

بسمه تعالی

هوش مصنوعی

مسائل ارضا محدودیتها - ۱

نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

منتظر با هر وزیر ما یک متغیر درنظر میگیریم
 مقادیری که متغیرها میتوانند به خودشان بگیرند
 شماره ردیفی است که اون وزیر توی ستون
 متناظرش قرار گرفته
 ازمون هدف: وزیرها هم را تهدید نکند

مقداره

- در یک م.ا.م. حالات بوسیله مقادیر مجموعه ای از متغیرها تعریف می شوند و آزمون هدف مجموعه ای از محدودیتهایی است که متغیرها باید ارضاء کنند.

مثال:

- وزیر^۸
- متغیرها X_1, X_2, X_3, \dots که هر یک از دامنه ای D_1, D_2, D_3, \dots انتخاب می شوند و مجموعه ای از محدودیتها C_1, C_2, C_3, \dots را باید ارضاء کنند.

متغیرهایی را نشان
 میدهد که ما بایدیک
 محدودیتی بین اونها
 درنظر بگیریم.

- مجموعه X : متغیرها
- مجموعه D : دامنه ها
- مجموعه C : محدودیتها
- هر محدودیت C_i بصورت $\langle \text{scope}, \text{rel} \rangle$

یعنی ما زمانی یک حل داریم که همه متغیرهای مسئله یک مقداری گرفته باشند و این مقادیر همه‌ی محدودیت‌های مسئله را ارضاء کنند.

- **یک حالت مسئله بوسیله انتساب** مقادیر به همه یا برخی از متغیرها تعریف می‌شود.
- یک انتساب که هیچ یک از محدودیتها را نمی‌شکند یک **انتساب سازگار** یا قانونی نامیده می‌شود.
- **انتساب کامل** وقتی که همه متغیرها مقدار گرفته اند.
- یک حل یک انتساب کامل و سازگار است.
- **انتساب جزئی** هنگامی فقط برخی از متغیرها مقدار گرفته اند.

مثال – رنگ آمیزی نقشه

دو تا استان همسایه نباید همرنگ باشند.
پس محدودیت مسئلمون اینه



متغیرها: {WA, NT, SA, QL, NSW, V, T}

■

4

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

هر استان را با سه
رنگ میتوانیم رنگ
کنیم

■ دامنه ها: {قرمز، سبز، آبی}

■ محدودیتها: هیچ دو ایالت مجاوری هم رنگ نباشند.

$$C = \{SA \neq WA, SA \neq NT, SA \neq Q, SA \neq NSW, SA \neq V, \\ WA \neq NT, NT \neq Q, Q \neq NSW, NSW \neq V\}.$$

■ مثلاً $WA \neq NT$ کوتاه شده $\langle (WA, NT), WA \neq NT \rangle$

عضو اول هر زوج
ایالتی که همسایه
هستند

■ در مجموعه $\{(قرمز, سبز), (سبز, قرمز), \dots\}$ (WA, NT)

مازیار پالهنجک

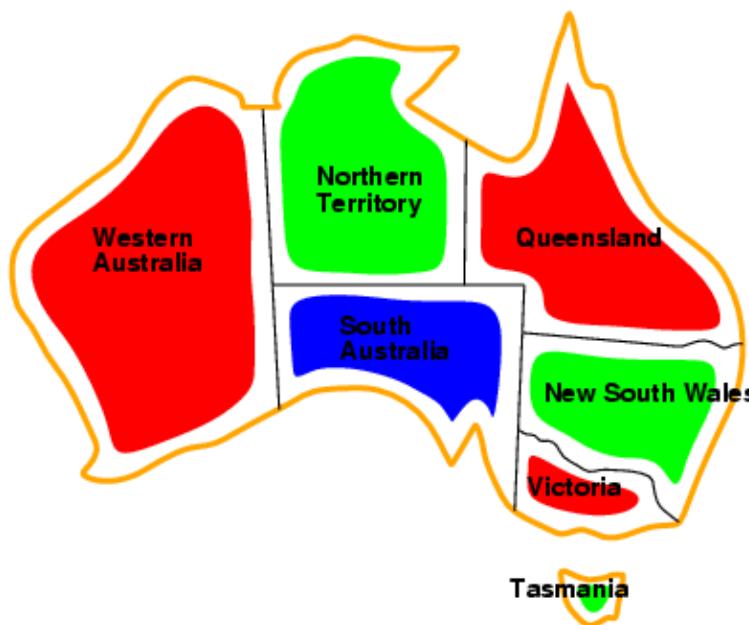
نمایش محدودیت ها

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

لیست جفت رنگ های
مجاز برای این دو
متغیر

5

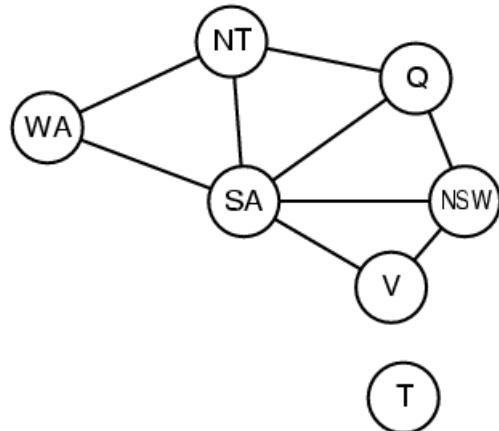
یک حل



■ می توان یک م.ا.م. را بوسیله یک گراف محدودیت به تصویر کشید.

■ رئوس: متغیرها، یالها: محدودیتها

بین همه‌ی ایالت‌های مجاور ما یال داریم.
چون محدودیت ما بین ایالت‌های مجاور تعریف شده است.



تدوین افزایشی یعنی به متغیرها یکی یکی مقدار بدیم مثل وقتی که وزیرها را یکی یکی روی تخته بگذاریم

یک تدوین کامل هم هست مثل مسئله‌ی هشت وزیر اگه همه‌ی وزیرها را روی تخته بگذاریم و بعداً بخاهمیم جابه‌جاشون کنیم این میشه تدوین کامل

جستجوی عمومی مثل روشی که در فصل عامل مسئله حل کن داشتیم میتوانیم مسئله‌ی ارضای محدودیت را هم همان طوری حل کنیم.

■ یک م.ا.م. را می‌توان بصورت یک مسئله جستجوی عمومی با

اجزای مسئله

تدوین افزایشی بیان نمود:

حالت اولیه: انتساب تهی

تابع تالی: انتساب مقداری به یک متغیر بی مقدار به شرطی که با متغیرهای مقدار گرفته برخورد نداشته باشد.

آزمون هدف: انتساب فعلی کامل و سازگار باشد.

هزینه مسیر: ثابت (مثلاً ۱) برای هر مرحله

از یک حالت اولیه شروع میکنیم که انتساب تهی است
بعد یکی یکی راس ها را بسط میدیم در پست دادن متغیری انتخاب میشود که هنوز مقدار نگرفته برای اون متغیر از دامنه ش یه مقداری را انتخاب میکنیم و بهش انتساب میدیم دقت کنیم که مقداری که به متغیر میدیم باید با محدودیت هایی که داریم سازگار باشه

- هر حل باید یک انتساب کامل باشد بنابر این در عمق n ظاهر خواهد شد اگر n متغیر وجود داشته باشد.

- چون عمق به n محدود است، بصورت امن می توان جستجوی عمق نخست را استفاده کرد.

- چون مسیر حل مهم نیست تدوین حالت کامل رامی توان استفاده کرد.

- هر حالت یک انتساب کامل که سازگار هست یا نیست.
- جستجوهای محلی برای این روش مناسب است.

حل ما یک انتساب کامل و سازگار است یعنی همین که متغیرها همشون مقدار گرفته باشند و محدودیت ها ارضاء شده باشد کافی است.

مازیار پالهنه

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

9

تلاش کنیم یه حالت
بهتر پیدا کنیم

الزاماً سازگار نیست

9

تنوع متغیرها

n
تا متغیرهم داریم

متغیرها گستته

دامنه محدود (رنگ آمیزی نقشه - هشت وزیر)
 $O(d^n)$

هر متغیری فقط دو تا
مقدار بتواند بگیره

حالت خاص: م.ا.م. بولی
دامنه نامحدود (زمانبندی کارها)

مثالاً مجموعه اعداد صحیح

متغیرها زمان شروع/پایان هر کار

نمی توان همه ترکیبات مجاز را فهرست کرد. به یک زبان محدودیت نیاز است. بطور
مثال: $StartJob_1 + 5 \leq StartJob_3$

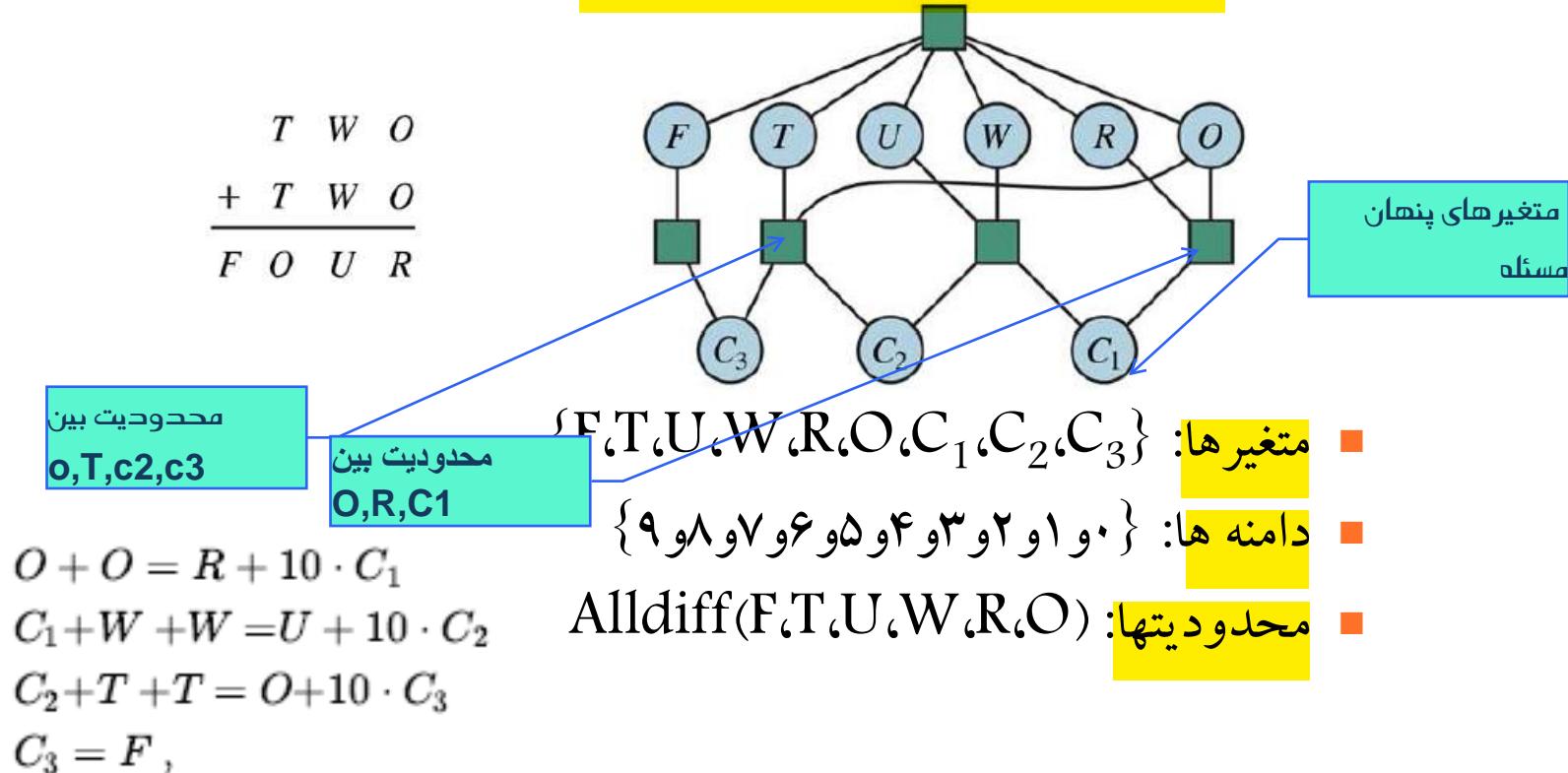
متغیرها پیوسته

زمانهای شروع/پایان رصد کردن توسط تلسکوپ هابل

تنوع محدودیتها

- **یکتائی** – مثلاً مردم یک استان از رنگ خاصی بدانند می‌آید:
 $NT!=\text{green}$
- با یک پیش‌پردازش می‌توان این مقدار را از دامنه متغیر متناظر حذف نمود.
- **دوتائی** – بین دو متغیر $WA!=NT$
- **بیشتر** – همانند معماهای ریاضی
- محدودیتی شامل تعدادی دلخواه متغیر، **محدودیت جهانی** نامیده می‌شود.

مثال محدودیت چند تائی - معماری ریاضی



- ابرگراف محدودیت برای محدودیتهای چند تائی (همانند شکل قبل)

- محدودیت مطلق: شکستن آن یک حل بالقوه را از بین می برد.

بهتر است که این محدودیت برقرار باشه ولی میتوانه هم برقرار نباشه مثلاً بگیم من ترجیح میدم یه کاری را بعداز ظهر

- محدودیت ترجیحی: بهتر است اینگونه باشد

- مثلاً در زمان بندی
- محدودیتهای ترجیحی را معمولاً می توان با افزودن هزینه به انتساب متغیرها حل نمود.

مازیار پالهنج

نوعی - نیمسال اول ۱۴۰۱-۰۲

مثلاً یه کاری را برای صبح اگه انجام بدیم
هزینش را یک بذاریم
اگه عصر انجام بدیم ۳
دنبال حل با هزینه کمتر هستیم

3

اندازه‌ی درخت خیلی
بزرگ میشه اگه
اینطوری جستجو را
انجام بدیم

جستجوی عقب‌گرد

n
تا متغیر داریم هر
کدامشون هم توی
دامنشون
d
تا مقدار دارن

به عمق ۲ ک برسیم
یکی از متغیرها
مقدار گرفته پس
(n-1)
تا متغیر داریم که
دامنشون
d
تا مقدار داره

- فرض کنید از عرض نخست استفاده کنیم.
- ضریب انشعاب در عمق ۱ برابر nd است.
- ضریب انشعاب در عمق ۲ برابر $(n-1)d$ است.
- در انتهای دارای $n!d^n$ انتساب کامل داریم.

$\{WA=green, NT=red\}$ با انتساب متغیرها جابجائی است یعنی $\{NT=red, WA=green\}$ یکسان است.

- بنابر این در هر مرحله فقط یک متغیر را مقدار می دهیم.
- حال d^n برگ خواهیم داشت.

- جستجوی عمق نخستی که هر بار فقط یک متغیر را مقدار می دهد، جستجوی عقبگرد نامیده می شود.
- جستجوی بنیادی م.ا.م. بصورت ناآگاهانه
- مسئله $n=25$ وزیر تا

تا ۲۵ وزیر هم
میتوانیم با این روش
حل کنیم و مشکل
حافظه پیدا نمیکنیم

انتساب اولیه تهی است

که برای ارضی محدودیت ها استفاده میشود

جستجوی عقبگرد

```
function BACKTRACKING-SEARCH(csp) returns a solution or failure
    return BACKTRACK(csp, {})
```

```
function BACKTRACK(csp, assignment) returns a solution or failure
    if assignment is complete then return assignment
    var  $\leftarrow$  SELECT-UNASSIGNED-VARIABLE(csp, assignment)
    for each value in ORDER-DOMAIN-VALUES(csp, var, assignment) do
        if value is consistent with assignment then
            add {var = value} to assignment
```

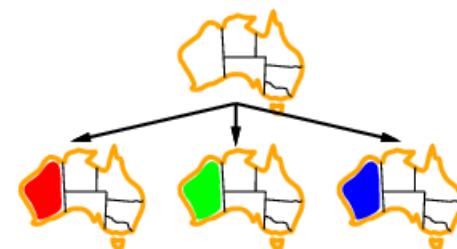
```
        result  $\leftarrow$  BACKTRACK(csp, assignment)
        if result  $\neq$  failure then return result
```

```
    remove {var = value} from assignment
    return failure
```

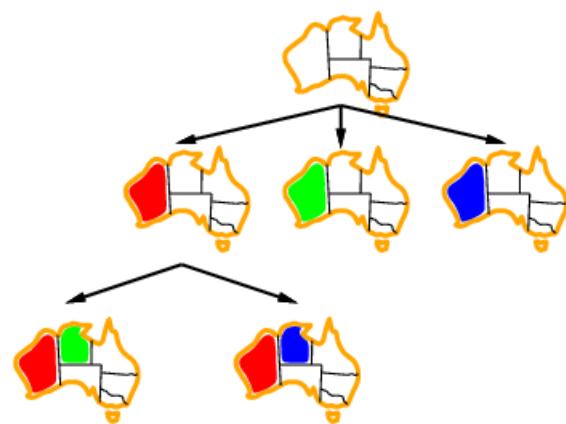
مثال جستجوی عقبگرد



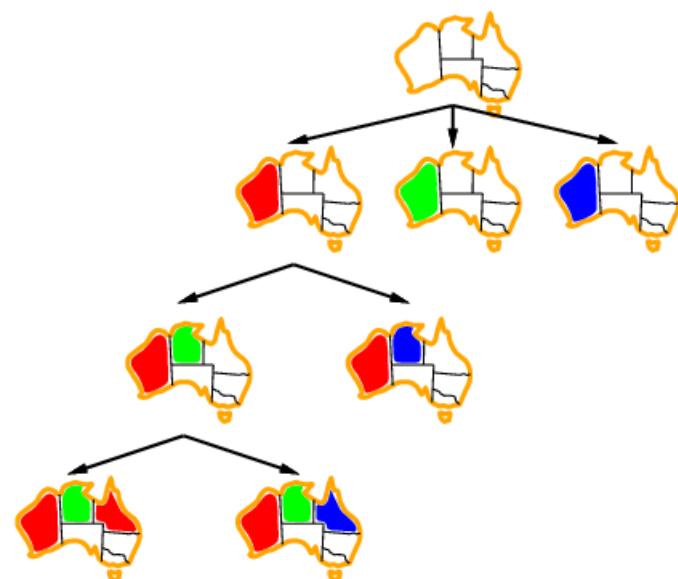
مثال جستجوی عقبگرد



مثال جستجوی عقبگرد



مثال جستجوی عقبگرد



بهود کارآئی جستجوی عقبگرد

- چه متغیری باید بعداً انتساب داده شود؟
- به چه ترتیبی مقادیر آن باید آزموده شوند؟
- آیا می توانیم شکستهای اجتناب ناپذیر را زودتر متوجه شویم؟

متغیر محدود شده بیشینه most constrained variable

نتخاب کن که کمترین مقادیر قانونی را دارد.

انتساب SA قبل از Q



- مکافه کمترین مقادیر باقیمانده (minimum remaining value): یک شکست اول

- اگر متغیر هیچ مقادیر باقیمانده ای نداشته باشد زودتر انتخاب

شده، و زودتر شکست می خوریم.

دوتا ایالتی که کنار قرمزه هستند میتوانند
یکی از دو رنگ سبز و آبی را داشته باشند
ولی بقیه ایالت ها میتوونند یکی از سه رنگ
را داشته باشند

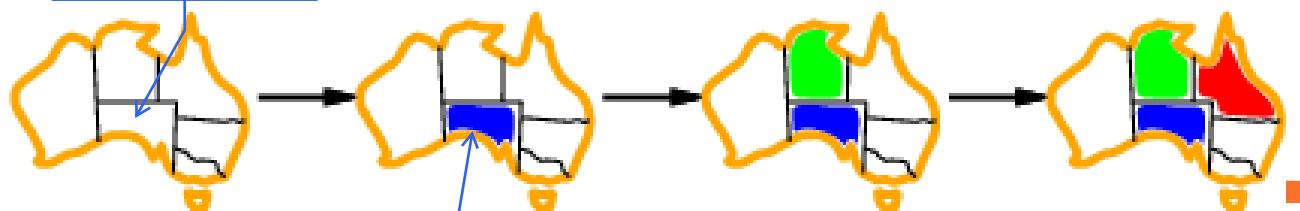
پس دوتا همسایه کمترین مقادیر باقیمانده را
دارند پس اولویت دارند در انتخاب برای
رنگ امیزی

هوش مصنوعی - نیمسال اول ۱۴۰۲

متغیر محدود کن بیشینه most constraining variable

- کدام متغیر **ابتدا** انتخاب شود؟
- متغیری را ابتدا انتخاب کن که کمترین مقدار را برای سایر متغیرها باقی می گذارد. (مکافه درجه)
- ضریب انشعاب را برای گزینه های آتی می کاهد

اولین راسی که
انتخاب میشه
بیشترین درجه را
داره



- قابل استفاده هنگامی که چند متغیر محدود شده بیشین قابل انتخاب هستند (و می خواهیم بین آنها انتخاب کنیم).

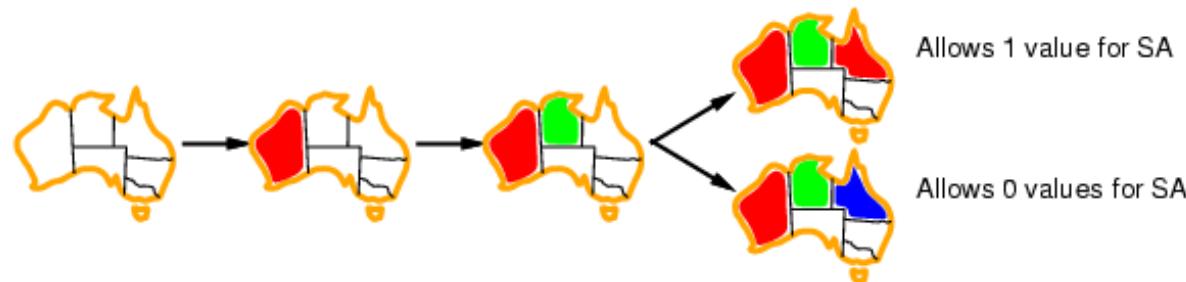
اینکه ما نمیتوانیم ابی را برای ایالت های دیگه انتخاب کنیم یعنی ضریب انشعاب کم شده چون برای تعداد بیشتری از متغیرها نمیتوانیم مقدار ابی را انتخاب کنیم

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

مقدار محدود کن کمینه

least constraining value

- هنگامی که متغیر انتخاب شد، ترتیب انتخاب مقادیر متغیر مهم است.
- متغیری که کمترین مقادیر را از متغیرهای باقیمانده حذف می کند.



- ترکیب مکافهه های گفته شده ۱۰۰۰ وزیر را نیز امکان پذیر می سازد.

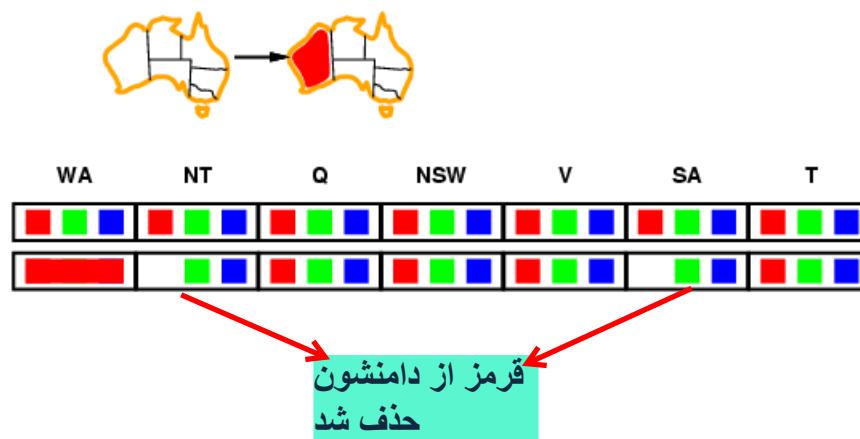
چک جلو

- دنبال کردن مقادیر متغیرهای باقیمانده انتساب نشده
- خاتمه جستجو هنگامی که متغیری هیچ مقدار قانونی نداشته باشد.



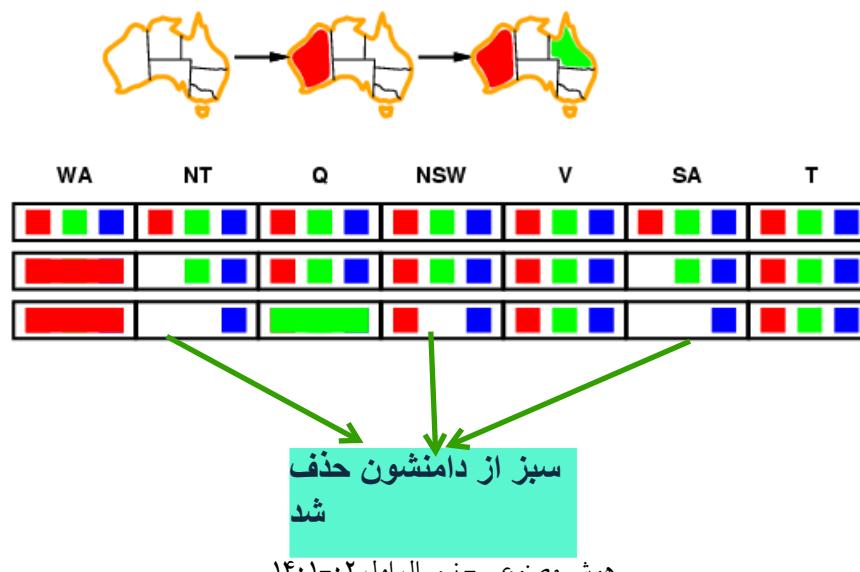
WA	NT	Q	NSW	V	SA	T
■ Red	■ Green	■ Blue	■ Red	■ Green	■ Blue	■ Red

چک جلو



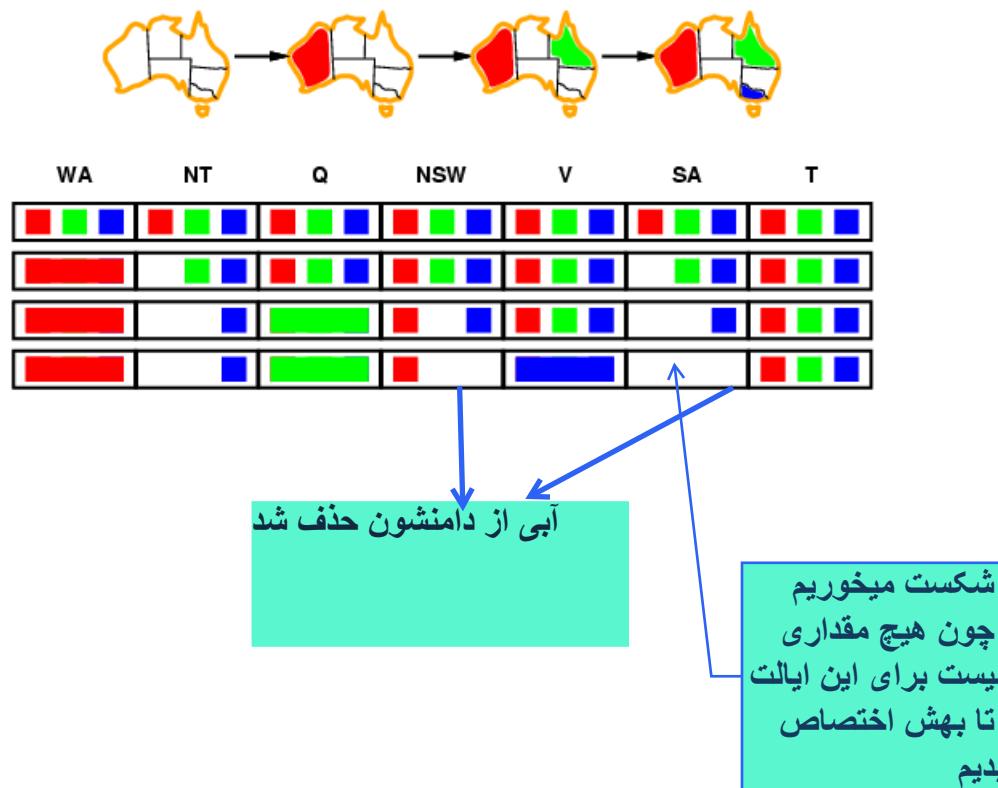
چک جلو

-
-



چک جلو

-
-



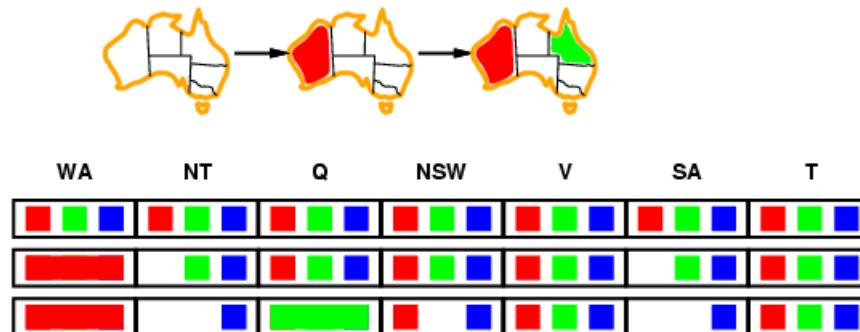
مازیار پالهنجک

28

28

انتشار محدودیت

چک جلو اطلاعات را از متغیرهای انتساب شده به نشده انتقال می دهد ولی اجازه تشخیص همه ناسازگاریهای زود هنگام را نمی دهد.



NT و SA نمی توانند هر دو آبی باشند.

مازیار پالهنجک

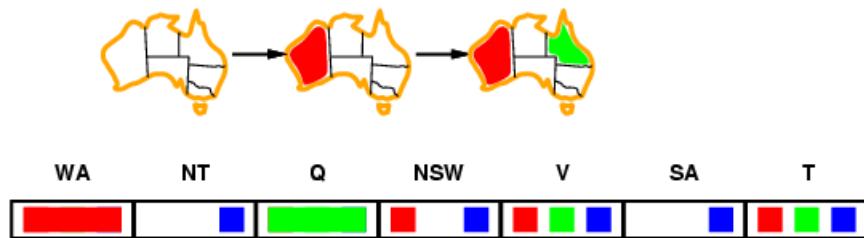
هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

29

سازگاری کمان

- کمان منظور یالی است در گراف محدودیت
- کمان $X \rightarrow Y$ سازگار گفته می شود اگر برای هر مقدار X در دامنه X مقدار y در دامنه Y وجود داشته باشد که با آن سازگار باشد.

■ مثال کمان سازگار



سازگاری کمان بین

SA و

NSW

برقرار است

چون در دامنه‌ی اس ای فقط ابی هست و در دامنه‌ی

NSW

علاوه بر ابی قرمز هم هست

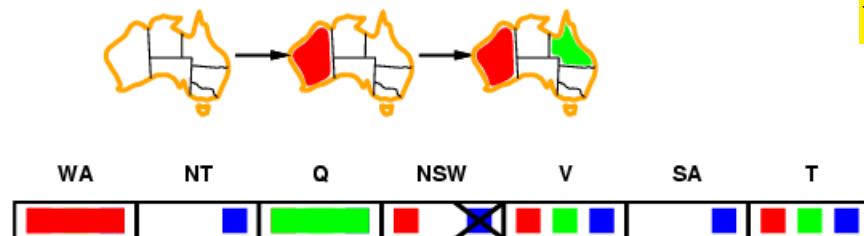
هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

30

سازگاری کمان

- کمان منظور یالی است در گراف محدودیت
- کمان $X \rightarrow Y$ سازگار گفته می شود اگر برای هر مقدار X در دامنه X مقدار y در دامنه Y وجود داشته باشد که با آن سازگار باشد.

■ مثال کمان ناسازگار



مازیار پالهنجک

اگه ابی انتخاب بشه در سمت چپ
در سمت راست دیگه مقداری نداریم که
بتوانیم انتخاب کنیم پس ناسازگار میشه

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

31

امام علی (ع) :

وَالْفُرْصَةُ تَمُرُّ مِنَ السَّحَابِ، فَانْتَهِزُوا فُرْصَ الْخَيْرِ

فرصت به مانند ابر سگدرا می گذرد، پس فرصت های نیک را غنیمت دانید

والسلام

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی

مسائل ارضا محدودیتها - ۲

نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

توی این مسایل باید مقادیر مناسبی که به متغیرها نسبت داده میشود را پیدا کنیم

اگه مقادیری که متغیرها نسبت میدیم با توجه به محدودیت هاشون باشه و محدودیت هاروارضا کنن میگیم انتساب سازگاره

اگه همه ی متغیرهای مسئله مقدار گرفته باشند انتساب کامل است ولی اگه تعدادی گرفته باشند و تعدادی نه میگیم جزئی

حل مسئله یک انتساب کامل و سازگار است در مثال رنگ آمیزی نقشه استرالیا متغیرهای ما ایالت های ما بود و هدف ما این بود که به هر ایالتی یک رنگ اختصاص بدم طوری که رنگ هیچ دو ایالت مجاوری یکسان نشه

دامنه ی متغیرها سه رنگ قرمز و سبز و آبی مسائل اراضی محدودیت را میشه بایک گراف اراضی محدودیت نمایش داد

در این گراف رئوس متناظر با متغیرهای ما هستند و یال ها متناظر با محدودیت هایی که بین دو یا چند متغیر وجود داره

برای پیدا کردن حل میشه از مسئله ی جستجو استفاده کرد فقط باید مسئله را به درستی تدوین کنیم حالات اولیه و اعمال و هدف ما و هزینه وتابع تالی چی هست

تنوع متغیرها: گستره و پیوسته محدودیت ها: یکتایی و دوتایی و محدودیت جهانی یک جستجوی مناسب برای پیدا کردن حل در مسائل اراضی محدودیت جستجوی عقبگرد است این جستجو به صورت عمقی عمل میکنه

یادآوری

- متغیرها، دامنه ها، محدودیتها
- انتساب سازگار
- انتساب کامل، جزئی
- مثال رنگ آمیزی نقشه
- گراف محدودیت
- حل بصورت یک مسئله جستجو
- تنوع متغیرها
- تنوع محدودیتها
- جستجوی عقبگرد
- متغیر محدود شده بیشینه
- متغیر محدود کن بیشینه
- مقدار محدود کن کمینه
- چک جلو
- سازگاری کمان

هر بار یکی از تالی ها را بسط میده ولی چون اینجا تعداد متغیر هامون ثابت است و در هر مرحله در جستجو فقط به یکی از متغیرها مقدار میدیم پس جستجوی ما تا عمق

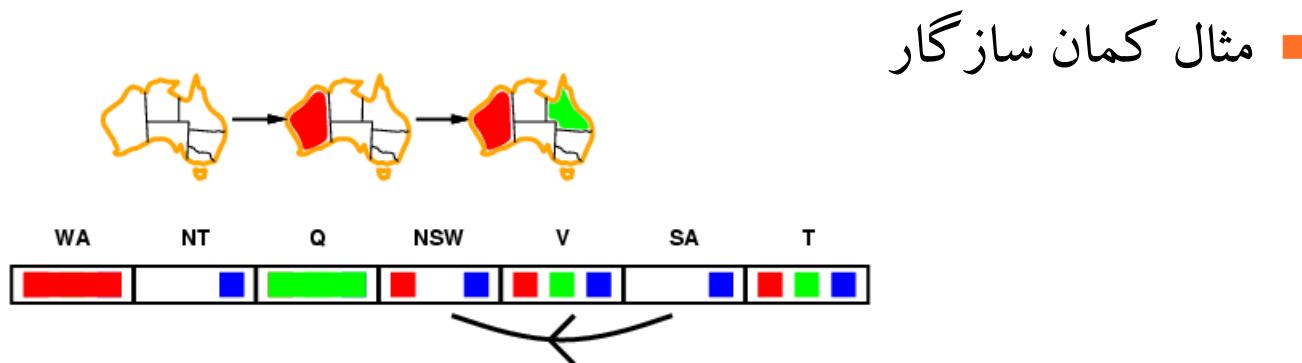
n

میره پس نگران نیسیم که توی حلقه ی بی نهایت قرار بگیره و تاعمق نامحدود بره و اینکه جستجوی عمق نخست از نظر حافظه مناسب است این روش جستجو میتوانه برای اراضی محدودیت ها مناسب باشه

در حالت اولیه از راس ریشه شروع میکنیم که هیچ متغیری مقداری نداره بعدش متغیری که مقدار نگرفته بش مقدار مناسب میدیم برای ساده شدن کار در هر مرحله فقط یک متغیری که مقدار نداره را انتخاب میکنیم هر بار که به یک متغیر میخواهیم مقدار بدیم چک میکنیم با متغیرهای قبلی که مقدار گرفتن سازگار است یا نه؟ اگر ناسازگار بود عقبگرد میکنیم و یه مقدار دیگه ای را برای اون متغیر چک میکنیم

سازگاری کمان

- کمان منظور یالی است در گراف محدودیت
- کمان $X \rightarrow Y$ سازگار گفته می شود اگر برای هر مقدار X در دامنه X مقدار y در دامنه Y وجود داشته باشد که با آن سازگار باشد.



جستجوی عقبگرد را میتوانیم بهینه تر پیاده سازی کنیم یعنی بگیم کدام یکی از متغیرها را اگه اول انتخاب کنیم بهتر است؟ در این حالت از یک مکافه استفاده کردیم ب اسم متغیر محدودشده بیشینه که میگه متغیری را زودتر انتخاب کن که دامنه ش مقادیری کمتری داره تا بتونیم زودتر متوجه شکست احتمالی بشیم

تا اگه قراره شکست بخوریم زودتر متوجه شیم

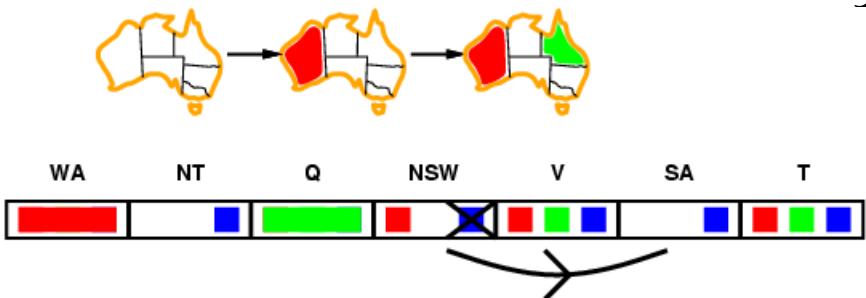
اگه چندتا متغیر محدودشده بیشینه داشته باشیم از بین اون متغیرها بهتر است متغیر محدودکن بیشینه را انتخاب کنیم یعنی اون متغیری را انتخاب کنیم که تعداد متغیرهایی که به واسطه ای مقدار دهی به این متغیر نمیتونن مقدار بگیرن بیشتر باشه یعنی درجه یا تعدادیال هایی که بهش وصل هستند بیشینه باشه

به این مکافه ای درجه هم میگیم که کمک میکنه فضای جستجو کاهش پیدا کنه و زودتر به حل برسیم

بعد از اینکه متغیر پیدا شد ما از مکافه ای محدودکن کمینه استفاده میکنیم یعنی از بین مقادیری که میتوانیم انتخاب کنیم اونی را انتخاب کنیم که کمتر برای متغیرهای دیگه محدودیت ایجاد کنه یعنی به متغیرهای اینده اجازه بدیم مقادیر مناسب بگیرن

- کمان $X \rightarrow Y$ سازگار گفته می شود اگر برای هر مقدار X در دامنه X مقدار y در دامنه Y وجود داشته باشد که با آن سازگار باشد.

■ مثال کمان ناسازگار

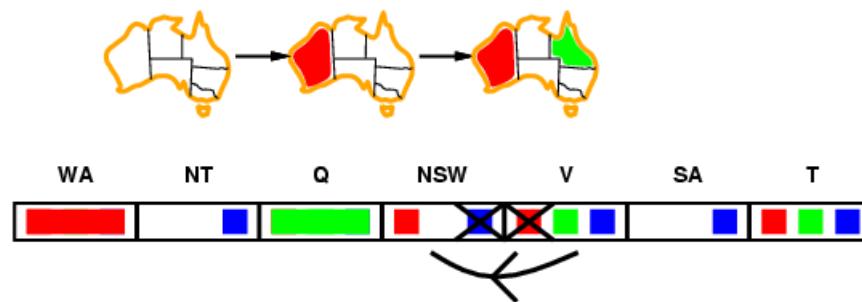


مکاشفه‌ی چک به جلو می‌گفت هر موقع به یک متغیر مقدار نسبت میدیم میرفتیم مقدار نسبت داده شده را از دامنه‌ی همسایه‌های اون متغیر حذف میکردیم که باعث میشد دامنه کوچک‌تر شه و درنهایت سریعتر جستجو به جواب برسه مشکل چک به جلو: تشخیص ندادن یک سری ناسازگاری برای رفع این مشکل از سازگاری کمان استفاده میکنیم سازگاری کمان یعنی بین هردو متغیری که در گراف محدودیت به هم مرتبط هستند نگاه کنیم که آیا برای هر مقداری که در دامنه‌ی متغیر اول هست مقداری پیدامیشه در دامنه‌ی متغیر دوم که به متغیر دوم نسبت بدم و محدودیت ها هنوز ارضا شوند اگه برای یکی از مقدایر متغیر اول نتونستیم مقداری در دامنه‌ی متغیر دوم پیدا کنیم که سازگار بشه اون مقدار را از دامنه‌ی متغیر اول حذف میکنیم به این صورت سازگاری کمان ایجاد میشه

- سعی کن همه کمانها را سازگار کنی
- اگر سعی کنیم NT و SA را سازگار کمان کنیم، به دامنه تهی می‌رسیم.
- سازگاری کمان می‌تواند قبل از جستجو به عنوان پیش پردازش اعمال شود.
- یا بعد از هر انتساب

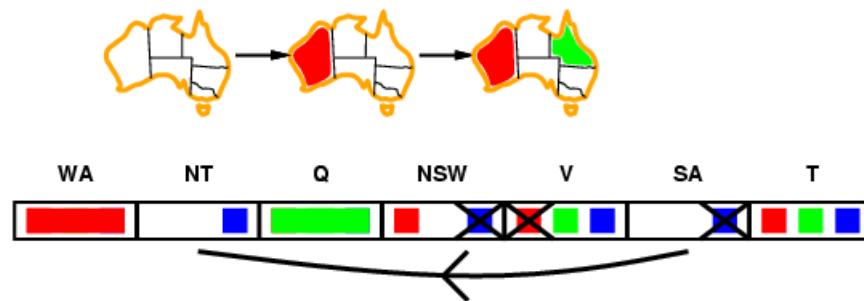
سازگاری کمان

- اگر X مقداری را از دست داد لازم است همسایگان آن چک شوند.
- چون حذف آن مقدار می تواند سازگاری کمان قبلی را از بین ببرد.



سازگاری کمان

- سازگاری کمان شکست را زودتر از چک جلو تشخیص می دهد.



سازگاری کمان

function AC-3(*csp*) **returns** false if an inconsistency is found and true otherwise
 $queue \leftarrow$ a queue of arcs, initially all the arcs in *csp*

```

while queue is not empty do
     $(X_i, X_j) \leftarrow \text{POP}(\text{queue})$ 
    if REVISE(csp,  $X_i$ ,  $X_j$ ) then
        if size of  $D_i = 0$  then return false
        for each  $X_k$  in  $X_i.\text{NEIGHBORS} - \{X_j\}$  do
            add  $(X_k, X_i)$  to queue
return true

```

function REVISE(*csp*, X_i , X_j) **returns** true iff we revise the domain of X_i
 $\text{revised} \leftarrow \text{false}$
for each x **in** D_i **do**
if no value y in D_j allows (x,y) to satisfy the constraint between X_i and X_j **then**
 delete x from D_i
 $\text{revised} \leftarrow \text{true}$
return revised

دامنهای
 D_i
 تهی شد یعنی هیچ
 مقداری دیگه نمیتوانه
 بگیره

اگه دامنهای ایکس
 ای را اصلاح کنیم

بقیهای همسایه های
 X_i
 رو برو بررسی کن

سازگاریهای دیگر

سازگاری رأس:

- یک متغیر (متناظر با یک رأس در گراف م.ا.م.) سازگار-رأس است اگر همه مقادیر دامنه آن محدودیتهای یکتائی را ارضاء کند.
- مثلاً اگر مردم SA رنگ سبز را دوست ندارند باید از دامنه آن حذف شود.
- یک گراف سازگار-رأس نامیده می شود اگر همه رئوس آن سازگار-رأس باشند.

وقتی حذف کنیم رنگ
سبز را بقیه‌ی
مقادیری که توی
دامنه باقی ماندند
سازگاری راس دارند

سازگاریهای دیگر

سازگاری مسیر:

- سازگاری کمان امکان دارد دامنه متغیرها را به یک عضو کاهش داده و پاسخ مسئله را بیابد،
 - یا دامنه برخی متغیرها بدون عضو بماند و مشخص شود که مسئله بدون پاسخ است.
 - اما همواره برخی مشکلات را متوجه نمی شود.
 - مثلاً اگر در مسئله رنگ آمیزی نقشه دامنه هر متغیر دارای دو رنگ {آبی، سبز} باشد.
 - گراف سازگاری کمان دارد ولی مسئله حل شدنی نیست.

سازگاریهای دیگر

- دو متغیر $\{X_i, X_j\}$ نسبت به متغیر X_m سازگار-مسیر هستند اگر:
- برای هر انتساب $\{X_i=a, X_j=b\}$ که با محدودیتهای (X_i, X_j) سازگار است، انتسابی برای X_m وجود داشته باشد که محدودیتهای (X_m, X_j) و (X_i, X_m) را ارضاء کند.
- بطور مثال تلاش برای سازگاری مسیر (WA, SA) نسبت به NT در حالتی که دامنه متغیرها دارای دو رنگ $\{\text{آبی} \text{ و } \text{قرمز}\}$ باشد.
- دو انتساب سازگار $\{WA=\text{red}, SA=\text{blue}\}$ یا $\{WA=\text{blue}, SA=\text{red}\}$
- هیچ رنگی برای NT باقی نمی‌ماند.

سازگاریهای دیگر

- سازگاری- k
- یک م.ا.م. سازگار- k است اگر برای هر انتساب سازگار به $k-1$ متغیر، یک انتساب سازگار برای هر متغیر k ام وجود داشته باشد.
- سازگاری-۱ همانند سازگاری رأس
- سازگاری-۲ همانند سازگاری کمان
- سازگاری-۳ همانند سازگاری مسیر
- یک م.ا.م. سازگار- k -قوی است اگر سازگار- k ، سازگار- $k-1$ ، سازگار- $k-2$ ،، تا سازگار-۱ باشد.

سازگاریهای دیگر

محدودیتهای جهانی:

همانگونه که قبلاً اشاره شد یک محدودیت جهانی شامل محدودیتی است که شامل چند متغیر (نه الزاماً همه) گردد.

بطور مثال Alldiff

روشی ساده برای تشخیص ناسازگاری در Alldiff: اگر m متغیر در محدودیت شرکت دارند و اگر آنها روی هم دارای n مقدار متفاوت هستند و $m > n$ در این صورت محدودیت قابل ارضاء نیست.

سازگاریهای دیگر

- منجر به یک راهکار ساده:
 - هر متغیر تک مقدار را از محدودیت حذف نمایید،
 - مقدار آن متغیرها را از دامنه دیگر متغیرها حذف نمایید،
 - مادامی که متغیر تک مقداری باقی مانده این کار را تکرار کنید،
 - هر زمان یک دامنه تهی تشخیص داده شد، یا تعداد متغیرها بیش از تعداد مقادیر باقی مانده بود یک ناسازگاری تشخیص داده شده است.
- بطور مثال: در مسئله رنگ آمیزی WA,SA,NT با دو رنگ ناسازگاری تشخیص داده می شود (سه متغیر و دو رنگ).

انتخاب یک متغیر
انتخاب نشده

انتساب اولیه تهی
است

مسئله‌ی ارضای
محدودیت

جستجوی عقبگرد

```
function BACKTRACKING-SEARCH(csp) returns a solution or failure
    return BACKTRACK(csp, {})
```

```
function BACKTRACK(csp, assignment) returns a solution or failure
    if assignment is complete then return assignment
    var  $\leftarrow$  SELECT-UNASSIGNED-VARIABLE(csp, assignment)
    for each value in ORDER-DOMAIN-VALUES(csp, var, assignment) do
```

```
        if value is consistent with assignment then
            add  $\{ \text{var} = \text{value} \}$  to assignment
```

```
            inferences  $\leftarrow$  INFERENCE(csp, var, assignment)
```

```
            if inferences  $\neq$  failure then
                add inferences to csp
```

```
                result  $\leftarrow$  BACKTRACK(csp, assignment)
```

```
                if result  $\neq$  failure then return result
```

```
                remove inferences from csp
```

```
                remove  $\{ \text{var} = \text{value} \}$  from assignment
```

```
return failure
```

استنتاج
به کاربردن سازگاری
راس و کمان و مسیر
و...

ایا این مقدار سازگار
هست با انتسابی که
تاالان داشتیم؟

چه مقداری را برابر
متغیر انتخاب کنیم

حذف استنتاج هایی
که تاالان انجام دادیم

به صورت بازگشتی
و عمقی میریم جلو

مازیار پالهنجک

اگه همه‌ی مقادیر چک شد
و سازگاری پیدا نشد ما
شکست بر میگردانیم

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

15

ایا میتوانیم از دامنه‌ی
متغیرها مقادیری
را حذف کنیم؟

15

جستجوی محلی برای ارضاء محدودیتها

- جستجوهای محلی از حالت کامل استفاده می کنند.
- در این حالت وضعیتی که همه متغیرها مقدار دارند.
- برای اعمال به م.ا.م.
- اجازه داشتن حالاتی که محدودیتها ارضا نشده اند.
- تغییر مقدار متغیرها (در جهت ارضاء کردن محدودیتها)
- انتخاب متغیر: بصورت تصادفی هر متغیر ناسازگار انتخاب شود.
- انتخاب مقدار با مکافه کمترین برخورد:
- انتخاب مقداری که کمترین محدودیتها را می شکند.

Figure 6.9

function MIN-CONFLICTS(*csp*, *max_steps*) **returns** a solution or *failure*
inputs: *csp*, a constraint satisfaction problem
max_steps, the number of steps allowed before giving up

```

current  $\leftarrow$  an initial complete assignment for csp
for i = 1 to max_steps do
    if current is a solution for csp then return current
    var  $\leftarrow$  a randomly chosen conflicted variable from csp.
    value  $\leftarrow$  the value v for var that minimizes CONFLICTS(csp, var, v, current)
    set var=value in current
return failure

```

حالت اولیه یک
انتساب کامل است که
لزوما هم سازگار
نیست

اگه حالت فعلی یک
حل است پس جواب
پیدا شده

The Min-Conflicts local search algorithm for CSPs. The initial state may be chosen randomly or by a greedy assignment process that chooses a minimal-conflict value for each variable in turn. The CONFLICTS function counts the number of constraints violated by a particular value, given the rest of the current assignment.

مازيار بالهنجك

هوش مصنوعي - نيمسال اول ۱۰۱-۱۰۲

17

اگه ماكس استپ تمام
شد شکست
برمیگردانیم

مقداری را انتخاب
میکنیم که کمترین
تداخل را ایجاد کنه

مثال: ۸ وزیر

حالات: ۸ وزیر در ۸ ستون (8^8 حالت)

انتخاب تصادفی یک متغیر

انتخاب مقداری برای آن متغیر با کمترین تضاد

امکان استفاده تا چند میلیون وزیر!

اگه از این مکاشفه
استفاده کنیم تا
چند میلیون وزیر راهم
میتوانیم با این روش
حل کنیم

این متغیر مشکل داره
پس باید مقدارشو
تغییر بدیم
پس توی همون
ستونی که هست
سطرش را عوض
میکنیم

مکافه میگه مقداری را
برای متغیر انتخاب کن که
کمترین تضاد و کانفلیکت
را ایجاد کنه

اگه وزیر را ببریم به
سطر بالایی باعث
ایجاد دوتا برخورد با
وزیرهای دیگه میشه

بین این دوتا یکی
را به تصادف
انتخاب میکنیم

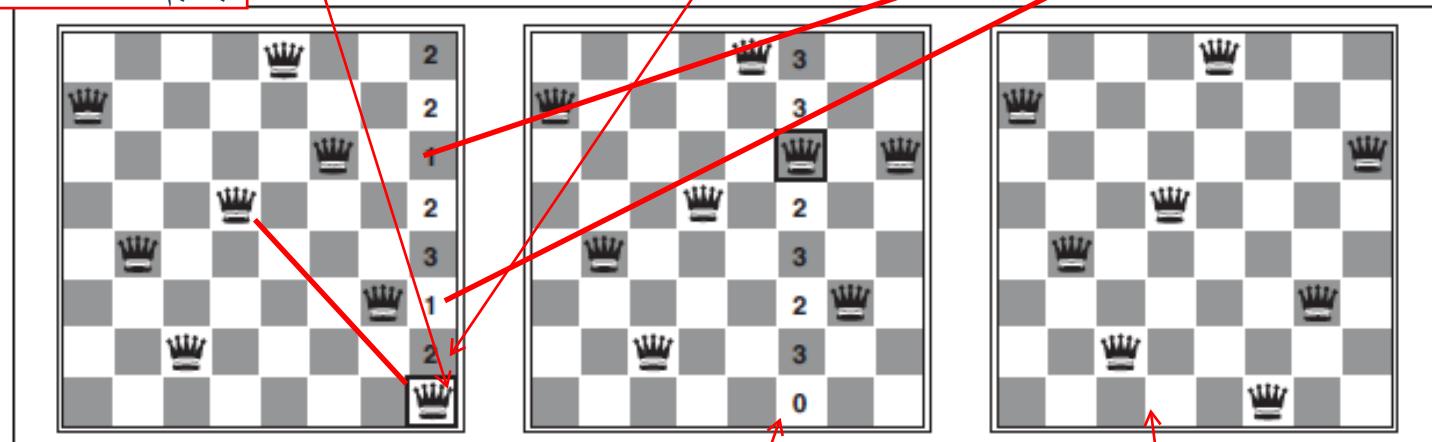
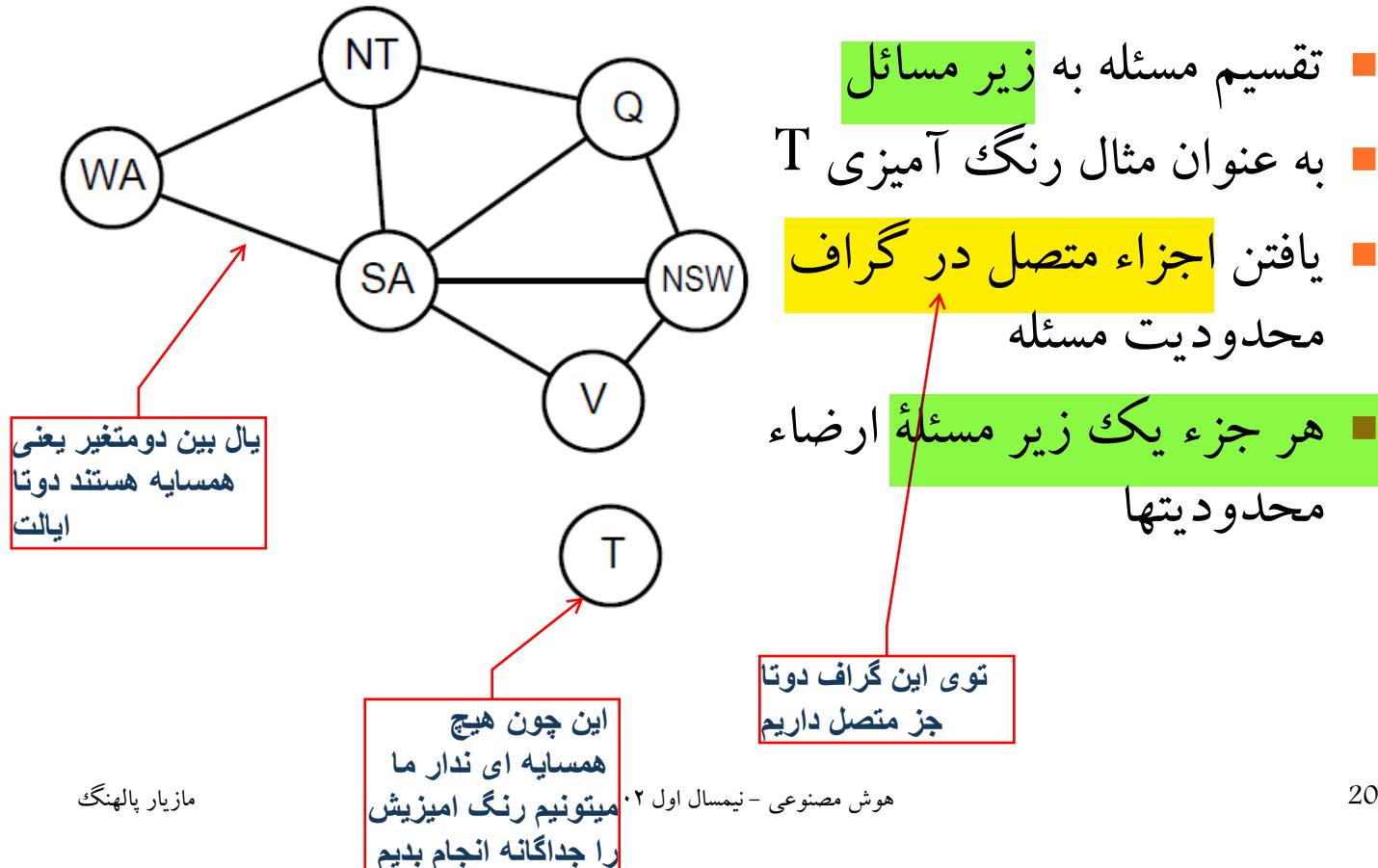


Figure 6.9 A two-step solution using min-conflicts for an 8-queens problem. At each stage, a queen is chosen for reassignment in its column. The number of conflicts (in this case, the number of attacking queens) is shown in each square. The algorithm moves the queen to the min-conflicts square, breaking ties randomly.

شکستن مسیله به تعدادی زیرمسئله و حل اون زیرمسئله ها و درنهایت حل هارا ترکیب کنیم

استفاده از ساختار مسئله



کل تعداد متغیر هامون
در کل مساله

اگر هر زیر مسئله دارای C متغیر از n متغیر مسئله باشد.

توی هرزیرمساله ما
 c
تا متغیر داریم پس تا عمق سی باید
بریم پایین
اندازه ای دامنه هم دی است یعنی
هرمتغیر
 d
مقدار میتونه به خودش بگیره که
مثل ضریب انشعاب است

تعداد زیر مسائل n/c

اگر d اندازه هر دامنه

هزینه بدترین حل $O(d^c \cdot n/c)$

در مقابل $O(d^n)$

ضریب انشعاب d

مسئله ای که ممکن است چند میلیون سال طول بکشد در چند
ثانیه (یا کسری از ثانیه) حل خواهد شد.

حل مساله بدون کمک
گرفتن از زیرمسائل

م.ا.م. با ساختار درختی

- یک گراف محدودیت یک درخت است هر گاه هر دو رأس فقط با یک مسیر به هم وصل باشند.
- هر م.ا.م. با ساختار درختی در زمانی خطی نسبت به تعداد متغیرها قابل حل است.

- یک م.ا.م. **سازگار کمان جهتدار** تحت یک ترتیب دهنده متغیرهای X_1, X_2, \dots, X_n است اگر و تنها اگر X_i با X_j سازگار

X_j
در ترتیب دهنده بعد از
 X_i
قرار گرفته باشد

بین متغیرهای مسئله
یک ترتیب گذاشته

کمان باشد برای $j > i$

سازگاری جهت دار
است یعنی از متغیری
که اندیشش کمتر
است به متغیری که
اندیشش بیشتر است

با این روش ترتیب متغیرها معلوم شد
سازگاری کمان هم جهت دار است پس
باید چک کنیم ایا
A با
B
سازگاری کمان دارد؟

با ساختار درختی

- یک متغیر را به عنوان ریشه انتخاب کنید،
- متغیرها را از ریشه به برگ به گونه ای ترتیب دهید که ولی هر رأس قبل از آن در این ترتیب دهی قرار گیرد.
- ترتیب دهی همبندی (topological order)

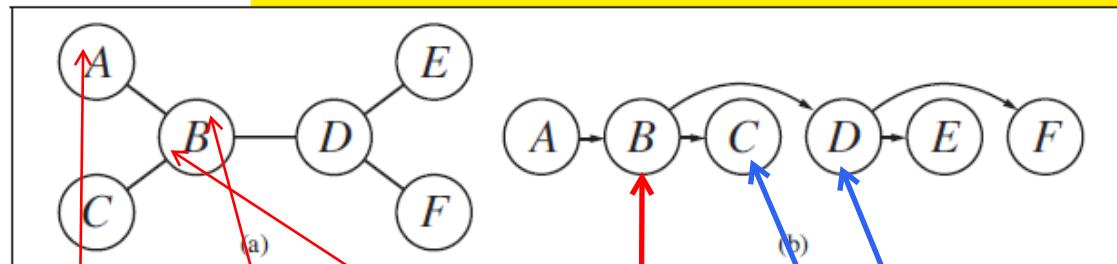


Figure 6.10 (a) The constraint graph of a tree-structured CSP. (b) A linear ordering of the variables consistent with the tree with A as the root. This is known as a **topological sort** of the variables.

مازیار پالهنجک

این راس را به
عنوان ریشه انتخاب
کردیم

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

این راس فرزند
A
است

راس بی دوتا فرزند
داره
C,D
پس باید بعد از بی
قرار بگیرن

23

23

Figure 6.11

```
function TREE-CSP-SOLVER(csp) returns a solution, or failure
    inputs: csp, a CSP with components X, D, C
```

n \leftarrow number of variables in *X*

انتساب تهی

assignment \leftarrow an empty assignment

ترتیب دهی به راس
ها

root \leftarrow any variable in *X*

X \leftarrow TOPOLOGICAL SORT(*X, root*)

ترتیب دهی به راس
ها

for *j* = *n* down to 2 **do**

 MAKE-ARC-CONSISTENT(PARENT(*X_j*), *X_j*)

ایجاد سازگاری کمان
بین ولی راس و
خودش

if it cannot be made consistent **then return** *failure*

for *i* = 1 to *n* **do**

assignment[*X_i*] \leftarrow any consistent value from *D_i*

وقتی
J=2

if there is no consistent value **then return** *failure*

میشه ولی راسمون
میشه راس یک که
ریشه است

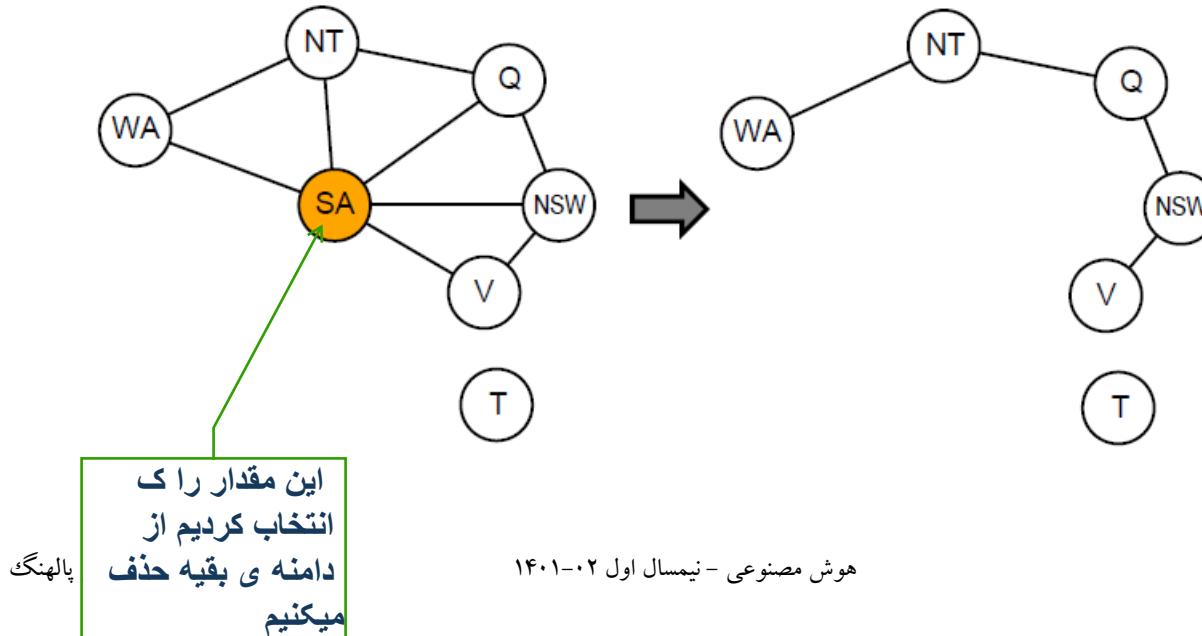
return *assignment*

The TREE-CSP-SOLVER algorithm for solving tree-structured CSPs. If the CSP has a solution, it finds it in linear time; if not, we will detect a contradiction.

همه ی گراف ها ساختار درختی ندارند و
برخی ساختار تقریباً درختی دارند
مثلًا اگه یه نود را از گراف زیر حذف کنیم
تبدیل به درخت میشه و میتوانیم از الگوریتم
صفحه قبل استفاده کنیم

۳.۱.۳. با ساختار تقریباً درختی

■ مقدار دهی یک متغیر، حذف مقدار آن از دیگر متغیرها، اجرای
الگوریتم درختی



م.ا.م. با ساختار تقریباً درختی

- ممکن است مقدار SA مناسب نباشد نیاز به تکرار
- بطور کلی:
- انتخاب مجموعه S از کل متغیرهای م.ا.م. به گونه ای که پس از حذف S گراف محدودیت درخت شود.
- برای هر انتساب سازگار متغیرهای S
- حذف مقادیر ناسازگار با S از دیگر متغیرها،
- اگر مابقی م.ا.م. دارای حل است آنرا به همراه مقادیر S بازگردان

اول یک انتساب سازگار برای متغیرهای توی

S

پیدامیکنیم بعد مقادیرناسازگار با متغیرهای اس را از متغیرهای دیگه حذف میکنیم بعد میریم برای بقیه ی متغیرها به روش درختی دنبال حل میگردیم اگه یه حل پیدا کردیم اون حل را به همراه مقادیری که برای اس پیدا کردیم برمیگردانیم اگه هم پیدانکردیم یه انتساب سازگار دیگه را برای متغیرهای مجموعه ی اس پیدامیکنیم و اونها را تست میکنیم



مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

27

خلاصه

اضافه کردن قسمت استنتاج
که سازگاری هایی که داشتیم را
چک میکردیم
اگه اون سازگاری ها باعث میشد
مقادیری از متغیرها حذف شه
حذف میکردیم
این باعث میشد جستجو بهتر
انجام شه.

الگوریتم جستجوی عقبگرد بصورت کاملتر

■ سازگاری کمان

■ معرفی سازگاریهای دیگر،

■ جستجوی محلی

■ استفاده از ساختار مسئله

■ ساختار درختی

■ ساختار تقریباً درختی

اون گراف های محدودیتی که
ساختار درختی دارند را اگه یک
ترتیب دهی برای متغیرهایشون
در نظر بگیریم و سازگاری کمانی
برآشون اعمال کنیم

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

پیدا کردن حل با
زمان خطی نسبت
به تعداد متغیرها

حذف چندین راس
از گراف محدودیت
برای تبدیل به
ساختاری درختی

28

28

امام علی (ع) :

وَالْفُرْصَةُ تَمُرُّ مِنَ السَّحَابِ، فَانْتَهِزُوا فُرْصَ الْخَيْرِ

فرصت به مانند ابر سگدرا می گذرد، پس فرصت های نیک را غنیمت دانید

والسلام

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی

عاملین منطقی - ۱

نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پاله‌نگ

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

مقدمه

- طراحی عاملینی که درباره محیط اطراف خود می دانند و استدلال می کنند.
- جزء اصلی عامل دانش-مبنا: پایگاه دانش
- پایگاه دانش = مجموعه ای از جملات به یک زبان رسمی

مثل هر عامل دیگه ای یک درک میگیره و
یک عمل پرمیگردانه

پایگاه دانش

یک عامل دانش - مبنای ساده

function KB-AGENT(*percept*) returns an *action*

persistent: *KB*, a knowledge base

t ← a counter, initially 0, indicating time

زمان

TELL(*KB*, MAKE-PERCEP-SENTENCE(*percept*, *t*))

action ← ASK(*KB*, MAKE-ACTION-QUERY(*t*))

TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))

t ← *t* + 1

return *action*

باید سوال را تبدیل
به زبانی کنیم که
پایگاه دانش
میفهمه

Figure 7.1 A generic knowledge-based agent. Given a percept, the agent adds the percept to its knowledge base, asks the knowledge base for the best action, and tells the knowledge base that it has in fact taken that action.

درکی که در لحظه
ی تی گرفتیم را به
مازیار
یک جمله‌ی معتبر
تبدیل کن و توی
پایگاه دانش ذخیره
کن

از پایگاه دانش
میرسیم در این
زمان چه عملی را
انجام بدیم بهتر
است؟

- یک عامل دانش - مبنا را در سه سطح می توانیم تعریف کنیم:
- سطح دانش: آنچه می داند و اهدافش چیست. "سی و سه پل جنوب اصفهان را به شمال آن متصل می کند".
- سطح منطق: کد گذاری آنچه می داند، Nisfahan)
- سطح پیاده سازی: رشتہ، آرایه

- پایگاه دانش می تواند ابتدا خالی بوده و با افزودن جملات یک به یک شکل بگیرد.
- به این روش، **روش توصیفی** گفته می شود.
- در **روش روالی**، پایگاه دانش از ابتدا کد می شود.

معرفی اجزای محیط

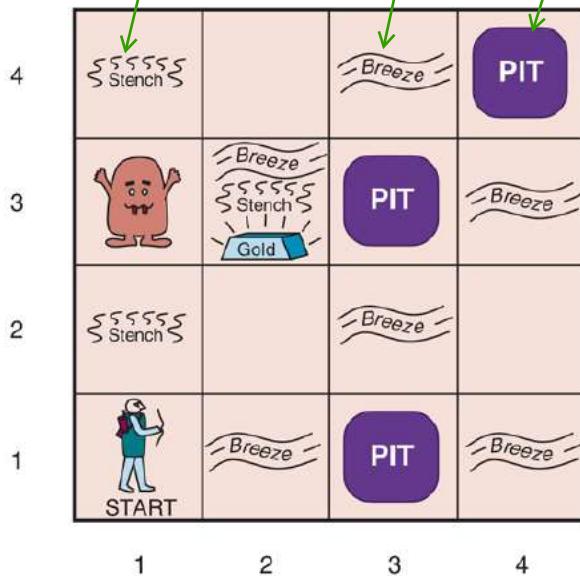
- حاکم
- حسگرها ۱.
- اقدامگرها ۲.
- کارایی ۳.
- محیط ۴.

دنیای دیو

نیمیم

چاه

بوی بد



کماح

معیار کارآئی

- خروج با طلا +۱۰۰۰
- نابودی -۱۰۰۰

۱- برای هر حرکت، ۱۰- برای تمام کردن تیرها
پایان بازی هنگام از بین رفتن عامل، یا خروج

عامل

خانه شامل گودال
یا دیو

محیط:

اتاقها بصورت 4×4

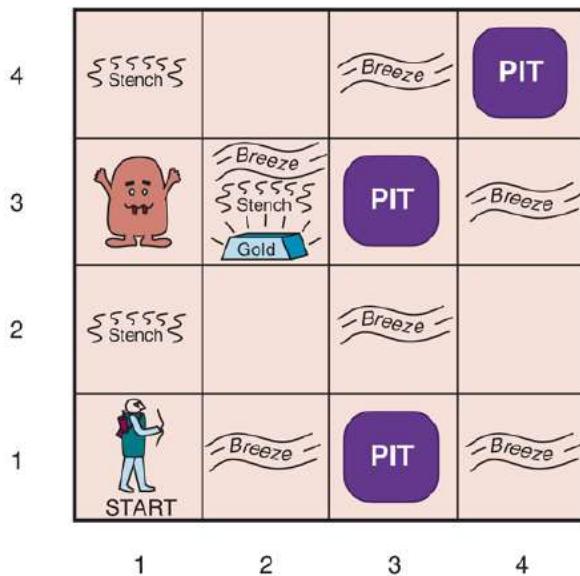
شروع از خانه [۱، ۱]، جهت عامل به سمت شرق
دیو، طلا یا گودال بصورت تصادفی در هر خانه
ای به جز خانه شروع

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

6

دنیای دیو



اعمال

- چرخش به چپ ۹۰ درجه،
- چرخش به راست ۹۰ درجه،
- حرکت به جلو،
- گرفتن طلا،
- تیر زدن
- بالا رفتن برای خروج از غار در خانه [۱,۱]

ادراکات

- بوی بد در خانه شامل و اطراف دیو
- نسیم در خانه های اطراف گودال
- درخشندگی در خانه شامل طلا
- ضربه بعد از بخورد عامل با دیوار
- جیغ هنگام کشته شدن دیو

ادراکات بصورت یک لیست با ۵ نماد به عامل داده می شود. بطور مثال:

[*Stench, Breeze, None, None, None*]

مازیار پالهنجک

اکتشاف در دنیای دیو

محلی که عامل
قرارگرفته

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

ok
یعنی امن است

اکتشاف در دنیای دیو

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

B = breeze

نسیم حس کرده
پس خانه های

اطرافش احتمال داره
گودال داشته باشند

گودال را با

p

نشان میدیم

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

9

اکتشاف در دنیای دیو

مشکوک به وجود
گودال

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

اکتشاف در دنیای دیو

S = بوی بد
دیو در خانه های
اطراف میتوانه باشد

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

v = visited
برگشت به عقب

اکتشاف در دنیای دیو

دیو اینجاست
ما توى ۱ و ۲ بوي بد
حس نكرديم پس
ميشهيم توى ۲ و ۲ ديو
نيست

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

چون توى ۱ و ۲ که
بوديم نسيم حس
نكرديم پس نتيجه
ميگيريم گودال توى
خانه ي ۱ و ۳ است

مازيار پالهنجك

هوش مصنوعي - نيمسال اول ۱۰۱-۰۲

12

اکتشاف در دنیای دیو

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

اکتشاف در دنیای دیو

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

اکتشاف در دنیای دیو

اس یعنی بُوی بد
به دلیل وجود دیو در
خانه‌ی ۱۹۳

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

نسیم حس کرده پس
خانه‌های اطرافش
ممکن است گودال
داشته باشن
(یکیشون حتماً داره)

جی یعنی درخندگی
پس میتونه طلا
رابگیره و از خانه
هایی که مطمئن شده
امن هستند برگردد
بره خانه ۱۹۱ و بره
بیرون غار

منطق

- یک زبان رسمی برای نمایش اطلاعات
- یک زبان نمایش دانش بو سیله دو جنبه تعریف می گردد:
- دستور (syntax): تشکیل جملات معتبر در زبان

$$x + 2 \geq y$$

$$x + 2 > \{ y \}$$

- معنا (semnatic): معنا یا درستی جملات نسبت به یک دنیای ممکن
- مثلاً $x+2 \geq y$ در دنیائی که $x=3$ و $y=1$ درست است.
- یک دنیای ممکن **مدل** نامیده می شود.

- اگر جمله α در مدل m درست باشد، گفته می شود جمله m جمله α را ارضامی کند، یا یک مدل برای α است.

ایجاب کردن

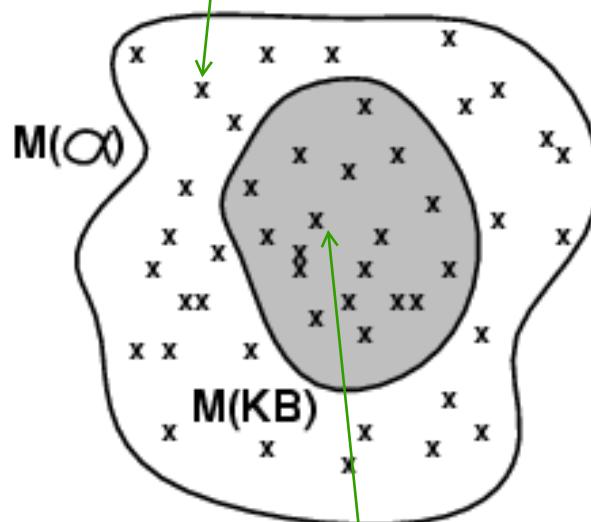
- اینکه جمله ای بطور منطقی از جمله ای دیگر تبعیت می کند.

$$\alpha \models \beta$$

آلفا ایجاب میکنے بتا را
یعنی اگه آلفا درست باشه بلاجبار بتا هم درسته
درست بودن بتا از درست بودن آلفا ایجاد میشه

یه سری مدل هستند که پایگاه دانش در انها درست است اگه آلفا هم توى اون مدل ها درست باشه از درستی پایگاه دانش درستی آلفا رو هم نتیجه گرفتیم

مدلهای



مازیار پالهنجک

مجموعه مدل هایی که پایگاه دانش درش درست است

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

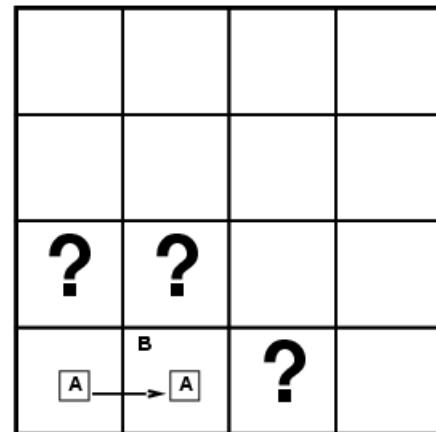
پایگاه دانش آلفا را ایجاد میکنیم

- می گوئیم m یک مدل برای جمله α است اگر α در m درست باشد.
- مجموعه همه مدلهای α $M(\alpha)$
- اگر و تنها اگر هر مدلی که در آن درست است، α نیز در آن درست باشد.

$$\text{اگر و تنها اگر } KB \models \alpha \\ M(KB) \subseteq M(\alpha)$$

ایجاب کردن در دنیای دیو

- وضعیت پس از تشخیص هیچ چیز در [۱۰] و نسیم در [۲۱]
- در نظر گرفتن همه مدلها (فقط با در نظر گرفتن گودال)
- ۳ گزینه بولی برای [۱۰۲]، [۲۰۲] و [۳۰۱]
در نتیجه ۸ مدل ممکن



مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

بود یا نبود گودال
توی یک خانه که با
بولین نشان میدهیم
دو تاحالت داره

20

گودال

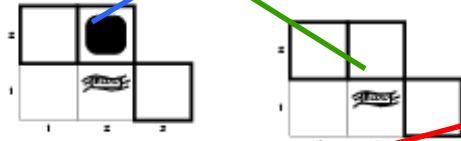
مدلها

توى يكى از سه
خانه گودال باشه

هېچکدام از سه
خانه گودال
نداشته باشند.

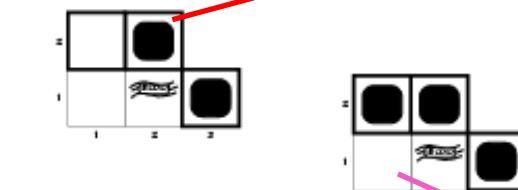


توى دوتا خانه گودال
باشه

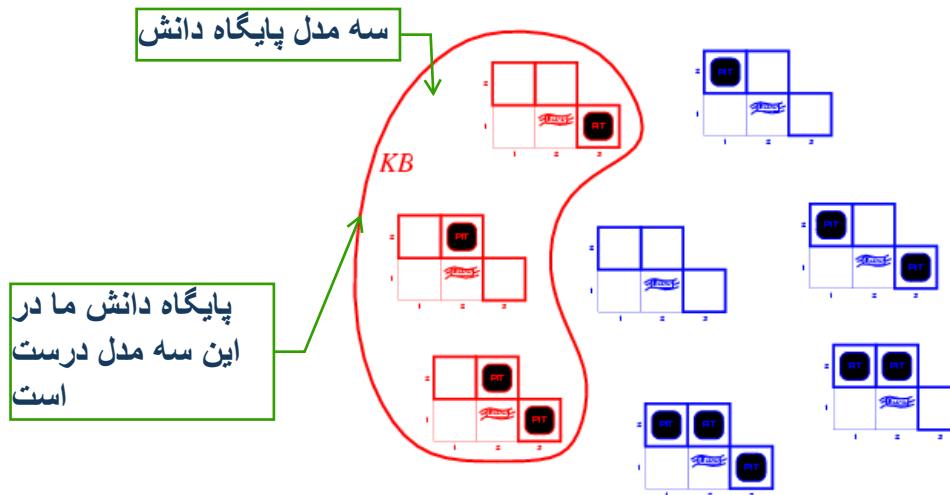


توى يكى از سه
خانه گودال باشه

در سه خانه گودال
باشه



مدلها



KB = قوانین دنیای دیو + مشاهدات ■

مازیار پالهنجک

توی خانه ۱ او ۱ که میدانیم نیست
پس یا توی خانه ۲ او ۲ است یا ۳ او ۳
پس مدل هایی از صفحه قبل انتخاب
میشن که این دو خانه را محتمل به
وجود گودال گرفتن

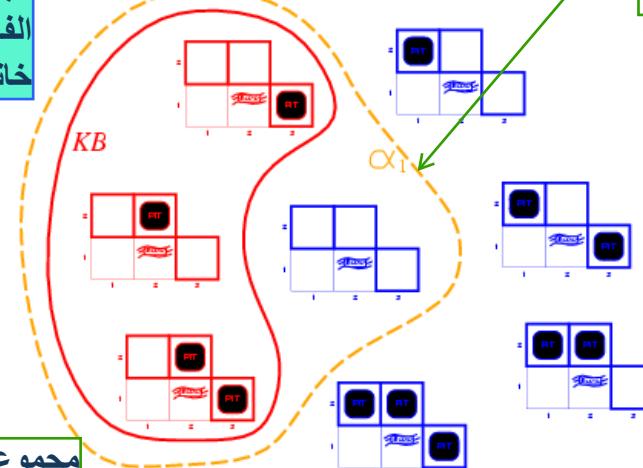
پایگاه دانش ما مشاهده کرده ک
توی خانه ۱ او ۲ نسیم هست
پس پایگاه دانش ما توی همه ی
مدل های صفحه قبل درست نیس
چرا؟ چون حس کردن نسیم درین
خانه یعنی گودال در یکی از خانه
های اطرافش هست

22

مدلها

سوال:
ایا آلفا
از پایگاه دانش ایجاب
میشه؟
الف1 یه جمله است که میگه
خانه ی ۱۰۲ امن است

خط چین زرد
مجموعه‌ی مدل‌های
آلفا ۱



یعنی اگه پایگاه دانش درست باشه حتما نتیجه میگیریم که خانه ی ۱۰۲ امن است.

مجموعه‌ی مدل‌های آلفا ۱ چیان؟
که توشون خانه ی موردنظر امن است
یعنی توشون گودال نیست

$$KB = \text{قوانين دنیای دیو} + \text{مشاهدات}$$

$$[1, 2] = \alpha_1 \quad \blacksquare$$

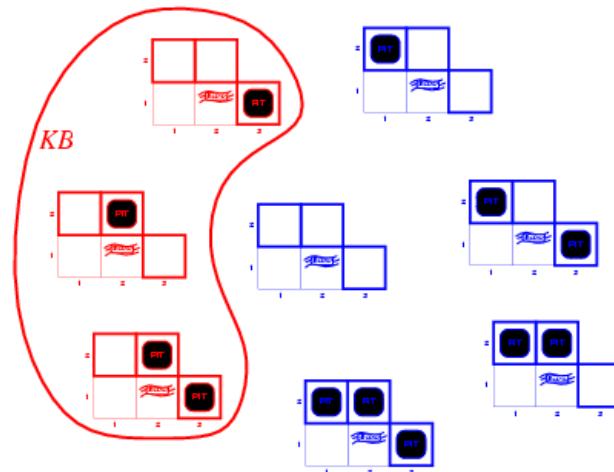
$$KB \models \alpha_1, \quad \blacksquare$$

مازیار پالهنجک

23

(مجموعه مدل‌های پایگاه دانش را در نظر بگیر منحنی قرمز چون مجموعه مدل‌های پایگاه دانش زیرمجموعه‌ی آلفا ۱ است پس میشه بگیم پایگاه دانش، آلفا ۱ را ایجاب میکنه

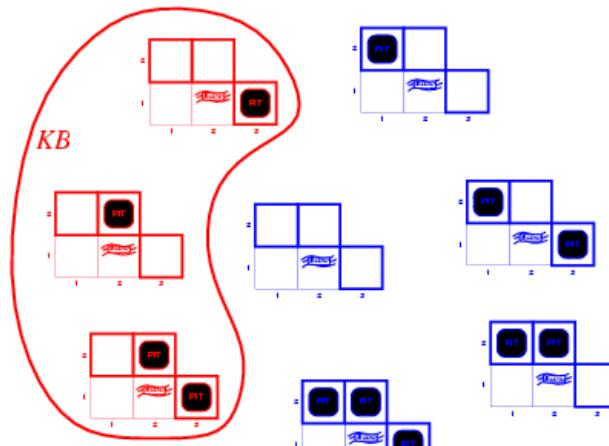
مدلها



قوانين دنیا دیو + مشاهدات $= KB$ ■

مدلهای

آیا پایگاه دانش آلفا ۲ را ایجاب میکنه؟
باید اول مدلهای آلفا ۲ را مشخص کنیم اگه مجموعه مدلهای پایگاه دانش زیرمجموعه‌ی مدلهای آلفا ۲ باشه میتوانیم بگیم که ایجاب میکنه



از مشاهدات من توی
پایگاه دانش نتیجه
گرفتیم که توی ۱ و ۲
نمی هست

= قوانین دنیای دیو + مشاهدات KB ■

"[2,2] = α2" ■

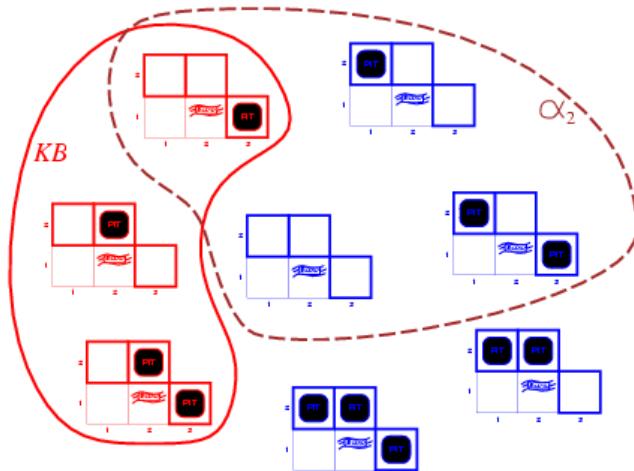
مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

25

یک جمله‌ی دیگه

مدلها



از درستی پایگاه دانش ، الزاما
درستی جمله α_2 نتیجه
نمیشه
پس الزاما نمیشه بگیم خانه ۲
موردنظر امن است

KB = قوانین دنیا دیو + مشاهدات
 $[2,2] = \alpha_2$ "امن است"
 $KB \not\models \alpha_2$

مازیار پالهنجک

مدل هایی که توشون
هوش مصنوعی - نیمسال اول
در خانه ۲ و ۳
گودال نیست

26

ایجاب کردن

$KB \models \alpha$ ■

پس علاقه داریم این
جمله هم ای متناظری
در واقعیت داشته
باشه که از واقعیات
بدست اومده

واقعیت

بدست می دهد

جمله

واقعیات

ایجاب می کند

با ایجاب کردن
میتوانیم بگیم اگه
جملات درست باشند
پس یک جمله هم باید
درست باشه

جملات چیزی را
نمایش میدهند که در
واقعیت وجود داره
ینی جملات
متناظر شون در دنیای
واقعی وجود داره

داخل عامل اون چیزی که بدست میاد یا ایجاب میشه یه
متناظر واقعی در دنیای خارجی داشته باشه

ایجاب کردن میتوانه ما را به استنتاج راهنمایی کنے

یه جمله را میدن ما
باید چک کنیم ایا اون
جمله از پایگاه دانش
ایجاب میشه یا ن؟

اون روال توانایی
ایجاد آلفا را داشته
باشه

استنتاج

- یک روال استنتاج یکی از دو کار را می تواند انجام دهد:
 - با داشتن یک KB
 - بیابد
 - با داشتن یک جمله ایجاب شدن آن توسط KB را بررسی کند.
 - یک روال استنتاج که فقط جملاتی که ایجاب می شوند را تولید می کند یک استنتاج موثر یا معتبر (sound) نامیده می شود.
 - $KB \vdash_i \alpha$ جمله α توسط روال استنتاج i از KB ایجاد می شود.
 - کامل بودن: i کامل است اگر $KB \models \alpha$ آنگاه

اگر آلفا از پایگاه دانشما ایجاب
باشه
در این صورت آلفا توسط روال
آی
از پایگاه دانش استنتاج بشه

۱۴۰۱-۰۲ - نیمسال اول هوش مصنوعی

روال استنتاج

زبانی که با ان دانش
را نمایش میدیم

منطق گزاره‌ای: دستور

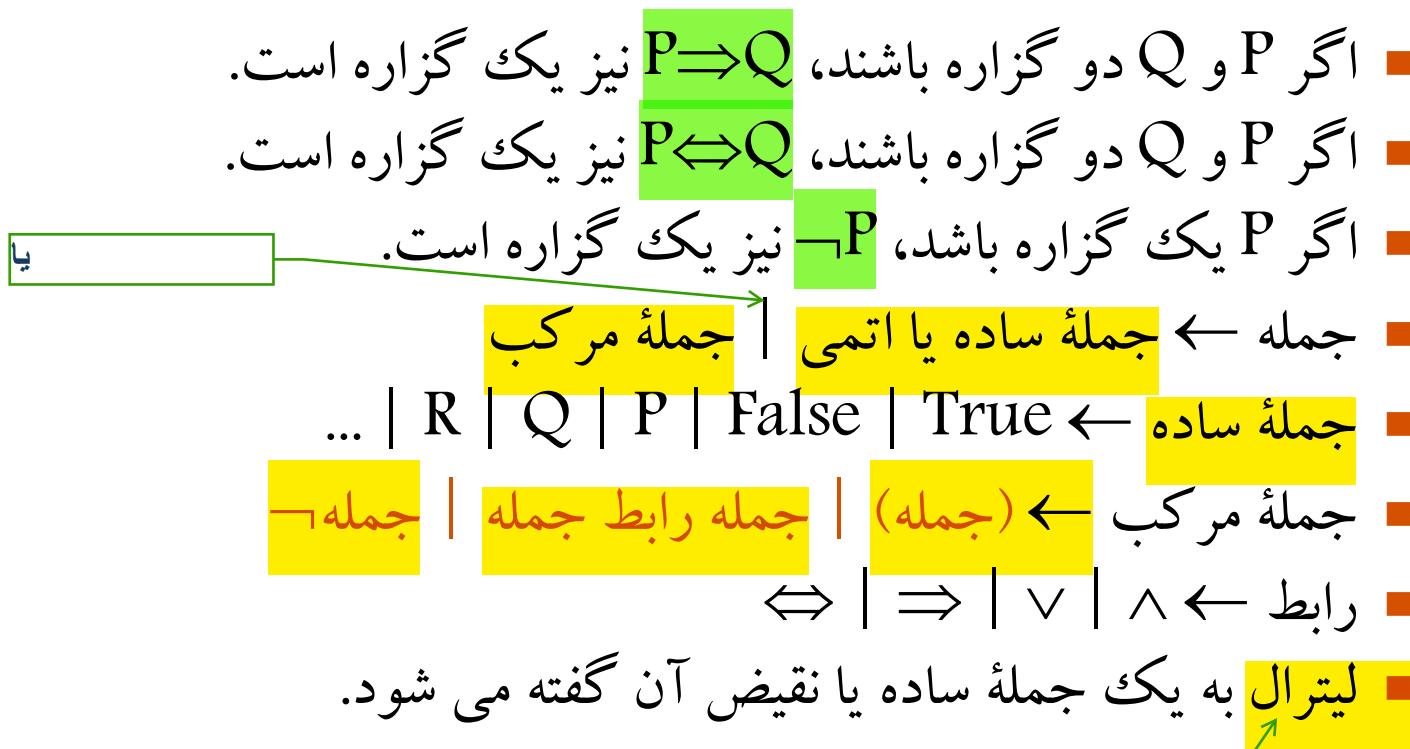
ساده‌ترین منطق

گزاره یک جملهٔ خبری که بتوان به آن ارزش درست یا نادرست
نسبت داد.

True, False

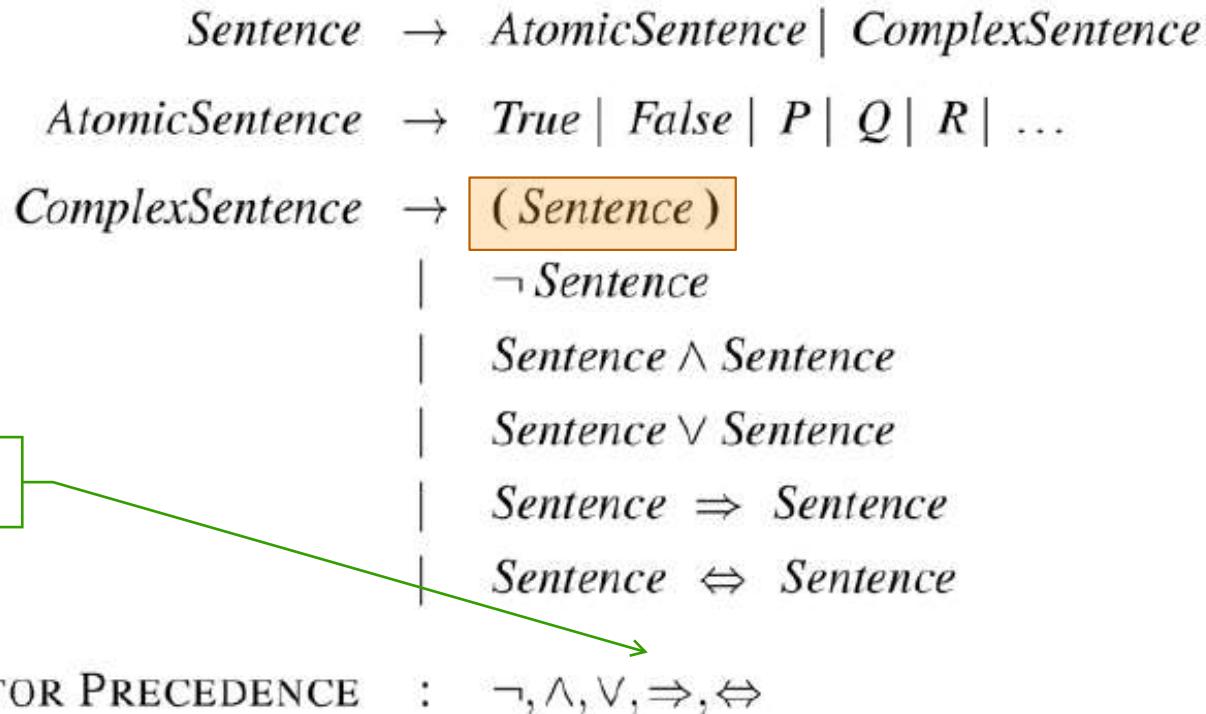
عطف دو گزاره

- نمادها: ثابت‌های منطقی (درست، نادرست)، متغیرهای گزاره‌ای (\dots, P, Q)، رابطه‌ای منطقی و پرانتزها
- ثابت‌های منطقی به تنهایی یک گزاره هستند
- اگر P و Q دو گزاره باشند، $P \wedge Q$ نیز یک گزاره است.
- اگر P و Q دو گزاره باشند، $P \vee Q$ نیز یک گزاره است.



ليترال مثبت اگه علامت
نقىض جلوش نباشه
اگه باشه ليترال منفى

Figure 7.7



A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

- برای جلوگیری از ابهام و افزایش خوانائی در صوت‌تر نیاز از پرانتز و کروشه استفاده می‌شود.

منطق گزاره‌ای : معنا

- معنا قوانینی را برای تعیین درستی یک جمله در یک مدل را بیان می‌دارد.
- در منطق گزاره‌ای، یک مدل مقادیر درستی نمادهای گزاره‌ای را می‌شاند.

در منطق گزاره‌ای
وقتی برای همه این
گزاره‌ها یه مقداری
را تعیین کنیم یه مدل
درست میشه

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

گودال توى خانه‌ی
دو و یک وجود داره؟

- معنای یک جمله نیز با داشتن یک مدل باید مشخص شود.
- درست همیشه یک واقعیت درست و نادرست یک واقعیت همیشه نادرست است.

مقدم

تالی

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

پایان



دانشگاه صنعتی اصفهان – مجموعه تالارها

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

35

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی

عاملین منطقی - ۲

نیمسال اول ۱۴۰۱-۱۴۰۲

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

ابزاری برای نمایش
دانش

عامل دانش - مبنا

■ سطح دانش، سطح منطق، سطح پیاده سازی

■ دنیای دیو

■ اکتشاف در دنیای دیو

■ منطق

■ ایجاد کردن

■ مدلها

■ استنتاج

■ یکی از دو وظیفه: تولید جملاتی که ایجاد می شوند، یا چک ایجاد شدن یک جمله

■ موثر

■ کامل

■ منطق گزاره ای

جملات در دنیای دیو

- فرض کنید درست باشد اگر یک گودال در $[x,y]$ باشد.
- فرض کنید درست باشد اگر نسیم در $[y,x]$ باشد.
- برای هر جمله یک برجسب R_i جهت رجوع در نظر می گیریم.
- می دانیم سلول $[1,1]$ گودال نیست:

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

- یک خانه نسیم دار است اگر و تنها اگر در خانه مجاور آن یک گودال باشد (یک جمله برای هر خانه):

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

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

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

3

یک قانون است

جملات در دنیای دیو

- جملات قبل درست در هر دنیای دیو.
- برای مثال ما:

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

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

نمیم نیست

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

تا متغیر ۷

داریم پس تعداد سطر های

جدولمون میشه دو به توان ۷
۱۲۸

باید بینیم از چه
متغیرهایی استفاده
شده تا حالا

در این خانه گودال
نیست

جدول درستی برای استنتاج

ایجاب می شود؟ KB از $\alpha_1 = \neg P_{1,2}$ یا $\neg P_{1,2}$ ■

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

آیا جاهایی که مدل های
پایگاهدانش درست است
مدل های آلفا ۱ هم درست
است؟

الان سه تا مدل داریم
که پایگاه دانش در
آن درست است

عی - نیمسال اول ۱۴۰۱-۰۲

باید بینیم با توجه به
مقداردهی این
متغیرها، پایگاه دانش
کی درست هست؟

5

آیا الفا از پایگاه
دانش ایجاب میشه؟

آیا الفا از پایگاه
دانش ایجاب میشه؟

استنتاج با فهرست کردن

■ فهرست کردن بصورت عمق نخست

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

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

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

مدلی که قراره بدست
بیاریم اول کار تهی

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

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

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

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

else do

$P \leftarrow FIRST(symbols)$

$rest \leftarrow REST(symbols)$

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

and

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

اگه درست بود باید
چک کنیم آیا الفا هم
در اون مدل درست
است یا نه؟

مازیار پالهنجک

آیا پایگاه دانش با این
مدلی که بدست اوردیم
درست است یا نه؟

اول ۱۰-۱۶

اگه پایگاه دانش برای اون مدل
درست نباشه مثل این است که بگیم
برامون مهم نیست که آلفا از پایگاه
ایجاب میشه یا ن

مدلی که داشتیم را
باید با مقداردهی
جدید متغیر جدیدمون
اجتماع بگیریم

- موثق چون همان تعریف ایجاب کردن را پیاده سازی می کند.
- کامل چون برای هر KB و α کار می کند و پایان می یابد.
- با n نماد پیچیدگی زمانی $O(2^n)$ و پیچیدگی فضا $O(n)$

چون هر دفعه یکی از متغیرها مقدار میگیره و ما تا عمق آن پایین میریم

هم ارزی منطقی

مجموعه مدل های
آلفا زیرمجموعه ای
مجموعه مدل های بتا

مجموعه مدل های بتا
زیرمجموعه ای
مجموعه مدل های آلفا

دو جمله هم ارز منطقی هستند اگر و تنها اگر در مدل های یکسانی درست باشند

$$\alpha \equiv \beta \text{ iff } \alpha \models \beta \text{ and } \beta \models \alpha$$

- $(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$ commutativity of \wedge جابه جایی
- $(\alpha \vee \beta) \equiv (\beta \vee \alpha)$ commutativity of \vee شرکت پذیری و
- $((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$ associativity of \wedge حذف نقيض دوگانه
- $((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$ associativity of \vee حذف نقيض دوگانه
- $\neg(\neg \alpha) \equiv \alpha$ double-negation elimination عكس و نقيض
- $(\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha)$ contraposition اصل شرطی
- $(\alpha \Rightarrow \beta) \equiv (\neg \alpha \vee \beta)$ implication elimination حذف دوشرطی
- $(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$ biconditional elimination حذف دوشرطی
- $\neg(\alpha \wedge \beta) \equiv (\neg \alpha \vee \neg \beta)$ de Morgan توزيع پذيری يا روی و
- $\neg(\alpha \vee \beta) \equiv (\neg \alpha \wedge \neg \beta)$ de Morgan توزيع پذيری يا روی و
- $(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$ distributivity of \wedge over \vee توزيع پذيری يا روی و
- $(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$ distributivity of \vee over \wedge توزيع پذيری يا روی و

اعتبار و قابل ارضاء بودن

یک جمله معتبر است اگر در همه مدلها درست باشد

e.g., True, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

استنتاج و اعتبار بصورت زیر به یکدیگر مرتبط هستند

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ معتبر باشد

حداقل یک مدل

یک جمله قابل ارضاء است اگر در مدلی درست باشد

e.g., $A \vee B$, C

یک جمله غیر قابل ارضاء است اگر در هیچ مدلی درست نباشد

e.g., $A \wedge \neg A$

قابل ارضاء بودن و استنتاج بصورت زیر به یکدیگر مرتبط هستند:

$KB \models \alpha$ if and only if $(KB \wedge \neg \alpha)$ قابل ارضاء نباشد

مسئله قابل ارضا بودن یک جمله در منطق گزاره‌ای، به مسئله SAT معروف است.

قوانین استنتاج

اگه آلفا انگاه بتا
درست باشه
و
آلفا درست باشه
میشه درستی بتا رو
هم نتیجه گرفت

قانون انتزاع :Modus Ponens

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

نتیجه‌ی حاصل از
جملات بالای خط را
زیر خط مینویسیم

حذف و :

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

بالای خط
اگه آلفا و بتا درست باشه
زیرخط

پس آلفا به تنهایی هم درست است

همه هم ارزیهای منطقی

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

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

خلاصه

- نمایش جملات دنیای دیو در منطق گزاره ای
- جدول درستی برای استنتاج
- هم ارزیها
- قوانین استنتاج

پایان



دانشگاه صنعتی اصفهان - مجموعه تالارها

مازیار پالهنج

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

13

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی
عاملین منطقی - ۳
نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنج
آزمایشگاه هوش مصنوعی
دانشکده مهندسی برق و کامپیوتر
دانشگاه صنعتی اصفهان

یاد آوری

- عامل دانش - مبنا
- منطق، ایجاد کردن
- دنیای دیو، اکتشاف در دنیای دیو
- مدلها
- استنتاج: یکی از دو وظیفه
- موثق، کامل
- منطق گزاره ای
- استنتاج با جدول درستی
- معتبر و قابل ارضاء بودن
- قوانین استنتاج
- قانون انتزاع، حذف و، هم ارزیها

قوانين استنتاج

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

$$\frac{\alpha \wedge \beta}{\alpha} \quad \text{حذف و:} \quad ■$$

$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)} \quad \text{همه هم ارزیهای منطقی} \quad ■$$
$$\frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$

مثال برای دنیای دیو

- نشان دادن $\neg P_{1,2}$
- قوانین تا کنون:

قانون اول $\rightarrow R_1 : \neg P_{1,1}$

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

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

$$\begin{array}{l} R_4 : \neg B_{1,1} . \\ R_5 : B_{2,1} . \end{array}$$

مثال برای دنیای دیو

R_2 ■ حذف دو شرطی به

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

R_6 و به ■ حذف و به

اینو که عکس و
نقیض کنیم میرسیم
به
 R_8

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

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

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

هم ارزی ■

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

قانون انزواع R_4 و R_8 ■

نقیض
 $b_{1,1}$
درست است
پس تالی هم درست
است

مثال برای دنیای دیو

دمو رگان ■

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

با حذف و ■

در این خانه گودال
وجود نداره 

جستجو برای استنتاج

■ امکان استفاده از روشهای جستجوی کلاسیک برای یافتن دنباله مراحلی که یک اثبات را شکل می دهند.

■ با تعریف مسئله بصورت:

■ حالت اولیه: پایگاه دانش اولیه

■ اعمال: همه قوانین استنتاج به همه جملاتی که به نیمة بالائی قانون استنتاج منطبق می شوند.

به پایگاه دانش

■ نتیجه Result: اضافه شدن جمله پائین قانون استنتاج

■ هدف: حالتی که شامل جمله ای است که می خواهیم اثبات کنیم.

در روش های استنتاج
جملات بی ربط کمتری
در نظر گرفته میشند

- کارآئی استنتاج بهتر از روش های جستجو،
- جملات بی ربط کمتر در نظر گرفته می شوند.
- خاصیت **یکنواختی**: monotonicity
- با اضافه شدن جملات به پایگاه دانش، جملات ایجاد شده فقط افزوده می شوند.
- در واقع چیزی که قبلاً ایجاد می شده حذف نمی شود.

$$\text{if } KB \models \alpha \text{ then } KB \wedge \beta \models \alpha .$$

اضافه شدن بتا
باعث نمیشند که دیگه آلفا از پایگاه دانش ایجاد نشود
به این خاصیت یکنواختی میگذرد

برای بررسی ایجاب
شدن یه جمله از
جمله ی دیگه باید
قوانین استنتاج کافی
در اختیار مون باشه

- می دانیم قوانین استنتاج موثق هستند.
- ولی اگر قوانین استنتاج کافی نباشند باز نمی توان ایجاب یک جمله را چک نمود.
- بطور مثال اگر قانون حذف دو شرطی وجود نداشت درمثال قبل نمی شد $P_{1,2} \neg$ را نتیجه گرفت.

ما دوست داریم قانون استنتاجی رو داشته
باشیم که هم موثق باشه هم کامل
یه قانون هست به اسم تحلیل که این
خاصیت رو داره

Resolution تحلیل

لیترال: جمله اتمی یا نقیض جمله اتمی

کلاوز Clause: فصل لیترالها

شکل عادی عطفی (Conjunctive Normal Form - CNF)

عطف کلاوزها

فصل دوتا لیترال

$$\text{E.g. } (A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$$

کلاوز

لیترال

نقیض یکی هم ارز
دیگری است

از جمله های بالایی
این نتیجه میشے که
همه ای لیترال ها باهم
فصل بشن
و
Li باید
از تو ش حذف بشه

قانون تحلیل تک (CNF) (حالت Unit Resolution)

$$\frac{\ell_1 \vee \dots \vee \ell_k, \quad m}{\ell_1 \vee \dots \vee \ell_{i-1} \vee \ell_{i+1} \vee \dots \vee \ell_k}$$

- که ℓ_i و m لیترالهای مکمل هستند

قانون تحلیل حالت کلی (CNF) (حالت Resolution)

$$\frac{\ell_1 \vee \dots \vee \ell_k, \quad m_1 \vee \dots \vee m_n}{\ell_1 \vee \dots \vee \ell_{i-1} \vee \ell_{i+1} \vee \dots \vee \ell_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n}$$

- که ℓ_i و m_j لیترالهای مکمل هستند.

مازیار پالهنجک

فصل لیترال های
جمله ای اول و لیترال
های جمله ای دوم

از جمله ای اول
Li
حذف میشے

از جمله ای دوم باید
mj
حذف شه

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

■ بطور مثال در دنیای دیو:

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

دو تا لیترال بالایی که
دورشون مستطیل
قرمز کشیدم حذف
میشه و این جمله که
فصل لیترال های باقی
مانده است ایجاد
میشه

اعمال قانون تحلیل یا
رزولوشن به این
دوجمله بالایی

نقیض این لیترال
توی جمله ی دوم
او مده
پس جمله ی پایین
ایجاد میشه

برای اینکه بتوانیم از قانون تحلیل استفاده کنیم باید جملات به

فرم
cnf

باشد یعنی بشه عطف کلاوز ها درحالی که خود کلاوز ها
فصل لیترال ها باشن

CNF به تبدیل

حذف دوشرطی: جایگزینی $\alpha \leftrightarrow \beta$ با $(\neg \alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$ ■

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

حذف شرطی: جایگزینی $\neg \alpha \vee \beta$ با $\alpha \Rightarrow \beta$ ■

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

بردن نقیض به داخل با استفاده از نقیض دوگانه و دمورگان: ■

$$\neg(\neg \alpha) \equiv \alpha \quad (\text{double-negation elimination})$$

$$\neg(\alpha \wedge \beta) \equiv (\neg \alpha \vee \neg \beta) \quad (\text{De Morgan})$$

$$\neg(\alpha \vee \beta) \equiv (\neg \alpha \wedge \neg \beta) \quad (\text{De Morgan})$$

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

▪ توزیع \vee روی \wedge :

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

- اگر در کلاوزی از یک لیترال چند نسخه وجود داشت فقط یکی نگاه داشته می شود.
- به این عمل **فاکتورگیری factoring** گفته می شود.

propositional
logic
منطق گزاره ای

میخایم نشون بدیم آیا الف از پایگاه
دانش ایجاد میشه یا نه
که معادل با اینه که بگیم عطف
پایگاه دانش با نقیض آلفا قابل ارضاء
نیست

Resolution

الگوریتم

اثبات با تناقض (برهان خلف)، نشان دهد که $KB \wedge \neg\alpha$ قابل ارضاء نیست. ■

Figure 7.13

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

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

$new \leftarrow \{\}$

while true **do**

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

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

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

$new \leftarrow new \cup resolvents$

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

$clauses \leftarrow clauses \cup new$

جملاتی که با اعمال
رزولوشن ایجاد
میشوند.

اول نقیض آلفا را به
پایگاه دانش اضافه
میکنیم
بعد به فرم
cnf
درمیاریم

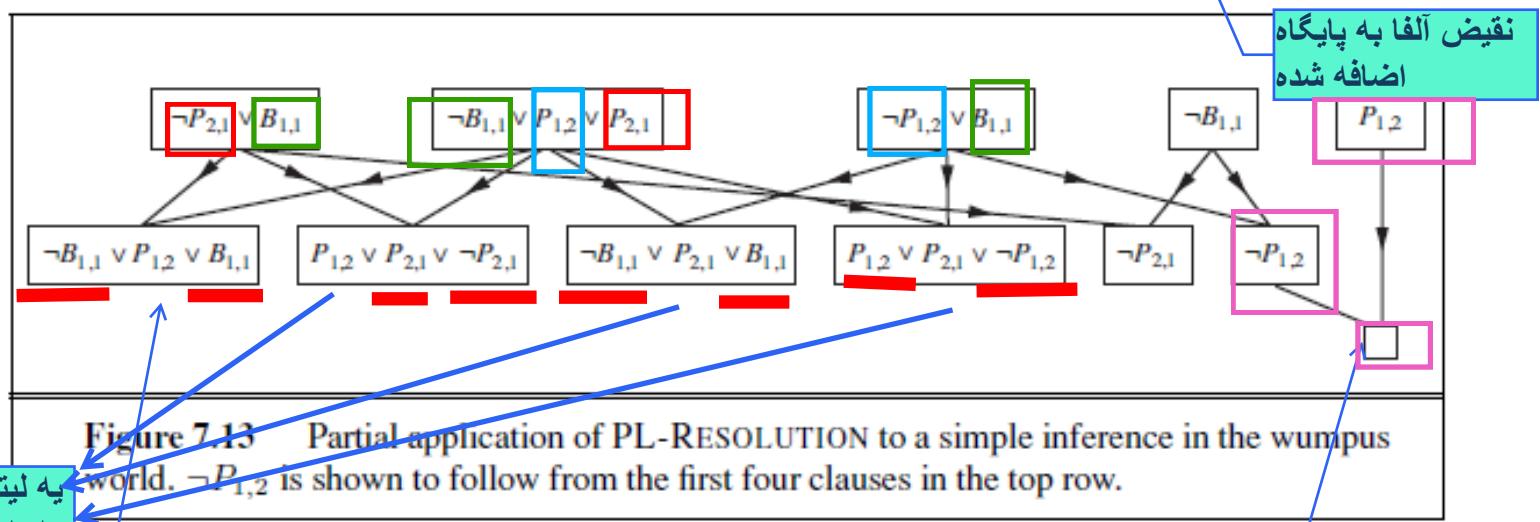
A simple resolution algorithm for propositional logic. PL-RESOLVE returns the set of all possible clauses obtained by resolving its two inputs.

15

اگه به کلاوز تهی
بررسیم یعنی آلفا از
پایگاه دانش ایجاد
میشه

مثال Resolution

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



- تحلیل یک روال استنتاج موثر و کامل است.
- همیشه به توان کامل آن نیاز نیست.
- بسیاری از پایگاههای دانش محدودیتهایی روی جملاتی که استفاده می کنند دارند،
- این باعث می شود که بتوان الگوریتمهای استنتاج محدودتر و کارآتری استفاده نمود.

پس کلاوز معین یک
کلاوز هرن حتما
هست

■ کلاوز معین Definite clause مثبت

■ بطور مثال ($\neg L_{1,1} \vee \neg Breeze \vee B_{1,1}$)

■ کلاوز هرن Horn Clause مثبت

■ کلاوز بدون لیترال مثبت کلاوز هدف Goal Clause

■ کلاوزهای هرن تحت تحلیل بسته هستند.

■ اگر تحلیل را روی دو کلاوز هرن اعمال کنیم حاصل یک کلاوز هرن خواهد شد.

یعنی اگه قانون تحلیل را روی دو تا کلاوز هرن استفاده کنیم نتیجه یه کلاوز هرن دیگه است.

$CNFSentence \rightarrow Clause_1 \wedge \dots \wedge Clause_n$
 $Clause \rightarrow Literal_1 \vee \dots \vee Literal_m$
 $Fact \rightarrow Symbol$
 $Literal \rightarrow Symbol \mid \neg Symbol$
 $Symbol \rightarrow P \mid Q \mid R \mid \dots$
 $HornClauseForm \rightarrow DefiniteClauseForm \mid GoalClauseForm$
 $DefiniteClauseForm \rightarrow Fact \mid (Symbol_1 \wedge \dots \wedge Symbol_l) \Rightarrow Symbol$
 $GoalClauseForm \rightarrow (Symbol_1 \wedge \dots \wedge Symbol_l) \Rightarrow False$

به مقدم شرط بدن
به تالی شرط سر
گفته میشه

- پایگاههای دانش به صورت کلاوزهای معین جالب هستند چون:
- یک کلاوز معین قابل تبدیل به فرم شرطی است که در آن مقدم عطف لیترالهای مثبت و تالی یک لیترال مثبت است.

یک کلاوز معین

بدنه و سر

مثال $(\neg L_{1,1} \vee \neg \text{Breeze}) \vee B_{1,1}$

تبدیل به $(\neg(L_{1,1} \wedge \text{Breeze})) \vee B_{1,1}$

تبدیل به $(L_{1,1} \wedge \text{Breeze}) \Rightarrow B_{1,1}$

در کو جملات به صورت شرطی ساده تر است.

استفاده از قانون
دمورگان

اگه عامل در خانه‌ی یک و یک
باشه و در کو ما نسیم باشه
در خانه یک و یک نسیم داریم

بیان واقعیت به
صورت شرطی

- واقعیت: یک کلاوز فقط با یک لیترال مثبت و بدون لیترال منفی ■
- مانند: $L_{1,1}$ ■
- می تواند به صورت $True \Rightarrow L_{1,1}$ نوشته شود. ■
- کلاوز بدون لیترال مثبت را می توان بصورت یک شرطی با سر
نوشت $False$ ■
- $(W_{1,1} \wedge W_{1,2}) \Rightarrow False$ هم ارز است با $\neg W_{1,1} \vee \neg W_{1,2}$ ■

- استنتاج با کلاوزهای هرن را می توان بصورت زنجیربندی به جلو یا زنجیربندی به عقب انجام داد.
- تعیین ایجاب کردن از روی کلاوزهای هرن در زمانی بصورت خطی نسبت به اندازه پایگاه دانش قابل انجام است.

خلاصه

- نمایش جملات دنیای دیو در منطق گزاره ای
- جدول درستی برای استنتاج
- هم ارزیها
- قوانین استنتاج
- جستجو برای استنتاج
- قانون Resolution
- تبدیل جملات بصورت CNF
- عدم نیاز به توان کامل تحلیل
- کلاوزهای معین و هرن

پایان



دانشگاه صنعتی اصفهان - مجموعه تالارها

مازیار پالهنج

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

24

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی

عاملین منطقی - ۴

نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پاله‌نگ

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

- عامل دانش - مبنا
- منطق، ایجاب کردن
- دنیای دیو، اکتشاف در دنیای دیو
- مدلها، استنتاج
- منطق گزاره ای
- استنتاج با جدول درستی
- معتبر و قابل ارضاء بودن
- قوانین استنتاج
- قانون انتزاع، حذف و، هم ارزیها
- تحلیل (resolution)
- تبدیل به شکل اصلی عطفی
- الگوریتم تحلیل
- مثال تحلیل در شکل اصلی عطفی
- پایگاه دانش به شکل کلاوز معین

- استنتاج با کلاوزهای هرن را می توان بصورت زنجیربندی به جلو یا زنجیربندی به عقب انجام داد.
- تعیین ایجاب کردن از روی کلاوزهای هرن در زمانی بصورت خطی نسبت به اندازه پایگاه دانش قابل انجام است.

به صورت ضمنی ما فرض میکنیم بین این جملات عطف است

جملاتی که توى پایگاه دانش داریم به این دو صورت هستند

زنجیر بندی به جلو و عقب

در شکل هرن پایگاه دانش عطف کلاوزهای هرن است.

- نماد گزاره ای، یا
- عطف نمادها \Leftarrow نماد

E.g., $C \wedge (B \Rightarrow A) \wedge (C \wedge D \Rightarrow B)$

قانون انتزاع (برای شکل هرن) کامل است.

اگه پایگاه دانش بخاد درست باشه همه ی این جملات باید درست باشند

$$\alpha_1, \dots, \alpha_n, \quad \alpha_1 \wedge \dots \wedge \alpha_n \Rightarrow \beta$$

β

تک تک اینها را میدانیم درست است اگه عطفشون را هم بنویسیم درست میشه جمله ی حاصل شده مازیار پالهنهک

الان مقدم این شرط درست میشه پس درست بودن تالی نتیجه گرفته میشه

می تواند با زنجیر بندی به جلو و عقب پیاده سازی شود.
در زمان خطی با اندازه پایگاه دانش اجرامی شوند

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

قانون انتزاع را با زنجیربندی به جلو و عقب میشه پیاده سازی کرد

قانون انتزاع میگفت اگه بدانیم $a \Rightarrow b$

درست است و بدانیم که a درست است میشه درست بودن بی را نتیجه بگیریم

4

یعنی از مقدم یه قانون میشه تالی اون قانون را نتیجه گرفت بعدش میتونیم اون تالی ک بدمست اومنده را به پایگاه دانش اضافه کنیم اگه تالی بدمست اومنده همون چیزیه که میخاستیم توقف میکنیم اگر نه هم ادامه میدیم و بازم چک میکنیم ایا چیز جدیدی میتونه به پایگاه دانش اضافه شه؟

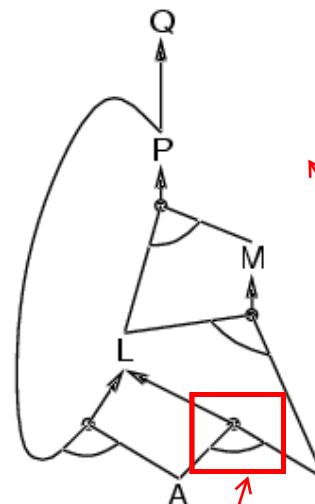
زنجیر بندی به جلو

- هر قانونی که مؤلفه های مقدمش در KB بدمست آمنده آتش می شود،
- تالی آن به KB اضافه شده و اگر به جواب رسیده ایم توقف می کنیم.

پایگاه دانش ما این جملات است.

واقعیت های یا فکت های A, B

$$\begin{aligned}
 P &\Rightarrow Q \\
 L \wedge M &\Rightarrow P \\
 B \wedge L &\Rightarrow M \\
 A \wedge P &\Rightarrow L \\
 A \wedge B &\Rightarrow L \\
 A \\
 B
 \end{aligned}$$



AND-OR
GRAPH

L
حاصل از عطف
 A, B
است

FORWARD CHAINING

زنجریندی به جلو

آیا کیو از پایگاه
دانش ایجاد میشه؟

کیو یک نماد گزاره
است

الگوریتم زنجریندی به جلو

```

function PL-FC-ENTAILS?(KB, q) returns true or false
  inputs: KB, the knowledge base, a set of propositional definite clauses
          q, the query, a proposition symbol
  count  $\leftarrow$  a table, where  $count[c]$  is the number of symbols in c's premise
  inferred  $\leftarrow$  a table, where  $inferred[s]$  is initially false for all symbols
  agenda  $\leftarrow$  a queue of symbols, initially symbols known to be true in KB

  while agenda is not empty do
    p  $\leftarrow$  POP(agenda)
    if  $p = q$  then return true
    if  $inferred[p] = \text{false}$  then
       $inferred[p] \leftarrow \text{true}$ 
      for each clause c in KB where p is in c.PREMISE do
        decrement  $count[c]$ 
        if  $count[c] = 0$  then add c.CONCLUSION to agenda
  return false

```

تعداد نماد های مقدم
سی
اگه سی به فرم
شرطی نوشته شده
باشه

جملات در پایگاه
دانش به صورت
کلاوز معین

پی توی چه
کلاوزهایی از پایگاه
دانش در مقدمشون
وجود داره؟ برو تمام
جملاتی که پی در
مقدمشان هست را
پیداکن

آیا نماد اس تاحلا
استنتاج شده؟
ایا درستیش را بدست
وردیم تاحلا؟
اول کار برای همه ی
نماد ها فالس است

یه صف است از
نمادها که درابتدا
کار دران نمادهایی از
پایگاه دانش که
میدانیم درست است
قرارمیگیره

زنجریندی به جلو موثر و کامل برای کلاوزهای هرن می باشد.

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

آیا تا الان درستی پی
را بدست اوردیم؟

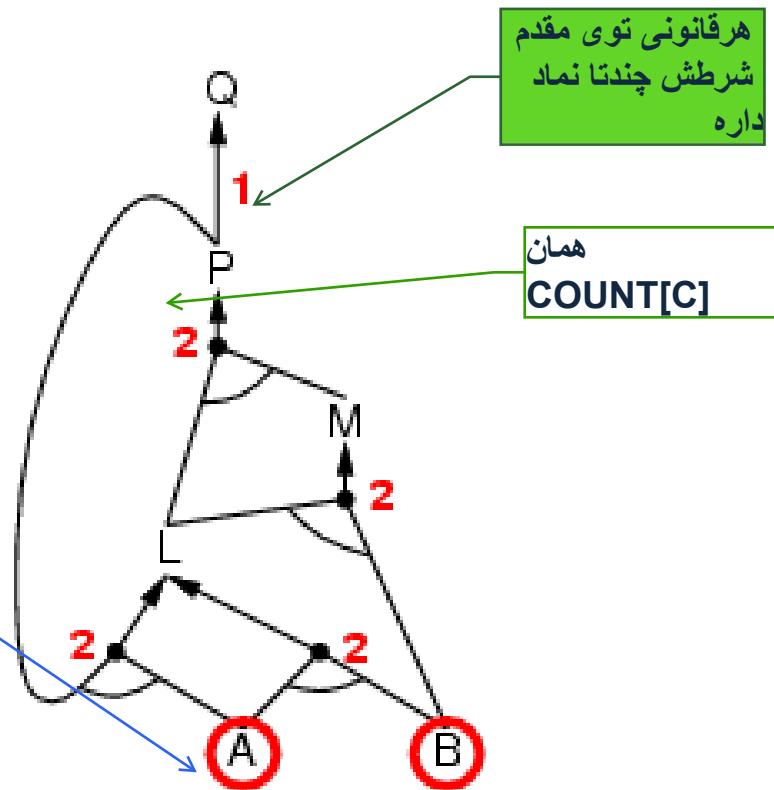
ما داریم زنجریندی به جلو انجام میدیم پس
شروع کارمون باید تعدادی نماد در پایگاه
باشه که بینیم درست است یا نه؟

اگه به صفر رسیده یعنی متوجه شدیم
که تمام مولفه های مقدم قانونمنون را
متوجه شدیم که درست است پس
میتوانیم تالی قانون را نتیجه بگیریم و
به صف دستور جلسه اضافه کنیم

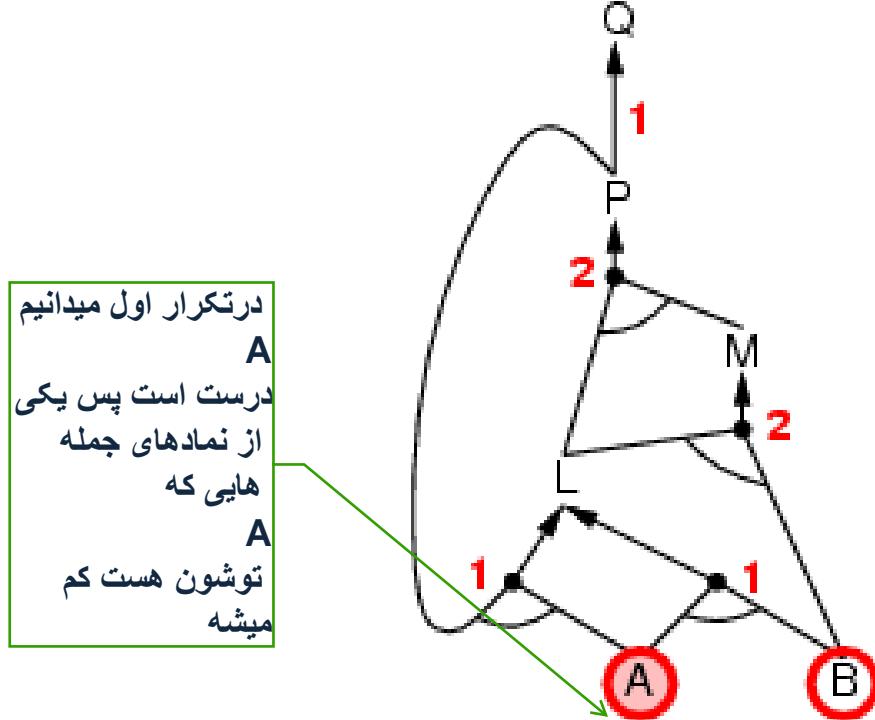
هر تالی میتوانه باعث بشه که یه نماد دیگه
ای نتیجه گیری بشه پرای همین بهش
زنجیربندی به جلو میگیم

مثال زنجیر بندی به جلو

ابتدا چیزهایی که
میشه بگیم درست
هستند همان واقعیت
های پایگاه دانش
هستند که
A,B
است

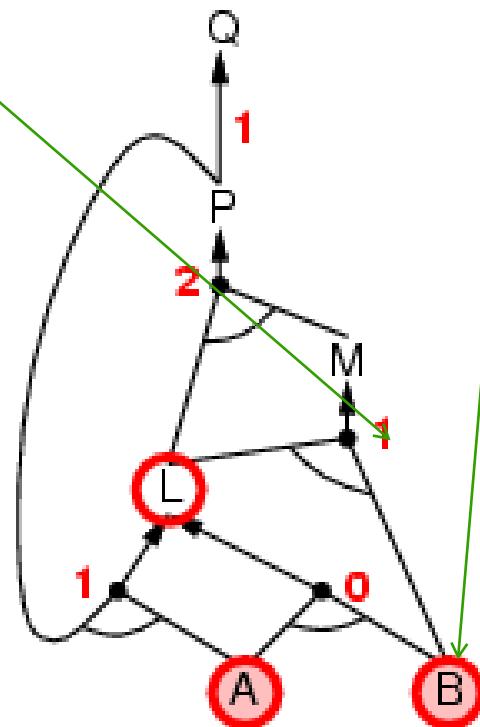


مثال زنجیر بندی به جلو



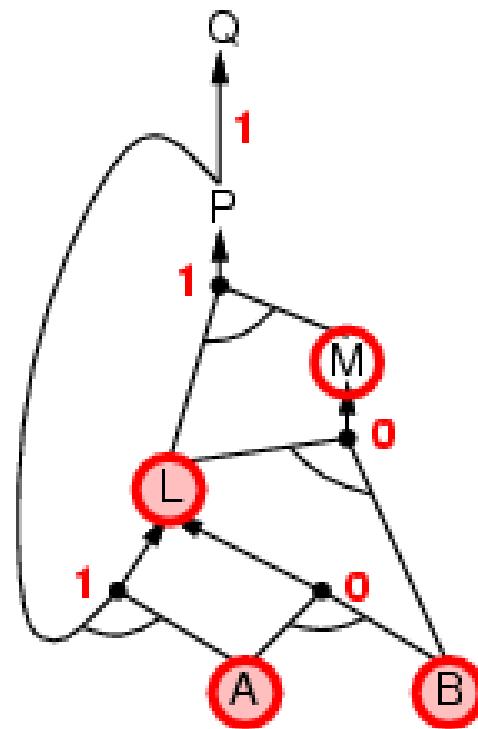
مثال زنجیر بندی به جلو

که اینجا بود ۲
به خاطر بی
یکی ازش کم شد

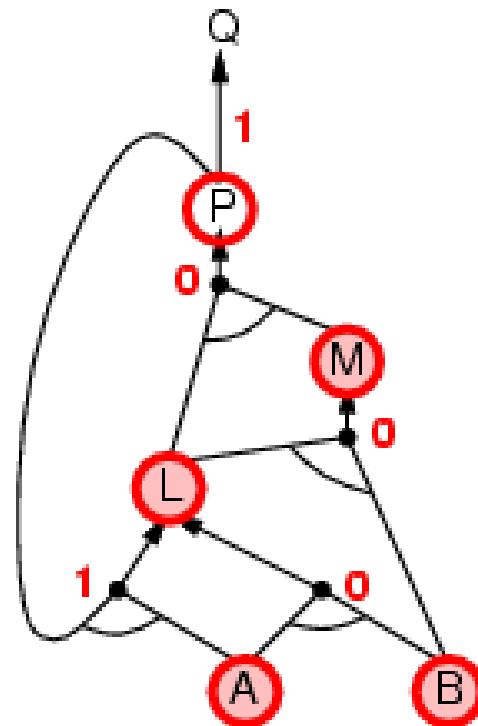


در تکرار بعدی از درست بودن بی
نتیجه میگیریم که باید یکی از نمادهای
جمله هایی ک تنوی مقدمشان بی بوده
کم کنیم
پس میشه اون جمله ای که
COUNT[C]=0
نتیجه بگیریم که تالیش درست است
پس تالی به دستور جلسه ما اضافه
میشه

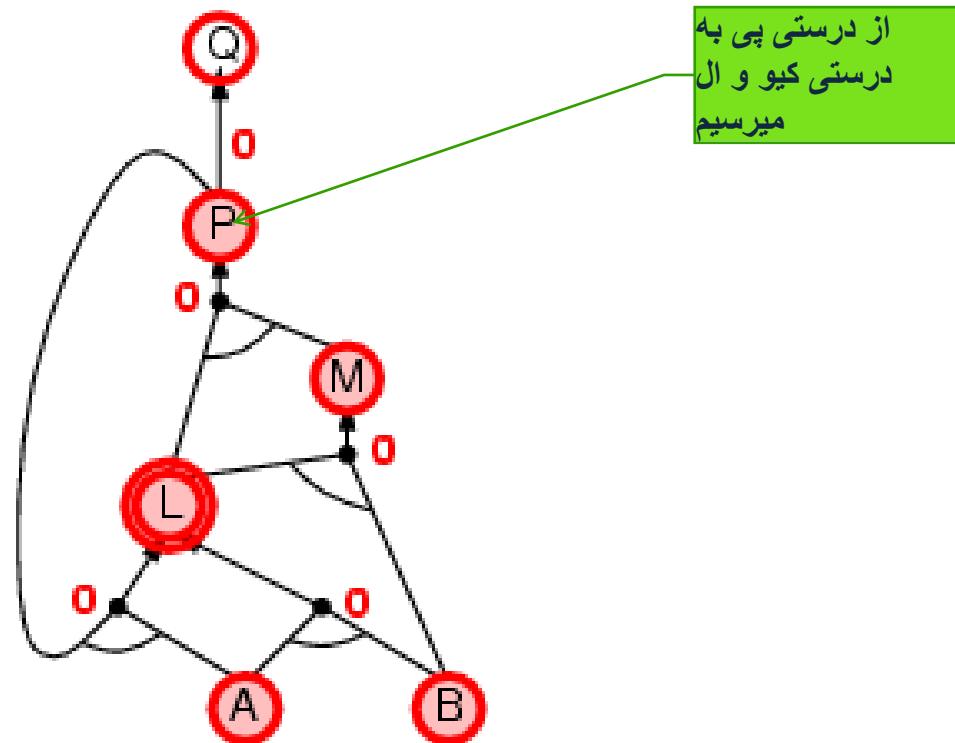
مثال زنجیر بندی به جلو



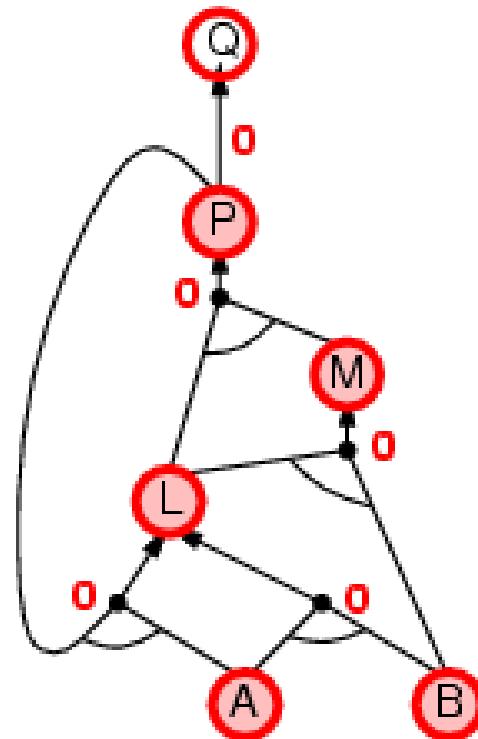
مثال زنجیر بندی به جلو



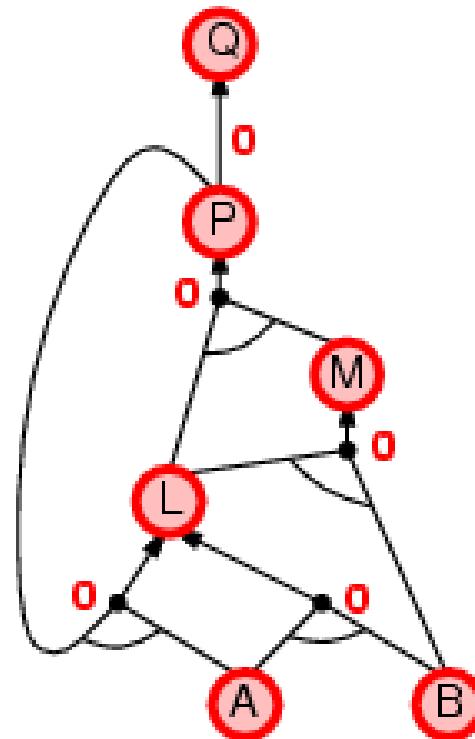
مثال زنجیر بندی به جلو



مثال زنجیر بندی به جلو



مثال زنجیر بندی به جلو



مبنای زنجیربندی به جلو قانون
انتزاع است و میدانیم قانون
انتزاع موثق است پس
زنجیربندی ب جلو هم موثق
است

اثبات کامل بودن

- ز.ج. چون اعمال قانون انتزاع است، بنابر این یک روال استنتاج موثق است.
- کامل بودن آن را باید بررسی کنیم.

هر کلاوز معینی که در پایگاه اولیه داریم توانی مدل آمایی که بش رسانیدیم درست است

اثبات کامل بودن

ز.ج. همه اتمهایی که از KB ایجاد می شوند را تولید می کند.

ز.ج. به نقطه ثابتی می رسد که دیگر اتمی قابل بدست آمدن نیست.
(و صعیت نهائی جدول (Inferred

حالت نهائی را به عنوان مدل m که مقادیر درست/نادرست به نمادها
انتساب شده در نظر بگیرید.

هر کلاوز معینی در KB اولیه در m درست است.

اگر اینکونه نباشد کلاوزی مثل $a_1 \wedge \dots \wedge a_k \Rightarrow b$ وجود دارد
نادرست است.

یعنی مقدم درست و تالی نادرست است

این در تناقض است که الگوریتم به نقطه ثابتی رسانیده است.

بنابر این m یک مدل KB است.

اگر $q, KB \models q$, q باید در هر مدل KB درست باشد.

بنابر این در m نیز درست است.

بنابر این هر جمله ایجاد شونده Q توسط الگوریتم استنتاج می شود.

.1

.2

.3

.4

.5

.6

.7

.8

.9

.01

مازیار پالهنج

هوش مصنوعی - نیمسال دوم ۱۴۰۰-۰۱

اگه چیزی قرار باشه
از جملات پایگاه
دانش ایجاد بشه
ایجاد خواهد شد

16

16

- ز.ج. در یک عامل می تواند در هنگام ورود یک درک استفاده شود.

با ورود این درک آیا میشه استنتاج های دیگه ای
هم داشته باشیم؟
یه دانشی تازه وارد شده چه چیزهایی میشه ازش
نتیجه گرفت؟ اطلاعات و نتیجه گیری های تازه

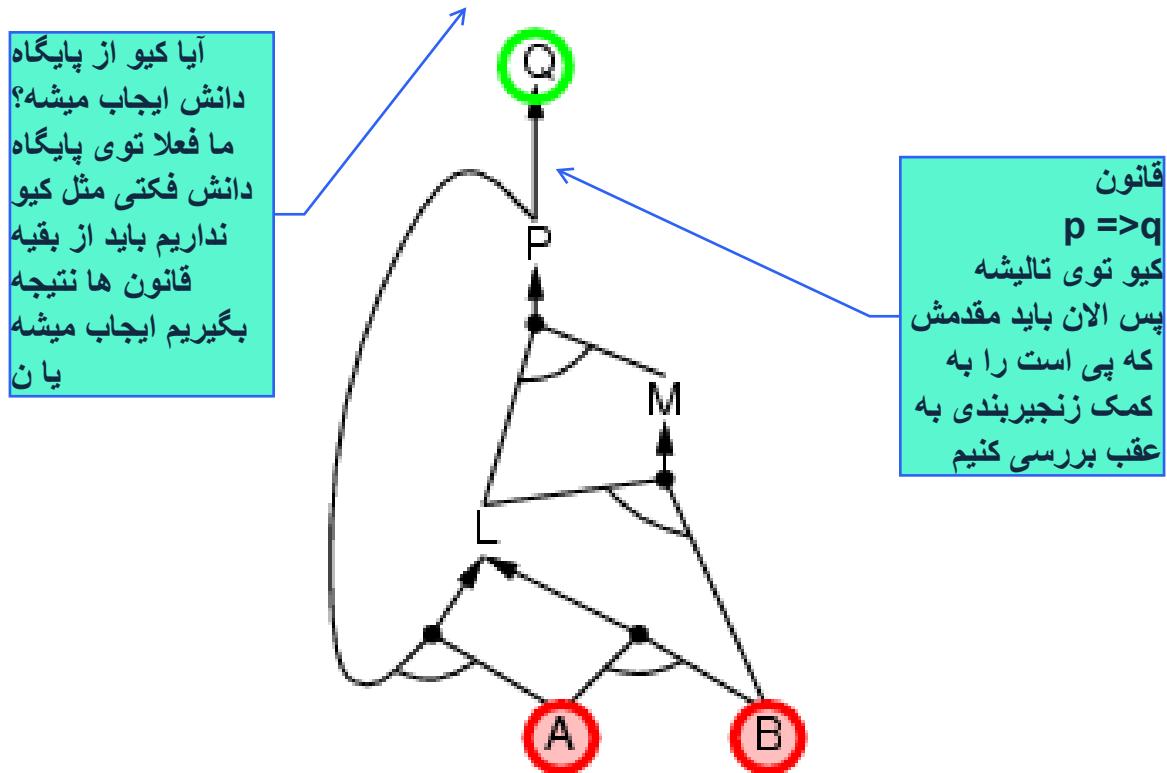
- Tell در
- در انسان تحت کنترل است

زنجبندی به عقب

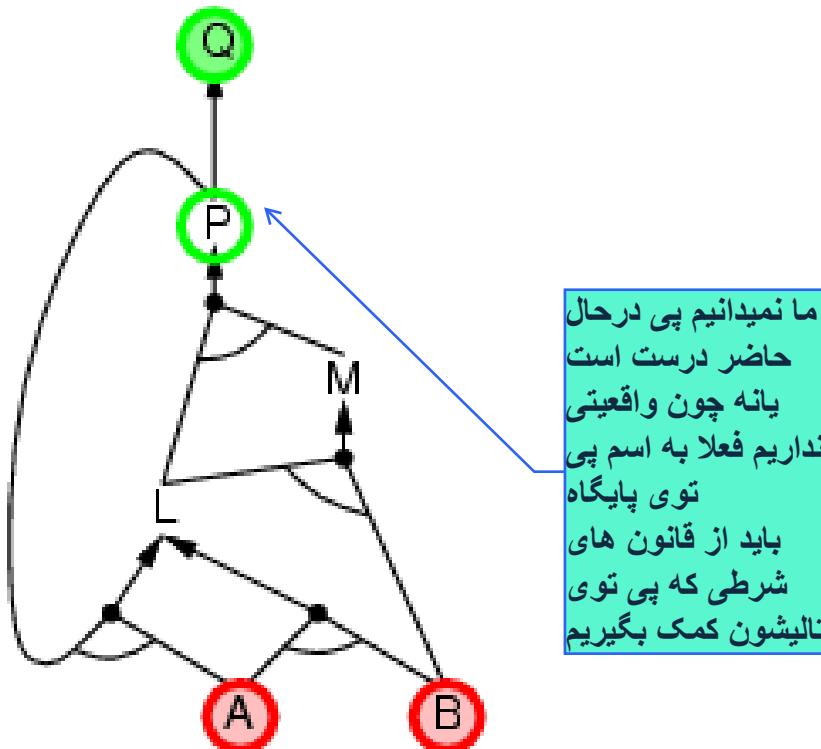
حرکت به سوی عقب از پرسش ۹

- بررسی این که ۹ هم اکنون درست است یا نه، یا
- بوسیله ز.ع. همه مؤلفه های مقدم قانونی که ۹ را به عنوان تالی دارد
- بررسی می کنیم.

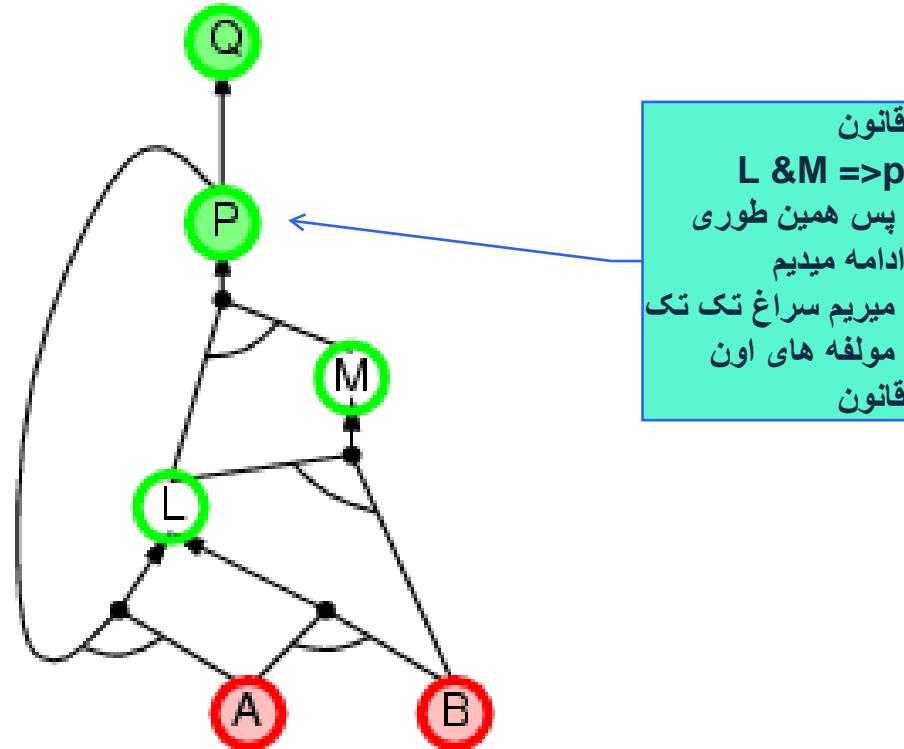
مثال زنجیربندی به عقب



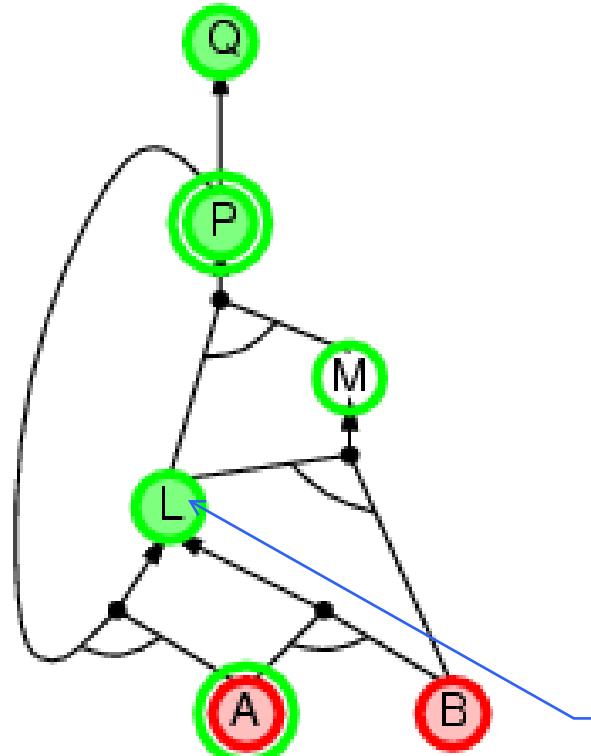
مثال زنجیربندی به عقب



مثال زنجیربندی به عقب



مثال زنجیربندی به عقب

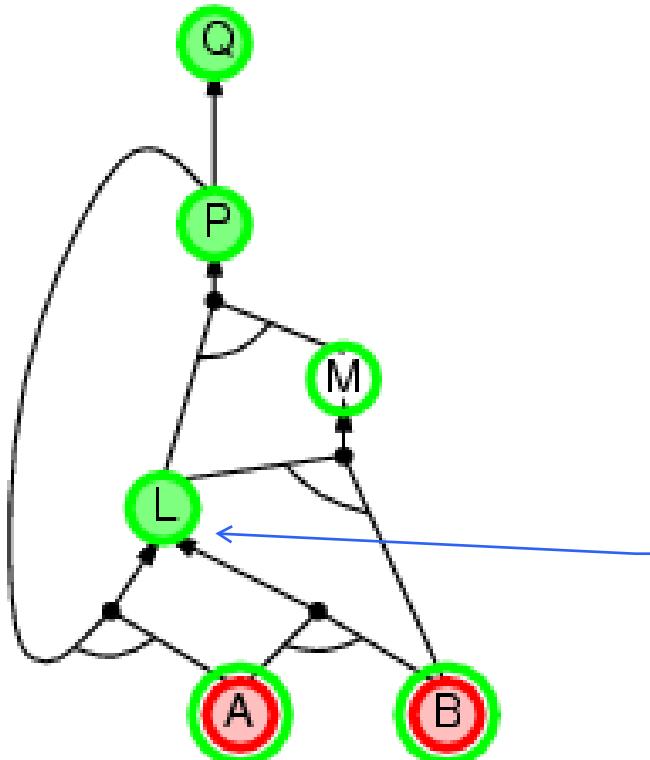


هوش مصنوعی - نیمسال دوم ۱۴۰۰-۰۱

مازیار پالهنجک

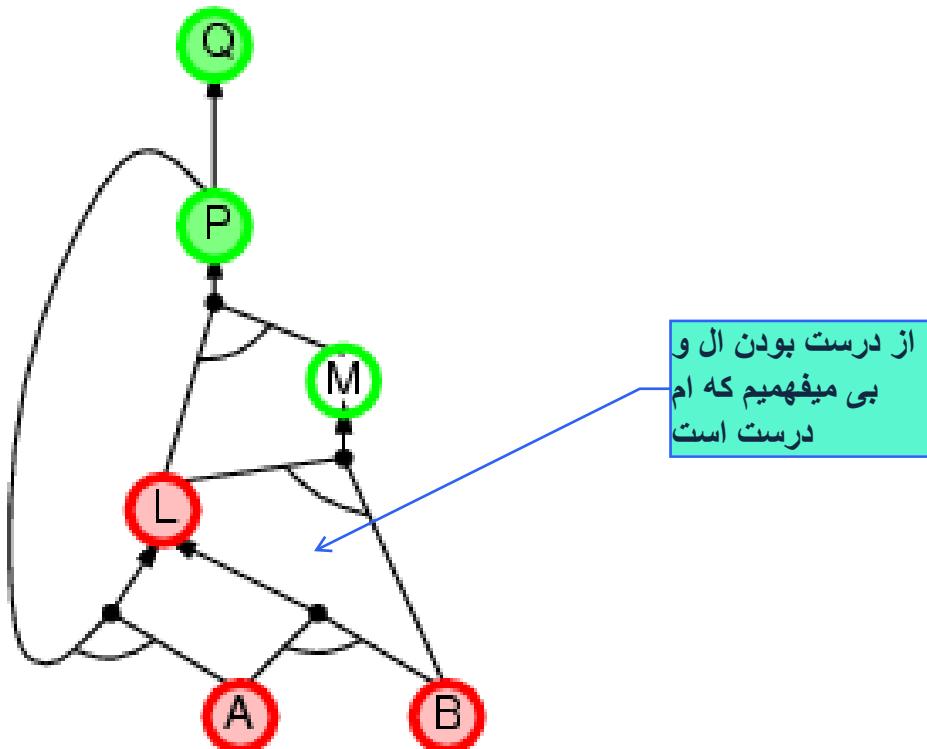
میخایم ال را بررسی
کنیم ولی نمیدوئیم
درست است یا نه؟
میریم سراغ قانونی
که ال توى تالیش
باشه به
A
میرسیم که میدوئیم
درست است چون
فکتمونه
حالا مثل استک میریم
جلو

مثال زنجیربندی به عقب

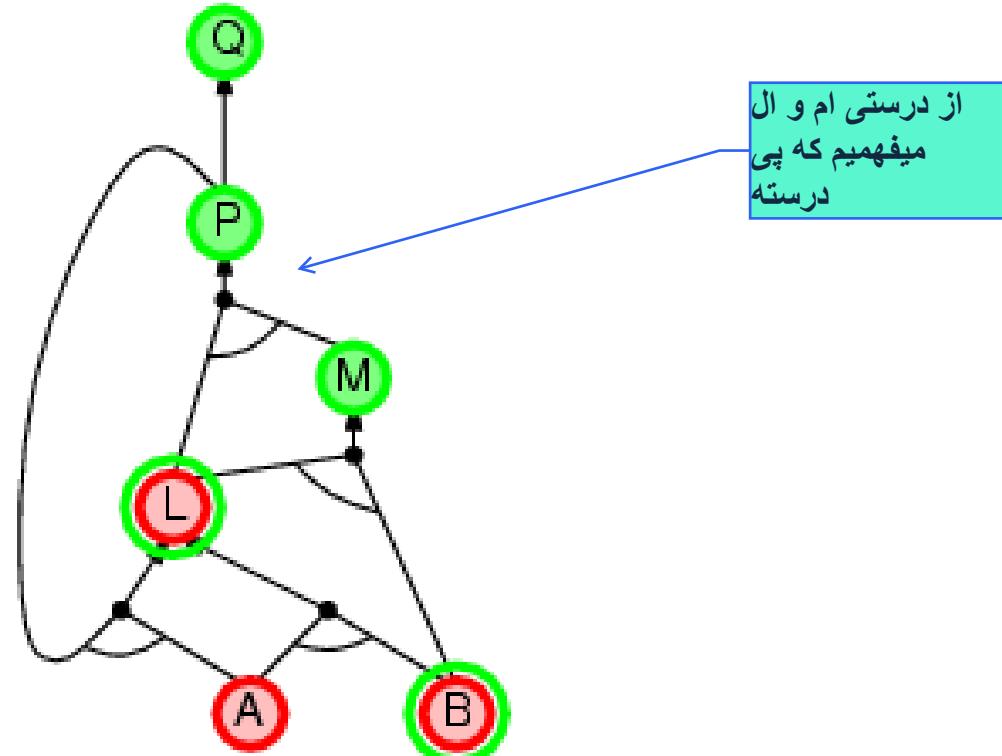


الآن مقدم این قانون
که ال را میده برای
هر دومولفه درست
شد پس میتوئیم
تالیش را نتیجه
بگیریم که تالیش
همان ال است پس ال
درسته

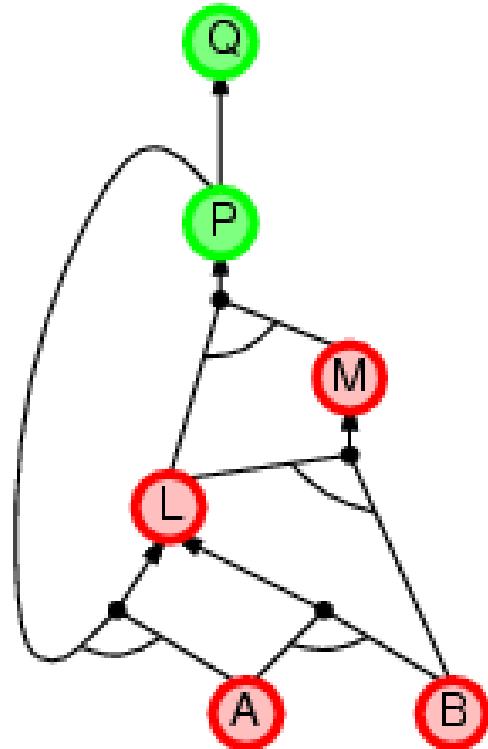
مثال زنجیربندی به عقب



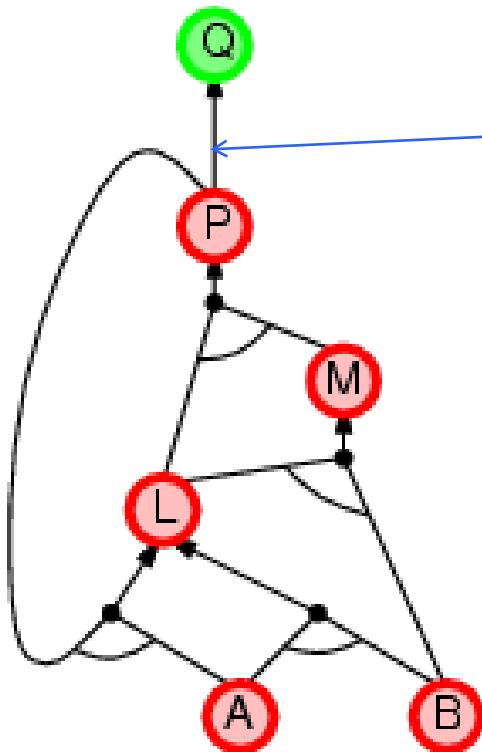
مثال زنجیربندی به عقب



مثال زنجیربندی به عقب

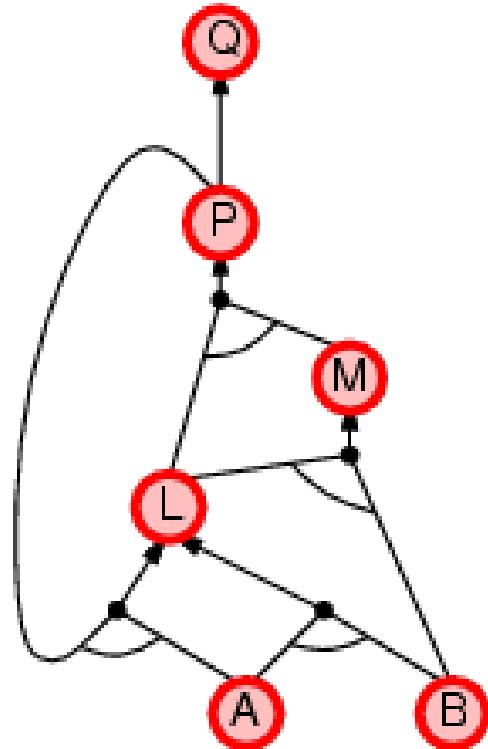


مثال زنجیربندی به عقب



از درست بودن مقدم
قانون پی انگاه کیو
میفهمیم که کیو
درسته

مثال زنجیربندی به عقب



استنتاج کارآ در منطق گزاره ای

- چک مدل در منطق گزاره ای همانند مسائل ارضاء محدودیتها
- جستجوی عقبگرد
- الگوریتم DPLL (Davis, Putnam, Logemann, Loveland)
- جستجوی محلی
- الگوریتم WalkSat

چون جمله ها از
عطف یه تعداد کلاوز
ایجاد میشوند

DPLL الگوریتم

- استفاده از چند مکاشفه برای بهبود چک کامل مدلها در فهرست کردن جدول درستی بهبودها:

اولین مکاشفه

چون یه کلاوز از
فصل یه تعداد لیترال
ایجاد شده

خاتمه زودهنگام

.1

یک کلاوز درست است اگر لیترالی از آن درست باشد
یک جمله نادرست است اگر کلاوزی از آن نادرست باشد

مکاشفه نماد خالص

.2

نماد خالص همیشه با یک علامت در کلاوزها ظاهر می شود
مثال: در $(A \vee \neg B), (\neg B \vee \neg C), (C \vee A)$ ، A و B خالص بوده ولی C نه

لیترال متناظر با نماد خالص را درست کن

(چون اگر جمله مدلی داشته باشد با درست کردن آن نماد خالص کلاوز نادرست نمی شود.)

مکاشفه کلاوز تک

.3

کلاوز تک فقط شامل یک لیترال است
آن تنها لیترال را درست کن

DPLL الگوریتم

```
function DPLL-SATISFIABLE?(s) returns true or false
  inputs: s, a sentence in propositional logic
```

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

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

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

این الگوریتم چون هر بار یه نماد را میگیره و مقدار میده و به صورت بازگشتی خودش را صدا میزنه و میره سراغ نمادهای بعدی به صورت عمقی داره جستجو میکنه
چون هر دفعه یه متغیر را میگیره و پایین میره توی درخت یک جستجوی عقبگرد است

مثل ارضای محدودیت ها یه تعدادی نماد هست که باید مقادیرشون مشخص بشه هدف ما هم اینه که بفهمیم جمله‌ی اس که ورودی است ارضا پذیر هست یا نه

نمادهای استفاده شده
در اس

آیا اس قابل ارضا
است؟
اس یه جمله در
منطق گزاره ای است

الگوریتم DPLL

```
function DPLL-SATISFIABLE?(s) returns true or false
  inputs: s, a sentence in propositional logic
  clauses  $\leftarrow$  the set of clauses in the CNF representation of s
  symbols  $\leftarrow$  a list of the proposition symbols in s
  return DPLL(clauses, symbols, {})
```

جملاتی که در منطق
گزاره ای داده شده را
ب تعدادی کلاؤز
تبدیل میکنیم که به
فرم cnf
هستند

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

```
  if every clause in clauses is true in model then return true
  if some clause in clauses is false in model then return false
  P, value  $\leftarrow$  FIND-PURE SYMBOL(symbols, clauses, model)
  if P is non-null then return DPLL(clauses, symbols - P, model  $\cup$  {P=value})
  P, value  $\leftarrow$  FIND-UNIT-CLAUSE(clauses, model)
  if P is non-null then return DPLL(clauses, symbols - P, model  $\cup$  {P=value})
```

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

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

جستجو برای
پیدا کردن نماد خالص

مازیار پالهنجک

کلاؤز تک را پیدا کن

۱۴۰۱-۰۲ نیمسال اول

از ابتدای نماد ها یه
نماد را بیرون
میکشیم و توی پی
میریزیم

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲
مقداری که برای پی
پیداشده رابه
مجموعه هی مدلمان
اضافه کن

در یکی از این حالت ها
ممکن است یه مدل پیدا کنیم
که کلاؤز هامون توی اون
مدل درست باشه چ با
p=true
این مدل درست برگردانه و
چه با فالس چون اور کردیم
باهم

استفاده از جستجوی محلی

- شروع از یک انتساب به همه متغیرها،
 ■ سعی در بهبود آن،
- با احتمال p انجام یک گام تصادفی،
- سعی در تغییر مقدار متغیری که تعداد کلاوزهای ارضا شده را
 ■ بیشینه کند.

اگه تعداد ماکس
دفعات تکرار تمام شد
و مدلی پیدا نشد که
همه را ارضاء کنه
شکست برگردان

پی احتمال است
که گام تصادفی است

حداکثر چندتا تغییر را
میتوانیم انجام بدیم؟

الگوریتم WalkSAT

Figure 7.18

```
function WALKSAT(clauses, p, max_flips) returns a satisfying model or failure
    inputs: clauses, a set of clauses in propositional logic
            p, the probability of choosing to do a “random walk” move, typically around 0.5
            max_flips, number of value flips allowed before giving up

    model  $\leftarrow$  a random assignment of true/false to the symbols in clauses
    for each i = 1 to max_flips do
        if model satisfies clauses then return model
        clause  $\leftarrow$  a randomly selected clause from clauses that is false in model
        if RANDOM(0, 1)  $\leq p$  then
            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
```

The WALKSAT algorithm for checking satisfiability by randomly flipping the values of variables. Many versions of the algorithm exist.

مازیار پالنگ

با احتمال پی میخایم
یه گام تصادفی
برداریم

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

مقدار یه سمبل توی
کلاوز را به صورت
تصادفی فلیپ کن اگه
فالسه ترو اگه تروهه
فالس میشه

34

- زنجیربندی به جلو و عقب
- چک مدل همانند مسئله ارضاe محدودیتها
- الگوریتم عقبگرد DPLL
- الگوریتم محلی WalkSat

پایان



دانشگاه صنعتی اصفهان - مجموعه تالارها

مازیار پالهنگ

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

36

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی

عاملین منطقی - ۵

نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

- عامل دانش - مبنا
- منطق، ایجاد کردن
- دنیای دیو، اکتشاف در دنیای دیو
- مدلها، استنتاج
- منطق گزاره ای
- استنتاج با جدول درستی
- معابر و قابل ارضابودن
- قوانین استنتاج
- قانون انتراع، حذف و، هم ارزیها
- تحلیل (resolution)
- تبدیل به شکل عادی عطفی
- الگوریتم تحلیل
- مثال تحلیل در شکل عادی عطفی
- پایگاه دانش به شکل کلاوز معین
- زنجیر بندی به جلو و عقب
- چک مدل همانند مسئله ارضاء محدودیتها
- الگوریتم عقبگرد DPLL
- الگوریتم محلی WalkSat

تعریف گزاره ها به
کمک گزاره های
منطقی

گزاره ها در دنیای دیو

این دو تا چیز را برای
همه خانه های
دنیای دیو تعریف
کنیم ما ۱۶ تا خانه
داریم

در این خانه گودال
نیست
توش دیو هم نیست

■ دانش اولیه

بیان معنای نسیم دار
بودن
بیان معنای بودار
بودن در یک خانه

■ و برای هر خانه

در خانه های
مجاورش یکی شون
دیو باشه

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

در یکی از خانه های
مجاورش گودال باشه

■ و اینکه فقط یک دیو وجود دارد:

این جمله میگه یه
دیوی توی این دنیا
هست

■ تعداد زیادی متغیر گزاره ای

مشکل این جمله به تنها یی اینه که
اگه توی دوتا خانه دیو باشه هم
درسته
اگه توی سه تا خانه دیو باشه
درسته به همین ترتیب

این زمانی نادرست
میشه که توی هر دو تا
خانه دیو باشه

$$W_{1,1} \vee W_{1,2} \vee \dots \vee W_{4,3} \vee W_{4,4} \\ \neg W_{1,1} \vee \neg W_{1,2} \\ \neg W_{1,1} \vee \neg W_{1,3} \\ \dots \\ \neg W_{4,3} \vee \neg W_{4,4} .$$

مازیار پالهنجک

- نیمسال اول ۱۴۰۱-۰۲

3

دوتا گزاره ای متفاوت

گزاره ها در دنیای دیو

- اگر حال Stench و قبلًاً Stench

چون قراره این دوتا را به پایگاه دانش اضافه کنیم باید زمانشون رو معلوم کنیم

احتیاج به معین کردن زمان

- مثالاً Stench³ و Stench⁴

و برای هر چیزی که با زمان تغییر کند.

متغیرهایی که جنبه های دائم محیط را نشان داده و به زمان وابسته نیستند **متغیرهای بدون زمان atemporal variables** نامیده می شوند.

می توان نسیم و بوی بد را پس از طریق متغیرهای

$$\begin{aligned} L_{x,y}^t &\Rightarrow (Breeze^t \Leftrightarrow B_{x,y}) \\ L_{x,y}^t &\Rightarrow (Stench^t \Leftrightarrow S_{x,y}) \end{aligned}$$

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

4

درک از ورودی در
لحظه‌ی تی

عامل در زمان تی در
خانه‌ی ایکس و واي
پاشه

مکان ایکس و واي
نسیم دار است

4

گزاره ها در دنیای دیو

اگه عامل ب سمت
شرق بره در زمان
صفر

در زمان صفر به
سمت جلو بره

عامل در زمان یک
در خانه ۱۹۱
است و در زمان یک
در خانه ۱۹۱
نیست

نیاز effect axioms

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

- برای نمایش اثر اعمال به اصول اثربخش است.

داره میگه اگه عمل
فوروارد رو انجام
بدیم چه اتفاق هایی
میفته

$$\text{ASK}(KB, L_{2,1}^1) = \text{true.}$$

- برای هر واحد زمانی، هر خانه، و هر یک از ۴ جهت و دیگر اعمال چنین اصولی باید نوشته شود.

$$\text{ASK}(KB, HaveArrow^1)$$

- حال می توان سؤال کرد:

- چه اتفاقی می افتد اگر سؤال شود:

آیا عامل در زمان
یک تیر داره؟

مثلثاً أَكَهُ كَارِي انجام
نَشَدَهُ كَهْ تَيْرَهَا كَمْ
بَشَهْ مَثَلًا عَالِمْ فَقَطْ
رَفْتَهْ جَلَوْ پَسْ بَايْدْ
تَيْرَهَاشْ سَرْجَاشْ
بَاشَهْ دِيَگْهْ

گزاره ها در دنیای دیو

باید بگیم بعد از
اینکه یه سری اعمال
انجام شد اون اعمالی
که تغییر نکردن را
باید یادت باشه وی
سری اعمال تغییر
نمیکنن

- پاسخ false خواهد بود.

نیاز است که اعلام کنیم پس از انجام برخی از اعمال، برخی امور
تغییر نمی کنند.

اصولی که بعد از اینکه
یه سری از اعمال
انجام میشن تغییری
نمیکنند

- نیاز به اصول قاب frame axioms

$$\begin{aligned} Forward^t &\Rightarrow (HaveArrow^t \Leftrightarrow HaveArrow^{t+1}) \\ Forward^t &\Rightarrow (WumpusAlive^t \Leftrightarrow WumpusAlive^{t+1}) \end{aligned}$$

مازیار پالهنجک

اگه در زمان تی عمل فوروارد انجام بشه اگه
در زمان تی تیر داشته باشیم در زمان تی
هوش بعلاوه یک هم تیر داریم
اگه در زمان تی دیو داشته باشیم در زمان تی
بعلاوه یک هم دیو داریم

عمل فوروارد

6

اف در زمان تی
بعلاوه یک
برقراره

عملی که باعث
درست شدن نات
 F_t
بشه انجام نشده باشه
ینی عملی انجام نشه
که اف تی از بین بره

را راه ها در دنیای دیو

successors-state

■ یا راه حل استفاده از **اصول حالت تالی**:
axioms

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

یه عملی باعث شده
باشه این اف ایجاد
بشه

■ مثال:

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

■ چون فرض شده دوباره تیر بدست نمی آورد.

اف در زمان تی
درست بوده و عملی
انجام نشده که باعث
از بین رفتاین اف
بشه

مصنوعی - نیمسال اول ۱۴۰۱-۰۲

عامل در زمان تی تیر داشته باشه
و عمل شوت کردن را در زمان تی
انجام نداده باشه
اگر و تنها اگر
عامل در زمان تی بعلاوه یک تیر
داشته باشه

بخاره به دیوار در
زمان تی بعلاوه یک

■ اصل حالت تالی برای مکان عامل:

$$L_{1,1}^{t+1} \Leftrightarrow (L_{1,1}^t \wedge (\neg Forward^t \vee Bump^{t+1}))$$

$$\vee (L_{1,2}^t \wedge (FacingSouth^t \wedge Forward^t))$$

$$\vee (L_{2,1}^t \wedge (FacingWest^t \wedge Forward^t)).$$

عامل از خانه ی یک و دو میره توی خانه ی یک و یک در زمان تی بعلاوه یک در خانه ی جدید است

■ نیاز به دانستن امن بودن خانه ها:

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

اینطوری نباشه که
درخانه ی ایکسو
اوای دیو باشه و اون
دیوه زنده باشه هو

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

درک هایی که عامل
گرفته در زمان های
 مختلف

گزاره ها در دنیای دیو

برای وضعیت:

$\neg Stench^0 \wedge \neg Breeze^0 \wedge \neg Glitter^0 \wedge \neg Bump^0 \wedge \neg Scream^0 ; Forward^0$ زمان صفر

$\neg Stench^1 \wedge Breeze^1 \wedge \neg Glitter^1 \wedge \neg Bump^1 \wedge \neg Scream^1 ; TurnRight^1$

$\neg Stench^2 \wedge Breeze^2 \wedge \neg Glitter^2 \wedge \neg Bump^2 \wedge \neg Scream^2 ; TurnRight^2$

$\neg Stench^3 \wedge Breeze^3 \wedge \neg Glitter^3 \wedge \neg Bump^3 \wedge \neg Scream^3 ; Forward^3$

$\neg Stench^4 \wedge \neg Breeze^4 \wedge \neg Glitter^4 \wedge \neg Bump^4 \wedge \neg Scream^4 ; TurnRight^4$

$\neg Stench^5 \wedge \neg Breeze^5 \wedge \neg Glitter^5 \wedge \neg Bump^5 \wedge \neg Scream^5 ; Forward^5$ زمان ۵

$Stench^6 \wedge \neg Breeze^6 \wedge \neg Glitter^6 \wedge \neg Bump^6 \wedge \neg Scream^6$

حال داریم.

استنتاج هایی که با
منطق گزاره ای قابل
فهم هستند

$(KB, L_{1,2}^0) = true,$

$ASK(KB, W_{1,3}) = true$

$ASK(KB, P_{3,1}) = true,$

مازیار پالهنجک

آیا عامل در زمان ۶
درخانه‌ی یک و دو
است؟ اگه از پایگاه
بپرسیم باید جواب
درست بده

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

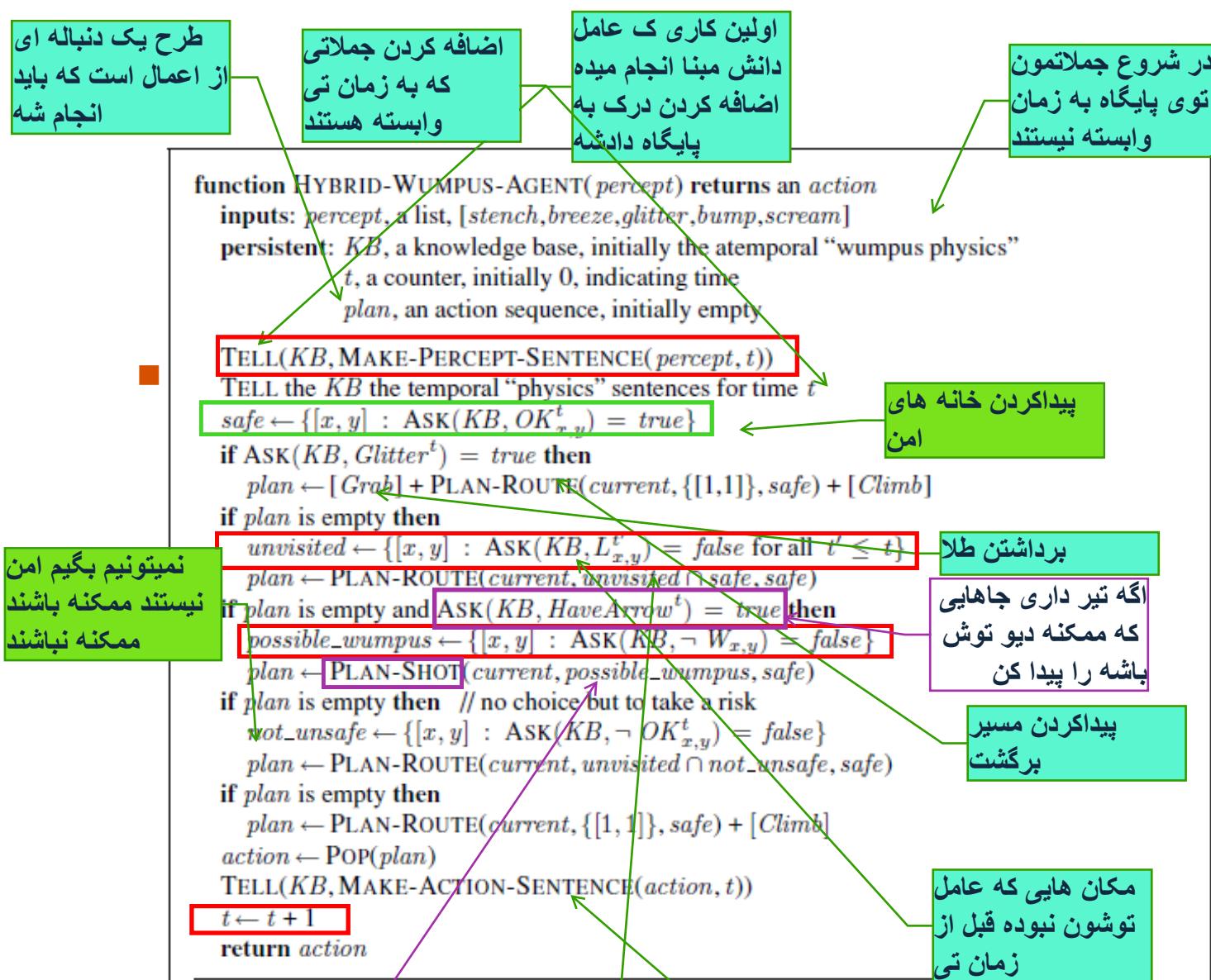
آیا در خانه‌ی ۱ و ۳
دیو هست؟ اره

آیا درخانه‌ی ۱ و ۳
گودال هست؟ اره

استفاده از استنتاج و
قوانين اگر انگاه و
الگوریتم های
جستجو

یک عامل ترکیبی

- با داشتن جملات کافی در پایگاه دانش می توان سوالات بسیاری را پرسید.
- توانائی استنتاج جنبه های مختلف حالت محیط، به همراه
- یک سری قوانین شرط-عمل، و
- الگوریتمهای جستجو،
- توانائی ایجاد یک عامل ترکیبی hybrid agent برای دنیای دیو را مهیا می سازد.

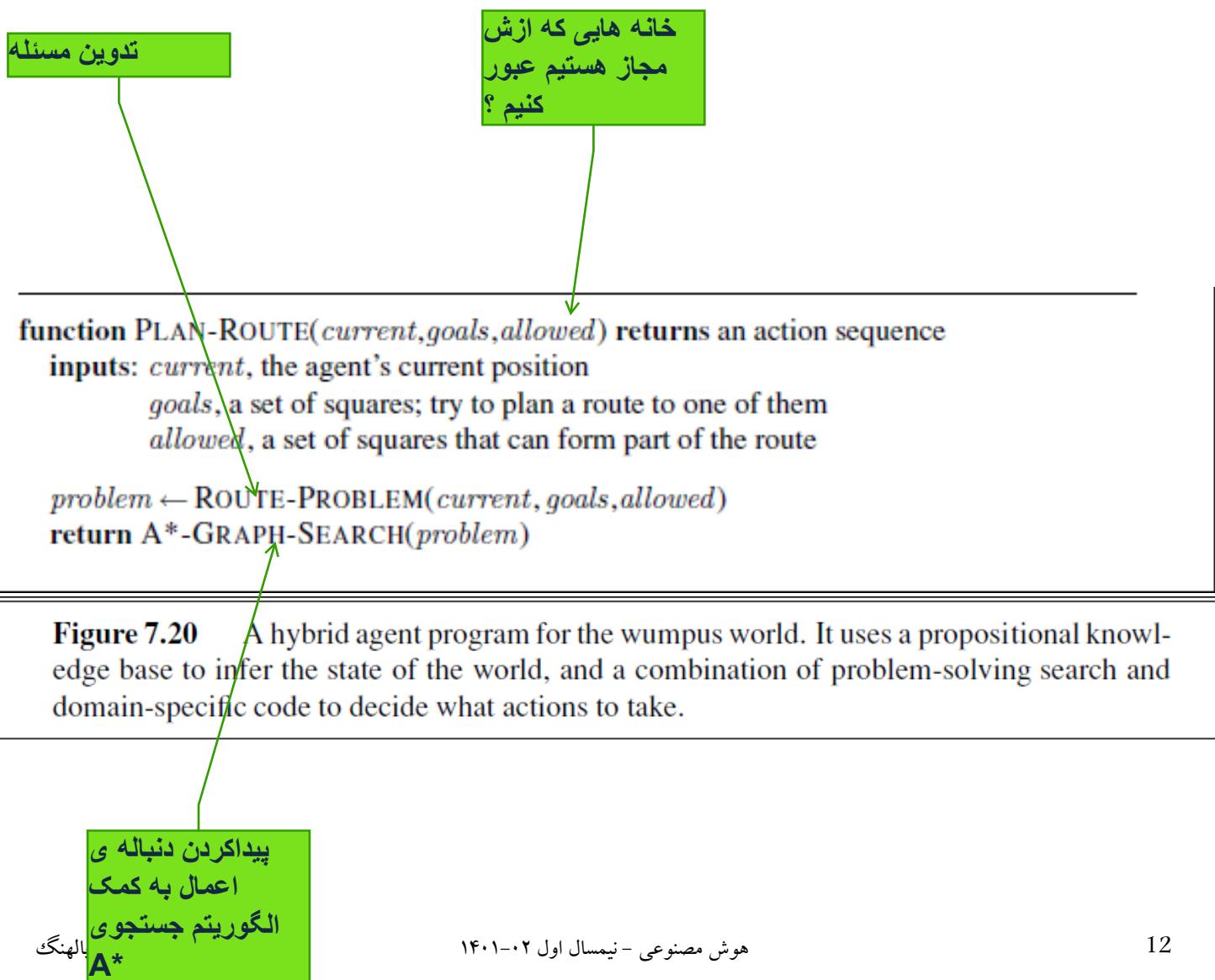


عاملی که ریسک‌پذیر است
میره سراغ خانه هایی که
صدق صدق مطمئن نیست که
امن هستند

از نزدیکترین خانه
ایی که ندیدی و امن
است یکی را انتخاب
کن

پایگاه دانش میگیم
ما این عمل را در
زمان تی انجام دادیم

۱۴۰۱-۰۲ نیمسال اول مصنوعی وش



- جملات در دنیای دیو
- اصول اثری
- اصول قاب
- اصول حالت تالی
- یک عامل ترکیبی

والسلام



دانشگاه صنعتی اصفهان – مجموعه تالارها

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

14

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی
منطق مرتبه اول - ۱
نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنجک
آزمایشگاه هوش مصنوعی
دانشکده مهندسی برق و کامپیوتر
دانشگاه صنعتی اصفهان

کوچکترین جزئی در منطق گزاره ای همان گزاره است

ولی توى منطق مرتبه اول ميشه به اشیا و روابط و توابع تبدیل کرد

منطق مرتبه اول

منطق گزاره ای فرض می کند دنیا شامل واقعیتها است.

منطق مرتبه اول فرض می کند دنیا شامل:

■ **اشیاء**: مردم، خانه ها، امین، رنگها، کشورها، اعداد، ...

■ **روابط**: برادر، بزرگتر از، درون، قسمتی از، دارای رنگ، دربر دارنده، ..

■ روابط یکتائی (خواص): قرمز، گرد، اول، ...

■ **توابع**: پدر، بهترین دوست، یکی بیش از، ...

منطق مرتبه اول

■ **مثال:** یک به اضافه دو برابر است با سه

■ **اشیاء:** یک، دو، سه، یک به اضافه دو

■ **رابطه:** برابر است با

■ **تابع:** به اضافه

یه اسمی برای شی

■ **مثال:** مربعهای مجاور دیو بودار هستند

■ **اشیاء:** دیو، مربع

■ **خاصیت:** بوداری

■ **رابطه:** مجاور بودن

یک رابطه‌ی یکتاوی

مسند تتها

دستور

Sentence → *AtomicSentence* | *ComplexSentence*

AtomicSentence → *Predicate* | *Predicate(Term, ...)* | *Term = Term*

ComplexSentence → (*Sentence*)

| \neg *Sentence*

| *Sentence* \wedge *Sentence*

| *Sentence* \vee *Sentence*

| *Sentence* \Rightarrow *Sentenee*

| *Sentence* \Leftrightarrow *Sentence*

| *Quantifier Variable, ... Sentence*

كمى ساز

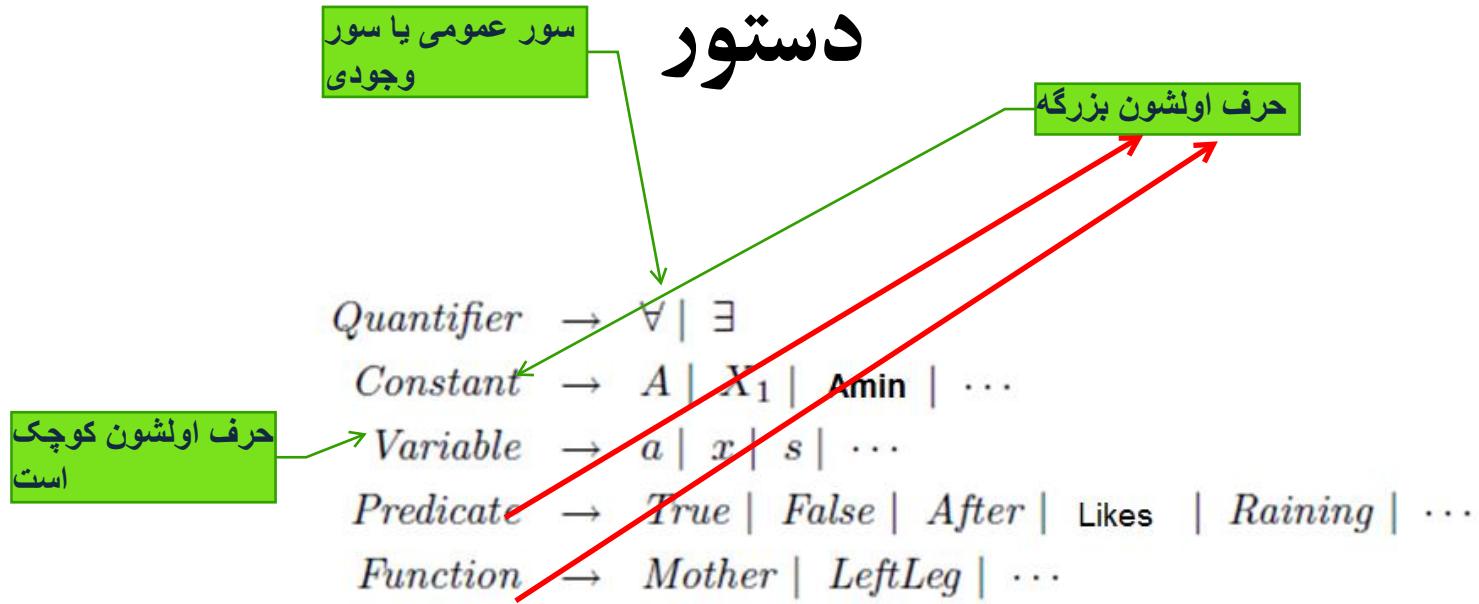
جمله‌ی مرکب

Term → *Function(Term, ...)*

| *Constant*

| *Variable*

دستور

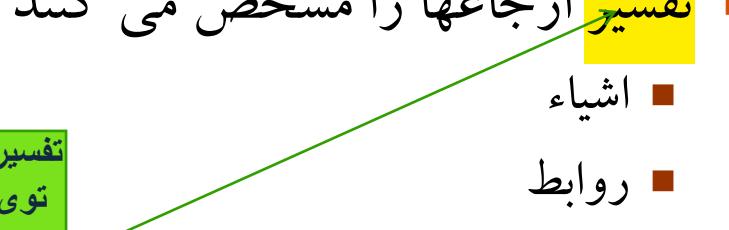


OPERATOR PRECEDENCE : $\neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow$

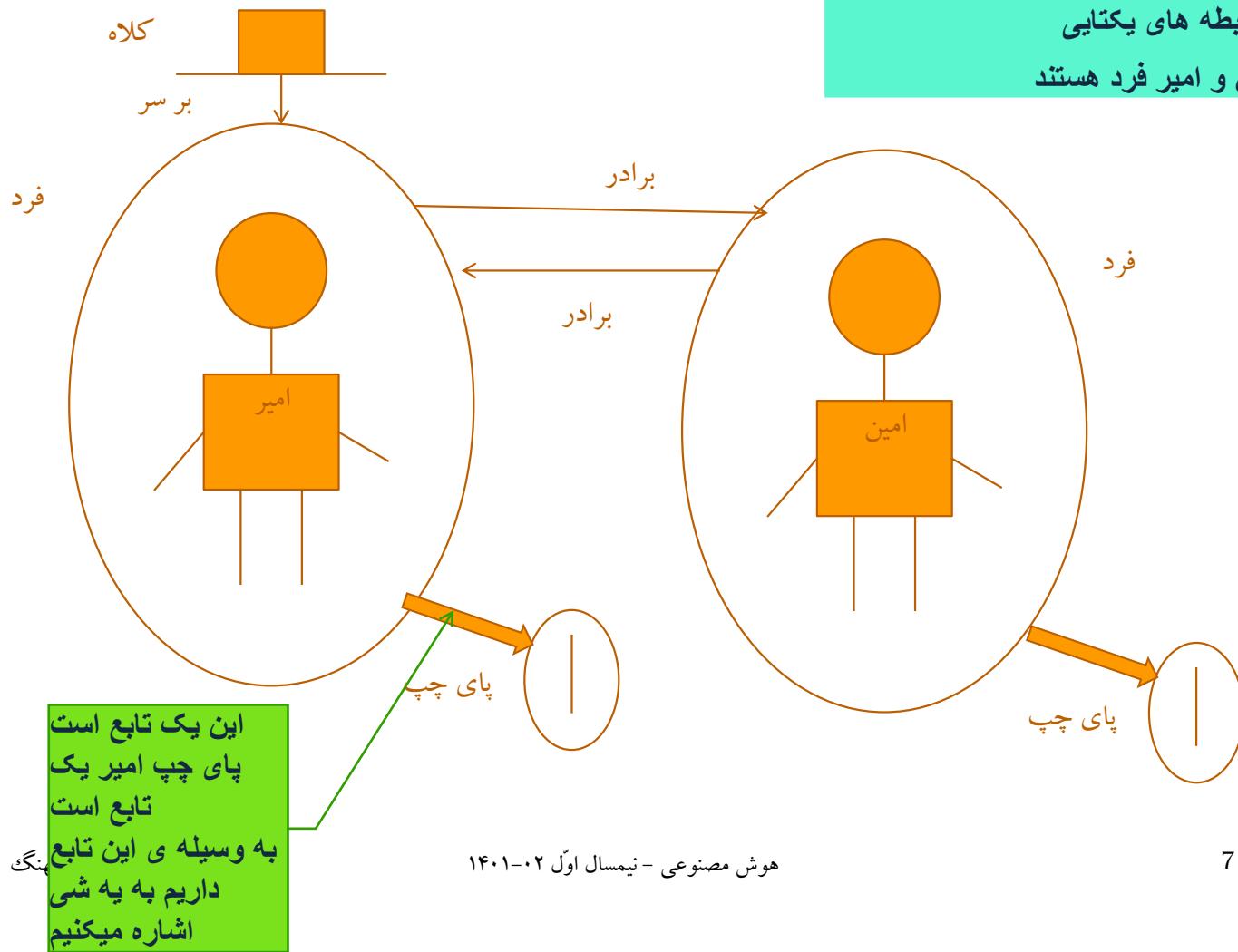
مازیار پالهنجک

اولویت بین عملگرها
از چپ به راست کم میشه

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

- جملات درست هستند نسبت به یک مدل و یک تفسیر
 - مدل شامل اشیاء (عناصر دامنه)، و روابط بین آنها است.
 - تفسیر ارجاعها را مشخص می کند برای
- اشیاء ■ روابط ■ توابع
- تفسیر میگه مثلًا اشیا
توی اون مدل به چه
چیزی رجوع میکنند
و توابع و روابط هم
همین طور
- 

شکل زیر یه دنیایی است که یه تعدادی
شی داره
روابط: امین برادر امیر است و برعکس
کلاه پرسر امیر است
وابطه های یکتایی
امین و امیر فرد هستند



- نمادهای ثابت: A، B، C، امین، ...
- تفسیر مشخص می کند چه شیئی در دنیا توسط هر ثابت اشاره می شود.
- هر نماد ثابت دقیقاً یک شیء
- همه اشیاء لازم نیست دارای نام باشند.
- برخی ممکن است نامهای متعددی داشته باشند.

- نمادهای مسندی: برادر، خواهر، ...
- تفسیر مشخص می کند نماد مسند به چه رابطه ای اشاره می کند
- در هر مدل داده شده، رابطه توسط مجموعه ای از چند تائیهای از اشیاء که آنرا ارضا می کند تعریف می شود
- یک چند تائی Tuple: مجموعه ای از اشیا که به ترتیب خاصی کنار هم قرار گرفته اند و توسط <> احاطه شده اند.
- رابطه برادری: {<امین، امیر>, <زینب، امین>}
- برادر یک مسند دوتائی

مازیار پالهنجک

امین برادر زینب
است
ولی زینب برادر امین
از زو ما نیست!!!!

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

9

■ نمادهای تابعی: پدر، پای چپ، سینوس، ...



والسلام

م. پالهنج
مازیار پالهنج

دانشگاه صنعتی اصفهان - پژوهشکده فاوا

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

11

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی
منطق مرتبه اول - ۲
نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنگ
آزمایشگاه هوش مصنوعی
دانشکده مهندسی برق و کامپیوتر
دانشگاه صنعتی اصفهان

منطق مرتبه اول

- منطق گزاره ای فرض می کند دنیا شامل واقعیتها است.
- منطق مرتبه اول فرض می کند دنیا شامل:
 - اشیاء: مردم، خانه ها، امین، رنگها، کشورها، اعداد، ...
 - روابط: برادر، بزرگتر از، درون، قسمتی از، دارای رنگ، دربر دارنده، ..
 - روابط یکتائی (خواص): قرمز، گرد، اول، ...
 - توابع: پدر، بهترین دوست، یکی بیش از، ...

یادآوری

- در درس ساختمانهای گستته:
- رابطه (دو تائی) R یک مجموعه از زوجهای مرتب است که می تواند زیرمجموعه ای از ضرب دکارتی دو مجموعه باشد:

$$R \subseteq A \times B$$

- مثلاً رابطه کوچکتر بودن:

$$<= \{ <1,2>, <2,3>, <3,4>, \dots \}$$

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

3

$Z * Z$
در مجموعه اعداد
صحیح تعریف شده

ترتیب مهم است

یادآوری

- می توان آن را بصورت زیر نمایش داد:

$<(1,2)$

LessThan(1,2)

روش های نمایش کوچکتر
بودن

$<(2,3)$

LessThan(2,3)

$<(3,4)$

LessThan(3,4)

...

...

- یا گاهی برای سادگی بصورت:

$1 < 2$

$2 < 3$

$3 < 4$

...

یادآوری

- به همین صورت رابطه n تائی R یک مجموعه از n تاییهای مرتب است که زیرمجموعه‌ای از ضرب دکارتی n مجموعه است.
- در واقع در اینجا از نماد مسند برای نمایش روابط استفاده می‌کنیم.

ان تا مجموعه را
باهم ضرب دکارتی
کنیم
اعضای ان تایی های
مرتب به ترتیب از
این ان تا مجموعه
هستند ترتیب
قرارگیری اشیا مهم
هست

ضرب دکارتی

طبق تعریف که برای
تابع داشتیم این
رابطه‌ی مربع یک
تابع است

یادآوری

- یک تابع (دو تائی) هم یک رابطه خاص بود که دامنه اش همه اعضای مجموعه اول در ضرب دکارتی بود و هر عضو از مجموعه اول نیز فقط به یک عضو از مجموعه دوم نگاشت می شد.

عضو اول توى
مجموعه اعداد حقیقی

- مثال رابطه squared روی مجموعه اعداد حقیقی:

$$\text{Squared} = \{ \dots, \langle 1, 1 \rangle, \langle 2, 4 \rangle, \langle 3, 9 \rangle, \dots \}$$

- که عضو دوم در هر یک از زو جهای مرتب را می توانیم با نماد Squared(3) و ... نیز نشان دهیم.

ازیار پالهنجک
عضو دوم توى
مجموعه اعداد حقیقی
و مربع عضو اول
است

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

این میشه یه شی
داره به نتیجه‌ی این تابع
که عدد ۴ است اشاره
میکنه پس شی است

هر عضو اول فقط به
یک عضو دیگه
نگاشت میشه

یادآوری

■ یا رابطه + بصورت:

$$+=\{\dots, <1,2,3>, <2,3,5>, <4,3,7>, \dots\}$$

■ که عضو سوم از هر سه زائی مرتب را می توانیم بصورت زیر نیز
رجوع کنیم:

Sum(1,2)
Sum(2,3)
Sum(4,3)

■ در این درس از نماد تابعی برای رجوع به یک شی استفاده شده است.

ترم میتوانه یک ثابت باشه
یا به صورت نماد تابعی باشه مثل پای چپ امین
یا ترم مرکب باشه

ترم

- ترم: یک عبارت منطقی که به یک شی رجوع می کند.
- ثابت: امین
- پای چپ امین (LeftLegOf(Amin))
- ترم مرکب: نماد تابع بدنبال یک لیست از ترمها داخل پرانتز

هرارگومان خودش
یه ترم

married
یه مسند است

جملات ساده یا اتمی

■ یک نماد مسندی بدنیال یک لیست از ترمها داخل پرانتز

Brother(Amin,Amir)

Married(Father(Amin), Mother(Amin))

■ یک جمله اتمی درست است اگر رابطه ای که بوسیله نماد مسندی به آن اشاره می شود بین اشیائی که بوسیله آرگومانها به آنها اشاره می شود برقرار باشد.

پدر امین یه تابع
است
که تابع خودش ترم
حساب میشد
ماریار پالهنجک

تفاوت تابع و مسند؟

اگه مسند بنویسیم معمولاً میشه بهش مقدار درست و غلط نسبت داد
ولی تابع داره به یک شی رجوع میکنه
نمیشه بش درست و نادرست نسبت داد

مثالاً پدر امین یه تابع است و نمیشه مقدار درست و غلط بش بدیم
ولی به برادر بودن و ازدواج کردن میشه ارزش درست و نادرست داد

۱۴۰۱-۰۲

جملات مرکب

- استفاده از رابطهای منطقی
 - Brother(A,B) ∧ Brother(B,A)
 - Older(A,30) ∨ Younger(B,30)
 - Older(A,30) ⇒ ¬Younger(A,30)

سورها

■ علاقمند به بیان خصوصیاتی مربوط به همه یا برخی از اشیاء

این جمله برای تمام اشیایی که در دامنه مدل ما هستند باید برقرار باشد

سور عمومی

برای هر شی x , اگر x یک مار باشد آنگاه x یک خزنده است.

$$\forall x \text{ Snake}(x) \Rightarrow \text{Reptile}(x)$$

در مدل m درست است اگر و تنها اگر برای هر شی در دامنه مدل m درست باشد. یا:

$$\text{Snake}(M1) \Rightarrow \text{Reptile}(M1) \quad \boxed{\wedge}$$

$$\text{Snake}(M2) \Rightarrow \text{Reptile}(M2) \quad \boxed{\wedge}$$

$$\text{Snake}(M3) \Rightarrow \text{Reptile}(M3) \quad \boxed{\wedge} \dots$$

اگه فرض کنیم
 $M1, M2, M3$
اشیا مدل ما هستند
جمله‌ی بالا که یه
شرطه باید برای همه
اینها برقرار باشه

یک تابع داریم که ارجومانش هم یه ثابت مثل امین است
چون ارجومانش ثابت است بش ترم زمینه ای میگیم
اگه بود Father(x)
دیگه بش ترم زمینه ای نمیگفتیم

سور عمومی

- یک ترم بدون متغیر یک ترم زمینه (ground term) نامیده می شود.
- مثال:

$Father(A \min)$

- دقت: همه مارها خزنده هستند باید بصورت شرطی نوشته شود.
- سور عمومی شامل همه اشیاء دامنه میشود حتی برای اشیائی که مار نیستند

■ بصورت زیر صحیح نیست

$\forall x \text{ Snake}(x) \wedge \text{Reptile}(x)$

چون باید برای همه اشیای دامنه ای مدل ما برقرار باشه اگه ایکس یه میز باشه این جمله درست نمیشه پس نمیشه جمله ای مازیار موردنظر را نشان داد اگه جمله شرطی باشه وایکس میز باشه به منتفی مقدم کل جمله درست میشه

سور وجودی

- بیان جمله ای که برای برخی از اشیاء دامنه درست است.
- مار M1 برادری دارد که مار است:

$$\exists x \quad Brother(x, M1) \wedge Snake(x)$$

اشیای دامنه را به
جای ایکس میگذاریم
و باهم فصل میکنیم
جمله هارا

■ هم ارز با

$$(Brother(M2, M1) \wedge Snake(M2)) \vee$$
$$(Brother(M3, M1) \wedge Snake(M3)) \vee$$
$$(Brother(M4, M1) \wedge Snake(M4)) \vee \dots$$

حداقل یکی از این کلاوز ها
که درست بشه کل جمله
درست میشه

سور وجودی

- دقت: برخی A ها B هستند را باید با عطف بیان کرد نه با شرطی
برخی سبیها قرمز هستند:

$$\exists x \quad Apple(x) \wedge Red(x)$$

- نه بصورت:

$$\exists x \quad Apple(x) \Rightarrow Red(x)$$

- در حالت شرطی اگر X سبی نباشد جمله به انتفای مقدم درست است.

به ازای همه ی جفت
اشیای دامنه این
رابطه برقرار است

اگه ایکس ولی وای
باشه انگاه وای
فرزند ایکس است

سورهای تو در تو

■ علاقمند به بیان جملات پیچیده تر

$$\forall x \forall y \text{ Parent}(x, y) \Rightarrow \text{Child}(y, x)$$

■
یا

$$\forall x, y \text{ Parent}(x, y) \Rightarrow \text{Child}(y, x)$$

برای ساده تر نوشتن
وقتی دوتا
سور عمومی پشت سر
هم هستند اینطوری
مینویسیم

■ هر کسی با فردی دوست است

$$\forall x \exists y \leftarrow \text{Friend}(x, y)$$

$$\exists x \forall y \text{ Friend}(x, y)$$

■ شخصی وجود دارد که دوست همه است

این ایکس وابسته به
سور وجودی است

این ایکس وابسته به
سور عمومی است

سورهای تو در تو

محدوده‌ی هر متغیری
به نزدیکترین
سورش وابسته می‌شود
تو این مثال ایکس
دو جا استفاده شده
معنای ایکس در
هر کدام از این پرانتز
ها به نزدیک ترین
سورش بستگی دارد

- ترتیب قرار دادن سورها بطور کلی مهم است.
- استفاده از پرانتز به درک بهتر کمک می‌کند
- می‌توان همانند حلقه‌های تو در تو تصور کرد
- هنگامی که دو سور از یک متغیر استفاده می‌کنند:

$$\forall x [Cat(x) \vee (\exists x \ Brother(A, x))]$$

- متغیر وابسته به نزدیکترین سوری است که در حوزه آن قرار دارد.
- بهتر است متغیرهای متفاوتی استفاده شوند:

$$\forall x [Cat(x) \vee (\exists z \ Brother(A, z))]$$

سور عمومی

ایکس سیب را دوست
داشته باشد

رابطه بین سورها

■ همه سیب را دوست ندارند.

■ هیچکسی وجود ندارد که سیب دوست داشته باشد

$$\forall x \neg Likes(x, Apples) \equiv \neg \exists x Likes(x, Apples)$$

■ یا

■ هر کسی سیب دوست دارد

■ هیچکس وجود ندارد که سیب را دوست نداشته باشد

$$\forall x Likes(x, Apples) \equiv \neg \exists x \neg Likes(x, Apples)$$

به ازای همه i
اشیای دامنه N قیض
پی درست است
هم ارز است با
وجودنده شی i است
که پی برآش درست
باشه

رابطه بین سورها

■ در واقع چون سور عمومی یک عطف بر روی دامنه اشیاء است و سور وجودی یک فصل رابطه فوق همان قانون دمورگان است.

$$\forall x \neg P \equiv \neg \exists x P$$

$$\neg \forall x P \equiv \exists x \neg P$$

$$\forall x P \equiv \neg \exists x \neg P$$

$$\exists x P \equiv \neg \forall x \neg P$$

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

$$P \wedge Q \equiv \neg(\neg P \vee \neg Q)$$

$$P \vee Q \equiv \neg(\neg P \wedge \neg Q)$$

اینطور نیست که به
ازای هر ایکس پی
درست باشد
معادل با اینکه
وجود داره ایکسی که
نقیض پی برآش
درسته
یا پی برآش نادرسته

اینطور نیست که به
ازای هر ایکسی
نقیض پی درست
باشد
یا پی نادرست باشد

اینطوری نیست که یه
ایکسی وجود داشته
باشد که پی برآش
نادرست باشد^۱

برابری

■ روش دیگر ساختن جملات ساده

ترم اول
پدر امین
ترم دوم
امیر

AtomicSentence → *Predicate* | *Predicate(Term, ...)* | *Term = Term*

■ نشان دادن اینکه دو ترم به یک شئ رجوع می کنند.

ایکوال مسند یا
رابطه‌ی ما میشه
که با مساوی نشان
دادیم

Father(Amin)=Amir

■ روش ساده‌تر برای

استفاده دیگر: ذکر اینکه دو ترم برابر نیستند.

A حداقل دو براuder دارد:

$\exists x, y \quad Brother(A, x) \wedge Brother(A, y)$

$\exists x, y \quad Brother(A, x) \wedge Brother(A, y) \wedge \neg(x = y)$

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

20

این دو ترم برابر
نیستند

دقت

■ امین دو برادر دارد:

$Brother(Amir, Amin) \wedge Brother(Hamid, Amin)$

■ لازم است ذکر شود که Amir و Hamid به افراد متفاوتی رجوع می کنند.

■ کاملتر، امین فقط دو برادر دارد:

$Brother(Amir, Amin) \wedge Brother(Hamid, Amin) \wedge$
 $Amir \neq Hamid \wedge (\forall x Brother(x, Amin) \Rightarrow (x = Amir) \vee (x = Hamid))$

- یک پیشنهاد استفاده از ایده مورد استفاده در پایگاههای داده
- هر نماد ثابت به شیء متفاوتی اشاره می کند.
- **فرض نامهای یکتا unique-names assumption**
- جملات اتمی که نمی دانیم درست هستند، در واقع نادرست فرض می شوند.

■ **فرض دنیای بسته closed-world assumption**

- هر مدل عناصر دامنه اش بیش از نمادهای ثابت استفاده شده نیست.

■ **فرض بسته بودن دامنه domain closure**

نماد ها فقط همون هایی هستند
که توى دامنه ى مدل استفاده
شدند و چيز اضافه تری نیست

■ در این شرایط جمله

Brother(Amir,Amin) ∧ Brother(Hamid,Amin)

■ دقیقاً بیان می کند که امین دو برادر دارد.



م. بالهنج
مازیار بالهنج

دانشگاه صنعتی اصفهان - پژوهشکده فاوا

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

والسلام

■ در تهیه این اسلایدها، از اسلایدهای سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی
منطق مرتبه اول - ۳
نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنگ
آزمایشگاه هوش مصنوعی
دانشکده مهندسی برق و کامپیوتر
دانشگاه صنعتی اصفهان

یادآوری

منطق گزاره ای محدود است و حتی
نوشتن جملات لازم هم براش سخته حتی
برای یه دنیای ساده
برای همین از منطق مرتبه اول استفاده
میکنیم

این منطق خیلی گویاتر و میشه جملات را
خیلی بهتر نسبت به منطق گزاره ای بیان
کرد
کوچکترین واحد در منطق گزاره ای گزاره
است که یک جمله‌ی خبری است که میشه
بهش نسبت درست و نادرست داد
گزاره را نمیشه به اجزای کوچکتر
شکست

در منطق مرتبه اول یه تعدادی شی و روابط بین اونها را
داریم
دستور منطق مرتبه اول میگه چطوری جملات را در این
منطق میشه ایجاد کرد
تفسیر میگه اشیا و روابط به چه چیزی در ان دنیا اشاره
میکنند

- معرفی منطق مرتبه اول
- اشیاء و روابط بین آنها
- دستور
- معنا - مدل ، تفسیر
- ثابت
- مسند
- تابع
- سورها

استفاده از منطق مرتبه اول

- تعامل با پایگاه دانش:
 - اضافه کردن جملات به پایگاه دانش با `:Tell`
 - `Tell(KB, Boy(Amin)).`
 - `Tell(KB, Person(Amir)).`
 - `Tell(KB, $\forall x \text{ Boy}(x) \Rightarrow \text{Human}(x)$)`
- به چنین جملاتی **اظهارات assertions** می گویند.

افزودن قوانین به
پایگاه دانش

:ASK

- سؤال کردن از پایگاه دانش با Ask(KB,Boy(Amin))
 - که درست باز می گرداند.
 - چنین جملاتی را سؤال goals می گویند.
 - هر جمله ای که بتواند از پایگاه دانش ایجاب شود می تواند سؤال شود:
- Ask(KB,Human(Amin))
 - که درست باز می گرداند.

چه ایکس هایی
انسان هستند؟
جایزگین هایی برای
متغیر ایکس را
برمیگردانه

- يا سؤال کرد:
 - Ask(KB, $\exists x \text{ Human}(x)$)
 - که درست جواب می دهد
 - ولی انتظار بیشتری داریم:
 - اشیائی که بجای X می توانند قرا بگیرند را باز می گرداند.

$$\{x/Amin\}$$

مرتبه اول

ولی دو فرد یکسان است
بر عکس این جمله شرطی درست نیست
ممکن هم زاد باشند
ولی برادر نباشه رابطشون

استفاده

دامنه خویشاندی

برادرها هم زاد هستند

مادر بودن یک تابع است

- $\forall x, y \ Brother(x, y) \Rightarrow Sibling(x, y)$

مادر هر کس ولی مؤنث اوست

- $\forall m, c \ Mother(c) = m \Leftrightarrow (Female(m) \wedge Parent(m, c))$

ولی بزرگ ولی ولی یک فرد است.

$$\forall g, c \ Grandparent(g, c) \Leftrightarrow \exists p \ Parent(g, p) \wedge Parent(p, c).$$

یک هم زاد فرزند دیگر ولی یک فرد است.

$$\forall x, y \ Sibling(x, y) \Leftrightarrow x \neq y \wedge \exists p \ Parent(p, x) \wedge Parent(p, y).$$

هم زاد بودن یک رابطه متقارن است.

- $\forall x, y \ Sibling(x, y) \Leftrightarrow Sibling(y, x)$

استفاده از منطق مرتبه اول

- برخی از جملات اصل (axiom) هستند (وابسته به جملات دیگر نیستند).
- برخی جملات هم قضیه (theorem) هستند و از اصول ایجاب می شوند.
- $\forall x,y \text{ } Sibling(x,y) \leftrightarrow Sibling(y,x)$
 - از نقطه نظر منطقی فقط اصول را باید در پایگاه دانش گذاشت.
 - از نقطه نظر عملی قضیه ها هم لازم هستند.
- چون در این صورت زمان زیادی برای بدست آوردن آنها صرف خواهد شد.

S(n)
یک تابع است که به
تالی ان اشاره میکند

استفاده از منطق مرتبه اول

مسند

$NatNum(0)$.

$\forall n \ NatNum(n) \Rightarrow NatNum(S(n))$.

صفر اولین عدد
طبیعی درنظرگرفته
شده پس نمیتوانه تالی
یه عبارتی باشه

$\forall n \ 0 \neq S(n)$.

$\forall m, n \ m \neq n \Rightarrow S(m) \neq S(n)$.

■ دامنه اعداد:

■ تعریف اعداد طبیعی:

■ اصولی برای تعریف تابع تالی S:

successor(n)

وقتی یه جمله
بامساوی بیان میشه
دوظرف ش باید ترم
باشه و یه راه ساختن
ترم هم استفاده از
تابع بود ترم هم به
شی اشاره میکنه

استفاده از منطق مرتبه اول

+
یک تابع است
داره به شی ای اشاره
میکنه که جمع صفر
و ام است

اپراتور قبل از
عملوند هاش میاد
میشه پرفیکس

■ تعریف جمع بر اساس تالی:

$$\forall m \ NatNum(m) \Rightarrow + (0, m) = m .$$

$$\forall m, n \ NatNum(m) \wedge NatNum(n) \Rightarrow + (S(m), n) = S(+ (m, n)) .$$

■ برای راحتی خواندن تعریف دوم را بصورت زیر نیز می توان
نوشت:

$$\forall m, n \ NatNum(m) \wedge NatNum(n) \Rightarrow (m + 1) + n = (m + n) + 1 .$$

جمع تالی ام و ان
مساوی است با تالی
جمع ام و ان

اینجا عملگر او مده
بین دو عملوند میشه
اینفیکس یا میانوند

اولی بصورت پیشوند prefix
دومی بصورت میانوند infix

مازیار پالهنج

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

تالی را با بعلاوه یک
نشان میدهیم

استفاده از منطق مرتبه اول

دامنه مجموعه ها

- مجموعه ها، مجموعه تھی و آنانی که بوسیله الحاقی چیزی به یک مجموعه ساخته شده اند هستند

$$\forall s \ Set(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2 \ Set(s_2) \wedge s = \{x | s_2\})$$

$$\forall s \ Set(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2 \ Set(s_2) \wedge s = Add(x, s_2)).$$

- مجموعه تھی دارای عضوی که به آن الحاق شده باشد نیست

$$\neg \exists x, s \ Add(x, s) = \{\}.$$

تکرار یک عضو در
مجموعه ها بی اثر
است

استفاده از منطق مرتبه اول

الحق کردن عنصری که هم اکنون عضو مجموعه است اثربن دارد

■ دامنه مجموعه ها

$$\forall x, s \ x \in s \Leftrightarrow s = Add(x, s).$$

■ تنها عناصر یک مجموعه، عناصری هستند که به آن الحق شده اند.

$$\forall x, s \ x \in s \Leftrightarrow \exists y, s_2 \ (s = Add(y, s_2) \wedge (x = y \vee x \in s_2)).$$

اون وای جدیدی که
داره به اس ۲ اضافه
میشه همون ایکس ما
است ک الان دیگه
عضو مجموعه است

هوش مصنوعی - نیمسال اول

ایکس قبلاتوی اس ۲
بوده

مازیار پالهنجک

تعريف زیرمجموعه
بودن

استفاده از منطق مرتبه اول

دامنه مجموعه ها ■

-
- $\forall S_1, S_2 \quad S_1 \subseteq S_2 \Leftrightarrow (\forall x \quad x \in S_1 \Rightarrow x \in S_2)$
- $\forall S_1, S_2 \quad (S_1 = S_2) \Leftrightarrow (S_1 \subseteq S_2 \wedge S_2 \subseteq S_1)$ برابری دو مجموعه
- $\forall x, S_1, S_2 \quad x \in (S_1 \cap S_2) \Leftrightarrow (x \in S_1 \wedge x \in S_2)$
- $\forall x, S_1, S_2 \quad x \in (S_1 \cup S_2) \Leftrightarrow (x \in S_1 \vee x \in S_2)$

درگ په لیست
عضوی است ۵

در زمان ۵
این درگ حاصل شده

دنیای دیو

■ جمله درگ، شامل لیست درگ و زمان:

$Percept([Stench, Breeze, Glitter, None, None], 5)$.

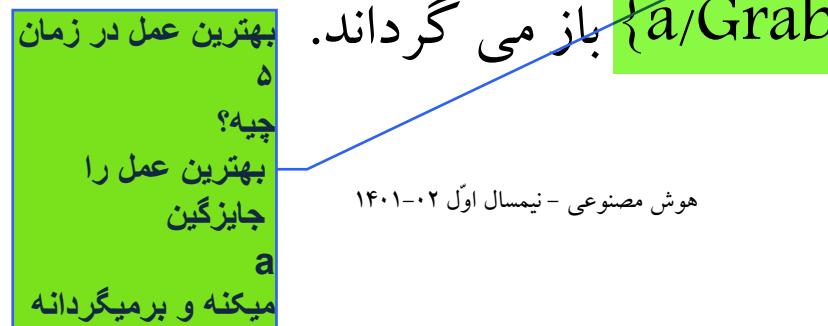
■ اعمال:

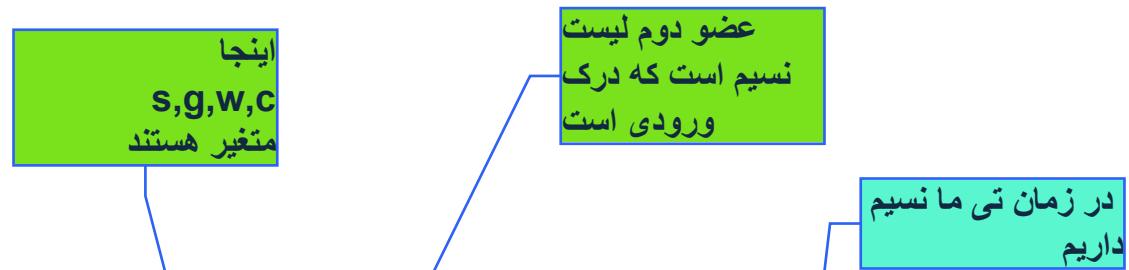
$Turn(Right), Turn(Left), Forward, Shoot, Grab, Climb$.

■ برای تعیین اینکه بهترین عمل برای انجام چیست؟

$\text{ASKVARS}(\exists a \ BestAction(a, 5))$,

■ که بطور مثال $\{a/Grab\}$ باز می گرداند.





■ استخراج واقعیات از درک خام:

$$\forall t, s, g, w, c \quad Percept([s, Breeze, g, w, c], t) \Rightarrow Breeze(t)$$

$$\forall t, s, g, w, c \quad Percept([s, None, g, w, c], t) \Rightarrow \neg Breeze(t)$$

$$\forall t, s, b, w, c \quad Percept([s, b, Glitter, g, w, c], t) \Rightarrow Glitter(t)$$

$$\forall t, s, b, w, c \quad Percept([s, b, None, g, w, c], t) \Rightarrow \neg Glitter(t)$$

در زمان تی نسیم
نداریم

■ در منطق گزاره‌ای باید برای هر زمان t یک جمله می‌نوشتیم.

ولی اینجا مینویسیم
Breeze(t)

در زمان تی ما
درخشندگی داریم

پایگاه دانش برای دنیای دیو

- در منطق گزاره‌ای باید برای هر زمان t یک جمله می‌نوشتیم.

در زمان t که
هرچی می‌توانه باشه

عاملی که با هر درکی
که می‌گیره سریعاً یه
اکشنی نشان میده

رفتار واکنشی

$$\blacksquare \forall t \text{ Glitter}(t) \Rightarrow \text{BestAction(Grab}, t)$$

اگه در زمان t ما
درخشنده‌ی حس
کردیم
سریعاً عمل برداشتن
را انجام بده

خانه‌ی ۱۰۲ مربع
است

مجاورت دو مربع

- چگونگی نمایش اشیاء: مربعها، گودالها، دیو، عامل؟
- استفاده از Square_{12} یا Square_{33} با
- بیان مجاورت دو مربع؟
- بهتر: نمایش با یک لیست دو عضوی $[x, y]$

$$\forall x, y, a, b \quad \text{Adjacent}([x, y], [a, b]) \Leftrightarrow \\ (x = a \wedge (y = b - 1 \vee y = b + 1)) \vee (y = b \wedge (x = a - 1 \vee x = a + 1))$$

اگه ایکس هاشون
برابر باشه
باید وای شون یکی
اختلاف داشته باشه
یا یکی زیادتر
یا یکی کمتر

- گودالها را هم می توان نام داد
- لزومی به جدا بیان کردن گودالها نیست

مسند



- فقط یک دیو: ثابت Wumpus
- فقط یک عامل: Agent



- مکان عامل: $At(Agent, s, t)$
- ثابت کردن مکان دیو: $At(Wumpus, [1,3], t)$
- اشیاء در هر زمان فقط در یک مکان می‌توانند باشند:

$$\forall x, s_1, s_2, t \quad At(x, s_1, t) \wedge At(x, s_2, t) \Rightarrow s_1 = s_2 .$$

خانه‌ی اس نسیم دار
است

استنتاج خواص پنهان

- خواص مربعها:

$$\forall s, t \text{ At}(\text{Agent}, s, t) \wedge \text{Breeze}(t) \Rightarrow \text{Breezy}(s)$$

به ازای هر اس اگه
اس نسیم دار باشه یه
مربع به اسم آر کنار
اس هست که آر یه
گودال توشه

- مربهای نزدیک گودال نسیم دار هستند:

قانون تشخیصی—استنتاج علت از اثر

$$\forall s \text{ Breezy}(s) \Rightarrow \exists r \text{ Adjacent}(r, s) \wedge \text{Pit}(r)$$

علت باعث یه اثری
شده

- قانون علی—استنتاج اثر از علت

$$\forall r \text{ Pit}(r) \Rightarrow [\forall s \text{ Adjacent}(r, s) \Rightarrow \text{Breezy}(s)]$$

هرخانه‌ی مجاور
مربع آر که گوداله
یک خانه‌ی نسیم
داره

مازیار پالهنجک

اصل حالت تالی

$$\forall t \text{ HaveArrow}(t + 1) \Leftrightarrow (\text{HaveArrow}(t) \wedge \neg \text{Action}(\text{Shoot}, t)).$$

علت: گودال داشتن

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

اثر: نسیم دار بودن

عملی انجام نداده
باشیم که تیرهایمان
از بین رفته باشند

19

19

منطق مرتبه اول

مهندسی دانش در م.م.ا.

باید پایگاه دانش را بسازیم
چطوری میتوانیم این پایگاه را بسازیم؟
مهندسی دانش به نحوه‌ی ساخت صحیح پایگاه
دانش میگن

مشخص کردن وظیفه

.1

- .1 چه سؤالاتی را باید پاسخگو باشد؟
.2 چه دانشی باید نمایش داده شود؟

یافتن دانش مرتبط

.2

- .1 یا خود خبره یا کمک گیری از یک خبره (اکتساب دانش)

یاخود کسی که داره
مهندسي دانش را
انجام میده خبره است
توى اون زمينه یا از
يه نفر ديگ کمک
میگيره

(ontology) هستی‌شناسی

کد کردن دانش عمومی دامنه

.3

کد کردن توصیف نمونه خاص مسئله

.4

سوال کردن از روال استنتاج و گرفتن پاسخ

.5

خطا زدائی پایگاه دانش

.6

.7

چون داریم از منطق
مرتبه اول استفاده
میکنیم باید مسندها و
توابع و ثابت‌ها
مشخص بشه

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

20

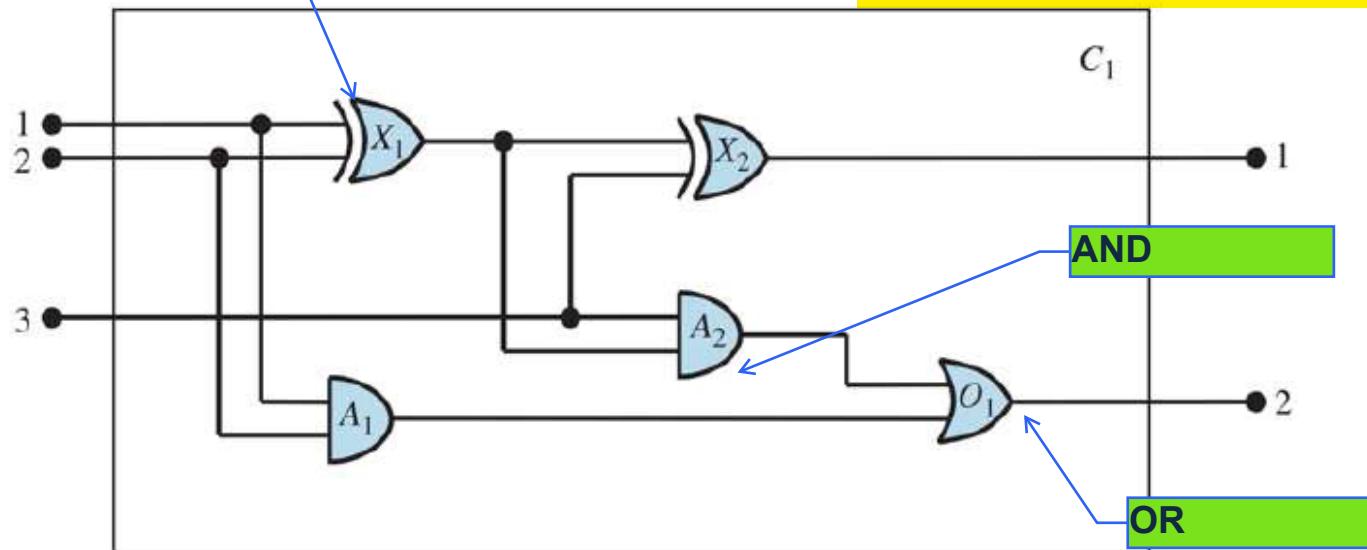
مازیار پالهنجک

علم پزشکی را کدکردیم
مرحله ۵ که نمونه‌ی خاصه میشه یه
بیمار

XOR
 $0,1 \Rightarrow 1$
 $1,0 \Rightarrow 1$
 $0,0 \text{ OR } 1,1 \Rightarrow 0$

دامنه مدارهای منطقی

تمام جمع کننده یک بیتی



A digital circuit C_1 , purporting to be a one-bit full adder. The first two inputs are the two bits to be added, and the third input is a carry bit. The first output is the sum, and the second output is a carry bit for the next adder. The circuit contains two XOR gates, two AND gates, and one OR gate.

دامنه مدارهای منطقی

.1 مشخص کردن وظیفه

- .1 آیا مدار واقعاً جمع را بدرستی انجام می دهد؟
.2 دروازه هایی که ورودی اول به آنها متصل است کدامند؟

.2 مشخص کردن دانش مرتبط

- .1 سیمها، دروازه ها، نوع دروازه ها (و، یا، یا مانعه الجمع، نقیض)
.2 نامربوط: اندازه، رنگ، قیمت

.3 تصمیم گیری در مورد واژه نامه

Gate(X1) .1

نمایش به کمک تابع

گزینه ها: .2

Type(X₁) = XOR .3

Type(X₁, XOR) .4

XOR(X₁) .5

به کمک مسند

دامنه مدارهای منطقی

کد کردن دانش عمومی .4

- اگر دو ترمینال به هم متصل باشند آنگه سیگنال روی آنها با هم برابر است:
- $\forall t_1, t_2 \text{ Terminal}(t_1) \wedge \text{Terminal}(t_2) \wedge \text{Connected}(t_1, t_2) \Rightarrow \text{Signal}(t_1) = \text{Signal}(t_2)$ سیگنال روی هر ترمینال یا یک یا صفر است.
- $\forall t \quad \text{Terminal}(t) \Rightarrow (\text{Signal}(t) = 1 \vee \text{Signal}(t) = 0)$
- $1 \neq 0$ دوتا ثابت یک و صفر
یکسان نیستند
- $\forall t_1, t_2 \quad \text{Connected}(t_1, t_2) \Rightarrow \text{Connected}(t_2, t_1)$
- چهار نوع دروازه وجود دارند:

$$\forall g \quad \text{Gate}(g) \wedge k = \text{Type}(g) \Rightarrow k = AND \vee k = OR \vee k = XOR \vee k = NOT.$$

سیگنالی که روی خروجی ۱ گیت ما است
یک است اگر ورودیها اگر
یه ان ای وجود داره که ان ورودی گیت ما
است و مقدارش یک است

خروجی یه گیت
OR
زمانی یک است که
حداقل یکی از
ورودی هاش یک
باشه

دامنه مدارهای منطقی

- $\forall g \text{ Gate}(g) \wedge \text{Type}(g) = \text{OR} \Rightarrow \text{Signal}(\text{Out}(1,g)) = 1 \Leftrightarrow \exists n \text{ Signal}(\text{In}(n,g)) = 1$
 - $\forall g \text{ Gate}(g) \wedge \text{Type}(g) = \text{AND} \Rightarrow \text{Signal}(\text{Out}(1,g)) = 0 \Leftrightarrow \exists n \text{ Signal}(\text{In}(n,g)) = 0$
 - $\forall g \text{ Gate}(g) \wedge \text{Type}(g) = \text{XOR} \Rightarrow \text{Signal}(\text{Out}(1,g)) = 1 \Leftrightarrow \text{Signal}(\text{In}(1,g)) \neq \text{Signal}(\text{In}(2,g))$
 - $\forall g \text{ Gate}(g) \wedge \text{Type}(g) = \text{NOT} \Rightarrow \text{Signal}(\text{Out}(1,g)) \neq \text{Signal}(\text{In}(1,g))$
- دروازه ها، ترمینالها و سیگنالها مجزا هستند:

$$\forall g, t, s \text{Gate}(g) \wedge \text{Terminal}(t) \wedge \text{Signal}(s) \Rightarrow g \neq t \wedge g \neq s \wedge t \neq s.$$

سیگنالی که روی
خروجی پایه ی یک
گیت هست
مخالف سیگنال
ورودی همان پایه
است

مازیار پالان
...

سیگنال ورودی روی
پایه ی یک گیت جی
اگه مخالف باشه با
سیگنال ورودی در
پایه ی دو ورودی
گیت

24

هوش مصنوعی - نیمسال اول ۲

خروجی اول گیت
ایکس۱
به ورودی اول
ایکس۲ وصل است

تا خروجی ۲
تاورودی ۳

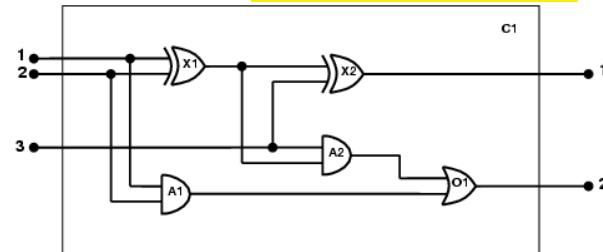
دامنه مدارهای منطقی

$Circuit(C_1) \wedge Arity(C_1, 3, 2)$
 $Gate(X_1) \wedge Type(X_1) = XOR$
 $Gate(X_2) \wedge Type(X_2) = XOR$
 $Gate(A_1) \wedge Type(A_1) = AND$
 $Gate(A_2) \wedge Type(A_2) = AND$
 $Gate(O_1) \wedge Type(O_1) = OR .$

مشخص کردن
ارتباط ها

$Connected(Out(1, X_1), In(1, X_2))$	$Connected(In(1, C_1), In(1, X_1))$
$Connected(Out(1, X_1), In(2, A_2))$	$Connected(In(1, C_1), In(1, A_1))$
$Connected(Out(1, A_2), In(1, O_1))$	$Connected(In(2, C_1), In(2, X_1))$
$Connected(Out(1, A_1), In(2, O_1))$	$Connected(In(2, C_1), In(2, A_1))$
$Connected(Out(1, X_2), Out(1, C_1))$	$Connected(In(3, C_1), In(2, X_2))$
$Connected(Out(1, O_1), Out(2, C_1))$	$Connected(In(3, C_1), In(1, A_2)) .$

کد کردن مسئله خاص .5



دامنه مدارهای منطقی

چه چیزی باید به جای

i_1, i_2, i_3

قرار بگیره تا خروجی پایه اول

$C_1=0$,

خروجی پایه دوم

$C_1=1$

سوال کردن .6

مجموعه مقادیر ممکن پایه ها؟

$$\exists i_1, i_2, i_3 \ Signal(In(1, C_1)) = i_1 \wedge Signal(In(2, C_1)) = i_2 \wedge Signal(In(3, C_1)) = i_3 \\ \wedge Signal(Out(1, C_1)) = 0 \wedge Signal(Out(2, C_1)) = 1.$$

$$\{i_1/1, i_2/1, i_3/0\} \quad \{i_1/1, i_2/0, i_3/1\} \quad \{i_1/0, i_2/1, i_3/1\}.$$

خطا زدایی .7

نبود $0 \neq 1$ در پایگاه دانش

مثال:

$\exists i_1, i_2, o$

$$Signal(In(1, C_1)) = i_1 \wedge Signal(In(2, C_1)) = i_2 \wedge Signal(Out(1, X_1)) = o$$

مازیار پالهنجک

فرض کنیم توی
پایگاه دانش نگفته
بودیم که یک مخالف
صفر است

هوش مادی

سوال
مقادیر
 i_1, i_2, o
چی میتوانه باشه با
این اطلاعات؟

26

26

■ برای XOR داریم:

$$\text{Signal}(\text{Out}(1, X_1)) = 1 \Leftrightarrow \text{Signal}(\text{In}(1, X_1)) \neq \text{Signal}(\text{In}(2, X_1))$$

■ اگر ورودیها ۱ و ۰ باشند

$$\text{Signal}(\text{Out}(1, X_1)) = 1 \Leftrightarrow 1 \neq 0.$$

■ ولی نمی‌دانیم ۱ و ۰ برابر نیستند و خروجی ۱ را نمی‌توانیم نتیجه بگیریم.

باید اضافه کنیم به
پایگاه دانش که این
سیگنال یک و صفر نگ
باهم مساوی نیستند

- استفاده از منطق مرتبه اول
- اضافه کردن جملات
- سؤال کردن
- چند دامنه نمونه
- مهندسی دانش درمنطق مرتبه اول



والسلام

م. پالهنج
مازیار پالهنج

دانشگاه صنعتی اصفهان - پژوهشکده فاوا

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

29

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی

استنتاج در منطق مرتبه اول - ۱

نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

استنتاج

همه‌ی انسان‌ها
فناپذیر هستند

■ فرض کنید:

$$\forall x \text{Human}(x) \Rightarrow \text{Mortal}(x)$$

Human(Socrat)

توی پایگاه دانش
داریم که سocrates
انسان است
ایا میشه نتیجه گرفت
سocrates فناپذیر است؟

■ تبدیل به حالت گزاره‌ای

میتونیم پایگاه دانش
را به حالت گزاره‌ای
دریابیم و از همان
قوانينی که توی
منطق گزاره‌ای برای
استنتاج داشتیم
استفاده کنیم

برای تبدیل پایگاه دانش به گزاره
ها باید این کارها را انجام بدیم

جي ترم زمينه اي
است چون متغيري
توش نداره

حاصلش يه جمله
است

قوانين استنتاج شامل سورها

■ جمله اي که از جايگزيني θ در جمله α بدست می آيد.

په لیستی که میگه به
جای هر ترمی ج
ترمی قرار بدیم

جايگزيني تتا روی
آلفا

SUBST(θ, α)

مثال:

■ $\text{SUBST}(\{x/\text{Ali}, y/\text{Amin}\}, \text{Likes}(x,y)) = \text{Likes}(\text{Ali}, \text{Amin})$

■ حذف عمومی: برای هر جمله α ، متغير v و ترم زمينه g :

متغير هامون جايگزين
ميشن يعني بهه جای
ایکس علی قرار
میگيره
به جای واي هم امين

$\forall v \alpha$

$\text{SUBST}(\{v/g\}, \alpha)$

جمله اي که ب
صورت سور عمومی
بیان شده

■ ترم زمينه: يك ترم بدون متغير

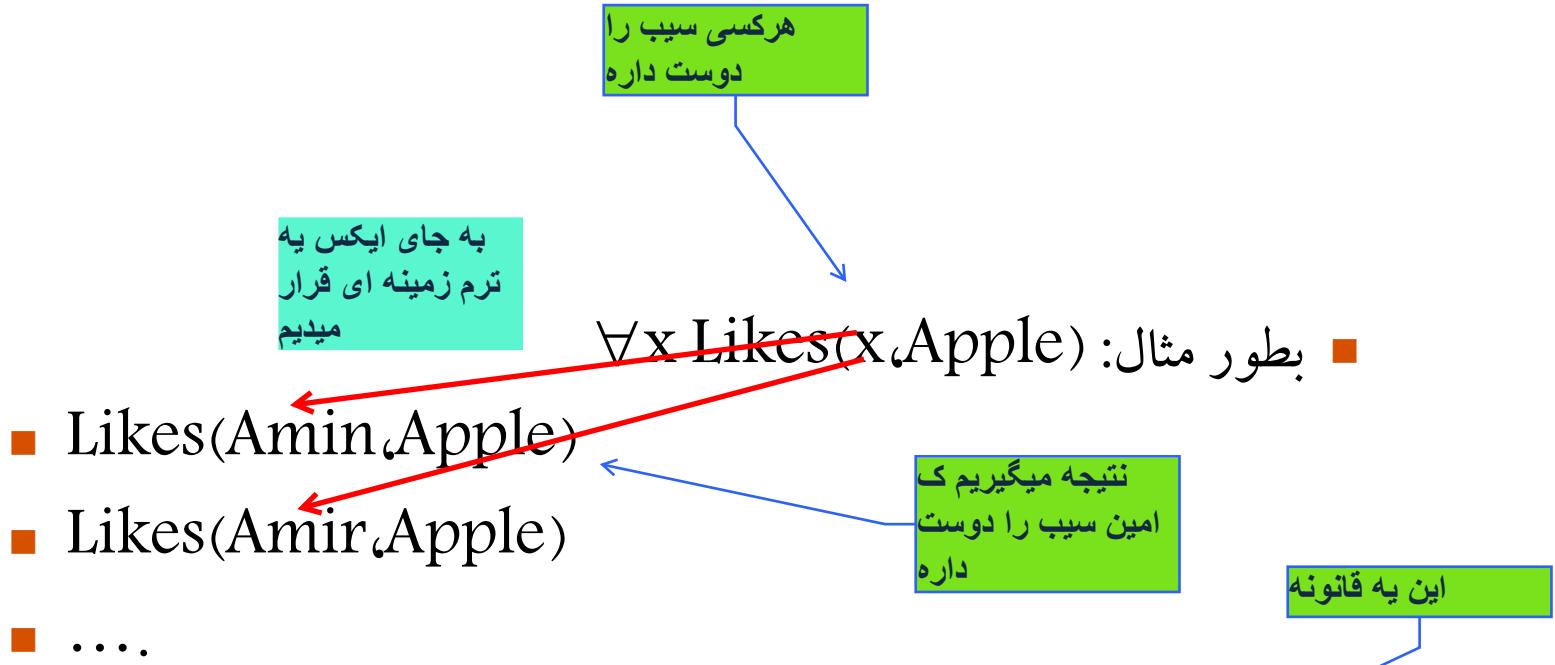
اگه اين قانون را
داشتيم ک ب ازاي هر
 v
آلفا درسته
ميشه جمله ي زيرخط
را نتيجه گرفت

مازيار پالهنگ

يه جمله اي خاheim
داشت که از
جايگزيني جي به
جای وي در آلفا
ایجاد ميشه

هوش مصنوعی - نیمسال اول ۰۲

3



حذف وجودی: برای هر جمله α ، متغیر v و نماد ثابت K که در جای دیگری از پایگاه دانش ظاهر نشده:

$$\frac{\exists v \alpha}{SUBST(\{v / K\}, \alpha)}$$

مازیار پالهنجک

میشه جمله‌ی آلفا را
نتیجه گرفت که به
جای وی ثابت
 k
رو قرار بدیم

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

4

نکته های
نباید
 k
جای دیگری توی
پایگاه دانش استفاده
شده باشه



■ نام جدید K ، ثابت اسکلم (Skolem constant)

■ حذف عمومی را بارها می توان اعمال نمود،

■ حذف وجودی فقط یک بار

مثلًا مثال بالا به جای
ایکس امین و امیر
گذاشتیم
به ازای تمام اشیای
داخل دامنه باید
برقرار باشه

فقط ثابت کی را میشه
گذاشت



تبدیل پایگاه دانش به حالت گزاره ای

اینجا یه جمله در
منطق مرتبه اول
داریم که میخاییم به
گزاره ها تبدیل کنیم

کاهش به استنتاج گزاره ای

$$\forall x \text{Human}(x) \Rightarrow \text{Mortal}(x)$$

به جای متغیر ایکس
اشیایی ک توی دامنه
هستند را میگذاریم

■ تبدیل می شود به:

$$\text{Human}(\text{Bograt}) \Rightarrow \text{Mortal}(\text{Bograt})$$

$$\text{Human}(\text{Arashmidos}) \Rightarrow \text{Mortal}(\text{Arashmidos})$$

$$\text{Human}(\text{Sograt}) \Rightarrow \text{Mortal}(\text{Sograt})$$

...

$$\text{Human}(\text{Sograt})$$

این خودش یه گزاره
است

از جمله ی شرطی بالا که درباره ی سقراط درست شد و از گزاره اینکه سقراط
انسان است

طبق قانون انتزاع میشه نتیجه گرفت که سقراط فناپذیره
چون مقدم درست شد و شرط هم درسته پس تالی به ناچار درسته

مازیار پالهنجک

6

خروجی پدر امیر یه
ترم زمینه ای است
چون خروجی تابع ها
ترم های زمینه ای
است تعداد خیلی
زیادی از این خروجی
تابع هارا میشه به
جای نماد ثابت
استفاده کرد

همه ی اینها ترم
زمینه ای هستند
چون هیچ متغیری
توشون استفاده نشده

کاهش به استنتاج گزاره ای

- مشکل: نمادهای تابعی، تعداد زیادی ترم‌های زمینه وجود دارند:
- مثال: Father(Father(Amir))

قضیه هربرند: اگر یک جمله از یک پایگاه دانش منطق مرتبه
اول ایجاب شد، توسط یک زیر مجموعه محدود پایگاه دانش
گزاره ای شده ایجاب می شود

For $n = 0$ to ∞ do

create a propositional KB by instantiating
with depth- n terms

see if α is entailed by this KB

پایگاه اولیه توسط یه
زیرمجموعه‌ی
محدودی قابل ایجاب
شدن است

یه پایگاه دانش گزاره
ای شده ایجاد کن

پایگاه‌داری ترم‌ها تا
عمق ان

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

از اول لازم یست پایگاه دانش را به صورت گزاره ای دربیاریم
چون اگه بخایم همه ی این توابع را تا عمق ان بسازیم یه پایگاه
خیلی بزرگ میشه پس به تدریج میسازیم و چک میکنیم که
ایجاب میشه یا ن؟

اون جمله را میشه از گزاره ای شدن پایگاه
دانش بدست اورد
ما میخایم ببینیم آیا جمله ی α از پایگاه دانشی
ک داریم ایجاب میشه؟
میگه شما اول این پایگاه گزاره ای شده را تا
عمق صفر چک کن اگه آلفا ایجاب میشد که پیدا
شد اگه نه عمق را یکی بزرگ نماید کن

پایگاه دانش ک گزاره ای بشه ما قوانین استنتاج را در منطق گزاره ای داشتیم ولی مشکلی ک داره اینه ک با حذف سور عمومی تعداد زیادی جمله ایجاد میشه چون باید همه ی اشیای دامنه را جایگزینش کنیم درحالی که خیلی هاش مرتبط نیست با اون استنتاجی که ما میخاییم انجام بدیم مثلا ما میخایستیم فناپذیری سقراط را فقط نتیجه بگیریم ولی برای ارشمیدس و بقیه اعضای دامنه هم ایجاد شد

مشکل: کار می کند اگر جمله ایجاب شود و گرنه در حلقه می افتد

قضیهٔ تورینگ، چرچ: ایجاب کردن در م.م.ا. نیمه قابل تصمیم گیری است (الگوریتمهای وجود دارد که به هر جملهٔ ایجاب شده بله بگوید، ولی الگوریتمی وجود ندارد که به هر جملهٔ ایجاب نشده نه یعنی با اطمینان نمیشه گفت که اگه یه جمله ای ایجاب اولین پیش بگوید.

یعنی با اطمینان نمیشه گفت که اگه یه جمله ای ایجاب نشده پس حتماً باید نمیشه

ما استنتاج در منطق مرتبه اول انجام دادیم تا حالا یعنی سوره‌های مون را به
گزاره تبدیل کردیم

مشکل: تعداد جملات نامربوط تولید شده می تواند خیلی زیاد باشد.

مثلاً اگر داشتیم:

اینجا ما باید همه‌هدهمی اشیای دامنه را به جای ایکس بذاریم
تعداد زیادی گزاره درست می‌شود
راه حل: قانون انتزاع تعمیم یافته

هر کی گرسنه باشه و
سیب در اختیارش
باشه میخوره سیب را

قانون انتزاع تعمیم یافته

■ فرض کنید پایگاه دانش بصورت زیر باشد:

$$\begin{aligned} \forall x \text{Hungry}(x) \wedge \text{Owns}(x, \text{Apple}) &\Rightarrow \text{Eats}(x, \text{Apple}) \\ \text{Owns}(\text{Amir}, \text{Apple}) \\ \text{Hungry}(\text{Amir}) \end{aligned}$$

ما میخایم فقط برای امیر به این
نتیجه برسیم که سیب را میخوره
نه برای همه اعضای دامنه
مثل ناصر و اصغر و اکبر

■ می خواهیم سریعاً نتیجه بگیریم:

$$\text{Eats}(\text{Amir}, \text{Apple})$$

■ حتی اگر داشتیم

کمک میکنیه تا سریع
تر به نتیجه ای که
میخایم برسیم

از اعمال ترا روی
جملات اتمی
 P'_i, P_i
جملاتی که ایجاد
میشند برابر هستند

قانون انتزاع تعمیم یافته

■ برای جملات اتمی P'_i, P_i و q که یک جایگزینی θ وجود دارد
بطوریکه $SUBST(\theta, P'_i) = SUBST(\theta, P_i)$ برای همه آنها:

اگه تارا روی p_1 و p_1 پریم
اعمال کنیم این دوتا جمله بعد از
جایگزینی باهم برابر میشن
تتا وقتی روی p_2 و p_2 پریم
اعمال بشه بعدش این دوتا جمله
باهم برابر میشن و

.....
اگه این اتفاق افتاد میشه تالی
قائون را سریعا نتیجه گرفت فقط
باید همون جایگزینی ای که
روی جمله های پی اعمال کردیم
را برای کیو هم اعمال کنیم

به این میگن قانون انتزاع تعمیم
یافته چرا تعمیم یافته؟

چون قانون انتزاع عادی میگفت

$$p \Rightarrow q$$

اگه این جمله شرطی درست
باشه و پی هم درست باشه کیو
را میشه نتیجه گرفت

$$SUBST(\theta, q) = \frac{p_1, p_2, \dots, p_n, (p_1 \wedge p_2 \wedge \dots \wedge p_n) \Rightarrow q}{SUBST(\theta, q)}$$

چندتا جمله ای اتمی
چون بینشون کاما
گذاشته ینی جدا جدا
توی پایگاه دانش
وجود دارن

جملات اتمی که باهم
عطف شدن

کیو هم خودش یه
جمله ای اتمی است

اعمال جایگزینی ترا
روی جمله ای اتمی
 P_i

■ ارتقاء قانون انتزاع

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

10
با این جایگزینی ما
میتوانیم مقدم را پیدا
کنیم بعد از جایگزینی
میتوانیم تالی را نتیجه
بگیریم

جايزگيني تتا روی پی

موثق بودن ق.ا.ت.

از درستی پی
درستی پی تتا ايجاب
ميشه

$$p \models p\theta$$

■ با حذف عمومی می دانیم:

$$p\theta = SUBST(\theta, p)$$

تتا روی تک تی
اینها اعمال ميشه

پس

$$(p_1 \wedge \dots \wedge p_n \Rightarrow q) \models (p_1 \wedge \dots \wedge p_n \Rightarrow q)\theta = (p_1\theta \wedge \dots \wedge p_n\theta \Rightarrow q\theta)$$

اعمال شدن تتا روی
این جمله

مقدم اين شرطه با جمله ی پايين ک ايجاب شده
برابر است پس تالي هم نتیجه گيری ميشه

و

$$p'_1, \dots, p'_n \models p'_1 \wedge \dots \wedge p'_n \models p'_1\theta \wedge \dots \wedge p'_n\theta$$

درستی اينها را به
صورت تکی داريم
پس عطف کلشون هم
درسته

■ در صورتی که $p_i\theta = p'_i\theta$ طبق ق.ا. می توان نتیجه گرفت $q\theta$

اعمال جايگزيني تتا
روي اين جمله

هوش مصنوعی - نيمسال اول ۱۴۰۱-۰۲

ما اين شرط را
داشتيم که جايگزيني
برقرار است

11

قانون انتزاع تعیین یافته کمک میکنے جملات نامربوط ساخته نشده همون جایگزینی که باعث شد مولفه های مقدم باهم یکسان بشن روی تالی اعمال میشه

یکسان سازی

- روال یکسان سازی (unification) دو جمله اتمی p و q را گرفته و یک جایگزینی بازمی گرداند که باعث شود p و q مشابه به نظر برسند.

$$\text{UNIFY}(p, q) = \theta \text{ where } \text{SUBST}(\theta, p) = \text{SUBST}(\theta, q)$$

جایگزینی تنا

- θ یکسان ساز نامیده می شود.

وقتی تنا روی پی
اعمال بشه و تنا روی
کیو اعمال بشه دوتا
جمله ایجاد میکنه ک
باهم برابر هستند

میخایم یه یکسان ساز
پیدا کنیم برای دو تا جمله‌ی
اتمی پی و کیو

امیر هر کی را که
میشناسه از ش متفرقه

یکسان سازی

■ فرض کنید:

$$\forall x \text{Knows}(Amir, x) \Rightarrow \text{Hates}(Amir, x)$$

■ فرض کنید می خواهیم بدانیم امیر از چه کسانی متفرق است؟

این دو تا جمله باهم
یکسان نیستند
چطوری میشه
یکسانشون کرد؟
اگه به جای ایکس در
جمله‌ی اول بگذاریم
حمید

p	q	θ
Knows(Amir,x)	Knows(Amir,Hamid)	$\{x/hamid\}$
Knows(Amir,x)	Knows(y,Parviz)	$\{x/Parviz,y/Amir\}$
Knows(Amir,x)	Knows(y,Mother(y))	$\{y/Amir,x/Mother(Amir)\}$
Knows(Amir,x)	Knows(x,Saber)	fail

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

13

ظاهرش شکسته
چون به جای ایکس
هم صابر میاد هم
امیر

13

یکسان سازی

- Knows(y,Saber) و Knows(x,Saber) یک معنا دارند. ■
- جداسازی استاندارد: دو جمله ای که در حال یکسان سازی هستند از متغیرهای با نامهای متفاوت استفاده کنند. ■
- Unify(Knows(Amir,x1),Knows(x2,Saber))={x1/Saber,x2/Amir}

امیر همه را میشناسه
همه صابر را میشناسن
پس امیر صابر را میشناسه

اینجا هم ایکس هم زد
هردو متغیر هستند
جایگزینی هنوز
اعمال نشده

یکسان سازی

p	q	θ
Knows(Amir,x)	Knows(y,z)	$\{y/Amir, x/z\}$
		$\{y/Amir, x/Amir, z/Amir\}$
		$\{y/Amir, x/z, w/Bahram\}$

چون ممکن است در اینده
نیاز به یکسان سازی
دیگری داشته باشیم
با اینکار ما آزادی این را
داریم که یکسان سازی
دیگری انجام بدیم

- جایگزینی اول عمومی تر از دیگران است.
- سعی می کنیم عمومی ترین یکسان ساز را بیابیم
- most general unifier (mgu)

مثال

- بر اساس قانون فروش سلاح به کشورهای متخاصم توسط یک ایرانی جرم است. کشور فرضی، دشمن ایران تعدادی موشک در اختیار دارد و همه آنها توسط مزدور که یک ایرانی است به آنها فروخته شده است.
- مزدور مجرم است؟

تبديل جمله های سوال به منطق مرتبه اول

۱. ... فروش سلاح به کشورهای متخاصل توسط یک ایرانی جرم است.

$Iranian(x) \wedge Weapon(y) \wedge$

$Hostile(z) \wedge Sells(x, z, y) \Rightarrow Criminal(x)$

به کمک این قانون
میتوانستیم تا تا
مولفه های کلاوز
هایی که باهم عطف
شدن را نتیجه بگیریم

ایکس موشک است

۱. فرضی ... تعدادی موشک در اختیار دارد.

$\exists x Owns(Farzi, x) \wedge Missile(x)$

با استفاده از حذف وجودی و استفاده از ثابت اسکلم و حذف و

$Owns(Farzi, M1)$

$Missile(M1)$

به جای ایکس
بالایی بید
M1
بگذاریم

به کمک ثابت اسکلم
په گزاره تبدیل
میکنیم

مازیار پالهنج

به کمک
حذف و
دوتا
جمله را
نتیجه
گرفتیم

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

.3

همه آنها توسط مزدور به آنها فروخته شده است.

$$\text{Missile}(x) \wedge \text{Owns}(\text{Farzi}, x) \Rightarrow \text{Sells}(\text{Mozdour}, \text{Farzi}, x)$$

.4

لازم است بدانیم: تمامی موشکها سلاح هستند.

$$\text{Missile}(x) \Rightarrow \text{Weapon}(x)$$

.5

نیز لازم است بدانیم: دشمن ایران به عنوان یک متخاصم به حساب می آید:

$$\text{Enemy}(x, \text{Iran}) \Rightarrow \text{Hostile}(x)$$

مزدور یک ایرانی است. .6

Iranian(Mozdour)

کشور فرضی دشمن ایران7

Enemy(Farzi,Iran)

پایگاه دانش نماد تابعی ندارد ■

به چنین پایگاهی Datalog گفته می شود. (کلاوزهای معین و بدون نماد تابعی) ■

first order
logic منطق مرتبه اول

آیا جمله‌ی آلفا از پایگاه دانش ایجاد می‌شود؟
اگه ایجاب شد اون جایگزینی که باعث ایجاب شدن شده را برمی‌گردانه

لگوریتم زنجیربندی به جلو

function FOL-FC-ASK(KB, α) **returns** a substitution or *false*
inputs: KB , the knowledge base, a set of first-order definite clauses
 α , the query, an atomic sentence

الفایک جمله‌ی اتمی

کلاوزهای معین در منطق مرتبه اول

while true **do**

$new \leftarrow \{\}$ // The set of new sentences inferred on each iteration

for each rule in KB **do**

$(p_1 \wedge \dots \wedge p_n \Rightarrow q) \leftarrow \text{STANDARDIZE-VARIABLES}(rule)$

for each θ such that $\text{SUBST}(\theta, p_1 \wedge \dots \wedge p_n) = \text{SUBST}(\theta, p'_1 \wedge \dots \wedge p'_n)$

for some p'_1, \dots, p'_n in KB

$q' \leftarrow \text{SUBST}(\theta, q)$

if q' does not unify with some sentence already in KB or new **then**

add q' to new

$\phi \leftarrow \text{UNIFY}(q', \alpha)$

if ϕ is not failure **then return** ϕ

اول باید استاندارد سازی متغیرها را انجام بده یعنی متغیرهای باید اسم های متفاوتی در هر قانون داشته باشند

اگه جملات جدیدی بدست امدن اونهارا به پایگاه اضافه کن و لگزنجیربندی به جلو را انجام بده تا چک کنیم آیا میشه از مقدم ها به تالی برسیم که موردنظر سوال است؟

if $new = \{\}$ **then return** *false*

add new to KB

کیوپریم تالی است که

بدست امده

آیا قبلا هم داشتیم؟

اگه جدیده به نیو اضافش میکنیم

۱۴۰۱-۰۲ - نیمسال اول

مصنوعی

تالی این قانون که

کیو است را نتیجه

میگیریم که جداساز تنا روشن اعمال شده

آیا کیوپریم با جمله‌ی آلفا قابل یکسان سازی است؟
اگه قابل یکسان سازی بود یعنی آلفا از پایگاه دانش با جایگزینی فی ایجاب می‌شده پس جایگزینی فی را برگردان

اگه نیو تهی است یعنی همه‌ی قانون ها را گشتم و هیچ جمله‌ای پیدا نکردیم که با آلفا یکسان بشه پس فالس برگردان

زنگیربندی به جلو

چندتا جمله‌ی اتمی در پایگاه دانش به شکل زیر داریم
از این جمله‌ها، کدام هاشون با مقدم قانون‌ها میتوانند یکسان شوند؟

Iranian(Mozdour) Missile(M1) Owns(Farzi,M1) Enemy(Farzi,Iran)

یه قانون دیگه داشتیم که میگفت اگه ایکس موشک باشه و فرضی اون را در اختیار داشته باشه میفهمیم که مزدور اون را به فرضی فروخته این دوتا با مقدم اون قانونی که داشتیم یکسان میشه

پس به جای ایکس میداریم

M1

و تالی اون قانون را نتیجه میگیریم که به جای متغیرش

m1

گذاشتیم

ذنجیربندی به جلو

قانون داشتیم که هرچی که موشک باشه سلاح هم هست

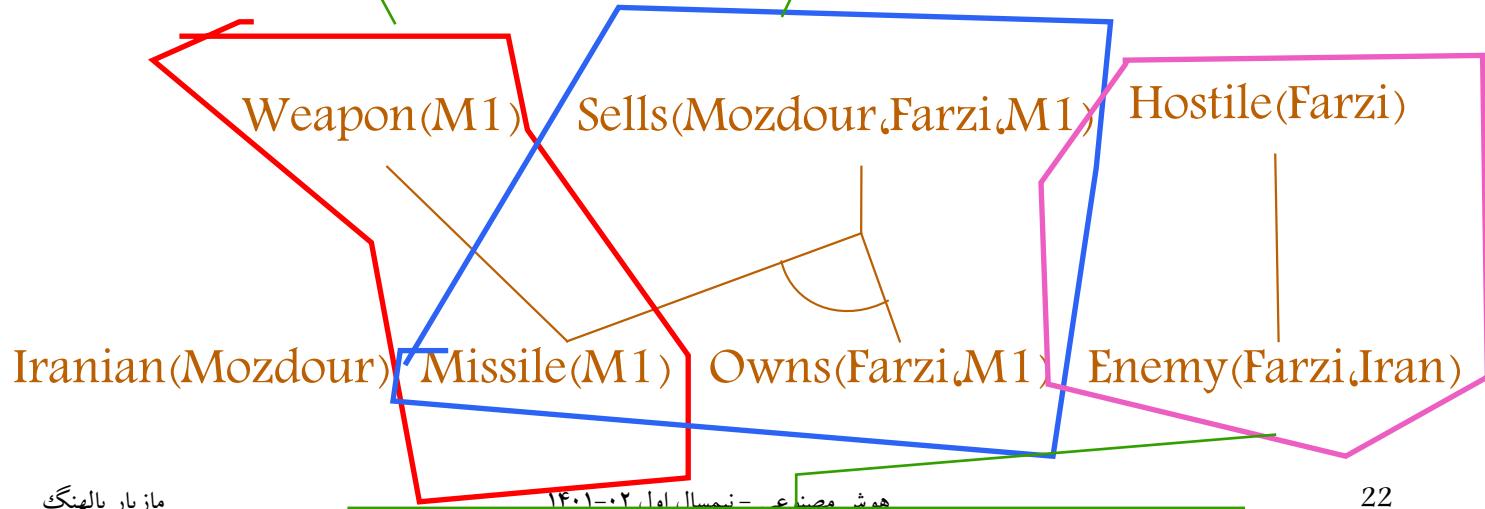
سره

Missle(M1)

با مقدم اون یکسان میشه که میشه تالی ش را نتیجه گرفت و به جای ایکس در تالی همان را گذاشت

M1

را گذاشت



مازیار پالهنجک

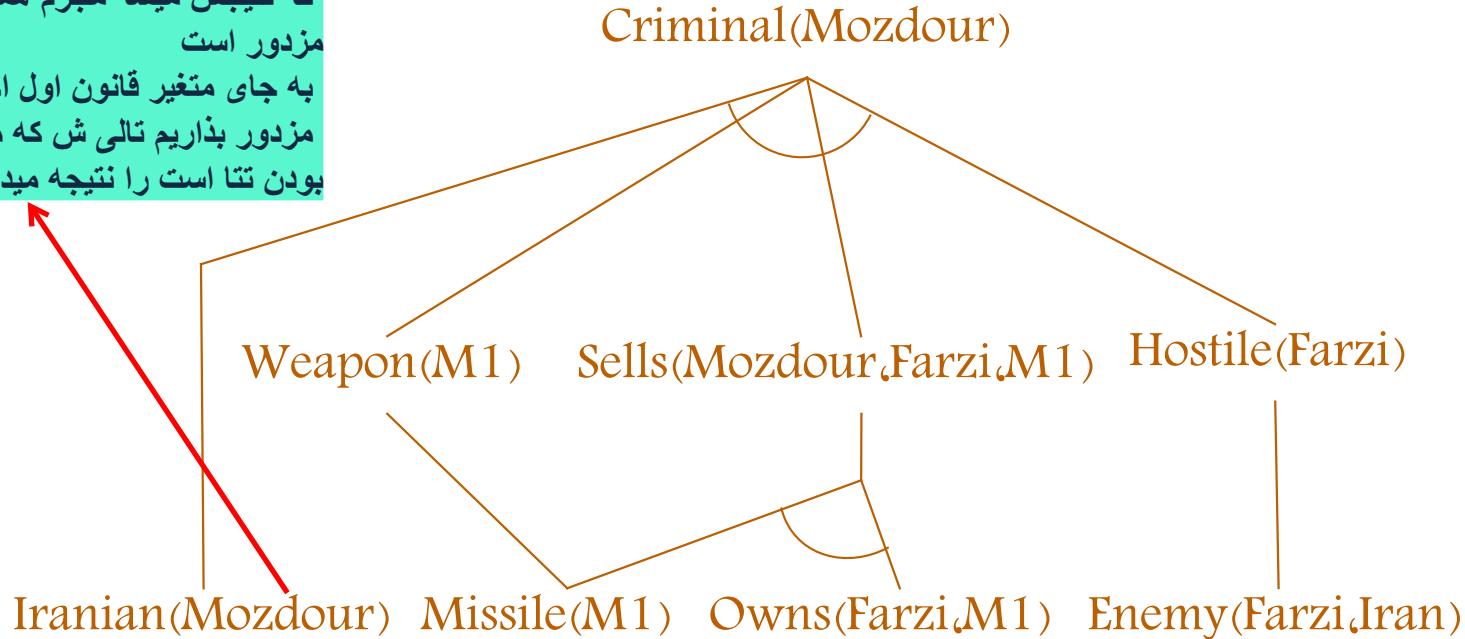
همش مصنوع - نمسا، آوا، ۱۴۰۱-۰۲

22

یه قانون داشتیم که میگفت دشمن متخاص است اینکه فرضی دشمن ایران است با مقدم اون قانون یکسان سازی میشه و نتیجه گرفته میشه که تالی به ازای ایکس مساوی فرضی برقرار است

زنگیربندی به جلو

یکسان سازی شدن همه ای اینها
با قانون اول
که نتیجش میشه مجرم همان
مزدور است
به جای متغیر قانون اول اگه
مزدور بذاریم تالی ش که مجرم
بودن تنا است را نتیجه میده



زنگیربندی به جلو

- موثق و کامل برای کلاوزهای معین مرتبه اول
- در حالت کلی اگر α ایجاب نشود ممکن است خاتمه نیابد.

جایگزینی پیدا شده در
شروع کار

آیا کوئری ما از
پایگاه دانش استنتاج
میشه؟

جایگزینی ای که
تحالا پیدا کردیم را
بشن میدیم
این میشه جایگزینی
ای که تحالا
پیدا کردیم تا تالی و
هدف یکی بشن

الگوریتم زنجیربندی به عقب

```
function FOL-BC-ASK(KB, query) returns a generator of substitutions
  return FOL-BC-OR(KB, query, {})
```

```
function FOL-BC-OR(KB, goal, θ) returns a substitution
  for each rule in FETCH-RULES-FOR-GOAL(KB, goal) do
     $(lhs \Rightarrow rhs) \leftarrow STANDARDIZE-VARIABLES(rule)$ 
    for each  $\theta'$  in FOL-BC-AND(KB, lhs, UNIFY(rhs, goal, θ)) do
      yield  $\theta'$ 
```

```
function FOL-BC-AND(KB, goals, θ) returns a substitution
  if  $\theta = failure$  then return
  else if LENGTH(goals) = 0 then yield  $\theta$ 
  else
    first, rest  $\leftarrow FIRST(goals), REST(goals)$ 
    for each  $\theta'$  in FOL-BC-OR(KB, SUBST( $\theta$ , first),  $\theta$ ) do
      for each  $\theta''$  in FOL-BC-AND(KB, rest,  $\theta'$ ) do
        yield  $\theta''$ 
```

آیا فیرست توى تالی
قانونی پیدا میشه یا
ن؟

توى قانون ها
میگردیم که
کدومشون هست که
تالی ش با کوئری ما
میتونه یکسان پشه؟
اون قانون هایی که
یکسان میشه را
میاریم بیرون و یه
حلقه روی تک تک
اون قانون ها میزنیم
هر قانونی که
پیدا کردیم متغیرهاشو
استاندارد میکنیم تا
اسم هاشون با
متغیرهای دیگه یکی
نبشه
درنهایت قانون زیر
درست میشه
 $lhs \Rightarrow rhs$

تک تک مولفه های
مقدم این را میبینم که
ایجاب میشن یا نه؟
ماز

زمانی که بخاهیم تالی های قانون ها را چک کنیم
or
را صدا میزنیم
وقتی میخاهیم بدنه ای قانون ها را چک کنیم چون شامل چندتا مولفه است که باهم اند
شدن باید
and
را صدا کنیم

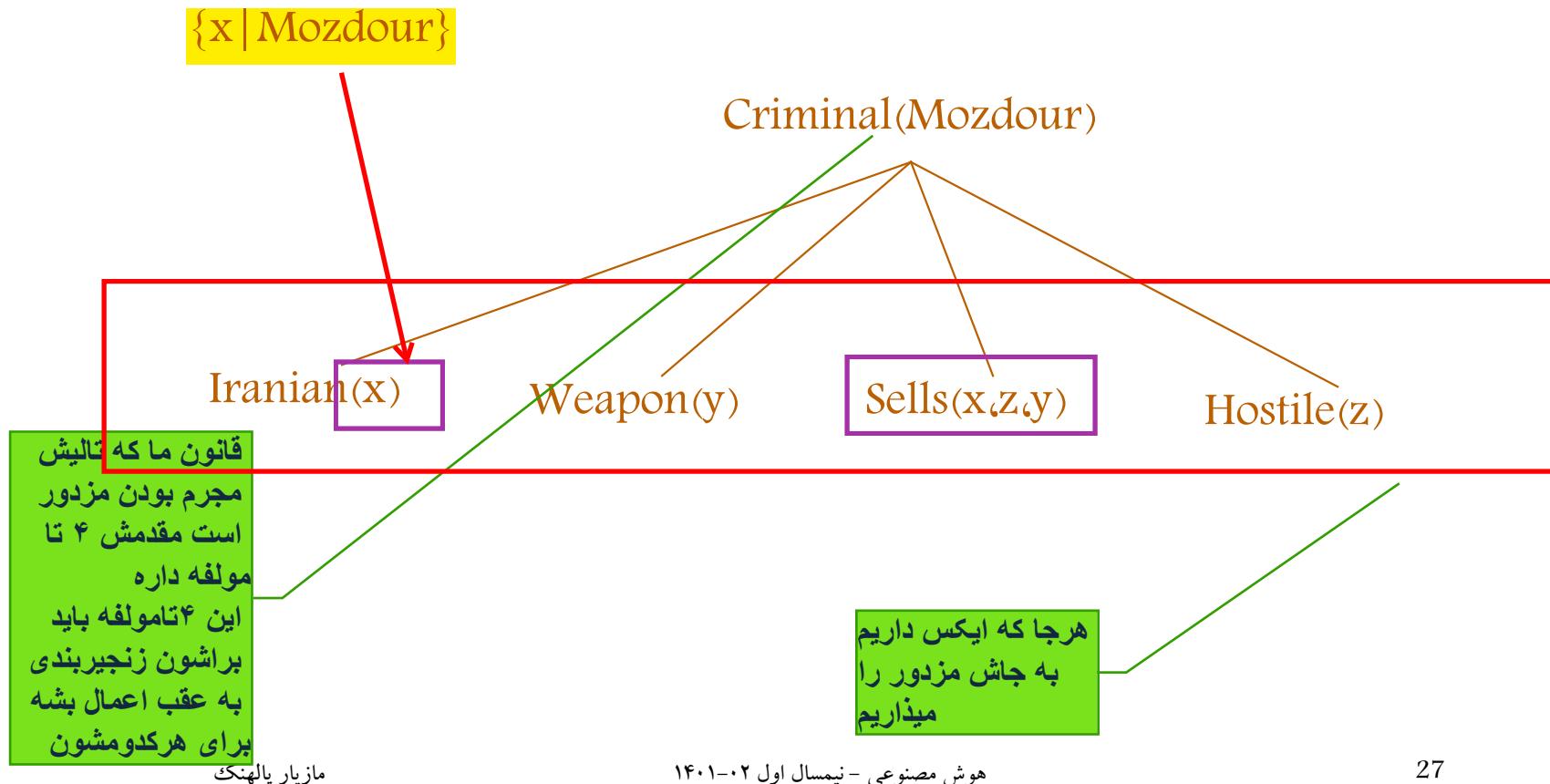
زنجبندی به عقب

Criminal(Mozdour)

سوال

آیا مزدور مجرم هست؟

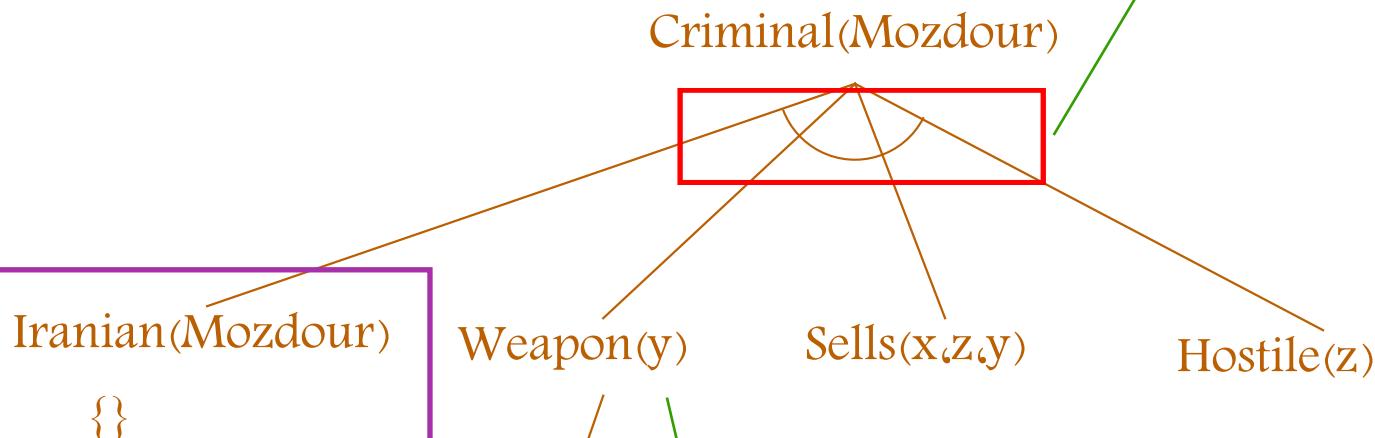
زنگیربندی به عقب



زنجریندی به عقب

{x | Mozdour}

and
همشون باید چک
بشن

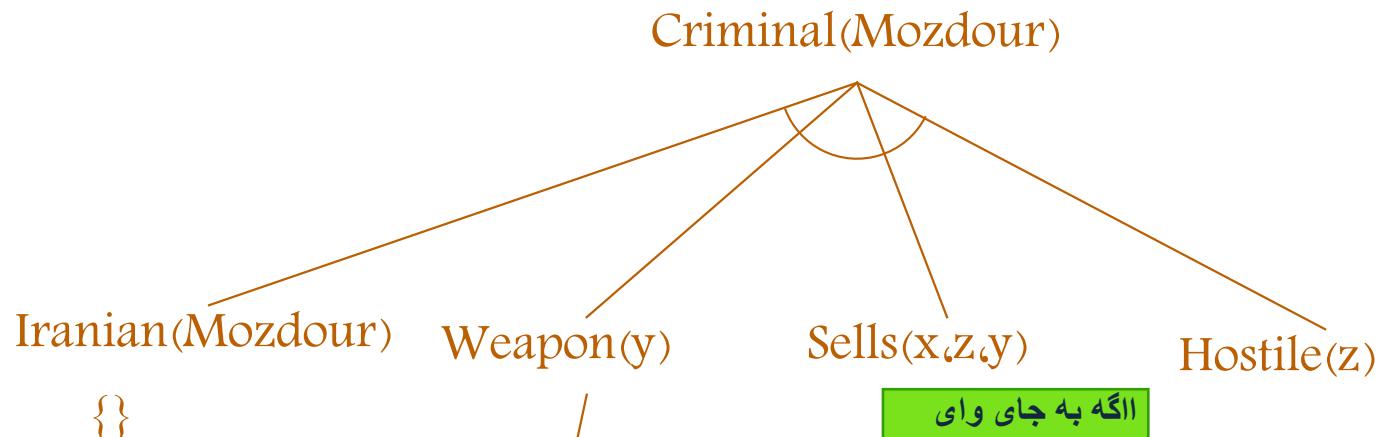


زنجریندی به عقب را ببریم میفهمیم که
به جای ایکس که مزدور بذاریم وقتی
زنجریندی به عقب را ببریم میفهمیم که
توی پایگاه دانش هست
Iranian(mozdour)
توی پایگاه دانش هست

مقدمش را باید چک
کنیم که مقدمش
Missle(y)
است

زنگیربندی به عقب

$\{x \mid Mozdour.y \mid M1\}$



$\{y \mid M1\}$

اگه به جای وای
بذاریم
M1
با یکی از جملات
پایگاهمون یکی
میشه پس مقدم را
نتیجه گرفتیم پس
تالی هم نتیجه
میگیریم یعنی باید به
جای وای بذاریم
M1

به جای زد
میذاریم فرضی

زنجربندی به عقب

{x | Mozdour.y | M1.z | Farzi}

Criminal(Mozdour)

Iranian(Mozdour)

Weapon(M1)

Sells(Mozdour,z,M1)

{}

{z | Farzi}

Missile(M1)

{y | M1}

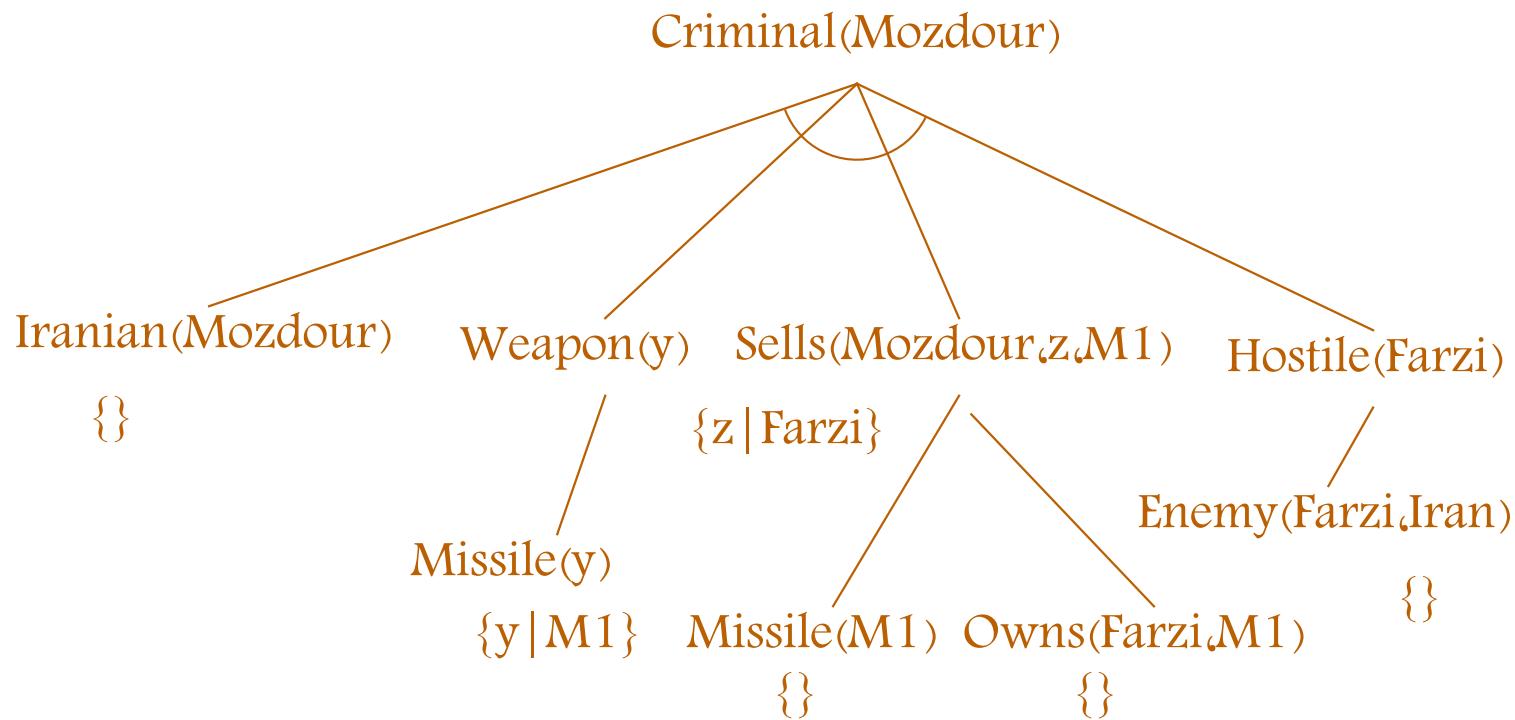
Missile(M1)

Owns(Farzi,M1)

Hostile(z)

زنجرندی به عقب

$\{x \mid Mozdour, y \mid M1, z \mid Farzi\}$



زنجیربندی به عقب

- عمل بصورت عمق نخست
- ناکامل بخاراطر امکان حلقه های بی پایان
- استفاده شده در برنامه نویسی منطقی

مثل پرولوگ

خلاصه

ایجاد تعداد زیادی
جملات نامریوط و
حجم زیادی میگیره

- چگونگی استنتاج در منطق مرتبه اول
- تبدیل به یک پایگاه منطق گزاره ای
 - استفاده از حذف عمومی
 - استفاده از حذف وجودی
- عدم کارآئی مناسب
- استفاده از قانون انتزاع تعمیم یافته
- بکارگیری بصورت زنجیربندی به جلو
- بکارگیری بصورت زنجیربندی به عقب

پایان



م. پالهنج

دانشگاه صنعتی اصفهان - مجموعه مفخر اصفهان

مازیار پالهنج

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

34

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی

استنتاج در منطق مرتبه اول - ۲

نیمسال اول ۱۴۰۱-۱۴۰۲

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

- تبدیل به پایگاه دانش گزاره‌ای
- قانون انتزاع تعییم یافته
- یکسان‌سازی
- جداسازی استاندارد
- عمومی‌ترین یکسان‌ساز
- زنجیربندی به جلو
- زنجیربندی به عقب

Resolution تحلیل

■ در حالت منطق مرتبه اول

$$\frac{\ell_1 \vee \dots \vee \ell_k, \quad m_1 \vee \dots \vee m_n}{\text{SUBST}(\theta, \ell_1 \vee \dots \vee \ell_{i-1} \vee \ell_{i+1} \vee \dots \vee \ell_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n)}$$

یکی از جملات کلاوز سمت چپ بتوانه با نقیض یکی از لیترال های سمت راست یکسان بشه ینی یه ترتیبی پیدا بشه که بتوونه این دوتا را مثل هم بکنه پس میتونیم این دوتا لیترال را حذف کنیم و بقیه لیترال ها را باهم فصل کنیم و یک کلاوز جدیدی ایجاد کنیم ولی بیسی مازیا باید اون یکسان سازی که بالا استفاده شد روی این کلاوز جدیده اعمال بشه

$$\text{UNIFY}(\ell_i, \neg m_j) = \theta$$

فرض می شود که دو کلاوز جدا سازی استاندارد شده اند

$$\text{Rich}(Amin)$$

$$\text{Unhappy}(Amin)$$

قانون تحلیل را به $\text{CNF}(\text{KB} \wedge \neg \alpha)$ اعمال می کنیم.

برای م.ا. کامل است.

این دوتا چون نقیض هم هستند پس میتونیم یکسانشون کنیم و یه یکسان ساز بدست بیاریم

یکسان سازشون میشه امین که به جای ایکس باید بذاریم امین پس بقیه ی کلاوز ها را به جز این دوتا با هم باید فصل کنیم و به جای متغیر ایکس بذاریم امین پس به کمک قانون تحلیل یا رزولوشن نتیجه میشه که این خوشحال نیست

- وقتی میخاییم ببینیم که آلفا از پایگاه دانش ایجاد میشه یا نه باید نقیض آلفارا به پایگاه دانش اضافه کنیم بعدش پایگاه رو به شکل **CNF** درمیاریم بعد میبینیم آیا به تناض میکنیم یا کلاوز تهی میرسیم یا نه؟

تبديل به شكل عطفى عادي

■ هر کسی که همه حیوانات را دوست دارد، کسی او را دوست دارد:

$$\forall x [\forall y Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y Loves(y,x)]$$

1- حذف دو شرطی و شرطی:

$$\forall x [\neg \forall y \neg Animal(y) \vee Loves(x,y)] \vee [\exists y Loves(y,x)]$$

2- حرکت \neg به داخل:

$$\begin{aligned} \forall x [\exists y (\neg Animal(y) \vee Loves(x,y))] \vee [\exists y Loves(y,x)] \\ \forall x [\exists y \neg \neg Animal(y) \wedge \neg Loves(x,y)] \vee [\exists y Loves(y,x)] \\ \forall x [\exists y Animal(y) \wedge \neg Loves(x,y)] \vee [\exists y Loves(y,x)] \end{aligned}$$

تبدیل به شکل عطفی عادی

■ ۳- جداسازی استاندارد متغیرها:

$$\forall x [\exists y Animal(y) \wedge \neg Loves(x,y)] \vee \exists z Loves(z,x)$$

■ ۴- اسکلم کردن (شکل کلی تر حذف وجودی)
■ باید دقت کرد:

ما میخاهیم این سورهای وجودی که داریم را حذف کنیم
در روشهای قبل اگفته بودیم
باید سوره را حذف میکردیم
و به جای آون متغیری که
توی سور وجودی استفاده
شده یه ثابت اسکلم میداشتیم
ولی اینجا باید دقت کنیم و از
تابع اسکلم استفاده کنیم نه
ثابت!

مثالی از تابع اسکلم

- هر کسی یک قلب دارد

$$\forall x \text{Person}(x) \Rightarrow \exists y \text{Heart}(y) \wedge \text{Has}(x, y)$$

سور وجودی را
حذف کردیم و به
جای متغیرش یه
ثابت اسکلم
گذاشتیم
یعنی داریم میگیم
همه ادم ها یه قلب
واحد دارن که
یکسان است خب
این اشتباهه

- اگر دقت نکنیم:

$$\forall x \text{Person}(x) \Rightarrow \text{Heart}(H) \wedge \text{Has}(x, H)$$

این اج ثابتی است که
نشان دهنده یک قلب
واحد است برای همه

- با استفاده از تابع اسکلم

$$\forall x \text{Person}(x) \Rightarrow \text{Heart}(F(x)) \wedge \text{Has}(x, F(x))$$

$F(x)$
ینی اون شی ای که
مربوط به ایکس
است که قلب است
هوش مصنوعی -

مازیار پالهنجک

اگه با ثابت اسکلم
بریم جلو ینی
همه یه حیوان خاصی
را دوست ندارند یا
یه فرد خاصی همه ی
اون افراد را دوست
داره که این منظور ما
نیست
پس از تابع اسکلم
استفاده میکنیم

تبدیل به شکل عطفی عادی

۳- جداسازی استاندارد متغیرها

$$\forall x [\exists y Animal(y) \wedge \neg Loves(x, y)] \vee [\exists z Loves(z, x)]$$

۴- اسکلم کردن (شکل کلی تر حذف وجودی)
باید دقت کرد:

$$\forall x [Animal(A) \wedge \neg Loves(x, A)] \vee Loves(B, x)$$


هر متغیر وجودی با یک تابع اسکلم از متغیرهای عمومی که در آن قرار گرفته جایگزین می شود.

این سور عمومی
حذف میشه

$$\forall x [Animal(F(x)) \wedge \neg Loves(x, F(x))] \vee Loves(G(x), x)$$

تبدیل به شکل عطفی عادی

۵- انداختن سورهای عمومی ■

$$[Animal(F(X)) \wedge \neg Loves(X, F(X))] \vee Loves(G(X), X)$$



۶- توزیع \vee روی \wedge ■

$$[Animal(F(X)) \vee Loves(G(X), X)] \wedge \\ [\neg Loves(X, F(X)) \vee Loves(G(X), X)]$$

اگه به جای ایکس بذاریم مزدور
یکسان سازی انجام دادیم بعدش که
رزولوشن یا قانون تحلیل انجام بشه
این دوتا که نقیض هم هستند باهم
میرن و بقیه میمونن که باید هرجا که
ایکس باشه مزدور بذاریم به جاش

بازدید مثال قبل

$$\begin{array}{l} \neg \text{Iranian}(x) \vee \neg \text{Weapon}(y) \vee \\ \neg \text{Hostile}(z) \vee \neg \text{Sells}(x, z, y) \vee \text{Criminal}(x) \end{array}$$

$$\neg \text{Criminal}(\text{Mozdour})$$

$$\text{Iranian}(\text{Mozdour})$$

$$\begin{array}{l} \neg \text{Iranian}(\text{Mozdour}) \vee \neg \text{Weapon}(y) \vee \\ \neg \text{Hostile}(z) \vee \neg \text{Sells}(\text{Mozdour}, z, y) \end{array}$$

توی پایگاه دانش
اینو داریم

$$\neg \text{Missile}(x) \vee \text{Weapon}(x)$$

$$\begin{array}{l} \neg \text{Weapon}(y) \vee \neg \text{Hostile}(z) \vee \\ \neg \text{Sells}(\text{Mozdour}, z, y) \end{array}$$

$$\text{Missile}(M1)$$

$$\begin{array}{l} \neg \text{Missile}(y) \vee \neg \text{Hostile}(z) \vee \\ \neg \text{Sells}(\text{Mozdour}, z, y) \end{array}$$

$$\begin{array}{l} \neg \text{Missile}(x) \vee \neg \text{Owns}(\text{Farzi}, x) \vee \\ \text{Sells}(\text{Mozdour}, \text{Farzi}, x) \end{array}$$

$$\begin{array}{l} \neg \text{Hostile}(z) \vee \\ \neg \text{Sells}(\text{Mozdour}, z, M1) \end{array}$$

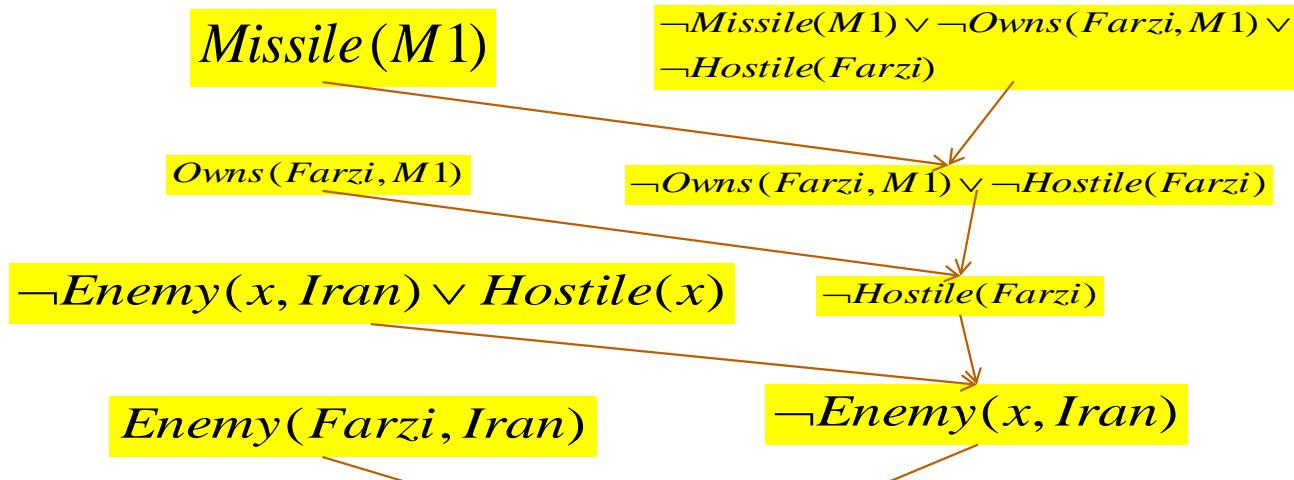
$$\text{Missile}(M1)$$

$$\begin{array}{l} \neg \text{Missile}(M1) \vee \neg \text{Owns}(\text{Farzi}, M1) \vee \\ \neg \text{Hostile}(\text{Farzi}) \end{array}$$

قانون تحلیل یا
رزولوشن انجام
میشه

قانونی که در پایگاه
دانش به صورت
CNF
درآومده
مازیار پالهنه

بازدید مثال قبل



مثال

- هر کسی همه حیوانات را دوست داشته باشد کسی او را دوست دارد.
- هر کسی که حیوانی را بکشد هیچکس او را دوست ندارد.
- امین همه حیوانات را دوست دارد.
- امین یا امیر گربه ای به نام گیارا کشتند.
- آیا امیر گربه را کشت؟

با منطق مرتبه اول میخایم بفهمیم امیر گربه را کشته
یا نه؟

پس جملات بالا را باید درگام اول به منطق مرتبه اول
تبدیل کنیم



- A. $\forall x [\forall y Animal(y) \Rightarrow Loves(x, y)] \Rightarrow [\exists y Loves(y, x)]$
- B. $\forall x [\exists y Animal(y) \wedge Kills(x, y)] \Rightarrow [\forall z \neg Loves(z, x)]$
- C. $\forall x Animal(x) \Rightarrow Loves(Amin, x)$
- D. $Kills(Amin, G1) \vee Kills(Amir, G1)$
- E. $Cat(G1)$
- F. $\forall x Cat(x) \Rightarrow Animal(x)$
- G. $\neg Kills(Amir, G1)$

همان گربه گ

باید صریحاً بگوییم که
گربه ها حیوان هستند

نقیض سوالمون را
به پایگاه دانش
اضافه میکنیم

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

12

جمله های بالا را باید به

cnf

تبديل کنیم

حذف شرطی و دوشرطی

حذف سورهای وجودی و عمومی

از تابع اسکلم اگ نیازه استفاده کن

نقیض ها را اعمال میکنیم از قانون دمورگان هم استفاده کن

- A1. $Animal(F(x)) \vee Loves(G(x), x)$
- A2. $\neg Loves(x, F(x)) \vee Loves(G(x), x)$
- B. $\neg Animal(y) \vee \neg Kills(x, y) \vee \neg Loves(z, x)$
- C. $\neg Animal(x) \vee Loves(\text{Amin}, x)$
- D. $Kills(\text{Amin}, \text{G1}) \vee Kills(\text{Amir}, \text{G1})$
- E. $Cat(\text{G1})$
- F. $\neg Cat(x) \vee Animal(x)$
- ¬G. $\neg Kills(\text{Amir}, \text{G1})$

اعمال قانون تحلیل روی جمله های بالا

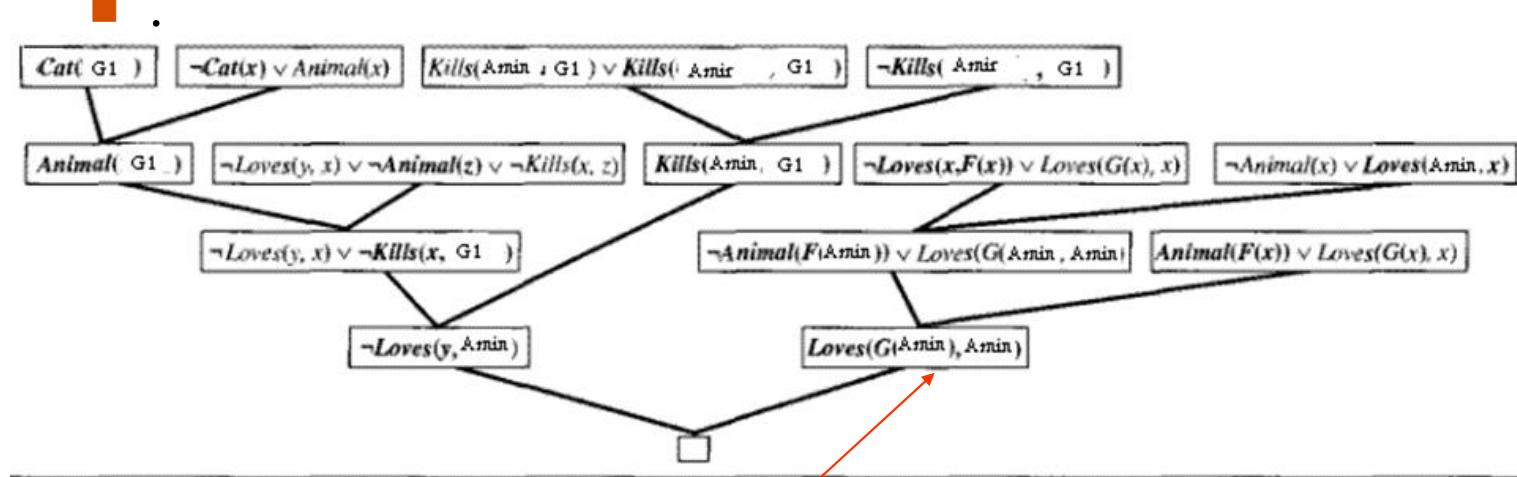


Figure 9.12 A resolution proof that Amir killed the cat. Notice the use of factoring in the derivation of the clause $Loves(G(Amin), Amin)$.

- اغلب علاقمند هستیم که بدانیم چه کسی گ1 را کشت؟
- سؤال بصورت $\exists w \text{ Kills}(w, G1)$
- شکل $\neg \text{Kills}(w, G1) : \text{CNF}$
- با اثباتی مشابه با جایگزینی $\{w / \text{Amir}\}$

$$\boxed{\text{Kills}(\text{Amin} , G1) \vee \text{Kills}(\text{Amir} , G1)}$$

$$\boxed{\neg \text{Kills}(\text{w} , G1)}$$

- مشکل: تحلیل با $\text{Kills}(\text{Amin}, \text{G1}) \vee \text{Kills}(\text{Amir}, \text{G1})$
- با $\{\text{w}/\text{Amin}\}$ به $\text{Kills}(\text{Amir}, \text{G1})$ می‌رسیم
- دوباره تحلیل با $\neg \text{Kills}(\text{w}, \text{G1})$ به تهی می‌رسیم!

وقتی که به جای

W

گذاشتیم امیر و به جواب نرسیدیم یه
عقبگرد کنیم که ایا میشد به جای

W

یه مقدار دیگه به جز امیر قرار بدم؟

■ حل ۱: محدود کنیم که جایگزینی برای متغیرهای سؤال فقط یک بار انجام شود.

■ با عقبگرد به جواب صحیح می‌رسیم

■ حل ۲: اضافه کردن لیترال پاسخ و نتیجه هر گاه به لیترال پاسخ تک رسیدیم.

■ در مورد مثال قبل: $\neg \text{Kills}(w, G1) \vee \text{Answer}(w)$

■ با جایگزینی صحیح به $\text{Answer}(\text{Amir})$ می‌رسیم.

■ با جایگزینی نادرست به $\text{Answer}(\text{Amin}) \vee \text{Answer}(\text{Amir})$

برخورد با برابری

- اگر در پایگاه داشته باشیم:
 - $A=B$
 - $P(A)$
 - سؤال: $P(X)?$
 - فقط $X=A$ پاسخ خواهیم گرفت.
- چه چیزهایی به جای
ایکس میتوانه قرار
بگیره؟

برخورد با برابری

- یک روش ایجاد چند اصل برابری:

- خواص انعکاسی، تقاریقی، و تراویائی:

$$\forall x \ x = x$$

$$\forall x, y \ x = y \Rightarrow y = x$$

$$\forall x, y, z \ x = y \wedge y = z \Rightarrow x = z$$

$$\forall x, y \ x = y \Rightarrow (P_1(x) \Leftrightarrow P_1(y))$$

$$\forall x, y \ x = y \Rightarrow (P_2(x) \Leftrightarrow P_2(y))$$

⋮

$$\forall w, x, y, z \ w = y \wedge x = z \Rightarrow (F_1(w, x) = F_1(y, z))$$

$$\forall w, x, y, z \ w = y \wedge x = z \Rightarrow (F_2(w, x) = F_2(y, z))$$

⋮

- و برای هر مسند و تابع:

مشکل این روش در
برخورد با برابری

بر خود با برابری

این سه تا ارجومان
میگیره

:demodulation

برای هر ترم X, y , و Z که Z در جائی از لیترال m_i ظاهر شده

$$\text{Unify}(X, Z) = \theta$$

برای یک لیترال تک

$$\frac{x = y, \quad m_1 \vee \dots \vee m_n}{\text{SUB}(\text{SUBST}(\theta, x), \text{SUBST}(\theta, y), m_1 \vee \dots \vee m_n)}$$

که $\text{SUB}(x, y, m)$ یعنی جایگزینی x با y هر جا که x در m ظاهر شده است.

اون چیزی که باعث
شده ایکس و زد باهم
یکسان بشن رو
باید هم روی ایکس
هم وای اعمال بشه

هرجایی که ایکس
توی ام حضور داره
به جاش وای بذاریم
هوش مصنوعی

برخورد با برابری

اگه به جای ایکس بذاریم
امین یکسان میشه جمله
ها

■ مثال: با داشتن:

$$\begin{aligned} Father(Father(x)) &= PaternalGrandfather(x) \\ Birthdate(Father(Father(\textcolor{red}{Amin})), 1320) \end{aligned}$$

■ می توانیم نتیجه بگیریم:

$$Birthdate(PaternalGrandfather(\textcolor{brown}{Amin}), 1320)$$

به کمک قانون
demodulation

برخورد با برابری

- بسط قانون قبلی به کلاوزهای غیر تک، بنام قانون paramodulation
- برای هر ترم X, y و Z که Z در جائی از لیترال m_i ظاهر شده است و $\text{Unify}(X, Z) = \theta$

$$\frac{\ell_1 \vee \dots \vee \ell_k \vee x=y, \quad m_1 \vee \dots \vee m_n}{\text{SUB}(\text{SUBST}(\theta, x), \text{SUBST}(\theta, y), \text{SUBST}(\theta, \ell_1 \vee \dots \vee \ell_k \vee m_1 \vee \dots \vee m_n))}.$$

اعمال تتراروی ایکس

اعمال تتراروی وای

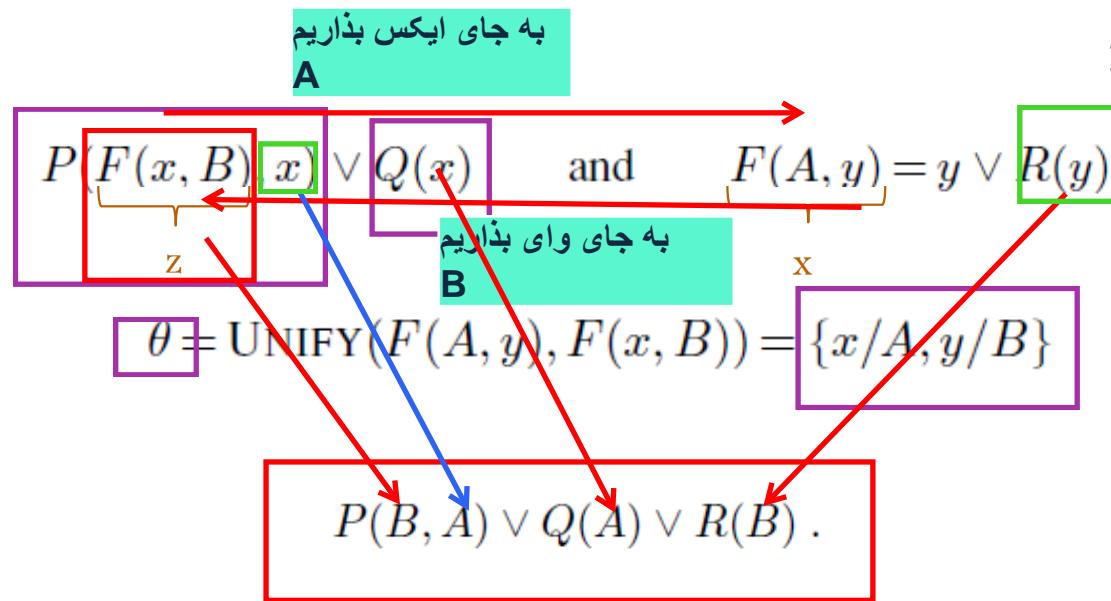
به چای زد
وای را میگذاریم
دققت کن که باید تترارا
روی همه ای لیترال
ها اعمال کنی

به کمک قانون
paramodulation

ایا میشه زد را با ایکس یکسان
کرد؟

برخورد با برابری

مثال: ■



خلاصه

- تحلیل در منطق مرتبه اول
- تبدیل به شکل عادی عطفی
- استفاده از تابع اسکلم
- برحورده با برابری

پایان



م.پالهنگ

دانشگاه صنعتی اصفهان - مجموعه مفاخر اصفهان

مازیار پالهنگ

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

25

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی

عدم قطعیت - ۱

نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنجک

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

مقدمه

- هنگامی که عامل واقعیتهای کافی در مورد محیطش می‌داند روش منطقی عامل را قادر می‌سازد که طرحهای را بدست آورد که ضمانت می‌دهند که کار کنند.
- در عمل اینگونه نیست.
- در این حالت، عمل تحت عدم قطعیت
- مثال دنیای دیو:

- محیط نیمه مشاهده پذیر
- عامل همیشه نمی‌داند کدام خانه گودال است
- ممکن است مجبور به قبول ریسک شود

- قبل از فضای باور برای برشورد با عدم قطعیت استفاده می کردیم.
- مشکلات:
 - فضای باور شامل همه حالت‌های ممکن است که می تواند بسیار بزرگ بوده و شامل حالاتی باشد که کمتر احتمال وقوع دارند.
 - ایجاد یک طرح اقتضائی که همه شرایط را در نظر بگیرد می تواند بسیار پیچیده باشد و شامل وضعیتهاست که کمتر محتمل هستند.
 - گاهی طرحی که صدرصد به هدف برسد وجود ندارد ولی باید کاری انجام داد. باید راهی برای مقایسه شایستگی چنین طرحهای وجود داشته باشد.

- دنیای واقعی بسیار پیچیده تر
- فرض کنید عامل می خواهد کسی را به فرودگاه برساند تا به پروازش به موقع برسد.
- طرح A90: ترک خانه ۹۰ دقیقه قبل از پرواز و رانندگی با سرعت معقول
- اگرچه فرودگاه دور نیست ولی عامل با قطعیت نمی تواند نتیجه بگیرد که A90 او را به موقع به فرودگاه خواهد رساند.
- در واقع طرح A90 به فرودگاه می رساند، اگر خودرو خراب نشود، سوخت تمام نشود، هواییما زودتر نرود، ...

- عامل منطقی که نتواند استنتاج کند که دنباله اعمالی او را به اهدافش می رساند نمی تواند عمل کند.

دندان درد

- مثال: تشخیص بیماری، عیب یابی و سایل،...

$Toothache \Rightarrow Cavity.$

کرم خوردگی

- ولی همه دندان دردها به دلیل کرم خوردگی نیست.
- دلیلهای دیگر هم ممکن است:

$Toothache \Rightarrow Cavity \vee GumProblem \vee Abscess \dots$

- تالی لیست نامحدودی از سببها

- قانون بصورت فوق، قانون تشخیصی گفته می شود.
- می توان قانون سببی نوشت:

Cavity \Rightarrow *Toothache.*

- همه کرم خوردگیها باعث درد نمی شوند.
- لیست کردن تمام عواملی که کرم خوردگی باعث دندان درد می شود در مقدم قانون
- امکان این وجود دارد که کرم خوردگی و دندان درد هر دو وجود داشته باشند و به هم مربوط نباشند.

- منطق شکست می خورد به دلایل:
- **تبیلی:** کار زیاد برای کامل کردن لیست مقدم و تالی، سختی استفاده از چنین قانونی
- **نادانی نظری:** عمل پزشکی ممکن است به دانش کامل نرسیده باشد.
- **نادانی عملی:** حتی اگر همه قانونها را بدانیم ممکن است در مورد بیمار خاصی همه آزمونها انجام نشده یا نتواند انجام شود.

- در این حالت **دانش عامل** در بهترین حالت می تواند یک درجه باور به جملات داشته باشد.
- بهترین ابزار: نظریه احتمال
- مثال: احتمال آنکه بیماری کرم خوردگی دندان داشته باشد ۸۰٪ است.
- عقاید بستگی به **ادراکاتی** دارد که عامل تا آن لحظه دریافت کرده است.
- با رسیدن ادراکات بیشتر، درجه اعتقاد عامل تغییر می کند.

- ادارکات دلایلی را تشکیل می دهند که محاسبه احتمالات بر اساس آنها انجام می گیرد.
- قبل از دریافت دلیل درباره احتمال پیشین یا بدون شرط صحبت می کنیم.
- بعد از دریافت دلیل درباره احتمال پسین یا شرطی صحبت می کنیم

عدم قطعیت و تصمیم معقول

- عامل منطقی هر طرحی که او را به هدفش برساند اجرا می کند.
- هر عمل بر حسب اینکه او را به هدفش می رساند قبول یا پذیرش می شود.
- عدم قطعیت وضعیت را تغییر می دهد.
- طرح A90 با احتمال ۹۵٪
- طرح A120 با احتمال ۹۸٪
- طرح A1440 (۲۴ ساعت قبل) با احتمال بیشتر کدام را عامل انتخاب کند؟

- انتخاب بستگی به ترجیح از دست دادن پرواز در مقابل زمان انتظار دارد.
- نظریه سودمندی برای نمایش و استنتاج ترجیحها استفاده می شود.
 - نظریه تصمیم = نظریه احتمال + نظریه سودمندی

دستور زبان

- درجات باور به گزاره ها اعمال می شوند.
- عنصر بنیادی: متغیر تصادفی
- رجوع به بخشی از دنیا که وضعیت آن در ابتدا نامعلوم است.
- هر متغیر تصادفی دامنه ای دارد.
- بولی: همانند کرم خوردگی با دامنه <درست، نادرست>
- cavity را با Cavity=true نمایش می دهیم
- cavity را با Cavity=false نمایش می دهیم.

گستته: همانند هوا با دامنه <آفتابی، بارانی، ابری، برفی>

مقادیر دامنه منفصل و کامل هستند.

Weather=snow را با snow نمایش می دهیم.

پیوسته

گزاره های بنیادی توسط رابطهای منطقی می توانند ترکیب شوند.

مثال Weather=sunny \wedge Cavity=false

sunny \wedge \neg cavity نمایش

متغیرها با اون مقادیری که بخودشون گرفتن دارن یه حالتی
از دنیا را نشان میدن
پس اگهه مثلثاً دوتا متغیر تصادفی داریم مثل
cavity , toothache
اگ فرض کنیم هردو بولین باشن پس ۴ تا حادثه‌ی اتمی
داریم یعنی ۴ تا وضعیت میتوانه دنیای ما داشته باشه

- **حادثه اتمی:** یک مشخص نمودن کامل حالت دنیا که عامل در مورد آن نامطمئن است (یک مدل).
- تصور از حادثه اتمی: انتساب مقادیر خاصی به همه متغیرهایی که دنیا از آن ترکیب شده است.
- اگر دنیائی فقط شامل متغیرهای Cavity و Toothache باشد در این صورت فقط ۴ حادثه اتمی وجود دارند.
- مجموعه همه دنیاهای ممکن: **فضای نمونه (sample space)**

مجموع احتمال های
همه ای حوادث اتمی
یک میشه

احتمال هر حادثه اتمی

یعنی حادثه اتمی
دیگری نیست که
بتونه وجود داشته و
باشه و توی فضای
نمونه ما نباشه

■ Ω : فضای نمونه و ω حادثه اتمی / نقطه نمونه / مدل / دنیای ممکن

$$0 \leq P(\omega) \leq 1 \text{ for every } \omega \text{ and } \sum_{\omega \in \Omega} P(\omega) = 1$$

■ خواص حوادث اتمی:

■ فقط یکی از آنها می تواند برقرار باشد.

■ مجموعه همه حوادث اتمی کامل است. حداقل یکی باید برقرار باشد.

■ یک حادثه A زیر مجموعه ای از Ω می باشد.

■ مثلاً احتمال اینکه مجموع اعداد پرتاب دو تاس برابر ۶ باشد.

■ یک گزاره را اینجا می توان به عنوان یک حادثه تصور نمود.

اگه ۲ و ۴ بیاد میشه
ی حادثه اتمی
اگه ۳ و ۳ بیاد یه
حادثه اتمی میشه

مازیار پالهنجک

میتوانه ترکیب چندتا صنوعی - نیمسال اول ۱۴۰۱-۰۲

حادثه اتمی باشه

15

a
یک گزاره است و
گزاره معادل یه حادثه
است

اصول احتمال

مجموع حوادث اتمی
که اون حادثه اتمی
متعلق باشه به گزاره
یا حادثه $\neg a$

برای هر گزاره A و B

$$0 \leq P(A) \leq 1$$

$$P(\text{false}) = 0 \text{ و } P(\text{true}) = 1$$

فصل دو تا حادثه

$$\begin{aligned} P(\neg a) &= \sum_{\omega \in \neg a} P(\omega) \\ &= \left[\sum_{\omega \in \neg a} P(\omega) + \sum_{\omega \in a} P(\omega) \right] - \sum_{\omega \in a} P(\omega) \\ &= \sum_{\omega \in \Omega} P(\omega) - \sum_{\omega \in a} P(\omega) \\ &= 1 - P(a) \end{aligned}$$

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

$$0 \leq P(\omega) \leq 1 \text{ for every } \omega \text{ and } \sum_{\omega \in \Omega} P(\omega) = 1$$

قانون آخر و

اصول کلمگرف (Kolmogrov) نامیده می شوند.

احتمال پیشین

- درجه باور در غیاب هر اطلاعات دیگر
- مثال $P(\text{sunny})=0.78$, $P(\text{cavity})=0.1$
- گاهی در مورد احتمال تمام مقادیر یک متغیر تصادفی صحبت می کنیم.
- مثال: $P(\text{Weather})$ برداری از مقادیر برای هر حالت هوا

احتمال اینکه هر
مقداری از اون متغیر
تصادفی چقدر را
میخایم

خود متغیر تصادفی
را نوشتیم

$$\begin{aligned}P(\text{Weather} = \text{sun}) &= 0.6 \\P(\text{Weather} = \text{rain}) &= 0.1 \\P(\text{Weather} = \text{cloud}) &= 0.29 \\P(\text{Weather} = \text{snow}) &= 0.01,\end{aligned}$$

یه بردار از مقادیر
احتمال که جمعشون
هم یکه

$$P(\text{Weather}) = \langle 0.6, 0.1, 0.29, 0.01 \rangle$$

- عبارت فوق توزیع احتمال متغیر تصادفی Weather را نشان می دهد.

تا
حادثه‌ی اتمی
میتوانیم داشته باشیم

باید ببینیم چندتا
حادثه‌ی اتمی
میتوانیم داشته باشیم؟

توزیع احتمال توأم

■ توزیع احتمال توأم برای یک مجموعه از متغیرهای تصادفی
احتمال هر حادثه اتمی روی آن متغیرهای تصادفی را نشان می‌دهد.

■ مثال (Weather,Cavity) یک ماتریس 2×4 از مقادیر است.

Weather	sunny	rainy	cloudy	snow
Cavity=true	0.144	0.02	0.016	0.02
Cavity=false	0.576	0.08	0.064	0.08

داره احتمال رخداد یه
حادثه‌ی اتمی را
نشان میده

- توزیع احتمال توأم کامل شامل همه متغیرهای موجود برای توصیف دنیا
- هر سؤال در مورد دامنه از این توزیع قابل پاسخگوئی است.

بی رخداده است

احتمال شرطی

هنگام دریافت دلیل بیشتر

احتمال a به شرط آنکه تمام آنچه که می دانیم b باشد.
 $P(a | b)$

$$P(\text{cavity} | \text{toothache}) = 0.8$$

نماد برای توزیع شرطی

بردار دو عضوی که هر عضو یک
 $P(\text{Cavity} | \text{Toothache})$

بردار دو عضوی است.

توزیع ینی برای یک متغیر تصادفی که ما احتمال رخداد همه مقدار ممکن را مینویسیم

دقت کن چون متغیر میداریم حرف اولشون بزرگه

احتمال شرطی

($p(b) \neq 0$ گاه) $P(a | b) = P(a \wedge b) / P(b)$ ■

$$P(a \wedge b) = P(a | b)P(b) = P(b | a)P(a) \quad ■$$

به این میگیم قانون ضرب

قانون ضرب ■

حالت کلی تر برای کل توزیع ■

$$P(\text{Weather}, \text{Cavity}) = P(\text{Weather} | \text{Cavity})P(\text{Cavity}) \quad ■$$

پی بزرگ به معنای
توزیع است

قانون زنجیری با اعمال مکرر قانون ضرب ■

$$\begin{aligned} P(X_1, \dots, X_n) &= P(X_1, \dots, X_{n-1})P(X_n | X_1, \dots, X_{n-1}) \\ &= P(X_1, \dots, X_{n-2})P(X_{n-1} | X_1, \dots, X_{n-2})P(X_n | X_1, \dots, X_{n-1}) \\ &= \dots \end{aligned}$$

hadthe ی اتمی اگ این
باشه
دندان درد براش
درسته
کشیدن درسته
کرم خوردگی هم
درسته

استنتاج با فهرست کردن

سه تا متغیر تصادفی
داریم

هر موقع او مگا درست
بود فی هم درست
باشه

■ شروع با توزیع احتمال توأم

	toothache		\neg toothache	
	catch	\neg catch	catch	\neg catch
cavity	.108	.012	.072	.008
\neg cavity	.016	.064	.144	.576

احتمال اون حادته
که از یه تعدادی
حادته اتمی ایجاد شده

■ برای هر گزاره φ ، گزاره های اتمی که در آنها درست است را جمع کن:

$$P(\varphi) = \sum_{\omega: \omega \models \varphi} P(\omega)$$

■

مازیار پالهنجک

حوادث اتمی که
متعلق به اون حادته
ما هستند

۱۴۰۱-۰۲ ول

محاسبه ی احتمال
هر گزاره ی فی
باید گزاره های اتمی
که برای اون گزاره
فی درست هستن را
جمع کنیم

استنتاج با فهرست کردن

۴تا
حادثه‌ی اتمی هست
که مقدار دندان درد
که یه حادثه است
توشون درست است

- شروع با توزیع احتمال توأم

	<i>toothache</i>	\neg <i>toothache</i>		
	<i>catch</i>	\neg <i>catch</i>	<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	.108	.012	.072	.008
\neg <i>cavity</i>	.016	.064	.144	.576

- برای هر گزاره φ ، گزاره‌های اتمی که در آنها درست است را جمع کن:

$$P(\varphi) = \sum_{\omega: \omega \models \varphi} P(\omega)$$

■

$$P(\text{toothache}) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2$$

$$P(Y) = \sum_{z \in Z} P(Y, z)$$

سند بالشگ

درنظر گرفتن همهٔ
مقادیر متغیرهای
دیگه توی مسئله

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

متغیرهای دیگه‌ای
ک توی مسئله وجود
داره را هم باید
درنظر گرفت

احتمال دندان درد

24

این یه توزیع توام است

$$P(Y) = \sum_{z \in Z} P(Y, z)$$

■ مجموعه متغیرهای دیگر به جز Y

■ مثال:

$$P(Cavity) = \sum_{z \in \{Catch, Toothache\}} P(Cavity, z)$$

متغیرهای دیگر توی
مسننه
متغیرهای نهان
مساله

$$P(Y) = \sum_z P(Y | z) P(z)$$

■ نوع دیگری از این قانون:

■ به این کار به حاشیه بردن گفته می شود.

marginalization

چون داریم متغیرهای
دیگه را به حاشیه
میبریم

استنتاج با فهرست کردن

- شروع با توزیع احتمال توأم ■

بدست اوردن احتمال
های شرطی به کمک
توزیع توأم

		<i>toothache</i>	\neg <i>toothache</i>
		<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	<i>catch</i>	.108	.012
	\neg <i>catch</i>	.072	.008
\neg <i>cavity</i>	<i>catch</i>	.016	.064
	\neg <i>catch</i>	.144	.576

- همچنین می توان احتمالهای شرطی را محاسبه نمود: ■

$$P(\neg cavity | toothache) = \frac{P(\neg cavity \wedge toothache)}{P(toothache)} = \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4 .$$

استنتاج با فهرست کردن

شروع با توزیع احتمال توأم ■

		<i>toothache</i>	\neg <i>toothache</i>		
		<i>catch</i>	\neg <i>catch</i>	<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	<i>catch</i>	.108	.012	.072	.008
	\neg <i>catch</i>	.016	.064	.144	.576

همچنین می توان احتمالهای شرطی را محاسبه نمود: ■

$$\begin{aligned} P(\text{cavity} | \text{toothache}) &= \frac{P(\text{cavity} \wedge \text{toothache})}{P(\text{toothache})} \\ &= \frac{0.108 + 0.012}{0.108 + 0.012 + 0.016 + 0.064} = 0.6 \end{aligned}$$

اینجا یکی از
متغیرهای مساله که
catch
است را در نظر
نگرفته پس سیگما را
روی مقادیر مختلف
کج باید بگیریم که کج
یه بار درسته یه بار
نادرسته

متغیر سوال
کویتی است
دلیلمن
toochache
متغیر مخفیمون کج
است

سی بزرگه
C

تی کوچیکه
t

عادی سازی

■ مخرج کسر را می توان به عنوان ثابت عادی ساز در نظر گرفت

$$P(Cavity / toothache) = \alpha P(Cavity, toothache)$$

$$= \alpha [P(Cavity, toothache, catch) + P(Cavity, toothache, \neg catch)]$$

$$= \alpha [<0.108, 0.016> + <0.012, 0.064>]$$

$$= \alpha <0.12, 0.08> = <0.6, 0.4>$$

	toothache		\neg toothache	
	catch	\neg catch	catch	\neg catch
cavity	.108	.012	.072	.008
\neg cavity	.016	.064	.144	.576

چون سی کویتی
بزرگه هم باید برای
مقادیر درستش
بنویسیم هم نادرست
پس یه بردار میخایم

برای جمع کردن
وقتی که کویتی
درسته باهم جمع
میشه وقتی نادرسته
باهم

ایده کلی: محاسبه توزیع روی متغیر سوال با ثابت در نظر گرفتن دلیل و

جمع گرفتن روی متغیرهای مخفی

حالت برای کویتی
درست میشه
0.108
برای نادرست
0.016
مازیار بالنه

$$0.12/(0.12+0.08) = \\ 0.12/0.2 = 0.6 \\ \Rightarrow \alpha = 1/0.2 = 5$$

هوش مصنوعی - نیمسال

الفاضrib عادی
سازه پس باید یه
کاری کنه جمع مقادیر
نهایی یک بشه
۰.۶ + ۰.۴ = ۱
پس الفا چیه؟

اگ یه توزیع شرطی بمون دادن یه متغیر سوال داریم یه دلیل هم بمون دادن به شرط دلیل که دلیل را که بخایم حساب کنیم یه سری متغیر مخفی هم داریم که میشه سایر متغیرهایی که توی مساله هستند ولی ما استفاده نکردیم متغیر مخفی مثل قبل کج است پس روی مقادیر مختلف کج باید جمع را انجام بدیم

$$P(X | e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$$

X، e، و Y با هم شامل همه متغیرهای موجود می شوند. ■

متغیر سوال

دلیل

ضریب عادی ساز

سایر متغیرهای نهان

خلاصه

منظور از فهرست
بکردن
پیدا کردن حوادث اتمی
که حادثه‌ی مورد نظر
ما برآشون درست
است را پیدا کنیم
مقادیر شون را
فهرست کنیم و باهم
جمع کنیم

نظريه‌ی احتمال به
نهایی کامل نیست و
کافی نیست تا عامل
تصمیم‌گیری را انجام
بده

وقتی میخاییم احتمال
شرطی را حساب کنیم
این پارامتر عادی ساز
به درد میخوره

عدم موفقیت عامل منطقی در حالت عدم قطعیت
عدم امکان استنتاج عملی که با موفقیت او را به هدفش برساند.
عدم امکان بوجود آوردن قوانینی که کامل باشند بدلاً این:

- تنبی
- نادانی نظری
- نادانی عملی

نظريه تصميم = نظريه احتمال + نظريه سودمندي

- متغير تصادفي، فضای نمونه، حادثه، حادثه اتمی
- اصول احتمال
- احتمال پیشین
- توزیع احتمال
- توزیع احتمال توأم
- احتمال شرطی
- توزیع شرطی
- استنتاج با فهرست کردن
- عادی سازی

احتمال بدون داشتن
هیچ اطلاعات قبلی
ای

توزیع احتمال برای
چندتا متغیر

۱۴۰۱-۰۲

نیمسال اوّل

یه سری دانش دیگه
هم داریم

استفاده از جدول
توزیع توأم کامل تا
استنتاج انجام دهیم

30



م.پالهنج

اصفهان - بوستان شهرستان

مازیار پالهنج

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

31

■ در تهیه این اسلایدها، از اسلایدهای سایت کتاب استفاده شده است.

بسمه تعالی

هوش مصنوعی
عدم قطعیت - ۲

نیمسال اول ۱۴۰۱-۱۴۰۲

دکتر مازیار پالهنگ
آزمایشگاه هوش مصنوعی
دانشکده مهندسی برق و کامپیوتر
دانشگاه صنعتی اصفهان

عامل منطقی باوجود عدم قطعیت نمیتوانه به خوبی توانی محیط استنتاج کنند و نمیتوانه دنباله ای اعمالی را پیدا کنند که بتوانند اونو به هدفش با قطعیت بررسونند

یادآوری

- عدم موقیت عامل منطقی در حالت عدم قطعیت
- عدم امکان استنتاج عملی که با موقیت او را به هدفش برساند.
- عدم امکان بوجود آوردن قوانینی که کامل باشند بدلایل:
 - تبلی، نادانی نظری، نادانی عملی
 - نظریه تصمیم = نظریه احتمال + نظریه سودمندی
 - متغیر تصادفی و دامنه آن
 - فضای نمونه، حادثه اتمی، حادثه
 - اصول احتمال
 - احتمال پیشین
 - توزیع احتمال
 - توزیع احتمال توأم
 - احتمال شرطی
 - توزیع شرطی
- استنتاج با فهرست کردن از روی توزیع احتمال توأم کلی
- عادی سازی هنگام یافتن یک توزیع شرطی

استقلال

A و B مستقل هستند اگر و تنها اگر:

$$P(A, B) = P(A)P(B) \text{ یا } P(B | A) = P(B) \text{ یا } P(A | B) = P(A)$$

مثال: به متغیرهای دندان درد، کشیدن و کرم خوردگی، متغیر وضع هوای را نیز اضافه کنیم

توزيع توأم کلی:

$$P(\text{Toothache}, \text{Catch}, \text{Cavity}, \text{Weather})$$

جدول با ۳۲ ورودی

متغیر وضعیت هوای
۴
مقدار را بخودش
میگرفت

مازیار پالهنجک

$$2^*2^*2^*4 = 32$$

هوش مصنوعی - نیمسال اول ۰۱-۱۴۰۱

3

دندان درد و کشیدگی
و کرمخوردگی به
ابری بودن ربطی
ندارن پس میشه
مستقل گرفت وضعیت
هوا را از اون ۳ تا

پیدا کردن احتمال این
په کمک قانون ضرب

$$P(\text{toothache}, \text{catch}, \text{cavity}, \text{cloudy}) =$$

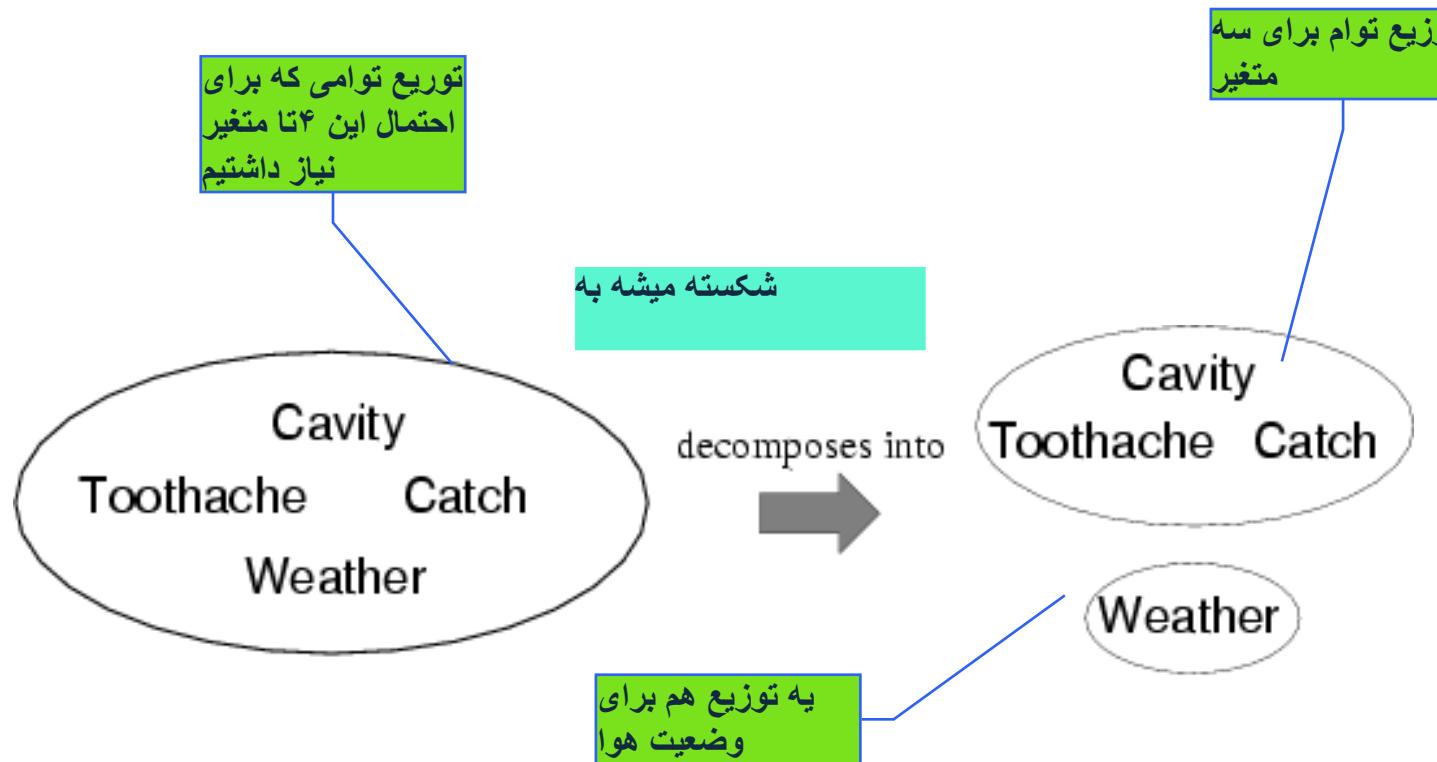
$$P(\text{cloudy} | \text{toothache}, \text{catch}, \text{cavity}) \times$$

$$P(\text{toothache}, \text{catch}, \text{cavity})$$

$$P(\text{toothache}, \text{catch}, \text{cavity}, \text{cloudy}) =$$

$$P(\text{cloudy}) \times$$

$$P(\text{toothache}, \text{catch}, \text{cavity})$$



$$8 + 4 = 12$$

- ۱۲ ورودی به جای ۳۲ ورودی
- استقلال مطلق به سختی وجود دارد.

مازیار پالهنجک

اگه بتونیم بین متغیرها استقلال پیدا کنیم حل
مسئله خیلی ساده تر میشه

استقلال شرطی
کج از دندان در
مستقل میشیم

استقلال شرطی

- توزيع $P(\text{Toothache}, \text{Cavity}, \text{Catch})$ را در نظر بگیرید.
- در صورت وجود کرم خوردگی، احتمال کشیده شدن دندان مستقل از داشتن دندان درد است.

$$P(\text{catch}|\text{toothache}, \text{cavity}) = P(\text{catch}|\text{cavity})$$

به همین صورت:

$$P(\text{catch}|\text{toothache}, \neg\text{cavity}) = P(\text{catch}|\neg\text{cavity})$$

یا:

$$P(\text{Catch}|\text{Toothache}, \text{Cavity}) = P(\text{Catch}|\text{Cavity})$$

ماریار پالهنج

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

6

به صورت توزیعی

$$p(a,b|c) = p(a|c) * p(b|c)$$

$P(a,b) = p(a) * p(b)$
رابطه ای که برای استقلال شرطی داشتیم

استقلال شرطی

- در صورت وجود کرم خوردگی، احتمال کشیده شدن دندان مستقل از داشتن دندان درد است.

$$P(\text{toothache, catch} | \text{cavity}) = P(\text{toothache} | \text{cavity})P(\text{catch} | \text{cavity})$$

$$P(\text{cavity} | \text{toothache, catch}) = \alpha P(\text{toothache} | \text{cavity})P(\text{catch} | \text{cavity})P(\text{cavity})$$

$$p(a|b) = (p(b|a)*p(a))/ p(b)$$

استقلال شرطی

▪ بطور کلی

$$\mathbf{P}(X, Y | Z) = \mathbf{P}(X | Z)\mathbf{P}(Y | Z)$$

ایکس و واي مستقل
از هم هستند با وجود
زد

قانون بیز

$P(a,b) = P(a | b)P(b) = P(b | a)P(a)$: قانون ضرب ■

$$P(b | a) = \frac{P(a | b)P(b)}{P(a)}$$

قانون بیز: ■

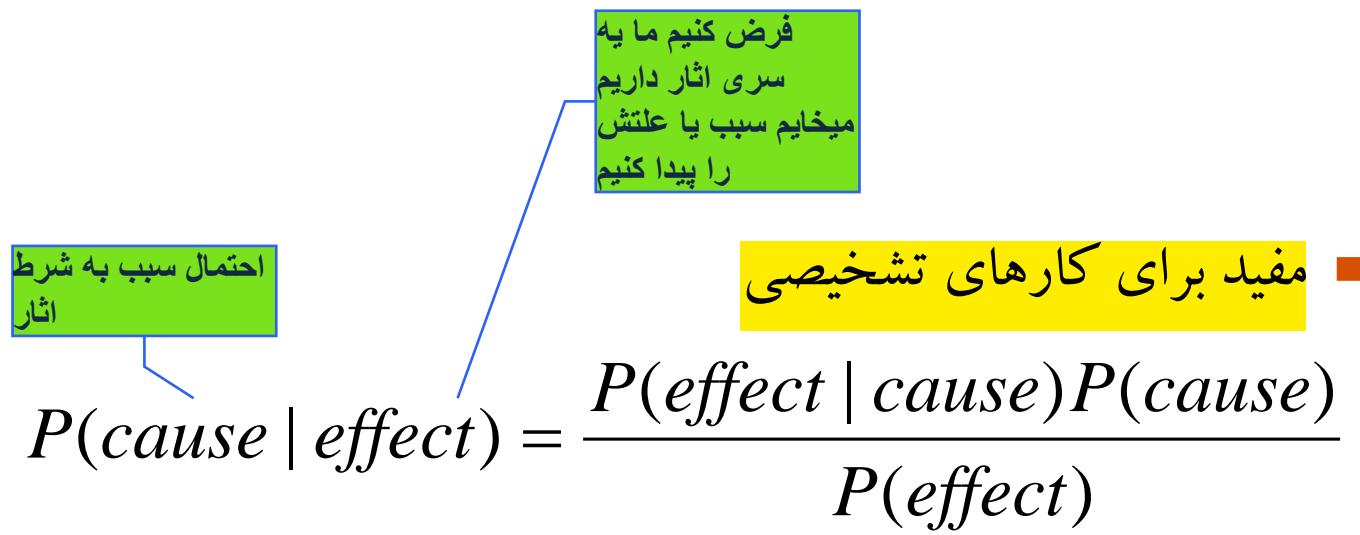
بصورت توزیعی: ■

$$P(Y | X) = \frac{P(X | Y)P(Y)}{P(X)}$$

قانون بیز

■ با داشتن دلیل e

$$\mathbf{P}(Y | X, \mathbf{e}) = \frac{\mathbf{P}(X | Y, \mathbf{e})\mathbf{P}(Y | \mathbf{e})}{\mathbf{P}(X | \mathbf{e})}$$



■ بطور مثال اگر M بیماری منژیت و S گردن درد باشد:

$$P(m|s) = \frac{P(s|m)P(m)}{P(s)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$$

مازیار پالهنجک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲
احتمال منژیت داشتن
به شرط گردن درد
داشتن چقدر

11

قانون بیز و استقلال شرطی

$$P(Cavity | toothache \wedge catch)$$

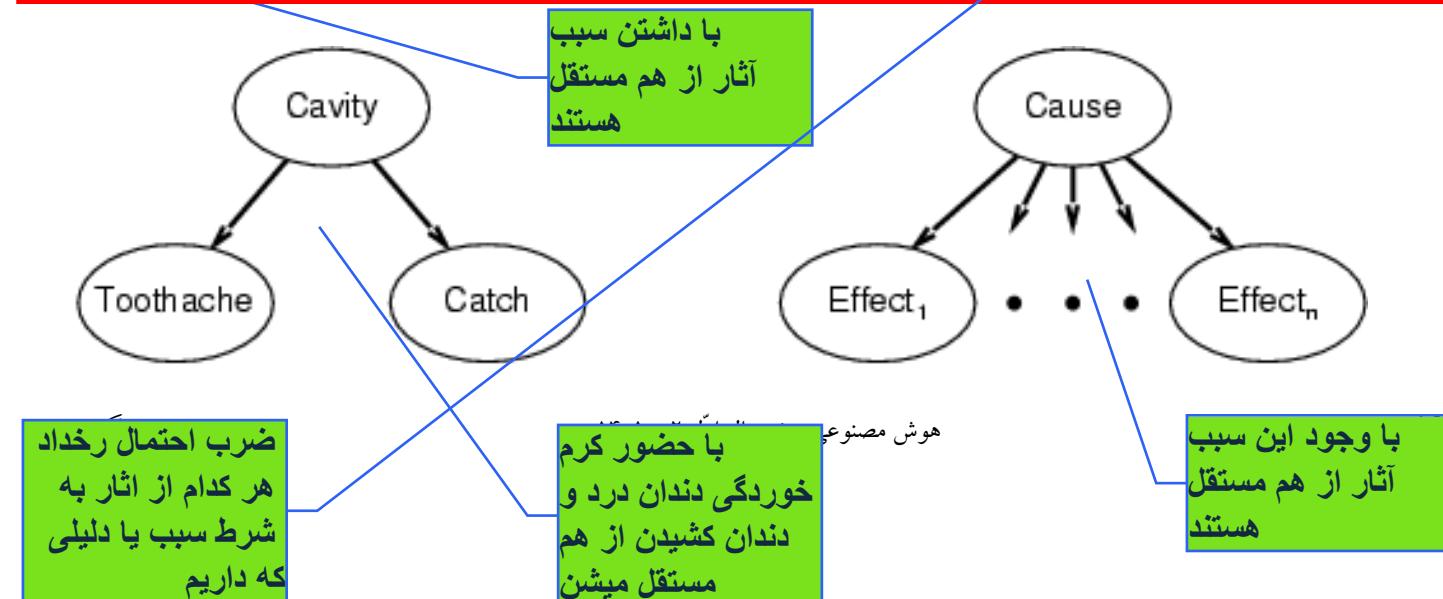
$$= \alpha P(toothache \wedge catch | Cavity) P(Cavity)$$

$$= \alpha P(toothache | Cavity) P(catch | Cavity) P(Cavity)$$

داشتهیم:

نمونه ای از مدل بیز ساده

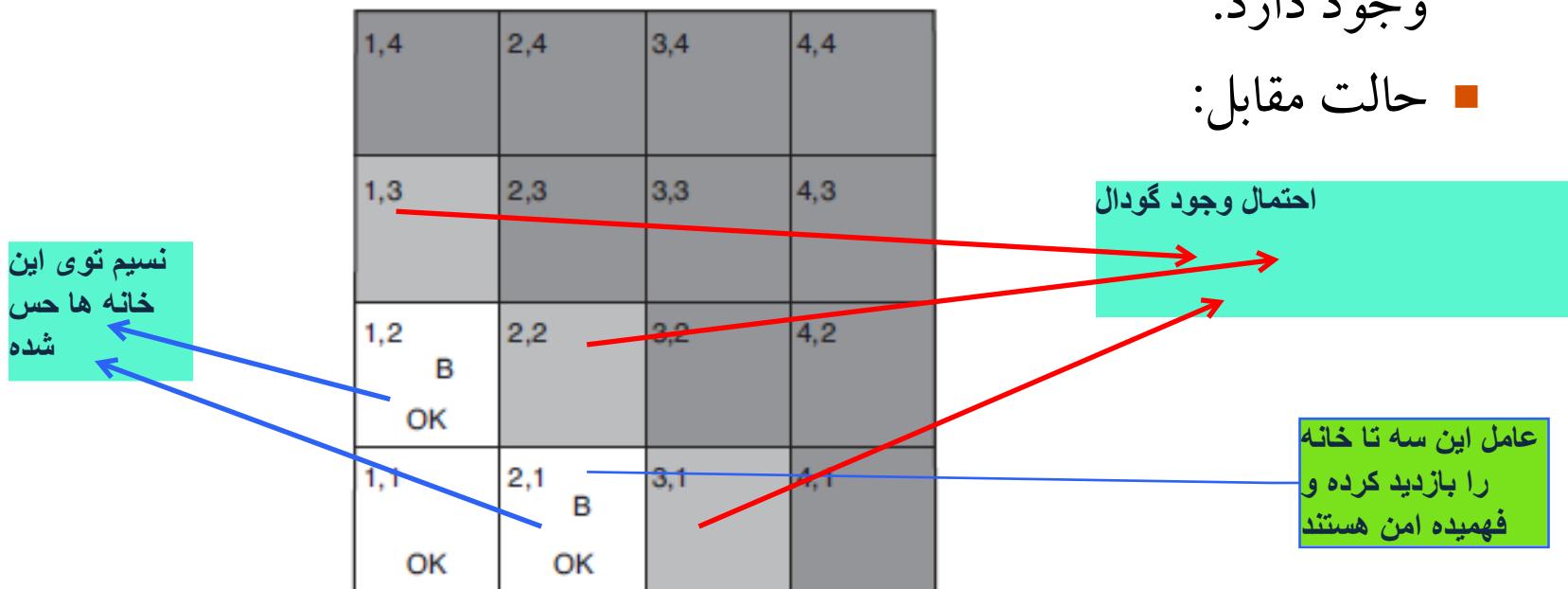
$$P(Cause, Effect_1, \dots, Effect_n) = P(Cause) \prod_i P(Effect_i | Cause)$$



دنیای دیو – بازدید دوباره

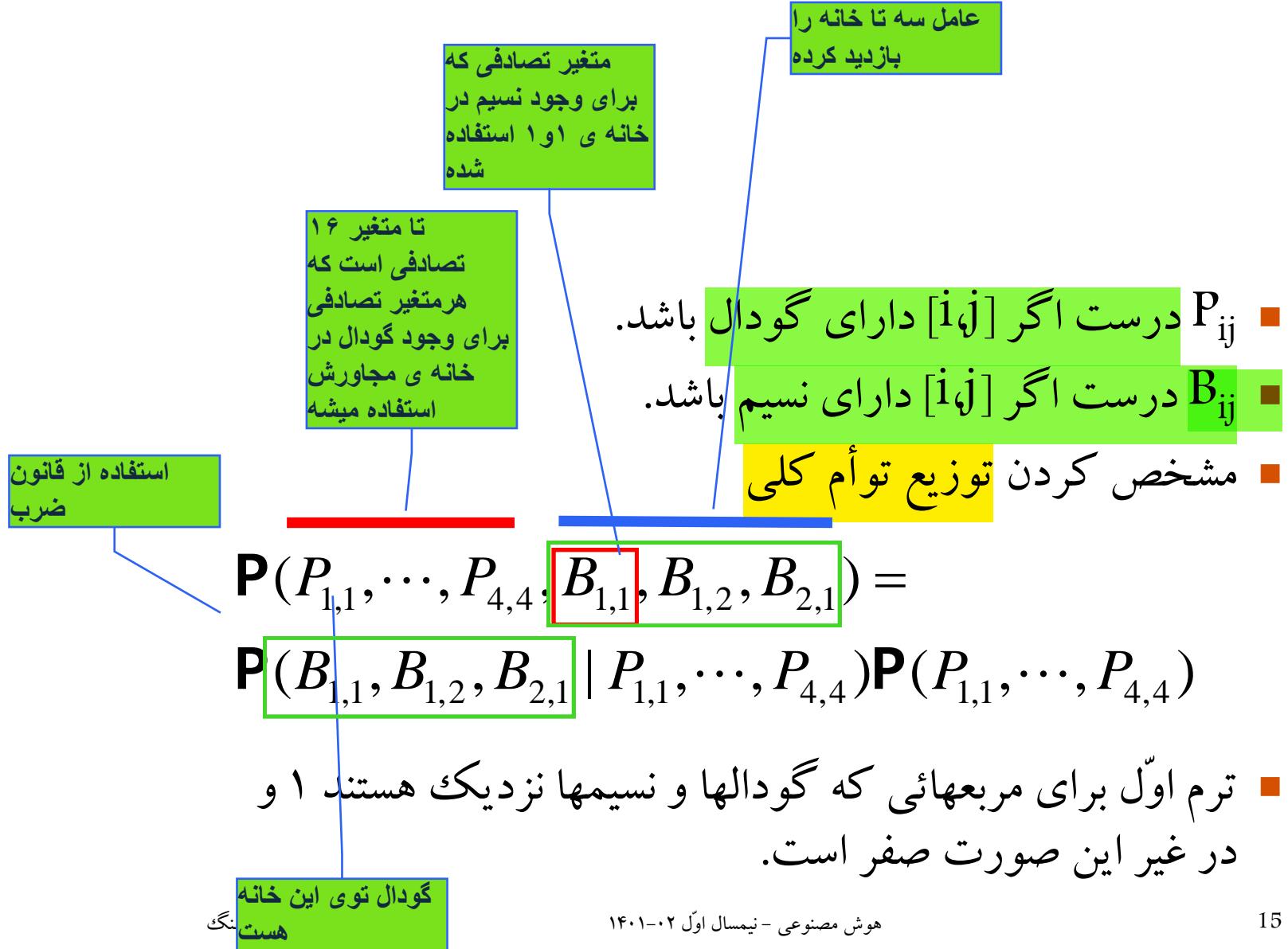
- دنیای دیو نیمه مشاهده پذیر است بدین دلیل عدم قطعیت در آن وجود دارد.

■ حالت مقابل:



عامل باید احتمال وجود گودال توى اوں سه تا خونه را
بدست بیاره بعد تصمیم بگیره پرہ یا نہ

- عامل منطقی خالص نمی تواند نتیجه بگیرد کدام مربع محتملتر است که امن باشد.
- می تواند بصورت تصادفی انتخاب کند.
- هدف: محاسبه احتمال آنکه هر یک از سه خانه دارای گودال باشد.
- فعلاً دیو و طلا را در نظر نمی گیریم.
- فرض: هر مربع به جز [۱۰۱] با احتمال ۰،۲ دارای گودال است.



وجود گودال در یک خانه مستقل از خانه‌ی دیگری است

ضرب احتمال روی همه‌ی خانه‌ها

داریم:

$$P(P_{1,1}, \dots, P_{4,4}) = \prod_{i,j=1,1}^{4,4} P(P_{i,j})$$

میدانیم تو خانه‌ی یک و یک نسیم نیست

تو خانه‌ی او تو خانه‌ی نسیم هست

دلالت فعلی: مشاهده یا عدم مشاهده نسیم در مربعهای بازدید

شده. ($b = \neg b_{1,1} \wedge b_{1,2} \wedge b_{2,1}$)

و اینکه این مربعها دارای گودال نیستند.

(known = $\neg p_{1,1} \wedge \neg p_{1,2} \wedge \neg p_{2,1}$)

علاقمند با پاسخ به سوالی همانند

توزیعش چقدر؟

$P(P_{1,3} | \text{known}, b)$

مازیار پالهنجک

اطلاعاتی که داریم کجاها گودال و نسیم هست یا نیست

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

با این احتمالات و اطلاعاتی که از مساله داریم احتمال وجود گودال در خانه‌ی ۳ و ۱ چقدر؟

جمع میزندیم روی
همه ی متغیرهای
ناشناخته یا نهان

گودال های شناخته
شده که سه تا بود

■ فرض **unknown** ترکیب متغیرهای گودال به جز مربعهای
شناخته شده و سؤال

$$\mathbf{P}(P_{1,3} | known, b) = \alpha \sum_{unknown} \mathbf{P}(P_{1,3}, unknown, known, b)$$

■ با داشتن توزیع توأم کلی می توان محاسبه فوق را انجام داد.

■ ۱۲ مربع ناشناس

■ جمع دارای $2^{12} = 4096$ ترم است.

خانه ی سوال که
میخایم ببینیم تو ش
گودال هست یا ن

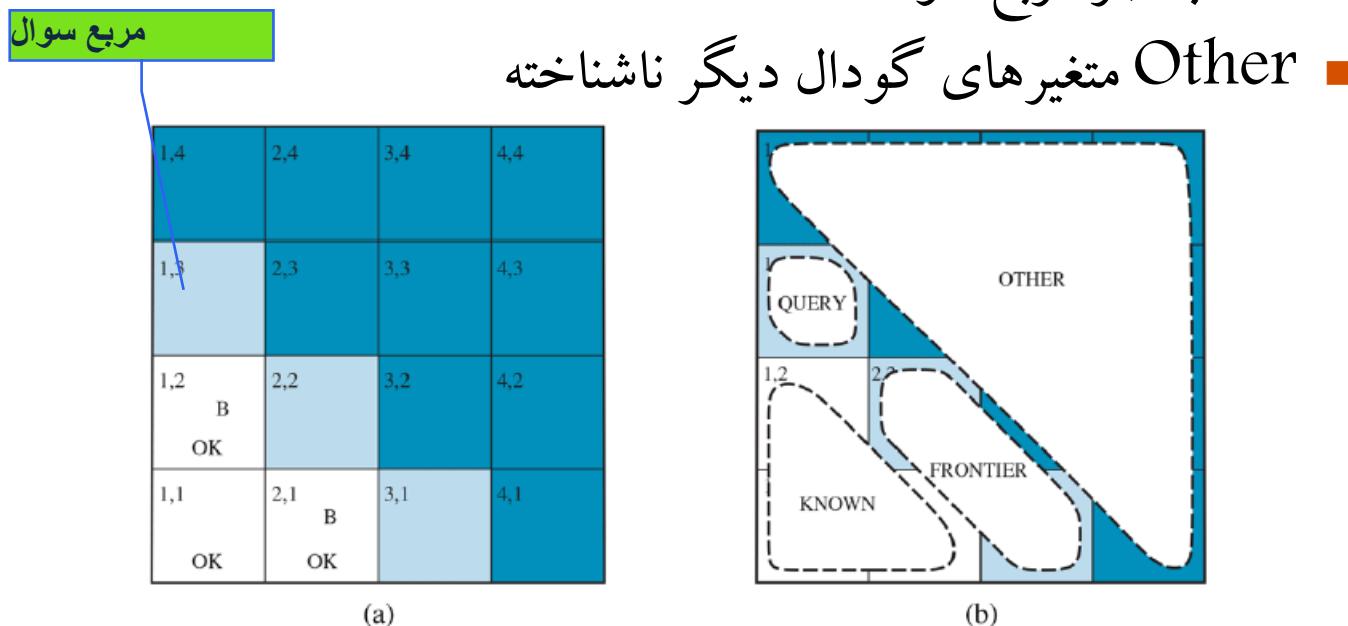
مازیار پالهنجک

- نیمسال اول ۱۴۰۱-۰۲

توى اين ۱۲ تا مربع
ميتونه گودال باشه یا
نباشه
پس ۴۰۹۶ تا حالت
داريم برای محاسبه
اين جمع

17

- آیا محتوای [4,4] بر روی گودال بودن [1,3] اثر دارد؟
- فرض frontier متغیرهای گودال نزدیک به مربعهای باز دید شده به جز مربع سؤال



(a) After finding a breeze in both [1,2] and [2,1], the agent is stuck—there is no safe place to explore. (b) Division of the squares into *Known*, *Frontier*, and *Other*, for a query about [1,3].

■ نسیمهای مشاهده شده بطور شرطی از دیگر متغیرها مستقل شرطی هستند به شرط داشتن متغیرهای frontier, known، و query

این قسمت فراتر
داره یه قسمت
other

$$P(X | e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$$

$$P(P_{1,3} | known, b)$$

$$\begin{aligned} &= \alpha \sum_{unknown} P(P_{1,3}, known, b, unknown) && \text{(from Equation (12.23))} \\ &= \alpha \sum_{unknown} P(b | P_{1,3}, known, unknown) P(P_{1,3}, known, unknown) && \text{(product rule)} \\ &= \alpha \sum_{frontier} \sum_{other} P(b | known, P_{1,3}, frontier, other) P(P_{1,3}, known, frontier, other) \\ &= \alpha \sum_{frontier} \sum_{other} P(b | known, P_{1,3}, frontier) P(P_{1,3}, known, frontier, other), \end{aligned}$$

مالهنج

احتمال وجود نسیم یعنی بی
به شرط داشتن
known,p13,frontier
از
other
مستقل است پس حذف میشه

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

$$\mathbf{P}(P_{1,1}, \dots, P_{4,4}) = \prod_{i,j=1,1}^{4,4} \mathbf{P}(P_{i,j})$$

$$\mathbf{P}(P_{1,3} | known, b)$$

$$= \alpha \sum_{frontier} \sum_{other} \mathbf{P}(b | P_{1,3}, known, frontier) \mathbf{P}(P_{1,3}, known, frontier, other)$$

$$= \alpha \sum_{frontier} \mathbf{P}(b | P_{1,3}, known, frontier) \sum_{other} \mathbf{P}(P_{1,3}, known, frontier, other)$$

$$= \alpha \sum_{frontier} \mathbf{P}(b | P_{1,3}, known, frontier) \sum_{other} \mathbf{P}(P_{1,3}) \mathbf{P}(known) \mathbf{P}(frontier) \mathbf{P}(other)$$

$$= \alpha \mathbf{P}(known) \mathbf{P}(P_{1,3}) \sum_{frontier} \mathbf{P}(b | P_{1,3}, known, frontier) \mathbf{P}(frontier) \sum_{other} \mathbf{P}(other)$$

$$= \alpha' \mathbf{P}(P_{1,3}) \sum_{frontier} \mathbf{P}(b | P_{1,3}, known, frontier) \mathbf{P}(frontier)$$

گودال ها از هم
مستقل هستند

مازیار پالهنجک

جمعش روی همه ی
وضعیت هایی است
که
other
به خودش میگیره
پس یک میشه

هوش مصنوعی - نی

= 1

21

$P(b | \text{known}, P_{1,3}, \text{frontier}) = 1$ است وقتی که
با مشاهدات نسیم سازگار باشد.

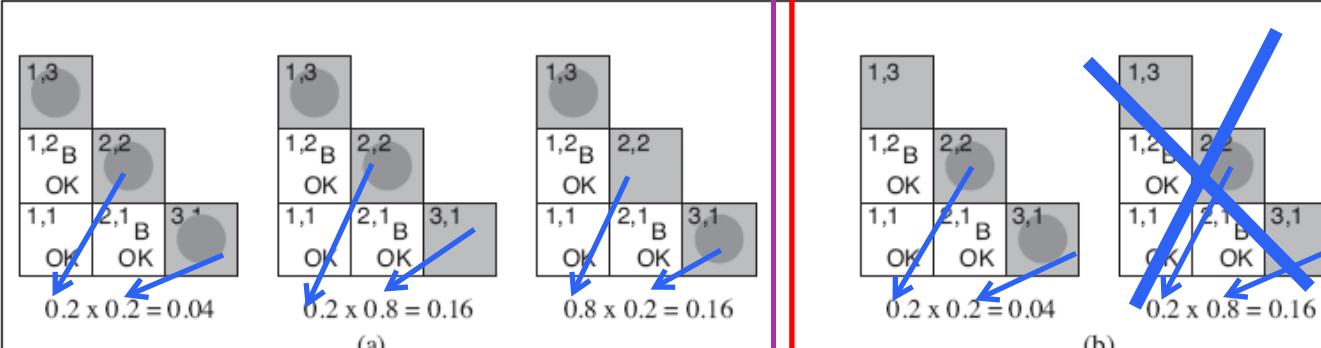


Figure 13.6 Consistent models for the frontier variables $P_{2,2}$ and $P_{3,1}$, showing $P(\text{frontier})$ for each model: (a) three models with $P_{1,3} = \text{true}$ showing two or three pits, and (b) two models with $P_{1,3} = \text{false}$ showing one or two pits.



$$P(P_{1,3} | \text{known}, b) = \alpha' \langle 0.2(0.04 + 0.16 + 0.16), 0.8(0.04 + 0.16) \rangle \\ \approx \langle 0.31, 0.69 \rangle$$

■ با احتمال ۰،۳۱ در [۱،۳] گودال وجود دارد.

■ به همینطور در [۱،۳]

■ بطور مشابه در [۲،۲] به احتمال ۰،۸۶

■ عامل حتماً باید از رفتن به [۲،۲] اجتناب کند.

■ برای عامل منطقی قبل سه خانه شرایط یکسانی داشتند.

در این خانه گودال است

مازیار پالهنج

عامل باید سعی کنه از خانه ای که احتمال وجود گودال توش بیشتر است اجتناب کنه

خلاصه

- استقلال متغیرها
- استقلال شرطی متغیرها
- بازدید دوباره دنیای دیو



م.پالهنج

اصفهان - بوستان شهرستان

مازیار پالهنج

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

25

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوء درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه این اسلایدها، از اسلایدهای سایت کتاب استفاده شده است.