لیترال وجود دارد

$$(l_{1,1} \vee \ldots \vee l_{1,k}) \wedge \ldots \wedge (l_{n,1} \vee \ldots \vee l_{n,k})$$

- الگوریتم resolution:
  - برای اینکه نشان دهیم KB|=a، مشخص میکنیم (KB ^ ┐a) ارضا کننده نیست
  - ابتدا (KB ^ ┐a) را به CNF تبدیل میکنیم
  - سپس قانون resulotion به عبارات کوچک حاصل اعمال میشود
  - هر جفتی که شامل لیترالهای مکمل باشد، resulotion میشود تا عبارت جدیدی ایجاد گردد
  - اگر این عبارت قبلا در مجموعه نباشد، به آن اضافه میشود
  - فرایند تا محقق شدن یکی از شروط زیر ادامه می یابد:
    - هیچ عبارت دیگری وجود نداشته باشد که بتواند اضافه شود. در این مورد، b استلزام a نیست
    - کاربرد قانون resolution، عبارت تهی را بدست میدهد که در این مورد، b استلزام a است

procedure. We illustrate the procedure by converting R_2, the sentence $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$, into CNF. The steps are as follows:

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

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

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$:

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

3. CNF requires $\neg$ to appear only in literals, so we "move $\neg$ inwards" by repeated application of the following equivalences from Figure 7.11:

$$\begin{aligned}
&\neg(\neg\alpha) \equiv \alpha \quad \text{(double-negation elimination)}\\
&\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta) \quad \text{(De Morgan)}\\
&\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta) \quad \text{(De Morgan)}
\end{aligned}$$

In the example, we require just one application of the last rule:

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

4. Now we have a sentence containing nested $\wedge$ and $\vee$ operators applied to literals. We apply the distributivity law from Figure 7.11, distributing $\vee$ over $\wedge$ wherever possible.

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

استنتاج براساس رزولوشن و فرم نرمال عطفی:

روال استنتاج با رزولوشن برای اثبات هر جمله از برهان خلف استفاده می کند. یعنی برای نمایش اینکه $kB|{=}\propto$ است نشان می دهد که $(kB \wedge \sim \propto)$ دارای تناقض بوده و ارضا ناپذیر است.

الگوریتم رزولوشن اول $(kB \wedge \sim \propto)$ را به CNF تبدیل می کند. سپس قانون رزولوشن به کلازهای حاصل اعمال خواهد شد. هر جفت که شامل لیترال های متضاد است با هم ادغام شده تا تولید یک کلاز جدید بکند، که به مجموعه اضافه خواهد شد. این فرآیند ادامه یافته تا یکی از موارد زیر روی دهد:

- کلاز جدیدی برای افزودن وجود نداشته باشد، که در این حالت β, α را مستلزم نمی سازد و اثبات صورت نمی گیرد.
- به کارگیری قانون رزولوشن یک کلاز تهی تولید کند که در این حالت β, α را مستلزم خواهد ساخت. چون کلاز تهی هم ارز با نادرست است. و بنابراین

$$kB \wedge \sim\alpha \equiv F$$
$$\sim \propto \equiv F$$
$$\propto \equiv T$$

مثال در دنیای هیولا:

وقتی عامل در [۱و۱] است هیچ نسیمی نیست پس چاهی در همسایگی نمی تواند وجود داشته باشد

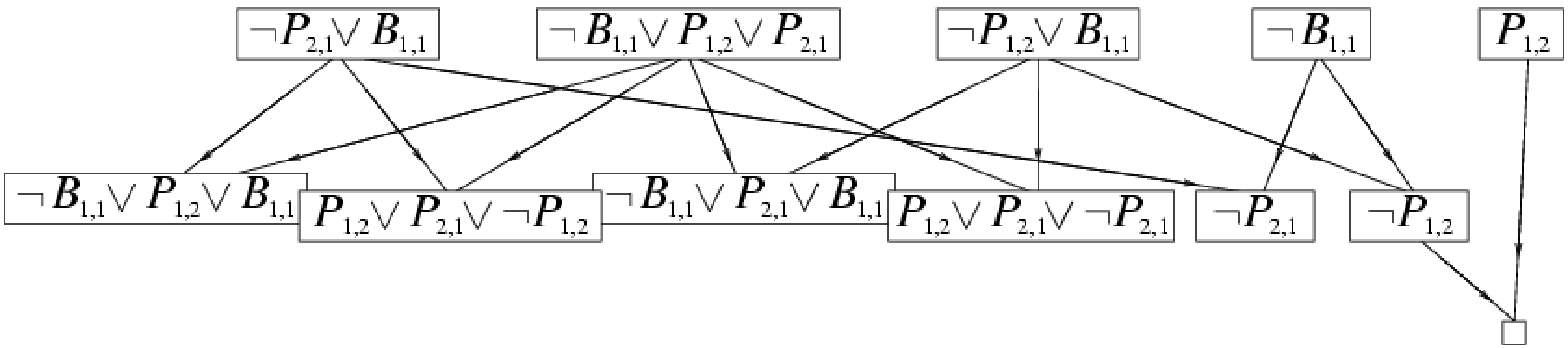
ثابت کنید $\sim P_{1,2}$

پایگاه معرفت مربوط به قرار:

$$KB = R_2 \wedge R_4 = \left(\left(B_{1,1} \Leftrightarrow P_{1,2} \vee P_{2,1}\right) \wedge \sim B_{1,1}\right)$$

است و ما می خواهیمα را ثابت کنمی که همان $\sim P_{1,2}$ است. در زمان تبدیل $(kB \wedge \sim\alpha)$ به CNF کلازهای نشان داده شده در بالای شکل ۶-۹ تولید خواهد شد. سطر دوم شکل تمامی کلازهای حاصل از اعمال رزولوشن جفتی بر سطر اول را نشان می دهد. اگر $P_{1,2}$ را با $\sim P_{1,2}$ ادغام کنیم به کلاز تهی منجر می شود که با مربع کوچکی نشان داده شده است. وارسی شکل ۶-۹ آشکار می سازد که مراحل رزولوشن پایان ناپذیرند. برای مثال کلاز $B_{1,1} \vee \sim B_{1,1} \vee P_{1,2}$ با $Trune \vee P_{1,2}$ هم ارز است که با True درست است کمک چندانی نمی کند. پس هر کلازی را که در آن دو لیترال متضاد وجود دارد دور می ریزیم.

مثال:الگوریتم resolution



KB = R1 ∧ R2

KB = (B1,1 ⇔ (P1,2∨ P2,1)) ∧ (¬ B1,1)

α = ¬P1,2

زنجیر پیشرو و عقبگرد

- **عبارات هورن**: ترکیب فصلی لیترالهایی است که فقط یکی از آنها مثبت است
  - هر عبارت **هورن** را میتوان به صورت یک استلزام نوشت که مقدمه آن ترکیب عطفی لیترالهای مثبت و تالی آن یک لیترال مثبت است
  - این نوع عبارات هورن که فقط یک لیترال مثبت دارند، **عبارات معین** نامیده میشوند
  - لیترال مثبت را **رأس** و لیترالهای منفی را **بدنه عبارت** گویند
  - عبارت معینی که فاقد لیترالهای منفی باشد، گزاره ای بنام **حقیقت** نام دارد
  - **عبارات معین** اساس **برنامه نویسی منطقی** را میسازد
- استنتاج با عبارات هورن، از طریق الگوریتم های زنجیر پیشرو و زنجیر عقبگرد انجام میگیرد

زنجیر پیشرو

الگوریتم زنجیر پیشرو تعیین میکند آیا نماد گزاره ای q(تقاضا)، توسط پایگاه دانش عبارات هورن ایجاب میشود یا خیر

R1 : ~P ∨ Q

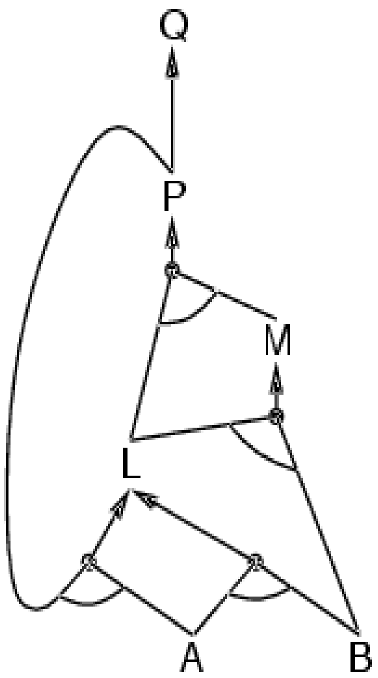
R2 : ~L ∨ ~M ∨ P = ~(L ∧ M) ∨ P

R3: ~B ∨ ~L ∨ M = ~(B ∧ L) ∨ M

R4:: ~A ∨ ~P ∨ L = ~(A ∧ P) ∨ L

R5: ~A ∨ ~B ∨ L = ~(A ∧ B) ∨ L

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



الگوریتم زنجیره پیش رو

```
function PL-FC-ENTAILS?(KB, q) returns true or false
  inputs: KB, the knowledge base, a set of propositional Horn clauses
          q, the query, a proposition symbol
  local variables: count, a table, indexed by clause, initially the number of premises
                   inferred, a table, indexed by symbol, each entry initially false
                   agenda, a list of symbols, initially the symbols known to be true in KB

  while agenda is not empty do
     p ← POP(agenda)
     unless inferred[p] do
        inferred[p] ← true
        for each Horn clause c in whose premise p appears do
           decrement count[c]
           if count[c] = 0 then do
              if HEAD[c] = q then return true
              PUSH(HEAD[c], agenda)
  return false
```

زنجیر پیشرو

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

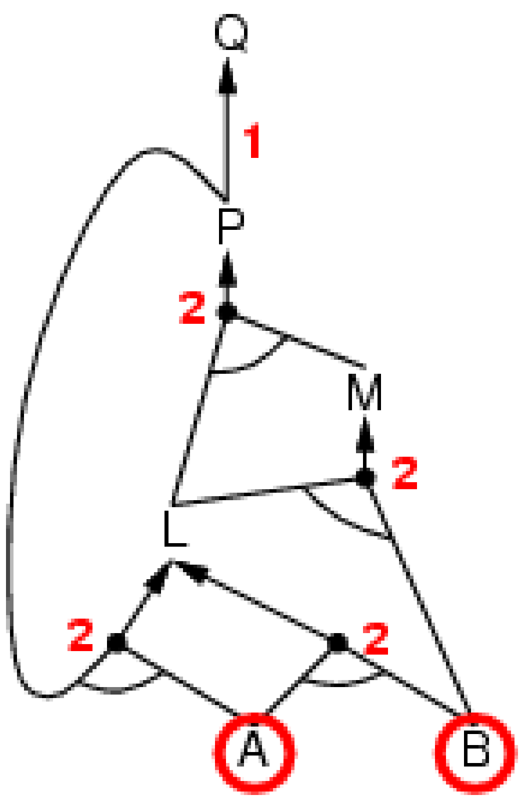
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



زنجیر پیشرو

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

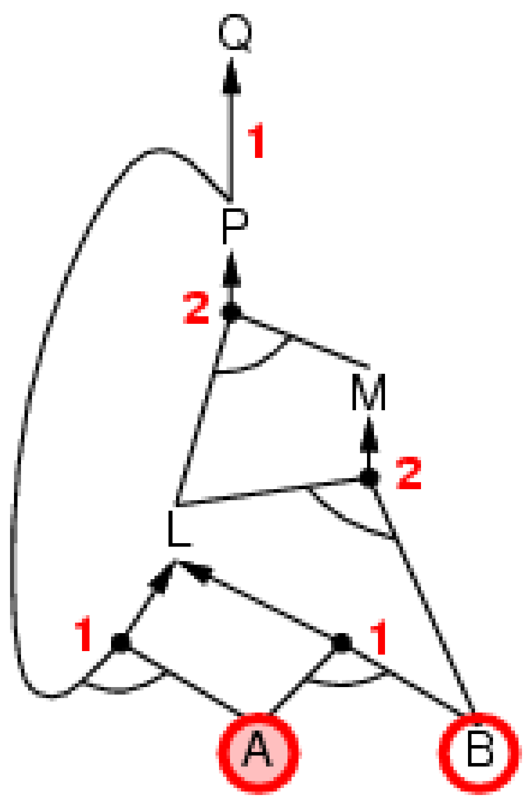
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



زنجیر پیشرو

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

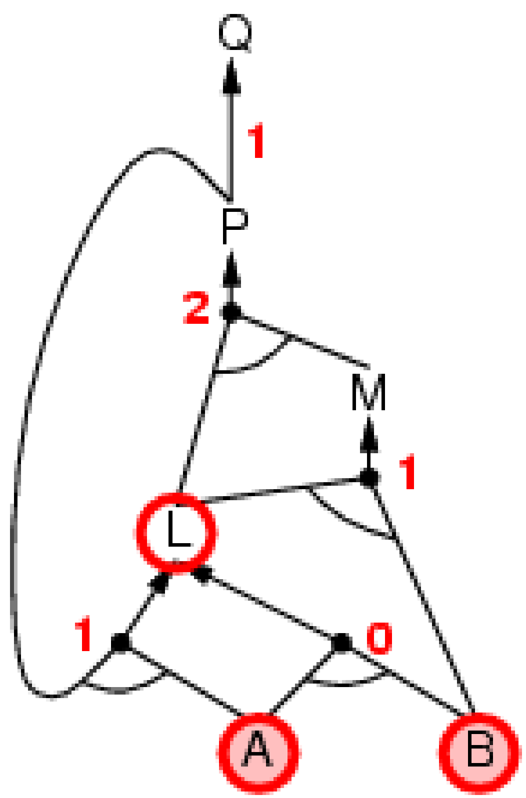
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

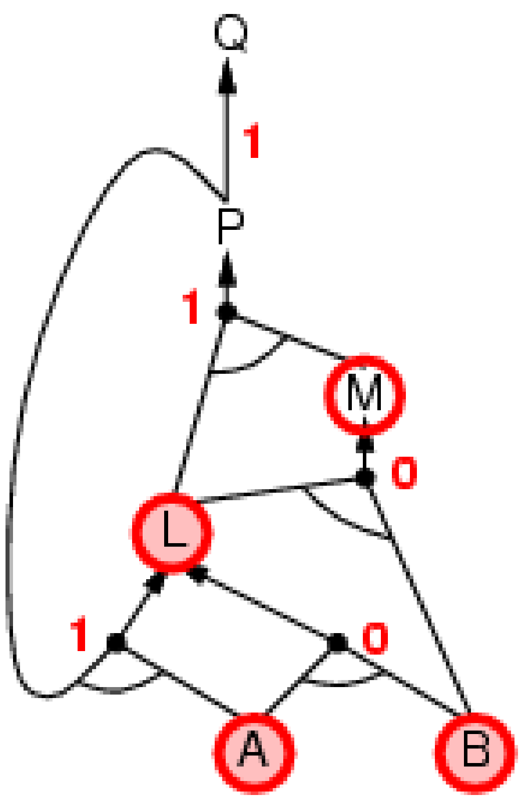
A

B



زنجیر پیشرو

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



زنجیر پیشرو

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

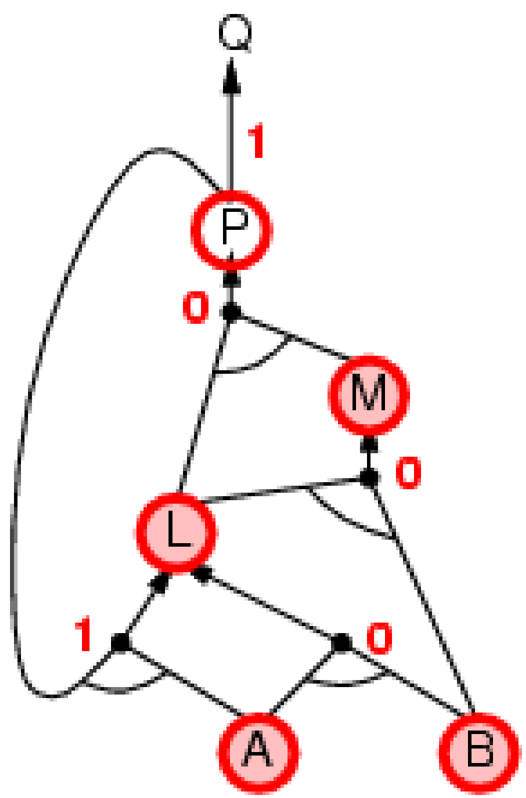
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

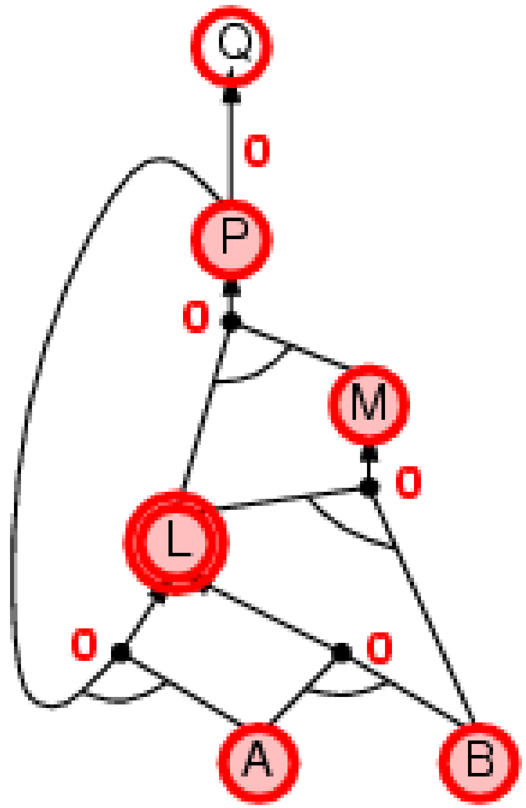
A

B



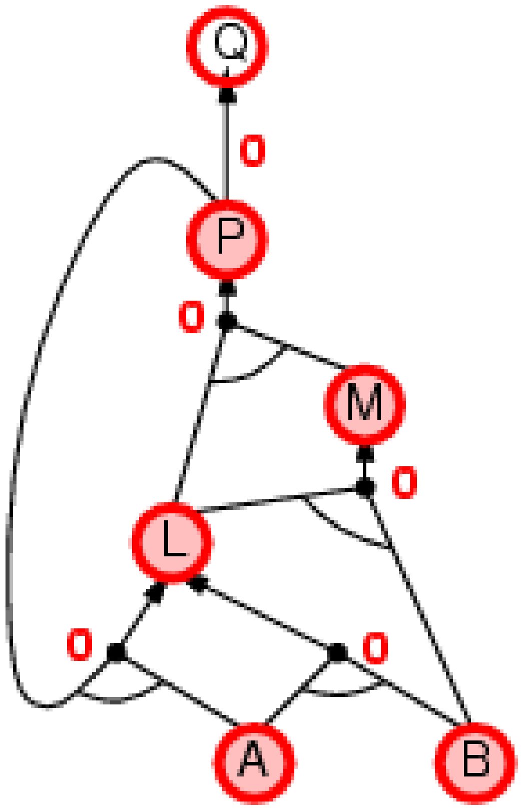
زنجیر پیشرو

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



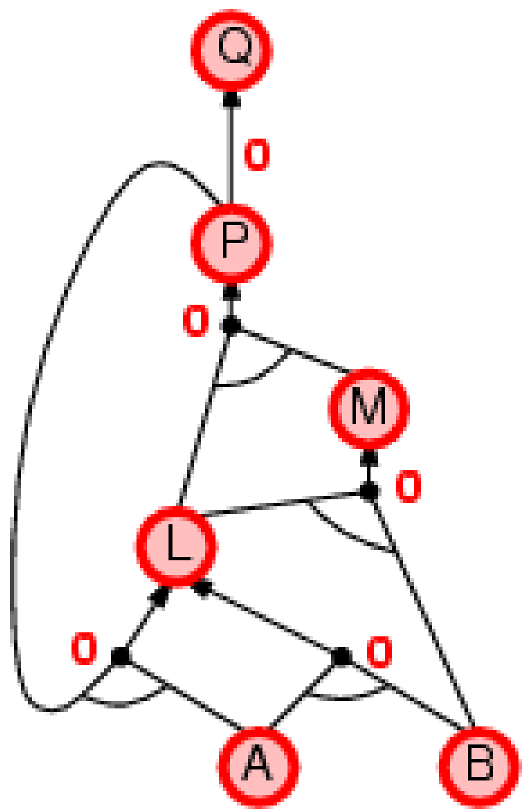
زنجیر پیشرو

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



زنجیر پیشرو

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



الگوریتم عقبگرد کامل

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

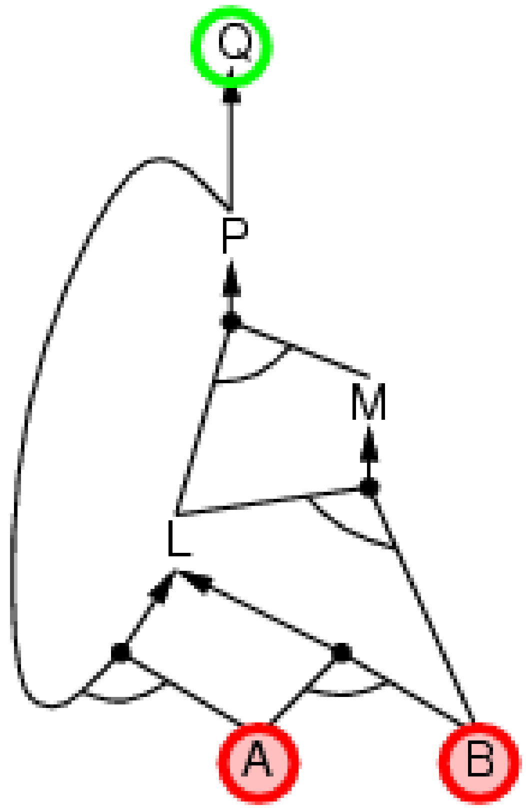
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

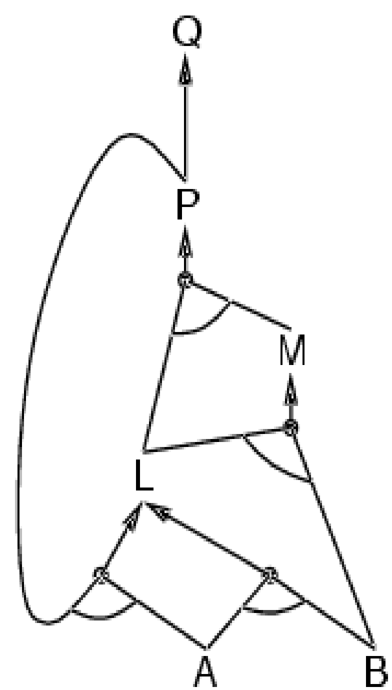
B



الگوریتم عقبگرد کامل

- تغییرات عمده: خاتمه زودرس، اکتشاف نماد محض، اکتشاف عبارت واحد

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



الگوریتم عقبگرد کامل

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

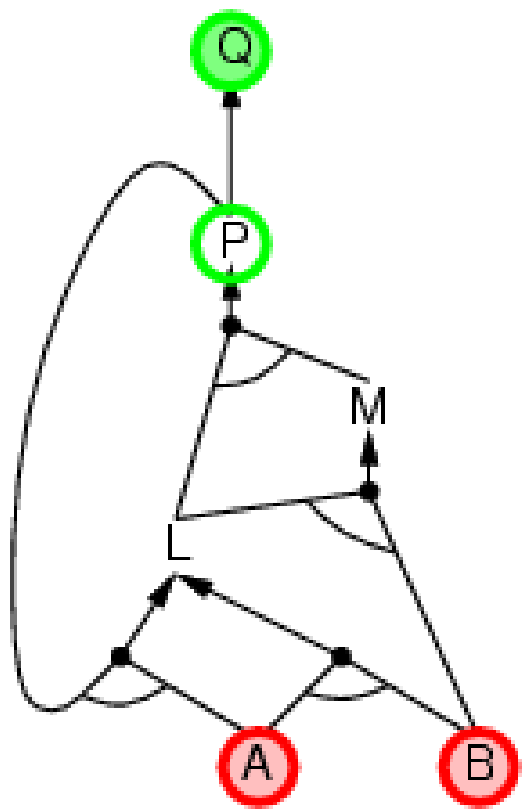
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



الگوریتم عقبگرد کامل

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

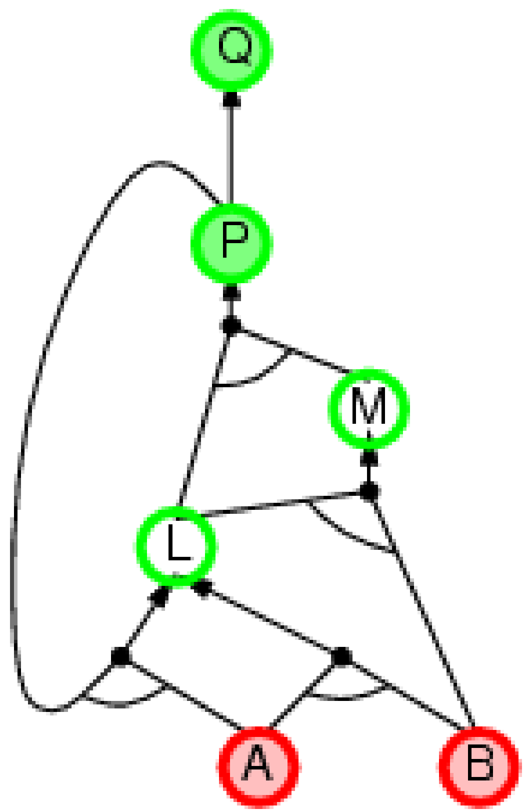
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



الگوریتم عقبگرد کامل

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

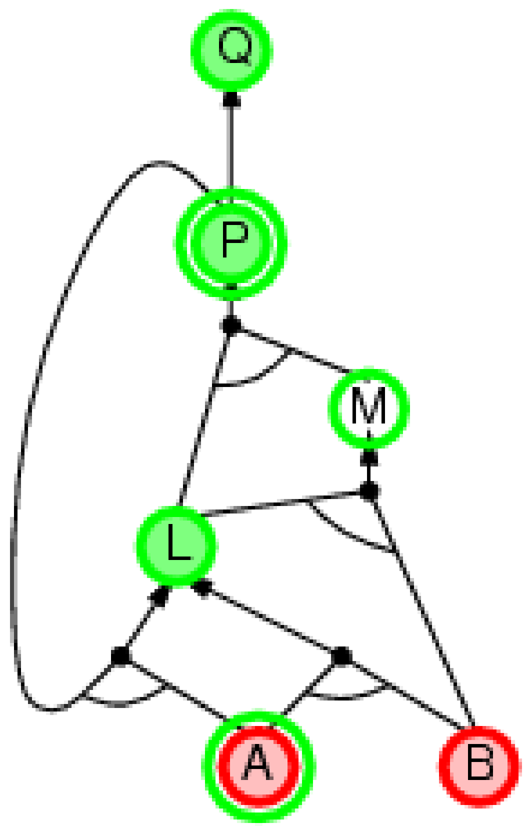
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

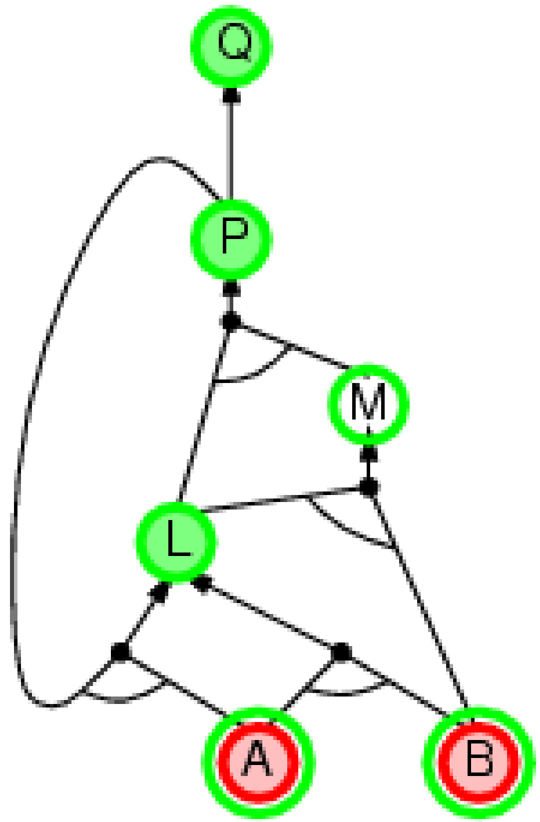
A

B



الگوریتم عقبگرد کامل

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



الگوریتم عقبگرد کامل

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

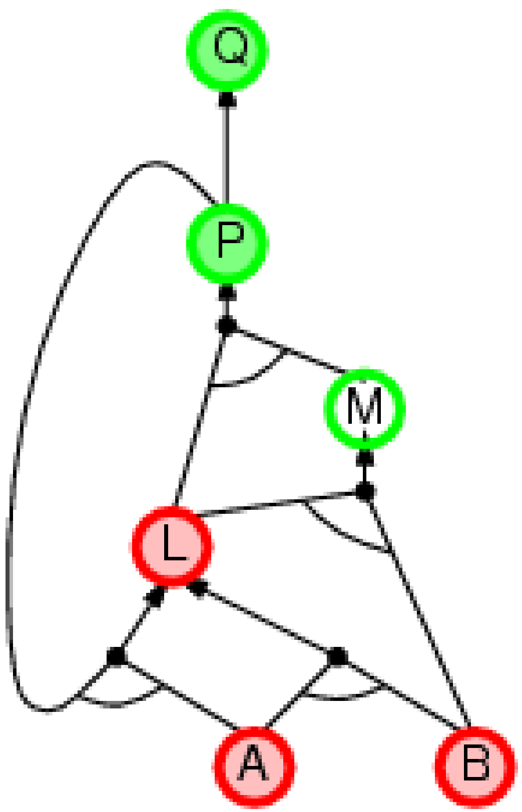
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

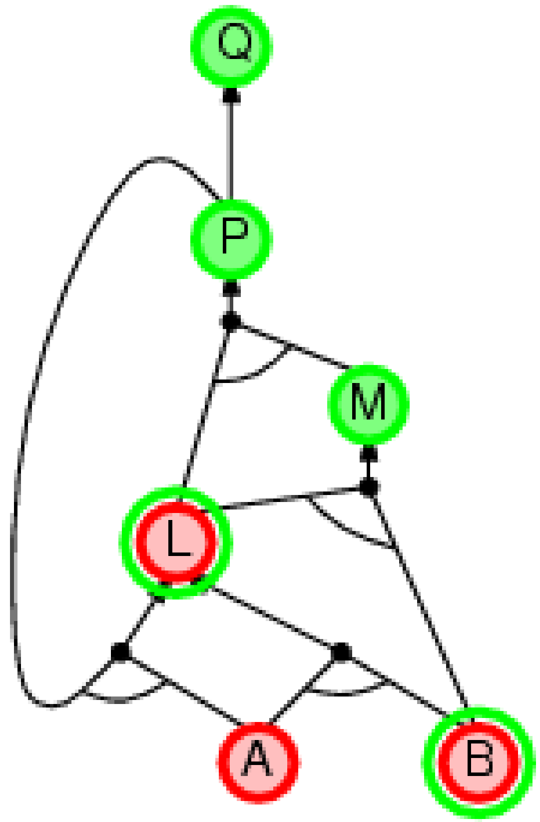
A

B



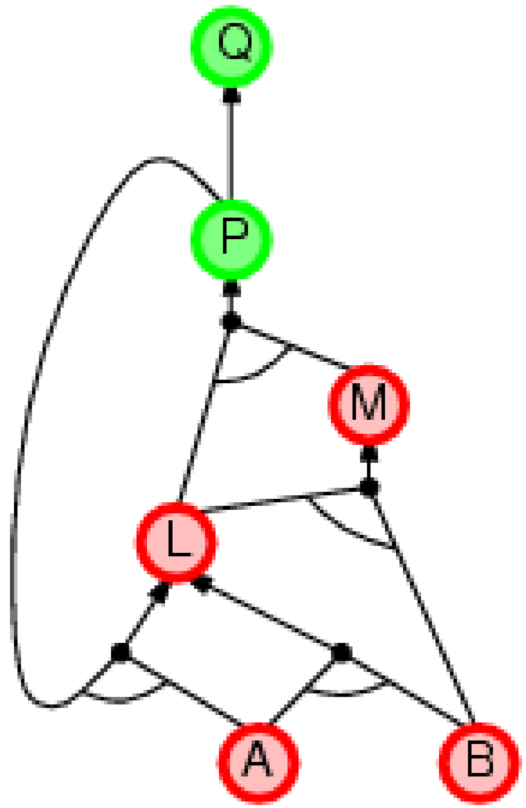
الگوریتم عقبگرد کامل

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



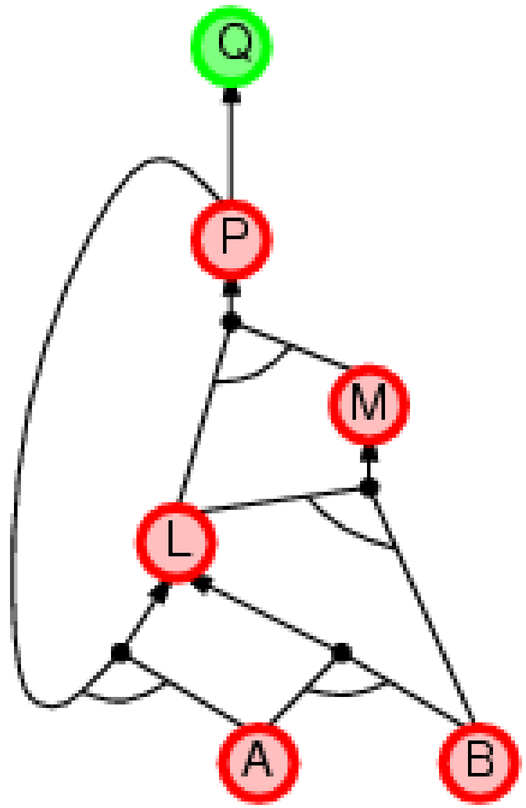
الگوریتم عقبگرد کامل

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



الگوریتم عقبگرد کامل

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



الگوریتم عقبگرد کامل

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B