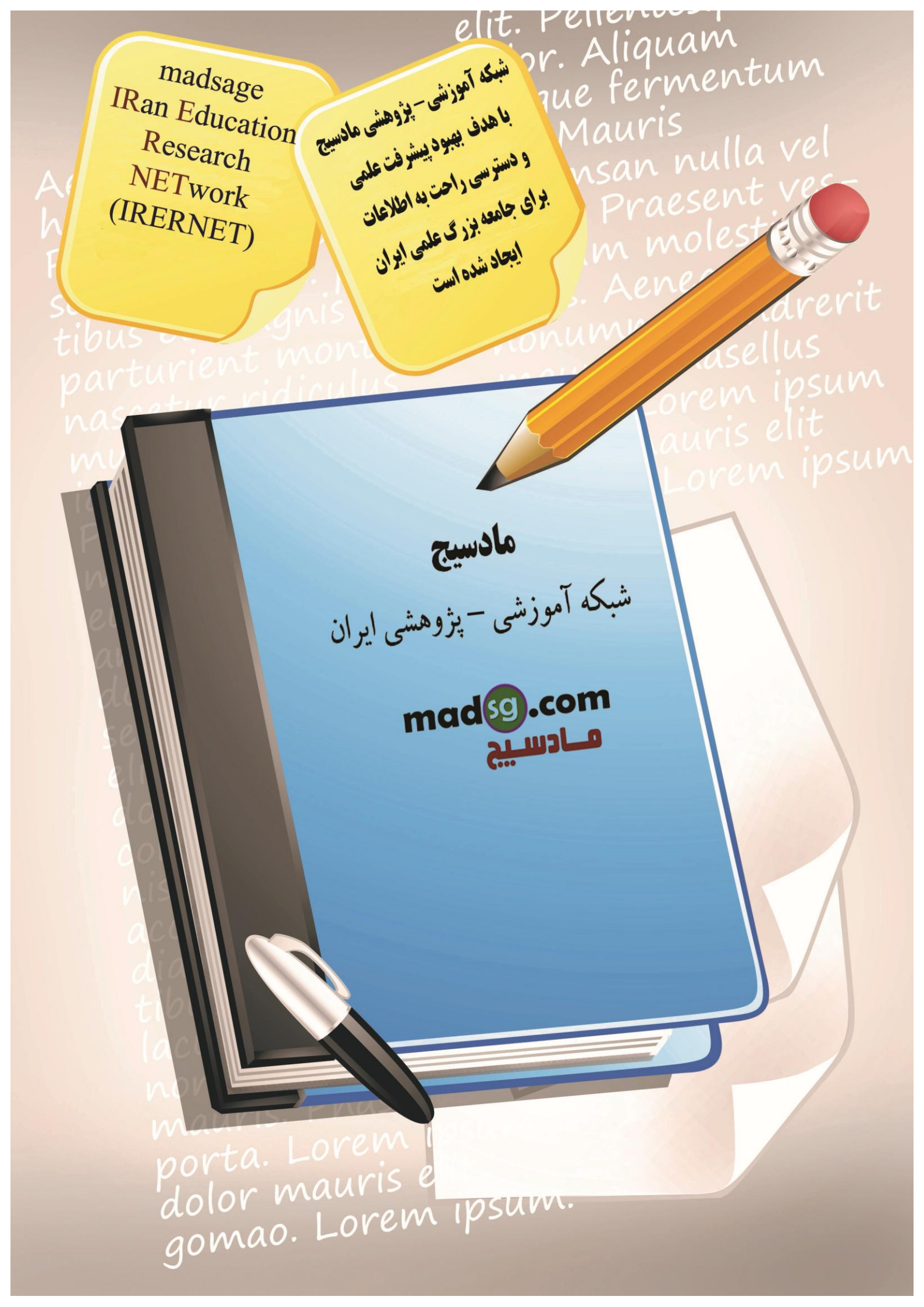

madsage
IRan Education
Research
NETwork
(IRERNET)
شبکه آموزشی-پژوهشی مادسیج
با هدف بهبود پیشرفت علمی
و دسترسی راحت به اطلاعات
برای جامعه بزرگ علمی ایران
ایجاد شده است
مادسیج
شبکه آموزشی - پژوهشی ایران
madsg.com
مادسیج

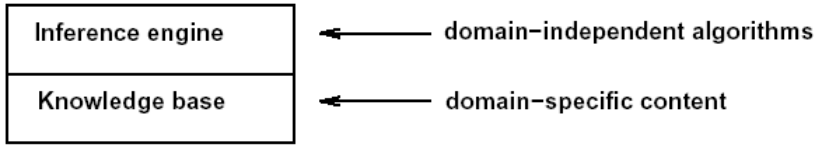
فصل هفتم

فصل هفتم
سید ناصر رضوی
E-mail:razavi@Comp.iust.ac.ir
۱۳۸۴

مقدمه

- عامل های مبتنی بر دانش
- محیط Wumpus
- منطق – مدل ها و استلزام
- منطق گزاره ای (بولین)
- هم ارزی، اعتبار و صدق پذیری
- قوانین استنتاج و اثبات تئوری
  - زنجیره استنتاج رو به جلو (forward chaining)
  - زنجیره استنتاج رو به عقب (backward chaining)
  - رزولوشن

پایگاه دانش



- پایگاه دانش = مجموعه ای از جملات در یک زبان رسمی
- رهیافت توصیفی برای ایجاد یک عامل (یا سیستم های دیگر):
  - به عامل آنچه را که نیاز دارد بداند، بگو (TELL)
  - سپس عامل می تواند از خود بپرسد (ASK) که چه عملی انجام دهد – پاسخ ها باید از KB **پیروی** کنند.
- می توان عامل ها را در سطح دانش در نظر گرفت:
  - یعنی، چه می دانند، بدون توجه به چگونگی پیاده سازی
- یا در سطح پیاده سازی:
  - یعنی، ساختارهای داده ای در KB و الگوریتم هایی که بر روی آنها کار می کنند.

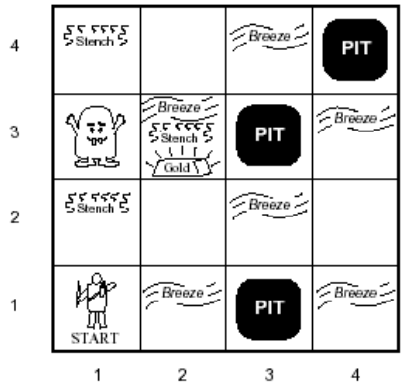
یک عامل ساده مبتنی بر دانش

```
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
```

- عامل باید قادر باشد:
  - حالات و اعمال و ... را بازنمایی کند.
  - ادراک جدید دریافت کند.
  - بازنمایی داخلی دنیا را بهنگام سازد.
  - خواص پنهان دنیا و اعمال مناسب را نتیجه گیری کند.

دنیای *Wumpus*

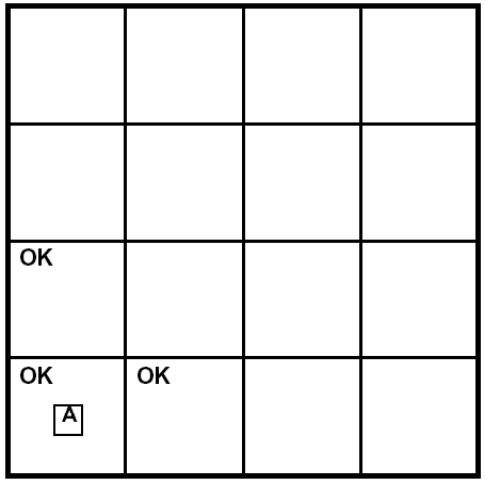


- **معیار کارآیی**
  - طلا 10000+، مرگ 1000-
  - هر عمل 1-، شلیک 10-
- **محیط**
  - خانه های مجاور وامپوس دارای بو هستند
  - خانه های مجاور چاله ها دارای نسیم هستند
  - در خانه حاوی طلا، درخشش وجود دارد
  - شلیک وامپوس را می کشد، اگر عامل رو به وامپوس باشد
  - تنها یک شلیک موثر است
  - اگر در خانه عامل طلا باشد، می تواند آنرا بردارد
  - عامل می تواند طلا را رها کند
- **حسگرها**
  - نسیم، درخشش، بو، ضربه و جیغ
- **محرک ها**
  - چرخش به چپ و راست
  - حرکت به جلو
  - برداشتن و رها کردن طلا و شلیک تیر

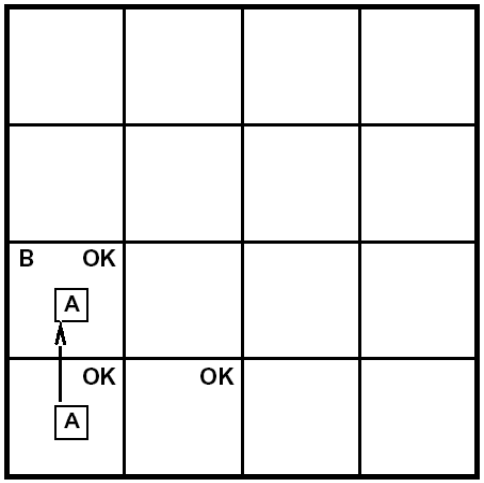
مشخصات محیط وامپوس

- دسترس پذیر؟؟ خیر-- تنها ادراک محلی میسر می باشد
- قطعی؟؟ بله– نتیجه اعمال کاملا مشخص است
- اپیزودیک؟؟ خیر
- ایستا؟؟ بله – وامپوس و چاله ها حرکت نمی کنند
- گسسته؟؟ بله
- تک-عاملی؟؟ بله – وامپوس اساساً یک ویژگی طبیعی است

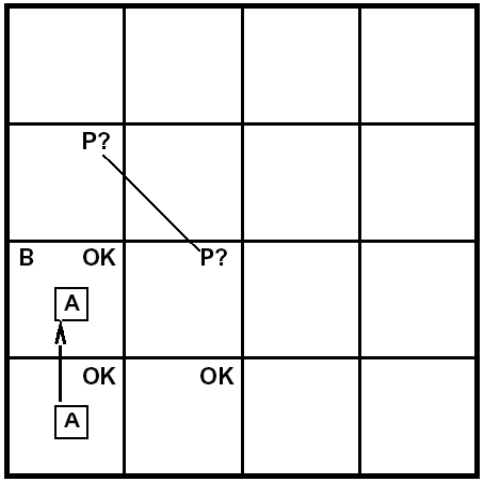
کاوش دنیای وامپوس



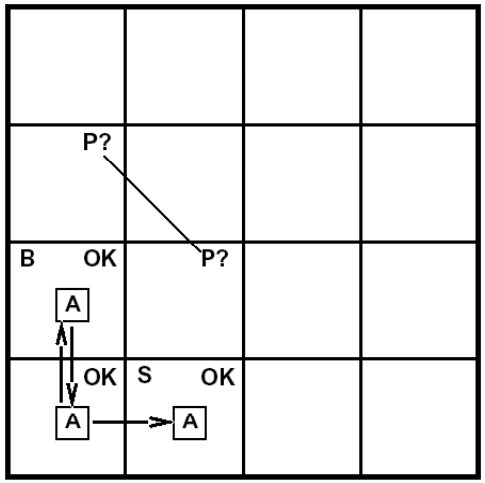
کاوش دنیای وامپوس



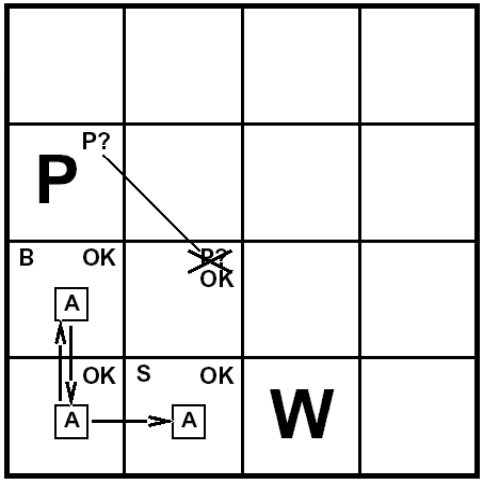
کاوش دنیای وامپوس



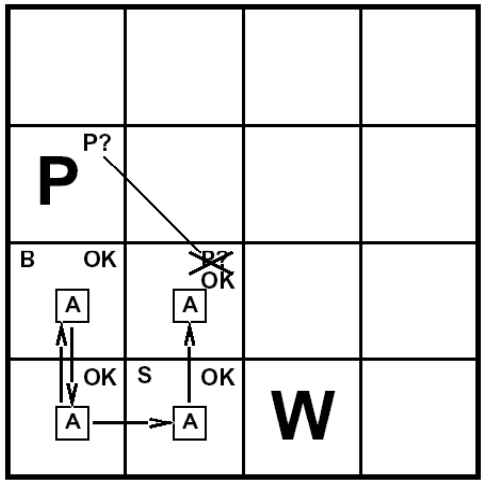
کاوش دنیای وامپوس



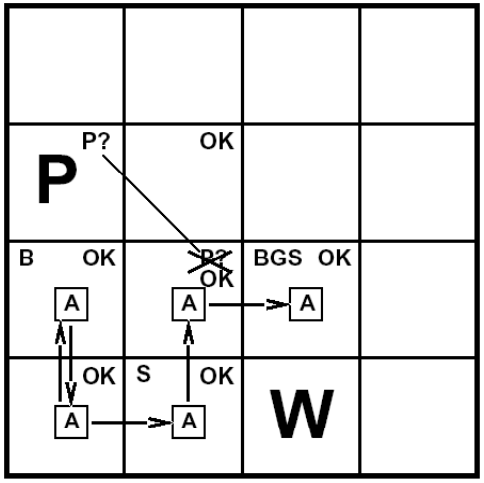
کاوش دنیای وامپوس



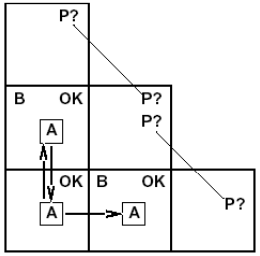
کاوش دنیای وامپوس

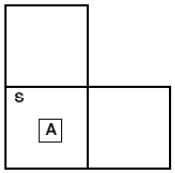


کاوش دنیای وامپوس



موقعیت های تصمیم گیری دیگر





- در خانه های (1, 2) و (2, 1) نسیم احساس می شود ← عمل مطمئنی وجود ندارد
- با فرض توزیع یکنواخت چاله ها، احتمال چاله در (2, 2) بیشتر

- احساس بو در (1, 1) ← قادر به حرکت نیست
- استفاده از استراتژی اجبار:

۱) به خانه روبرو شلیک کن
۲) اگر وامپوس آنجا بوده ← مرده ← امن
۳) اگر وامپوس آنجا نبوده ← امن

منطق

- منطق یک زبان رسمی برای بازنمایی دانش بطوری که بتوان از آن نتیجه گیری نمود.

- دستور ساختاری (syntax) : ساختار جملات زبان را تعریف می کند
- معنا (semantic): معنای جملات را تعریف می کند
  - یعنی، تعریف درستی یک جمله در یک دنیا

- مثال: زبان ریاضی
- $x + 2 \geq y$ جمله
- $x2 + y \geq$ جمله نیست
- جمله $x + 2 \geq y$ در دنیایی با $x=7, y=1$ درست و در دنیایی با $x=0$ و $y=6$ نادرست می باشد.

استلزام (entailment)

- استلزام بدین معناست که چیزی از چیز دیگری پیروی می کند:

$KB \models \alpha$

- پایگاه دانش KB مستلزم جمله α است، اگر و فقط اگر
- α در تمام دنیاهایی که در آن KB درست است، درست باشد.

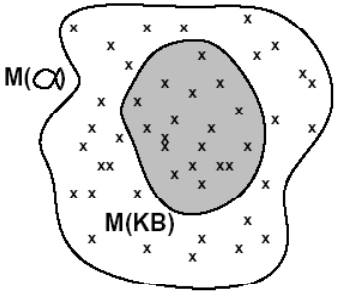
- مثال: $x + y = 4$ مستلزم $x + y = 4$ می باشد.

- استلزام رابطه ایست که بین ساختار جملات (syntax) و بر مبنای معنای جملات تعریف می شود.

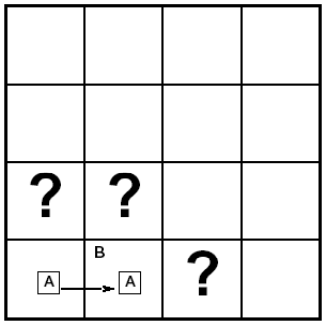
مدل ها

- منطق دانان عموماً بر حسب **مدل ها** فکر می کنند، که بطور رسمی دنیاهای ساخت یافته ای می باشند که درستی را می توان نسبت به آنها ارزیابی کرد.

- می گوییم m **مدلی از** جمله α می باشد اگر α در m درست باشد
- $M(\alpha)$ مجموعه تمام مدل های α می باشد
- بنابراین $KB \models \alpha$ اگر وفقط اگر $M(KB) \subseteq M(\alpha)$

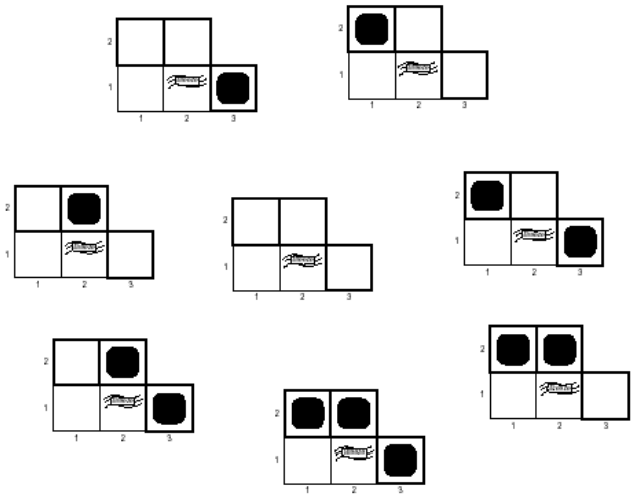


استلزام در دنیای وامپوس

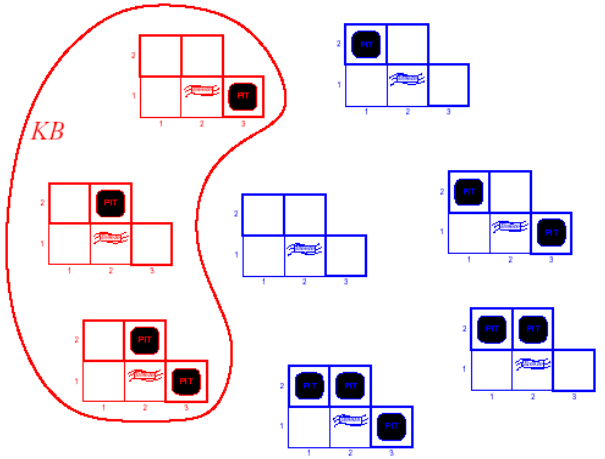


- موقعیت پس از [1, 1]، رفتن به راست، دریافت نسیم در [2, 1]
- مدلهای ممکن برای ? ها را تنها با فرض چاله ها در نظر بگیرید
- سه انتخاب بولین ← هشت مدل مختلف

مدل های وامپوس

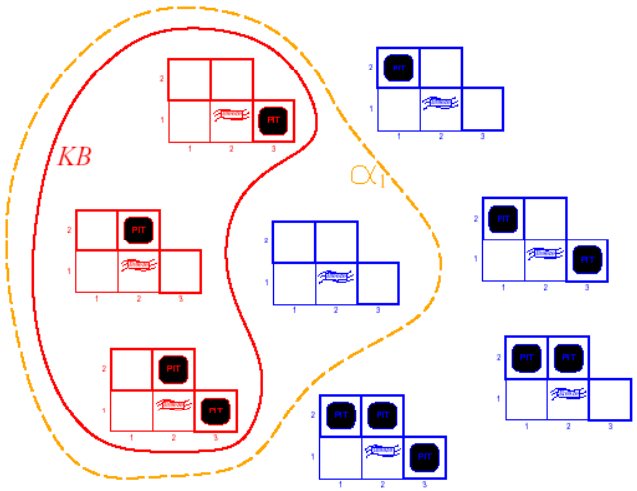


مدل های وامپوس



KB = Wumpus-world rules + observations

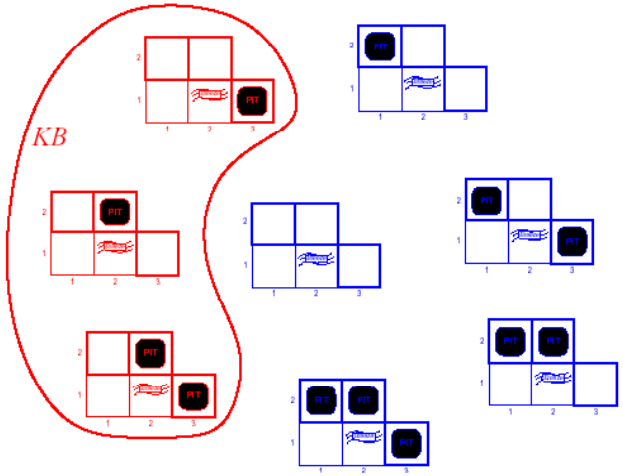
مدل های وامپوس



KB = Wumpus-world rules + observations

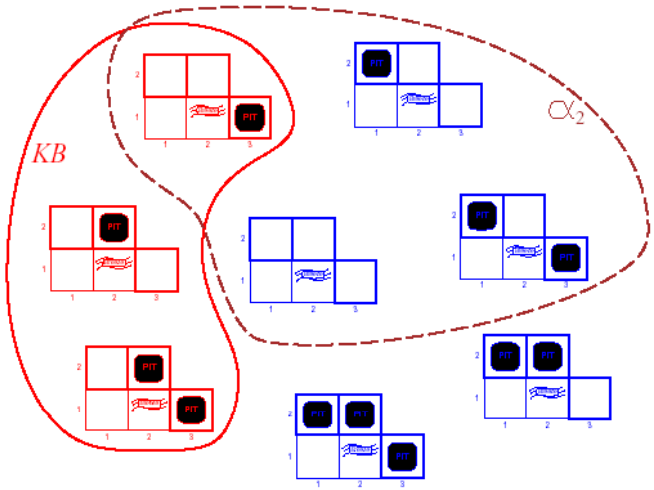
α_1 = "[1, 2] is safe", KB $\models \alpha_1$, proved by model checking

مدل های وامپوس



KB = Wumpus-world rules + observations

مدلهای وامپوس



KB = Wumpus-world rules + observations

α_2 = "[2, 2] is safe", KB $\not\models \alpha_2$

استنتاج (Inference)

- $KB \vdash_i \alpha$ = جمله α بوسیله رویه i از KB قابل اشتقاق می باشد.
- نتایج KB مانند یک انبار کاه می باشد، و α مانند یک سوزن
- استلزام = سوزن در انبار کاه؛ استنتاج = یافتن سوزن
- صحت (soundness): رویه i صحیح است اگر

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

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

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

- مثال: در منطق مرتبه اول (First Order Logic) یک رویه استناج کامل و صحیح وجود دارد.

منطق گزاره ای : ساختار

- منطق گزاره ای ساده ترین نوع منطق است – برای بیان ایده های ساده و مبنایی

- سیمبولهای گزاره ای P_1، P_2 و ... هر کدام یک جمله می باشند
- True و False ثابت های گزاره ای می باشند و هر کدام به تنهایی یک جمله اند

- اگر S جمله باشد، آنگاه $\neg$ S نیز یک جمله است (نقیض)
- اگر S_1 و S_2 جمله باشند، $S_1 \wedge S_2$ نیز یک جمله است (ترکیب عطفی)
- اگر S_1 و S_2 جمله باشند، $S_1 \vee S_2$ نیز یک جمله است (ترکیب فصلی)
- اگر S_1 و S_2 جمله باشند، $S_1 \Rightarrow S_2$ نیز یک جمله است (ترکیب شرطی)
- اگر S_1 و S_2 جمله باشند، $S_1 \Leftrightarrow S_2$ نیز یک جمله است (ترکیب دوشرطی)

منطق گزاره ای: معنا

- 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*

جملات دنیای وامپوس

- 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 می 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 یک خانه نسیم می وزد اگر و فقط اگر چاله ای مجاور آن باشد"

استفاده از جدول درستی برای استنتاج

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
false	*false*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*false*	*false*	*false*	*false*	*false*	*true*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*false*

استنتاج بوسیله شمارش

- شمارش تمام مدل ها به روش اول – عمق صحیح و کامل است

```
function TT-ENTAILS?(KB, α) returns true or false
    symbols ← a list of the proposition symbols in KB and α
    return TT-CHECK-ALL(KB, α, symbols, [])

function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
    if EMPTY?( symbols) then
        if PL-TRUE(KB, model) then return PL-TRUE(α, model)
        else return true
    else do
        P ← FIRST( symbols); rest ← REST( symbols)
        return TT-CHECK-ALL(KB, α, rest, EXTEND(P, true, model)) and
               TT-CHECK-ALL(KB, α, rest, EXTEND(P, false, model))
```

برای n سیمبول $O(2^n)$

هم ارزی منطقی

- دو جمله هم ارز منطقی می باشند، اگر و فقط اگر هر دو در مدلهای یکسانی درست باشند.
- $\alpha \equiv ß$ iff $\alpha \models \beta$ and $\beta \models \alpha$

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

اعتبار و صدق پذیری

- یک جمله معتبر (valid) است اگر در تمام مدل ها درست باشد
  - مثال: True، $A \vee \sim A$ ، $A \Rightarrow A$ ، $(A \wedge (A \Rightarrow B) \Rightarrow B)$
- ارتباط معتبر بودن با استنتاج:

$KB \models \alpha$ iff $(KB \Rightarrow \alpha)$ is ***valid***

- یک جمله صدق پذیر (satisfiable) اگر در بعضی از مدل ها درست باشد
  - مثال: $A \vee B$
- یک جمله صدق ناپذیر است اگر در هیچ مدلی درست نباشد
  - مثال: $A \wedge \sim A$
- ارتباط صدق پذیری با استنتاج:

$KB \models \alpha$ iff $(KB \wedge \sim\alpha)$ is ***unsatisfiabe***

روش های اثبات

- روش های اثبات به دو نوع تقسیم می شوند:
- اعمال قوانین استنتاج:
  - تولید صحیح جملات جدید از جملات قدیمی
  - اثبات = دنباله ای از اعمال قوانین استنتاج

  می توان از قوانین استنتاج به عنوان عملگرها در الگوریتم استاندارد جستجو استفاده کرد.
  - اغلب نیاز به تبدیل جملات به یک شکل نرمال دارند.
- بررسی مدل:
  - شمارش جدول درستی (برحسب n نمایی)
  - عقبگرد بهبود یافته (DPLL)
  - جستجوی هیوریستیک در فضای مدل (صحیح اما نا کامل)

استنتاج رو به جلو و رو به عقب

- شکل نرمال Horn (HNF):
  KB = ترکیب عطفی عبارت های Horn
- عبارت Horn =
  - سیمبول گزاره ای
  - (سیمبول گزاره ای) ⇒(ترکیب عطفی سیمبول های گزاره ای)

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

  - مثال :

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

- قانون استنتاج Modes Ponens برای شکل Horn:

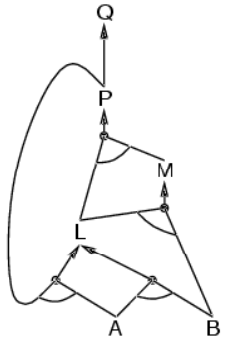
$$\frac{\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n, \quad \alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n \Rightarrow \beta}{\beta}$$

- می تواند در هر دو روش روبه جلو و روبه عقب بکار رود.
- این روشها بسیار طبیعی هستند و در زمان خطی (برحسب اندازه KB) اجرا می شوند.

استنتاج روبه جلو

- 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.

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

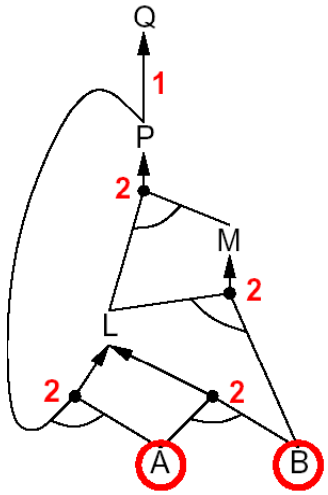


الگوریتم استنتاج روبه جلو

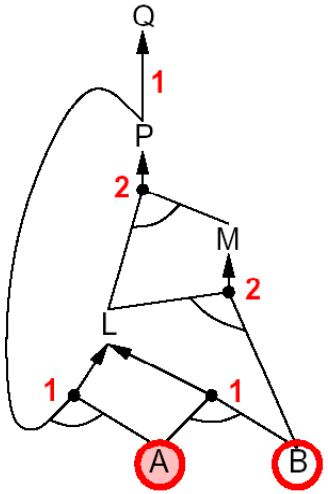
```
function PL-FC-ENTAILS?(KB, q) returns true or false
  local variables: count, a table, indexed by clause, initially the number of premises
                   inferred, a table, indexed by symbol, each entry initially false
                   agenda, a list of symbols, initially the symbols known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

- استنتاج روبه جلو برای پایگاه دانش در شکل Horn کامل و صحیح است.

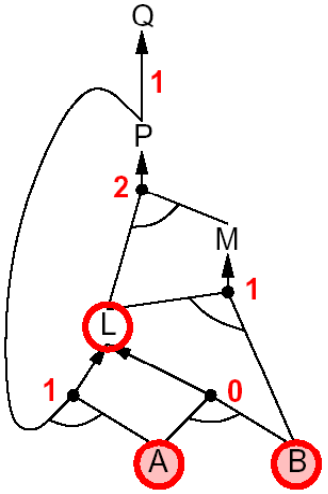
مثال استنتاج رو به جلو



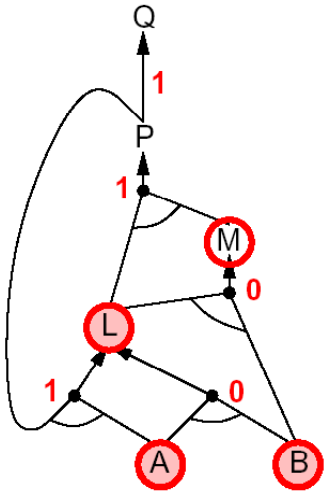
مثال استنتاج رو به جلو



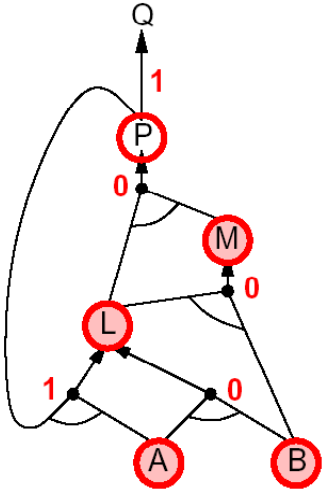
مثال استنتاج رو به جلو



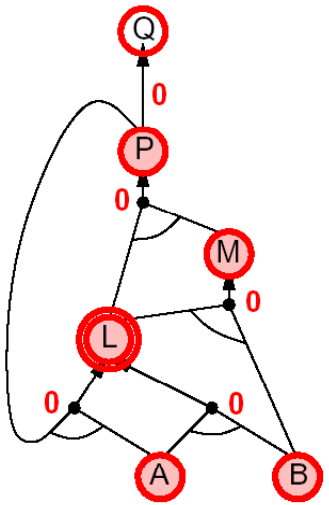
مثال استنتاج رو به جلو



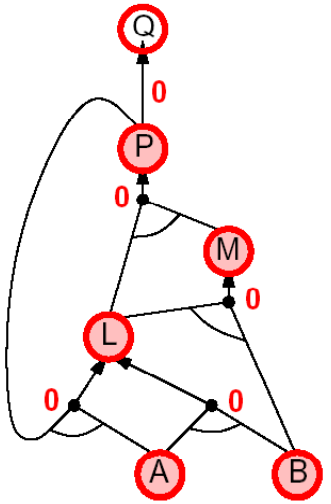
مثال استنتاج رو به جلو



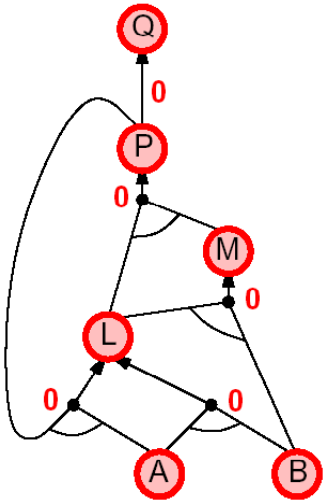
مثال استنتاج رو به جلو



مثال استنتاج رو به جلو



مثال استنتاج رو به جلو



استنتاج رو به عقب

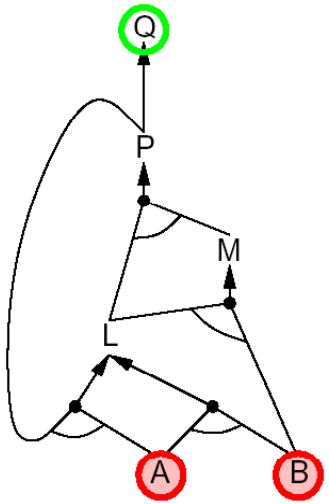
- ایده: برای اثبات q به سمت عقب حرکت کن
- برای اثبات q بوسیله BC،
  - بررسی کن که آیا q اکنون ثابت شده می باشد، یا
  - بوسیله BC تمام شرایط برخی از قوانین را که نتیجه آنها q است اثبات کن

- اجتناب از حلقه: بررسی قرار داشتن زیرهدف جدید روی پشته هدف
- اجتناب از اعمال تکراری: بررسی اینکه آیا زیرهدف جدید

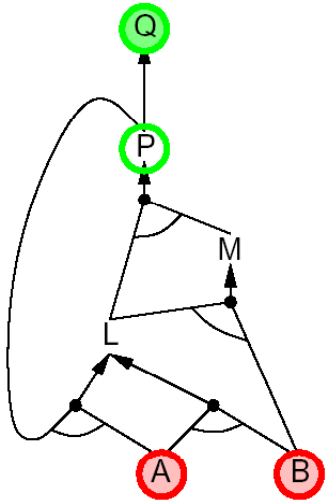
۱) قبلا درستی اش اثبات شده، یا

۲) قبلا شکست خورده است (fail)

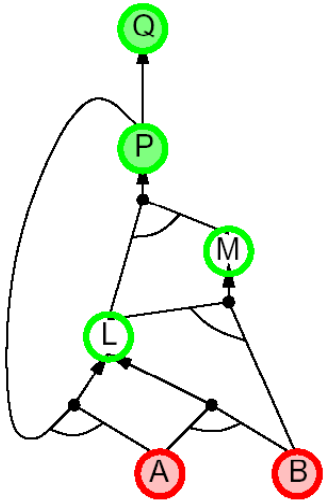
مثال از استنتاج رو به عقب



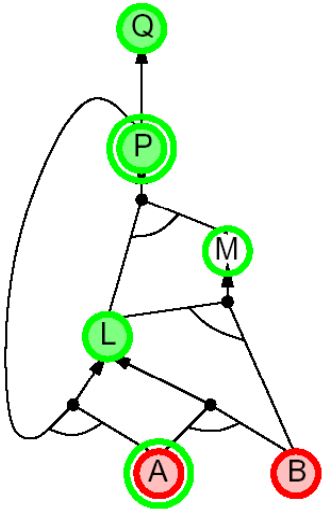
مثال از استنتاج رو به عقب



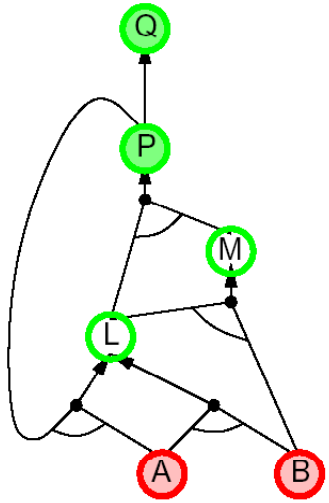
مثال از استنتاج رو به عقب



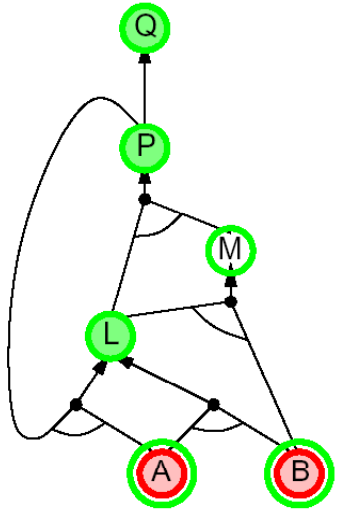
مثال از استنتاج رو به عقب



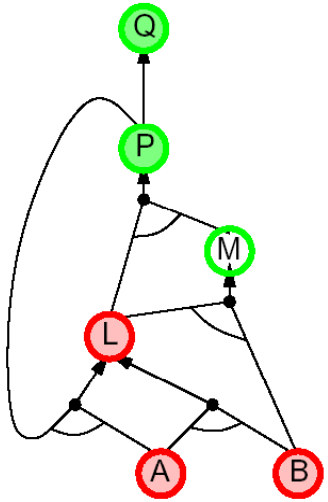
مثال از استنتاج رو به عقب



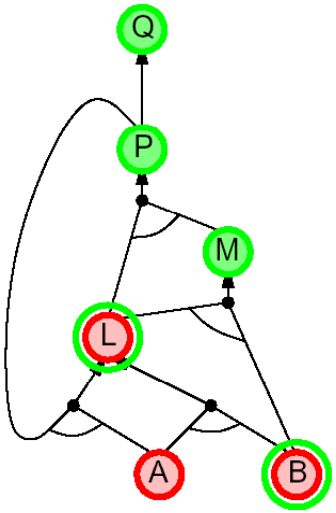
مثال از استنتاج رو به عقب



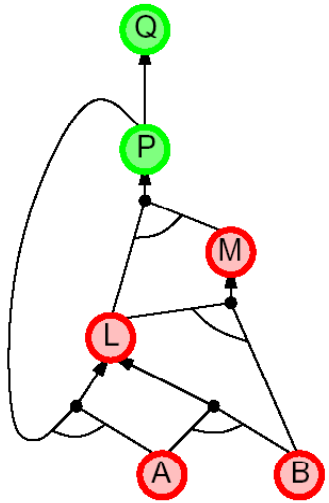
مثال از استنتاج رو به عقب



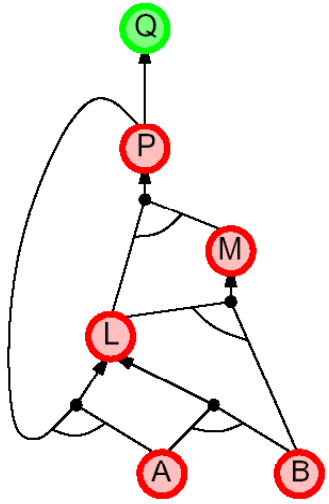
مثال از استنتاج رو به عقب



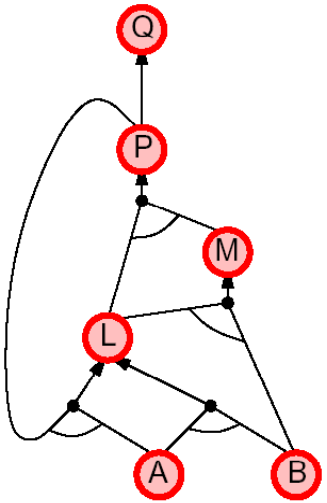
مثال از استنتاج رو به عقب



مثال از استنتاج رو به عقب



مثال از استنتاج رو به عقب



مقایسه دو روش

- FC:
  - بر مبنای داده (data driven)
  - ممکن است کارهای بسیاری انجام دهد که به هدف مربوط نمی شوند
- BC:
  - بر مبنای هدف (goal driven)
  - پیچیدگی BC می تواند بسیار بهتر از خطی نسبت به اندازه KB باشد.

Resolution

- شکل نرمال عطفی (CNF)

conjunctions of *disjunctions* of *literals*

Clauses

- مثال:

$$(A \vee \sim B) \wedge (B \vee \sim C \vee \sim D)$$

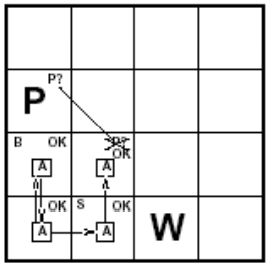
- قانون استنتاج رزولوشن (برای CNF): صحیح و کامل برای منطق گزاره ای

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

l_i و m_j نقیض یکدیگرند.

Resolution

- مثال :



$$\frac{P_{1,3} \vee P_{2,2}, \quad \neg P_{2,2}}{P_{1,3}}$$

- رزولوشن برای منطق گزاره ای صحیح و کامل می باشد.

Resolution

- صحت قانون استنتاج رزولوشن

$$\frac{\neg(l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k) \Rightarrow l_i \qquad \neg m_j \Rightarrow (m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)}{\neg(l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k) \Rightarrow (m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)}$$

تبدیل به CNF

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

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 distributivity law ($\wedge$ over $\vee$) and flatten:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Resolution الگوریتم

- اثبات بوسیله تناقض، یعنی نشان بده KB $\wedge\sim\alpha$ صدق ناپذیر است

function PL-RESOLUTION(*KB*, α) **returns** *true* or *false*

clauses ← the set of clauses in the CNF representation of KB $\wedge\sim\alpha$

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*

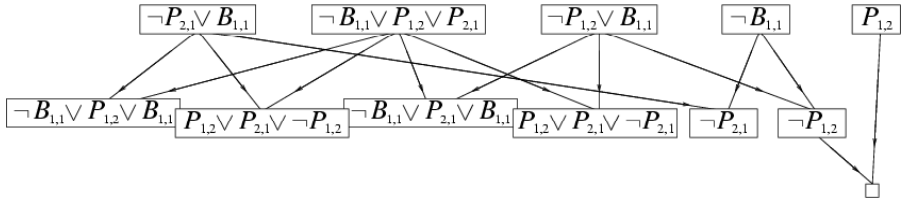
new ← *new* $\cup$ *resolvents*

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

clauses ← *clauses* $\cup$ *new*

مثال برای رزولوشن

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



الگوریتم های استنتاج کارآ در منطق گزاره ای

- دو خانواده از الگوریتم های استنتاج کارآ برای منطق گزاره ای

- الگوریتم های کامل جستجوی عقبگرد:
  - الگوریتم DPLL (Davis, Putnam, Logemann, Loveland)
- الگوریتم ناکامل جستجوی محلی
  - الگوریتم WalkSAT

DPLL الگوریتم

- تعیین کن که آیا یک جمله ورودی در زبان منطق گزاره ای (در شکل نرمال CNF) صدق پذیر است یا خیر.
- بهبودها نسبت به روش شمارش جدول درستی:

۱- خاتمه زود هنگام

- یک فراکرد (Clause) درست است اگر هر یک از لیترال ها درست باشد.
- یک جمله نادرست است اگر هر یک از فراکردهای آن نادرست باشد.

۲- هیوریستیک سیمبول محض

- سیمبول محض: سیمبولی که در تمام فراکرد ها با یک علامت ظاهر شود.
- مثال: در سه فراکرد $(A \vee \neg B), (\neg B \vee \neg C), (C \vee A)$ سیمبولهای A و B سیمبول محض می باشند، اما C یک سیمبول محض نیست.
- لیترال یک سیمبول محض را درست تلقی کن.

۳- هیوریستیک فراکرد واحد

- فراکرد واحد: تنها شامل یک لیترال می باشد. یا فراکردی که تمام لیترال های آن غیر از یک لیترال، نادرست می باشند.
- تنها لیترال موجود در یک فراکرد واحد باید درست باشد.

الگوریتم DPLL

```
function DPLL-SATISFIABLE?(s) returns true or false
  inputs: s, a sentence in propositional logic
  clauses ← the set of clauses in the CNF representation of s
  symbols ← 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 ← FIND-PURE-SYMBOL(symbols, clauses, model)
  if P is non-null then return DPLL(clauses, symbols–P, [P = value|model])
  P, value ← FIND-UNIT-CLAUSE(clauses, model)
  if P is non-null then return DPLL(clauses, symbols–P, [P = value|model])
  P ← FIRST(symbols); rest ← REST(symbols)
  return DPLL(clauses, rest, [P = true|model]) or
         DPLL(clauses, rest, [P = false|model])
```


الگوریتم WalkSAT

- الگوریتم جستجوی محلی و ناکامل
- تابع ارزیابی: هیوریستیک حداقل درگیری برای کمینه کردن تعداد فراکردهای ارضاء نشده
- تعادل میان میزان حریصانه بودن و تصادفی بودن

الگوریتم 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
          max-flips, number of flips allowed before giving up
  model ← 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 ← 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
```


مسائل ارضاءپذیری سخت

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

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

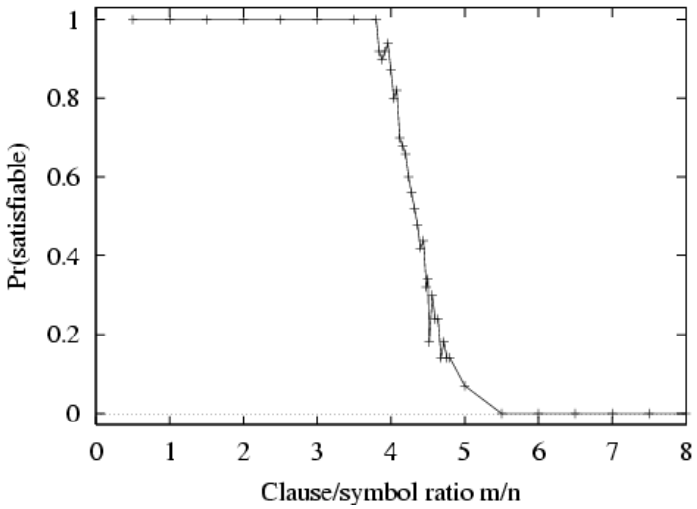
5 سیمبول و 5 فراکرد – ۳۲ انتساب و ۱۶ مدل – به طور متوسط دو حدس کافی می باشد (برای یافتن مدل)

m = تعداد فراکردها

n = تعداد سیمبول ها

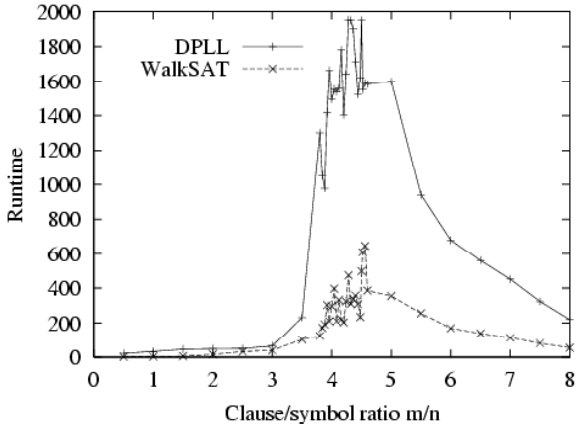
به نظر می رسد مسائل سخت نزدیک $m/n = 4.3$ باشند (نسبت بحرانی)

مسائل ارضاءپذیری سخت



$n = 50$

مسائل ارضاءپذیری سخت



زمان اجرا برای 100 جمله 3-CNF ارضاء پذیر، $n = 50$

مسائل ارضاءپذیری سخت

- سه نکته واضح در شکل قبل:

– 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 $2^{50} \approx 10^{15}$ برای شمارش جدول درستی.

– در کل محدوده، الگوریتم WalkSAT بسیار سریعتر از DPLL می باشد.

عامل های مبتنی بر استنتاج در دنیای وامپوس

- یک عامل دنیای وامپوس با استفاده از منطق گزاره ای:

$\neg P_{1,1}$
$\neg W_{1,1}$
$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y})$
$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$
$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$
$\neg W_{1,1} \vee \neg W_{1,2}$
$\neg W_{1,1} \vee \neg W_{1,3}$
…

- ۶۴ سیمبول گزاره ای متفاوت
- ۱۵۵ جمله

```
function PL-WUMPUS-AGENT(percept) returns an action
  inputs: percept, a list, [stench, breeze, glitter]
  static: KB, initially containing the "physics" of the wumpus world
          x, y, orientation, the agent's position (init. [1,1]) and orient. (init. right)
          visited, an array indicating which squares have been visited, initially false
          action, the agent's most recent action, initially null
          plan, an action sequence, initially empty

  update x,y,orientation, visited based on action
  if stench then TELL(KB, S_{x,y}) else TELL(KB, ¬ S_{x,y})
  if breeze then TELL(KB, B_{x,y}) else TELL(KB, ¬ B_{x,y})
  if glitter then action ← grab
  else if plan is nonempty then action ← POP(plan)
  else if for some fringe square [i,j], ASK(KB, (¬ P_{i,j} ∧ ¬ W_{i,j})) is true or
          for some fringe square [i,j], ASK(KB, (P_{i,j} ∨ W_{i,j})) is false then do
      plan ← A*-GRAPH-SEARCH(ROUTE-PB([x,y], orientation, [i,j], visited))
      action ← POP(plan)
  else action ← a randomly chosen move
  return action
```


محدودیت های منطق گزاره‌ای

- به طور کلی منطق گزاره ای از قدرت بیان کافی برخوردار نیست:
  - برای هر مربع شامل جملات "فیزیکی" آن مربع
  - برای هر زمان t و هر مکان [x, y]

$$L_{x,y}^{t} \wedge FacingRight^{t} \wedge Forward^{t} \Rightarrow L_{x+1,y}^{t+1}$$

خلاصه

- عامل های منطقی از استنتاج بر روی یک پایگاه دانش برای اشتقاق دانش جدید و تصمیم گیری استفاده می کنند
- مفاهیم مبنایی منطق:
  - ساختار (syntax)
  - معنا (semantics)
  - استلزام (entailments)
  - استنتاج (inference)
  - صحت (soundness)
  - کامل بودن (completeness)
- استنتاج رو به جلو و رو به عقب برای عبارت های Horn دارای زمان خطی هستند و کامل می باشند.
- رزولوشن برای منطق گزاره ای کامل است
- منطق گزاره ای در بازنمایی دانش ضعیف می باشد

شبکه آموزشی- پژوهشی مادسیج
با هدف بهبود پیشرفت علمی
و دسترسی راحت به اطلاعات
برای جامعه بزرگ علمی ایران
ایجاد شده است

madsg.com

مادسیج

IRan Education & Research NETwork
(IRERNET)