

روش های بازنمایی دانش

بازنمایی دانش (نمایش دانش)

در ادامه، تکنیک های مختلف در کد کردن دانش در یک سیستم خبره را مورد بحث قرار می گیریم.
ارائه دانش = کد کردن دانش

تفاوت برنامه های متداول با سیستم های خبره از دید دانش

- 0 برنامه های متداول روی داده ها عمل پردازش انجام می دهند (داده را پردازش می کنند)
- 0 سیستم های خبره روی دانش عمل پردازش انجام می دهند (دانش را پردازش می کنند)

دلایل نمایش دانش در سیستم های خبره

1. پوسته سیستم های خبره طوری طراحی می شوند که برای روش خاصی از نمایش دانش مثل قواعد، منطق یا گراف مناسبند.
2. روش ارائه دانش بر روی سرعت، توسعه و کارایی و نگهداری سیستم موثر است.

Domain Expert: اشاره به فرد خبره و متخصص انسانی دارد. یعنی فردی که می تواند یک مساله را به گونه ای حل نماید که دیگران قادر به آن نیستند. منظور از Domain این است که یک محدوده خاصی داشته باشد مثل مکانیک، مهندسی، پزشکی و ...

$$|\text{Expert}| = \text{Expert} - \text{Non Expert} = \text{Knowledge}$$

👉 نکته: تفاوت Expert و Non Expert در دانش است؛ به عبارت دیگر، تفاوت یک فرد خبره و یک فرد عادی در داشتن دانش تخصصی است.

👉 پرسش: دانش چیست؟ دانش به فهم و دانایی اشاره می کند. از نظر فلسفی، دانش مفهومی است که همه می دانند ولی قادر به تعریف آن نیستند. روشن است که هر چه دانش بیشتر باشد توانایی هم بیشتر خواهد شد.

Knowledge is Power (Francis Bacon)

توانا بود هر که دانا بود (فروسی)

و سیستم خبره قوی تر، دارای دانش بیشتری است. با توجه به دانش، یک تعریف از سیستم خبره به صورت زیر است:

$$\text{Expert System} = \text{Knowledge} + \text{Reasoning}$$

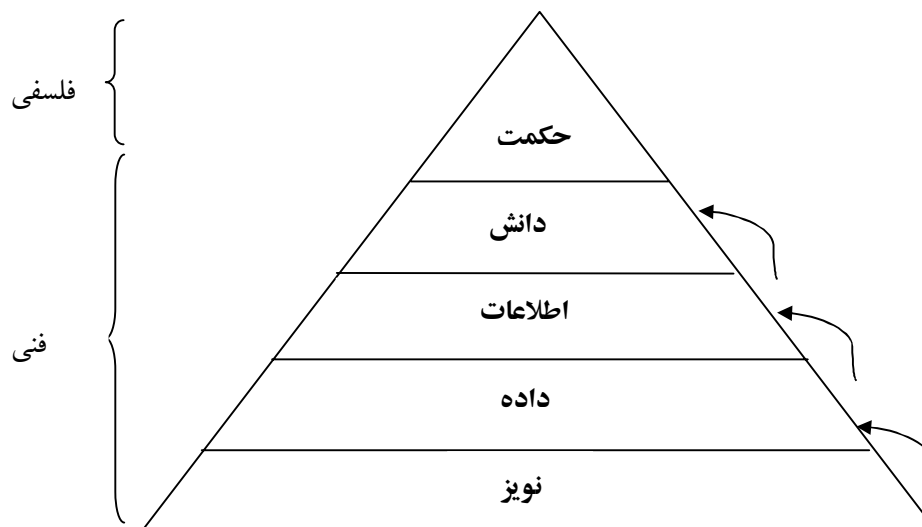
استدلال + دانش = سیستم خبره

همانطور که یک برنامه، ترکیبی از الگوریتم ها و ساختمان داده هاست.

$$\text{Program} = \text{Data structure} + \text{Algorithms}$$

دانش، قدرت است. (فرنسیس بکن)

هرم سلسله مراتبی دانش



شکل 5-1: هرم دانش

- 0 در پایین ترین سطح نويز قرار دارد.
- 0 داده: قسمت های مفید دریافتی است.
- 0 اطلاعات: داده های پردازش شده است.
- 0 دانش: اطلاعاتی که پردازش شده است
- 0 حکمت (خرد): دانشی است که مرتبط با هدف و ارزش است و به زندگی و فلسفه آن مربوط است.

نکته: اگر داده ای، هیچ مفهومی نداشته باشد نويز خواهد بود. اما اگر بدانیم این سمبل ها و نمادها برای منظوری است، داده محسوب می شود. به طور مثال اگر شخصی به زبانی سخن بگوید که برای شما ناآشناست، سخنان آن شخص برای شما نويز محسوب می شود چون هیچ مفهومی برای شما ندارد.

- **نويز:** داده ای که هیچ مفهومی ندارد.
- **داده:** می دانیم برای منظوری استفاده می شود.
- **دانش:** فهم در یک موضوع خاص است.

مثال: فرض کنید یک رشته 24 عددی داریم که از هر چیزی ممکن است تشکیل شده باشد. اگر ندانیم این رشته به چه منظور استفاده می شود نويز است ولی اگر بدانیم برای بیان قیمت، نمره و ... استفاده شود، داده است. برای این که از داده به اطلاعات برسیم به الگوریتم نیاز داریم. پس تبدیل داده به اطلاعات توسط یک الگوریتم صورت می گیرد. مثلاً اگر رشته 24 عددی را 8 تا 8 تا جدا کنیم هر 8 کاراکتر نمایانگر یک کد اسکی است و می توانیم به کد اسکی برگردانیم.

Information is not knowledge. The only source of knowledge is experience.

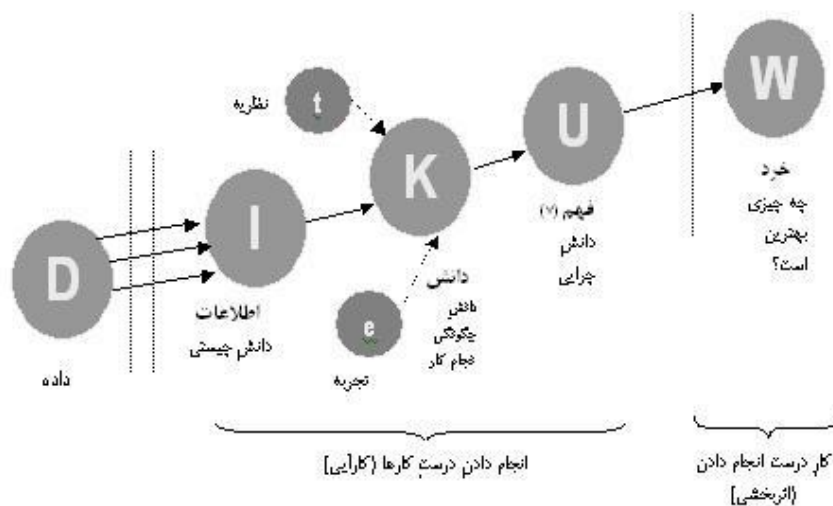
اطلاعات، دانش نیست. تنها منبع دانش، تجربه است. (آلبرت اینشتین)

یا تصور کنید این عدد به Gold + تبدیل شود که خود یک اطلاعات است. پس الگوریتم مبدل داده به اطلاعات است که Gold + نمایانگر افزایش قیمت طلا باشد. اگر از این اطلاعات در جایی استفاده شود تبدیل به دانش می شود مثلاً "چون قیمت طلا در حال افزایش است پس طلا خریداری شود" یک دانش را ایجاد کرده است.

- دانش به کار گیری اطلاعات است

نکته: تعاریف بالا برای داده، اطلاعات، دانش و حکمت (شکل 1-5)، نسبی است یعنی ممکن است چیزی که در محیطی دانش، در نظر گرفته می شود، در محیط دیگر داده باشد.

پوشش: شکل 2-5 را تفسیر کنید. تفاوت بین کارآیی و اثربخشی را با مثال توضیح دهید.¹

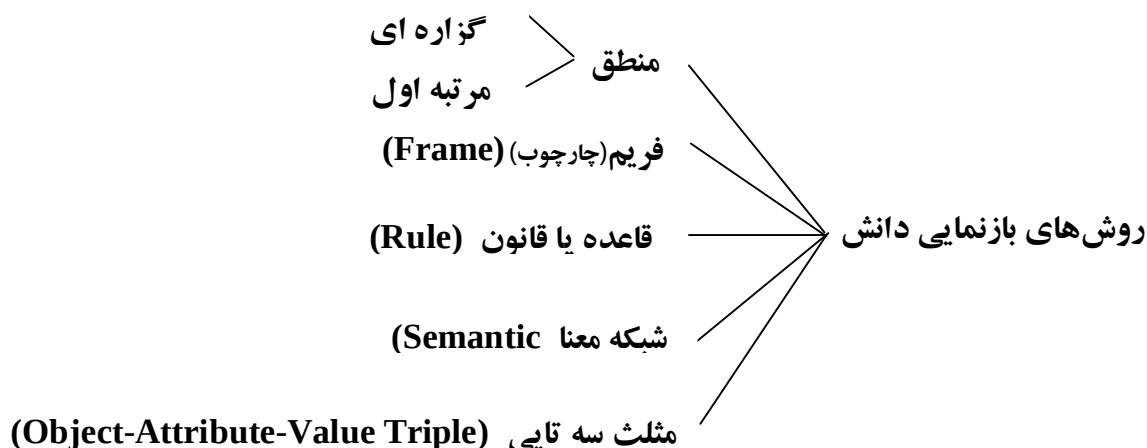


شکل 2-5: روند تبدیل از داده تا خرد

¹ - میان اثربخشی (انجام کار درست) و کارآیی (انجام درست کار) تفاوت وجود دارد. بهتر است انسان کار درست انجام دهد تا کار را به روش درست انجام دهد. (پیر دراکر)

بازنمایی دانش

نمایش دانش یا بازنمایی دانش، مجموعه روش‌هایی است که برای کد کردن دانش در پایگاه دانش سیستم خبره استفاده می‌شود. پس از آنکه دانش از یک فرد خبره در دامنه خوش - متمرکز (Well-focused Domain) به دست آمد، بایستی در یک سیستم خبره کد شود. برای کد کردن دانش نیاز به ساختار بندی دانش است به گونه‌ای که سیستم خبره قادر به حل مساله به روشی مشابه فرد خبره باشد.



روش‌های دیگری علاوه بر روش‌های بالا، برای بازنمایی دانش وجود دارد.

نکته: در علوم شناختی، حقایق (Facts) بلوک‌های سازنده در سازماندهی دانش انسان است. حقایق فهم و دانایی ما از یک رویداد یا مساله است که قالبی از دانش توصیفی است. در هوش مصنوعی و سیستم‌های خبره، حقایق بخشی از چارچوب‌ها (Frames)، شبکه‌های معنا و قوانین است که اغلب به یک گزاره اشاره دارد. گزاره: عبارتی است درست یا نادرست. مثال: "هوا بارانی است" یک گزاره است که ممکن است درست یا نادرست باشد.

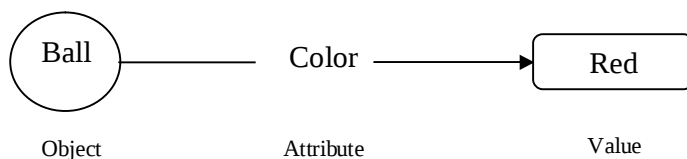
1- روش مثلث سه تایی OAV

OAV یک نوع گزاره پیچیده‌تر است که یک عبارت را به سه بخش تقسیم می‌کند. OAV به گزاره، ساختار یا بخش‌بندی می‌دهد. این بخش‌بندی تشکیل شده است از Object و Attribute و Value.

نکته: یک ویژگی OAV این است که امکان مدیریت بخش‌های متفاوت گزاره دارد که سبب می‌شود OAV جذاب‌تر از بازنمایی گزاره‌ای ساده باشد.

The ball's color is red

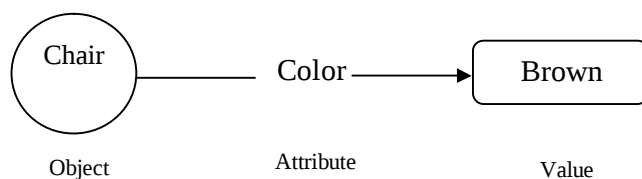
در شکل 3-5، برای گزاره بالا، سه بخش اصلی OAV مشخص شده است.



شکل 3-5: سه تایی OVA برای توپ

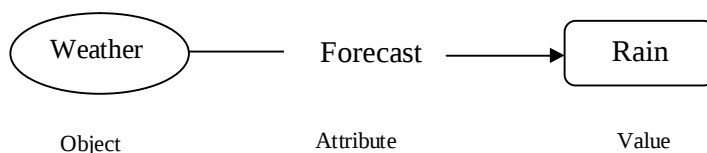
و یا در گزاره دیگری داریم:

The Chair's color is brown.



شکل 5-4: سه تایی OVA برای صندلی

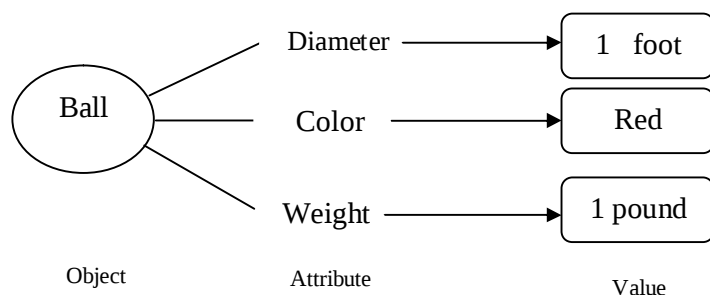
درمورد گزاره It's raining داریم



شکل 5-5: سه تایی OVA برای آب و هوا

یک شی با چندین ویژگی (Object with Multiple Attribute)

در شکل زیر یک Object با چندین Attribute نمایش داده شده است که نمایانگر این است که یک شی ممکن است چندین ویژگی داشته باشد.



شکل 5-6: سه تایی OVA با چندین ویژگی

این دانش، دانش سه تایی نامیده می شود زیرا دانش را به سه بخش تبدیل می کند که مدیریت و سازماندهی آن ساده تر است ولی اساساً بیانگر یک گزاره است.

🔗 نکته 1: یک Object می تواند مفهومی فیزیکی (ماشین، توپ و ...) یا انتزاعی (عشق، دوستی و ...) باشد.

🔗 نکته 2: Value می تواند یک مقدار داشته باشد (Single value facts) و یا چند مقدار (Multi Value facts) داشته باشد.

فشارسنج مثالی است از single value facts که از سه مقدار در حال افزایش، ثابت و در حال کاهش در هر لحظه تنها یک مقدار می‌گیرد. یعنی در هر لحظه فشارسنج تنها یکی از مقادیر در حال افزایش، در حال کاهش و یا ثابت، نشان می‌دهد. (در فرم‌ها این خاصیت مشابه Radio Box ها است)

سطح تحصیلات درجات مختلفی دارد که نشانگر Multi value facts است به طور مثال اگر درجات تحصیلی را به دیپلم، لیسانس، فوق لیسانس و دکترا تقسیم کنیم، اگر شخصی دارای مدرک فوق لیسانس باشد مسلماً دیپلم و لیسانس هم دارد پس چند Value را در بر می‌گیرد. پس در Multi value facts یک Attribute بیش از یک Value دارد. (در فرم‌ها این خاصیت مشابه Check box ها است)

انواع دانش

دانش می‌تواند به انواع گوناگونی تقسیم شود:

دانش رویه‌ای (Procedural Knowledge)

دانش رویه‌ای دانشی است که چگونگی حل یک مساله را توضیح می‌دهد و به بررسی قوانین، استراتژی‌ها، لیست اهداف و پروسیجرها می‌پردازد.

دانش توصیفی (Declarative Knowledge)

دانش توصیفی دانشی است که در مورد بخش‌های شناخته‌شده‌ی مساله توضیح می‌دهد که شامل یک سری عبارت‌های دارای ارزش درست یا نادرست است.

فرا دانش (Meta Knowledge)

فرا دانش دانشی است که چگونگی استفاده از دانش را توضیح می‌دهد. این دانش، کمک می‌کند چه دانش‌هایی در حل یک مساله مناسب‌تر هستند.

دانش هیورستیک (Heuristic Knowledge)

دانش هیورستیک که گاهی قواعد سرانگشتی نامیده می‌شود فرآیند استدلال را هدایت می‌کند. این دانش تجربه‌ای است و فرد خبره از طریق تجربه به دست آورده است.

دانش ساختاری (Structural knowledge)

دانش ساختاری دانشی است ساختار یک دانش را توصیف می‌کند. این نوع دانش مدل کلی ذهنی یک خبره در مورد یک مساله تشریح می‌کند که این مدل ذهنی شامل مفاهیم، زیر مفاهیم و ... است.

در جدول 1-5 نمونه‌هایی برای انواع دانش آورده شده است:

جدول 5-1: انواع دانش

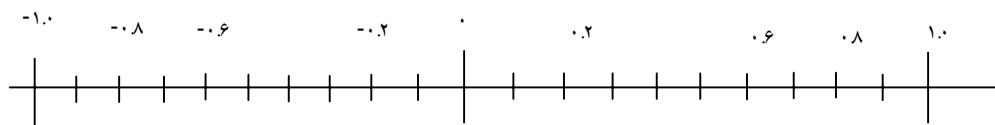
Types of Knowledge	
Procedural	قوانین Rules استراتژی ها Strategies اهداف Agendas رویه ها Procedures
Declarative	مفاهیم Concepts اشیا Objects حقایق Facts
Meta	Knowledge about the Other types of knowledge And how to use them دانشی که معین کند از کدام دانش استفاده شود و چگونه از آنها استفاده شود
Heuristic	قواعد سر انگشتی Rules of thumb
Structural	مجموعه قواعد Rule sets ارتباطات مفهومی Concept Relationship مفاهیم ارتباط اشیا Concept to object Relationship

حقایق قطعی (Certain Facts) و حقایق غیر قطعی (Uncertain Facts)

دنیای اطراف ما، یک دنیای سیاه و سفید نیست و رویدادهای اطراف ما تمام درست یا نادرست نیستند. برای درک بهتر حقایق قطعی و غیر قطعی به تعریف ضریب قطعیت نیاز داریم.

ضریب قطعیت (Certainty Factor = CF)

ضریب قطعیت بیانگر درجه باور نسبت به یک حقیقت است. اولین بار ضریب قطعیت در سیستم پزشکی MYCIN مورد استفاده قرار گرفت. شکل 5-7 مربوط به CF سیستم MYCIN می باشد. در این شکل CF مقادیری از 1.0 - تا 1.0 + دارد.



شکل 5-7: ضریب قطعیت

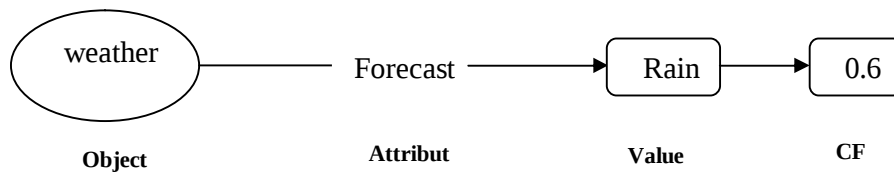
که در این نمودار اعداد نمایانگر مقادیر زیر است:

1.0- : کاملاً نادرست 0.8- : تقریباً نادرست 0.6- : احتمالاً نادرست

0.2- تا 0.2 : شناخته شده نیست

0.6 : احتمالاً درست 0.8 : تقریباً درست 1.0 : کاملاً درست

در مثال زیر همان گزاره It is raining با CF بیان می‌شود.



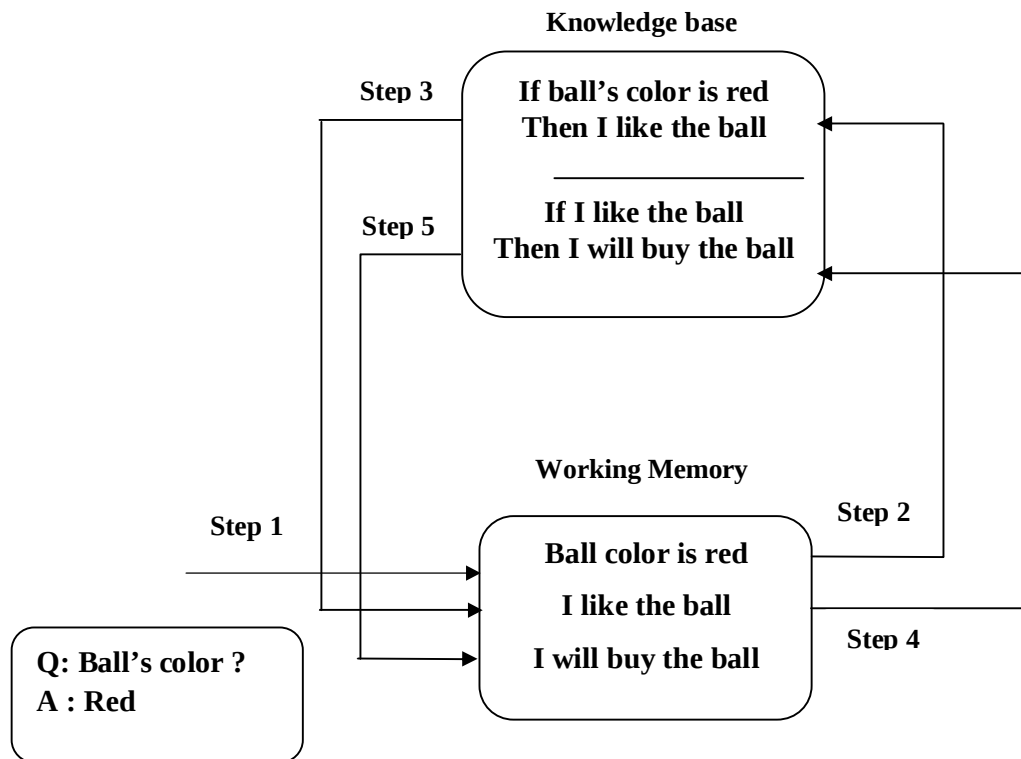
شکل 5-8: سه تایی OVA همراه با ضریب قطعیت

2- قانون (Rule)

قانون ساختاری از دانش است که اطلاعات شناخته‌شده‌ای را به اطلاعات شناخته‌شده دیگر مرتبط می‌سازد. این ساختار می‌تواند مورد استنتاج و نتیجه‌گیری قرار گیرد. به طور مثال فرض کنید سیستم خبره ای دارای دو قانون زیر باشد:

Rule1: IF the ball's color is red THEN I like the ball

Rule2: IF I like the ball THEN I will buy the ball



شکل 5-9: روند استنتاج در مثالی ساده (Durkin, 1994)

شکل 5-9، روند استنتاج در این مثال نشان می‌دهد. در سوال پرسیده می‌شود توپ چه رنگی است؟ پاسخ می‌دهد: قرمز

وقتی قرمز به سیستم داده شد قانون شماره 1 فعال می‌شود و نتیجه می‌گیرد توپ رادوست دارد و سپس قسمت پایین فعال می‌شود و نتیجه می‌گیرد توپ را خواهد خرید.

انواع قانون

قانون‌های رابطه‌ای (Relationship Rules)

IF the battery is dead
Then the car will not start

اگر باتری مشکل داشته باشد ماشین روشن نخواهد شد.
در این نوع قانون رابطه‌ای بین علت و معلول وجود دارد.

قانون‌های توصیه‌ای (Recommendation Rules)

IF the car will not start
THEN take a cab

اگر ماشین روشن نمی‌شود، کاپوت ماشین را بالا بزن.
قانون‌های توصیه‌ای حاوی یک پیشنهاد یا توصیه هستند.

قانون‌های دستوری (Directive Rules)

IF the car will not start
AND the fuel system is ok
THEN check out the electrical system

اگر ماشین روشن نمی‌شود و سیستم سوخت مشکلی ندارد سیستم برق را بررسی کن.
این قوانین به یک جمله امری ختم می‌شود.

قانون‌های راهبردی (Strategy Rules)

IF the car will not start
THEN first check out the fuel system
THEN check out the electrical system

اگر ماشین روشن نمی‌شود ابتدا سیستم سوخت را چک کن و سپس سیستم برق را چک کن.
قوانین راهبردی مراحل انجام کار را بیان می‌کند.

قانون‌های حسی-تجربی (Heuristic Rules)

IF the car will not start
AND the car is a 1957 Ford
THEN check the float

اگر ماشین روشن نمی‌شود و مدل ماشین فورد سال 1957 است پس شناور آن را چک کن
قوانین تجربی از روی تجربه‌های قبلی به انجام کار می‌پردازد.

در مثال زیر قبل از X علامت سوال می بینید این بدین معنی است که بهتر است در قوانین از متغیرها استفاده شود چون باعث می شود قوانین عام تر و کلی تر گردند.

IF ?X is Employee
AND ?X Age > 65
THEN ?X can retire

پرسش: اصطلاح Pattern Matching Rules یعنی تطبیق مقدار با متغیر در سیستم های قانون پایه چیست؟

فرا قانون (Meta Rules)

قانونی است که بیان می کند از چه قانونی استفاده شود.

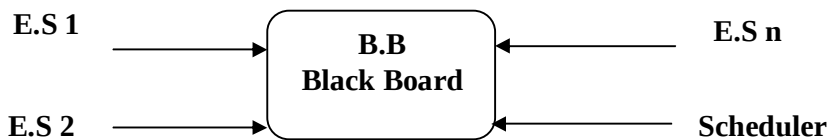
IF the car will not start
AND the electrical System is operating normally
THEN use rules concerning the fuel system

اگر ماشین روشن نمی شود و سیستم برق آن به درستی کار می کند قوانین مربوط به سیستم سوخت را به کار ببر.

تخته سیاه (Black Board)

حل مسائل توزیع شده (DPS) Distributed problem solving

ممکن است در مسائلی نیاز باشد چندین خبره در مورد یک مساله و برای حل آن بیندیشند. به عبارت دیگر چندین خبره به حل یک مساله بپردازند. چنین مسائلی از نوع مسائل توزیع شده است و یکی از مکانیزم های پیاده سازی آنها تخته سیاه یا Black Board است. مفهوم تخته سیاه بخشی از برنامه است که برای اشتراک گذاری راه حل ها استفاده می شود (شکل 5-10).



شکل 5-10: مفهوم تخته سیاه

در پیاده سازی سیستم خبره، تخته سیاه یک فایل نظیر یک فایل متنی است که راه حل های مسائل حل شده را داخل این فایل قرار می گیرند و تمام سیستم های خبره به این فایل دسترسی دارند و می توانند در صورت نیاز از راه حل های دیگران استفاده نمایند. به طور مثال یک پردازش متن چند زبانه یک مساله توزیع شده است، یعنی یک متن دارای چند زبان باشد و کسی که این متن را تهیه می کند می بایست خبرگی در زبان های مختلف داشته باشد و یا افراد متفاوتی که در یک مکانیکی کار می کنند که هر کدام در بخشی از اتومبیل نظیر بخش سوخت، بخش الکتریکی، موتور، چرخ ها و غیره، خبره هستند یا محسوب می شوند.

مثال: سیستم خبره عیب یاب کامپیوتر که پس از شناخت عیوب خبره‌های متفاوتی برای حل مشکل فرا خوانده می‌شوند، خبره در زمینه مانیتور، خبره در زمینه Mother Board و...

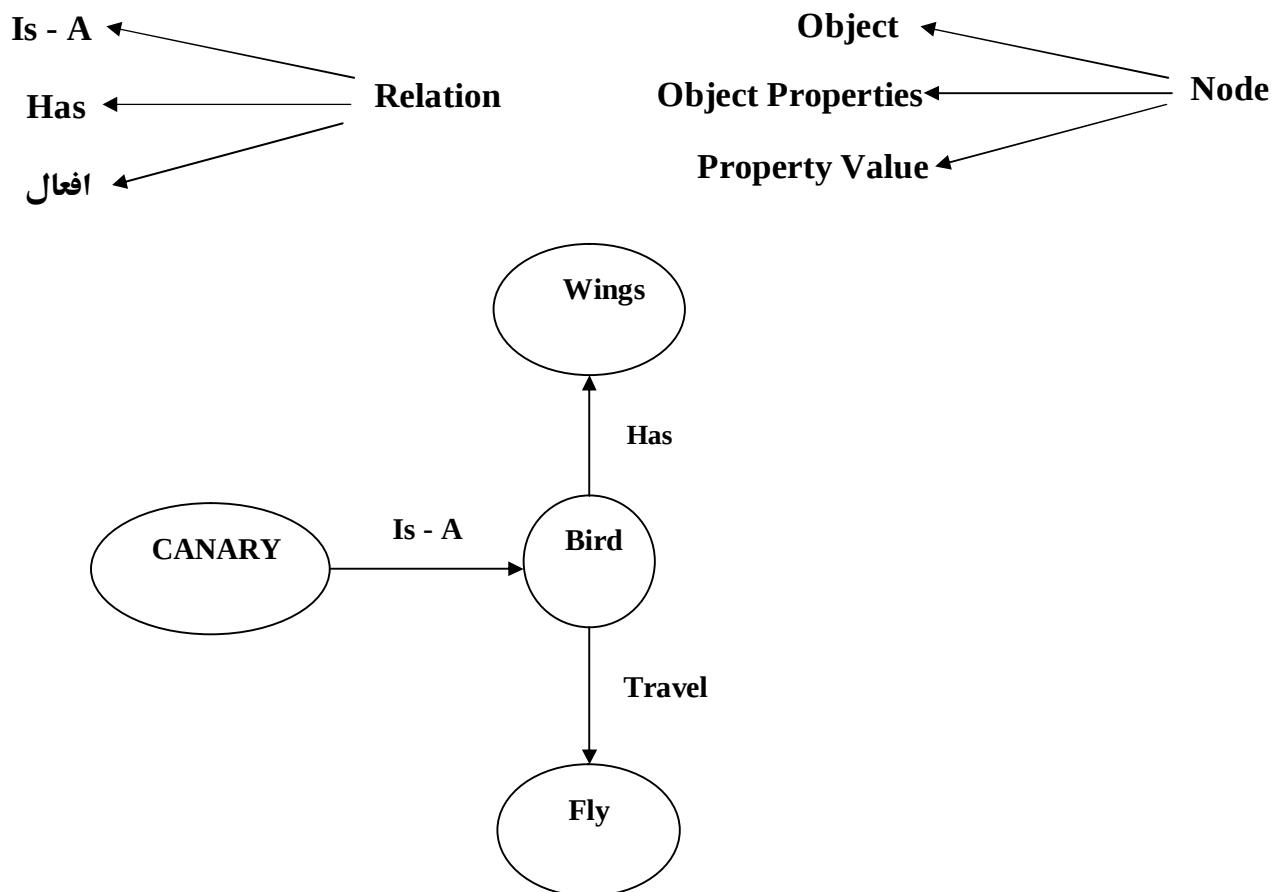
مؤلفه‌های اصلی سیستم‌های خبره توزیع شده:

- 1- مجموعه‌ای از سیستم‌های خبره (Community of different Expert System)
- 2- تخته‌سیاه (Black Board)
- 3- زمان‌بند (Scheduler)

سیستم تخته سیاه علاوه بر این که کمیت‌های از سیستم‌های خبره است بخشی برای قرار دادن راه حل‌ها (Black Board) نیز دارند و یک زمان‌بند نیز برای تقسیم زمان موجود و در اختیار قرار دادن زمان در میان خبره‌ها دارند.

3- شبکه معنا یا شبکه مفهومی (Semantic Network)

شبکه معنا روشی برای بازنمایی دانش مبتنی بر گراف است که از گره‌ها و یال‌ها تشکیل شده است به گونه‌ای که گره‌ها بیانگر اشیا و یال‌ها نشانگر ارتباط بین اشیا است.



شکل 5-11: شبکه معنا برای پرند

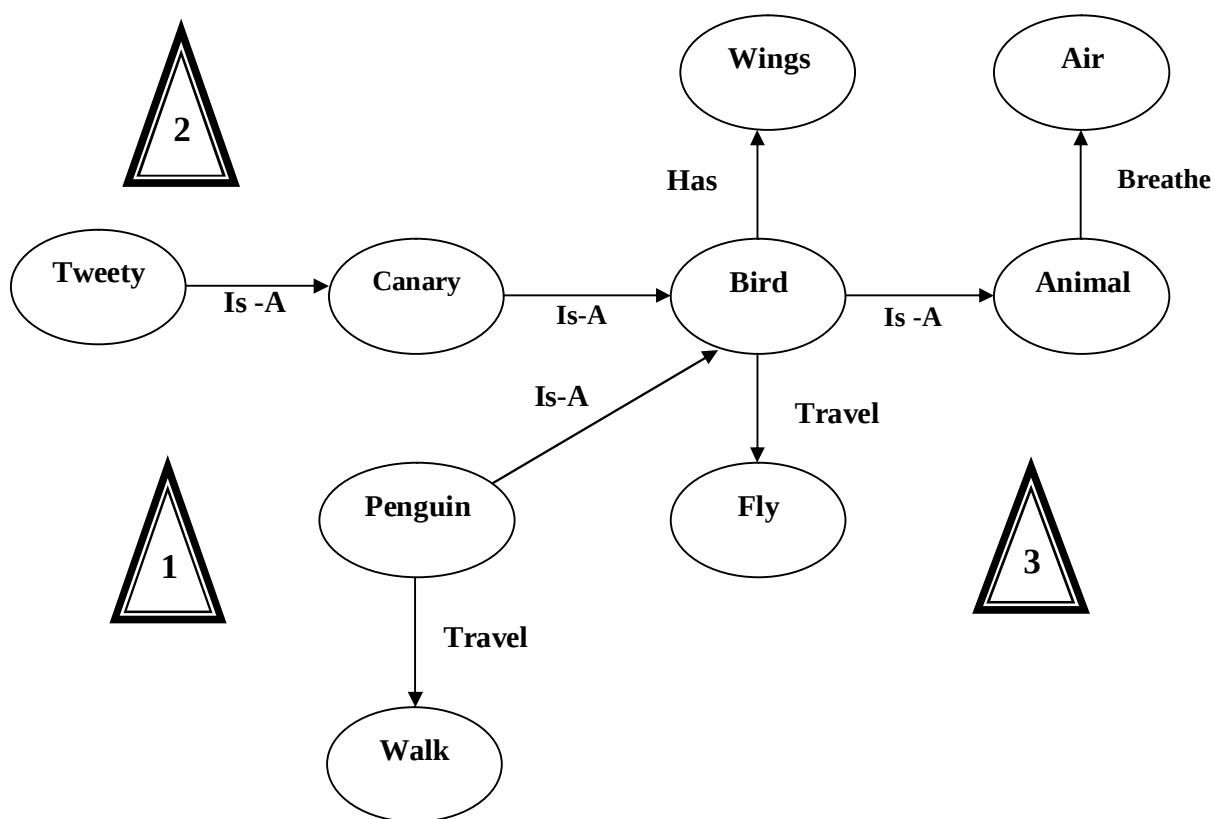
این مثال در مورد پرنده است. گراف دارای سه صفت داشتن (Has)، جابجا شدن (Travel) و نوع پرنده (Is-A) است. نکته قابل توجه در مورد شبکه معنا آن است که جهت در گراف شبکه‌های معنا مهم است:

- پرنده بال دارد
- پرنده با پرواز جابه‌جا می‌شود (پرواز می‌کند)
- قناری نوعی پرنده است

شبکه معنا می‌تواند با استفاده از سه روش گسترش پیدا کند:

- 1- مفاهیم مشابه
- 2- خاص‌تر شدن
- 3- عام‌تر شدن

مثال



شکل 5-12: گسترش شبکه معنا

- در بخش (1) مفاهیم مشابه، نمایش داده شده است، یعنی قناری و پنگوئن هر دو پرنده هستند.
- در بخش (2) خاص‌تر نمودن مفاهیم نمایش داده شده قناری یک نوع پرنده است و Tweety یک نمونه خاص از قناری‌هاست.
- در بخش (3) عام‌تر شدن مفاهیم نمایش داده شده است پرنده‌گان گونه‌ای از حیوانات هستند.

وراثت در شبکه‌های معنا (Inheritance)

یک ویژگی بسیار مهم در شبکه‌های معنا، وراثت است. وراثت بدین معناست که مفهوم یا خاصیتی از گره‌ای به ارث برده شود. این ویژگی در شبکه‌های معنا به وسیله $IS-A$ نمایش داده می‌شود. در مثال بالا تمام ویژگی‌های Animal را Bird دارد و Animal ها موجوداتی هوایی هستند.

چند نکته در مورد شبکه‌های معنا

- وراثت باعث کاهش حجم پایگاه دانش می‌شود و سبب می‌شود هر مفهومی چندین بار تکرار نشود.
- در مثال بالا مشاهده می‌شود سطح و راستای پنگوئن و قناری در گراف یکی است و هر دو با یک یال به پرنده وصل شده است و این نمایانگر این است که بسیاری از مفاهیم در پنگوئن و قناری شبیه هستند.
- در پاسخگویی به شبکه‌های معنا جستجویی روی گراف صورت می‌گیرد تا پاسخ یافته شود.
- مشکل اصلی شبکه‌های معنا استاندارد نبودن روش بازنمایی آن است که ممکن است سلیقه‌ای عمل شود.
- نمونه تکمیل شده شبکه‌های معنا، همان نمودارهای کلاس در UML خواهد بود.
- روش شبکه‌های معنا جزء روش‌های توصیفی است، در نتیجه زبان‌های توصیفی همچون Prolog, Lisp به خوبی می‌تواند شبکه معنا را کد کند.

مدیریت استثناها در شبکه‌های معنا (Exception Handling)

مثال: پنگوئن یک پرنده است ولی پرواز نمی‌کند که این قابلیت توسط Exception Handling تصحیح می‌شود. این مفهوم مشابه مفهوم Override در زبان‌های شی گرا است.

Override the Inherited Information

4- فریم یا چارچوب (Frame)

اگر به شبکه‌های معنا، مفهوم رویه یا Procedure اضافه شود، فریم یا چارچوب شکل می‌گیرد.

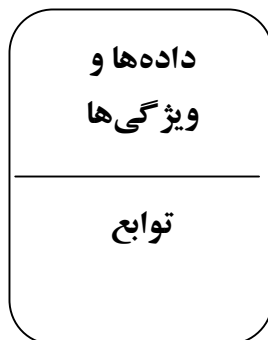
Semantic Network + Procedure = Frame

نکته: روش بازنمایی فریم هم توصیفی است هم رویه‌ای.

مفهوم فریم، مشابه مفهوم Object (شیء) است.

Object = Data (Properties) + Procedure

Object



فریم: فریم ساختمان داده‌ای برای بازنمایی دانش کلیشه‌ای مشابه مفهوم شیء است.

نکته: شیء یا Object در مهندسی نرم افزار، تقریباً معادل مفهوم فریم در سیستم خبره است.

نکته: تفاوت فریم با شبکه‌های معنا: فریم‌ها علاوه بر مفاهیم، دارای پروسیجرها هم هستند.

CRC: خیلی از مفاهیم شی، توسط فرم‌ها و کارت‌ها نمایش می‌دهند که به این کارت‌ها CRC گویند. مثال زیر (شکل 5-13) بیانگر کارتی در مورد دانشجو است.

Report Card

Student Name

Address

Course

Grade

Chemistry	
Math	
English	

شکل 5-13: نمایش کارت یا فرم

مثال بالا کارنامه‌ای است که در آن نام و آدرس دانشجو و نام دروس ترم جاری و نمرات آنها را نمایش می‌دهد.

ساختار کلی چارچوب: در شکل 5-14، ساختار کلی فریم نمایش داده شده است.

Frame Name

Object 1

Class

Object 2

Properties

Property	Value1
Property	Value2
***	***
***	***

شکل 5-14: ساختار کلی فریم

در یک کارت CRC می‌بایست نام فریم، کلاس، ویژگی‌ها و مقادیر آن ویژگی‌ها آورده شود. فیلد class یک فیلد اختیاری در چارچوب است که مقدار آن object2 نام قاب دیگر مرتبط با object1 است. این رابطه معمولاً از نوع Is-A است.

Object1 Is-A Object2

Object1 اطلاعاتی از Object2 به ارث می‌برد.

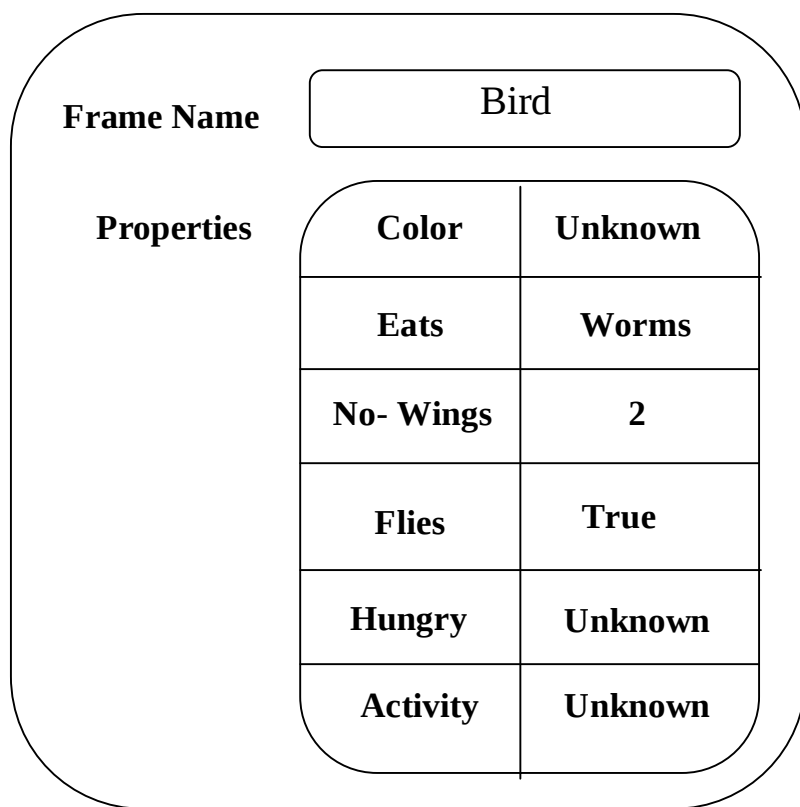
چارچوب کلاس (Class Frame)

به طور مثال CRC شکل 5-15، یک فریم در مورد یک پرنده است. پس نام فریم پرنده است و ویژگی‌های آن رنگ، نوع خوراک، تعداد بال‌ها، قابلیت پرواز، گرسنگی و فعالیت پرنده است.

در این نوع مسائل مشاهده می‌شود که ویژگی‌ها دو حالت پویا و غیر پویا (استاتیک و دینامیک) دارند

0 ایستا: رنگ، تعداد بال

0 پویا: گرسنگی و فعالیت



شکل 5-15: فریم پرنده

نکته: چون روش فریم، گسترش یافته روش شبکه معناست، ویژگی وراثت را دارد.

رفتارهای ارثی در چارچوب (Inheriting Behavior)

در کنار ارث‌بری اطلاعات از کلاس، یک نمونه می‌تواند رفتارهای مهم را نیز به ارث ببرد. برای این منظور در قاب کلاس یک پروسیجر یا متد، اعمالی که بایستی قاب انجام دهد تعریف می‌شود مثلاً متدی که بگوید اگر پرنده گرسنه باشد چه کاری بایستی انجام شود.

وجوه (Facets)

Facet کنترلی روی مقادیر ویژگی‌هاست. به بیانی می‌توان گفت Facet نوعی محدودیت و قیدگذاری است. به طور مثال می‌توان مقدار عددی را به محدوده خاص محدود نمود، یا تنها قادر به پذیرفتن مقادیری خاص باشد.

Facet به دو نوع تقسیم می‌شود:

1. IF Needed در صورت نیاز

2. IF change در صورت تغییر

اگر Facet ما برای دستور دادن، به این منظور که یک ویژگی چگونه مقداری بگیرد IF Needed است و اگر به منظور این باشد که عملیات مشخصی را در صورت تغییر مقدار تعیین نماید، یعنی تعیین کند چه کاری انجام دهد اگر مقدار آن تغییر یابد IF Change نامیده می‌شود.

IF Needed

```
IF Tweety has less than two wings  
THEN Tweety can't fly
```

اگر Tweety کمتر از دو بال داشته باشد قادر به پرواز نیست. به صورت Default در نظر گرفته شده است هر پرنده ای قادر به پرواز است حالا اگر در نمونه ای مشاهده شد که کمتر از دو بال دارد پس آن مقدار Can not می‌شود. در IF needed فقط مقداری عوض می‌شود به طور مثال اینجا Fly به Not Fly تبدیل شد.

IF Change

```
IF self : Hungry = True  
THEN self : Activity =Eating #self =Eats
```

اگر ویژگی گرسنگی مثبت باشد، فعالیت صورت گرفته، خوردن خواهد بود. در IF Change با توجه به ویژگی خاص عمل خاصی صورت خواهد گرفت یعنی به ازای مثبت یا منفی شدن ویژگی خاصی عمل خاصی صورت خواهد گرفت. Self های قبل از ویژگی‌ها مشابه کلمه THIS در برنامه‌نویسی شیء گراست.

5- منطق (Logic)

قدیمی‌ترین روش بازنمایی دانش در کامپیوتر، منطق است. اما طراحان حرفه‌ای از این روش به ندرت استفاده می‌کنند. روش منطق، پایه و اساس بازنمایی دانش است. به طور مثال Prolog زبانی برای بازنمایی منطق است.

دو نوع منطق اصلی

1. منطق گزاره‌ای (مرتبه صفر)

2. منطق مسندی (مرتبه یک)

منطق گزاره‌ای (مرتبه صفر)

در منطق گزاره‌ای، هر گزاره با یک نماد یا سمبل بازنمایی می‌شود و با بررسی جدول‌های درستی به استنتاج می‌پردازیم.

A = The car will start

در این مثال روشن شدن ماشین با نماد A بازنمایی شده است.

در منطق حرکت پذیر منظره‌ای نمی‌تواند وجود داشته باشد. (وینستون)

برای استنتاج از جداول درستی استفاده می‌شود. به طور مثال جدول درستی برای عمل عطف $A \wedge B$ به صورت زیر خواهد بود.

A	B	$A \wedge B$
F	F	F
F	T	F
T	F	F
T	T	T

جدول درستی عمل فصل $A \vee B$ به صورت زیر خواهد بود.

A	B	$A \vee B$
F	F	F
F	T	T
T	F	T
T	T	T

و برای هم ارزی داریم $A \equiv B$:

A	B	$A \equiv B$
F	F	T
F	T	F
T	F	F
T	T	T

و برای عملگر NOT داریم:

A	NOT A
F	T
T	F

و برای عملگر IMPLIES یا عملگر آنگاه داریم:

$$C = A \rightarrow B \equiv \sim A \vee B$$

A	B	C
F	F	T
F	T	T
T	F	F
T	T	T

منطق مسندی (منطق مرتبه یک)

حساب مسندی که گسترش یافته منطق گزاره‌ای است، بازنمایی دقیق‌تری از دانش را امکان‌پذیر می‌سازد. این روش با استفاده از متغیرها و توابع، پردازش دانش را بهبود می‌بخشد. در منطق مسندی از نماد یا سمبل، ثابت، متغیر، تابع و مسند استفاده می‌شود.

برای استفاده از Constant آنها را با حروف کوچک نمایش می‌دهیم مثل اسم افراد.

نمونه‌ای از مسند:

like(john,mary)

Predicate رابطه بین مفاهیم است در این جا like مسند یا Predicate است و john, mary پارامترهای Predicate است. همانطور که مشاهده می‌کنید john, mary هر دو ثابت هستند زیرا با حروف کوچک شروع شده‌اند.

متغیرها را با حروف بزرگ نمایش می‌دهیم:

likes(X,Y)

در این مثال like مسند است و پارامترهای آن متغیر هستند زیرا X,Y را با حروف بزرگ نمایش داده شده است.

نکته: خروجی تابع یک Object است در حالی که خروجی Predicate درست یا نادرست (True یا False) است.

father(jack) = bob
mother(judy) = kathy

تمام عملگرهای موجود در منطق گزاره‌ای قابل استفاده در منطق مسندی هم هست. به طور مثال عملگرهایی نظیر آنگاه، عطف و فصل و ... در منطق مسندی قابل استفاده هستند. با ترکیب Predicate ها با استفاده از عملگرها قادر خواهیم بود دانش‌های پیچیده‌تر را بازنمایی کنیم.

$\text{likes}(X,Y) \wedge \text{likes}(Z,Y) \text{ Implies } \text{Not like}(X,Z)$
OR
 $\sim \text{likes}(X,Z) \rightarrow \text{likes}(X,Y) \wedge \text{likes}(Z,Y)$

اگر X,Y یکدیگر را دوست داشته باشند و Z,Y نیز یکدیگر را دوست داشته باشند نمی‌توان نتیجه گرفت که X,Z نیز یکدیگر را دوست دارند.

مفهوم سور وجودی و سور عمومی

وقتی سور عمومی به کار می‌بریم که موردی برای تمامی موارد درست باشد یا بخواهیم وجود آن را در تمامی موارد بررسی کنیم. سور وجودی زمانی به کار می‌رود که حداقل یک مورد را بخواهیم بررسی کنیم.
همه mary را دوست دارند.

$\forall X \text{ likes}(X, \text{mary})$

حداقل یک نفر وجود دارد که mary را دوست داشته باشد.

$\exists \text{ likes}(X, \text{mary})$

نکته: در Prolog تنها سور عمومی وجود دارد و برای مفهوم سور وجودی نیز با ایجاد یک سری محدودیت به سور عمومی تبدیل می‌شود.

نکته: تبدیل زبان گفتاری (محواره‌ای) به زبان منطق مرتبه یک کار ساده‌ای نیست زیرا زبان گفتاری دارای مرتبه Π و منطق مسندی مرتبه یک است.

Modus Ponens

تا کنون بررسی شد که Modus Ponens اطلاعات جدید را از داده مساله اولیه ایجاد می‌کند. این روش گزینه مناسبی در کاربردهایی است که آموختن و ایجاد اطلاعات از اطلاعات اهمیت دارد با این حال در کاربردهایی نیاز داریم که اطلاعاتی مشخص برای اثبات هدفی جمع‌آوری شود.
مفهوم Modus Ponens همان قیاس است.

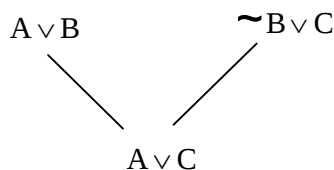
IF A is True
AND $A \rightarrow B$ is True
THEN B is True

اگر A درست باشد و $A \rightarrow B$ نیز درست باشد در نتیجه B هم درست است.

منطق، تورا از انب برب می‌برد. در حالی که تحیل تورا به بهر جامی برد. (آلبرت اینشتین)

اثبات (Resolution)

Resolution یک استراتژی استنتاج است که در سیستم‌های منطقی برای تعیین درستی بیانیه یا حکم (Assertion) استفاده می‌شود. این روش تلاش می‌کند تا تئوری یا هدفی مانند گزاره P را با استفاده از مجموعه‌ای از اصول مرتبط با مساله اثبات نماید. در واقع سعی می‌کند در عمل ثابت کند که $\sim P$ نمی‌تواند درست باشد از این رو اثبات با نقیض یا همان برهان خلف نامیده می‌شود. (proof by Refutation). در روند اثبات با این روش، عبارت‌های جدیدی ایجاد می‌شود که Resolvent نامیده می‌شود.



شکل 5-16: روند رزولیشن بر روی یک مثال

در این مثال $A \vee C$ یک Resolvent است. Resolvent بدست آمده به اصول موجود اضافه می‌شود و این روند ادامه می‌یابد و...

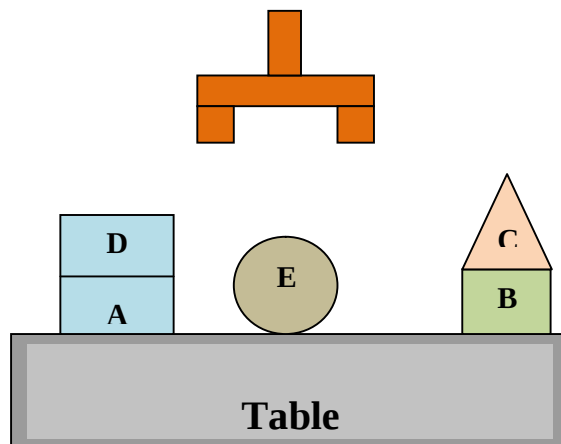
پرسش: روند ایجاد Resolvent ها تا چه زمانی، ادامه می‌یابد؟ 

تا زمانی که در عبارات تناقض وجود داشته باشد. تناقض زمانی رخ می‌دهد که دو اصل از نظر منطقی در تضاد با یکدیگر باشند. $\sim B$ و B در مثال بالا، روش Resolution این است که موارد مختلف را گرفته و آنها را ساده نموده تا نتیجه نهایی بدست آید.

مقایسه استنتاج Modus Ponens و Resolution

در M.P داده‌ها از طریق اطلاعات اولیه و فرم است آنگاه جستجو روی این قالب بدست می‌آید. اگر مساله‌ای الزامات شرطی زیادی داشته باشد این فرآیند باید روی تمام اطلاعات ممکن انجام شود در حالیکه Resolution تمرکز روی هدفی که در جهت اثبات آن است، دارد و در نتیجه تنها الزاماتی که مرتبط با این هدف یا فرض است، در نظر می‌گیرد.

نکته: یکی از کاربردهای منطق در بازنمایی محیط حل مساله برای یک روبات جهت برنامه‌ریزی (Planning) روبات‌هاست. به طور مثال در شکل 5-17، یک روبات، نشان داده شده است.



شکل 5-17: طرح ریزی در جابجایی بلوک‌ها

قوانین زیر در مورد روبات شکل 5-17 در نظر گرفته شده است:

- 1) cube(a) cube(b) cube(d) pyramid(c) sphere(e) hand(hand) table(table)
- 2) on(a, table) on(b, table) on(d, a) on(c, b) on(e, table)
- 3) holding(hand, nothing)

این قوانین محیط حل مساله را تشریح می نماید به بیان دیگر یک فریم وضعیت هستند. Cube نماد مکعب ها، pyramid نماد هرم، sphere نماد کره، hand دستگیره یا بازوی ربات، table میز است. تابع On برای قرار گرفتن شی ای روی شیء دیگر استفاده می شود و نشان می دهد پارامتر اول روی پارامتر دوم قرار دارد. تابع holding برای نمایش شیء ای که توسط بازوی روبات برداشته شده است به کار می رود. در اینجا بدین معنی است که ربات چیزی را برداشته است. در این مثال با تغییر موقعیت هر آیتم توابع و مقادیر تغییر کرده و می تواند به برنامه ریزی یا Planning پرداخته و وضعیت را Trace کند. حال مفهومی تحت عنوان putOn(b,a) تعریف می شود که مفهوم آن این است که b را روی a قرار بده.

$hand_holding(b) \wedge clear(a) \rightarrow putOn(b,a)$

این قانون بدین معنی است که اگر بازوی روبات، b را برداشته باشد و روی a خالی است، می توان b را روی a قرار داد.

تمرین ها

تمرین 5-1*: چگونه منطق و زبان های مبتنی بر منطق می توانند در برنامه ریزی روبات ها به طور تئوری و عملی استفاده شوند؟

تمرین 5-2: تفاوت و شباهت روش منطق و روش قانون در بازنمایی دانش چیست؟ با مثال این تفاوت را توضیح دهید.

تمرین 5-3: آیا با روش رزولیشن و با توجه به مجموعه قوانین موجود در پایگاه دانش، می توان درستی E_3 را اثبات کرد؟

$E_1 \rightarrow E_2$

$E_2 \rightarrow E_3$

E_1

تمرین 5-4*: روش SLD-Resolution در منطق چیست؟ چه تفاوتی با Resolution معمولی دارد؟

تمرین 5-5*: بررسی کنید امروزه علاوه بر روش های معرفی شده برای بازنمایی دانش، چه روش های دیگری وجود دارد.

تمرین 5-6: مساله قاب (Frame Problem) در برنامه ریزی چیست؟ چه راهکار یا راهکارهایی برای حل آن پیشنهاد شده است؟

Fear is the main source of superstition. To conquer fear is the beginning of wisdom.

ترس اصلی ترین ریشه خرافات است. غلبه یافتن بر ترس، آغاز خرد است. (برتراند راسل)