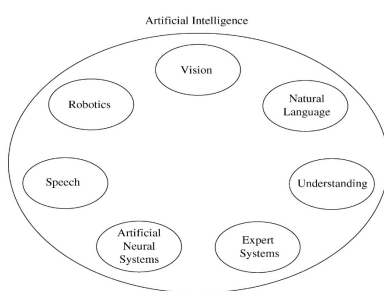


سیستم های خبره

دکتر سیامک سرمدی

تعريفات



□ **سیستم های خبره:** یکی از شاخه های هوش مصنوعی است که در آن **نرم افزار** هایی میسازیم که:

□ به نحوی از قابلیت تصمیم گیری **افراد خبره** تقلید کرده و با استفاده از دانش تخصصی به **حل مسائل** می پردازند.

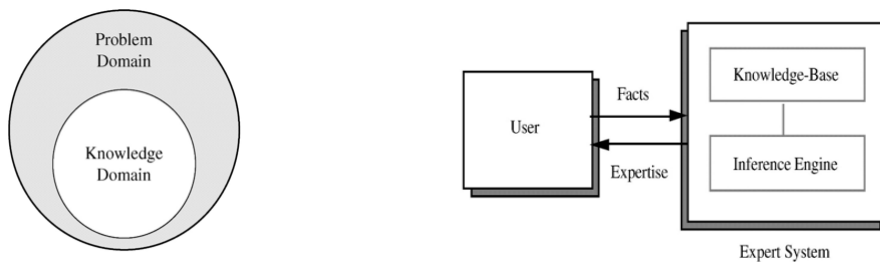
□ از **دانش** و **روشهای استنتاج** برای حل مسائلی استفاده می کند که به دلیل **مشکل** بودن، نیاز به تجربه و مهارت انسان دارند.

□ **فرد خبره:** کسی است که در **یک زمینه** خاص دارای **تجربه** و مهارت و در یک کلام خبرگی است (مثال: یک مکانیک، تعمیرکار تلویزیون، پزشک...).

□ **نحوه کار:** کاربر اطلاعاتی را (شامل حقایق یا وقایع ... Facts, Events, ...) به سیستم خبره داده و در پاسخ تجربه، تخصص و توصیه هایی دریافت می کند.

ساختار

- از نظر ساختار ، سیستم خبره از دو بخش اصلی تشکیل می شود:
- پایگاه دانش: حاوی دانش یک زمینه مشخص است که معمولاً به شکل **جملات و قواعد منطقی** ذخیره شده است.
- موتور استنتاج: با استفاده از اطلاعات و دانش موجود در پایگاه دانش **نتیجه گیری** نموده و پاسخ کاربر را می دهد.
- معمولاً **دامنه** دانش سیستم **محدود** به دانشی که به شکل **جملات منطقی** به آن داده شده میباشد و طبیعتاً کل دانش موجود یک حوزه را پوشش نمیدهد.



نمایش دانش و استنتاج

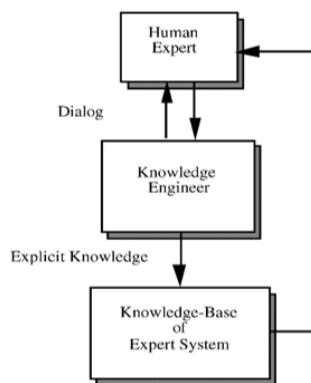
- **استنتاج:** سیستم به همان روشی که افراد خبره راه حل مسئله را میابند به استدلال یا استنتاج می پردازد. به این معنی که با داشتن **دانش و منطق**، و همچنین یک مجموعه از **واقعیتها**، به نتیجه گیری میپردازد.
- مثال: اگر اتومبیل روشن نشود، ممکن است نتیجه گرفته شود که بنزین آن تمام شده. البته این یکی از چندین استنتاجی است که ممکن است صورت بگیرد.
- **قواعد (rules):** دانش موجود در یک سیستم خبره را میتوان به شکل های **مختلفی نمایش داد (representation)** و ذخیره کرد. یکی از روشهای عمومی نمایش دانش، استفاده از **تعدادی از قواعد منطقی (Rules)** به شکل **اگر ... آنگاه ...** هستند. مثال:

اگر "چراغ قرمز" است آنگاه "متوقف شو".

اگر - الگو (pattern) - عمل (action)

- هنگامی که **واقعیت** "چراغ قرمز است" وجود داشته باشد، این **واقعیت با الگوی** "اگر چراغ قرمز است... " **منطبق** می شود. با توجه به اینکه **مقدم** قاعده **ارضاء** شده، عمل یا اقدام این قاعده یعنی "متوقف شو" انجام میگردد.

نحوه تهیه قواعد و دانش



- **مهندسی دانش: فرآیند ساخت** یک سیستم خبره (بخصوص ساختن قواعد) را گویند که توسط یک مهندس دانش صورت میگیرد.
- **مهندس دانش:** یک سیستم خبره کلاسیک حاوی دانشی است که در جایی نوشته نشده است. مهندس دانش به **کسب دانش** از **فرد خبره** و تبدیل آن به زبان قابل فهم برای ماشین (مثلا قواعد) اقدام میکند. دانش باید از طریق **گفتگوی طولانی** بین یک فرد خبره و یک مهندس دانش استخراج شود.
- تبدیل دانش معمولا فرآیند **آسانی نیست** و ممکن است **اشتباهاتی** در آن صورت گیرد. همچنین در دانش فرد خبره ممکن است **ناسازگاریها** (inconsistencies)، **ابهامات** (ambiguities)، **دوگانگی** (duplications) و یا مشکلات دیگری وجود داشته باشد و بعضا تا زمانی که این دانش در یک سیستم خبره مورد استفاده قرار نگرفته، مشکلات ظاهر نمیشوند.

محدودیت اساسی سیستم های خبره

- **عدم وجود دانش علت و معلولی:** یکی از محدودیتهای عملی سیستم های خبره کمبود دانش علت و معلولی است. یعنی سیستم خبره **درک عمیقی** از **علت و معلول** های اساسی **ندارد**.
- برنامه نویسی سیستمهای خبره با **دانش سطحی** (shallow knowledge) صورت می گیرد که بر اساس **دانش تجربی**، ابتکاری یا هیوریستیک (Heuristic knowledge) است. ساختن این سیستم ها بسیار **ساده تر** از ساختن سیستمی بر اساس **دانش عمیق** (Deep knowledge) است که بر اساس عملکرد، ساختار و رفتار بنیانی اشیاء استوار است.
- **مثال:** تهیه سیستم خبره ای که برای یک فرد مبتلا به سر درد **آسپیرین** تجویز کند ساده تر از سیستمی است که تمام اصول بیوشیمی، فیزیولوژی، کالبد شناسی (anatomy) و عصبی بدن انسان را بداند و تجویز دارو را بر اساس آن دانش انجام دهد. برنامه ریزی همه دانش علت و معلولی بدن کار بسیار دشواری است و حتی اگر موفقیت آمیز باشد می تواند بسیار کند باشد.

محدودیت اساسی سیستم های خبره

- **هیوریستیک:** یکی از انواع دانشهای **سطحی**، دانش **ابتکاری** یا هیوریستیک است. لغت هیوریستیک در زبان یونانی به معنی **کشف** است.
- روشهای هیوریستیک برخلاف روشهای الگوریتمی حل موفقیت آمیز مسئله را **تضمین نمی کنند**. بلکه این روشها یکسری قواعد **تجربی** و یا قوانین **سر انگشتی** هستند که **از** آزمایشها و **تجربیات** بدست آمده اند و ممکن است در حل مسئله کمک کنند ولی لزوما نمی توانند آنها بدرستی حل کنند.
- مثال: اگر موقع فشاردادن ترمز صدای صوت می آید، لنت های ترمز ایراد دارند و باید عوض شوند.
- در بعضی زمینه های پزشکی و مهندسی، روشهای هیوریستیک نقش مهمی دارند. حتی اگر یک **راه حل دقیق** برای مسئله ای وجود داشته باشد، ممکن است از نظر **زمانی** یا **هزینه غیر عملی** باشد و راه حل های هیوریستیک **میانبر** هائی را ارائه کنند که در زمان و هزینه صرفه جویی نماید.

امکانات دیگر سیستم های خبره

- **تسهیلات توضیح دهنده (Explanation Facility):** معمولا لازم است که سیستم خبره بتواند **نحوه استدلال** و **استنتاج** خود برای رسیدن به جواب را **توضیح دهد** تا بتوان استدلال آن را ارزیابی کرد. معمولا برای اینکار، سیستم قواعد مورد استفاده و نحوه نتیجه گیری منطقی پاسخ **از** آن **قواعد** را نشان میدهد.
- **القاء قاعده (Rule Induction):** بعضی سیستم های خبره این قابلیت را دارند که **قواعد** را **با** استفاده از **مثالها** یاد بگیرند. در این روش سیستم، قواعد را با استفاده از جداول اطلاعات ایجاد می نماید. معمولا اطلاعات بصورت جدولی (مثلا فایل Excel به برنامه ای داده میشود و برنامه بر اساس آن مثالها قواعدی را میسازد)
- اگرچه القاء قواعد (یادگیری با استفاده از ارائه مثالها) به توسعه دانش کمک می کند ولی **فقط انواع محدودی** از دانش را میتواند از این طریق وارد سیستم خبره کرد.

حل مسائل با قواعد منطقی

□ در انتهای دهه 1950 برنامه های زیادی برای "حل مسائل بطور عمومی" نوشته شدند. مشهورترین نمونه "حلال عمومی مسائل" (General Problem Solver) بود که توسط نیول (Newell) و سیمون نوشته شده بود. این دو نتیجه گرفتند که:

حل بیشتر مسائل توسط انسان، ادراک یا فهم را می توان با قواعد تولیدی (Production rule) از نوع "اگر... آنگاه..." بیان کرد. به عنوان مثال: اگر هوا بارانی به نظر می رسد، آنگاه چتر بردار.

□ هر قاعده متناظر با معادل با مجموعه ای کوچک از دانش است که به آن یک جزء یا چانک (Chunk) گفته میشود. مثال:

اگر ماشین حرکت نمیکند و
درجه سوخت نشان میدهد که باک خالی است
آنگاه مخزن را پر کنید.

حافظه دراز مدت - محل ذخیره قواعد

□ حافظه درازمدت جایی است که دانش ما در آن ذخیره می شود. حافظه دراز مدت تعداد بسیار زیادی از اینگونه قواعد "اگر ... آنگاه ..." در خود ذخیره کرده است. برای مثال یک متخصص شطرنج ممکن است حدود 50000 جزء دانش درباره الگوهای بازی شطرنج بداند.

□ نمونه ای از قواعدی که ما در مغز خود ذخیره کرده ایم:

اگر شعله ای وجود داشته باشد، پس آتش وجود دارد.
اگر دود وجود داشته باشد پس ممکن است آتش وجود داشته باشد.
اگر صدای آژیر شنیده شود پس ممکن است آتش وجود داشته باشد.

□ توجه کنید که دو قاعده آخر با قاطعیت بیان نشده اند و به صورت احتمالی می باشند.

حافظه کوتاه مدت -

□ از حافظه کوتاه مدت برای ذخیره موقت دانش در هنگام حل مسئله (استنتاج) استفاده می شود. این حافظه عملیاتی (Working memory) کوچک بوده و در حدود 4 الی 7 جزء ظرفیت دارد. برای مثال بیشتر مردم فقط می توانند 4 الی 7 عدد را در یک لحظه در ذهن خود تصور کنند. حافظه درازمدت ظرفیت بسیار بالاتری از حافظه کوتاه مدت دارد.

□ یک تئوری بیان می کند که حافظه کوتاه مدت نشانگر تعداد اجزاء دانشی است که بطور همزمان میتوانند فعال شوند. در واقع حل مسائل در انسان با در نظر گیری این اجزاء دانش فعال شده در ذهن صورت می گیرد.

□ اگر چندین قاعده بطور همزمان فعال شوند، پردازش گر باید به روشی تصمیم بگیرد که کدام قاعده بالاترین ارجحیت را دارد. سپس قاعده با اولویت بالاتر اجرا خواهد شد. برای مثال اگر هر دو قاعده زیر بر اثر اطلاعات ورودی فعال شوند، در آن صورت باید به علت اولویت یکی از قواعد فوق زودتر اجرا شود.

اگر آتش سوزی وجود دارد، پس محل را ترک کنید.
اگر لباسهایتان آتش گرفته اند پس آنها را خاموش کنید.

کاربردهای سیستم های خبره (بر پایه قواعد)

1. تشخیص و رفع اشکال دستگاهها و سیستم ها: تشخیص و پیشنهاد راه حل برای ایرادات دستگاهها و سیستم های مختلف
2. برنامه ریزی و زمانبندی: تشخیص اعمال لازم برای رسیدن به یک هدف خاص و ترتیب آن اعمال با در نظر گرفتن نیروی انسانی، مواد و سایر محدودیت ها، این گروه از ابزارها دارای پتانسیل تجاری بالایی هستند. مثال: زمانبندی برای خطوط هوایی (پرواز ها، پرسنل، گیت ها، انتقال بار...)، زمانبندی و برنامه ریزی تولید، زمانبندی امور در شرکت ها
3. آرایش (انتخاب اجزاء بر اساس سفارش) و اعمال تنظیمات بر روی دستگاههای تولید شده: برای مثال انتخاب اجزاء کامپیوترها و انجام تنظیمات سخت افزاری و نرم افزاری آنها، تولید خانه های پیش ساخته و سایر تولیداتی که پروسه مهندسی و تولید پیچیده ای دارند.
4. تصمیم گیری های مالی: در مورد ارائه وام به شرکتها و اشخاص، تشخیص ریسک و تعیین قیمت در بخش بیمه، خرید فروش سهام، خرید و فروش ارز
5. نشر اطلاعات و علم: ارائه انتخابی مطالب و ارائه راهنمایی به کاربران در زمینه مشکلات خاص. برای مثال: راهنمایی گرامری، مشاور مالیاتی برای ارائه راهنمایی در مورد نحوه تنظیم و ارائه گزارشات مالیاتی، راهنمای حقوق برای ارائه مشاوره های حقوقی
6. مونیتورینگ و کنترل پروسه ها: آنالیز فوری اطلاعات و تشخیص ایرادات احتمالی در یک پروسه، مثلا در صنعت تولید فولاد و نفت و ...
7. طراحی و تولید: ارائه کمک در طراحی فیزیکی و مفهومی محصولات، تا خود پروسه تولید

چه موقع از سیستم خبره استفاده کنیم

- قبل از شروع به ساخت سیستم خبره باید مطمئن شد که سیستم خبره روش مناسبی برای حل یک مسئله هست یا نه.
- آیا میتوان مساله را با استفاده از روش های معمول برنامه نویسی حل کرد؟ اگر پاسخ مثبت است در آن صورت استفاده از سیستم خبره بهترین روش حل مساله نیست.
- اگر تمام حالات و علائم حاکی از بد کارکردن یک سیستم یا تجهیزات و راه حل های آنها شناخته شده باشد ، در اینصورت یک جدول جستجوی ساده و یا یک درخت تصمیم گیری برای حل مسئله کافی است.
- سیستم های خبره برای حل مسائلی که برای آنها الگوریتم مشخص و شناخته شده ای وجود ندارد و دارای ساختار ناقصی هستند (وجود احتمالات زیاد) مناسب است. در اینگونه موارد راه رسیدن به یک راه حل خوب ، استدلال است.

چه موقع از سیستم خبره استفاده کنیم

- برای مثال اگر علائم کافی برای تشخیص یک بیماری موجود نباشد... معمولاً به همین علت یا علل مشابه (کمبود اطلاعات) است که برای تشخیص های پزشکی امکان ایجاد الگوریتم ها و برنامه های متداول نیست.
- یک فرد مسافر تصمیم به مسافرت گرفته ولی اطلاعات کافی برای انتخاب مشخصات مسافرت خود به آژانس مسافرتی ارائه نکرده است. مثلاً مقصد سفر مشخص نیست، بودجه کافی در اختیار مسافر نیست و زمان دقیقی نیز مد نظر نیست و مسافر تنها قصد سفر در اسرع وقت را دارد. در این مثال خاص ساختار اطلاعاتی ناقص است و به علت وجود احتمالات و انتخاب های زیاد روشی الگوریتمی برای مسئله وجود ندارد.
- برخلاف الگوریتمها و برنامه های متداول که انتظار دریافت ورودی های مشخص با ترتیب مشخصی را دارند، سیستم های خبره باید امکان واکنش به هر نوع ورودی (مثلاً ورودی های ناقص) را داشته باشند.

ابزارهای مورد استفاده

□ یکی از تصمیم های اساسی **انتخاب ابزارهای** ایجاد سیستم خبره است. معمولاً **بهتر است** از ابزارهای موجود برای این منظور استفاده کرده و بدنبال برنامه نویسی نرویم. زبانهایی مانند LISP و Prolog برای انجام **عملیات سمبولیک** (حقایق و قواعد با استفاده از عبارات اولیه یا سمبول ها بیان میشوند) مناسب هستند و بنابراین برای ساخت سیستم های خبره مناسب هستند ولی اهداف و کاربرد آنها **خیلی وسیع** تر است. بجای ساختن تمام اجزاء سیستم خبره از ابتدا با برنامه نویسی بهتر است از ابزارهای تخصصی تر مانند CLIPS استفاده شود. اینگونه سیستمها **محدودیت** هایی دارند ولی ساده تر بوده و برای ساختن این سیستمها مناسب ترند..

□ سیستم های خبره را بر اساس **مشخصه** های مختلف می توان **توصیف و مقایسه** کرد. برای مثال:

□ نحوه نمایش دانش (قواعد، ...)

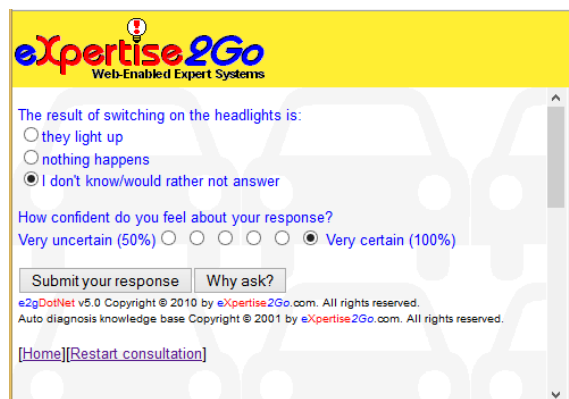
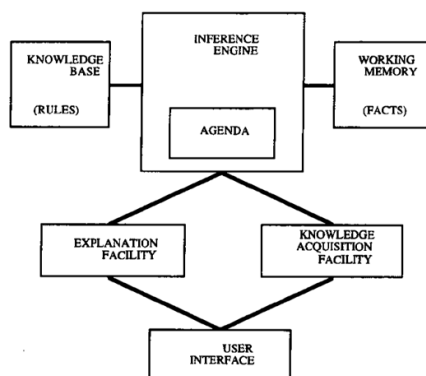
□ قابلیت استنتاج پیشرو یا پسرو

□ پذیرش عدم قطعیت

□ امکانات توضیح راه حل

اجزاء سیستم

□ **رابط کاربری (User Interface):** ارتباط کاربر و سیستم خبره را برقرار می کند. این بخش می تواند به شکل متنی (مثلاً بر مبنای وب)، گرافیکی یا حتی مکانیکی باشد و همچنین ممکن است ساده و یا پیچیده و با امکانات متعدد طراحی گردد.



اجزاء سیستم

- امکانات توضیح راه حل: نحوه استدلال سیستم را برای کاربر توضیح می دهد. در سیستمهای مبتنی بر قاعده پاسخ به این سوال که چگونه سیستم به یک نتیجه خاص رسیده آسان تر است، زیرا سابقه قواعد فعال شده و محتویات حافظه کاری را میتوان نگهداشت.
- پایگاه دانش (حافظه تولیدی یا Production Memory): شامل کلیه دانش موجود در سیستم است که به شکل قواعد ذخیره گردیده اند. قواعد تولید را می توان به شکل "اگر ... آنگاه..." بیان کرد. هر قاعده با یک نام شناخته میشود. پس از نام، اگر و سپس "مقدم، قسمت شرطی، قسمت الگو، یا قسمت سمت چپ (LHS)" می آید.
- برای مثال یک سیستم ساده تصمیم گیری عبور از خیابان را در نظر بگیرید که دو قاعده زیر در آن وجود دارند:

The light is Red → Stop

The light is Green → Go

```
Rule: Red_light
IF
    the light is red
THEN
    stop

Rule: Green_light
IF
    the light is green
THEN
    go
```

اجزای سیستم - قواعد

- نمونه زیر از قواعد سیستم خبره MYCIN است که برای تشخیص بیماریهای عفونی خون و مننژیت ساخته شده است:

```
IF
    The site of the culture is blood, and
    The identity of the organism is not known with
        certainty, and
    The stain of the organism is gramneg, and
    The morphology of the organism is rod, and
    The patient has been seriously burned
THEN
    There is weakly suggestive evidence (.4) that
        the identity of the organism is pseudomonas
```

اگر محل کشت میکروب در آزمایشگاه خونی است و هویت ارگانیسم با قطعیت شناخته نشده است و رنگ ارگانیسم به صورت گرام منفی است و شکل ارگانیسم، میله ای است و بیمار دچار سوختگی شدید است.

آنگاه

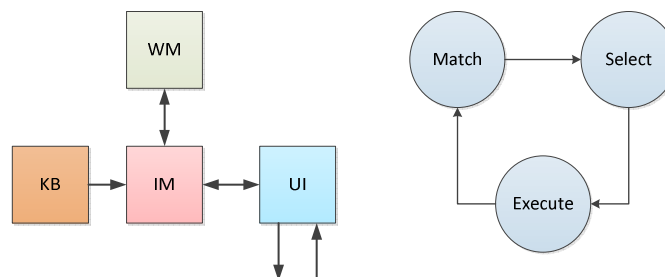
شواهد ضعیفی (۰/۴) نشان می دهند که هویت ارگانیسم، شبه تک یاخته است.

اجزای سیستم - قواعد

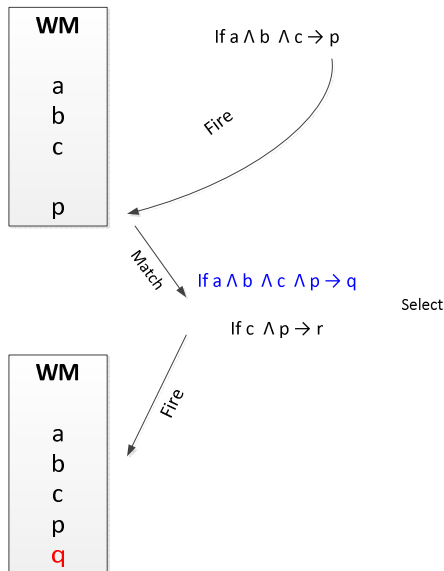
- **حافظه کاری:** وقایع (events) و حقایق (facts) که توسط قواعد بکار گرفته می شوند در آن ذخیره می گردد (مانند حافظه کوتاه مدت انسان عمل می کند).
- **موتور استنتاج:** تعیین می کند که کدام قواعد توسط واقعیات و یا اطلاعات وارد شده ، فعال و ارضاء می شوند (از طریق ارضاء قسمت شرط آنها).
- **دستور کار یا برنامه عملیات (agenda):** یک لیست اولویت بندی شده از قواعد که توسط موتور استنتاج تهیه می شود. قسمت شرطی قواعد موجود در این لیست توسط حقایق و یا اشیاء موجود در حافظه کاری ارضاء شده است.

سیکل اجرا

- چرخه اجرای سیستم خبره به شکل زیر است (چرخه تا زمانی که بر اثر اجرا، قاعده جدیدی فعال نگردد ادامه میابد):
- **تطبیق:** ابتدا قواعد موجود در پایگاه دانش با حقایق ارائه شده مقایسه شده و تعدادی از قواعد که قسمت شرط آنها ارضاء می شود برای آتش شدن فعال می گردند.
- **انتخاب (حل اختلاف یا Conflict resolution):** اگر بیش از یک قاعده در مرحله قبل فعال شده باشند (و در دستور کار باشند)، یکی از قواعد در این مرحله انتخاب می شود.
- **اجرا:** قاعده انتخاب شده آتش (اجرا) می شود. در اثر اجرای قاعده ممکن است حقایق جدیدی (از تالی قاعده اجرا شده) بوجود آیند و در حافظه کاری قرار گیرند. در اینصورت اگر در اثر تغییر حافظه کاری (حقایق) قسمت شرطی تعدادی از قواعد فعال شده دیگر ارضاء نمیشود، آن قواعد از دستور کار حذف می گردند. بعضی از قواعد ممکن است ایجاد خروجی، دریافت ورودی یا توقف را فعال کنند.



عملیات منطقی استنتاج



- **استنتاج پیشرو (Forward Chaining):** این نوع استنتاج، از حقایق شروع به استدلال میکند تا به نتایجی برسد که از این واقعیات (بر اساس قواعد) ناشی میشوند.
- **مثال 1:** اگر قبل از خروج از منزل باران بیاید (واقعیت)، یک چتر با خود بر میدارید (نتیجه).
- **مثال 2:** در تصویر مقابل قاعده فوق با حقایق موجود در حافظه کاری (a, b, c) فعال میشود. قاعده آتش شده و یک حقیقت جدید را تولید کرده و در حافظه کاری قرار می دهد. سپس مجموعه حقایق موجود در این مرحله (a,b,c,p) دو قاعده دیگر را انتخاب و فعال میکند. از بین ایندو قاعده، قاعده اول به علت اولویت آتش میشود و حقیقت q را تواید میکند. قاعده بعدی ممکن است در مرحله بعد آتش شود (در صورتیکه هنوز فعال باشد).

عملیات منطقی استنتاج

- بنابراین در طول سیکل اجرا تعداد قواعد فعال شده دائم در حال تغییر است.
- ممکن است قاعده ای علی رغم قرار گرفتن در حافظه کاری (دستور کار) هیچگاه آتش نشود (در مرحله انتخاب ، به علت اهمیت بیشتر، اولویت بالاتر یا دلیل دیگر، قاعده دیگری برای آتش شدن انتخاب گردد و آن قاعده غیر فعال شود)
- **استنتاج پسرو (Backward Chaining):** با روشی معکوس استدلال میکند. یعنی سعی میکند از یک فرضیه یا در واقع نتیجه بالقوه به واقعیات و یا حقایقی که پشتیبان آن نظریه هستند (میتوانند منتج به آن نتیجه شوند) برسد.
- **مثال:** اگر شما هوای بیرون خانه را ندیده باشید و شخصی با چتر خیس وارد خانه شود، فرضیه شما این خواهد بود که باران آمده است.
- برای اثبات فرضیه فوق از شخص سوال میکنیم که آیا باران آمده است؟ اگر پاسخ شخص مثبت باشد فرضیه به یک واقعیت یا حقیقت بدل می شود. بنابراین یک فرضیه را میتوان بعنوان یک واقعیتی در نظر گرفت که در صحت آن شک هست و باید ثابت شود.

عملیات منطقی استنتاج

- بسته به طراحی ، موتور استنتاج ممکن است بتواند با روش پسرو و یا پیشرو استنتاج کند.
- OPS5 و CLIPS برای استنتاج پیشرو طراحی شده اند. EMYCIN برای استنتاج پسرو ساخته شده است. بعضی موتورهای استنتاج مانند ART و KEE با هر دو روش کار میکنند.
- در مسائل تشخیصی بهتر است از روش **پسرو** استفاده کنیم و در مسائل **پیش بینی** ، نظارت و کنترل روش **پیشرو** بهتر است.