هوش مصنوعی

فصل هفتم

عامل های منطقی

مدرس:ایمان مختاری
دانشگاه پیام نور شهرکرد

عاملهای مبتنی بر دانش (دارای دانش درونی)

دلایل مطالعه عاملهای مبتنی بر دانش

۱-دانش و استدلال برای عامل های مصنوعی اهمیت دارد.(برای موفقیت بیشتر موثرند)

۲-دانش و استدلال در ارتباط با محیط های نیمه رویت پذیر نیز نقش اساسی دارد(وجود دانش قبلی)

جان از پنجره الماس را دید و خواست آن را بردارد

جان آجر را به سمت پنجره پرت کرد و آن را شکست

آن چیست؟

۳-انعطاف پذیری عاملهای مبتنی بر دانش

امکان استفاده از دانش قبلی و فراگیری دانش جدید برای حل مسائل واضح

عامل های مبتنی بر دانش و معرفی پایگاه دانش (KB)

جزء اصلی عامل های مبتنی بر دانش، پایگاه دانش(KB) آن است.

- پایگاه دانش: مجموعه ای از جملات
- جمله: هر جمله در **زبانی**،**به نام** زبان بازنمایی دانش بیان می شود و اظهاری را درباره دنیا بازنمایی می کند.

برای اضافه کردن جملات به پایگاه دانش و پرس و جو از دانسته ها

TELLو ASK

هر دو وظیفه ممکن است دارای استنتاج(به دست آوردن جملات جدید و دید جدید از اطلاعات قبلی) باشند.

استنتاج باید به این صورت باشد که هرگاه سوالی از پایگاه دانش پرسیده شد پاسخ باید از آنچه قبلا به پایگاه دانش گفته شده(tell) نتیجه شود.

پایگاه دانش ، مجموعه ای از عبارت ها است که به وسیله ی یک زبان رسمی بیان شده اند .

یک عامل عمومی مبتنی بر دانش

```
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
```

همانند سایر عاملها درک ها (سنسورها) را دریافت و یک عمل بر می گرداند
۱-جمله ای شامل درک از محیط را به پایگاه دانش می گوید
۲-پایگاه دانش ، اقدام لازم را برای انجام بر می گرداند
۳-ثبت اقدام در پایگاه دانش

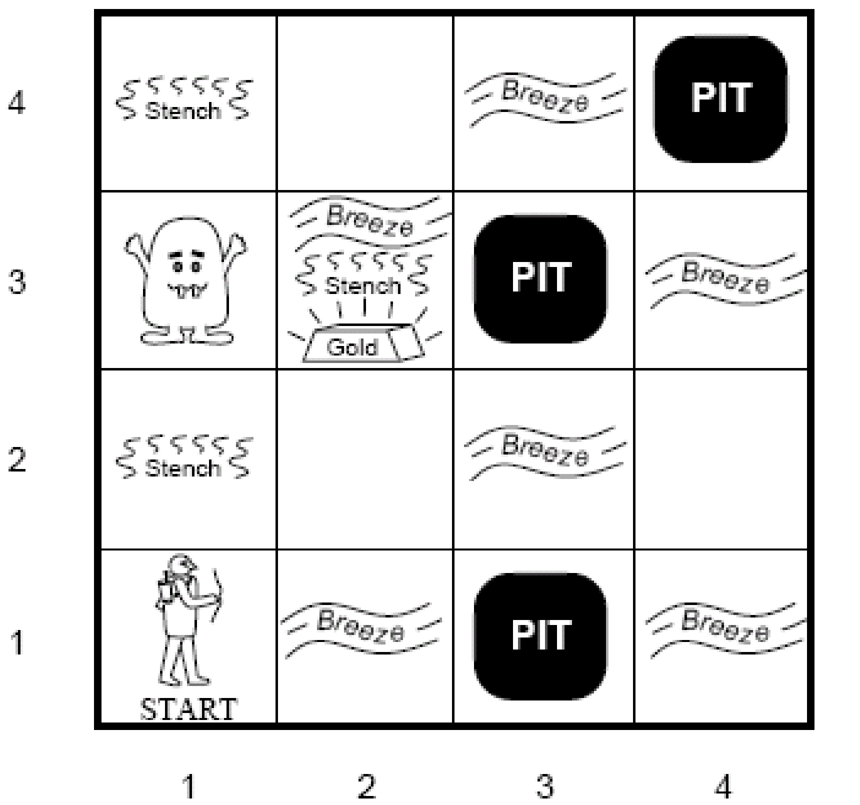
عاملهای مبتنی بر دانش

- عامل پایگاه دانش خیلی شبیه به عاملهایی با حالت درونی است که در فصل ۲ شرح داده شد.

- عاملها در دو(سه) سطح متفاوت تعریف می شوند:
  - سطح دانش: عامل چه می داند و اهدافش چیست تا رفتارش را تنظیم کند(بدون توجه به پیاده سازی)
  - (سطح منطقی): بیان دانش با جملات خاص زبان ((Wife(john,marya)
  - سطح پیاده سازی: ساختمان داده ای فیزیکی که عامل از آن استفاده می کند و چگونگی دستکاری آنها

- انواع رویکردها در ساخت سیستم
  - رویکرد اعلانی :دانش اولیه + ساخت زبان بازنمایی جهت کار با سیستم
  - رویکرد رویه ای:رفتارهای مطلوب را به صورت مستقیم و استدلالی وجود ندارد و به یک روش در زمان اجرا مورد استفاده قرار می گیرد
- تمامی قابلیت های عامل های مبتنی بر دانش نظیر بازنمایی، استدلال و فراگیری مدیون توسعه نظریه و فن آوری منطق در طول قرن های طولانی می باشد.

دنیای ومپوز



Breeze نسیم
PIT گودال
Stench بوی بد

دنیای ومپوز

فرض کنید وارد غاری شده اید که دارای اتاقهای مختلفی است و شما تیرکمانی با یک تیر دارید و قصد دارید طلایی که در غار مخفی است را پیدا کنید و از در ورودی خارج شوید،یک ومپوز انسانی بدبو و غارنشین است ممکن است ما را بکشد از سویی در غار تعدادی گودال وجود دارد که باعث مرگ ما می شوند در اتاقهایی همجوار سطری یا ستونی یک اتاق ،در صورت وجود گودال وزش نسیم در صورت وجود ومپوز بوی بد و درخشش درصورت وجود طلا حس خواهد شد

PEAS در ومپوز

- **معیار کارایی:**
  - ۱۰۰۰+ برداشتن طلا، ۱۰۰۰- افتادن در گودال یا خورده شدن توسط ومپوز، ۱- انجام هر اقدام، ۱۰- برای استفاده از تیر
- **محیط:**
  - اتاق های مشبک ٤*٤
  - شروع به حرکت عامل از مربع [۱و۱] به طرف راست
  - مکان های طلا و ومپوز مربع هایی به جز مربع شروع است و به صورت تصادفی با تابع توزیع یکنواخت انتخاب شده است.
  - هر مربع به جز مربع شروع می تواند با احتمال ۲/. گودال باشد.

دنیای ومپوز

- **اقدامات:**
  - ۹۰ درجه چرخش به چپ، ۹۰ درجه چرخش به راست، جلو رفتن، برداشتن شیئ، انداختن، تیراندازی در خط مستقیم، مرگ در هنگام ورود به مربع گودال یا ومپوز

- **حسگرها:**
  - بو بد،در مربع دارای ومپوز و مربع های مجاور
  - نسیم، در مربع های مجاور با گودال
  - درخشش، در مربعی که در آن طلاست
  - ضربه، هنگام برخورد با دیوار
  - فریاد، هنگام کشته شدن ومپوز

خصوصیات دنیای wumpus

قابل مشاهده بودن ؟؟ نه – فقط دارای ادراک محلی است .

قطعیّت ؟؟ بله – نتایج دقیقا مشخص است .

دوره ای نه – دارای ترتیب در سطح عملکردها است .

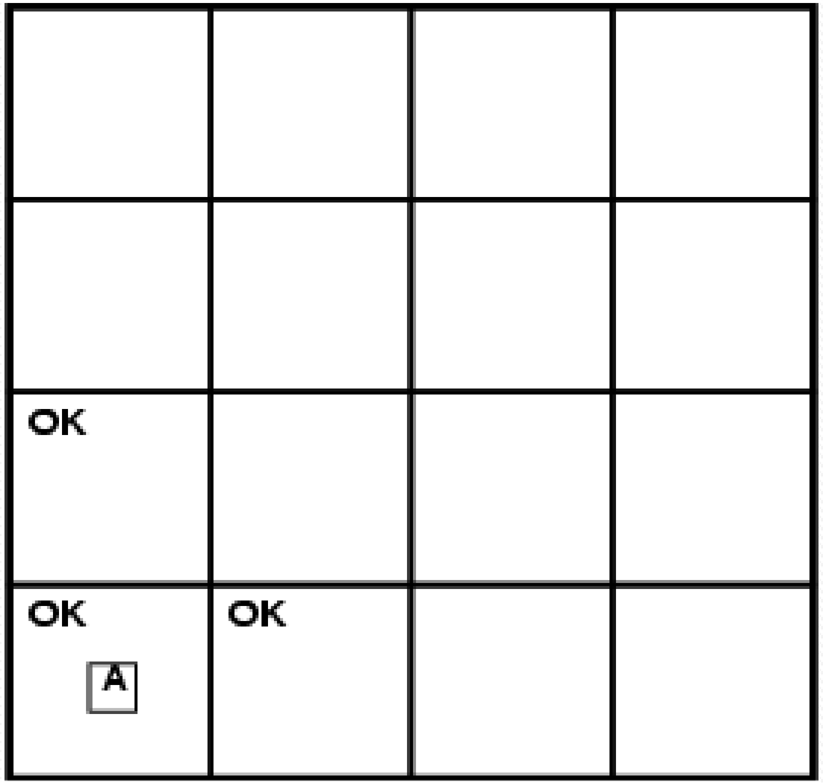
ایستایی ؟؟ بله – wumpus و چاله ها نمی توانند حرکت نمایند .

گسسته؟؟ بله .

تک عاملی ؟؟ بله wumpus در اصل یک طرح طبیعی می باشد .

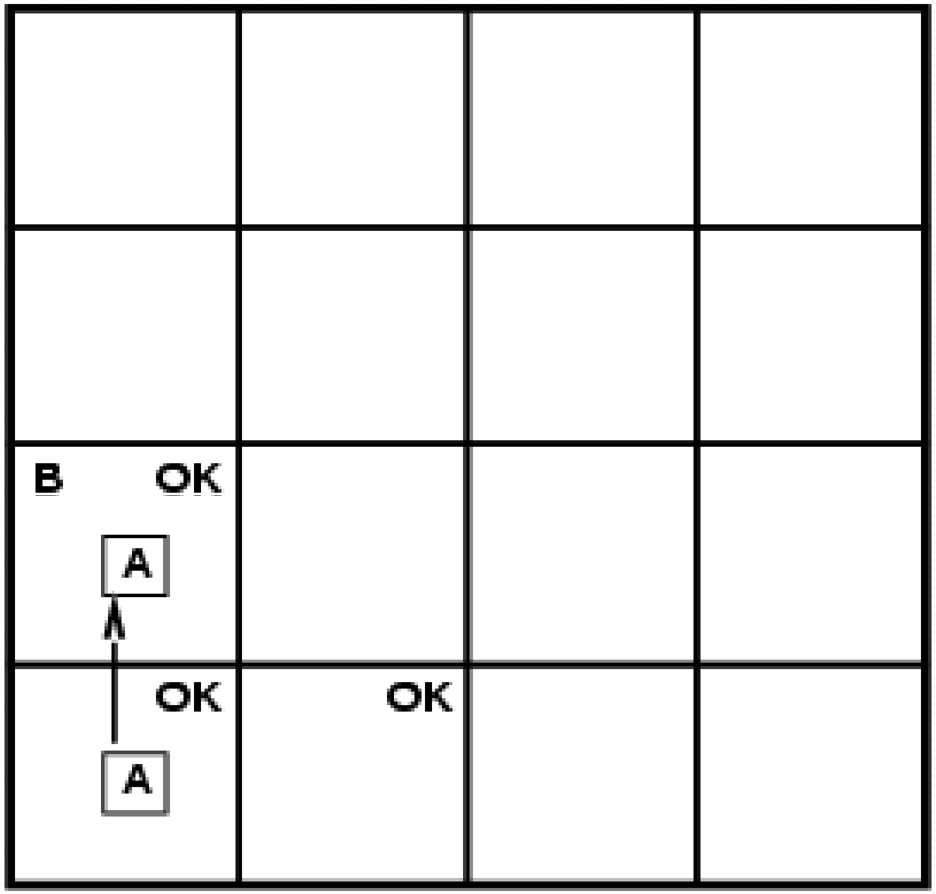
دنیای ومپوز

A = عامل
B = نسیم
G = درخشش طلا
OK = مربع امن
P = گودال
S = بوی بد
V = ملاقات شده
W =ومپوز



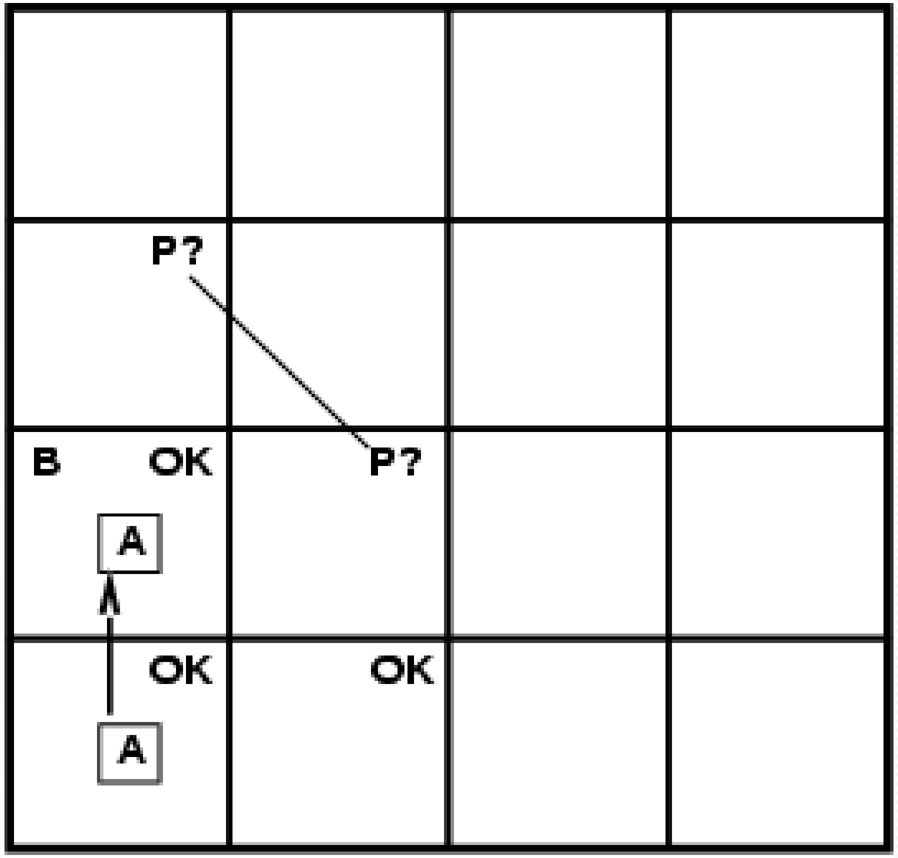
دنیای ومپوز

A = عامل
B = نسیم
G = درخشش طلا
OK = مربع امن
P = گودال
S = بوی بد
V = ملاقات شده
W =ومپوز



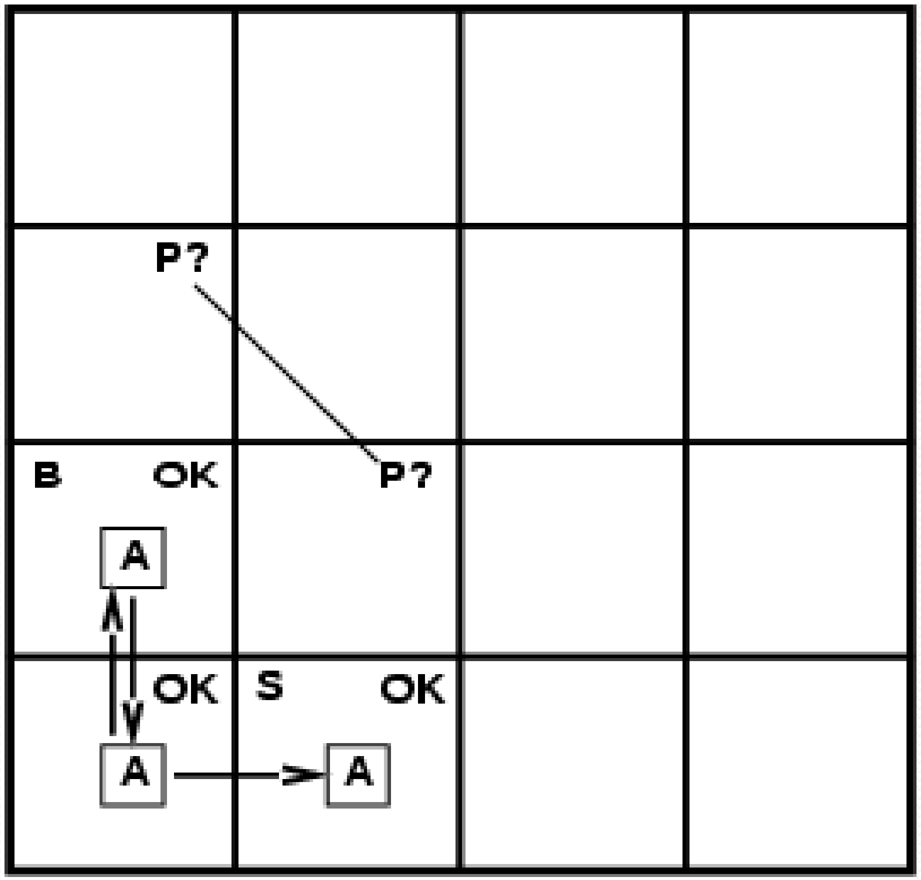
دنیای ومپوز

A = عامل
B = نسیم
G = درخشش طلا
OK = مربع امن
P = گودال
S = بوی بد
V = ملاقات شده
W =ومپوز



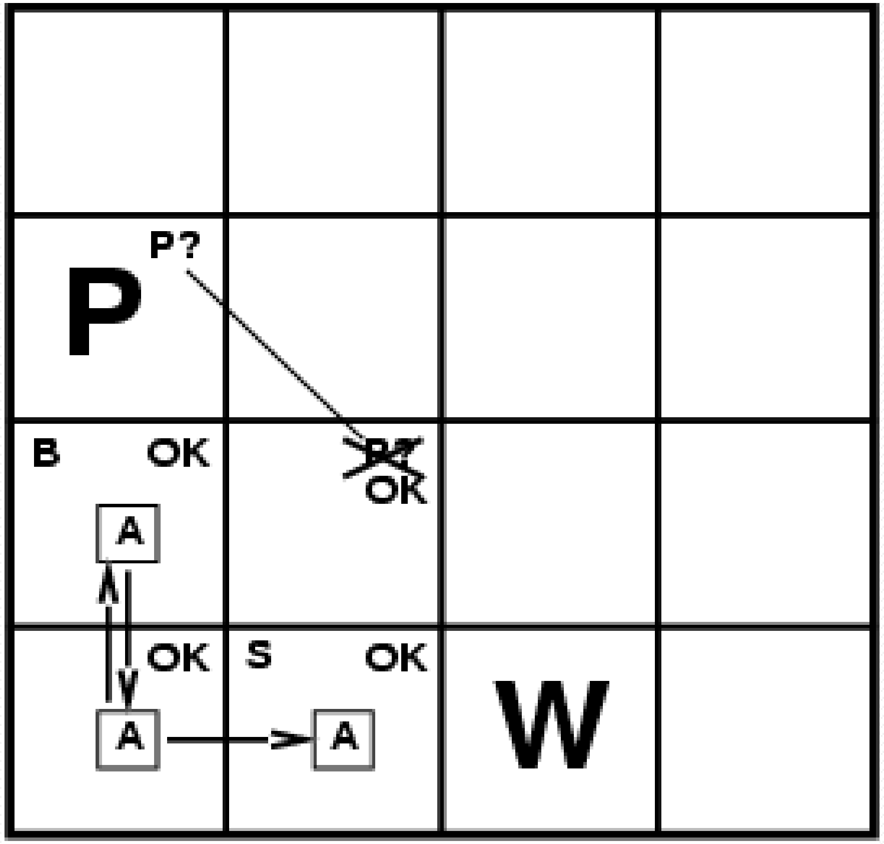
دنیای ومپوز

A = عامل
B = نسیم
G = درخشش طلا
OK = مربع امن
P = گودال
S = بوی بد
V = ملاقات شده
W =ومپوز



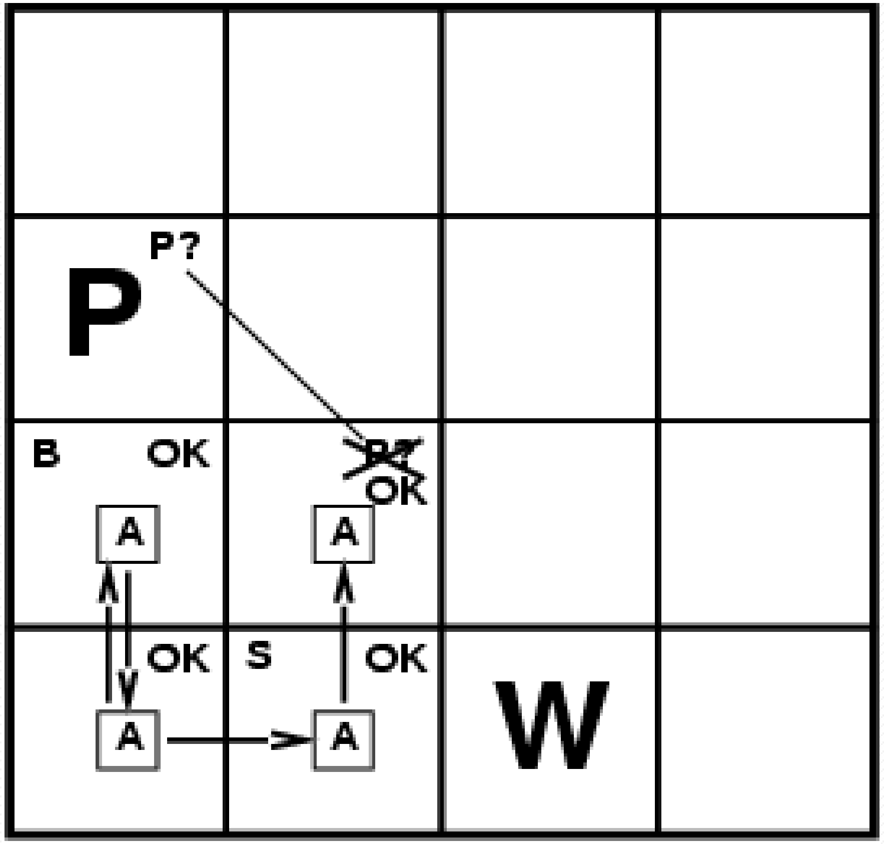
دنیای ومپوز

A = عامل
B = نسیم
G = درخشش طلا
OK = مربع امن
P = گودال
S = بوی بد
V = ملاقات شده
W =ومپوز



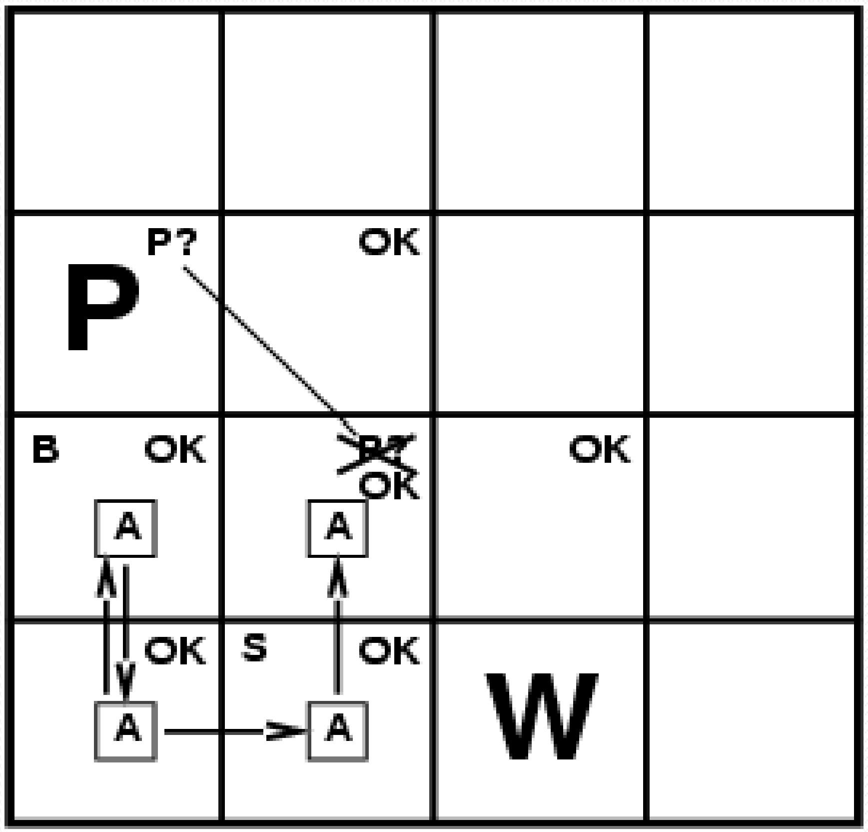
دنیای ومپوز

A = عامل
B = نسیم
G = درخشش طلا
OK = مربع امن
P = گودال
S = بوی بد
V = ملاقات شده
W =ومپوز



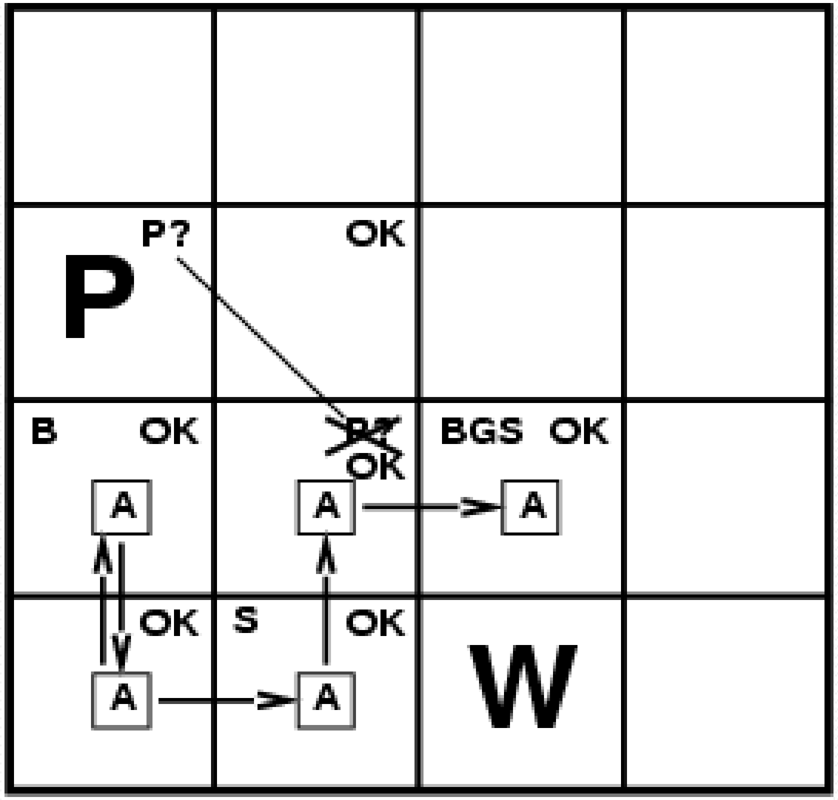
دنیای ومپوز

A = عامل
B = نسیم
G = درخشش طلا
OK = مربع امن
P = گودال
S = بوی بد
V = ملاقات شده
W =ومپوز



دنیای ومپوز

A = عامل
B = نسیم
G = درخشش طلا
OK = مربع امن
P = گودال
S = بوی بد
V = ملاقات شده
W =ومپوز



منطق

منطق یک سیستم رسمی برای تعریف حالتهاست که دارای :

- **نحو زبان(جمله بندی)**: چه ترکیبی برای یک جمله درست است.(خوش شکل)

مثلا x+y=4 خوش شکل و x2y+= بد شکل است

- **معانی زبان**: باید معنی جملات را توضیح دهد.
  - در تعریف دقیق تر منطق، معانی زبان، درستی هر جمله را در نسبت به هر دنیای ممکن تعریف میکند.

x+y=4

این معنی صحت جمله را در دنیایی که x=1 و y=1 است رد می کند.

- در منطق های استاندارد هر جمله در هر **دنیای ممکن** یا درست است یا نادرست (boolean)(یعنی بینابین وجود ندارد).
- برای دقت بیشتر کلمه "مدل" را به جای "دنیای ممکن"به کار می بریم.وقتی می گوییم m مدلی از a است یعنی جمله a در مدل m درست است.یا جمله a باعث ایجاد دنیای m می شود

مثال : منطق

مثال، در منطق زبان ریاضیات

- $X+2 >= y$ یک جمله خوش شکل است اما $x2+y$ چنین نیست.
- $X+2 >= y$ در دنیایی درست است که: $x=7$ و $y=1$
- $X+2 >= y$ در دنیایی نادرست است که: $x=0$ و $y=6$

ایجاب منطقی (استلزام) $\models$

- ایجاب منطقی بین جملات این است که جمله ای از جمله دیگر بطور منطقی استنباط می شود(ایجاد جمله جدید از جملات قبلی):

$\alpha \models \beta$

- جمله α ، جمله β را ایجاب میکند.
- اگر و فقط اگر، در هر مدلی که α درست است، β نیز درست باشد.
- اگر α درست باشد، آنگاه β نیز درست است.
- درستی β مستلزم درستی α است.

- مثال: جمله $x+y=4$ جمله $4=x+y$ را ایجاب می کند.

مدل چيست؟

منطق دانان عموماً بر حسب مدل ها فكر مي كنند، كه بطور رسمي دنياهاي ساخت يافته اي مي باشند كه درستي را مي توان نسبت به آنها ارزيابي كرد.

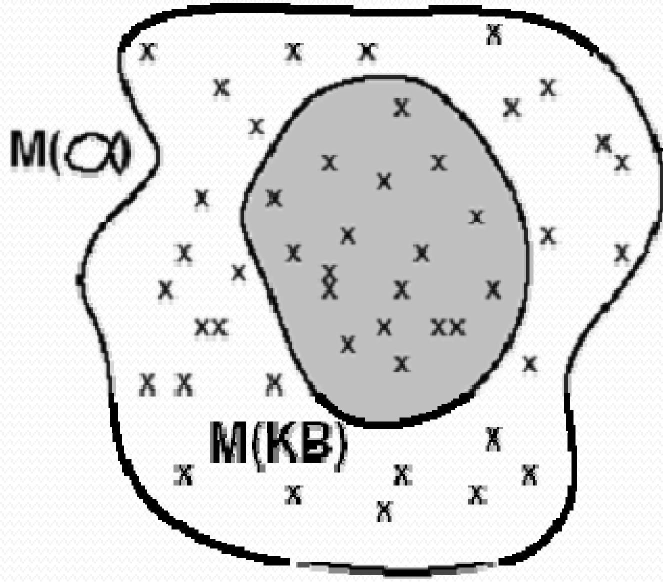
مي گوييم m مدلي از جمله α مي باشد اگر α در m درست باشد

$M(\alpha)$ مجموعه تمام مدل هاي α مي باشد

بنابراين $KB \models \alpha$ اگر وفقط اگر $M(KB) \subseteq M(\alpha)$

قد علی ۲ متر است

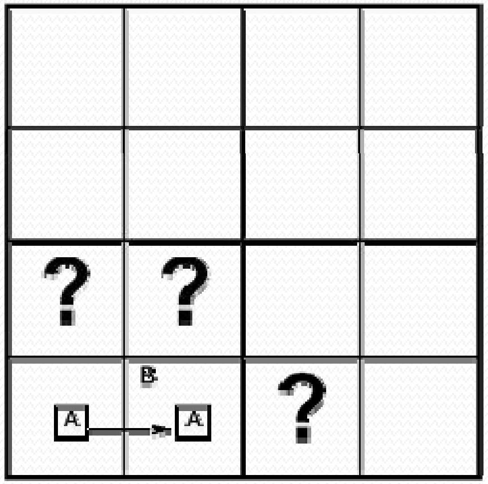
قد علی کمتر از ارتفاع زردکوه است



مدلهای ممکن برای ومپوز

موقعیت پس از [1, 1]، رفتن به راست،

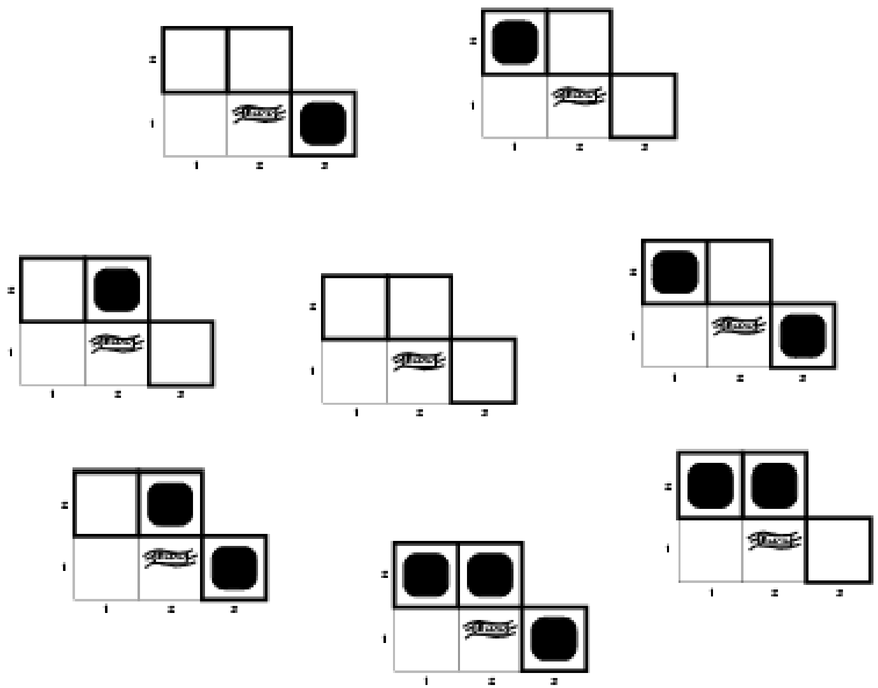
دریافت نسیم در [2, 1]



مدلهای ممکن برای ? ها را تنها با فرض چاله ها در نظر بگیرید

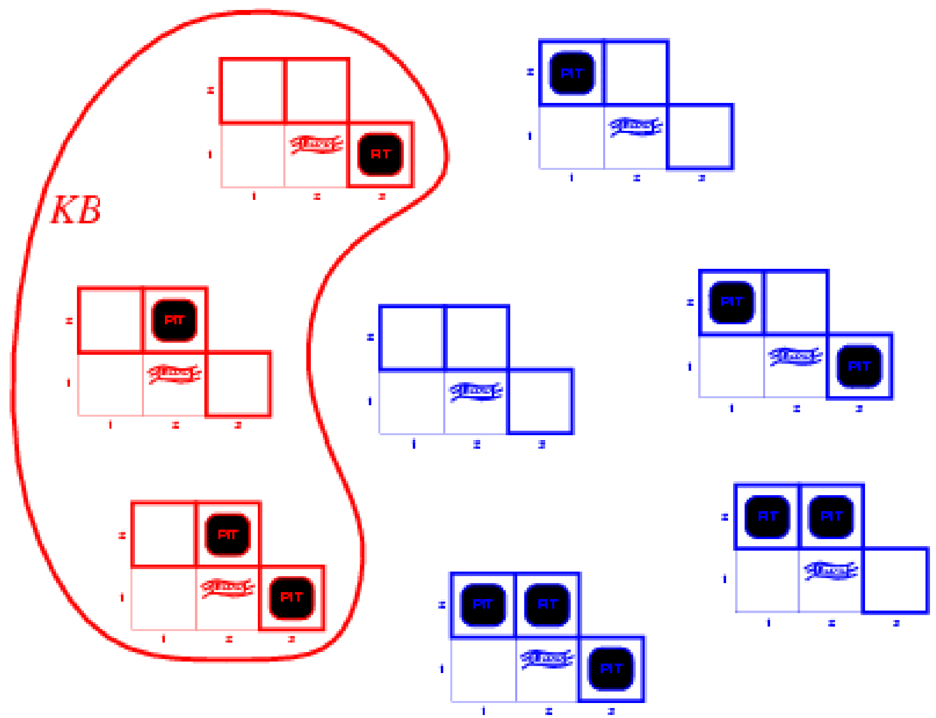
سه انتخاب بولین ← هشت مدل مختلف

مدلهای ومپوز



۲*۲*۲=۸ مدل ممکن

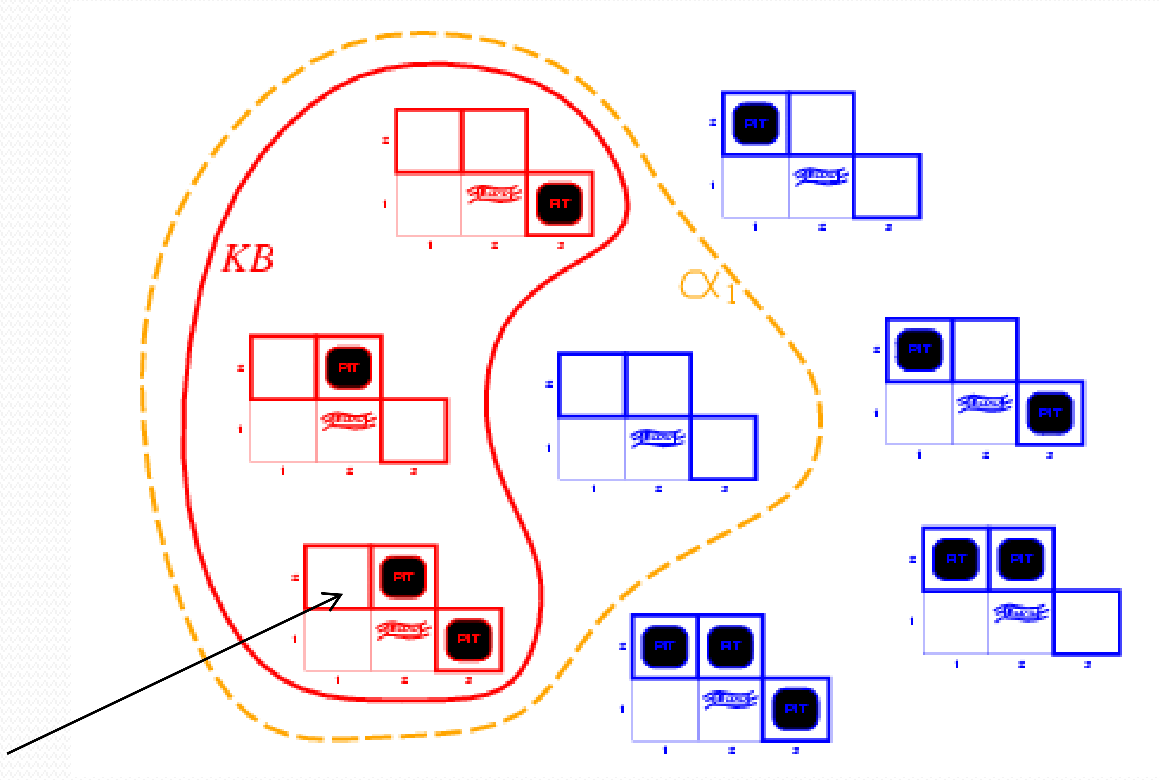
مدلهای ومپوز



KB شامل قواعد دنیای ومپوز و حقایق موجود در این دنیاست

$KB =$ حقایق+ قواعد

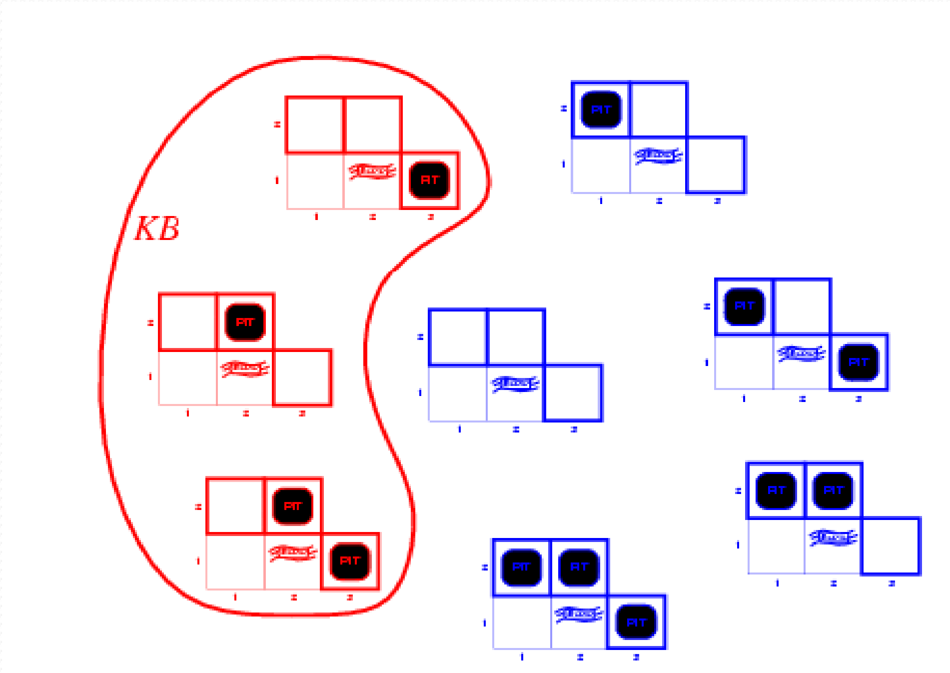
مدلهای ومپوز



هیچ گودالی در (۲و۱)وجود ندارد = α1

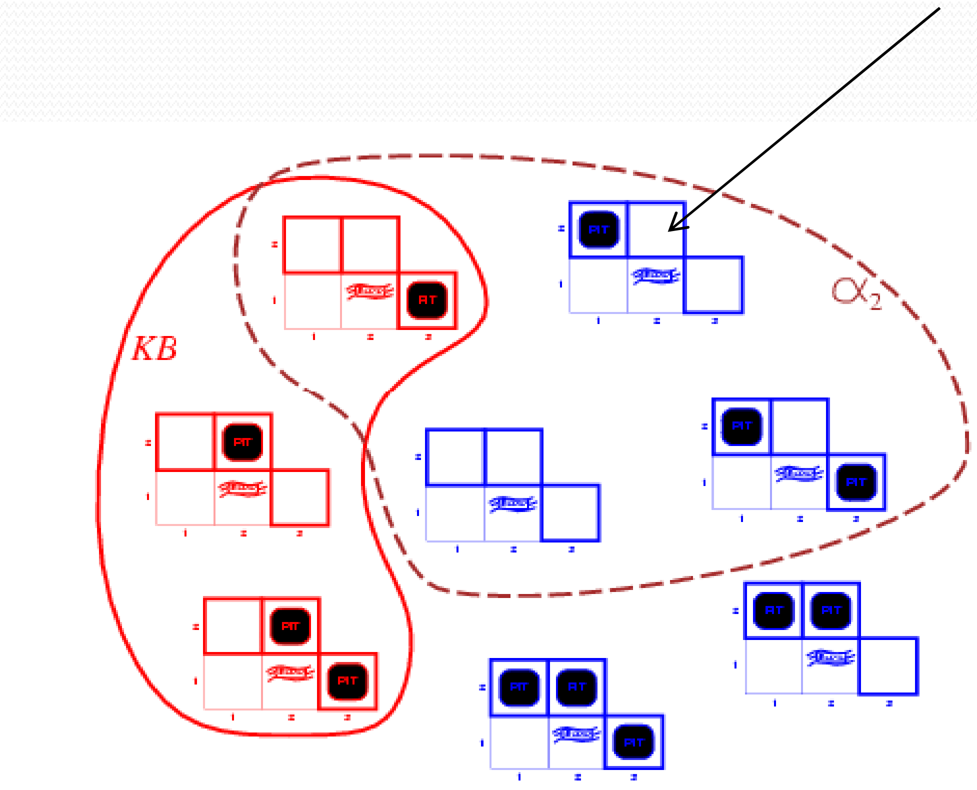
☑ $KB \models \alpha 1$,"چراکه KB زیر مجموعه(یا مساوی)α است

مدلهای ومپوز



واقعیت ها +قواعد = KB

مدلهای ومپوز



$KB =$ حقایق +قواعد

$\alpha 2 =$ " هیچ گودالی در (۲و۲)وجود ندارد ", $KB \not\models \alpha 2$

استنتاج $\vdash_i$

- استنتاج منطقی ، پردازشی است که ارتباط استلزامی(ایجاب) بین جملات را پیدا می کند
- مثال سوزن و کاه:
  - نتایج KB را به صورت یا مجموعه کاه در نظر بگیرید
  - ایجاب دانستن وجود سوزنی مثل α در کاه است
  - استنتاج به منزله ی یافتن سوزن است

$KB \vdash_i \alpha$ = جمله α بوسیله رویه i از KB قابل اشتقاق می باشد.

استنتاج

- الگوریتم استنتاج مثال قبل را ***وارسی مدل*** می گویند، زیرا تمامی مدلهای ممکن را برشماری می کند تا وارسی کند که آیا α در تمام مدل های درست KB، درست است یا خیر.

۱. الگوریتم استنتاجی که فقط جملات ایجابی را به دست می آورد، ***صحیح یا حقیقت نگهدار*** نامیده می شود.

۲. یک الگوریتم استنتاج ***کامل*** است در صورتی که بتواند هر جمله ایجاب شدنی را به دست آورد.

صحت (soundness): رویه i صحیح است اگر

$$KB \vdash_i \alpha \Rightarrow KB \models \alpha$$

کامل بودن (completeness): رویه استنتاج i کامل است اگر

$$KB \models \alpha \Rightarrow KB \vdash_i \alpha$$

بررسی اولین و ساده ترین منطق :منطق گزاره ای

- گزاره ها (حقایق) توسط سمبل یا نمادها نمایش داده می شوند که هر نماد می تواند نماینده یک حقیقت باشد مثلا "الان روز است" با نماد A نمایش داده می شودکه ممکن است درست باشد یا نادرست

می دانیم در هر منطقی دو جزء وجود دارد :
- نحو
- معانی

- نحو منطق گزاره ای ، جملات مجاز را تعیین میکند.
- جملات بسیط یا اتمیک (عناصر نحوی غیر قابل تجزیه) از یک نماد یا سمبل گزاره ای تشکیل می شود.
- مدل نسبت دادن True و False به اجزای اتمیک است با سه نماد می توان ۸ مدل ایجاد کرد
- هر یک از این نمادها نمایانگر یک گزاره است که می تواند درست یا نادرست باشد.
  - نمادها از حروف بزرگ مثل R,Q,P استفاده میکنند.

- دونماد گزاره ای با معنی ثابت:
  - True: گزاره همواره درست
  - False:گزاره همواره نادرست

نحو منطق گزاره ها

- جملات مرکب به کمک رابطهای منطقی، از جملات ساده تر تشکیل می شوند.
- انواع رابطها:
- $\neg$ (not) جمله ای نظیر $W_{1,3}$ $\neg$ نقیض $W_{1,3}$ است.
- یک لیترال، یا یک جمله بسیط(لیترال مثبت)است ، یا نقیض جمله بسیط (لیترال منفی).
- $\wedge$ (and) مثل $W_{1,3} \wedge P_{1,3}$ ترکیب عطفی نام دارد. اجزای عطفها هستند.
- $\vee$ (or) مثل $(W_{1,3} \wedge P_{3,1}) \vee W_{2,2}$ ترکیب فصلی مربوط به فاصلهای $W_{2,2}$ و $P_{3,1} \wedge W_{1,3}$
- $=>$ (استلزام): $(W_{1,3} \wedge P_{3,1}) \vee \neg W_{2,2}$ استلزام یا شرطی نامیده میشود. فرض آن یا مقدم آن $W_{1,3} \wedge P_{3,1}$ و نتیجه یا تالی آن $\neg W_{2,2}$ است.
- $\Leftrightarrow$ (اگر و فقط اگر)جمله $W_{2,2} \Leftrightarrow W_{1,3}$ دو شرطی نام دارد.

معانی منطق گزاره ها

- معانی، قواعد تعیین درستی یک جمله نسبت به یک مدل خاص را تعریف می کند.
  - مثال مدل $x+y=4$ و بحث دنیایی که $x=1$ و $y=3$
- در منطق گزاره ای، یک مدل، مقدار درستی را برای هر نماد گزاره ای تعیین می کند.
- معانی منطق گزاره ای، باید چگونگی محاسبه مقدار درستی هر جمله با توجه به یک مدل را مشخص کند. این کار به صورت بازگشتی انجام می شود
- تمام جملات از جملات بسیط و پنج رابط ساخته می شوند.

- هر مدل درست بودن/غلط بودن سیمبول های گزاره ای را مشخص می کند
- مثلا $P_{1,2}$ (درست)، $P_{2,2}$ (درست) ، $P_{3,1}$ (نادرست)
- قوانین ارزیابی درستی نسبت به یک مدل m

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
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*

جداول درستی برای پنج رابط منطقی

P	Q	┐P	P ^ Q	P v Q	P=>Q	P⇔Q
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

منطق گزاره ای برای دنیای ومپوز

اجازه دهید $P_{i,j}$ درست باشد، اگر و فقط اگر در خانه $[i, j]$ چاله باشد.

اجازه دهید $B_{i,j}$ درست باشد، اگر و فقط اگر در خانه $[i, j]$ نسیم باشد.

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

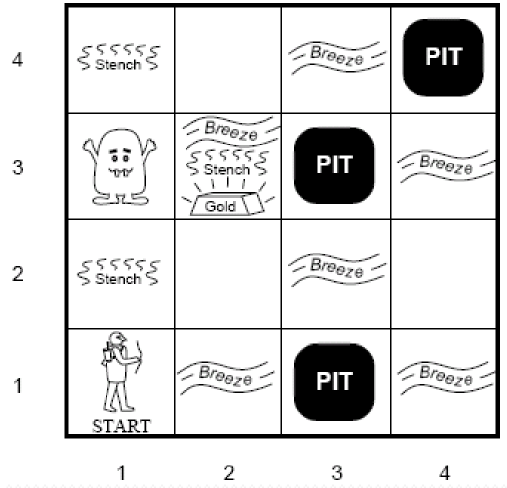
$$\neg B_{1,1}$$

$$B_{2,1}$$

" چاله ها باعث وزش نسیم در خانه های مجاور مي شوند".

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

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



" در يك خانه نسيم مي وزد اگر و فقط اگر چاله اي مجاور آن باشد"

استنتاج در منطق گزاره ای

- می دانیم هدف استنتاج منطقی ، این است که ثابت کند برای جمله ای مثل α مي توان گفت $KB \models \alpha$ ؟

۱. **روش اول** : بررسی شمارشی و خطی این موضوع که α در هر مدلي که KB در آن درست است ، درست باشد یعنی بررسی برای اینکه آیا در همه حالتهای سمبلها معتبر است یا خیر؟
   - شمارش تمام مدلها به روش اول عمق صورت مي گيرد
   - این الگوریتم صحیح(حقیقت نگهدار) است چون مستقیما ایجاب را پیاده سازي مي کند
   - این الگوریتم کامل است زیرا براي هر KB و α کار مي کند و همیشه خاتمه مي یابد
   - اگر n نماد موجود باشد 2^n مدل به وجود مي آید(n سمبل و مقداردهیT-F)
   - پیچیدگي زماني $o(2^n)$ و پیچیدگي فضايي $o(n)$

۲. **روش دوم** :استفاده از الگوریتمهای استنتاجی

مفاهیم زیربنایی الگوریتم های استنتاجی: هم ارزی، اعتبار و ارضا پذیری

- ***هم ارزی منطقی*** : دو جمله a , b ***هم ارز منطقی*** هستند اگر در مجموعه مدل های یکسان، درست باشند.این مفهوم را به شکل $b \Leftrightarrow a$ مینویسیم.

$\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

- ***اعتبار*** : یک جمله ***معتبر*** است در صورتی که در تمامی مدلها، درست باشد.جملات معتبر به عنوان ***بدیهیات*** نیز شناخته می شوند.

True، $A \vee \sim A$

- ***ارضا پذیری (صدق پذیر)***: یک جمله ***ارضا پذیر*** است اگر در بعضی مدلها صدق کند و درست باشد.مثلا جمله x=1 در صورتیکه x برابر ۱ باشد صدق پذیر است یا... $A \vee B$

برهان خلف: اضافه کردن نقیض چیزی که به دنبالش هستیم و اثبات ناسازگاری وضعیت پیش آمده. اثبات با تناقض(اثبات با تکذیب)

هم ارزی های منطقی استاندارد

$\alpha \equiv \beta$ اگر و فقط اگر $\beta \models \alpha$ و $\alpha \models \beta$ باشد .

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$: جابه جایی پذیری $\wedge$

$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$: جا به جایی پذیری $\vee$

$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$: شرکت پذیری $\wedge$

$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$: شرکت پذیری $\vee$

$\neg(\neg\alpha) \equiv \alpha$: حذف نقیض دوگانه

$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$: مفهوم مخالف (عکس نقیض)

$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$: حذف استنتاج

$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$: حذف شرط دو طرفه

$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$: قانون دمورگان

$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$: قانون دمورگان

$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$: توزیع پذیری $\wedge$ روی $\vee$

$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$: توزیع پذیری $\vee$ روی $\wedge$

الگوهای استدلال در منطق گزاره ای

قواعد استنتاج: الگوهای استنتاج استانداردی که می توانند برای حصول زنجیره ای از نتایج که به هدف مطلوب منجر می شود به کار برده شوند.

۱. قیاس استثنایی(Modus Ponens): با استفاده از ترکیب عطفی، میتوان هر عطف را استنتاج کرد(یعنی هر وقت جمله ای به شکل $\alpha \Rightarrow \beta$ و α داده شود، جمله β را میتوان استنتاج کرد.)

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

$\alpha \Rightarrow \beta$: True

α : True

β:True

۲. حذف عطف (And Elimination): هر عطف را می توان از ترکیب عطفی استنتاج کرد(با این شرط که نتیجه ترکیب شرطی True است)

$$\frac{\alpha \wedge \beta}{\alpha} \quad \text{یا} \quad \frac{\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \ldots \alpha_n}{\alpha_i}$$

Or β

استفاده از هم ارزی های منطقی برای استنتاجهای جدید

$$\frac{(\alpha => \beta) \wedge (\beta => \alpha)}{\alpha <=> \beta} \qquad \frac{\alpha <=> \beta}{(\alpha => \beta) \wedge (\beta => \alpha)}$$

استفاده از دنباله ای از کاربردهای قواعد استنتاج ، اثبات نامیده میشود

خاصیت یکنواختی

مجموعه جملات ایجاب شده فقط زمانی می توانند افزایش یابند که اطلاعاتی به پایگاه دانش افزوده شود.

برای جملات a و b داریم:

$$KB \models \alpha \Rightarrow KB \wedge \beta \models \alpha$$

استفاده قوانین استنتاج به منظور یافتن نتیجه از یک پایگاه دانش، به طور صریح مبتنی بر خواص عمومی منطقهای است که یکنوایی (monotonicity) نامیده می‌شود.

می‌توانیم خواص یکنوایی منطق را به طور زیر شرح دهیم:

$$if\ KB_1 \models_{\alpha}\ then\ (KB_1 \cup KB_2) \models \alpha$$

قاعده استنتاج تحلیل(مهم)

- تحلیل واحد، یک عبارت(ترکیب فصلی(OR) لیترالها) و یک لیترال را گرفته، عبارت دیگری تولید می کند.یعنی از صورت مخرج را استنتاج می کند و از دو عبارت فصلی ظاهر شده در صورت کسر یک عبارت فصلی(ترکیبی)برای مخرج ایجاد می شود، مواردی که در دو عبارت به یک لیترال به صورت نقیض اشاره می کنند در مخرج حذف می شوند

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

m و l_i هر دو به یک لیترال به صورت برعکس اشاره می کنند

- قاعده تحلیل واحد می تواند به قاعده تحلیل کامل تعمیم داده شود:

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

m_j و l_i هر دو به یک لیترال به صورت برعکس اشاره می کنند

نرمال عطفی(مهم)

- شکل نرمال عطفی$_{AND}$(CNF):مزیت شکل نرمال عطفی آن است که برخلاف قاعده تحلیل که فقط برای ترکیبهای فصلی لیترالها به کار برده می شد.جمله ای که به صورت ترکیب عطفی ، ترکیبهای فصلی لیترالها بیان شود شکل نرمال عطفی یا CNF گفته می شود

- هر عبارت موجود در جمله k-CNF دقیقا k لیترال وجود دارد

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

مراحل تبدیل به نرمال عطفی

۱. حذف ⇔ با جایگزینی

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

A⇔B : (A =>B) ^ (B=>A)

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

۲. حذف => با

A=>B : ┐A v B

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

۳. اعمال ┐ به لیترالها (داخل بردن)

┐(┐A)≡A

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

┐(A^B) ≡ (┐A v ┐B)

┐(AvB) ≡ (┐A ^ ┐B)

۴. استفاده از قانون توزیع پذیری v نسبت به ^ برای ترکیب عطفی فصلیها

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

دو راه برای اثبات استلزام

ساده کردن

- *Deduction Theorem*:
  $KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

برهان خلف

- Or,. . .
  $KB \models \alpha$ if and only if $(KB \wedge \neg\alpha)$ is unsatisfiable
  - *reductio ad absurdum*

(تابع PL-Resolution)

(برهان خلف)

function PL-RESOLUTION(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  clauses ← the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
  new ← { }
  loop do
    for each C_i, C_j **in** *clauses* **do**
      resolvents ← PL-RESOLVE(C_i, C_j)
      if *resolvents* contains the empty clause **then return** *true*
      new ← *new* ∪ *resolvents*
    if *new* ⊆ *clauses* **then return** *false*
    clauses ← *clauses* ∪ *new*

مثال الگوریتم تحلیل برای منطق گزاره ای (تابع PL-Resolution)

مجموعه تمامی بندهای ممکن که با حل دو ورودی بدست می آید را بر می گرداند

یک الگوریتم تحلیل:(از روش برهان خلف(قراردادن نقیض و ثابت کردن ناسازگاری))

مثال مهم : آیا از عبارت زیر می توان D→C را استنتاج کرد

۱

A → (B→C)
┐ D ∨ A
B

۲

A → (B→C)
┐ D ∨ A
B
┐ (D→C) افزودن نقیض نتیجه

۳ حذف شرطی

┐ A ∨ (B→C)≡ ┐ A ∨ (┐ B∨C)
┐ D ∨ A
B
┐ (┐ D ∨ C)

۴ اعمال ┐

┐ A ∨ (┐ B∨C)
┐ D ∨ A
B
D ∨ ┐ C

۵ نوشتن ∨ در خطوط جداگانه

┐ A ∨ (┐ B∨C)
┐ D ∨ A
B
D
┐ C

۶ حذف دو به دو متضادها

┐ A ∨ ┐ B ∨ C
┐ D ∨ A
B
D
┐ C

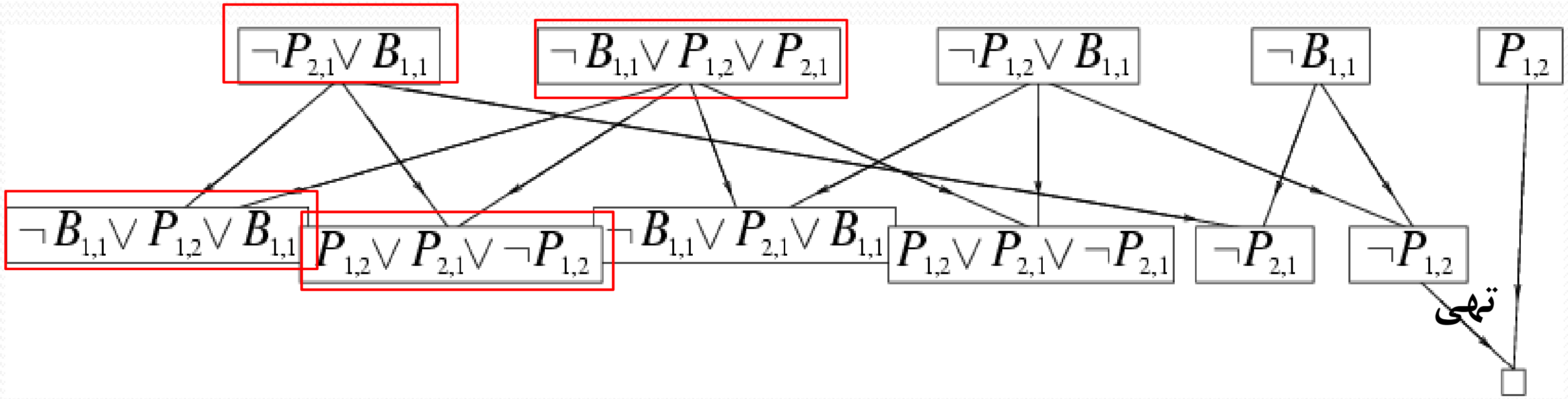
با رسیدن به تهی استلزام ثابت می شود

مثال تابع PL-Resolution روی مساله دنیای ومپوزها

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

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. Move $\neg$ inwards using de Morgan's rules and double-negation:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$
4. Apply distributive law ($\wedge$ over $\vee$) and flatten:
   $\mathbf{(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})}$

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



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

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

کامل بودن تحلیل

- RC(S) یا بستار تحلیل :

عبارت است از تمام بندهای به دست آمده به واسطه اعمال مکرر قواعد تحلیل به بند موجود در S یا مشتقات آنها

- قضیه تحلیل زمینه یا کامل بودن تحلیل:

اگر مجموعه ای از بندها،ارضا پذیر باشند .آنگاه بستار تحلیل (مجموعه تمامی بندهای ممکن که با حل دو ورودی بدست می آید) آن بند ها ، بند تهی را در بر خواهد داشت

زنجیره ای پیش رو و پس رو

- بند های هورن: که در پایگاههای دانش واقعی ایجاد می شود ترکیب فصلی لیترالهایی است که فقط یکی از آنها مثبت است.

علت سخت گیری در مثبت بودن فقط یکی از لیترالها:

۱.هر بند **هورن** را میتوان به صورت یک استلزام نوشت که مقدمه آن ترکیب عطفی لیترالهای مثبت و تالی آن یک لیترال مثبت است

$(\neg L_{1,1} \vee \neg \text{Breeze} \vee B_{1,1})$ .... $(L_{1,1} \wedge \text{Breeze}) \Rightarrow B_{1,1}$

- این نوع بندهای هورن که فقط یک لیترال مثبت دارند، بندهای **معین** نامیده میشوند
- لیترال مثبت را **سر**و لیترالهای منفی را **بدنه بندها** گویند
- عبارت معینی که فاقد لیترالهای منفی باشد، گزاره ای بنام **واقعیت** نام دارد.
- **بند های معین** اساس **برنامه نویسی منطقی** را میسازد

۲.استنتاج با بندهای هورن، از طریق الگوریتم های **زنجیره ای پیش رو** و **زنجیره ای پس رو** انجام میگیرد

۳.زمان تصمیم گیری در مورد ایجاب در بندهای هورن می تواند بر حسب اندازه پایگاه دانش به صورت خطی باشد

زنجیره ای پیش رو و پس رو

$$P_1 \wedge P_2 \; ... \wedge P_n \; \Rightarrow Q$$

$$\neg P_1 \; \vee \; ... \; \vee \neg \; P_n$$

شكل نرمال Horn (HNF):

KB = تركيب عطفي عبارت هاي Horn

عبارت Horn =

- سيمبول گزاره اي
- (سيمبول گزاره اي) ⇐(تركيب عطفي سيمبول هاي گزاره اي)

$$P_1 \wedge P_2 \wedge \ldots \wedge P_n \Rightarrow Q$$

- مثال :

$$C \wedge (B \Rightarrow A) \wedge (C \wedge D \Rightarrow B)$$

قانون استنتاج Modes Ponens براي شكل Horn:

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n, \qquad \alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n \Rightarrow \beta}{\beta}$$

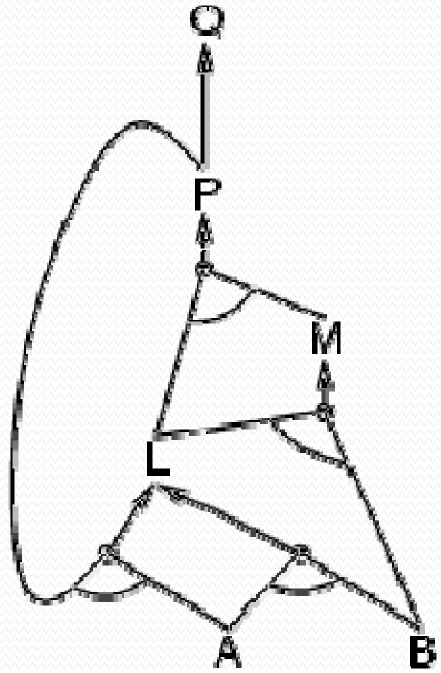
مي تواند در هر دو روش روبه جلو و روبه عقب بكار رود.

اين روشها بسيار طبيعي هستند و در زمان خطي (برحسب اندازه KB) اجرا مي شوند.

زنجیره ای پیش رو

ایده : هر قانونی که بخش شرایط آن در KB ارضاء شده را اعمال کن (fire) و نتیجه قانون را به KB اضافه کن، تا زمانیکه پاسخ پیدا شود و یا استنتاج دیگری ممکن نباشد.

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

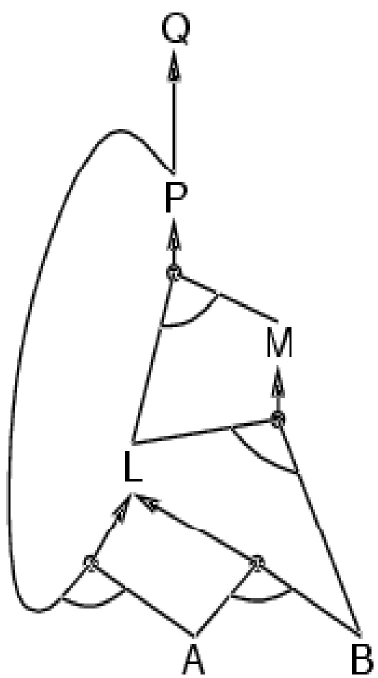


زنجیر پیش رو

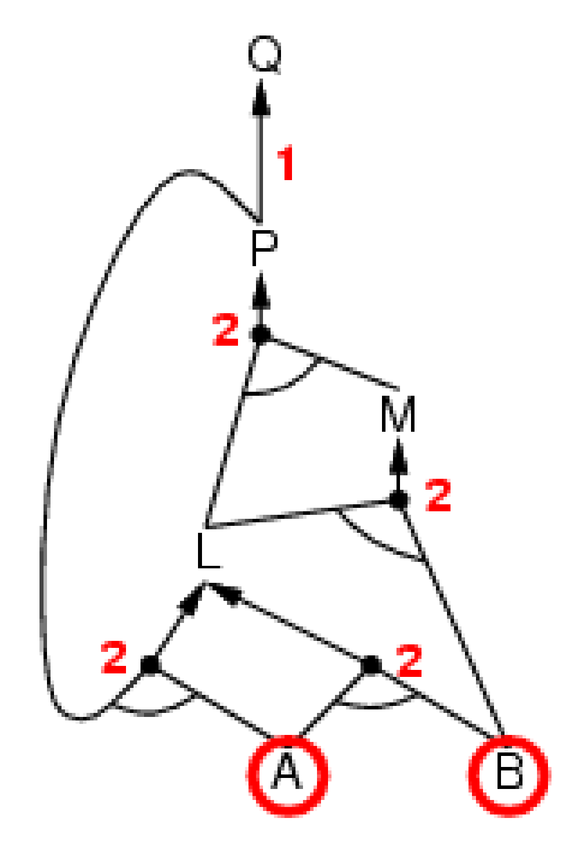
الگوریتم زنجیر پیش رو تعیین میکند آیا نماد گزاره ای q(تقاضا)، توسط پایگاه دانش بند های هورن ایجاب میشود یا خیر

از یک جمله ،جلو می رود و جملات دیگر را نتیجه می گیرد(استدلال داده گرا)

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

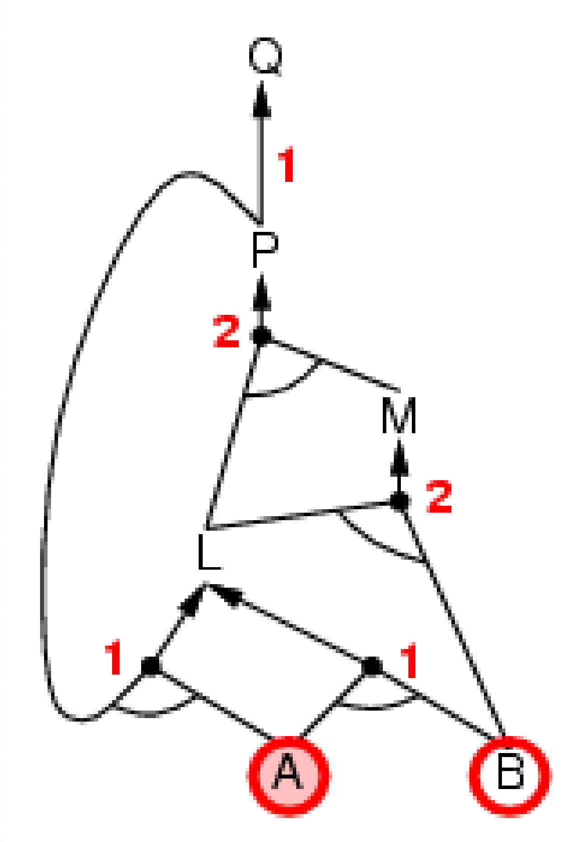


زنجیر پیش رو

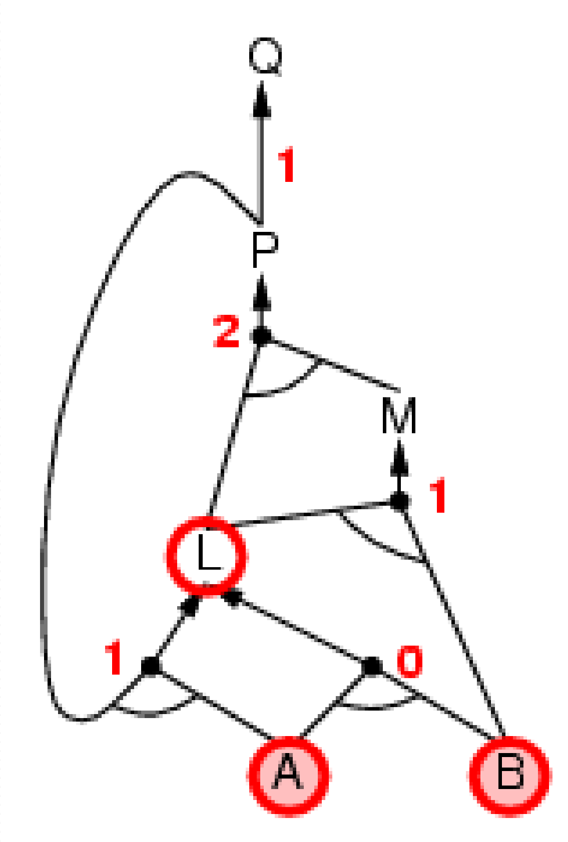


بارون بیاد
زمین خشک باشه

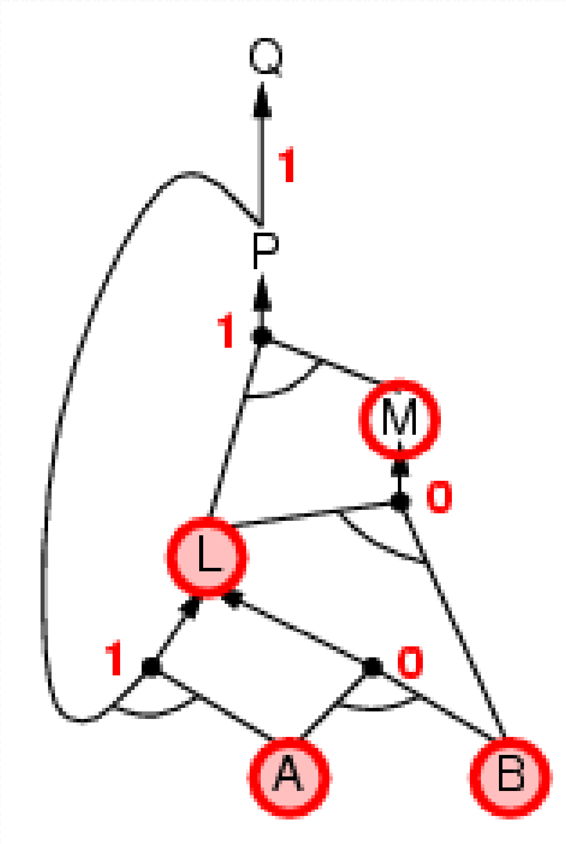
زنجیر پیش رو



زنجیر پیش رو

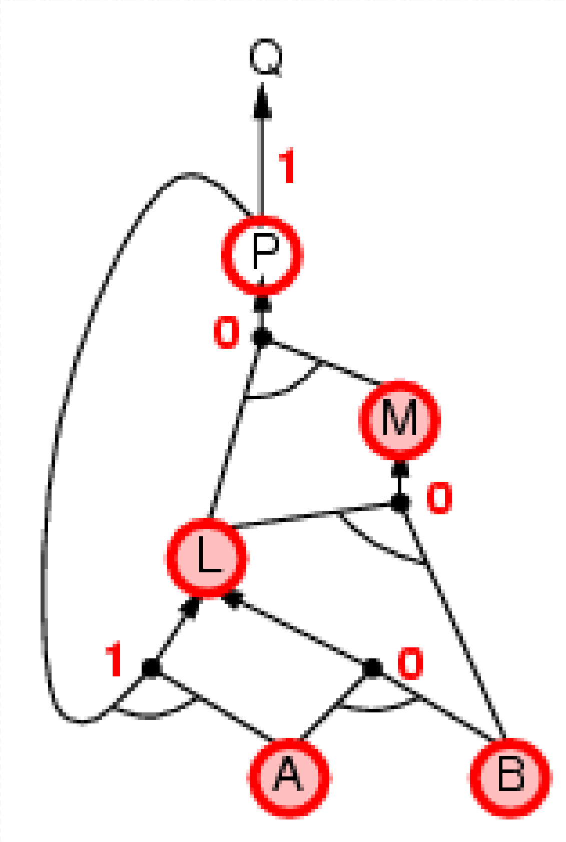


زنجیر پیش رو



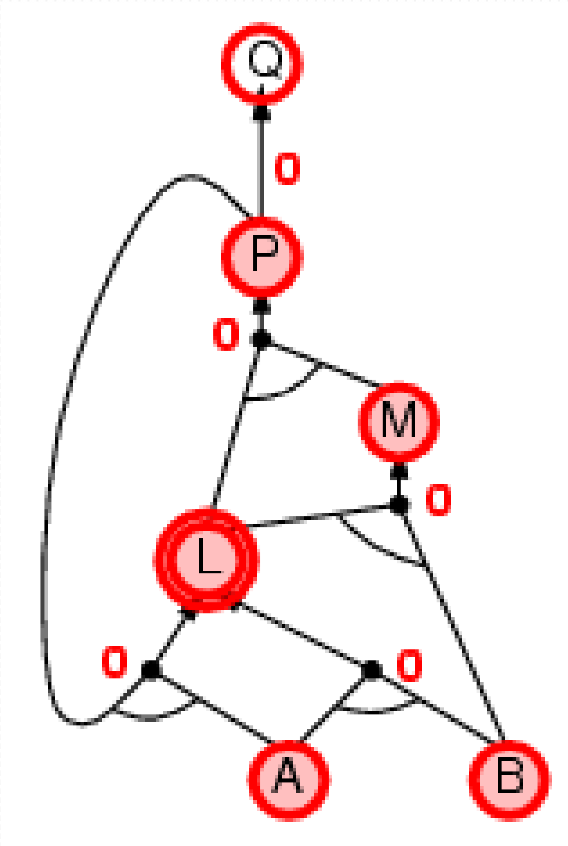
آب جاری میشه

زنجیر پیش رو



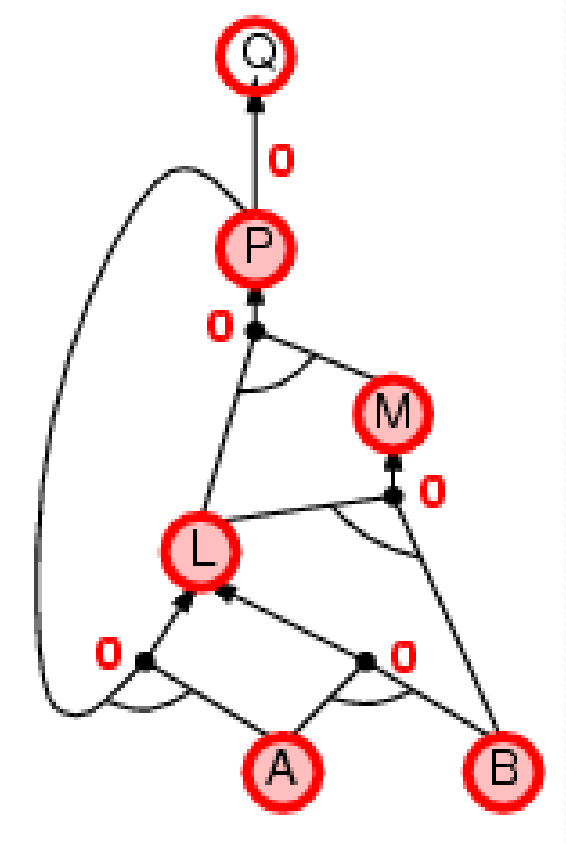
شهر پر آب میشه

زنجیر پیش رو



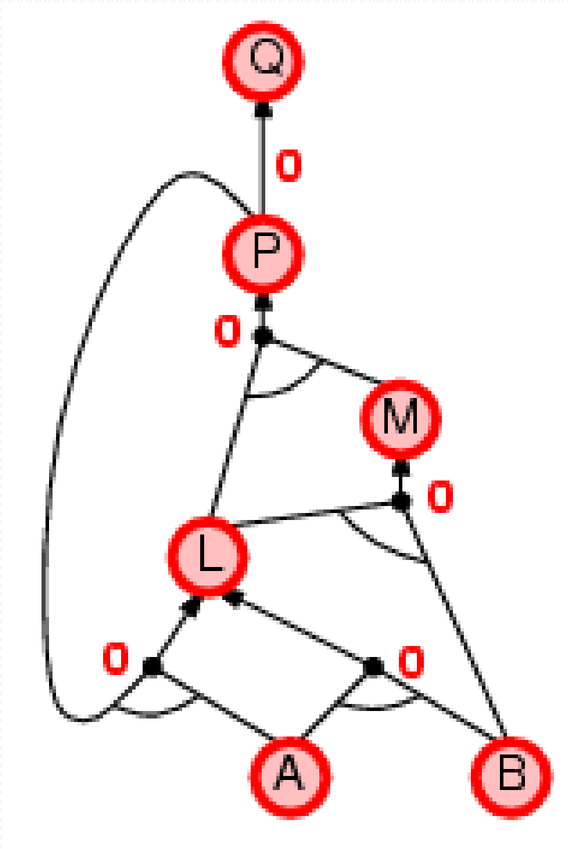
خونه ها ویران میشن

زنجیر پیش رو



شهر و آب میبره

زنجیر پیش رو



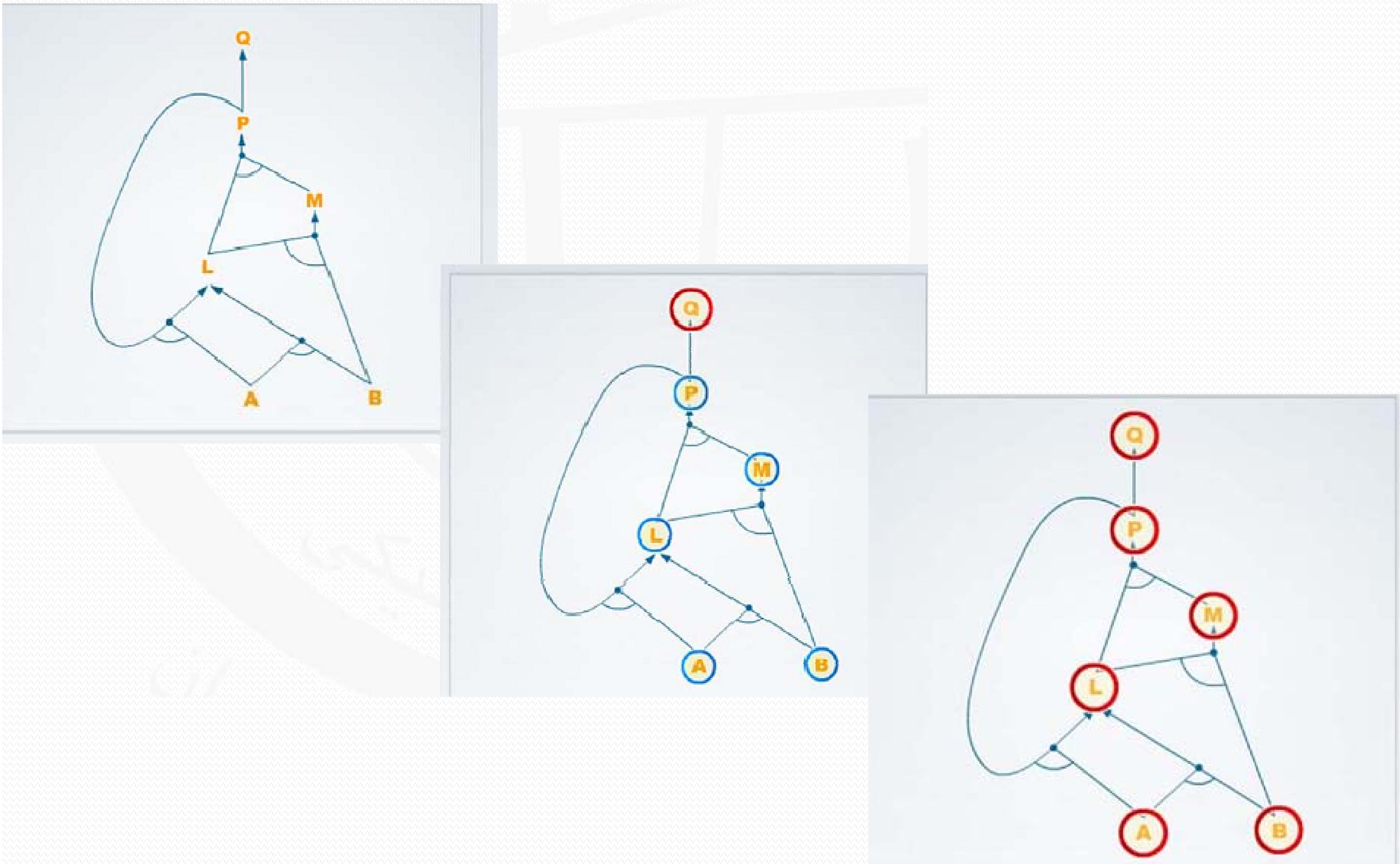
زنجیر پس رو

ایده‌ی اصلی: بررسی کنید برای اثبات q آیا قانونی وجود دارد که به نتیجه‌ی q برسد، اگر قانونی وجود داشت باید ببینید که پیش‌شرطهای آن چیست؟

نکته‌ی مهم این است که روند الگوریتم درLoop نیفتد.

عدم تکرار دوباره‌ی کارها باعث افزایش سرعت الگوریتم می‌شود.

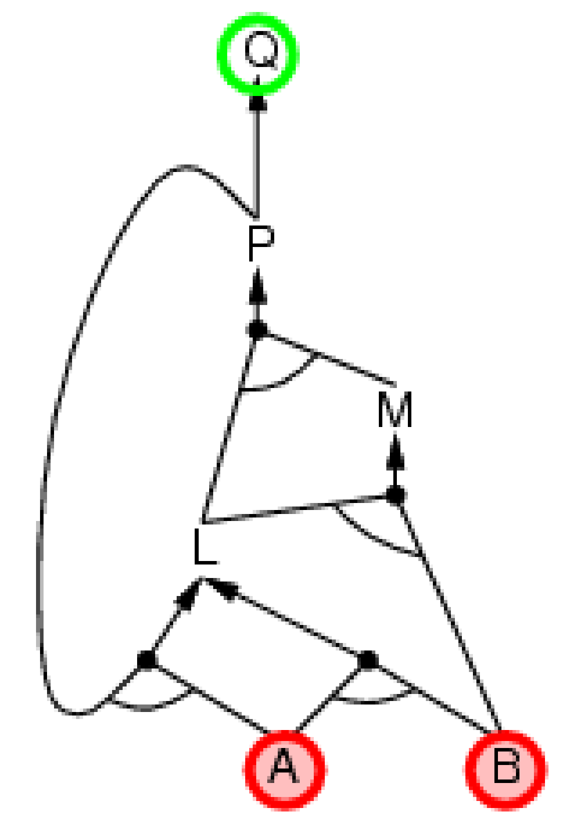
اثبات یک جمله از روی چندین جمله در پایگاه دانش


Q
P
M
L
A
B
Q
P
M
L
A
B
Q
P
M
L
A
B

تفاوت الگوریتم FC و BC

- الگوریتم Forward Chaining مبتنی بر داده‌هایی است که در اختیار داریم(Data Driven).
- الگوریتم Backward Chaining مبتنی بر هدف می‌باشد (Goal Driven).
  - در الگوریتم عقب‌رو شما قادر به مطرح کردن سوال خاص هستید.
  - این الگوریتم‌ها بر اساس کاربرد و دانشی که در اختیار دارید می‌توانند مفید و یا مضر باشند.
  - به عنوان مثال اگر سیستمی در اختیار داشته باشید که اطلاعات بسیار زیادی در آن وجود داشته باشد، بهتر است از الگوریتم Forward Chaining استفاده نکنید، زیرا ممکن است حجم زیادی از اطلاعات بی‌فایده تولید شود.
- الگوریتم‌های Resolution، Forward Chaining و Backward Chaining سه روش پایه‌ی استنتاج با استفاده از قوانین استنتاج هستند.
- در ادامه روش‌های دیگر استنتاج را مورد بررسی قرار خواهیم داد.

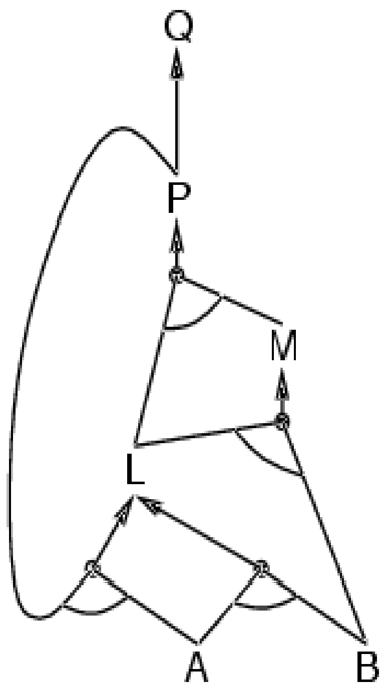
زنجیر پس رو



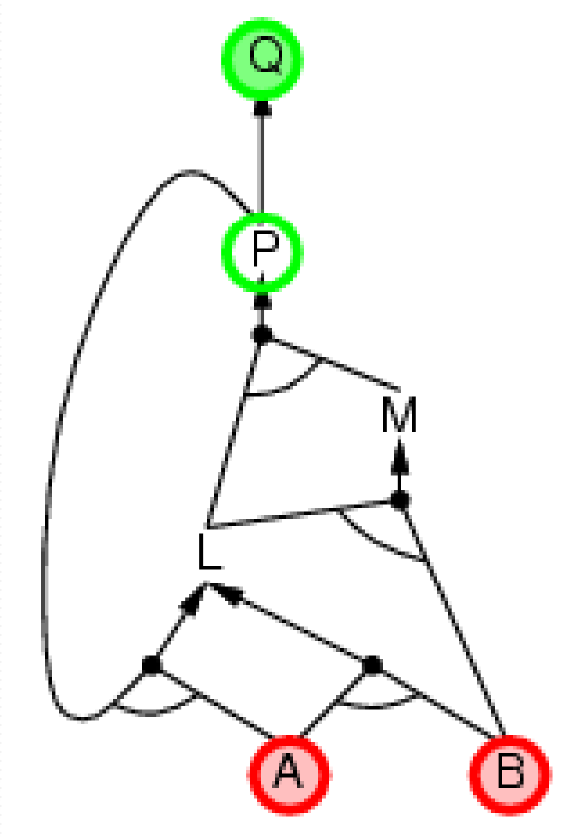
زنجیر پس رو

- تغییرات عمده: خاتمه زودهنگام، هیوریستیک نماد محض، هیوریستیک بند واحد

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

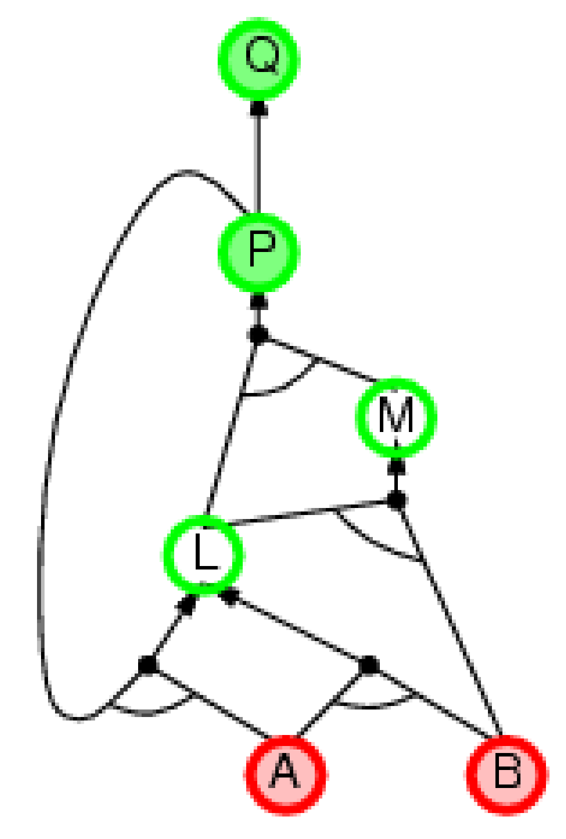


زنجیر پس رو



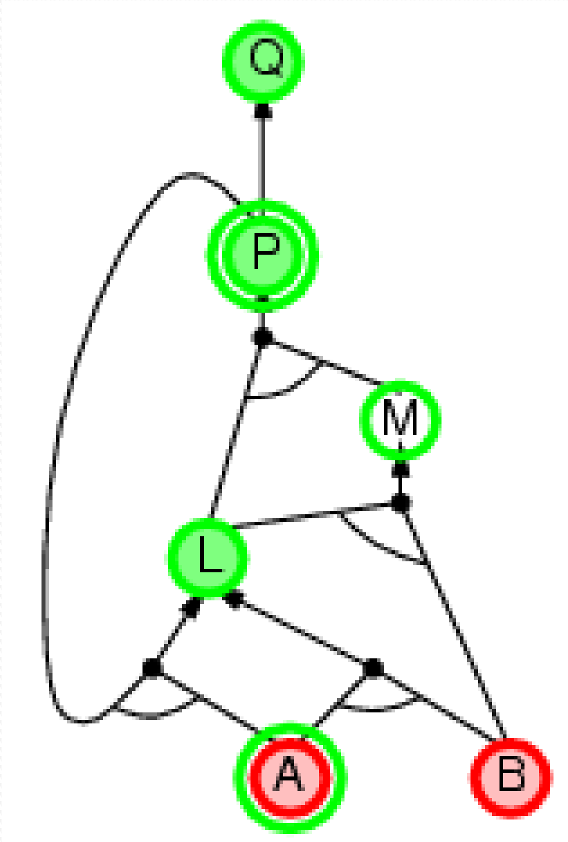
شهر و آب برده

زنجیر پس رو



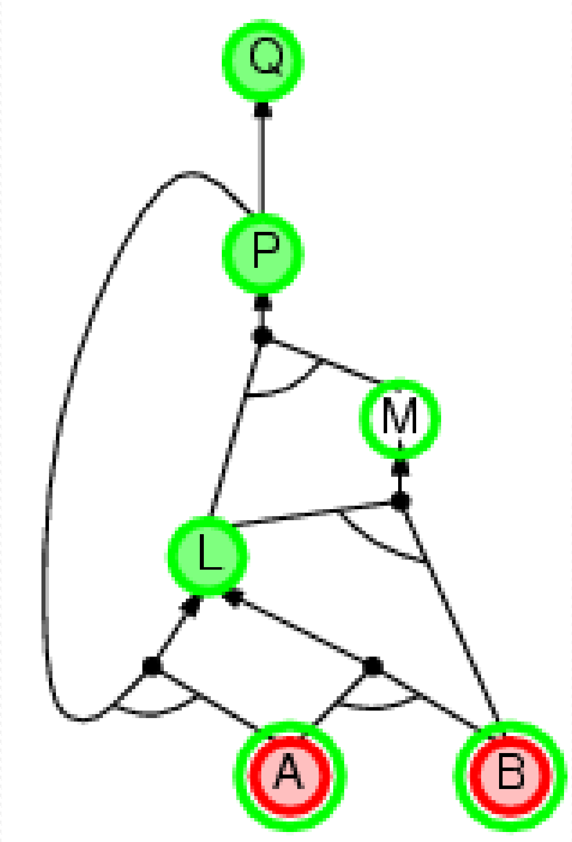
خونه ها ویران شدن

زنجیر پس رو



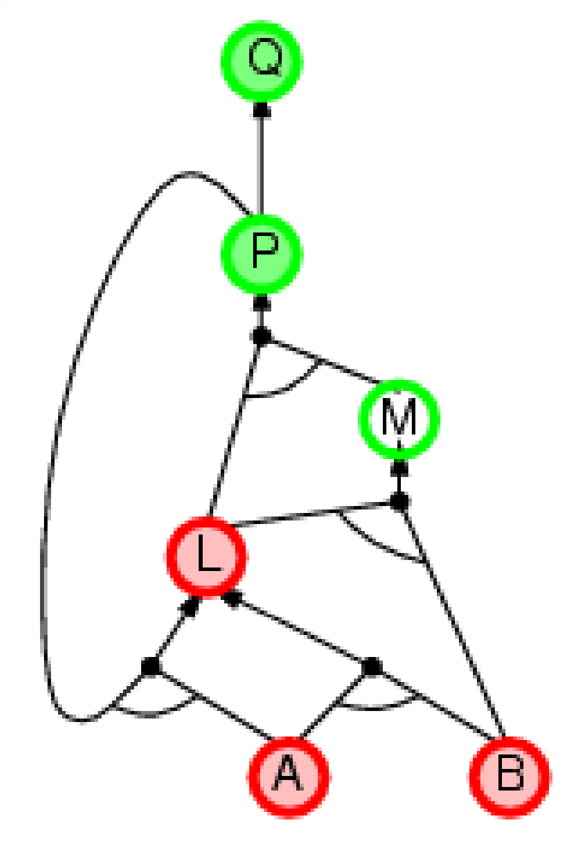
شهر پر از آبه

زنجیر پس رو



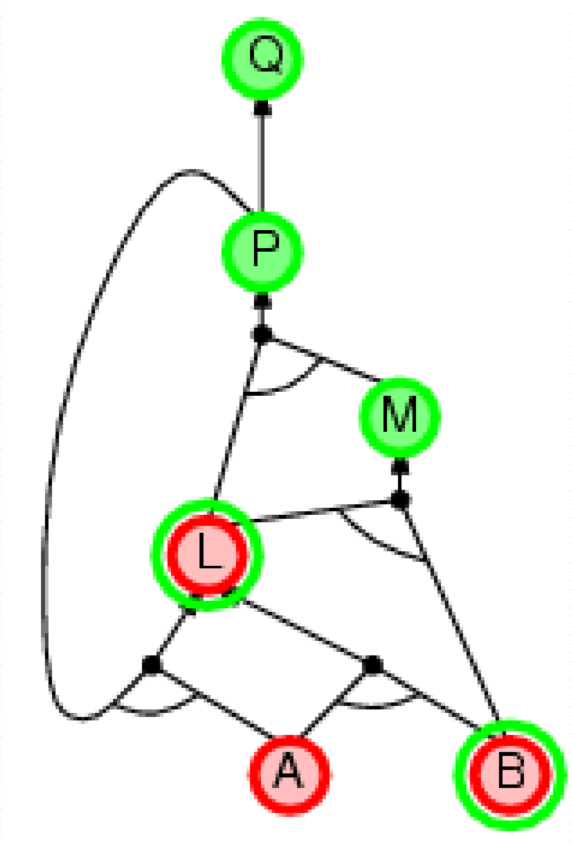
آب جاریه؟

زنجیر پس رو

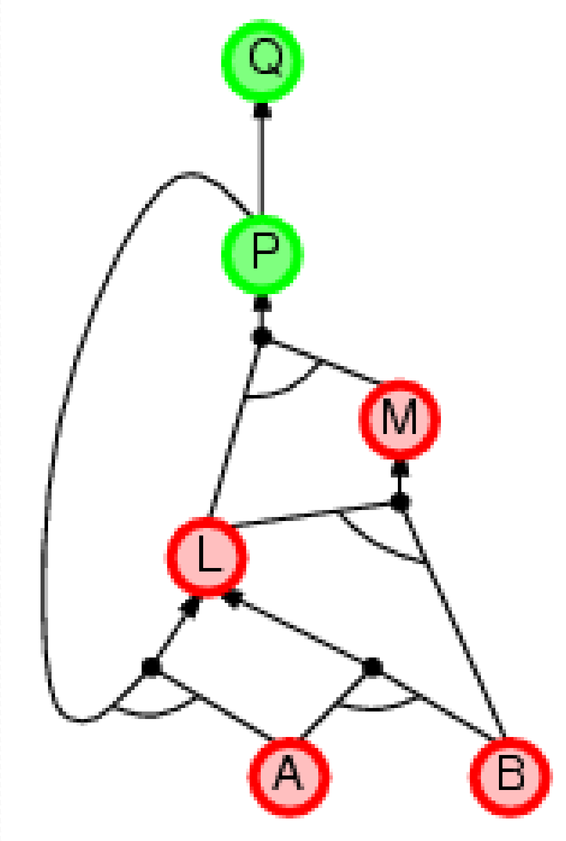


بارون اومده
زمین خشک بوده

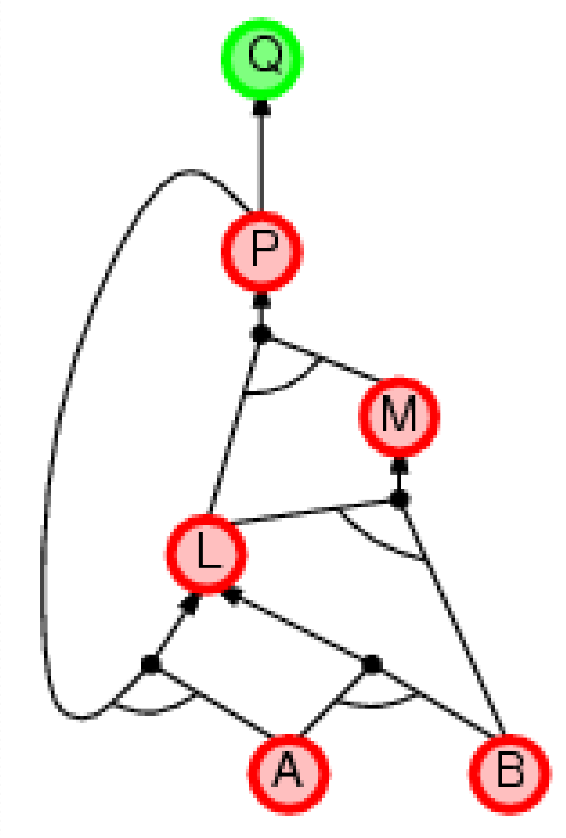
زنجیر پس رو



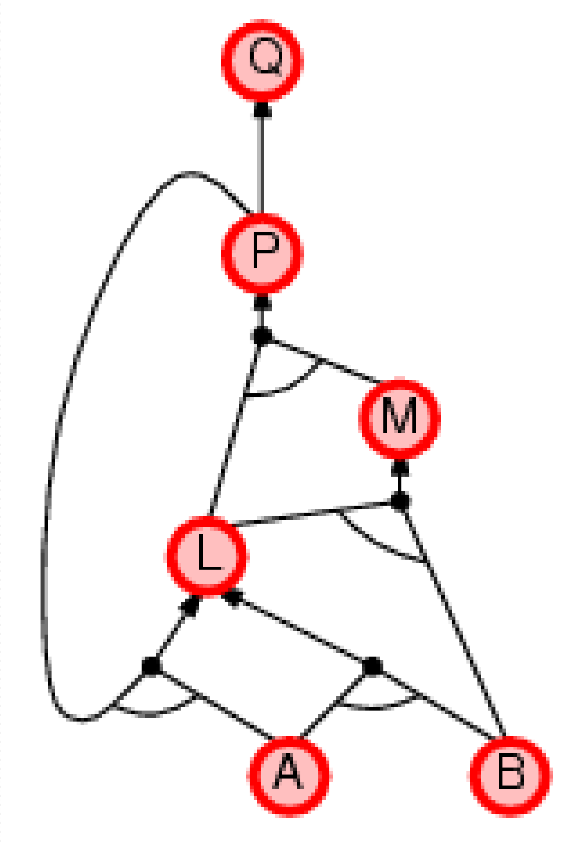
زنجیر پس رو



زنجیر پس رو



زنجیر پس رو



الگوریتم استنتاج کارا برای منطق گزاره ای

دو خانواده از الگوریتم های استنتاج کارآ برای منطق گزاره ای

الگوریتم های کامل جستجوی عقبگرد:

دیویس پاتنام
پس گرد کامل

– الگوریتم DPLL (Davis, Putnam, Logemann, Loveland)

الگوریتم ناکامل جستجوی محلی

– الگوریتم WalkSAT

الگوریتم پس گرد کامل

الگوریتم DPLL

تعيين كن كه آيا يك جمله ورودي در زبان منطق گزاره اي (در شكل نرمال CNF) صدق پذير است يا خير.

بهبودها نسبت به روش شمارش جدول درستي:

۱- خاتمه زود هنگام

- يك فراكرد (Clause) درست است اگر هر يك از ليترال ها درست باشد.
- يك جمله نادرست است اگر هر يك از فراكردهای آن نادرست باشد.

۲- هيوريستيك سيمبول محض

- سيمبول محض: سيمبولي كه در تمام فراكرد ها با يك علامت ظاهر شود.　　　A همیشه مثبت
- مثال: در سه فراكرد $(A \vee \neg B), (\neg B \vee \neg C), (C \vee A)$ سيمبولهای A و B سيمبول محض می باشند، اما C يك سيمبول محض نيست.
- ليترال يك سيمبول محض را درست تلقی كن.

۳- هيوريستيك فراكرد واحد　　(بند واحد)

- فراكرد واحد: تنها شامل يك ليترال مي باشد. يا فراكردی كه تمام ليترال های آن غير از يك ليترال، نادرست مي باشند.
- تنها ليترال موجود در يك فراكرد واحد بايد درست باشد.

(B v –C)
B=false　--- (false v –c) = (-c)

الگوریتم جستجوی محلی

مجموعه تناقضات در ارضای محدودیت

الگوریتم WalkSAT

- در اینجا قصد داریم الگوریتم Min Conflict را به WalkSAT تبدیل کنیم تا برای کار منطق سازگارتر باشد.
- ایده‌ی اصلی: ابتدا به تمام متغیرها مقدار اتفاقی true یا false می‌دهیم، سپس هربار بررسی می‌کنیم که کدام‌یک از متغیرها در صورت تغییر باعث کاهش تعداد ناسازگاری‌ها می‌شود.

تابع ارزیابی: هیوریستیک حداقل درگیری برای کمینه کردن تعداد فراکردهای ارضاء نشده

تعادل میان میزان حریصانه بودن و تصادفی بودن

مسائل سخت ارضا پذیری

- جملات 3-CNF تصادفی را در نظر بگیرید، مثلا:

$(\neg D \vee \neg B \vee C) \wedge (B \vee \neg A \vee \neg C) \wedge (\neg C \vee \neg B \vee E) \wedge (E \vee \neg D \vee B) \wedge (B \vee E \vee \neg C)$

5 سیمبول و 5 فراکرد — ۳۲ انتساب و ۱۶ مدل — به طور متوسط دو حدس کافی می باشد (برای یافتن مدل)

m = تعداد فراکردها (بند)

n = تعداد سیمبول ها

به نظر می رسد مسائل سخت نزدیک $m/n = 4.3$ باشند (نسبت بحرانی)

مسائل سخت ارضا پذیری

- مسائل نزدیک نقطه بحرانی بسیار سخت از دیگر مسائل تصادفی هستند.
- حتی در مسائل سخت، الگوریتم DPLL نسبتا کارآمد است – چند هزار مرحله به طور میانگین در مقایسه با $10^{15} \approx 2^{50}$ برای شمارش جدول درستی.
- در کل محدوده، الگوریتم WalkSAT بسیار سریعتر از DPLL می باشد.

مثال : انواع عاملها بر اساس منطق گزاره ای

۱. عاملهای مبتنی بر استنتاج

از الگوریتمهای استنتاج برای دنبال کردن دنیا و اسنباط خواص پنهان استفاده می کنند

۲. عاملهای مبتنی بر مدار

گزاره ها را به صورت بیتهای موجود در ثباتها بازنمایی و انتشار سیگنالها در مدارات منطقی آنها را به روز می کنند