

به نام خدا

جزوه درسی مربوط به درس

سیستم‌های خبره

Expert Systems

حمید رضا نیرومند

<http://niroomand.ir>

فهرست مطالب

۴	تعریف سیستم خبره.....
۴	تعریف استدلال (Reasoning).....
۴	انواع استدلال.....
۶	سیستم‌های استدلال خودکار.....
۷	اجزای سیستم‌های خبره.....
۸	تاریخچه سیستم‌های خبره.....
۱۲	زمستان هوش مصنوعی.....
۱۳	معرفی ابزار ES-Builder.....
۱۵	درخت تصمیم‌گیری (Decision Tree):.....
۱۸	معماری نرم‌افزاری سیستم‌های خبره.....
۲۰	نمایه‌سازی.....
۲۰	تعریف نمایه.....
۲۰	تعاریف دیگر از نمایه.....
۲۰	دلایل نیاز به نمایه‌سازی.....
۲۰	اهداف نمایه‌سازی.....
۲۱	نمایه یک فرا داده است.....
۲۱	تعریف فراداده یا Metadata.....
۲۱	آنتولوژی چیست؟.....
۲۲	انواع وب.....
۲۲	1- وب ۱ (وب سنتی) (Traditional Web).....
۲۲	۲- وب ۲ (وب اشتراکی) (Social Web).....
۲۲	3- وب ۳ (وب معنایی یا وب هوشمند) (Semantic Web).....

۲۲ روش‌های معنا بخشیدن به صفحات وب
۲۳ XML چیست؟
۲۵ URI چیست؟
۲۵ URL چیست؟
۲۶ RDF چیست؟
۲۶ ۳ موضوع اصلی در RDF
۲۶ مزایای RDF نسبت به XML:
۲۷ روش اعمال RDF به یک صفحه با HTML5
۲۸ مدل مخفی مارکوف چیست؟
۳۰ کاربردهای مدل مخفی مارکوف
۳۲ مزایای سیستم‌های خبره
۳۲ معایب سیستم‌های خبره
۳۳ کاربردهای سیستم‌های خبره
۳۵ سیستم‌های تشخیص الگو یا Pattern Recognition Systems
۳۵ ANN: Artificial Neural Networks
۳۷ Knowledge engineering دانش مهندسی
۳۷ مهندسی مبتنی بر دانش (Knowledge-based engineering (KBE))
۳۷ مزایای استفاده از مهندسی دانش در صنعت:
۳۸ طراحی به کمک رایانه (CAD)
۳۸ PLM
۴۲ Knowledge representation and reasoning

تعریف سیستم خبره

In artificial intelligence, an expert system is a computer system emulating the decision-making ability of a human expert.

در هوش مصنوعی، یک سیستم خبره یک سیستم کامپیوتری است که توانایی تصمیم‌گیری یک متخصص انسانی را تقلید می‌کند.

Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if-then rules rather than through conventional procedural code.

سیستم‌های خبره طراحی شده‌اند تا مسائل پیچیده را با کمک استدلال از طریق دانشی که معمولاً در قالب یک سری قوانین اگر-آنگاه نمایش داده شده حل کنند و نه از طریق کدهای رویه‌ای مرسوم.

تعریف استدلال (Reasoning)

استدلال یا گواه‌آوری (Reasoning)، ترکیب قانون‌مند قضیه (های) معلوم برای رسیدن به قضیه (های) تازه است. در استدلال، ذهن بین چند قضیه، ارتباط برقرار می‌کند تا از پیوند آن‌ها، نتیجه زاده شود و به این ترتیب نسبتی مشکوک و مبهم به نسبتی یقینی تبدیل شود به عبارتی دیگر به تجمیع تصدیقات (به جملات خبری گفته می‌شود که بر سلب یا ایجاب مطلبی دیگر امر دارد. مانند: جیوه فلز است. انسان حیوان نیست.) برای اثبات تصدیقی دیگر استدلال گفته می‌شود.

انواع استدلال

1 - استدلال استقرایی یا جزء به کل یا Induction:

نوعی استدلال است که در آن ذهن از جزء به کل سیر می‌کند. یعنی چند مورد جزئی را مشاهده می‌کند و سپس یک حکم کلی می‌دهد.

مثال: علی انسان است. علی دو پا دارد. انسان دو پا دارد.

مثال: در چند مورد آب را حرارت می‌دهیم و می‌بینیم که در صد درجه سلسیوس (سانتی‌گراد) می‌جوشد و از این نتیجه می‌گیریم که هر آبی در صد درجه سلسیوس می‌جوشد.

2 - استدلال قیاسی یا استنتاج یا Deduction:

وقتی ذهن از قضیه‌های کلی به نتیجه‌های جزئی می‌رسد و به عبارت مختصرتر از کل به جز می‌آید؛ آن را قیاس می‌نامند. مثال:

۱. هر انسان فانی است.
۲. سقراط انسان است.
۳. پس سقراط فانی است.

یا

۱. همه فرانسوی‌ها پنیر دوست دارند.
۲. پی‌یر فرانسوی است.
۳. بنابراین پی‌یر، پنیر دوست دارد.

اساسی‌ترین و اعتمادپذیرترین شکل استدلال در منطق، استدلال قیاسی است؛ چرا که نتیجه‌دهی قیاس همواره یقینی است.

سؤال چالشی: داستان مناظره امام صادق علیه السلام با ابوحنیفه را مطالعه کنید. چه تفاوتی میان آن قیاسی که امام صادق آن را باطل می‌داند و این قیاسی که در اینجا بحث شده وجود دارد؟

3 - تمثیل (منطق) (Argument from analogy)

تمثیل سرایت دادن حکم یک موضوع به موضوع دیگر به دلیل مشابهت آن دو به یکدیگر است. واژه تمثیل در این بافت به معنای تشبیه کردن یا مانند کردن چیزی به چیزی دیگر به کار رفته است. در استدلال تمثیلی می‌کوشیم تا بر اساس همانندی بین دو چیز، حکمی را که برای یکی از آن‌ها درست می‌دانیم برای دیگری هم درست بدانیم. توجه: هر چه همانندی میان دو پدیده بیشتر باشد، استدلال تمثیلی استوارتر و نیرومندتر است؛ اما نتیجه آن هیچ‌گاه قطعی نیست.

4 - مغالطه (یا سفسطه یا Fallacy)

مُغَالَطَه یا مَغَالَطَه، آوردن دلیل اشتباه یا غیرمجاز برای استدلال است که از نظر علم منطق به یکی از دلایل زیر نادرست باشد:

- نخست آنکه دست کم یکی از مقدمات گزاره نادرست باشد؛
- دوم آنکه مقدمات گزاره، متضمن نتیجه گزاره نباشد.

مغالطه جزئی از برهان است که به طور قابل اثباتی در منطق آن ایراد وجود دارد و بنابراین کل برهان را نامعتبر می‌سازد. مغالطه ممکن است برای وارونه کردن حقیقت (ها) به کار رود.

مثال:

ماست از شیر است؛ شیر برای اسهال مضر است؛ پس ماست برای اسهال مضر است.

سیستم‌های استدلال خودکار^۱

در علوم کامپیوتر، به ویژه در ارائه دانش و استدلال و متالوگ، حوزه استدلال خودکار به درک جنبه‌های مختلف استدلال اختصاص دارد. مطالعه استدلال خودکار به تولید برنامه‌های رایانه‌ای کمک می‌کند که به رایانه‌ها اجازه می‌دهد به طور کامل یا تقریباً به طور کامل و همینطور به صورت خودکار استدلال کنند. اگرچه استدلال خودکار به عنوان زیرمجموعه‌ای از هوش مصنوعی در نظر گرفته می‌شود، اما با علم کامپیوتر و فلسفه نظری نیز ارتباط دارد.

توسعه یافته‌ترین نوع استدلال خودکار، «اثبات خودکار قضیه» (Automated Theorem Proving) و «بررسی اثبات خودکار» (Automated Proof Checking) (که به عنوان یک تضمین برای استدلال صحیح تحت مفروضات ثابت تلقی می‌شود) است.

اجزای سیستم‌های خبره

- سیستم خبره از دو زیرسیستم تشکیل شده:

- Knowledge Base (پایگاه دانش)
- حاوی حقایق و قوانین است (Facts and Rules)
- Inference Engine (موتور استنتاج)
- قوانین را روی حقایق اعمال می‌کند تا حقایق جدید به دست آید.
- موتور استنتاج می‌تواند دارای قابلیت توصیف (Explanation) و خطایابی (Debugging) نیز باشد.

- راه‌های کسب دانش:

- استفاده از مستندات موجود
- مشاهده
- مصاحبه
- پرسشنامه

تاریخچه سیستم‌های خبره

- دهه ۱۹۴۰: ظهور کامپیوتر
- اوائل ۱۹۵۰: بررسی پتانسیل (ظرفیت) کامپیوترها در «فکر کردن مانند انسان»
- اواخر ۱۹۵۰: بررسی تقلید decision-making توسط کامپیوتر، به کارگیری کامپیوتر در تشخیص بیماری در زمینه‌های پزشکی و بیولوژی
- برخی از نمونه‌های اولیه سیستم‌های خبره عبارتند از:

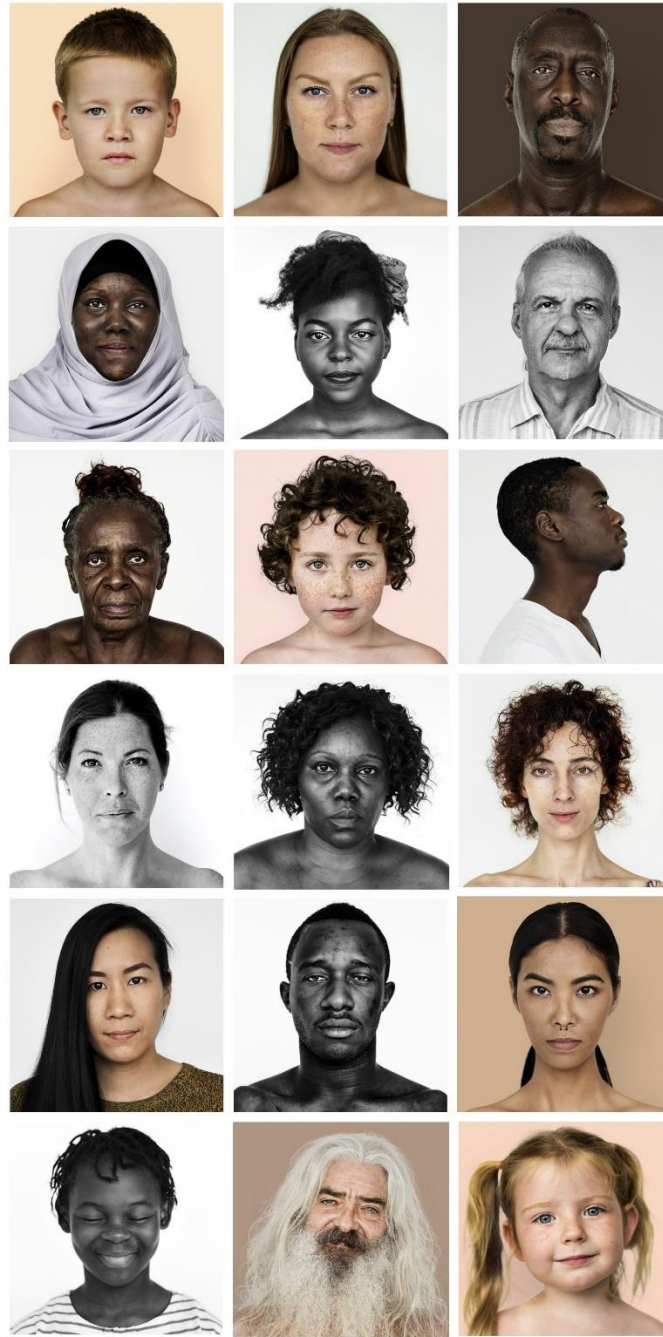
- MYCIN
- INTERNIST-I
- CADUCEUS
- Dendral

- سیستم‌های خبره اولین بار در سال ۱۹۶۵ با پروژه برنامه‌نویسی اکتشافی استنفورد که توسط Edward Feigenbaum (که گاهی پدر سیستم‌های خبره شناخته می‌شود) مدیریت می‌شد، معرفی شد. آن‌ها در این پروژه به دنبال تشخیص بیماری‌های عفونی (پروژه MYCIN) و شناسایی مولکول‌های اورگانیک ناشناخته (پروژه Dendral) بودند.
- این جمله از آقای فیگنباوم، گام بلندی در زمینه سیستم‌های خبره بود:

intelligent systems derive their power from the knowledge they possess rather than from the specific formalisms and inference schemes they use

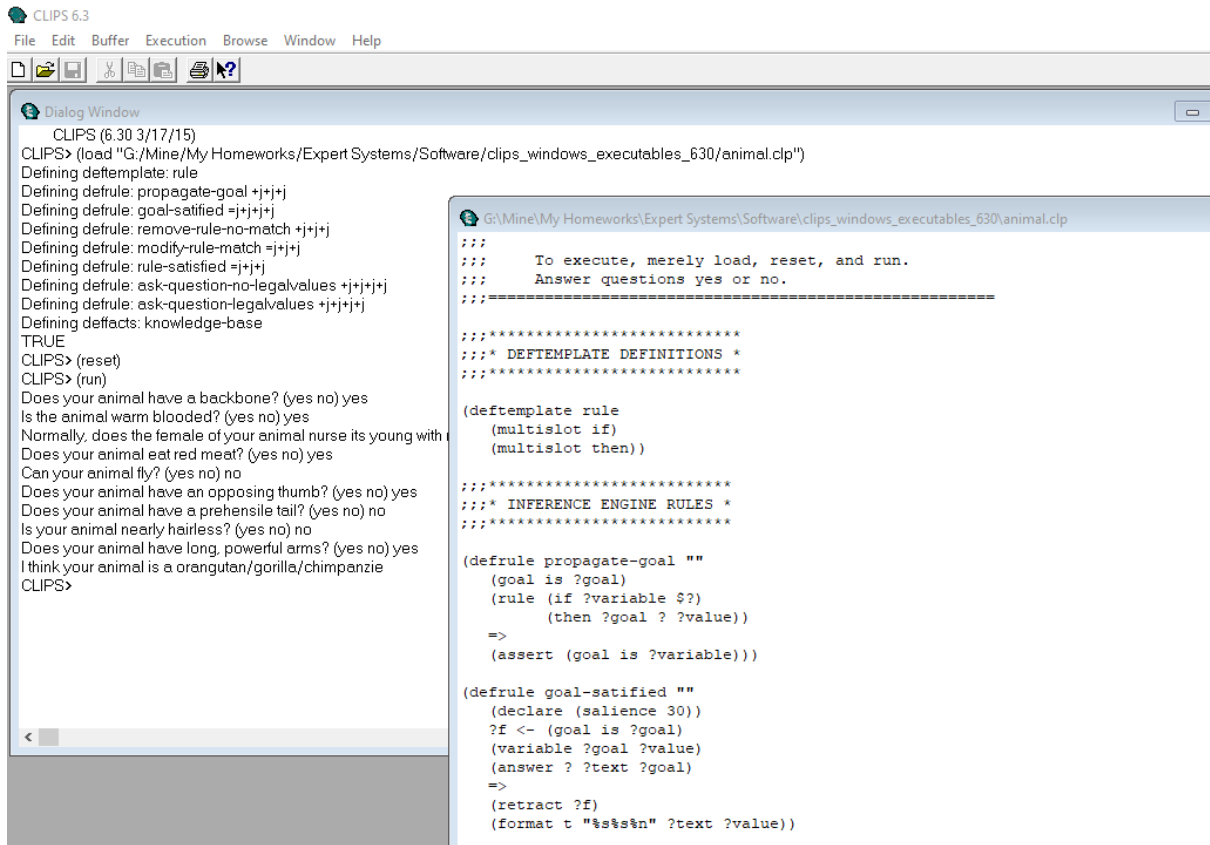
سیستم‌های خبره قدرت خود را از **دانشی** که دارند به دست می‌آورند و نه از توصیف رسمی و شمای استنتاجی که استفاده می‌کنند. مثال: پروژه World Face Project که هدف آن ایجاد یک پایگاه دانش بزرگ از صورت‌های انسانی بود.

- اولین سیستم‌های خبره در دهه ۱۹۷۰ ساخته شدند و در دهه ۱۹۸۰ رشد آن‌ها افزایش یافت.
- سیستم‌های خبره را می‌توان اولین شکل موفق نرم‌افزارهای هوش مصنوعی دانست.



rawpixel

- کار بر روی سیستم‌های خبره در فرانسه نیز پویا بود. در حالی که در آمریکا تمرکز بر روی کدنویسی سیستم خبره rule-based با زبان LISP و ایجاد Shell برای سیستم خبره بود، در فرانسه تمرکز بر روی استفاده از زبان Prolog بود.
- مزیت استفاده از Shell این است که کار با سیستم خبره را برای کاربران نابرنامه‌نویس راحت‌تر می‌کند.
- مثال: زبان Clips برای تولید سیستم خبره



- مثال قرآنی: داستان گاو بنی اسرائیل

پیشنهاد: فیلم‌های زیر را تماشا کنید:

- The Wild Child 1970
- AI 2001
- The Imitation Game

- در سال ۱۹۸۰ توجه به سیستم‌های خبره شدت گرفت؛ دانشگاه‌ها دوره‌هایی با این مضامین تعریف کردند.

- در سال ۱۹۸۱ کامپیوتر PC با Ms-DOS معرفی شد.

- در همین سال‌ها تفاوت بین قدرت Mainframe ها و PC ها باعث پیدایش مفهومی به نام «مدل کلاینت/سرور» در دنیای کامپیوتر شد.

- قبل از ظهور PC ها، سیستم‌های خبره بر روی Lisp Machine ها از شرکت Xerox و... اجرا می‌شدند اما بعد از آن بر روی کامپیوترهای شخصی قابل اجرا بودند.

- در سال ۱۹۸۲ اولین سیستم خبره که در طراحی محصولات با ابعاد بالا به کار گرفته می‌شد با نام SID¹ تولید شد.
- در اواسط دهه ۱۹۷۰، دیدگاه‌ها نسبت به سیستم‌های خبره خوشبینانه‌تر شد و انتظار می‌رفت که سیستم‌های خبره کاملاً خودکار تولید شود.
- در اوائل ۱۹۷۰ انتظارها در مورد توانایی کامپیوترها بسیار ایده‌آل‌گرایانه بود. اما Richard M. Karp با ارائه مقاله مشهور خود با عنوان "Reducibility Among Combinatorial Problems" و طرح مسأله "Karp's 21 NP-complete problems" انتظارها از کامپیوتر را منطقی‌تر کرد.

- His findings describe what computers can do and what they cannot do.

- در دهه ۱۹۹۰ و بعد از آن، واژه «سیستم خبره» و ایده یک سیستم مجزا برای هوش مصنوعی از دایره لغات فناوری اطلاعات حذف شد. این اتفاق به صورت تفسیر می‌شود: ۱- سیستم‌های خبره با شکست مواجه شدند؛ یعنی نتوانستند آنچه قول داده بودند را عرضه کنند. ۲- برعکس تفسیر قبلی، برخی معتقدند: سیستم‌های خبره قربانی موفقیت خود شدند! (victims of their success) به معنا که بسیاری از شرکت‌هایی مانند SAP و Oracle که زمانی پشتیبان سیستم‌های خبره بودند، این سیستم‌ها را به عنوان بخشی از ابزارهای دیگر و در بطن آن‌ها به کار گرفتند و به این صورت سیستم‌های خبره از «مستقل بودن»، به «بخشی از یک ابزار بودن» تبدیل شدند.

- رویکردهای امروزی به سیستم‌های خبره:

- The limitations of the previous type of expert systems have urged researchers to develop new types of approaches. They have developed more efficient, flexible and powerful approaches in order to simulate the human decision-making process. Some of the approaches that researchers have developed are based on new methods of artificial intelligence (AI), and in particular in machine learning and data mining approaches with a feedback mechanism

- محدودیت‌های سیستم‌های خبره قبلی، باعث شد که پژوهشگران رویکردهای جدیدی را توسعه دهند. آن‌ها روش‌های کارا تر، انعطاف‌پذیرتر و قدرتمندتری برای شبیه‌سازی رفتار انسان ارائه

¹ Synthesis of Integral Design

کردند. برخی از این رویکردها مبتنی بر روش‌های جدید در هوش مصنوعی و به ویژه یادگیری ماشینی و داده کاوی همراه با مکانیزم‌های بازخورد^۱ هستند.

- در مجموع امروزه به جای عبارت «سیستم‌های خبره» از عبارت «سیستم‌های هوشمند» یا Intelligent Systems استفاده می‌شود.

زمستان هوش مصنوعی

در تاریخ هوش مصنوعی، زمستان هوش مصنوعی دوره کاهش بودجه و علاقه به تحقیقات هوش مصنوعی است. سال ۱۹۹۰ کمترین میزان رغبت به هوش مصنوعی ثبت شد اما پس از آن کم‌کم علاقه و سرمایه‌گذاری بهبود یافت تا اینکه در سال ۲۰۱۲ یک افزایش شگفت‌انگیز در سرمایه‌گذاری در هوش مصنوعی شکل گرفت.

^۱ به سیستمی که خروجی‌اش به عنوان ورودی برای همان سیستم استفاده شود، در اصطلاح سیستم با مکانیزم بازخورد گفته می‌شود.

Feedback occurs when outputs of a system are routed back as inputs as part of a chain of cause-and-effect that forms a circuit or loop.

معرفی ابزار ES-Builder

<https://www.mcgoo.com.au/esbuilder/>



Attribute

همان if (اگر) است. یعنی همان سؤالی که از کاربر می پرسید.

Value

همان then (آنگاه)ها هستند. یعنی پاسخهای شما به سؤالات.

Conclusion

نتیجه نهایی یک مجموعه گزینش پاسخها توسط کاربر.

مثالی از یک سیستم خبره پشتیبانی یک پروژه:



ES-Builder Web - Decision T

Testa ES (38342) - By HR N (Last saved: 14th N

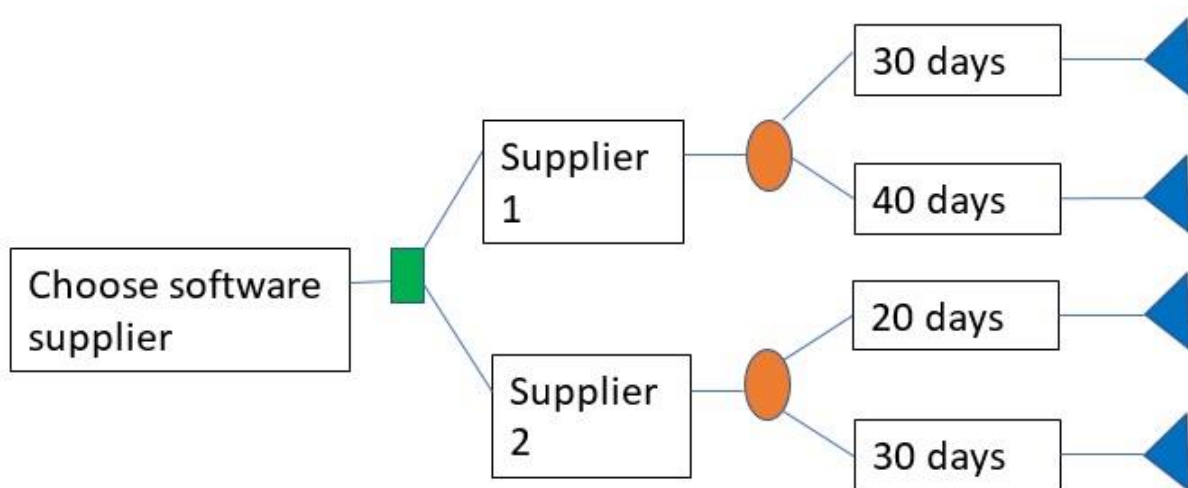


نحوه تبدیل درخت تصمیم‌گیری به سیستم خبره

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

در درخت تصمیم‌گیری، هر گره، در حقیقت یک «آزمون» بر روی یک «صفت» است. (مثلاً آیا حیوان مورد نظر، خون گرم است یا خون سرد؟) هر شاخه، در حقیقت خروجی آن آزمون است و هر برگ، برچسبی است که در نتیجه محاسبه همه صفات به داده مورد نظر نسبت داده می‌شود. مسیری که از ریشه شروع شود و به برگ ختم شود، در واقع یک قانون (rule) در سیستم خبره خواهد بود.

با توجه به اینکه درخت تصمیم‌گیری پیوسته است، ممکن است رسم آن به صورت افقی، مشکل باشد؛

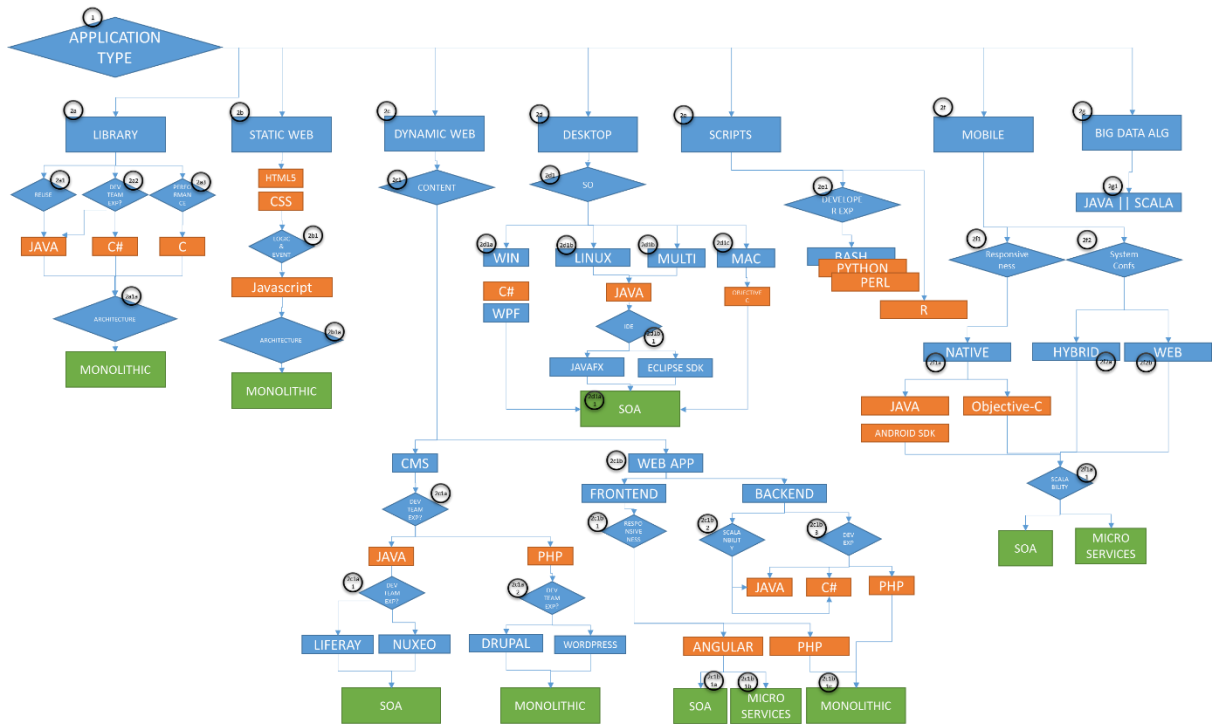


بنابراین می‌توان یکی از دو راه حل زیر را جایگزین کرد:

- نوشتن قوانین به جای رسم درخت:

if condition1 and condition2 and condition3 then outcome.

- رسم درخت تصمیم‌گیری به صورت عمودی:



معماری نرم‌افزاری سیستم‌های خبره

سیستم خبره نمونه‌ای از یک سیستم دانش‌محور است. سیستم‌های خبره اولین سیستم‌های تجاری بودند که از معماری دانش‌محور استفاده کردند. یک سیستم دانش‌محور اساساً از دو زیر-سیستم تشکیل شده است: پایگاه دانش (Knowledge-base) و موتور استنتاج (Inference Engine).

پایگاه دانش بیانگر حقایق (facts) در مورد جهان است. در سیستم‌های خبره اولیه مانند Mycin و Dendral، این حقایق عمدتاً به عنوان مقادیر ساده‌ای که به متغیرها نسبت داده می‌شدند، در نظر گرفته می‌شدند، اما در سیستم‌های خبره بعدی که با Shell‌های تجاری توسعه یافتند، پایگاه دانش ساختار به‌روزتر و قوی‌تری به خود گرفت و از مفاهیم برنامه‌نویسی شیء‌گرا استفاده کرد. جهان به عنوان کلاس‌ها، زیر-کلاس‌ها و نمونه‌ها (instance) نمایش داده شد و ویژگی‌ها به عنوان مقادیر صفات شیء در نظر گرفته می‌شد و قوانین همان پرس‌وجوی مقادیر و نسبت دادن آن‌ها به صفات اشیاء به حساب می‌آمد.

موتور استنتاج یک سیستم استدلال خودکار است که وضعیت فعلی پایگاه دانش را ارزیابی می‌کند، قوانین مربوطه را اعمال می‌کند و دانش جدید را در پایگاه دانش قرار می‌دهد. موتور استنتاج همچنین ممکن است دارای توانایی توضیح (Explanation) باشد، بنابراین می‌تواند زنجیره استدلال مورد استفاده برای رسیدن به یک نتیجه خاص را با ردیابی مجدد قوانینی که منجر به ادعا شده است، برای کاربر توضیح دهد.

برای درک بهتر نحوه عملکرد موتور استنتاج، سعی کنید تشخیص دهید عکس زیر مربوط به چه حیوانی است. به پردازش‌هایی که در مغز شما اتفاق می‌افتد دقت کنید:



برای موتور استنتاج عمدتاً دو حالت وجود دارد: «زنجیره پیش‌رو یا زنجیره جلوسو» (Forward Chaining) و «زنجیره پس‌رو یا زنجیره پشت‌سو» (Backward Chaining). در زنجیره پیش‌رو از سمت چپ یک قانون به

سمت راست آن می‌رسیم و در زنجیره پس‌رو از سمت راست یک قانون به سمت چپ می‌رسیم. به عنوان مثال، قانون زیر را در نظر بگیرید:

if “the temperature is 100+” then “the water is boiling”

یعنی اگر «دما ۱۰۰ درجه است» آنگاه «آب می‌جوشد».

اگر با دیدن یک آب که در حال گرم شدن در دمای ۹۰ درجه است به این نتیجه برسیم که طبق قانون بالا آب به زودی جوش خواهد آمد؛ این یک استنتاج پیش‌رو است. اما اگر با دیدن یک آب که در حال جوشیدن است به این نتیجه برسیم که دقایقی پیش دمای آب از ۱۰۰ درجه عبور کرده، این یک استنتاج پس‌رو است.

با تکامل سیستم‌های خبره، بسیاری از **تکنیک‌های جدید** در انواع مختلف موتورهای استنتاج گنجانیده شدند. برخی از مهمترین این موارد عبارتند از:

۱. Truth Maintenance (حفظ درستی داده‌ها): این تکنیک کمک می‌کند که اگر

یک دانش به هر دلیلی تغییر کرد، دانش‌های مرتبط با آن نیز تغییر کند تا صحت و درستی نتایج مختل نشود؛ مثلاً: تا پیش از این می‌دانستید که لمس سطوح آلوده به کرونا موجب بیماری کرونا می‌شود؛ حالا پژوهش‌ها نشان می‌دهد که اینطور نیست. مغز شما یک سیستم حفظ درستی دارد که دانش‌های مغز شما را متناسب با این دانش جدید تغییر می‌دهد...

۲. Hypothetical reasoning (استدلال فرضی): این تکنیک کمک می‌کند که

همه احتمالات (possibilities) یک دانش به طور همزمان (Parallel) بررسی شود؛ مثلاً: فرض کنید به شما اعلام کنند که یک نفر از دنیا رفته. مغز شما تمام احتمالات ممکن برای دلیل آن را بررسی می‌کند: با احتمال چقدر کرونا گرفته، با احتمال چقدر تصادف کرده و...

۳. Uncertainty systems (سیستم‌های عدم قطعیت): در این نوع سیستم‌ها فقط دو

حالت قطعی 0 و 1 تعریف نمی‌شود بلکه هر حالتی بین این دو حالت نیز ممکن است تصور شود. به این ایده در اصطلاح منطق فازی (Fuzzy Logic) گفته می‌شود.

۴. Ontology classification (دسته‌بندی بر اساس هستی‌شناسی): سیستم خبره

می‌تواند برای درک بهتر دانشی که دارد به علم هستی‌شناسی (آنولوژی) متوسل شود.

نمایه‌سازی

تعریف نمایه^۱

An Index is a list of words or phrases (heading) and associated Pointers (locators) to where useful material relating to that heading can be found in a Document.

نمایه، لیستی است از لغات یا عبارات (که به آن سرآیند یا heading گفته می‌شود) به انضمام اشاره‌گرها (که به آن locators یا «جای‌نما» گفته می‌شود) به محل‌هایی که مطالب مفید و مرتبط با آن سرآیند در سند مورد نظر قابل مشاهده است.

تعاریف دیگر از نمایه

- به طور ساده نمایه، سیاهه یا لیستی نظام یافته، برای هدایت منظم یک متن، محتوا، مجموعه‌ای از مدارک یا هر گونه اطلاعات ضبط شده است.
- نمایه استفاده‌کنندگان را از اطلاعات معلوم به اطلاعات اضافی ناشناخته هدایت می‌کند.

دلایل نیاز به نمایه‌سازی

- ۱- تحلیل مؤثر و سریع اسناد
- ۲- افزایش حجم اطلاعات
- ۳- سرعت پایین بازیابی اطلاعات در روش‌های کتابداری سنتی
- ۴- تغییر روش‌های ذخیره و بازیابی اطلاعات

اهداف نمایه‌سازی

- ۱- زمان و تلاش برای یافتن اطلاعات را به حداقل میزان کاهش دهد.
- ۲- موفقیت جستجوی استفاده‌کننده را به حداکثر برساند.
- ۳- برقراری ارتباط میان مفاهیم
- ۴- تنظیم شناسه‌ها به ترتیبی نظام‌مند و مؤثر
- ۵- شناسایی سریع مدارک در یک مجموعه

۶- سازماندهی اطلاعات به قصد بازیابی سریع و آسان مدارک

به طور خلاصه، هدف نمایه‌سازی عبارت است از: **انتخاب واژه مناسب با زمان کم و تلاش کمتر**

نمایه یک فرا داده است

تعریف فراداده یا Metadata

به اطلاعات اضافه که درباره یک داده ثبت می‌شود تا در طبقه‌بندی از آن‌ها استفاده شود گفته می‌شود. (داده‌ای در مورد داده)

مثال ۱: مشخصات عکسی که توسط دوربین دیجیتال گرفته می‌شود (مانند زمان گرفتن عکس، سایز عکس، نام و مدل دوربین، نام و مدل لنز دوربین، موقعیت جغرافیایی محل گرفتن عکس و ...) در header عکس ذخیره می‌شود. این اطلاعات به دسته‌بندی عکس توسط برنامه‌های سیستم عامل کمک می‌کند. (به طور مثال در سیستم عامل مک، عکس‌ها به طور خودکار نسبت به محلی که گرفته شده‌اند دسته‌بندی می‌شوند: عکس‌های شمال، عکس‌های مشهد و...)

مثال ۲: در طراحی سایت، اطلاعاتی اضافه در مورد هر صفحه در بین تگ `<head>` در داخل یک `<meta>` قرار داده می‌شود. این اطلاعات به موتورهای جستجو در دسته‌بندی سایت‌ها کمک می‌کند.

آنتولوژی چیست؟

آنتولوژی در لغت از دو کلمه `onto` به معنی «هستی» و `logy` به معنی «مطالعه» تشکیل شده است. ریشه آنتولوژی در فلسفه است و به ارسطو نسبت داده می‌شود. در علوم کامپیوتر و در حوزه وب معنایی، آنتولوژی مفهوم کلمات و ارتباط بین آن‌ها در حوزه‌ای که مورد استفاده قرار می‌گیرند را نشان می‌دهد. امروزه با حجم عظیمی از اطلاعات مواجه هستیم. به منظور دستیابی مؤثر به اطلاعات، به سیستمی نیاز است که بتواند به طور مناسبی، اطلاعات را از وب استخراج کرده و به طور مناسبی به کاربران ارائه دهد، اما این کار به سادگی میسر نمی‌باشد. آنتولوژی به وجود آمده است تا این مشکلات را مرتفع سازد.

انواع وب

۱- وب ۱ (وب سنتی) (Traditional Web)

این وب همان وب اولیه است که توسط آقای تیم برنرزلی ابداع شد. حاوی صفحات ساده‌ی بدون جذابیت و بدون هوشمندی.

۲- وب ۲ (وب اشتراکی) (Social Web)

در این دوران سایت‌های اشتراک جمعی یا شبکه‌های اجتماعی (Social Networks) مثل فیس‌بوک، توئیتر، اینستاگرام و... ظهور کردند و کاربران توانستند داده‌های خود را به راحتی با یکدیگر به اشتراک بگذارند و بنابراین دسترسی به داده‌ها آسان‌تر شد.

۳- وب ۳ (وب معنایی یا وب هوشمند) (Semantic Web)

در این وب پیش‌بینی می‌شود صفحات وب بیش از گذشته برای موتورهای جستجو به طور کامل شناخته شده باشند، بنابراین نسبت به نیاز کاربر بهترین نتایج یافته و نمایش داده خواهد شد.

روش‌های معنا بخشیدن به صفحات وب

۱- XML

۲- URI

۳- RDF

XML چیست؟

XML مخفف Extensible Markup Language و به معنی «زبان نشانه گذاری قابل توسعه» است.

فایل XML یک فایل متنی ساده و یک پایگاه داده قابل حمل به حساب می آید.

در این زبان تگ ها توسط برنامه نویسی تعریف می شود و خود زبان، تگ های مشخصی ندارد. (پس یعنی دیگران

هستند که زبان را «توسعه» می دهند. پس این، یک «زبان قابل توسعه» است)

معمولاً هر تگ والد، نماد یک موجودیت و تگ های داخلی، صفات آن موجودیت یا همان سرستون ها در پایگاه

داده هستند.

این زبان امروزه در تمامی نرم افزارها برای انتقال اطلاعات یا تنظیمات از یک محل به محل دیگر کاربرد دارد.

مثال: جدول زیر را به زبان XML تبدیل کنید؟

books

A1	A2	A3	A4
E1	E2	E3	E4
D1	D2	D3	D4
S1	S2	S3	S4

<BOOKS>

<BOOK>

<A1>E1</A1>

<A2>E2</A2>

<A3>E3</A3>

<A4>E4</A4>

</BOOK>

<BOOK>

<A1>D1</A1>

<A2>D2</A2>

<A3>D3</A3>

<A4>D4</A4>

</BOOK>

<BOOK>

<A1>S1</A1>

<A2>S2</A2>

<A3>S3</A3>

<A4>S4</A4>

</BOOK>

</BOOKS>

مثال: جداول موجود در پایگاه داده‌ی یک کتابخانه به شکل زیر است، از داده‌های آن یک خروجی XML بگیرید.

Books

bid	Code	ISBN	title	author	publisher	pubDate
1	1001	978-964-531-102-3	Learn Internet	Hamid Reza Niroomand	Aftabgardan	2015
2	1002	978-964-531-302-1	C# in 24 Hours	Qomi	Naqous	2010

Members

mid	Code	FirstName	LastName
1	100	Ali	Hasani
2	101	Hojjat	Rahimi

```
<library>
  <books>
    <book>
      <bid>1</bid>
      <code>1001</code>
      <ISBN>978-964-531-102-3</ISBN>
      <title>Learn Internet</title>
      <author>Hamid Reza Niroomand</ author>
      < publisher >Aftabgardan</ publisher >
      <author>2015</ author>
    </book>
    <book>
      <bid>2</bid>
      <code>1002</code>
      <ISBN>978-964-531-302-1</ISBN>
      <title>C# in 24 Hours</title>
      <author>Qomi</ author>
      < publisher >Naqous</ publisher >
      <author>2010</ author>
    </book>
  </books>
</library>
```



```
<members>
  <member>
    <mid>..</mid>
    به همین صورت، مانند بالا ادامه دهید...
  </member>

</members>
...
```

```
</library>
```

URI چیست؟

مخفف Uniform Resource Identifier و به معنی «شناسه یک شکل منابع» و شناسه یا آدرسی است که از روی اجزای آن می توان اطلاعات کافی در مورد آن منبع از جمله رده یک شی و رده های بالاتر آن به دست آورد.

مثالی از یک URL بدون به کارگیری مفاهیم وب معنایی:

```
http://aftab.cc/linkestan/showlink.php?id=1400
```

از روی این آدرس نمی توان فهمید که این لینک در مورد چه چیزی است.

همان لینک با به کارگیری مفاهیم وب معنایی:

```
http://aftab.cc/linkestan/link/pdf/learn-c-sharp-in-24-hours
```

از روی این لینک می توان فهمید که لینک به یک pdf در مورد آموزش C# است.

URL چیست؟

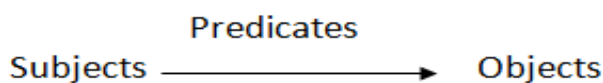
مخفف Uniform Resource Locator و به معنی «مشخص کننده ی موقعیت منبع به صورت یک شکل» است که یک نوع URI به شمار می آید. منظور از URL همان آدرس عکس ها و ویدئوها و صفحات روی اینترنت است.

RDF چیست؟

مخفف Resource Description Framework و به معنی «چهارچوب توصیف منابع» است. زبانی است که به لغات یک صفحه برای موتورهای جستجو معنا می بخشد و ارتباط بین لغات را مشخص می کند. با استفاده از XML می توانیم موجودیت های مختلف و صفات آنها را تک گذاری کنیم اما معنای آنها را نمی توانیم بیان کنیم به عنوان مثال: اگر کسی برچسب قیمت را با PRICE نشانه گذاری کند و شخص دیگری با COST نشانه گذاری کند هیچ راهی برای فهم مترادف بودن این دو عبارت به وسیله XML وجود ندارد، اینجاست که RDF به کار می آید.

هر عبارت RDF به صورت یک سه-تایی (Tripple) بیان می شود: فاعل، گزاره، مفعول.

۳ موضوع اصلی در RDF

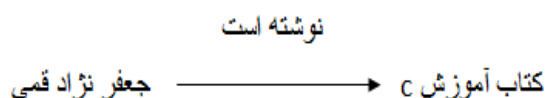


۱- Subjects: فاعل جمله

۲- Predicates: صفت یا فعلی که فاعل آن را انجام می دهد.

۳- Objects: مفعول یا اسمی که جانشین مفعول شده است و کار، روی آن انجام می شود.

مثال:



مزایای RDF نسبت به XML:

۱- RDF یک فهم مشترک از یک مفهوم واحد را ارائه می دهد زیرا گاهی اوقات ما برای یک مفهوم از دو لغت متمایز استفاده می کنیم یا بلعکس از یک لغت برای دو مفهوم متفاوت استفاده می کنیم.

۲- RDF بین مفاهیم در وب و دنیای واقعی ارتباط برقرار می کند.

مثال: جمله «آقای تیم برنزی وب را اختراع کرد» را به صورت RDF نمایش دهید.

<وب#><اختراع کرد#><آقای تیم برنزی#>

روش اعمال RDF به یک صفحه با HTML5

مثال: عبارت [نام من حمید رضا نیرومند است. من ایران را دوست دارم. کتاب مورد علاقه من «آینده خود را خلق کنید» نوشته «برایان تریسی» است] را به صورت HTML5+RDFa بنویسید؟

```
<html prefix="dc: http://purl.org/dc/elements/1.1/" lang="fa">
  <head>
    <title>سایت شخصی حمید رضا نیرومند</title>
  </head>
  <body about="http://niroomand.ir">
    <h1>صفحه اصلی</h1>
    <p>است و من <span property="foaf:nick">حمید رضا نیرومند</span> نام من<br>
    <a href="http://www.iran.ir/" rel="foaf:interest" lang="fa">ایران</a>
    را دوست دارم.
  </p>
  <p>
    <span rel="foaf:interest" resource="urn:ISBN:0752820907">
      <span about="urn:ISBN:0752820907">
        <cite property="dc:title">
          آینده خود را خلق کنید
        </cite> نوشته
        <span property="dc:creator">برایان تریسی</span> است</span></span>.
      </p>
    </body>
  </html>
```

- ۱- در تگ html با استفاده از صفت prefix نسخه rdf ای که استفاده می کنید را تعیین کنید. (این آدرس در حقیقت آدرس یک Vocabulary است و به موتور جستجو می فهماند که کلماتی که شما برای معنا بخشیدن به عبارات استفاده کرده اید از آن «فرهنگ لغات» یا «فضای نام» انتخاب شده. مثلاً وقتی می خواهید بگویید «ایران»، یک علاقه مندی برای «حمید رضا نیرومند» است، از چه کلمه ای استفاده کرده اید و کجا تعریف شده که این کلمه به معنی علاقه مندی است؟)
- ۲- در تگ body (یا هر تگ کلی دیگر) یک صفت about اضافه کرده و نام فاعل (یا یک آدرس اینترنتی در مورد فاعل) که این صفحه در مورد آن فاعل است را بیان کنید.
- ۳- در صفحه، کلمات مهم مانند نام شخصیت، نام کتاب، .. که با فاعل رابطه ای دارند را با تگی مانند احاطه کنید و به این تگ یک صفت rel (مخفف relation به معنی رابطه) اضافه کنید و در مقدار آن، رابطه ای آن کلمه با فاعل را بنویسید. (مثلاً «کتاب آینده خود را خلق کنید» به عنوان interest یعنی علاقه مندی برای صاحب این صفحه معرفی شده)

۴- اگر کلمه‌ای در صفحه بود که صفت فاعل بود را با تگ span احاطه کنید و یک صفت property به آن اضافه کرده و بگویید که این کلمه چه صفتی برای فاعل است. (مثلاً «حمید رضا نیرومند» یک nickname یا اسم و فامیل برای فاعل به حساب می‌آید)

نکته: توجه کنید که نام صفات و ارتباطات باید حتماً در vocabulary که در ابتدای صفحه به آن آدرس داده‌اید وجود داشته باشد.

مدل مخفی مارکوف چیست؟

یک مدل آماری است که در آن سیستم مدل شده به صورت یک فرایند مارکوف با State‌های پنهان فرض می‌شود این مدل برای یافتن State نهایی یک سیستم با توجه به State اولیه کاربرد دارد. (state=حالت) به طور خلاصه:

یعنی ما با توجه به دانسته‌های قبلی و با توجه به یک سری داده اولیه یا حالت آغازین، به یک حالت یا State پایانی برسیم.

مثال:

دو برادر و خواهر به نام‌های آلیس و باب را در نظر بگیرید. آن‌ها دور از هم زندگی کرده و در انتهای هر روز درباره کارهای روزمره‌شان با هم تلفنی صحبت می‌کنند. فعالیت‌های باب شامل "قدم زدن در پارک"، "خرید کردن" و "تمیز کردن آپارتمان" می‌شود. انتخاب اینکه هر روز کدام کار را انجام دهد منحصراً بستگی به هوای همان روز دارد. آلیس اطلاع دقیقی از هوای فعلی محل زندگی باب ندارد ولی از تمایلات کلی وی آگاه است (بنا به نوع هوا چه کاری را دوست دارد انجام دهد). بر اساس گفته‌های باب در پایان روز قبل، آلیس سعی می‌کند هوای آن روز را حدس بزند.

در حقیقت آلیس هوا را یک زنجیره گسسته مارکوف می‌پندارد که دو حالت "بارانی" و "آفتابی" دارد. اما به طور مستقیم هوا را مشاهده نمی‌کند. بنابراین حالات هوا بر او مخفی است. در هر روز احتمال اینکه باب به "قدم زدن"، "خرید کردن" و "تمیز کردن" پردازد بستگی به هوا داشته و دارای یک احتمال مشخص است و این احتمال را آلیس به مرور و بر اساس صحبت‌های قبلی باب یاد گرفته است. مشاهدات مسأله شرح فعالیت‌هایی است که باب در انتهای هر روز به آلیس می‌گوید.

حالا تصور کنید الان آلیس می‌خواهد به باب زنگ بزند و می‌خواهد بداند او خانه است یا بیرون از خانه؟

مسئله بالا توسط الگوریتم آقای مارکوف به صورت زیر مدل می‌شود:

```
states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
    'Rainy': {'Rainy': 0.7, 'Sunny': 0.3},
    'Sunny': {'Rainy': 0.4, 'Sunny': 0.6},
}

emission_probability = {
    'Rainy': {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
    'Sunny': {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
}
```

حالات کلی: بارانی یا آفتابی بودن هوا

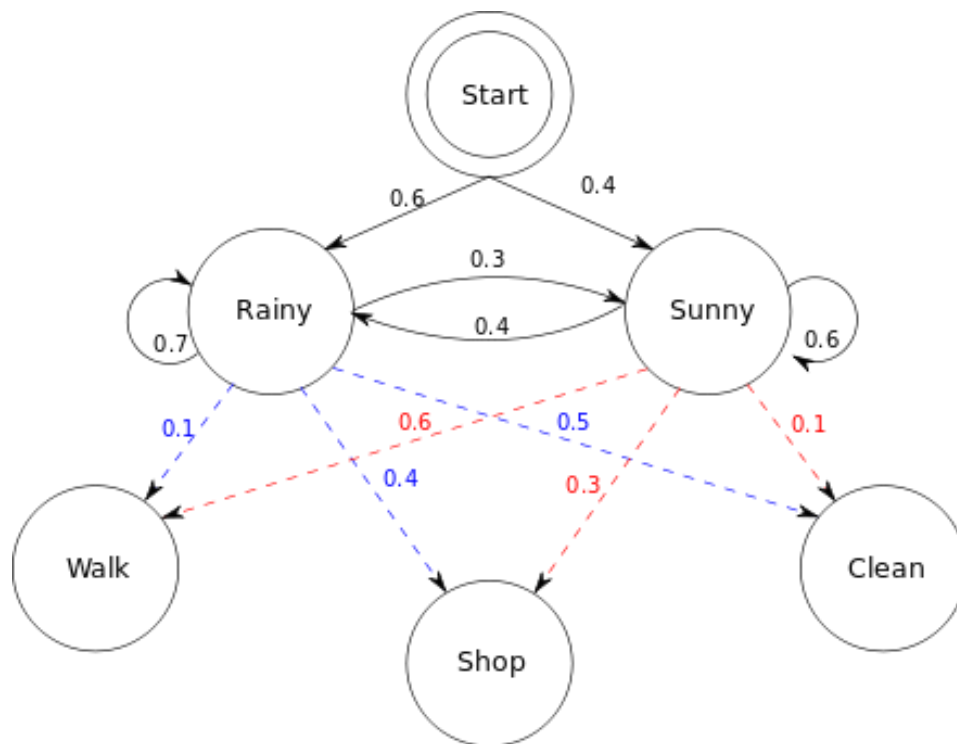
مشاهدات: قدم زدن، خرید کردن، تمیز کردن خانه

احتمالات آغازین (یعنی احتمالاتی که آلیس به مرور و طی مدت‌ها صحبت یاد گرفته): به احتمال ۶۰ درصد هوای محل زندگی باب بارانی است و به احتمال ۴۰ درصد آفتابی.

احتمال انتقال از یک حالت به حالت دیگر: اگر دیشب باب گفته باشد که هوا بارانی است، به احتمال ۷۰ درصد امروز نیز بارانی است و به احتمال ۳۰ درصد آفتابی است و اگر دیشب گفته باشد که هوا آفتابی است، به احتمال ۴۰ درصد امروز آنجا بارانی است و به احتمال ۶۰ درصد همچنان آفتابی است...

احتمال بروز یک کار: اگر بارانی باشد به احتمال ۱۰ درصد باب بیرون قدم می‌زند، به احتمال ۴۰ درصد خرید می‌کند و به احتمال ۵۰ درصد خانه است و خانه را تمیز می‌کند. و اگر هوا آفتابی باشد به احتمال ۶۰ درصد قدم می‌زند، به احتمال ۳۰ درصد خرید است و به احتمال ۱۰ درصد در حال تمیز کردن خانه.

این مدل را می‌شود به صورت گراف زیر نشان داد:



حالا می توان راحت تر پاسخ داد: آلیس می خواهد به باب زنگ بزند، دیشب باب گفته است که هوا آنجا بارانی بوده. به نظر شما الان باب در حال انجام چه کاری است؟

پاسخ: نگاه کنید به حالت یا State یا همان دایره ی Rainy، یال هایی که عدد بیشتری دارند را دنبال کنید... به احتمال ۷۰ درصد به خودش می رود (یعنی وقتی گفته دیشب بارانی بوده به احتمال ۷۰ درصد امروز هم بارانی است) و به احتمال ۵۰ درصد به Clean می رود، پس او در حال نظافت و در خانه است...

کاربردهای مدل مخفی مارکوف

این مدل در زمینه های بسیار زیادی کاربرد دارد:

- تبدیل گفتار به صدا (تشخیص گفتار):

تصور کنید شما در حال خواندن یک متن برای یک نرم افزار هستید که او آنرا تایپ کند. اگر تلفظ یک کلمه شبیه یک کلمه دیگر باشد (مثلاً می گوئید: Important Object نرم افزار ممکن است Import an Object بشود...) چگونه تشخیص دهد که کدام منظور شما بوده؟ با مدل مخفی مارکوف

و چیزهایی که قبلاً یادش داده اید...

- تشخیص چهره:

مثلاً تصور کنید یک نرم افزار تشخیص چهره، فاصله بین دو مردمک چشم را ملاک شناسایی قرار دهد. اگر برای دو نفر از افراد یک سازمان این فاصله یکسان باشد، کدام یکی الان جلو دستگاه ایستاده؟ با توجه به دانسته‌های دیگر می‌شود فهمید...

- ترجمه ماشینی:

همه با Google Translate کار کرده‌اند و می‌دانند که گاهی چقدر اشتباه ترجمه می‌کند! دلیل؟ هنوز دیتابیس مربوط به مدل مخفی مارکوف گوگل کامل نشده. به همین دلیل است که گوگل، Community Help را راه‌اندازی کرده و از شما انسان‌ها خواسته به مرور به آن نرم‌افزار یاد بدهید که وقتی فلان کلمه کنار فلان کلمه قرار می‌گیرد ترجمه‌اش چه می‌شود؟

- پیش‌بینی ژن

- هم تراز کردن توالی:

تشخیص قرار گرفتن ماشین بر روی خط، تشخیص در یک راستا قرار گرفتن نقاط یک خط

- تشخیص فعالیت:

پزشک هوشمند، خانه‌های هوشمند

مزایای سیستم‌های خبره

هدف سیستم‌های مبتنی بر دانش این است که اطلاعات حیاتی لازم برای کار سیستم را دقیق و روشن کند (و نه مبهم). در یک برنامه کامپیوتری سنتی، منطق در کد برنامه تعبیه شده است و معمولاً فقط توسط یک متخصص فناوری اطلاعات قابل بررسی است. در یک سیستم خبره هدف، تعیین قوانین در قالبی بود که به راحتی قابل فهم، بازبینی و حتی ویرایش توسط متخصصان حوزه مرتبط با آن سیستم خبره باشد. این نوع نمایش دانش، باعث می‌شود توسعه سیستم سریع‌تر و نگهداری آن آسان‌تر شود.

نمونه‌ای از تعریف آسان قانون توسط یک نابرنامه‌نویس: سایت ifttt.com

هر چند هدف سیستم‌های خبره آسان‌سازی تعریف قوانین توسط کاربران نابرنامه‌ساز بود اما در عمل، تعریف قوانین به شکل ساده‌تر نیز مشکلات خاص خودش را داشت؛ از جمله: آن قوانین نیز مانند کدهای برنامه‌نویسی ساختارهای خاص خود را داشت که در صورت رعایت نکردن، خطاهایی رخ می‌داد. و از طرفی مباحث مربوط به یکپارچه‌سازی سیستم‌ها با یکدیگر و همچنین پایگاه داده‌های حجیم چالش دیگری برای سیستم‌های خبره بود.

معایب سیستم‌های خبره

رایج‌ترین نقطه ضعف ذکر شده برای سیستم‌های خبره در ادبیات دانشگاهی، مسأله کسب دانش است. با توجه به اینکه دسترسی به یک متخصص انسانی برای کسب دانش یک چالش بزرگ است، در سال‌های اخیر، بسیاری از پژوهش‌ها به سمت تولید ابزارهایی برای کسب دانش رفته است تا فرایندهای تولید، خطایابی و نگهداری قوانینی که قرار بوده توسط انسان خبره انجام شود را این ابزارها انجام دهند.

معایب دیگری که در همه سیستم‌های کامپیوتری مشاهده می‌شود در سیستم‌های خبره نیز وجود دارد: یکپارچه‌سازی، کلان‌داده‌ها، کارایی و...

کاربردهای سیستم‌های خبره

شاخه	توضیح	مثال
تفسیر (Interpretation)	استنباطِ توصیفِ موقعیت از طریق داده‌های سنسور (یا حسگر)	سیستم تشخیص حریق، سیستم تشخیص تصادف در ماشین تسلا
پیش‌بینی (Prediction)	استنباطِ عواقب احتمالی موقعیت‌های فرضی	سیستم پیش‌بینی وضع هوا، پیش‌بینی بورس (Meta Trader) و...
تشخیص عیب (Diagnosis)	استنباط عملکرد ناصحیح سیستم از روی مشاهدات	سیستم‌های تشخیص بیماری، آنتی‌ویروس‌ها، سیستم دیاگ خودرو (عیب‌یابی خودرو) و...
طراحی (Design)	پیکربندی اشیاء بر اساس یک سری قید (شرط)	سیستم خبره پیشنهاد طراحی مناسب برای یک سایت، سیستم خبره پیشنهاد معماری مناسب برای ساختمان و...
برنامه‌ریزی (Planning)	طراحی عملیات / طراحی انجام کارها	سیستم خبره برنامه‌ریزی پروژه، سیستم خبره برنامه‌ریزی تحصیلی (کنکور و...) و...
نظارت (Monitoring)	مقایسه مشاهدات برای برنامه‌ریزی آسیب‌پذیری‌ها	سیستم‌های خبره تشخیص نفوذ به شبکه (سیستم‌های IDS^1)، سیستم تشخیص حضور دزد در خانه یا خانه‌باغ و... پیشنهاد: برای سیستم‌های نرم‌افزاری که تولید می‌کنید، یک IDS در بطن آن‌ها قرار دهید.
اشکال‌زدایی (Debugging)	ارائه راه‌حل‌های گام به گام برای مسائل پیچیده	دیاگرام‌های Troubleshooters در ویندوز و...

¹ An Intrusion Detection System (IDS) is a monitoring system that detects suspicious activities and generates alerts when they are detected

<p>پیشنهاد: برای سیستم‌های نرم‌افزاری که تولید می‌کنید، یک Troubleshooter در بطن آنها قرار دهید تا کاربران سیستم با کمک آن بتوانند مشکلات عمومی خود را حل کنند. حالت Dev_Mode می‌تواند بخشی از این پیاده‌سازی باشد.</p>		
<p>سیستم خبره‌ای که با نصب آن، باگ‌های امنیتی سیستم عامل اصلاح شود.</p>	<p>اجرای یک طرح و برنامه برای به انجام رساندن یک نسخه‌ی تجویز شده</p>	<p>تعمیرات (Repair)</p>
<p>سیستم خبره آموزش زبان،</p>	<p>تشخیص، ارزیابی و اصلاح رفتار دانش‌جویان</p>	<p>آموزش (Instruction)</p>
<p>سیستم خبره فضاپیمای شاتل، سیستم خبره کنترل رفتار فرزند در ویندوز (سیستم Parental Control)، سیستم خبره کنترل از راه دور خانه و... پیشنهاد: درباره IoT مطالعه کنید. پیشنهاد: درباره سوئیچ هوشمند سونوف تحقیق کنید.</p>	<p>تفسیر، پیش‌بینی، تعمیر و نظارت رفتارهای سیستم</p>	<p>کنترل (Control)</p>

سؤال: تفاوت Diagnosis و Debugging و Repair را با یک مثال در بحث خودرو یا پزشکی و... تبیین کنید.

سؤال: جارو برقی هوشمند را از نظر سیستم خبره بودن تشریح کنید.

سیستم‌های تشخیص الگو یا Pattern Recognition Systems

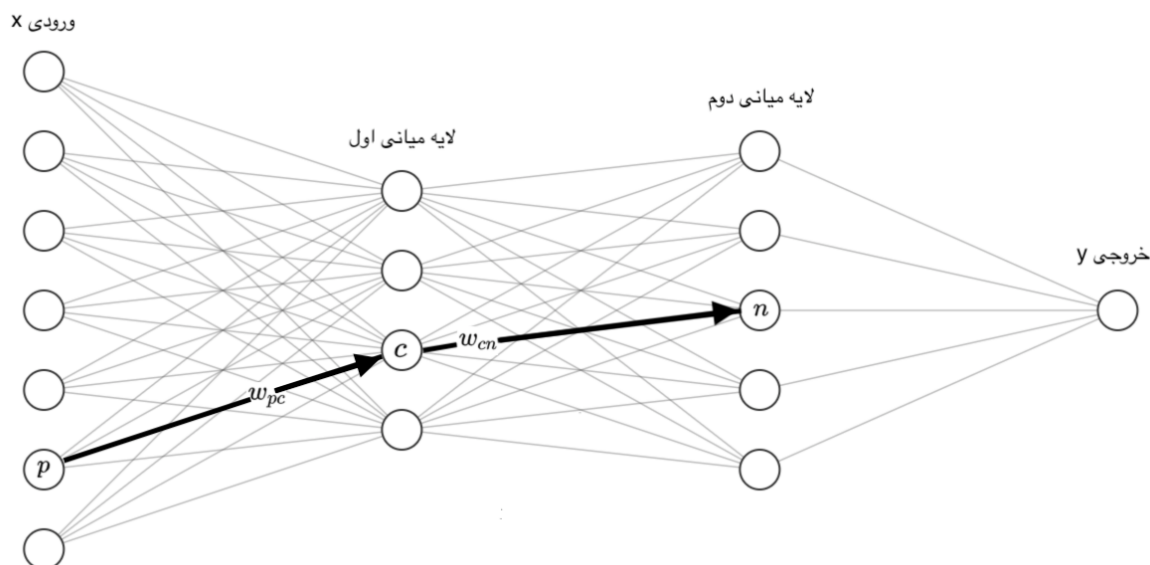
سیستم‌هایی که به دنبال یک الگو در داده‌های نویزدار (Noisy Data) می‌گردند؛ مثال: فرض کنید داده‌های مربوط به خریدهای مشتریان دیجی کالا را الگویابی کنیم. فکر می‌کنید چه الگوهایی از این داده‌ها استخراج خواهد شد؟

نکته: Voice Recognition یک نوع Pattern Recognition به حساب می‌آید.

با توجه به اینکه بسیاری از مسائل تشخیص الگو، پیچیده‌تر از آن هستند که قابل پیاده‌سازی با راهکار rule-based (مانند سیستم خبره) باشند، راهکارهای دیگر مانند «شبکه‌های عصبی در هوش مصنوعی» مطرح‌تر شدند.

ANN: Artificial Neural Networks

شبکه‌های عصبی مصنوعی یا شبکه‌های عصبی صنعتی (Artificial Neural Networks - ANN) یا به زبان ساده‌تر شبکه‌های عصبی سیستم‌ها و روش‌های محاسباتی نوین برای یادگیری ماشینی، نمایش دانش و در انتها اعمال دانش به دست آمده در جهت پیش‌بینی پاسخ‌های خروجی از سامانه‌های پیچیده هستند. ایده اصلی این گونه شبکه‌ها تا حدودی الهام گرفته از شیوه کارکرد سیستم عصبی زیستی برای پردازش داده‌ها و اطلاعات به منظور یادگیری و ایجاد دانش میباشد. عنصر کلیدی این ایده، ایجاد ساختارهایی جدید برای سامانه پردازش اطلاعات است.



این سیستم از شمار زیادی عناصر پردازشی فوق‌العاده بهم پیوسته با نام نورون تشکیل شده که برای حل یک مسئله با هم هماهنگ عمل می‌کنند و توسط سیناپس‌ها (ارتباطات الکترومغناطیسی) اطلاعات را منتقل می‌کنند. در این شبکه‌ها اگر یک سلول آسیب بیند بقیه سلول‌ها می‌توانند نبود آن را جبران کرده، و نیز در بازسازی آن سهیم باشند. این شبکه‌ها قادر به یادگیری‌اند. مثلاً با اعمال سوزش به سلول‌های عصبی لامسه، سلول‌ها یاد می‌گیرند که به طرف جسم داغ نروند و با این الگوریتم سیستم می‌آموزد که خطای خود را اصلاح کند. یادگیری در این سیستم‌ها به صورت تطبیقی صورت می‌گیرد، یعنی با استفاده از مثال‌ها وزن سیناپس‌ها به گونه‌ای تغییر می‌کند که در صورت دادن ورودی‌های جدید، سیستم پاسخ درستی تولید کند.

مهندسی دانش Knowledge engineering

مهندسی دانش (KE) به کلیه جنبه‌های فنی، علمی و اجتماعی مربوط به ساخت، نگهداری و استفاده از سیستم‌های مبتنی بر دانش گفته می‌شود.

مهندس دانش کسی است که می‌تواند سه فعالیت استخراج، تحلیل و مدل‌سازی دانش را انجام دهد. این سه فعالیت منجر به تولید یک پایگاه دانش ساخت یافته مبتنی بر مدل‌های دانش با قابلیت استفاده مجدد می‌شود که می‌تواند به عنوان محتوای ورودی در یک سیستم مبتنی بر دانش استفاده شود.

مهندسی مبتنی بر دانش (Knowledge-based engineering (KBE))

به کاربرد فناوری سیستم‌های مبتنی بر دانش در زمینه طراحی و ساخت در اصطلاح مهندسی مبتنی بر دانش گفته می‌شود.

مزایای استفاده از مهندسی دانش در صنعت:

- 1 - بهبود در یکپارچه‌سازی^۱
- 2 - بالا رفتن قابلیت استفاده مجدد^۲
- 3 - نگهداری بهتر سیستم‌ها^۳
- 4 - خودکارسازی بیشتر و بهتر امور (اتوماسیون)^۴

تمرین: برای هر یک از موارد بالا یک شغل سنتی را نام ببرید و بیان کنید که مهندسی دانش چگونه می‌تواند در آن شغل بهبود ایجاد کند.

مثال: برای اداره برق چه پیشنهادهایی در ۴ مورد بالا دارید؟

- 1 - کنتور برق را هوشمند و با سیستم بانکی یکپارچه کنیم تا در صورت عدم پرداخت به موقع، کنتور خود به خود قطع شود.
- 2 - تولید برق از حرکت ماشین‌ها روی یک مانع
- 3 - سیستم تشخیص منازل که رمز-ارز کشت می‌کنند (Farmها)

¹ Improved integration

² More re-use

³ Better maintenance

⁴ More automation

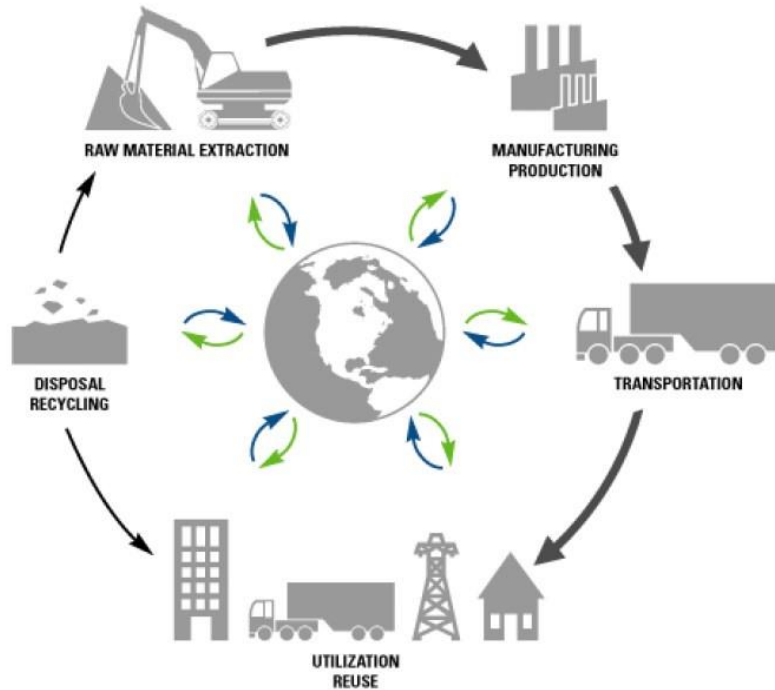
طراحی به کمک رایانه (CAD)

CAD مخفف Computer-aided Design است و به دسته‌ای از کاربردهای کامپیوتر که مرتبط با طراحی گرافیکی به کمک کامپیوتر است، اطلاق می‌شود. یکی از **اولین** نمونه‌های کاربرد مهندسی نرم‌افزار، استفاده در زمینه CAD بود.



PLM

مخفف Product Lifecycle Management است. در صنعت، مدیریت چرخه حیات محصول (PLM) فرآیند مدیریت کل چرخه حیات محصول از ابتدای تولید آن (از طریق مهندسی، طراحی و ساخت) و همچنین پشتیبانی و محو کردن محصولات تولیدی است. PLM افراد، داده‌ها، فرآیندها و سیستم‌های تجاری را یکپارچه می‌کند.



سیستم‌های PLM به سازمان‌ها کمک می‌کند تا با مشکلات روزافزون پیچیدگی و مهندسی تولید محصولات جدید برای بازارهای رقابتی جهانی کنار بیایند.

PLM را می‌توان یکی از چهار سنگ‌بنای ساختار مرتبط با فناوری اطلاعات هر شرکت دانست. این چهار سنگ‌بنا عبارتند از:

- CRM: Customer Relationship Management

- سیستم مدیریت ارتباط با مشتری
 - مدیریت تماس‌ها
 - مدیریت فاکتورها
 - مدیریت روابط بین مشتریان
 - مدیریت ملاقات‌ها
 - ارسال ایمیل و پیامک و... به همه مشتریان یا گروهی از مشتریان

.۴

- SCM: Supply Chain Management

- سیستم مدیریت زنجیره تأمین
 - عبارتست از فرایند برنامه‌ریزی، اجرا و کنترل عملیات مرتبط با زنجیره تأمین در بهینه‌ترین حالت ممکن. مدیریت زنجیره تأمین دربرگیرنده تمامی

جابجایی‌ها و ذخیره مواد اولیه، موجودی در حین کار و محصول تمام شده از نقطه شروع اولیه تا نقطه پایان مصرف می‌باشد.

- مدیریت ارتباط با مشتری
- مدیریت خدمات مشتری
- سبک مدیریت تقاضا
- تحقق سفارش
- مدیریت جریان تولید
- مدیریت رابطه تأمین‌کننده
- توسعه و تجاری‌سازی محصول
- مدیریت برگشت کالا



لیستی از نرم‌افزارهای SCM رایگان و توضیحات کامل درباره SCM در مطلب زیر موجود است:

<https://www.goodfirms.co/blog/free-open-source-supply-chain-management-software>

- ERP: Enterprise Resource Planning

- برنامه‌ریزی منابع سازمانی
- منظور از منابع، 4M است:
- Man (نیروی انسانی)

- Machine (ماشين آلات)
- Materials (مواد)
- Money (پول)

PLM: Product Lifecycle Management -

Knowledge representation and reasoning

Knowledge representation and reasoning (KR², KR&R) is the field of artificial intelligence (AI) dedicated to representing information about the world in a form that a computer system can utilize to solve complex tasks such as diagnosing a medical condition or having a dialog in a natural language.

بازنمایی دانش و استدلال (KR²، KR&R) زمینه‌ای از هوش مصنوعی (AI) است که مربوط است به نمایش اطلاعات جهان در شکلی که یک سیستم رایانه‌ای بتواند با استفاده از آن وظایف پیچیده مانند تشخیص یک بیماری پزشکی یا گفتگو به زبان طبیعی را به انجام رساند.

بازنمایی دانش با یافته‌هایی از رشته روانشناسی درگیر است؛ در مورد اینکه انسان‌ها چگونه مسائل را حل می‌کنند و یا دانش را نمایش می‌دهند. تا از این طریق بتوان فرمول‌هایی یافت که طراحی و ساخت سیستم‌های پیچیده را آسان‌تر کند. بازنمایی دانش و استدلال همچنین با یافته‌هایی از منطق نیز درگیر است؛ برای مثال برای خودکار کردن انواع استدلال‌ها، نیاز است که با به کارگیری قوانین یا روابط مجموعه‌ها و زیرمجموعه‌ها آشنا شد.

نمونه‌هایی از نمایش دانش به شکل رسمی و فرموله‌شده عبارتند از:

- semantic nets
- systems architecture
- frames
- rules
- ontologies

نمونه‌هایی از موتورهای استنتاج خودکار عبارتند از:

- inference engines
- theorem provers

ثبات قضیه خودکار در علوم رایانه به بررسی راه‌های ممکن برای اثبات قضیه‌ها به صورت خودکار (معمولاً با برنامه کامپیوتری) می‌پردازد. اثبات قضیه خودکار یکی از مهم‌ترین شاخه‌های استدلال خودکار به شمار می‌آید، اما همچنین به مقدار زیادی به علوم رایانه نظری و فلسفه مربوط است. منظور از قضیه در این جا قضیه ریاضی است.

نرم‌افزارهای اثبات قضیه خودکار قادرند برخی مسائل و قضایای ریاضی را اثبات نمایند و حتی در مواردی موفق به کشف اثبات‌های کوتاه‌تر برای برخی از قضایای ریاضی شده‌اند.

- classifiers

Classifierها یا طبقه‌بندی‌کننده‌ها برنامه‌هایی هستند که به طور خودکار به داده‌ها برچسب می‌زنند.

تمرین: به عنوان نمونه‌ای از Classifierها نرم‌افزارهای Google Lens و Bixby Vision را بررسی کنید. از یک گل عکس بگیرید و بررسی کنید که آیا به درستی نام آن گل را بیان می‌کند؟ (آیا برچسب درستی به آن گل می‌زند؟)

https://en.wikipedia.org/wiki/Learning_classifier_system

https://en.wikipedia.org/wiki/Rule-based_machine_learning