به نام خدا

هوش مصنوعی

فصل هفتم

عامل های منطقی

LOGICAL AGENTS

فهرست :

- عامل های مبتنی بر دانش
- منطق گزاره ای
- الگو های استدلال در منطق گزاره ای
- الگوریتم Resolution

مقدمه:

- دو مفهوم اساسی در در حوزه هوش مصنوعی **بازنمایی دانش** (یعنی نحوه نمایش و ذخیره دانش) و **فرآیند های استدلال** هستند و **عامل مبتنی بر دانش** (Knowledge-based agent) بر اساس دو مفهوم فوق بنا می شود
- انسان ها دارای دانش هستند و براساس آن استدلال می کنند. وجود دانش و استدلال برای عامل های مصنوعی نیز اهمیت دارد، زیرا منجر به رفتار موفق آنها می شود که در صورت عدم استفاده از دانش و استدلال، امکان پذیر نیست
- دانش در مورد نتایج اعمال، عامل حل مسأله را قادر می سازد در محیطهای پیچیده به خوبی عمل کند.
- عامل مبتنی بر دانش در مقایسه با عامل حل مسئله تفاوت ها و برتری های دارد

تفاوت ها و برتری های عامل مبتنی بر دانش در مقایسه با عامل حل مسئله:

❖ **عامل حل مسئله، دانش بسیار محدودی دارد:** عامل حل مسئله فقط تعریف مسئله ای که باید حل کند و احتمالا یک تابع هیستوریک در مورد مسئله دارد. عامل های مبتنی بر دانش، دانش خود از یک مسئله را در قالب مجموعه از گزاره های منطقی ذخیره و از روش استنتاج منطقی برای تصمیم گیری استفاده می کنند.

❖ **دانش عامل حل مسئله بسیار خاص و غیر قابل انعطاف است:** عامل های مبتنی بر دانش، می توانند با داشتن دانش کلی، از ترکیب دانش خود برای رسیدن به هدف استفاده کنند. (مثل اثبات یک قضیه با استفاده از اصول کلی اولیه و فرض های موجود در صورت قضیه)

❖ **عامل های مبتنی بر دانش در مواجهه با محیط های قابل مشاهده جزئی موفق ترند:** عامل مبتنی بر دانش می تواند دانش کلی را با ادراک فعلی ترکیب کند و قبل از انتخاب عمل، جنبه های مخفی حالت فعلی را کشف کند.

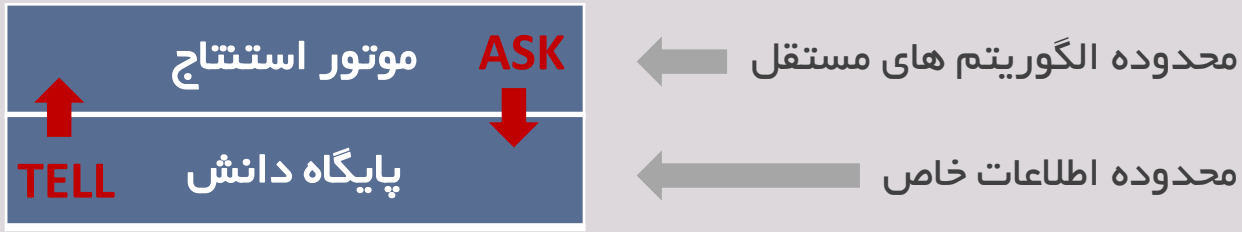
❖ **عامل های مبتنی بر دانش انعطاف پذیرند:** این عامل ها وظایف جدیدی را در قالب اهداف جدید می پذیرند، می توان به آن ها اطلاعات جدیدی داد یا خودشان اطلاعاتی را در مورد محیطشان یاد بگیرند. با بروزرسانی دانش می توانند خود را با تغییرات محیط وقف دهند.

عامل های مبتنی بر دانش:

عامل مبتنی بر دانش دو مولفه دارد:

1. پایگاه دانش یا Knowledge base (به اختصار KB)
2. موتور استنتاج یا Interface Engine

پایگاه دانش، مولفه اصلی این عامل است و مجموعه ای از **جملات** است. **جمله**، زبان نمایش دانش است. هر جمله، یک واقعیت در مورد دنیای عامل را نشان می دهد.



عامل های مبتنی بر دانش (ادامه):

- پایگاه دانش می تواند شامل دانش پیش زمینه (دانش اولیه) باشد یا شامل دانشی باشد که به مرور یاد می گیرد.
- برنامه ی عامل ابتدا به پایگاه دانش می گوید چه چیزی را ادراک کرده است سپس از پایگاه دانش سوال می کند چه فعالیتی باید انجام دهد.
- پس بر روی KB دو عمل انجام می شود:

1. TELL: جمله ای را به پایگاه دانش اضافه می کند
2. ASK: جمله ای را از پایگاه دانش استخراج می کند

- هر دو کار ممکن است شامل استنتاج باشند. استنتاج یعنی جملات جدید از جملات قدیم مشتق شوند.

عامل های مبتنی بر دانش (ادامه):

❖ عامل مبتنی بر دانش باید بتواند:

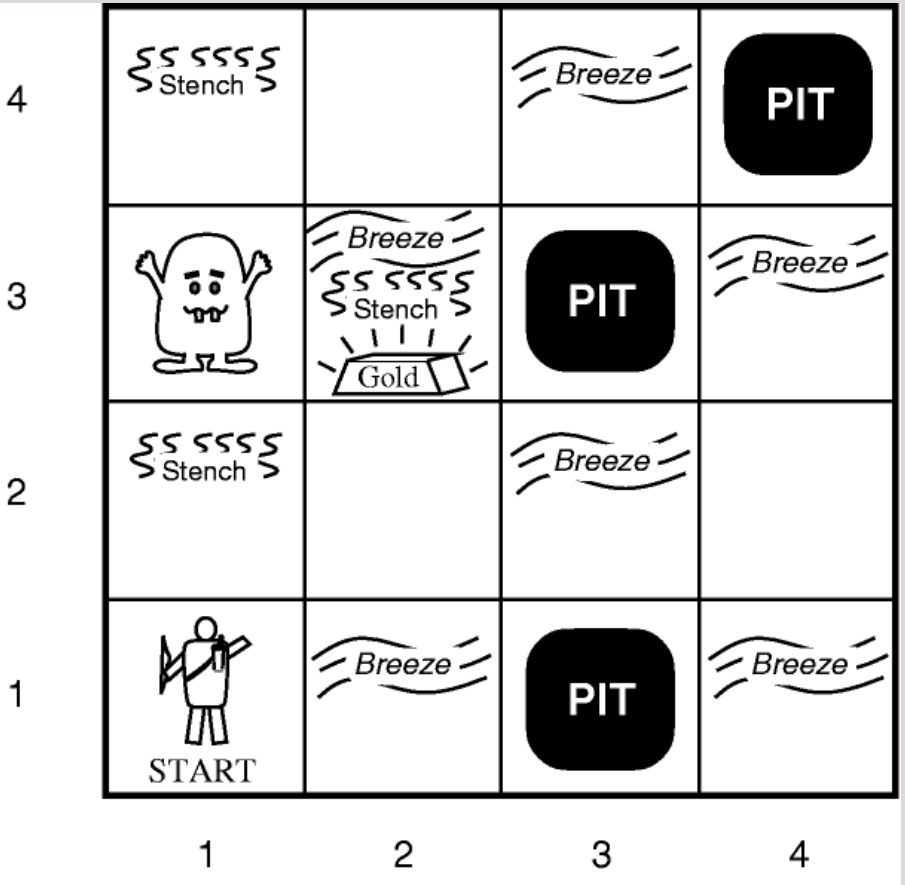
1. حالت و فعالیت های مختلف را نمایش دهد
2. ترکیب ادراک جدید
3. بروز کردن تصور داخلی خود از جهان
4. استنباط خصوصیات مخفی جهان
5. استنتاج فعالیت های مناسب

توجه: عامل مبتنی بر دانش، بسیار شبیه به عامل هایی با حالت درونی (داخلی) است.

❖ یک عامل را می توان در دو سطح بررسی کرد:

1. **در سطح دانش**: عامل چه چیزی را می داند و اهداف آن کدامند، بدون توجه به پیاده سازی
2. **در سطح پیاده سازی**: ساختمان داده اطلاعات پایگاه دانش و چگونگی دستکاری آن ها

دنیای وامپوس (The Wumpus World):



- دنیای وامپوس، یک محیط فرضی را نشان می دهد. غاری شامل اتاق هایی با راهروهای متصل به هم است. در جایی از این غار، موجود تخیلی به نام وامپوس کمین کرده است
- وامپوس حیوانی است که هر کس وارد اتاقش شود او را می خورد. عامل می تواند به وامپوس شلیک کند ولی فقط یک تیر در اختیار دارد. در اطراف خانه وامپوس، بوی بد به مشام می رسد
- در بعضی از اتاق ها، چاله وجود دارد و هر کسی که به آن اتاق ها وارد شود (به جز وامپوس که بزرگ است و در آن، جا نمی شود)، در دام می افتد
- تنها سودی که برای زندگی کردن در این محیط وجود دارد، یافتن خانه حاوی طلاست

عامل های منطقی (مثال دنیای وامپوس):

❖ معیار کارآیی:

- ۱۰۰۰+: برای انتخاب طلا
- ۱۰۰۰-: برای افتادن در گودال یا خورده شدن توسط وامپوس
- ۱-: برای هر مرحله (هر عملی که انجام شود)
- ۱۰-: برای استفاده از تیر

❖ محیط:

- بوی بد در مربعهای همجوار وامپوس، نسیم در مربع های همجوار گودال، درخشش در مربع حاوی طلا، کشته شدن وامپوس با شلیک در صورت مقابله، تیر فقط مستقیم عمل می کند، برداشتن و انداختن طلا

❖ حسگر ها:

- بو بد، نسیم، درخشش

❖ محرک ها:

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

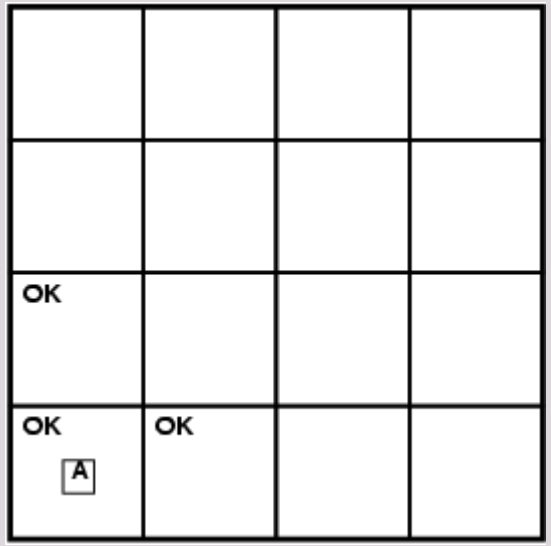
عامل های منطقی (مثال دنیای وامپوس):

- **قابل مشاهده کامل**: خیر، فقط ادراک محلی و قابل مشاهده جزئی
- **قطعی**: بله، نتیجه دقیقا مشخص است
- **رویدادی**: خیر، ترتیبی از فعالیت هاست
- **ایستا**: بله، وامپوس و گودال ها حرکت نمی کنند
- **گسسته**: بله، گسسته است
- **تک عامله**: بله، تک عاملی است

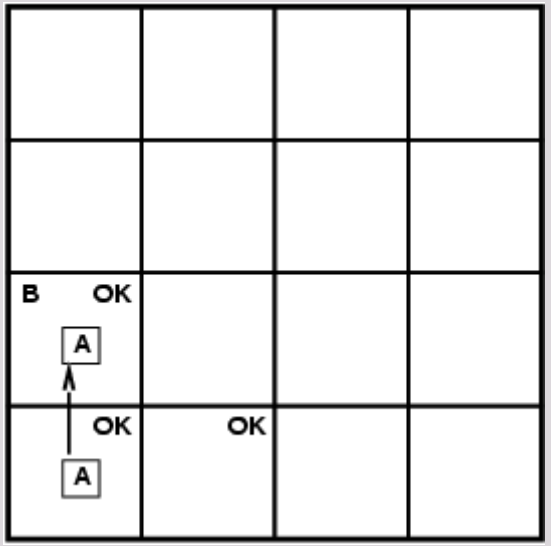
توجه: وامپوس یک عامل نیست بلکه یک ویژگی محیط است.

کاوش در دنیای وامپوس:

- عامل = A
- نسیم = B
- درخشش،طلا = G
- مربع امن = OK
- گودال = P
- تعفن = S
- ملاقات شده = V
- وامپوس = W



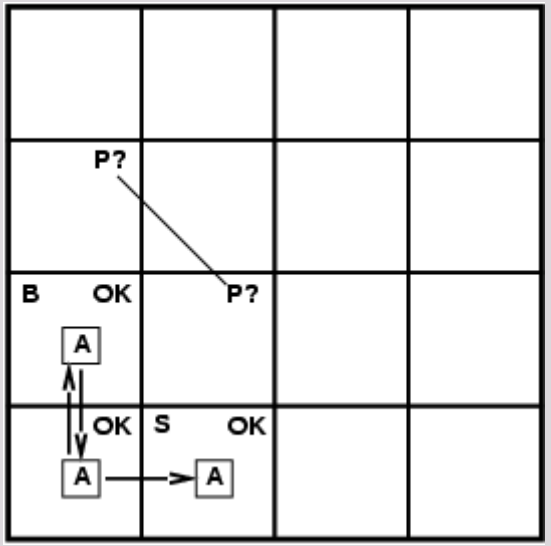
شماره ۱



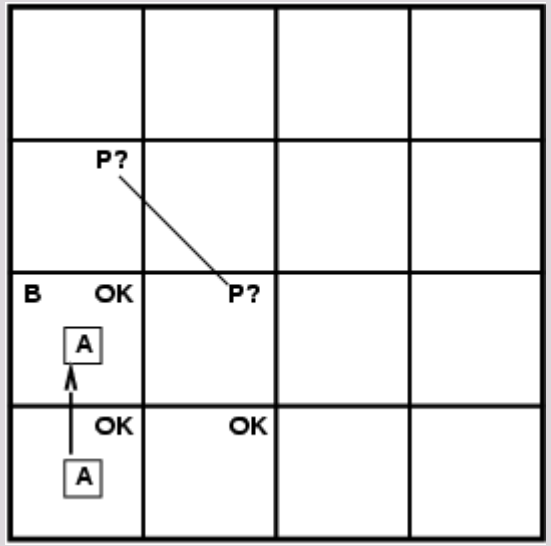
شماره ۲

کاوش در دنیای وامپوس (ادامه):

- عامل = A
- نسیم = B
- درخشش،طلا = G
- مربع امن = OK
- گودال = P
- تعفن = S
- ملاقات شده = V
- وامپوس = W



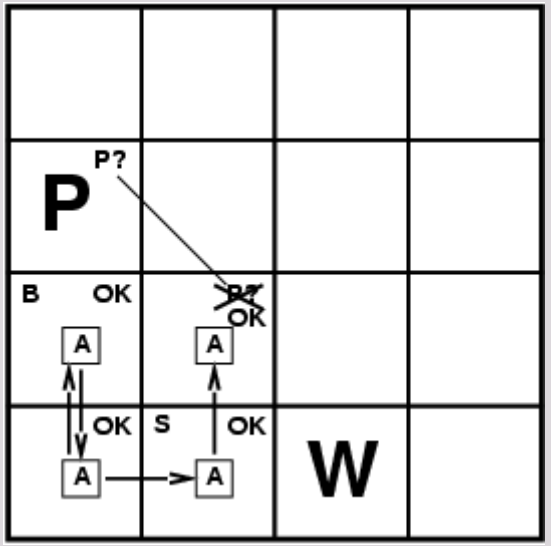
شماره ۴



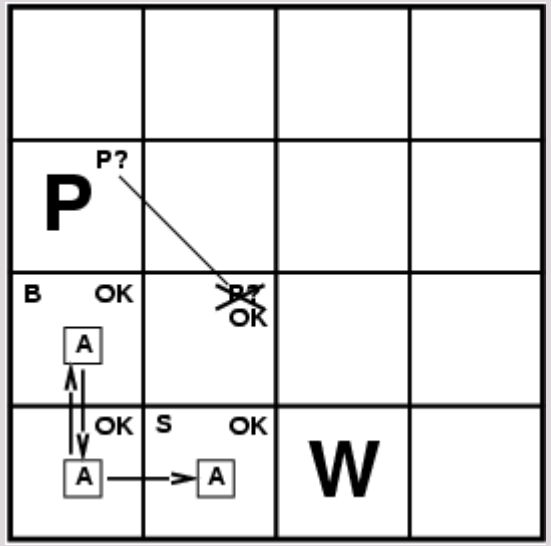
شماره ۳

کاوش در دنیای وامپوس (ادامه ۲):

- عامل = A
- نسیم = B
- درخشش،طلا = G
- مربع امن = OK
- گودال = P
- تعفن = S
- ملاقات شده = V
- وامپوس = W



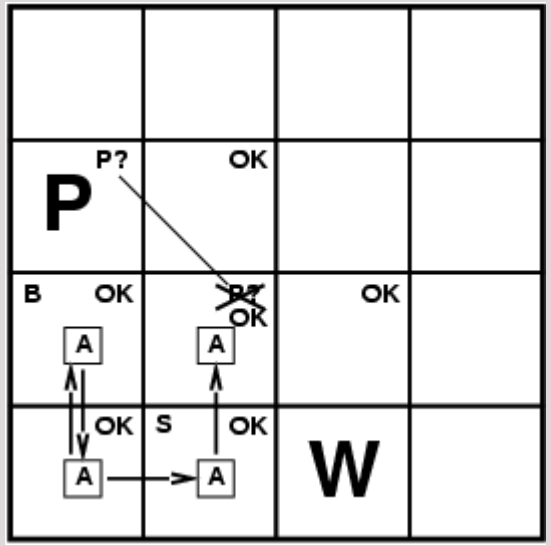
شماره ۶



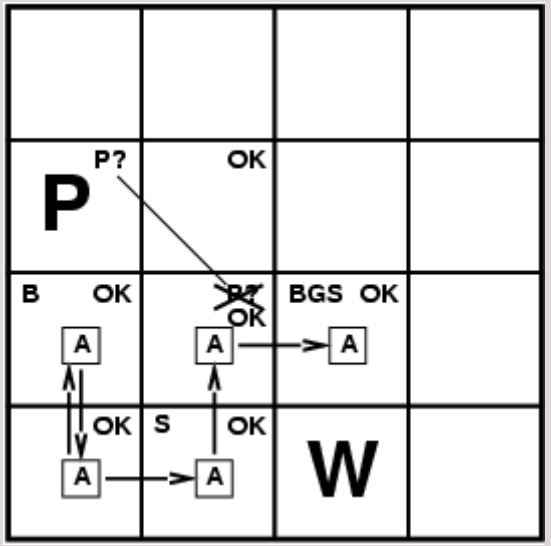
شماره ۵

کاوش در دنیای وامپوس (ادامه ۳):

- عامل = A
- نسیم = B
- درخشش،طلا = G
- مربع امن = OK
- گودال = P
- تعفن = S
- ملاقات شده = V
- وامپوس = W



شماره ۷



شماره ۸

منطق:

- **منطق**، یک زبان رسمی برای بازنمایی دانش است بطوریکه بتوان روی آن، استدلال و نتیجه گیری کرد
- هر زبان بازنمایی دانش باید دو مورد شامل **نحو** (گرامر، ساختار یا Syntax) و **معنا** (Semantics) را تعیین کند. نحو، ساختار جملات زبان را تعیین می کند (چه کلمه بندی صحیح است یا اصطلاحا خوش فرم) و معانی زبان، درستی یک جمله را در هر دنیای ممکن تعریف می کند
- پایگاه های دانش شامل جملات هستند. این جملات بر اساس نحو زبان بازنمایی دانش نشان داد می شوند و این زبان همه جملاتی که خوش فرم هستند را نمایش می دهد
- **مثال**: طبق **گرامر** زبان ریاضی، عبارت $x + y = 4$ یک جمله خوش فرم است در حالی که $=+x2y$ خوش فرم نیست
- طبق **معانی زبان** در ریاضی، $x + y = 4$ در دنیایی $x = 2$ و $y = 2$ درست است و در دنیایی که $x = 1$ و $y = 1$ است، نادرست می باشد
- وقتی می گوییم m مدلی از α است، یعنی جمله ی α در مدل m درست است

استلزام:

استلزام، یک روش استدلال منطقی است و تعیین می کند جمله ای بطور منطقی از جمله دیگر بدست می آید. استلزام را با نماد $\models$ نشان می دهیم

- جمله a استلزام جمله b است
- جمله a جمله b را ایجاد می کند
- 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

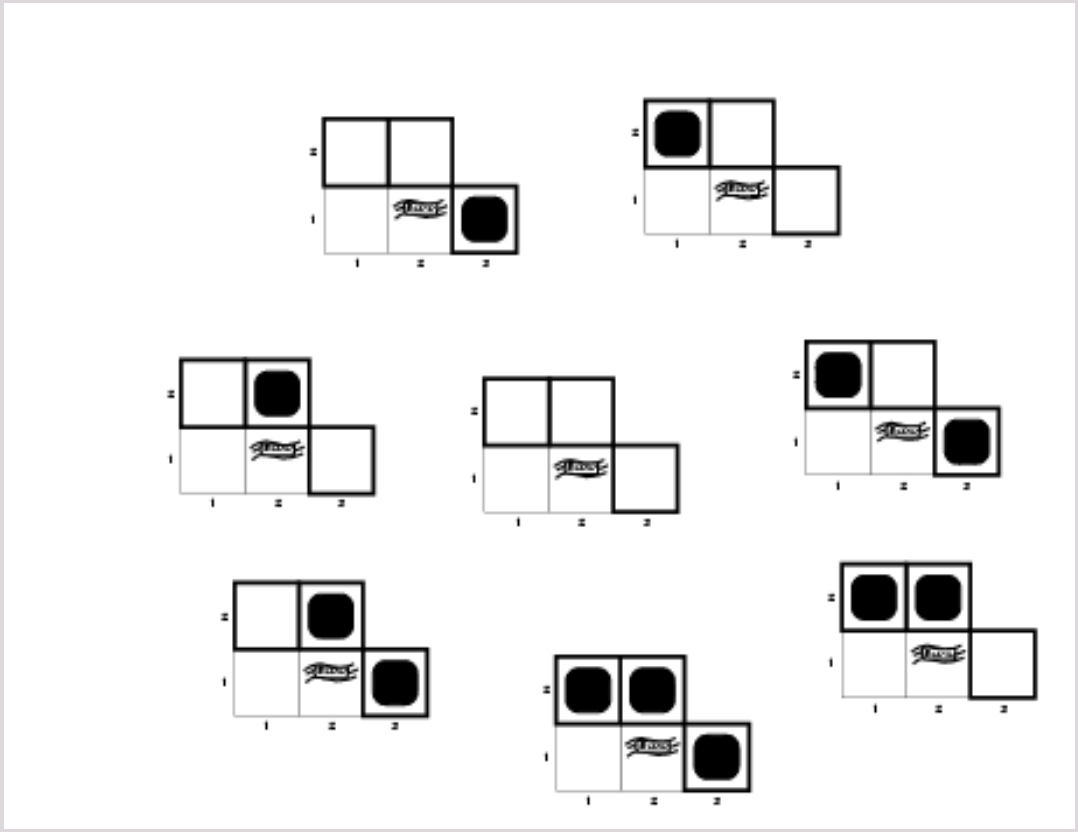
مثال: جمله $x + y = 4$ مستلزم جمله $x + y = ۴$ است

مدل های وامپوس:

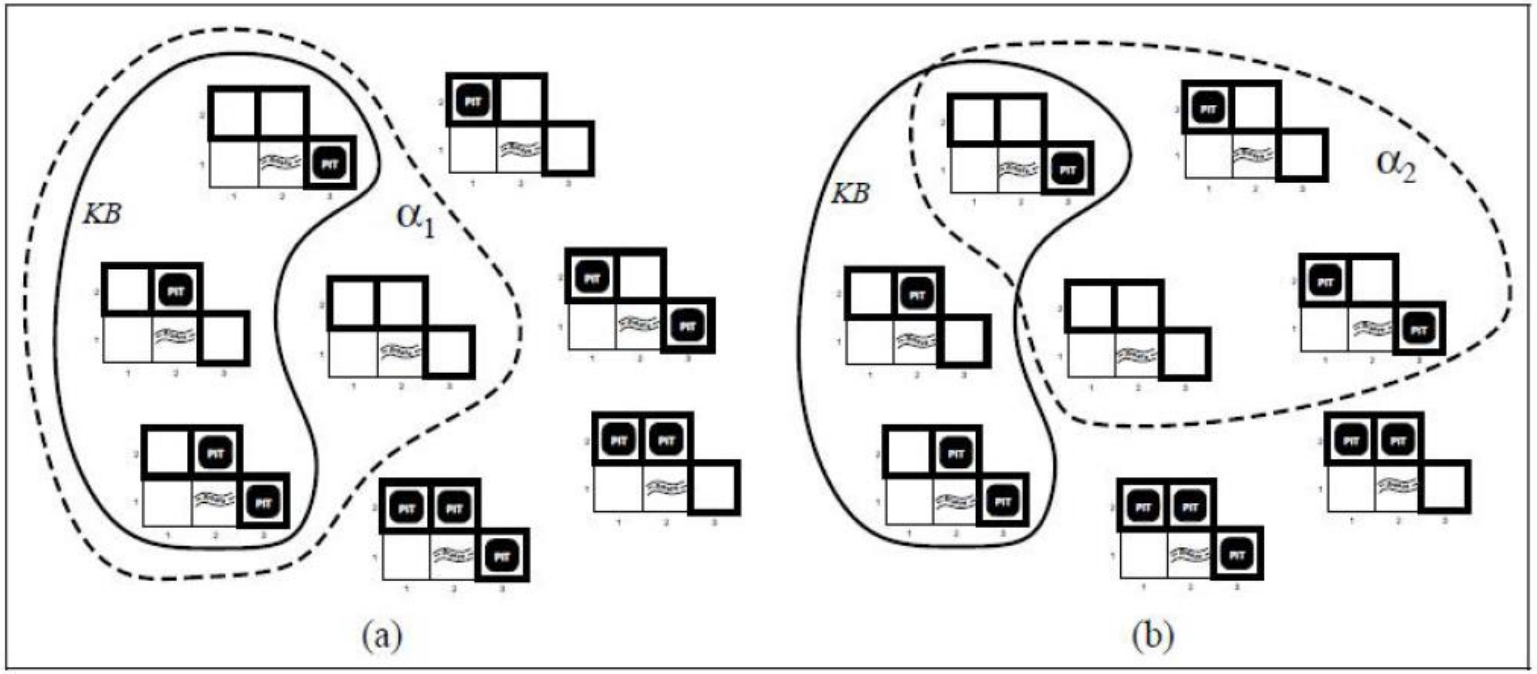
- عامل [۱,۱] را امن دید و در [۲,۱] نسیمی را احساس کرد. عامل مایل است بداند در مربع های همجوار گودالی وجود دارد یا خیر.
- سه مربع همجوار داریم که هر کدام می توانند گودال باشند یا نباشند، در این صورت:

۸ = ۳ ^ ۲ برابر است با تعداد گودال های ممکن

- پایگاه دانش در مدل هایی که متناقض با دانش عامل است، **نادرست** می باشد.

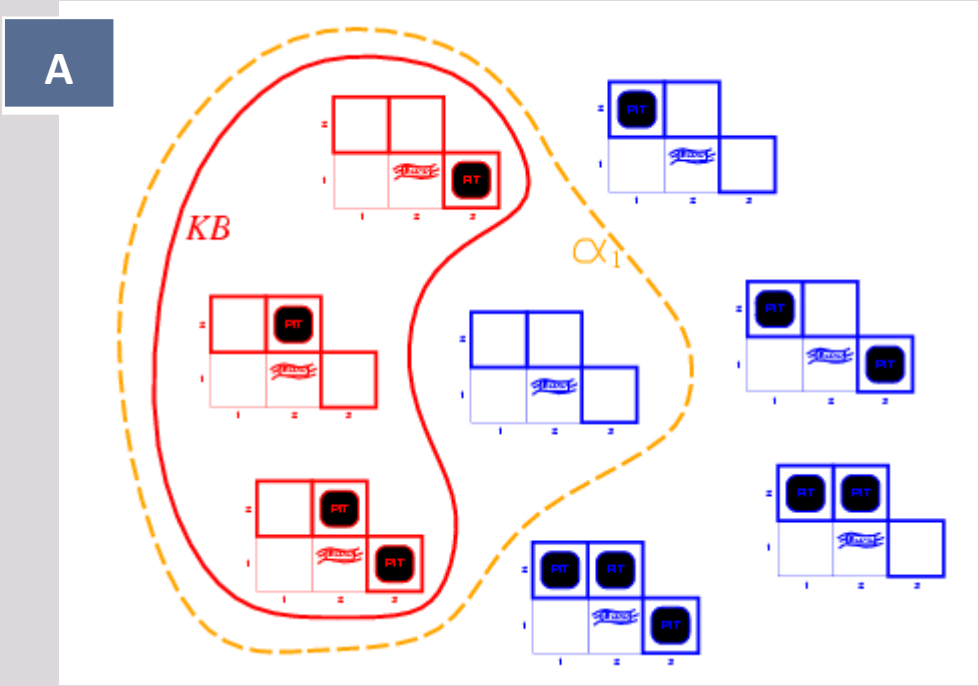


مدل های وامپوس (ادامه):



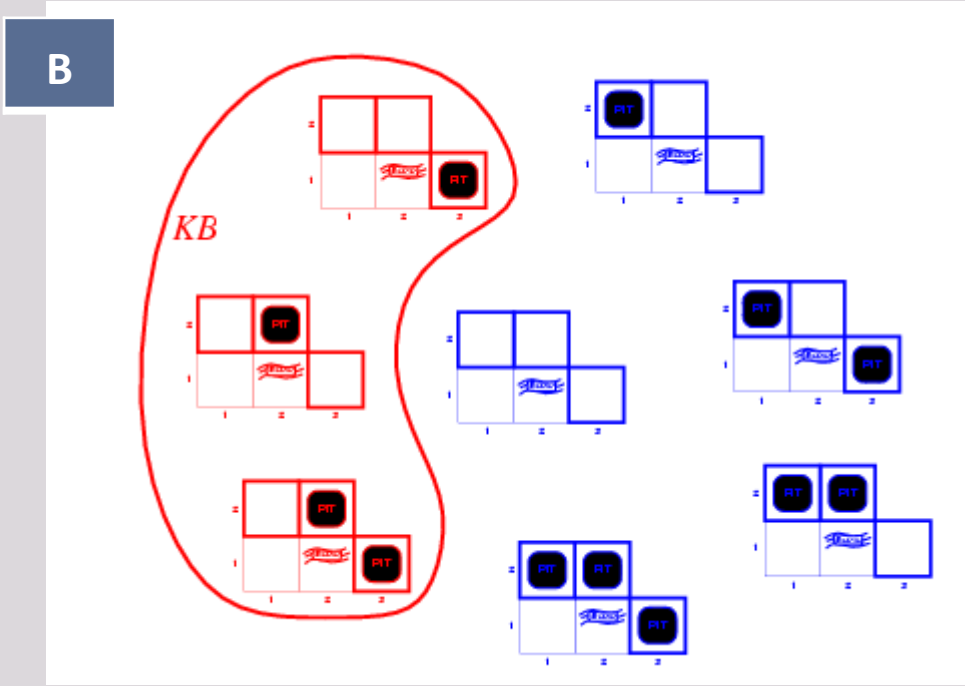
شکل ۷-۵: مدل‌های ممکن برای نمایش چاله‌ها در دنیای وامپوس

مدل های وامپوس (ادامه ۲):



❖ پایگاه دانش (KB) برابر است با:

قوانین دنیای وامپوس + مشاهدات



❖ α1 = [1,2] امن است:

KB ⊨ α1

منطق گزاره ای:

- منطق گزاره ای را **منطق بولی** نیز می نامند.
- نحو منطق گزاره ای، جملات مجاز را تعریف می کند. **جملات اتمی** (عناصر غیر قابل تجزیه از نظر نحوی)، از یک سمبل گزاره ای تشکیل شده اند.
- هر یک از این سمبل ها به گزاره ای اختصاص دارد که می تواند **درست** یا **نادرست** باشد. برای نام سمبل های گزاره ای از حروف بزرگ مثل R,Q,P استفاده می کنیم
- جملات پیچیده با استفاده از رابط های منطقی، از جملات ساده تر ساخته می شوند. True و False هر کدام یک جمله اند. به آن ها **ثابت های گزاره ای** می گوییم

رابط های منطقی:

- ¬ یا Not: جمله ای مثل $\neg W_{1,3}$ ، نقیض $W_{1,3}$ است. لیترال می تواند شامل یک جمله اتمی (لیترال مثبت) یا نقیض یک جمله اتمی (لیترال منفی) باشد
- ∧ یا Add: جمله ای که رابط اصلی آن ∧ است، مثل $W_{1,3} \wedge P_{1,3}$ ، ترکیب عطفی نام دارد. هر بخش از آن یک عطف نامیده می شود
- ∨ یا Or: جمله ای که از ∨ استفاده می کند، مثل $W_{2,2} \vee (W_{1,3} \wedge P_{3,1})$ ، ترکیب فصلی از عبارت های $W_{1,3} \wedge P_{3,1}$ و $W_{2,2}$ نامیده می شود
- ⇒ یا Implication: جمله ای مانند $(W_{1,3} \wedge P_{3,1}) \Rightarrow \neg W_{2,2}$ شرطی نامیده می شود، مقدم آن $(W_{1,3} \wedge P_{3,1})$ و تالی آن $\neg W_{2,2}$ است. جملات شرطی دستورات If-then نیز نامیده می شوند
- ⇔ یا Bi-Conditional: جمله $W_{1,3} \Leftrightarrow W_{2,2}$ دو شرطی نام دارد

مثال: عبارت "اگر و فقط اگر ۵ به ۲ بخش پذیر باشد آنگاه ۵ زوج است." یک ترکیب دو شرطی برای عبارت‌های "۵ به ۲ بخش‌پذیر است." و "۵ عدد زوج است." محسوب می‌شود

گرامر BNF جملات در منطق گزاره ای:

❖ شکل روبرو یک گرامر رسمی برای منطق گزاره ای نشان می دهد که BNF نام دارد. این گرامر در مورد پرانتز ها دقیق می باشد. هر جمله ای که با استفاده از رابطه دودویی ساخته می شود، باید در پرانتز قرار گیرد

❖ ترتیب تقدم در منطق گزاره ای از بیشترین تقدم به کمترین تقدم (از چپ به راست) به صورت زیر است:

$\neg , \wedge , \vee , \Rightarrow , \Leftrightarrow$

Sentence ⟶ Atomic Sentence | Complex Sentence

Atomic Sentence ⟶ True | false | symbol

Symbol ⟶ P | Q | R | …

Complex Sentence ⟶ ¬ Sentence

| (Sentence ∧ Sentence)

| (Sentence v Sentence)

| (Sentence => Sentence)

| (Sentence ⇔ Sentence)

عامل های منطقی (هم ارزی، اعتبار):

- **مفهوم هم ارزی منطقی**: دو جمله ی α و β از نظر منطقی هم ارز هستند. اگر در مجموعه یکسانی از مدل ها درست باشند. هم ارزی را بصورت $\alpha \equiv \beta$ می نویسیم.

- **مفهوم اعتبار**: یک جمله معتبر است اگر در تمام مدل ها درست باشد. جملات معتبر، **تاتولوژی** نیز نامیده می شوند.

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

جدول درستی پنج رابطه منطقی

عامل های منطقی (هم ارزی های منطقی استاندارد):

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

هم ارزی منطقی استاندارد

الگوریتم شمارش جدول درستی:

```
function TT-ENTAILS?(KB, α) returns true or false
  inputs: KB, the knowledge base, a sentence in propositional logic
          α, the query, a sentence in propositional logic

  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))
```

نکته: قوانین مربوط به هر رابطه را می توان با استفاده از جدول درستی بیان کرد

استنتاج بوسیله ی شمارش جدول درستی:

- فرض کنید در پایگاه دانش داریم:

$$p \wedge (\sim q \vee r) \wedge (p \Rightarrow q)$$

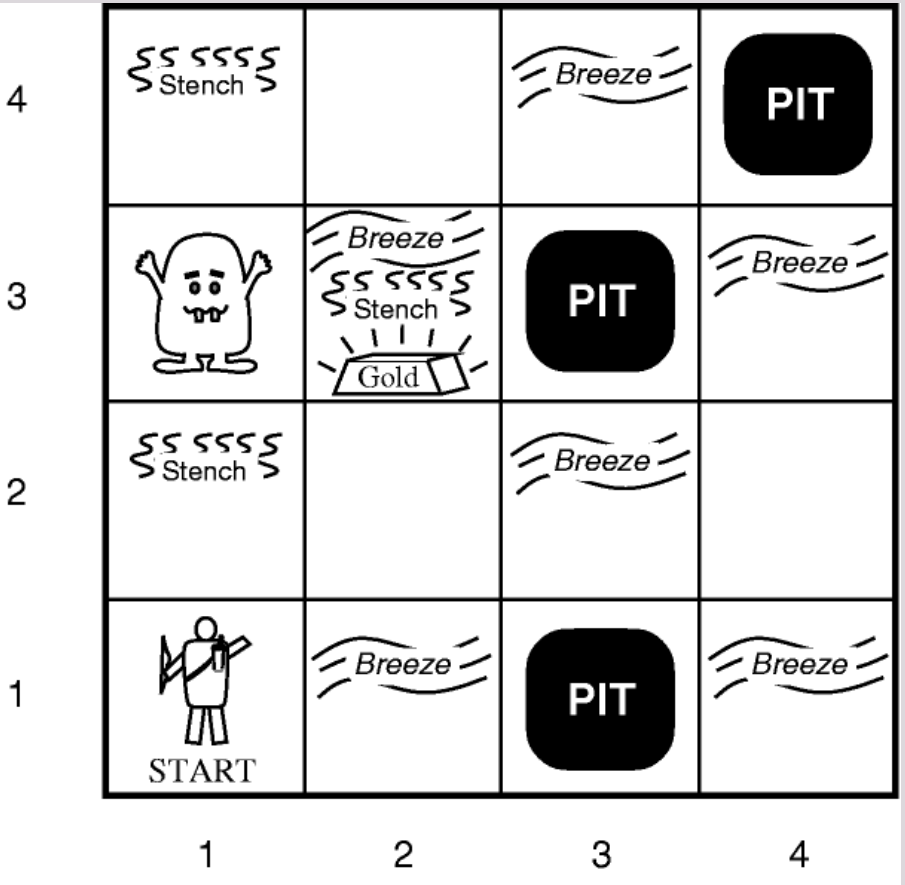
آیا از این پایگاه دانش می توان درستی r را نتیجه گرفت؟ ($KB \models r$)

- برای استنتاج $KB \models \alpha$، اگر KB و α حاوی n نماد باشند، آنگاه $2 \wedge n$ مدل وجود دارد

P	q	r	P	$\sim q \vee r$	$p \Rightarrow q$	r
0	0	0	0	1	1	0
0	0	1	0	1	1	1
0	1	0	0	0	1	0
0	1	1	0	1	1	1
1	0	0	1	1	0	0
1	0	1	1	1	0	1
1	1	0	1	0	1	0
1	1	1	1	1	1	1

با توجه به تصویر، $KB \models r$

منطق گزاره ای در دنیای وامپوس:



- معنی $B_{1,1}$ این است که در خانه [۱,۱] نسیمی وجود دارد:

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

- در [۱,۱] گودالی وجود ندارد:

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

- **توجه:** جمله شرطی یک طرفه در دنیای وامپوس درست است اما کامل نیست: $B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})$

این جمله در مدل هایی با $B_{1,1}$ نادرست و $P_{1,2}$ درست، صحیح می باشد اما در واقع در این مدل قوانین دنیای وامپوس نقض می شود

الگوهای استدلال در منطق گزاره ای:

- هدف استنتاج این است که برای جمله ای مثل α تصمیم بگیرد که آیا KB ⊨ α برقرار است یا نیست
- **قوانین استنتاج**: الگوهایی استاندارد که زنجیره ای از نتایج را برای رسیدن به هدف ایجاد می کند
- **قیاس استثنایی**: با استفاده از ترکیب عطفی، میتوان هر عطف را استنتاج کرد (یعنی هر وقت جمله ای به شکل $a \Rightarrow b$ داده شود، جمله b را می توان استنتاج کرد)
- می توان از (WumpusAhead ∧ WumpusAlive) و (WumpusAhead ∧ WumpusAlive) ⇒ Shoot) پرتاب (Shoot) را استنتاج کرد

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

الگوهای استدلال در منطق گزاره ای (ادامه):

- **قانون حذف عطف**: قانون استنتاج مفید دیگر، حذف عطف (حذف and) است که طبق آن، هر لیترال عطف را میتوان از ترکیب عطفی استنتاج کرد

$$\frac{\alpha \wedge \beta}{\alpha}$$

- می توان جمله WUMPUSAlive را از (WumpusAlive ∧ WumpusAhead) استنتاج کرد
- **خاصیت یکنواختی**: مجموعه ای از جملات استلزامی ، فقط می تواند در صورت اضافه شدن اطلاعات به پایگاه دانش رشد کند. برای جملات α و β داریم:

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

کاربرد قوانین استنتاج در جهان وامپوس:

اکنون می توانیم یک پایگاه دانش برای دنیای وامپوس بسازیم. با پایگاه دانش حاوی R1 تا R5 شروع کرده و چگونگی اثبات $\neg P_{1,2}$ (یعنی گودالی در [۱,۲] وجود ندارد) را بررسی می کنیم:

- در خانه [۱ , ۱] چاله ای وجود ندارد: $R_1: \neg P_{1,1}$
- در یک خانه نسیم دریافت می شود اگر و فقط اگر چاله ای در خانه همجوارش وجود داشته باشد. برای همه خانه ها باید این مطلب بیان شود. فعلا خانه های مرتبط را درنظر می گیریم: $R_2: B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$ و $R_3: B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- جملات قبلی در تمام دنیای وامپوس درست هستند. اکنون دریافت نسیم را برای دو خانه اول که ملاقات شد، در نظر میگیریم:

$$R_4: \neg B_{1,1} \quad \text{و} \quad R_5: B_{2,1}$$

- بنابراین، پایگاه دانش تا به اینجای کار شامل جملات R_1 تا R_5 است

کاربرد قوانین استنتاج در جهان وامپوس (ادامه):

- ابتدا حذف دو شرطی را در R_2 اعمال می کنیم: $R_6: (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
- سپس حذف عطف را در R_6 اعمال می کنیم: $R_7: ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
- با توجه به عکس نقیض داریم: $R_8: (\neg B_{1,1} \Rightarrow (\neg (P_{1,2} \vee P_{2,1}))$
- اکنون قانون مودس پوننس را به R_8 اعمال می کنیم و مشاهده R_4 را در نظر می گیریم:

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

- سرانجام، قانون دمورگان را اعمال کرده و به دست می آوریم: $R_{10}: \neg P_{1,2} \wedge \neg P_{2,1}$

نتیجه: پس از حذف عطف مشخص می شود که خانه [۲ , ۱] و خانه [۱ , ۲] فاقد چاله هستند

قانون رزولوشن (Resolution):

- در این بخش، یک قانون استنتاج یکتا به نام رزولوشن را بررسی می کنیم که در ترکیب با هر الگوریتم جستجوی کامل، منجر به الگوریتم استنتاج کامل می شود. در ادامه نسخه ساده ای از قانون رزولوشن را در دنیای وامپوس استفاده می کنیم:

- درنظر می گیریم عامل از [۱ , ۲] به [۱ , ۱] برمی گردد و سپس به [۲ , ۱] می رود که در آنجا، تعفن را احساس می کند ولی نسیمی احساس نمی کند. حقایق زیر را به پایگاه دانش اضافه می کنیم:

$$R_{11}: \neg B_{1,2} \text{ و } R_{12}: B_{1,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$$

با همان فرایندی که قبلا به R_{10} منجر شد، به عدم وجود چاله [۲ , ۲] و [۳ , ۱] می رسیم (می دانیم خانه [۱ , ۱] فاقد چاله است:

$$R_{13}: \neg P_{2,2} \text{ و } R_{14}: \neg P_{1,3}$$

قانون رزولوشن (ادامه):

- حذف دو شرطی را به R_3 اعمال می کنیم و با درنظر گرفتن R_5 و با استفاده از مودس پوننس به این حقیقت می رسیم که چاله ای در [۱ , ۱] یا [۲ , ۲] یا [۱ , ۳] وجود دارد: $R_{15}: P_{1,1} \vee P_{2,2} \vee P_{1,3}$
- اکنون به **اولین کاربرد قانون رزولوشن** می رسیم. در R_{13} لیترال $\neg P_{2,2}$ و در R_{15} لیترال $P_{2,2}$ وجود دارد، لذا داریم: $R_{16}: P_{1,1} \vee P_{1,3}$
- استنتاج فوق بدین معنی است که اگر چاله ای در یکی از خانه های [۱ , ۱] یا [۲ , ۲] یا [۱ , ۳] وجود داشته باشد و مطمئن باشیم که در خانه [۲ , ۲] چاله ای نیست آنگاه چاله در خانه [۱ , ۱] یا [۱ , ۳] است
- **رزولوشن دوم**، یعنی به طور مشابه در R_1 لیترال $\neg P_{1,1}$ و در R_{16} لیترال $P_{1,1}$ وجود دارد لذا داریم: $R_{17}: P_{1,3}$
- **به طور کل**، اگر چاله ای در [۱ , ۱] یا [۱ , ۳] نباشد یعنی چاله در خانه [۱ , ۳] است و این دو مرحله از استنتاج، مثال هایی از قانون استنتاج رزولوشن واحد است

قانون رزولوشن (ادامه ۲):

- قانون رزولوشن واحد، یک عبارت و یک لیترال را گرفته، عبارت دیگری تولید می کند:

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

- قانون رزولوشن واحد می تواند به قانون رزولوشن کامل، تعمیم داده شود:

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

نکته: قانون رزولوشن فقط به لیترال های ترکیب فصلی اعمال می شود لذا برای به کارگیری آن فقط به پایگاه های دانش و پرسش های شامل این ترکیبات فصلی نیاز است

شبه کد الگوریتم رزولوشن:

```
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 ∧ ¬α
  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
```

الگوریتم رزولوشن

الگوریتم رزولوشن:

❖ شکل نرمال عطفی یا (CNF): جمله ای که بصورت ترکیب عطفی از ترکیبات فصلی لیترال ها بیان می شود. 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 رزولوشن: رویه های استنتاج مبتنی بر رزولوشن، براساس اصل اثبات برهان خلف عمل می کنند. یعنی برای اینکه نشان دهیم $KB \models \alpha$ باید مشخص کنیم جمله $KB \wedge \neg \alpha$ ارضا نشدنی است و برای این منظور از برهان خلف استفاده می کنیم.

الگوریتم رزولوشن (ادامه):

❖مراحل الگوریتم رزولوشن به شرح زیر است:

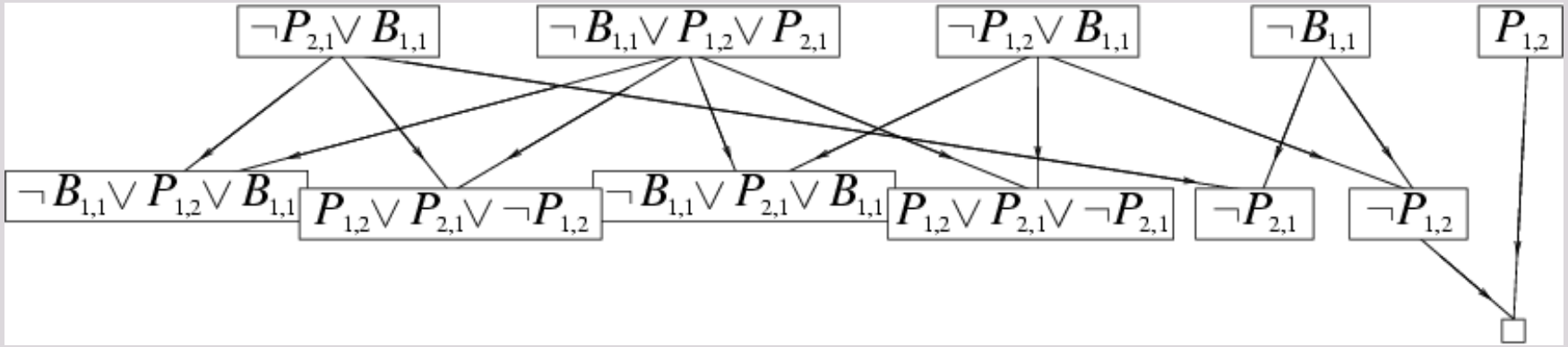
1. ابتدا $KB \wedge \neg \alpha$ را به CNF تبدیل می کنیم
2. سپس قانون رزولوشن به عبارات کوچک حاصل اعمال می شود
3. هر جفتی که شامل لیترال های مکمل باشد، رزولوشن می شود تا عبارت جدیدی ایجاد گردد
4. اگر این عبارت قبلا در مجموعه نباشد، به آن اضافه می شود
5. فرایند ادامه می یابد تا یکی از دو مورد زیر اتفاق بیافتد

شرط اول، آنست که هیچ فراکرد دیگری وجود نداشته باشد که بتواند اضافه شود. در این مورد KB جمله α را نتیجه نمی دهد

شرط دوم، آنست که کاربرد قانون رزولوشن، فراکرد تهی ایجاد کند که در این مورد KB جمله α را نتیجه می دهد

نکته: رویه رزولوشن را می توان به استنتاج خیلی ساده ای در دنیای وامپوس اعمال کرد

الگوریتم رزولوشن در دنیای وامپوس:



الگورتیم رزولوشن برای دنیای وامپوس

- پایگاه دانش عامل به صورت زیر است:

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

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

پایان فصل هفتم