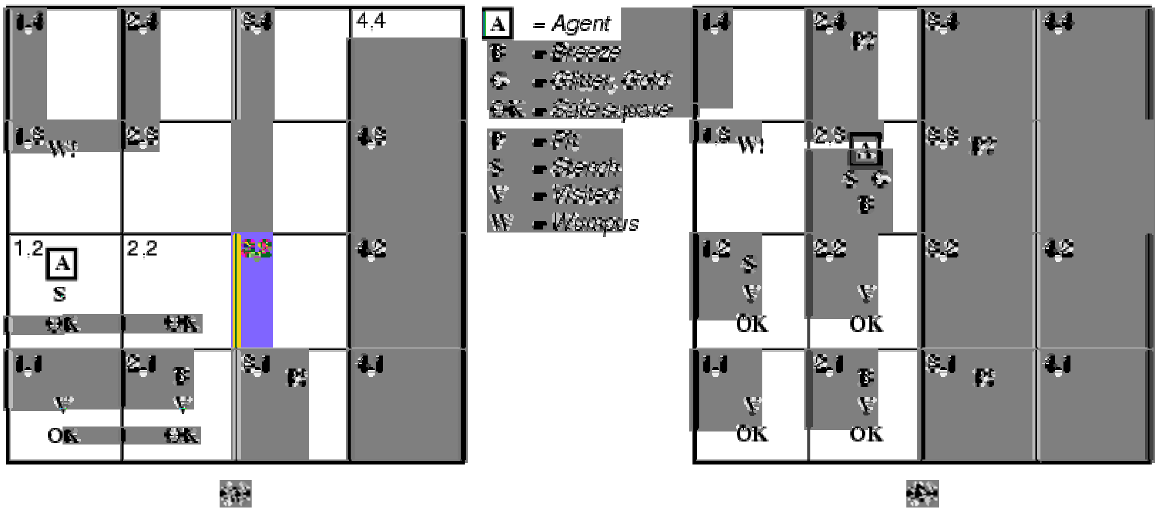
در خانه ی [۲و۱] ، از بوی تعفن خانه می فهمیم کـه Wumpus در خانـه ی [۳و۱] یـا [۲و۲] اسـت . Wumpus در خانه ی [۱و۱] یا [۲و۲] یا [۱و۲] هم نیست . در نتیجه Wumpus در خانه ی [۳و۱] اسـت . بنابراین ، خانه ی [۲و۲] ، امن است چون بوی خـوش در خانـه ی [۲و۱] نیسـت ؛ بنـابراین ، چالـه در خانـه ی [۱و۳] می باشد . بنابراین ، به خانه ی [۲و۲] می رویم .



از خانه ی [۲و۲] به خانه ی [۳و۲] می رویم . در خانه ی [۳و۲] درخشش طلا ، بوی بد و بوی خوش را داریم ، بنابراین طلا را بر می داریم و از بوی خوش موجود در این خانه نتیجه می گیریم که چاله در خانه ی [۳و۳] یا [۴و۲] می باشد . سپس از مسیر امنی که قبلا آمده ایم برمی گردیم و در نهایت ، بازی خاتمه می یابد .

منطق چیست ؟

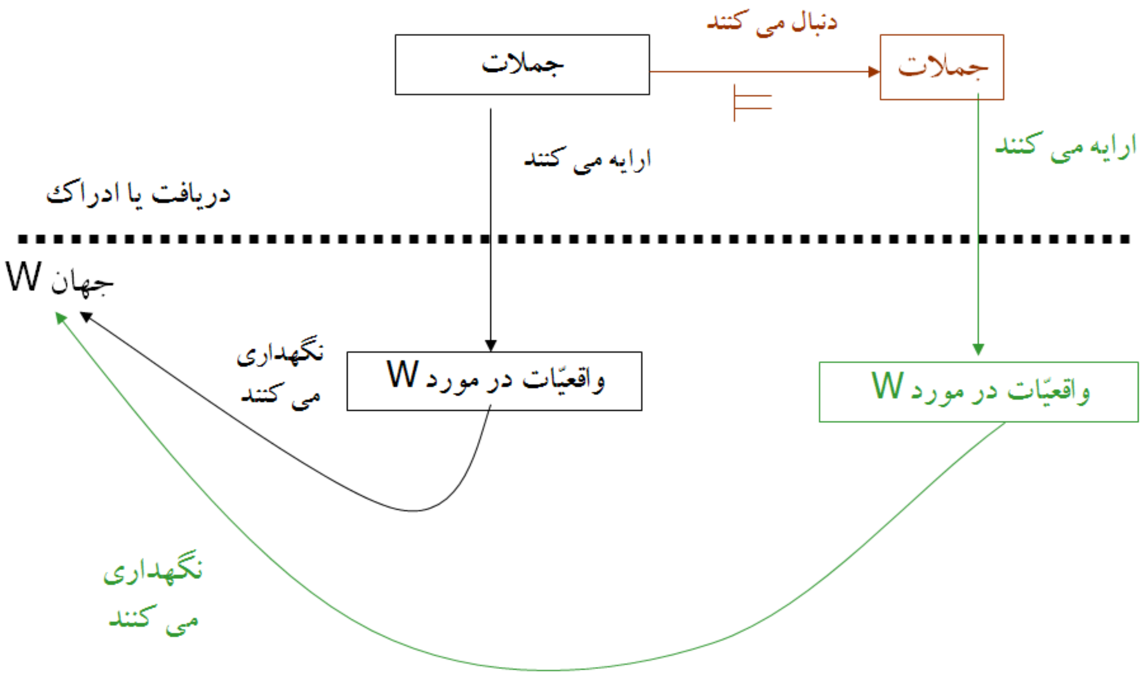
منطق ، روشی کلاسیک برای ارایه ی دانش عامل می باشد و به ما اجازه می دهد که عامل ها را به صورت اعلانی (اظهاری) ، برنامه ریزی و توصیف کنیم . به بیان دیگر ، منطق ، یک زبان رسمی است که

دارای ظاهر [1] و سمانتیک [2] می باشد ، ظاهر (نحو یا گرامر) ، می گوید که چه عبارت هایی مجازند مثلا در ریاضیّات ، عبارت $x+2=5$ به صورت گرامری درست می باشد ، امّا $x+=3$ به صورت گرامری درست نمی باشد . به عنوان مثال دیگر ، در زبان ریاضی ، $x+2\geq y$ یک جمله است ولی $x2+y>$ یک جمله نمی باشد . و معنا یا سمانتیک ، مشخص می نماید که یک عبارت در چه موقعی درست یا در چه موقعی ، غلط می باشد ؛ معنای $x+2=5$ در زمانی درست است که $x=3$ باشد و در غیر این صورت غلط می باشد . به عنوان مثال دیگر ، $x+2\geq y$ درست است در صورتی که عدد $x+2$ کم تر از y نباشد . $x+2\geq y$ در جهان در صورتی که $x=7,y=1$ باشد صحیح می باشد . $x+2\geq y$ در جهان در صورتی که $x=0,y=6$ غلط است . *توجه کنید که ، عبارت های منطقی ، باید یا درست باشند و یا غلط باشند ؛ به بیان دیگر ، "درجه ی درستی ، وجود ندارد" .*

ارتباطات موجود در جهان

[1] Syntax

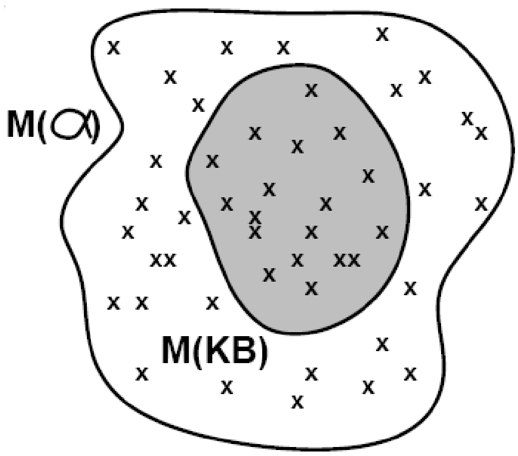
[2] Semantics



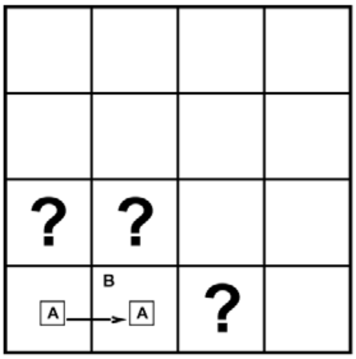
دنبال کردن [۱] – معنای دنبال کردن این است که چیزی توسط دیگری دنبال شود : $KB \models \alpha$. **پایگاه دانش KB جمله ی α را دنبال می کند اگر و فقط اگر α در همه ی جهان ها در جایی که KB درست است ، درست باشد .** به عنوان مثال ، $x+y=4$ ، $4=x+y$ را دنبال می نماید . دنبال کردن ، ارتباطی میان جملات است که براساس **سمانتیک ها یا معناها** می باشند . به عنوان مثال دیگر ، اگر پایگاه دانش ، دارای " پیراهن ، سبز رنگ می باشد " و "پیراهن ، خط دار (راه راه) می باشد " باشد ، در این صورت " پیراهن ، سبز رنگ است یا راه راه است را دنبال می نماید " . به صورت تکنیکی (مطابق اصول فنی) ، دنبال کردن ، می گوید که : برای همه ی مدل هایی که a درست است ، b هم درست می باشد ؛ مثلا ، $a=3 \models a+2=5$. دنبال کردن به ما اجازه خواهد داد که استدلال نماییم و واقعیّات جدیدی را به پایگاه دانش عامل مان اضافه نماییم .

[۱] Entailment : استلزام یا ایجاب کردن

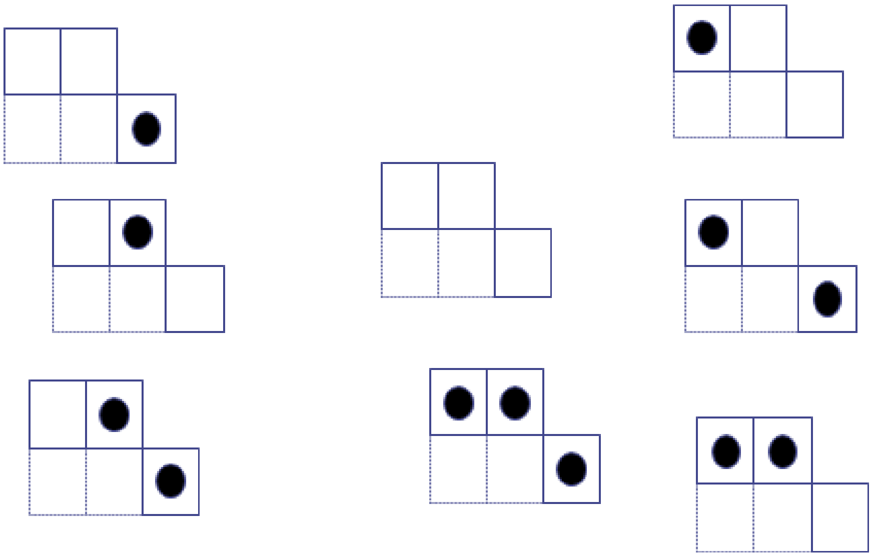
مدل ها – مدل ها ، تعریف های رسمی وضعیّت های ممکن جهان هستند . گوییم **m** **مدلی از** یک جمله ی **α** است در صورتی که **α** در **m** درست باشد . اگر $M(\alpha)$ مجموعه ای از تمام مدل های α باشد در این صورت $KB \models \alpha$ اگر و فقط اگر $M(KB)$ زیر مجموعه یا مساوی $M(\alpha)$ باشد .



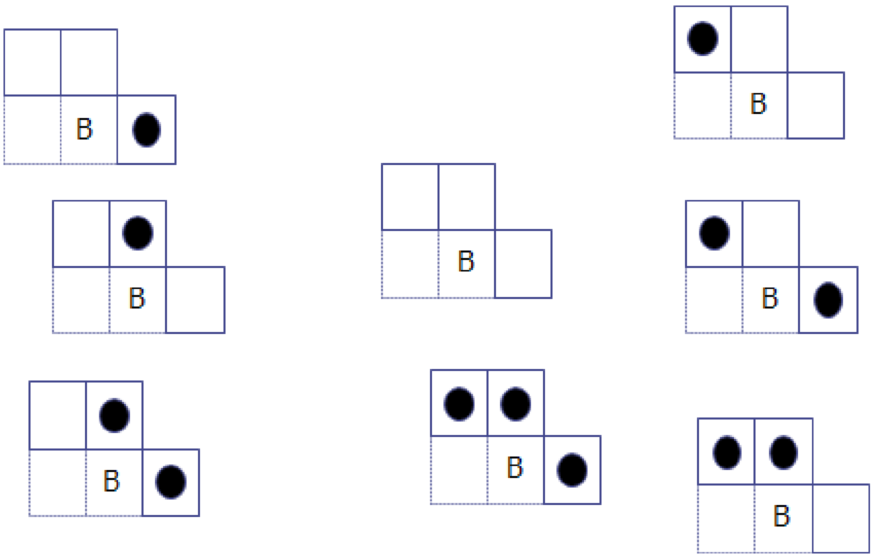
دنبال کردن در دنیای wumpus – بعد از اینکه در خانه ی [۱و۱] هیچ چیزی را پیدا نکردیم حرکت بعدی ما ، حرکت به راست در خانه ی [۱و۲] خوش هواست . مدل های ممکن برای ؟ برای احتمال وجود چاله یا عدم وجود چاله چیستند ؟ . سه انتخاب بولین وجود دارد و در نتیجه ۸ مدل ممکن است .



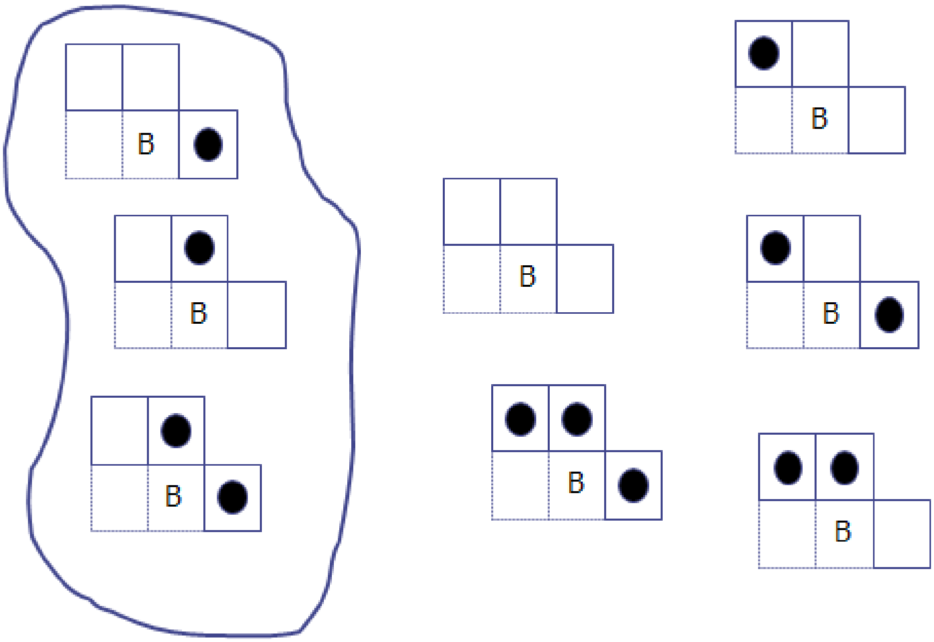
مدل های Wumpus



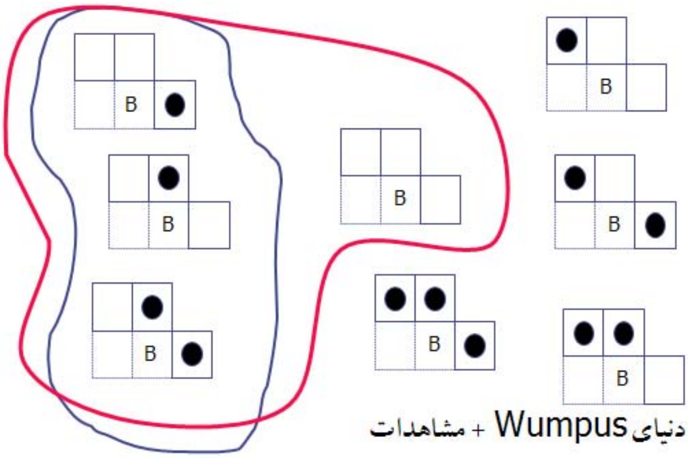
مدل های Wumpus



مدل های Wumpus



مدل های Wumpus

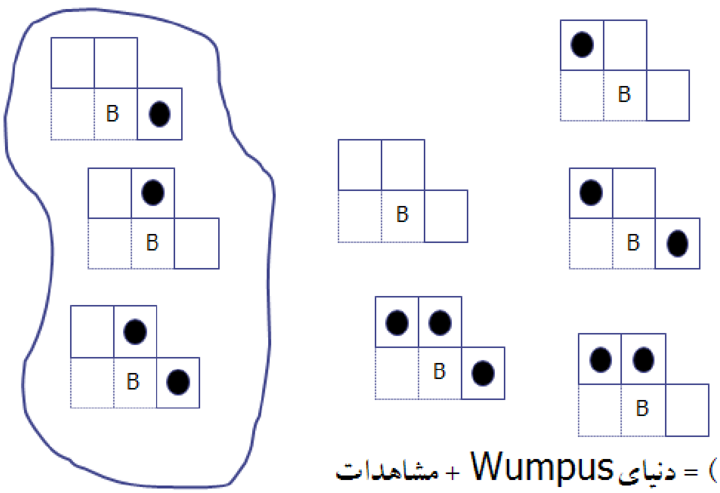


پایگاه دانش (KB) = دنیای Wumpus + مشاهدات

α_1 = "ایمن است [۲و۱]"

KB |= α_1

مدل های Wumpus

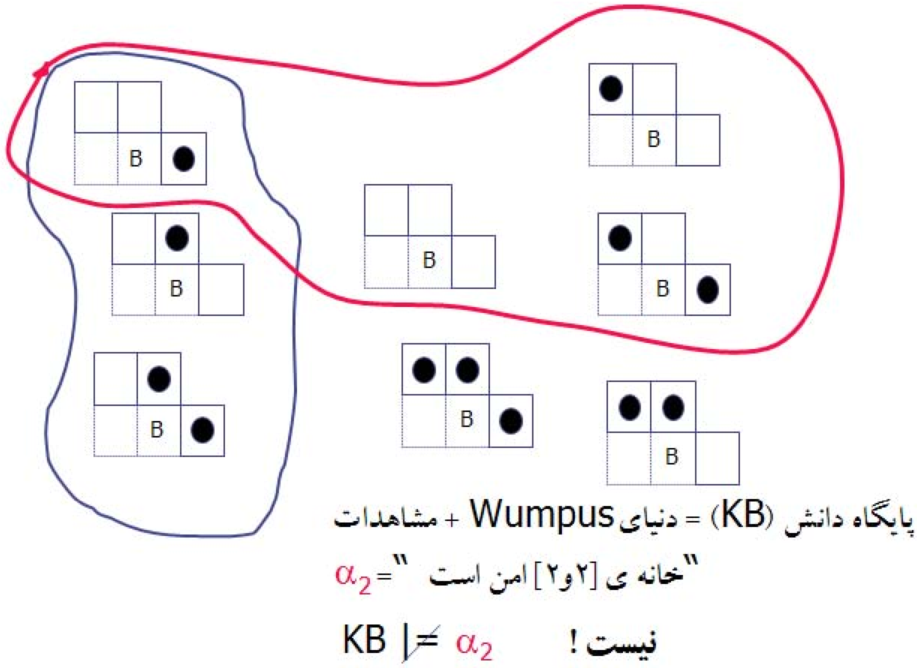


پایگاه دانش (KB) = دنیای Wumpus + مشاهدات

α_2 = "خانه ی [۲و۲] امن است"

KB |= α_2 ??

مدل های Wumpus



استنتاج منطقی

نظریّه ی دنبال کردن ، که قبلا آن را بررسی کردیم ، می توانـد بـرای اسـتنتاج منطقـی مـورد اسـتفاده قرار گیرد . بررسی مدل [۱] (مثال Wumpus را ببینید) همه ی مدل های ممکن را بررسـی مـی کنـد و چـک می کند که آیا α درست است .

در صورتی که یک الگوریتم فقط جمله های دنبـال شـده را بـه وجـود مـی آورد ، ***نگهدارنـده ی درستی*** [۲] **یا کامل** [۱] نام دارد ، در غیر این صورت فقط چیزها را به وجـود مـی آورد . *i در صـورتی کامـل است که در جایی که $KB \vdash_i \alpha$ ، این هم درست باشد که $KB \models \alpha$.*

---

[۱] model checking
[۲] truth preserving

کمال یا تمامیّت [۲] : الگوریتم می تواند هر عبارتی که دنبال می شود را به وجود آورد . i در صورتی دارای تمامیّت یا کمال است که در هر جایی که $KB \models \alpha$ ، این هم درست باشد که $KB \vdash_i \alpha$.

استنتاج [۳]

$KB \vdash_i \alpha$ ؛ یعنی ، جمله ی α می تواند از KB توسط روال [۴] i مشتق شود (به وجود آید) .

کامل بودن [۵] : *i در صورتی کامل است که در جایی که $KB \vdash_i \alpha$ ، این هم درست باشد که $KB \models \alpha$.*

کمال یا تمامیّت [۶] : *i در صورتی دارای تمامیّت یا کمال است که در هر جایی که $KB \models \alpha$ ، این هم درست باشد که $KB \vdash_i \alpha$.*

ما می خواهیم منطقی را که به اندازه ی کافی رسا می باشد برای گفتن تقریباً هر چیزی که می خواهیم و برای این که بفهمیم کدام یک صحیح است و نتیجه ی کدام زیر برنامه کامل می باشد ، تعریف نماییم .

مثال هایی از منطق

منطق گزاره ای : $A \wedge B \Rightarrow C$

---

[۱] sound

[۲] completeness

[۳] inference

[۴] procedure

[۵] soundness

[۶] completeness

منطق مرتبه ی اوّل : $(\forall x)(\exists y)$ Mother(y,x)

منطق عقیده ای یا زود باور [1] : B(John,Father(Zeus,Cronus))

منطق گزاره ای

ظاهر – منطق گزاره ای ، ساده ترین منطقی است که ایده های اساسی را توضیح می دهد . نمادهای گزاره ای ، P1,P2 و ..... می باشند . اگر S یک جمله باشد ؛ $\neg S$ ، عبارت نقیض [2] آن می باشد .

برای عبارات S_1 و S_2 ؛

$S_1 \wedge S_2$ ؛ AND (ترکیب فصلی [3] یا جدایی ؛ در صورتی که ورودی (ها) صحیح باشند ، نتیجه درست خواهد بود) این دو خواهد بود .

$S_1 \vee S_2$ ، OR (ترکیب عطفی [4] ؛ در صورتی که از S_1 و S_2 یکی درست باشد ، نتیجه درست خواهد بود) این دو خواهد بود .

$S_1 \Rightarrow S_2$ نتیجه گیری (استلزام یا استنباط [1]) خواهد بود و در صورتی درست است که A درست باشد و B هم درست باشد ؛ مثلا ، اگر الان هوا بارانی باشد ، آن گاه الان هوا ابری می باشد . استلزام به ما امکان استنتاج را می دهد .

---

[1] Logic of Belief
[2] negation
[3] conjunction
[4] disjunction

نکته : $S_1 \Rightarrow S_2$ معادل است با $\neg S_1 \vee S_2$.

$S_1 \Leftrightarrow S_2$ استنباط دو شرطی (اگر و فقط اگر [۲]، S_1 اگر و فقط اگر S_2) خواهد بود . به عنوان مثال ، $(P \wedge Q) \Rightarrow R$ و $(A \Rightarrow B) \vee (\neg C)$.

ترتیب اولویّت ها [۳]

به ترتیب از چپ به راست عبارتند از $\Rightarrow$ $\vee$ $\wedge$ $\neg$ ؛ به عنوان مثال ، $\neg A \vee B \Rightarrow C$ برابر است با $((\neg A)\vee B)\Rightarrow C)$.

مدل های موجود در منطق گزاره ای

مدل ، نسبت دادن درست یا غلط به هر عبارت اتمیک (تجزیّه ناپذیر) است ؛ اگر A ، B ، C و D سمبل های گزاره ای باشند ؛ $m = \{A=true, B=false, C=false, D=true\}$ ، یک مدل است ؛ همچنین ، $m' = \{A=true, B=false, C=false\}$ هم یک مدل است . چند مدل می توانند برای n سمبل گزاره ای تعریف شوند ؟ ۲^n

سمانتیک ها یا معناهای منطق گزاره ای ، هر مدل درست [۴] / غلط [۵] را برای هر نشان [۶] مشخص می نماید .

---

[۱] implication
[۲] biconditional
[۳] Order of Precedence
[۴] true
[۵] false
[۶] symbol

" چاله ها باعث ایجاد هوای خوش در خانه های اطراف می شوند "

$P_{i,j}$ در صورتی که یک چاله در [i,j] وجود دارد ، درست می باشد .

$B_{i,j}$ در صورتی که یک هوای خوش (breeze) در [i,j] وجود داشته باشد ، درست می باشد .

$$\neg P_{1,1}$$
$$\neg B_{1,1}$$
$$B_{2,1}$$

" چاله ها باعث ایجاد هوای خوش در خانه های اطراف می شوند "

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

" یک خانه خوش هوا است *اگر و فقط اگر* یک چاله در اطرافش وجود داشته باشد "

جداول صحّت برای استنتاج

KB	R_5	R_4	R_3	R_2	R_1	$P_{3,1}$	$P_{2,2}$	$P_{2,1}$	$P_{1,2}$	$P_{1,1}$	$B_{2,1}$	$B_{1,1}$
غلط	غلط	درست	درست	درست	درست	غلط	غلط	غلط	غلط	غلط	غلط	غلط
غلط	غلط	درست	غلط	درست	درست	درست	غلط	غلط	غلط	غلط	غلط	غلط
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
غلط	درست	درست	غلط	درست	درست	غلط	غلط	غلط	غلط	غلط	درست	غلط
درست	درست	درست	درست	درست	درست	درست	غلط	غلط	غلط	غلط	درست	غلط
درست	درست	درست	درست	درست	درست	غلط	درست	غلط	غلط	غلط	درست	غلط
درست	درست	درست	درست	درست	درست	درست	درست	غلط	غلط	غلط	درست	غلط
غلط	درست	درست	غلط	غلط	درست	غلط	غلط	درست	غلط	غلط	درست	غلط
.	.	.	.	.	.	.	.	.	.	.	.	.

. . درست	. . درست	. . درست	. . درست	. . درست	. . درست	. . درست	. . غلط	. . درست	. . درست	. . غلط	. . درست	. . غلط

در شمارش سطرها (با انتساب های مختلف به سطرها) ، در صورتی که KB در سطر درست است ، از α هم صرف نظر نمایید .

استنتاج با شمارش

شمارش Depth-first برای همه ی مدل ها صحیح و کامل است .

تابع TT-Entails?(KB,α) درست یا غلط را برمی گرداند

ورودی ها : KB ، که یک پایگاه دانش می باشد و یک جمله در منطق گزاره ای می باشد و α که یک صف می باشد و به صورت یک جمله ی منطق گزاره ای می باشد

یک لیست از سمبل های گزاره ای در KB و α → symbols

TT-Check-All(KB,α,symbols,[ ]) را برگردان

تابع TT-Check-All(KB,α,symbols,model)درست یا غلط را برمی گرداند

در صورتی که Empty?(symbols) صحیح می باشد ،

در صورتی که PL-True?(KB,model) دارای ارزش درست است ، PL-True?(α,model) را برگردان .

در غیر این صورت درست (true) را برگردان

در غیر این صورت

$P \leftarrow First(symbols); rest \leftarrow Rest(symbols)$

TT-Check-All(KB,α,rest,Expand(P,true,model)) و TT-Check-All(KB,α,rest,Expand(P,false,model)) را برگردان

$O(2^n)$ برای n سمبل می باشد ؛ مسأله به صورت co-NP-complete می باشد .

تساوی های منطقی

دو عبارت در صورتی معادل منطقی می باشند که در مدل های یکسان برابر باشند :

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

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$: جابه جایی پذیری[1] $\wedge$

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

$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$: شرکت پذیری[2] $\wedge$

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

$\neg(\neg\alpha) \equiv \alpha$: حذف نقیض دوگانه[3]

$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$: مفهوم مخالف[1]

---

[1] commutativity

[2] associativity

[3] double-negation elimination

$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$: حذف استنتاج[۲]

$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$: حذف شرط دو طرفه[۳]

$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$: قانون دمورگان[۴]

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

$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$: توزیع پذیری[۵] $\wedge$ روی $\vee$

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

صحّت و قابلیّت ارضا – یک عبارت در صورتی **درست** است که در **همه ی** مدل ها درست باشد . مثلاً ، $True, A \vee \neg A, A \Rightarrow A, (A \wedge (A \Rightarrow B)) \Rightarrow B$. صحّت مربوط می شود به نتیجه گیری از راه قضیّه ی قیاس[۶] : $KB \models \alpha$ اگر و فقط اگر $(KB \Rightarrow \alpha)$ مجاز باشد . یک عبارت *راضی کننده* است اگر در برخی از مدل ها درست باشد ؛ مثل ، $A \vee B, C$ و یک جمله *ناراضی کننده* است اگر در هیچ مدلی درست نباشد ؛ مثل ، $A \wedge \neg A$

قابلیّت ارضا وابسته به نتیجه ی به دست آمده از راه زیر است: $KB \models \alpha$ اگر و فقط اگر $(KB \wedge \neg\alpha)$ ناراضی کننده باشد . α را به وسیله ی برهان خلف[۷] اثبات نمایید .

---

[۱] contraposition

[۲] implication

[۳] biconditional elimination

[۴] De Morgan

[5] distributivity

[۶] Deduction Theorem

[7] reductio ad absurdum

متدهای اثبات [1]

به طور کلّی متدهای اثبات به دو دسته تقسیم می شوند:

- ✓ **استفاده از قوانین استنتاج :** در این روش ، معمولاً به ترجمه ی عبارات به صورت یک فرم نرمال نیازمندیم و برای اثبات ، یک سری از قوانین استنتاج را به کار می بریم .
- ✓ **بررسی مدل** [2] : که در این روش ، از جدول صحت استفاده می نماییم .

زنجیره ی مستقیم و معکوس

زنجیره ی مستقیم : از قوانین و چیزهای موجود استفاده می کنیم تا چیزهای اضافی را به دست آوریم و اگر به عبارت مورد نظر رسیدیم ، دیگر این کار را ادامه نمی دهیم . این روش ، استنتاج مشتق شده از داده ها [3] هم نام دارد و برای به دست آوردن واقعیّات یا قوانین جدید ، بسیار مفید می باشد .

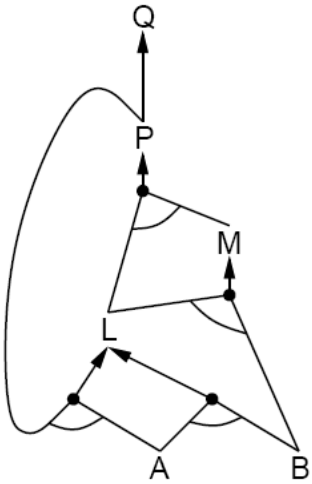
$$P \Rightarrow Q, L \wedge M \Rightarrow P, B \wedge L \Rightarrow M, A \wedge P \Rightarrow L, A \wedge B \Rightarrow L, A, B$$

---

[1] Proof methods

[2] Model checking

[3] data-driven reasoning



الگوریتم زنجیره ی مستقیم

تابع PL-FC-Entails?(KB,q) درست یا غلط را برمی گرداند

ورودی ها ، KB ، که پایگاه دانش ، یک مجموعه از شروط گزاره ای شیپوری [1]

q ، صف ، که یک سمبل گزاره ای می باشد

متغیّرهای محلی [2]: count ، یک جدول شاخص گذاری شده توسط شرط ، به صورت اوّلیه حاوی تعداد permis ها می باشد

---

[1] یک عبارت شیپوری ، عبارت هایی " یایی " هستند که در آن ها حداکثر یک لفظ مثبت وجود دارد ؛ مثل ، $\neg A\vee\neg B\vee\neg C\vee D$ و $\neg A \vee \neg B$ ؛ عبارت های شیپوری می توانند به صورت استنتاج هایی با یک نتیجه (تالی) نوشته شوند ؛ مثلا ، $\neg A\vee\neg B\vee\neg C\vee D$ می تواند به صورت $A\wedge B\wedge C\Rightarrow D$ نوشته شود . عبارت های شیپوری ، مبنا یا پایه ی برنامه نویسی منطقی هستند .

[2] local variables

inferred یک جدول شاخص گذاری شده توسط سمبل ، هر ورودی دارای مقدار اوّلیـه ی false مـی باشد

agenda ، لیستی از سمبل ها ، به صورت اوّلیه سمبل ها در KB شناخته می شوند

مادامی که agenda خالی نمی باشد کارهای زیر را انجام بده

p← Pop(agenda)

و گرنه برای inferred[p] کارهای زیر را انجام بده

inferred[p] ← true

برای هر شرط شیپوری c که در p قضیّه به دست می آید کارهای زیر را انجام بده

count[c] را کاهش بده

در صورتی که count[c]=0 می باشد کارهای زیر را انجام بده

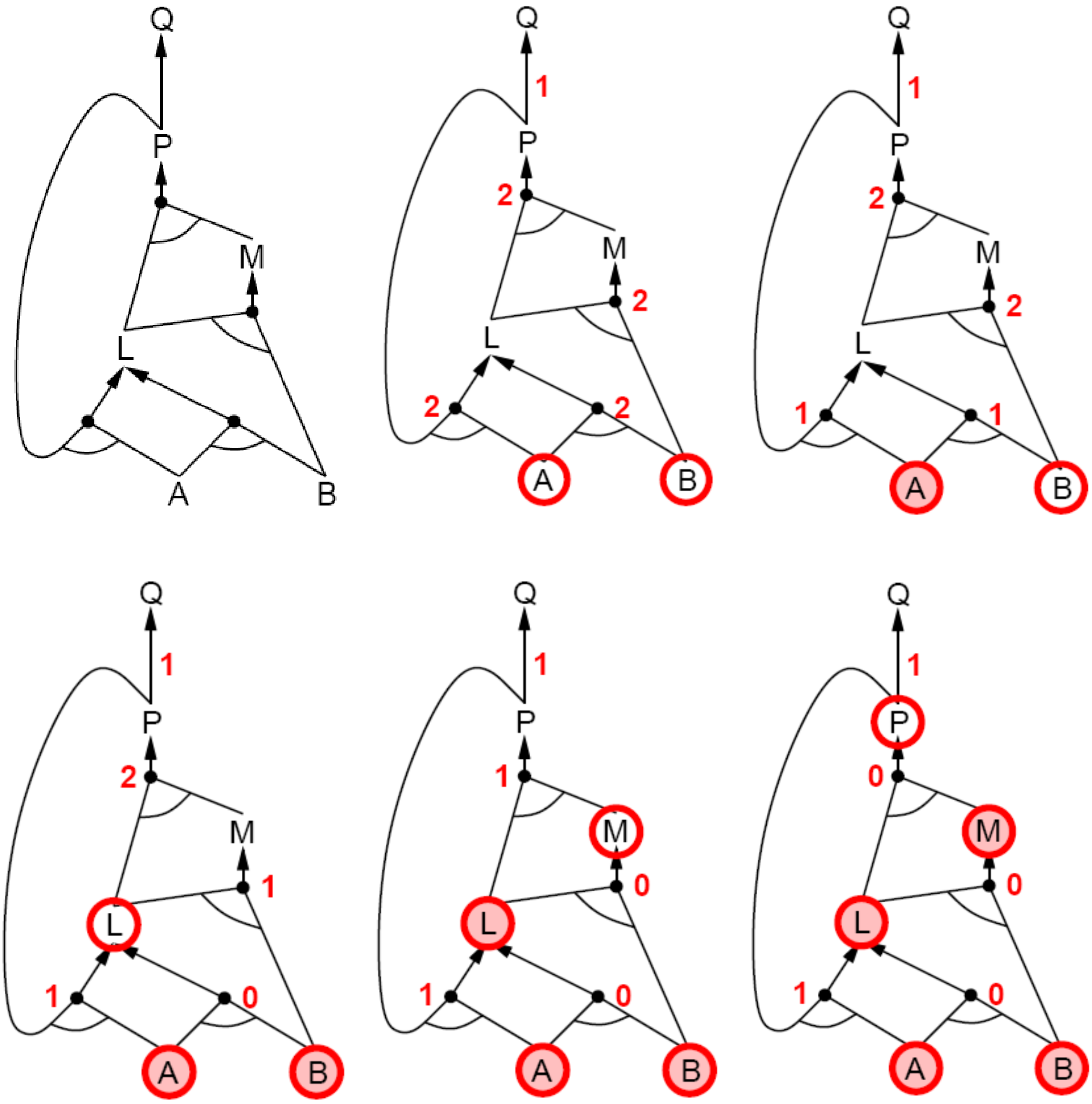
در صورتی که Head[c]=q می باشد true را برگردان

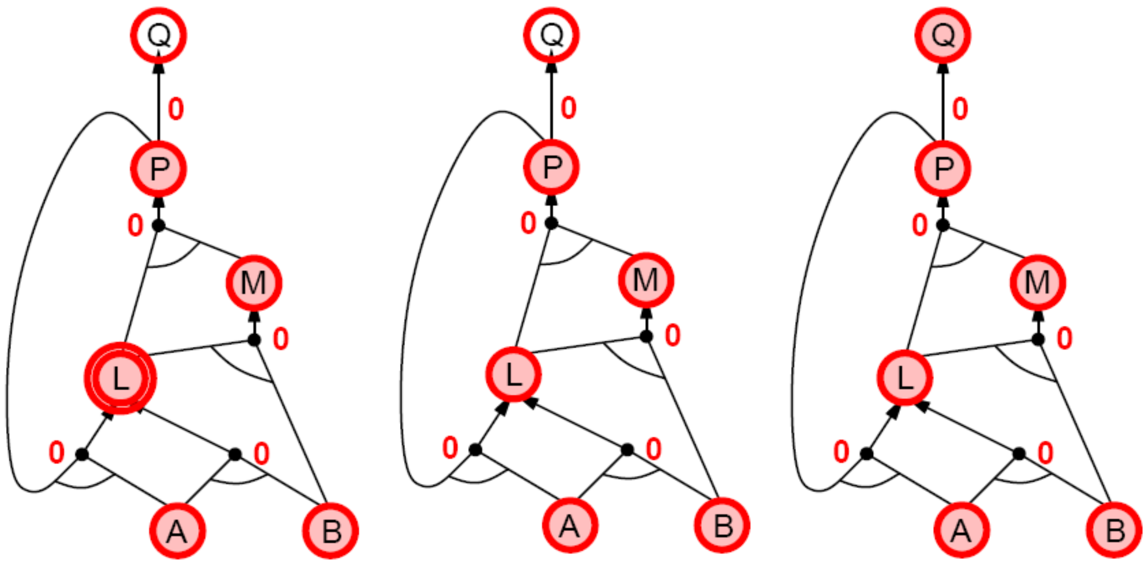
Push(Head[c],agenda)

false را برگردان

مثال زنجیره ی مستقیم

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


Q
P
M
L
A
B



اثبات تمامیّت

FC یا زنجیره ی مستقیم ، هر عبارت اتمیک که توسط KB به وجود آمده است را نتیجه می دهد ؛ FC در جایی که هیچ عبارت اتمیک جدیدی مشتق نمی شود به یک **نقطه ی ثابت** [1] می رسد ؛ توجه کنید که حالت نهایی به صورت یک مدل m ، نسبت دهنده ی درست / غلط به سمبل ها می باشد ؛ هر شرط در KB اصلی ، در m درست است ؛ از این رو m مدلی از KB می باشد ؛ در صورتی که KB|=q ، q در هر مدل KB ، شامل m درست می باشد .

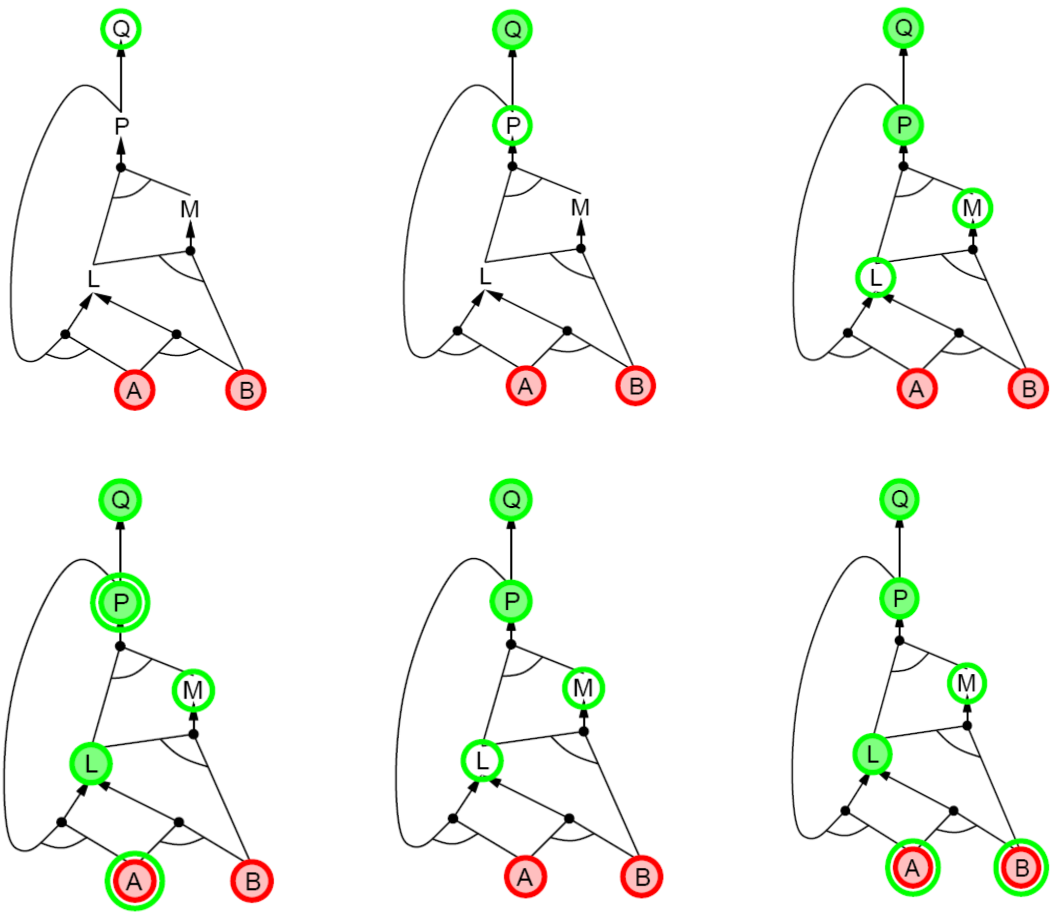
زنجیره ی معکوس

از نتیجه ، با پیدا کردن قانون هایی که می توانند نتیجه را به وجود بیاورند به عقب برمی گردد ؛ سپس تلاش می کند تا مقدمات قانون ها را به دست آورد . این روش برای حل مسأله مناسب است و برخی اوقات ،

[1] fixed point

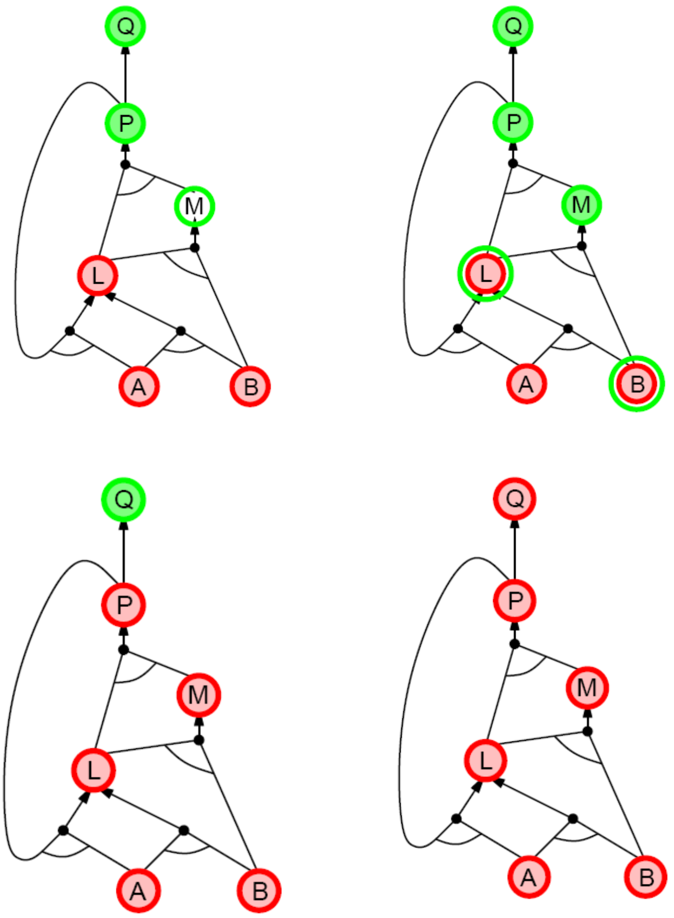
استنتاج مشتق شده از پرس و جو [۱] نامیده می شود . در این روش ، احتمال به وجود آوردن اطّلاعـات جدیـد و ناشناخته ، کم تر می باشد . زبان برنامه نویسی پرولوگ هم از زنجیره ی معکوس استفاده می نماید .

مثال زنجیره ی معکوس (شکل ها را از چپ به راست دنبال نمایید .)



---

[۱] query-driven reasoning



زنجیره ی مستقیم (FC) و معکوس (BC)

در FC از داده ها استفاده می کنیم و ممکن است تعدادی کار را که به هدف نامربوطند انجام دهیم . در BC از نتیجه ها شروع می کنیم و برای حلّ مسأله مناسب می باشد .

تحلیل

قوانین قبلی ، درست هستند ولی الزاماً کامل نمی باشند . خوشبختانه ، قانون کاملی برای استنتاج وجود دارد که این قانون ، *تحلیل* نام دارد . قانون تحلیل پایه شبیه این می باشد ، $A \vee B$ و $\neg A \vee C$ به ما اجازه می دهند که $B \vee C$ را استنتاج کنیم . در موقع تحلیل ، باید پایگاه دانش ما به صورت CNF باشد .

صورت نرمال ربط دهنده [1] (CNF)

اغلب ، مفید است که فرمول ها را به صورت های نرمال بنویسیم . صورت های نرمال با آنچه که قبل از نرمال کردن بودند فرقی ندارند و تفاوت آن ها در ظاهر آن هاست و برای استدلال ، مناسب ترند . یک حرف [2] ، یک چیز تجزیه ناپذیر [3] یا نقیض آن است مثل ، P یا نقیض آن که $\neg P$ است . یک فرمول به صورت CNF است اگر به صورت $A_1 \wedge A_2 \wedge \ldots A_k$ باشد که A_i تشکیل شده از یا (OR) گزاره ها یا نقیض آن ها . به عنوان مثال ؛

$(p \vee q) \wedge r \wedge (\neg p \vee \neg r \vee s)$ ، به صورت CNF است .

$\neg(p \vee q) \wedge r \wedge (\neg p \vee \neg r \vee s)$ ، به صورت CNF نمی باشد .

$(p \vee q) \wedge r \wedge (p \Rightarrow (\neg r \vee s))$ ، به صورت CNF نمی باشد .

نکته : *هر عبارت منطق گزاره ای می تواند به فرم نرمال ربط دهنده تبدیل شود .*

تبدیل یک عبارت به CNF

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

---

[1] Conjunctive Normal Form یا CNF
[2] literal
[3] atom

۱. حذف کردن $\Leftrightarrow$ ؛ جایگزین کردن $\alpha \Leftrightarrow \beta$ با $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$

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

۲. حذف کردن $\Rightarrow$ ؛ جایگزین کردن $\alpha \Rightarrow \beta$ با $\neg\alpha \vee \beta$

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

۳. $\neg$ را با استفاده از قوانین دمورگان و قرینه سازی دوگانه به درون عبارات ، حرکت دهید :

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

قوانین توزیع و گسترش را روی $\vee$ و $\wedge$ به کار ببرید:

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

مثال :

فرمول نهایی (به صورت CNF)	فرمول اوّلیه
$(A \vee B) \wedge (A \vee C)$	$A \vee (B \wedge C)$
$(B \vee A) \wedge (C \vee A)$	$(B \wedge C) \vee A$

مزیّت های منطق گزاره ای

- اعلانی یا اظهاری بودن این روش – دانش می تواند از استنتاج ، مجزّا باشد .

- می تواند اطّلاعات جزیی را به کار بگیرد .
- می تواند عبارت های پیچیده تر را به صورت ساده تر تولید نماید .
- مکانیزم های صحیح و کامل دارد (برای عبارت های شیپوری ، کارآمد می باشد)

معایب منطق گزاره ای

- افزایش تشریحی در تعداد لفظ ها
- راهی برای تشریح ارتباط های میان اشیا وجود ندارد .
- راهی برای بیان کیفیّت ، در مورد اشیا وجود ندارد .

منطق مرتبه ی اوّل ، روشی برای رسیدگی کردن به این مسایل می باشد .

فصل دهم

منطق مرتبه ی اوّل[1] به بیان ساده

[1] First-Order Logic(FOL)

منطق مرتبه ی اوّل ، تمام خصوصیّات منطق گزاره ای را داراست . منطق مرتبه ی اوّل براساس این ایده است که جهان مرکب از دو نوع موجودیت [۱] می باشد : اشیا [۲] و ارتباطات [۳] ؛ اشیا ، معمولاً به نام ها و چیزها اشاره می نمایند ، به عنوان مثال ، جورج بوش [۴] و خورشید و ارتباطات معمولاً به خصوصیات [۵] و ویژگی ها [۶] اشاره می کنند ، به عنوان مثال ، جورج بوش زنده است . خورشید ، داغ است . لیلی پاتر [۷] ، مادر هری پاتر [۸] است .

در تفسیر زبان طبیعی از " هر ... " یا " همه ... " استفاده می کنیم ؛ برای مثال ، " هر شخصی دارای یک والد است . " ، " همه ی پرندگان پرواز می نمایند . " ؛ این مطالب اظهار می کنند که همه ی اشیا دارای یک ویژگی معین هستند . همچنین از " برخی ... " هم استفاده می نماییم ؛ برای مثال ، " برخی از پرندگان

---

[۱] entity
[۲] objects
[۳] relations
[۴] George Bush
[۵] properties
[۶] attributes
[۷] Lily Potter
[۸] Harry Potter

نمی توانند پرواز نمایند " ، " برخی از افراد از شطرنج لذت می برند " ؛ این مطالب اظهار می کنند که لااقل ، یک شی دارای یک خصوصیت معین می باشد .

یک زبان مرتبه ی اوّل شامل موارد زیر است :

۱- یک مجموعه از ثابت ها [۱] ؛ که با حروف کوچک مشخص می شوند می باشد ، نظیر a ، b ، c و ... ؛ به طور حسّی ، یک ثابت نام یک شی می باشد .

۲- یک مجموعه از متغیّرها ، معمولاً با حروف کوچک مشخص می شوند ، نظیر x ، y ، z و ...

۳- یک مجموعه از سمبل های تابعی ، که معمولاً با حروف کوچک نمایش داده می شوند ، نظیر f ، g ، h و ... ؛ هر سمبل تابعی دارای یک مقدار صحیح و مثبت می باشد .

مجموعه ای از واژگان با قوانین [۲] زیر تعریف می شود :

۱. یک ثابت ، یک واژه می باشد

۲. یک متغیّر ، یک واژه می باشد

۳. یک عبارت [۳] $f(t_1,...,t_n)$ در صورتی یک واژه است که یک سمبل تابعی و هر t_i واژه باشند

۴. هیچ چیز دیگری واژه نمی باشد

گزاره ها [۱]

---

[۱] constants

[۲] rules

[۳] expression

معنی یک فرمول منطق مرتبه ی اوّل

در منطق مرتبه ی اوّل ، معنی $\neg ,\vee ,\wedge$ و $\rightarrow$ مثل معنی آن ها در منطق گزاره ای می باشد ، به عنوان مثال در صورتی که φ و ψ دو فرمول باشند آن گاه $\varphi \vee \psi$ در صورتی درست می باشد که لااقل یکی از φ یا ψ درست باشند . در زمانی که سور ها وجود دارند ، صحّت یک فرمول باتوجه به دامنه ی سخن تشخیص داده می شود ؛ یک دامنه که معمولاً با حرف D نمایش داده می شود فقط یک مجموعه می باشد ، یک مجموعه می تواند مثل اعداد طبیعی {۱،۲،۳،...} یا مجموعه ای از افراد باشد . به فرمول $\forall x : \varphi$ و یک دامنه ی D توجه نمایید ، فرمول $\forall x : \varphi$ در دامنه ی D درست می باشد اگر و فقط اگر $\varphi(x|d)$ برای هر $d \in D$ درست باشد ، که $\varphi(x|d)$ فرمولی گرفته شده از φ بعد از جایگزین نمودن هر وقوع x در φ با d می باشد . به عنوان مثال ، به فرمول $\forall x : has\_chair(x)$ و یک دامنه ی D تشکیل شده از دانش آموزانی که در حال حاضر در OMB 31 هستند توجه نمایید . فرمول $\forall x : has\_chair(x)$ در دامنه ی D درست است اگر و فقط اگر $has\_chair(Sanjeev)$ درست باشد ، $has\_chair(John)$ درست باشد و همین طور این مورد برای هر دانشجوی درون OMB 31 هم درست باشد . حال به فرمول $\exists x : \varphi$ و دامنه ی D توجه نمایید ؛ فرمول $\exists x : \varphi$ در دامنه ی D درست است اگر وفقط اگر $\varphi(x|d)$ برای برخی از $d \in D$ درست باشد که $\varphi(x|d)$ فرمولی گرفته شده از φ بعد از جایگزین نمودن هر وقوع x در φ با d می باشد به عنوان مثال ، به فرمول $\exists x : s\tan ding(x)$ و یک دامنه ی D مرکب از افرادی که در حال حاضر در OMB 31 هستند توجه نمایید ؛ فرمول $\exists x : s\tan ding(x)$ در دامنه ی D درست است اگر و فقط اگر لااقل یک شخص در OMB 31 ایستاده باشد و بنابراین standing(Lecturer) درست و داریم : $\exists x : s\tan ding(x)$ درست می باشد . توجه کنید که یک فرمول ممکن است در برخی از دامنه ها درست باشد امّا در برخی دیگر نادرست باشد ؛ به عنوان مثال ، به فرمول $\forall x \exists y : (x = y^2)$ توجه نمایید ؛ این فرمول در دامنه ی اعداد حقیقی مثبت درست می باشد ولی در دامنه ی همه ی اعداد حقیقی این عبارت غلط می باشد ؛ اگر x منفی باشد ، x نمی تواند با هر y^2 ای برابر باشد ، از آن جایی که $y^2 \geq 0$ می باشد .

تعریف – یک فرمول که در همه ی دامنه ها درست است یک *همانگویی*[۱] *(همیشه درست)* می باشد .

تعریف – یک فرمول که در همه ی دامنه ها غلط است یک *خلاف گویی*[۲] *(تناقض)* است .

بیان نسبت خانوادگی

منطق مرتبه ی اوّل یک زبان ارایه ی قدرتمند برای دانش می باشد ؛ به ارایه ی نسبت خانوادگی به زبان انگلیسی توجه نمایید ، گزاره ها ممکن است شامل موارد زیر باشند :

- Parent(_ , _)[۳]
- Child(_ , _)[۴]
- Grandparent(_ , _)[۵]
- Sibling(_ , _)[۶]
- Male(_)[۷]
- Female(_)[۱]

---

[۱] tautology
[۲] contradiction
[۳] پدر یا مادر
[۴] بچه ، فرزند
[۵] پدر بزرگ یا مادر بزرگ
[۶] برادر یا خواهر
[۷] مرد

- - = (_ , _)

راهی کلّی برای تبدیل عبارت های انگلیسی به منطق مرتبه ی اوّل وجود ندارد ؛ در این مورد نظریّه های [۲] غیردقیق [۳] سادگی [۴] و اختصار [۵] وجود دارد . در زیر جملاتی بیان شده و معادل آن ها در منطق مرتبه ی اوّل هم آورده شده است :

- " James پدر Harry است "
  - $Parent(James, Harry) \wedge Male(James)$
- " Lily مادر Harry است "
  - $Parent(Lily, Harry) \wedge Female(Lily)$

ارتباطات خانوادگی

- " مردان [۶] و زنان [۷] کاملاً متمایز هستند
  - $\forall x : (Male(x) \leftrightarrow \neg Female(x))$
  - $p \leftrightarrow q$ خوانده می شود " p معادل با q است و کوتاه نویسی برای $((p \to q) \wedge (q \to p))$ می باشد

---

[۱] زن
[۲] notions
[۳] imprecise
[۴] simplicity
[۵] conciseness
[۶] males
[۷] females

- برادر و خواهر [1] افرادی دقیقا متمایز هستند که دارای یک والد مشترک می باشند
  - $\forall x \forall y (Sibling(x,y) \leftrightarrow (\neg = (x,y) \wedge \exists p \ (Parent(p,x) \wedge Parent(p,y)))$

منطق مرتبه ی اوّل و منطق گزاره ای

در منطق مرتبه ی اوّل $\forall$ لازم است ؛ به فرمول $\forall x : P(x)$ توجه نمایید ، برای یک دامنه ی محدود $D = \{a_1,...,a_n\}$ این عبارت دقیقا در زمانی درست است که $P(a_i)$ برای $1 \le i \le n$ درست باشد : $\forall x : P(x) \equiv P(a_1) \wedge ... \wedge P(a_n)$ و برای یک دامنه ی نامحدود $D = \{a_1,...,a_n,...\}$ این عبارت دقیقا در زمانی درست است که $P(a_i)$ برای هر i در عبارت زیر درست باشد : $\forall x : P(x) \equiv P(a_1) \wedge ... \wedge P(a_n) \wedge ...$ ، امّا یک فرمول نمی تواند دارای یک طول بی نهایت باشد .

در منطق مرتبه ی اوّل ، $\exists$ هم لازم است ؛ به فرمول $\exists x : P(x)$ توجه نمایید ، برای یک دامنه ی محدود $D = \{a_1,...,a_n\}$ این عبارت دقیقا در زمانی درست است که $P(a_i)$ برای $1 \le i \le n$ درست باشد : $\exists x : P(x) \equiv P(a_1) \vee ... \vee P(a_n)$. برای یک دامنه ی نامحدود $D = \{a_1,...,a_n,...\}$ این عبارت دقیقا در زمانی درست است که $P(a_i)$ برای هر i در عبارت زیر درست باشد : $\exists x : P(x) \equiv P(a_1) \vee ... \vee P(a_n) \vee ...$ ، امّا یک فرمول نمی تواند دارای یک طول بی نهایت باشد .

نقیض و سورها

داریم :

- $\neg \forall x : P(x) \equiv \exists x : \neg P(x)$
- $\forall x : \neg P(x) \equiv \neg \exists x : P(x)$

---

[1] siblings

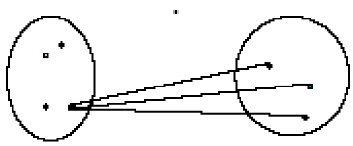
- $\neg\forall x : \neg P(x) \equiv \exists x : P(x)$
- $\forall x : P(x) \equiv \neg\exists x : \neg P(x)$

مثال برای نقیض و سورها

به گزاره ی (_)male در دامنه ی دانش آموزان درون OMB 31 توجه نمایید . معنای $\neg\forall x : male(x)$ این است که موردی وجود ندارد که در آن هرکس در OMB 31 ، مرد باشد ؛ این مطلب معادل با این عبارت است که کسی در OMB مرد نمی باشد ، ظاهرا معادل $\exists x : \neg male(x)$ می باشد . سایر معادل ها عبارتند از : $\exists x \exists y : P(x, y) \equiv \exists y \exists x : P(x, y)$ و $\forall x \forall y : P(x, y) \equiv \forall y \forall x : P(x, y)$.

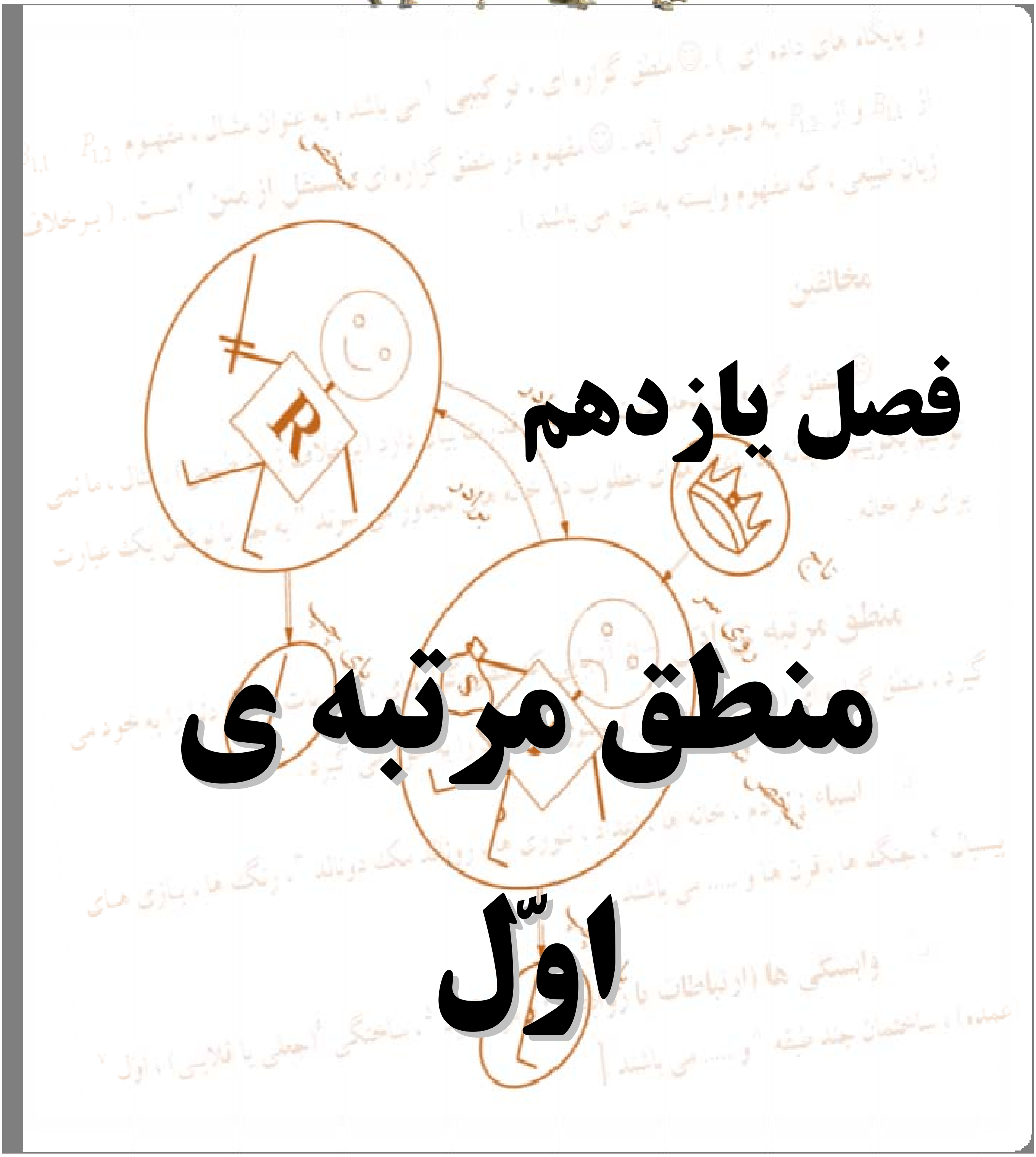
به عنوان مثال دیگر ، به دامنه ی $G1 \cup G2$ از دو گروه افراد G1 و G2 توجه نمایید . (_)Group1 و (_)Group2 گزاره ها ی یگانی هستند که نشان می دهند که به ترتیب کسی در گروه G1 و G2 هست . (_ , _)Know هم یک گزاره ی دودویی نشان دهنده ی این که دو نفر همدیگر را می شناسند می باشد :

- $\forall x \forall y (Group1(x) \wedge Group2(y) \rightarrow Know(x, y))$



[redacted]

$\forall y \forall x (Group1(x) \wedge Group2(y) \rightarrow Know(x, y))$



فصل یازدهم

منطق مرتبه ی اوّل