جزوه

سیستم های خبره

Expert Systems

مدرس:

کمال میرزایی

ویرایش چهارم

به نام خدا

پیشگفتار

سیستم‌های خبره، برنامه‌های کامپیوتری هستند که نحوه تفکر یک متخصص در یک زمینه خاص را شبیه‌سازی می‌کنند. این نرم‌افزارها، الگوهای منطقی که یک متخصص بر اساس آن‌ها تصمیم‌گیری می‌کند، شناسایی کرده و سپس بر اساس آن الگوها، مانند انسان‌ها تصمیم‌گیری می‌کنند.

در واقع، بیشتر دستاوردهای هوش مصنوعی در زمینه تصمیم‌گیری و حل مسئله بوده است که اصلی‌ترین موضوع سیستم‌های خبره را شامل می‌شود. به عبارت دیگر، به آن نوع از برنامه‌های هوش مصنوعی که به سطحی از خبرگی می‌رسند که می‌توانند به جای یک متخصص در یک زمینه خاص تصمیم‌گیری کنند، سیستم‌های خبره، گفته‌می‌شود. این سیستم‌ها، دارای پایگاه دانشی هستند که انباشته از اطلاعات و دانشی است که انسان‌ها، هنگام تصمیم‌گیری درباره یک موضوع خاص، براساس آن‌ها تصمیم می‌گیرند.

در این جزوه، تلاش‌شده تا به تمامی سرفصل‌های درس سیستم های خبره، اشاره‌شود. بنابراین به هیچ وجه جایگزین مراجع اصلی این درس نخواهد بود. در گردآوری آن، بیشتر از کتاب معروف سیستم‌های خبره **جان دارکین** و بخشی هم از کتاب هوش‌مصنوعی **راسل**، استفاده‌شده‌است:

[1] J. Durkin, "**Expert Systems, Design and Development",** 1994 .
[2] S. Russell and P. Norvig, "**Artificial Intelligence; A Modern Approach**", 1995, 2003.

در خواندن این جزوه درسی، توجه‌داشته‌باشید که مرجع و ملاک اصلی، منابع و مراجع درس سیستم‌های خبره به زبان اصلی است، بنابراین اگرشما، احیاناً مطالب این جزوه را در مواردی متناقض با مطالب این منابع و مراجع یافتید، ملاک را در صحت بر مبنای کتب زبان اصلی این درس قراردهید.

در کنار این دو کتاب، می‌توانید به کتب فارسی مرتبط با سیستم‌های خبره هم مراجعه‌کنید. در پایان، در صورت تمایل، نظرات و پیشنهادات خود را در مورد این جزوه، به آدرس پست الکترونیکی، ارسال نمائید. امید است که مجموعه گردآوری‌شده، یاریگر شما در یادگیری بهتر و بیشتر مطالب درس، باشد.

تهیه و تنظیم: کمال میرزایی

mirzaie_kamal@yahoo.com

خبرگی به چه معناست ؟

خبرگی(Expertise) دانشی است تخصصی که برای رسیدن به آن، نیاز به مطالعه مفاهیم تخصصی یا دوره‌های ویژه، وجود دارد.

سیستم‌های خبره

سیستم‌های خبره یکی از زیر شاخه‌های هوش مصنوعی می‌باشد و یک سیستم خبره به برنامه کامپیوتری گفته‌می‌شود که دارای خبرگی در حوزه خاصی می‌باشد و می‌تواند در آن حوزه برای تصمیم‌گیری یا کمک به خبره جهت تصمیم‌گیری بکاررود .

نکته: سیستم‌های خبره برای حل مسائلی بکار می‌روند که :

۱. الگوریتم خاصی برای حل آن مسائل وجود ندارد.
۲. دانش صریح برای حل آن مسئله وجود دارد.

بنابراین اگر سیستمی با استفاده از روش‌های علم آمار اقدام به پیش‌بینی دمای هوای فردا کند، در حوزه سیستم‌های خبره قرار نمی‌گیرد . اما اگر سیستمی با استفاده از این قاعده که "در این فصل از سال دمای هوا معمولا ثابت می‌باشد" و این واقعیت که " دمای امروز ۲۵ درجه سانتی‌گراد می‌باشد " به این نتیجه دست یابد که " دمای فردا حدود ۲۵ درجه خواهد بود " در حوزه سیستم‌های خبره قرار خواهد گرفت .

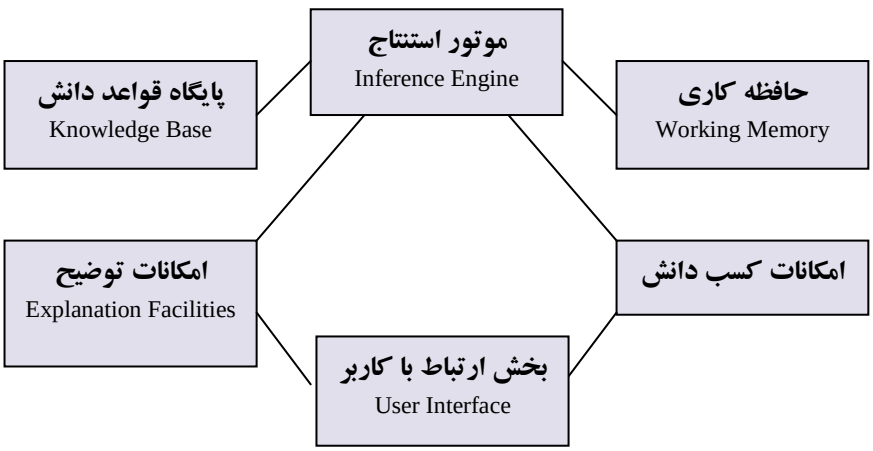
از سیستم خبره نباید انتظار داشت که نتیجه بهتر از نتیجه یک خبره را بیابد . سیستم خبره تنها می‌تواند همسطح یک خبره اقدام به نتیجه‌گیری نماید. سیسنم‌های خبره همیشه به جواب نمی‌رسند .

با توجه به این که علوم مختلفی وجود دارد در نتیجه خبرگی در شاخه های علمی متفاوت مطرح است . یک فرد خبره یا Expert فردی است که در زمینه‌ای خاص مهارت دارد به طور مثال یک پزشک ، یک مکانیک ویک مهندس افرادی خبره هستند . این مسئله بیانگر این است که دامنه کاربرد سیستم‌های خبره، گسترده است ومی‌توان برای هر زمینه کاری یک سیستم خبره طراحی نمود .

بیان خبرگی در قالب دانش یا بازنمایی دانش

برای این که این خبرگی، یک سیستم خبره تشکیل دهد لازم است این خبرگی درقالب دانش بیان شود. بازنمایی دانش تکنیکی است برای بیان خبرگی در قالب دانش . بازنمایی دانش برای ایجاد و سازماندهی دانش یک فرد خبره در یک سیستم خبره استفاده می شود .

اجزای اصلی سیستم خبره



شکل ۱-۱: اجزای اصلی یک سیستم خبره

پایگاه دانش: یکی از مولفه های مهم سیستم های خبره پایگاه دانش یا مخزن دانش است . محلی است که دانش خبره به صورت کدگذاری شده و قابل فهم برای سیستم ذخیره‌می‌شود . **پایگاه قواعد دانش،** محلی است که در آن بازنمایی دانش صورت می گیرد . بازنمایی دانش بعد از اتمام مراحل به پایگاه قواعد دانش تبدیل می شود.

به کسی که دانش خبره را کدکرده و وارد پایگاه دانش می‌کند مهندس دانش knowledge engineer گفته می‌شود
بطورکلی دانش به صورت عبارات شرطی و قواعد در پایگاه دانش ذخیره می‌گردد." اگر چراغ قرمز است آنگاه متوقف شو"
هرگاه این واقعیت وجود داشته باشد که "چراغ قرمز است" ، آنگاه این واقعیت با الگوی " چراغ قرمز است " منتطبق می‌شود در این صورت این قاعده برآورده می‌شود و دستور متوقف شو اجرا می شود .

موتور استنتاج: یعنی از دانش موجود استفاده و دانش را برای حل مسئله به هم ربط دهیم . موتور استنتاج با استفاده از قواعد منطق و دانش موجود در پایگاه دانش و حقایق حافظه کاری اقدام به انجام کار خاصی می‌نماید . این عمل یا بصورت افزودن حقایق جدیدی به پایگاه دانش می‌باشد یا به صورت نتیجه‌ایی برای اعلام به کاربر یا انجام کار خاصی می‌باشد .

حافظه کاری: حافظه ای برای ذخیره پا سخ سوال های مربوط به سیستم می باشد .

امکانات کسب دانش: راهکارهایی برای ایجاد و اضافه نمودن دانش به سیستم. امکاناتی است که اگر بخواهیم دانشی به سیستم اضافه کنیم باید یک بار از این مرحله عبور کنیم اگر این دانش قبلاً در سیستم وجود نداشته باشد به موتور استنتاج می رود روی آن پالایشی صورت می گیرد و سپس در پایگاه دانش قرار می گیرد.

امکانات توضیح: برای نشان دادن مراحل نتیجه‌گیری سیستم خبره برای یک مساله خاص با واقعیت خاص به کاربر به زبان قابل فهم برای کاربر بکار می‌رود . این امکانات این فایده را دارد که کاربر با دیدن مراحل استنتاج اطمینان بیشتری به تصمیم گرفته‌شده توسط سیستم خواهد داشت و خبره‌ایی که دانش او وارد پایگاه دانش شده است اطمینان حاصل خواهد کرد که دانش او به صورت صحیح وارد شده است.
اگر در ارتباط با سیستم سوال و جوابهایی مطرح شود وسیستم به ما یک سری راهکار پیشنهاد کند و توضیحی در زمینه این که چرا چنین سوالی پرسیده‌می‌شود ؟ (Why) و چگونه به این نتیجه رسیده‌ایم ؟(How) را در ناحیه ای ذخیره نماییم ، امکانات توضیح را تشکیل می دهد.

بخش ارتباط با کاربر: مربوط به بخشی است که به طور مستقیم با کاربر در ارتباط است .

کاربردهای سیستم خبره

۱- جایگزینی برای فرد خبره (سیستم اینترنتی در زمینه مشاوره محصولات یک شرکت)
- تداوم کار در صورت عدم دسترسی به فرد خبره
- کاهش هزینه
- احساساتی نبودن سیستم و خستگی ناپذیری آن

۲- کمک و دستیار (برنامه های Pspice یا Autocad یا MS Project برنامه هایی هستند که دانشی برای انجام عمیاتی برای کمک به افرادی خاص را دارند)

انواع مسائل قابل حل با سیستم خبره

۱- **مسائل کنترلی**(Control) **:** مثل کنترل وضعیت یک بیمار در بیمارستان به طوری که موقعیت های ویژه و حساس مدیریت شوند . در این گونه مسائل داده هایی از محیط مسئله گرفته می شود و از این داده ها برای فهم حالت سیستم یا پیش بینی حالت آینده سیستم استفاده می شود . به نوعی داده های جمع آوری شده تفسیر می شود و متناسب با این تفسیر اعمال مورد نیاز تعیین و اجرا می شوند .

۲- **طراحی** (Design): در سیستم های طراحی خبره با مجموعه ای از محدودیت ها وشرایط مسئله مورد نظر را پیکر بندی می کند مانند برنامه‌های Pspice یا Autocad

۳- **تشخیص** (Diagnosis) : تشخیص عیب خودرو ، تشخیص بیماری

۴- **آموزش** (Learning) : سیستم خبره برای آموزش ریاضی

۵- **تفسیر** (Interpertation) : داده‌های جمع‌آوری‌شده به صورت نمادین در قالب متن و نمودارهایی نمایش داده شود مانند نوار قلب و یا تفسیر حالات فرد از روی چهره

۶- **پایش** (Monitoring) : سیستم های پایش ، اطلاعات دریافتی از محیط مسئله را با وضعیت های حیاتی و حساس مقایسه می‌کند تا در صورت بروز مشکل ، تشخیص داده شده و اقدامات لازم انجام شود

۷- **برنامه‌ریزی**(Planning) : عبارتست از انجام یک سری عملیات برای رسیدن به هدفی خاص تحت شرایط مسئله. یعنی گام هایی را در نظر بگیریم که در صورت طی گام ها به هدف مورد نظر برسیم.

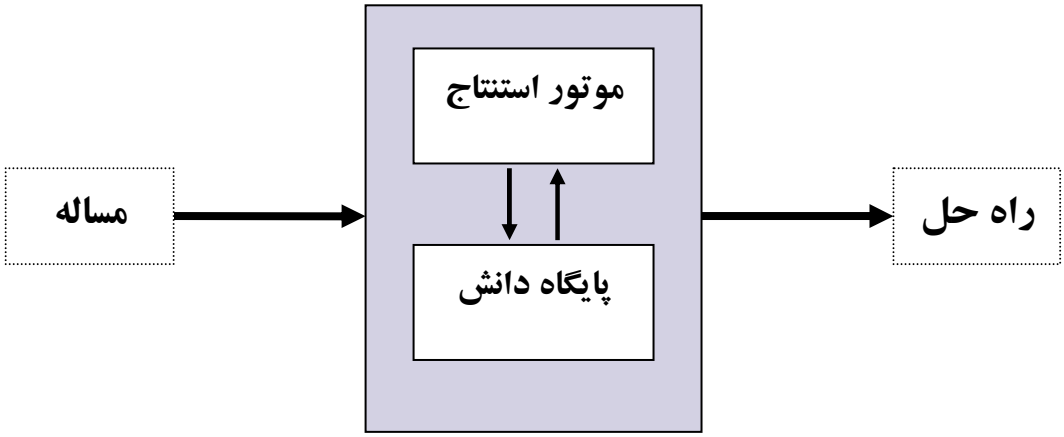
۸- **شبیه‌سازی**(Simulation) : سیستمی به گونه ای مدل سازی شود که رفتار سیستم مدل سازی شده تا حدودی بیانگر رفتار واقعی سیستم در دنیای واقعی باشد به طور مثال بررسی اثرات زلزله روی یک محصول جدید در یک محیط مجازی

۹- **پیش‌بینی**(predication): مثلا برنامه ای به نام pant در سال ۱۹۸۳ که تعیین خسارت غلات به وسیله افات را پیش بینی می‌کرد.

۱۰- **تجویز**(Prescription)**:** یکسری توصیه به افراد می‌کند و کاربرد اصلی ان در شاخه پزشکی می باشد .مثلا blue box برنامه ای از این نوع برای افسردگی است.

پرسش: علاوه بر مسائلی که در بالا آمده، آیا مسائل دیگری می‌توانید مثال بزنید که با سیستم های خبره، قابل حل باشد؟

تمرین: دو مولفه اصلی سیستم خبره ، پایگاه دانش و موتور استنتاج می باشد. برای افراد خبره نظیر پزشکان، مهندسان، تعمیرکاران، محتوای این دو مولفه چه خواهد بود.؟ چگونگی حل مساله توسط آنها چگونه است؟ با مثال یا مثال‌هایی چگونگی حل مساله با توجه به این دو مولفه، توضیح دهید.



شکل ۱-۲: ساختار کلی یک سامانه خبره

پیاده‌سازی سیستم های خبره

برای پیاده‌سازی سیستم‌های خبره

- ابزار (Tools)
- زبان (Language)
- پوسته (Shell)

وجود دارد.

ابزار

محیط برنامه‌نویسی است که علاوه بر قابلیت‌های یک محیط برنامه‌نویسی وظایف دیگری نظیر مدیریت فایلها ، نمایش گرافیکی و تبدیل کدها به زبان دیگر را فراهم‌می‌کند برخی از این ابزارها به گونه‌ای طراحی شده‌اند که کاربر لازم نیست هیچ کدی را واردکند و می‌تواند روش حل مسئله را به صورت جداول یا نمودارهایی وارد نماید و پس از آن ابزار مزبور، کد را ایجاد نماید مانند ابزاری به نام KVision که دانش را به صورت دیداری ایجادمی‌کند .

زبان

امکانات لازم برای کدنویسی فراهم می‌کنند.

مثال: زبان‌هایی مانند Lisp, Prolog ,C برای ایجاد سیستم خبره استفاده‌می‌شوند .

پوسته

معمولا به سیستم خبره‌ای فاقد دانش اشاره دارد و کاربر برای ایجاد سیستم خبره فقط باید پایگاه دانش آن را پر نماید .

به طور مثال سیستم خبره پزشکی MYCIN، سیستم خبره‌ای برای تشخیص بیماری مننژیت می‌باشد. حال اگر این سیستم فاقد دانش تشخیص بیماری خاص باشد به یک Shell تبدیل‌می‌شود که آن را Empty MYCIN یا E-MYCIN می‌نامند که اگر دانش بیماری دیگری در آن قراردادهشود سیستمی برای تشخیص بیماری دیگری خواهد بود .

الگو های برنامه‌نویسی

۱- **زبان‌های رویه‌ای**(Procedural)

در این زبان‌ها باید روند و چگونگی کار برای سیستم بیان شود تا سیستم ساخته شود.

زبان‌هایی همچون C, Pascal زبان‌های رویه‌ای، محسوب‌می‌شوند.

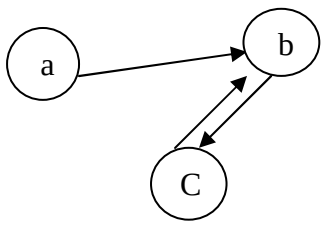
۲- **زبان‌های توصیفی**(Declarative)

در این زبان‌ها تعریف صورت مسئله در سیستم، منجر به حل مسئله می‌گردد. مانند زبان‌های SQL , Prolog , CLIPS , Protoje

پرسش: در سیستم‌های خبره و به طور کلی کاربردهای هوش مصنوعی بهتر است از زبان‌های توصیفی استفاده گردد. چرا؟ با این وجود، آیا امکان پیاده‌سازی سیستم‌های خبره با زبان‌های سطح پایین یا سطح میانی وجود دارد ؟

مثالی برای بیان قابلیت‌های زبان توصیفی

فرض کنید گراف ساده‌ای به صورت زیر داریم



شکل۲-۱: گرافی با سه گره

پرسش گراف بالا در زبان C چگونه کد می‌شود؟

پاسخ: در زبان C می بایست یک ماتریس تعریف شود و به ازای وجود یال بین دو گره در درایه متناظر عدد یک و در صورت فقدان مسیر عدد صفر قرار گیرد.

همین گراف در زبان توصیفی نظیر زبان Prolog ، به صورت زیر تعریف می شود .

```
Arc(a,b)
Arc(b,c)
Arc(c,b)
Path (x,y)  if  arc(x,y)
Path (x,y)  if  arc(x,z) && path(z,y)
```

در این زبان ،تنها کافیست مسیرهای موجود تعریف شوند.

به طور کلی، زبانهای غیر رویه‌ای به دو دسته زیر تقسیم می‌شوند:

۱- توصیفی :

- مبتنی بر منطق مانند Prolog
- مبتنی بر قاعده مانند CLIPS
- مبتنی بر فریم مانند Protoje

۲- غیر توصیفی

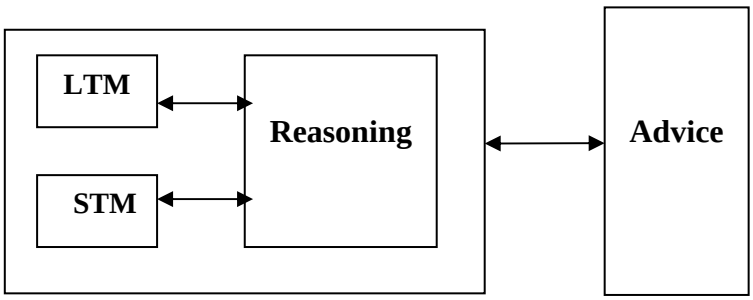
- شبکه‌های عصبی و سیستم‌های استقرایی: این زبان‌ها از یک سری نمونه و جزئیات، به کلیات می‌رسند. به بیان دیگر این زبان‌ها قابلیت تعمیم دارند .

برای کسب اطلاعات بیشتر از انواع **زبان‌های برنامه‌نویسی**، به مباحث درس **طراحی و پیاده‌سازی زبان‌های برنامه‌سازی**(PL) ، مراجعه کنید.

ساختار سیستم های خبره

مؤلفه‌های اصلی سیستم‌های خبره

در شکل۳-۱، ساختار و مولفه‌های اصلی یک انسان خبره، نشان‌داده‌شده‌است . این شکل نمایانگر وجود دو نوع حافظه و یک مولفه Reasoning یا استدلال است.



شکل۳-۱: مولفه‌های اصلی یک انسان خبره

- (Long Term Memory = LTM) یا حافظه‌بلندمدت: دانش‌های مرتبط با حل مسئله؛ حافظه‌ای که در تصمیم‌گیری از آن استفاده‌می‌شود.
- (Short Term Memory = STM) یا حافظه‌کوتاه‌مدت: حافظه‌ای است برای جمع آوری اطلاعات از محیط برای پردازش
- Reasoning یا استدلال : مجموعه روش‌های استفاده از دانش

اگر فردی به فرد خبره مراجعه‌نماید و از او راهنمایی بخواهد ابتدا حقایقی در مورد مسئله، گرفته‌می‌شود به این حقایق case fact گفته‌می‌شود که توسط سوال و جواب‌هایی که بین فرد خبره و شخص مقابل او صورت می گیرد، جمع‌آوری‌می‌شود. در مرحله پایانی فرد خبره بر اساس اطلاعاتی که دارد و LTM و STM یک پیشنهاد (Advice) می‌دهد.

مثال:

به عنوان مثال اگر پزشک را به عنوان خبره در نظر بگیرید:

حافظه کوتاه مدت شرح حال و نتایج آزمایشات و شواهد است و حافظه بلند مدت تجربیات و مطالعات و دانشی که قبلاً کسب نموده‌است و استدلال نتیجه‌ای است که با استفاده از شواهد و سوال و جوابها و دانش قبلی گرفته می‌شود .

مثالی از Knowledge Base

فرد خبره یک مکانیک است:

RULE 1:
IF the car will not start
THEN the problem may be in the electrical system

اگر ماشین روشن نشد ممکن است از سیستم برق باشد

RULE 2:
IF the problem may be in the electrical system
AND the battery voltage is below 10 volts
THEN the fault is a bad battery

اگر مشکل از سیستم برق باشد و ولتاز باتری از ۱۰ ولت کمتر باشد پس مشکل از باتری است

یادآوری: پایگاه دانش بخشی از سیستم خبره است که شامل دانش تخصصی است.

وقتی حافظه‌بلندمدت به حوزه فنی و مهندسی دانش(سیستم خبره) می‌آید تبدیل به مفهوم پایگاه دانش، می‌شود پس معادل حافظه بلند مدت در سیستم خبره همان پایگاه دانش خواهد بود.

معادل حافظه‌کوتاه‌مدت، حافظه‌کاری یا Working Memory است، حافظه کاری بخشی از سیستم خبره است که شامل یافته‌ها و حقایق کشف‌شده در یک جلسه سوال و جواب بین سیستم خبره و کاربر (session) می‌باشد.

در انسان سیستم استدلال (Reasoning) و در سیستم خبره موتور استنتاج (Interence Engine) وجود دارد. به بیان دیگر Reasoning انسان هنوز به طور کامل قابل مدل‌سازی نیست. چرا؟ بخشی از Reasoning که قابل مدل‌سازی است Inference، نامیده‌می‌شود.

موتور استنتاج(Interence Engine)

موتور استنتاج یک پردازنده در سیستم خبره است که حقایق گنجانده شده در حافظه کاری را با دانش گنجانده شده در پایگاه دانش تخصصی موجود در پایگاه دانش برای رسیدن به نتیجه تطبیق‌می‌دهد .

STEP 1:
Expert System : Does the car not start?
USER : True

سیستم خبره : آیا ماشین روشن‌نمی‌شود ؟
کاربر : بله؛ روشن نمی‌شود

User Assert : the car will not start
System Assert : the problem may be in the electrical system

بیانیه کاربر : ماشین روشن نمی‌شود
بیانیه سیستم : ممکن است مشکل از سیستم الکتریکی باشد .

STEP 2 :
Expert System : is the battery voltage below 10 olts ?
USER : True

سیستم خبره : آیا ولتاژ باتری زیر ۱۰ ولت است ؟ کاربر : بله

USER ASSERT : The battery voltage is below 10 volts
SYSTEM ASSERT : The fault is a bad battery

بیانیه کاربر : ولتاژ باتری زیر ۱۰ ولت است
بیانیه سیستم : مشکل از باتری است .

تسهیلات توضیح (Explanation Facility)

سیتم خبره باید قابلیت توضیح یا توجیه پاسخ‌های خود را داشته باشد . دو نوع توضیح مورد نیاز است

۱. Explaining How چگونه به این پاسخ رسیده است ؟
۲. Explaining Why چرا به این جواب رسیده است ؟

این توضیحات جهت اعتبار دهی به پاسخ هاست .

توضیح چگونگی:

EXPERT : The battery is bad
PERSON : How ?
EXPERT : Since your car wont start I assumed there was a problem with the electrical system . once I found the battery voltage was below 10 volts I knew the battery was bad.

فرد خبره: باتری مشکل دارد ؟
شخص: چگونه به این جواب رسیدی ؟
فرد خبره: وقتی دیدم ماشین روشن نمی‌شود حدس‌زدم که سیستم الکتریکی مشکل دارد و هنگامی که دیدم ولتاژ باتری زیر ۱۰ ولت است فهمیدم مشکل از باتری است .

سیستم های خبره در پاسخ به چگونگی، قوانین استفاده شده در استنتاج را بیان می کنند.

توضیح چرایی:

EXPERT : will the car not start
PERSON : why ?
EXPERT : if I know that the car wont start then I usually assume the problem is in the electrical system

فرد خبره: ماشین روشن نمی‌شود ؟
شخص: چرا این سوال را می پرسی ؟
فرد خبره: اگر من بدانم ماشین روشن نخواهد شد معمولاً فرض می کنم مشکل از سیستم برق است .

سیستم‌های خبره درپاسخ به چرایی، قانونی را که در حال بررسی است، توضیح‌می‌دهند.

واسط (Interface)

واسط بخشی است که با کاربر در ارتباط است.

مشخصه‌های سیستم خبره

۱. **جداسازی دانش از کنترل**

یک سطح پایین‌تر این مبحث، در پایگاه داده قابل مشاهده است . در پایگاه داده سعی بر این است که داده‌ها از رویه‌های پیاده‌سازی‌شونده روی داده‌ها، مجزا باشند . در این درس، داده‌ها همان دانش و رویه‌ها همان موتور استنتاج می‌باشد. مزیت این جداسازی این است که تعمیم یافتگی در سیستم، افزایش‌می‌یابد .

۲. **برخورداری از دانش خبره و تخصصی**
۳. **تمرکز بر روی تخصص‌های خاص و ویژه**
۴. **استدلال با نمادها**

Data Processor : برنامه معمولی

Knowledg Processor : برنامه سیستم‌های خبره

۵. **استدلال هیورستیک و تجربی:** استدلالی که بر اثر تجربه، بدست‌می‌آید.
۶. **قابلیت استدلال نادقیق**: یعنی با قوانین احتمالی هم استدلال نماید.

سیستم خبره باید بتواند در محیط‌هایی که اطلاعات نادقیق است(کامل نیست) استدلال کنند . این استدلال می‌تو اند اشتباه باشد چون اطلاعات کامل نیست .مثلا پزشکی را در نظر بگیرید که تجربه دارد و تازه‌کار هم نیست، ولی زمانی که وضعیت بحرانی پیش‌می‌آید باید بتواند با اطلاعات کم ، بهترین تصمیم را بگیرد .

۷. **محدودیت نسبت به مسایل قابل حل:** تنها مسائل قابل حل، توسط سیستم‌های خبره، قابل پیاده‌سازی، می‌باشند .

تا مساله‌ای حل‌نشده باشد، سیستم‌خبره نمی‌تواند به آن پاسخ‌دهد . باید یک فرد خبره‌ای باشد که اطلاعات از او گرفته شده و در سیستم قرارداده‌شود.

۸. **مناسب بودن سیستم خبره از نظر پیچیدگی**: مسایل سیستم خبره نباید خیلی سخت و نه خیلی راحت باشد.
۹. **احتمال اشتباه:** ممکن است سیستم خبره در تعیین راه حل دچار مشکل شود .

☜ **پرسش:** تفاوت Data و Knowledge چیست؟

مهندسی دانش

پرسش: سیستم‌های خبره را با سیستم‌های معمولی، مقایسه کنید؟

در جدول ۴-۱، مقایسه‌ای بین سیستم‌های خبره و سیستم‌های معمولی، صورت گرفته است.

Conventional program	Expert system
Numeric	Symbolic
Algorithmic	Heuristic
Information and control Integrated	Knowledge separate from control
Difficult to modify	Easy to modify
Precise Information	Uncertain Information
Command Interface	Natural dialogue with explanation
Final result given	Recommendation with explanation
Optimal solution	Acceptable solution

جدول۴-۱

- برنامه‌های معمولی، برنامه‌های عددی هستند در حالی که بر نامه‌های سیستم‌های خبره، برنامه‌های نمادین می‌باشند .
- برنامه‌های معمولی، الگوریتمیک هستند در صورتی که برنامه‌های معمولی هیورستیک هستند .
- در برنامه‌های معمولی داده و کنترل یکپارچه شده در حالی که سیستم‌های خبره، دانش از کنترل، جدا می‌شود .
- تغییر در برنامه معمولی سخت اما در سیستم‌های خبره آسان است .
- برای بر نامه‌های معمولی، داده‌های دقیق نیاز است، در حالی که سیستم‌های خبره با داده‌های غیر دقیق نیز کار می‌کنند.
- برنامه‌های معمولی باید به جواب نهایی رسیده شود در حالی که سیستم خبره در انتها ممکن است، تنها به نتیجه‌گیری ختم شود .
- برنامه‌های معمولی دنبال راه حل دقیق و قطعی هستند اما سیستم‌های خبره دنبال جواب قابل قبول می‌باشند.

برنامه‌نویسی در مقایسه با مهندسی دانش (Programming vs. Knowledge Engineering)

برنامه نویسی(فرآیند تولید برنامه) سه گام اصلی دارد :

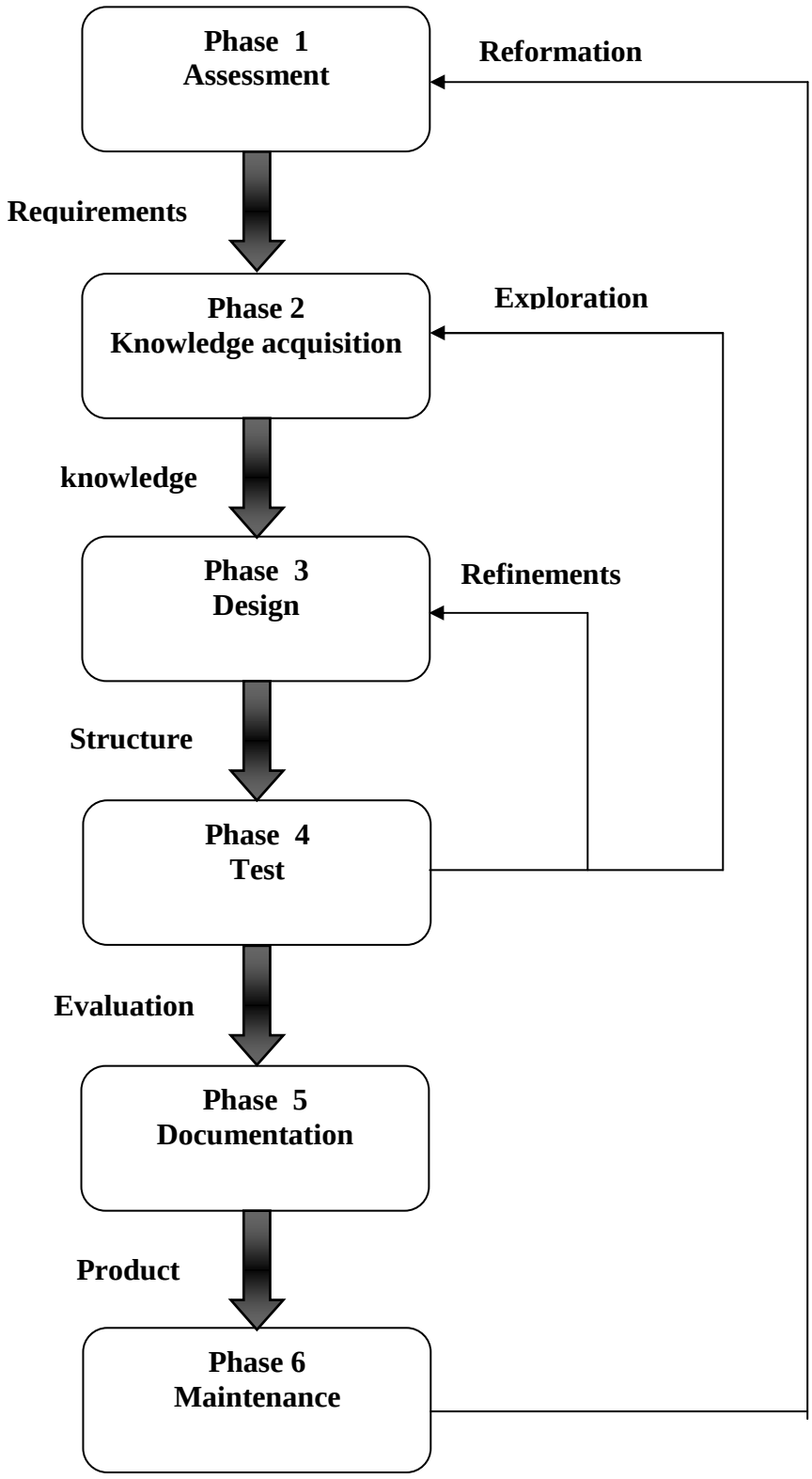
۱. طراحی (Design)
۲. کد نویسی (Code)
۳. اشکال زدایی (Debug)

مهندسی دانش

فرآیند ساخت یک سیستم‌خبره ، مهندسی دانش گفته می‌شود . ایجاد یک سیستم خبره، فرآیندی تکراری است و مراحل مختلف آن در طی پروژه تکرار می شود ، یعنی اینکه طراح قسمتی از سیستم را می سازد و تست می‌کند و پس از آن دوباره سیستم را توسعه و بهبود می بخشد .

فازهای مهندسی دانش

در شکل ۴-۱، شش فاز مهندسی‌دانش، نشان‌داده‌شده‌است.



شکل۴-۱: شش فاز مهندسی‌دانش

فاز ۱ : ارزیابی(Assessment)

مشابه فاز امکان‌سنجی در مهندسی نرم‌افزار است در این فاز مطالعات و بررسی‌هایی انجام می‌شود تا :

۱. عملی بودن مساله مورد نظر تایید شود .
۲. دلایل توجیه کننده برای ایجاد سیستم خبره مورد نظر پیدا شود .
۳. اهداف کلی پروژه تعیین شود .
۴. منابع مورد نیاز تعیین شود .

فاز ۲ : اکتساب دانش(knowledge acquisition)

فرایند اکتساب ، سازماندهی و مطالعه دانش، اکتساب دانش گفته می شود .این فرایند شامل مصاحبه ، گفتگو و تکمیل فرم ها و به طور کلی اکتساب خبرگی، از فرد خبره است . این فاز، فراهم‌کننده دانش مورد نیاز در حل مساله است . خروجی این فاز Knowledge است و دانش استخراج می‌شود و به مرحله بعد می‌رود .

فاز ۳ : طراحی(Design)

در طی این مرحله ، ساختار و سازماندهی کلی دانش سیستم، نظیر روش‌های پردازش دانش و ابزارها و نرم‌افزارها برای نمایش و استدلال دانش تعریف‌می‌شود. ابزارها و روش‌های مناسب انتخاب می‌شود و دانش استخراج شده در مرحله اکتساب دانش در واحدهای مناسب ذخیره می‌شوند . معمولا در فاز طراحی یک نمونه اولیه یا Prototype ، جهت فهم بهتر مساله ایجاد می‌گردد .

فاز ۴ : آزمون(Test)

مرحله تست Feed backهایی به فازهای ۳ و ۲ و دارد . این مساله بدین معنی است که عملیات آزمون در تمامی مراحل پروژه و سرتاسر مراحل ایجاد سیستم اعمال می شود . هدف فاز تست بررسی صحت و اعتبار ساختار کلی سیستم و دانش استخراج شده است . در این مرحله از راهنمایی‌های فرد خبره هم استفاده خواهدشد . خروجی طراحی به تست، Structure می‌باشد یعنی در این مرحله دانش ، دانش ساختار یافته مورد آزمون قرار می‌گیرد.
بعد از گذر از فاز تست و اعمال feed back ها Evaluation می شود و بعد از آن به صورت یک سری مستند یا Documentation می‌شود .

فاز ۵ : مستندسازی(Documentation)

در این مرحله اطلاعاتی در مورد پروژه در قالب متن و نمودار، فراهم‌می‌شود .

فاز ۶ : نگهداری(Maintenance)

این فاز به معنای ارتقا، بهبود سیستم و رفع اشکالات احتمالی در نظرگرفته‌شده‌است تا سیستم در جهت رشد و افزایش دانش و کاهش نواقص احتمالی، پیش رود. مسیر برگشت در شکل۴-۱، نشان دهنده فرآیند برگشت است یعنی گاهی می‌توان به مراحل قبل بازگشت و بخشی را اصلاح کرد و حلقه‌های مشاهده شده در شکل، اشاره به روند تکاملی سیستم دارد و مدل، یک مدل تکراری می‌باشد.

تمرین: با تحقیق و پژوهش، به پرسش‌های زیر پاسخ دهید:

الف- امروزه چه ابزارهای کیس(CASE Tools) برای فازهای مختلف مهندسی‌دانش وجود دارد. این ابزارها بیشتر در در کدام فازها موثرتر و کارآیی بیشتری داشته‌اند؟ چه متدولوژی‌هایی برای مهندسی‌دانش، پیشنهادشده‌است؟

ب- مفهوم مهندسی‌دانش را با مهندسی‌نرم‌افزار، مقایسه‌کنید. چه شباهت‌ها و تفاوت‌هایی بین این دو مفهوم وجود دارد؟

بازنمایی دانش

بازنمایی دانش (نمایش دانش)

در ادامه، تکنیک‌های مختلف در کدکردن دانش در یک سیستم خبره را مورد بحث، قرارمی‌دهیم .

ارائه دانش = کد کردن دانش

تفاوت برنامه‌های متداول با سیستم‌های خبره از دید دانش

- برنامه‌های متداول روی داده‌ها عمل پردازش انجام می‌دهند (داده ها را پردازش‌می‌کنند)
- سیستم‌های خبره روی دانش عمل پرداش انجام‌می‌دهند (دانش را پردازش می‌کنند)

دلایل نمایش دانش در سیستم‌های خبره

۱. پوسته سیستم های خبره طوری طراحی می شوند که برای روش خاصی از نمایش دانش مثل قواعد یا منطق مناسبند .

۲. 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 ای‌حل نماید که دیگران قادر به آن نیستند . منظور از Domain این است که یک محدوده خاصی داشته باشد مثل مکانیک ، مهندسی ، پزشکی و...

| Expert |= Expert – Non Expert = knowledge

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

پرسش: دانش چیست ؟

دانش به فهم و دانایی اشاره‌می‌کند . از نظر فلسفی، دانش مفهومی است که همه می‌دانند ولی قادر به تعریف آن نیستند.

چیزی که واضح است این است که هرچه دانش بیشتر باشد توانایی بیشتر است

Knowledge is Power

توانا بود هرکه دانا بود

و سیستم خبره قوی تر، دارای دانش بیشتری است .

با توجه به دانش، یک تعریف از سیستم خبره به صورت زیر است :

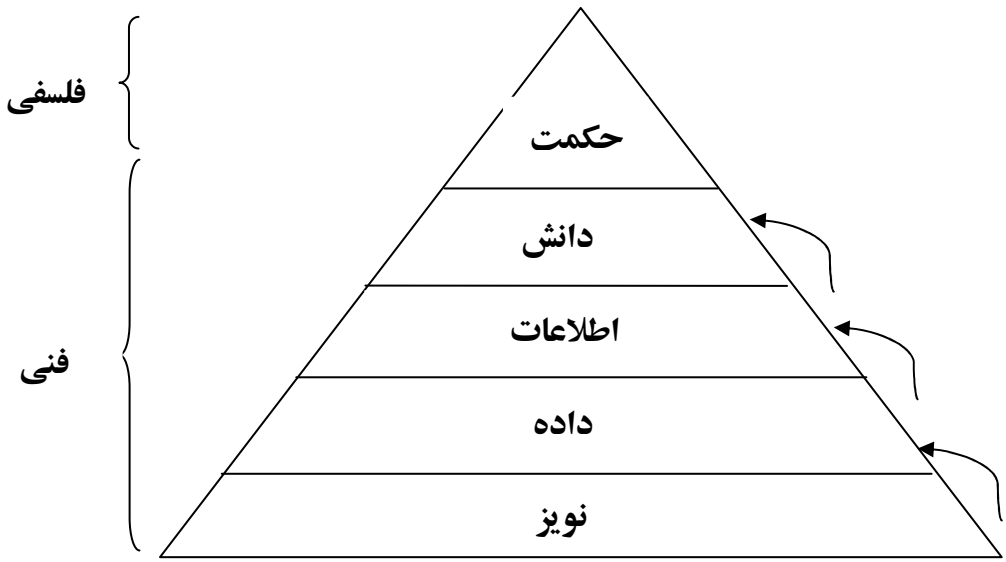
Expert System = knowledge + Reasoning

سیسستم خبره = دانش + استدلال

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

Program = Data structure + Algorithms

هرم سلسله مراتبی دانش



شکل۵-۱: هرم دانش

- در پایین ترین سطح نویز قرار دارد سپس داده و بعد ازآن اطلاعات و سپس دانش و حکمت قرار دارد.
- داده؛ نویزهای پاک شده است و قسمت‌های غیرمفید حذف شده‌اند.
- اطلاعات یعنی داده‌هایی است که پردازش‌شده است.
- دانش یعنی اطلاعاتی که پردازش‌شده است
- حکمت(خرد) دانشی است که هدف و ارزش دارد و به زندگی و فلسفه آن، مربوط است.

نکته: اگر داده‌ای، هیچ مفهومی نداشته باشد نویز است. اما اگر بدانیم این سمبل‌ها و نمادها برای چه منظوری است داده‌می‌باشد .به طور مثال اگر شخصی به زبانی سخن بگوید که برای شما ناآشناست، سخنان آن شخص برای شما نویز محسوب‌می‌شودچون هیچ مفهومی برای شما ندارد .

نویز : داده‌ای که هیچ مفهومی ندارد .
داده : حداقل می‌دانیم برای چه منظوری استفاده می‌شود.
دانش : فهم در یک موضوع خاص است.

مثال:
فرض کنید یک رشته ۲۴ عددی داریم که از هر چیزی ممکن است تشکیل شده باشد . اگر ندانیم این رشته به چه منظور استفاده می شود نویز 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 می‌گیرد. مثلا اگر رشته ۲۴ عددی را ۸ تا ۸ تا جداکنیم هر ۸ کاراکتر نمایانگر یک کد اسکی است و می‌توانیم به کد اسکی برگردانیم .

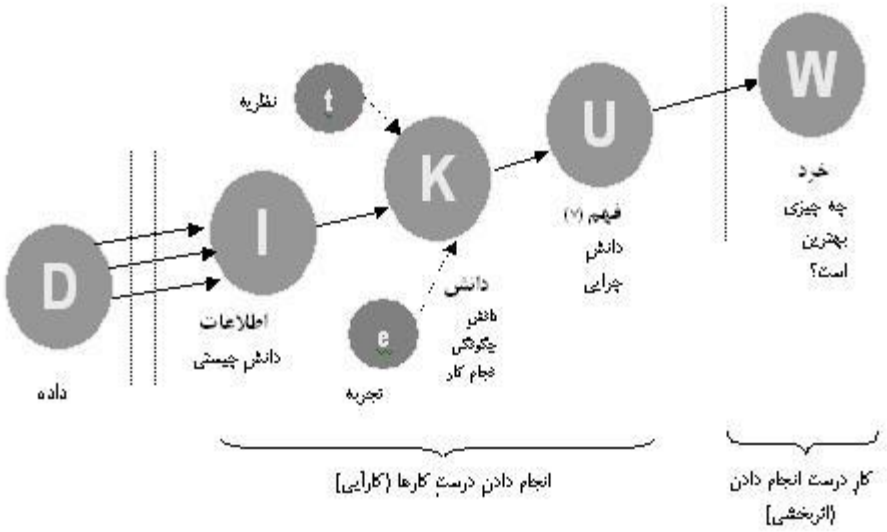
یا تصورکنید این عدد به + Gold تبدیل شود که خود یک اطلاع می باشد. پس الگوریتم مبدل داده به اطلاعات است که + Gold نمایانگر افزایش قیمت طلا باشد . اگر از این اطلاع در جایی استفاده‌شود تبدیل به دانش می شود مثلا "چون قیمت طلا در حال افزایش است پس طلا خریداری شود " یک دانش را ایجاد کرده است.

دانش به کار گیری اطلاعات است

فوق دانش meta Knowledge، دانش چگونگی استفاده از دانش و تجربه است .

نکته: قابل‌توجه‌است که تمامی این تعاریف، نسبی هستند مثلا ممکن است چیزی که در محیطی دانش، درنظرگرفته‌می‌شود در محیط دیگر داده‌باشد .

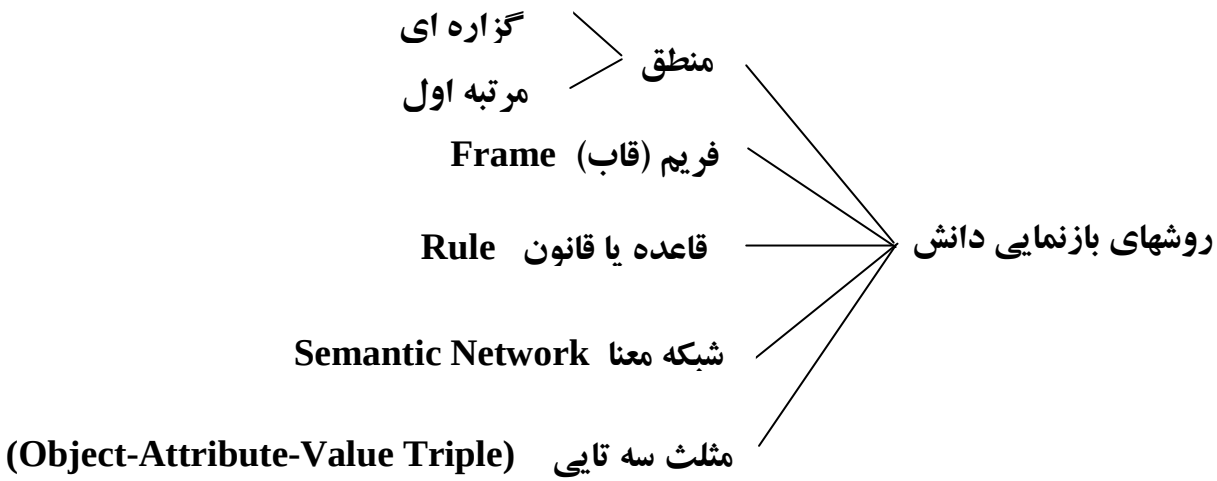
تمرین: شکل ۵-۲ را تفسیر کنید.



شکل۵-۲: روند تبدیل از داده تا خرد

بازنمایی دانش

نمایش دانش یا بازنمایی دانش روشهایی است که برای کد کردن دانش در پایگاه دانش سیستم خبره استفاده می شود. بعد از آنکه دانش از یک فرد خبره در دامنه خوش – متمرکز (well-focused domain) بدست آمد، باستی در یک سیستم خبره کد شود. برای کدکردن دانش نیاز به ساختار بندی دانش است به گونه‌ای که سیستم، قادر به حل مساله به روشی مشابه فرد خبره باشد.



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

در علوم شناختی ، حقایق (fact) 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 سیستم خبره حقایق و واقعیت‌ها بخشی از قالبها Frames و شبکه های 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

روش مثلث سه تایی OAV :

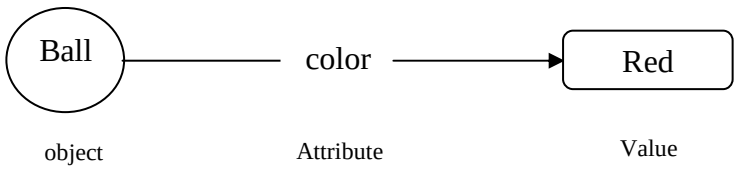
OAV یک نوع گزاره پیچیده تر است که یک عبارت را به سه بخش تقسیم می‌کند .

OAV وظیفه دارد به گزاره ، ساختاریا بخش بندی بدهد این بخش بندی تشکیل شده است از Object و Attribute و Value . مزیت OAV این است که با عبارتهای کمتری می‌توان دانش بیشتری را ارائه‌نمود.

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

The ball's color is red

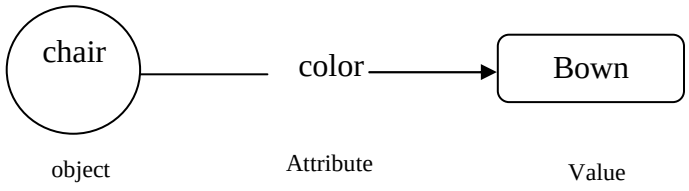
حال در این گزاره سه بخش بالا را معین می کنیم:



شکل۵-۳: سه تایی OVA برای توپ

و یا در گزاره دیگری داریم :

Chair color is brown



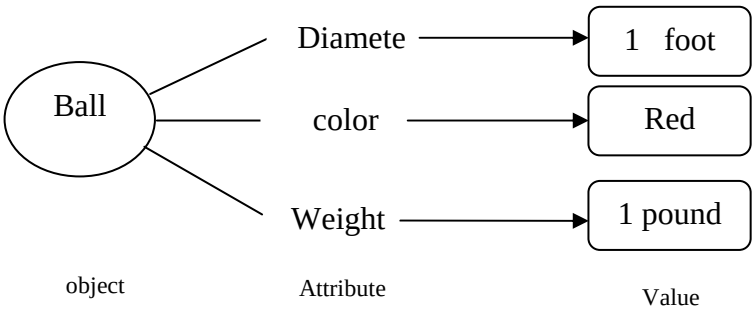
شکل۵-۴: سه تایی OVA برای صندلی

درمورد گزاره It's raining داریم

شکل۵-۵: سه تایی OVA برای آب و هوا

یک شی با چندین ویژگی Object with multiple attribute :

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



شکل۵-۶: سه تایی OVA با چندین ویژگی

به این دانش ، دانش سه تایی گفته می‌شود زیرا به سه بخش تبدیل می‌شود و مدیریت و سازماندهی آن ساده‌تر است ولی بیانگر یک گزاره است.

نکته ۱: یک Object میتواند مفهومی فیزیکی(ماشین ، توپ و ...) یا انتزاعی (عشق ، دوستی و...) باشد .

نکته ۲: Value می‌تواند یک مقدار داشته باشد (Single value facts) و یا چند مقدار داشته باشد (Multi Value facts)

فشارسنج مثالی است ازsingle value facts که سه مقدار دارد ولی از سه مقدار درحال افزایش ، ثابت ، در حال کاهش در هر لحظه تنها یک مقدار می‌گیرد. یعنی در هر لحظه فشارسنج تنها یکی از مقادیر در حال افزایش یا در حال کاهش و یا ثابت، نشان‌می‌دهد . در فرمها این خاصیت با Radio Boxها نمایش‌داده‌می‌شود.

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 مسلما 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 فرم ها این خاصیت با Check box ها نمایش داده‌می‌شود.

تمرین: تحقیق کنید علاوه بر روش‌های بانمایی معرفی شده، چه روش‌های دیگری وجود دارد.

انواع دانش

دانش به انواع گوناگونی تقسیم می شود :

دانش رویه ای Procedural knowledge :

دانش رویه ای دانشی است که چگونگی حل یک مساله را توضیح می دهد و به بررسی قوانین ، استراتژی ها ، لیست اهداف و پروسیجرها می‌پردازد.

دانش توصیفی Declarative:

دانش توصیفی، دانشی است که در مورد بخش‌های شناخته‌شده‌ی مساله توضیح‌می‌دهد 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 مسائلی 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 بدست‌آورده است .

دانش ساختاری Structural knowledge:

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

در جدول ۵-۱ نمونه‌هایی برای انواع دانش آورده شده است :

جدول ۵-۱: انواع دانش

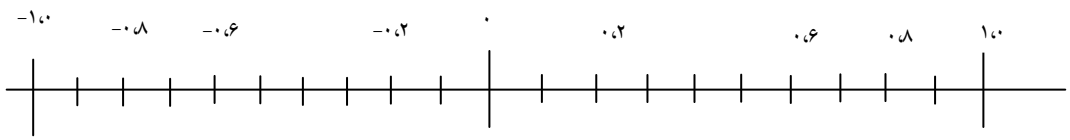
Types of knowledge	
Procedural	Rules قوانین Strategies استراتژی ها Agendas اهداف Procedures رویه ها
Declarative	Concepts مفاهیم Objects اشیا Facts حقایق
Meta	Knowledge about the Other types of knowledge And how to use them دانشی که معین کند از کدام دانش استفاده شود وچگونه از آنها استفاده شود
Heuristic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

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 ما نسبت به یک حقیقت می باشد . اولین بار ضریب قطعیت در سیستم پزشکیMYCIN مورد استفاده قرار گرفت.
شکل ۵-۷ مربوط به CF سیستم MYCIN می‌باشد در این شکل CF مقادیری از ۱،۰ - تا ۱،۰ + دارد.



شکل۵-۷: ضریب قطعیت

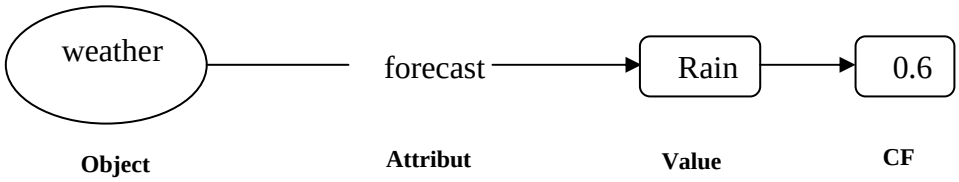
که در این نمودار اعداد نمایانگر مقادیر زیر می باشد :

۱،۰- : کاملا نادرست ۰،۸- : تقریبا نادرست ۰،۶- :احتمالا نا درست

۰،۲- تا ۰،۲ : شناخته شده نیست

۰،۶ : احتمالا درست ۰،۸ : تقریبا درست ۱،۰ : کاملا درست

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



شکل۵-۸: سه تایی OVA همراه با ضریب قطعیت

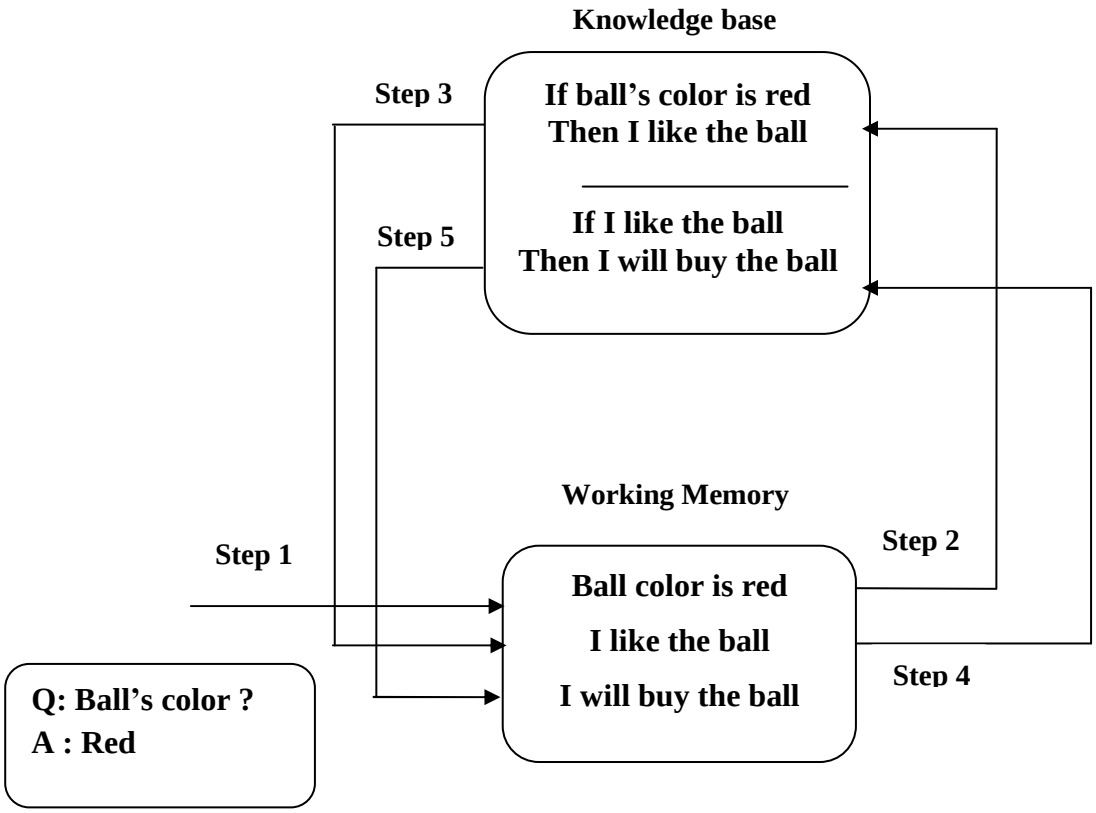
قانون Rule :

قانون ساختاری از دانش است که اطلاعات شناخته شده‌ای را به اطلاعات شناخته‌شده دیگری مرتبط می‌سازد که می‌تواند این ساختار مورد استنتاج و نتیجه‌گیری قرار گیرد .

به طور مثال :

If the ball's color is red
Then I like the ball

اگر توپ قرمز باشد من این توپ را دوست دارم .



شکل۵-۹: روند استنتاج در مثالی ساده

شکل ۵-۹، روند استناج در این مثال نشان می دهد. در سوال می پرسد توپ چه رنگی است ؟ پاسخ می دهد : قرمز

وقتی قرمز به سیستم داده شد قانون شماره ۱ فعال می شود و نتیجه می گیرد توپ رادوست دارد و سپس قسمت پایین فعال می شود و نتیجه می گیرد توپ را خواهد خرید .

انواع قانون

قانون‌های رابطه‌ای Relationship Rules

IF the battery is dead
Then the car will not start

اگر باطری مشکل داشته باشد ماشین روشن نخواهد شد .
در این نوع قانون رابطه ای بین علت و معلول وجود دارد .

قانون های توصیه‌ای Recommendation Rules

IF the car will not start
THEN take a cab

اگر ماشین روشن نمی شود . کاپوت ماشین را بالا بزن .
قانون های توصیه ای 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 نمی شود و سیستم سوخت مشکلی ندارد سیستم برق را بررسی کن.
این قوانین به یک جمله امری ختم می شود .

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 می کند .

قانون‌های تجربی Heuristic Rules

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

اگر ماشین روشن نمی شود و مدل ماشین فورد سال ۱۹۵۷ است پس شناور آن را چک کن
قوانین تجربی از روی تجربه های قبلی به انجام کار می پردازد .

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 می‌شود قوانین عام تر و کلی تر گردند

IF ?X is Employee
AND ?X Age>65
THEN ?X can reti**re**

Pattern Matching Rules یعنی تطبیق مقدار با متغیر

Meta Rules

Meta Rules قانون یا فرا قانونی است که بیان می کند از چه قانونی استفاده شود.

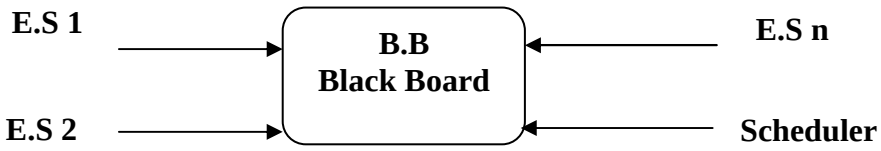
IF the car will not start
AND the electrical System is operating normally
THEN use rules concerning the fuel system

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

Black Board

حل مسائل توزیع شده Distributed problem solving (DPS)

ممکن است در مسائلی نیاز باشد چندین خبره در مورد یک مساله و برای حل آن بیندیشند و به عبارت دیگر چندین خبره با هدف حل نمودن یک مساله بپردازند . چنین مسائلی از نوع مسائل توزیع شده می باشند . و یکی از مکانیزمهای پیاده سازی آنها تخته سیاه Black Board می باشد. بخشی از برنامه مفهوم تخته سیاه را ایجاد می کند که این تخته سیاه وسیله‌ای برای اشتراک گذاری راه حل‌ها می باشد .



شکل۵-۱۰: مفهوم تخته سیاه

در پیاده‌سازی این بخش یک فایل است. به طور مثال یک فایل متنی است و به علت این که سیستم‌های خبره برنامه هستند راه حل‌های مسائل حل شده را داخل این فایل قرار می‌دهند 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 می‌کند می بایست خبرگی در زبان‌های مختلف داشته باشد و یا افراد متفاوتی که در یک مکانیکی کار می کنند هر کدام در بخشی از اتومبیل خبره هستند ، بخش سوخت ، بخش الکتریک ، موتور ، چرخها و ...

مثال:

سیستم خبره عیب یاب کامپیوتر که پس از شناخت عیوب خبره های متفاوتی برای حل مشکل فرا خوانده می شوند ، خبره در زمینه مانیتور ، خبره در زمینه Mother Board و...

مولفه های اصلی سیستم های خبره توزیع شده:

۱- مجموعه ای از سیستم های خبره Commiunity of different Expert System

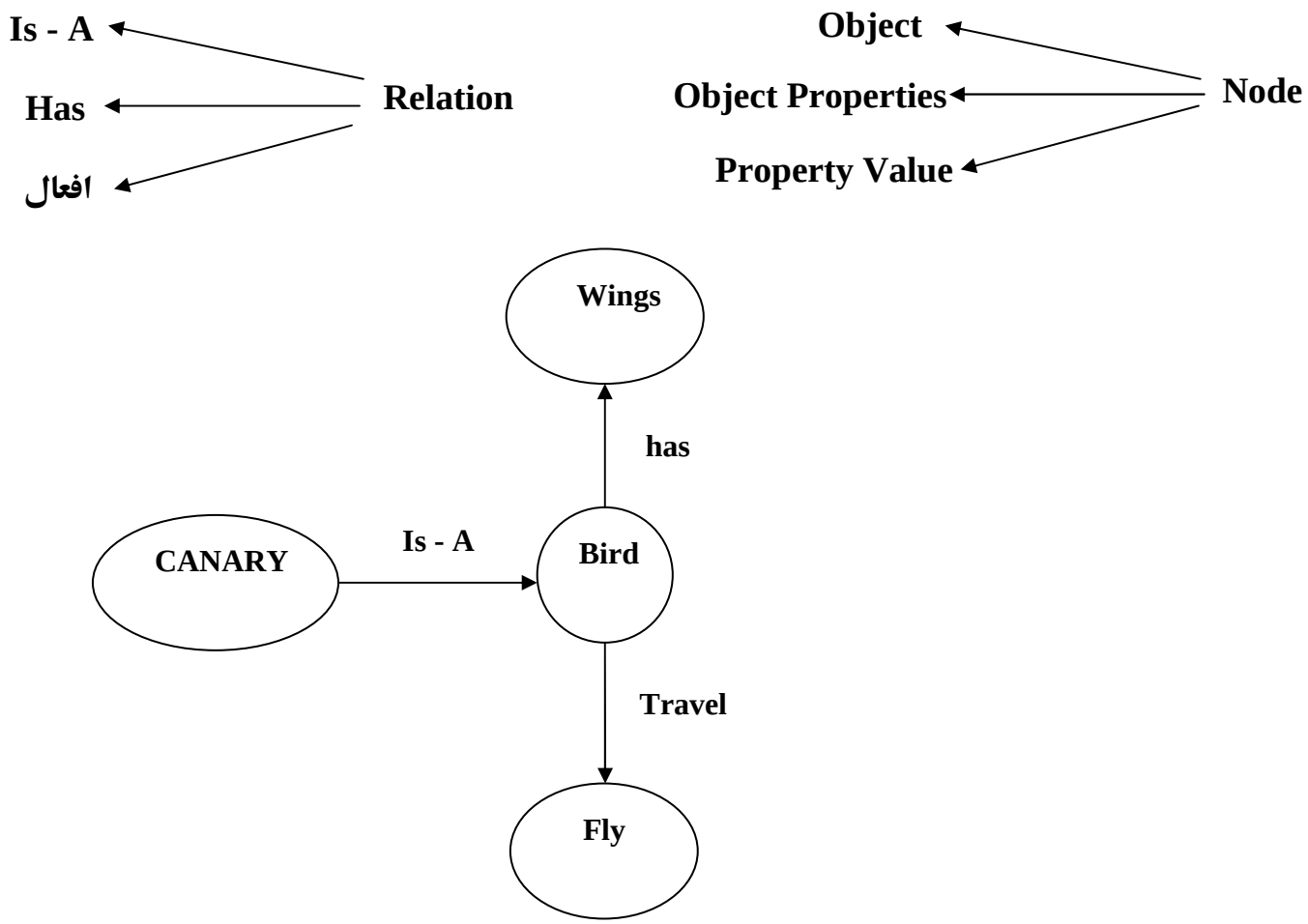
۲- تخته سیاه Black Board

۳- زمان بند Schedular

سیستم تخته سیاه علاوه بر این که کمیته ای از سیستم های خبره است بخشی برای قرار دادن راه حل ها (Black Board) نیز دارند و یک زمانبند نیز برای تقسیم زمان موجود و در اختیار قرار دادن زمان در میان خبره ها دارند

شبکه های معنا Semantic Network

شبکه های معنایی یک روشی برای بازنمایی دانش با استفاده از گراف هستند که از گره ها و یالها تشکیل شده است به طوری که گره ها بیانگر اشیا و یالها نشانگر ارتباط بین اشیا می باشد.



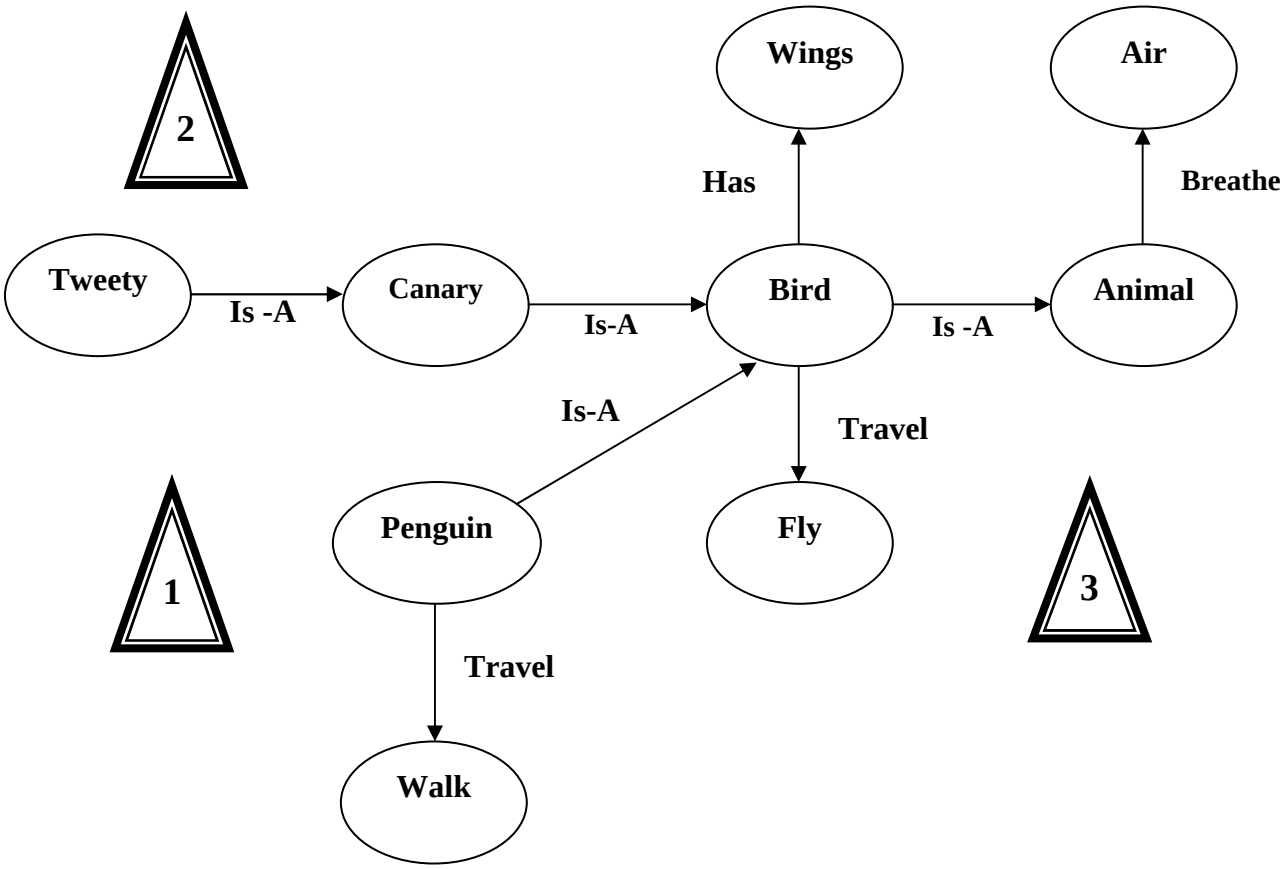
شکل۵-۱۱: شبکه معنا برای پرنده

این مثال در مورد پرنده است. این گراف دارا ی سه صفت داشتن ، جابجا شدن ، و نوع پرنده است. نکته ای که در اینجا قابل مشاهده می باشد این است که جهت در گراف شبکه های معنایی مهم هستند :

- پرنده بال دارد
- پرنده با پرواز جابه جا می شود
- پرنده مورد مطالعه قناری است .

این شبکه می‌تواند توسعه پیدا کند و گسترش با استفاده از سه روش صورت می گیرد.
شبکه معنا با استفاده از ۱- خاص‌تر شدن ،۲- عام‌تر شدن و ۳- مفاهیم مشابه گسسترش می یابد .

مثال



شکل۵-۱۲: گسترش شبکه معنا

در بخش اول(۱) مفاهیم مشابه نمایش داده شده است یعنی قناری و پنگوئن هر دو پرنده هستند.
در بخش دوم (۲) خاص تر نمودن مفاهیم نمایش داده شده قناری یک نوع پرنده است و Tweety یک نمونهخاص از قناری هاست.
در بخش سوم (۳) عام تر شدن مفاهیم نمایش داده شده است پرندگان گونه ای از حیوانات هستند.

وراثت Inheritance

یک ویژگی بسیار مهم در شبکه های معنا وراثت است . وراثت بدین معناست مفهوم یاخاصیتی از گره‌ای به ارث برده شود این ویژگی در شبکه‌های معنا به وسیله $Is - A$ نمایش‌داده‌می‌شود. در مثال بالا تمام ویژگی های Animal را Bird دارد و Animal ها موجوداتی هوازی هستند .

وراثت باعث کاهش حجم پایگاه دانش می شود و سبب می شود هر مفهومی چندین بار تکرار نشود .

در مثال بالا مشاهده می شود 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 شبکه های معنا جستجویی روی گراف صورت می گیرد تا پاسخ یافته شود .

مشکل اصلی شبکه های 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 این بدین معنی است که هر چه وجود دارد توصیف می کند در نتیجه زبانهای توصیفی همچون Lisp ,SQL ,Prolog, به خوبی می‌تواند شبکه معنا را کد کند.

مدیریت استثناها Exception Handling

مثال :پنگوئن ی‌ک پرنده است ولی پرواز نمی کندکه این قابلیت توسط exception Handling تصحیح می شود .

ای‌ن مفهوم مشابه مفهوم Override در زبان های شی گرا است .

Override the Inherited Information

فریم یا چهارچوب Frame:

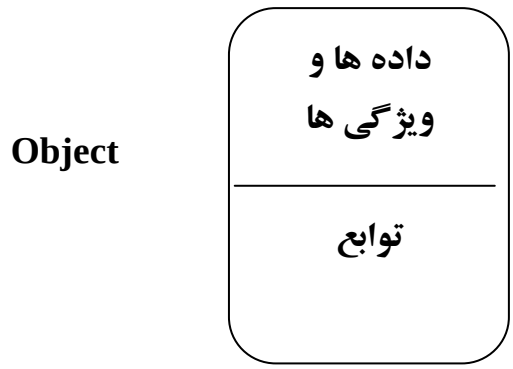
اگر به شبکه های معنا، مفاهیم رویه ها Procedure ، اضافه شود، تشکیل فریم می دهد .

Semantic Network + Procedure = Frame

فریم هم توصیفی است هم رویه ای.

مفهوم فریم ، مشابه مفهوم Object (شیء) است.

Object = Data (Properties) + Procedure



فریم : فریم ساختمان داده ای برای بازنمایی دانش کلیشه ای مشابه مفهوم یا شیء می باشد .

Object در مهندسی نرم افزار، همان فریم در سیستم خبره است .

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

CRC : خیلی از مفاهیم شی را توسط فرم ها و کارت ها نمایش می‌دهند که به این کارت ها CRC گویند. مثال زیر بیانگر کارتی در مورد دانشجوست.

Report Card

Student Name	
Address	

Course	**Grade**
Chemistry	
Math	
English	
***	

شکل۵-۱۳: نمایش کارت یا فرم

مثال بالا کارنامه ای است که در آن نام و آدرس دانشجو و نام دروس ترم جاری و نمرات آنها را نمایش می دهد .

ساختار کلی قاب : در شکل زیر یک ساختار کلی فریم نمایش داده شده است :

Frame Name	Object 1	
Class	Object 2	
Properties	**Property1**	**Value1**
	Property2	**Value2**
	***	***
	***	***

شکل۵-۱۴: ساختار کلی فریم

در یک کارت CRC می بایست نام فریم ، کلاس ، و ویژگی ها و مقادیر آن ویژگی ها آورده شود. فیلد class یک فیلد اختیاری در قاب می باشد که مقدار آن obj2 نام قاب دیگر مرتبط با obj1 است . این رابطه معمولا از نوع Is – A می باشد .

Obj1 Is – A Obj2

Obj1 اطلاعاتی از obj2 به ارث می برد .

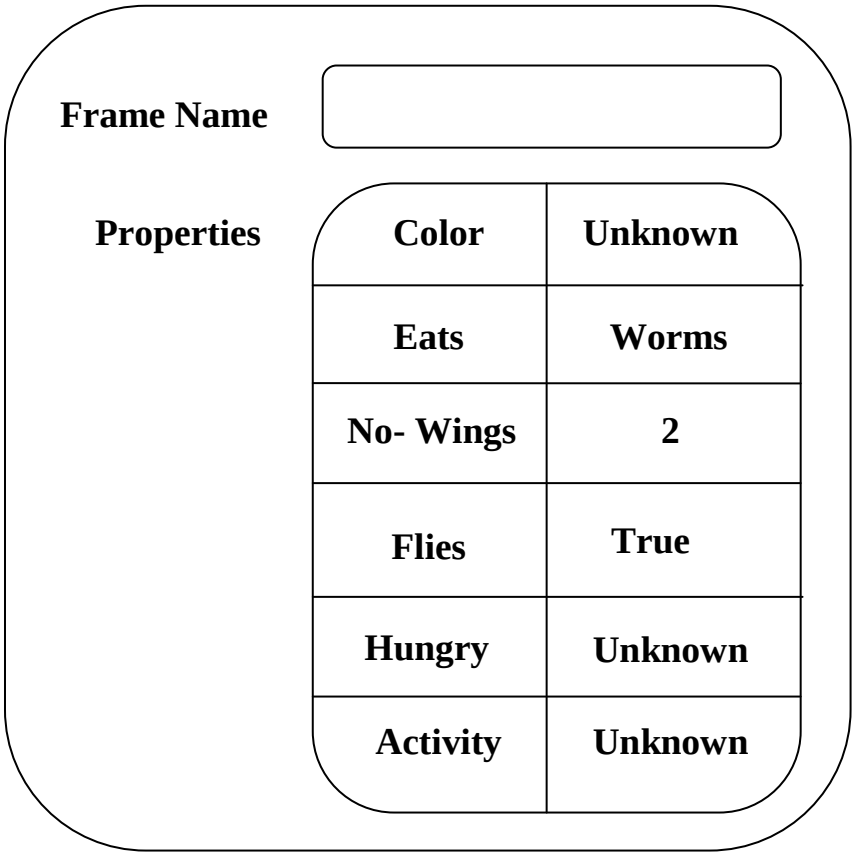
قاب کلاس Class Frame

به طور مثال CRC زیر یک فریم در مورد یک پرنده است .

پس نام فریم پرنده است و ویژگی های آن رنگ ، نوع خوراک ، تعداد بالها ، قابلیت پرواز ، گرسنگی و فعالیت پرنده است .

در این نوع مسائل مشاهده می شود که ویژگی ها دو حالت پویا و غیر پویا (استاتیک و دینامیک) دارند :

- ایستا: رنگ ، تعداد بال
- پویا: گرسنگی و فعالیت



شکل۵-۱۵: فریم پرنده

چون فریم توسعه یافته روش Semantic Network است ویژگی وراثت را حتما دارد .

رفتارهای ارثی Inheriting Behavior

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

وجوه Facets

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

Facet به دو نوع تقسیم می شود :

۱. IF Needed در صورت نیاز
۲. IF change در صورت تغییر

اگر Facet ما برای دستور دادن ، به این منظور که یک ویژگی چگونه مقداری بگیردIF Needed است و اگربه 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 نامیده‌می‌شود .

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 می شود . در IF needed فقط مقداری عوض می شود 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 صورت‌گرفته، خوردن خواهد بود. در IF Change با توجه به ویژگی خاص عمل خاصی صورت خواهدگرفت یعنی به ازای مثبت یا منفی شدن ویژگی خاصی عمل خاصی صورت خواهد گرفت. Self های قبل از ویژگی ها مشابه کلمه THIS در برنامه‌نویسی شیء‌گراست.

منطق Logic

قدیمی ترین 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 و ارتباط زبان Prolog با سیستم خبره مانند ارتباط زبان اسمبلی با زبانهای برنامه‌سازی است و تمام زبانهای بازنمایی در پایین‌ترین سطح، به منطق می‌رسند همانگونه که زبانهای برنامه‌سازی در پایین‌ترین سطح، به اسمبلی می‌رسند، یعنی مفاهیم پایه را در بر می گیرد .

دو نوع منطق وجود دارد

۱. منطق گزاره ای
۲. منطق مسندی یا مرتبه اول

منطق گزاره‌ای

در منطق گزاره ای هر گزاره با یک نماد یا سمبل بازنمایی می شود و با بررسی جدول‌های درستی به استنتاج می پردازیم.

A= The car will start

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

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

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

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

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

و برای هم ارزی داریم $B \equiv A$:

A	B	$B \equiv A$
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 عمگر آنگاه داریم :

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

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

حساب مسندی

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

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

نمونه ای از مسند:

like (john,mary)

Predicate رابطه بین مفاهیم است در این جا like مسند یا Predicate است و mary , John پارامتر های Predicate هستند و همانطور که مشاهده می کنیدmary , John هر دو ثابت هستند زیرا با حروف کوچک شروع شده اند.

متغیر ها را با حروف بزرگ نمایش می‌دهیم

Likes(X,Y)

در این مثال like مسند میباشد و پارامتر های آن متغیر هستند زیرا X,Y را با حروف بزرگ نمایش داده‌ایم .

خروجی تابع یک Object است در حالی که خروجی predicate غلط یا درست False یا True می باشد .

Father(jack) = bob

Mother(judy) = kathy

تمام عملگرهای موجود در منطق گزاره ای قابل استفاده در منطق مسندی هم می باشد. به طور مثال عملگرهایی نظیر آن گاه ، عطف و فصل و ... در منطق مسندی قابل استفاده هستند. با ترکیب Predicate ها با استفاده از عملگر ها قادر خواهیم بود دانش های پیچیده تر را بازنمایی کنیم .

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

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

مفهوم سور وجودی و سور عمومی:

وقتی سور عمومی به کار می بریم که موردی برای تمامی موارد وجود داشت باشد یا بخواهیم وجود آن را در تمامی موارد بررسی کنیم . سور وجودی زمانی به کار می رود که حداقل یک مورد را بخواهیم بررسی کنیم .

همه mary را دوست دارند.

$\forall$ X likes(X,mary)

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

$\exists$ likes(X,mary)

نکته: در Prolog تنها سور عمومی وجود دارد و برای مفهوم سور وجودی نیز با ایجاد یک سری محدودیت به سور عمومی تبدیل می کنیم. تبدیل زبان گفتاری و محاوره ای به زبان منطق مرتبه اول کار ساده ای نیست زیرا زبان گفتاری دارای مرتبه n است و مرتبه منطق مسندی، یک است .

Modus Ponens

تا کنون بررسی‌شد که Modus Ponens اطلاعات جدید را از داده مساله اولیه ایجاد می کند. 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 جمع آوری شود.

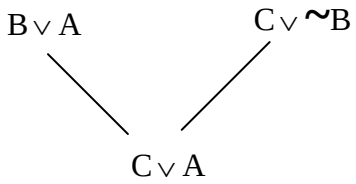
مفهوم Modus Ponens همان قیاس است.

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.

Resolution

Resolution یک استراتژی استنتاج است که در سیستم های منطقی برای تعیین درستی بیانیه یا حکم (Assertion) استفاده می شود .این روش تلاش می کند تا تئوری یا هدفی مانند گزاره P را با استفاده از مجموعه ای از اصول مرتبط با مساله اثبات نماید. در واقع سعی می کند در عمل ثابت کند که 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 جدید ی ایجاد می شود که Resolvent نامیده می‌شود.



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

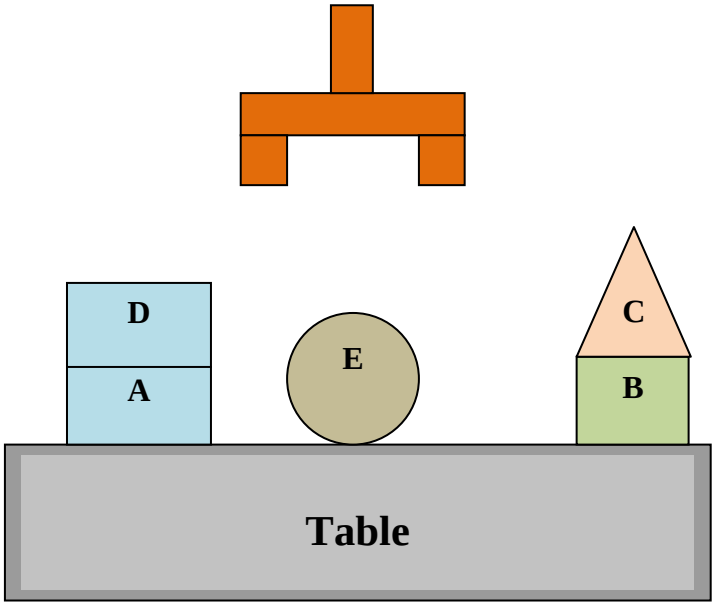
پرسش: روند ایجاد Resolvent ها تا چه زمانی، ادامه‌می‌یابد ؟

تا زمانی که در عبارات تناقض وجود داشته باشد . تناقض زمانی رخ می دهد که دو اصل از نظر منطقی در تضاد با یکدیگر باشند.
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 داده ها از طریق اطلاعات اولیه و فرم آنگاه و 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 ربات جهت برنامه ریزی یا Planning روبات‌ها می‌باشد . به طور مثال در شکل۵-۱۶، یک روبات، نشان‌داده‌شده‌است.



شکل۵-۱۶: طرح‌ریزی در جابجایی بلوک‌ها

قوانین زیر در مورد ربات بالا 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 مکعب‌ها ،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 می‌دهد پارامتر اول روی پارامتر دوم قرار دارد.تابع holding برای نمایش شیء ای که توسط بازوی ربات برداشته شده است به کار می رود. در اینجا بدین معنی است که ربات چیزی را بر نداشته است. در این مثال با تغییر موقعیت هر آیتم توابع و مقادیر تغییر کرده و می‌تواند 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 .

$$\text{Hand –holding(b)} \wedge \text{clear(a)} \rightarrow \text{putOn(b,a)}$$

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

تمرین: تحقیق کنید منطق و زبان‌های مبتنی بر منطق، چگونه می‌توانند در برنامه‌ریزی روبات‌ها استفاده شوند.

تکنیک های استنتاج

فرآیند کار با دانش، حقایق و استراتژی‌های حل مساله برای رسیدن به نتیجه را استدلال یا استنتاج می‌گویند.

پرسش: مفهوم استنتاج و استدلال چه شباهت‌ها وتفاوت‌هایی با یکدیگر دارند؟

استدلال استنباطی(Deductive Reasoning)

روش استنتاج استنباطی، روشی از کل به جزء می باشد.

Implication: I will get wet if I am standing in the rain
Axiom: I am stunding in the rain
Conclusion: I will get wet

من خیس خواهم شد، اگر زیر باران بایستم
من زیر باران ایستاده‌ام ، پس خیس خواهم‌شد.

$$B \text{ IF } A \equiv \frac{A \rightarrow B \quad A}{B}$$

در استنتاج استنباطی از Modus Ponens استفاده می‌کنیم.

استدلال استقرایی (Inductive Reasoning)

روش استنتاج استقرایی، جزء به کل می باشد. انسانها استدلال استقرایی را برای رسیدن به نتایج کلی، با استفاده از مجموعه‌ای محدود از حقایق و با کمک فرآیند تعمیم به کار می‌برند.

Premise: Monkeys in the pittsburgh zoo eat bananas
Premise: Monkeys in the cleveland zoo eat banana
Conclusion: In general , all monkeys eat banana

میمونها در باغ وحش A موز می‌خورند.
میمونها در باغ وحش B موز می‌خورند.
در نتیجه تمام میمونها موز می‌خورند.

در این مثال مشاهده می‌شود که با دیدن نمونه‌هایی محدود، با استفاده از تعمیم ، این قانون را به کل مجموعه نسبت‌داده شده‌است، ولی این نتیجه‌گیری همیشه درست نیست.

نکته: روش استقرایی برای ایجاد قوانین استفاده‌می‌شود، در حالی که روش استنباطی از قوانین برای رسیدن به نتایج، استفاده می‌کند.

استدلال معکوس یا انتزاعی (Abductive Reasoning)

در استدلال استنباطی، از علت به معلول می‌رسیم در صورتی که در استدلال معکوس، از معلول به علت می رسیم. این استدلال را استنتاج با ظاهر فریبنده یا توجیه کننده (Plausible Inference) می‌نامند.

Implication: Ground is wet if it is raining
Axiom: Ground is wet
Conclusion: It is raining ?

اگر باران بیاید، زمین خیس می‌شود.
زمین خیس است.
باران آمده ؟

$$\frac{A \rightarrow B \quad \equiv \quad B \text{ IF } A}{A}$$

اسستدلال قیاسی (Analogical Reasoning)

استدلال قیاسی، از مقایسه ی مفهومی با مفهوم دیگر به دست می آید.

Tiger Frame :
Specialization of : Animals
Number of legs : 4
Eats : meat
Lives : India and Southeast Asia
Color : tawny wuth stripes

در این مثال ویژ گی‌های ظاهری و بعضی از صفات ببر ها، آمده‌است. حال اگر شخصی باشد که تا به حال ببر، ندیده باشد ولی شیر دیده باشد برای این که به او ببر را بشناساند. می‌گوید تفاوت‌ها و شباهت های شیر ها با ببر ها چیست تا شخص با استفاده از تجربیات قبلی یک تصویر ذهنی از ببر تشکیل دهد. در این نوع استدلال افراد یک مدل ذهنی از مفاهیم را از طریق تجربیات پیشین در ذهن خود تشکیل می‌دهند.

استدلال عقل سلیم (Common Sense Reasoning)

این استدلال، از منطق خاصی پیروی نمی‌کند . در این روش به گونه‌ای استدلال می‌شود که انتظار می رود، بهترین استدلال ممکن باشد، در صورتی که ممکن است بهترین استدلال نباشدولی در اکثر موارد درست است.
فرض کنید شما می خواهید به مهمانی بروید بدون این که بدانید میزبان از گل خوشش می‌آید یا نه ؟ برای او دسته گلی می‌خرید. شما این نتیجه را به این دلیل گرفته‌اید که تصور می‌کردید، اکثر افراد از گل خوششان می‌آید.

یکی از وجوه تمایز هوش مصنوعی و هوش طبیعی عقل سلیم است که مدل‌سازی آن بسیار سخت و دشوار است. به عبارت دیگر از پیچیدگی زیادی برخوردار است. استدلال عقل سلیم معمولا ترکیبی از چندین استدلال است.

تمرین: تحقیق کنید که آیا استدلال پیش فرض(Default Reasoning)، همان استدلال عقل سلیم است. اگر یکی نیستند، چه تفاوت‌ها وشباهت‌هایی با یکدیگر دارند؟

استدلال غیر یکنواخت (Non Monotonic Reasoning)

در این استدلال با افزوده‌شدن حقایق جدید ممکن است حقایق و قواعد قدیم باطل و نادرست شوند. به طور مثال، اگر در یک سیستم خبره پزشکی به وجود نوعی بیماری پی ببریم ولی در طول مدتی که بیماری تشخیص داده شود، ممکن است بیماری از بین رفته باشد و یا بیماری های دیگری اضافه شود. به طور مثال اگر بیماری سرماخوردگی دیر تشخیص داده شود به بیماری ذات الریه تبدیل می‌شود. استدلال های احتمالی از نوع استدلال های غیر یکنواخت محسوب می شوند.

پرسش: استدلال‌های یکنواخت ساده‌ترند یا استدلالهای غیر یکنواخت؟ چرا؟

استدلال‌های یکنواخت ، به علت این که بحث کنترل ناسازگاری در آنها وجود نداشته و در طول زمان طراحی سیستم خبره حقایق و قوانین، ثابت‌خواهدبود.

تمرین: تحقیق کنید که علاوه بر روش‌های استدلال نام‌برده‌شده، چه روش‌های استدلال دیگری وجود دارد.

استنتاج (Inference)

استدلال یا Reasoning یک مفهوم عام است و می‌تواند برای انسان نیز به کار رود، ولی استنتاج یا Inference مفهوم فنی‌تر است و در مرحله پیاده‌سازی به کار می‌رود. Inference بخش‌های قابل پیاده‌سازی Reasoning است.

کاربرد قانون Modus Ponens در زنخیره استنتاج

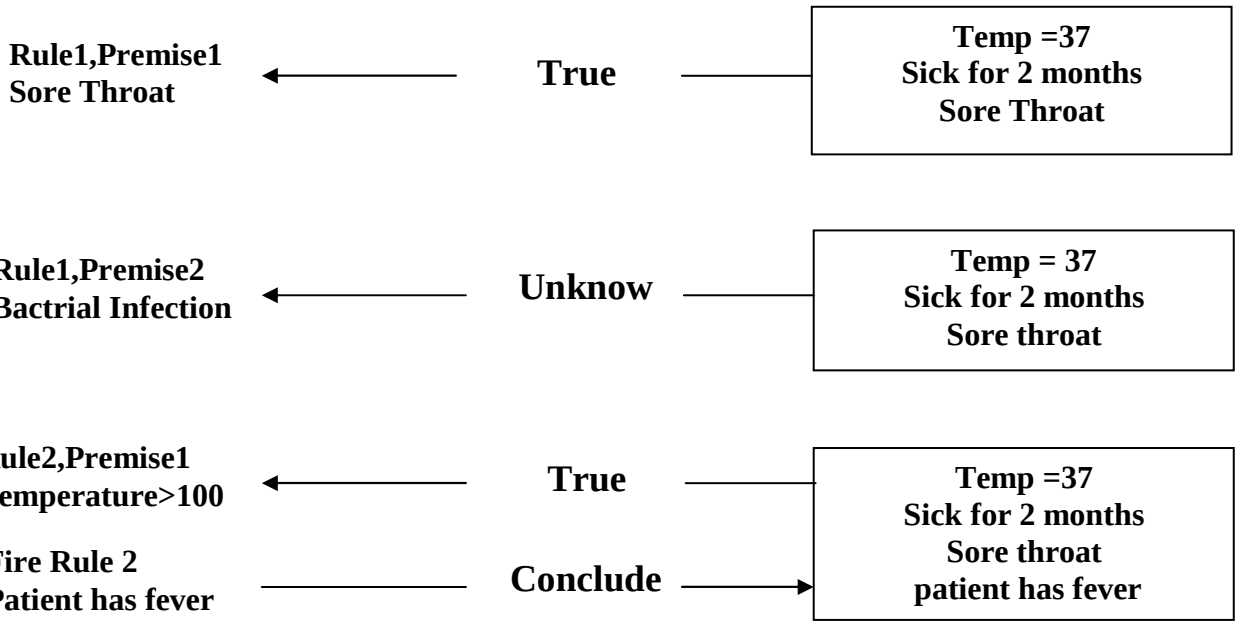
State-0 $E^1 \rightarrow E^2$ $E^2 \rightarrow E^3$	**State-1** $E^1 \rightarrow E^2$ $E^2 \rightarrow E^3$ $E^1 \rightarrow$temperature >37º
State-2 $E^1 \rightarrow E^2$ $E^2 \rightarrow E^3$ $E^1 \rightarrow$temperature >37º $E^2 \rightarrow$patient has high temperature	**State-3** $E^1 \rightarrow E^2$ $E^2 \rightarrow E^3$ $E^1 \rightarrow$temperature >37º $E^2 \rightarrow$patient has high temperature $E^3 \rightarrow$Advice two Aspirins

در state0 ، E^1 نتیجه‌می‌دهد E^2 و E^2 نتیجه‌می‌دهد E^3، حال مفهوم E^1 در state1 آمده است که دمای بالای ۳۹ درجه است و E^2 دمای بالای بیمار است و E^3 توصیه ۲ تا آسپرین است. این روند را روند استنتاج پیشرو می گویند.

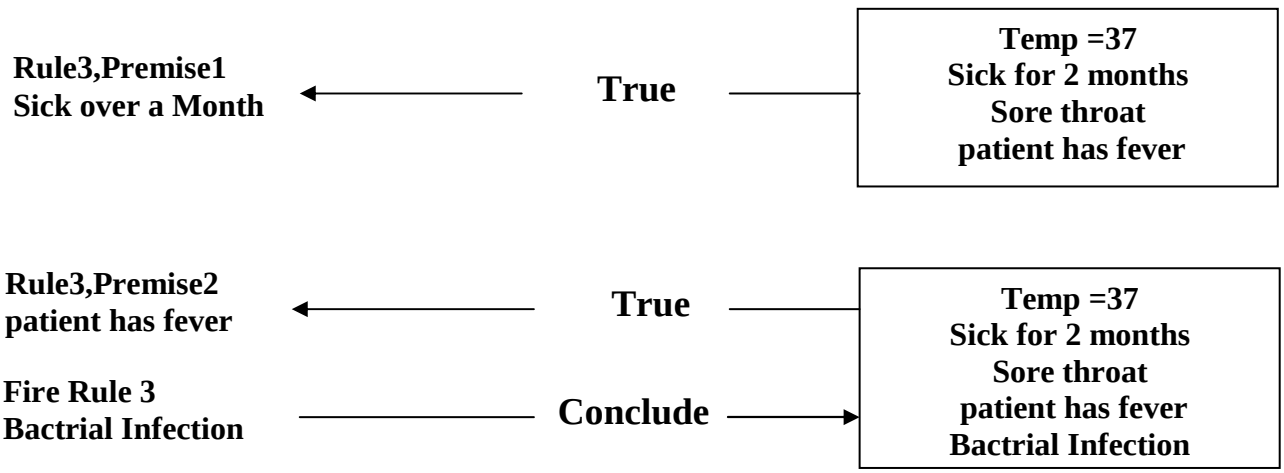
زنجیره پیشرو (Forward Chainig)

زنجیره پیشرو استنتاجی است که با مجموعه‌ای از حقایق شناخته شده، آغاز می‌شود و با استفاده از آنها، نتایج جدیدی گرفته‌می‌شود و این روند تا زمانی که دیگر نتوان از قانونی در روند استنتاج استفاده‌کرد، ادامه می‌یابد.

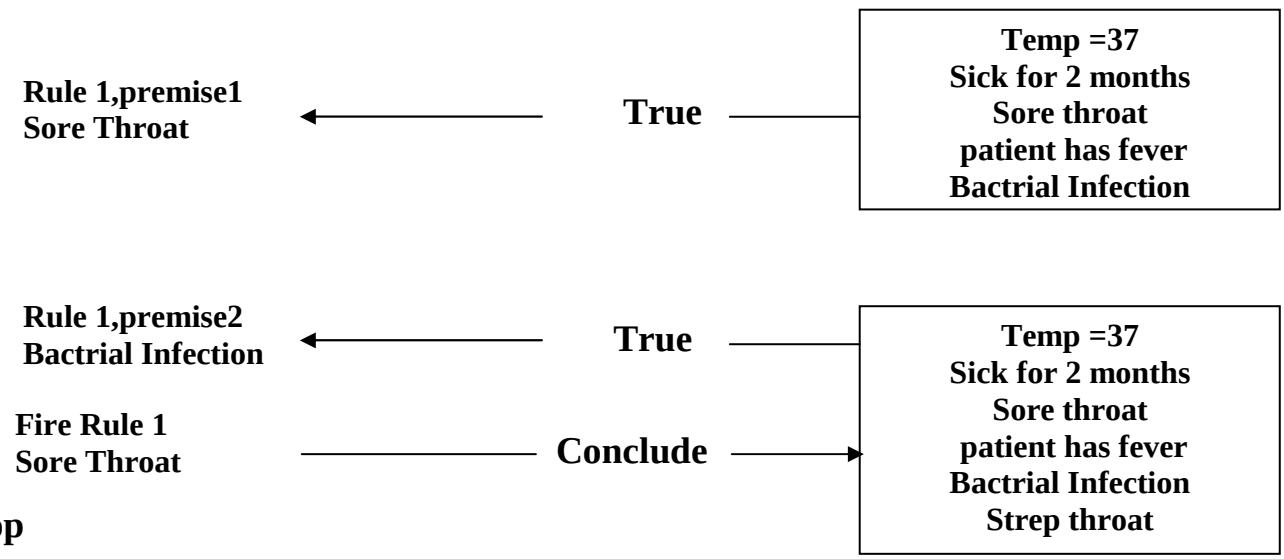
Cycle 1



Cycle 2 Note : Rule 1 again unknown and Rule 2 fired



Cycle 3



Stop

Rule 1:
If the paitent has a sore throat
AND we suspect a bacterical Infection
THEN we believe the paitent has strep throat

اگر بیمار گلو درد داشته باشد و ما به عفونت باکتریایی مظنون باشیم.
ما گمان به گلو درد میکروبی می بریم.

Rule 2:
IF the paitent's tempreture is>37
THEN the patient has a fever

اگر دمای بدن بیمار بالای ۳۷ درجه است . بیمار تب دارد .

Rule 3:
IF the paitent has been sick over a month
AND the paitent has a fever
THEN we suspect a bactrial infection

اگر بیمار بیشتر از یک ماه بیمار باشد و بیمار تب نیز داشته باشد.
ما به گلو درد عفونی مظنون خواهیم شد.

Rule 4:
IF the patient has a fever
THEN the paitent can't go out on a date

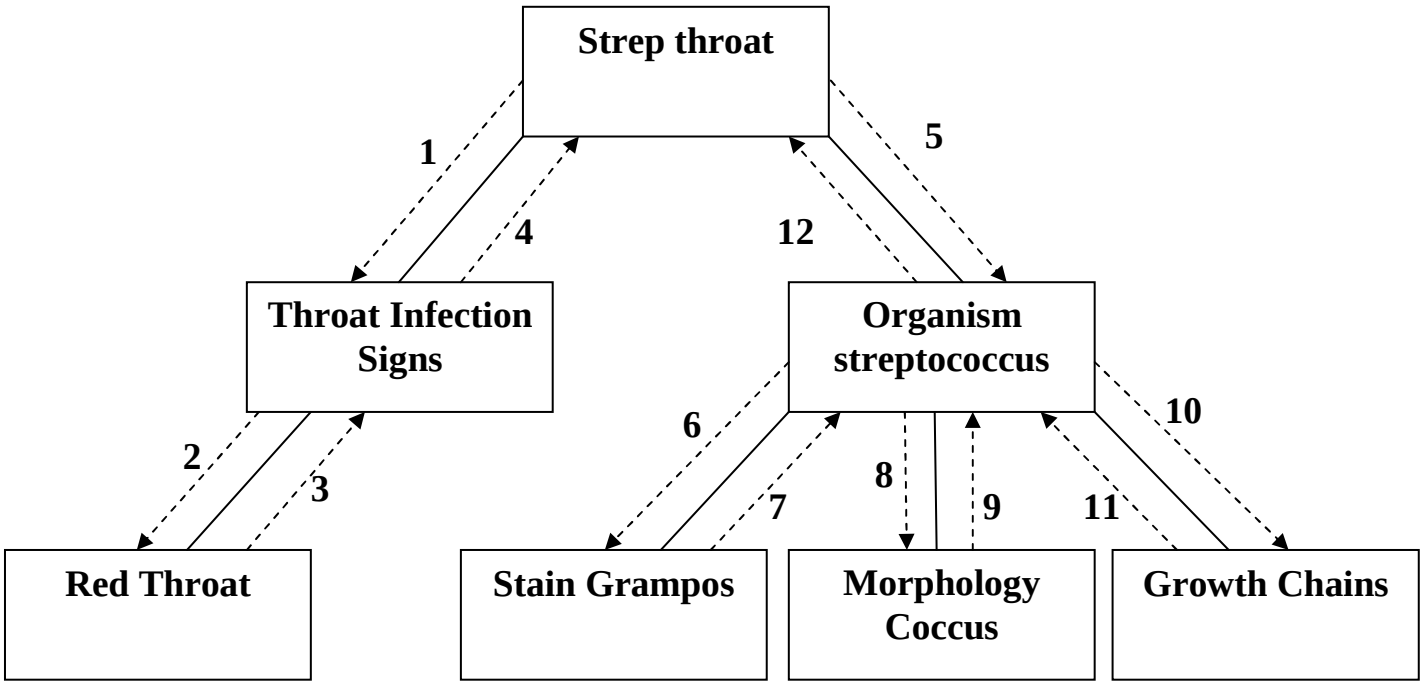
اگر بیمار تب دارد . بیمار برای مدتی نمی تواند از خانه خارج شود.

Rule 5:
IF the paitent can't go out on a date
THEN the paitent should stay home and read a book

اگر بیمار نتواند از خانه برای مدتی خارج شود پس باید در خانه بماند و کتاب بخواند .

زنجیره پسرو (Backward Chaining)

زنجیره های پیشرو و پسرو هر دو از استنتاج Modus Ponens استفاده‌می‌کنند. زنجیره پسرو استراتژی است که تلاش می‌کند تا فرضیه‌ای را با جمع‌آوری اطلاعات مرتبط اثبات نماید. در پایگاه دانش، قوانینی به نام قوانین اولیه وجود دارد؛ قوانین Primitive یا اولیه، قانونی است که توسط هیچ قانون دیگری نتیجه نشده‌است و به عبارت دیگر در روند استنتاج، هنگامی که به این قوانین رسیدیم استنتاج با مشاهده و بررسی حافظه کاری؛ به پایان می‌رسد.
در شکل ۶-۱، زنجیره پسرو برای بیماری Strep Throat بررسی می شود :



شکل۶-۱

در شکل۶-۱ با دنبال کردن اعداد throat Infection signs (علایم عفونت گلو) به Red throat (تورم گلو) می رسیم . پس اگر گلو متورم و قرمز باشد به مرحله بالا بر می گردد و نتیجه را ok می کند. در این بخش Red throat یک قانون اولیه یا primitive است چون بعد از آن چیزی چک نمی شود.

در این مثال پزشک فرض می‌کند که بیمار به بیماری Strep throat مبتلاست .سپس به دنبال علایم عفونت می‌گردد و با دیدن تورم و قرمزی گلو علایم را مشاهده کرده است و فرض اثبات شده است.

در روش Backward حدسی زده می‌شود و با توجه به قانون‌ها موجود به دنبال داده هایی می‌گردد که سبب اثبات درستی حدس ها شود. با مطالعه نمودار از پایین به بالا هر کدام از موارد یک Rule می‌باشد به طور مثال قرمزی و تورم گلو یک علامت برای عفونت است و اگر علامت عفونت مشاهده شود می تواند به خاطر بیماری strep throat باشد.

پس مطالعه از بالا به پایین (از ریشه به سمت برگها)روند استنتاج backward می باشد و از پایین به بالا (از برگها به سمت ریشه) نمایانگر Rule ها می‌باشد.

از بالا به پایین نمودار گویای این موضوع است که اگر کسی بیماری گلو درد داشته باشد که می‌تواند منشاء باکتریایی یا عفونی داشته باشد .اگر منشاء عفونی داشته باشد گلو متورم میشود و اگر باکتریایی باشد می بایست سه علامت Stain Grampos و morphology coccus و growth Chains را داشته‌باشد.

ممکن است این روش چندین روش را اثبات نماید. در اینگونه سیستم‌ها مجموعه اهدافی موجود است که باید تک تک به اثبات برسند.

پرسش: اگر مجموعه ای از اهداف داشته باشیم، روش Forward مناسب‌تر است یا Backward ؟

روش Backward مناسب‌تر است به علت اینکه روند از هدف به فرض مطمئن‌تر و مقرون به صرفه‌تر می‌باشد و نیازی به پیمودن مسیرهای اضافی برای رسیدن به هدف نیست.

حال اگر سیستم دارای چند هدف باشد بهتر است این اهداف متناسب با قوانین ، اهداف Module بندی شود تا روند استنتاج با کارایی و سرعت بیشتری انجام شود. این ماژول بندی باعث می‌شود تا سیستم سوال و جواب های مرتبط‌تری را بپرسد.

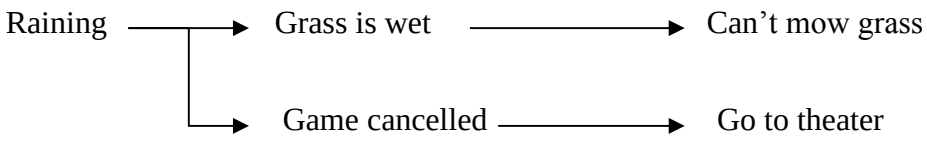
کاربر هوشمند(Intelligent User)

بعد از این که اهداف Module بندی شد حال چند سیستم خبره کوچکتر خواهید داشت و قتی سیستم Run می‌شود چند هدف برای اثبات دارید و از شما پرسیده خواهدشد شما در پی اثبات کدام هدف هستید. با دانستن هدف شما سیستم سوالات مرتبط با هدف شما را خواهد پرسید پس کاربر باید هوشمند فرض شود یعنی با قرار دادن لیستی از اهداف از کاربر کمک بخواهیم تا با انتخاب اهداف مورد نظر محدوده جستجو را مشخص کند.

معایب و مزایای تکنیک‌های زنجیره‌ای(روش‌های پیشرو و پسرو)

مزایای زنجیره‌ای پیشرو

۱. روش‌های پیش رو برای حل مسائل با جمع آوری اطلاعات شروع می کند بنابراین ای روشها برای مسائلی که در ابتدا نیاز به جمع آوری اطلاعات دارند مناسب هستند .زبان CLIPS نمونه ای از استنتاج پیشرو است.



۲. روش‌های پیشرو برای مسائلی نظیر طراحی planning ، پایش Monitoring ، کنترل ، تفسیر Interpretation مناسب است زیرا این نوع مسائل مبتنی بر جمع آوری اطلاعات است.

۳. در اینگونه مسائل با داده کم می‌توان نتایج زیادی یافت. مانند مثال بالا که با آمدن باران، چندین نتیجه گرفته شده است. (تفسیر های متنوع)

معایب زنجیره‌ای پیشرو

۱. روش‌های پیش رو فاقد اولویت‌دهی یا تعیین اهمیت سوالها هستند.

Q. Do you believe you are suffering a heart attack?
A. YES…YES…YES
Q. Is your nose running?
A. WHAT!!!!!

آیا تصور می کنید که دچار حمله قلبی شده اید ؟ بله

آیا آبریزش بینی دارید ؟ چی!!!!!!!!!!

۲. در سیستمهای پیش رو ممکن است سوالات نامربوطی پرسیده شود. چون هدف در این روش مشخص نیست و سیستم می بایست در پی جمع‌آوری اطلاعات باشد.

Q. Do you have a high temperature?
Q. Have you visited England lately?

آیا دمای بدن شما بالاست ؟

آیا به تازگی به انگلستان رفته‌ای ؟

مزایای زنجیره پسرو

روشهای پسرو، رای مسائلی که در آغاز فرضی بنا شده و سپس در جهت اثبات ان باشیم مناسب است.

۱. به دلیل بنا نمودن یک فرض سوالات مربوط به هم مطرح می شود .
۲. جستجو تنها بخشی از پایگاه دانش مرتبط با مساله را پیدا می کند .
۳. برای کاربردهایی نظیر عیب یابی ، تشخیص و توصیه مناسب است.

معایب زنجیره پسرو

روشهای پسرو به دلیل خط و سیر استدلال خاصی که دارند ممکن است به بن بست منتهی شوند. فراخوانهاMeta rule می توانند کمک موثری برای جلوگیری از این مشکل باشند. زبان Prolog نمونه‌ی از روش Backward می باشد .

انتخاب بین روش‌های پیشرو و پسرو

حال ببینیم برای حل مسائل روش پیشرو را انتخاب کنیم یا روش پسرو؟

ابتدا باید مساله را خوب بشناسیم اگر مساله به گونه‌ای است که در ابتدا نیاز به جمع آوری اطلاعات وسپس نتیجه گیری دارد روش پیشرو مناسب تر است.

اما اگر ابتدا حدسی زده می‌شود و سپس با جمع آوری اطلاعات، در تلاش برای اثبات حدس خود هستید روش پسرو، مناسب‌تر است.

در مسائلی ممکن است بخشی از مساله با روش Forward و بخش دیگر با روش Backward حل شود. به طور مثال در سیستم پزشکی ، پزشک ابتدا حدسی در مورد بیماری می زند(Backward) و با انجام آزمایشات و مشاهده نتایج ممکن است حدس پزشک تغییر نماید(Forward).

روشهای ترکیب دو روش Forward و Backward

چگونگی ترکیب این دو روش به صورت فنی دو روش دارد :

۱. سیستم های جدا از هم طراحی نماییم . ۲- استفاده از Demon Rules

طراحی سیستم‌های جدا از هم

برای مسائل پیچیده طراحان سیستم خبره اغلب ابتدا مساله را به چندین زیر وظیفه تقسیم می کنند، سپس برای هر مساله ، سیستم خبره جداگانه‌ای طراحی‌می‌شود. هر یک از این سیستم‌ها، بخشی از مساله اصلی را حل خواهندکرد و در حل مساله کنترل می‌تواند بین زیر سیستم‌ها، جابه‌جا شود.

Forward
IF patient is suffering persistent headahes
AND patient is suffering dizziness
THEN there is evidence of meningitis

اگر بیمار سر درد و سر گیجه دارد. احتمالا مننژیت دارد .

Backward
IF there is evidence of meningitis
THEN Load meningitis Expert

اگر علایم مشاهده شده مربوط به مننژیت است، سیستم خبره مننژیت را بارگذاری کن.

استفاده ازقوانین نگهبان یا Demon Rules

قوانینی هستند که وقتی شرایط آنها با محتوای حافظه کاری تبطیق یافت فعال می شوند .

Demon Rule 1 : Tank pressure problem
IF power is off
AND tank pressure>1000
THEN problem = tank pressure problem

اگر منبع خاموش است و فشار منبع بالاتر از ۱۰۰۰ است ، مشکل فشار منبع می باشد .

Demon Rule 2 : Emergency situation
IF problem = tank pressure problem
Then situation =Emergency

اگر مشکل فشار منبع می باشد، موقعیت اضطراری است .

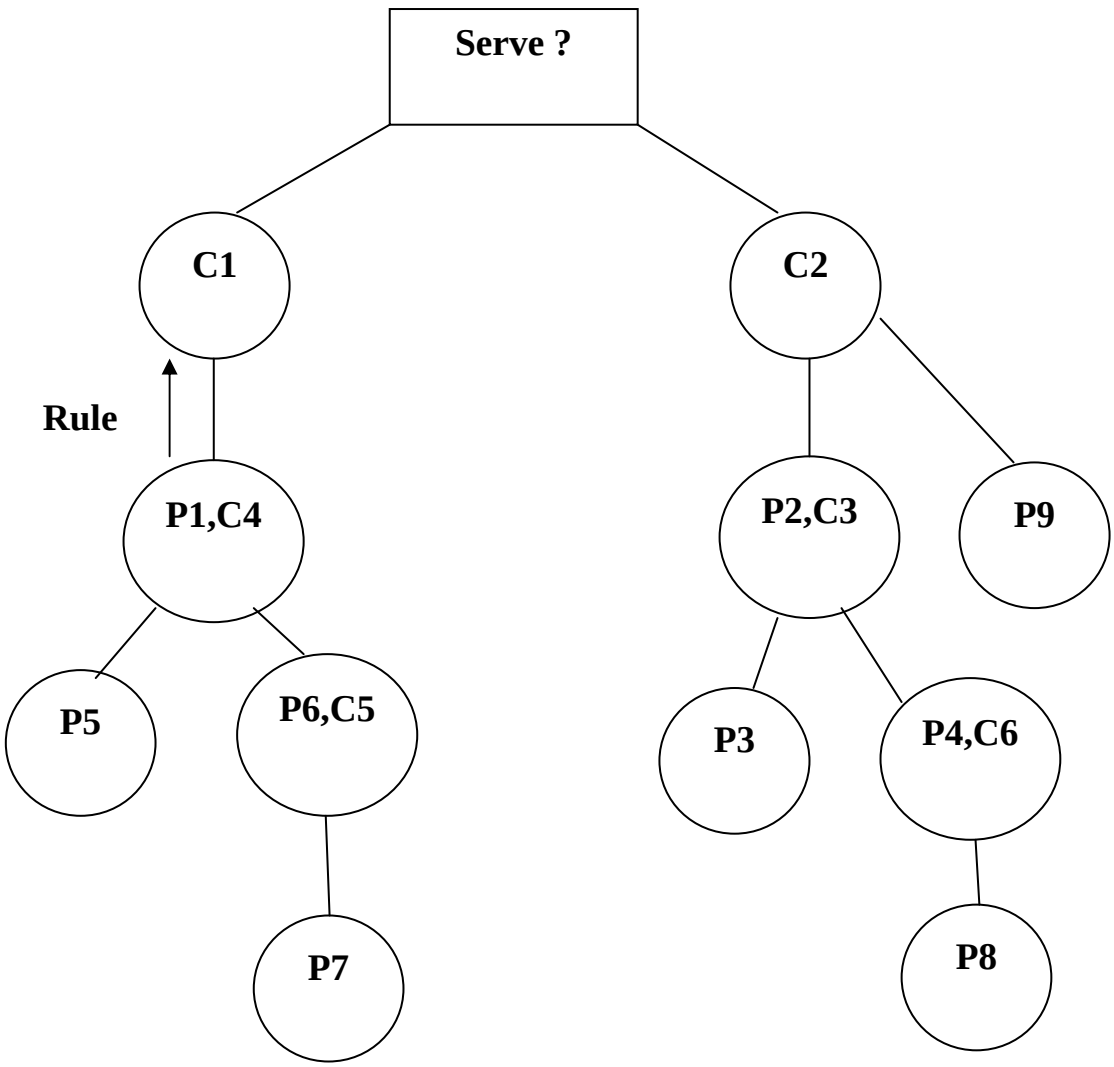
Demon Rule 3 : Evacuate
IF situation =Emergency
THEN Response =Evacuate personnel

اگر موقعیت اضطراری است پرسنل را مطلع ساز

این برنامه در هر لحظه در حال اجرا می باشد یعنی در هر لحظه فشار را چک می کند و وضعیت ها را بررسی‌می نماید و اگر موقعیت را اضطراری تشخیص داد توصیه یا عملی را انجام دهد .مانند آنتی ویروس‌ها یا Fire Alarm

تکنیک‌های پایه جستجو

فضای مساله :فضای مساله می تواند گراف یا درخت باشد .به طوری که گره ها بیانگر حالت های مساله و شاخه ها بیانگر رابطه بین حالت ها هستند .



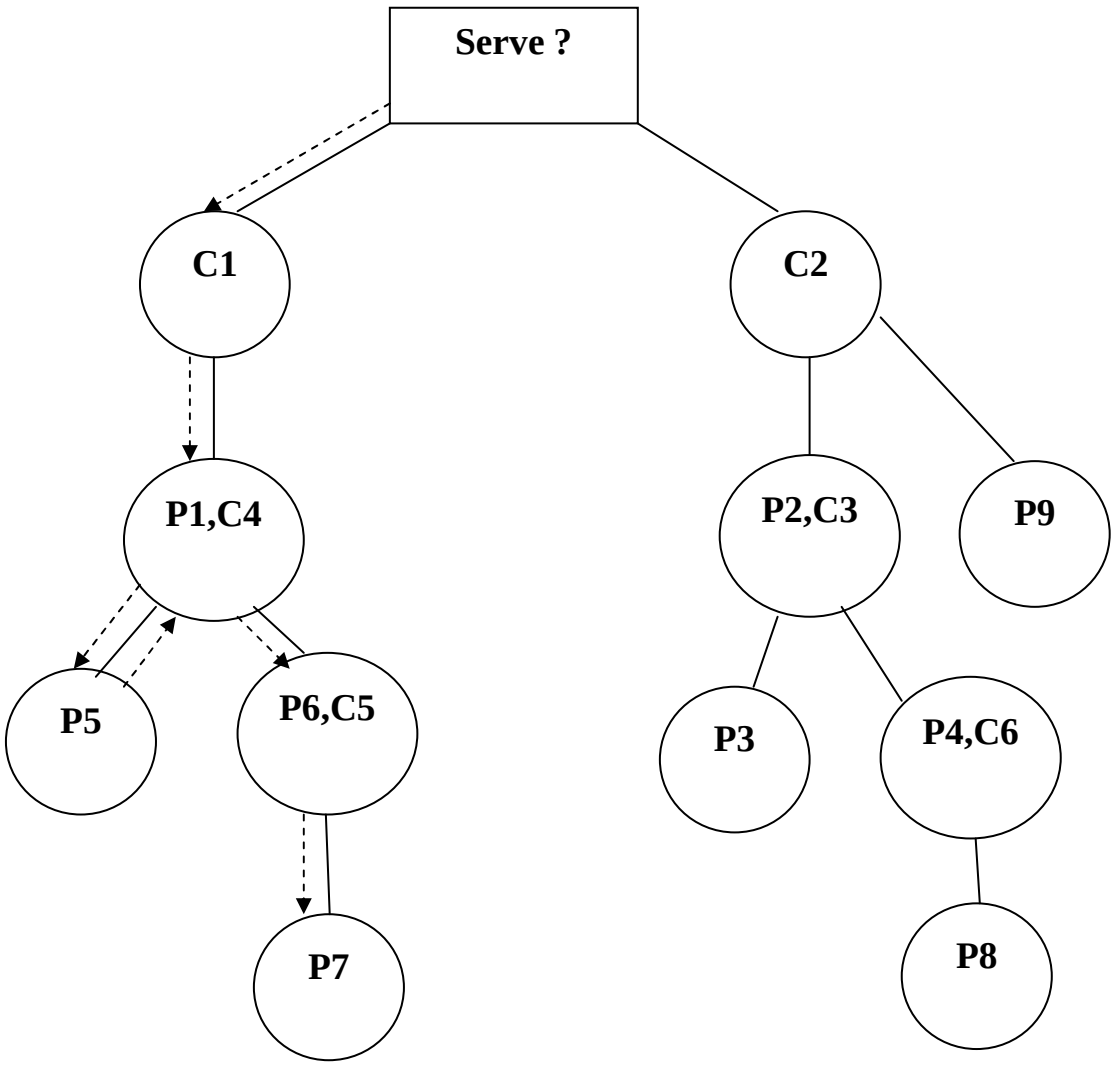
شکل۶-۲

Rule1
IF you purchase meat –P1
THEN you should serve red wine –C1

با مشاهده Rule1 و گراف بالا مشاهده می شود Rule ها به نوعی ارتباط بین گره ها در یک درخت را بیان می کنند. به این درخت ها، درخت های شبکه استنتاج یا Inference network می‌گویند. پس هر کدام از یالها بیانگر یک Rule می‌باشد.

جستجوی عمقی Depth First Search(DFS)

میتوان بر روی فضای مساله که یک گراف است جستجو انجام داد که اولین روش مورد بحث جستجوی عمقی است. در این جستجو یک شاخه را می‌گیریم تا به جواب برسیم اگر نرسیدیم، پسگرد می‌کنیم . روش عمقی در شکل۶-۳ نشان داده شده است.



شکل۶-۳

مزایای جستجوی عمقی

۱. اگر راه حلی موجود باشد یافتن آن را تضمین می نماید.
۲. سریع در مساله عمیق و جزئی می شود.
۳. بر روی یک راه حل خاص تمرکز می‌کند ،سوالات مرتبطی پرسیده می‌شود.

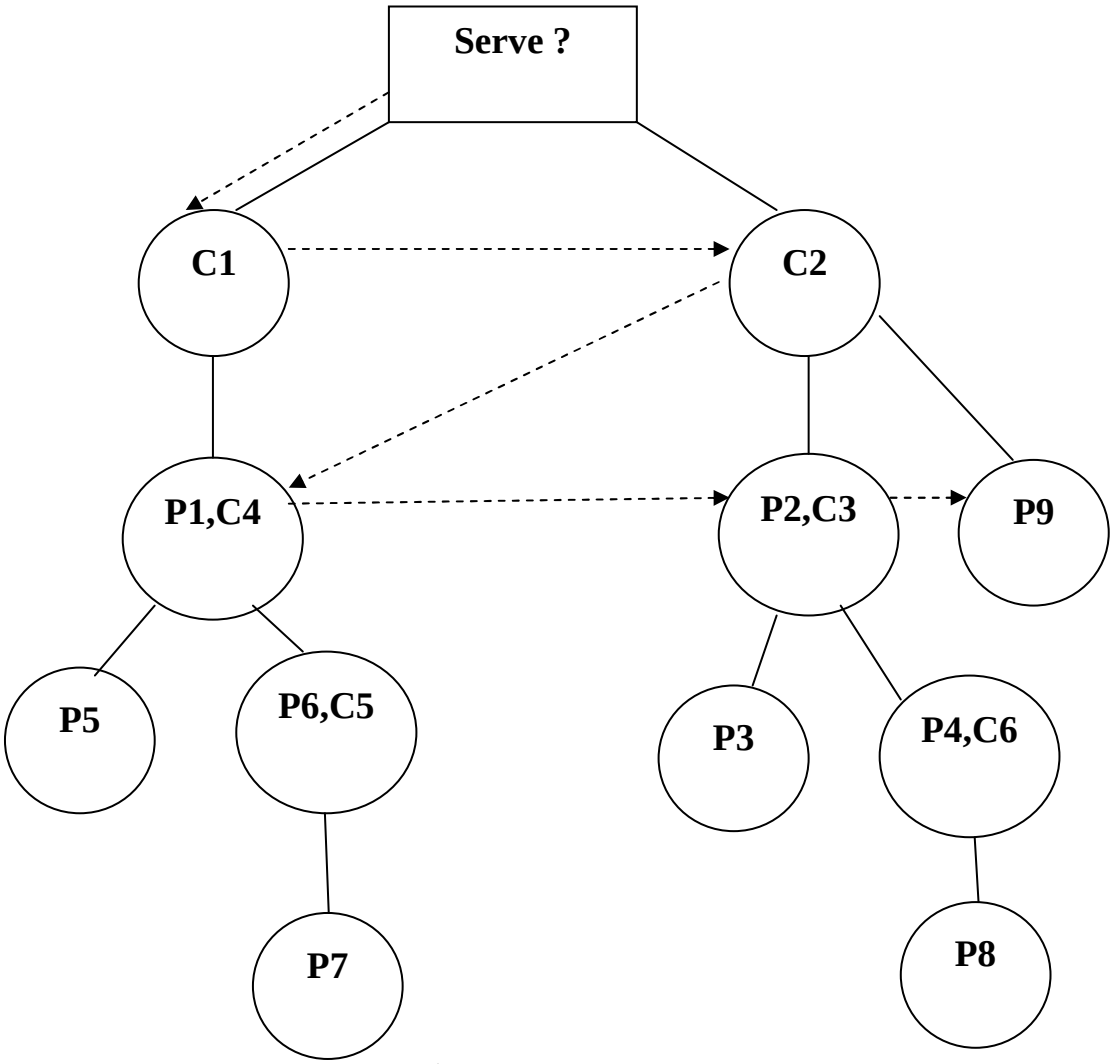
معایب جستجوی عمقی

۱. به دلیل نا آگاهی و جستجوی کورکورانه ، زمان بر است.

۲. برای مسائل با فضاهای بزرگ نامناسب است.

۳. برای مسائلی که راه حل های آنها سطحی است کارایی ندارد.

جستجوی سطحی Breadth First Search(BFS)



شکل۶-۴

مزایای جستجوی سطحی

۱. اگر راه حلی موجود باشد یافتن آن را تضمین می نماید .به علت این که گراف ها دور ندارند .

۲. راه حل های ساده و مسیرهایی با عمق کم را از دست نمی دهند .

معایب جستجوی سطحی

۱. نا آگاهانه است، زیرا در آن از تابع تخمین H استفاده نمی شود.

۲. زمان بر است.

۳. برای مسائل با فضای بزرگ کاربرد ندارد.

٤. رابطه آن با کاربر ضعیف است . زیرا در روش سطحی شاخه های متفاوتی چک می شود پس سوالات پرسیده شده سوالات مرتبط به هم نیستند.

جستجوی اول بهترین

در این روش، از دانش مساله جهت استدلال استفاده کنیم . در هوش مصنوعی آن را به عنوان تابع ارزیابی می شناسیم و در سیستم خبره به عنوان قواعد Heuristic شناخته می‌شوند.

موارد کاربرد قواعدHeuristic

۱. مرتب کردن اهداف

۲. مرتب سازی فرض ها

۳. استفاده از فرا قاعده ها یا Meta Rules

٤. استفاده از قواعد با اولویت

٥. استفاده از ضرایب اطمینان (CF)

مزایای جستجوی اول بهترین

۱. روشی آگاهانه هستند یعنی از دانش حل مساله برای هدایت جستجو استفاده می کنند.

۲. مدلسازی استنتاج انسان گونه

معایب جستجوی اول بهترین

در این روش تضمینی برای یافتن راه حل وجود ندارد .چون ممکن است Heuristic درست نباشد.

استنتاج یکنواخت

استنتاج یکنواخت بدین معناست که در روند استنتاج فرض های مساله تغییر نکند. مثلا برای تشخیص عیب ترانزیستور :

Assertion: Transistor Q1 is bad
Rule: IF Transistor Q1 is bad
THEN Replace Transistor Q1
Conclusion: Replace Transistor Q1

اگر ترانزیستور خراب است باید با یک ترانزیستور سالم جایگزین شود. پس نتیجه گرفته می شود جایگزین شود زیرا در طول استنتاج ترانزیستور سالم نخواهد شد .

استنتاج غیر یکنواخت

اسستنتاج غیر یکنواخت روشی برای استنتاج است که اجازه تغییر در استدلال به دلیل تغییرات در حقایق یا واقعیت ها را می دهد.اکثر استدلال های انسانها غیر یکنواخت است .

Assertion: It is raining –FACT1
Rule: IF It is raining
THEN carry an umbrella
Conclusion: Carry an umbrella—FACT2

اگر باران می آید چتر بردار .حال ممکن است تا می‌آید بیرون، باران قطع شود.

سیستم های خبره قانون پایه

تولید (Production)

تولید واژه‌ای است که در روانشناسی شناختی ، برای تشریح روابط بین وضعیت‌ها و عمل‌ها استفاده می‌شود و معمولا متناظر با مفهوم قانون(قاعده) در سیستمهای خبره قانون پایه است.

شکل کلی تولید :

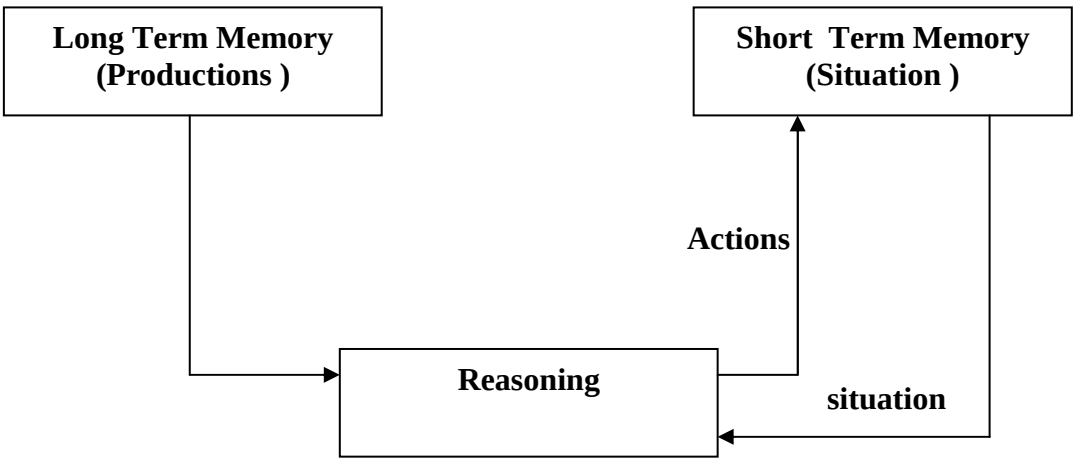
ANTECEDENT(شرایط) → CONSEQUENT(نتایج)

Situation → Action

If situation Then Action

در مدل پایین سه مولفه مشاهده می شود.

- Long Term Memory (LTM)
- Short Term Memory (STM)
- Reasoning



شکل۷-۱: معماری یک سیستم تولید

STM ها حاوی situation ها یا توصیف محیط حل مساله می باشند .

LTM ها حاوی تولیدات می باشند؛ یعنی هر وضعیتی منجر به چه عملی خواهند شد .

پس از استدلال ، حاصل استدلال Action ای است که به STM (حافظه کوتاه مدت) می رود به طور مثال فکری که برای انجام کاری در سر می‌پرورانید، در حافظه کوتاه مدت می باشد .

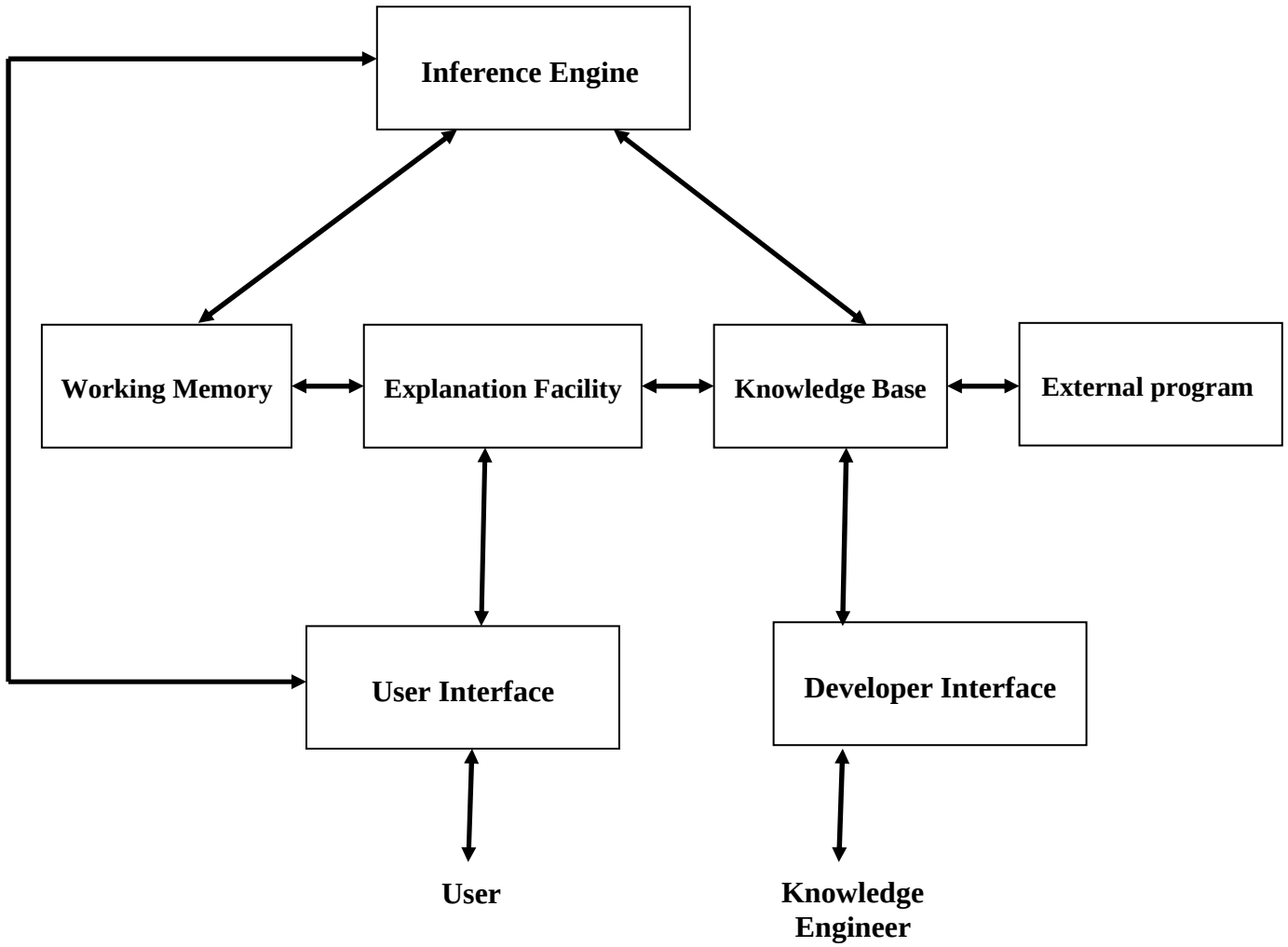
نکته: Action در اینجه بیشتر به طرح عمل اشاره دارد تا خود عمل .

معمار ی سیستم‌های قانون پایه

متناظر مفهوم سیستم‌های تولید در روانشناسی، سیستم‌های قانون پایه یا مبتنی بر قاعده در طراحی سیستم‌های خبره است.
این معماری شامل ۴ مولفه است که به ترتیب زیر است :

۱. واسط کاربر User interface (بخشهایی که با کاربر در ارتباطند مانند فرم ها و جدول ها و ...)
۲. واسط توسعه دهنده Developer interface (برنامه نویس)
۳. تسهیلات توزیع Explanation facility (چراب به این نتیجه رسیده اند)
٤. برنامه های خارجی External programs (مانند پرونده های بیمارها و پایگاه های داده و ...)

نمودار معماری سیستم های قانون پایه



شکل۷-۲: معماری سیستم های قانون پایه

مزایای سیستم های قانون پایه

۱. بیان طبیعی
۲. جداسازی کنترل از دانش
۳. پیمانه ای بودن دانش
٤. سادگی در گسترش و توسعه
٥. رشد نسبی هوشمندی
٦. استفاده از دانش مرتبط برای حل مساله
۷. به کارگیری از امکانات توضیح
۸. بررسی سازگاری
۹. امکان استفاده از دانش Heuristic
۱۰. استفاده از دانش غیر قطعی
۱۱. استفاده از متغیر ها

معایب سیستم های قانون پایه

۱. نیاز به تطبیق دقیق :

به مثال زیر توجه کنید :

The motor is running hot
The motor's temperature is hot

همانطور که مشاهده می شود دو جمله معانی و مفاهیم یکسانی دارد ولی به علت این که از واژگان متفاوتی استفاده کرده اند برای سیستم خبره‌ای، ممکن است متفاوت یا ناشناخته باشد.

۲. داشتن روابط مبهم

IF C THEN D , IF B THEN C , IF A THEN B

۳. محاسبات ممکن است کند باشد .

٤. **برای هر مساله ای مناسب نیست و به کار نمی رود.** (هم عیب است و هم ویژگی) امروزه طراحان برنامه نویس یافته اند که سیستم های تولید به سادگی می توانند مسائل حوزه خاصی را مدلسازی کنند اما برای سایر مدلها مناسب نیستند

٥. **قانون فقط یکی از روشهای بازنمایی دانش است** و هر دانشی را نمی توان با قانون (rule) بازنمایی کرد .

سیستم های قانون پایه با زنجیره پس رو

قوانین هدف یا Goal Rules

قوانین هدف، قوانینی هستند که هدف آنها می تواند اثبات شود اگر یکی از اهداف قانون فعال شود ، یک قانون هدف مشابه هر قانون دیگر هنگامی فعال می شود که شرایط آن درست باشد ، شرایط این قانون ممکن است خود هدف قانون دیگری باشد ، این روند ادامه پیدا می کند تا به شرایطی برسیم که توسط هیچ قانون دیگری پشتیبانی نمی شود، به این شرایط ، شرایط اولیه یا primitive گفته می شود.

Primitive فرضی است که نتیجه هیچ قانونی نیست .و اگر به این قسمت رسیدیم باید یا محیط مساله یا حافظه کاری را بررسی کنیم . اما اگر primitive نباشد باید پایگاه دانش بررسی شود زیرا خود آن قانون هدف یا نتیجه یک قانون بوده است .

مثال : بهتر است در این گونه مسایل موضوع را به سیستم های کوچکتری تقسیم نموده و برای هر کدام سیستم خبره ای طراحی نماییم :

- **Determine if the patient has meningitis**
- **Determine a prescription that should be given to a patient with meningitis**
- **Formulate a different prescription if the recommended one is not acceptable to the user**

- معین کندآیا بیمار مننژیت دارد
- معین کند که چه توصیه ها و تجویزی را به آن بیماری داشته باشد .
- اگر بیمار نپذیرفت جایگزینی برای توصیه خود داشته باشد .

این قسمت ها هدف های نهایی هستند که باید در نهایت طراحی شود.

به مثال زیر توجه کنید :

```
Goal 1 : Infection  is meningitis
Rule 1 : meningitis Infection
        IF          the phyician  know th patient has meningities
        OR          We suspect meningitis
        THEN        Infection is meningitis
        AND Display  Infection
Note : information to be displayed must be designed and acceaaed by this statement
        ELSEDisplay No infection found

Rule2 : Suspect Meningitis from tests or symptoms
        IF          We suspect meningitis from test results
        OR          We suspect meningitis from patient symptoms
        THEN        We suspect meningitis

Rule3 : We suspect meningitis from tests
        IF          tests were run
        AND cultures were seen
        AND         cultures look like meningitis
        THEN        We suspect meningitis from test results

Rule 4 : Cultures look like meningitis
        IF          The appearance of the culture is coccus
        AND The stain of the culture is grampos
        THEN        cultures look like meningitis

Rule5 ; We suspect meningitis from symptoms
        IF          the patient is suffering persistent headaches
        AND the patient is suffering dizziness
        AND the patient has been lethargic
        THEN        we suspect meningitis from patient symptoms
```

پایگاه دانش این مثال از ۵ قانون تشکیل شده ، قانون اول برای تشخیص عفونت است ، قانون دوم مربوط به از روی آزماایشات و نشانه ها به وجود مننژیت گمان برده شود ، قانون سوم با آزمایشات حدس به وجود مننژیت زده شود .

قانون چهارم مربوط به علایم مشابه مننژیت و قانون پنجم مربوط به نشانه های بیماری است .

با به راه انداختن برنامه حافظه کاری یا working memory خالی است .

Working memory

به علت اینکه قرار است Backward کار کند ،چون باید دنبال قانونی بگردد که قسمت THEN آن با هدفی که ما به دنبال آن هستیم طابقت داشته باشد ، ابتدا قسمت THEN قانون اول فعال می شود . قسمت THEN قانون اول می گوید عفونت مننژیت است و هدف این قانون تشخیص عفونت مننژیت است .

در گام دوم بررسی می شود آیا شرایط قانون یک در حافظه کاری هست یا نه که چون حافظه کاری خالی است جواب منفی است .

در گام سوم به دنبال این می گردد که آیا این فرض در قسمت then هیچ قانونی هست؟ پاسخ منفی است .

گام چهارم : این فرض آیا یک قانون اولیه است ؟ بله ، چون Then هیچ قانونی نیست .

حال سیستم می پرسد : آیا شما می دانید بیمار مننژیت دارد ؟

ای سوال برای این است که سیستم متوجه شود آیا پزشک از وجود مننژیت مطلع است ؟ چون اگر مطلع باشد ادامه این کار مناسب نیست .

کاربر پاسخ منفی می دهد ، یعنی هیچ اطلاعی وجود ندارد .

پس قانون اول غلط بود .

Working memory
Patient known to have meningitis _false

در گام پنجم به فرض اول نگاه می کند : این فرض می گوید :

پزشک می داند بیمار مننژیت دارد که این قسمت غلط بود حالا به بخش OR می رویم ، این قسمت می گوید حدس به مننژیت زده شود .و چون حدس به مننژیت اثبات نشده ، شرط آن در حافظه کاری نیامده است .

در گام ششم می گوید آیا قسمت or قانون اول then هیچ قانونی هست ؟

پاسخ مثبت است ، then قانون دوم شرط قانون اول است

در گام هفتم : به اولین شرط قانون شماره ۲ نگاه می کند و می بیند آیا شرایط قانون ۲ در حافظه کاری قرار دارد یا نه ؟

جواب منفی است ، چون در حافظه کاری اطلاعاتی مرتبط با موضوع جستجو نداریم .

در گام هشتم : بررسی می کنیم آیا شرایط قانون شماره ۲ در قسمت then هیچ قانون دیگری هست ؟

پاسخ مثبت است ، قانون شماره ۳ . در قانون شماره ۳ شرایط بدین صورت است که آزماش انجام شود ، به همین دلیل سیستم می پرسد کآیا آزمایش انجام داده اید ؟

و کاربر پاسخ مثبت می دهد .

در گام دهم سیستم می پرسد آیا علایم بیماری دیده شده اند ؟ کاربر پاسخ می دهد : بله .

در گام یازدهم دوباره سیستم می پرسد آیا این علایم مشابه ویروس coccus هست ؟

کاربر از سیستم می پرسد که چرا این سوال را پرسیده است ؟

سیستم پاسخ می دهد این سوال ها به تعیین ای که آیا این علایم مشابه مننژیت هست یا خیر ؟و با دانستن این اطلاعات سیستم می تواند قانون شماره ۴ را بررسی کند .

به قانون شماره ۴ می رود ، که اگر علایم با آن ویروس خاص مشابه باشد ، علایم به مننژیت نزدیک است .

در گام دوازدهم :دوباره دلیل پرسیده می شود ؟ و در پاسخ سیستم می گوید با دانستن این اطلاعات می تواند قانون شماره ۳ را بررسی کند .

در گام سیزدهم : سیتم سوال می پرسد که آیا علایم همان علایم coccus است ؟

و کاربر پاسخ منفی می دهد.

چون جواب منفی است سیستم به بررسی شرط دوم قانون ۲ می پردازد .

Working memory
Patient known to have meningitis _false
Tests run – True
Cultures seen – True
Appearance of culture is coccus -- False

در گام چهاردم می پرسد آیا شرط دوم قانون دوم در حافظه کاری هست ؟

پاسخ منفی است .

در گام پانزدهم بررسی می کند که شرط دوم قانون دوم Then کدام قانون است ؟ قانون پنجم .

در گام شانزدهم : شرایط قانون ۵ را بررسی می کنیم ، هیچکدام از شرایط قانون ۵ شناخته شده نیست ، پس سوال می پرسد :

آیا بیمار سر درد دارد ؟ بله

آیا سرگیجه دارد ؟ بله

آیا بیمار سست و بیحال است ؟ بله

در گام هفدهم :وقتی همه این قسمت ها مثبت بود قانون شماره ۵ به تتیجه رسیده است . و نتیجه می گیرد که ما می توانیم از روی علایم به مننژیت پی ببریم . و چون شرایط قانون پنجم then هیچ قانونی نیست پس primitive است .

Working memory
Patient known to have meningitis _false
Tests run – True
Cultures seen – True
Appearance of culture is coccus – False
Patient has heaadaches—True
Patient has dizziness—True
Patient has been lethargic – True
Suspect meningitis from symmtoms—True
We suspec meningitis—True
Infection is meningitis-- True

سیستم های تولید

تولید

یادآوری: تولید واژه‌ای است که در روانشناسی شناختی ، برای تشریح روابط بین وضعیت‌ها و عمل‌ها استفاده می‌شود.

If Situation **Then** Action

تاریخچه و مرورتاریخی سیستم‌های تولید

سیستم تولید پست

Post: Proved in 1943 that any system of mathematics or logic can be described as a certain type of production system. Post production systems have no rule ordering or any form of control strategy.

اولین بار توسط پست، در سال ۱۹۴۳ معرفی شد. موضوع شگفت انگیز در کار پست؛ هر سیستم ریاضی و یا سیستم منطقی را می‌توان به نوع خاصی از سیستم‌های تولید، تبدیل کرد.

مثال:

Unordered rule base.

$ba \rightarrow ab$

$ca \rightarrow ac$

$cb \rightarrow bc$

	Working mem.	Conflict set	Fired
0	cbaca	1,2,3	3
1	bcaca	2	2
2	bcaac	2	2
3	bacac	1,2	1
4	abcac	2	2
5	abacc	1	1
6	aabcc	{}	Halt

شکل۸-۱: سیستم تولید پست

پرسش: به نظر شما، با توجه به شناخت و درکی که از سیستم‌های خبره تا کنون یافته‌اید، در ایده پیشنهادی توسط پست؛ چه چیزی فراموش‌شده‌است؟

الگوریتم مارکوف

Markov: Introduced rule ordering to production systems, the Markov algorithm.
مارکوف با ایجاد ساختار کنترلی، گام بعدی را در ایجاد سیستم‌های تولید برداشت. الگوریتم مارکوف، مجموعه مرتب‌شده از قوانین تولید است که هر یک بر اساس میزان اولویت، روی رشته ورودی اعمال‌می‌شوند.

پرسش: الگوریتم مارکوف، در چه حالت‌هایی متوقف‌می‌شود؟

مثال

Ordered rule base.
1. $ba \rightarrow ab$
2. $ca \rightarrow ac$
3. $cb \rightarrow bc$

	Working mem.	Conflict set	Fired
0	cbaca	1,2,3	1
1	cabca	2	2
2	acbca	3,2	2
3	acbac	1,3	1
4	acabc	2	2
5	aacbc	3	3
6	aabcc	{ }	Halt

شکل۸-۲: سیستم تولید با الگوریتم مارکوف

الگوریتم رته

پرسش: الگوریتم مارکوف، چه ویژگی‌ها و مزایایی دارد؟ چه محدودیت‌هایی دارد؟

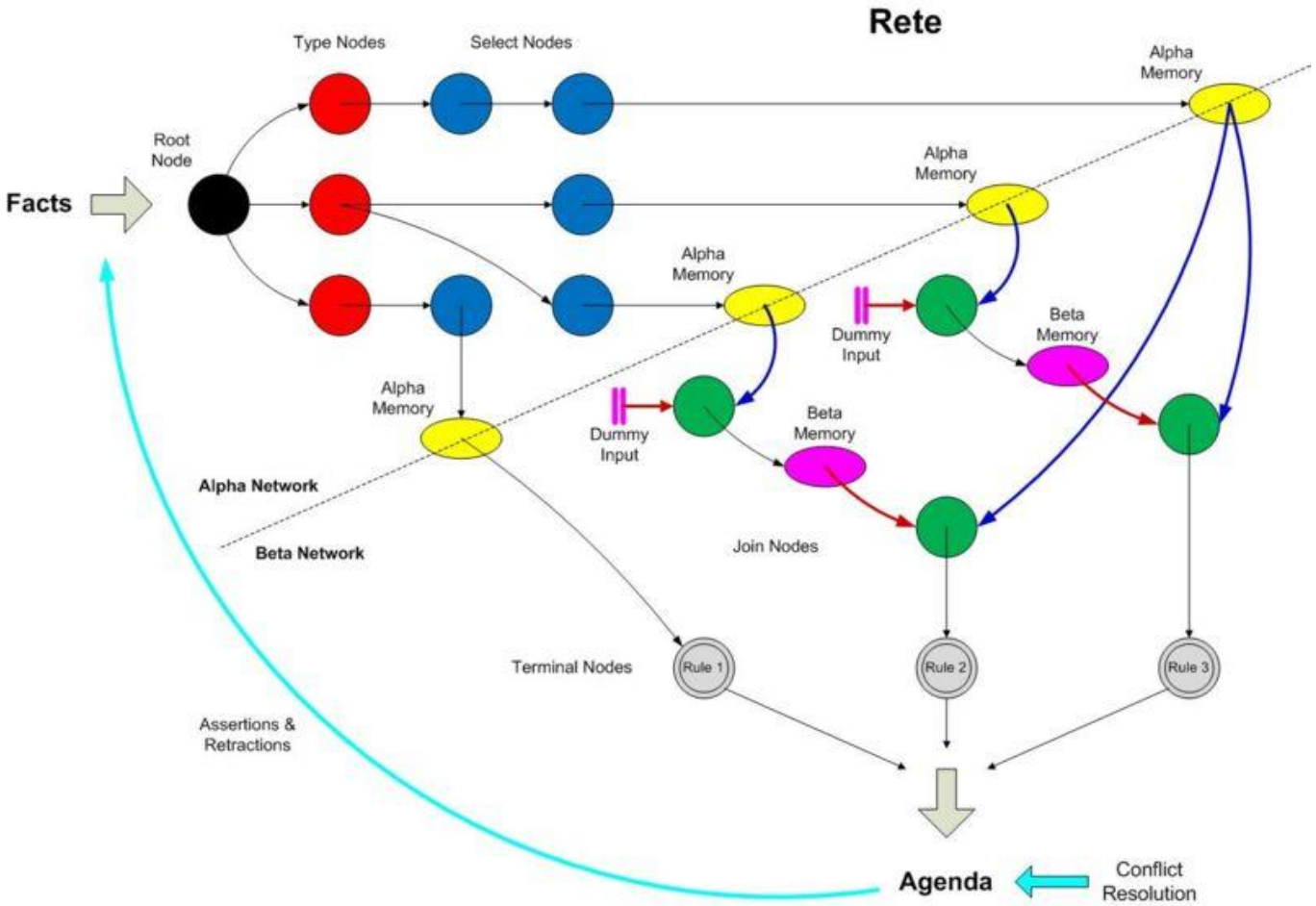
Forgy: Introduced pattern matching to detect changes in matches, the Rete algorithm.
Compiles rule memory into networks that avoid duplication between rules and over time, as the network is updated only when the rule base is changed.
در سال ۱۹۷۹، چارلزفوجی از دانشگاه کارنگی ملون، الگوریتم رته را پیشنهاد داد که منجر به OPS شد.

OPS = Official Production System

OPS → OPS2 → OPS3→ OPS4 → OPS5 → OPS89 → … CLIPS

اطلاعات مربوط به قواعد(قانون‌ها یا همان تولیدها) در قالب شبکه، ذخیره‌می‌شود. بنابراین Rete یک گراف جهت‌دار، حلقوی و ریشه‌دار است. هر مسیر از ریشه تا برگ، سمت چپ قوانین است و هر گره، جزئیاتی که بر قوانین، منطبق هستند، ذخیره‌می‌کند.

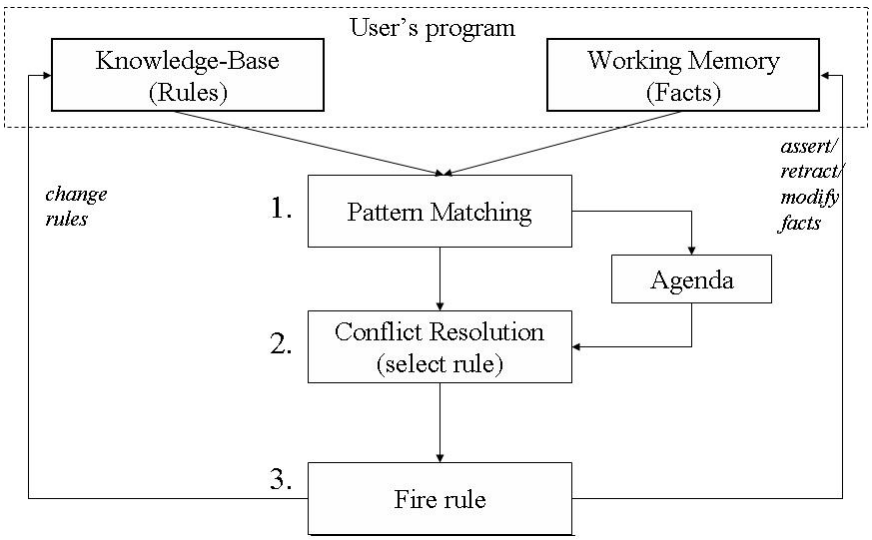
الگوریتم رته به جای سیکل تشخیص واجرای همه قواعد، تنها به بررسی قواعدی می‌پردازد که تغییرکرده‌اند. به محض اینکه واقعیت‌ها عوض‌می‌شوند، واقعیت‌های جدید در رته از ریشه به سمت، برگ‌ها پخش‌می‌شوند.



شکل۸-۳: گراف رته

نکته: CLIPS یکی از پوسته‌هایی است که از الگوریتم رته، در موتور استنتاج خود بهره می‌گیرد.

CLIPS = C Language Integrated Production System
Design using the C programming language at NASA/Johnson Space Center with the specific purpose of providing high portability, low cost, and easy integration with external systems.



شکل۸-۴: چرخه استنتاج در CLIPS

مثال: با فرض پایگاه دانش و حافظه‌کاری، شبکه رته را آنالیز کنید.

Rule Memory:

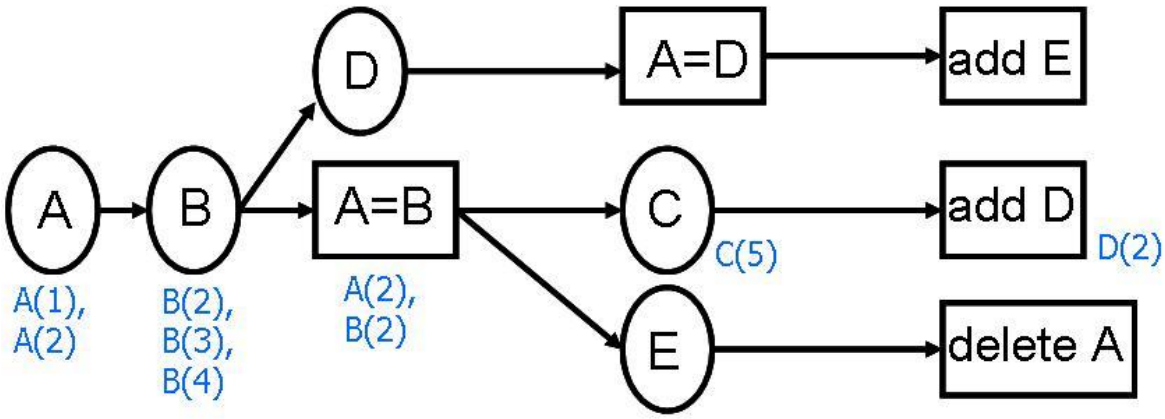
$A(x) \wedge B(x) \wedge C(y) \Rightarrow$ add D(x)

$A(x) \wedge B(y) \wedge D(x) \Rightarrow$ add E(x)

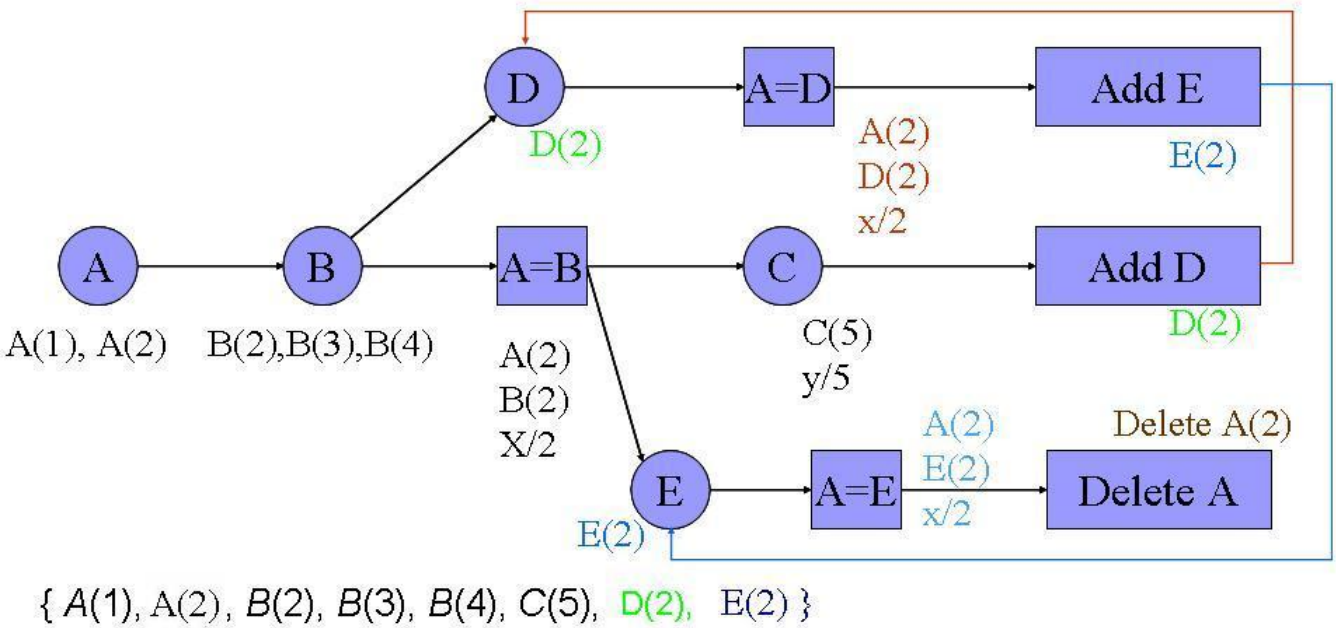
$A(x) \wedge B(x) \wedge E(x) \Rightarrow$ delete A(x)

Working Memory:

{A(1), A(2), B(2), B(3), B(4), C(5)}



شکل۸-۵: شبکه رته

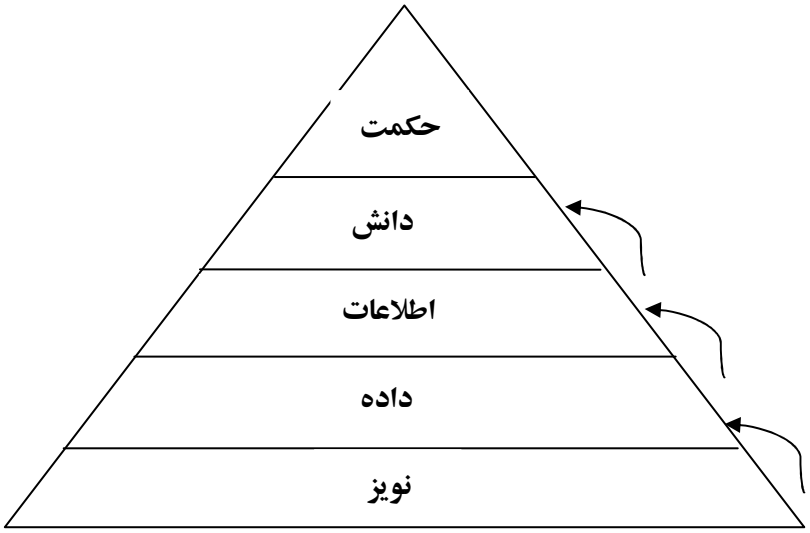


شکل۸-۶: تطبیق رته

تمرین: تحقیق کنید که الگوریتم رته، چه مشکلات و محدودیت‌هایی ممکن است، داشته باشد.

استخراج دانش

در دو دهه قبل توانایی‌های فنی بشر برای تولید و جمع آوری داده‌ها به سرعت افزایش یافته‌است. عواملی نظیر استفاده گسترده از بارکد برای تولیدات تجاری، به خدمت‌گرفتن کامپیوتر در کسب و کار، علوم، خدمات دولتی و پیشرفت در وسائل جمع‌آوری داده، از اسکن‌کردن متون و تصاویر تا سیستم‌های سنجش از دور ماهواره‌ای، در این تغییرات نقش مهمی دارند.



شکل۹-۱: هرم دانش

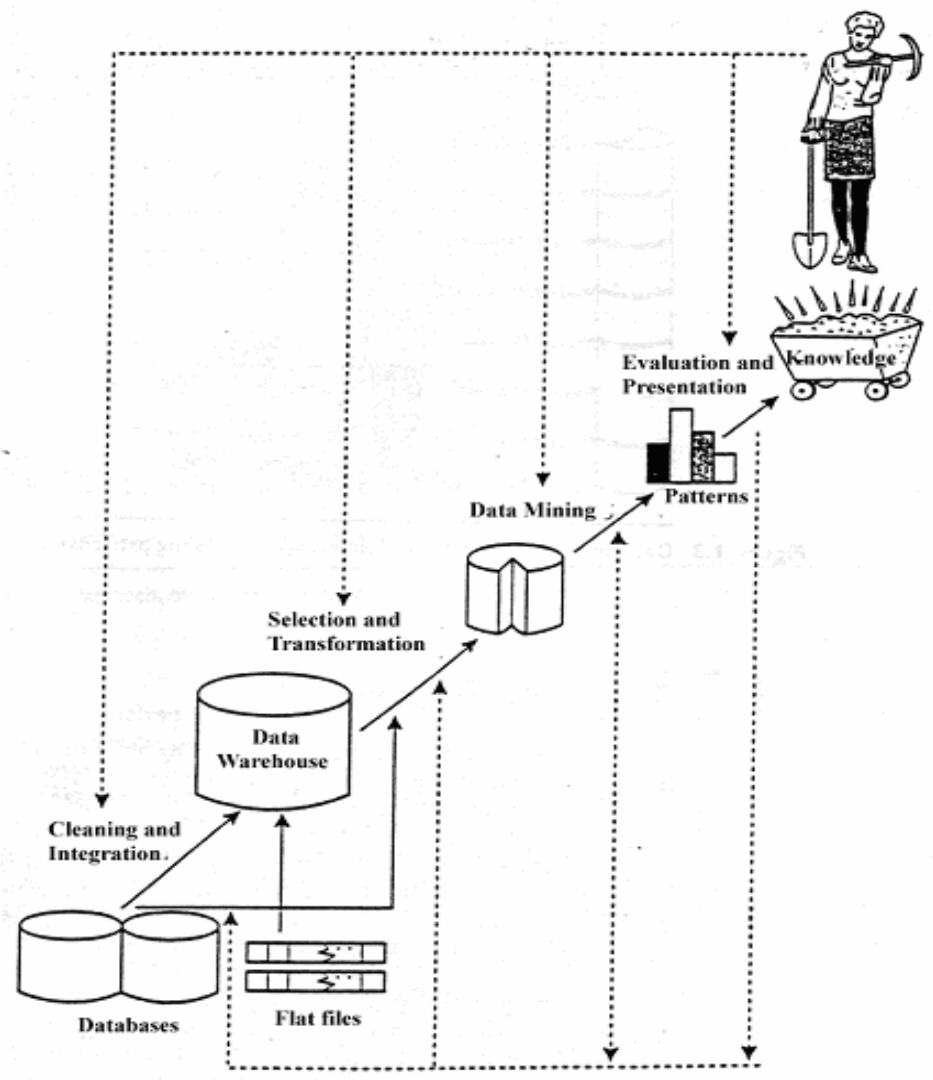
استخراج دانش نحوه تبدیل داده به دانش است.

- اگر این کار به صورت مهندسی انجام‌شود، اکتساب دانش خواهد بود (فرم‌های خاصی پر شود یا نمودارهایی کشیده شود) اکتساب دانش برای هر مساله‌ای امکان پذیر نیست.
- راه حل دیگر داده کاوی یا استخراج دانش است یعنی از داده‌ها، به صورت خودکار دانش، استخراج شود. داده کاوی، استخراج خودکار دانش از حجم زیاد داده‌هاست.

واژه های «داده‌کاوی» ، «کشف دانش در پایگاه داده» و «استخراج دانش از پایگاه داده» اغلب به صورت مترادف یکدیگر، مورد استفاده قرارمی‌گیرند. استخراج دانش به عنوان یک فرآیند در شکل۹-۲، نشان داده شده است. استخراج دانش از پایگاه داده، فرایند شناسایی درست، ساده، مفید، و نهایتا الگوها و مدلهای قابل فهم در داده‌هاست. داده‌کاوی، مرحله ای از فرایند کشف دانش می باشد و شامل الگوریتم‌های مخصوص داده کاوی است، بطوریکه، تحت محدودیتهای مؤثر محاسباتی قابل قبول، الگوها و یا مدلها را در داده کشف می‌کند. به بیان ساده‌تر، داده‌کاوی به فرایند استخراج دانش ناشناخته، درست و بالقوه مفید از داده اطلاق می‌شود. تعریف دیگر اینست که، داده کاوی گونه‌ای از تکنیک‌ها برای شناسایی اطلاعات و یا دانش تصمیم‌گیری از قطعات داده می‌باشد، به نحوی که با استخراج آنها، در حوزه‌های تصمیم گیری، پیش بینی، پیشگویی، و تخمین مورد استفاده قرار گیرند. داده‌ها اغلب حجیم ، اما بدون ارزش می باشند، داده به تنهایی قابل استفاده نیست، بلکه دانش نهفته در داده ها قابل استفاده می باشد. به این دلیل اغلب به داده کاوی، تحلیل داده‌ای ثانویه گفته می‌شود.

چه چیزی سبب پیدایش داده کاوی شده است؟

اصلی ترین دلیلی که باعث شد داده کاوی کانون توجهات در صنعت اطلاعات قرار بگیرد، مساله در دسترس بودن حجم وسیعی از داده ها و نیاز شدید به اینکه از این داده‌ها اطلاعات و دانش سودمند استخراج کنیم. اطلاعات و دانش بدست آمده در کاربردهای وسیعی از مدیریت کسب و کار وکنترل تولید و تحلیل بازار تا طراحی مهندسی و تحقیقات علمی مورد استفاده قرار می‌گیرد.



شکل۹-۲: فرایند استخراج دانش یا داده کاوی

داده کاوی چه کارهایی نمی تواند انجام دهد؟

داده کاوی فقط یک ابزار است و نه یک عصای جادویی. داده کاوی به این معنی نیست که شما راحت به کناری بنشینید و ابزارهای داده کاوی همه کار را انجام دهد. داده کاوی نیاز به شناخت داده ها و ابزارهای تحلیل و افراد خبره در این زمینه ها را از بین نمی برد. داده کاوی فقط به تحلیلگران برای پیدا کردن الگوها و روابط بین داده ها کمک می‌کند و در این مورد نیز روابطی که یافته‌می‌شود باید به وسیله داده‌های واقعی دوباره بررسی و تست گردد.

نکته: داده کاوی: استخراج الگوها یا دانش از حجم زیادی داده است.

تمرین: تحقیق کنید که چه روش‌ها و تکنولوژی‌هایی می‌تواند در داده‌کاوی، استفاده شود.

تصمیم گیری

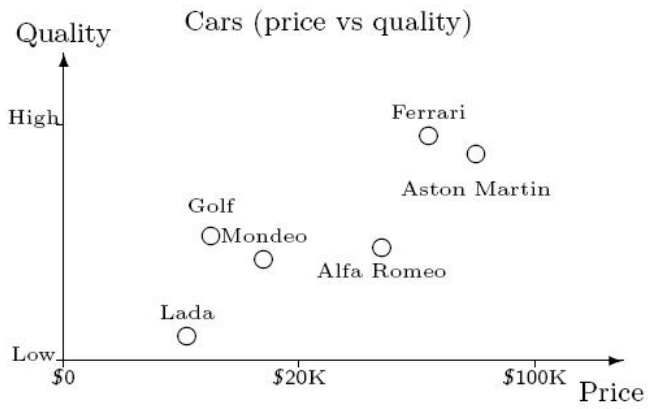
پرسش: تصمیم چیست؟ تصمیم‌گیری به چه معناست؟ تصمیم‌گیری چه نقشی در طراحی سیستم‌های خبره دارد؟

تابع سودمندی(Utility Function)

$$a \gtrsim b \quad \text{if and only if} \quad u(a) \geq u(b)$$

تصمیم گیری چند معیاره (Multi-criteria Decision Making)

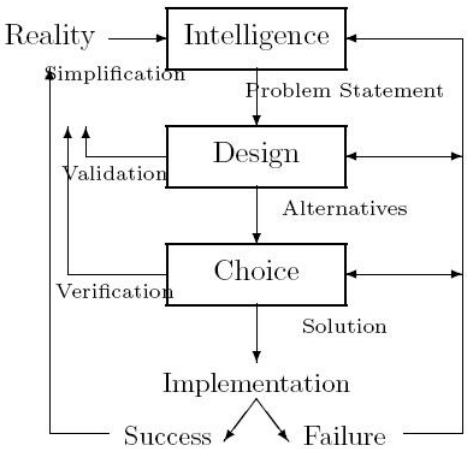
چندین معیار در تصمیم‌گیری، تاثیر دارد.



شکل ۱۰-۱: مثال انتخاب خودرو

پرسش: در تصمیم گیری چندمعیاره، تابع سودمندی چیست؟

سه فار مهم در تصمیم گیری(از نگاه سایمون)



شکل ۱۰-۲: سه فاز تصمیم‌گیری

درخت تصمیم گیری

در بسیاری از مواقع اکتساب دانش از فرد خبره سخت و مشکل است؛ این مساله بدین دلیل است که وقتی میزان خبرگی بالا می‌رود، فرد خبره، مسایل و جزئیاتی را بدیهی فرض‌می‌کند، درحالیکه مهندس دانش نیاز به این جزئیات، برای طراحی سیستم خبره دارد. به عبارت ساده، در این موارد فرد خبره با سوال و جواب، پاسخ های مناسبی به مهندس دانش، نمی‌دهد. راه حل یا راه حلها کدامند؟

در این موارد فرد خبره نمی‌تواند بگوید مساله را چگونه حل نموده است و مهندس دانش، خود باید راه حل مساله را با روش استقرا، بیابد.

استقراء (Induction)

ایجاد قوانین کلی از مجموعه مثال‌هاست. (مثال‌ها در قالب جداولی نمایش‌داده‌می‌شود که به آنها، جداول‌تصمیم‌گیری، گفته‌می‌شود.) از طریق مثال‌ها، دانش استخراج‌می‌شود و به همین دلیل استقرا یک نوع یادگیری است.

دانش استخراج شده درقالب درختی به نام، درخت تصمیم‌گیری، بازنمایی‌خواهدشود.

نکته: درخت تصمیم‌گیری؛ روشی برای نمایش یک سری از قوانین هستند که منتهی به یک رده یا مقدار می‌شوند.

روند کلی: از روی جدول تصمیم‌گیری(مثال‌ها)، درخت تصمیم‌گیری، ساخته‌شود و از روی درخت، قوانین‌کلی استخراج می‌شود. برای ایجاد درخت تصمیم‌گیری، از مفهوم آنتروپی استفاده‌می‌شود .

پرسش: اگر سکه‌ای، یک میلیون بار بیندازیم، چنانچه این نتایج را بخواهیم برای فرد دیگری ارسال کنیم، به چه تعداد بیت نیاز خواهد بود؟

برای پاسخ به این سوال دو حالت زیر را در نظر بگیرید: ۱) سکه سالم باشد ۲) سکه ناسالم باشد

حالت اول : سکه سالم باشد

احتمال وقوع خط = احتمال وقوع شیر (در هر پرتاب)

تعداد بیتها جهت نمایش وضعیت هر پرتاب = ۱ بیت

تعداد کل بیتهای مورد نیاز = 1000 ‘000 bits

حالت دوم: سکه ناسالم باشد

فرض کنید احتمال شیرآمدن 1/1000 و احتمال خط آمدن 999/1000 باشد.

در یک میلیون پرتاب انتظار داریم ۱۰۰۰ بار شیر بیاید.

پرسش: چگونه راه حل را می‌توان بهتر کرد؟

نکته: دنباله نمایش سکه های ناسالم، حاوی اطلاعات کمتری نسبت به دنباله ی نمایش سکه های سالم است، لذا به بیتهای کمتری برای انتقال نیاز دارد.

محتوای اطلاعاتی (Information Content): تعداد بیتهای لازم برای ارسال یک دنباله با استفاده از یک کدگذاری بهینه

همیشه می توان از یک کدگذاری که کمتر بهینه است، استفاده کرد. اما افزایش بیتها هرگز به معنای افزایش محتوای اطلاعاتی نیست.

مثال: ۱۰۰۰ بار پرتاب یک تاس ۸ وجهی را در نظر بگیرید:

الف) تاس سالم باشد:

تعداد بیتهای مورد نیاز برای بیان وضعیت پرتاب تاس : ۳بیت

به ارسال ۳۰۰۰ بیت نیاز خواهیم داشت.

$$\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\frac{1}{32}+\frac{1}{64}+\frac{1}{128}+\frac{1}{128}=1$$

A B C D E F G H

ب) تاس ناسالم باشد: فرض کنید احتمال آمدن وجه $i(i=1 \cdots 8)$ به صورت زیر باشد:

فرض کنید کدهای زیر را به هر یک از ۸ حالت تاس اختصاص دهیم:

A ⟶ 0
B ⟶ 10
C ⟶ 110
D ⟶ 1110
E ⟶ 11110
F ⟶ 111110
G ⟶ 1111110
H ⟶ 1111111

$$\text{متوسط بیتهای مورد نیاز برای ارسال} = \sum \text{تعداد بیتهای هر حالت} * \text{نرخ وقوع آن حالت}$$

آنتروپی

فرض کنید دو نفر X و Y وجود دارند که یکی از آنها یک سوال ساده می پرسد و دیگری به آن سوال پاسخ بله یا خیر می دهد.

مقدار اطلاعاتی که در این سوال–جواب ها بین طرفین رد و بدل می شود، آنتروپی نامیده می شود.

فکر می کنید با یک سوال–جواب چه مقدار اطلاعات بدست خواهید آورد؟

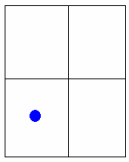
مثال: توپی داخل یکی از دو جعبه قرار دارد. الف : آیا توپ داخل جعبه سمت راست است؟ ب : بله

با یک سوال–جواب می توان فهمید که توپ کجاست. پس می توان گفت که **مقدار اطلاعات ۱ بیت است.**

مثال: الف: آیا توپ داخل جعبه های بالایی است؟ ب : خیر

الف: آیا توپ داخل جعبه سمت راستی است؟ ب : خیر

مقدار اطلاعات ۲ بیت است.



پرسش: ۰ بیت چه چیزی را تعبیر می کند؟ صفر بیت اطلاعات دریافت کرده اید؟!

کلاس‌بندی و آنتروپی

فرض کنید ۵ کلاس A,B,C,D,E را داریم و می خواهیم یک شی را در آنها کلاس بندی کنیم

الف) اگر عدم قطعیت نداشته باشیم:

	A	B	C	D	E
احتمال	0	1	0	0	0

$H(X) = 0\text{Log}(1/0) + 0\ \text{Log}(1/1)+0\text{Log}(1/0)+\cdots$

$= 0$ (عدم قطعیت نداریم)

ب) اگر بین دو کلاس شک داشته باشیم:

	A	B	C	D	E
احتمال	0.5	0.5	0	0	0

$H(X) = 0.5\text{Log}(1/0.5)+0.5\text{Log}(1/0.5)+0\cdots = 1$

یک بیت عدم قطعیت داریم،اگر اطلاعات کلاس را بخواهیم ارسال کنیم به یک بیت نیاز خواهیم داشت.

ج) اگر بین چهار کلاس شک داشته باشیم:

	A	B	C	D	E
احتمال	0.25	0	0.25	0.25	0.25

$H(X) = 0.25\text{Log}(1/0.25)+0\text{Log}(1/0)+\cdots \quad = 2$

د) اگر حالت زیر را بین کلاسها داشته باشیم:

	A	B	C	D	E
احتمال	0.5	0.125	0.125	0.125	0.125

$H(X) = 2.$

اگر بخواهیم اطلاعات را ارسال کنیم به طور متوسط (زمانی) به دو بیت احتیاج داریم.

A = 1 B= 000 , C= 001 , D= 010 , E= 011

نکته: هر چه همه‌ی کلاس‌ها به طوریکنواخت محتمل شوند،H(X) بزرگتر می‌شود؛ یعنی میزان عدم قطعیت در انتخاب کلاس، افزایش‌می‌یابد. **آنتروپی(عدم قطعیت)** و **اطلاعات** دو روی یک سکه اند! در تئوری اطلاعات تنها سمبلهایی که برای گیرنده دارای عدم قطعیت هستند، به عنوان اطلاعات قلمداد می‌شوند. (تنها اطلاعاتی که برای فهمیدن اساسی هستند باید فرستاده شوند.)

مفهوم اطلاعات

تئوری اطلاعات محتوای اطلاعات را بر حسب **بیت** اندازه گیری می کند. وقتی می خواهیم در مورد محتوای اطلاعاتی یک چیز صحبت کنیم مثل اینست که با چند پاسخ آری یا خیر وضعیت آن را مشخص کنیم. وضعیت یک سکه دو برآمد دارد پس با یک بیت می توانیم وضعیت آن را مشخص کنیم. اگر در مورد سکه ۰،۹۹ احتمال شیر باشد پس تقریبا همیشه وضعیت شیر رخ میدهد لذاسوال و جوابی نیاز نیست و محتوای اطلاعاتی به صفر نزدیک می شود.

بهره ی اطلاعاتی(information Gain)

بعد از آزمون یک صفت به چند بیت اطلاعات دیگر نیازمندیم؟ صفت مورد آزمایش چقدر به ما اطلاعات می دهد؟

Gain (A) = Info (I) - Reminder (A) = Info (I) - Info (I,A)

نکته: در استفاده از بهره ی اطلاعاتی باید آن صفت یا attribute ای را انتخاب‌کرد که بیشترین بهره‌ی اطلاعاتی را داراست و معمولا از فرمول نرمال‌سازی‌شده در محاسبه بهره اطلاعاتی استفاده می‌شود.

$$GainRatio(A) = \frac{Info(I) - Info(I,A)}{SplitInfo}$$

Split(A) عدم قطعیت نمونه ها نسبت به صفت A است.

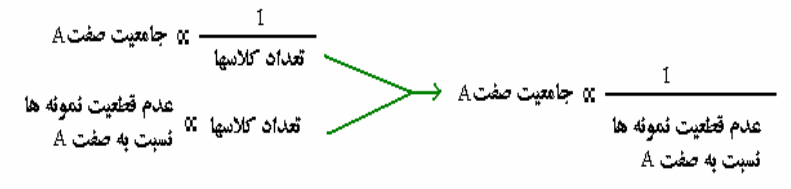
$$Split(A) = -\sum_{j=1}^{k} \frac{\#classj}{\#I} \log_2 \frac{\#classj}{\#I}$$

Split(A) در دو حالت بزرگ می شود:

۱- تعداد شاخه های K(مقادیر صفت A) بیشتر شود.

۲- تعداد نمونه ها در هر کلاس j کم باشد.

بروز دو حالت فوق موجب جامعیت کمتر صفت می شود.



پرسش: چرا از فرمول بهره ی اطلاعاتی به صورت نرمال شده استفاده می کنیم؟

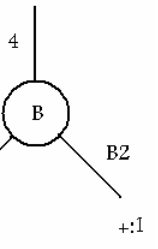
ما دنبال صفت Aای هستیم که : اولا بهره ی اطلاعاتی بالاتری داشته باشد. ثانیا جامعیت کلاس بندی آن بیشتر باشد

مثال: فرض کنید صفت خروجی برای ۵ ویژگی ورودی به صورت زیر باشد. مقدار Gain و Gain Ratio را برای هر صفت بدست آورید.

صفت A	صفت B	صفت C	صفت D	صفت E	صفت خروجی
					+
					+
					+
					−

ابتدا عدم قطعیت نمونه ها نسبت به صفت خروجی را بدست می آوریم

$$H(I) = -\frac{3}{4}\log_2^{\frac{3}{4}} - \frac{1}{4}\log_2^{\frac{1}{4}} = 0.81bit$$



$$H(I,B) = \frac{3}{4}H(B1) + \frac{1}{4}H(B2)$$

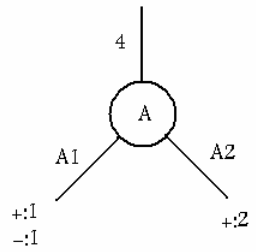
$$H(B1) = -\frac{2}{3}\log_2^{\frac{2}{3}} - \frac{1}{3}\log_2^{\frac{1}{3}} = 0.9183$$

$$H(B2) = 0bit$$

$$Gain(B) = H(I) - H(I,B) \approx 0.1212$$

$$Split(B) = -\frac{3}{4}\log_2^{\frac{3}{4}} - \frac{1}{4}\log_2^{\frac{1}{4}} = 0.81$$

$$GainRatio(B) = \frac{0.1212}{0.81} \approx 0.15$$



$$H(I,A) = \frac{2}{4}H(A1) + \frac{2}{4}H(A2)$$

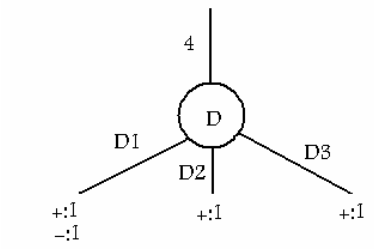
$$H(A1) = -\frac{1}{2}\log_2^{\frac{1}{2}} - \frac{1}{2}\log_2^{\frac{1}{2}} = 1bit$$

$$H(A2) = -\frac{2}{2}\log_2^{\frac{2}{2}} - \frac{0}{2}\log_2^{\frac{0}{2}} = 0bit$$

$$Gain(A) = H(I) - H(I,A) = 0.81 - \frac{1}{2}\times 1 = 0.31$$

$$Split(A) = -\frac{2}{4}\log_2^{\frac{2}{4}} - \frac{2}{4}\log_2^{\frac{2}{4}} = 1bit$$

$$GainRatio(A) = \frac{0.31}{1} = 0.31$$



$$H(I,D) = \frac{2}{4}H(D1) + \frac{1}{4}H(D2) + \frac{1}{4}H(D3)$$

$$H(D1) = 1bit$$

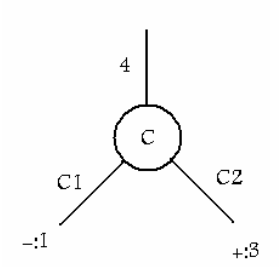
$$H(D2) = 0$$

$$H(D3) = 0$$

$$Gain(D) = 0.81 - 0.5 = 0.31$$

$$Split(D) = -\frac{2}{4}\log_2^{\frac{2}{4}} + \frac{1}{4}\log_2^{\frac{1}{4}} + \frac{1}{4}\log_2^{\frac{1}{4}} = 1.5$$

$$GainRatio(D) = \frac{0.31}{1.5} = 0.201$$



$$H(I,C) = \frac{3}{4}H(C1) + \frac{1}{4}H(C2)$$

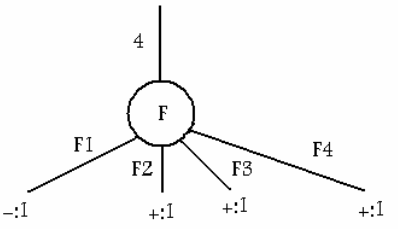
$$H(C1) = 0bit$$

$$H(C2) = 0bit$$

$$Gain(C) = H(I) - H(I,C) = 0.81$$

$$Split(C) = -\frac{1}{4}\log_2^{\frac{1}{4}} - \frac{3}{4}\log_2^{\frac{3}{4}} = 0.81$$

$$GainRatio(B) = \frac{1}{0.81} = 1.23$$



$$H(I,F)=\frac{1}{4}H(F1)+\frac{1}{4}H(F2)+\frac{1}{4}H(F3)+\frac{1}{4}H(F4)$$

$$H(F1)=0$$

$$H(F2)=0$$

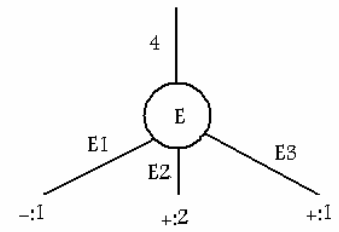
$$H(F3)=0$$

$$H(F4)=0$$

$$Gain(F)=0.81-0=0.81$$

$$Split(F)=-\frac{1}{4}\log_2^{\frac{1}{4}}-\frac{1}{4}\log_2^{\frac{1}{4}}-\frac{1}{4}\log_2^{\frac{1}{4}}-\frac{1}{4}\log_2^{\frac{1}{4}}=2$$

$$GainRatio(F)=\frac{0.81}{2}=0.405$$



$$H(I,E)=\frac{1}{4}H(E1)+\frac{2}{4}H(E2)+\frac{1}{4}H(E3)$$

$$H(E1)=0$$

$$H(E2)=0$$

$$H(E3)=0$$

$$Gain(E)=0.81$$

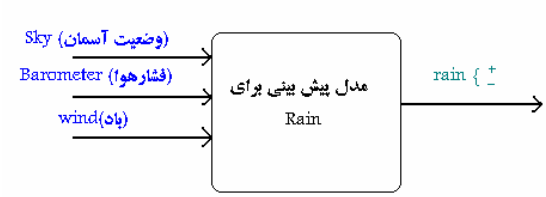
$$Split(E)=-\frac{1}{4}\log_2^{\frac{1}{4}}-\frac{2}{4}\log_2^{\frac{2}{4}}-\frac{1}{4}\log_2^{\frac{1}{4}}=1.5$$

$$GainRatio(E)=\frac{0.81}{1.5}=0.54$$

اگر نقطه ی Gain مدنظر قرار گیرد: **C=F=E>A=D>B**

اگر GainRatio مدنظر قرار گیرد: **C>E>F>A>D>B**

مثال: استفاده از الگوریتم ID3 برای پیش بینی بارش باران بر اساس متغیرهای ورودی (فاکتورهای) زیر:



	Input attributes or Decision factors			result
	Sky	barometer	Wind	Rain
1	clear	rising	north	-
2	cloudy	rising	south	+
3	cloudy	steady	north	+
4	clear	falling	north	-
5	cloudy	falling	north	+
6	cloudy	rising	north	+
7	cloudy	falling	south	-
8	clear	rising	south	-

$$H(I)=-\sum_{i=1}^{2}\frac{\#classi}{\#I}\log_2^{\frac{\#classi}{\#I}}$$

عدم قطعیت در کل نمونه ها

یا

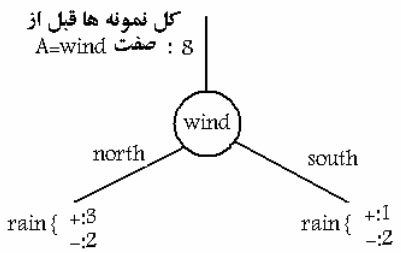
تعداد بیتها برای بیان عدم قطعیت

$$=-\frac{4}{8}\log_2^{\frac{4}{8}}-\frac{4}{8}\log_2^{\frac{4}{8}}$$

$$=1bit$$

حال با صفات (فاکتورهای) wind , Sky, Barometer مواجه ایم. برای هر کدام از این صفات باید Gain را محاسبه کنیم.

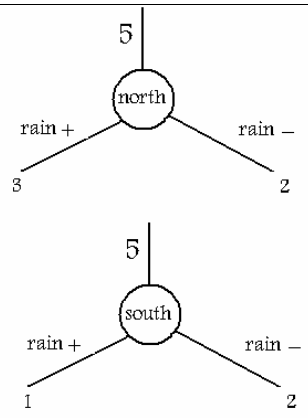
حالت اول: A=wind



$$H(I,wind)=\sum_{i=1}^{2}\frac{\#classi}{\#I}H(classi)$$

$$=\frac{\#class(north)}{\#I}H(north)+\frac{\#class(south)}{\#I}H(south)$$

$$=\frac{5}{8}H(north)+\frac{3}{8}H(south)\quad ①$$



$$H(north) = -\sum_{j=1}^{2} \frac{\#classj}{north} \log_2^{\frac{\#classj}{north}}$$

$$= -\frac{\#class+}{north} \log_2^{\frac{\#class+}{north}} - \frac{\#class-}{north} \log_2^{\frac{\#class+}{north}}$$

$$= -\frac{3}{5}\log_2^{\frac{3}{5}} - \frac{2}{5}\log_2^{\frac{2}{5}} = 0.971bits \quad ②$$

$$H(south) = -\frac{1}{3}\log_2^{\frac{1}{3}} - \frac{2}{3}\log_2^{\frac{2}{3}} = 0.918bits \quad ③$$

①، ②، ③

$$H(I, wind) = \frac{5}{8}H(north) + \frac{3}{8}H(south)$$

$$= \frac{5}{8}0.971 + \frac{3}{8}0.918 = 0.951bits$$

$$Gain(wind) = H(I) - H(I \mid wind)$$

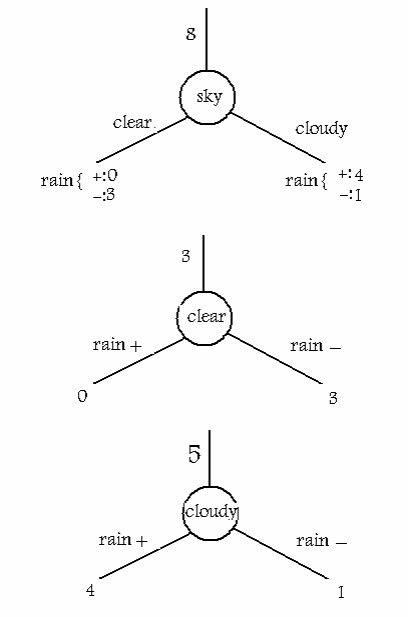
$$= 1 - 0.951 = 0.049bits$$

$$Split(wind) = -\sum_{j=1}^{2} \frac{\#classj}{\#I} \log_2^{\frac{\#classj}{\#I}}$$

$$= -\frac{5}{8}\log_2^{\frac{5}{8}} - \frac{3}{8}\log_2^{\frac{3}{8}} = 0.9544$$

$$GainRatio(wind) = \frac{0.049}{0.9544} = 0.0513$$

حالت دوم: A=sky



$$H(I, sky) = -\sum_{j=1}^{2} \frac{\#classj}{\#I} H(classj)$$

$$① \quad = \frac{3}{8}H(clear) + \frac{5}{8}H(cloudy)$$

$$H(clear) = -\frac{0}{3}\log_2^{\frac{0}{3}} - \frac{3}{3}\log_2^{\frac{3}{3}} = 0bits \quad ②$$

$$H(cloudy) = -\frac{4}{5}\log_2^{\frac{4}{5}} - \frac{1}{5}\log_2^{\frac{1}{5}} = 0.722bits \quad ③$$

①، ②، ③

$$H(I, sky) = -\frac{3}{8} \times 0 + \frac{5}{8} \times 0.722$$

$$= 0.45bits$$

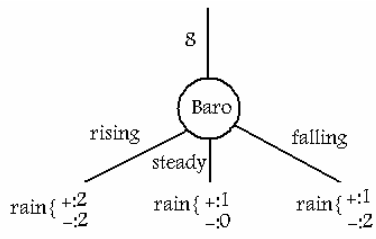
$$Gain(sky) = H(I) - H(I \mid sky)$$

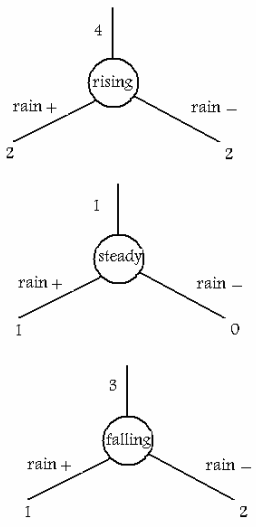
$$= 1 - 0.45 = 0.55bits$$

$$Split(sky) = -\frac{3}{8}\log_2^{\frac{3}{8}} - \frac{5}{8}\log_2^{\frac{5}{8}} = 0.9544$$

$$GainRatio(sky) = \frac{0.55}{0.9544} = 0.5763$$

حالت سوم: A=Barometer





$$H(I, Baro) = -\sum_{j=1}^{3} \frac{\#classj}{\#I} H(classj)$$

$$① \quad = \frac{4}{8}H(rising) + \frac{1}{8}H(steady) + \frac{3}{8}H(falling)$$

$$H(rising) = -\frac{2}{4}\log_2^{\frac{2}{4}} - \frac{2}{4}\log_2^{\frac{2}{4}} = 1bits \quad ②$$

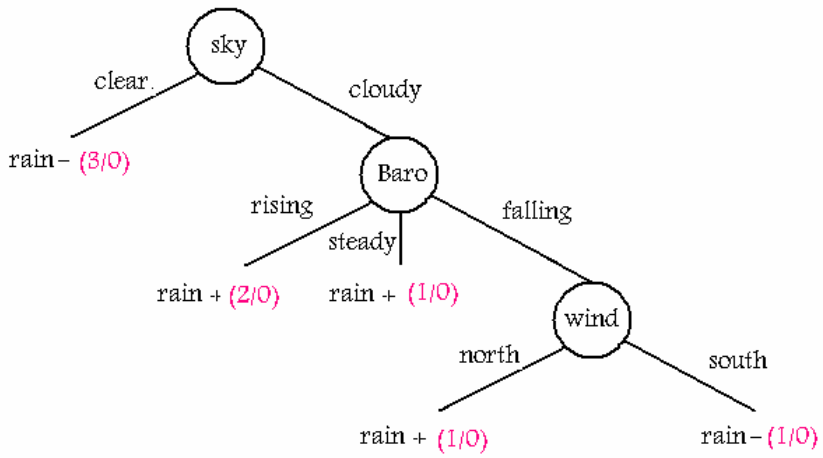
$$H(steady) = -\frac{1}{1}\log_2^{\frac{1}{1}} - \frac{0}{1}\log_2^{\frac{0}{1}} = 0bits \quad ③$$

$$H(falling) = -\frac{1}{3}\log_2^{\frac{1}{3}} - \frac{2}{3}\log_2^{\frac{2}{3}} = 0.9184bits \quad ④$$

①، ②، ③، ④

$$H(I,Baro)=-\frac{4}{8}\times 1+0+\frac{3}{8}\times 0.9182$$
$$=0.8443bits$$
$$Gain(Baro)=H(I)-H(I\mid Baro)$$
$$=1-0.8443=0.1557bits$$
$$Split(Barometer)=-\frac{4}{8}\log_2^{\frac{4}{8}}-\frac{1}{8}\log_2^{\frac{1}{8}}-\frac{3}{8}\log_2^{\frac{3}{8}}$$
$$=1.406$$
$$GainRatio(Barometer)=\frac{0.1557}{1.406}=0.1107$$

صفتی را که بیشترین بهره اطلاعاتی را دارد انتخاب می کنیم. در نهایت درخت تصمیم گیری به صورت زیر در می آید:



تمرین‌ها

تمرین۱۰-۱: اگر جدولی به صورت زیر داشته‌باشیم، در خت تصمیم‌گیری آن به چه صورت خواهدبود؟

A	B	C	خروجی
A1	B1	C1	+
A1	B2	C1	+
A2	B1	C2	-
A2	B2	C2	-

تمرین۱۰-۲: تصمیم‌گیری در سیستم‌های پشتیبان تصمیم‌گیر یا DSS چه نقشی دارد؟ نقش DSS در سیستم‌های خبره چیست؟

استدلال فازی

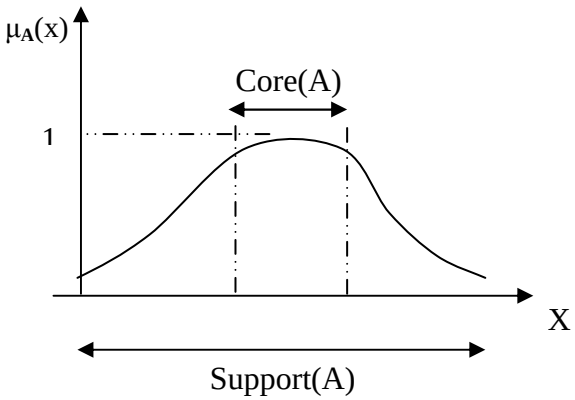
پرسش: فازی چیست؟ چرا منطق فازی مطرح شد؟ منطق و استدلال فازی چه نقشی در طراحی سیستم‌های خبره دارد؟

مرز بین خیلی از مفاهیم مشخص و واضح نیست.

مثال: یکسری افراد با سن از مجموعه X و با درجه عضویت از مجموعه Young به صورت زیر موجود باشد :

X={ 5 , 15 , 25 , 35 , 45 , 55 , 65 , 75 , 85 }
Young = { 0 , 0.2 , 0.4 , 0.8 , 0.4 , 0.1 , 0 , 0 , 0 }

مفهوم Support و Core و α ـ Cut – Set



شکل ۱۱-۱: مفهوم Support و Core

عملگرهای فازی

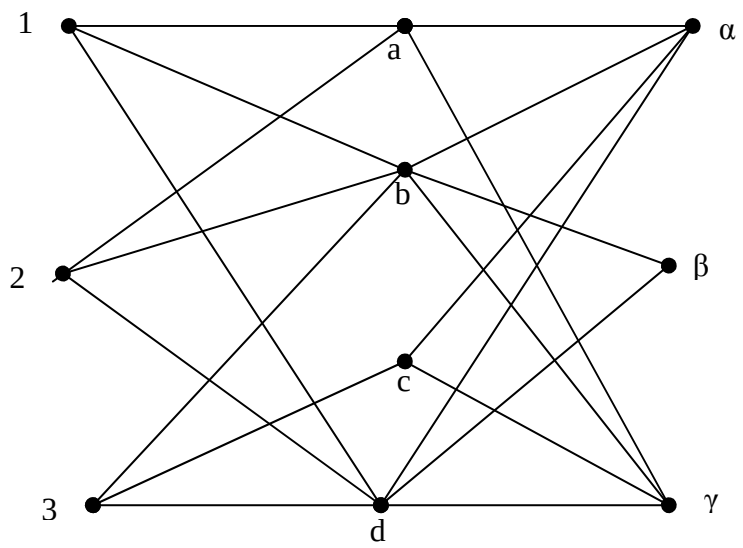
- **تفاضل متقارن**
- **تفاضل معمولی**
- **فاصله همینگ**
- **ترکیب توابع فازی**

فرض کنید دو رابطه R , S را داشته باشیم SoR یا S(R) (ترکیب S با R)را بسازیم.

R	a	b	c	d
1	0.1	0.2	0	1
2	0.3	0.3	0	0.2
3	0.8	0.9	1	0.4

S	α	β	γ
a	0.9	0	0.3
b	0.2	1	0.8
c	0.8	0	0.7
d	0.4	0.2	0.3

S.R	α	β	γ
1	0.4	0.2	0.3
2	0.3	0.3	0.3
3	0.8	0.9	0.8



شکل ۱۱-۲: ترکیب S با R

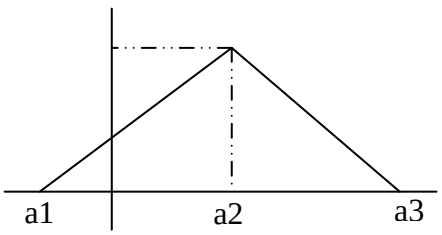
- **جمع دو مجموعه فازی**
- **تفریق دو مجموعه فازی**

نمایش اعداد فازی

۱ـ مثلثی ۲ـ ذوزنقه‌ای

مثلثی: هر مثلثی با سه پارامتر مشخص‌می‌شود. $A(a_1, a_2, a_3)$

درجه $a_1, a_3 = 0$ و درجه $a_2 = 1$



$$\mu_A(X) = \begin{cases} 0 & x < a_1 \\ \dfrac{x - a_1}{a_2 - a_1} & a_1 \le x \le a_2 \\ \dfrac{a_3 - x}{a_3 - a_2} & a_1 \le x \le a_2 \\ 0 & x > a_3 \end{cases}$$

a_1 تا a_2 خطی با شیب و ضریب مثبت است و a_2 تا a_3 خطی با شیب منفی است

پرسش: فرض کنید A= { -5 , -1 , 1} باشد آنگاه $A_{0.5}$ را بدست آورید.

استدلال فازی

If A(x) then B(x) یا If X is A then Y is B

R(x,y) , A(x) → B(y) رابطه بین A و B

به

Generalized modus ponens : Gmp
Fact : X is A' : R(x)
Rule : if x is A then Y is B
result : Y is B' : R(y) = R(x) o R(x,y)

به این قانون Gmp گویند که modus ponens تعمیم یافته است. رابطه بین A(x) و B(x) در R(x) ذخیره می شود .

Generalized modus tolens
Y is B' : R(y)
If X is A then Y is B

A': R(x) = R(y) o R(x,y)

مثال:

"x and y are approximately equal" جمله فازی x,y تقریبا مساوی هستند

For this rule , a premise is given like (approximately)
"x is small". جمله فازیست و x,y تقریبا مساوی هستند.

R(x,y)=approximately – Equal(x,y) R(x) = small (x)
R(x):

X	1	2	3	4
μ_R (x)	1	0.6	0.2	0

Membership degrees of R(x,y)

Y / x	1	2	3	7
1	1	0.5	0	0
2	0.5	1	0.5	0.5
3	0	0.5	1	0.5
4	0	0	0.5	1

R(y)=R(x) o R(x,y)

Membership degrees R(y)

Y	1	2	3	4
μ_R (y)	1	0.6	0.5	0.2

همانطور که می بینید y مفهومش مشابه x است.

مثال: فرض کنید $B \subseteq H, A \subseteq T$

A مفهوم ارتفاع و B مفهوم تقریباً بلند و T مفهوم دما و H مفهوم رطوبت بوده و T,H مجموعه هایی فراگیر هستند.

A:" high" $A \subseteq T$ R (t): t is A
B:"fair high" $B \subseteq H$ R (h): h is B
R (t, h) = if t is A, then h is B
R (t, h): R (t) → R (h)

Rule ها به این صورت است:

If temperature is high then اگر دما بالا باشد

Humidity is fairy high رطوبت تقریبا بالاست (این جملات نمایانگر $R(t,h): R(t) \rightarrow R(h)$ است)

Humidity: H رطوبت

Fairy high: B مجموعه تقریبا رطوبت بالاست

High: A مجموعه

Temperature: دما

پس دو مجموعه داریم : یکی برای دما و دیگری برای رطوبت

فرض کنید T (دما) به صورت زیر باشد:

T	20	30	40
μ_A (t)	0.1	0.5	0.9

و رطوبت (H) به صورت زیر باشد :

H	20	50	70	90
μ_B (t)	0.2	0.6	0.7	1

و اگر داشته باشیم :

$$R(t,h) = A*B = \int \mu_A(t) \wedge \mu_B(h) \Big/ (t,h)$$

نسبت

که مقادیر هر خانه از جدول R(t,h) را با استفاده از فرمول بالا بدست می آوریم.

h / t	20	50	70	90
20	0.1	0.1	0.1	0.1
30	0.2	0.5	0.5	0.5
40	0.2	0.6	0.7	0.9

اکتساب دانش

مهندس دانش، دانش را از منابع زیر اکتساب می‌کند:

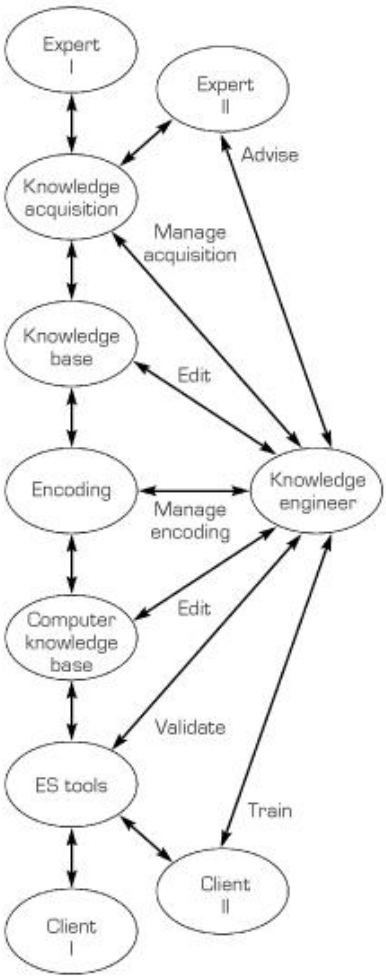
- افراد خبره
- متون و کتب
- موارد مطالعه (Case Studies)

افراد خبره اصلی‌ترین منابع دانش هستند. در اغلب موارد، اکتساب دانش، شامل مصاحبه‌ها طولانی میان مهندس دانش و فرد (یا افراد) خبره می‌باشد. سپس مهندس دانش باید دانش اخذ شده به زبان طبیعی را به دانشی که قابل ذخیره در پایگاه دانش باشد، نگاشت کند.

اکتساب دانش را چنین تعریف نموده اند:

"تبدیل و انتقال خبرگی بالقوه لازم برای حل مسئله از یک منبع به یک برنامه"

مهندسی دانش امروزه شامل بر متدها و تکنیک‌هایی برای به دست آوردن، مدل‌سازی، نمایش و ارائه و استفاده از دانش است. تکنیک‌های نوین در مدل سازی دانش باعث شده که بتوان از دانش به صورت استفاده مجدد در حوزه های مختلف یک دامنه از مسائل استفاده نمود. در حال حاضر سیستم‌های مبتنی بر دانش، سیستم‌های نرم‌افزاری هستند که در آن دانش بشر برای حل مسائل و استدلال در لایه‌ای جدا از لایه کنترل و استدلال ذخیره شده است. به این ترتیب لایه دانش از لایه استدلال جدا شده و معماری و طراحی سیستم‌های مبتنی بر دانش با معماری سیستم‌های نرم‌افزار متداول تفاوت‌های زیادی پیدا کرده است.

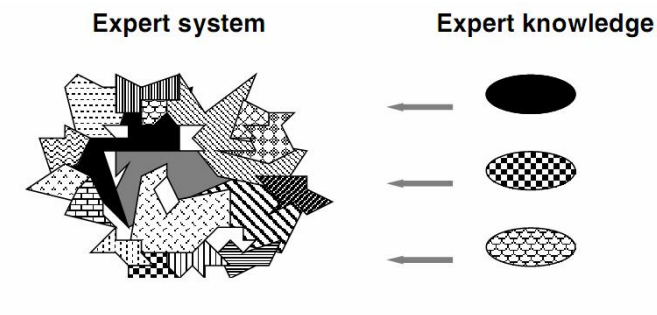


شکل۱۳-۱: نقش مهندس دانش در فرایند ساخت سیستم خبره

دانش در ذهن خبره و در پایگاه دانش

فرضیه‌ی سیستم نماد فیزیکی اظهارمی‌کند که ساختمان سمت چپ در شکل۱۳-۲، گونه‌ای از نقشه‌ی دانش یک دامنه می باشد.

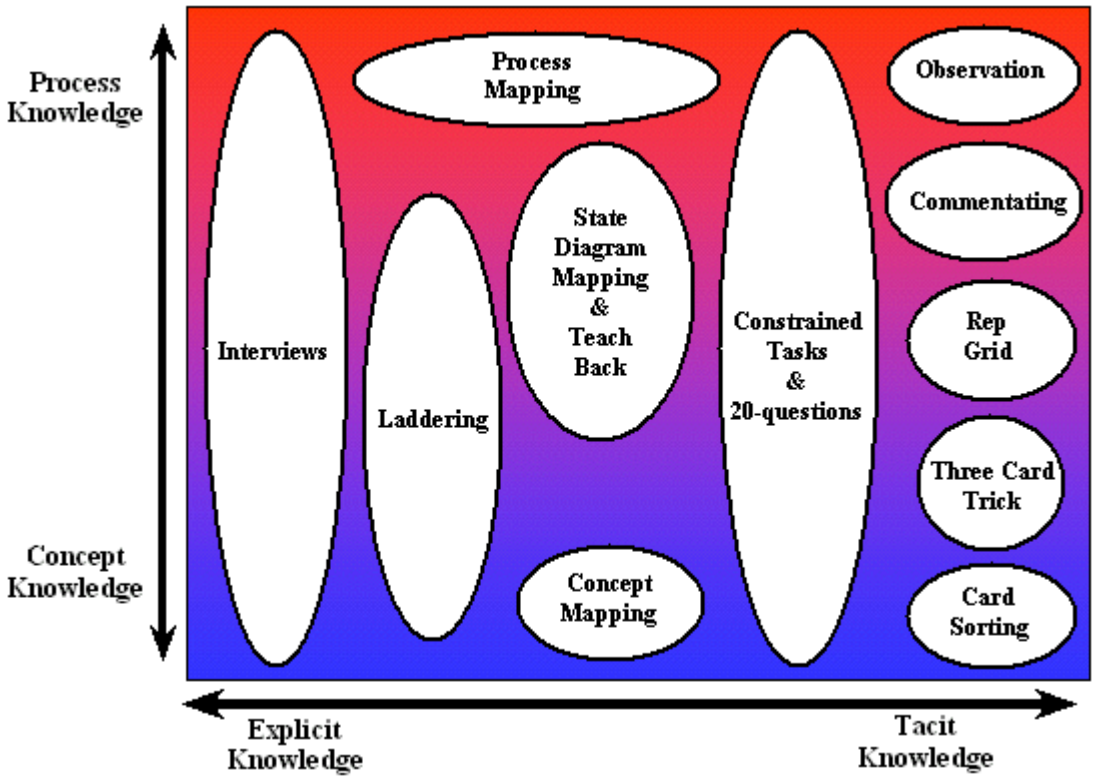
پرسش: فرضیه ی سیستم نماد فیزیکی چیست؟



شکل۱۳-۲: دانش در سیستم خبره

مقایسه تکنیک‌های KA

برای KA تکنیک‌های مختلفی پیشنهادشده‌است که در شکل ۱۳-۳، این تکنیک‌ها با هم مقایسه‌شده‌است.



شکل ۱۳-۳: مقایسه تکنیک‌های KA

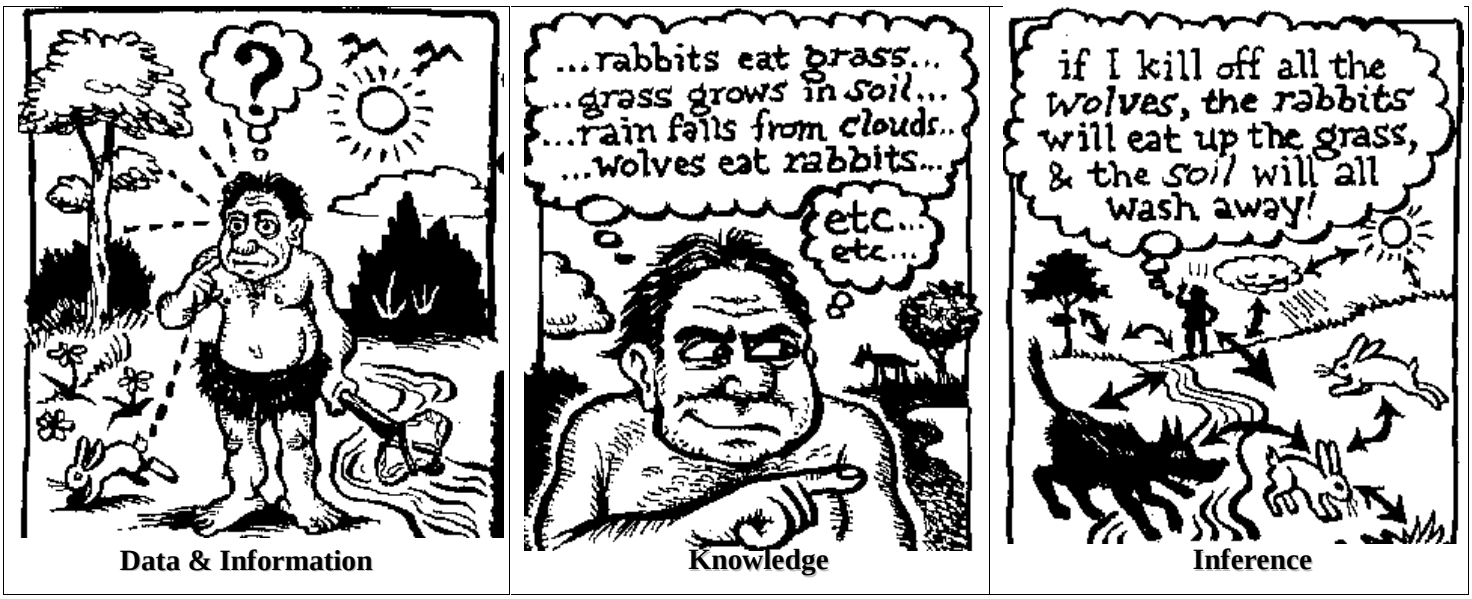
تمرین‌ها

تمرین۱۳-۱: تحقیق کنید که تکنیک‌های KA در شکل ۱۳-۳، چه مشکلات و محدودیت‌هایی ممکن است، داشته باشند.

تمرین۱۳-۲: همانطورکه اشاره‌شد، مصاحبه یکی از رایج‌ترین و متداول‌ترین روش‌ها در KA است. تحقیق کنید که برای اکتساب دانش موثر، سبک و نوع سوالات در جلسات مصاحبه با افراد خبره چگونه باید باشد. مهندس دانش چه نکاتی را باید در اکتساب دانش، توجه داشته باشد؟

موضوعات تکمیلی

دانش: به‌کارگیری داده و اطلاعات است.



شکل۱۴-۱: داده و اطلاعات، دانش، استنتاج

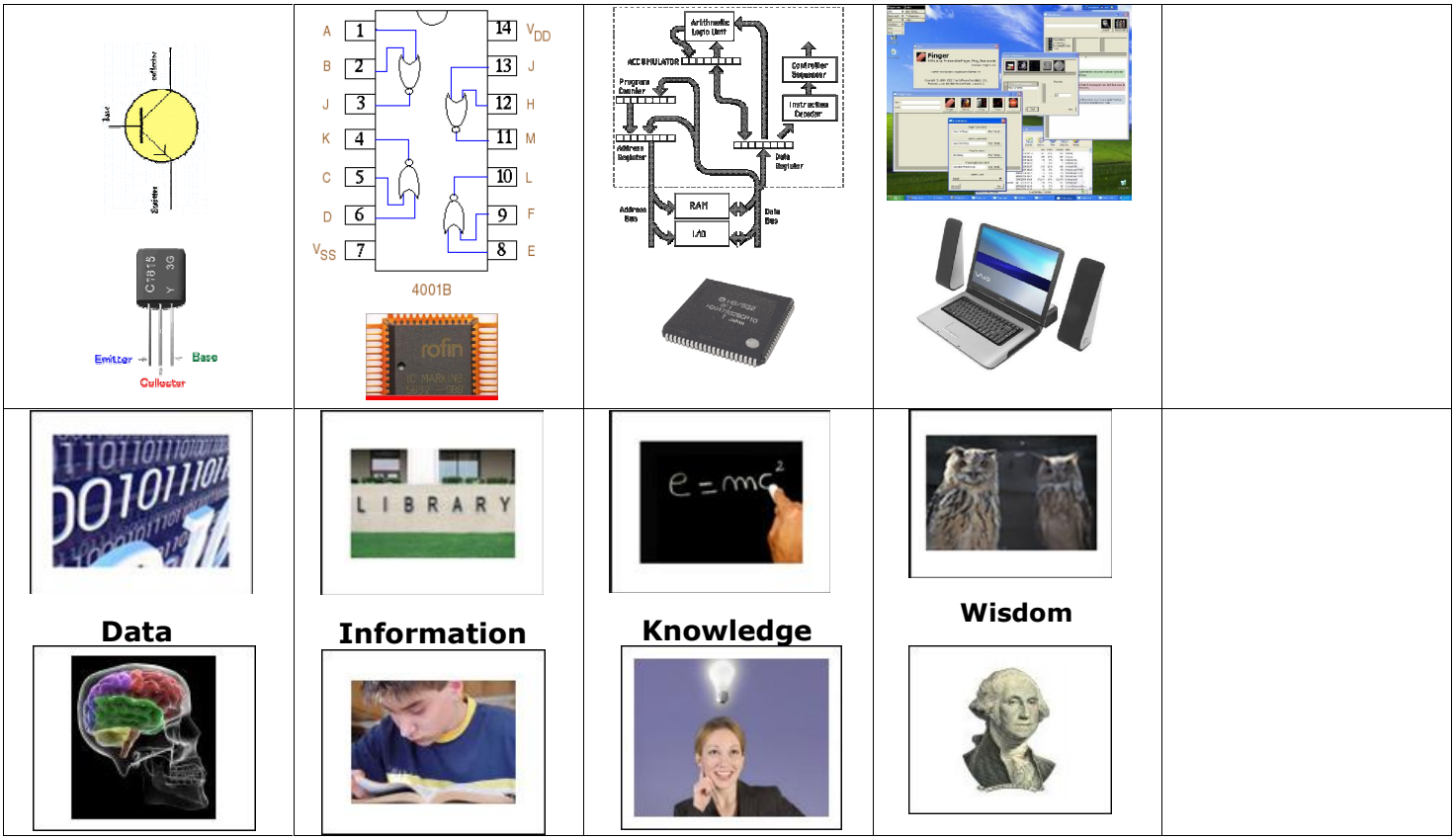
انواع دانش

- **Procedural Knowledge**
  - Knowledge of processes
  - Example: how to clean your face? (men v.s. women)
- **Declarative Knowledge**
  - Knowledge of propositions or facts (declarations) that are true or false
  - Example: rabbits eat grass
- **Tacit Knowledge**
  - Knowledge which cannot be expressed in language
  - Example: how to drive a car?
- **Meta-knowledge**
  - Knowledge about knowledge
  - Example: According to my study experience, ‘‘types of knowledge’’ will be a question in the mid-term exam.

انواع استنتاج

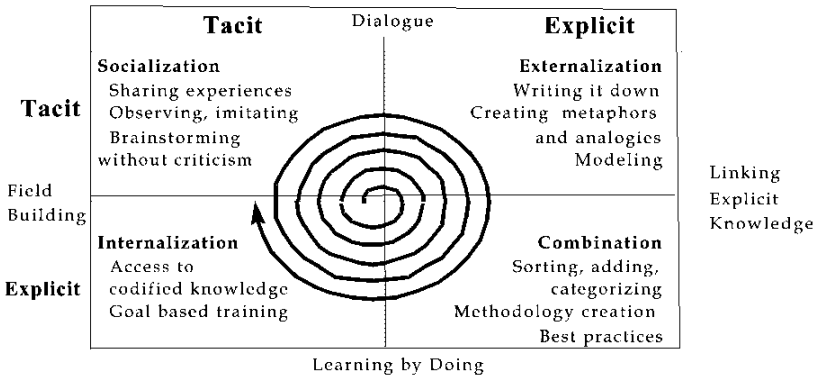
- **Deduction**: reasoning where conclusions must follow from premises
- **Induction:**– inference is from the specific case to the general
- **Intuition:** no proven theory
- **Heuristics:** rules of thumb based on experience
- **Generate and test**: trial and error
- **Abduction:**– reasoning back from a true condition to the premises that may have caused the condition
- **Default:**– absence of specific knowledge
- **Nonmonotonic:** New evidence may invalidate previous knowledge
- **Analogy:** inferring conclusions based on similarities with other situations

نکته: دانش؛ شکل گیری الگوهای متفاوت در مقیاس‌های متفاوت است. !!



شکل ۱۴-۲: شکل گیری دانش و خرد

چرخه‌ی تبدیل دانش؛ چرخه نو ناکا(Nonaka's spiral of knowledge)



شکل ۱۴-۳: چرخه معروف نو ناکا

دانش ضمنی: دانش‌ضمنی به دانش اجتماعی‌سازی معروف است که از طریق تعاملات اجتماعی به تبادل تجربیات و مدل‌های ذهنی و انتقال دانسته‌های مهارتی و تجربی خود می پردازد. مثال: داستان‌گویی ، آموزش‌ها در کلاس و ... .دانش ضمنی به سادگی قابل صریح‌شدن نیست ولی قابل فهم است.

دانش ضمنی به ضمنی(اجتماعی‌سازی): این نوع دانش‌ها در ارتباطات رو در رو به دست می‌آید مانند مطالبی که سر کلاس درس، آموزش‌داده‌می‌شود و صحبت‌هایی که افراد با هم دارند ، به این نوع تبادل دانش، اجتماعی‌سازی گویند.

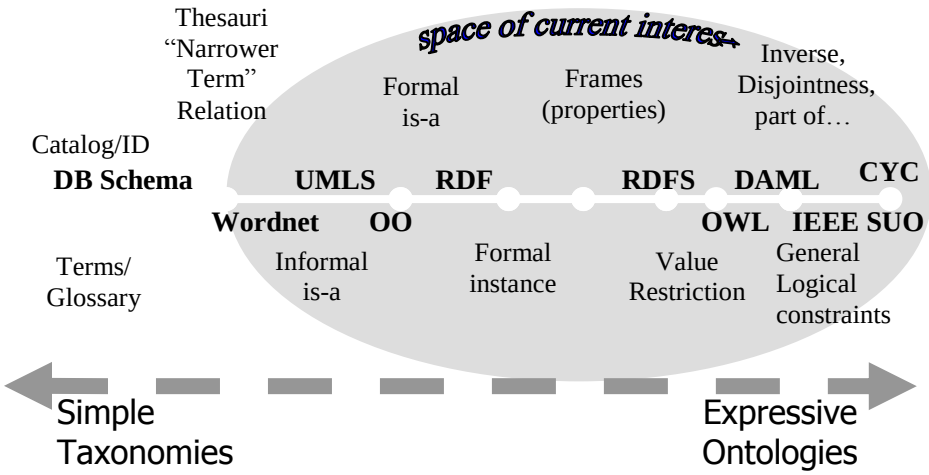
دانش ضمنی به صریح(برون‌سازی): بیان واضح و کد کردن صریح دانش است . در این تبدیل دانش ، دانش از داخل به خارج منتقل می‌شود. مثل نوشتن یک کتاب ، کشف علمی و ...

دانش صریح به صریح (ترکیب): که در کامپیوتر بیشتر رایج است و به دانش صریح به صریح ، ترکیب نیز گفته می‌شود که اجزای مختلف آن با روش‌های مختلفی هم بسته شده‌اند . این فرآیند می تواند هم به وسیله ی انسان‌ها و هم به وسیله‌ی کامپیوتر انجام گیرند.

دانش صریح به ضمنی (درونی‌سازی):در این تبدیل دانش به وسیله ی تجربه، ترکیب‌می‌شود و این دانش ترکیب‌شده، ارزیابی و به صورت یک تجربه‌ی ضمنی، شکل‌می‌گیرد. تجربه‌ها و مهارت‌های افراد و تیم‌ها از این دسته هستند.

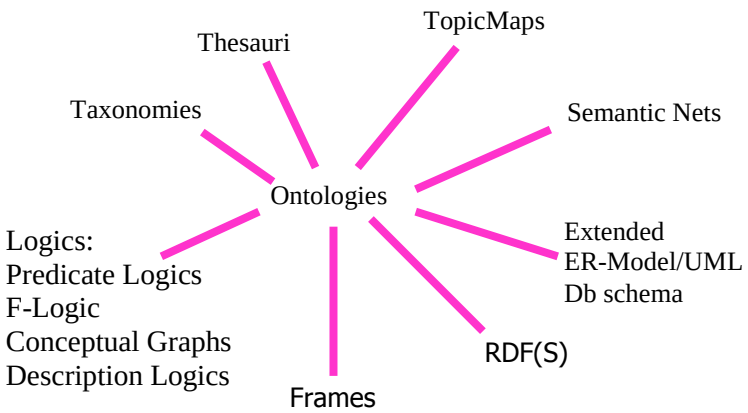
آنتالوژی

An explicit formal specification of how to represent the objects, concepts and other domain entities and relationships among them.



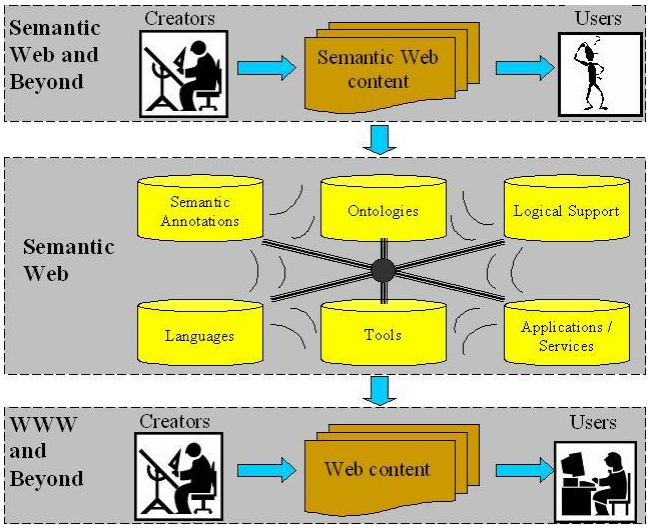
شکل۱۴-۴: زبان‌های آنتالوژی

نکته: آنتالوژی به عنوان یک زبان بازنمایی دانش است.



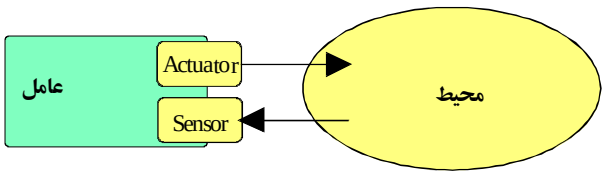
شکل۱۴-۵: آنتالوژی به عنوان روش بازنمایی دانش

کاربردهایی از آنتالوژی



شکل۱۴-۶: کاربرد آنتالوژی در وب معنایی

عامل‌ها و سیستم‌های خبره



شکل۱۴-۷: ساختار کلی عامل

انواع عامل‌ها (یادآوری از هوش مصنوعی)

(0) Table-driven agents :

use a percept sequence/action table in memory to find the next action. Implemented by a (large) lookup table

(1) Simple reflex agents

Based on condition-action rules, implemented with an appropriate production system; stateless devices with no memory of past world states

(2) Agents with memory

have internal state that is used to keep track of past states of the world

(3) Agents with goals

Agents that have state and goal information that describes desirable situations. Agents of this kind take future events into onsideration.

(4) Utility-based agents

base decisions on classic axiomatic utility theory in order to act rationally

تمرین‌ها

تمرین۱۴-۱: تحقیق کنید که چرخه نانوکا چه کاربرد و اهمیتی در سیستم‌های خبره دارد؟ پاسخ خود را همراه با مثال، توضیح دهید.

تمرین۱۴-۲: تحقیق کنید که آنتالوژی در چه کاربردهای و نرم‌افزارهای دانشی، استفاده‌می‌شود؟ آنتالوژی به عنوان یک روش بازنمایی دانش، با سایر روش‌های بازنمایی دانش مقایسه کنید.