فهرست

پیشگفتار

کتاب حاضر برگرفته از تجربیات عملی و تدریس نگارنده در خصوص هوش مصنوعی است . بخصوص که این تجربه با تدریس و طراحی سوال در دانشگاه پیام نور نیز تکمیل گردیده است . اغلب دانشجویان پیام نور در اقصی نقاط ایران به دلایلی نظیر اشتغال به کار در ایام هفته یا کمبود استاد در دروس تخصصی کامپیوتر, از نعمت استاد و تدریس تمام وقت محرومند .از طرفی فرآیند طراحی سوال به گونه ای است که همه موضوعات کتاب را شامل شده و تفاوتی بین دانشجویانی که از کلاس استاد بهره برده اند با دانشجویانی که به صورت خودخوان درس را امتحان میدهند ندارد متاسفانه ترجمه کتابهای معروف از دانشگاهها معروف به چند دلیل منبع خوبی برای مطالعه دانشجویان نیست. یکی از مهمترین دلایل حجم کتاب است زیرا در دانشگاههای دیگر استاد فصولی از کتاب را برای تدریس و آزمون انتخاب میکند که آنها را درس داده و برای ارایه درس نیز به ازای هر واحد ۱۸ ساعت در نظر گرفته میشود با این اوصاف بخش عمده ای از کتاب تدریس نمیگردد حال اینکه در پیام نور برای رشته کامپیوتر به ازای هر واحد درسی در بهترین شرایط ۱۱ ساعت برای مرور درس و رفع اشکال در نظر گرفته شده و امکان حذف مطالب توسط استاد نیز غیر ممکن است زیرا سوالات بصورت متمرکز طراحی و توزیع میگردد .

ضرورت استفاده از منبعی که به صورت خودآموز بوده و حاوی مطالب اصلی باشد و با فرهنگ دانشگاه پیام نور تطبیق داشته باشد کار بسیار دشواریست . نگارنده این تلاش را به عنوان گامهای نخست برای اغنای منابع درسی خودخوان در رشته کامپیوتر تلقی مینماید و به همین دلیل اثر را خالی از اشکالات شکلی و محتوایی نمیداند و در این مسیر از همه مطالعه کنندگان کتاب بخصوص دانشجویان تقاضای انتقال بازخورد خودخوان و تدریس این درس را دارد و با این امید به خود اجازه داده است با گردآوری مطالب کلیدی و البته کاربردی و به روز دانشجویان را با مفاهیم مرتبط با این حوزه آشنا سازد .با این اعتقاد که بهترین افراد برای اشکالزدایی کتابهای منبع در پیام نور همانا دانشجویان با تجربه هستند لذا سعی شده دانشجویان

زیادی بخصوص افرادی که در آستانه فارغ التحصیلی هستند کتاب حاضر را مطالعه و نقد کنند.در ابتدای هر فصل راهنمایی برای طراح سوال و دانشجو که اشاره به اهمیت فصل دارد ارایه خواهد شد تا با یاری خدا هم استاد و هم دانشجو و هم طراح سوال با الگو و شاخصی مشابه از مفاهیم کتاب بهره برداری نمایند. در این راستا راهنمایی اساتید در وزن دهی و ارزش گذاری فصول و انتقال به نگارنده بسیار حایز اهمیت است .همچنین سعی شده در پایان هر فصل موضوعاتی برای تحقیق مطرح شود تا عمق دانسته های دانشجو در قالب ٦ نمره تحقیق درس افزون گردد.

تجربه نگارنده تاکید دارد که برای درک مفاهیم نظری باید امکان آزمون پذیری مفاهیم کتاب فراهم گردد بخصوص در شرایطی که استادی برای رفع اشکال در دسترس نیست . لذا علاقه مند بودم در شروع کتاب مفاهیم برنامه نویسی در سیستمهای هوشمند با زبانهایی مانند لیسپ و پرولوگ را که با قواعد پیاده سازی نیازها با زبانهای رایج مانند c++ و جاوا و دلفی کاملا متفاوت است را به عنوان مطلب اصلی ارایه نمایم اما با تبادل نظر با اساتید با تجربه دانشکده فناوری اطلاعات تصمیم گرفته شد مطلب به صورت چکیده و در قالب ضمیمه به کتاب اضافه گردد لذا دو ضمیمه این کتاب مروری کوتاه بر برنامه نویسی در محیطهای پرولوگ و لیسپ خواهد داشت.

.

مقدمه

انسان از دیرباز سوالات زیادی در خصوص روش درک خود و چگونگی شکل گیری عقل و استنتاج در ذهن داشته است . پاسخ به این سوالات در گذشته بسیار دور برای فلاسفه و بعد برای ریاضی دانانِ علاقه مند به توسعه منطق ریاضی و سپس برای روانشناسان بسیار مفید بود چنانکهَ ریشه بسیاری از قوانین استقراء ایجاد منطق ریاضی و تحلیل روان شناختی از انسان از ترکیب این پاسخها حاصل شده است . پس از اختراع رایانه بسیاری از دانشمندان به فکر تولید موجودات هوشمند افتادند هر چند که در ابتدا سرعت پردازش و حافظه های در دسترس بسیار کم بود اما دانشمندان علوم رایانه با ادغام یافته های علوم مختلف بخصوص ریاضیات راه را برای خلق ابزاها و موجودات هوشمند باز نمودند. هوش مصنوعی سعی در ارایه روشهای حل مسئله با رایانه دارد بگونه ای که بیننده در تشخیص عامل بودن انسان یا ماشین دچار تردید گردد.شاید این آغاز راهی بود که امروز در جای جای زندگی روزمره ما گسترش یافته و تمام ابزارهای زندگی عادی ما از ماشین لباس شوییهایی که زمان کار خود را با میزان کثیفی لباس ما تطبیق میدهند تا فروشگاههایی که به محض ورود ما میدانند به دنبال چه هستیم از این را به یکدیگر مرتبط هستند.

این کتاب از ۱۲ فصل و ۲ ضمیمه تشکیل شده است .در ابتدای هر فصل مقدمه ای اختصاصی آورده شده است که خواننده دلایل مطالعه فصل را بهتر درک کند و اگر این کتاب مرجع درسی خارج از مقطع کارشناسی پیام نور بود با توجه به این مقدمه بتوان فصولی از کتاب را حذف نمود.فصل اول شامل تاریخچه هوش مصنوعی , کاربردهای هوش مصنوعی , اهدافی که هوش مصنوعی دنبال میکند و آینده آن است .این فصا برای ترغیب دانشجو جهت دنبال کردن موضوعات اساسی کتاب بوده و اهمیتی کمتر از یک دوازدهم کتاب دارد.فصل دوم به مفاهیم منطق و استنتاج میپردازد و اشاره ای کوتاه به سیستمهای هوشمند قاعده گرا و زبان پرولوگ دارد. فصل سوم به روشهای فراگیری و نمایش دانش اختصاص دارد . فصل چهارم در خصوص روشهای استدلال مبتنی بر دانش و نظریه ها و زبانهای نمایش دانش است.فصل پنجم به عدم قطعیت اختصاص دارد و بحثی را در خصوص روشهای لحاظ کردن احتمالات را در استنتاجها مطرح مینماید.در این فصل نظریه های معروف از جمله دمستر و شافر در خصوص مشاهدات بیان شده و اشاره مختصری به منطق فازی که ابتکار دکتر عسکری زاده بوده و دنیای

صنعت را دگرگون نموده خواهد شد. در فصل شش دانشجویان با روشهای جستجو که بیشترین کاربرد را در استنتاجهای هوشمند آشنا خواهند شد. دیدن سخن گفتن و شنیدن ویژگیهای خاص موجودات هوشمند است .فناوری های هوش مصنوعی شامل بینایی ماشین و پردازش زبانهای طبیعی و تولید صدا در فصل هفتم مورد بحث و بررسی قرار خواهد گرفت دانشجویان با مروز این فصل درک واقعی از کاربردهای هوش مصنوعی را در زندگی امروز بدست خواهند آورد . فصل هشتم شامل مروری بر سیستمهای خبره میباشد و با مطالعه آن تصور مناسبی از نقش فرد خبره , مهندس دانش و روشهای استنتاج به منظور تولید یک سیستم خبره به دانشجو منتقل خواهد شد.در این فصل رهیافتهای برنامه نویسی و ابزارهای مورد استفاده و کاربردها نیز معرفی خواهند شد. فصل نهم مروری بر شبکه عصبی با تکیه بر شناخت سیستم مشابه انسانی داشته و ضمن اشاره به چگونگی فرآیند یادگیری ماشین و محدودیتهای آن تفاوت های آنرا با سیستم انسانی مقایسه خواهد کرد.فصل دهم اشاره ای به روش استدلال موردی و چند کاربرد پیاده شده با این روش در زندگی امروز خواهد داشت تا دانشجو با نمونه های دیگری از استنتاج و فهم ماشین آشنا شود.فصل یازدهم برنامه ریزی محدود و کاربردهای آن بخصوص آلگوریتم تنظیم رشته DNA و پروتوئینها را توضیح میدهد و نتیجه کاربردی کتاب در فصل دورازدهم به دانشجویان ارایه خواهد شد.این کاربردها شامل تشریح نمونه های موفق در صنعت , پزشکی و توریسم الکترونیکی و تجارت الکترونیکی میباشد .در پایان نیز دو ضمیمه که شرح مختصری از دو زبانی که مختص سیستمهای هوشمند آفریده شده اند یعنی پرولوگ و لیسپ اشاره خواهد شد .به احتمال زیاد شما این کتاب را به همراه یک لوح فشرده شامل مطالب کمک آموزشی و دو محیط پیاده سازی برای زبانهای لیسپ و پرولوگ دریافت خواهید کرد .اما اگر این مهم به دلایل مدیریتی و اجرایی تحقق نیافت سایت **www.askarzadeh.ir** به منظور ارایه مطالب فوق و مباحث مربوط به مولف در نظر گرفته خواهد شد.

.

فصل اول

هوش مصنوعی در یک نگاه

اهداف کلی

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

۱. هوش مصنوعی شامل چه مفاهیم و تعاریفی است

۲. هوش مصنوعی چگونه رشد نمود و تاریخچه آن چیست.

۳. کاربرد های هوش مصنوعی در زندگی ما کدامند

٤. آینده هوش مصنوعی تحقق کدامیک از اهداف بشر را نشانه گرفته است

نگاه کلی به هوش مصنوعی

مقدمه

کامپیوترهای اولیه با محاسبات عددی در ارتباط بودند. کامپیوترهای کنونی علاوه بر محاسبات عددی ، با استدلال بر مبنای دانش هم، درگیر هستند . با فناوری های هوش مصنوعی که از این به بعد به اختصار AI می نامیم نقش کامپیوترها از وسیله ای مفید به وسیله ای ضروری و حیاتی تغییر می کند.

هدف AI این است که سعی کند کامپیوتر ها را به انجام کارهایی که بشر متمایل است درآن ماهر باشد ، وادار کند.

از یک جهت ، این یک تحقیق برای سعی در وادار کردن کامپیوترها به رفتار کردن به روشی زیرکانه تر می باشد.

نام واقعی AI به وسیله جان مک کارتی [1] در دهه ۶۰ ابداع شده است ، او طراح زبان LISP بود.

AI فاصله میان دانشمندان رفتار بشری و دانشمندان کامپیوتر را پر می کند.

چیزی که ما در مورد کامپیوتر ها می دانیم این است که مجموعه ای از دستورات نوشته شده به زبان برنامه نویسی کامپیوتر ، که همیشه به طور دقیق اجرا خواهند شد به آنها داده شده است . بنابراین دانشمندان علوم بشری می توانند تئوری های خود را در مورد رفتار بشر با تبدیل قوانین آنها به برنامه های کامپیوتر تست کنند و ببینند که آیا رفتار کامپیوتر دراجرای این برنامه ها شبیه رفتار طبیعی یک موجـــود بشـری یا حداقل زیر مجموعه کوچکی از رفتار **بشر** ، که مطالعه می کنند ، می باشد .

دانشمندان کامپیوتر می توانند به مدل کردن رفتار بشری به عنوان یک مقایسه با توانائیهای برنامه نویسی خودشان نگاه کنند :

اگر یک شخص بتواند کاری انجام دهد ، آیا ما می توانیم یک برنامه کامپیوتری ،که همان کار را انجام دهد ، بنویسیم ؟

---

1 John McCarthy

نام « AI » نسبتاً یک نام برخواسته از احساسات است ، هر چند ا کنون تصویب شده و بعید است تغییر کند. بحث کردن در مورد اینکه آیا موجودات بشری فقط « ماشین های گوشتی » خیلی پیچیده ای هستند یا اینکه از نظر تئوری ، امکان دارد یک « AI » واقعی ایجاد کرد در حالی که یک برنامه کامپیوتری است که واقعاً مانند یک موجود انسانی کامل رفتار کند ، بسیار جالب باشد . اما مطالعه AI کسی را درگیر این گونه باورها نمی کند .

درحقیقت تنها واقعیت تصدیق شده در مورد AI ، این است که موجودات بشری نسبت به چیزی که آنالیز های رایانه ای در نظر میگیرند پیچیده تر هستند. پیش گویی هایی نظیر اینکه AI مشابه انسان به زودی ساخته می شوند ، اغلب فقط پیروی از احساسات و عواطف رسانه ها است وتا این لحظه درستی آن ثابت نشده است.

چنانکه میدانیم برنامه ی فضایی که منجر به نشستن سفینه روی کره ماه شد ،هرگز به هدف خود یعنی اقامت انسان در ماه منجر نشد بلکه به پیشرفت تکنولوژی فضایی منجر شد.

به نظر می رسد در مورد AI نیز چنین باشد یعنی منافع واقعی در تحقیق دراین زمینه به سمتی حرکت می کند که در کل باعث پیشرفت علم کامپیوتر می شود.

درمـــوارد خاص ، تحقیقات AI اغلب با مشکلات برنامه نویسی سخت خاصی مواجه شده است ، بسیاری از مفاهیم در طراحی زبان برنامه نویسی و متدولوژی برنامه نویسی که اکنـــون در علوم کامپیوتر عمومی هستند ، از برنامه نویسی AI نشات گرفته اند.

۱-۱ تعریف

گفته می شود که بسیاری از فعالیت های فکری بشر مثل نوشتن برنامه ها ، فهمیدن زبان ، درگیرشدن در برداشت استدلال یا حتی راندن یک اتومبیل ، به هوش احتیاج دارد. متجاوز از چند دهه گذشته کامپیوتر ها برای انجام دادن چنین وظایفی ساخته شده بودند .به طور خاص سیستم هایی وجود دارند که می توانند به طور خودکار کد کامپیوتری تولید کنند ، تفکیک و تجمع سمبولیک را انجام دهند ، متنی را به اندازه

مشخص درک کنند و غیره . درنمونه های ذکر شده گفته میشود سیستم ها درجاتی از هوش مصنوعی را دارا هستند.

با این مقدمه ، می توانیم AI را از زوایای مختلف تعریف کنم .مثلا :

- AI مطالعه نحوه وادار کردن کا مپیوتر ها به انجام چیزهایی است که مردم در آن زمینه ها هم اکنون بهتر از کامپیوترها هستند .
- AI شاخه ای از علم کامپیوتر است که با سمبل گذاری ، و روشهای غیر الگوریتمی حل مسائل سروکار دارد.
- AI بخشی از علم کامپیوتر است که به طراحی سیستم های کامپیوتری هوشمند مربوط است ، این سیستم ها خصوصیاتی را ارائه می دهند که فقط در رفتار موجودات هوشمند مانند انسان قابل مشاهده است .

AI مجموعه ای از تفکرات یا مفاهیم فکری است راهـــی برای نگـــاه کردن و حل مشکلات از یک نقطه نظر خاص .

آنچه AI را از سایر علوم کامپیوتر و مهندسی متمایز مینماید روش اکتشافی حل مسئله است که ما در این کتاب با مفاهیم ، روشها و تکنیکهای آن آشنا میشویم

مشکلات AI طیف بسیار وسیعی دارد. این مشکلات در حالت های جزئی و خاص پدیدار می شوند.تکنیک های AI برای دستکاری سمبل ها همانند اطلاعات عددی طراحی شده اند که حوزه وسیعی از آنها به وسیله محققان AI گسترش پیدا کرده است. این کتاب محدوده ای از این تکنیک ها را معرفی می کند.

۱-۱-۲ چرا AI ؟

کامپیوتر ها انسانها را از انجام دادن بسیاری وظایف بی نیاز کرده اند ، نمونه ای از این وظایف به صورت زیر است :

۱ـ محاسبات عددی : کامپیوتر ها در محاسبات عددی به طور قطعی سریع تر و دقیق تر از انسانها هستند. برای مثال یک کامپیوتر می تواند ۱۸۹۲۵٤ را در ۸۲۹۰۹۹۷٤۳ ضرب کند و تقریباً نتیجه را فوراً بدهد درحالی که برای بشر به طور متوسط دو دقیقه

طول می کشد که همین عملیات را انجام دهد. همچنین در مورد انسان ، امکان بروز خطا بیشتر است.

۲ـ ذخیره سازی اطلاعات : کامپیوتر ها مقدار زیادی از اطلاعات را می توانند ذخیره کنند. حجم محدود شده فقط به خاطر در دسترس بودن اطلاعات است این یک تباین با بشر است ، در جائی که به نظر می رسد که فقط مقدار معینی از آگاهی می تواند ذخیره شود کامپیوترها کاراتر هستند چون بر خلاف انسان به سرعت این آگاهی ها را میتوان تکثیر یا در چند نقطه نگهداری کرد.

۳ـ عملیات تکراری : روشن است که بشر موقع انجام عملیات تکراری بی حوصله و مرتکب خطا می شود .

ازطرف دیگرکامپیوتر به طور خاص برای انجام دادن یک چنین کارهایی ساخته شده است. به این دلیل که کامپیوترها در هنگام انجام عملیات تکراری نه شکایت می کنند و نه خسته می شوند.

لیست بالا به کارهای مکانیکی یا بدون نیاز به تفکر را مشخص می کنند . انسانها درانجام دادن کارهای هوشمند بهتر از کامپیوتر ها هستند . قبل از جلو رفتن در بحث بهتر است سعی کنیم معنی هوشمندی را درک کنیم .

هوشمندی یک عبارت تعریف شده نسبی است و هیچ تعریف دقیق مجردی دراین زمینه وجود ندارد.

رفتاری که برای یک فرد ، هوشمندانه به نظر می رسد ، ممکن است برای فرد دیگری اینگونه نباشد اما خصوصیات زیر قابلیت های ضروری برای هوشمندی هستند.

- پاسخ دادن به موقعیت ها از قبل تعریف نشده با انعطاف خیلی بالا.
- معنی دادن به پیام های مبهم یا نادرست.
- اختصاص دادن اعتبار نسبی به عناصر برای یک موقعیت.
- پیداکردن شباهت ها ولو اینکه موقعیت ها متفاوت باشند.
- درک تمایز ها بین موقعیت ها ولو اینکه شباهت های بسیاری بین آنها وجود داشته باشد.

به فرض اینکه تعریف بالا را در هوشمندی قابل پذیرش باشد ، موارد زیر یک لیست از وظایفی است که هوشمندی لازم دارد .

- تولید و درک گفتار.
- تشخیص الگو
- حرکت در یک فضای پر از مانع دینامیک
- اثبات قضیه ریاضی
- استدلال

اکنون تلاشی برای فهم اینکه AI چیست و چرا به AI نیاز داریم ، انجام داده ایم . اجازه دهید ، نگاهی بر پیشرفت AI از زمان شروع آن تاکنون ، داشته باشیم. به بیان دیگر، به منظور درک بهتر از اینکه از کجا شروع کرده و در کجا هستیم ، نگاهی به تاریخچه AI خواهیم داشت.

یکی ازنتایج اولیه کار با AI ، درک و تفکیک مسایلی بود که برای هوش انسان ، سخت و برای هوش مصنوعی آسان و برعکس . ملاحظه می کنیم که بشر می تواند وظایف ریاضی فکری را به سرعت انجام دهد ، از آن جایی که ما اکنون با ماشین حساب هایی آشنا هستیم که می تواند عملیات ریاضی را سریع انجام دهند .

نوشتن برنامه های کامپیوتری برای مرتب کردن پازل ها ، که تست های IQ را تشکیل می دهد ،آسان است.

اما نوشتن برنامه هایی که بتوانند کارهایی را انجام دهند ، که بشر به طور طبیعی انجام می دهـــد ، خیـــلی سخـت است ، حتی آنهایی که به طور متوسط ، هوشمندی پائینی می خواهند. برای مثال ، نوشتن برنامه های کامپیوتری برای فهمیدن جملات ساده به زبانهای محاوره ، یا تفسیر تصاویر ساده ، وظیفه سختی است که تاکنون به طور کامل حل نشده است.

این استدلال به حقیقتی منجر می شود مبنی بر اینکه کامپیوترها ماشین های ضروری برای اجرا کـــردن دستورات به صورت سریع و با دقت هستند . بشر با بینشی کارهای

خود را انجام میدهد که روانشناسان هنوز کامل آنرا درک نکرده اند.
در مورد انسان دریافته ایم که بیش از تعداد محدودی قوانین یا موضوعات را نمی توانیم به خاطر آوریم ولی هنگام حل مسائلی که به قوانین کوچک شکسته نمی شوند ، می توانیم همه ترکیبهای اطلاعاتی را که از تجربه در طول زندگی بدست آورده ایم به ذهن بیاوریم .
به عنوان مثال آنالیز هایی از دو جمله زیر را در نظر بگیرید.
« مـــن بـــرای رنگ کردن پنجره روی میز ایستادم .آن شکست و من روی بوته گل افتادم . »
« من برای رنگ کردن پنجره روی میز ایستادم .آن شکست و من روی قالی افتادم . »
این یک مثال ساده از یکی از نمونه هایی است که درآن زبان بشر مبهم است . موقعی که ما می گوئیم « آن» همیشه نمی دانیم منظورکدام شی است. در هر دو مورد « آن» می تواند میز یا پنجره باشد. اگر جملات بخشی از یک پاراگراف بزرگتر بودند « آن» به چیزی غیر از اولین ذکر شدن در جمله اولیه اشاره می کند . اگر جمله دومی چون "من نباید آن کار را انجام می دادم که میز آسیب ببیند ." دراین مثال وجود داشت ، آنگاه « آن» در جمله اول از عمل نسبت به یک شی خاص اشاره می کرد.
به هرحال در جمله اول به طور طبیعی فرض می کنیم که « آن » به معنی پنجره و در جمله دوم به معنی میز است . البته از نظر خوانندگان میتواند موضوع برعکس تلقی شود که خود گواه بر پیچیدگی و ابهام زبان محاوره ای است.
از این مثال این طور استدلال می شود که احتمالاً بوته های گل در طرف دیگر پنجره بوده و قالی زیر میز قرار دارد. از این مثال این طور دریافت می شود که آنالیز جملات زبان طبیعی احتیاج به آگاهی از حس مشترک از دنیا دارد. اگر یک برنامه کامپیوتری به منظور تفسیر جملات زبان طبیعی وذخیره اطلاعات لازم آن با استفاده از مجموعه ای از جملات منطقی ، بنویسیم ، اطمینان از اینکه این برنامه همیشه« آن» را به چیزی که انسان به طورطبیعی تفسیر می کند ، ترجمه کند ،کار سختی خواهد بود.
درحقیقت ، قرار دادن قوانینی برای پوشش هر موقعیت ممکن ، تقریباً غیر ممکن

است.

۱-۱-۳ تفاوت بین ارائه سمبلیک و غیرسمبلیک

برای ساختن سیستم های هوشمند ، لازم است ابتدا مطابق با فعالیتهای دستی درک شده از رفتار هوشمند از سمبل هایی که با فعالیت ادراکی مطابقت دارد استفاده شود. این ارائه های سمبلیک به دنیای خارجی اشاره می کنند .(شکل ۱,۱)

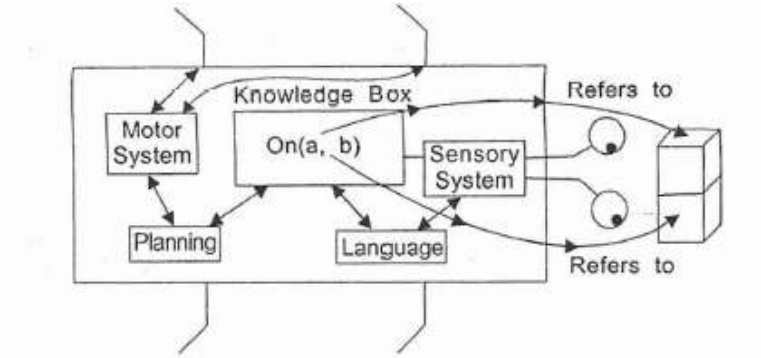


Figure 1.1 Symbolic representation.

مزایای ارائه سمبلیک عبارتند از :

ـ سازنده سیستم می تواند چیزی را که سیستم می داند ، بخواند.

ـ آگاهی و دانش با جملاتی به زبان رسمی ارائه شده است.

ـ خواندن ارائه و فهمیدن معنی دانش امکان پذیر است.

در حقیقت ، آگاهی با اوزان روی ارتباطات موجود در یک شبکه نشان داده می شود.(شکل ۱,۲)

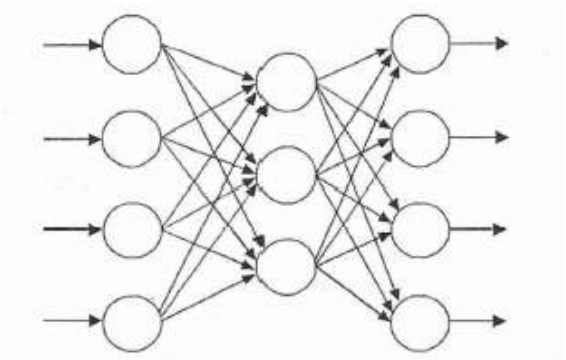


Figure 1.2 Non-symbolic representation.

ارائه غیر سمبلیک می تواند با ترکیباتی از خصوصیات ، مثل یک تصویر ارتباط برقرار کند و تحمل پذیری بیشتری نسبت به پارازیت۱ دارند.

۱-۲ تاریخچه AI

زمانی که آگاهی از تاریخچه AI برای فهم موضوعی ضروری نیست ، این تاریخچه متنی را ارائه می کند که می توانیم پیشرفت های اخیر را تفسیر کنیم و از پیشرفتی که AI تا امروز داشته است ، قدردانی کنیم .

افرادی که توانایی ها و تکنیک های سیستم AI را ثابت کرده اند آلن تورنیگ ۲،وارن مک کلوچ ۳، کلود شانون ٤، نوربرت وینر۵، جان مک کارتی ٦، ماروین مینسکی ۷، نول۸ ، سایمن۹ ، آرتور ساموئل و دیگران بودند.

۱-۲-۱ تست تورینگ

یکی از مقاله های اولیه برای نشان دادن سئوال هوش ماشین که به طور اخص در ارتباط با کامپیوتر دیجیتال امروزی است در سال ۱۹۵۰ به وسیله ریاضی دان انگلیسی ،

---

1 Noise
2 Alan Turing
3 Warren McCulloch
4 Claude Shannon
5 Norbert Wiener
6 John McCarthy
7 Marvin Minsky
8 Newell
9 Simon

آلن تورینگ ، نوشته شد.

تورنیگ که بیشتر به خاطر همکاری در تئوری محاسبه پذیری شناخته شده ، به سئوالی که آیا ماشینی واقعاً می تواند ساخته شود که فکر کند ، تلاش زیادی کرد.

نکته وجود ابهامات اساسی دراین سئوال است ابهاماتی چون فکر کردن چیست ؟ یا یک ماشین چیست ؟ این ابهامات مانند سدی در مقابل هر پاسخ مستدلل قرار داشت ، او برای گذر از این سد پیشنـــهاد کـرد که سئوال هوش با یک تست تجربی تعریف شده ی واضح تر جایگزین شود.

تست تورینگ کارایی ماشین هوشمند را در برابر انسان اندازه می گیرد چون انسان بهترین و تنها استاندارد برای رفتار هوشمند است .

این تست که تورینگ آن را « بازی تقلید » نامیده ، ماشین و انسان را در کنار هم و در اتاقی جدا از انسان سوم ، که آنرا محقق مینامد، قرار داد. محقق قادر نیست با هیچ کدام از آن دو (انسان و ماشین کنارش) به طور مستقیم صحبت کند و نمی داند کدام موجودیت واقعاً ماشین است و تنها با استفاده از یک ترمینال متنی می تواند با آنها ارتباط برقرار کند. شاید امروز خوانندگان با ترمینال متنی (در مقابل ترمینال گرافیکی) آشنایی نداشته باشند زیرا درآن زمان کامپیوترهای شخصی بوجود نیامده بود و معمولا افراد با استفاده از یک دستگاه شبیه مانیتورهای امروزی و با یک کیبورد که فقط وظیفه ارسال و دریافت اطلاعات با کامپیوتر مرکزی را داشت نیازهای خود را مرتفع میساختند.

از محقق خواسته شده که از طریق ترمینال متنی سوالاتی برای هر دو مطرح کند و براساس جوابهایی که دریافت کرده کامپیوتر را از انسان تشخیص دهد.

اگر محقق نتواند ماشین را از انسان تشخیص دهد ، آنگاه تورنیگ استدلال می کند که ماشین می تواند هوشمند فرض شود.

تست تورنیگ با جداکردن محقق از کامپیوتر و فرد شرکت کننده دیگر ، اطمینان حاصل می کند که تشخیص محقق با ظاهر ماشین یا مشخصه مکانیکی صدا تحت تأثیر قرار

.

نگرفته است. محقق در جهت تشخیص هویت کامپیوتر آزاد است هر سوال منطقی یا غیر منطقی را مطرح کند .

برای مثال محقق می تواند هم از کامپیوتر و هم از انسان ، درخواست انجام دادن یک محاسبه ریاضی نسبتاً پیچیده را بکند ، به فرض اینکه احتمال درست بودن جواب کامپیوتر نسبت به جواب انسان بیشتر باشد ، دربرخورد با این استراتژی ، کامپیوتر نیاز خواهد داشت بداند چه موقع باید به چنین سئوالاتی جواب نا درست بدهد ، به منظوراینکه شبیه انسان به نظر بیاید.

برای کشف هویت انسانی براساس طبیعت احساسی ، ممکن است محقق از هر دو برای پاسخ به یک شعر یا کار هنری مطلبی بپرسد . کامپیوتر برای مقابله با این استراتژی نیازمند شناخت و آگاهی از ساختار احساسی انسان است

ویژگی های مهم این تست عبارتند از :

الف) این تست یک مفهوم معقول از هوش به ما می دهد. مثال ، رفتار یک موجود هوشمند شناخته شده ، در پاسخ به یک مجموعه خاص از سئوالات . این تست یک استاندارد برای معین کردن هوش ارایه میدهد که از بحث های غیر قابل اجتناب روی طبیعت درستی اش پرهیز می کند.

ب) این تست ما را ازمنحرف شدن از مسیر اصلی با سئوالات گیج کننده و غیرقابل پاسخگویی منع می کند ، در همه حال کامپیوتر از پردازش های داخلی مناسب استفاده می کند و به هر حال ماشین واقعاً از اعمالش آگاه است

ج) این تست با وادار کردن محقق تنها با تمرکز کردن روی محتوای پاسخ سئوالات ، هر محرک تشخیصی را که قابلیت تشخیص موجود زنده را فراهم میکند حذف می کند.

به دلیل این مزایا تست تورنیگ ، تکیه گاهی برای بسیاری از طرح های ۱واقعی استفاده شده برای ارزیابی برنامه های AI مدرن میباشد.

برای ارزیابی هوشمندی یک نرم افزار که به استناد بعضی از شاخصهای فنی هوشمند

---

1 Scheme

تشخیص داده شده میباید کارایی آنرا ، برای حل مجموعه از مسائل با یک انسان خبره مقایسه نمود .

این تکنیک ارزیابی فقط یک نمونه از تست تورنیگ است .

از گروهی از انسانها خواسته شده است که کارایی یک کامپیوتر و یک انسان را در یک مجموعه خاص از مسائل مقایسه کنند. همان طور که خواهیم دید این روش ابزاری ضروری هم درگسترش وهم در تأیید سیستم های خبرهٔ امروزی شده است .

تست تورینگ علیرغم درک مستقیم ظاهری اش نسبت به تعدادی از انتقاد های قابل توجیه ، آسیب پذیر است . یکی از مهم ترین انتقادها ، به حمایت از حل مسئله به روش سمبلیک است. اگرچه درک مستقیم جزء مهمی از هوش انسانی هستند ، ولی توانایی هایی که نیازمند قدرت ادراک یا مهارت باشند را آزمون نمی کند .

برعکس بعضی اوقات به نظر میرسد که تست تورنیگ ، هوش ماشین را وادار به تطبیق یافتن با قالب انسانی می کند . شاید هوش ماشین متفاوت از هوش انسان است و تلاش برای ارزیابی آن با شرایط انسانی یک اشتباه اساسی است. به عنوان مثال آیا مایلیم یک ماشین عملیات ریاضی را به کندی و بی دقتی یک انسان انجام دهد ؟

آیا یک ماشین هوشمند نباید سرمایه های خودش مثل حافظه ، سریع و مطمئن را نسبت به تلاش در رقابت با ادراک بشری افزایش دهد ؟

درحقیقت تعدادی از صاحب نظران امروزی AI مانند فرد ۱و هایس۲ ، در پاسخ به این چالش، تست تورینگ را به عنوان یک اشتباه و گمراهی بزرگ در پیشرفت تئوری های عمومی برای کشف و بیان مکانیزمهای هوش در انسانها و ماشین ها ، و به کار بردن این تئوری ها برای گسترش ابزارها برای حل مسائل عملی خاص میدانند .

گرچه دانشمندان بسیاری با دغدغه های فرد و هایس موافق هستند اما هنوز تست تورینگ را به عنوان یک جزء مهم در تأیید و اعتبار نرم افزار مدرن AI قبول دارند .

تورینگ نظریه امکان پذیر پیاده سازی یک برنامه هوشمند را روی یک کامپیوتر

---

1 Ford
2 Hayes

دیجیتال را نیز معرفی کرد.

تورینگ با ارایه یک مدل خاص از ماشین محاسبه حالت گسسته الکترونیک تخمین مستحکمی در مورد ظرفیت ذخیره ، پیچیدگی برنامه وفلسفه طراحی اساسی مورد نیاز برای یک چنین سیستمی را به وجود آورد . در نهایت ، تورینگ نقایص معنوی ، فلسفی و علمی امکان ساختن چنین برنامه ای را بر حسب یک تکنولوژی واقعی بیان کرد. مثالهایی هم در مورد همکاری انسان و برنامه هوشمند وجود دارد که باعث بهبود کارایی انسان گردیده است.

سیستم (DENDRAL) مثالی خوب برای این مورد است که برای تشخیص ساختار شیمیایی مواد از طیف نگار جرم هر ماده است . با استفاده از این نرم افزار قوانینی کشف شد که تا آن زمان توسط افراد خبره نیز شناخته نشده بودند.

حال میدانیم که (DENDRAL)یک برنامه بسیار موفق بوده و در حقیقت درچنین آنالیزها کاملاً خبره است. حال هوشمندی یک ماشین به چه معنا است؟

تورینگ در مورد این مسأله در دهه ۵۰ فکر کرد و تست زیر را ایجاد کرد.

یک انسان محقق می تواند سئوالاتی را از طریق گفتن یک VDU ، به یک ماشین در یک اتاق و یک انسان در اتاقی دیگر بپرسد. اگر محقق از جوابهای دریافت شده ، قادر نباشد تصمیم بگیرد که کدام به ماشین وصل است در این صورت تورینگ می گوید ماشین فوق باید هوشمند پنداشته شود.

برنامه تقلید مکالمه با یک روانشناس (ELIZA) از اولین تلاش ها برای ساخت یک برنامه بود که بتواند در تست تورینگ قبول شود. ابتدا موقعی که شخصی این برنامه را می بیند ، کاملاً مؤثر به نظر می رسد اما اگر برای مدت کو تاهی از آن استفاده کند حقه زدن به آن و درحقیقت نحوه استنتاج و کار کردن این برنامه خیلی آسان کشف میکند . یک پیاده سازی از ELIZA را بعداً خواهیم دید.

وا ضح است به منظور قبولی در تست تورینگ ، سیستم باید زبان طبیعی را بفهمد . گفتن حداقل برای بشر که همه نوع فرضیات و استنتاج ها را به منظور معنی دادن جملات و سخن ها انجام می دهند ، کار خیلی سختی

است. باور همه این است که برای بکار بردن زبان طبیعی در یک حالت محاوره ای با انسان سیستم باید استدلال برداشت متعارف را با دسترسی به نوعی از اطلاعات عمومی که یک انسان انتظار دارد بداند، ترکیب کند.

تلاشی برای تامین کردن اطلاع پایه، پروژه Cyc از Lenat است .تورینگ درحال ساخت یک سیستم عظیم است که سعی می کند تمامی دانش بشر را تسخیر کند. به کمپانی CYCORPعرضه این تکنولوژی واگذار شده است .

این تیم متون لازم را از دایره المعارف ها و روزنامه ها و ... بررسی و برای قابل استفاده کردن چیزی که نویسنده فرض می کند ماشین هوشمند از دنیا می داند آنرا در پایگاه داده درج می کند . به طور معمول چندین میلیون ورودی (مدخل) دراین پایگاه دانش وجود دارد.

۱-۳ کاربردهای AI

۱-۳-۱ پردازش زبان طبیعی

وقتی افراد با یکدیگر ارتباط برقرار می کنند ، تقریباً بدون تلاش خاصی وخیلی پیچیده با یکدیگر تعامل وهمکاری می کنند و با اطلاعات ناچیز می توانند فرآیند ها را درک کنند . ساختن کامپیوترهایی که توانایی تولید ودرک حتی کسر کوچکی از زبان طبیعی مثل انگلیسی را داشته باشد ، خیلی سخت است . این سختی ، به دلیل اینکه زبان به عنوان رسانه ارتباطی موثر در بین افراد با هوش توسعه یافته است ، می باشد . به نظر می رسد در نتیجه انتقال ذره ای از ساختار فکری ، تحت شرایط محیط ، از یک مغز به مغزی دیگر ، هر مغزی دارای ساختارهای بزرگ فکری مشابه است ، این متون به عنوان متن عمومی مشترک در درک زبان بکار گرفته می شود . این تشابه در متون ، در ایجاد و فهم پیام های خیلی فشرده کمک می کند . بنابراین درک زبانهای طبیعی یک مشکل پیچیده و بزرگ از رمزگذاری و رمز گشایی است .

یکی از اهداف دائمی و همیشگی هوش مصنوعی ، خلق برنامه هایی که در درک و ایجاد زبان بشری توانا هستند ، می باشد . به نظر نمی رسد که تنها توانایی استفاده و

درک زبان طبیعی ، خصوصیت بنیادی هوش بشری باشد ، اما عدم کنترل و هدایت موفق دستگاه ها به طور خودکار ، یک ضربه باور نکردنی روی قابلیت استفاده و سودمندی کامپیوترها دارد . تلاش های بسیاری در نوشتن برنامه هایی که زبان طبیعی را درک می کنند ، صورت گرفته است . اگرچه این برنامه ها درمتون محدود به موفقیت هایی دست یافته اند ولی سیستم هایی که بتوانند زبان طبیعی را با انعطاف پذیری و عمومیتی که بشر صحبت می کند ، استفاده کنند هنوز تولید نشده است زیرا فراسوی گفتار با روش و چارچوب رایج است .

پیچیدگی درک زبان طبیعی خیلی بیشتر از تجزیه جملات در قسمتهای منحصر به فرد صحبت کردن و جستجوی آن لغات در فرهنگ لغت است . ادراک واقعی بستگی به عوامل زیادی دارد از جمله به وسعت دانسته های قبلی درباره قلمرو مباحثه و اصطلاحات استفاده شده در آن قلمرو و نیز توانایی بکار بستن آگاهی وابسته به قراین عمومی ، جهت رفع و حذف ابهاماتی که در یک قسمت نرمال از گفتار بشر وجود دارند ، بستگی دارد .

بنابراین به منظور ساخت سیستم های کامپیوتری که بتوانند زبان طبیعی را درک کنند ، هم آگاهی وابسته به قراین و هم فرآیند جهت ساخت استنباط های موثر ، لازم است . محققان زیادی مجذوب این قلمرو خیلی مهم از سیستمهای هوشمند شده اند .

۱-۳-۲ بازیابی هوشمند از پایگاه داده

سیستم های پایگاه داده ، شامل حوزه وسیعی از حقایق ، درباره بعضی موضوعات هستند که جهت پاسخ به پرس و جو های ۱کاربران در رابطه با آن موضوع استفاده می شوند . برای اطلاعات هر هنرپیشه که در پایگاه داده وارد شده است ، ممکن است داده هایی نشاندهنده سن ، قد ، وزن ، تعداد کلاه ، کل درآمد ، ... وجود داشته باشد . با این حقایق ذخیره شده در یک شکل منظم ، هر کسی می تواند پاسخ پرس وجو ها یی نظیر " لیست همه هنرپیشه گانی که قد آنان بیش از ۱۷۰ سانتی متر است ." یا "متوسط حقوق سالیانه هنرپیشه X چقدر است ؟" را به دست آورد . طراحی سیستم های پایگاه

---

1 QUERY

داده هنوز یک فیلد بسیار فعال در محدوده علم کامپیوتر است . از نقطه نظر AI ، اگر پاسخ ها به استدلال های قیاسی با حقایق موجود در پایگاه داده ، نیاز داشته باشیم این فیلد حتی از این هم مهمتر خواهند شد .

برای ایجاد سیستم های بازیابی هوشمند داده ها تلاش ویژه ای صورت گرفته اما مشکلات فراوانی نیز وجود دارد. اولاً ، مشکل درک پرس و جوهای اظهار شده در یک زبان طبیعی نظیر انگلیسی . ثانیاً ، با فرض اینکه سیستم پرس و جوها را کاملاً درک کرده باشد ، چگونه باید پاسخ را از داده های موجود در پایگاه داده استنتاج کند . ثالثاً ، برای اطلاعات داده شده چندین پرس وجو که نیازمند اطلاع عمومی است ، وجود دارند . برای مثال ، اگر هنرپیشه P کاپیتان باشد ، آنگاه سیستم باید در وضعیت دانستن اینکه P تیم را همیشه در یک میدان مسابقه راهنمایی می کند و قسمتی از هسته گروه که تیم را انتخاب کرده است ، باشد . چنین حقایقی ممکن است در طی توسعه پایگاه داده به صراحت اظهار نشده باشد .

۱-۳-۳ سیستم های خبره

سیستم های خبره ، سیستم های مشاور اتوماتیک هستند . این سیستم ها استنتاج های خبره راجع به محدوده هایی که در آن تخصص یافته اند را ارایه می کنند . به عنوان مثال سیستم های خبره طوری ساخته شده اند که می توانند عیوب موجود در سیستم های نظامی نظیر هواپیماها ، رادارها و سایر ادوات نظامی را تشخیص دهند ، اعضاء یک نوع خاص از موجودات را به صورت علمی طبقه بندی نمایند، پیش بینی بر روی ساختار های شیمیایی ممکن از مواد مختلف ارایه نمایند، عامل بالقوه سنگ معدنی که ته نشین می شود را شناسایی کنند، یا حتی امراض را با توجه به شواهد موجود تشخیص دهد .

در طراحی هر سیستم خبره برای حل هر مسئله دو کلید ترکیب شده وجود دارد . یکی از این کلیدها نشاندهنده دانش موجود از موضوع ودیگری کاربرد این دانش در هنگام

ارایه نتایج است . نمایش دانش قدری پیچیده است زیرا که دانش می تواند مبهم و یا نامعلوم باشد . اساساً دانش به عنوان یک مجموعه بزرگ از قوانین ساده ارائه و در ساختارهایی که قالب۱ و متون اخباری ۲، نامیده می شود ، نمایش داده میشود . نتایج یا پاسخ عموماً از تکنیک های استنتاج قوانین پایه ای ، حاصل می شوند . در سال های اخیر شکل های دیگر استنتاج مانند استنتاج احتمالی به طور جدی گسترش یافته اند .

نرم افزار DENDRAL یکی از سیستم های اولیه برای به کار بردن دانش حوزه خاص جهت حل مسائل بود که در استنفرد۳ و در اواخر دهه ۱۹۶۰ توسعه داده شد. DENDRAL برای استنباط ساختار بنیانی مولکول ها از فرمول های شیمیایی شان وبا تکیه بر دانش وسیعی از تجزیه و روابط شیمیایی موجود در مولکول ها ، طراحی شده بود . چون مولکول های بنیادی میل دارند که خیلی گسترده باشند ، تعداد ساختارهای ممکن برای این مولکولها نیز تمایل به بزرگ و گسترده شدن دارند . DENDRAL به وسیله بکار بردن دانش اکتشافی شیمیدانان خبره و متخصص ، جهت مسئله روشن سازی ساختار مشکل این فضای وسیع تحقیق را حل می کند . روشهای DENDRAL ، فوق العاده موثر بودند ، تنها بعد از یک آزمایش کوتاه ، ساختار صحیح مولکولها را از میان میلیونها احتمال پیدا می کنند . این عملکرد باعث تضمین بکارگیری نسل های بعدی این سیستم در آزمایشگاه های شیمیایی و دارویی در سراسر دنیا گردید .

از آنجاییکه DENDRAL ، یکی از اولین برنامه ها در استفاده موثر دانش در حوزه خاص جهت دست یافتن به کارایی حل مسائل در سطح دانشمندان خبره بود ، MYCIN نمونه دیگری از برنامه هاست که با رویکرد ایجاد برنامه خبره در حوزه خاص پایه گذاری شد. MYCIN از دانش متخصصین و خبرگان پزشکی جهت تشخیص و تعیین درمان تورم ستون فقرات و عفونت میکربی در خون استفاده می کرد. MYCIN که در استنفرد و اواسط دهه ۱۹۷۰ توسعه داده شده ، یکی از اولین برنامه

---

1 Frame
2 Script
3 Stanford

ها جهت نشان دادن مشکلات استدلال با اطلاعات نامعلوم وناقص بود . MYCIN شرح وتفسیر استدلالش را واضح ومنطقی تهیه کرد و از ساختار کنترل مناسب برای حوزه مسائل خاص و ضوابط شناخته شده جهت ارزیابی کارایی اش به طور قابل اعتماد ، استفاده کرد.

۱-۳-٤ اثبات قضیه

اثبات یا رد کردن قضایای ریاضی یک کار کاملاً فکری است . این مسئله به این دلیل است که اثبات یا رد این قضایا به استنتاج هایی از فرضیات نیاز دارد و علاوه بر آن با حکم هم درگیر می باشد . این حکم بر روی حجم وسیعی از اطلاع تخصصی ، پایه گذاری شده است و برای حدس صحیح راجع به چیزی که قبلاً ثابت شده است ، قضیه با دلایل موجود کمک می کند . این موضوع در شکستن مسائل به مسائل کوچکتر جهت انجام کار به طور مستقل کمک می کند . چندین برنامهٔ اثبات قضیه خودکار که توانایی به طور مستقل کار کردن را دارند ، توسعه داده شده اند . فرمول بندی تکنیک های قیاسی که از زبان منطق گزاره ای استفاده می کند ، در درک اجزاء استدلال ، به طور واضح تر کمک می کند . وظایف غیر رسمی بسیاری مانند تشخیص پزشکی می تواند مسائل اثبات قضیه را فرمول بندی کنند .

بیشترین جاذبه مکانیزه شدن اثبات قضیه ها ، دقیق وعمومی بودن منطق است . چون منطق یک روش رسمی است ، منطق خود به فرآیند مکانیزه شدن کارها منجر می شود. طیف وسیعی از مسائل از طریق نشان دادن خود مسئله و اطلاعات زمینه ای به عنوان اصول منطقی در رابطه با آن مسئله و نمونه های قضایایی که اثبات شده اند ، می توانند وجود داشته باشند . یک دلیل برای علاقمندی اثبات قضیه ها بصورت اتوماتیک ، تفهیم مسائل است که چنین سیستم هایی ، توانایی حل مسائل خیلی پیچیده را بدون کمک بشر و به طور مستقل ندارند . اما بصورت کاربردی در اثبات بسیاری از قضایای امروزی ، به عنوان دستیاران هوشمند وظایف سخت تجزیه مسائل بزرگ به مسائل کوچکتر و استنتاج اکتشافی برای جستجو در فضای استدلال منطقی را انجام دهند .

بنابراین اثبات قضیه ساده تر انجام می شود .این سیستمها هنوز به دنبال انجام اثبات اصل موضوع ، بازبینی تخمین های کوچکتر ، و کامل کردن خصوصیات رسمی یک اثبات طرح ریزی شده به وسیله همکاری ومشارکت با انسان هستند وبه همین دلیل قضیه یک زیر بخش مهم از AI است .

۱-۳-۵ رباتیک

انسانها قادرند در محیط شان جابجای یا تغییر صورت دهند مثلا جعبه اسباب بازی را جابجا کنند یا وضعیت خاموش و روشن بودن کلید را تغییر دهند. اگرچه به طور ناخودآگاه این اعمال به وسیله انسان انجام می شود ، اما انجام آن با پیچیدگی ها ی فراوانی همراه است . وقتی برای نوشتن برنامه ای که رفتار مشابه را در ماشین های هوشمند بوجود آورد تلاش می کنیم ، درمی یابیم که این کار نیازمند توانایی های بسیاری است که فقط در حل مسائل بسیار عقلانی تر وجود دارد.

در فرآیند طراحی رباتی که کارهایش را در رابط با جهان خارج با درجه ای از حساسیت و انعطاف پذیری انجام میدهد یکی از مهمترین نکات برنامه ریزی است. به طور خلاصه ، در برنامه ریزی ، رباتی را در نظر میگیریم که قابلیت انجام اعمال تجزیه ناپذیر معین را دارد . برنامه ریزی برای یافتن ترتیبی از آن اعمالی که بعضی وظایف سطح بالاتر نظیر حرکت از یک سو به سوی دیگر در یک اتاق پر از مانع را انجام خواهد داد ، تلاش می کند .

به دلایل زیادی برنامه ریزی کار سختی است ،نه بدلیل اینکه حالتهای حرکات پیوسته ی ممکن زیاد است زیرا حتی یک ربات بی نهایت ساده هم توانایی انجام تعداد زیادی از حرکتهای پیوسته زا به صورت بالقوه دارد . برای مثال ، رباتی را که می تواند به سمت جلو ، عقب ، راست یا چپ حرکت کند ، تصور کنید ، و چگونگی راه های مختلفی که ربات احتمالاً می تواند اطراف اتاق حرکت کند ، در نظر بگیرید . همچنین فرض کنید که در اتاق موانعی وجود دارد که ربات مجبور است مسیری را انتخاب کند که با یک روش موثر و کارآمد آن را اطراف اتاق حرکت دهد . نوشتن برنامه ای که بتواند به طور هوشمندانه بهترین مسیر را بدون دست پاچه شدن به سبب زیاد بودن

تعداد احتمالات و تحت شرایط محیط پیدا کند ، ربات به تکنیک های خبره وماهر برای ارائه دانش فاصله ها و کنترل جستجوی سراسر محیط ممکن ، نیازمند می باشد . تحقیق در مورد رباتیک به توسعه وگسترش بسیاری از تکنیک های AI برای حالتهای مدلسازی از دنیا و تغییر شکل دادن از یک فرم به فرمی دیگر که پیدا شده اند کمک می کند .

۱-۳-٦ مسائل زمان بندی و ترکیبی

یک دسته از مسائل با زمان بندی بهینه مشخص ، مربوط شده اند . یک مثال کلاسیک ، مسئله مسافرت فروشندگان است ، جائی که مسئله ، پیدا کردن یک تور با کمترین مسافت ، عازم شدن به یکی از چندین شهرها ، دیدن شهرهای دیگر فقط یک بار و سپس بازگشت به شهر مبدأ است . مسئله ، به یافتن مسیر با کمترین هزینه روی لبه های گراف که شامل n گره[1] می باشد در حالی که هر گره بیش از یک بار ملاقات نمی شود ، تعمیم پیدا می کند . در چنین مسائلی دامنه ترکیبات یا ترتیبات احتمالی از چیزی که یک جواب خیلی وسیع دارد ، انتخاب می شود . Brute force برای ایجاد راه حل که اغلب به انفجار ترکیبی از احتمالات که حتی منابع کامپیوترهای بزرگ را تمام می کند ، منجر می شود . چنین مسائلی به وسیله محاسبه به عنوان مسائل کامل NP ، نامیده می شوند .

محققین AI روی متدهایی برای حل گونه های مختلفی از مسائل ترکیبی ، کار کرده اند . کلید حل چنین مسائلی ، اطلاع دربارهٔ دامنهٔ مسئله می باشد . متدهای توسعه داده شده ای که جهت حل مسائل ترکیبی ای اثبات شده اند ، در حل دیگر مسائل سختی که حالت ترکیبی کمتری دارند ، موثر می باشند .

۱-۳-۷ مسائل ادراکی

تلاشهایی برای ساختن کامپیوترهایی که ببینند (با متناسب کردن شان با ورودی های تلویزیون) و بشنوند (با قرار دادن ورودی های میکروفن) صورت گرفته است . این

---

1 Node

تلاشها تنها موفقیت محدودی بعد از پردازش مفید

در AI ، پردازش ادراکی به عنوان یک مجموعه ای از عملیات مورد مطالعه قرار گرفته است . یک منظره بصری ممکن است به وسیله گیرنده های حسی و نیز نمایش به عنوان ماتریسی با ارزش که نمایش داده شده ، رمزی شده باشد . این اعمال به وسیلهٔ ردیاب هایی که اجزاء تصویر پیشین مانند سگمنت های خطی ، منحنی های ساده ، زوایا و ... را جستجو می کنند ، پردازش شده اند . برعکس این اعمال برای استنباط کردن اطلاعات راجع به اشیاء در منظره پردازش شده اند . هدف نهایی ارائه یک منظره با یک مدل مناسب است . این مدل ممکن است یک توصیف سطح بالا ازمنظره ، مانند یک تپه با یک درخت روی آن ، باشد .

نکته مشکل درک کلی ، تأمین یک نسخه خلاصه شدهٔ ورودی از مقادیر وسیع غیر قابل مدیریت داده ورودی خام است. اشکال اصلی در مسائل ادراکی ، تعداد بسیار زیاد توصیف های کاندید از یک منظره است. یک راه حل برای این مورد ، استراتژی ساختن فرضیه از هر توصیف و سپس تست کردن آنها به منظور رسیدن به هدف است. برای تشکیل این فرضیه ، دانش زیادی در بارهٔ مناظر مورد انتظار نیاز است .

۱-۳-۸ معماریهای وابسته به سلسله اعصاب .

معماری های عصبی به عنوان مکانیزمهایی برای پیاده سازی هوش برای بعضی اهداف جذاب هستند . برنامه های سنتی AI بی دوام و خیلی حساس به پارازیت هستند ، چنین برنامه هایی تمایل دارند یا درست باشند یا اینکه کلاًّرد شوند .

هوش انسان انعطاف پذیر تر است . ما در تفسیر ورودی پر سروصدا مثل تشخیص یک چهره در یک اتاق تاریک از یک زاویه عجیب یا دنبال کردن یک مکالمه دریک مهمانی شلوغ ماهر هستیم . حتی جایی که یک انسان ممکن است قادر به حل بعضی مسائل نباشد ، می توانیم برای رسیدن به راه حل یک حدس معقول بزنیم.

چون معماری های عصبی ، دانش را در واحد های با قطعه کوچک ضبط می کنند به نظر می رسد که پتا نسیل بیشتری برای تطبیق جزئی داده ی پر خش و ناکامل را دارند. همچنین معماری های عصبی تنومند تر هستند چون دانش تا حدی به طور یکنواخت

درشبکه توزیع شده است . به علاوه ، معماری های عصبی یک مدل طبیعی برای مشابهت تأمین می کنند ، چون هر رشته عصبی یک واحد مستقل است .
Hillis روی حقیقتی که بشردر یک وظیفه هنگامی که دانش بیشتری کسب می کند، سریع تر عمل می کند ، در حالی که کامپیوتر ها تمایل به کم کردن سرعت دارند ، توضیح داده است .
این کم کردن سرعت به منظور هزینه جستجوی ترتیبی یک پایگاه دانش است . یک معماری به طور عظیم شبیه مغز انسان از این مشکل رنج نمی برد .

۱-۳-۹ بازی کردن game

بیشتر تحقیق اولیه در وضعیت فضای جستجو با استفاده از بازیهای تخته ای نظیر چکرز ، شطرنج و معمای ۱۵ تکه انجام شده است .
علاوه بر ظاهر عقلانی ذاتی آنها ، بازیهای تخته ای خاصیت های معینی دارند که آنها را موضوعات ایده آل برای کار اولیه ساخته است .
اغلب بازی ها با استفاده از مجموعه قوانین خوش- تعریف انجام می شوند : این قوانین تولید فضای جستجو را آسان می کند و محققان را از ابهامات زیاد و پیچیدگی های ذاتی درمسائل کمتر ساخت یافته رها می سازد . شکل های تخته استفاده شده در این بازیها به آسانی روی یک کامپیوتر بدون نیاز به رعایت ظاهر پیچیده لازم برای گرفتن مهارت های معنایی دامنه های مسئله پیچیده تر ، ارائه می شوند. بازی ها می توانند فضاهای جستجوی خیلی بزرگتری تولید کنند. این بازی ها به قدری بزرگ و پیچیده هستند که نیازمند تکنیک های قدرتمند برای تعیین چاره هایی به منظور کاوش در این فضای مسأله می باشند. این تکنیک ها مکاشفه ای نامیده می شوند و یک محدودهٔ عمده از تحقیقات AI را تشکیل می دهند. بیشتر چیزی که ما معمولاً هوش می نامیم ، به نظر می رسد در مکاشفه های استفاده شده به وسیله انسانها برای حل مسائل ، مستقر شده باشد.

۱-٤ اهداف AI

.

اهداف اصلی تحقیق AI عبارتند از :

● فهمیدن ادراک انسان . مثلاً چگونه انسان ها مسائل را حل می کنند؟ سعی برای بدست آوردن دانش عمیق حافظه انسان ، توانایی های حل مسأله ، آموزش و تصمیم گیری و....

● خودکار سازی صرفه اقتصادی ، انسانها را در وظایف هوشمند جایگزین می کند. آیا برنامه هایی که به خوبی بشر کاری را انجام می دهند ، وجود دارند ؟

● توسعه هوشمند صرفه اقتصادی سیستم هایی را که به انسانها کمک می کنند بهتر ، سریعتر و عمیق تر فکر کنند ، می سازند . برای مثال سیستمی که به GP کمک می کند تا بیماریها را تشخیص د هد.

● هوش فوق بشری برنامه هایی می سازد که از هوش انسان تجاوز می کند.

●حل مسأله عمومی، حوزه وسیعی ازمسائل راحل می کند . سیستم هایی به وسعت ذهن .

● مباحثه دارای نتیجه منطقی با انسانها به وسیله زبان طبیعی ارتباط برقرار می کند . یک مکالمه هوشمند انجام می دهد.

● خود مختاری ، سیستم های هوشمندی دارد که عامل بر روی ابتکار خویش است . باید به دنیای واقعی عکس العمل نشان دهند.

● آموزش (استقراء) سیستم باید به نحوه گرد آوری داده علاوه بر اینکه خودش آن را جمع آوری کند ، قادر باشد . عمومیت دادن ، فرض کردن، به کاربردن/آموزش اکتشافی ، برهان با مشابهت سازی .

● ذخیره اطلاعات و دانستن چگونگی بازیابی آن .

۱-۵ برنامه نویسی هوش مصنوعی

رهیافت مهندسی نرم افزار مرسوم به برنامه نویسی به شروع با نیازمندی ها ، گسترش خصوصیت های رسمی ، شکستن مسأله به قسمت ها و در مرحله آخر پیشرفت ، کد کردن ،تأکید دارد . به هر حال برنامه نویسی سیستم های هوش مصنوعی به راحتی با مدل های مهندسی نرم افزار مرسوم تطبیق نمی کنند . به دلیل طبیعشان ، مسائل AI

نمی توانند به طور رسمی مشخص شوند . درستی یک سیستم AI به وسیله کارایی اش که مطابق بعضی خصوصیات رسمی باشد اندازه گیـــری نمی شـــود ، بلکه به وسیلـــه نزدیکی اش به کارایی انسان اندازه گرفته می شود. این نتیجه می دهد که یک رهیافت به برنامه نویسی ، به عنوان « اکتشافی » یا « شکل اولیه » توصیف می شود. یعنی برنامه ها به روشی کمتر رسمی ، نسبت به مهندسی نرم افزار مرسوم توسعه داده شده اند ، ولی ممکن است به عنوان الگوهای تجربی نسبت به محصولات نیمه کامل مورد توجه قرار گیرند . یک برنامه AI ممکن است با یک محدوده ازمثال های انسانی تست شوند و در صورت لزوم طوری اصلاح شوند که مثل یک انسان روی آن مسائل عمل کنند. بعضی از اصلاحات ممکن است در بهبود کارایی (با توجه به نزدیکی به کارائی بشر) ، در مثال های مختلف موفق شوند . بقیه ممکن است فقط در یک یا دو مثال مفید واقع شوند و در بقیه مثال ها کارایی کمی داشته باشند. به هرحال ، زمانی که اصلاحات انباشته می شوند ، ساختار برنامه ممکن است موقع واکنش اصلاحات به بعضی اصلاحات دیگر، پیچیده شود . سرانجام ، این نکته ممکن است جایی حصول شود که برنامه واگذار شده است ولی درس های آموخته شده از ساخت آن ، نگهداری شده اند و برای ساخت یک برنامه بهتر استفاده می شوند. این پـــردازش مـمکن است چندین چرخه انجام دهد ، هر چرخه یک الگوی بهتر می سازد. این به عنوان رهیافت به برنامه اجرا ـ فهم ـ ا شکال زدایی ـ ویرایش (RUDE) توصیف شده است . اگرAI را به عنوان جنبه هایی از برنامه نویسی که ضرورتاً با استفاده از متد RUDE نسبت به رهیافت مهندسی نرم افزار رسمی تر ، نوشته شده اند ، تعریف کنیم ، آنگاه بازی شطرنج ، که ما قبلاً گفتیم یک بخش مشکوک AI است ، می تواند به عنوان یک بخش ثابت AI دیده شود. اکنون ممکن است قادر به نوشتن برنامه ای باشیم که بتواند برهمه به جز استثنائی ترین شطرنج بازان انسانی غلبه کند . اما نمی توانیم مشخصات رسمی ای بنویسیم که بهترین حرکت بعدی را در هر موقعیت داده شده تعیین کند. اصلاح مدل اساسی بازی شطرنج با اضافه کردن تغییراتی به رویه جستجوی اصولی که بـــه

روش تجـربی با اجرای برنامه کامل دربرابر بازیکن انسانی تست می شوند ، درگیراست .

نکته اینکه پروسه مهندسی نرم افزار تولید یک برنامه از مشخصات رسمی ، یک مسأله AI است ، درغیر اینصورت ما مجبور به استخدام برنامه نویسان برای انجام آن نبودیم و می توانستیم آن ر ا به طور خودکار انجام دهیم. « برنامه نویسی خودکار» خودش یک شاخه از AI است . نکته اینکه این اصطلاحی است که معنی اش را با گذشت زمان تغییرداده است . وقتـی برنامه نویسان به زبان ماشین برنامه نوشتند ، کامپایلرهایی که زبان سطح بالای اولیه (مثل فرترن) را به کد ماشین ترجمه کردند ، به عنوان سیستم های « برنامه خودکار» توصیف شدند. امروزه این اصطلاح را برای آن سیستم هایی به کار می بریم که برنامه های اجرایی از مشخصات غیر قابل اجرا تولید می کنند . آن مشخصات ممکن است درمنطق یا درزبان مشخصات رسمی دیگر باشد ، ولی شاخه دیگر برنامه نویسی خودکار« ترکیب وتلفیق مثالها » برنامه هایی را از مشخصات غیر رسمی که شامل مثال هایی از ورودی و خروجی های مورد انتظار ازاین ورودی ها است ، تولید می کند . برنامه نویسی AI اغلب از زبانهای اخباری که برای نزدیک ترشدن به روش استدلال انسان نسبت به زبانهای برنامه نویسی مرسوم درباره مسائل تلاش می کنند ، استفـــاده می کند . طبیعت اکتشافی برنامه نویسی AI به این معنی است که ضروری است برنامه های AI قابل فهم بودنشان را حفظ کنند و در جزئیاتی که نیازمند دستکاری حافظه کامپیوتر است درگیر نشوند . کار اخیر در AI در زبان LISP انجام شده ، یک پیشرو به زبان تابعی Miranda . استدلال استفادهٔ LISP ، استفاده از لیست ها برای ذخیره داده و تأکید روی دستکاری نمادهای مناسب تر از آرایه ها و داده عددی علمی / مهندسی (در فرترن) و پردازش داده (در کوبول) برای کار AI ، بود. PROLOG ، زبان ارجح برای بسیاری برنامه نویسان AI است که خصوصیات پردازش لیست و دستکاری نماد LISP را دارد. ایده آل توسعه دهندگان PROLOG

منطق قابل اجرا بودن روی ماشین بو د، به این معنی که برنامه PROLOG خواستار

مشخصه خودش است .

بنابراین یک برنامه AI به یک روش اکتشافی گسترش پیدا کرد که کاملاً مشخص شده است . تساوی الگوریتم = منطق + کنترل ، پیشنهاد شده است ، پیشنهادی که برنامه نویس نیاز دارد به آسانی روابط منطقی خاص بین داده و کنترل را که خواستارخصوصیت مجزا که با کامپیوتر مطابق با قوانین بیان شده در منطق دستکاری شده مدیریت شده است ، مشخص کند .اظهار شده که زبانهای دستوری ، به دلیل ادغام جنبه های منطقی برنامه نویسی با جنبه های کنترلی آن گیج کننده هستند .

یک جنبه مهم جدایی منطق و کنترل که جایگاه مهمی در برنامه نویسی AI دارد ، فرق بین زبان وشبه زبان است . برنامه نویسی مرسوم یک جدایی بین داده برنامه و خود برنامه نوشته شده در زبان برای دستکاری آن داده ایجاد می کند. برنامه نویسی AI اغلب یک جدایی سه گانه در داده ، قوانین دستکاری داده و قوانین بیان چگونگی دستکاری قوانین داده ی استفاده شده را ، ایجاد می نماید. شبه برنامه یک مجموعه قوانین سطح بالا است ؛ برنامه ای که با یک برنامه به عنوان یک داده برخورد می کند.

۱-٦ انتقاد از AI

یک انتقاد برحق از AI این است که AI روی micro – worlds متمرکز است و واقعاً به هوش به عنوان مدل رفتار بشر نگاه نمی کند ـ AI در باره حل مسائل مهارت آمیز است . شاید این مشکل بیشتر درمورد نام « هوش مصنوعی » است .

یقیناً می توانیم به AI ، به عنوان یک برنامه مفید ودرستی که به روش های نوشتن برنامه های کامپیوتری که می توانند مسائل سخت را حل کنند ، توجه دارد ، بنگریم چه این برنامه ها بتوانند به عنوان بلاکهای ساخت در برابر هدف بزرگتر ساخت «موجودات بشری مصنوعی » در نظر گرفته شوند یا نه . به هرحال درپی این انتقادات یک روند رشد در AI برای تلاش و گرفتن بعضی جنبه های ا ستدلال « برداشت متعارف » بوده است . برای نمونه ، منطق پیشگویی استاندارد می تواند به دلیل ناتوان بودن در ارائه حقیقتی که غالباً فرض های قراردادی برایش در نظر می گیریم ، مورد

انتقاد واقع شود.

برای مثال می دانیم که اگر حقایق $p(x) \rightarrow q(x)$ را وقتی x متغیر است و p(a) موقعی که a یک ثابت است ، داشته باشیم می توانیم q(a) را نتیجه بگیریم . به عنوان نوعی مثال فرض کنید بگوییم « همه دانشجویان علوم کامپیوتر درس منطق را می گیرند. » می توانیم بیان کنیم که $CompSci(x) \rightarrow Logic(x)$. و اگر حقیقت « Gopel یک دانشجوی علوم کامپیوتر است » را داشته باشیم با توجه به $CompSci(x) \rightarrow Logic(x)$ می توانیم نتیجه بگیریم Logic (Gopel) یعنی Gopal ، درس منطق را می گیرد .

اگر چه این یک استدلال شناخته شده است که هر کسی می تواند p(a) را از q(a) نتیجه بگیرد با وجود $P(x) \rightarrow q(x)$ ، اما اگر حقیقت « Rajesh ، درس منطق را می گیرد» داشته باشیم ، نمی توانیم « Rajesh دانشجوی علوم کامپیوتر است » نتیجه بگیریم . چون ممکن است دانشجویان ریاضی هم این درس را بگیرند .

مردم اغلب این گام غلط را در استدلال دارند و همیشه نمی توانیم آنها را از غیر منطقی بودن رها کنیم . ممکن است موردی باشد که همه دا نشجویان علوم کامپیوتر درس منطق را بگیرند، ولی یک یا دو دانشجو از دیگر گروههای آموزشی هم وجود دارند که درس منطق را می گیرند . پس این یک استنتاج برداشت عمومی معقول است که اگر داشته باشیم « Rajesh درس منطق را می گیرد » بگوییم که Rajesh دانشجوی رشته علوم کامپیوتر است مگر اینکه دلیلی داشته باشیم که خلاف آن را بگوید . منطق پیشگویی متداول این نوع جهش را نمی تواند انجام دهد ، اما تحقیق در زمینه منطقی که قابل تغییر باشد ، بوده است . دراین مورد یک عنصر احتمال درگیر وجود دارد : فرض ، یک برداشت متعارف معقول تحت شرایط داده شده بالا است ، اما معقول نیست اگر ما دانشجویان علوم کامپیوتر را فقط جز کوچکی از آنهایی که درس منطق را می گیرند ، می پنداشتیم .

ایدهٔ micro-worlds وابسته به « فرض دنیای محدود » است ، این فرضی است که داده ایی که ما ارائه کرده ایم ، همه چیزی است که شناخته شده است . این ما را قادر

می کند ، فرض کنیم که اگر نتوانیم درستی چیزی را ثابت کنیم ، پس آن چیز باید غلط باشد . این فرض ممکن است که در یک micro-world معقول باشد ، ولی نه موقعی که ما درمورد دنیای واقعی استدلال می کنیم . برای مثال : اگر قوانینی که ما در مورد اینکه چه کسی درس منطق را می گیرد فقط $CompSci(x) \rightarrow Logic(x)$ و $Maths(x) \rightarrow Logic(x)$ باشد : می توانیم نتیجه بگیریم اگر (Ravi) Compsci یا Maths (Ravi) را نداشته باشیم Logic(Ravi) غلط است . (یعنی Ravi درس منطق را نگرفته است.)

این استنتاج درفرض دنیای محدود ، همیشه درست خواهدبود ، ولی دنیای واقعی محدود نیست و در عمل ممکن است استدلال هایی دربارهٔ دانشجویی که درس منطق را می گیرد ، جدا از اینکه او دانشجوی کامپیوتر است یا ریاضی ، وجود داشته باشد که نمی شناسیم . البته هنوز می توانیم فرض کنیم Ravi درس منطق را نمی گیرد ولی مثل مورد بالا این فقط فرضی است که تا زمانیکه دانش بیشتر بدست بیاوریم ، در نظر می گیریم .

۱-۷ آیندهٔ AI

AI دردهه های بعدی در بسیاری زمینه ها گسترش خو اهد یافت . از منظرهای مختلف نسبت به آینده ، یک نوع خاص آن ، در جهت تعامل با بسیاری رویه های دیگر اهمیت دارد . این اصول به مواردی که بشر نقش مهمی را در اعمال و تصمیمات آنها بازی می کند ، مربوط است . به جهت دقت بیشتر ، منظورمان اعمال پیچیده ای است که یکی درمیان به وسیله بشر و ماشین انجام می شوند . چنین موقعیت هایی به طور متناوب وقتی توانایی ماشین آنقدر برای انجام دادن همه اعمال کافی نیست ، اتفاق می افتد . ولی می تواند بخش هایی از آن اعمال و هنگامی که برنامه ریزی و اجرای اعمال ناپیوسته شده اند ، انجام دهد .

ایـــن درتضاد با فعالیت های اخیر در AI در دهه های گذشته است ، فعالیت هایی که به طور عمده از یک نقطـــه نظر منطقـــی انجام شـــده انـــد . سیستم هـــا نقش یک

جعبه سیاه ۱ را داشتند و استدلال منطقی دغدغه اصلی بود . در حقیقت این جعبه سیاه اثر خوبی روی داستانها برای ارائه ، داشت .

در کاربردهای عملی ، سیستم های بر مبنای دانش ، گسترش پیدا کردند ، برای مثال جهت تشخیص (هم در زمینهٔ پزشکی هم در دامنه های تکنیکی)، پیکربندی و طراحی و برای برنامه ریزی عملی .

طولی نکشید ، که کافی نبودن منطق به تنهایی به عنوان یک دیدگاه درک شده بود . بنابراین منطق با فرم های مختلف استدلال عدم قطعیت تکمیل شد ، مثل فاکتور های عدم قطعیت ، شبکه های Bayesian ، مدارک ، مجموعه های فازی ، استدلال مشابهت و ....

یک مشاهده این است که این تلاشها اغلب فرض های ریاضی دقیق داشته اند که تکمیل شده بودند . همه این رهیافت ها مجدداً موفقیت های قابل توجه داشتند ، که این موفقیت ها ، درسیستم های مستقل یا به عنوان بخش هایی از سیستم های برمبنای دانش عمومی تر، تحقق یافته اند . همچنین واضح بود که سیستم های AI کلاسیک قادر نبودند ، با پدیدهٔ ظاهر شده درمحتوای « برداشت متعارف » به صورت رضایت بخش ارتباط برقرار کنند.

در حدود ۲۰ سال قبل ، ایده غالب این بود که این مساله به سبب این حقیقت می باشد که دانش برداشت متعارف خیلی پیچیده بود : بشر به سال ها تحصیلات به منظور تسلط بر دنیای هر روزه احتیاج داشت . از این نقطه نظر این فقط یک گام ساده برای شروع پروژهٔ CYC (D. Lenat در تگزاس) بـــا هـــدف جمـــع آوری و به ساختار در آوردن حدود ۱۰۰ میلیون واحد دانش (این تعداد با فرض اینکه انسان در روز ۱۰/۰۰۰ دانش در طول ۳۰ سال یاد بگیرد محاسبه شده است) بود .

نتایج نشـــان داده اند که آن روش درکل موفق نبود . (با اینکه شایستگی های زیادی داشت) . یکی از دلایل نتایج منفی ، مجموعه های بزرگ از واحد های دانش بود ، و کنترل کردن تعامل آنها با هم خیلی سخت است. درسال های گذشته یک ایده بسط

---

1 Black Box

داده شد که دراصل جدید نبود اما توجه سیستماتیک به آن نشد. آن رهیافت تقسیم کردن کار بین انسانهایی که ایده های خلاق و برداشت عمومی دارند ، بود واز یک طرف این افراد می توانند ایده های پایه تولید کنند و از طرف دیگر ماشین هایی که جستجوی بدون تعقل اما از نظر زمانی به صرفه می توانند انجام دهند ، توانایی مدیریت پایگاه داده های بزرگ را دارند و می توانند محاسبات وسیع انجام دهند. این کار به ایدهٔ یک سیستم دستیار منجر شده بود. درسیستم های دستیار گذشته ، راه حل هایی برای منظوری خاص جهت تکمیل کردن سیستم های برمبنای دانش ، اگر آنها قادر به دریافت نتایج دلخواه نبودند ، وجود داشتند . اغلب نقش بشر کمک کردن در هنگام ناتوان بودن ماشین در حل مساله بود .

یک رهیافت سیستماتیک به رسیدگی انواع مسائل زیر نیازمند است :

۱ـ توصیف اینکه چه کارهایی را بشر بهتر ازماشین می تواند ، انجام دهنـــد وبرعکس.

۲ـ استنتاج یک برنامه کاری که در اصل کاری سخت بین انسانها و عامل های نرم افزاری تقسیم می شود.

۳ـ توصیف اصولی همانند جنبه های تکنیکی درارتباط بین عامل های مختلف.

٤ـ یکپارچگی استدلال برداشت عمومی و استفاده از مفاهیم غیر رسمی با تفکر منطقی و محاسبات کامپیوتر . این به تحقیق درباره سیستم های نیمه رسمی منجر می شود.

به طور مختصرً دراینجا دو عوالم با هم روبرو می شوند : فرمالیسم ها۱ دربرابر مفاهیم غیر رسمی ، حقیقت منطقی دربرابراستدلال تخمینی ، از کل به جز رسیدن (استنتاج) در برابر استقرا و یادگیری . درتضاد با ایده های پروژه CYC ، هر شخصی سعی می کند بر پیچیدگی دنیا با پذیرش نا درستی ، تسلط پیدا کند . این راهی است که انسانها به آن اقدام کردند و سیستم های دستیار باید این توانایی ها را یکی کنند. درهمکاری انسان ـ ماشین ، به طور واضح انسانها شریک های برتری هستند. انسانها تصمیم گیری های نهایی را انجام می دهند و نسبت به عواقب کار ، مسئولیت دارند . یک روش

1 Formalism

جهت توصیف نقش تابعی ماشین ها ، درنظر گرفتن آنها به عنوان خدمتکارانی است که انسانها را با تولید دانش لازم به شکل مفید در همان لحظه ، پشتیبانی می کنند. ساختار سازمانی برای این منظور به عنوان مدیریت دانش که اهمیت افزاینده دارد ، شناخته شده است . مشکل دیگری که فوراً ایجاد می شود، کنترل بی دقتی است چون خطا های مطلق نمی توانند پذیرفته شوند . این به معنی داشتن محدوده های امن وقیاس مدیریت خطای عددی در استدلال سمبلیک است.

زمینه های مختلفی درگیر با این موضوع ها وجود دارند که برجسته ترین آنها استدلال بر مبنای مشابهت و استدلال فازی ، هستند . برای توسعه آینده AI درزمینه این ملاحظات مهم ترین گامها ، گسترش حوزهٔ تکنیک های AI سنتی نیست بلکه گشودن دید گاهی برای استدلال که توسط بشر انجام می شود ، می باشد . این یک رهیافت مرحله ای است که انسانها را به شکل سیستم های دستیارمتحد می کند ولی به ماشین اجازه می دهد با انواع روش های فکری انسان با مفاهیم غیر رسمی ارتباط داشته باشد. مطمئناً این مزیت بزرگی برای کاربرد های بسیاری خواهد بود ، مثل تجارت الکترونیکی، پزشکی ، تحصیلات و... . دراینجا یک پیشرفت هم در اساس و بنیاد و هم در سیستم های کاربردی نیاز داریم .

تحقیق و پیشرفت آینده در AI می تواند در ۳ عبارت زیر گنجانده شود :

- ایجاد اطلاعات
- خود مختاری
- وضعیت

۱-۷-۱ ایجاد اطلاعات

مقدار زیادی داده های جمع شده در پایگاه داده های موجود ، کاربرد خیلی کمی دارد، اگر فقط مکانیزم های بازیابی معمولی استفاده شوند . پرسش سئوالات درست و اتصال داده به روشی که با سئوالات مناسب باشد ، اطلاعات مربوط را ارائه می کند . کلیه متدها اکنون برای انجام این کار مثل انبار داده ۱(data warehouse) ، متدهای

1 Data Warehouse

استاتیک کلاسیک ، شبکه های عصبی و الگوریتم های یادگیری ماشین در دست است. متـــدهای تـــوسعه یافته اخیر حـتی می توانند به روشی کار آمد از عهده حجم زیاد داده برآیند . AI کمک اساسی به این تکنولوژی کرده است که به صورت تجاری در سالهای اخیر استفاده می شود و معمولاً داده کاوی نامیده می شود ، برای داده ی فرمت نشده متن کاوی ، وب کاوی یا کاوش تصویری نامیده می شود.

اطلاعات تولید شده به این روش از یک فضای خالی ، ایجاد نشده است ، بلکه به طور ضمنی در یک داده گنجانده شده است ، به یک مفهوم ، این اطلاعات یک خلاصه داده است . این خصوصیتی است که باید داده کاوی را که یک زمینه جذاب برای محققان AI است ، درکنار متدها ایجاد کند . اغلب به یادگیری ماشین به عنوان یک نمونه برای یادگیری انسان ، توجه می شود . این دیدگاه ، قطعاً غلط نیست ولی خیلی محدود است . از علوم شناخته شده در می یابیم که در یادگیری انسان ، آمار و ارقام هم نقش مهمی دارند ، پس می توانیم فرض کنیم که یادگیری ، یک پردازش پیچیده در جایگاهی است ، که متدهای مختلفی با هم در تعاملند . این مورد به وسیله آمیختن متدها در داده کاوی به خوبی منعکس شده است . اگر چه متدهای مختلف جداگانه استفاده می شوند ، ولی این متدها در حقیقت به هم گره نخورده اند و با یک پردازش ایجاد اطلاعات مجرد ، استفاده می شوند . این یک وظیفه تحقیق آینده است که این مسأله را حل می کند . یک راه حل ، فقط کارایی تکنولوژی داده کاوی موجود را بهبود نمی دهد ، بلکه یک جنبهٔ شناختی دارد . می توانیم فرضیه ای که بشر، دانش را از تجربیات بدست می آورند دنبال کنیم ، که با انتزاع مجموعه های بزرگی ازمثالها که بشر درطول زندگیش می بینند ، دانش خبره نامیده می شود . پردازش انتزاع درقسمت شناسایی دقیق و شناسایی بی دقت اجرا می شود و به وسیله دانش پایه و دانش یک موضوع تخصصی راهنمایی می شود . این پردازش ممکن است با تکنیک های داده کاوی پیشرفته شبیه سازی شود و بدین گونه ممکن است به طور اتوماتیک ، آگاهی خبره از داده تولید شود.

.

۱-۷-۲ خود مختاری

درعلم کامپیوتر ، سیستم های توزیع شده از چند سال پیش گسترش پیدا کرده اند و اثبات شده است ، که ابزار قدرتمندی برای افزایش کارایی حل مساله ، هستند به علاوه دامنه های کاربرد جدیدی برای تکنولوژی کامپیوتر می گشایند . همکاری AI در زمینه سیستم های گسترده فقط به منظور گسترش الگوریتم های جدید برای این سیستم ها نبود ، بلکه برای مجهز کردن اجزای سیستم های گسترده به درجاتی از خود مختاری بود .

تاکنون سیستم های کامپیوتری به عنوان ماشین های محاسبه و ذخیرهٔ حجم زیادی از داده و استفاده جهت پشتیبانی تصمیم گیری ، دیده شده اند . بشر برای پذیرش تصمیمات گرفته شده به وسیله سیستم های کامپیوتری مردد بود . به هر حال دربعضی دامنه های تکنیکی ، قبلاً کامپیوتر ها تأسیسات بزرگی را که شامل تصمیم گیری در بعضی موارد بود ، کنترل می کردند ، اگرچه بشر هنوز به عنوان ناظر و سرپرست عمل می کند . پس سیستم های کامپیوتری قبلاً خود مختار شدن را آغاز کرده اند . ولی این تنها شروع فرآیند بزرگ است . در سیستم های چند عاملی ، که کمک به AI به منظور گسترده شدن هستند ، خودمختاری یک موضوع کلیدی است. یک جزء از سیستم می تواند به عنوان یک عامل که فقط درجه ای ازخود مختاری را دارد ، دیده یا طرح ریزی شود ، درغیراینصورت به عنوان یک جزء منفعل نگریسته می شود . خود مختاری می تواند با تعدادی ویژگی مشخص شود . ابتدا ، این به معنی توانایی انتخاب چند عمل ازمجموعه ای از اعمال ممکن در موقعیت معین ، شامل تصمیم بین غیرفعال بودن یا موقع نیاز فعال شدن ، می باشد . این توانایی Proactivity نامیده می شود . ویژگی مهم دیگر توانایی ابقا کردن اهداف است . نه تنها لازم است که یک عامل ، اهدافی داشته باشد که برنامه ریزی و فعالیتش را راهنمایی کند بلکه اینکه عامل بتواند این اهداف را تغییر دهد ، رها کند یا اهداف جدیدی اتخاذ کند . ویژگی سوم توانایی ارتباط فعال و همکاری است.کلمه" active " (فعال) در اینجا مهم است چون تبادل پیام ها به منظور فراخوانی های تابعی بین اجزا منفعل معمولا ارتباط نامیده می شود ،

اما اینجا این معنی را نمی دهد . ارتباط و همکاری بین عامل ها یک رفتار proactive است ، یک عامل می تواند ارتباط را شروع کند یا برای همکاری با بقیه هر موقع که باور دارد لازم است ، جستجو کند. ویژگی خود مختاری یک رابطه جدید بین عامل ها ایجاد می کند چه آنها به عنوان ربات و یا به عنوان سیستم های نرم افزاری و انسانهایی که خصوصیت مشارکت دارند ، درک شوند . عامل های خود مختار نمی توانند به عنوان ماشین های صرف که در هنگام نیاز شروع می شوند ، کارشان را انجام می دهند و تمام می شوند ، در نظر گرفته شوند . ممکن است شخصی به این عوامل به دید خدمتکاران نگاه کند ، ولی خدمتکاران خواسته و توانایی هایی ادراکی پیچیده خودشان را دارند ، و این موارد باید به عامل ها واگذار شوند . ما باید انتظار زندگی در یک جامعه پیچیده تر در آینده را نسبت به اینکه امروز ما با عامل ها به عنوان نوعی از موجودیت های زنده کار می کنیم ، داشته باشیم . این عوامل در زندگی روز مره ما حاضر خواهند شد ، مثل دستیار های شخصی در سیستم های کامپیوتری مان ، به عنوان راننده در ماشینمان ، یا مثل سرایدار خانه مان . محققان AI باید با جامعه شناسان در باره مسائلی که ممکن است از این دید ایجاد شوند ، در ارتباط باشند .

۱-۷-۳ جایگزینی

در یک دید کلی ، همه سیستم های کامپیوتری در محلی قرار گرفته اند ، اما سیستم های سنتی موجود در موقعیت های خوش- تعریف خیلی محدود شده که کاملاً به وسیله بشر معین شده ، وجود دارند . موقعیت فقط یک موضوع مهم تحقیق درباره ظهور عوامل در سیستم چند عاملی و قابل تطبیق با رباتهای متحرک ، مناسب بوده . بدیهی است که موقعیت به پیامد خود مختاری مربوط می شود . فقط سیستم های خود مختار باید خودشان در جهت وجایگاه مناسب قرار گیرند و در موقعیتها عمل کنند. در اینجا موقعیت به معنی یک مجموعه تاثیرها از محیط است که حداقل تقریبا غیر قابل پیش بینی اند اما اهمیت زیادی برای سیستم دارند به طوری که مجبور به نشان دادن واکنش در یک روش مناسب است . یک سیستم جایگزین مجبور است ۲ مسئله اصلی

را حل کند ، برای مثال چگونه موقعیت را حس کند و چگونه واکنش مناسب را انتخاب کند . دریافت یک موقعیت ممکن است یک کار ساده برای یک عامل نرم افزاری صرف در یک محیط خوش ساخت باشد . به هر حال اگر به فعالیت عوامل در مثلاً اینترنت فکر کنیم ، کارها خیلی پیچیده تر می شوند چون این محیط به شدت غیر ساخت یافته و دینامیک است . حتی حس کردن با رباتها پیچیده تر است . در اینجا سیگنالهای فیزیکی اولیه ، تبدیل شده به داده هستند ، وظیفه ای که آنها ممکن است به فیزیک دانان و مهندسان الکترونیک محول کنند . ولی در مرحله بعد ، سیگنال ها از منابع مختلف ، باید با هم قرار داده شوند تا یک توصیف از وضعیت که سیستم جایگزین را قادر به واکنش مناسب کند ، ایجاد کنند . این وظیفه ترکیب حس گر نامیده می شود و در جایی که متدهای AI وجود دارند ، هستند . از علوم شناخته شده در می یابیم که فهم یک موقعیت یک فرایند خیلی پیچیده است که نیاز به اطلاع پایه ای و فعالیتهای عصبی زیادی دارند . مغز موقعیت را از ورودی ها متفاوت ایجاد می کند . هنوز درباره این فرایند چیز کمی می دانیم و برای استدلالهای واضح ، دوباره سازی آن سخت است . شبیه سازی های به وسیله متد های AI ، ممکن است در به دست آوردن بینشی نسبت به آن کمک کنند . اگر شبیه سازی ممکن باشد ، می تواند با سیستم های جایگزین مصنوعی جهت توسعه توصیف یک موقعیت برای اهدافشان استفاده شود.

برای سیستمهای جایگزین هدف اصلی از ایجاد توصیفهای جایگزینی ، استفاده آنها برای فعالیتهای خودشان است . وظایف اصلی موقعیت یابی خود و جهت یابی و وظیفه مشتق شدهٔ فعالیت می تواند فقط روی اساس توصیفهای جایگزینی انجام شود . از دیگر سو به این معنی که ایجاد سیستم توصیف جایگزینی ، همیشه برای پشتیبانی سیستم جایگزین در تکمیل وظایفش هدفدار شده است ؛ و این در خودش یک پایان نیست . ضوابطی نظیر کمال و پایداری اولویت ندارند ؛ بلکه یک توصیف اگر به انتخاب اعمال مناسب کمک کند ، احتیاجات یک سیستم جایگزین را ارضاء می کند . بعضی انواع زمینه های دانش جایگزین ، نیازمند در بر داشتن اطلاع پایه ای وسیع و میزان قابل توجهی آگاهی مربوط ، که لازم نیست با یکدیگر و یا با اطلاع پایه ای

سازگاری داشته باشند ، می باشد . دانش در این مقادیر قابل توجه ممکن است حتی در اشکال مختلفی ارائه شود . متدهای انتخاب میزان اطلاع صحیح ، برای ترکیب آنها و برای تغییر شکل دادن دانش از یک شکل به شکل دیگر باید توسعه داده شوند . به طور آشکار ، سیستمهای جایگزین باید قابلیت برنامه ریزی جهت انتخاب ترتیب فعالیت ها داشته باشند . همچنین باید توانایی فراگیری را هم داشته باشند ، چون همانطوری که در بالا ذکر شد ، تاثیراتی که یک وضعیت را تشکیل می دهند ، تقریباً غیر قابل پیش بینی اند . اگرچه ، موقعیت های جدید کاملاً با هم تفاوت ندارند ، ولی در همه ی محیط های معقول ، شباهت هایی بین آنها وجود دارد مثلاً در یاد گیری ، تشخیص و طبقه بندی موارد مشابه . یاد گیری می تواند رفتار سیستم جایگزین را بهبود دهد . برای خلاصه : ایجاد اطلاعات ، خود مختاری و وضعیت می تواند به عنوان مرکز توجه در تحقیقات و توسعه AI در آینده در نظر گرفته شوند . به منظور مطرح شدن این چالش ها ، متدهای ساده زیادی باید برای سیستم های بزرگتر جمع آوری شده باشند . پس جهت کلی تحقیق و توسعه AI می تواند با گسترش سیستم های پیچیده ای که متد های مختلفی را یکی می کند و این ۳ نیازمندی را تکمیل می کند ، مشخص شود .

فصل دوم

منطق نمادین

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

منطق

گزاره ها

شکل های نرمال در منطق گزاره ای

نتایج منطقی

اصل تفکیک پذیری

جبر گزاره ای

فرمولهای خوش – ساخت (WFFS)

فرم عبارتی

قوانین استنتاج

یکسان سازی

تفکیک پذیری

۲ – ۱ مقدمه

یکی از توانای های هوش انسان توانایی ایجاد استنتاج های منطقی است .
آگاهی از این حقیقت علاقه به اثبات تئوری های هوش ماشین را افزایش داد.
پیشرفت در بهبود تکنیک های اثبات تئوری،با پیشرفت در به کاربردن این تکنیک ها در AIهمزمان بود . تکنیک ها و تئوری های فوق برای پاسخ سئوال استنتاجی و بعداً در تکنیک های حل مسأله ترکیب برنامه ، آنالیز برنامه و مسائل بسیار دیگر به کار گرفته شدند.
موقعی که ما از اثبات تئوری های هوش ماشین حرف می زنیم با« منطق » درگیر هستیم وقتی منطق مربوط به AI است به آن « منطق نمادین » گفته می شود. از زوایای مختلفی می توانیم منطق نمادین را مطالعه کنیم اما به طور سنتی همواره این منطق با گرایش های ریاضی و فلسفی مطالعه شده است .
برای حل مسائل خیلی هوشمندانه از منطق نمادین استفاده میکنیم .برای نمایش مشکلات و یافتن راه حل های آنها می توانیم از منطق نمادین استفاده کنیم .حال ببینیم چگونه منطق نمادین می تواند برای نمایش مسائل استفاده شود.برای درک بهتر مثال های زیر را در نظر میگیریم :
مثال ۱ : فرض کنید حقایق زیر را داریم :

(۱) F : اگر هوا گرم ومرطوب باشد آنگاه باران خواهد بارید .
(۲) F : اگر هوا مرطوب باشد آنگاه گرم است .
(۳) F : الآن هوا مرطوب است .

سئوال این است که آیا باران خواهد بارید ؟
حقایق بالا به فارسی نوشته شده اند. ما باید از نماد ها برای ارائه آنها استفاده کنیم .P و Q و R را به ترتیب « گرم » ، « مرطوب » ، « باران خواهد بارید » درنظر بگیرید.ما به نشانه های منطقی نیز نیاز داریم دراین مثال از « $\wedge$ » برای نمایش « and» و « $\rightarrow$ »

.

برای نمایش « نتیجه می دهد » استفاده می کنیم .آنگاه ۳ حقیقت بالا به صورت زیر نمایش داده می شود .

$$F(1) : P \wedge Q \rightarrow R$$
$$F(2) : Q \rightarrow P$$
$$F(۳) : Q$$

پس ما این جملات را به فرمول های منطقی ترجمه کرده ایم . خواهیم دید وقتی (۱) F و (۲)F و (۳) F درست باشند فرمول R : (٤) F درست است .

پس نتیجه می گیریم (٤)F به طور منطقی از (۱) F و (۲) F و (۳) F پیروی می کند.

و(٤) F یعنی « باران خواهد بارید » .

مثال ۲ ما حقایق زیر راداریم .

(۱) F : کنفوسیوس یک انسان هست .

(۲) F: هر انسان فانی است .

چه چیزی قابل استنتاج است ؟

برای نمایش (۱) F و (۲) Fبه مفهوم جدیدی به نام گزاره نیازمندیم.

P(x) و q(x) را به ترتیب « x یک انسان است » و « x فانی است » در نظر می گیریم.

همچنین (Vx) « برای همه x ها » استفاده می گردد .

آنگاه حقایق بالا به صورت زیر نمایش داده می شوند :

$$F(۱) : P(Confucius)$$
$$F(2) : (\forall x)(Px \rightarrow Q)(x))$$

بعداً خواهیم دید که به طور منطقی می توانیم از (۱) F و (۲) F (Confucius)Q: (۳) F را که به معنی کنفوسیوس فانی است نتیجه بگیریم. در ۲ مثال بالا ضرورتاً باید ثابت کنیم که یک فرمول منطقاً از فرمول های دیگر منتج می شود. ما عبارتی را که یک

فرمول منطقاً از فرمولها نتیجه می شود را یک تئوری می نامیم .
نمایش و اثبات یک تئوری و استنتاج یک فرمـــول از فرمول های دیگر ، اثبـــات قضیـــه نامیده می شود (Proof) . اثبات تئوری های و نظریه های هوش ماشین با در نظـــر گرفتـــن متد هـــای این تئوری مشکـــل است .

۲- ۲ منطق

دیدگاه های مختلفی درباره تعریف « منطق » وجود دارند ولی در این متن منظور از منطق هنر استدلال صحیح می باشد .چرا "صحیح"؟
فرض کنید مجموعه ای از عبارات به ما داده شده است. این موضوعات عبارتنداز:
(a) امکان به دست آوردن همه حقایق صحیح از مجموعه ای عبارات داده شده به واسطه استدلال وجود دارد ؟ (به عنوان استنتاج هم شناخته شده) این مورد به تمامیت استدلال اشاره می کند.

(b) می توانیم یک حقیقت غلط از طریق استدلال مشتق کنیم ؟ این مورد به صحت استدلال اشاره می کند.
از اینرو مـــوقعی که استدلال ما دو ویژگی بالا را در همه عبارات داده شده دارا باشد نتیجه می گیریم که استدلال درست است . به این هنر استدلال درست ، منطق می گویند.
ولـــی اینجا در مـورد منطق نمادین صحبت می کنیم . هدف این منطق نه تنها نمادین (سمبلیک) کردن استدلال های ریاضی بلکه نمادین کردن این استدلالها درزندگی روزانه است .دراین فصل ما با ساده ترین منطق نمادین به نام منطق گزاره ای سر و کارداریم .
چه چیزی به وسیله منطق گزاره ای استنتاج می شود؟ گفتیم که هدف منطق نمادین ، نمادین کردن استدلال است . در منطق گزاره ای ، این نمادین کردن یا نشان گذاری با استفاده از عبارات اخباری انجام میشود چه این گزاره ها درست و چه غلط باشند .

بنابراین منطق گزاره ای به روش استدلالی ای که در یک مجموعه عبارات اخباری داده شده اند ، اشاره می کند. این عبارت اخباری،به عنوان یک گزاره شناخته شده است.
پس اکنون دو نکته عمده قابل توجه داریم . اولین نکته مطالعه موضوعات در رابطه بین گزاره ها ودومین مربوط به منطق است . اجازه دهید با گزاره ها شروع کنیم.

۲-۳ گزاره ها

۲-۳-۱ عبارت

در منطق گزاره ای ما به جملات اخباری که می توانند درست یا غلط باشند ، نه هر دو توجه می کنیم . هر جمله خبری یک گزاره یا عبارت نامیده می شود.

تعریف ۱ :

یک « گزاره » جمله خبری ای است که یا درست باشد یا غلط اما نه هر دو
مثالهایی از گزاره ها عبارتند از : " آسمان آبی است ". درست یا غلط بودن نسبت داده شده به یک گزاره « ارزش درستی » گزاره نامیده می شود.
معمولاً ما درست را با T و غلط را با F نشان می دهیم .به علاوه ما باید از سمبل با حروف بزرگ یا رشته ای از سمبلهای با حرف بزرگ را برای مشخص کردن یک گزاره استفاده کنیم . برای مثال ممکن است ما گزاره بالا را به صورت زیر مشخص کنیم :

P= آسمان آبی است

تعریف ۲ :

سمبل ها مثل P , Q , R که برای مشخص کردن گزاره ها استفاده می شوند، « فرمول های اتمی » یا « اتم ها » نامیده می شوند.
ازگزاره ها می توانیم به وسیله « عطف های منطقی » ، « گزاره های مرکب » بسازیم .
مثال هایی ازگزاره های مرکب عبارتند از :« برف سفید است وآسمان روشن است » و «

اگر Gopal در خانه نیست آنگاه Gita در خانه است » عطف های منطقی در دو گزاره مرکب بالا « and » و « if ..... then » است. در منطق گزاره ای ما باید از پنج عطف منطقی استفاده کنیم : (not) و(or)و(and) و (if ..... then) و (if and only if) . این پنج عطف منطقی می توانند برای ساخت گزاره های مرکب استفاده شوند. به طورکلی تراین ۵ عطف منطقی می توانند برای ساخت گزاره های مرکب پیچیده تر از گزاره های مرکب با اعمال آنها به صورت تکراری استفاده شوند . اجازه دهید مثالی را در نظر بگیریم وقتی که « رطوبت بالا است » را با P و «دما بالا است » را با Q و « شخصی احساس راحتی می کند » را با C نشان می دهیم . آنگاه جمله « اگر رطوبت و دما بالا باشد آنگاه شخصی احساس ناراحتی می کند » می تواند با $((P \wedge Q) \rightarrow (\sim C))$ نمایش داده شود.

بنابراین می بینیم که یک گزاره مرکب می تواند یک ایده نسبتاً پیچیده را بیان کند.

در منطق گزاره ای یک عبارت مثل P که نمایش دهنده یک گزاره است ، یا یک گزاره پیچیده مثل $((P \wedge Q) \rightarrow (\sim C))$ ، یک فرمول خوش ـ ساخت نامیده می شود.

تعریف ۳ :

« فرمولها » در منطق گزاره ای به صورت بازگشتی تعریف شده اند مانند زیر :

۱ـ یک اتم یک فرمول است .

۲ـ اگر G یک فرمول باشد آنگاه (G -) یک فرمول است .

۳ـ اگر G و H فرمول باشند آنگاه $(G \wedge H)$و $(G \vee H)$و $(G \rightarrow H)$ و $(G \leftrightarrow H)$ فرمول هستند.

٤ـ همه فرمول ها با به کاربردن قوانین بالا تولید می شوند .

فرض می کنیم G و H ۲ فرمول باشند. آنگاه ارزش درستی فرمول های تولید شده با استفاده از عطف های منطقی روی H و G طبق روش زیر به ارزش های درستی

.

GوH مربوط می شوند.

۱ـ G ~ وقتی G غلط باشد ، درست است و برعکس . G ~ نقیض G است .

۲ـ $(G \wedge H)$ وقتی درست است که G و H هر دو درست باشند درغیر این صورت $(G \wedge H)$ غلط است . $(G \wedge H)$ ترکیب عطفی G و H است.

۳ـ $(G \vee H)$ وقتی درست است که حداقل یکی از G و H درست باشد درغیر اینصورت غلط است . $(G \vee H)$ تفکیک G و H نامیده می شود.

٤ـ $(G \rightarrow H)$ غلط اســت اگر G درســت و H غلط باشد . در غیر اینصورت درست است.

$(G \rightarrow H)$ به صـورت « اگر G آنگاه H » یا « G نتیجه می دهد H را » خوانده می شود.

5ـ $(G \leftrightarrow H)$ درست است هرگاه G و H ارزش های درستی یکسان داشته باشند.

در غیر اینصورت غلط است .

رابطه های بالاهمانطور که در جدول ۱ . ۲ نشان داده شده اند می توانند به سادگی ارائه شوند.

جدول ۲,۱

G	H	G	$(G \wedge H)$	$(G \vee H)$	(G→H)	(G↔H)
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

۲-۳-۲ تفسیر فرمول ها

فرض کنید P و Q دو جمله اتمی هستند و به ترتیب دارای ارزش های درستی درست و غلط می باشند.

با توجه به دومین ردیف از جدول بالا با جایگزینی P و Q به ترتیب به جای G و H

در می یابیم که ارزش های درستی (~P) . $(P \wedge Q)$ ، $(P \vee Q)$ ، $(P \to Q)$ و $(P \leftrightarrow Q)$ به ترتیب غلط ، غلط ، درست ، غلط ، غلط هستند . مشابهاً ارزش درستی هر فرمولی می تواند برحسب ارزش های درستی جملات اتمی ارزیابی شوند.

مثال ۳. فرمول $G = (P \sim Q) \to \big(R \leftrightarrow (s)\big)$ را در نظر بگیرید .

اجزا این فرمول S, R , Q , P هستند . فرض کنیدارزش های درستی S, R , Q , P به ترتیب T, F , T و T هستند.

آنگاه $(P \wedge Q)$ غلط است زمانی که Q غلط باشد (~S) غلـط است زمانی که S درست باشد. $\big(R \leftrightarrow (\sim s)\big)$ غلط است هرگاه R درست و(~S) غلط باشد و$\big(R \leftrightarrow (\sim s)\big)$ $(P \wedge Q) \to$ درست است هرگاه$(P \wedge Q)$ و$\big(R \leftrightarrow (\sim s)\big)$ هر دو غلط باشد.

بنابراین فرمول G درست است اگرS, R , Q , P به ترتیب T, F , T و T قلمداد شوند ، این تفسیر فرمول G نامیده خواهد شد.

چون هر کدام از S, R , Q , P می توانند T یا F باشند ، ۱٦ تفسیر ازفرمول G وجود دارد .

جدولی که ارزش های درستی G را ، برای همه انتساب های ممکن این ارزش ها به اجزا ئی(اتم ها ئی) که در G هستند ، نشان می دهد ، جدول درستی G نامیده می شود.

تعریف ٤ :

فرمول گزاره ای G داده شده ، $A_1 , A_2 \ldots\ldots A_n$ را اجزائی (اتم هایی) که در فرمول G هستند در نظر بگیرید. آنگاه یک تفسیراز G ، یک انتساب ارزش درستی به A_1 تا A_n می باشد که درآن به هر A_i ، T یا F نسبت داده شده است.

تعریف ٥ :

گفته می شود یک فرمول G « تحت یک تفسیر درست است » اگر وفقط اگر G در تفسیر به صورت T ارزیابی شود درغیر اینصورت گفته می شود G « در این تفسیر غلط است » .

اگر n جز(اتم) مجزا در یک فرمول وجود داشته باشند آنگاه ۲ به توان n تفسیر مجزا برای فرمول وجود خواهد داشت .

بعضی اوقات اگر A_1 تا A_n همه اجزا(اتمهای) موجود در یک فرمول باشند ممکن است نشان دادن یـــک تفسیـر با یک مجموعه m_1 تا m_n که m_i همان A_i یا $\sim A_i$ است ، راحت تر باشد . برای مثال مجموعهٔ P ، Q~ ، R~ و S تفسیری ارائه می کند که درآن به S, R , Q , P به ترتیب F, F, T و T نسبت داده شده است .

به این معنی که اگر یک اتم A در یک مجموعه ای باشد که تفسیری ارائه می کند آنگاه به A ، T اختصاص داده می شود.

درصورتیکه اگر نقیض A در مجموعه باشد آنگاه به A ، F نسبت داده می شود .

۲-۳-۳ درستی و تناقض فرمول ها

فرمول $G = ((P \to Q)P) \to Q$ را در نظر بگیرید.

اجزای (اتم های) این فرمول P و Q هستند فرمول G تفسیرهای ٤= (۲) Sqr را دارد ، ارزش های درستی G تحت همه ٤ تفسیرش در جدول ۲ـ ۲ داده شده است

جدول۲-۲.

P	Q	(P→Q)	(P→Q)P	((P→Q))→ Q
T	T	T	T	T

T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

می بینیم فرمول G تحت همه تفسیرها یش درست است . این فرمول، یک فرمول معتبر یا یک درست نما نامیده خواهد شد.

حالا فرمول $G = (P \to Q) \wedge (P \to \sim Q)$ را در نظر بگیرید : جدول درستی برای G به صورت داده شده درجدول ۳. ۲ است .

جدول۲-۳

P	Q	~Q	(P→ Q)	(P^~Q)	(P→Q)(P^~Q)
T	T	F	T	F	F
T	F	T	F	T	F
F	T	F	T	F	F
F	F	T	T	F	F

می بینیـم G تحت همه تفاسیرش غلط است . این فرمول متناقص یا یک تناقص نامیده می شود.

تعریف ۱:

یک فرمول« معتبر » گفته می شود اگر وفقط اگر تحت تمام تفاسیرش درست باشد.

یک فرمول « غیر معتبر » گفته می شود اگر و فقط اگر معتبر نباشد.

تعریف ۲ :

یک فرمول « متناقض » نامیده می شود اگر و فقط اگر تحت تمام تفاسیرش غلط باشد.

یک فرمول « نا متناقض » نامیده می شود ا گر و فقط اگر متناقض نباشد.

پس :

۱ . یک فرمول معتبر است اگر نقیض آن متناقض باشد.

۲ . یک فرمول متناقض است اگرنقیض آن معتبر باشد .

۳. یک فرمول نامعتبر است اگر حداقل در یک تفسیر غلط باشد.
٤. یک فرمول نامتناقص است اگر تحت حداقل یک تفسیر درست باشد.
٥ . اگر یک فرمول معتبر باشد آنگاه آن نامتناقص است ولی عکسش صادق نیست.
٦ . اگر یک فرمول متناقض باشد آنگاه غیر معتبر است ولی عکسش صدق نمی کند .

٤-۲ شکل های نرمال در منطق گزاره ای

موقعی که با فرمولها سرو کار داریم ، با ارائه ونمایش این فرمولها ارتباط داریم .
پس در یک برداشت کلی کسی میتواند بگوید که اگر همه فرمولها به طور مشابه ارائه شوند یا به دیگر دارای فرم یکسا نی باشند ، استنتاج روی یک مجموعه گزاره آسانتر خواهد بود . پس لازم است یک فرمول را از شکلی به شکل دیگر تبدیل کنیم مخصوصاً « فرم نرمال ». برای مثال لازم است مجموعه داده شده از فرمولها را به فرم استاندارد کلی تبدیل کنیم.
این تبدیل با جایگزینی یک فرمول در فرم داده شده ، با فرمول معادل آن انجام می شود واین فرآیند تا زمانی که فرم دلخواه حاصل شود تکرار می گردد.

تعریف ۱:
دو فرمول F و G معادل گفته می شوند (F=G) ، اگر و فقط اگر ارزش های درستی F و G تحت هر تفسیری از این دو (F و G) یکسان باشند.

تعریف ۲ :
یک « رشته » یک اتم یا نقیض اتم است . در منطق گزاره ای یکی از ۲ نوع مدل ارائه یا فرم های فرمول ها را استفاده می کنیم . آنها در زیر آمده ا ند.

تعریف ۳ :
یک فرمول F به فرم نرمال ترکیب عطفی است اگر و فقط اگر F فرم

F=F1^…^Fn,n>=1
داشته باشد ، که هر F_1 تا F_N یک انفصال رشته ها باشد.
موقعی که می گوئیم F_1 یک انفصال از رشته ها است یعنی F_1 به فرم

$$F_1 = A_1 \vee ... \vee A_n$$

است جائی که A_1 تا A_n ، لیترال هستند .

تعریف ٤ :

یک فرمول F به فرم نرمال فصلی است اگر وفقط اگر F فرم

$$F = F_1 \vee ..... \vee F_n, n>=1$$

داشته باشد وقتی هر کدام F_1 تا F_n یک ترکیب عطفی از لیترال ها باشند. وقتـــــی می گوئیم F_1 یک ترکیب عطفی از لیترال ها است یعنی F_1 به فرم

$$F_1 = A_1 \wedge ....A_n$$

است که A_1 تا A_n لیترال هستند.

به منظور انجام تبدیل فرمول ها به فرم دلخواه ، به یک منبع مناسب فرمول های معادل نیاز مندیم ' $' را برای مشخص کردن فرمولی که همیشه درست است و ' # ' را برای مشخص کردن فرمولی که همیشه غلط است در نظر بگیرید . جملاتی که در زیر آمده بعضی زوج های مفید از فرمول های معادل است که قوانین نامیده می شوند. در این قسمت F, G و H فرمولها هستند.

1. $F \leftrightarrow G = (F \rightarrow G)\ (G \rightarrow F)$
2. $F \rightarrow G = F \vee G$

قوانین جابه جایی پذیر

3. (a) $F \vee G = G \vee F$, (b) $F \wedge G \wedge F$

قوانین شرکت پذیری

4. (a) $(F \vee G) \vee H = F \vee (G \vee H)$; (b) $(F \wedge G) \wedge H = F \wedge (G \wedge H)$

قوانین توزیعی

5. (a) $F \vee (G \wedge H) = (F \vee G) \wedge (F \vee H)$; (b) $F \wedge (G \vee H) = (F \wedge G) \vee (F \wedge H)$

قوانین دمرگان

6. (a) $\sim(F \vee G) = \sim F \wedge \sim G$; (b) $\sim(F \wedge G) = \sim F \vee \sim G$

قوانین دیگر

7. (a) $F \vee \# = F$; (b) $F \wedge \$ = F$
8. (a) $F \vee \$ = \$$; (b) $F \wedge \# = \#$
9. (a) $F \vee \sim F = \$$; (b) $F \wedge \sim F = \#$
10. $\sim(\sim F) = F$

۲-۵ نتایج منطقی

در ریاضیات مثل زندگی روزمره ، باید تصمیم بگیریم که آیا یک جمله ازبعضی عبارات دیگر نتیجه می شود یا نه. این به مفهوم نتیجه منطقی منجر می شود.

تعریف :

فرمول های F_1 تا F_n و فرمول Gداده شده اند،G یک نتیجه منطقی F_1 تا F_n است اگر و فقط اگر برای هر تفسیر I در جایی F_n ..... $F_1 \wedge$ درست باشند ، G هم درست است.

F_1 تا F_n « بدیهیات » G نامیده می شوند .

قضیه ۱ :

(قضیه استنتاج) : فرمول های F_1 تا F_n وفرمول G داده شده ، G نتیجه منطقی از F_1 تا F_n گفته می شود اگرو فقط اگر فرمول $(F_1 \wedge F_2 \wedge .... \wedge Fn) \rightarrow G)$ معتبر باشد.

اثبات : فرض کنید G نتیجه منطقی از F_1 تا F_n باشد . I را یک تفسیر دلخواه می گیریم .

اگر F_1 تا F_n در I درست باشند آنگاه با تعریف نتیجه منطقی ، G نیز در I درست است .

بنابراین فرمول $(F_1 \wedge ....F_n) \rightarrow G)$ در I درست است از طرف دیگر اگر $F_۱$ تا F_n در I غلط باشند آنگاه $((F_1 \wedge F_2....F_n) \rightarrow G)$ تحت هر تفسیری درست است . یعنی $((F_1 \wedge F_2....F_n) \rightarrow G)$ یک فرمول معتبر است .
فرض کنید $((F_1 \wedge F_2....F_n) \rightarrow G)$ یک فرمول معتبر باشد. برای هر تفسیر I اگر $F_1 \wedge ....F_n$ در I درست باشد ، G باید در I درست باشد. پس G یک نتیجهٔ منطقی از $F_1 \wedge ....F_n$ است .

قضیه ۲ :

فرمول های $F_۱$ تا F_n و G داده شده . G یک نتیجه منطقی از $((F_1 \wedge F_2....F_n) \rightarrow G)$نامیده می شود اگر وفقط اگر فرمول $(F_1 \wedge F_2 \wedge .... \wedge F_n \neg G)$ متناقض باشد.

اثبات : بنا به قضیه ۱ ،G یک نتیجه منطقی از $F_۱$ تا F_n است اگر وفقط اگر فرمول $(F_1 \wedge ....F_n) \rightarrow G)$ معتبر باشد. از این رو G نتیجه منطقی از $F_۱$ تا F_n است اگر و فقط اگر نقیض فرمول $(F_1 \wedge F_2 \wedge ... \wedge F_n) \rightarrow G)$ متناقض باشد پس :

$$
\begin{aligned}
\sim((F1 \wedge F2 \wedge \ldots \wedge Fn) \rightarrow G) &= (\sim(F1 \wedge F2 \wedge \ldots \wedge Fn) \vee G) \\
&= (\sim((F1 \wedge F2 \wedge \ldots \wedge Fn)) \wedge \sim G) \\
&= (F1 \wedge F2 \wedge \ldots \wedge Fn) \wedge \sim G \\
&= F1 \wedge F2 \wedge \ldots \wedge Fn \wedge \sim G
\end{aligned}
$$

نتیجه می گیریم قضیه ۲ درست است.

قضایای ۱ و ۲ خیلی مهمند ،این قضایا نشان می دهند که اثبات یک فرمول خاص ، یک نتیجه منطقی ازیک مجموعه متناهی از فرمولها است ، این مجموعه متناهی با

.

اثبات یک فرمول وابسته معین ، که معتبر یا متناقض است ، معادل می باشد.
اگر G نتیجه منطقی F_1 تا F_n باشد ، فرمول $(F_1 \wedge F_2 \wedge ... \wedge F_n) \rightarrow G)$ یک قضیه نامیده می شود و G نتیجه قضیه نامیده می شود.
متد اثبات بالا به عنوان یک متد استنتاج طبیعی شناخته شده است . متد دیگر ، اصل تفکیک پذیری است که در زیر مطرح شده است .

۲-۶ اصل تفکیک پذیری

برای هر دو عبارت C_1 و C_2 اگر یک لیترال L_1 در C_1 وجود داشته باشد که مکمل یک لیترال L_2 در C_2 باشد آنگاه L_1 و L_2 را به ترتیب از C_1 و C_2 حذف کنید و ترکیب فصلی از عبارات باقیمانده را بسازید . عبارت ساخته شده ، حل مساله C_1 و C_2 است .
مثال ٤ عبارات زیر را در نظر بگیرید :

$$C_1 = P \vee R$$
$$C_2 = \neg P \vee Q$$

عبارت C_1 یک لیترال P دارد که مکمل P در C_2 است .
بنا براین با حذفP از C_1 وP ازC_2 به ترتیب و ایجاد ترکیب فصلی از عبارات باقیمانده R و Q ، جواب $R \vee Q$ را بدست می آوریم .

۲-۷ جبر گزاره ای

جبر گزاره ای یا منطق گزاره ای مرتبه اول ($FOPL$) یکی ازقدیمی ترین و مهمترین طرح ونقشه های (شماهای) ارائه دانش استفاده شده در AI است .
می توانیم بگوییم که منطق گزاره ای یک زبان انسان گرای سطح بالا برای توصیف مسائل و متدهای حل مساله می باشد . این منطق به وسیله منطق دانان به عنوان وسیله ای برای استدلال رسمی ومسائل مقدماتی در زمینه های ریاضیات ، توسعه داده شد.

آشنایی با $FOPL$ برای دانشجوی AIجهت اهداف مختلف ، مهم است.

- آن فقط رهیافت رسمی به استدلال را که یک اساس تئوری درست دارد پیشنهاد می کند.
- ساختار منطق گزاره ای به اندازه کافی برای مجوز ارائه دقیق عبارات زبان طبیعی به طور معقولانه انعطاف پذیر است.
- منطق گزاره ای در بسیاری از جاها به عنوان یکی از مفید ترین متدهای نمایش به وسیلهٔ ایجاد کنندگان(کارکنان) در زمینه AI پذیرفته شده است.

منطق یک متد رسمی استدلال است. مفاهیم زیادی که می توانند به صورت شفاهی بیان شوند، قابلیت ترجمه به ارائه های نمادین را دارند که به دقت به معنی این مفاهیم نزدیک هستند و این ساختارهای نمادین می توانند در برنامه ها به منظور استنتاج حقایق مختلف برای انجام یک شکل از استدلال خودکار دستکاری شوند.

۲-۷-۱ نحو منطق گزاره ای

اجزای اساسی منطق گزاره ای عبارتند از:

۱. عطف ها :

~ یا ¬ نقیض را نشان می دهد .

& یا ∧ ترکیب عطفی یا AND را نشان می دهد

| یا ∨ ترکیب فصلی یا OR را نشان می دهد.

→ مستلزم بودن (if ..... then) را نشان می دهد.

↔ هم ارزی یا اگروفقط اگررا نشان می دهد.

۲ . سورها :

∃ : سور وجودی (وجود دارد)

∀ : سور عمومی (برای همه مقادیر)

۳ . ثابت ها :

عبارات مقدار ثابت متعلق به دامنه داده شده هستند . معمولاً با حروفی مثل ابتدای الفبای انگلیسی و اعداد مشخص می شوند. مثل a،D , b ، ۱۰ ، ۲/۵

٤ . متغیر ها :

عباراتی که می توانند در یک دامنه داده شده ارزش های مختلفی بگیرند .

این عبارات معمولاً با کلمه یا آخرین حروف های الفبای انگلیسی مثل z, y , x مشخص می شوند.

۵ . نشانه های کمکی :

() و [ ] و { } برای نقطه گذاری استفاده می شوند.

٦ . توابع :

سمبل های توابع تعریف شده روی یک دامنه n(n>0) عنصر را به یک عنصر از دامنه نگاشت می دهند.

دراینجا n رتبه یادرجه تابع نامیده می شود.حروفf وg و h و کلماتی مثل age-of( ) و Cause - of ( ) توابع را مشخص می کنند.

یک تابع درجه n به صورت f (t_1 , t_2 , ... t_n) نوشته می شود جایی که t ، عبارات تعریف شده روی دامنه هستند . یک تابع درجه صفر ، یک ثابت است.

Term: متغیرهای ثابت و توابع term گفته می شوند.

۷ . گزاره ها رابطه ها یا نگاشت :

گزاره ها ارتباط یا نگاشت تابعی از عناصر دامنه به ارزش های درستی صحیح یا غلط را مشخص می کنند.

حروف و کلمات در وسط الفبا مثل P , Q ,R , EQUAL, ... برای مشخص کردن گزاره ها استفاده می شوند. گزاره ها ، همانند توابع می توانند n (n>0) عبارت به عنوان آرگومان داشته باشند. یک گزاره درجهٔ صفر ،قضیه است .قضایا ، گزاره های ثابت هستند.

اتـــــم :

گزاره ها به فرمول های اتمی یا اتمها نسبت داده می شوند . وقتی می خواهیم به فرمول اتمی یا نقیض آن اشاره کنیم . از لیترال استفاده می کنیم . یک گزاره که متغیری ندارد اتم پایه نامیده می شود

۲-۷-۲ نمایش حقایق ساده در منطق

عبارات زیر را در نظر بگیرید:

S1: همه کارمندان هر سال ۳۵۰۰۰ ریال یا بیشتر مالیات می پردازند.

S2: امروز تعدادی از کارمندان در مرخصی نمی باشند.

S3:هیچ کدام از کارمندان درآمدی بیشتر از رئیس ندارند.

برای ارائه این عبارات در منطق گزاره ای باید اختصارهایی برای گزاره ها و توابع تعریف

کنیم . برای مثال :

EMP (X) برای X یک کارمند است .

PRD(X) برای X رئیس است .

inc(x) برای درآمد x

GE(u, v) برای u بزرگتر یا مساویv است.

ON – LEAVE(x) برای x امروز مرخصی است .

TAX (x) برای x مالیات می پردازد.

استفاده ۳ اختصار S_1 و S_2 و S_3 به عنوان

$$
\begin{aligned}
&s1 \ : \ \forall x \ ((EMP(x) \wedge GE \ (inc(x),35000)) \rightarrow TAX(x) \\
&s2 \ : \ \exists y \ (EMP(y) \rightarrow ON\_LEAVE \ (y) \ ) \\
&s3 \ : \ \forall xy \ ((EMP(x) \wedge PRD(y))) \rightarrow \neg \ GE \ (inc(x), inc(y)).
\end{aligned}
$$

۲-۷-۳ موضوع های ایجاد شده در تبدیل جملات انگلیسی به منطق گزاره ای

جملات زیر وفرم گزاره ای آنها را در نظر بگیرید :

Marcus (a) یک انسان است ـ MAN (Marcus)

(b) Marcus اهل Pompeian بود. Pompeian (Marcus)

(c) اهالی Pompeian رومی بودند.

$\forall x$ POMPEIAN (x) $\rightarrow$ ROMAN (x).

(d) قیصر یک فرمانروا بود. RULER(Caesar)

(e) همه رومی ها یا به قیصر وفادار بودند یا از او متنفر بودند.

(f) هر شخص به کسی وفادار است .

(g) مردم سعی می کردند حاکمانی را که به آنها وفادار نبودند ، به قتل برسانند.

(h) Marcus سعی کرد قیصر را بکشد . سعی در به قتل رساندن (tryassassinate)(مارکوس،قیصر)

۱) حقیقتی را بیان می کند که Marcus انسان است اما بعضی اطلاعات را به زبان انگلیسی نمی تواند ارائه کند برای مثال مفهوم زمان گذشته .

پذیـــرش این حـــذف به کاربرد مورد نیاز ما در محیط دانش ، بستگی دارد.

در (۲) و (۳) و (٤) از این حقیقت صرف نظر می کنیم که اسامی مناسب اغلب به یک فرد خاص منتسب نمی شوند.

از آنجایی که تعداد زیادی از افراد اسم یکسان دارند، تصمیم اینکه چندین نفر اسم یکسان دارند مستلزم دانش و استدلال به مقدار کافی خواهد بود.

یک مشکل عمده موقعی ایجاد می شود که تبدیل جمله فارسی به جمله منطقی در حیطه سورها باشد .

(٦) بعضی اوقات به صورت $\exists y \forall \times loyalto(x, y)$ نوشته می شود یعنی « وجود دارد کسی که همه به او وفا دارند » .

در (۷) و (۸) از " tryassassinate " برای سادگی استفاده کردیم .

در استفاده از این مدل نمایش ، نمی توان به آسانی ارتباطی بین سعی برای کشتن و واقعاً کشتن ایجاد کرد.

هرگاه چنین ارتباط هایی ضروری باشند ، مجبوریم مدل نمایشی دیگری انتخاب کنیم .

از این مثالها باید۳ موضوع مهم در طی تبدیل جمله فارسی به عبارت منطقی واستفاده

آنها برای استنباط عبارات جدید ، مورد توجه قرار گیرد.

- بسیاری جملات فارسی مبهم هستند. انتخاب تفسیر صحیح ، سخت خواهد بود.
- اغلب یک انتخاب از راههای ارائه دانش وجود دارد. ارائه یک مجموعه خاص از جملات ، به کاربرد محتوای دانشی که در جمله ، قرار داده خواهد شد ، بستگی دارد.
- حتی در یک موقعیت خیلی ساده ، بعید است که یک مجموعه از جملات شامل همه اطلاعات ضروری برای استدلال درباره یک موضوع باشد.

۲-۸ فرمول های خوش ـ ساخت (WFFS)

مثال های قبلی درفرم منطق گزاره ای ، به عنوان فرمول های خوش ـ ساخت شناخته شده اند.

این فرمولها به طور بازگشتی به صورت زیر تعریف می شوند.

یک فرمول اتمی ، یک فرمول خوش- ساخت (wff)است.

اگر P و Q wff باشند آنگاه :

$(P \vee Q), (P \wedge Q), \neg(P)$ و

$\exists X\ P(X), \forall x P(x), (p \leftarrow Q), (P \rightarrow Q)$. wff هستند .

مثالی از عباراتی که wff نیستند.

$\forall P\ P(x), MAN(\sim marcus), father\_of\ (Q(x)).$

حوزهٔ یک سور فرمولی است که سور در آن اعمال می شود.

وقوع یک متغیر در یک فرمول محدود است اگر و فقط اگر وقوع متغیر درحوزهٔ سور بکارگیری متغیر باشد. در غیر اینصورت متغیر آزاد نامیده می شود.

۱-۸-۲ خواص WFF

۱ . تفسیر :

در موقع انتساب ارزش های داده شده به هر سمبل گزاره در یک wff ، می گوئیم تفسیر به یک wff داده شده است

اگر ارزش یک تفسیر درست باشد ، آنگاه آن تفسیر مدلی از wff نامیده می شود. .

تفسیری از یک فرمول در منطق گزاره ای شامل دامنه غیر تهی D و انتساب ارزش ها به هر مقدار ثابت ، سمبل تابع و سمبل گزاره درآن است مانند زیر :

- به هر ثابت / متغیر یک عنصر در D نسبت می دهیم .
- به هر سمبل تابع درجه n . نگاشت از D^n به D نسبت می دهیم.
- به هر سمبل گزاره ای درجه n ، یک نگاشت از D^n به (T , F) نسبت می دهیم .

wff زیر را در نظر بگیرید .

$$(\forall x)(\exists y)p(x,y)$$

می توانیم تفسیر را روی دامنه { ۲ و ۱ } = D مثل زیر تعریف کنیم :

P (۱,۱) = true و P(۱ , ۲) = false و P (۲ , ۱) = false , p (۲ , ۲) = true

۲. اعتبار :

یک wff معتبرگفته می شود ، اگر تحت هر تفسیری درست باشد ، در غیراینصورت wff غیر معتبر نامیده می شود.

Wff های پایه معتبر به عنوان درست نما شناخته می شوند .

۳. صدق پذیری :

یک wff که برای بعضی تفاسیر درست است صدق پذیر است و wff ای که تحت هر تفسیری غلط باشد غیر صدق پذیر گفته می شود (متناقض)

Wff معتبر صدق پذیر است و wff های متناقض غیر معتبرند.

٤. نتیجه منطقی :

در ریاضیات مثل زندگی روزمره اغلب مجبور به تصمیم گیری درباره این که آیا یک عبارت از عبارات دیگر نتیجه می شود ، هستیم . این تصمیم گیری به مفهوم نتایج منطقی منجر می شود. می توانیم آن را به صورت زیر تعریف کنیم :

فرمولهای داده شده $w_۱$ تا w_n وفرمول G که نتیجهٔ منطقی $w_۱$ تا w_n است (به

عبارت دیگر G به طور منطقی از w_1 تا w_n نتیجه می شود.) اگر وفقط اگر برای هر تفسیر در w_1 تا w_n درست باشد ، G هم درست است .

w_1 تا w_n ، بدیهیات یا اصول موضوع G نامیده می شوند .

۵. برابری :

اگر ارزش های درستی ۲ wff یکسان باشند ، صرف نظر از تفاسیر ، می گوئیم این wff ها معادلند.

۲-۸-۲ عبارات منطقی هم ارز

$\neg (\neg W) = W$ (Double negation)
$W1 \wedge W2 = W2 \wedge W1$
$W1 \vee W2 = W2 \vee W1$ (Commutativity)

$(W1 \wedge W2) \wedge W3 = W1 \wedge (W2 \wedge W3)$
$(W1 \vee W2) \vee W3 = W1 \vee (W2 \vee W3)$ (Associativity)

$W1 \vee (W2 \wedge W3) = (W1 \vee W2) \wedge (W1 \vee W3)$
$W1 \vee (W2 \vee W3) = (W1 \wedge W2) \vee (W1 \wedge W3)$ (Distributivity)

$\neg(W1 \wedge W2) = \neg W1 \vee \neg W2$
$\neg(W1 \vee W2) = \neg W1 \wedge \neg W2$ (D'Morgan's Law)

$W1 \rightarrow W2 = \neg W1 \vee W2$
$W1 \leftrightarrow W2 = (W1 \rightarrow W2) \wedge (W2 \rightarrow W1) = (\neg W1 \vee W2) \wedge (\neg W2 \vee W1)$

$\forall x\ W(x) \vee W1 = \forall x(W(x) \vee W1)$
$\forall x\ W(x) \wedge W1 = \forall x(W(x) \wedge W1)$

$\exists x\ W(x) \vee W1 = \exists x(W(x) \vee W1)$
$\exists x\ W(x) \wedge W1 = \exists x(W(x) \wedge W1)$

$\neg((\exists x)W(x)) = \forall x\ (\neg W(x))$
$\neg((\forall x)W(x)) = \exists x\ (\neg W(x))$

$\forall x\ W1(x) \wedge \forall x\ W2(x) = \forall x\ (W1(x) \wedge W2(x))$
$\exists x\ W1(x) \wedge \exists x\ W2(x) = \exists x\ (W1(x) \wedge W2(x))$

۲-۹ فرم عبارتی

یک عبارت می تواند به عنوان یک wff شامل ترکیب فصلی لیترال ها تعریف شود مثل $P_1 \vee P_2 \vee \ldots\ldots \vee \neg P_n$ در جایی که P_1 تا P_n لیترال هستند.

۲-۹-۱ فرم نرمال عطفی (C N F)

اگر w_1 تا w_n هر کدام شامل ترکیب فصلی از لیترال ها باشند می گوئیم $W1 \wedge W2 \wedge \ldots \wedge Wn$ به فرم نرمال عطفی است . مثل

$$(\neg P \wedge Q) \vee (P \wedge \neg Q \wedge R) \vee \neg R$$

۲-۹-۲ فرم نرمال فصلی (D N F)

اگر w_1 تا w_n هر کدام شامل ترکیب عطفی لیترال ها باشند آنگاه می گوئیم $W_1 \vee W_2 \ldots\ldots \vee W_n$ به فرم نرمال فصلی است . به عنوان مثال:

$$(\neg P \wedge Q) \vee (P \wedge \neg Q \wedge R) \vee \neg R$$

نکته : هر wff می تواند به این دو فرم نرمال تبدیل شود.

در نظر بگیرید $(P \wedge Q) \vee (R \wedge S)$ به فرم DNF است و می تواند به CNF تبدیل شود. این عبارت با عبارات منطقی معادل مانند عبارات زیر استفاده می شوند :

$$((P \wedge Q) \vee R) \wedge (P \wedge Q) \vee S)$$
$$((P \vee R) \wedge (Q \vee R)) \wedge ((P \vee S) \wedge (Q \vee S))$$
$$(P \vee R) \wedge (Q \vee R) \wedge (P \vee S) \wedge (Q \vee S)$$

۲-۹-۳ فرم نرمال Prenex

یک wff فرم نرمال Prenex گفته می شود اگر وفقط اگر آن فرمول،از فرم Qixi M باشد که ، Qi هر سوری باشد و M یک فرمول بدون سور است. اینجا Qixiپیشوندو M ماتریس فرمول نامیده می شود.

تئوری استنتاج :

فرمول های w_1 تا w_n و فرمول G داده شده ، G یک نتیجه منطقی از w_1 تا w_n است اگر وفقط اگر فرمول ($(W_1 \wedge W_2 \wedge \ldots\ldots \wedge W_n) \rightarrow G$)) معتبر باشد.

قضیـــه : فرمـــول هـای داده شده w_1 تا w_n و فرمول G ، G یک نتیجه منطقی w_1 تا w_n است اگر وفقط اگر فرمول $(W_1 \wedge W_2 \wedge .... \wedge W_n \wedge \neg G)$ متناقض باشد.

تعاریف :

عبارات صریح (عبارات horn) : یک عبارت دقیقاً شامل یک ve+ وصفر یا بیشتر لیترالهای ve– می شود.

عبارت واحد مثبت : یک عبارت قطعی که هیچ لیترال ve– ندارد.

عبارت واحد منفی : یک عبارت منفی با دقیقاً یک لیترال.

عبارت منفی : یک عبارت با صفر یا بیشتر لیترال های منفی وهیچ گونه لیترال ve+ .

عبارت واحد : عبارت واحد ve+ یا ve– است.

عبارت تهی : عبارت منفی بدون لیترال .

عبارات غیرصریح (عبارات غیر horn) : عبارات شامل حداقل ۲ لیترال ve+ وصفر یا تعداد بیشتر لیترال ve– .

۲-۱۰ قوانین استنتاج

درجبر گزاره ای قوانین استنتاج وجود دارند ، که می توانند برای wff معین به کار برده شوند و مجموعه ای از این wff ها برای تولید wff های جدید استفاده می شوند. این قوانین عبارتند از:

۱ـ Modus Ponens : عملی است که W2 wff را از wff های زیر تولید می کند.

$W_۱$ and

$W_۱ \rightarrow W_۲$

۲ـ Modus tollens : عملی است که $\neg W_1$ ، را از $\neg W2$ and $\rightarrow W_۲$ و

W_1 تولید می کند.

۳ـ Universal Specialization : W(A) را از wff های $\forall x W(x)$ تولید می کند جائیکه A یک نماد مقدار ثابت است .

با استفاده از قوانین بالا $W_2(A)$ را از wff های زیر می توانیم تولید کنیم .

$W_1(A)$ and

$\forall x W(x) \rightarrow W_2(x)$

برای مثال عبارت « Leo یک شیر است » و استنباط « همه شیرها وحشی هستند » را در نظر بگیرید.

می توانیم نتیجه « Leo وحشی است » را استنباط کنیم.

W1 : LION (leo)
W2 : ∀x LION(x) → FEROCIOUS(x)
FEROCIOUS(leo)

از w_1 و W_2 بایـد نشان دهیـم (FEROCIOUS (Leo یک نتیجه منطقی از w_1 و W_2 است .

W1 ∧ W2 : LION(leo) ∧ (∀x LION(x) → FEROCIOUS(x))

چون w_1 و W_2 درست هستند و $Lion(x) \rightarrow Ferocious(x)$ برای همه X ها درست می باشد ، می توانیم x را با Leo جایگزین کنیم . پس W_2 ، $\neg LION(leo) \vee FEROCIOUS(leo)$ خواهد بود ، ولی $\neg LION(leo)$ غلط است پس FEROCIOUS(leo) درست می باشد.

٤ـ قاعده زنجیری : عملی است که $P \rightarrow R$ را از wff های $P \rightarrow Q$ و $Q \rightarrow R$ تولید می کند.

قوانین استنتاجی wff ها را از wff های موجود در جبرهای گزاره ای مشتق می کنند. این wff های مشتق شده ، قضایا نامیده می شوندو دنباله ای از کاربرد قوانین استنتاج اثبات قضیه را تشکیل می دهد.

۲-۱۰-۱ جایگزینی

در مثال قبل برای استنتاج FEROCIOUS(leo) ، جایگزینی x ضروری بود. جایگزینی ها ، اجزای ضروری فرآیند استنتاج هستند.

تعریف :

یک جایگزینی به عنوان مجموعه ای از زوج های ti و Vi تعریف می شونددر حالی که vi متغیرهای مجزا و ti عبارات هستند.

در عبارت ، ti متناظر vi جایگزین می شود. اینها با حروف یونانی نمایش داده می شوند.مثل:

$$a = \{t_1/v_1, t_2/v_2, \ldots, t_n/v_n\} \quad (n\geq 1)$$

برای مثال اگر $C = P(x, y) \vee Q(x, f(y))$ and $\beta = \{a/x, g(b)/y\}$ ،آنگاه بعد از به کاربردن β در C داریم : $C' = C\beta = P(a, g(b)) \rightarrow Q(a, f(g(b)))$.

یک نمونه جایگزینی از یک عبارت ، با جایگزینی ترم ها برای متغیرها ، درآن عبارت حاصل می شود.

برای مثال نمونه های $W = P(x, f(y), b)$ عبارتند از :

$W\{z/x, w/y\} = P(z, f(w), b)$
$W\{a/y\} = P(x, f(a), b)$
$W\{g(z)/x, a/y\} = P(g(z), f(a), b)$
$W\{c/x, a/y\} = P(c, f(a), b)$

۲-۱۰-۲ ترکیب جایگزینی ها

ترکیب ۲ جانشینی α و β با $\alpha\beta$ مشخص می شود ، که با به کاربردن β در عبارات α و اضافه کردن جفت هایی از متغیرهای β که در α وجود ندارند ، بدست می آید.

α = {g(x, y)/z}
β = {a/x, b/y, c/w, d/z} then
$\alpha\beta$ = {g(a, b)/z, a/x, b/y, c/w}

نکته : می تواند نشان داده شود که :

$$(E\alpha)\beta = E(\alpha\beta)$$

در حالی که E یک عبارت است و $(\alpha\beta)\beta\alpha$ یک ترکیب و حالت شرکت پذیری می باشد نه جا به جایی پذیری .

۲-۱۱ یکسان سازی

هر جایگزینی که ۲ یا تعداد بیشتری عبارات را معادل کند ، متحد کننده برای عبارات نامیده می شود.

می گوئیم یک مجموعه از عبارات {Ei} قابل اتحاد است اگر یک جایگزینی « a » وجود داشته باشد مثل:... =$E1a = E$۲a که a متحد(یکسان) کننده برای مجموعه {Ei} نامیده می شود.

مثال : α = {a/x, b/y} مجموعهٔ {P(x, f(y), b), P(x, f(b), b)}

را برای حاصل {P(a, f(b), b)} یکسان می کند.

اینجا می گوئیم α عمومی ترین یکسان کننده (M G U) مجموعه است ، اگر بقیه یکسان کننده ها یک نمونه از α باشد.

الگوریتم های زیادی برای یکسان سازی یک مجموعه عبارات با قابلیت یکسان شدن ، وجود دارند ،که وقتی مجموعه نتواند یکسان شود خطا را گزارش می دهد. رویه بازگشتی داده شده در زیر می تواند برای یکسان کردن مجموعه ای از ۲ رویه قابل یکسان سازی استفاده شود.

Procedure UNIFY (E1, E2)

اگر $E_۱$ یا $E_۲$ یک اتم (مثلاً یک نماد گزاره ای ، یک سمبل تابع ، مقدار ثابت ، یک علامت نقیض یا یک متغیر) باشند ، آرگومانها را با هم تبادل می کنند به طوریکه $E_۱$

یک اتم باشد .

```
BEGIN
   IF E1 and E2 are identical THEN return NIL
   IF E1 is a variable DO
   BEGIN
      IF E1 occurs in E2, return FALL
      Return (E2/E1)
   END
   IF E2 is a variable, return (E1/E2)
   Return FAIL
END

F1 := First element of El, T1 := Rest of El
F2 := First element of E2, T2 := Rest of E2
Z1 := UNIFY (F1, F2)
IF Z1 == FAIL, return FAIL
Gl := Result of applying Z1  to T1
G2 := Result of applying Z1  to T2
Z2 := UNIFY (G1, G2)
IF Z2 == FAIL, return FAIL
return the composition of Z1 and Z2
```

برای مثال اگر {P(x, f(y), b), P(x, f(a), b)} را یکسان سازی کنیم ، رویه بالا مجموعه یکسان شده { a/y} را بر می گرداند.

عبارت زیر موجود در الگوریتم قبلی را در نظر بگیرید

IF E1 occurs in E2, return FAIL

این تست ، تست بررسی رخداد نامیده می شود وهدفش نپذیرفتن قیود های خود ـ ارجاعی است مثل X/F(x)

اگر به این قید ها توجه نکنیم ، به حلقه بی نهایت وخرابی سیستم منجر می شود.

یکسان سازی را برای کشف اینکه آیا یک لیترال می تواند با دیگری مطابقت کند یا نه ، استفاده می کنیم . امکان وجود متغیرهایی در هر دو لیترال می باشد ، وممکن است این متغیرها دارای عبارات جایگزین برای آنها باشند که لیترال ها را یکسان کنند .

۲-۱۲ تفکیک پذیری

تفکیک پذیری یک قاعده مهم استنتاج است که برای یک مجموعه عبارات می تواند به کار رود. این متد به وسیله رابینسون در ۱۹٦٥ توسعه داده شد. برای به کار بردن این قاعده (Resolution) مجبوریم همه wff ها را به فرم عبارتی تبدیل کنیم .

رویه تبدیل در زیر داده شده :

۱ـ حذف همه سمبل های دلالت .

$A \rightarrow B$ with $\neg A \vee B$ and
$A \leftrightarrow B$ with $(\neg A \vee B) \wedge (\neg B \vee A)$

۲ـ کاهش حیطه منفی(نقیض) .

(از قوانین دمرگان و قوانین معادل دیگر می توان استفاده کرد.)

هدف به کاربردن هر سمبل منفی (نقیض) برای یک فرمول اتمی است .

۳ ـ استاندارد سازی متغیر ها :

درحوزهٔ یک سور، یک محدوده متغیر به وسیله سور، یک متغیر مصنوعی است.

یک متغیر مصنوعی می تواند با متغیر دیگری جایگزین شود ، که در جای دیگری ، در سراسر محدودهٔ سور ، بدون تغییر ارزش wff استفاده نشده است .

این کار برای اطمینان از اینکه هر سور دارای متغیر مصنوعی یکتای خودش است ، می باشد.

برای مثال $\forall x\ P(x) \rightarrow \exists x\ Q(x)$ بصورت $\forall x\ P(x) \rightarrow \exists y\ Q(y)$ می تواند نوشته شود.

٤ـ حذف سورهای وجودی .

این عمل برای جایگزینی هر رویدادی از متغیرهای کمیت سنج (سور) وجودی با یک تابع Skolem می باشد ،که آرگومان هایش متغیرهای عمومی ای هستند که حوزه آنها شامل حوزه سور وجودی است .

برای مثال wff را در نظر بگیرید :

$\forall x\ (\exists y\ P(x, y));$

در اینجا وجود y به ارزش x که صریحاً به وسیله تابع g(x) تعریف می شود ، بستگی دارد ، که هر x را به y نگاشت می دهد. پس می توانیم y را با g(x) در wff قبلی جایگزین کنیم که $\forall x p(x, g(y))$ خواهدشد. (این فرآیند Skolemization نامیده می شود) .

نکته : تابع استفاده شده Skolem باید یکتا باشد. اگر E در قلمرو چیزی چون A نباشد ، می توانیم تابع Skolem را بدون هیچ آرگومانی بگیریم.

۵ـ تبدیل به فرم Prenix :

دراین مرحله هیچE وجود ندارد و هر A متغیر خودش را دارد. پس می توانیم A را به ابتدا منتقل کنیم .

٦ ـ تبدیل به CNF

۷ـ حذف :

در این مرحله همه متغیرهای مورد استفاده عمومی اند . پس می توانیم حضور صریح A را حذف کنیم. اکنون ما در چپ ماتریس درفرم CNF هستیم .

۸ـ حذف ∧ :

این عمل با نوشتن عبارت به شکل $(W_1 \wedge .... \wedge W_n)$ به صورت $\{W_1,....,W_n\}$ انجام می شود. اینجا هر Wi (i=1,2,…,n) یک ترکیب فصلی رشته ها است.

.

۹ـ تغییر نام متغیرها :

متغیر می تواند تغییر نام داده شود طوریکه هیچ متغیری در بیش از یک عبارت ظاهر نشود.

مثل:

$$S = \forall x \{P(x) \rightarrow \{\forall y[P(y) \rightarrow P(f(x,y))] \wedge \neg(\forall y)[Q(x,y) \rightarrow P(y))]\}\}$$

1. $\forall x \{\neg P(x) \vee \{\forall y[\neg P(y) \vee P(f(x,y))] \wedge \neg(\forall y)[\neg Q(x,y) — P(y)]\}\}$
2. $\forall x \{\neg P(x) \vee \{\forall y[\neg P(y) \vee P(f(x,y))] \wedge \exists y[Q(x,y) \wedge \neg P(y)]\}\}$
   (By using $\neg(\forall x)P(x) = \exists x (\neg P(x)$ and D'Morgan's Law)
3. $\forall x \{\neg P(x) \vee \{\forall y[\neg P(y) \vee P(f(x,y))] \wedge \exists w[Q(x,w) \wedge \neg P(w))]\}\}$
4. $\forall x \{\neg P(x) \vee \{\forall y[\neg P(y) \vee P(f(x,y))] \wedge [Q(x, g(x)) \wedge \neg P(g(x))]\}\}$
5. $\forall x \forall y \{\neg P(x) \vee \{[\neg P(y) \vee P(f(x,y))] \wedge [Q(x,g(x)) \wedge \neg P(g(x))]\}\}$
6. $\forall x \forall y \{[\neg P(x) \vee \neg P(y) \vee P(f(x,y))] \wedge [\neg P(x) \vee Q(x,g))] \wedge [\neg P(x) \vee \neg P(g(x))]\}$
7. $\{[\neg P(x) \vee P(y) \vee P(f(x,y))] \wedge [\neg P(x) \vee Q(x,g(x))] \wedge [\neg P(x) \vee \neg P(g(x))]\}$
8. $\{\neg P(x) \vee \neg P(y) \vee P(f(x,y)),$
   $\neg P(x) \vee Q(x,g)),$
   $\neg P(x) \vee \neg P(g(x))\}$
9. $\{\neg p(x1) \vee \neg P(y) \vee P(f(x1,y)),$
   $\neg P(x2) \vee Q(x2,g(x2)),$
   $\neg P(x3) \vee \neg P(g(x3))\}$

قضیه: اگر یک فرمول x منطقاً از یک مجموعه wff ها نتیجه شود ،S، آنگاه به طور منطقی از مجموعه ای از عبارات با تبدیل wff ها در S به فرم Clausal نتیجه می شود.

۲-۱۲-۱ تفکیک پذیری(Resolution) عبارات پایه

فرض کنیم ۲ عبارت پایه $P1 \vee P2 \vee \ldots Pn$ و $Q1 \vee Q2 \vee \ldots \vee Qn$ داریم فرض می کنیم Q_J, P_i مجزا هستند و یک لیترال در یک عبارت ، نقیض لیترال در یک عبارت دیگر را در برداشته باشد. در اینجا

جواب با گرفتن ترکیب فصلی ۲ عبارت وحذف جفت های مکمل P_1 و $\neg P_1$ محاسبه می شود.

بعضی موارد خاص resolutionعبارتند از :

Parent clause	Resolvants	
P and ¬P ∨ Q	Q	(Modus ponen)
P ∨ Q and ¬P ∨ Q	Q	(Q and Q collapses to Q)
P ∨ Q and ¬P ∨ ¬Q	P ∨ ¬P, Q ∨ ¬Q	(both are tautologies)
¬P and P	NIL (empty clause)	(sign of contradiction)
¬P ∨ Q and ¬Q ∨ R	¬P ∨ R	(Chain rule)

۲- ۱۲- ۲ تفکیک پذیری(resolution)عمومی

اینجا هر عبارت به عنوان یک مجموعه از لیترالها با عطف بین آنها در نظر گرفته می شود.

عبارات پدر متناظر که با { Li} و { Mj } داده شده ا ند ، در نظر بگیرید.

فرض کنید {li} یک زیر مجموعه از { Li} و {mj } زیر مجموعه ای از { Mj } است که MGU هایی برای هر اجتماع مجموعه های { Li} و { mj } وجود دارد.

$\{\{L_i\} - \{l_i\}\}s \cup \{\{M_j\} - \{m_j\}\}$

می گوییـــم { Li} و { Mj } حل شوند و ها را به عنوان راه حل می دهیم.

برای مثال عبارات زیر را در نظر بگیرید :

$$P(x, f(a)) \vee P(x, f(y)) \vee Q(y) \text{ and } \neg P(z, f(a)) \vee Q(z)$$

With $l_i = P(x, f(a))$ and $m_i = \neg P(z, f(a))$
we will get the resolvant $P(z, f(y)) \vee \neg Q(y) \vee Q(y)$.

With $l_i = P(x, f(a)), P(x, f(y))$ and $m_i = \neg P(z, f(a))$,
we will get the resolvant as $Q(a) \vee \neg Q(z)$.

۲-۱۲-۳ تئوری کامل بودن تفکیک پذیری(resolution)

اگر resolution مرتباً به مجموعه ای از عبارات صدق پذیر اعمال شود سرانجام عبارت تهی تولید خواهد شد.

۲-۱۲-٤ تفکیک پذیری (resolution) در تئوری تفکیک پذیری

.

تئوری یا مبنای تفکیک پذیری ، اثبات سیستم های طراحی شده ، جهت ایجاد دلیل از طریق تناقض یا برهان خلف می باشد. در برهان خلف ، ابتدا هدف wff را خنثی می کنیم و سپس عامل خنثی شده(نقیض) را به مجموعه عبارات (حکم) اضافه می کنیم.
این احکام به مجموعه ای از عبارات تبدیل می شود و با استفاده از تئوری تفکیک پذیری ، تناقض ارائه شده از طریق عبارت تهی (NIL) منتج خواهد شد.
اثبات: فرض کنیم $wff\ W$ به طور منطقی از مجموعه ای از wff های S منتج شود، با تعریف هر تفسیر قانع کننده ودرست از S ، W هم درست می باشد.
بنابراین هیچ کدام ازتفاسیر درست S نمی تواند ، $\neg W$ را تفسیر کند.
برای مثال، هیچ تفسیری نمی تواند اجتماع S و $\neg W$ را بیان کند .
پس اگر W به طور منطقی از S منتج شود ، آنگاه $S Y \neg W$ درست نمی باشد.
با توجه به کلیت تئوری ، $S Y \neg W$ قابل استنتاج نیست ، پس تفکیک پذیری (resolution) و عبارت تهی (NIL) ایجاد خواهد شد.
برای مثال عبارات زیر را در نظر بگیرید
"هر کسی که بتواند بخواند ، با سواد است

$$\forall\ x\ R(x) \rightarrow L(x) \qquad (۱)$$

"میمونها با سواد نیستند."

فصل سوم

فراگیری و نمایش دانش

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

هوش ماشین

مهندسی دانش

رویه برای فراگیری دانش

نمایش حقایق

طرح های نمایش منطقی

طرح های نمایش رویه ای

طرح های نمایش شبکه

طرح های نمایش ساخت یافته

۳-۱ مقدمه

حتی بحث کردن راجع به کار ماشینهای نسبتاً ساده مانند ماشین های لباسشوئی یا چرخ های خیاطی سخت به نظر می آید ، مگر اینکه کارکرد هایی را که این ماشین ها ، بر اساس آنها جهت انجام وظایفشان طراحی شده اند ، قابل درک باشد . ازآنجائی که هوش مصنوعی با پیچیده ترین نوع ماشینهای قابل تصورمان ، ارتباط دارد ، 'ماشین

های هوشمند ' ، شاید باید در تعریف کردن انتظارمان از وظایف چنین ماشین هایی ، کوشش کنیم . بدیهی است که انتظار داریم این ماشین هاا هوشمند باشند . اما منظورمان از هوشمندی چیست ؟ فرهنگ لغت تعاریف زیر را ارائه می دهد :

هوشمندی : توانائی درک کردن

همچنین درک کردن : دریافت و درک چیزی یا تشخیص معنای آن .

این تعاریف مفهومی است که به اندازه کافی واضح و روشن به نظر می آید . وقتی این مفاهیم را به طور ذهنی به کار می بریم ، به نظر می رسد که با تجربه شخصی مان از چیزی که مایل است هوشمند باشد یا از هوش مان استفاده کند را بطه ای بدون خطا دارند . متاسفانه ، به کارگیری این مفاهیم هنگامی که جهت بکاربستن شان به طور معقول و با وجود حالت خارجی ، و بررسی هوش به عنوان یک توانائی ذهنی تلاش می کنیم ، رو به زوال است ، که ممکن است این توانائی ذهنی با دیگر موجودیتها خواه زنده یا مکانیکی ، مشترک و سهیم باشد .

مسئله اصلی این است که ، می دانیم برای درک چیزی چه احساس می کنیم و به طور عام مایل هستیم به دیگر افراد یا اشیاء با احساس مشابهٔ احساس خودمان اعتبار دهیم . یک مثال ساده مانند ، یک قطعه آشنا از ماشین آلات ، مثل ترموستات درسیستم حرارت مرکزی را در نظر بگیرید . ترموستات فقط زمان بالاتر یا پائین تر بودن دما از یک حد معین را تشخیص نمی دهد ، بلکه با یک عمل مناسب به این وضعیت پاسخ می دهد . به نظر می آید که در یک رابطه کاملاً محدود شده ، ترموستات دارای درک و فهم می باشد و این موضوع را با واضح ترین روش ممکن ، از طریق رفتاری هوشمندانه اثبات می کند . اگر ترموستات هوشمند باشد ، ارزش لغت مربوط به موضوع را در جایی که بی معنی باشد، می کاهیم . هوشمندی عام به این نتیجه رسیده است که هنگامی که در عمل ، و تمامی سئوالات مورد استفاده در آزمون هوش که به عنوان معیاری برای سنجش میزان ارزشمندی یا شایستگی افراد جهت شغلهای کاملاً مباحثه ای ، مورد استفاده قرار می گیرد ، یک تصور کلی از مقداری مشکوک است . بنابراین آیا لازم

است که هوشمندی را به توانائی های ذهنی جداگانه ای ازقبیل ادراک ، دلیل ، خلاقیت تجزیه کنیم ؟ اگر چنین است ، تفاوت میان هوشمندی و دانش چیست؟

یکی از معدود نتایج غیر قابل تغییر و ثابت که از سه دهه اول تحقیقات AI شناخته شده است ، این است که " هوشمندی به دانش نیاز دارد ." برای جبران سرمایه ی مقاومت ناپذیر این نتیجه ، ضرورتاً دانش دارای چندین ویژگی مطلوب کمی است که شامل موارد ذیل می باشد :

- حجیم هستند .
- برای توصیف و مشخص کردن به صورت درست و دقیق سخت ودشوار است .
- پیوسته در حال تغییر است.
- ازمرحله داده تا سازمان دهی شده متفاوت می باشد.

راههای مختلفی برای طبقه بندی انواع دانش وجود دارد ، یکی از مهمترین امتیاز ها تفاوت میان دانش القاء شده و دانش استنتاج شده ، است . این بهترین تعبیر به وسیله یک مثال است .

ملاحظه یک مهارت معمولی که بیشتر بچه ها در سنین بین پنج تا ده سالگی بر آن تسلط دارند ، در اختیار گرفتن توپ است . ممکن است برای کودکی در سن پنج سالگی گرفتن یک توپ بزرگ و سبک و پرتاب کردن آن به سمت بالا حتی برای کسری از یارد مشکل باشد ، اما چندین سال بعد شاید قادر به در اختیار گرفتن یک توپ تنیس وپرتاب آن به هوا تا بیست یارد دورتر هم باشد . بشر در سالهای اولیه به طور موثر قادر به کسب مهارت تکنیکی و فنی جهت محاسبه مسیر گلوله های پرتابه ای نبود . فهم یک کودک به وسیله قیاس بدست می آید . این فهم در نتیجه مشاهده مسیرهای توپ های زیادی و تلاش برای در اختیار گرفتن آنها حاصل می شود که بچه ها قادر به پیش بینی مسیر توپ بعدی که می خواهند دراختیار بگیرند هستند . از طرف دیگر ، یک سیستم کامپیوتری با تکیه بر اطلاعات شامل سرعت سیر پرتابه و مسیر آن ، مکان بعدی پرتابه با استفاده از قوانین نیوتن را محاسبه می کند . این موضوع به

مجموعه فرمول های واضح ، دقیق ، وابسته به قوانین ریاضی و برنامه ریزی شده در کامپیوتر، بستگی دارد . این برنامه ، کامپیوتر را قادر می سازد تا در مورد مسیر پرواز یک پرتابه به وسیله مراجعه به مجموعه قوانین رسمی ریاضی ، نتیجه گیری کند .

تعداد کمی از مردم ممکن است در مورد محاسبه قضیه مسیر یک پرتابهٔ وابسته به قوانین ریاضی که به هوشمندی بیشتری نسبت به توانایی در اختیار گرفتن یک توپ نیاز دارد ، با هم بحث کنند .

پس یک تمایز مهم وجود دارد که میان دانش و هوشمندی ایجاد شده است .

همچنین واضح است که امکان دارد یک ماشین لزوماً بدون داشتن هوش ، دانش را ذخیره کند ، ولی وجود یک ماشین هوشمند که دانش نداشته باشد غیر ممکن به نظر می آید .

سئوال راجع به چگونگی ، میزان وسعت و اندازه که می تواند به وسیله دانش در یک ماشین رسوخ کند ، در همه زمینه های هوش مصنوعی یک سئوال بنیادی است .

۳-۲ هوش ماشین

بیشتر از فرو رفتن در مجادلاتی که هوش ذاتی انسان را فرا می گیرد ، شاغلین و صاحبنظران در هوش مصنوعی ، صرفنظر از خصوصیات نظری ، جهت تعریف اهداف خودشان برگزیده شده اند . این صاحبنظران اظهار کردند که یک ماشین هوشمند ، ماشینی است که قادر به انجام چیزهایی است که به وسیله بشر انجام شده است که برای قضاوت نیاز به هوش دارد . بنابراین تعریف هوش غیر ضروری است . این مطلب خیلی کاربردی به نظر می رسد . اگر ما نمی توانیم هوش را تعریف کنیم ، اما قادر به تشخیص اینکه افراد کی هوشمند هستند ، می باشیم .

متاسفانه ، این بحث به سادگی مغلوب می شود . برای قیاس منطقی در رابطه با نیاز مردم به هوش ، برای انجام یک وظیفه ، و هوشمند بودن یک ماشین که نسخه برداری و تکرار می کند ، افراد بدبین و شکاک پاسخ می دهند که یک ماشین می تواند کاری را یک بار انجام دهد ، آنگاه اثبات شده است که آن وظیفه ، یک کار مکانیکی و غیر

.

فکری می باشد . اگر قادر به توضیح نحوه کار یک ماشین باشیم ، بی درنگ هوشمندی آن را توضیح داده ایم . اگر هم بپذیریم که یک ماشین هوشمندانه رفتار می کند ، آیا ممکن است این رفتار هوشمند همان هوشی نباشد ، که به عنوان یک خصوصیت از افرادی می شناسیم که کامپیوتر و نه خود ماشین را برنامه ریزی کرده اند ؟ همچنین آیا می توان تمایزی میان رفتاری هوشمند و رفتاری که فقط هوشمند به نظر می رسد ، دریافت کرد ؟

به طور طبیعی ، زمانی که نمی دانیم چه در سر ما رخ می دهد ، پذیرش این بحث ، دلالت ضمنی بر این مطلب دارد که هوش خود ما ممکن است به سبک و روشی غیر واقعی باشد . شاید فقط فکر می کنیم ، که هوشمند هستیم . در مورد یک ماشین می توانیم خصوصیت و طبیعت فرآیندهای دخلی را ایجاد کنیم . اگر ماشین و انسان رفتار یکسانی نشان دهند ، آیا فرآیندهای داخلی در چند روش قابل قیاس هستند ؟

اگر دانش یکسان و همگن شده بیشتر از انباشتگی داده ، نباشد ، حتی یک کامپیوتر نسبتاً متوسط بیشتر از هوشمند ترین افراد ارزیابی شده است . به طور مسلم ، ظرفیت محض حافظه کامپیوتر از حافظه خود ما پیشی گرفته است . دقیقاً این مطلب ، دلیل مفید واقع شدن کامپیوترها می باشد که در یک میلیونیم ثانیه می توانند به فروشنده بلیط اطلاع دهند ، که آیا جائی در پرواز فردا به لوس آنجلس باقی مانده است یاخیر . تنها حافظه بزرگ قابل دسترسی است ، زیرا فردی عامل ترمینال ، وجود دارد که سئوالات کاملاً درست ، و دقیقاً با روش و سفارشی صحیح را می پرسد .

موضوعات کلیدی که یک طراح سیستم AI با آنها روبرو می شود ، عبارتند از:

- فراگیری دانش
- ارائه دانش
- دستکاری دانش

دستکاری دانش اصولاً به واسطه استنتاج و استنباط و غالباً استراتژی کنترل جستجوگرا یا موتور استنباط (که موارد دانش دست یابی شده ، نتایج ایجاد شده ، و ترتیب گامهای بکار برده شده را تعیین می کند .) رخ می دهد .

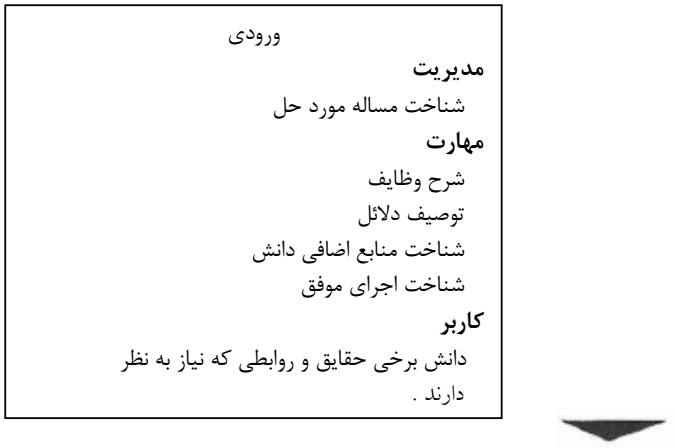
فراگیری دانش چیست ؟ فراگیری دانش چنین تعریف شده است : انتقال و تغییر شکل نظریه فنی حل مساله بالقوه از چند منبع دانش به یک برنامه . بوچانان۱ ۱۹۸۳
فرآیند فراگیری دانش نیز می تواند به عنوان وظیفهٔ شناختن و استخراج کردن دانش دامنه مربوط ، در مقادیر کافی برای ایجاد توانمندی جهت حل مسائل پیچیده تعریف شود . غالباً این فرآیند به انتقال نظریه فنی اشاره می کند ،که با استخراج دانش از منابع متنوع و نمایش آن در یک غالب مناسب درگیر است .

۳-۳ مهندسی دانش

فیلد مهندسی دانش می تواند به عنوان فرآیند مسائل ارزیابی ، دانش فراگیری و سیستم های ساختمان برمبنای دانش تعریف شود .
(هارمون۲ و کینگ۳ ۱۹۸۵)
سیستمهای کامپیوتری مرسوم به طور سنتی از طریق تحلیل گر سیستم ، برنامه ریزی و طراحی شده اند . تحلیل گر ارتباطی بین کاربری که دانش کمی دارد یا فاقد دانش در مورد جزئیات تکنیکی سخت افزار و نرم افزار است و یک برنامه نویس که ایده کوچکی در مسائل خاص و تقاضاهای کاربر را دارد ، فراهم می آورد .
هرچند یک تفاوت عمده وجود دارد ، اما سیستم های بر مبنای دانش (شکل ۳,۱) نیازمند رهیافت مشابهی هستند . تحلیل گران سیستم ها ، سیستم های کامپیوتری ای که روالهای٤ خاصی را انجام می دهند ، طراحی می کنند . به هر حال ، توسعه و پیشرفت بر مبنای دانش ، با الحاق دانش خبره در یک سیستم ، درگیر است . به همین دلیل ، مهندس دانش توسعه سیستم را به طور سنتی عهده دار شده است (آدلی۵ ۱۹۸۸) . یک مهندس دانش با تحلیل گر سیستم قابل مقایسه است اما با فرآیندهای خبره ارتباط بیشتری دارد .

---
1 Buchanan
2 Harmon
3 King
4 Procedure
5 Adeli

شکل ۳-۱ سیستم بر مبنای دانش



هارت۱ (۱۹۸٦) پیشنهاد داد که مهندس دانش ، هرگز نباید متخصص در تهیه دانش باشد . این موضوع برمبنای فرضی بیان شده است که مهندس دانش معمولاً دارای دانش ناکافی درباره برنامه نویسی و تکنیکهای سیستم مبتنی بر دانش خواهد بود ، و شرح و توصیف کامل و دقیق دانش شان ، دشوار خواهد بود .

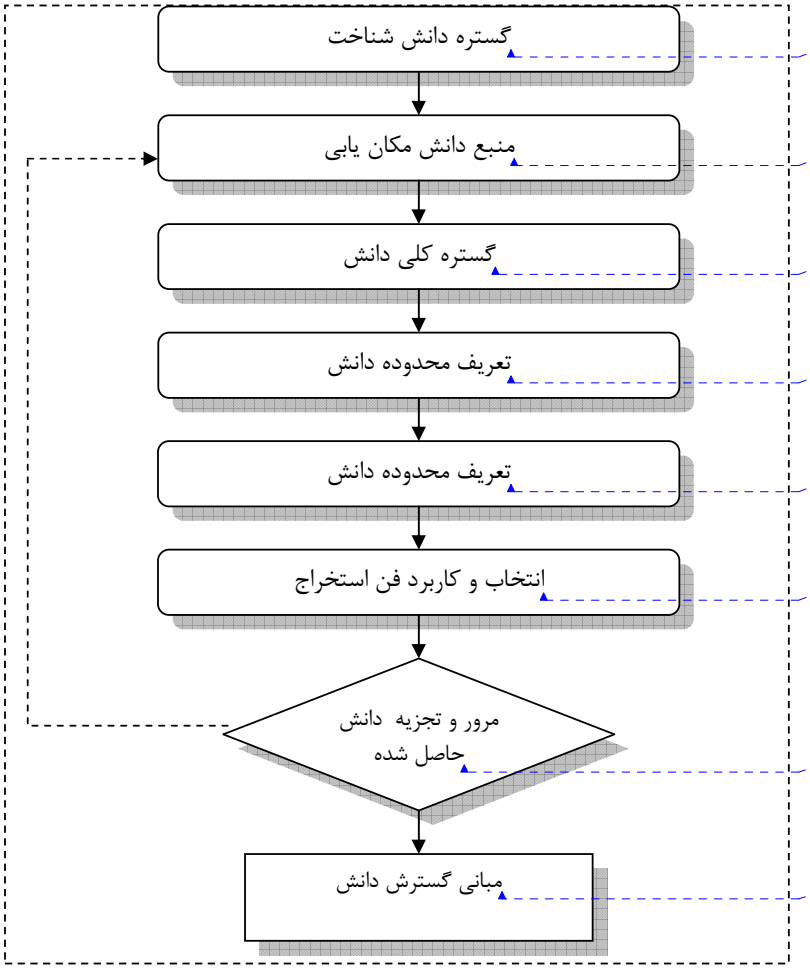
این نقطه نظر می تواند بر زمینه هائی از چنین فلسفه ای اعتراض داشته باشد که به زیان های این قبیل رهیافت ها رسیدگی و توجهی نمی کند . یک مهندس دانش باید با دامنه دانش تحت آزمایش ، چیزی که اغلب به آسانی قابل دست یافتن نیست ، آشنا باشد . خصوصاً این حالت برای دامنه های دانش مهندسی می باشد . آدلی (۱۹۸۸) معتقد است ، در آینده ، سیستم های مبتنی بر دانش مهندسی ، به وسیله مهندسین وارد به کار و مطلع در دامنه های کاریشان توسعه خواهد یافت ، مخصوصاً چنانکه مهندسین بیشتری معرفی شده باشند و در باره ابزارها و فنون AI مطالبی را آموخته باشند .

---

1 Hart

۳-٤ رویه برای فراگیری دانش

در ابتدا ، دامنه دانش(شکل ۳,۲) باید برای تضمین کردن توسعه یک سیستم محکم و متمرکز ، کاملاً شناخته شده باشد . این دستاورد از طریق یک آزمایش دقیق بر روی اهداف سیستم پیشنهاد شده ، به دست می آید .



Formatted: Font: 12 pt, Complex Script Font: B Nazanin, 12 pt

Formatted: Font: 12 pt, Complex Script Font: B Nazanin, 12 pt

Formatted: Font: 12 pt, Complex Script Font: B Nazanin, 12 pt

Formatted: Font: 12 pt, Complex Script Font: B Nazanin, 12 pt

Formatted: Font: 12 pt, Complex Script Font: B Nazanin, 12 pt

Formatted: Font: 12 pt, Complex Script Font: B Nazanin, 12 pt

Formatted: Complex Script Font: B Nazanin

Formatted: Complex Script Font: B Nazanin

شکل ۳-۲ رویه ای برای فراگیری دانش

منابع دانش دامنه مربوط و مناسب می توانند برای تحلیل و استخراج بعدی ، مکان یابی شوند . پیش از تعریف مرزهای دامنه ، یک نگاه کلی به موضوع اصلی مقتضی و قابل توصیه است . این نگاه کلی ، یک بصیرت و ساختاری برای کارهای متعاقب فراهم می کند و همین طور مشکلات بالقوه را نیز مشخص و پر رنگ می کند . تعریف مرزهای دامنه تضمین می کند که در طی فرآیند استخراج هیچ گونه حذفیات عمده یا تلاش هدر رفته در ارتباط با گردآوری اطلاعات ناخواسته ، صورت نمی گیرد. انتخاب یک تکنیک استخراج مناسب ، از روش برگزیده شده که برای تجزیه کردن اطلاعات لازم بکار برده شده است ، پیروی می کند . تنوع و گوناگونی تکنیک ها قابل دسترسی هستند .

زمانی که دانش شناخته شده و فراگرفته شده باشد ، آخرین وظیفه در پردازش این است که تضمین کند ، دانش درست و بی عیب ، در صورت تکرار قابل اعتماد و متمرکز شده در یک قالب مناسب است . توانگری دانش که در دامنه تحت رسیدگی و بررسی شرکت می کند ، باید قابل فهم و واضح باشد .

استخراج ، بخش های۱ انفرادی دانش را که لازم است در تمام موارد یکسان شده ، سازمان دهی شده باشد ، تحویل می دهد .

این که چه میزان سازمان دهی لازم است ، بستگی بسیاری به روشی که بانک اطلاعات اجرا و استفاده شده است ، دارد . برخی پیاده سازی ها به بخش هایی نیاز دارند که در گروه ها ودنباله ها ترتیب داده شده اند ، که این نظم و ترتیب ارتباطات میان بخش های موجود در بانک اطلاع را نشان می دهد . ترتیب آنها می تواند استفاده شود ، به عنوان مثال ، برای تعیین دنباله که در بخش های مختلف استفاده شده است _یک تصمیم مهم که اگر چندین بخش بتوانند به طور معتبر در نقطه معین مذاکره استفاده شوند ، سیستم مجبور به ایجاد آن است .

برای نقل کردن یک مثال جامع و محکم ، خیلی از سیستم ها به اطلاعاتی در باره

1 Item

مشتری و بیمار از قبیل : نام ، سن ، جنسیت و سایر موارد ، نیاز دارد . سیستم مجبور به انتخاب ترتیبی برای بدست آوردن این اطلاعات دارد . یک راه ساده دنبال کردن ترتیب متنی این بخش ها در مبنای دانش است . در موارد دیگر ممکن است ضروری باشد که این اطلاعات در یک مجموعه ترتیبی استفاده شده باشد .

۳-٤-۱ منابع دانش

سه منبع اصلی دانش عبارتند از :

- ادبیات
- مهارت ها
- مثال ها.

سه رکن مختلف دانش عبارتند از :

- قوانین علمی
- تجربه
- الگوها

می دانیم که دانش ، اطلاعاتی است که به ما در حل مسائل موجود در یک دامنه خاص یاری می دهد . سودمندترین دانش ، بیانی ازبعضی قواعد است که به ما برای پیش بینی چیزی که بعداً اتفاق خواهد افتاد یا توضیح و تشرح نحوه و دلیل برخی مواردی که اتفاق افتاده است ، یاری می رساند . قویترین قوانین با قاعده که می شناسیم ، قوانین علمی می باشند . بنابراین ، فراگیری دانش در زمینه علمی از زمینه های دیگر ساده تر است ، و اینکه آیا این قوانین ازکتاب به دست آمده اند یا از مهارت ، چندان اهمیت ندارد .

در دیگر دامنه های ضعیفتر موافق با اصول علمی ، غالباً قاعده ها ضعیفتر هستند و به طور واضح بیان نشده اند . از قوانین موجود در باغبانی یا توصیه سرمایه گذاری سخن به میان نمی آوریم ، با این حال مهارت در این زمینه ها ، آگاهی از قواعد جالب توجه است و روزانه از آنها استفاده می شود . در اینجا سئوال از منابع دانش ضروری تر می باشد.

در زمینه های کاربردی ، امکان ندارد یک تجربه خبره تدوین و نوشته و ثبت شده باشد . ما حقیقتاً به یک خبره زنده نیاز داریم ، زیرا میزان کمی در شکل نوشته شده قابل دسترسی است . تاکنون اکثریت کاربردهای سیستم خبره در این حوزه ها بوده اند . آنها به بسیاری تجربه اندوخته شده ، تکیه دارند ، اما فاقد فرمول علمی می باشند . بزرگترین مشکل در این زمینه ها این است که امکان ندارد خبره توانائی استفاده از دانش را به صورت عملی داشته باشد . و غالباً این موضوعات فراگیری دانش را ، هم برای خبره و هم برای مهندس دانش کاری سخت و دیربازده می سازد .

دانش ، حاوی حقایق ، روالها و قوانین قضائی است و به طور وسیع منتشر شده است . بسیاری موارد مختلف دیگر است . مهارت های انسانی ، افراد با درجه و رتبه ثابت شده ، می توانند انواع مختلفی از اطلاعات را فراهم آورد . ویس۱ و کولیکوسکی۲ (۱۹۸۳)تشخیص داده اند که :

- تجارب شخصی از مسائل حل شده گذشته .
- نظریه یا روشهای شخصی برای حل مسائل .
- دانش شخصی درباره استدلالها برای انتخاب روشهای بکاررفته .

بهر حال ، برای موارد کاربردی سیستم مبتنی بر دانش مهندسی ، این رهیافت ، ناکافی است . مسائل مهندسی معدودی ، کاملاً اکتشافی هستند و منابع دانش و همچنین تخصص بشر که باید تحقیق و رسیدگی شود ، به شرح ذیل می باشد :

- نظریه خبره
- داده مشهور و تاریخی
- کدهای تمرین
- روال های مهندسی
- دادهٔ آزمایشی

---

1 Weiss
2 Kulikowski

- ادبیات تکنیکی
  - کتابهای متنی
  - مجلات
  - دست نویس ها
  - اطلاعات تولید کننده
- معادلات مهندسی بنیاد شده

از اینرو ممکن است دانش نیز به عنوان گامهائی جهت متابعت کردن در حل یک مسئله ، استفاده از معادله و یا داده صحیح و تصحیح استفاده از برنامه های کاربردی تشخیص داده شود .

۳-٤-۲ انواع دانش

سئوال بعدی برای اسباب و لوازم کار تنها این است که ، فراگیری دانش چه چیزی را گردآوری می کند ؟ ما در پی سه نوع مختلف دسته بندی دانش هستیم .

اولین دسته بندی ، ساده ترین آن است ، و با نام دانش ادراکی همراه است . این دسته دانش حقایق ساده و روابطی مانند رنگ بلور های کوارتز ، نقطه ذوب آهن یا ماکزیمم گشتاور یک موتور را پوشش می دهد . اگر بخواهیم دقیق شویم ، نظر به اینکه سیستم های خبره نیازی به نگهداری این بخش ها ندارند ، می توانند پاسخ های سئوالات را به دست آورند . اما عملاً هر سیستمی به دسته بندی کردن چنین اطلاعاتی نیاز دارد .

سطح بعدی که یافته ایم ، چیزی است که اغلب افراد در دانش ملاحظه می کنند : مفاهیم و روابط . در اینجا قوانین علمی از قبیل معادلات نیوتن و قانون بویل و نیز مشاهدات اکتشافی نظیر " تب نشان دهنده بیماری است . " را یافتیم . این سطح ، اصول سیستم های خبره است .

سومین سطح که کاملاً مهم ترین سطح هم می باشد ، وجود دارد . خبره ها در یک مسئله ، نه فقط دانش علمی و تجربیاتشان را ، بلکه از دانش چگونگی شروع کردن یک مسئله ، نحوه مطالعه و روش تحقیق مشکلات ، و اینکه در هنگامی که آنها به هم پیوسته اند چه کاری انجام دهیم ، استفاده می کنند . این می تواند دانش استراتژیک

نامیده شود .
سودمندی یک سیستم خبره می تواند کاملاً به دانش استراتژیک وابسته باشد .
این سه سطح نه تنها قدرت این بخشها را نشان می دهند بلکه نحوه سخت و دشوار نگهداری را نیز نمایان می سازند .

۳-۵ نمایش حقایق

مسئله نمایش حقایق به عدم مطابقت بین بشر و حافظه کامپیوتر مربوط می شود ، مثلاً نحوه رمز گذاری دانش ، به طوری که این بازتابی مناسب از دانش خبره است و می تواند به وسیله کامپیوتر دستکاری شود . تحقیق روانشناسی اشاره بر این مطلب دارد که ما انواع رفتار های استدلالی را نمایش نمی دهیم ، بلکه با سیستم های استقرائی یا سیستم های اثبات قضیه ارتباط می دهیم . انسانها از وضعیت ها برای اعمالی که از نتایج منطقی بهره می برند ، استدلال می کنند .
برنامه نویسی یک کامپیوتر جهت حل مسائل غیر کمی ، با مسائل غیر ملموسی نظیر ایده ها ، مفاهیم و روابطشان ، قابلیت رسمی ارائه و دستکاری موجودیتهای نسبتاً انتزاعی که باید تعبیه و برنامه ریزی شوند ، سرو کار دارد . این کار مستلزم ابزارهای ساختاری و پردازش است که ماورای داده ، متن ، و دستکاری عددی و سیستم های شمارش گرا و زبانهای غالب امروزی می باشد . این مسائل را ساختار های قویتر مبنای دانش ، و عملکرد های دستکاری شده بر روی اصول دانش ، برنامه های موتور استنباط می نامیم .

۳-۵-۱ چه چیزی ارائه می شود ؟

در ابتدا اجازه دهید ، لزوم نوع دانشی را که ممکن است در سیستم های AI ارائه شده باشند ، را مورد بررسی قرار دهیم :
اشیاء : حقایق درباره اشیاء در گستره دنیای ما ، به عنوان مثال ، گیتار سازی دارای سیم است و ترامپت یک آلت موسیقی برنجی است.

.

رخدادها : اعمالی که در دنیای ما رخ می دهد ، به عنوان مثال ، استیو وی۱ در گروه فرانک زاپا۲ گیتار می نوازد.

اجراء : رفتاری شبیه نواختن گیتار با دانشی درباره نحوه اجرای کارها درگیر است .

فرادانش : دانش ، درباره چیزهائی که می شناسیم . مانند ، خواهرم برنامه یک سفر دارد . خواهرم می داند که می تواند علائم خیابان موجود در طول مسیرش را جهت آگاهی از موقعیت آن خیابان ، بخواند. بنابراین در حل مسائل موجود در AI باید دانش و اطلاع ارائه دهیم و دو موجودیتی که با آنها سروکار داریم ، عبارتند از :

حقایق : واقعیت هائی درباره دنیای واقعی و چیزی که ما نمایش می دهیم . حقایق می توانند به عنوان یک سطح دانش مورد توجه قرار گیرد .

ارائه حقایق : آنچه که دستکاری می کنیم . ارائه حقایق می تواند به عنوان سطح نماد مورد اعتنا قرار گیرد ، از اینرو معمولاً نمایش در بخش هائی از نمادها که می توانند به وسیله برنامه ها دستکاری شده باشند ، را تعریف می کنیم .

می توانیم این موجودیتها را در دوسطح پی ریزی و ساختاربندی نمائیم :

سطح دانش ، جائی که حقایق تعریف شده اند .

سطح نماد ، که ارائه اشیاء در بخش هایی از نمادها که می توانند در برنامه ها دستکاری شوند ، تعریف شده اند .

زبان انگلیسی یا زبان طبیعی یک روش واضح از ارائه و بررسی حقایق است . منطق ، ما را در بررسی حقایق زیر توانا می سازد . "اسپوت یک سگ مانند سایر سگها است ." سپس می توانیم از طریق عبارت $\mathrm{dog(x)} \rightarrow \mathrm{hasatail(x)}$ استنباط کنیم که "همه سگها دارای دم هستند ."

بنابراین می توانیم نتیجه بگیریم : که اسپوت دم دارد .

استفاده یک تابع نگاشت بازگشتی مناسب در یک جمله فارسی نظیر" اسپوت دمی دارد که می تواند ایجاد شود ."

---

1 Steve Vai
2 Frank Zappa

توابع در دسترس همیشه یک به یک نیستند ، بلکه چند به چند می باشند که مشخصه ای از ارائه های انگلیسی است .

هر دو جمله " همه سگها دمهائی دارند ." و "هر سگ یک دم دارد. " بیان می کنند که هر سگ یک دم دارد اما اولین جمله را می توان چنین تفسیر کرد که هر سگ بیش از یک دم دارد ؛ این کار را مجدداً با تعویض دم با دندان انجام دهید . وقتی یک برنامه AI یک ارائه داخلی از حقایق را دستکاری می کند ، این ارائه های جدید نیز باید به عنوان ارائه های جدیدی از حقایق قابل تفسیر باشند .

۳-۵-۲ کاربرد دانش

به طور خلاصه جائی که دانش در سیستمهای AI بکار گرفته شده است را ذکر کردیم . اجازه دهید ، قدری بیشتر درباره چگونگی کاربرد ها و چند وچون دانشی که ممکن است استفاده شده باشد ، بررسی کنیم .

یادگیری : به معنای دانش فراگیری است . فراگیری بیش از افزودن یک حقیقت جدید به مبنای دانش است . در این فرآیند دانش جدید به مبنای دانش افزوده شده است و دانشی که پیش تر حاصل شده تصفیه یا اصلاح شده است .

بازیابی : ارائه طرح وبرنامه استفاده شده ، می تواند یک اثر بحرانی روی بازده و کارآرائی یک روش بگذارد . انسانها در این کار خیلی خوب و مناسب هستند .

استدلال : استنباط حقایق از داده های موجود .

اگر یک سیستم تنها بداند که :

رمو۱ یک نوازند ه جاز است .

همه نوازندگان جاز می توانند به خوبی بنوازند .

اگر چیزهائی شبیه : " آیا رمو یک نوازنده جاز است ؟" یا " آیا نوازنده های جاز می توانند به خوبی بنوازند؟" پرسیده شده باشد ، آنگاه پاسخ سئوالات به سهولت از ساختارها و رویه های داده به دست می آید . به هر حال ، سئوالی نظیر " آیا رمو خوب

---

1 Remo

می نوازد ." به استدلال نیاز دارد . همه موارد بالا به هم مربوط هستند . برای مثال ، کاملاً واضح و روشن است که یادگیری و استدلال با بازیابی درگیر هستند .

۳-۵-۳ فرم vs . محتوی دانش

حالتی از افزایش ارقام را در نظر بگیرید . این کار می تواند از طریق ذخیره کردن یک جدول مراجعه ای شامل مجموع همه جفتهای ورودی اعداد صحیح قابل پذیرش ، اجرا گردد . یک شمارنده الکترونیکی می تواند به طور متناوب استفاده شده باشد ، که این شمارنده پی در پی به وسیله دو عدد صحیحی که مجموعشان به دست می آید ، می تواند افزایش یابد . از نقطه نظر تابعی یا محتوائی ، دو رهیافت پاسخهای یکسان ایجاد خواهند کرد . اما از نقطه نظر ارائه ای و نمایشی تفاوت های مهمی وجود دارد که به طور مؤثر بر نحوه توانایی ما در انجام وظایف محوله تاثیر می گذارد . جدول مراجعه ای خواستار سرعت زیاد است ، اما اگر اجباراً با ارقام بزرگ سروکار داشته باشیم ، این سرعت نیازمند نگهداری و ذخیره یک حافظه بزرگ است . شمارنده الکترونیکی ، تمایل به کارآمدی بسیار زیادی در نیازمندیها و الزامات سخت افزاری اش دارد ، اما در ایجاد پاسخهای ملزوم خیلی کندتر است. بنابراین ، ساختارهای خاص به وسیله دانشی که در قالب رمزی شده اش، توصیف شده است ، می تواند اثر مهمی برروی استفاده اش در حل مسائل داشته باشد . به علاوه ، از آنجائیکه هیچ نمایش یا مدل انفرادی نمی تواند همه جنبه های یک شئ واقعی را در برگیرد ، یک موجودیت هوشمند باید طیف گسترده ای از نمایشها و ارائه ها را جهت ارتباط و سروکار داشتن با دنیا ، بکار گیرد .

۳-۵-٤ ارائه دانش

نمایش و ارائه یک وضعیت (یا شئ ، یا مسئله) یک ترجمه و برگردان از آن وضعیت در سیستم دربرگیرنده یک واژه است که اشیا و ارتباطات ، و عملکردها که می توانند روی این اشیاء ، و حقایق و محدودیتهای راجع به این اشیا اجرا شوند ، را نام می نهند . مشخصات تشخیص دهنده اولیه از یک نمایش و ارائه عبارتند از:

الف) چه اطلاعاتی صریح و روشن ساخته شده اند .

ب) چگونه اطلاعات به طور فیزیکی رمز گذاری می شوند .

هدف از نمایش و ارائه ، ساده و مختصر کردن مسئلهٔ پاسخ دهی یک دسته محدود از سئوالات درباره موقعیت داده شده است ، و بنابراین انتخلب یک ارائه باید در مسیر هدف باشد .

حداقل دو ارائه مجزا برای مطابقت صلاحیت و شایستگی چند مکانیزم محاسبه ای داده شده جهت تقاضای حل یک مسئله دنیای واقعی لازم است : اولین ارائه ، لوازم سمبلیک و نمادین کارآمد را برای پاسخ دهی به سئوالات درباره وضعیت داده شده ، فراهم می کند ، و دومی ، تکنیک های راه حل اولی در عملیات و ساختار های ذخیره سازی ذاتی در ماشین را ترجمه می کند .

بنابراین می توانیم در مورد مبانی دانش در بخش های یک نگاشت بین اشیاء و روابط موجود در دامنه یک مسئله و موضوعات محاسباتی و روابط موجود در یک برنامه فکر کنیم . نتایج استنباطها بر روی مبانی دانش باید با نتایج اعمال یا مشاهدات موجود در دنیا مشابه باشد . موضوعات محاسبه ای ، روابط و استنباطهای قابل دسترس ، برای برنامه نویسان به وسیله زبان ارائه دانش که انتخاب می کنند ، تعیین شده اند . یک زبان مناسب می تواند به برنامه ریز ، در اندوختن ، سازماندهی و غلط یابی مبانی دانش کمک کند .

طرح های ارائه بی شماری پیشنهاد و انجام شده اند ، هرکدام نقاط ضعف و قوت خود را دارند . مایلوپولس۱ و لوسک۲ (۱۹۵٤) این طرحها را به چهار گروه رده بندی کرده اند :

الف) طرح نمایش منطقی

ب) طرح نمایش رویه ای

ج) طرح نمایش شبکه

د) طرح نمایش ساخت یافته

طرح های نمایش منطقی

---

1 Mylopoulos

2 Levesque

این دسته از نمایش ها از عبارات موجود در منطق قراردادی برای نمایش دادن یک مبانی دانش بهره می گیرد . قوانین استنباطی و رویه های برهانی این دانش را ، برای نمونه های مسئله بکار می گیرد . اولین ترتیب جبر گزاره ای ، طرح و برنامه های نمایش منطقی است که بیشترین مورد استفاده را در بسیاری از جاها داشته ، اما این فقط یکی از یک عدد از نمایش های منطقی است .(ترنر۱ ۱۹۸٤) : پرولوگ یک زبان برنامه نویسی ایده ال برای اجرای طرحهای نمایش منطقی است .

طرح های نمایش رویه ای

طرحهای رویه ای ، دانش را به مثابه مجموعه ای از دستور العمل ها برای حل یک مسئله نمایش می دهد . این کار با نمایش های اعلانی که به وسیله شبکه های منطق و معنائی فراهم شده اند ، مغایرت دارد . در یک سیستم قانون مند ، به عنوان مثال ، قانون if…then ، ممکن است قانون به عنوان یک رویه برای حل کردن یک هدف در دامنه مسئله تفسیر شود .

برای حل استنتاج ، حل کردن فرض های قبلی اولویت دارد . سیستم های تولیدی ، مثال هائی از طرح های نمایش رویه ای هستند.

طرحهای نمایش شبکه

نمایش های شبکه ای ، دانش را به عنوان یک گراف دربر می گیرد که این گراف در گره هایش موضوعات یا مفاهیم موجود در دامنه مسئله و یالهایش روابط یا وابستگی بین آنها را ارائه می دهد . مثالهائی از نمایش های شبکه شامل ''شبکه های معنائی" و ''وابستگی های ادراکی '' و "گراف های ادراکی" می باشند .

طرح های نمایش ساخت یافته

زبان های نمایش ساخت یافته ، شبکه ها را با پذیرش اینکه هر گره یک ساختار داده ای پیچیده شامل شکافهای نامبرده با مقادیر ضمیمه شده باشد ، توسعه می دهند . این مقادیر ممکن است ارقامی ساده یا داده نمادین ، اشاره گر هائی به دیگر قالبها ، یا حتی رویه هائی برای اجرای وظیفه ای خاص باشند . مثال هائی از نمایشهای ساخت یافته

1 Turner

شامل اسکریپت ها وقالبها۱ و اشیاء۲ است .

۳-٦ طرح های نمایش منطقی

طرحهای نمایش این دسته مشتمل بر جبر گزاره ای و مسندی است . این جبرها ، زبان های نمایش برای AI هستند . هر دوی این گونه جبرها در فصل قبل شرح داده شده اند .

۳-۷ طرحهای نمایش رویه ای

این طرحهای نمایشی چیزهائی را که به عنوان سیستم های خبره قانون گرا نامیده شده اند ، تشکیل می دهد . یک سیستم خبره یک برنامه دانش گرا است که راه حل های "کیفیت خبره " برای مسائل موجود در گستره ای خاص ، را فراهم می کند . مثال ساده ۳,۱ ، از یک مبانی دانش که کاربرد یک ارائه اعلانی غیر متوالی را ثابت می کند ، در زیر نشان داده شده:

مثال ۳-۱

اگر چراغ روشن باشد
آنگاه بهترین گیاه بگونیا است
اگر چراغ کم نور باشد
آنگاه بهترین گیاه پیچک است
اگر نور ، نور خورشید باشد
آنگاه نور ، درخشان است
اگر نور، نور لامپ باشد
آنگاه نور کم سو است
اگر مکان ، مکانی سرباز باشد
آنگاه نور ، نور خورشید است

---

1 Frame
2 Object

اگر مکان سرپوشیده باشد
آنگاه نور ، نور لامپ است

۳-۸ طرحهای نمایش شبکه

تئوری های انجمن ها مفهوم یک موضوع در بخش های یک شبکه از انجمن ها با سایر موضوعات در یک ذهن یا مبانی دانش را تعریف می کند . اگرچه ، نمادها ، موضوعات موجود در دنیا را معنی و مشخص می کنند ، اما این تشخیص و معنی و مفهوم به وسیله ذخیرهٔ دانش مان ، در میان واقع شده است . زمانی که راجع به یک موضوع درک و استدلالی داریم ، در ابتدا آن درک و احساس در مفهوم موجود در ذهن ما نگاشت شده است . این مفهوم قسمتی از دانش تمام و دست نخورده ما از جهان است و با سایر مفاهیم روابطی مناسب برقرار کرده است . این روابط یک فهم و ادراک از ویژگیها و خواص و رفتار یک موضوع مانند "برف " را تشکیل می دهد ، برای مثال به واسطه تجربه ، مفهوم "برف" را با مفاهیم دیگر از قبیل سرما ، سفیدی ، آدم برفی ، لیزی ، و یخ ارتباط می دهیم . درک ما از برف و درستی عباراتی مانند "برف سفید است " به واسطه شبکه انجمن ها ، حقیقت خودش را آشکار و معلوم می کند .

در اینجا دو نوع از طرحهای نمایش شبکه ای که در زیر مطرح شده اند را خواهیم دید :

۳-۸-۱ شبکه های معنائی

شبکه های معنائی ، تناوبی برای منطق گزاره به عنوان یک شکل و فرم از نمایش دانش هستند . این ایده بر این اصل استوار است که ما می توانیم دانش خود را در شکل یک گراف با گره هائی که بیانگر موضوعات در جهان و یالهائی که ارائه دهنده روابط بین آن موضوعات هستند ، بگنجانیم .

گرافها از طریق ایجاد معانی ارائه صریح از روابط مورد استفاده یالها و گره ها ، ثابت کرده اند که رسانه و وسیله ای مناسب برای رسمی کردن نظریه های شرکت کننده دانش هستند . یک شبکه معنائی ، دانش را به عنوان یک گراف با گره هائی متناظر با حقایق یا مفاهیم و یال هائی که بین مفاهیم ارتباط و وابستگی برقرار می کنند ، را ارائه

می دهد . بعضی از اصول شبکه های معنائی (شکل ۳-۳) پیروی می کند از :

- شبکه های معنائی روابط بین اشیاء را که به عنوان گره ها ارائه شده اند ، را وصف و تشریح می کنند .
- این گره ها که نام گذاری شده اند ، مدور هستند .
- ارتباطات بین گره ها به وسیله یالهائی که به این دوایر وصل هستند ، ارائه شده اند .
- یک شبکه معنائی می تواند برای تولید ساختارها و و موضوعات استفاده شده باشد .
- یک شبکه معنائی می تواند جهت تولید قوانین برای یک مبنای دانش استفاده شده باشد .

شبکه معنائی ای که در شکل ۳-۳ نشان داده شده ،خواهان ارائه داده های زیر است :

تام یک گربه است .

تام یک پرنده می گیرد .

تام مال جان است .

تام رنگ تندی دارد .

گربه ها خامه دوست دارند .

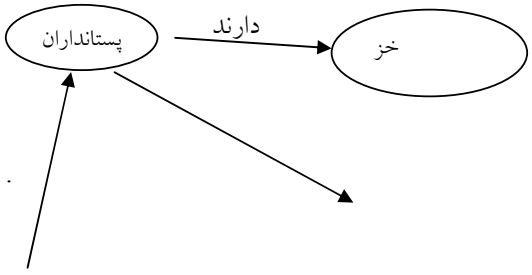
گربه روی پادری نشسته است .

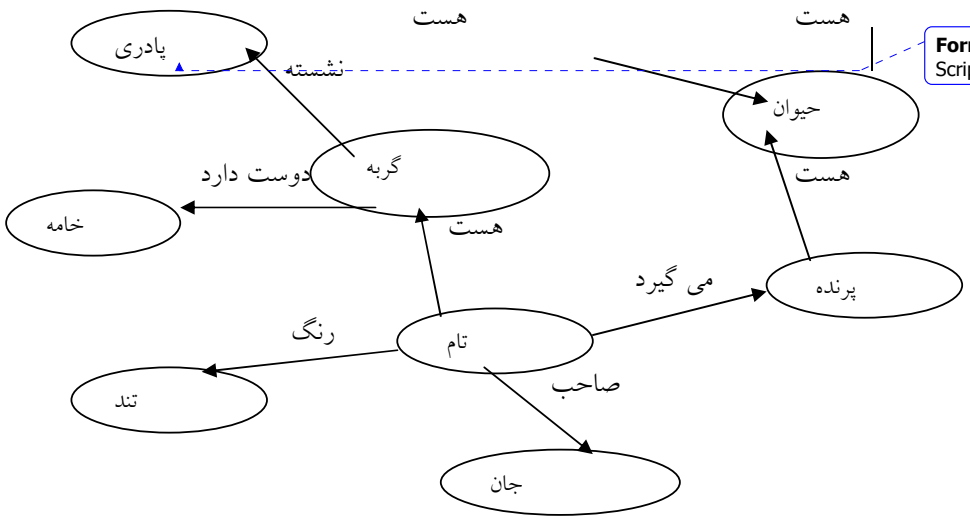
گربه جزو پستانداران است .

پرنده یک حیوان است .

همه پستانداران جزو حیوانات هستند .

پستانداران خز دارند .





Formatted: Font: 12 pt, Complex Script Font: 12 pt

شکل ۳-۳

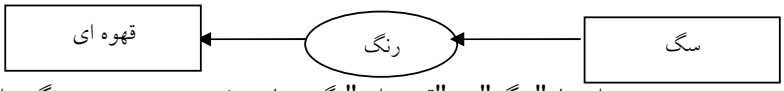
این شکل و قالب از نمایش که بحث شده ، به روش دانش ساختاری بشر نزدیکتر است ، که این دانش ساختاری بشر به وسیله ساختمان ذهنی بین چیز هائی نسبت به منطق گزاره ای که پیش از این مطرح کرده ایم ، پیوند می دهد .

کلمه "هست" یک پیوند است که دو معنی متفاوت دارد . می تواند به این معنی باشد که یک موضوع یک اصطلاح منفرد از یک دسته است ، برای مثال تام عضو دسته گربه هاست ، یا اینکه یک دسته ، یک زیر مجموعه از دیگری است ، برای مثال دسته گربه ها زیر مجموعه دسته پستانداران است . این گیجی و اشتباه گرفتن در منطق رخ نمی دهد .

۳-۸-۲ گرافهای ادراکی

یک گراف ادراکی یک گراف متناهی ، متصل و دوقسمتی است . گره های گراف یا ارتباطات مفهومی یا ادراکی هستند . گرافهای ادراکی از یال های برچسب دار ؛ در عوض گره های رابطه ادراکی که رابطه بین مفاهیم را نمایش می دهند ، استفاده نمی کنند . چرا که گرافهای ادراکی دوقسمتی هستند ، مفاهیم می توانند یالهائی برای روابط

مفهومی و برعکس داشته باشند . مثال زیر را ملاحظه کنید :



در اینجا "سگ" و "قهوه ای" گره های مفهومی هستند و رنگ رابطه مفهومی است . برای تشخیص دادن این نوع گره ها ، مفاهیم به وسیله جعبه ها (مستطیل) و روابط مفهومی به وسیله بیضی ها نمایش داده می شوند . شکل ٤-۳ یک شبکه معنائی را نمایش می دهد که روابط بین اجزاء یک پرنده و یک هواپیما را نشان می دهد .

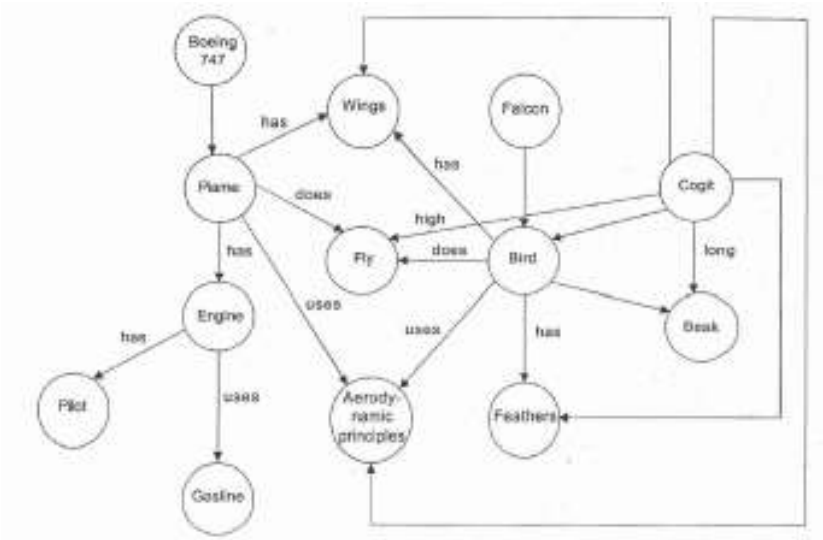


Figure 3.4 Semantic net showing relationships between parts of a bird and a plane.

خصوصیات گراف های ادراکی :

- گره های مفهومی یا موضوعات انتزاعی و یا واقعی از جهان مباحثه را نمایش می دهند .
- گره های ارتباط مفهومی رابطه ای که با یک یا چندین مفهوم درگیر می باشد ، را نشان می دهد .

.

- هر گراف مفهومی یک گزاره منفرد را نمایش می دهد . یک مبانی دانش نوعی محتوی یک شماره ، از چنین گراف هائی خواهد بود . ممکن است ، به طور قراردادی ، گراف ها پیچیده باشند ، اما باید حتماً متناهی باشند .
- نظریه گراف های مفهومی مشتمل بر شماری از عملیات ها است که به ما اجازه می دهد گرافهای جدید را از گرافهای موجود تشکیل دهیم .

عملیات ها روی گراف های ادراکی :

عملیاتی روی گراف های ادراکی ، به ما اجازه ایجاد یک گراف جدید به وسیله تخصیص یافتن یک گراف موجود را می دهد . این کار به وسیله چهار عملیات که کپی کردن ، محدود کردن ، متصل کردن و مختصر کردن نام دارند ، انجام گرفته است .

فرض کنید که $g1$ و $g2$ گراف های مفهومی باشند ، آنگاه :

- قانون کپی به ما اجازه شکل دهی گراف جدید ، G ، که یک کپی عینی از $g1$ است ، را می دهد .
- محدودیت به گره های مفهومی در یک گراف اجازه می دهد تا جایگزین گرهی شوند که تخصص آنها را نمایش می دهد .
- قانون متصل کردن ، به ما اجازه ترکیب دو گراف در قالب یک گراف منفرد را می دهد .
- اگر یک گراف محتوی دو رابطه تکراری باشد ، آنگاه ممکن است یکی از آنها حذف شود . این قانون ، مختصر کردن نام دارد.

۳-۸-۳ وابستگی مفهومی

راجر شانک۱ و دانشجوانش یک نظریه که وابستگی مفهومی نامیده شده ، را به کمک مدعای او مبنی بر اینکه خیلی از داستانها به وسیله فقط تعدادکمی از مفاهیم می توانند نمایش داده شوند ، جهت امتحان و فهم خطوط داستان پایه ای ، توسعه داد . او شماری از سیستم های NL را که تمایل به نگاشت توصیفات انگلیسی داستانهای موجود در ارائه خودش را داشته ، بسط و توسعه داد .

---

1 Roger Schank

جملات مختلف انگلیسی با معنای یکسان ، خواستار این هستند که با نمایشی یکسان نگاشت شوند .

در وابستگی ادراکی (CD) ساختارهای شکاف و پرکننده ، برای نمایش نوعی دانش درباره وقایعی که معمولاً در جملات زبان طبیعی نقل می شوند ، استفاده شده اند . هدف این کار نمایش دانش برای موارد :

- تسهیل کردن رسم استنباطها از جملات .
- مستقل بودن از زبان در جملاتی که به صورت اصلی بیان شده اند .

نظریه وابستگی ادراکی

- برای هر دو جمله ای که در معنی یکسان هستند ، صرف نظر از زبان ، فقط باید یک نمایش وجود داشته باشد .
- هر اطلاعاتی در جمله که تلویحی و ضمنی است باید در نمایش معنی آن جمله به صورت صریح ایجاد شود .

بلوک های ساختمان :

وابستگی ادراکی : ۱۱ عمل اولیه :

ATRANS : انتقال یک رابطه انتزاعی (مانند ارائه دادن)

PTRANS : انتقال موقعیت فیزیکی یک شئ (مانند رفتن)

PROPEL : کار برد نیرو فیزیکی در یک شئ (مانند فشار دادن)

MOVE : جابجائی عضوی از بدن به وسیله صاحب خودش (مانند لگد زدن)

GRASP : نگاه داشتن یک شئ از طریق یک عملگر (مانند کلاچ)

INGEST : بلعیدن یک شئ به وسیله یک حیوان (مانند خوردن)

EXPEL : اخراج و دفع چیزهائی از بدن یک حیوان (مانند گریه کردن)

MTRANS : انتقال اطلاعات ذهنی (مانند گفتن)

MBUILD : درست کردن اطلاعاتی جدید از روی گذشته (مانند تصمیم گرفتن)

SPEAK : تولید اصوات (مانند گفتن)

.

ATTEND : توجه یک اندام حسی به طرف محرک (مانند گوش دادن)

چهار مجموعه مفهومی اولیه برای ایجاد ساختار های وابسته :

ACTs : اعمال

PPs : اشیاء (تولید کننده های تصویر)

AAs : تعدیل کننده های اعمال (عمل aider)

PAs : تعدیل کننده های تصویر (تصویر aider)

وابستگی ها در میان تصورات با روابط معنایی در میان مفاهیم اساسی مشابه هستند .

مزایای اصلی وابستگی مفهومی

شرح قوانین استنتاجی با دانشی که می تواند دستکاری شده باشد ، ساده تر است . قوانین می توانند یک بار برای هر عمل (ACT) سریعتر از تمامی لغاتی که آن عمل (ACT) را توصیف می کند ، نمایش داده شوند . به عنوان مثال دادن ، گرفتن ، دزدیدن و بخشیدن همه نمونه هائی از ATRANS تلقی می شوند و می توانند استنتاج های یکسانی درباره کسی که این شئ را دارد ، و کسی که یکبار آن شئ را داشته است ، ایجاد کنند .

برای ایجاد نمایش CD اطلاعات صریحی که درمتن بیان نشده اند ، را به وجود می آوریم . مانند" Gopal کتاب را از Gita گرفت " ، و اطلاعاتی مانند " گیتا مدت زیادی کتاب را در اختیار نداشته است . " را به طور صریح ایجاد می کنیم . این کار ممکن است ، فهم عبارت بعدی را آسان تر کند . مانند " گیتا چیزی برای خواندن ندارد . "

باید رئوس مطالب یک مختصر سازی از یکی از سیستم های شانک که مکانیزم تقاضا کننده اسکریپت (SAM) نامیده شده ، را تهیه کنیم . این عمل خواهان در اختیار گرفتن نمایشی از یک داستان و ادغام آن با یک اسکریپت استاندارد می باشد که این اسکریپت می تواند برای بیرون آوردن سناریو و ایجاد تفسیری از موقعیت ، استفاده شده باشد . بنابراین ، اینها خواستار فهمی از داستان در اصطلاحات اسکریپت می باشد . آنها معتقدند که نسبتاً به تعداد کمی اسکریپت ها برای فهم خیلی از موقعیت های بشر

نیاز داریم .

شانک حدود ۱۱ عمل (ACTs) پایه ای برای ارائه و نمایش معانی به کار برد . که فقط چهار مورد آنها را مختصر سازی عقاید نامعلوم فهرست وار می آوریم :

۱- (شئ ۱ ، شئ۲ ، مکان ۱، مکان ۲) ptrans :موقعیت یک شئ فیزیکی (شئ۱) را به موقعیت شئ ۲ از مکان ۱ به مکان ۲ تغییر می دهد .

۲- بلعیدن (بازیگر ، غذا ، ثروتمند) : گرفتن چیزهائی از داخل یک شئ جاندار - بازیگر غذا را با الگوبرداری از انسانهای ثروتمند می خورد.

۳- (بازیگر ۱، شئ ، بازیگر ۲ ، بازیگر ۳) atrans ، یک رابطه انتزاعی ازقبیل مالکیت یا تصرف با احترام به یک شئ -بازیگر۱ را از تصرف بازیگر ۲ برای در اختیار قرار دادن بازیگر ۳ تغییر می دهد .

٤- (بازیگر۱ ، info ، بازیگر ۲) mtrans ، اطلاعات را انتقال می دهد و info را از بازیگر ۱ به بازیگر ۲ منتقل می کند .

گذشته از ACTS ، مفاهیم دیگری از قبیل زمان ، موقعیت و خواص نیز وجود دارد .

۳-۹ طرح های نمایش ساخت یافته

چگونه یک کامپیوتر می تواند توجه خودش را فقط بر روی جنبه هائی از مسئله داده شده که به راه حل مربوط است ، معطوف کند ؟ این یک مسئله قالبی است (که به وسیله مارتین مینسکی اختراع شده است) .

Marvin Minsky

.

یک سیستم هوشمند ، در تلاش برای حل یک مسئله یا انجام یک عمل چگونه می تواند دریابد ، که از چه اطلاعاتی در پایگاه داده باید صرف نظر شود و چه اطلاعاتی به مسئله موجود مربوط است ؟

۳-۹-۱ قالب ها

هرچه وظایف پیچیده تر می شوند ، لازم است که ارائه و نمایش آنها ساخت یافته تر باشد . ساختیافته تر بودن سیستم ، برای بکار بردن قالبها سودمندتر می باشد . یک قالب ، مجموعه ای ازخصوصیات یا شکاف ها و مقادیر شرکت کننده می باشد که بعضی موجودیت های دنیای واقعی را توصیف می کند . قالبها به شکل خاصی در ساختار خودشان مفید نیستند ، اما سیستمهای قالب یک روش قدرت مند از رمزگذاری اطلاعات برای پشتیبانی کردن استدلال می باشند . نظریه مجموعه ها یک پایه و اساس خوب و مناسب برای درک سیستم های قالب فراهم می کنند .

دانش را به عنوان نمایش و ارائه های شبکه کاربردی سازمان دهی شده و پیوند یا وابستگی صریح بین اشیاء موجود در اصل دانش در نظر می گیریم . متناوباً می توانیم دانش را در واحد های پیچیده تر که موقعیت ها یا اشیاهای پیچیده در گستره را ارائه می دهند سازمان دهی کنیم . این ها " قالبها " یا "طرحها " نامیده شده اند .

هر قالب منفردی ممکن است به عنوان یک ساختار داده ، همانند خیلی ازرابطه ها برای رکورد سنتی آنها در نظر گرفته شده باشد ، که این رکورد حاوی اطلاعاتی مربوط به ، موجودیت های کلیشه ای می باشد.

شکاف ها درقالب حاوی اطلاعاتی از قبیل موارد زیر هستند :

الف) اطلاعات بازشناسی قالب

ب) رابطه بین یک قالب و قالبهای دیگر . " تلفن هتل " ممکن است نمونه خاصی از تلفن باشد ، یا به شکلی دیگر نمونه ای از یک "دستگاه ارتباطی " باشد .

پ) توصیف کنندگان تقاضاها برای مطابقت قالبها . به طور مثال ، جایگاه یک صندلی ، بین ۲۰ تا ٤۰ سانتیمتر از کف فاصله دارد و قسمت پشتی آن ارتفاعی بیش از ٦۰ سانتیمتر دارد . این تقاضاها ممکن است برای تصمیم گرفتن به کار گرفته شوند ، زمانی

که اشیاء جدید به وسیله قالب ، مناسب کلیشه ها تعریف شده اند .

ت) اطلاعات رویه ای در استفاده از ساختار های توصیف شده . یک خصوصیت مهم قالبها توانائی الصاق کردن کد رویه ای به یک شکاف است .

ث) اطلاعات قراردادی قالب . این اطلاعات مقادیر شکاف هستند که برای صحیح بودن در زمانی که هیچ مدرکی ، مخالف یافت نشده است در نظر گرفته شده اند . به عنوان مثال ، صندلی ها چهار پایه دارند ، تلفن ها دکمه فشاری دارند ، یا تخت های هتل به وسیله کارمندان ساخته شده اند .

ج) اطلاعات نمونه جدید . خیلی از شکاف های قالب ممکن است نامعلوم بمانند تا زمانی که یک مقدار برای یک نمونه خاص داده شده یا برای حل مساله لازم شده باشد . برای مثال رنگ چادر شب تختخواب ممکن است در تعریف تختخواب نامعلوم بماند .

مثال زیر را در نظر بگیرید :

مثال ۲-۳

شخص هست یک : پستاندار

مرد بالغ هست یک : شخص

بازیکن راگبی هست یک : مرد بالغ

قد:

وزن:

پست بازی :

تیم :

رنگ تیمها :

مشکی هست یک : بازیکن راگبی

کسب امتیاز توسط : مایک – هال

نمونه : پشت

.

ارتفاع : ۰-٦

پست : میانی

تیم : کاردیف – آر اف سی

رنگ های تیم : سیاه / آبی

تیم راگبی هست یک : تیم

تعدادتیم : ۱۵

مربی :

در اینجا قالبهای شخص ، مرد بالغ، بازیکن راگبی ، تیم راگبی ، همگی طبقه بندی هستند و قالبهای رابرت- هاولی و کاردیف –آر اف سی نمونه های آنها می باشند .

کاردیف –آر اف سی

نمونه : تیم راگبی

تعداد تیم : ۱۵

مربی : تری هومز

بازیکن ها : { رابرت هاولی ، جویان – جونز ، .....}

بعضی از خصوصیت های مهم قالبها عبارتند از :

- قالب ها شبکه های معنائی را در تعدادی از روش های مهم ، که مهم ترین قالب در مؤلفه های سازمان دهی دانش موجود در ساختارهائی که یک موضوع مهم از مبنای دانش است ، گسترش می دهند .
- الحاق رویه ای ، یک خصوصیت مهم ویژه از قالبها است ، از اینرو ، دانش و اطلاع معین با نمایش ها و ارائه های اعلانی به خوبی وفق داده نشده اند .
- نمایش دادن دانش با سیستم قالب ، اگرچه دارای اطلاعات ناتمام می باشد ، حداقل تاحدی به ما اجازه استدلال کردن و استنتاج سریع حقایقی که به طور صریح مشاهده و آشکار نشده اند را می دهند.
- یکی از مسائل با نمایش قالبی ، دشواری بنیان نهادن مقادیر قراردادی به طور صحیح و دقیق برای یک قالب می باشد .

۳-۹-۲ اسکریپت ها

به وسیله تعریف ، یک اسکریپت یک ارائه ساخت یافته است که یک رشته حوادث و وقایع کلیشه ای از حقایق را در یک محتوای مخصوص توصیف می کند . در اصل اسکریپت به وسیله شانک و گروه تحقیق اش به عنوان سازمان دهی ساختار های وابستگی مفهومی موجود در توصیفات وضعیت های نوعی ، طراحی شده اند . اسکریپت ها در سیستم های ادراکی زبان های طبیعی ، به عنوان بانک اطلاع برحسب وضعیت هائی استفاده شده اند ، که این وضعیت ها سیستمی برای درک کردن ، می باشند .

اجزای یک اسکریپت عبارتند از :

- توصیف گران یا شرایط ورودی از دنیا که باید برای اسکریپتی که فراخوانده شده ، صحیح باشند . در یک اسکریپت رستوران ، اسکریپت ها مشتمل بر رستورانی است که باز است و دارای مشتری گرسنه می باشد .
- نتایج یا حقایق وقتی که اسکریپت خاتمه یافته است ، درست هستند . برای مثال ، آن مشتری گرسنه نیست و پول کمی هم دارد . صاحب رستوران پول بیشتری دارد.
- اثاثیه۱ یا هر چیزی که محتوای اسکریپت را درست می کند . این اثاثیه ها ممکن است شامل میزها ، پیشخدمت ها ، و منو غذا باشد . این حالت اجازه می دهد ، فرض های قراردادی قابل استدلال درباره موقعیت ها داشته باشیم ، برای مثال ، فرض شده است که یک رستوران میز و صندلی دارد مگر اینکه لفظی غیر از این را بیان کرده باشد.
- نقش ها اعمالی هستند که شرکت کنندگان فردی آنها را انجام می دهند . پیشخدمت سفارشها را می گیرند ، غذا تحویل می دهند و صورت حساب را ارائه می نمایند . مشتری سفارش می دهد ، می خورد و صورت حساب را پرداخت می نماید .

---

1 Props

- شانک اسکریپت را به ترتیبی به صحنه۱ ها شکست ، که هر کدام از آنها یک خصوصیت موقتی از اسکریپت را ارائه می دهد. در رستوران ورود ، سفارش و خوردن و ... وجود دارد .

اسکریپت ها در شرح یک موقعیت معین از قبیل "سرقت از بانک " مفید هستند . که ممکن است با موارد زیر درگیر باشند :

- به دست آوردن یک تفنگ .
- سرقت مسلحانه از بانک .
- گریختن از بانک به همراه پول .

اینجا اثاثیه ممکن است شامل موارد زیر باشد:

- تفنگ ، ت .
- غارت ، غ .
- کیف ، ک .
- فرار ماشین ، م .

نقش ها ممکن است شامل موارد زیر باشد :

- سارق ، S
- صندوق دار ، M
- مدیر بانک ، O
- پلیس ، P

شرایط ورودی ممکن است شامل موارد زیر باشد :

- S فقیر است .
- S بینوا است .

نتایج ممکن است شامل موارد زیر باشد :

- S پول بیشتری دارد
- O عصبانی است

---

1 Scene

- M در حالت شوک است
- P گلوله است .

سه صحنه وجود دارد : به دست آوردن تفنگ ، سرقت بانک و گریختن. اسکریپت کامل می تواند همان طور که در جدول دیده شده توصیف شده باشد .

برخی نکات اضافی برای یادداشت روی اسکریپت ها :

- اگر یک اسکریپت خاص به کار برده شده باشد ، باید فعال شده باشد و این فعال بودن به اهمیتش بستگی دارد .
- اگر موضوع در حال عبور ذکر شده باشد ، آنگاه یک اشاره گر می تواند برای آن اسکریپت نگهداری شده باشد .
- اگر موضوع با اهمیت است ، آنگاه اسکریپت باید باز شده باشد .
- دروغهای خطرناک در داشتن تعداد زیادی اسکریپت فعال حتی به اندازه یکی هم ممکن است پنجره های زیاد بازی را با پرده یا فراخوان های بازگشتی بسیار در یک مسئله ، داشته باشد .
- حوادث ایجاد شده ، یک دنباله شناخته شده را دنبال می کند که ما می توانیم از اسکریپت ها برای ارائه اعمال درگیر شده و جهت پاسخ دهی به سئوال های جزئیاتی بکار گیریم .
- دنباله های مختلف ممکن است برای پی آمد های متفاوت از یک اسکریپت مجاز باشد (به عنوان مثال سرقت بانک اشتباه است .)

مزایای اسکریپت ها :

- توانائی جهت پیشگوئی وقایع
- تفسیر منسجم انفرادی ممکن است از مجموعه ای از مشاهدات ساخته شده باشد .

معایب اسکریپت ها :

- عمومیت کمتر نسبت به قالبها .

.

- ممکن است برای ارائه کلیه انواع دانش مناسب نباشد .

در اینجا مثال دیگری از اسکریپت ها (مثال ۳,۳)، در ابتدا به فارسی و سپس در غالب هر فرم قالب ارائه ای داده شده است .

اسکریپت های یک رستوران

۱- مشتری به رستوران می رود .
۲- مشتری به سمت میز می رود .
۳- پیشخدمت غذا می آورد .
٤- مشتری غذا می خورد
۵- مشتری به پیشخدمت پول پرداخت می کند .
٦- مشتری رستوران را ترک می کند.

حال اسکریپت را به عنوان یک اسکریپت تاکیدی PROLOG (نام ، اسکریپت ، قرارداد ها) که نام فقط همان نام اسکریپت ، اسکریپت لیستی از ارتباطات با مقادیر آزاد تشریح اسکریپت در مؤلفه هائی از ACTs می باشد ، و پیش فرض ها یا قراردادها لیستی از زوجها ، هم بسته کردن مقادیر برای نام های پیش فرض می باشند ، اگر آنها به وسیله مثال های واقعی که با زیر مجموعه ای از اسکربپت ها ترکیب می شود معرفی نشده باشد .

اسکریپت(رستوران ، [ptrans(بازیگر ، بازیگر ، مکان اولیه ، رستوران)،
Ptrans(بازیگر، بازیگر، در، میز)،
Ptrans(پیشخدمت ، غذا، آشپزخانه ، میز)،
بلعیدن(بازیگر، غذا، دهان(بازیگر))،
Atrans(بازیگر،پول،بازیگر،پیشخدمت)،
Ptrans(بازیگر، بازیگر، رستوران ،جائی دیگر)]،
[(بازیگر، مشتری)،(مکان اولیه ، مکان ۱)،
(رستوران،رستوران)،(در،در)،
(میز،میز)،(غذا،غذا)،

(پیشخدمت، پیشخدمت)،(آشپزخانه،آشپزخانه)،
(پول،صندوق)،(جائی دیگر،مکان۲)]).

فصل چهارم

استدلال و سیستم های KRR

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

استدلال و برهان

سیستم استدلال و نمایش دانش KKR

زبان های نمایش دانش

مدل سازی دامنه

سیستمهای استدلالی شبکه هاس معنایی (شبکه های شرکت پذیر)

٤-١مقدمه

یک سیستم بانک اطلاعاتی ، در اصل ، سیستمی اخباری(اطلاعاتی) است و بدین لحاظ لزومی ندارد به ترتیب ورود ، ذخیره یا پردازش هر اطلاعی بستگی داشته باشد ، مگر اینکه دلایل خوبی جهت تحمیل پیمانه ای کردن روی اطلاعات وجود داشته باشد. به این معنی که دو مشکل اصلی بر موتور استنباط وجود دارد:

- یک سیستم اطلاعاتی باید روشی جهت تصمیم گیری مکانی برای شروع داشته باشد.قوانین و حقایق در یک پایگاه دانش ایستا مقیم شده است.باید راهی برای پردازش استدلالی جهت آغاز، وجود داشته باشد.
- موتور استنباط باید مجدداً تضادهائی که در هنگام خطوط تناوبی پدید آمدن استدلال رخ می دهد ،را حل کند.برای مثال، این طور می تواند باشد که این

سیستم به نقاطی که چهار یا تعداد بیشتری از قوانین آماده برای برانگیختن دارند، نیل کند.موتور استنباط باید قانونی برای امتحان کردن موارد بعدی انتخاب کند .

٤-۲ استدلال و برهان

استدلال رسمی با دستکاری نحوی و ترکیبی ساختارهای داده ای ، برای استنباط موارد جدید ،برطبق قوانین از پیش تعیین شده ی استنباط درگیر است.استدلال رسمی ،نوعی استدلال است که در منطق پایه ای و قاعده پایه ای سیستم های نمایشی، به طور طبیعی ترین و عمومی ترین حالت، مورد استفاده قرار گرفته است. این استدلال شامل استنباط قوانینی از جمله روش برجسته۱ می باشد. اگر P ، Q را نتیجه دهد و P درست باشد ، آنگاه Q باید درست باشد.

استدلال رویه ای(برای مثال یک برنامه که می تواند مجموع دو عدد را بگیرد) شامل روال های اختصاصی یا رویه هایی برای پاسخ به سوالات و حل مسائل است. امروزه،استدلال رویه ای ، به طور عمومی در قالب پایه ای و سیستم های شبکه معنایی استفاده شده است ،این استدلال برای فضائی کوچکتر در برنامه های قراردادی برای هر پوسته برنامه کالبدی ایجاد شده اند که می توانند یک کلاس از مشکلات مشخص را حل کند.

استدلال به وسیله قیاس با برون یابی (یا استقرا) حقایق جدید از اطلاعات موجود درگیراست.برای مثال ،اگر الیزابت نمره A در درس های ریاضی و شیمی گرفته باشد ، می توانیم حدس بزنیم که شاید در درس فیزیک نیز نمره ی خوبی گرفته باشد.به نظر می رسد که این روش ، سبک طبیعی تفکر برای مردم باشد ، اما تاکنون برای راه اندازی در یک برنامه ی هوش مصنوعی ،دشوار بوده است.اگر این توانایی قادر باشد در یک برنامه با موفقیت اجرا شود ، می خواهد قابلیت وفق دادن و یادگیری آن برنامه را به وسیله خودش داشته باشد.

تعمیم و انتزاع پردازش های استدلالی طبیعی نیز برای بشر که جهت اجرا در یک برنامه دشوار بوده است، می باشد.این محدودهٔ تحقیقی،جهت وجود داشتن در هسته و مرکز

---

1 Modus Ponens

اصلی آموخته ها و علم بشری می باشد.این موضوع با قوانین استقرائی مانند اگر Q برای هر نمونه شناخته شده ، درست باشد و اگر اعداد به اندازه کافی بزرگ از چنین نمونه ها یی وجود داشته باشد آنگاه Q همیشه درست است.اگر بدانیم که سینه سرخ ها بال دارند ، گنجشک ها بال دارند کبوترها بال دارند سرانجام نتیجه خواهیم گرفت که همه ی پرنده ها بال دارند.این یک مثال جهت تعمیم دادن ایجاد چکیده ی بیشترِ قانون عمومی (تعمیم یافته) برای یک مجموعه اطلاعات شامل تعداد زیادی مثال های اختصاصی یا سابقه ها ی وضعیت (کیس) است. برعکس، ممکن است دانش خود رابه طور پی در پی ، از طریق ایجاد مشخصتر و جزئی تر شده به عنوان آموخته ها یمان از مثالهای بیشتر، تصحیح کنیم.استدلال سطح متا ا[1]با بکارگیری اطلاع ،مخصوصاً اطلاع و دانش دربارهٔ وسعت دانش و اطلاعتان و دربارهٔ اهمیت حقایق معین جهت حل یک مشکل درگیر است. تحقیقات اخیر نشان می دهد که استدلال سطح متا نقش اصلی را در درک و شناخت بشر بازی می کند.

٤-٢-١زنجیره ی پیش رو و پس رو

دو سبک اساسی بکار بردن قوانین، برای تعریف دستورالعملی از حرکت به واسطه یک فضای تحقیقاتی یا در شکلی دیگر نمایش یک پردازش حل مشکل ، وجود دارد. این دو سبک از حقیقتی که روش برجسته می تواند در دو روش مختلف بکار برده شود،پیروی می کند.

در ابتدا ،می توانیم با A به عنوان یک داده سروکار داشته باشیم و به مفهوم$A \rightarrow B$ به عنوان عملگر مربوط یا قانون توجه کنیم .در این روش ، داده مسئله را برخلاف سوابق همهٔ قوانین در دسترس تطبیق می دهیم ، مجموعه ای از قوانین قابل اجرا را بیابید، قانون(های) مخصوصی که برانگیخته می شوند، را انتخاب کنید و سپس عملیات دیکته شده توسط پی آمد جدول(ها)ی برانگیخته شده را بگیرید.از این کاربرد درست از روش برجسته ، به طورطبیعی B را نتیجه می گیریم .این نوع استدلال، استدلالی داده-گرا[2] یا زنجیره ی پیش رو است.

روش دوم استدلال از این حقیقت پیروی می کند که توانائی نوشتن یک قانون را داریم که این قانون در آنچه که در ابتدا نتیجه ،بدون اینکه تغییری در معنی قانون ایجاد

---

1 Metalevel
2 Data-driven

کند،ظاهر می شود .این موضوع جانشین نوشتن می شود.

IF **A**
THEN B

می توانیم یک فرمول معادل بنویسیم به صورت

B
IF A

این وارون سازی از قانون اصلیِ، پیشنهاد می کند که نتایج یا اهداف را قبل از نگرانی درباره ی اینکه آیا داده برای پشتیبانی نتایج نشان داده شده در دسترس هست یا خیر، امتحان کنیم.

از اینرو، در روش دوم ، از طریق امتحان کردن جوانب دست راست قوانین برای دیدن آنچه که قوانین به عنوان نتایج شان دارند، شروع می کنیم که امیدواریم اهداف مسائل به دست آیند . برای آن قوانینی که خروجی های مطلوب دارند،B در مورد ساده بالا ، سوابق را امتحان می کنیم ، در اینجا A برای فهمیدن و دیدن چیزی که حقایق برای قادر ساختن این قوانین که برانگیخته شده اند .اگر حقایق مربوطه و مناسب برای راه اندازی این قوانین در دسترس نباشد ، استقرار این حقایق مربوطه را به عنوان زیر هدف بکار می بریم ،و محصولات(نتایج) را به طور بازگشتی در این سبک جستجو می کنیم.این یک استدلال هدف گرا۱ یا زنجیرهٔ پس رو است. از اینرو به B به عنوان هدفی که به دست آمده توجه می کنیم و A داده ای بازبینی شده، یا شاید مستقر شده به عنوان یک زیر هدف در فرآیند تحقیق، می باشد.

چهار عامل بر این سئوال که آیا بهتر است اسدلال و دلیل پیش رو باشد یا پس رو مؤثر است:

- آیا احتمال بیشتری برای حالات شروع یا حالات هدف وجود دارد؟ تمایل داریم از مجموعه ی کوچکتر حالات به مجموعه ی بزرگتر حالات حرکت کنیم(و بنابراین برای یافتن آسانتر است)
- در هر مسیر آیا فاکتور انشعاب بزرگتر است؟(برای مثال میانگین تعداد گره ها که مستقیماً از یک گره ساده می تواند به دست آید). تمایل داریم در جهتی با فاکتور انشعاب پائین تر اقدام کنیم.

---

1 Goal-driven

- آیا از این برنامه برای هم ترازی فرآیند استدلالیش برای یک کاربر درخواست خواهد شد؟ اگر چنین است ، پیش رفتن در دستورالعملی که با روش گمانی کاربر به صورت نزدیکتر رابطه دارد ، مهم است.
- کدام نوع از رویدادها یک فقره حل مشکل را راه اندازی می کند؟اگر این مورد ورود یک واقعیت جدید است، استدلال پیش رو معنی و مفهوم را ایجاد می کند. و اگر جستجوئی برای یک پاسخ مطلوب صورت گیرد ،زنجیره ی پس رو طبیعی تر است.

زنجیره پیش رو:

زنجیرهٔ پیش رو مستلزم ساختن استنتاج ها بوسیلهٔ تطبیق جوانب شرطِ قوانین اگر . . . آنگاه ، با حقایق موجود و دم دست است.این خصوصیتی از استدلال استقرایی (قیاسی) است.تطبیق یک حقیقت با وضعیتی ناقص از یک قاعده به طور قابل استفاده برابر با برطرف کردن دو شرط منطقی است.از آنجا که زنجیرهٔ پیش رو از حقایق برای استنتاج عمل می کند،طریقه استنباط زنجیره پیش رو نیز روش سابق ، روش رویداد گرا[1] ، یا روش داده گرا ، را فراخوانی کرده است .

استدلال در یک سیستم زنجیرهٔ پیش رو ، به عنوان یک چرخه بازشناخت کار[2] تعریف شده است.ابتدا قوانینی که می توانند موفق باشند ،تشخیص و شناخته شده اند، سپس یک قانون انتخاب می شود و نتیجه پایانی(یا عمل) در حافظه کاری اثبات می شوند . سپس سیستم این نتیجه(یا عمل) را به عنوان دادهٔ ابتدایی به کار می برد و به سوی چرخه ی بعدی پیش می رود. استنتاج های ساخته شده همیشه با داده های موجود و بخش های دانش سازگار هستند اما آنها ممکن است بی ارتباط باشند،زیرا امکان ندارد کاربر به نتایجی که منتج شده، علاقمند باشد .

رویهٔ استنتاج زنجیره پیش رو با یک مجموعه از حقایق و قوانین شروع می شود و در یک روش تکراری برای یافتن قانون یا قوانین موجود تلاش می کند که این وضعیت مکان این قوانین با یک حقیقت مطابقت می کند (شاید با یک جانشینی متغیر یا ملزومات اجباری).از این تطابق، حقایق جدیدی مشتق شده (استنتاج شده)است. جمع کردن حقایق جدید و جستجو برای تطابق قوانین را تا زمانی ادامه می دهیم که قوانین

---

1 Event-driven
2 Recognize-act

بیشتری بکار نرود(قوانین تمام شود)، یا مسئله حل شده باشد و حالت هدف حاصل شده است.وقتی بیش از یک قانون با یک حقیقت مطابقت دارد،یک وضعیت تضاد ممکن ،ایجاد شده است که باید مقرر شده باشد. باید در مورد اینکه،در ابتدا کدام قانون را با الزامات متغیر مربوطه اش و حقایق جدید حاصل شده اش به کار بریم،تصمیم بگیریم؛ و سپس باید تعیین کنیم که آیا باید برای تطبیق دادن تنها حقیقت جدید مشتق شده یا یکی از تطابق های قدیمی تر که به طور موقت زمان رسیدگی اش را عقب انداخته ایم تلاش کنیم.

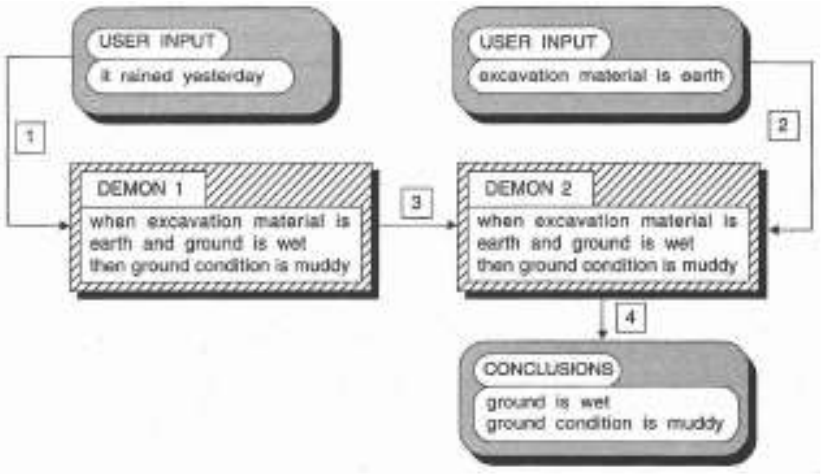
پردازش داده گرا، چیزی را که به عنوان demons شناخته شده است،بکار می گیر یک Demon رویه ای است که به یک عنصر داده ای مربوط شده است،هر زمانی که موقعیت مربوط به آن داده درست باشد،demon عملکرد پردازشی مناسبی را انجام می دهد.

شکل ٤-١ مثالی از یک رویه زنجیرهٔ پیش رو را از میان یک بانک اطلاع ساده نشان می دهد.قانون ۱و۲ در شکل به ترتیب به عنوان demon1 و demon2 ارائه شده اند(یک demon از طریق کلمه کلیدی "وقتی" متمایز می شود.)

دو قسمت اطلاعات اصلی آماده شده است:

"دیروز باران بارید."

"زمین حاوی منابع اکتشافی است."



شکل ٤-١ زنجیرهٔ پیش رو

از ادعای "دیروز باران بارید" موتور استنباط demon1 را برمی انگیزد و نتیجه "زمین خیس است " را استنباط می کند.این مورد در ترکیب عطفی با دومین قسمت از اطلاعات اصلی برای بر انگیختن demon2 استفاده شده است و استنباط می کند که " زمین در وضعیت گل آلود است ". سپس موتور استنباط دو نتیجه پایانی را به کاربر گزارش می دهد.سیستم های زنجیرهٔ پیش رو تعدادی معایب مهم دارند. برای یک مورد آنها فاقد کانون و مرکز توجه هستند.از آنجا که این موضوع داده گرا است، یک سیستم قاعده زنجیرهٔ پیش رو به پی بردن و اکتشاف تمامی نتایج منطقی از مجموعه حقایق و اطلاع تحت رسیدگی،گرایش دارد چه درخواست شده باشد یا نشده باشد. به علاوه، اگر مشخصات مسئله این چنین باشند که حقایق یا مدارک،توانائی راه اندازی تعداد زیاد و متمایزی از قوانین قابل اجرا را داشته باشند،در ابتدا درخت جستجو چند شاخه ای است ومسئله رفع ناسازگاری می تواند از سنگینی درخت جستجو بکاهد.این امکان وجود دارد که چرخه های کامپیوتری بی شمار اتلاف سیستم را که حقایقی را مشتق می کند، بیا بیم که این حقایق هیچ ارتباطی به مسئله تحت بررسی ندارند.

استراتژی زنجیرهٔ پیش رو مخصوصاً در موقعیتی که داده محدود شده است و جمع آوری آنها پرهزینه است، مناسب می باشد.قلمرو های نوعی برنامه ریزی مالی،کنترل پردازش،پیکربندی سیستم های پیچیده و میزان سازی سیستم هستند.

زنجیرهٔ پس رو:

سیستم های زنجیرهٔ پس رو به ایجاد کانون توجه هدف گرا که در سیستم های زنجیرهٔ پیش رو ناپیدا هستند ،گرایش دارد.زنجیره های پس رو از فرضیه ها برای حقایق یا مدارک عمل می کنند و به این دلیل روش استنتاج زنجیرهٔ پس رو به طور متناوب روش نتیجه بخش،هدف گرا،و فرضیه گرا خوانده شده است.زنجیرهٔ پس رو با فرضیه ای اثبات نشده،شروع می شود و تلاشی برای اثبات آن فرضیه به وسیله یافتن قوانینی که فرضیه را نشان می دهند،می کند و سپس برای بازبینی حقایقی که قانون می تواند به عمل بیاورد تلاش می کند.

بنابراین،فرضیه ای که می خواهد اثبات شود با زیر اهداف آن جایگزین می شود اگر(زیر اهداف) اثبات شد قضیه اثبات می شود.اولین مرحله شامل جستجوی مکان نتایج به سمت پایین از یک لیست قوانین و تشخیص یکی از آن ها به همراه متغیر های مناسب، جانشین ها، یا الزامات می باشد که با فرضیهٔ اثبات نشده مطابقت دارد. مرحلهٔ بعدی تلاش برای بررسی حقایق در موقعیت مکانی است که هر قانون را قابل اجرا می سازد. موقعیت مکانی معمولاً شامل مجموعه ای مرکب از یک زیر هدف یا بیشتر می باشد. اگر این زیراهداف با حقایق شناخته شده منطبق نشده باشند،آنگاه آن زیر اهداف برای مرور مجدد لیست قوانین به کار برده می شوند،که این بار برای جستجوی حقایق یا قوانینی که نتایج پایانی اش با زیر اهداف مطابقت می کند،می باشد.

در این روش تکرار حقایق و قوانین تا زمانی ادامه پیدا می کند که یک خط کامل از استدلال بازشناخته شده باشد،که اتصالات یا زنجیره های فرضیهٔ اثبات نشده،کاملاً به زیر اهداف برمی گردد،تا آنجا که در نهایت به تعدادی حقایق یا مدارک شناخته شده متصل می شوند.

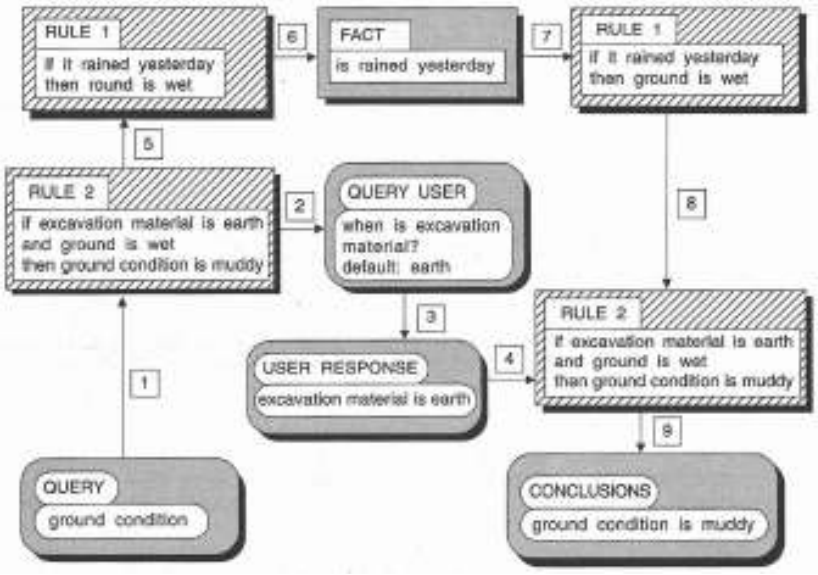
همچنانکه این موردی با زنجیرهٔ پیش رو است،این پردازش تطبیق قوانین با جایگزین های متغیر به طور مؤثر با رفع دو عبارات منطقی معادل است. زنجیرهٔ پس رو این امتیاز را دارد که استنتاج های غیرضروری را انجام نمی دهد.زنجیرهٔ پس رو وقتی که یک نیاز، جستجو یا هدف ابراز شده،نمایان و برجسته شده باشد به صورت تک کاره(فاقد عمومیت) عمل می کند.

شکل ٤-۲ مثالی از یک رویه زنجیرهٔ پس رو از میان یک بانک اطلاعی ساده را نشان می دهد.موتور استنتاج در ابتدا با هدف "وضعیت زمین هست؟ " شروع می کند که

سیستم تمامی قوانینی که استنتاجی دربارهٔ هدف را ایجاد می کند،دوباره به دست می آورد.این مورد در مثال ساده شده قانون ۲ انتخاب شده است.

اگر قانون بتواند برانگیخته شود،آنگاه هر وضعیت در قسمت مقدم قانون برای فهمیدن، ارزیابی می شود.سیستم کاربری را که به "زمین حاوی منابع اکتشافی است" پاسخ می دهد را جستجو می کند.اگر کاربر پاسخ" ناشناخته"را ارائه داده باشد،سیستم مقدار پیش فرض را در نظر می گیرد که به عنوان "زمین" مشخص شده است.شرایط به نوبت ارزیابی می شوند و شرط بعدی قانون۲ "زمین مرطوب است"در زنجیرهٔ موتور استنباط، موجود در نتایج پایانی قانون۱ نتیجه می دهد. درستی بخش مقدم قانون۱ که ادعا می کند"دیروز باران آمد"از طریق یک حقیقت بانک اطلاعی ثابت می شود.

سپس موتور استنباط از این اطلاعات برای پیگرد کامل قوانینی که به ترتیب برانگیخته می شوند استفاده می کند،همانطور که درسمت راست شکل قبلی نشان داده شده است. این نتایج در سیستم پاسخی از "وضعیت زمین گل آلود است " فراهم می آورد.



شکل٤-۲زنجیرهٔ پس رو

عموماً مکانیسم های زنجیرهٔ پس رو در جایی که کمیت داده به صورت بالقوه خیلی

بزرگ است،و نیز در جایی که تعدادی خصوصیت مشخص از سیستم تحت رسیدگی سودمند و مورد توجه است،استفاده شده اند.موارد کاربردی نوعی،شامل تشخیص پزشکی و عیبجوئی می باشد،و بیشتر پوسته های سیستم خبره به تعدادی شکل های زنجیرهٔ پس رو تکیه می کنند.

زنجیرهٔ پس رو همانند زنجیرهٔ پیش رو،فاقد مزاحمت نیست.زیراهداف می توانند از نظر تعداد و پیچیدگی زیاد شوند،ایجاد اثبات آخرین هدف بسیار مشکل و وقت گیر است.به علاوه،وقتی که برای یافتن یک قانون که نتیجه اش با یک هدف یا زیر هدف مطابقت می کند، تلاش می کنیم،ممکن است بیش از یک امکان موجود بیا بیم که ما را دوباره در موقعیت تضاد قرار می دهد.در ابتدا کدام قانون و زیر اهداف نتیجه ای آن قانون را باید بیازماییم؟ بیشتر احتمال دارد که کدام یک به نتیجه ای مفید منجر شود و کدام یک به بن بست برسد؟ انبوه ترین درخت با هدف به عنوان نقطهٔ آغازینش، بدترین مشکل بیان می شود.سیستم های زنجیرهٔ پس روی کاربردی تمامی مجموعه های مفاهیم سازمان دهی شده را برای کمک جهت غلبه براین مشکلات به کار می گیرد.

سیستم های ترکیبی:

تفاوت توانائی ها و ضعف های زنجیرهٔ پیش رو و پس رو میل به تلاش برای کامل کردن آنها در برخی دسته های جستجوی دوسویه (هدف دار و داده گرا) مرکب از بهترین هر دو عالم را افزایش داده است.در ترکیب هر دو روش زنجیره ای،ممکن است دادهٔ ورودی مبهم یا بی اعتبار مسیرهای ممکن بسیار زیادی را برای استدلال زنجیرهٔ پیش روی محض و فرضیه های بسیار زیادی برای زنجیرهٔ پس روی محض تولید کند.به هر حال،زنجیرهٔ پس رو می تواند برای کنترل رشد نمائی احتمالات و اجتناب از مجموعه بی نیاز از حقایق و نتایج وقتی که درخت جستجوی زنجیرهٔ پیش رو به حد افراط شروع به پهنای خروجی می کند، استفاده شود.توجه و تمرکز بر روی زنجیرهٔ پس رو می تواند برای پشگویی های محتمل با یاری گرفتن از زنجیره پیش رو افزایش یابد.

بعضی اوقات جوانب خاصی از یک مسئله، از طریق زنجیره پیش رو و دیگر جوانب به وسیله زنجیرهٔ پس رو است به بهترین حالت بکار برده شده اند.برنامه تشخیص پزشکی

زنجیرهٔ پیش رو را در نظر بگیرید.این برنامه ممکن است بیست یا تعداد بیشتری از حقایق دربارهٔ وضعیت بیمار را بپذیرد،سپس زنجیرهٔ پیش رو روی آن حقایق برای کاهش نوع وطبیعت و یا علت بیماری تلاش می کند. حال در نظر می گیریم که در برخی نقاط، قسمت چپ یک قانون نزدیک به گفتار قانع کننده And/or (ناخوشی) بوده، نه خروجی از ده پیش موقعیت های آن ملاقات شد(بررسی شد).این موضوع ممکن است برای به کار بردن استدلال پس رو جهت قانع کردن دهمین پیش موقعیت در روش مستقیم موثر باشد که از انتظار برای ذنجیره ی پیش رو جهت تهیه ی حقیقت از طریق تصادف (اتفاق) سریعتر است.یا شاید موقعیت دهم به امتحان(تست) های درمانی بیشتری نیاز دارد.در این صورت ذنجیره ی پس رو می تواند برای پرس و جو از کاربر به کار رود.

٤-٣ سیستم استدلال و نمایش دانش KKR

امروزه محققین بر روی روش های محاسبه ای نظری و عملی برای نمایش دادن و به کار بردن معانی زبان طبیعی تمرکز یافتند.برخی زبان های طبیعی رانده شده به صورت مدل های محاسبه ای اجازه میدهد پیشرفت صادقانه هوشمند باشد.سیستم های کامپیوتری زبان طبیعی را در اطلاعات بشری و پردازش دانش شبیه سازی می کند.توسعه ی سیستم های نمایش معلومات و استدلال بر پایه ی زبان طبیعی است که یک روند جدید در هوش مصنوعی ایجاد می کند. سیستم های نمایش معلومات و استدلال مبتنی بر زبان طبیعی و اجراهای در مقیاس بزرگ آنها به دقت مانند محاسبه ی ماشینی نمایشی و استدلالی زبان طبیعی است که متفاوت با هر سیستم نمایش معلومات نظری و عملی و هر سیستم استدلالی یا سیستم های استدلالی خودکار دیگر است.

این راه جدید سیستم های نمایش معلومات و استدلال مبتنی بر زبان طبیعی به رسیدگی مؤثر در مقیاس بزرگ ،دانش چند منظوره ، استدلال و معنای زبان طبیعی بستگی دارد (وابسته است).اما اکثریت سیستم های نمایش معلومات و استدلال به طور مناسب مشخصات مهم زبان طبیعی را منعکس نمی کند و به طور استنباطی و نمایشی وابستگی(رابطه) ضعیفی با زبان طبیعی دارد.به طور تاریخی هدف نمایش معلومات و استدلال، آدرس دهی نمایشی و استنتاجی نیازهای پردازش زبان طبیعی است.اگرچه هدف نمایش معلومات و استدلال به سرعت و اساساً از این(موضوع) فاصله گرفته.در

چندین سال گذشته تعداد کمی تعامل و تمایل بین بیان معلومات و مجموعه پردازش های زبان طبیعی وجود داشته است.جستجوی مستقیم زبان طبیعی بر پایه ی سیستم های بیان معلومات و استدلال یک تغییر شگرف در چگونگی مشاهده ی نقش زبان طبیعی در یک سیستم کامپیوتری هوشمند ایجاد می کند.در بسیاری از جاها،چشم انداز و مشاهدات سنتی در AI و مجموعه های زبان شنا سی محاسبه ای نگهداری شده،زبان طبیعی را به عنوان یک راه فاصل یا خط مقدم برای یک سیستم همانند یک سیستم خبره یا پایگاه دانش د نظر بگیرید.در این دیدگاه،وظایف استنتاج و سایر اطلاعات جذاب و پردازش دانش قسمتی از پردازش زبان طبیعی نیست.از طریق تباین،مدل محاسبه ای زبان طبیعی،زبان طبیعی را به عنوان یک سیستم نمایش معلومات و استدلال به همراه ماشین استنتاج نمایشی محاسبه ای جذاب خاص خودش نمایش می دهد. این نظریه تعدادی حقایق را آشکار می سازد ، مقدار زیادی حقایق ناشناخته از رابطهٔ بین زبان طبیعی و ذهن بشر را روشن می کند.این روند تعیین می کند که ساختار ذهن بشرنیز نسبت به زبان طبیعی بسته است و زبان طبیعی ضرورتاً زبان ناشی از تفکر بشر است. در سیستم های استدلال و نمایش دانش بر مبنای زبان طبیعی،دانش و اطلاعات (۱)در سیستم های کامپیوتری از طریق زبان طبیعی در شکل متن یا گفتگو ثبت و وارد شده اند(۲) به وسیله الگوریتم ها و شبیه سازی دقیق جملات و معانی زبان طبیعی ارائه و ترکیب شده اند(۳) دربارهٔ مکانیسم های استنتاج،استنتاج های شبیه سازی کردن دقیق که بشر در زبان طبیعی ایجاد کرده،استدلال کرده اند،و(٤) به وسیله زبان طبیعی در شکل پاسخ های زبان طبیعی به پرس وجوها از سیستم های کامپیوتری خارج شده اند.این به معنای آن است که همهٔ وظایف پردازش اطلاع و دانش ، رفتار هوشمند سیستم کامپیوتری که در سطح زبان طبیعی رخ می دهد،پشتیبانی می کند. مزیت زبان طبیعی مشابه یکنواختی نمایش و استدلال،یک معماری کامپیوتری ساده و قوی است.برای محدوده هایی که به خوبی رسمی نشده است ، می توان این طور بیان کرد که استدلال و دانش مبتنی بر زبان طبیعی سودمند تر از نمایش استدلال و دانشی است که از طریق زبان طبیعی برانگیخته و تحریک نشده اندعی است.

معمای طبیعی محاسبه ای مربوط به زبان طبیعی فقط به صورت الگوریتمی نیست اما ساختمان الگوریتمی آن با شکل بسیار متفاوتی با زمینه ی تراکم تعداد محاسبه کامپیوتر

سنتی و تراکم نشانه ی محاسبه هوش مصنوعی سنتی به نظر می رسد.
یک توافق همه جا گسترده(عمومی) وجود دارد که برنامه های هوش مصنوعی نباید متن زبان کاملاً طبیعی را به عنوان یک زبان نمایش معلومات به کار برند.زبان طبیعی نمی تواند به آسانی به صورت الگوریتمی اداره شود. که پر از ابهام است. این مفهوم وابسته به متن است. این جمله به شدت وابستگی پیچیده و قوی به معانی دارد.مرتبط کننده ها(موضوع(قضیه) ها ، حروف تعریف(شرط ها)،حروف عطف) به طور قابل ملاحظه مبهم و غیر اصولی هستند.تعداد کمی روش های قوی برای استدلال رشته های زبان طبیعی وجود دارد.زبان طبیعی عمل با استنتاج را کم انجام می دهد.زبان در سطح تفکر است.یک دانش پایه ای بشر در جهان به صورت زبان شنا سی در ذهن بیان نشده است.زبان برای ارتباط با مردمی است که نمی توانند چشمان شما را تشخیص دهند یا نمی توانند استدلال شما را منتشر کنند.بنابراین نمی توان تفکر را به طور مستقیم بیان کرد.

تقسیم واقعی تلاش به صورت نمایشی و استنتاجی بین زبان طبیعی و سیستم های دیگری مانند ذهن بشر، پایگاه داده یا دانش پایه ،)سوال ، که به طور وسیع وارسی شده است را باقی می گذارد.همچنین نمایش استدلال و استفاده انواعopenیک جستجوی باز دانش از ورودی های بسیار گسترده ی زبان طبیعی مهم هستند .در روش سنتی، زبان طبیعی یک نقش نمایشی و استنتاجی کمتری بازی می کند و همچنین سیستم پردازش زبان طبیعی با هیچ زبان طبیعی انگیخته شده توسط سیستم پردازش زبان طبیعی، پشتیبانی نمی شود.

دیگر روش های جالب توجه ، کسب معلومات اتوماتیک از ورودی های زبان طبیعی در مقیاس متوسط تا (support) بزرگ است.این روش های فراگیری بر پایه ی زبان طبیعی است و جنبه های متنوع الگوریتم آنرا متحد می کند.

آنها استعداد بالقوه ای برای جایگزینی دانش دستی برای مدارک متنی به جای یک پردازش خطا دار و گران دارند.این روش ها امید به اینکه_ ما نیاز بیشتری به بشر، برای فرستادن زباله ی اکتسابی سیستم که آن را بازرسی کرده و از بین می برد، نداشته باشیم_ را افزایش می دهد.جستجو گر های پردازش زبان طبیعی و شاغل ها به طور ثابت با نامناسبی ساختار های داده جهت نمایش زبان طبیعی روبرو هستند زیرا معانی و جمله ی آنها به طور مناسب جمله و معانی زبان طبیعی را نمی گنجاند.استنتاج در زبان

طبیعی به وسیله ی الگوریتم های استدلال سنتی به خوبی شبیه سازی نمی شود.این امر فعالیت پردازش زبان طبیعی لازم برای آگاهی یابی اتوماتیک را بسیار پیچیده می کند.
سیستم های استدلال و نمایش معلومات بر پایه ی زبان طبیعی، یک ترکیب منحصر به فرد با کیفیت شبیه انسانی از پردازش دانش با سود های کم کامپیوتر را می پذیرد. این سیستم ها به کامپیوترها اجازه می دهد که دانش در فرم زبان طبیعی مشابه با مردم را پردازش کند و سپس آنرا با کیفیت های کامپیوتری غیر انسانی مشابه حافظه غیر محدود واقعی، معنایی و سریع ترکیب می کند. فقط چنین ترکیب های انسانی و غیر انسانی از اطلاعات و پردازش دانش ، برای یک پردازش(جستجو) عمقی از اطلاعات و دانش در حجم بزرگی از ورودی های زبان طبیعی امیدواری می دهد.چنین حجم بالایی شامل مدارکی مانند متون و دیالوگ رونویسی شده است که بر روی شبکه ی ارتباطی جهانی در دسترس است. این افزایش حجم پایدار به طور جاری به صد مدرک متنی تخمین زده می شود.

٤-۳-۱ملاقات مجدد نمایش حقایق

یک نمایش حقایق چیست؟این ایده می تواند به بهترین نحو بر حسب نقش های مجزایی که بازی می کند،و هر شرایط سخت برای انجام وظیفه موجود دریافت شود.یک نمایش حقایق(KR) به طور کاملاً اساسی یک جانشینی است،و این جانشینی برای چیزی از خودش می باشد.این کار برای توانا ساختن یک موجودیت برای تعیین نتایج از طریق تفکر نسبت به ایفای نقش استفاده شده است،برای مثال از طریق استدلال دربارۀ جهان بیشتر از عمل کردن در آن به کار می رود.این یک مجموعه از الزامات وابسته به هستی شناسی است،برای مثال پاسخی به یک سوال.که(آن سوال) این است که در چه شرایطی باید دربارۀ جهان تفکر کنم؟
این یک تئوری تکه تکه شده از استدلال هوشمند است که در سه جزء شرط بیان می شود:۱)مفهوم اساسی نمایش استدلال هوشمند۲)مجموعه استنتاج های نمایش قوانین و۳) مجموعه استنتاج هایی که آن بیان می کند.
این یک حد متوسط برای محاسبه ی موثر مستدل است یعنی محیط محاسبه ای، در آنچه فکر کردن انجام می شود.یک سهم برای این کارایی مستدل به وسیله ی راه نمایی یک نمایش جهت سازمان دهی اطلاعات برای آسان کردن ساخت، تهیه می

شود.استدلال های توصیه شده.این یک مقدار متوسط از عبارت انسانی است یعنی یک زبان در آنچه درباره ی جهان گفته می شود.

نمایش حقایق(KR) به عناوین عمومی از چگونگی اطلاعات که می توانند به طور مناسب رمز گذاری شوند و در مدلهای محاسباتی ادراکی مورد استفاده قرار گرفته،اشاره می کند.این نسبتاً یک زمینه گسترده با پیوستن به منطق ، زمینهٔ کامپیوتر، روانشناسی شناختی و ادراکی ،زبان شناسی،و دیگر قسمت های زمینهٔ ادراکی است.بعضی نمایش حقایق (KR) بر اهداف معقول در روان شناسی یا زبان شناسی کار می کنند،اما تعداد زیادی،بیشتر از طریق ارتباطات مهندسی،و محرکی که در تمام فیلدهای AI فعال می شود،برانگیخته و تحریک شده است.کار KR به طور نوعی،موضوعات صرفاً فلسفی را نادیده می گیرد،اما با محدوده هائی در فلسفه که شامل تجزیه و تحلیل نمایش ذهنی ،استدلال استقرایی و "زبان تفکر" فلسفه زبان ،و منطق فلسفی می باشد، مرتبط است.

٤-۳-۲ استدلال

استدلال یک پردازش است که به صورت داخلی می رود،اما آن می خواهد فقط به صورت خارجی درباره ی وجود داشتن (زیستن) استدلال بیاورد.ممکن است توسط یک برنامه (شخص) در برنامه نویسی اسمبلی یک دوچرخه بکار برده شود، و سیستم ممکن است مجبور شود درباره ی موجودیت هایی مانند چرخ ها، ذنجیر ها،چرخ دنده ها،و فرمان ها استدلال بیاورد،اما چنین چیزهایی فقط در جهان خارجی وجود دارند. این دوگانگی اجتناب ناپذیر یک اصل اساسی و ابتدایی و نقشی برای نمایش یک است.این به عنوان یک جانشینی داخلی استدلال کننده، یک جانشین برای چیزهایی که در جهان وجود دارد،عمل می کند.عملکرد ها روی نمایش ها و با نمایش ها ، به جای عملکرد های روی چیزهای حقیقی جانشین می کند، آن جانشینی به جای رابطه ی مستقیم با جهان است.در این دیدگاه استدلال خودش است،در یک قسمت(اندازه) ،یک جانشینی به جای عمل است در جهان وقتی که ما نمی توانیم یا هنوز نمی خواهیم آن عمل را انجام دهیم.

اکتساب دانش یک صورت در ساختمان دانش بر پایه ی سیستم هوشمند است.سیستم های بر پایه ی دانش یک نوع برنامه ی کامپیوتر است که دانش تخصصی(فنی) را ، یا مهارت برای حل مشکل را می پذیرد.اکتساب دانش شامل دانش تخصصی مناسب، ثبت آن و تبدیل آن به شکل محاسبه پذیر است،بنابراین ماشین حل مشکل از سیستم

هوشمند می تواند آنرا بپذیرد.اکتساب دانش مهمترین بخش از ساختمان و نگهداری از سیستم های هوشمند است.

سیستم های بر مبنای دانش (KBS) یک زیر زمینه از هوش مصنوعی هستند که با تولید برنامه هائی که نظریه استدلال کارشناسان انسانی را در بر می گیرد،ارتباط دارد.در شرایط ساده،تمام معانی،شکلی از تولید مثل عقلانی است،و این برای یافتن اشخاصی با مهارت استدلال می باشد که این مهارت مهم و کمیاب است.(برای مثال، یک تشخیص دهنده بیماری خبره،بازی کنندهٔ شطرنج،و داروساز (شیمیدان))،صحبت کردن با آنها برای تعیین اینکه آنها چه دانش اختصاصی دارند و چگونه استدلال می کنند،می باشد،و سپس آن دانش و استدلال را در یک برنامه تجسم می کنیم.

نمایش حقایق (KR) به عنوان یکی از عناصر اصلی از هوش مصنوعی و قسمت انتقادی تمامی حل مسائل،در نظر گرفته شده است.زیرفیلدهای KR از جنبه های صرفاً فلسفی معرفت شناسی،برای مسائل کاربردی تر ادارهٔ میزان زیادی از داده تغییر می کند.این گوناگونی به وسیلهٔ مشکل مرکزی رمزگذاری دانش بشری در همهٔ شکل های متنوع در چنین روشی که دانش می تواند به کار رود، یک شده است.

هر پردازش هوشمند تجسم شده شامل اجزاء دستورالعمل خواهد بود، که ما به عنوان مشاهده کننده ی وسیع به طور طبیعی ، به نمایش یک مقدار گزاره ای از دانش می پردازیم، که پردازش شفاهی را نمایش می دهد، و جدا از چنین تخصیص معنایی خارجی ،یک نقش قراردادی اما سببی و ضروری در تهیه ی حرکتی که آن دانش را آشکار می کند ،را بازی می کند.

یک نمایش موفقیت آمیز از مقدار کمی از دانش سپس باید در یک شکل که توسط بشر قابل فهم است و باید سیستم را سبب شود، دانش را برای رفتار کردن به کار ببرد اگر آنرا بشناسد." اجزاء ساختاری" که این اهداف را انجام می دهد، به صورت نوعی در یافت می شوند، به هر دو(صورت) اجرایی و نظری، در کل سال ها گسترش یافته اند.KR زبان ها برای برنامه نویسی یک پردازش از نمایش دانش است.حالت متوسط استفاده شده،برای ارائه،و زبان های برنامه نویسی، پیاده سازی ضوابط به وسیلهٔ فرضیهٔ KR به خوبی راضی کننده نیست.این برنامه نویسی ها بر رفتار که با حقایق نمایش داده شده سازگار است اثر می گذارد،اما آنها برای درک و استنباط به دشواری مشهورند.

٤-٤زبان های نمایش حقایقKR

ویلیام وودز۱ تعریف کرد که مشخصات یک زبان KR عبارتند از:

یک زبان KR باید به طور روشن هر تفسیری از یک جمله(شایستگی منطقی) را نمایش دهد،یک روش برای ترجمه از زبان طبیعی برای آن نمایش دارد وباید برای استدلال قابل استفاده باشد.

تعریف وودز صرفاً یک ساده سازی از فرضیهٔ KR در جائی که" استدلال" تنها روش "ایجاد رفتار است که دانش را آشکار و معلوم می کند". استدلال برای KR ضروری است،و مخصوصاً برای زبان های KR ،حتی هنوز توانائی های استدلال ساده می تواند به مسائل جدی منجر شود و همچنین باید به خوبی درک و با دقت به کار برده شود. در۳۰ سال گذشته یکی از پیشرفت های مهم در کاربرد KR طرح پیشنهادی ، مطالعه و پیشرفت زبان هایKR مبتنی بر قالب و چارچوب بوده است.در حالیکه زبان های KR مبتنی بر قالب در تنوع درجه ها از یکدیگر متفاوت هستند،اصول مرکزی این سیستم ها یک نکته بر پایهٔ مشخصات اشیاء(مفاهیم) و روابط آنها با یکدیگر است.مشخصات اصلی چنین زبانی عبارت است از:

۱.شیء گرائی

همه ی اطلاعات درباره ی یک مفهوم خاص به همراه آن مفهوم با عنوان متضاد ذخیره می شود،برای مثال ،برای سیستم های مبتنی بر قانون جائیکه اطلاعات درباره ی یک مفهوم ممکن است در سراسر مبنای قانون پخش شود.

۲.عمومی کردن /اختصاصی کردن.

عمومی یا اختصاصی کردن به مذت طولانی به عنوان یک دیدگاه کلیدی از شناخت و ادراک بشر شناخته شده،زبان های KR روشی طبیعی برای مفاهیم گروه در سلسله مراتب مفاهیم سطوح بالاتر ویژگی های عمومی تر و به اشتراک گذاشتهٔ مفاهیم زیر را ارائه می دهد.

۳.استدلال

توانایی قرار گرفتن دریک روش رسمی که موجودیت بعضی قسمت های دانش به موجودیت برخی دیگر دلالت می کند یک روش متفاوت برای استدلال فراهم می کند.KR مهم است.هر زبان KR سابقاً جزء شناخته نشده ی دانش برای توانایی

1 William Woods

تصمیم گرفتن را فراهم می کند اگر یک KR طبقه بندی .ارائه ی یک توصیف مفهوم شایسته(مناسب) آن توصیف باشد،این در حقیقت یک فرم اختصاصی عمومی برای استدلال است.

٤.دسته بندی

یک توصیف انتزاعی از یک مفهوم داده شده ،اغلب زبان های KR قابلیتی ایجاد می کند که تعیین کند اگر یک مفهوم مناسب با آن توصیف باشد،در حقیقت این مفهوم شکلی خاص و معمول از استدلال است.

موقعیت یابی و عمومی سازی شیء به ساخت حقایق ارائه داده شده قابل فهم تر برای بشر کمک می کند؛استدلال و طبقه بندی به ایجاد سیستمی کمک می کند که به گونه ای رفتار می کند که انگار می داند چه چیزی نمایش داده شده است.سیستم های مبتنی بر قالب با اهداف فرضیهٔ KR مواجه می شوند.

دریافتن توانایی ها و محدودیت های نمایش های مبتنی بر قالب،مخصوصاً در مقایسه با دیگر صورتها مهم است.با تمامی تکنیک های KR نمادین که در یک روش یا دیگری از منطق مرتبهٔ اول (FOL)،وبه عنوان یک دلیل جهت ارائه دانشی که تغییر نکرده است،مشتق شده اند،شروع کنید.ممکن است سیستم های مختلفKR قابلیت سروکار داشتن با تغییرات غیر یکنواخت در دانش نمایش داده شده را داشته باشند،اما فرض پایه بر اساس است آن تغییر است که اگر در حال حاضر استثنائی نسبت به قانون باشد.

سیستم های تولید و سیستم های پایگاه داده دو شکل دیگر اعلانع عمده KR هستند.سیستم های تولید به عبارت ساده و طبیعی KRدو خبر اصلی صورت های اجازه می دهد.اگرچه این سیستم ها به صورت کاملاً بازگشتی نمایش داده می شود وقتی مشکلات بزرگی راIf_thenقانون اعمال می کند،وقتی هیچ نظمی برای قوانین وجود ندارد و استنتاج ها نمی توانند مجبور شوند که فقط با اشیای مورد علاقه از ارتباط دور نگه داشته شوند.سیستم های تولیدی با سیستم های بر پایه ی قالب استنتاج می شوند که در مجموع توانایی های استنتاج طبیعی را مانند طبقه بندی و وراثت فراهم می کند،به خوبی تکنیک های ساخت دانش از قبیل عمومی سازی و تعیین موقعیت شیء.

سیستم های پایگاه داده فقط برای نمایش ادعاهای ساده،بدون استنتاج تهیه می شوند.قوانین استنتاج قسمت های مهم دانش دربارهٔ یک دامنه هستند.سیستم های بر پایهٔ قالب به طور جاری وقتی با دانش رویه ای سروکار دارند،شدیداً محدود می شوند.یک مثال قانون رویه ای قانون گرانش نیوتون است_جاذبهٔ بین دو جرم به طور معکوس متناسب با مجذور فاصله های آنها از دیگری است.دو قالب نمایشی دو جسم با شکاف های نگهداری موقعیت ها و جرم شان ارائه داده شده،مقدار جاذبهٔ کششی بین آنها نمی تواند به صورت اعلانی در استفاده از مکانیسم های استدلالی استاندارد موجود در زبانهای KR بر پایهٔ قالب استنتاج شود،اگرچه یک تابع یا رویه در زبان برنامه نویسی می تواند مکانیسمی برای اجرای این "استدلال " به خوبی ارائه دهد.

سیستم های بر پایهٔ قالب که می توانند با این نوع دانش سروکار دارشته باشند با اضافه کردن یک زبان رویه ای به نمایش آن عمل می کند.دانش در یک روش بر پایهٔ قالب نمایش داده نشده است ؛ آن به عنوان کد C یا لیسپ که در سراسر شکاف در قالب دستیابی شده است ،نمایش داده شده است.این یک امتیاز مهم است _دانشی وجود داردکه در آن توابع لیسپ رمز گذاری می شود،که کاملاً در دسترس نیست.سیستم می تواند به وسیلهٔ آن دانش استدلال کند اما نه دربارهٔ آن (دانش)؛ به عبارت دیگر ما می توانیم برخی روش های مربوط (وابسته) را برای محاسبه یا استنتاج مقدار یک شکاف بر پایهٔ برخی دیگر به کار بریم اما نمی توانیم بپرسیم چگونه آن مقدار به دست آورده می شود.

٤-٥ مدلسازی دامنه

مدلسازی دامنه،فیلدی در کاربرد KR برای دامنه های خاص استکه مطالعه و اجرا شده است.

دانش دامنه به دانش های تسخیر شده توسط متخصص دامنه اشاره می کند که باید در برخی مدل ها کد گذاری شود.این دانش به خوبی تعریف نشده و به طور منصفانه دسترسی برای بقیه مشکل است. مدل متا به آداب KR ،به طور نوعی یک زبان KR که به عنوان سطح نماد برای ارائه ماشین این دانش استفاده خواهد شد،اشاره می کند.معرفی به پردازش گرفتن دامنه ی دانش و نمایش فیزیکی اشاره می کند،آن مدل متا را بکار می برد.این پردازش بعضی اوقات به عنوان آگاهی یابی اشاره می شود.مدل دامنه به پایگاه معلومات که از نمونه سازی ناشی می شود ، اشاره می کند و اهداف

اجرایی به عنوان نمونه رسماً نمایش داده نمی شوند،اما به استدلال مدل دامنه ی ساخته شده و آنچه به کار می برد اشاره می کند.این جداسازی سنتی بین برنامه ها و مدل های دامنه سبب مشکلات در طول مدلسازی یک مدل دامنه می شود که فقط شامل دانش اشیاء و مشخصات نیست ،اما دانش جنبه های رویه ای پردازش به خوبی در ارتباط با دامنه است.مشکلات ناشی ازاین حقیقت است که مدلسازی دامنه یک نظم دادن در پیشنهادات به طور توسعه بخش است با ساختن بر فراز سیستم های قبلی. برخی از نتایج بسیار مهم شکلی از اسلوب شناسی است که به دیگر قالب سازی های دامنه، برای اجتناب از دام ها و به کار بردن روش هایی که موثر واقع می شوند ،کمک می کنند.

٤-٥-١ تحلیل وابسته به هستی شناسی

روش های شناسایی برجسته(عمده) برای مدلسازی دامنه به طور واضح نشان می دهد که نمونه سازی مدل وقت گیر ترین قسمت است و اینکه مهمترین قسمت نمونه سازی، تحلیل وابسته به هستی شناسی است.هستی شناسی برای رده بندی های عمومی اشیاء، فراوان است، و به نظر می رسد راهنمایی های واضحی برای پیشرفت یک نوع جدید آن باشد.

تعداد کمی راهنما برای دانش مورد نیازجهت نمایش دانش رویه ای و استدلال درباره ی آن وجود دارد.،مخصوصاً وقتی روش های مورد نیاز نمایش به عنوان نرم افزار اجرا می شود.زمینه ی بیشتری برای رسم فوق وجود ندارد ،متفاوت با سیستم های اطلاعاتی نرم افزاری ،تا آنجا که هستی شناسی ها و اسلوب شناسی ها برای مدلسازی آنچه نرم افزار انجام می دهد.پایان تحلیل بر پایه ی هستس شناشی یک قسمت بزرگ تلاش برای جستجو است و ربای همین این هرگز قبلاًانجام نشده بوده است.

لغت "هستی شناسی " به معنای "مطالعه ی وضعیت فوق " است.یک هستی شناسی وضعیت های موجود یک مجموعه ی منحصر به فرد از اشیاء را توصیف می کند. این توصیف معمولاً از قواعدی که هر چیزی را تعریف می کند، ساخته می شود.در نمایش معلومات هستی شناسی، اصطلاح تعریف شده برای یک قسمت از مدل دامنه داده شده است که نمونه هایی را مستثنی می کند،در عین حال آنچه آنها می توانند باشند را توصیف می کند.تحلیل وابسته به هستی شناسی، پردازشی برای تعریف این قسمت از

مدل می باشد.
آنچه دامنه ی خاص هستی شناسی را می سازد ،با توانایی های نمایشی مدل متای زبان به کار برده شده، برای ساخت مدل محدود می شود.هر زبان نمایش معلومات، در رفتارش و دامنه ی عبارت متفاوت است.به طور کل یک هستی شناسی شامل سه قسمت است:تعاریف مفاهیم، تعاریف نقش ها، وتعاریف استنتاج های بیشتر.
تعاریف مفاهیم تمامی انواع اشیاء در دامنه را می سازد.در عبارات مقصود گرا اینها تعاریف کلاس خوانده می شود،و در عبارات پایگاه داده اینها موجودیت های مستقل خوانده می شود.
نقش پیش فرض ها برای هر مفهوم معین می کند که مقادیر پیش فرض برای هر مشخصه چیست.نقش محدودیت ها برای هر مفهوم هر محدودیت بر روی مقادیر نقش را مشخص می کند از قبیل نوع مقادیر باید چه چیزی باشد،چه تعدادی از مقادیر می تواند موجود باشد،و غیره.یک نقش یک مشخصه ی یک شیء است.
قسمت آخر یک هستی شناسی مشخصات استنتاج اضافی است که زبان، فراهم می کند. مثال های اینف قوانین ذنجیره ی پیش رو ، and/or ،استنتاج طبقه بندی path ذنجیره ی پس رو،گرامر های and/or ها و غیره.demon

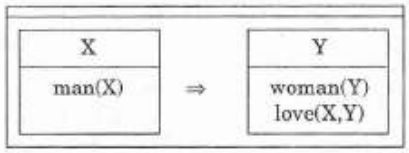
٤-٥-٢ تئوری بیان نمایش

یک فرضیه برای بیانDRT برپایهٔ تئوری نمایش، یک متغیر تجزیه ناپذیر است.(DRT) یا(Kamp) تئوری بیان نمایش زبان های طبیعی پیشنهاد می دهد.
(DRS) شامل دو قسمت است،یکی مجموعهٔ نشانه های بیان ،که برای نمایش اشیاء معرفی شده ودیگری بیان یک ساختار بیان نمایش ها به صورت اختصاصی و به صورت چارچوب به کار برده می شود و بقیه موقعیت ها روی این اشیاء هستند.
نشانه ها در قسمت بالاتر جعبه نمایش داده می شود و شرایط در قسمت پایین تر جعبه نمایش داده می شود.شکل زیر(**A** man walks) مربوط به جملهٔ(DRS) است.

x
man(X) walk(X)

مثال بعدی با "موقعیت ها" و تعیین کننده " هر" سروکار دارد.این مثال به عنوان

ارتباطی بین دو DRS را ارائه داده. این DRS برای جمله :
“every man loves a woman”
می تواند به صورت زیر رسم شود:



تفسیر یک DRS می تواند به صورت زیر تعریف شود.D در مدل واقعی M درست است ،وقتی

- الزامی از B وجود دارد که هر نماد مباحثه ای در قسمت بالائی D برای منحصر به فرد بودن در M همانند هر موقعیتی در D با توجه به B به موضوعی درست در M تبدیل می شود.
- A⇐ وضعیتی تعریف شده است که اگر برای تمامی الزامات B (که توسعه ای از B هستند ،این الزام ⇐ موقعیت موجود ، فقط برای نشانه گرهائی که در چپ DRS معرفی شده)که سمت چپ DRS را به صورت صحیح در M ایجاد می کند،B وجود داشته باشد، توسعه ای از B فقط برای آن نشانه گر هائی که در سمت راست DRS که درست بودن سمت راست DRS در M را ایجاد می کند ، معرفی کرده است.

٦-٤ سیستم های استدلالی شبکه های معنایی(شبکه های شرکت پذیر)

معرفی های شبکه معانی از ساختمان و نمایش(معرفی) ساختار در دانش فراهم می کند.در یک شبکه قسمت های دانش در یک گروه معنایی منسجم با هم جمع می شود.بنابراین شبکه ها یک روش عادی تربرای طرح و از زبانی عادی تراز طرح های نمایشی را فراهم می کنند.نمایش های شبکه ، نمایشی تصویری از اشیاء ، مشخصه های آنها ،و روابط بین آنها و دیگر موجودیت ها ارائه می دهد.

شبکه های شرکت پذیر گرافهائی با گره های برچسب دار و یالها رسم شده اند.زبان در ساخت یک شبکه بر پایهٔ نخستین دامنهٔ انتخاب شده برای اشیاء و روابطی به خوبی برخی اصول اولیهٔ عمومی بکار برده شده. شبکه های شرکت پذیر به وسیله Quillian برای مدل معنایی عبارات و لغات انگلیسی معرفی شده اند.Quillian ساختارش را

.

شبکهٔ معنایی برای معنا بخشیدن به کاربرد مورد نظرش نامید.او یک سیستمی که معانی بین لغات را به وسیلهٔ مسیرهای مرتبط با آن یافت ،را توسعه بخشید.روابط در سراسر یک نوع از " فعالسازی انتشار" بین دو واژه معین می شود.این مدل کمی جاذبه ی ذاتی(با درک مستقیم) دارد وقتی که اطلاعات مربوط با هم در سراسر اتصالات مربوطه جمع و محدود می شوند.دانش لازم برای کارایی برخی فعالیت ها، به طور نوعی به همراه یک دامنه ی کم پهنا یا "مجاورت معنایی " از فعالیت در برگرفته می شود.این نوع سازماندهی با مسیر آگاهی ذخیره شده و باز یافت شده در بشریت شباهت دارد.هیچ ترکیب(نحو) مقبول عمومی یا معنا شناسی برای شبکه های شرکت پذیر وجود ندارد.

زبان شبکه های شرکت پذیر از حروف الفبا ، در دو حالت حروف بزرگ و کوچک ،نشانه های رابطه ای،مجموعه اعضاء و زیر مجموعه نشانه ها،ارقام ده دهی ،گره های مربعی یا بیضی شکل،و کمان های مستقیم طول های قراردادی ساخته شده اند.عنوان های ذیل مثالی از تعداد کمی از سیستم هایKRR را که بر روی طرح شبکه معنائی ساخته شده ، شرح می دهد.

٤-٦-١ سیستم پردازش شبکه معنایی(SNePS)

سیستم پردازش شبکه معنایی(SNePS) به وسیلهٔ شاپیرو[1] و گروهش برای ایجاد سیستمی جهت ارائه باورهائی از زبان طبیعی مورد استفاده در سیستم هوشمند،طراحی شده است. همیشه این هدف بوده که سرانجام بانک اطلاع مبتنی بر SNePS ،نه به وسیله مهندسین دانش یا برنامه نویس که نمایش دانش را در بعضی زبان های رسمی می کنند، ساخته شود.SNePSچهار نوع گره دارد: گره های پایه، گره های متغییر،گره های ذره ای ، گره های الگو. گره های پایه و متغییر کمان ناشی شده از آنها را ندارند. یک گره پایه تعدادی موجودیت های مخصوص که به صورت مفهومی متفاوت با موجودیت نمایش داده شده با هر گره دیگر است، را نمایش می دهد.در مجموع خصوصیات موجودیت به صورت اعلانی توسط باقیمانده ی شبکه ی در ارتباط با گره هایش تعیین می شود.همچنین گره های متغیر هیچ کمانی ناشی از آنها ندارند و اختیاری ، اختصاصی ،گزاره ها و غیره را نمایش می دهد.مجدد توسط باقیمانده ی شبکه ذره ای معین می شود و گره های الگو کمان هایی ناشی از آنها ندارند و به طور

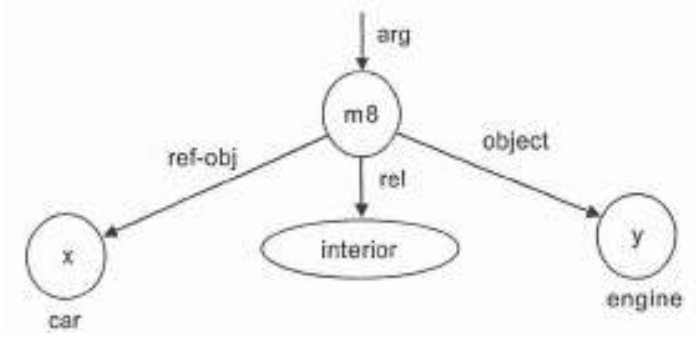
1 Shapiro

بنیادی توسط آن کمان ها ، گره های اشاره کرده به کمان های وابسته و غیره معین می شود .گره های ذره ای گزاره ها را نمایش می دهد ،شامل قوانین و "ساختارهای اختصاصی " است.گره های الگو، با جملات باز یا اصطلاحات تابعی با متغییر های آزاد گزاره ی استاندارد عمومی ،مشابه است.هر گره یک شناسه دارند،که به صورت منحصر به فرد آن را مشخص می کند.گره های پایه ممکن است شناسه ها را توسط کاربر معین کند.همهٔ گره های دیگر شناسه ها را توسط سیستم خلق می کند. شکل٤-٣ مثالی از ارائه SNePS را نشان می دهد که یک رابطهٔ جزئی از اجزاء ساختاری مشکل ماشین را که می تواند تفسیر هائی به صورت " موتورها در داخل اتومبیل ها هستند" باشد.

.

٤-٦-٢ نمایش Schubert

درشبکهٔ Schubert ،که در شکلهای مبتنی بر شبکه معنائی استفاده می شوند،متغیرها گره های اتمیک در شبکه هستند.نوع ومحدودیت های دیگر به وسیله ی اتصالات به گره های متغیر مشخص می شوند.



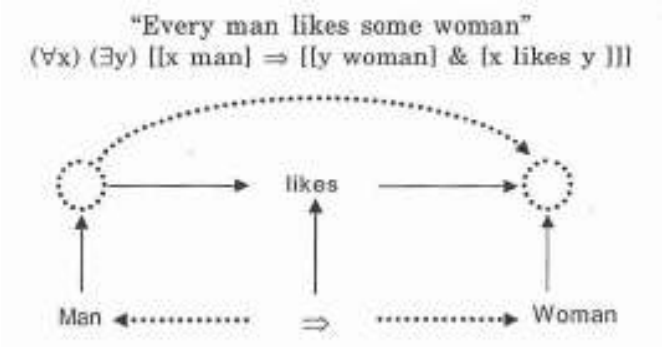
شکل ٤-٣ نمایش SNePS

هیچ کمیت سنج عمومی واضح یا وجودی وجود ندارد.متغییر های آزاد به طور ضمنی عمومی واجد شرایط هستند؛کمانهای Skolem کمیت گره های متغییر را بطور وجودی تعیین می کنند. در شکل ٤-٤ فلشهای نقطه گذاری شده یک وابستگی هدفمند بین کمیت عمومی" man" و کمیت وجودی " woman" را تفکیک می کنند.

اتصال به متغیر ها $\Rightarrow$ ارائه می شود) به وسیلهٔFOPLاین دلالت (همانطوری که در

.

نمایش داده شده است ،جملات با کمیت سنجی های شاخه ای می "likes"و مسند تواند مشخص شود.



شکل ٤- ٤ نمایش Schubert برای Every man likes a woman

اگرچه ،جداسازی متغیر ها از نمایش محدودیت های جملات آنها، رابطه ی طبیعی به زبان طبیعی نسخه اصلی نیست .علاوه بر این ،از آنجا که محدودیت های روی محتواها ی ممکن برای متغییر ها به صورت نوع محدودیت های ساده ظاهر می شود،هیچ نمایشی برای عبارت اسمی با مکمل های جملات رابطه ای محدود کننده و بالنتیجه هیچ نمایشی برای جملات احمقانه وجود ندارد،(جمله احمقانه یک نوع خاص از جملات است.جمله احمقانه مشکلات را مطرح می کند وقتی ما برای ترجمه ی آن به منطق گزاره ی استاندارد تلاش می کنیم.)

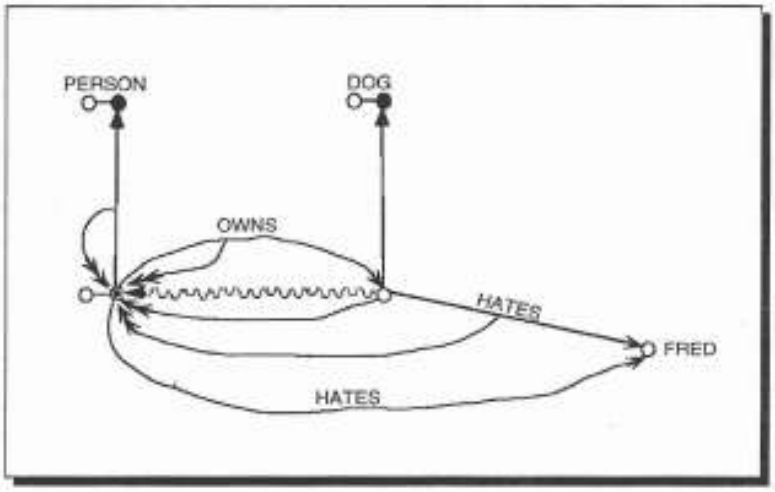
٤-٦-٣ نمایش NETL مربوط به Fahlman

NETL یک زبان شبکه ای معنایی برای معماری های پیوندگرا است . این یک سیستم برای نمایش و به کار بردن داده های واقعی جهانی است. وظیفه Fahlman تفکیک کردن دو نمایش برای متغیرها می باشد.نمایش اول به عنوان یک گره نوعی است که "عضو نوعی " از یک مجموعه را نمایش می دهد.این روش از مشکلات وراثتی نوعی وابسته به چنین نمایش هایی رنج می برد و به آن اغتشاش کپی می گوید. نمایش دوم برای متغییر ها با گره های کمیتی عمومی ارتباط دارند که این گره ها – EVERY NODEs نامیده شده است. محدودیت بر روی این گره ها می تواند محدودیت های نوعی عمومی یا محدودیت های عبارات وابسته باشد و بطور مستقیم به -EVERY NODE متصل شده اند.

متغییر های کمیتی وجودی از نوعی ازINDV با کمانهای Skolem برای متغیر های EVERY- NODE جهت نشان دادن وابستگی های هدفمند هستند.

نمایش برای جمله

" Any person that owns a dog that hates Fred also hates Fred" در شکل ٤-٥ نشان داده شده است. گره با نقطه مرکزی مشکی شخصی واجد شرایط وجودی است که صاحب سگی است که از Fred متنفر است. جفت گره های پر نشده برچسب دار یا پر شده برچسب دار با کلاسها در ارتباط است. کمانهای با سرفلشهای دوبل با چیزی که Fahlman عبارات نام نهاده در ارتباط هستند که این عبارات بطور مستقیم با محدودیتهای عبارتی بر روی متغیرها مربوط هستند. درنهایت ارتباط نفرت داشتن(hates) از جمله سطح بالائی بوسیله فلش از گره انسان (person) به گره Fred نشان داده شده است.



شکل ٤-٥ نمایش NETL برای Any person who owns a dog that hates Fred also hates Fred

وغیره است به دلیل نمایش محدودیت های جملات پیچیده .اگرچه ، شکل نمایش و Schubertاین عمومی تر از نمایش متغییر

تفسیر آن (به عنوان مثال تعریف یک مجموعه بیشتر از موقعیت یک گزاره) دوباره رابطه ا ی غیر طبیعی به جمله ای که آن بیان

.

توانایی نمایش جملات احمقانه و جملات کمیت سنج شاخه ای را نشان Fahlmanمی کند، است.بطور کلی نمایش متغییر های
می دهد؛اگرچه او در این باره بحث نکرده است و معانی چنین نمایش هایی واضح نیست.

فصل پنجم

عدم قطعیت

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

استدلال غیر یکنواخت ویکنواخت

عامل اطمینان

۵-۱ مقدمه

قبلاً در این کتاب ارائه دانش و استراتژیهای استنباط را شرح دادیم با این فرض که دانش حوزه مسئله کامل وبدون ابهام است.

علاوه بر این حقایق ، تعاریف وروابط بین هر موجودیت حقایق نامحدودی هستند که ما آنها را با قطعیت ۱۰۰ درصد درست می دانیم .برای مثال در یک حوزه مسئله ای b,aدو موجودیت یا حقیقت هستندو a+b رابطه بینشان می باشد .

هیچ استثنایی برای رابطه ی بالا واینکه اگر a درست باشد آنگاه b درست است وجود ندارد. اغلب دانش انسان ، حل مسائلی را که عدم قطعیت دارند در بر می گیرد.

این عدم قطعیت باید ارایه شود و در اغلب مسایل روزمره با موفقیت حل شود.برای مثال کارهای برنامه ریزی با عدم قطعیت، حول برهمکنش مراحل برنامه مشخص شده انجام می شوند .

برنامه های استراتژیک بیشتر با عدم قطعیت در مورداعمال و مقاصد مخالف پیچیده

شده است.آگاهی با کمیتهای شلوغ زیادی وداده های مبهم مشخص شده است. دانش مبنی بر این داده ، برای حل یک مسئله که غالب اوقات ناقص است استفاده شده است.این یک نمونه از ناشناسی کران دار را نشان می دهد و گهگاه نادرست است.
موقعیت زیادی وجود دارد که انسان نیاز دارد که یک استنتاج مناسب علی رغم دانش حل مسئله که ناقص است ، بسادگی روی تجربه حل مسئله اش ، در گستره های مشابه ایجاد کند.
برای مثال اگر کسی موقعیت عمومی مقصدش را بداند می تواند با مسیر بهینه ی به آن برسد .به عنوان مثالی دیگر می توان گفت اگر اوبداند چگونه یک ماشین را پارک کند ،او می تواند این کار را در جایی دیگربدون دانش قبلی از شرایط پارک در آنجا پارک کند . ما می توانیم مثالهای دیگری از زندگی روزمره بیابیم که یک نفر یک استنباط انجام می دهد ومسئله ای را بدون پردازش کامل و دانش روشن و غیر مبهم حل می کند.استدلال با عقل سلیم برای شرح چنین عملیاتی استفاده میشود بنابراین انسان قادر است تا هرگاه اطلاعات ناقص وناجوری در دسترس بودِ ،استدلال کند.نتیجه یه دست آمده معمولاً بر اساس دانش مطرح شده می باشد.
این نتایج موقتاً برای رقابت با اطلاعات ورودی که فرضیه ها را تصحیح می کنندِ معتبر می باشند.مطالعات بیشمار ، مواجهه ی مستقلی با انطباق این گونه استدلال قابل تجدید در ۱۵ سال اخیر دارد.شیوه های مورد استفاده متنوع شده اند.بعضی کاملاً نمادی هستند.بقیه از اعداد برای کمی کردن عدم قطعیت استفاده می کنند.بعضی محدود به منطق رسمی و بقیه که کمتر رسمی هستند بیشتر برای این منظور خاص می باشند.
در بحث جاری ، ما بسیاری از این سیستمها را که عدم قطعیت و نقص را در پایگاه دانششان و روشهای استنباطشان مدیریت می کنند ، می بینیم.ما باید یک مثال را از زندگی روزمره ی خود در نظر بگیریم و ببینیم اصول اساسی زمینه ی هر یک از این شیوها چیست . این می تواند در ارزیابی شباهتها وتفاوتها و تواناییشان برای رسمی کردن جنبه های قطعی که استدلال عقل سلیم نامیده می شود ، به ما کمک کند .
.ما شماها ، مثالها ومنطق پیش فرض را در نظر خواهیم گرفت.

منطق عدم قطعیت مانند منطق امکانات واحتمالات ،تابع عقاید مانندتئوری DSِ،منطق BAYESIAN و مدل عامل قطعیت .

منطق تیره

تعدادی از طرح های شرح داده وابسته به دسترسی به زمان می باشند.استدلال نادرست در علوم معمول می باشد. و با عباراتی چون "هنر خوب حدس زدن"ودیدگاههای ملایم تر علم مشخص شده اند .H,R(1960)

فقط یک بخش کوچکی از علوم طبیعی درست است .، ناحیه ی ریاضیات محض،در چندین ناحیه از علوم طبیعی کاربردی از طرف دیگرِ تصمیمهاِ پیش بینیها و تفاسیر که بعداز رویه های درست آمیخته با کارشناسی غیر رسمی ساخته شده اند

۵-۱-۱ استدلال نمادین در مقابل آماری

روشهای نمادین اساساً نشان می دهند که عقیده ی عدم قطعیت(ابهام) می تواند درست ، غلط ، یا نه درست نه غلط باشد. همچنین بعضی روشها با دانش ناقص ومتناقض مسائلی دارند.

روشهای آماری یک روش برای ارایه ی عقایدی ایجاد می کند که قطعی (غیرقطعی) نیستند اما ممکن است بعضی مدارک پشتیبانی یا ضد و نقیض وجود داشته باشد.

مزایای روش آماری در دو سناریو:

Genuine randomness : بازیهای کارتی مثال خوبی است .ما غیر ممکن است که بتوانیم با قطعیت نتیجه را پیشگویی کنیم اما راجع به احتمال ضریب همبستگی آیتم ها اطلاع داریم واینگونه میتوانیم کار کنیم.

استثنائها :روشهای نمادین میتوانند اینها را ارایه دهند .با این وجودِ اگر تعداد استثنائها زیاد باشدِ چنین سیستمی تمایل دارد تا بیشتر حواس معمولی و استدلال خبره را تقسیم بندی کندِ بطور مثالِ روشهای آماری میتوانند استثناهای بزرگ را بدون تکیه به شمارش خلاصه کنند.

۵-۱-۲ احتمالات

.

شیوه ی اساسی روشهای آماری مواجهه با عدم قطعیت از طریق اصل بدیهی احتمالات را اتخاذ میکند. احتمالات اعداد حقیقی بین ۰ تا ۱ هستند.

$P(A)=0$ که احتمالA می باشدعدم قطعیت مطلق A را نشان میدهد $p(A)=1$قطعیت مطلق را نشان میدهدو مقادیر بین این دو عدد درجه هایی از(غیر) قطعیت را نشان

میدهند. احتمالات میتوانند به چند روش محاسبه شوند.

(تعداد کل نتایج)/(تعداد نتایج مطلوب)=احتمال

بنابراین احتمال تک شدن در بسته بازی با کارت بدینگونه است:

(تعداد کارتها در یک دسته ورق)۵۲/(تعداد تک ها)٤=احتمال میشود۳ ۱/۱. به همین نحو احتمال پیک۱۳/۵۲ مباشد که میشود ۱/٤ .

اگر شما یک انتخاب k تایی از n تایی را انتخاب کنید فرمولی برای یافتن راههای این انتخاب وجود دارد

احتمال شرطی $P(A \text{ I } B)$,احتمال رویداد A به شرطی که Bاتفاق بیافتد را نشان می دهد .

۵-۲ استدلال غیر یکنواخت ویکنواخت

برای انجام چنین استدلال غیر قطعی ای به یک تکنیک برای ایجاد چندین فضای موازی عقیده نیاز داریم که هر کدامشان در ارتباط با عقیده ی یک کارشناس میباشند.

چنین تکنیک هایی با این حقیقت که فضاهای عقیده کارشناسان متنوع است پیچیده هستند اگر چه یکسان نیستند اما به اندازه ای شباهت دارند که به طرز غیر قابل قبولی برای ارایه ی آنها بعنوان دانش مجزای کاملاً اساسی غیر کارآمد هستند .تکنیک لازم برای چنین استدلالی به عنوان استدلال غیر یکنواخت شناخته شده است.

سیستمهای استدلال قراردادی همچون منطق پیشگویی برای کار با اطلاعاتی طراحی شده اند که سه خصوصیت دارند:

این سیستمها توسط رابطه با گستره ی بهره و علاقه کامل هستند.به عبارتی تمام اعمالی که برای حل مسئله لازم است در سیستم ارائه می شوند یا می توانند از قوانین

شرطی مشتق شوند .

این سیستمها پایدارند.تنها راه تغییر آنها اینست که اعمال جدیدی می توان به آنها اضافه کرد تا قابل دسترس شوند .اگر این اعمال جدید با اعمال دیگری که قبلاً ثابت شده اند سازگار باشند سپس هیچ چیز نباید از مجموعه ی اعمالی که درست شناخته شده اند جمع شود.به این خصوصیت یکنواختی می گویند. اگر هر یک از این خصوصیات قانع کننده نباشند ِسیستمهای استدلالی مبنی بر منطق شرطی نامناسب می شود .سیستم استدلال غیر یکنواخت برای توانایی حل مسائل در جاییکه همه ی این خصوصیات ممکن است ناپیدا باشند طراحی شده اند .برای این منظور ما باید چند کلید را آدرس دهی کنیم که شامل موارد ذیل باشد.

پایگاه دانش چگونه گسترش یابد تااستدلال ها در فقدان علم همانند حضور آن بوجود آیند؟

پایگاه دانش چگونه میتواند وقتیکه واقعیت جدیدی به سیستم (یا وقتی یه چیز قدیمی دور ریخته می شود)اضافه می شود ِ بطور مناسب بهنگام شود؟

پایگاه دانش چگونه می تواند ناسازگاری ها را وقتیکه چندین استدلال غیر یکنواخت متناقض ممتد وجود دارد ، ِ حل کند؟

با توجه به سیستهای استدلالی مورد استفاده در سیستمهای متداول علمی ، تمایز مشخصی بین استدلالهای یکنواخت وغیر یکنواخت دیده میشود .در یک سیستم استدلالی یکنواخت تمام مقادیر در طی دوره ی مذاکره درست باقی می ماند .موجودیتهایی که درست شده اند درست باقی می مانند و میزان اطلاعات درست پیوسته افزایش می یابد.در سیستمهای استدلالی غیر یکنواخت موجودیتی که درست بوده ممکن است کنار گذاشته شود .

طراحی وبرنامه ریزی مثال های خوبی برای مسائلی هستند که خواستار استدلال غیر یکنواخت می باشند. در وهله ی اول در یک مسئله ی برنامه ریزی، ممکن است تصور

.

شود مقادیر مطمئنی را تقبل کند بعد برای اطلاع بیشتر در دسترس باشد، این مقادیر ممکن است تغییر کنند.تغییر مقادیر یک مشخصه ی مجرد مشکل نیست ، نرم افزارهای متداول اجازه می دهند شناسه ها دوباره استفاده شوند یا مقدار جدیدی به آنها تحمیل شود .ردیابی مفاهیمی که بر اساس واقعیت خاصی است مشکل می باشد . بیشتر سیستمهای علمی که در بازار است فقط به استدلال های غیر یکنواختی که با دقت کنترل شده اند یا آنهاییکه یک مهارت what ifرا فراهم می کنند تا شناسه ها قابل تغییر باشند ، مجوز می دهند،اگر چه تحمیل فرضیه های جدید به قدیمی باید به دقت بازبینی شود.

۳-۵ عامل اطمینان

یک عامل اطمینان(CF)یک عدد در بازهٔ [-1,1] می باشد که درجهٔ اطمینان فرضیه را برمی گرداند عوامل اطمینان مثبت نشان می دهند که بر معتبر بودن فرایض گواهی وجود دارد.

وقتی cf = 1باشد ،فرضیه صحیح است.به عبارتی CF منفی می گوید که فرضیه غلط است.یک CF کوچک یعنی اینکه باور بزرگتری که نا معتبر باشد وجود دارد cf=-1 یعنی فرضیه کاملاً مردود است cf=0 یعنی نمیتوان در خصوص غلط یا درست بودن فرضیه ها گواهی داد.

فرضیه ها عباراتی اند درخصوص ارزش پارامترها ، برای گره های گوناگون در درخت سابقه هستند .

درMYCIN

فرضیه های نمونه :

H1= شناسه های سازمان -۱ streptococcus ،است.

h2=بیمار -۱ تب دارد .

hS=نام بیمار -۱ جان است.

X =Cf(h,, E) می گوید که عامل قطعیت برای فرضیه ی h بر پایه ی e می باشد.

بنابراین ما می توانیم داشیم: CF(hl, *E)* = 8, CF(h2, ***E)*** **= -3** and CF(h3,

1 = ***E)*** و غیره.

در اینجا عوامل قطعیت به دو شیوه استفاده می شوند .یکی اینکه ارزش هر پارامتر بالینی با cf همبسته ذخیره می شود . در این شرایط E همواره در MYCIN برای تمام اطلاعات جاری می ماند.

بنابرین اگر برنامه نیاز داشته باشد که هویت سازمان -۱ممکن است در مجموعه ی حقیقت دیده شود و دریافت شود که هویت سازمان-۱ streptococcus با CF=0.8 باشد .دومین کاربرد CF ها در جملات قواعد تصمیم گیری خودشان است.در این شرایط گواهی E مشابه موقعیت قضیه ی قواعد است.

A A B A C ~ D نمایش .CF(D, **A** A B **A C)** = X در وقتی است که تشخیص تحت تا ثیر مسئله مریضی ازیک لیست فرایض متناقض باشد.این فرایض با هم درcf شان برای هر گره در درخت انتخاب ذخیره می شوند .در موفقیت عملکردیشان قانع کننده اند.

اگر چه احتمالات شرطی یک ساختار ریاضیاتی در گرفتن وزنهای قطعیت فراهم می کنند آنها به خاطر کمبود داده در محاسبه ی مقادیرو همچنین بخاطر حضور وسیع و دانش غیر دقیق خیلی هم مناسب نیستند . مدل عامل قطعیت یک تخمین برای احتمالات شرطی می باشد .برای مثال با استفاده از برهانBays میتوانیم عدم قطعیت را در علم تحت تسخیر داشته باشیم.اگر E تمام گواه باشد و D

فرضیه ی تحت توجه می باشد سپس P(dllE)احتمال شرطی برای فرضیه ی D خواهد بود .

فرضیه Bays مفید است چونکه به P(dllE) اجازه می دهد که از ترکیب احتمالات شرطی ساخته شود .در اینجا di یکی از فرایض گسسته است.برای مثالdi Di میتواند یک مریضی و Eممکن است مجموعه ی از نشانه ها باشد.از وقتی که مدارک مداوماً در سیستمهای خطایاب گردآوری شده اندمی شوند ما باید از قضیه ی Bays برای افزایش پله ای احتمالات شرطی فرایض استفاده کنیم.اینکار با اصلاح

قضیه یBays به صورت زیر ممکن است:

(PdllE v sl) =
P(sIl(di A E)) * P(diIE)
Cj P(s1ldj A (djlE))

کهE مجموعه مدارک قبلی می باشد.diها فرایض اند و s مدرک جدید می باشد .برنامه هایی که از قضیه ی Bays به این فرم استفاده می کنند به میزان زیادی داده آماری نه تنهاP(s|d) برای هر قطعه داده s درE بلکه همچنین رابطه متقابلs با هریک از فرایض به عنوان مدارک بیشتر گرداوری شده اند.رویه ای به نظر می رسد اما در عمل مناسب نیست.همچنین ما فقط علوم قابل داوری را استفاده می کنیم بنابراین نگرش Bayesian می تواند تخمین زده شود.CF(d1E) = P(djlEP) وقتی که p(d) کوچک است .

وقتیکه ما باید برای فرضیه ای به طور دائم جستجو کنیم به یک تابع ترکیبی نیاز داریم .این تابع در MYCIN :

(1 - min(MB, ***MD)***
CFco,(X, ***Y)*** = X, Y(l - ***X)***
where X, Y > 0
where X or Y ***c*** 0
= - CFC,,(-X, -Y)
where X, Y c 0

می باشد.,,,cfجابجاپذیری را حفظ می کند.این امر به این دلیل لازم است .

فصل ششم

تکنیکهای جستجو

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

جستجوی مهارتها

مشکل ارائه (نمایش)

تعاریف

ارائه برنامه

حل مسئله در هوش مصنوعی

تکنیک وسیع جستجو

روش ارتباط و انشعاب با رنامه ریزی دینامیکی (پویایی)

روش مینی مکس

٦-١ جستجوی مهارتها

در هوش مصنوعی جستجو ، به بدنه بزرگی از عقاید اصلی که منجر به ا ستنباط و استنتاج و برنامه ریزی و استدلال و اثبات قضیه و پردازش میشود ، مربوط است . .
کاربرد این عقاید کلی در پردازش زبان طبیعی و بازیابی اطلاعات و برنامه ریزی اتوماتیک و رباتها و تجزیه مراحل و بازی و سیستمهای خبره و اثبات قضیه های ریاضی یافت می شود.
به طور کلی سیستم های جستجو سه جزء دارند:

۱- پایگاه داده
۲- عملگرها
۳- استراتژی کنترل
پایگاه داده دامنه کاری جاری و هدف یا به عبارت دیگر اعمال کار را شرح می دهد .
عملگر ها برای دستکاری پایگاه داده استفاده می شوند
استراتژی کنترل تصمیم می گیرد که چه عملگری و کجا به کار گرفته شود
منظور از هر تکنیک جستجویی به کارگیری یک ترتیب مناسبی از عملگرها برای یک دامنه اولیه کاری برای رسیدن به هدف است.
دو راه برای رسیدن به هدف وجود دارد :
۱- استدلال پیشین
۲- استدلال پسین
استدلال پسین به کاربرد عملگرها
به آن ساختارهایی در پایگاه داده که دامنه کاری را به منظور ایجاد یک موقعیت تعیین شده شرح می دهند مربوط می شود .مثل روش استدلال پایین به بالا یا داده اشتقاقی ، که هدف این است موقعیت را از حالت اولیه به حالت پیشرو بیاورد .
برای مثال یک بازی شطرنج را در نظر بگیرید.
موقعیت اولیه . مشخص کردن مکان مهره های شطرنج روی تخته در شروع بازی
هدف. هر وضعیتی از تخته بازی که کیش و مات کند
عملگرها. قوانینی برای حرکت قانونی در شطرنج

استدلال پیشین یا بالا- پایین یا استدلال هدف هدایت شده ، جملات هدف (مسئله) را به " زیرهدف " ها یی می شکند است که امیدوارانه ، برای حل کردن آسانتر است و راه حل آن برای حل مسئله اصلی کافی است .
به عنوان مثال مسئله انتگرال گیری را در نظر بگیرید:

$$1/\cos 2 \; dx$$

اگرعملگر اجازه دهد :

1/cos2 dx
آنگاه ما می توانیم مجددا" به این صورت بیان کنیم:
Sec2x dx
جالب توجه است که بسیاری از استدلالهای ما استدلال پیشین است.

٦-۲ مشکل ارائه (نمایش)

یک سیستم حل مسئله هر دو صورت استدلال " پیشین" یا " پسین" را استفاده می کند که هر عملگر فقط برای تولید یک شیء جدید یا یک حالت جدید پایگاه داده کار می کند که به آن گفته می شود که مسائل را در یک فضای حالت نمایش دهد .
در این جا جالب است که توجه شود که در مورد استدلال پیشین ؛ دو حالت رخ می دهد:
۱-کاربرد یک عملگر یک مسئله جدیدی را که اندازه یا سختی آن کمتر از مسئله اصلی است به بار می آورد .
۲-کاربرد یک عملگر در یک سیستم مجتمع یک مسئله را به مجموعه ای از "زیر مسائل" کاهش می دهد
شاید اهمیت کمتری از مسئله اصلی داشته باشد.
ارائه کاهش مسئله به چنین سیستمی گفته می شود
در هر حال بین کاهش ارائه مسئله و فضای حالت یک رابطه وجود داردو ممکن است برای حل مسائل در ترتیب " زیر مسئله " ها محدودیت وجود داشته باشد یا وجود نداشته باشد .
برای مثال : اگر عملگر اصلی در انتگرال گیری به صورت زیر باشد
[f(x)+g(x)] dx
بسته به نوع نمایش مسائل جدید میتواند به دو حالت دیده شده وجود داشته باشد .
. دو مسئله یکپارچه جدید که می تواند در هردر حل شود
. دو مسئله یکپارچه بعلاوه یک مسئله سوم f در انتگرال جمع شود
در آخرین حالت؛ تکلیف سوم نمی تواند اجرا شود مگر اینکه دو تکلیف اول زودتر کامل شوند.

.

۶-۳ تعاریف

در این کتاب ما بعضی از تکنیکهای جستجو را به دقت بررسی می کنیم. اما قبل از انجام این کار تعدادی از تعاریف اولیه را در اینجا شرح می دهیم:

نمودار: یک نمودار یک شیء داده ای است که شامل دو مجموعه است که رئوس و لبه نامیده می شوند. V یک مجموعه متناهی و غیر تهی از گره هاست و E یک مجموعه متناهی از جفت گره ها ست. هر جفت در E یک لبه در نموداراست. اگر هر جفت از رئوس به صورت(i,j) مرتب (که متفاوت از (j,i) می باشد) شده باشد سپس نمودار مستقیم است در غیر این صورت نمودار غیر مستقیم است .

درخت: یک درخت یک مجموعه متناهی از یک یا چند گره است. ویک گره ویژه که ریشه نامیده می شودوجود دارد.؛ گره های باقیمانده در مجموعه های گسسته ی T1.....Tn دسته بندی می شوند که هر یک از این مجموعه ها یک درخت است. T1.....Tn "زیر درخت" نامیده می شوند.

گره ریشه، گره نهایی و بچه ها: گره ی بالای درخت که پدر ندارد ریشه درخت نامیده می شود.

گره قعر درخت که فرزندی ندارد گره نهایی نامیده می شود. اگر هر گره ای به گره ها ی دیگر وصل شده باشد یا گره شاخه ها که گره های دیگر به آن وصل شدند فرزند آن گره نامیده می شوند.

فاکتور انشعاب،گسترش،شروع، پایان: اگر تعداد فرزندان همیشه برای همه گره هایی که فرزند دارند یکسان است این تعداد فاکتور انشعاب نامیده می شود. پردازش فرزندان به دست آمده از گره ها را گسترش گره گویند. گره ها گفته می شود باز شوند تا اینکه گسترش داده شوند سپس آنها مسدود می شوند(گره پایان).

مسیر:یک مسیر از راس Up به راس دیگرUq یک تناوبی از رئوس Up,Ui1,....Uin است چنان که (Up,Ui1) ,.....(Uin,Up) لبه ها در E(G) هستند. طول مسیر تعداد لبه ها در آن است .

هدف: هدف یک مسیر از ریشه درخت به حالت هدف است. حالت هدف ممکن است به دوصورت تعریف شود: به صورت صریح و واضح ؛ یا ترکیبی از حالتها که به حالتی

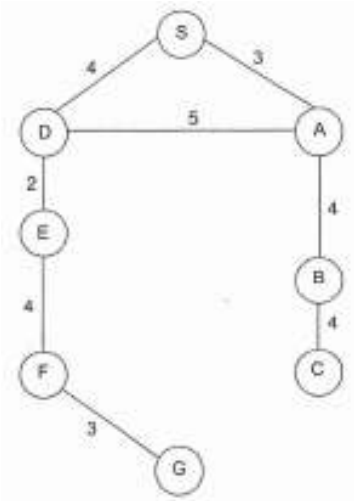
معلوم دلالت می کند.
جستجوی یک راه حل بوسیله ایجاد یک درخت فضای حالت که شامل مسیر راه حل است انجام می شود.
برگشت در لیست:برگشت در لیست یکی از معمولترین تکنیکها در طراحی الگوریتم ارائه می دهد. برای به کارگیری این روش ، راه حل مطلوب باید بر یک بازه n تایی به صورت (g1.....gn) دلالت کردکه gi از مجموعه محدود از Di انتخاب می شود. اغلب مسئله ی که حل شده است برای بیشینه ساختن مقیاس کارکرد تابع P(g1....gn) فراخوانی می شود .
اکنون فرض کنید m اندازه Di باشد. آنگاه n تایی m = m1,m2m...,mn وجود دارد که ممکن است برای حل مقیاس تابع P کاندید شده باشد.
رویه"back track" فقط تا پلهایی که منجر به راه حل بهینه می شوند ارزیابی می کند. آن یک عقیده اساسی است تا بردار راه حل یک جزءی در یک زمان بالا ببرد و مقیاس اصلاح شده تابع Pi(g1,...,gn) را استفاده کند برای تست اینکه آیا مسیر موجود هیچ شانسی برای موفقیت دارد یا نه.
اگر در حالتی احساس شود که مسیر انتخاب شده نمی تواند به سوی یک راه حل بهینه هدایت شود آنگاه ممکن است آزمایش مسیر انتخاب شده کاملا" نادیده گرفته شود.
.برنامه نویسی پویا: برنامه نویسی پویا یک روش طراحی الگوریتم است . که می تواند وقتی استفاده شود که حل یک مسئله به عنوان نتیجه یک توالی از تصمیمات دیده شود. به هر حال در بعضی از حالات ؛چنین تصمیم تدریجی (فقط بر اساس اطلاعات محلی) ممکن است برای ساختن مقدور نباشد. یک راه برای بدست آوردن راه حل بهینه در چنین مواردی این است که همه مسیرهای ممکن را امتحان کرده وبهترین مورد را انتخاب کنیم. برنامه نویسی پویااغلب به طور موثری میزان شمارش را بوسیله جلوگیری از شمارش ترکیبی از تصمیماتی که نمی توانند بهینه باشند کم می کند.
شاخه و حد: شاخه و حد به همه روشهای جستجوی فضای حالت ، که همه فرزندان گره ریشه یا گره جاری قبل از هر گره دیگری که می تواند گره جاری شود ایجاد شده اند مر بوط می شود .
قاعده انتخاب ایجاد گره بعدی ، هیچ برتری به یک گره نمی دهد که شانس خوبی

برای رسیدن از جستجو به گره جواب دارد .
این روش جستجوی نیروی حیوان صفت می تواند به وسیله ارائه رتبه ای به گره ها که تا کنون کاوش نشده اند بهینه شود. لازم است هزینه محاسبات یک " گره پاسخ" از گره جاری را بدست آوریم.
این هزینه می تواند تعداد گره های لازم برای رسیدن به هدف یا مجموع تعداد سطوحی که تا رسیدن به نزدیکترین گره هدف نیاز است ، باشد .
توجه کنید که در عمل بکارگیری ضوابط بالا خیلی مشکل است چون نیروی مورد نیاز برای محاسبه هزینه گره ممکن است خیلی زیاد باشد که شامل هزینه جستجوی " زیر درخت " از گره جاری است.

٦-٤ شماهای ارائه

در تجزیه به خوبی حل؛ ساختار درختی بطور معمول برای نمایش استراتژی کنترل در جستجو استفاده می شود.
در نمایش فضای حالت ، یک درخت ممکن است برای نمایش مجموعه ای از حالتهای مسئله استفاده شود که بوسیله کاربرد عملگرها تولید می شود.
در چنین نمایشی ، ریشه درخت حالت مسئله اولیه یا موقعیت را ارائه می دهد . هر یک از حالات جدید میتوانند بوسیله اعمال یک عملگربه ریشه که به عنوان گره جانشین گره ریشه ارائه می شوند بدست بیایند.
کاربرد عملگر برای این گره ها جانشین بعدی را تولید می کنند.
به هر حال در عمل ؛ حالت ها بوسیله یک گراف نمایش داده می شوند تا یک درخت زمانی که ممکن است مسیرهای مختلفی از ریشه به گره داده شده وجود داشته باشد.
در کنار درخت و گرافها شماهای نمایش دیگر شامل گرافهای AND/OR است که که در روش حل مسئله شامل کاهش مسئله استفاده می شوند.
آنها ابزاری را برای بدست آوردن "زیرهدف" ها فراهم می آورند که ترکیبی است که برای بدست آوردن هدف مطلوب کافی است.
در اینجا شکل ٦-١ یک نمایش ساده از یک درخت است که حالتهای مختلفی را با هم و با تغییر نمایش می دهد. همچنین نمایشی از تغییر ازحالتی به حالت دیگر با کاربرد

یک عملگر در یک شاخه از درخت درشکل ٦-۲ داده شده است.



شکل ٦-۱

یک مسئله جستجوی پایه ای در شکل ٦-۱ نشان داده شده است. یک مسیر از گره شروع (S) به گره هدف (G) پیدا شده است .
شکل٦-۲ رویه های جستجو درختها را مانند اینها کاوش می کنند .

شکل ٦-۲

این پیش زمینه به ما اجازه می دهد که به رویه معمولی حل مسئله در هوش مصنوعی نظری بیندازیم .

٦-۵ حل مسئله در هوش مصنوعی

.

رویه اولیه برای حل یک مسئله در AI می تواند به صورت زیر نوشته شود :

رویه تولید

۱- داده ⟵ پایگاه داده اولیه

۲- بعضی قوانینیR را از مجمو عه ای از قوانین که می توانند به داده اعمال شوند انتخاب کن

۳- داده ⟵ نتیجه اعمال قاعده به داده ها

پایان پردازش

در رویه بالا مسئله کنترل بنیادی انتخاب قوانین R برای کاربرد در پایگاه داده است یک مشخصه مهم در این ملاحظه میزان اطلاعات و آگاهی در دست که این تخمین ها به کار می برند .

چنین رویه جستجویی اطلاعاتش را مانند رویه جستجوی " کشف کننده/مطلع" بکار می برد بر خلاف یک رویه جستجوی " کور/نابینا "جایی که انتخابها دلخواهانه انجام می شود و مؤثر بودن محاسبات یک جستجو بستگی به طیف (بی اطلاعی / مطلع) در استراتژی کنترل دارد .

این موثر بودن با دونوع هزینه اندازه گیری می شود: هزینه کاربرد قوانین و هزینه کنترل استراتژی کنترل کاملا" بی اطلاع فقط یک محاسبات کنترلی کوچک به وجود می آورد. به هر حال نتیجه چنین استراتژی در هزینه ،درخواست قوانین بالا است و تعداد زیادی از درخواست قوانین برای پیدا کردن راه حل برای آگاهی دادن یک استراتژی کنترل درباره قلمرو مسئله را لازم دارد .

٦-٦ تکنیک های جستجوی کورکورانه

اگر چه تکنیک های جستجوی نا آگاهانه برای مسائل کلی عملی نیستند. به هر حال یک پایه ای را برای درک استراتژی جستجوی آگاهانه فراهم می آورد.

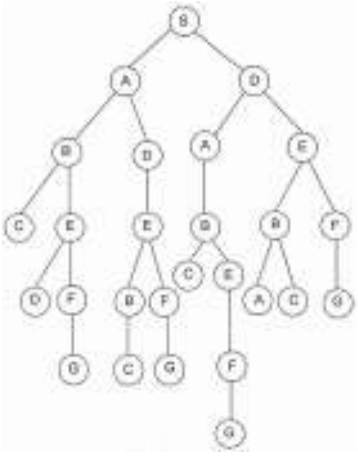
همه الگوریتم های جستجو فرضهای زیر را به وجود می آورند:

یک گره می تواند بوسیله بعضی رویه های شناخته شده مانند رویه گسترش ، توسعه یابد.

گراف فضای حالت یک درخت است که دلالت دارد به اینکه فقط یک حالت شروع و یک راه منحصر به فرد از یک گره به دیگری وجود دارد.

هر وقت یک گره گسترش داده می شود تا فرزندانش را تولید کند یک اشاره گر به

عقب از فرزندان به والد ایجاد می شود .
بنابراین وقتی یک گره هدف بدست می آید ویژگیها دنبال کردن مسیر راه حل را ممکن می کنند . برای تمام روشهای جستجو درخت داده شده در شکل ۳-٦ حالت شروع مسئله را نشان می دهد.



شکل ٦-۳

٦-٦-۱ جستجوی سطحی

در جستجوی سطحی ، ما از یک گره اولیه S شروع می کنیم و آن را به عنوان گره ملاقات شده علامت گذاری می کنیم .
سپس S برای بدست آوردن همه گره های دیگر به ترتیب نزدیکی به S ، گسترش داده می شود . این بوسیله تعداد لبه های بین آنها اندازه گیری می شود. ما پیمان نامه ای که در شیوه " چپ به راست " آزموده شده است را استفاده می کنیم. یعنی تمام عملگرهای ممکن سطح n قبل از هر عملگر دیگری در سطح n+1 در نظر گرفته می شود .
اگر چه این رویه ممکن است خیلی طولانی باشد با این حال یک راه بهینه برای هدف پیدا می شود اگر چنین راهی وجود داشته باشد.
جستجوی سطحی بوسیله الگوریتم زیر شرح داده می شود:

۱- از یک صف یک عاملی شامل گره ریشه .

۲- تا زمانی که صف خالی است یا هدف در دسترس است تصمیم می گیریم که اولین عامل در صف گره هدف است .

a) اگر عنصر اول گره هدف است سپس عملی انجام ندهید.

b) اگر عنصر اول گره هدف نیست سپس عنصر اول را از صف حذف کنید و فرزندان عنصر اول را اضافه کنید.

۳- اگر هدف یافت شد اعلام موفقیت است در غیر اینصورت اعلام شکست است.

در جستجوی سطحی پردازش سطح به سطح رو به پایین انجام می شود تااینکه هدف به دست آید

ما می توانیم مشاهده زیر را در باره جستجو های سطحی ایجاد نماییم.

و آن یک استراتژی جستجوی سیستماتیک است که همه گره ها در سطح یک و سپس همه حالتها در سطح ۲ و غیره در نظر می گیرد.اگر راه حلی وجود داشته باشد این جستجو تضمین می کند که آنرا بیابد .و اگر چندین راه حل وجود داشته باشد جستجوی سطحی ابتدا کم عمق ترین وضعیت هدف را پیدا می کند .

٦-٧ رویه ارتباطات و انشعاب با برنامه ریزی پویا

۱-از صف مسیرهای ناقص_صف اولیه شامل طول مسیر صفر و قدم صفر از گره ریشه به هیچ مکانی است.

۲-تا زمانی که صف خالی است یا به هدف دسترسی پیدا کرده است.اگر اولین مسیر در صف به گره ی هدف دسترسی پیدا کرد ، مشخص کنید:

a) اگر اولین مسیر به گره ی هدف دسترسی پیدا کرد کاری انجام ندهید

b) اگر اولین مسیر به گره هدف دسترسی پیدا نکرد:

I) برداشت مسیر اول از صف

ii) در مسیر جدید توسعه اولین قدم بوسیله ی مسیر برداشته شده.

iii) اضافه کردن مسیرهای جدید به صف

iv) مرتب سازی صف بوسیله ی جمع بندی هزینه در مقابل با هزینه ی کمتر مسیرها

v) اگر ۲ مسیر یا مسیرهای بیشتری به یک گره دست یاقت تمام مسیرهای دیگر را حذف کنید به جز ان مسیری که به آن گره با کمترین هزینه دسترسی دارد.

۳)اگر هدف پیدا شد اعلام موفقیت است وگر نه اعلام شکست است.

پایان ارتباط و انشعاب با برنامه ریزی پویا
جستجوی ارتباط و انشعاب بوسیله ی برنامه ریزی پویا مشخص می کند که مسیر S_D_E_F_G بهینه است. ارقام در کنار گره ها فاصله ها را جمع کرده اند. گره های حذف شده بی فایده مشخص شده اند. گره های کمتر با عملیات جستجوی ارتباط و انشعاب بدون برنامه ریزی پویا گسترش پیدا کرده اند.

رویه A :
رویه A یک جستجوی ارتباط و انشعاب است که با براوردی از فاصله باقی مانده با روند برنامه ریزی پویا ترکیب شده است. اگر تخمین فاصله ی باقی مانده ی کمتر از فاصله واقعی باشد. سپس رویهA راه حل بهینه را ارائه می کند. بنابر این رویه جستجوی A این است:
روش A:
۱- از یک صف مسیرهای ناقص_ صف اولیه شامل طول مسیر صفر، گام صفر از گره ی ریشه به هیچ مکانی است.
۲- تا زمانی که صف خالی است یا به هدف دسترسی پیدا کرده است اولین مسیر در صف را که به گره هدف دسترسی پیدا کرده را مشخص کنید.
a) اگر اولین مسیر به گره هدف دسترسی پیدا کرد کاری انجام ندهید.
b) اگر اولین مسیر به گره ی هدف دسترسی پیدا نکرد:
I) برداشتن اولین مسیر از صف
II) در مسیر جدید توسعه اولین قدم بوسیله ی مسیر برداشته شده
III) اضافه کردن مسیر جدید به صف
IV) مرتب سازی صف بوسیله ی جمع بندی هزینه در مقابل با مسیرهایی با کمترین هزینه
V) اگر ۲ یا تعداد بیشتری به یک گره دست یابند تمام مسیرهای دیگر را حذف کنید به جز آن مسیری که به آن گره با کمترین هزینه دسترسی دارد.
۳-اگر هدف پیدا شد اعلام موفقیت است وگر نه اعلام شکست است .
پایان A

٦-٨ جستجوی بازی

بازی هایی مانند شطرنج ، چکرز، تیک تاک نو و غیره نوع دیگری جستجو نیاز دارند. گره دردرخت بازی نشان دهنده ی ترکیبات صفحه ی شطرنج و شاخه ها نشان دهندهٔ تغییر شکل ازترکیب یک صفحه به صفحه ی دیگر است. تصمیمات بوسیله ی ۲ نفر یا حریفان گرفته می شوند که هر کدام یک تصمیم در هنگام نوبتشان می گیرند. توجه کنید در بازی هایی مانند شطرنج یک حرکت به معنی حرکت منفرد یک بازیکن و جواب حریفش می باشد. در این جا بطور غیر رسمی حرکت به عمل منفرد یک فرد بر می گردد.

بازی ها نیازمند رویه های جستجوی مختلفی هستند نسبت به چیزی که ما با آن قبلا مواجه شدیم. روشهای جستجوی اجباری حیوانی قطعا خارج شده اند. این به خاطر ان است که به عنوان مثال در شطرنج اگر ما فاکتور انشعاب موثر را ١٦ و عمق ۱۰۰ در نظر بگیریم سپس تعداد انشعابها احتمال خسته کننده ای بطور تقریبی ۱۰۰ خواهد بود که بطور مسخره آمیزی یک عدد بزرگ است.

از طرف دیگر اگر مقداری را بعنوان ازمایشی برای انتخاب حرکت بعدی مان فرض می کنیم سرانجام ما نتایج ضعیفی را خواهیم گرفت. بنابر این این قبیل تکنیک ها نیازمند یک آنالیزور موقعیت است که می تواند بعد از آنکه بازی از طریق حرکات مراحل مختلف توسعه پیدا کرد استفاده شود. باید به حد کافی در دقت شود که در مجموعه فوران رخ ندهد. به این دلیل پس از یک عمق منطقی ، موقعیت ها می توانند بر طبق یک حرکت رو به جلو نگهداری شوند. این بر اساس یک تصوری است که حرکات موقعیت های قطعی را به عنوان پیشرفت های بازی مشخص می کنند. باید مورد توجه قرار گیرد که این گمان یک فرض قابل بحث است.

٦-٨-١ روش مینی مکس

فرض کنید ما یک آنالیزور داریم که تمام قضاوتها در مورد موقعیتهای صفحه شطرنج را به یک عدد کیفیتی تغییر می دهد. هم چنین فرض کنید که اعداد مثبت نشان دهنده حمایت از یک فرد و اعداد منفی برای حریف او هستند درجه حمایت به ارزش واقعی عدد بستگی دارد .

پروسه تعیین کیفیت عدد " ارزیابی آماری" نامیده می شود .در پایان تعداد حرکا ت است که امتیازات ارزیابی آماری را پیدا کند که بوسیله آنالیزورهای موقعیت که ارزیابی

کنندگان آماری نامیده می شوند فراهم شده است.
امید بازیکن برای اعداد مثبت بازیکن بیشتر و حریف او بازیکن کمتر نامیده می شود. در سطح میانی در درخت ،ارزشها در وضعیت پایانی داده شده اند .ارزش یک وضعیت غیر پایانی بوسیله برگشت دوباره از مراحل پایانی محاسبه شده است . این روش که بوسیله اطلاعات امتیاز دهندگی شما را از درخت بازی می گذراند رویه مینی مکس نامیده می شود از وقتی که امتیاز در هر گره حداقل یا حداکثر از امتیازات آنی پایین تر است.

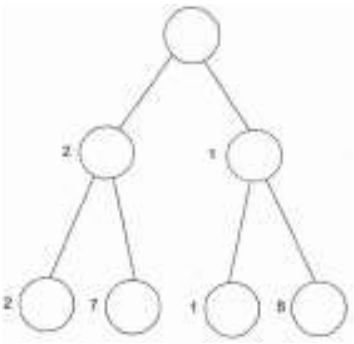
رویه مینی مکس

۱- مشخص کنید اگر محدودیت جستجو مورد دسترسی قرار گرفته است یا اگر یک سطح حداقل یا یک سطح حداکثر است.

a) اگر محدودیت جستجو در دسترس باشد ارزش آماری موقعیت جاری وابسته به بازیکن خاص محاسبه می شود . نتیجه را گزارش کنید .

b) اگر مرحله سطح حداقل است مینی مکس را روی فرزندان در موقعیت جاری استفاده کنید. حداقل نتایج را گزارش کنید.

c) در غیر این صورت اگر مرحله سطح حداکثر است مینی مکس را روی فرزندان موقعیت جاری استفاده کنید . حداکثر نتایج را گزارش بدهید.



تصویر ٦-۱۱ : روش مینی مکس از جریان کسب کردن هدف عبور می کند.

رویه مینی مکس از یک ارزیابی کننده آماری را برای محاسبه تعداد فواید موقعیت بازی در انتهای یک درخت بازی به طور جزئی توسعه یافته استفاده می کند .. اگر رویه

مینی مکس مورد استفاده واقع شود ارزیاب کننده آماری باید در هر وضعیت که در انتهای هر درخت پیدا می شود مورد استفاده قرار گیرد خوشبختانه روش هایی وجود دارد که اعداد ارزیابی ها می تواند با کاهش تعداد انشعابها در درخت کاهش پیدا کند.

٦-٨-٢ روش آلفا – بتا

روش آلفا – بتا این طور رفتار می کند : اگر حریف یک جواب دارد که پتانسیل حرکت بد است نیازی برای بررسی جواب های دیگر برای پتانسیل حرکت نیست.
بنابراین رویه آلفا – بتا به صورت زیر است:

۱- مشخص کنید سطح بالاترین سطح است یا اگربه محدودیت جستجو رسیده است یا اگر مرحله سطح پایین تری است یا اگر سطح بالاتری است.

a) اگر سطح بالاترین مرحله است اجازه دهید آلفا ∞- باشد و بتا ∞+ باشد.

b) اگر به محدودیت جستجو رسیده است مقدار آماری موقعیت جاری که وابسته به بازیکن خاص است را محاسبه کنید.نتیجه را گزارش دهید.

c) اگر مرحله یک سطح پایین است

i) تا وقتی که همه بچه ها مورد آزمایش با مینی مکس بوده اند آلفا بزرگتر از بتا است.

A: بتا را به کوچکتر از بتای معین شده تنظیم کنید و کوچکترین مقدار گزارش داده شده بوسیله مینی مکس را روی فرزندان بکار برید.

B : مینی مکس را بر روی فرزند بعدی آلفا و بتا استفاده کنید

ii) بتا را گزارش دهید.

d) اگر مرحله یک سطح بالاتر است :

i) تا زمانی که همه بچه ها با مین مکس یا آلفا مورد آزمایش قرار گرفته اند آنها بزرگتر از بتا هستند .

A : آلفا را با بزرگترین مقدار آلفای معین شده تنظیم کنید و بزرگترین مقدار گزارش شده بوسیله مینی مکس را روی بچه ها به کار برید.

B : مینی مکس را بر روی فرزند بعدی موقعیت جاری استفاده کنید آخرین تقاضای جدید مینی مکس از آخرین آلفا و بتا را نگهداری کنید .
ii) آلفا را گزارش دهید

برای بکار گرفتن یک جستجوی ما نیاز داریم که گره های پایه ، براساس مقدار برگشتی از تابع کشف کننده مرتب شوند.
ما می توانیم چنین جستجو را برای الگوریتم جستجوی عمومی به کار ببریم.تابعی که بهترین گره را برای گسترش دادن انتخاب می کند تابع BEST-FIRST-SEARCH می نامیم.
در حقیقت نمی توانیم مطمئن باشیم که بهترین گره را اول گسترش دادیم به عبارت دیگر این جستجو در همه حالات ما را از حالت جاری به حالت هدف نمی رساند.
اما به ما این دانش را میدهد که باور کنیم آن بهترین گره برای گسترش است و این تابع ،

BEST-FIRST-SEARCH(problem, EVAL-FN) یک رشته از راه حلها را بر می گرداند.

ورودی: Problem , a problem
EVAL-FN یک تابع ارزیابی
Queueing-Fn یک تابع که گره ها را بوسیله EVAL-FN مرتب می کند

٦-۸-۳ جستجوی حریصانه

جستجوی حریصانه پیاده سازی فلسفه جستجوی بهترین است . آن روی یک اصل علمی کار می کند که بزرگترین بیت از مسئله گرفته می شود.
جستجوی حریصانه سعی می کند تا هزینه تخمین زده شده برای رسیدن به هدف را حداقل کند.
برای انجام این کار گره هایی را که فکر می کند به حالت هدف نزدیکترند گسترش می دهد.برای انجام این کار تابع کشف کنندگی را بکار می برد .

با بدست آوردن یک تابع کشف کننده h ؛ ما می توانیم یک جستجوی حریصانه را به کار بگیریم مانند زیر:

تابع GREEDY-SEARCH(problem) یک راه حل شکست را برمی گرداند.

Return BEST-SEARCH(problem,h)

٦-٨-٤ جستجوی تپه نوردی

بررسی های لازم شاید راه های مرتب سازی به صورت تناوبی یا انتخابی بهبود ببخشد جستجوی تپه نوردی یک تکنیکی است که شامل مرتب سازی به صورت جستجوی عمقی است.

یک طرح مرتب سازی ، بر اساس مسافتهای باقیمانده مثل خط مستقیم یا مسافت پروازی است.

شکل ٦-٦ رویه جستجوی تپه نوردی را نشان می دهد

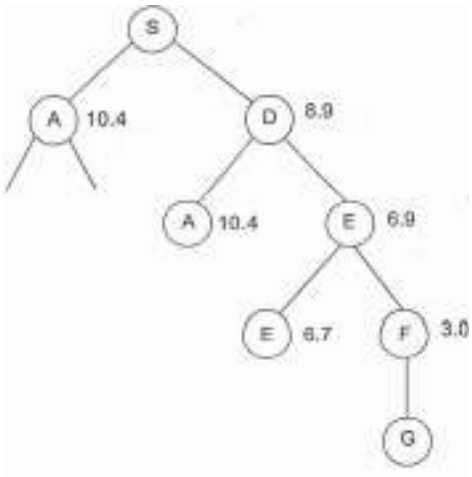
رویه جستجوی تپه نوردی:

۱- از یکی از عناصر صف شامل گره ریشه

۲- تا زمانی که صف هست خالی یا هدف بدست می آید تعیین کن که آیا اولین عنصردر صف گره هدف است : a) اگر اولین عنصر گره هدف است کاری انجام نده

b) اگر اولین عنصر؛ گره هدف نیست ، اولین گره را حذف کن. فرزندان آن گره را اگر وجود دازد مرتب کن ، مسافت باقی مانده را تخمین بزن.و آنها را در ابتدای صف اضافه کن .

۳- اگر هدف پیدا شد موفقیت را نشان بده در غیر این صورت شکست را نشان بده.



شکل ٦-٦

جستجوی تپه نوردی ، یک جستجوی عمقی با یک اندازه گیری اکتشافی است که گره هایی را که گسترش می دهیم مرتب می کند. شماره کنار گره ها ، مسافت خط مستقیمی است که گره با گره هدف دارد.

٦-٨-٥ جستجوی A*

در بخش قبلی ما جستجوی حریصانه را در نظر گرفتیم . این روش جستجو هزینه رسیدن به هدف را با استفاده ازتابع کشف کننده کاهش می دهد.

جسجوی حریصانه میتواند زمان جستجو را کاهش دهد اما نه کامل است نه بهینه.

در مقایسه جستجو با هزینه یکسان هزینه مسیر را نیز حداقل می کند .

جستجوی با هزینه یکسان هم بهینه هست هم کامل اما می تواند بسیار بی فایده باشد.

اگر ما بتوانیم دو استراتژی را برای دست یافتن به مزایای هر دو جستجوترکیب کنیم ، بهترین کار را انجام می دهیم . خوشبختانه می توانیم با ترکیب دو تابع ارزیابی به این امر دست یابیم .

$$F(n) = g(n) + h(n)$$

$g(n)$ هزینه مسیر از گره شروع به گره n را می دهد و $h(n)$ هزینه تخمینی از ارزانترین مسیر از n به هدف است . ما داریم :

.

F(n) =هزینه تخمین زده شده از ارزانترین راه حل از طریق n

چیز خوب درباره این استراتژی این است که با وجود قرار دادن محدودیتی روی تابع h بهینه , کامل است .
ما می توانیم جستجوی A* را به صورت زیر به کار ببریم:
تابع A*-SEARCH (problem) راه حل شکست را بر می گرداند.

Return BEST-FIRST-SEARCH(problem , g + h)

توجه کنید که اگر شما از این الگوریتم به صورت دستی استفاده می کنید همیشه کم هزینه ترین گره را روی ریشه هر جاییکه گره در درخت جستجواست گسترش دهید.
به این منظور گره ای که شما برای گسترش بعدی انتخاب می کنید فقط به گره هایی که تولید شده اند محدود نشده است. البته این در الگوریتم به عنوان تابع صف به صورت اتوماتیک به ترتیب گره ها ساخته شده است .

٦-٨-٦ کشف کننده قابل قبول

محدودیتهایی را که در بالا برای تابع h نام بردیم این است که تابع h نباید هرگز هزینه ای بیش از تخمین برای رسیدن به هدف داشته باشد.
چنین تابع h ای را تابع کشف کننده قابل قبول می نامیم. راه دیگر توصیف تابع قابل قبول این هست که بگوییم آنها خوشبینانه هستند.وقتی که آنها همیشه فکر می کنند که هزینه رسیدن به هدف کمتر از مقدار واقعی آن است.
آن واضح است که تابع کشف کننده SLD قابل قبول است بطوریکه ما هرگز نتوانیم یک مسیر کوتاه تر بین هر دوشهر پیدا کنیم .
A* هم بهینه هست هم کامل اما این خبر خوبی نیست. می تواند نشان داده شود که تعدادی از گره ها که بدست آمده اند هنوز به صورت تابع نمایی از طول فضای جستجو برای بیشتر مسائل هستند .
این حالت دلایلی برای زمان جستجودارد اما معمولا" برای فضای مورد نیاز خیلی جدی تر می شود.

٦-٨-٧ معمای ٨

معمای ۸ شامل هشت مربع و یک فضای خالی است. هدف چیدن هر مربع در یک زمان تا رسیدن به حالت هدف است .
مسئله در شکل ٦-۷ نشان داده شده است

5	4		1	2	3
6	1	8	8		4
7	3	2	7	6	5
Initial State			Goal State		

شکل٦-۷

این مسئله فقط مرحله مستقیم است که آن را برای مطالعه سخت می سازد در عین حال آنچنان سخت نیست که ما توی باتلاق گیر کنیم
یک راه حل معمول آن حدود ۲۰ مرحله است . فاکتور انشعاب آن ۳ است (٤- وقتی که خانه خالی در مرکز قرار دارد . ۲- وقتی که خانه خالی در گوشه ها باشدو ۳- برای حالتهای دیگر) این به این معنی است که دریک جستجوی خسته کننده در حدود ۳^۲۰ حالت وجود دارد. به هر حال بوسیله نگهداری اثر حالتهای تکراری می توانیم آنرا کاهش دهیم چون فقط ۳,٦۲,۸۸۰=۹! ترتیب متفاوت از ۹ مربع وجود دارد .

اما حتی ۳,٦۲,۸۸۰ تعداد زیادی برای جستجو است بنابراین کار بعدی یافتن یک تابع کشف کننده خوب است.

اگر ما بخواهیم یک راه حل بهینه پیدا کنیم به یک تابع کشف کننده احتیاج داریم که هرگز تخمین بالایی از تعداد مراحل در رسیدن به هدف را ارائه ندهد.. به یک کشف کننده قابل قبول احتیاج داریم.
اینجا دو امکان وجود دارد :
H1 که تعداد خانه هایی است که در مکانهای نادرست هستند.در شکل بالا در قسمت حالت اولیه از هشت سفال ؛ هفت تای آن در خارج از مکان واقعی هستند.
این کشف کنند ه به صورت واضح قابل قبول است که هر یک از خانه هایی که خارج از مکان خود هستند نیاز دارند که حداقل یک بار حرکت داده شوند تا به محل صحیح

.

برسند.

H2 : مجموع فواصل خانه ها از مکان هدفشان . روشی که ما استفاده می کنیم برای محاسبه مقدار فواصل خانه ها از موقعیت هدفشان جمع کردن تعداد افقی و عمودی موقعیت هااست.

این تابع کشف کننده همچنین قابل قبول است چون هر حرکتی فقط می تواند یک خانه را یک مرحله به هدف نزدیکتر کند .

فاصله مان هاتان برای حالت شروع در بالا هست:

$$H2 = 2 + 3 + 3 + 2 + 4 + 2 + 0 + 2 = 18$$

برای خانه های ۱ تا ۸ .

چگونه ما تصمیم می گیریم که کدام یک برای استفاده بهترین است؟

یک روش این است که مثالهای زیادی از مسئله را ایجاد کنیم و از تابع کشف کننده استفاده کنیم و ببینیم که کدام یک به ما بهترین راه حل را می دهد

جدول ٦,١ نتیجه ای از ۱۰۰ مسئله اجرا شده در عمق (طول راه حل) ۲و٤و٦و...و ۲٤ که از A* با تابع ها ی کشف کننده h1,h2 را نشان می دهد.

هزینه جستجو عددی را نشان می دهد که نشان دهنده میانگین تعداد گره هایی است که گسترش داده شده اند.

Table 6.1

Search cost				EBF		
Depth	IDS	A*(h1)		A*(h2)	A*(h1)	IDS / A*(h2)
۲	۱۰	٦	٦	۲,٤٥	۱,۷۹	1.79

٤	۱۱۲	۱۳	۱۲	۲,۸۷	۱,٤۸	۱,٤٥
٦	٦۸۰	۲۰	۱۸	۲,۷۳	۱,۳٤	۱,۳۰
۸	٦۳۸٤	۳۹	۲٥	۲,۸۰	۱,۳۳	۱,۲٤
۱۰	٤۷۱۲۷	۹۳	۳۹	۲,۷۹	۱,۳۸	1.22
		۲۲۷	۷۳	۲,۷۸	۱,٤۲	۱,۲٤
				۱۲	۳٦٤٤۰٤	
۱٤	۳٤۷۳۹٤۱	٥۳۹	۱۱۳	۲,۸۳	۱,٤٤	۱,۲۳
		۲۱۱			۱,٤٥	1.25
					۱٦	۱۳۰۱

از این نتایج واضح است که h2 کشف کننده بهتری است که با بسط دادن تعداد کمتری گره به نتیجه می رسد.اما چرا این گونه است؟

یک دلیل واضح این است که تعداد گره ها یی که گسترش داده می شوند فاکتور انشعاب هستند . اگر فاکتور انشعاب بالا است پس گره های بیشتری گسترش داده می شوند. بنابراین یک راه برای اندازه گیری کیفیت تابع کشف کننده ؛ درک فاکتور انشعاب میانگین است .

اگر ما از A* برای مسئله استفاده کنیم و عمق راه حل d باشد سپس b* فاکتور انشعابی است که یک درخت یکنواخت باعمق d خواهد داشت تا گره های N را نگه دارد. بنابراین:

$$N = 1 + b^* + (b^*)^2 + \ldots + (b^*)^d$$

.

یک مثال برای محسوس کردن مطلب ارائه می دهیم. اگر A* یک راه حل در عمق ۵ پیدا کند. از ۵۲ گره استفاده می کند سپس فاکتور انشعاب موثر ۱,۹۱ است :

$$52 = 1 + 1.91 + (1.91)^2 + (1.91)^3 + (1.91)^4 + (1.91)^5$$

فاکتور انشعاب مؤثردر جدول برای استراتژی های مختلف جستجو نشان داده شده است.
ما می توانیم بفهمیم ازاین که A* که h2 را بکار می برد فاکتور انشعاب مؤثر پایین تری دارد بنابراین h2 یک کشف کننده بهتر از h1 است.
ما می توا نیم بپرسیم آیا h2همیشه بهتر از h1 است؟در حقیقت آن اسان است که ببینیم برای هر گره n ؛ $h2(n) > h1(n)$.
این حالت را " سلطه " می نامیم و می گوییم که h2 بر h1 غلبه می کند.تسلط مستقیما به مؤثر بودن ترجمه می شود .
آن چیزی است که می توانیم بگوییم h2 بر h1 سلطه دارد اما چطور می توانیم h2 را در اولین مکان مطرح کنیم.
هردو تابع h1,h2 تعداد حرکات را برای حل نهایی تخمین می زنند. ممکن است که ما احتیاج داشته باشیم که حرکات بیشتری را تخمین بزنیم. اما اگر ما قوانین بازی را به روشنی تغییر دهیم سپس ما می توانیم این کشف کننده ها را بسازیم که یک طول مسیر دقیق که برای حل مسئله نیاز است را نشان می دهند.
به طور مثال ؛ اگر قوانین چنان تغییر کنند که ما بتوانیم هر خانه را به سوی مربع خالی حرکت دهیم سپس h1 می تواند مقدار دقیق طول مسیر مورد نیاز برای حل مسئله را نشان دهد.
به طور مشابه اگر یک خانه بتواند در هر مسیری حتی به سمت یک مربع اشغال شده حرکت کند ؛ سپس h2 یک طول مسیر دقیق از راه حل را می دهد.این تغییرات در قوانین ؛ قوانین و مسئله را که محدودیت عملگر دارند راحت تر می کند که یک مسئله راحت شده نامیده می شود.این معمولا در موردی است که هزینه راه حل دقیق یک مسئله راحت ، کشف کننده خوبی برای مسئله اصلی است .

٦-۹ ویژگی برنامه ریزی

این بخش شما را مرحله به مرحله در ایجاد برنامه های لیسپ برای حل مسائل

جستجوی عمقی راهنمایی می کند و سپس مسائل حریفان و ادم خواران را با استفاده از prolog نمایش میدهد. نوشتن یک برنامه برای جستجوی عمقی آسان است. دو رویه بازگشتی یک درخت را به صورت موثر می شکافند. اگرچه که اصلاح این کدها و وفق دادن آنها برای جستجوی سطحی و دیگر جستجوها آسان نیست.

به ما اجازه دهید که روی یک جسجوی صف گرا کار کنیم. صف ما شامل مسیر های ناقص است. استفاده از صفهایی با مسیر های ناقص در ابتدا سخت است. اما یکبار که ما جستجوی عمقی را انجام دهیم ، تغییر دادن این کدها برای انجام جستجوهای دیگر آسان خواهد شد .

جستجو به سادگی اولین بحث خودش را به یک صف تک عاملی برای سوداوری جستجو تبدیل می کند .. سپس جستجو صف را امتحان می کند و اولین مسیر در صف را برای موفقیت تست می کند. اگر آخرین گره در اولین مسیر ، گره پایانی نیست جستجو مسیر را توسعه می دهد و صف را اصلاح می کند و صف را به کپی دیگری از جستجو بر می گرداند .

۱٤۲- جستجوی اولیه (شروع پایان)

(جستجوی اول (فهرست شروع) پایان) مقدار دهی :

(شرط (صف پوچ) NIL): .برگشت NIL اگر صف خالی است.

(پایان برابر (صف ماشین) (T) : برگشت T اگر هدف پیدا شده است.

جستجوی اولT :

< ادغام خاص (توسعه (صف ماشین) و صف > پایان .

در اینجا گسترش ، فرزندان یک گره را بر می گرداند که گره به عنوان یک بحث تعیین می شود .قبل از اینکه گسترش را بنویسیم اجازه دهید نگاهی به نمایش داده در برنامه بیندازیم.اگر ما فقط با درختان مواجه شویم فهرستهای خانگی باید به خوبی انجام شوند . اگر همچنین ما بخواهیم تمام شبکه ها را کنترل کنیم بهتر است که از نشانه ها و خصایص استفاده کنیم .گره ها و فرزندانشان می توانند بوسیله نشانه ها ارائه شوند و کمانها می توانند به وسیله ی خصایص نشان داده شوند.به عنوان مثال :

SETF (GET ' S CHILDREN) (LO)

حقیقت را درک کنید که S یک والد است و بچه هایش L و O هستند.

SETF (GET 'L CHILDREN) (M F)

حقیقت این است که L یک والد است و بچه هایش M و F هستند.

با تکرار انواع حالات ما می توانیم کل یک درخت را شرح دهیم. به عنوان مثال یک درخت نمونه می تواند به صورت زیر باشد.

```
SETF (GET 'S  CHILDREN )    (LO)
SETF (GET 'L  CHILDREN )    ( MF )
SETF (GET 'M CHILDREN )     ( N )
SETF (GET 'N  CHILDREN )    ( F )
SETF  (GET 'O  CHILDREN )   ( PQ )
SETF  (GET 'P   CHILDREN )  ( F )
SETF  ( GET 'Q  CHILDREN )  ( F )
```

قبلا مشخص شد که چگونه گره ها به هم متصل شده اند ما اکنون برای نوشتن کد برای گسترش آماده هستیم.

DEFUND EXPAND (NODE)(GET NODE CHILDREN)

روش تلفیق فرزندان جدید در صف قدیمی بستگی به استراتژی جستجو دارد. برای جستجوی عمقی فرم مخصوص این است:

( )(EXPAND(CAR QUEUE)(CDR QUEUE)افزودن)

بنابراین یک جستجوی عمقی بی تجربه به این صورت است:

```
DEPTH   (DEFOUND DEPTH (Start Finish):    توجه تغییردر اسم
        (DEPTH (List  Start) Finish):     مقدار دهی اولیه
DEPTH   (DEFOUND DEPTH (Queue Finish):    توجه تغییردر اسم
        (cond (null Queue) N/L).          برگشت NULL اگر صف خالی است
        (Equal  Finish(Car Queuet)        برگشت T اگر هدف پیدا شده است
        ( T (depth , t,y)                 تلاش دوباره با صف جدید
        (append (expand (car Queue)       گره جدید در سر
        (CDR Queue));                     بقیه صف
        FINISH))))
```

این برنامه کار کوچکی را انجام می دهد. به سادگی t یا null را بر می گرداند و خوب است

اگر ما بدانیم که گره ها در طی مسیر ما را به سوی هدف سوق می دهند. هم چنین این برنامه نمی تواند شبکه ها را کنترل کند وهیچ آزمونی برای جلوگیری از رفتن آنها به سوی حلقه های بی نهایت وحود ندارد.
چگونه به برنامه ای برسیم که مسیر را برگرداند ؟ ما باید اطلاعات بیشتری از عوامل ساختمان داده ای صف جمع آوری کنیم . تا به حال نمایش عواملی که گره ها نگه می دارند تست شده است. بنابراین صف می تواند شبیه این باشد:

(s)
(L O)
(M F O)
(N F O)
(F F O)

در عوض ما می خواهیم عوامل نه تنها گره ها بلکه مسیرها را نشان دهند. هر مسیر با گره آغازی شروع می شود و به گره ای که فرزندانش هنوز زیاد نشده اند توسعه می یابد. سپس صف مانند زیر توسعه پیدا می کند

(((S)))
((LS)(OS))
((MLS) (F L S) (O S))
((N M L S) (F L S) (OS))
((F N M L S) (F L S) (O S))

به خاطر شکل جدید در صف لازم هست که ما عمق را عوض کنیم . اول پایان) (CAAR QUEUE با (CAR QUEUE) مقایسه شده است . سپس به جای برگشت T مسیری که T در آن قرار گرفته شده بر گردانده می شود . در آخر یک سازگاری کوچک برای نمایش نتیجه به صورت بر عکس که یک ترتیب طبیعی است ساخته می شود که از منبع تا هدف نام گذاری شده است.

(DEFUN DEPTH(START FINISH);	توجه تغییر در اسم DEPTH است
DEPTH LIST (LIST START) FINISH) ;	توجه تغییر محسوس
(DEFUND DEPTH (QUEUE FINISH);	توجه تغییر در اسم DEPTH
(COUD ((NULL QUEUE) N/L)	برگست null اگر صف خالی است
((EQUAL FINISH ((CAAR QUEUE))	توجه تغییر کوچک در CAAR

.

```
REVERSE (CAR QUEUE)))          توجه تغییر کوچک در CAAR
(T ( DEPTH1))                  تلاش دوباره با صف جدید
(APPEND (EXPAND (CAR QUEUE)    گره جدید در قسمت سر
(CDR QUEUE))                   بقیه صف
  (FINISH))))
```

همچنین ما باید گسترش را تغییر بدهیم .علاوه بر گرفتن یک گره و برگرداندن یک لیست از فرزندانش ، آن باید یک مسیر را بگیرد ، در پایان مسیر فرزندان یک گره را پیدا کنند و فهرستی از مسیرهای جدید را بر گرداند . هر مسیر جدید شامل مسیر اصلی با یکی از بچه هایش خواهد بود .این می تواند مانند زیر مرتب شود:

```
DEFUN EXPAND (PATH)
MAPCAR <> (LAMBDA(CHILD)(CONS CHILD PATH))
(GET (CAR PATH), CHILDREN)))
```

نقشه کار برای یک مسیر جدید که برای هر فرزند متعلق به انتهای مسیر قدیمی ساخته شده ، تنظیم شده است و هنوز اگر حلقه هایی وجود داشته باشد کار نمی کند. اگر ما آرزو داریم که شبکه ها را به خوبی درختان مدیریت کنیم ، باید مسیرهای پیشنهاد شده بوسیله عملیات گسترش را امتحان کنیم ، بررسی کنیم که ببینیم آیا گره جدیدی در جای دیگری ارائه شده است و آنرا استخراج کنیم .

```
DEFUN EXPAND (PATH);
REMOVE-IF
LAMBDA(PATH) (MEMBER(CAR PATH)CDR PATH)))
MAPCAR<> (LAMBDA(CHILD)(CONS CHILD PATH))
(GET (CAR PATH) CHILDREN)))
```

فصل هفتم

تکنولوژی هوش مصنوعی

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

- آشنایی با سه فناوری مختلف هوش مصنوعی، شامل چشم کامپیوتری ، پردازش زبان طبیعی و شناسایی کلام یا گفتار
- آشنایی با الگوریتم شناسایی چهره و چگونگی محاسبه دقت در آن
- ارائه تعریف گرامر یک زبان و انواع آن
- آشنایی با چگونگی اشتقاق جملات از یک گرامر و چگونگی عملکرد تجزیه کننده ها
- آشنایی با گرامر و برنامه نویسی منطقی
- آشنایی با زبان نمایش دانش و سپس تعریف معنی شناسی جملات
- تعریف چگونگی پردازش سیگنال صوتی

۷-۱ مقدمه

امروزه فناوری های زیادی در زمینه هوش مصنوعی در حال مشهور شدن هستند.این فناوری ها به عنوان زمینه های مختلف تحقیقاتی مطرح شده اند .در این فصل ما راجع به سه فناوری مختلف هوش مصنوعی بحث خواهیم نمود.ابتدا با مبحث دید کامپیو تری (computer vision) که برای کاربردهای ارتشی و نظامی توسعه یافته است شروع خواهیم نمود.

امروزه ما توسعه های بی نظیری از computer vision را در زمینه هایی از چشم روباتها گرفته تا دید ماشینی (چشم ماشینی machine vision) شاهد هستیم.

سپس به توضیح پردازش زبان طبیعی خواهیم پرداخت .این فرایند ما را قادر به مکالمه با کامپیوتر به زبان طبیعی می سازد.این مبحث در بسیاری از پایگاههای داده ,سیستمهای خبره , روباتها و غیره مشهور است.همکار اصلی NLP یک بازشناس کلام یا گفتار است که در بخش نهایی این فصل به ان پرداخته خواهد شد.

۷-۲ بینایی کامپیوتری[1]

مشکل اصلی یک چشم کامپیوتری تشخیص و فهم یک شیئ یا یک منظره با خواص سه بعدی ان در یک تصویر یا یک توالی از تصاویر است.

دید یا vision پردازشی است که به وسیله ان شرح مناظر فیزیکی از تصویر انها استنتاج می شود.کاربردهای متنوعی از چشم کامپیوتری وجود دارند همانند: انالیز تصاویر پزشکی , مجتمع کردن (assembly) , کشتیرانی و ناوبری , واسط انسان و کامپیوتر و غیره.

یک کامپیوتر دید انسان را در ٤ مرحله تقلید می کند که به ترتیب عبارتند از ۱)اکتساب تصویر ۲) پردازش تصویر ۳) تجزیه و تحلیل تصویر ٤) فهم تصویر .

امروزه دستگاهها و ابزار خوبی برای گرفتن تصاویر واضح وجود دارد.

وقتی تصویری گرفته می شود , تایید کیفیت تصاویر به وسیله مکانیزم پردازش تصویر انجام می شود..در اینجا اثرات سیگنالهای نویز که می توانند به علت وجود خرابی در دستگاه

---

1 computer vision

تصویربرداری یا تنوع در نورپردازی و یا وجود اشتباه در فرایند دیجیتالی کردن باشند ,نادیده گرفته شده است.از ان پس فاز تحلیل اغاز شده که با فاز فهم تصویر دنبال می شود و با شناسایی و تشخیص اشیا مختلف در تصویر سروکار دارد.

اجازه دهید با یک مساله تشخیص چهره با استفاده از یک eigenspace representation شروع کنیم.

فرض کنید یک مجموعه M تایی تصویر با سایز N x N پیکسل که هر تصویر شامل چهره یک فرد و تقریبا" تصویر فقط روی موقعیت چهره با روشنایی کافی است , داشته باشیم. و یک تصویر هم داریم که ان را با سری تصاویر مقایسه نموده و تصمیم می گیریم کدام یک از تصاویر ان مجموعه برابر با عکس مورد نظر است.

۱-۲-۷ Eigenspace Representation of images

یک تصویر N x N می تواند به عنوان یک نقطه در فضای تصویر N^2 بعدی بیان شود که در ان هر بعد با یکی از پیکسلها در تصویر مرتبط است و ارزش ممکن برای هر بعد با سطح خاکستری بودن هر پیکسل ارتباط دارد.به عنوان مثال یک تصویر 512x512 که در ان هر پیکسل یک عدد صحیح در بازه ۰ تا ۲۵۵ (یک پیکسل در یک بایت ذخیره می شود) است.فضای تصویر یک
144,262 بعدی است که هر بعد ۲۵٦ ارزش ممکن دارد .

اگر ما سری M تایی تصاویر را به عنوان M تا نقطه در فضای تصویر در نظر بگیریم یک راه شناسایی چهره یک فرد در یک تست جدید این است که نزدیکترین تصویر در ان سری در فضای تصویر را پیدا کنیم.ولی این را ه حل به دلیل اینکه اندازه فضای حالت بسیار بزرگ است بسیار کند خواهد بود . به این دلیل به جای این روش اجازه دهید هر تصویر را در یک فضا با ابعاد کمتر در نظر بگیریم , فضایی که فضای چهره (face space) یا eigen space نامیده می شود.

فرض کنید ما M' تا تصویر $E_1,E_2,....E_{M'}$ داریم که eigen space یا eigen vectorsنامیده می شود.این تصاویر, تصاویر پایه مجموعه را تعریف می کند.بنابراین هر تصویر در دوره ای مشخص می کند که چقدربه هر کدام از تصاویر مجموعه پایه شباهت دارد.

ما می توانیم یک تصویر دلخواه I را به عنوان ترکیب وزن دار (خطی) از این eigenvector ها نمایش دهیم مانند زیر:

میانگین تصویر را از تمام تصاویرازمایشی $I_1,I_2.....I_M$ محاسبه کنید: (A)

$$A = \frac{1}{M}\sum_{i=1}^{M} I_i$$

برای K=1,2,.....M' ارزش حقیقی وزن W_K را که میزان شباهت بین تصویر ورودی I و k eigenvector ام را نشان می دهد (E_K) حساب کنید.

$$W_K = E_{K^*}^T \left(I - A\right)$$

جاییکه تصویر I مفروض است و نشان دهنده بردار ستون از طول N^2 است, E_K , K امین تصویر از eigen face می باشد و بردار ستون از طول N^2 می باشد , A هم بردار ستون از طول N^2 است.

* عمل حاصلضرب نقطه ای و عمل _ تفریق پیکسل به پیکسل می باشد.بنابراین W_K ارزش حقیقی قابل سنجش می باشد.

$W = [w_1, w_2.....w_{M'}]^T$ بردار ستونی از وزنهاست که سهم هر کدام از تصاویر eigen face در نمایش I را نشان می دهد.

بنابراین به جای نمایش تصویر I در فضای تصاویر ما ان را به عنوان W point در فضای M'

بعدی وزن نمایش می دهیم که ما انرا فضای چهره یا eigen spaceمی نامیم.

از اینرو هر تصویر از اصل eigenspace طراحی شده است.به عنوان متراکم سازی , هر تصویر به وسیله اعداد حقیقی 'M نمایش داده می شود, به معنی اینکه برای هر ارزش نوعی مانند M=10 و ۳۲ بیت برای هر وزن , ما احتیاج به ۳۲۰ bits/image برای رمزگذاری ان در فضای چهره داریم.

بدیهی است که ما همچنین باید ' M تا تصویر eigen face را ذخیره کنیم که اینها هر کدام N^2 پیکسلی می باشند ولی این هزینه صرف تمام تصاویر ازمایشی (training) می شود.بنابراین می تواند به عنوان یک هزینه اضافی کوچک مطرح شود.

تصویر I حدودا" می تواند توسط W مانند زیر بازسازی شود:

$$I \approx A + \sum_{i=1}^{M'} w_i * E_i$$

این بازسازی و احیا صحیح و کامل می شود اگر $M' = \min(M, N^2)$.

بنابراین نمایش یک تصویر در eigen space در صورتی که عکس نوسازی و مطابق اصل بازسازی نشده باشد ,درست و دقیق نخواهد بود.ولی برای فرق قائل شدن بین تصاویر همین اندازه شباهت کافیست.

در این مرحله ارزشی برای 'M انتخاب کنید و سپس بهترین تصاویر eigen vector را مشخص کنید.این عمل به وسیله تکنیک امارشناسی که مولفه اصلی تحلیل(Principal Components Analysis) نامیده می شود انجام می گردد.

ذاتا" این تکنیک 'M تا تصویر را انتخاب می کند که حجم اطلاعات را در متراکم سازی بیشینه می کند.

بهترین 'M تا تصاویر eigen face به صورت زیر حساب می شوند:

برای هر تصویر ازمایشی (training) I_i , به وسیله تفریق میانگین نرمال می شود:

$$Y_i = I_i - A$$

ماتریس کوواریانس N^2xN^2 را حساب کنید.

:

eigen vector های C را که با M' تا از بزرگترین eigen values مرتبط می باشند را بیابید.

eigen vector ها را $E_1,E_2....E_{M'}$ بنامید.اینها تصاویر eigen face ای هستند که به وسیله الگوریتم بالا استفاده شده اند.

به دلیل اینکه C بسیار بزرگ است , این روش از لحاظ محاسباتی بسیار سخت است.با این حال هستند روش های سریع مشابه برای پیدا کردن K تا از بزرگترین eigen vectorها.

$$C = \frac{1}{M}\sum_{i=1}^{M} Y_i Y_i^T$$

۷-۲-۱۲الگوریتم شناسایی چهره(FaceRecognition Algorithm)

کل الگوریتم شناسایی و تشخیص چهره در مراحل زیر می تواند خلاصه شود:

۱) با داشتن یک مجموعه از تصاویرازمایشی چهره ها, M' تا از بزرگترین eigen vectorها را محاسبه کنید:$E_1,E_2....E_{M'}$.

M'=10 یا 20 یک ارزش نوعی می باشد.توجه داشته باشید این مرحله تنها یکبار انجام می شود.(offline).

۲) برای هر فرد در مجموعه ازمایشی , اصل همبستگی را با شخص در ان eigen space محاسبه کنید.ربا ی این کار از فرمول بالا استفاده کنید.$W = [w_1, w_2....w_{M'}]$.

توجه داشته باشید که این مرحله نیز تنها یکبار انجام می شود.(offline)

۳) با داشتن تصویر ازمایشی I_{test}, , ان را به وسیله محاسبه W_{test} توسط فرمول بالا به یک M' eigen space بعدی تبدیل کنید.

.

٤) نزدیکترین چهره از تصاویر ازمایشی به تصویر مورد نظر را بیابید:

$$d = \min_k \|W_{test} - W_K\|$$

هنگامیکه W_K نقطه ای در eigen space مرتبط با شخص k ام در مجموعه مورد ازمایش است و||x|| فاصله اقلیدسی در eigen space را مشخص می کند.

٥) فاصله تصویر مورد ازمایش را از eigen space بیابید.

$$dffs = \|Y - Yf\|$$

زمانیکه $Y = I_{test} - A$.

٦) اگر dffs<Threshold1

- تصویر ازمایشی به اندازه کافی به eigen space شبیه است. در مقایسه با کل تصاویر می توان مطمئن بود که این تصویر یک چهره است نه چیز دیگر.

سپس اگر d< Threshold2

- می توان I_{test} را با عنوان تصویری که شامل چهره شخص k ام است دسته بندی کرد , زمانیکه k نزدیکترین چهره در eigen space به W_{test} است.

در غیر این صورت

- تصویر I_{test} را به عنوان شخص نا شناخته دسته بندی کن.

در غیر اینصورت

- تصویر I_{test} را به عنوان تصویری که شامل چهره نیست دسته بندی کن.

مثال ۱-۷

فرض کنید دو تصویر ازمایشی ۳*۳ داریم که N=3,M=2 به صورت زیر تعریف شده است:

ما این دو تصویر را به عنوان دو آرایه (۳*۳=۹) در نظر می گیریم.

$I_1=[0,0,0,10,10,10,10,0,0,0]^T$

$I_2=[0,10,0,0,10,0,0,10,0]^T$

حال فرض کنید از یک شبه فضای (فضای فرعی) یک بعدی استفاده می کنیم.که $M=1$ و eigen vector به صورت زیر محاسبه شده است:

$E=[5,0,5,0,10,5,0,5,0]^T$

میانگین تصویر یا A توسط محاسبه هر پیکسل از I_1,I_2 بدست می اید.

میانگین سطح خاکستری بودی دو تصویر از پیکسلهای متناظر بدست می اید.بنابراین پیکسل دوم در A برابر است با $(0+10)/2=5$

از اینرو A={0,5,0,5,10,5,0,5,0}.

اکنون ما می توانیم بفهمیم که به یک eigen space یک وسیله محاسبه W_1 طراحی شودهنگامیکه

0	10	0
0	10	0
0	10	0
Image I_2		

چگونه I_1 بعدی به می

0	0	0
10	10	10
0	0	0
Image I_1		

$w_{1,1}=E_1^T*(I_1-A)$.

بنابراین ما اینجا داریم:

$$I_{1'} = I_1 - A = [0,-5,0,5,05,0,-5,0]^T$$
$$w_{1,1} = 5*0 + -5*0 + 5*0 + 10*5 + ... + 5*0 = 0$$
$$W_1 = [0]$$

اکنون فرض کنید تصویر ازمایشی زیر راداریم:

0	7	3
0	10	10
0	10	0
Image I_{test}		

با تصویر کردن یا تجسم I_{test} به فضای چهره داریم: $W_{test} = [w_{test,1}]$

زمانیکه :

$$w_{test} = E_1^T * (I_{test} - A)$$
$$= [5,0,5,10,5,10,5,0,5] * [0,2,3,-5,0,5,0]^T$$
$$= 15$$

بنابراین W_{test} =(۱۵) که به این مفهوم است که W_{test} به I_1 بیشتر شبیه است تا به I_2 .

بنابراین ما I_{test} را جزء کلاس یا گروه I_1 طبقه بندی می کنیم.

۷-۲-۳ دقت شناسایی چهره:(Face Recognition Accuracy)

کارایی استفاده از eigen space ۲۰ بعدی به طبقه بندی صحیح ۹۵٪ ای از یک پایگاه شامل ۷۵۰۰ تصویر از ۳۰۰۰ نفر را نتیجه داد.

اگر مجموعه تصاویر ازمایشی شامل دو تصویر از هر شخص است پس برای هر شخص نقطه میانگین در eigen space را باید از نقاط محاسبه شده برای هر

تصویر از یک شخص محاسبه کنید.این متد نیازمند این است که همه تصاویر در پایگاه شامل چهره هایی از یک سایز (اندازه) و یک موقعیت و یک جهت باشند به طوریکه با استفاده از این تابع عمومی فاصله بتوان انها را در eigen space مقایسه کرد .
اگر از یک شیء سه بعدی چندین تصویر وجود دارد(به عنوان مثال :سر یک شخص از جهات مختلف) سپس نقاط موجود در eigen space که متناظر با دیدهای ۳ بعدی مختلف است می توانند به وسیله تطابق یک فراسطح (Hyper surface) به همه نقاط ترکیب شوند.

این hyper surface یا فراسطح می تواند به عنوان یک توصیف از شخص در eigen space ذخیره شود.در حال حاضر محصولات تجاری مختلفی وجود دارند که بر اساس این متد eigen face عمل می کنند.

۷-۳ پردازش زبان طبیعی[۱]

پردازش زبان طبیعی یکی از بزرگترین کاربردهای هوش مصنوعی است.
این مبحث به تکنیک هوش مصنوعی برای ایجاد ارتباط با کامپیوتر با یک زبان طبیعی رجوع می کند.استفاده از یک واسط زبان طبیعی یک راه معقول برای پرداختن به طراحی نوعی خاص از واسط است.
مزیت ان این است که به هیچ مهارت خاصی غیر از داشتن توانایی ابتدایی در تایپ یک یا دو جمله به وسیله صفحه کلید نیاز ندارد.
از طرف دیگر عیب واسطهای زبان طبیعی این است که به اندازه کافی مختصر و موجز نیستند.
از لحاظ مفهومی ۲ نوع واسط NL (Natural language) وجود دارد.۱) آنهایی که خود را به یک زیر مجموعه از زبان انگلیسی محدود می کنند (یا یک زبان دیگر)

---

1 Natural language Processing

۲) آنهایی که سعی می کنند یک پوشش کامل (کمتر یا بیشتر) از یک زبان را تامین کنند(توجه داشته باشید حتی انسانها همه کلمات و معانی آنها را در زبان مادری خود نمی دانند).

همه واسط های محدود شده با مشکل قابلیت سکنی (Habitability) مواجه می شوند. آیا یک کاربر می تواند دقیقا" همان زیرمجموعه ای از یک زبان را یاد بگیرد که واسط ان را پوشش می دهد؟

کاربردهای دیگر پردازش زبان طبیعی (NLP) شامل ترجمه به زبان ماشین (MT) و سیستمهای

NL CAI (Computer aided Instruction) می باشد.

بودجه مصرف شده برای خدمات ترجمه سالانه بیش از میلیاردها دلار است.قسمت اعظم این ترجمه ها , ترجمه های تجاری و اسناد تکنیکی , قراردادها و کتابهای راهنماست.

ساختار یک زبان مانند انگلیسی به خصوص ساختار نحوی ان معمولا" به وسیله قوانین گرامری بیان می شود.این قوانین یک جمله یا یک واحد از زبان را به واحدهای کوچکتر تجزیه می کند .به طور مثال ممکن است قانونی داشته باشیم (در انگلیسی) که بگوید یک جمله می تواند با یک عبارت اسمی که با یک عبارت فعلی دنبال می شود وجود داشته باشد.این قانون به اختصار می تواند به این صورت بیان شود:

(عبارت فعلی ,عبارت اسمی → جمله) $(S \rightarrow NP,VP)$

در پردازش زبان طبیعی دو تکنیک برای تجزیه و تحلیل زبان طبیعی یا NL وجود دارد.

۱) تطبسق با قالب (همچنین key board analysis).در این روش سیستم جمله وروردی را برای کلمات کلیدی خاصی پویش می کندو وقتی پیدا شدند , سیستم با یک پاشخ موجود واکنش نشان می دهد.

۲) تجزیه نحوی (Syntactic driven Parsing) :در این روش دانش قواعد یک

زبان برای تجزیه و تحلیل استفاده می شود.

۷-۳-۱ گرامر (Grammer)

گرامر یک توصیف رسمی از ساختار یک زبان می باشد.یک گرامر ساختارهای مجاز یک زبان را مشخص کرده و به یک جمله اجازه تجزیه و تحلیل شدن می دهد.تجزیه یک جمله یافتن یک ساختار مجاز ممکن را شامل می شود.نتیجه معمولا" به صورت یک درخت است.(به نام درخت تجزیه).یک نمونه در شکل ۱-۷ نشان داده شده است.

۷-۳-۲ پارسر یا تجزیه کننده (Parser)

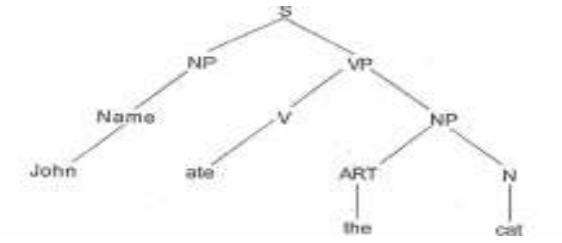
پارسر یک الگوریتم برای تحلیل یک جمله با گرامر معلوم است که به صورت:

۱)فقط جواب بلی/خیر به سوال پاسخ می دهد.آیا این جمله از قواعد داده شده پیروی می کند؟

این چنین پارسری یک پذیرنده (accepter) نامیده می شود.

۲) همچنین یک ساختار توصیفی برای جملات صحیح تولید می کند .برای مثال درخت در شکل ۱

۱-۷ به صورت لیست زیر می تواند نشان داده شود.



شکل ۱-۷

درخت مورد نظر در زبان LISP :

.

```
(S (NP (NAME john))
  (VP (V ate)
    (NP (ART the))
      (N cat)))
```

درخت مورد نظر در PROLOG :

```
S(np(name(john)),
  Vp(v(ate),
     Np(art(the),
       N(cat))))
```

توضیح: گرامر مستقل از متن:

1. S →NP VP
2. VP →V NP
3. NP →NAME
4. NP →ART N
5. NAME →john
6. V →ate
7. ART →the
8. N →cat

توضیح: یک مجموعه ۵ تایی است به صورت(P,A,N,T,S) :

- P یک مجموعه از قوانین مستقل از متن
- A مجموعه ای از سمبولهای الفبایی از قوانین
- N یک مجموعه از سمبولهای غیر ترمینال است.
- T یک مجموعه از سمبولهای ترمینال است.
- S یک سمبول غیر ترمینال به نام سمبول شروع است.

به طور مثال:

```
T= {ate, cat, john, the}
N ={S, NP , VP, N, V, NAME, ART}
```

P ={S→NP VP , VP →V NP,….}

توجه داشته باشید چگونه قوانین تولید به قوانین گرامری و فرهنگ لغات بالا منشعب می شود.

NAME V,N, و ART سمبولهای لغوی یا pre-terminal نامیده می شود.

در گرامر زبان برنامه نویسی قواعد مستقل از متن مثل زیر می باشد:

While Statement →**while** condition **do** Statement List **end**
Statement List →Statement
Statement List → Statement ; Statement List

۷-۳-۳ انواع گرامر

گرامر می تواند به یکی از صورتهای زیر باشد:

- قوانین نامحدود (unrestricted grammars)
- قوانین حساس به متن (context-sensitive grammar)
- قوانین مستقل از متن (context-free grammars)
- گرامرهای با قاعده (regular grammars)

این ٤ نوع از گرامر در بازنویسی قواعد الفا --> بتا با یکدیگر فرق دارند.

- قواعد نامحدود:

هیچ محدودیتی در قواعد ان وجود ندارد.قواعد نامحدود به صورت گسترده استفاده نمی شوند.قدرت و توانایی زیاد آنها استفاده از آن را مشکل ساخته است.

- قواعد حساس به متن یا گرامر دگرگونی: (transformational grammar)

.

طول رشته در سمت چپ قانون(آلفا) باید کمتر یا مساوی با طول رشته در سمت راست(بتا) قانون باشد.

قوانین تولید در گرامر های حساس به متن می توانند بری تبدیل جملات معلوم به جملات مجهول متناظر استفاده شود.

- قواعد مستقل از متن یا گرامر ساختار عبارت (phrase structure grammar):

همه قوانین باید به فرم $A \rightarrow alpha$ باشند که در آن A یک سمبول غیر ترمینال است و alpha یک رشته دلخواه از سمبولهاست.

- گرامرهای با قاعده یا گرامر خطی راست (right linear grammar):

همه قوانین یکی از این دو فرم را در بر می گیرند: $A \rightarrow t$ و $A \rightarrow tN$, طوریکه A,N سمبولهای غیرترمینال و t عضو واژگان می باشد (سمبول نهایی)

گرامرهای با قاعده به اندازه کافی قدرتمند و توانا نیستند تا بتوانند به راحتی زبان طبیعی را توصیف کنند .(حتی زبان برنامه نویسی را).آنها گاهی می توانند برای توصیف بخشی از از زبانها استفاده شوند و این مزیت را دارند که می توانند سریعتر تجزیه شوند.

این محدودیتها به قوانین انواع گرامرها اعمال می شود.

۷-۳-٤ اشتقاق جملات از یک گرامر

برای اشتقاق جمله از یک گرامر , از سمبول S شروع کرده و ان را به عنوان رشته جاری در نظر بگیرید.

مکررا" پروسه های بازنویسی را به صورت زیر اجرا کنید:

- قانونی را انتخاب کنید که LHS (قسمت سمت چپ) ان در رشته جاری رخ دهد.(در گرامرهای مستقل از متن LHS باید سمبول غیر ترمینال باشد)

- LHS ان قانون را با RHS (قسمت سمت راست)قانون در رشته جاری جایگزین کنید.و یک رشته جاری جدید تولید کنید.

این عملیات را تکرار کنید تا زمانیکه هیچ غیر ترمینالی در رشته جاری باقی نماند.

سپس این رشته جاری تولید شده یک جمله در زبانی است که توسط گرامر ایجاد شده.(قبلا" آن یک ترم یا عبارت محسوب می شد.)

Current string	*Rewriting*
S →NP VP	S
→NAME VP	NP
→ john VP	NAME
→ john V NP	VP
→ john ate NP	V
→ john ate ART N	NP
→ john ate the N	ART
→ john ate the cat	N

تجزیه کردن شاید معکوس این پردازش باشد.(انجام گامهایی که در بالا نشان داده شده , تجزیه پایین به بالا و راست به چپ john ate the cat را تشکیل می دهد.)

۷-۳-۵ تجزیه بالا به پایین (Top –Down Parsing)

در این روش تجزیه , ما حدس می زنیم (پیش بینی می کنیم) که کدام محصول بعدا" بکار برده می شود ودر صورتیکه حدس ما اشتباه باشد , پیشنهادهای متناوب را به صورت پشته در می آوریم و در صورت نیاز به آنها بر می گردیم .

این الگوریتم می تواند پیچیدگی زمانی نمایی در جملات با گرامرهای مستقل از متنی که خوب نیستند داشته باشد.

با گرامر well behaved یک الگوریتم می تواند زمان خطی داشته باشد.گرامرهای NL معمولا" well behaved نیستند.نمونه زیر یک تجزیه بالا به پایین را نشان می دهد.

S →NP VP
NP →ART N | NAME
PP →PREP NP

VP → V | V NP |V NP PP |V PP

Sentence :1 The 2 dogs 3 cried 4.

Backup states	*Position*
S →NP VP	1
→ART N VP	1
NAME VP	1
→ (The) N VP	2
NAME VP	1
→(dogs) VP	3
NAME VP	1
→ V	3
V NP	3
V NP PP	3
V PP	3
NAME VP	1
→ (cried.)	4

۷-۳-۶ تجزیه پایین به بالا (Bottom –up Parsing)

Thee dogs cried → ART N V
→ NP V
→ NP VP
→ S

با استفاده از متد تجزیه پایین به بالا همه گرامرهای مستقل از متن می توانند در n x n گام تجزیه شوند طوریکه n طول جمله باشد.

به دلیل اینکه ممکن است تجزیه قابل پیش بینی پیچیدگی زمانی نمایی داشته باشد لذا ممکن است قسمتهایی از جمله خصوصا" قسمتهای گیج کننده مجددا" تجزیه شوند.

۷-۳-۷ تجزیه نموداری (Chart Parsing)

نمودار یا chart رکوردی از تمام زیرساختارهاست که در طول تجزیه شدن ساخته شده اند.نمودار گاهی ممکن است well-formed substring table نامیده

شود.نمودارهای واقعی به سرعت پیچیده می شوند.

نمودارها برای جملات محذوف یا elliptical نیز مورد استفاده قرار می گیرند.

1 : Q .How much are apples?
2 : A .Thirty cents each.
3 : Q .Plums?

تجزیه جمله سوم به عنوان جمله حذفی می باشد ولی تمام آنها محذوف نیستند و در تجزیه " آلو" به عنوان NP روی چارت می باشد.

تجزیه کامل کل مکالمه به عنوان نوعی ساختار می تواند مفید باشد.

الگوریتم تجزیه پایین به بالا مبتنی بر چارت (A Bottom up Chart-based Parsing Algorithm)

این الگوریتم تاثیر پردازش تجزیه پایین به بالا را بیان می کند.

تجزیه جمله به طول n با n x n x n گام ضمانت شده است و بهتر از گرامر well-behaved با (n x n steps or n step) کار می کند.

الگوریتم اجزاء اصلی جمله را می سازد.(عبارتی یا کلمه ای)

نکات ۲ تا ۹ زیر به طور کامل مشخص نمی کند که کدام گامهای تجزیه انجام می شوند.

یک راه معقول اینست که یک کلمه را پویش (مانند نکته ۳) و بعد تمامی گامهای تجزیه ممکن در(٤ تا ۷) را قبل از پویش کلمه دیگر انجام داد.

تجزیه زمانی کامل می شود که آخرین کلمه خوانده شود و تمامی گامهای تجزیه برای آن اجرا شود.

ورودی های پارسر:جمله , کلمه , گرامر.

عملیات تجزیه کننده:

۱) این الگوریتم در دئ ساختار داده ای عمل می کند.نمودار فعال که مجموعه ای از کمانهای فعال و جزء اصلی است از ابتدا خال هستند.

۲) گرامر برای در بر گرفتن قوانین الحاق کلمات مورد توجه قرار می گیرد.برای مثال اگر" fly " در فرهنگ لغات به عنوان کلمه استفاده شده باشد و در همان مدخل لغوی به عنوان فعل نیز به کار برده شود سپس به صورت قسمتی از گرامر زیر خواهد بود:

N → fly
V → fly

۳) کلمه ای مانند fly پویش شده است, اجزاء اصلی متناظر با عنوانهای کلمه ای ساخته می شوند:

N1: N → fly FROM 2 TO 3, and
V1: V → fly FROM 2 TO 3

٤) اگر گرامرشامل قوانینی مانند NP → ART ADJ N و جزء اصلی مانند ART1: ART → the FROM m TO n در جمله یافته شود سپس کمان فعال ARC1: NP → ART1 * ADJ N FROM m TO n به چارت یا نمودار فعال اضافه می شود.

علامت (*) در کمان فعال نشانه کرانه بین جزء اصلی یافت شده و جزء اصلی هنوز یافت نشده می باشد.

۵) پیش روی (*): اگر نمودا ریا چارت فعال کمان فعلی مانند زیر راداشته باشد:

ARC1: NP →ART1 * ADJ N FROM m TO n

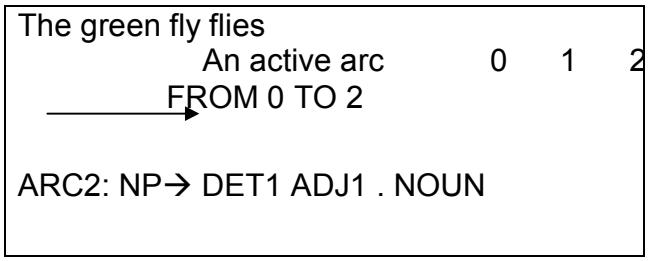
و یک جزء اصلی در نمودار نوع ADJ (مثلا" اولین آیتم بعد از *) وجود داشته باشد می گوییم:

ADJ1: ADJ → green FROM n TO p

اگر FROM در جزء اصلی با موقعیت TO در کمان فعال تطابق پیدا کند , سپس (*) می تواند پیشرفت کند و یک ئکمان فعال جدید تولید کند.

ARC2: NP → ART1 ADJ * N FROM m TO p

در شکل زیر کامن جدید از ۰ تا ۲ تولید شده اند.



٦) اگر پردازش برای پیش روی * یک کمان فعال تولید کند که * در قسمت دورتری در سمت راست قانون قرار بگیرد :همانند:

ARC3: NP → ART1 ADJ1 N1 * FROM 0 TO 3

سپس این کمان به یک جزأ اصلی تبدیل می شود:

NP1: NP → ART1 ADJ1 N1 FROM 0 TO 3

همه کمانهای فعال در این جهت کامل نمی شوند.

۷) هم اجزاء اصلی و هم عبارات می توانند در گامهای ٤ و ٥ استفاده شوند.مثلا" اگر گرامر شامل قانون S → NP VP باشد , به محض اینکه جزء اصلی NP1 در گام ٥ ام ساخته شد , ساختن کمان فعال جدید ممکن خواهد بود.

ARC4: S → NP1 * VP FROM 0 TO 3

۸) وقتی جزء اصلی بعدی ساخته شد ر می توانند نامهایی مثل NP2,NP3,ADJ2 و ... داشته باشند.

۹) هدف اصلی تجزیه بدست اوردن جزء اصلی عبارت (معمولا" از نوع S) که FROM آن 0 و TO آن طول جمله می باشد است.

نمونه ای از تجزیه چارت یا نمودار:(Chart parsing)

گرامر عبارتست از :

1 . S → NP VP
2 . NP → ART ADJ N
3 . NP → ART N
4 . NP → ADJ N
5 . VP → AUX V NP
6 . VP → V NP

گامهای تجزیه :

* sentence *: **the** * * position *:1
* constituents *:
ART 1: ART →the FROM 0 TO 1
* active – arcs *:
ARC1: NP →ART * ADJ N FROM 0 TO 1 [rule 2]
ARC2: NP →ART1 * N FROM 0 TO 1 [rule3]

* sentence *: the **large** * position *:2
* constituent *: add
ADJ1: ADJ → large FROM 1 TO 2
* active – arcs *:add
ARC3: NP →ART1 ADJ1 * N FROM 0 TO 2 [arc1*→]
ARC4: NP →ADJ1 * N FROM 1 TO 2 [rule4]

* sentence *: the large **can** * position*: 3
* constituents * :add
NP2: → ART1 ADJ1 N1 FROM 0 TO 3 [arc3*→]
NP1: NP → ADJ1 N1 FROM 1 TO 3 [arc4 * →]
N1: N → can FROM 2 TO 3
AUX1: AUX → can FROM 2 TO 3

V1: V → can FROM 2 TO 3
*active – arcs *: add
ARC5: VP → V1 * NP FROM 2 TO 3 [rule 6]
ARC6: VP → AUX1 * V NP FROM 2 TO 3 [rule 5]
ARC7: S → NP1 * VP FROM 1 TO 3 [rule1]
ARC8: S →NP2 * VP FROM 0 TO 3 [rule 1]

*sentence * : the large can can * position*:4
constituents : add
ARC9: VP → AUX1 V2 * NP FROM 2 O 4 [arc6 *→]
ARC10: VP → V2 * NP FROM 3 TO 4 [rule6]
ARC11: VP → AUX2 * V NP FROM 3 TO 4 [rule 5]

sentence : the large can can hold
position : 5
constituents: add
N3: N → hold FROM 4 TO 5
V3: V → hold FROM 4 TO 5
active – arcs : add
ARC 12: VP → AUX2 V3 * NP FROM 3 TO 5 [arc11* →]
ARC13: VP → V3 * NP FROM 4 TO 5 [rule 6]

sentence: the large can can hold the
position: 6
constituents : add
ART2: ART → the FROM 5 TO 6
active-arcs : add
ARC14: NP → ART2 * ADJ N FROM 5 TO 6 [rule 2]
ARC15: NP → ART2 * N FROM 5 TO 6 [rule 3]

sentence : the large can can hold the water
* position * : 7
* constituents * : add
S2: S → NP1 VP2 FROM 1 TO 7 [arc7 * →]
S1: S → NP2 VP2 FROM 0 TO 7 [arc8* →]
VP2: VP → AUX2 V3 NP3 FROM 3 TO 7[arc12 * →]

VP1: VP → V3 NP3 FROM 4 TO 7 [arc13 * →]
NP3: NP → ART2 N4 FROM 5 TO 7 [arc15* →]
N4: V → water FROM 6 TO 7
V4: V water FROM 6 TO 7
active-arcs: add
ARC16: VP → V4 * NP FROM 6 TO 7 [rule 6]
ARC17: S → NP3 * VP FROM 5 TO 7 [rule 1]

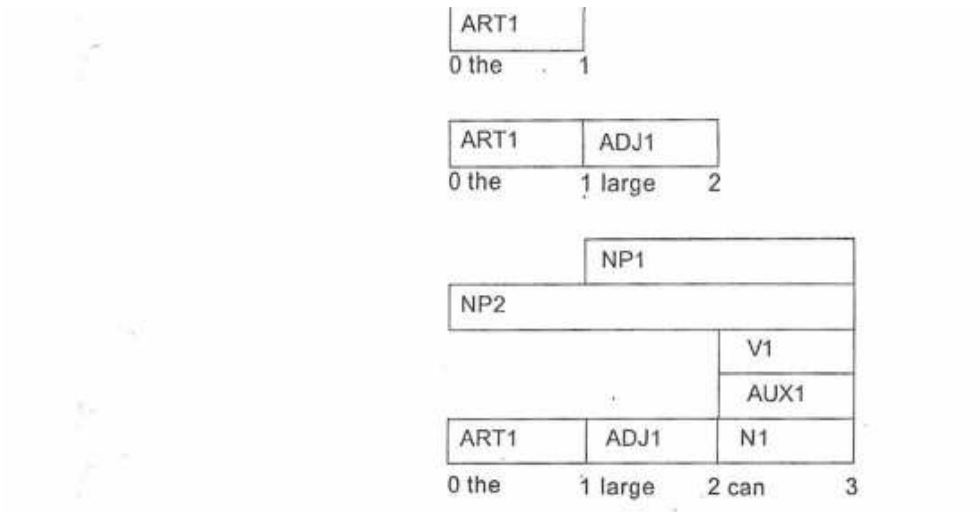
این گامها را به صورتی که در شکل ۲-۷ نشان داده شده است تکرار کنید.
عیب روش پایین به بالا این است که جز اصلی بی ارتباط را نیز پیدا خواهد کرد مانند "hold the water" که این جزء توسط روش تجزیه بالا به پایین مورد توجه واقع نخواهد شد.یک تجزیه کننده بالا به پایین مستقل از متن می تواند یک چارت داشته باشد.

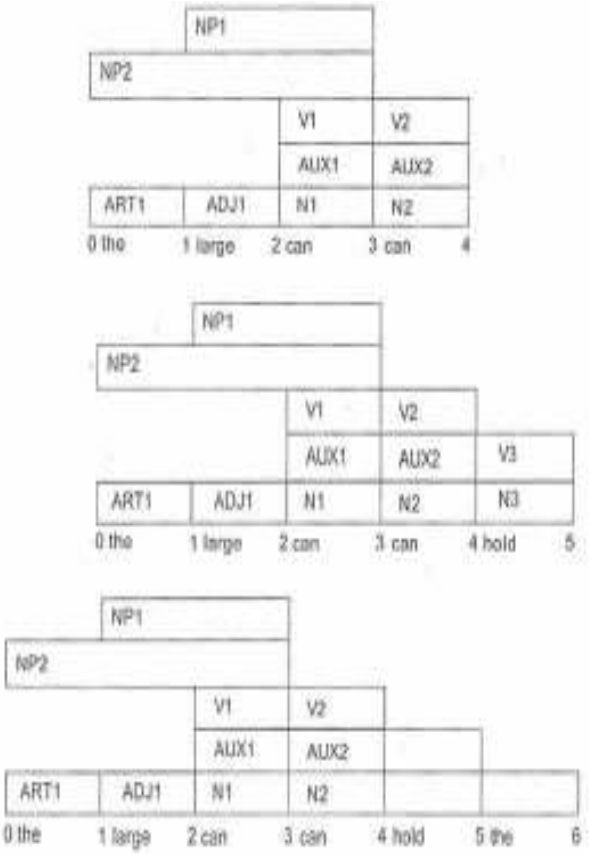
ثبت ساختار جمله(Recording Sentence structure) :

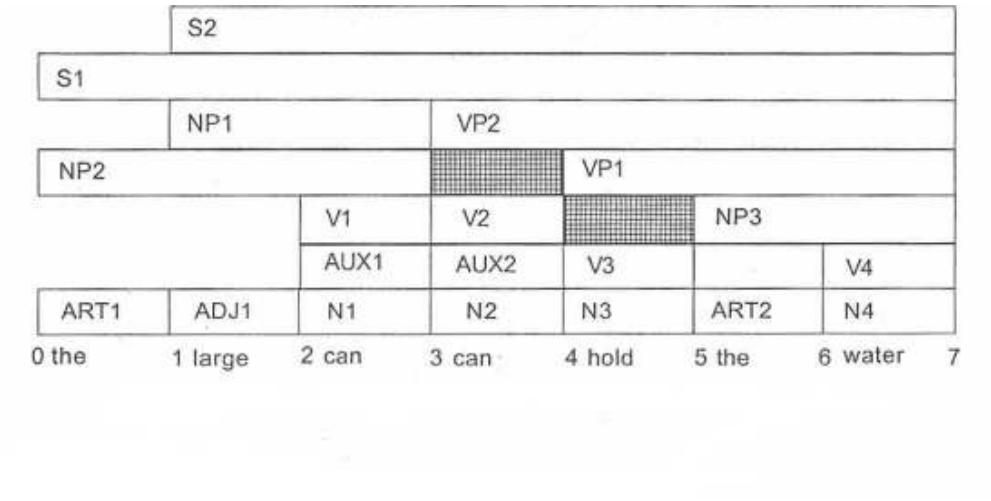
جمله زیر را در نظر بگیرید:

" Jack found a dollar"
(S SUBJ (NP NAME jack)
MAIN-V find
TENSE past
OBJ (NP ART a HEAD dollar))

S(subj(np(name(jack))),
Main(find),
Tense(past),
Obj(np (art (a),head (dollar))))


ART1
0 the 1
ART1
ADJ1
0 the 1 large 2
NP1
NP2
V1
AUX1
ART1
ADJ1
N1
0 the 1 large 2 can 3


NP1
NP2
V1
V2
AUX1
AUX2
ART1
ADJ1
N1
N2
0 the
1 large
2 can
3 can
4
NP1
NP2
V1
V2
AUX1
AUX2
V3
ART1
ADJ1
N1
N2
N3
0 the
1 large
2 can
3 can
4 hold
5
NP1
NP2
V1
V2
AUX1
AUX2
ART1
ADJ1
N1
N2
0 the
1 large
2 can
3 can
4 hold
5 the
6



شکل ۲-۷

۷-۳-۸ گرامرو برنامه نویسی منطقی

:(Grammars & Logic programming)

قوانین گرامر می تواند مستقیما" در PROLOG کدنویسی شود.

S → NP VP ……s(p1,p3) :- np(P1,P2), vp(P2,P3).

این عبارت بدین معناست که S وجود دارد از موقعیت P1 به موقعیت P3 به شرط اینکه وجود داشته باشد NP از P1 به P2 و VP از P2 به P3 .همینطور:

VP →V NP ……vp(P1,P3) :- v(P1,P2), np(P2,P3).
NP → NAME …….np(P1,P2) :-propernoun(P1,P2).
NP → ART N …….np(P1.P3) :- art(P1,P2), noun (P2,P3).

لغات می توانند توسط گزاره ها تعریف شوند.مانند:

NAME → John ….. isname(john).
V → ate ….. isverb(ate).
ART →the ….. isart(the).

N → cat ….. isnoun(cat).

برای هرعنوان لغتی مانند ART یک گزاره (predicate) تعریف می کنیم مثلا" برای ART

Art (FROM,TO) :_word (Word, FROM ,TO), isart (Word)

این مطلب زمانی درست است که کلمه ای که موقعیت آن مشخص شده , از همان عنوان باشد.

در مثال زیر:

Word (Word , FROM ,TO)

عبارت بالا مشخص می کند که word در جمله ورودی , بین موقعیت های FROM و TO قرار دارد.به هماین صورت:

```
Noun(FROM,TO) :- word(Word ,FROM , TO), isnoun(Word).
V(FROM ,TO) :- word (Word , FROM ,TO), isverb(Word).
Propernoun(FROM, TO ):- word(Word , FROM ,TO), isname(Word).
```

با استفاده از سیستمی که قبلا" توصیف شده بود,می توانیم موقعیت کلمات در جمله هایشان را ثابت کنیم.

```
Word(john ,1,2).
Word(ate ,2, 3).
Word(the ,3 ,4).
Word(cat, 4 ,5).
```

سپس از query پرولوگ استفاده کنید?s(1,5)

نتیجه آ ن ** بله خواهد بود.

برای درک ساختار جمله, آرگومان ها و استدلال های اضافی برای عبور از گلوگاهها اضافه کنید.

```
S(p1,p3, S(np,vp)) :- np(P1,P2,NP), vp(P2,P3,VP),
Vp(P1,P3,vp(v(Verb),NP)) :- v(P1,P2,Verb), np(P2,P3,NP).
Np(P1,P2,np(name))) :-proper(P1,P2,Name).
Np(P1,P3,np (art(Art),noun(Noun)))
Art(P1,P2,Art),
Noun(P2,P3,Noun).

Art(FROM,TO,Word) :- word(Word,FROM,TO),
isart(Word).
Noun(FROM,TO,Word) :-
WORD(Word,FROM,TO),isnoun(Word).
V(FROM,TO,Word) :- word(Word,FROM,TO),
isverb(Word).
Proper(FROM,TO,Word) :-
word(Word,FROM,TO),isname(Word).
```

۷-۳-۹ زبان نمایش دانش (Knowledge Representation Language)

موضوع نمایش دانش مرتبط با سیستم NLP که شامل اطلاعات کلی درباره جهانی که نیازمند زبانی قابل فهم است , می باشدو نوعی دانش مخصوص که با سیستم NLP مرتبط است.

زبان نمایش دانش یا (KRL=knowledge Representation language)

به زبان مخصوصی اشاره دارد که برای کدبندی دانش استفاده می شود.

مجموعه اطلاعات و دانشی که توسط سیستم استفاده می شود پایگاه دانش (KB) نام دارد.

نمایش بر پایه FOPC :

زبانی که استفاده می شود اشتراکاتی با زبان منطقی دارد.بخشی از زبان شامل ثابتها مانند john1 ,تابع های کاربردی مانند father (john1) و متغیرها مانند xو y می

باشد.توجه داشته باشید فرم منطقی زبان از ثابتها استفاده نمی کند , همه چیز با متغیرهای مستقل بیان می شد تا نمایش متن مستقل را نگه دارد.برای مثال , فرم منطقی (NAME j1 "john") در متن خاصی ممکن است به عنوان ثابت john1 درKB نمایش داده شود.

تعاریف محدود شده در KRL معنی پیدا می کنند همانطور که در زبان منطقی معنی دارند.

$\exists x : Person(x) Happy(X)$ There is a happy person
$\forall x : Person(x) Happy(X)$ All people are happy

عبارت دوم معادل است با $\forall x Person(x) \rightarrow Happy(X)$.

Many KR systems do not explicitly use quantifiers. Their variables are all tacitly
universally quantified, as in PROLOG. Thus eats(john1, X) :- fried(X). means "John
eats anything if it's fried". Literally anything, of course-if fried(cartyres1) is stored
in the KB, then eats(john1, cartyresl) is true.

تعداد زیادی سیستم KR به طور واضح و آشکار از کمیت سنج استفاده نمی کنند.متغیرهای آنها همگی به طور فراگیر سنجیده شده اند چنان که در PROLOG است.بنابراین

Eats(john1, X) :-fried(x) به معنی "john eats anything if it's fried." است.

ثابتهای SKOLEM و توابع:

متغیرهای با سور وجودی به وسیله تکنیکی به نام Skolemization بکاربرده شده

اند به طوریکه متغیرها را با ثابتهای جدید جایگزین می کند.برای مثال فرمول $\exists y \forall x loves(x, y)$
به صورت فرمول روبرو کدبندی می شوند. $loves(X, sk1)$ جایی که sk1 یک ثابت جدید است که بجای شئ ای که وجود را ثابت می کند قرار می گیرد.
وابستگی های سورها با استفاده از توابع جدیدی به نام Skolem function نشان داده می شود.به عنوان مثال فرمول $\forall y \exists x loves(x, y)$ به صورت فرمول loves(sk2(Y),Y) کدبندی خواهد شد.جاییکه sk2 یک تابع جدید است که به صورت بالقوه یک شئ جدید برای هر مقدار Y تولید می کند.

معنی شناسی(Semantics) :

وظایف زیادی تحت عنوان semantic وجود دارند.(semantic به معنای مطالعه معانی می باشد.برای مبحث مورد مطالعه ما , semantic به معنی پردازش تعیین معنی جملات ورودی است.)

- کلمه صحیح را برای معنی کردن انتخاب کنید.
- ابهامات را از آن بردارید.زمانی که از لحاظ نحوی تفسیرهای متعددی ممکن است , تفسیری را انتخاب کنید که معنی بدهد.

" امروز صبح ,جان سگ مرا دید که به طرف محل کارش رانندگی می کرد."

- مراجع ضمیر را مشخص کنید.

" بیل دوچرخه جان را خواست.
او آن را فروخت."

- نمایشی از معنی زبان ورودی را تولید کنید.این باز نمایی ممکن است تا حدی از ورودی متفاوت باشد.

"جان خرگوشش را به بیل فروخت.
آیا بیل یک حیوان خانگی خریده است؟"

- اطلاعات حذف شده را پر کنید.

" من امروز از کار دیرم شد.

ماشین من استارت نمی زد.

باتری آن تمام شده بود.

باتری قسمتی از اتومبیل من است.تمام شدن باتری باعث شد اتومبیل من استارت نزند.استارت نزدن ماشین باعث شد که من دیر سرکار برسم برای اینکه من معمولا" برای رفتن به سر کار از ماشین استفاده می کنم و اگر ماشینی استارت نزند , نمی تواند حرکت کند.

ابهامات(Ambiguity) :

ابهامات زیادی بیش از آنچه بتوانیم تصور کنیم در یک زبان طبیعی وجود دارد.

ابهامات لغوی:

یک کلمه می تواند نقش های متعددی در سخن داشته باشد و برای هر نقش دارای یک معنی به خصوص باشد.

تعداد ترکیبات معنایی , محصول تعدد معانی برای هر کلمه است.

ابهامات نحوی:

عبارات ممکن است به قسمتهای مختلفی از جمله مربوط باشند.مخصوصا" ترکیبات عطفی و عبارت گزاره ای قسمتهای پر زحمت هستند.

۷-۳-۱۰ مثالها

در این قسمت ما به دومثال برای رفع برخی مشکلات NL با استفاده از گرامر شرط قطعی

(DCG=Definite Clause Grammar) می پردازیم.

اولین مثال نشان می دهد که چگونه یک عبارت فعل-اسم (noun-verb) می

تواند از متغیرهایی که خود می توانند به صورت غیر ترمینال درDCG ظاهر شوند بدست آید.

فقط فعلهای زمان حال شرح داده خواهند شد.سپس ما به مشکلات ترجمه از یک زبان به زبان دیگر خواهیم پرداخت .

در اینجا یک جمله به زبان فرانسه تجزیه شده و در همان زمان ترجمه انگلیسی در یکی از متغیرها انجام شده است.

ما در ابتدا با معرفی کلی از نحو و ترجمه DCG در PROLOG آغاز می کنیم.بیشتر نسخه های PROLOG توانایی تعریف زبان و عملیات بر روی آنها توسط DCG را دارند. گرامر های DCG بسیار شبیه گرامر های مستقل از متن هستند ولی DCG ها صراحتا" به دلیل داشتن حساسیت نسبت به متن , قویترند.

آسانترین فرم (Simplest form):

پایه ای ترین فرم DCG اساسا" همان گرامر Context Free یا مستقل از متن است.

با یک مثال توضیح می دهیم.یکبار قوانین به مترجم PROLOG نشان داده می شود وآنها به شرطها و عبارات خالص در PROLOG ترجمه می شوند.

اگر فهرست نویسی انجام شد سپس ترجمه قابل مشاهده خواهد بود.(فهرستی از شرطها)

مثال ۱:

در اینجا گرامری برای تشخیص انواع فرم های اعداد داریم.اجازه دهید ببینیم چقدر به تعریف نرمال گرامرهای مستقل از متن نزدیک است؟

```
Digit → [0], [1], [2] ,[3] ,[4] ,[5] ,[6] ,[7] ,[8] ,[9].
```

Nat num → digit .

Nat num → (توالی از ارقام)

digit,nat num (

Int →nat

num.

Int → sign ,nat num.

Real → int.

Real →int,[.],nat num.

Sign → [-],[+].

اشیا درون " [] " عناصر ترمینال زبان هستند.شناسه های دیگر نقش غیر ترمینال را دارند.

قوانین در سمت راست می توانند توسط علامت | ازهم تمیز داده شوند.قوانین بالا به صورت زیر در پرولوگ ترجمه می شوند.

```
Nat num(A,B) :-
                Digit(A,B).
Nat num(A,B) :- digit(A,C),
Nat num(C,B).
Sign(A,B) :-
                ( “C” (A,-B) |
                  “C” (A ,+,B).
Real(A,B) :-
Int(A,B).
Real(A,B) :-
                Int(A,C),
                “C” (C,”.”,D),
                Nat num (D,B).
Real (A,B) :-
                Sign(A,C),
                “C” (C,”.”,D),
                Nat num(D,B).
```

```
Int(A,B) :-
                Nat num (A,B)
Int (A,B) :-
                Sign(A,C),
                Nat num (C,B).
Digit (A,B) :-
                (“C”(A,0,B ) |
                 “C” (A,1,B) |
                 “C” (A,2,B) |
                 “C” (A,3,B) |
                 “C” (A,4,B) |
                 “C” (A,5,B) |
                 “C” (A,6,B) |
                 “C” (A,7,B) |
                 “C” (A,8,B) |
                 “C” (A,9,B) ).
```

اگر ما nat num(A,B) را برداریم , سپس تفسیر این گزاره اینگونه خواهد بود که توالی عناصر B , قسمت انتهایی A می باشد.به عبارت دیگر زوج (A,B) را برای ایجاد لیست تفاضل استفاده می کنیم.

تا حالا مثال nat num([1,2,3],[]) درست بوده است وبه طور کلی تر عبارت مذکور برای هر x ای درست است.گزاره داخلی “C” برای تشخیص ترمینالها , و عبارت “C”(X,Terminal,Y) به این معنی است که ترمینال هد یا سر X , و دنباله آن Y است (در صورتی که صورت روبرو تعریف شده باشد. “C”([X|Xs],X,Xs)

برای آسانتر نمودن مثال بالا میتوانیم رویه ای بنویسیم که می تواند اعداد در نمایش نرمال را به یک توالی از سیمبولهای جداگانه تبدیل کند.به عنوان مثال عدد 123+ به صورت لیست [+,1,2,3] و عدد 1.23 به صورت [1,”.”,2,3] تبدیل خواهد شد.

توجه داشته باشید که وجود "." ضروری می باشد در غیر اینصورت پرولوگ آن را به

.

عنوان یک ترمیناتور در نظر خواهد گرفت و پیغام خطا خواهد داد.همچنین اگر عددی شامل علامت باشد باید به صورت یک اسم که نیازمند اتم است, نقل شود.(عدد +123 یک اتم نیست چون با + شروع شده).

```
Convert ti list(Atom,Lst):-
        Name(Atom,Atom Lst),
        % convert to list of asci numbers
         List to vals(AtomLst,Lst).
Asci to char(Asc,Chr):-
         Name(Chr,[Asc]).
         List to vals([],[]). % apply down a list
List to vals([H|T],[HV|TV]):-
         Asci to val(H,HV),
         List to vals(T,TV).
```

به جای استفاده کردن از گزاره های ترجمه شده برای تشخیص عناصر یک زبان , می توانیم همچنین از عبارات گزاره ای داخلی (built-in) استفاده کنیم.عبارت (NT,Lst) به این معنی است که L ام به وسیله غیر ترمینال NT تولید شده است.به عنوان مثال عبارت (real[1,2,".",4,5]) درست است.

اکنون گرامر بالا اعدادی که با صفر شروع می شوند مانند ۰۰۳ را می پذیرد.دگرگون ساختن گرامر توسط معرفی غیر ترمینال ها برای نپذیرفتن و رد چنین اعدادی خیلی مشکل نیست. تغییرات به صورت زیر می باشد:

```
Nonz digit →[1] |[2]| [3] | [4] | [5] | [6] |
                [7] | [8] | [9] .% a nonz digit is in 1-9
Digit → [0] | nonz digit.
Red nat num → digit | nonz digit, nat num.
Red int → red nat num | sign , red nat num.
Red real → red int | red int, [“.”], nat num.
```

برای عبارت (red nat num(0,0,2,3)) اشتباه ولی برای (nat num [0,0,1,2]) درست خواهد بود.

مثال پیچیده تر:

DCG همچنین این امکان را دارد که کدهای پرولوگ را در محفظه (body) جاسازی کند.این کار توسط جا دادن کد مورد نیاز در {} انجام می گیردوهر آنچه داخل {} باشد , غیر قابل تغییر توسط مفسر خواهد بود.پارامترها نیز می توانند به عنوان آرگومان در سمبول های غیرترمینال ظاهر شوند بنابراین نتایج می توانند در محاسبات دیگر استفاده شوند یا به عنوان side effect در تشخیص یا فهم یک زبان در نظر گرفته شوند.

مثال ۱:

پارامتری را به مثال قبلی اضافه کردیم بنابراین بعد از تشخیص عدد درست , آن پارامتر حاوی مقدار یا ارزش آن عدد خواهد بود.همچنین لازم است بدانیم چند تا رقم در عدد ما ظاهر می شوند.

مثلا" عبارت (nat num (N),[1,2,3,1]) متغیر N را به مقدار 1231 معرفی می کند.

```
Digit (0) →[0]. % a digit is in 0-9
Digit (1) →[1].
Digit (2) →[2].
Digit (3) →[3]. % etc.
….
Digit (9) →[9].
Nat num (N,1) →digit(N).% a natural number is a sequence of digits
Nat num(N,ND) → digit (D), nat num (N2,ND1),
                                % ND is the number of digits.
{plus (ND1,1,ND),
                Power ten(D,ND1,P),
                Plus (P,N2,N)}.
                Int(N,D) → nat num (N,D). % an intiger is a natural number
                % possibly with a sign
Int(N,D) →[+],nat num(N,D).
```

```
Int(N,D) →[-], namt num(N1,D),
                { N is – N1}.
Real(R) → int ( R ) . % a real is given in normal decimal notation
Real (R ) → int(I),[“.”] , nat num(N, ND),
                { neg power ten(ND,inv P),
                 ( I >= 0 → R is I+ N * InvP;
                 R is I – N * Inv P)}.
Real ( R) → [+],[“.”], nat num(N,D),
               { neg power ten(D,inv P),
               R is –N * invP}.
Real (R ) →[-],[“.”] , nat num (N,D),
                 { neg power ten (D,InvP),
                  R is – N * InvP }.
                 Power ten(D,0,D) :-!.
Power ten(D,E1,P) :-
                 E1 >0,
                Plus(E,1,E1),
                Power ten(D,E,P1),
                P is P1 * 10,! .
Neg power ten(0,1_ :- !.
Neg power ten(N,P) :-
                 N>0,
                 Succ(N1,N),
                 Neg power ten(N1,P1),
                 P is P1/10.
```

یک فراخوانی مثلا" real (R,[-2,3,”.”,45],[]) مقدار درست را بر می گرداند و R معادل است با 23.45- و یک فراخوانی عبارت (int(N,D),[-,1,2,3]) N را به عدد 123- معرفی نموده و D تعداد ارقام عدد که ۳ است را نشان می دهد.

مثال ۲:

```
% parse simple English sentence
% for time being just present as a list of identifiers
% e.g. [ the ,big ,cat,kicks,the,black,dog]
```

```
% first we have a straightforward generator
% will not repeat adjective  like big  big girl!
% also check whether we need an “an” or an “a”.
Adjnounph (CV) → noun(CV).
Adjnounph (CV) → adjective(CV,Adj), noun().
Adjnounph (CV) → adjective (CV,Adj),adjective(Adj2),
                    { Adj \== Adj2}
                     Noun ().
Nounphrase →det(CV),adjnounph(CV).
      Sentence → nounphrase, verb , nounphrase.
% now some explicit examples
Det(cons) →[the] | [a].
Det(vowel) → [an].
Verb → [hit] | [kicks] | [kisses].
Noun(cons) → [cat] | [boy] |[girl].
Noun(vowel) → [owl] | [ox].
Adjective(cons,big) → [big].
Adjective(cons,black) →[black].
Adjective(cons,brown) → [brown].
Adjective(cons,tabby) → [tabby].
Adjective(vowel,awful) → [awful].
Adjective(vowel,awesome) →[awesome].
```

گرامر بالا جمله "an awful owl hits an ox" را پذیرفته و جمله " an awesome cat kisses an awful girl" را رد خواهد کرد.

اجازه دهید به سیستم هایی بپردازیم که سعی می کنند انگلیسی (ترجیحا" هر زبان طبیعی)را به عنوان ورودی بفهمند .فقط این اواخر است که ما برنامه هایی داریم که می توانند از عهده جمله های بسیار پیچیده برایند و در حالی که بسیار بزرگ و کند هم می باشند .موفقیت های بیشتری می توانند وجود داشته باشند در صورتی که دامنه سخن محدود شود.همه برنامه های اولیه به این فرم بودند.

۷-۳-۱۱ الیزا (ELIZA)

این تلاشی است برای بازی کردن نقش یک روانپزشک (Rogerian therapist) که تنها با علم نحو (pure syntactic) کار می کند .

این روش مانند این است که روانپزشک چیزی را به بیمار خود تلقین کند نه اینکه بیمار را وا دارد تا احساسات و افکار خود را بروز دهد.الیزا لیستی از قالب ها یا الگوهایی دارد که سعی می کند سوالهای ورودی را با لیستی از کلمات کلیدی طبقه بندی شده یا منظم تطبیق دهد.

الگوها به یکی از فرم های زیر می باشند:

۱) کاراکتر های مطابق با کاراکترها

۲) <string> %

۳) <string> % < string>

رشته ها لیستی از کاراکترها می باشند که باید دقیقا" مطابقت داشته باشند و علانت ٪ با زیر رشته ای که باقی مانده , تطبیق می یابد.

مقدار ٪ برای دادن پاسخ معقولتر استفاده خواهد شد بنابراین ما احساس می کنیم که ماشین واقعا" ورودی ها را فهمیده است .البته فقط در وضعیت ۱ است که می دانیم کاربر دقیقا" چه چیزی را به عنوان ورودی وارد می کند.

اجازه دهید به قالبی که با فرم دوم مطابقت می کند بپردازیم. پاسخ های ممکن عبارتند از:

الف) چه فرقی می کند که من اعتقاد دارم یا نه؟

ب) شاید اعتقاد دارم و شاید هم ندارم.

ج) بله , اعتقاد دارم.

این پاسخ ها در جواب سوال زیر است:

آیا شما به خدا اعتقاد دارید؟ سپس یک پاسخ ممکن با توجه به پاسخ دوم, خواهد بود: شاید من به خدا اعتقاد دارم و شاید هم ندارم.

تکنیک hashing برای بازیابی سریع و تطبیق الگوها به کار برده می شود.برای مثال در اجرای یونیکس , ابتدا تطبیق در دو کاراکتر اول انجام می شود.اگر هیچ الگوی

تطبیقی وجود نداشته باشد سپس یک جستجو با استفاده از کلمات کلیدی در جملات ورودی انجام می شود.کلمات کلیدی مرتب شده اند.لیست پاسخ های ممکن برای کلیدی (difficult) می تواند به صورت زیر باشد:

a) tell me about your difficulties
b) what do you mean by difficult

اگر هیچ کلمه کلیدی در جمله نباشد که با کلمات کلیدی در پایگاه داده مطابقت داشته باشند سپس تعدادی ملاحظات یا تبصره های غیرالزامی استاندارد می تواند به صورت زیر ایجاد شود.

a) let's change the subject بیاید موضوع را عوض کنیم
b) go on ادامه بده!

یک رهیافت جایگزین , انتخاب تصادفی از یکی از کلمات کلیدی که قبلا" استفاده شده می باشد.رکوردی از ٤ کلمه کلیدی که قبلا" استفاده شده نگهداری می شود. بنابراین در اینجا سیستم از تاریخچه محدود شده استفاده می کند.این تکنیک اساسی است که الیزا از آن استفاده می کند به غیر از زمانی که یک قالب یا الگویی که با یکی از فرم های ۲ یا ۳ تطبیق یافته , کنار گذاشته شود. چند تا پیش پردازش باید برای تولید پاسخ درست از لحاظ گرامری اجرا شوند.

اگر کاربر ضمیر شخصی (من) را در ورودی ذکذ کند, دیگر در جایی که هست با ضمیر دیگری نمی تواند جایگزین شود.

فرض کنید الگویی وجود داشته باشد به صورت % because , همراه با یک قالب پاسخ.

% ? that's incredible!

سپس در ورودی داریم:

Because I am clever.

یک جایگزین صریح و آسان به صورت زیر نتیجه خواهد داد:

I am clever ? that's incredible !

سپس یک جایگزینی به جای i به صورت زیر خواهد بود.

You am clever? that's incredible!

البته این جمله هنوز درست نیست و باید تغییراتی روی آن صورت گیرد. you am به you are تغییر یابد.این دو مرحله fix و grammar نامیده می شوند.

Fix تغییرات زیر را انجام می دهد:

You → me
I → you
Me → you
That you → that I

همچنین تغییرات دیگری نیز انجام می گیرد.

در اینجا بعضی از خطاهای گرامری ساده اصلاح می شوند.

You am → you are
Me are → I am

با دیگر تغییرات مشابه.

مهم است که بدانیم مرحله fix قبل از مرحله grammar اجرا می شود.

Do you think I am clever?
⇓
May be I think I am clever and may be I don't.
⇓ *fix*
may be I think you am clever and may be I don't.
⇓ *grammar*
May be I think you are clever and may be I don't.

این روش نمی تواند تمامی خطاهای گرامری را رفع کند و نسبتا" برای آن آسان است که جوابهای غیر گرامری بدهد.برای مثال در اجرای یونیکس جاری , توجه نمی کند که چه چیزی باید به جای fix, you شود. پس برای سوءال زیر خواهیم داشت:

Do you belive in what you do?

جواب خواهیم گرفت:

May be I do belive in what **me** do or maybe I don't.

البته در اینجا آسان است که تغییرات fix را برای اصلاح کردن اعمال کنیم ولی مثالهای زیادی وجود دارند که می توانند برنامه را فریب دهند.تنها اطلاع دقیقی که می توانیم درباره برنامه ای بگوییم این است که شامل fix یا grammar است ولی هیچ قدرت قیاسی برای یک نمونه وجود ندارد.

به هیچ عنوان این فکر درستی برای با هوش بودن نیست.برای بهبود دادن , نیازمند fix های بیشتری برای مواجه شدن با موردهای جدید است.

۷-٤ شناسایی کلام یا گفتار۱

شناسایی سخن عمل تشخیص از بازسازی کلماتی که یک سیگنال صوتی معلوم را تولید می کنند است. به عبارت دیگر, مسئله ترجمه و انتقال سیگنالهای صوتی دیجیتالی کد شده یک گوینده در زبان طبیعی (انگلیسی) به متن آن در همان زبان می باشد.

با توجه به ابهاماتی که در سطوح مختلف این مسئله وجود دارد(مثل صدای پس زمینه, لهجه گوینده, صدای دیجیتال شدن و ...) می توانیم مسئله را رسما" به صورت زیر مشخص کنیم:

$$\text{Argmax}_{words}\ P(words \mid signal)$$

طوریکه words , رشته کلمات در زبان طبیعی مانند انگلیسی می باشد و سیگنال , ترتیبی از داده های صوتی گرفته شده که دیجیتالی شده اند می باشد.

۷-٤-۱ پردازش سیگنال (Signal processing)

---

1 Speech Recognition

صحبت کردن یکی از اصلی ترین و عمده ترین روشهایی است که انسانها برای ایجاد ارتباط از آن استفاده می کنند.کامپیوترها در طول سالها پیوسته در حال افزایش توانایی محاسبه بوده اند و در کنار آن قابلیت های شناسایی کلام نیز افزایش پیدا کرده است.

کلام به دو دسته عمومی تقسیم می شود:

۱) تشخیص و شناسایی کلام

۲) فهم کلام

با توجه به هدف و مقصود این مبحث, از آنجایی که تشخیص کلام زیر مجموعه ای از فهم کلام نیز می باشد, فهمیدن کلام مورد مطالعه قرار خواهد گرفت .

تشخیص کلام عمل نگاشت از سیگنالهای صوتی دیجیتالی شده به رشته ای از کلمات می باشد.سیستم باید به سه سئوال زیر پاسخ دهد:

۱) گوینده با چه صدایی کلام یا سخن را ادا کرد؟

۲) گوینده چه کلماتی را با آن صدای سخن می خواست بیان کند؟

۳) گوینده چه مفهومی را می خواست با بیان آن کلمات برساند؟

برای پاسخ دادن به سئوال اول , ابتدا باید فرق بین صداهای معمولی و صدای سخن یا کلام را از هم تمیز داد. همه زبانهای بشری , ترکیبی از ٤٠ تا ٥٠ صوت متمایز می باشند که phone نامیده می شوند. Phone صوتی می باشد که متناظر با یک مصوت واحد یا حرف صامت واحداست . به هر حال قابل ذکر است ترکیب مشخصی از حروف می توانند صوت خاص مانند

(*th*) را تولید کنند و ترکیب حروف به خصوصی می تواند تعداد مختلفی صوت , بسته به مفهوم یا متن را تولید کنند.(a در cat و rat).

ابتدا باید تمامی اصوات ممکن ار مشخص کنیم, سپس آنها را به روشی خاص توصیف کنیم و بنابر آن می توانیم به آسانی در دیکشنری صداها به آنها رجوع کنیم.آسانترین و واضح ترین روش, تجزیه و تحلیل اصوات توسط تمیز دادن آنها به وسیله فرکانس یا دامنه نوسان یا ... است.

سپس لازم است این خصوصیات را در کتاب مرجع یا دیکشنری جایگذاری کرد که از

آنجا می توانیم صداها را به درستی تشخیص دهیم.
بهترین روش مرتب کردن بر اساس تلفظ می باشد به دلیل اینکه spelling یا املاء دراین مقوله بی ربط و نامناسب می باشد.
سئوال بعدی این است که چگونه مشخص کنیم که گوینده چه کلماتی را می خواست به شنوندگان بگوید؟ با وجود اینکه ما یک لغت نامه یا دیکشنری بر اساس تلفظ مرتب شده داریم ولی هنوز مسائلی وجود دارد که باید حل شوند.یکی از مسائل تصادف یا وقوع هم صداها (homophones) می باشد به این معنی که دو یا بیشتر از ۲ تا کلمه وجود دارند که تلفظ یکسانی دارند مانند their,they're,there,... .مسئله دیگر مشخص کردن رشته درستی از کلمات که هیچ شکافی بین آنها وجود ندارد است.بنابراین ما باید روشی را برای تعیین این که چه وقت کلمه ای به پایان می رسد و چه وقت کلمه ای دیگر آغاز می گردد, بدست آوریم.اگر نتوانیم کلمات را از یکدیگر تمیز دهیم , لازم است دوباره نگاهی به لغت نامه برای هر صوت بیندازیم به این دلیل که ممکن است هر صوت جدید آغازکننده کلمه دیگری باشد.
پاسخ به سئوال آخر هنوز کاملا" حل نشده است.معنی کلماتی که بیان می شود چیست؟
برای پاسخ به این سوال ابتدا باید مشخص کنیم کدام کلمه ها گفته شده اند, سپس به تحلیل گر هایی که سعی دارند معنی را به کمپیوتر انتقال دهند واگذار کنیم.
سیگنال صوتی خام که توسط میکروفون گرفته می شود , درآغاز دیجیتالی (digitized) و پیش پردازش می شوند.Diditization شامل نمونه براداری از سیگنالهای آنالوگ معمولا" بین 8,16 khz می باشد و سپس درجه بندی هر نقطه نمونه معمولا" از 8 تا 12 بیت ارزش برای هر نقطه است.در قدم بعدی زیر دنباله های روی هم افتاده و دارای اشتراک داده های دیجیتالی شده پردازش می شوند.زیر دنباله های به طول حدودا" 10msec (شامل ۸۰-۱۶۰ نقطه داده ای) برای تعریف ترتیبی از فریم ها استفاده می شود.در داخل هر فریم مجموعه ای از خصوصیات که

بعدا" کشف خواهند شد, وجود دارد.برای مثال کل انرژی در یک فریم و اختلاف انرژی بین فریم جاری و فریم قبلی.

در یک فریم حدود ۸ تا ٤٠ ویژگی یافت خواهد شد.

این n-D بردار از ویژگی های خود بعدا" توسط پردازشی به نام Vector quantization, quantized یا درجه بندی می شود مثلا" 256 bin .

هر فریم یا چارچوب اکنون توسط یکی از ٢٥٦ برچسب ممکن توضیح داده می شود.نتیجه کل این پردازش توضیح فشرده ای از فضاهای روی هم انباشته از سیگنالهای صوتی است که برای شناسایی کلمه کافی می باشد.

اساس رهیافت شناسایی و تشخیص کلام استفاده از قانون بیز (Bayes) می باشد که مسئله را به قسمت های قابل مدیریت می شکند.

$$P(words \mid signal) = P(words)\, P(signal \mid words) / P(signal)$$

از آنجایی که ما سیگنال های دیجیتالی از نوع توصیف شده در بالا داریم و هدف ما پیدا کردن ترتیبی از کلمات می باشد که P(words | signal) رابیشینه می کند, P(signal) ثابتی برای ورودی های صوتی معلوم می باشد بنابراین می توانیم به آسانی این مسئله را رها کنیم .بنابراین هدف جدید ما محاسبه مقدار زیر است:

$$Argmax_{words}\, P(signal \mid words)\, P(words)$$

P(words) نشان دهنده مدل زبان می باشد که تصریح کننده احتمالات قبلی از رشته کلمات خاصی می باشد.بنابراین باید تعین کننده میزان احتمال وقوع برخی کلمات انگلیسی باشد.

P (signal | words) مدل صوتی می باشد که مشخص کننده احتمالهایی از اصواتی از کلمات که بیان شده اند, می باشد.این قسمت توسط حقیقتی که اکثر کلمات به راهها و طرق مختلف تلفظ می شوند, پیچیده تر می شود.قسمت بعدی جزئیات بیشتری در مورد مدل زبان و مدل صوتی را بیان می کند.

۲-٤-۷ مدل زبان (The Language Model)

p(words) احتمالهای قبلی از ترتیبی از کلمات words=$w_1w_2...w_n$. می باشد که محتمل تر در زبان طبیعی می باشند.مثلا" I have agun " محتمل تر از "I have a gub" می باشد.

یک راه برای بیان احتمال های وابسته به استفاده از قانون های زنجیره ای به صورت زیر می باشد :

$P(w_1,w_2.....w_n) = P(w_1)\ P(w_2 \mid w_1) \ldots P(w_{n-1} \mid w_1....,w_{n-2})\ P(w_n \mid w_1,...w_{n-1})$

اکنون ما نیاز به محاسبه احتمال کلمه اول در جمله (w_1) و محاسبه احتمال کلمه دوم (w_2) در صورتی که کلمه اول در جمله باشد داریم. در انتها محاسبه احتمال کلمه w_n است در صورتی که n-1 کلمه قبلی شامل کلمات $w_1....w_{n-1}$ باشند.این نوع بیان بسیار پیچیده است به دلیل اینکه نیازمند این است که ما احتمالهای شرطی از ترتیبی از کلمات ممکن را مشخص کنیم .اگر سری کلمات شامل n تا کلمه و زبان ما شامل m تا کلمه باشد, سپس برای محاسبه $P(w_n \mid w_1,....w_{n-1})$ نیازمند محاسبه m^{n-1} حالت از ترتیب کلمات (شروع کننده جمله) ممکن می باشد.

به جای این کار می توانیم از فرضیه آسانی به نام First-order Markov استفاده کنیم به این صورت که احتمال وقوع یک کلمه فقط وابسته به کلمه قبلی خود می باشد.

$$P(w_n \mid w_1,....,w_{n-1}) = \approx P(w_n \mid w_{n-1})$$

با استفاده از این فرضیه اکنون ما بیان ساده تری برای محاسبه احتمالات متصل داریم.

$$P(w_1w_2.....w_n) = P(w_1)P(w_2 \mid w_1)....P(w_n \mid w_{n-1})$$

این مدل ساده شده مدل bigaram نامیده می شود به خاطر اینکه مربوط به زوج های متوالی از کلمات می باشد واین مدل کمترین مقدار متن برای تعیین احتمال هر کلمه در جمله را فراهم می سازد.واضح است استفاده بیشتر از متن, صحیح تر می باشد ولی

.

برای محاسبه هزینه بیشتری می طلبد.(به عنوان مثال فرضیه second order Markov که در این صورت مدل trigram خواهیم داشت.)

می توانیم جدولی برای نمایش مدل bigram توسط محاسبه احتمالهایی از توالی زوج کلمه های ممکن در یک مجموعه بزرگ آزمایشی کلمات ایجاد کنیم.به عنوان مثال اگر حرف (a) ۱۰۰۰۰ بار در مجموعه آزمایشی ظاهر شده و (a) توسط کلمه (gun) ۳۷ بار دنبال شود بنابراین (gun | a) برابر با 37/10000 یا 0.0037 می باشد.

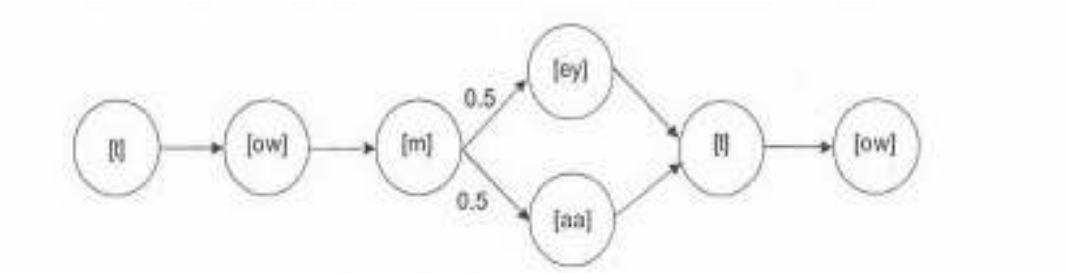
به جای نمایش مدل bigram در یک جدول می توانیم از آتاماتای احتمالی محدود(Probabilistic finite automata) استفاده کنیم.گره ای (حالت) برای هر کلمه ممکن ایجاد کنید و کمانی از هر گره به همه گره های دیگر رسم کنید .هر کمان را با احتمالی که کلمه با گره منبع(مبداء) مربوط است و پیرو کلمه ای که با گره مقصد پیوسته است, ارزش گذاری یا برچسب گذاری کنید.در انتها گره ای به نام شروع یا start اضافه کنید و کمانی از آن به همه گره ها بکشید.این کمانها را با احتمالی که کلمه با گره مقصد در ارتباط است و می تواند یک جمله را شروع کند ارزش گذاری کنید.

ما می توانیم از این (FSM)Finite state Machine برای تعیین احتمال جمله ای که با گره start شروع می شود و به گره هایی که مربوط به کلمات متوالی در جمله می باشد و گذار پیدا می کند, استفاده کنیم.

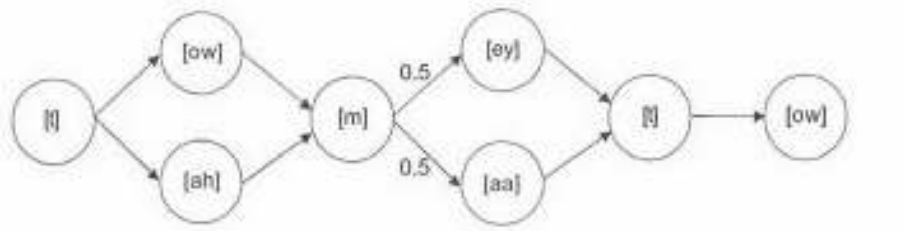
۷-٤-۳ مدل صوتی ((The Acuostic Model P(signal | words)):

مدل صوتی به ما می گوید که چه صدایی پس از ادای کلمات شنیده خواهد شد.مدل Markov به ما تلفظ های متناوب کلمات را نشان می دهد .این تحول و گذار فقط مربوط به حالت جاری است نه به حالتهای قبلی .این مدل تمامی راههای ممکن را برای رسیدن به هدف, براساس حالت کنونی آن و نه حالتهای قبلی , نشان می دهد.

شکل شماره ۳-۷ مدل مارکوف



Co- articulation زمانی اتفاق می افتد که شخص آرام ی اتند حرف می زند و صداها را می آمیزد ومبهم می کند و اساسا" اندکی تلفظ آنها را عوض می کند .این به انسانهایی اشاره می کند که خیلی تند حرف می زنند و برخی کلمات را با هم می آمیزند یا مانند انسانهایی که لهجه دارند.



شکل شماره ٤-٧ co- articulation

مدل پنهانی Markov (The Hidden Markov model) سعی می کند به اختلاف در گفتار توجه کند و سعی می کند $P(words \mid phones)$ و $P(signal \mid phones)$ را کمینه کند همچنین به خطاهای ممکن یا تکه هایی از گفتار که در حالت تعلیق (hang) می باشد , توجه می کند.

فصل هشتم

سیستم خبره

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

■ با برنامه نویسی مبتنی بر هوش مصنوعی که در سیستمهای مبتنی بر دانش تجلی یافته است آشنا میشویدو بر تفاوتهای آن با بر نامه نویسی معمولی پی میبرید.

■ تعاریفی از سیستمهای خبره و همینطور معماری و خصوصیات آنها را مشاهده میکنیدو مختصرا" در مورد تاریخچه آن اطلاع کسب میکنید

■ با مفهوم مهندسی دانش و دانش دامنه آشنا میشویدو بطور کامل در مورد مراحل اکتساب دانش آگاهی بدست می آورید.

■ در مورد رویه ی استنتاج در حساب گزاره ای و مسندی و در سیستمهای تولید مبتنی بر قانون خواهید خواند.

■ باابزارهای توسعه سیستمهای خبره که در دسترس قرار دارند مانند زبانهای الگوریتمیک و زبانهای سمبولیک آشنا خواهید شد.و در مورد زبانهایی مانند پرولوگ و لیسپ که به عنوان زبانهایی برای برنامه نویسی هوش مصنوعی هستند مطالبی خواهید خواند.

■ و در آخر چند نمونه از کاربردهای سیستمهای خبره مانند والیزاو مایسین و غیره راخواهید دید. PARRY

۸-۱ مقدمه

در سالهای اخیر تحقیقات در زمینه هوش مصنوعی , موفقیت های بزرگی را کسب نموده است. درمیان همه این موفقیت ها بارزترین آنها توسعه سیستم های کامپوتری قدرتمندی به نام سیستمهای خبره یا سیستمهای مبتنی بر دانش بوده است.برنامه هایی طراحی شده اند تا دانش واقعی در زمینه های خاص تخصصی را برای حل مسائل مورد استفاده قراردهند.به عنوان مثال تلاشهایی با همکاری متخصصین و توسعه دهندگان صورت گرفته است که نتیجه آن طراحی سیستمهای تشخیص بیماری, پیکربندی سیستمهای کامپوتری و اکتشاف کانیها و معادن با کیفیت اجرای برابر با کیفیت کار متخصیصین واقعی بوده است.

سمبولهایی که در سیستم مبتنی بردانش استفاده می شوند برای بازنمایی مجازی هر نوع شیء اعم از مصنوعات,انسانها, اجسام , ارگانهای زیستی, کلاسهای مختلف اشیا, مفاهیم و ....است.

مطلب قابل توجه در اینجا این است که ما قصد مشاهده و دستکاری سمبولها به عنوان چیزی غیر از اعداد محض را داریم. بنابراین ما به خصوصیا ت و تسهیلات زبانهای برنامه نویسی متفاوتی نیاز داریم.

از تفاوتهای دیگر بین برنامه نویسی الگوریتمیک معمول و برنامه نویسی مبتنی بر هوش مصنوعی- که در سیستمهای مبتنی بر دانش تجلی یافته است- می توان روش برنامه نویسی را نام برد.در برنامه نویسی معمولی که اغلب برنامه نویسی رویه ای نیز نامیده می شود, ما به کامپیوتر باید بگوییم که با داده هایی که به صورت متوالی وارد کامپیوتر می شود چه کار کند. بنابراین در برنامه نویسی رویه ای , رویه ها بازنمایی چگونگی انجام کاری هستند که ما می خواهیم.

از سوی دیگر , در برنامه نویسی مبتنی بر دانش , ما در حوزه خاصی, دانش را به صورت اخباری بازنمایی می کنیم. ما تنها آنچه را که می دانیم بازنمایی می کنیم بدون اینکه دقیقا" و به صورت پیشرفته به چگونگی استفاده از آن توجه کنیم.

جدول ۱-۸ تفاوتهای بین این دو نوع برنامه نویسی را خلاصه می کند.دو نکته در اینجا خارج از مبحث ما هستند.اول, سازگاری با دو حالتی بودن رویه ای در مقابل روش اخباری که توضیح داده شد، به این معنی که در برنامه نویسی رویه ای ما درباره تغییر و

دستکاری داده ها صحبت می کنیم در حالی که در برنامه نویسی اخباری ما درمورد بازنمایی دانش بحث می کنیم. نکته دوم و مهم تر صفت جداسازی دانش از کنترل است که مشخصه ای از برنامه نویسی مبتنی بر دانش است.

نرم افزار مبتنی بر دانش	برنامه نویسی معمول
• قابلیت اداره کردن داده های گم شده یا غیرقطعی • تصمیم گیری ترتیب وتوالی توسط موتور استنتاج • مناسب برای تغییر سیمبولها • واسط های زبان طبیعی • امکان توضیح در حین اجرا • سیستم خبره = مسئله + کنترل + داده	• نیازمند داده های صحیح است • ساختار رویه ای ثابت • مناسب برای پردازش عددی • فقط یک برنامه نویس آن را می فهمد • توضیح در حین اجرا غیر ممکن • برنامه = الگوریتم + داده

۸-۱-۱ تعاریف:

تعاریف مختلفی از سیستمهای خبره وجود دارد هرچند که بسیاری از آنها مشابه هستند.پرفسور فایگن بام از دانشگاه استنفورد یک سیستمهای خبره را اینگونه توصیف نموده است:

" یک برنامه کامپیوتری هوشمند است که از دانش و رویه های استنتاج برای حل مسائل دشواری که نیازمند کارشناسان خبره هستند, استفاده می کند .دانش مورد نیاز برای اجرا در چنین سطحی به همراه رویه های استنتاج مورد استفاده, می توانند به عنوان مدلی از متخصصین و کارشناسان در یک زمینه خاص در نظر گرفته شود."

یک تعریف ارائه شده که خیلی هم مورد استفاده قرار گرفته , تعریفی است که از سوی " گشینگ , ریبو و رایتر " ارائه شده است:

" سیستمهای خبره, برنامه های کامپیوتری دارای اثر متقابل هستند که عمل تلفیق

قضاوت, قوانین , شهود و دیگر تخصص ها را برای تامین یک توصیه قابل درک و زیرکانه در امور مختلف , انجام می دهند."

یک تعریف جامع تر از سیستمهای خبره را برکمن ارائه داد:

" یک سیستم خبره سیستمی است که دارای قوانین کارشناسانه است و از جستجوهای کورکورانه اجتناب می کند, با تغییر و دستکاری سیمبولها استدلال می کند, اصول اساسی و بنیادی در یک حوزه خاص را می فهمد و می یابد , متدهای ضعیف استدلال برای عقب نشینی در مواقعی که قوانین خبره جوابگو نیستند دارد .

با مسائل دشوار در زمینه های پیچیده سروکار دارد. می تواند تشریحی از یک مسئله به صورت عبارات غیر تخصصی گرفته و آنها را به بازنمایی داخلی مناسب برای پردازش با قوانین کارشناسی و تخصصی خود تبدیل کند. می تواند برای دانش درونی خود نیز استدلال کند, به خصوص برای بازسازی منطقی مسیرهای استنتاج برای توضیح و تفسیر و توجیه کردن خود."

سیستمهای مبتنی بر دانش سعی در یافتن تکنیک ها و روشهای ساخت سیستمهای انسان – ماشین با تخصص های خاص حل مسائل است.

تخصص شامل دانش دریک زمینه به خصوص , فهم وتشخیص مشکلات آن حوزه و مهارت در حل بعضی از آنهاست. معمولادانش در هر رشته تخصصی بر دو قسم است:

- دانش خصوصی
- دانش عمومی

۸-۱-۲ دانش عمومی:

دانش عمومی شامل توصیفات,حقایق و تئوری های منتشر شده است که در کتابهای درسی , روزنامه ها و مقالات تحقیقی غیره است.

۸-۱-۳ دانش خصوصی:

دانش عمومی تنها منبع متخصصین انسانی نمی باشد. متخصصان معمولا" دارای دانش خصوصی هستند.

دانش خصوصی به طور گسترده ای شامل قوانین thumb به نام کشف کنندگی هستند. کشف کنندگی ها متخصصین را قادر می سازد تادر صورت لزوم، حدس های آگاهانه بزنند, تشخیص رهیافت های محتمل برای حل یک مسئله وسروکار با داده های

ناکامل و پر از خطا به صورت موثر است.

۸-۲ مهارت در مقابل دانش

مهارت در حل مسائل ، عموما دلالت بر سرعت, کارآیی , خطاهای کاهش یافته , بار ادراکی کاهش یافته, نیرومندی و غیره دارد .از سوی دیگر , دانش به انسان اجازه می دهد بواسطه قیاس, ادراک و شعور متعارف ,تجزیه و تحلیل و غیره به حل مسائل جدید بپردازد.حد اقل تا کنون مسائل دشوار و جالب توجه راه حل های الگوریتمیک نداشته اند. بسیاری از امور مهم درمتون و مفاهیم اجتماعی – فیزیکی رخ می دهند, که عموما" در مقابل توصیفات دقیق و تجزیه و تحلیل های سخت مقاومت می کنند .به عنوان مثال طرح ریزی استدلال های مشروع تشخیص های پزشکی, امکانات عیب شناسی ماشینهای تحلیل موقعیت نظامی و غیره.

انسانهای کارشناس به کارایی برجسته ای دست می یابند زیرا آنها مطلع و با تجربه اند. دلیل سوم برای متمرکز شدن بر دانش , تشخیص ارزش ذاتی خود آن است.دانش یک منبع کمیاب است که تکثیر و پالودگی ان , توانگری به همراه می آورد.دانش بشری توسط آموزش و کارآموزی انتقال و پخش میشود.

با استفاده از دانش , انسان ها به موفقیت های قابل توجهی در حل مسائل دشوار دست یافته اند.در صورتی که این دانش از متخصصین اقتباس شده و در یک برنامه کامپیوتری قرار گیردکه استفاده شود, سپس آن برنامه هم باید به سطح بالایی از کارایی نائل آید.اقتباس دانش از کارشناسان و قرار دادن آن در فرم های قابل محاسبه , هزینه های تولید مجدد دانش و بهره برداری از دانش را به نحو قابل توجهی خواهد کاست.همزمان , دانش خصوصی به واسطه قرار گرفتن در دسترس عموم برای ارزیابی و آزمایش می تواند باعث سرعت گرفتن پیشرفت علم شود.

در بیشتر موارد در روش های حل مسئله مبتنی بر دانش یا هوشمند , ما با مسائلی روبرو هستیم که راه حل های رسمی یا الگوریتمیک ندارند.بنابراین روشهای کشف کنندگی باید استفاده شوند .راه حل های موءثر بستگی به استفاده به هنگام و به جا از دانش برای شناسایی تصمیمات بالقوه ای دارند که می توانند نتایج محتمل و امیدبخش به بار آورند و اتنخاب های غیر ثمر بخش را حذف کنند.این ایده بنیادی از هر روش حل مسئله مبتنی بر دانش می تواند به صورت زیر خلاصه شود:

۱) دانش = حقایق + عقاید + کشف کنندگی
۲) موفقیت = یافتن یک روش حل مسئله خوب با در اختیار داشتن منابع در دسترس.
۳) بازده جستجو مستقیما" بر موفقیت تاثیر می گذارد.
٤) عوامل موثر بریک راه حل کارامد:

- دانش قابل اجرا , صحیح و قابل تفکیک و تمایز.
- حذف سریع دیدگاه های غیر ثمر بخش
- منابع دانش مضاعف مشترک
- تقسیم راه حل هل به سطوح متفاوتی از تجرد

۵) مشکلات اساسی در رهیافت مبتنی بر دانش:

- دانش اشتباه یا خطا
- وجود امکان های مختلف برای ارزیابی کردن
- وجود رویه های پیچیده جهت حذف احتمالات
- مسائلی که به صورت پویا تغییر می کنند

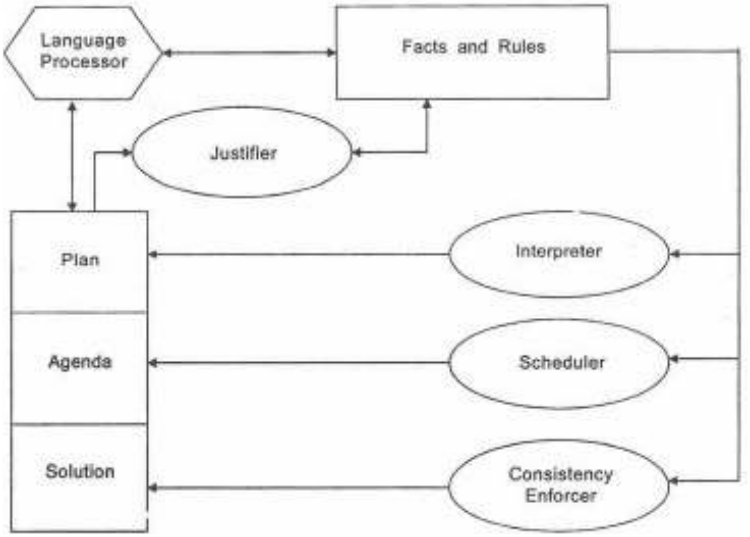
بر اساس تجربه مان در حل مسائل هوشمند, ایده اساسی می تواند بر اساس موارد زیر تشریح شود:
۱) ساخت راه حلها به صورت کارامد وبه طور انتخابی از فضایی از پیشنهادات متناوب
۲) شناسایی راه حل های مفید و سپس کاوش بیشتر در آنها
۳) هرس کردن راه حل تا رسیدن به بهترین راه حل

یک حل کننده مسئله ایده آل باید موارد زیر را داشته باشد:
۱) پردازنده زبان برای مکالمات مبتنی بر مسائل
۲) چرکنویس برای ثبت نتایج میانی
۳) دانش در مورد یک حوزه که می تواند شامل حقایق , کشف کنندگی و عقاید باشد
٤) دانش برای چک کردن ثبات یک راه حل پدیدار شده
۵) دانش برای برنامه ریزی استراتژی راه حل مسئله بعدی

.

٦) دانش برای ارزیابی راه حل های جزئی

حل کننده های اولیه مسائل هوشمند به سمت سیستم های خبره رشد کردند.سیستم های خبره , سیستم های مبتنی بر دانش ساده شده اند زیرا کل دانش حل مسئله در هر زمینه ای بسیار پیچیده است و نمی تواند توسط ساختارهای نحوی ساده تسخیر شود.بنابراین دامنه مسئله باید تنگ تر شود .و بنابراین همه اهداف کاربردی و عملی دانش محدود شده تا بتواند در زبانهای بازنمایی دانش ساده شده تسخیر شود.قبل از اینکه به جزئیات سیستم های خبره بپردازیم , شایسته است که ابتدا بتوانیم تفاوتهای بین برنامه های معمولی و سیستم های خبره را بشناسیم .این موضوع یک دورنما یا منظر متناوب برای توصیف سیستم های خبره به ما می دهد.



(شکل ۱–۸) معماری ایده آل سیستم خبره

۸–۲–۱ چگونه سیستم های خبره از برنامه های معمولی تمییز داده می شوند؟

اساسی ترین تفاوت میان این دو این است که سیستم های خبره دانش را تغییر می دهند در حالی که برنامه های معمولی داده ها را دستکاری می کنند.(جدول ۲–۸)

پردازش دانش	پردازش داده

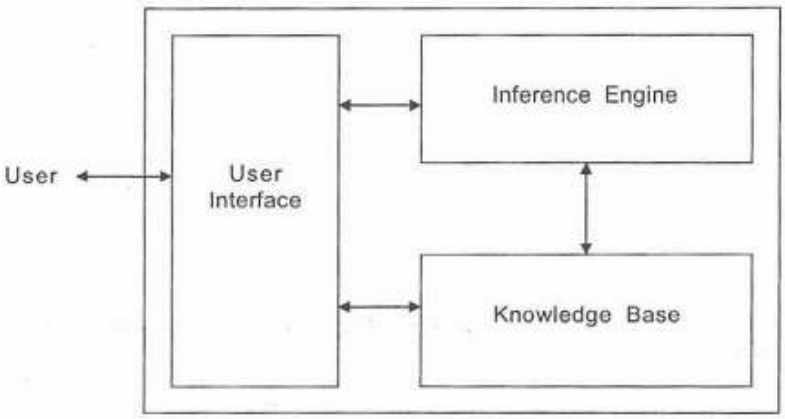
• بازنمایی و استفاده داده + کنترل = دانش • کشف کنندگی • پردازش های استنتاجی • کنترل گسترده و مقدار کم داده با هم نگه داری می شوند.	• بازنمایی و استفاده از داده های استاتیک • الگوریتم ها • پردازش تکراری • برای کنترل و مقدار زیاد داده جداگانه نگه داشته می شود.

به هرحال تفاوتهای اساسی بین پایگاه های داده و پایگاه دانش به صورت زیر است (جدول ۳–۸):

پایگاه دانش	پایگاه داده
• اطلاعات در سطح بالاتری از تجرد قرار دارند. • بیشتر بر روی کلاسی از اشیا عملیات انجام می دهد تا یک شئ واحد • از قدرت استنتاجی بهره مند است • بازنمایی به وسیله منطق , قوانین یا فریم ها یا مستندات یا شبکه های معنایی انجام می شود. • مورد استفاده برای تحلیل داه ها و برنامه ریزی	• مجموعه ای از داده های نمایش دهنده حقایق • فقط بر روی یک شئ واحد عمل می کند • اطلاعات باید صراحتا" توصیف شده باشند • به صورت سلسله مراتبی یا ارتباطی یا براساس مدل شبکه نمایش داده می شود. • برای اهداف عملیاتی ابقا می شود

.

شکل ۲-۸ معماری یک سیستم خبره نوعی را نشان می دهد.



(شکل ۲-۸) ساختار یک سیستم خبره

۸-۳ خصوصیات اولیه یک سیستم خبره

یک سیستم خبره باید موارد زیر را داشته باشد:

۱) متخصصی که باید کارایی تخصصی و کارشناسانه داشته باشد.آنها باید صاحب درجه بالایی از مهارت باشند و به اندازه کافی قدرتمند باشند.

۲) به دلیل اینکه دانش آنها سمبولیک است , استدلال سمبولیک را اجرا کنند.

۳) آنها باید قادر باشند قوانین پیچیده را استفاده کنند و دامنه های مشکل مسائل را اداره کنند.

٤) self-knowledge آنها باید قادر به تست و امتحان کردن قدرت استدلال خود باشند و بتوانند عملیات خود را توضیح دهند.

میزان مفید بودن یک سیستم خبره مستقیما" وابسته به کیفیت دانش آماده شده توسط طراحان است.بنابراین بسیار مهم است که سیستم به صورت بازگشتی تصفیه و معتبر شود.

۸-٤ تاریخچه مختصری درباره سیستم های خبره

در کنفرانس معتبر و بین المللی در زمینه هوش مصنوعی در سال ۱۹۷۷ , پرفسور فایگن بام , در یک مقاله , کلید بینش یک سیستم خبره را ارائه داد:" قدرت یک سیستم

خبره به دلیل اشتقاق از دانشی است که دارا می باشد نه از یک فرمالیسم خاص یا تمهیدات استنتاجی"

در سالهای نخست هوش مصنوعی , تصور می شد که همراه کردن چند قانون استدلال با یک کامپیوتر قدرتمند می تواند سیستم خبره ای تولید کند که قادر به اداره هر نوع مسئله ای در هر حوزه ای باشد.مانند (GPS=General Problem Solver) .با افزایش تجربه در این زمینه , قدرت شدیدا" محدود GPS، در نهایت منجر به درک این موضوع شد که GPS برای حل مسائل پیچیده بسیار ضعیف است. به همین دلیل کارشناسان نسبت به ساخت حل کننده های عمومی مسائل، بیشتر شروع به تفکر در زمینه مسائل با دامنه های محدودتری نمودند

در اواسط دهه ۷۰ , سیستمهای خبره متعددی پدیدار شدند.تعداد محدودی از محققان که نقش اصلی دانش در این سیستم ها را متوجه بودند شروع به تلاشهایی برای توسعه تئوری های جامع بازنمایی دانش و سیستم های چند منظوره نمودند.آنها نیز موفق نشدند به این دلیل که دانش نمی تواند به وسیله ساختارهای متناهی تسخیر شود زیرا که دانش بسیار متنوع و گسترده است.

از سوی دیگر رهیافت های مختلف دیگری برای بازنمایی دانش در آمدند که برای حل مسائل در زمینه های خاصی طراحی شده بودند.

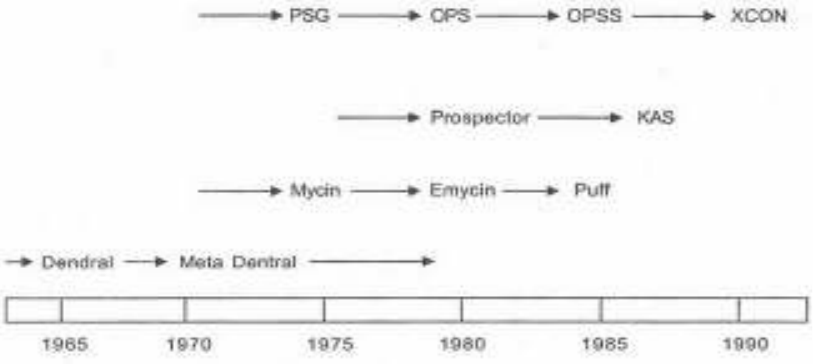
۸-٤-۱ منظور ما از دانش در یک دامنه چیست؟

در یک نگاه مجرد وعمومی , دانش شامل توصیفات , ارتباطات و رویه ها در یک زمینه مورد علاقه است.

توصیفات در یک زمینه دانش که اشیا و کلاسها را شناسایی و تمییز می دهد , شامل جملات به یک زبان خاص است .زبان می تواند به صورت یک سیستم رسمی مانند منطق کلاسیک (مانند منطق حساب گزاره ای , حساب مسندی یا خبری) باشدطوری که هر موجودیت خوش ترکیب است و یک علامت تفکیک معنایی مشخص دارد یا زبان اصلاح شده همانطور که در منطق غیر رسمی وجود دارد مانند منطق قراردادی , منطق مدل , منطق غیر یکنواخت , منطق احتمالاتی و منطق شولا یا fuzzy است که همگی از منطق کلاسیک مشتق شده اند.همین طور به وسیله اضافه نمودن قوانین

استدلالی خاص یا به وسیله پیوستن یک مدل کانونیکال.هر سیستمی دارای فواید و عیوب خاص خودش است و هیچ کدام از آنها برای بازنمایی کل دامنه دانش به اندازه کافی مناسب نیستند.
همچنین ارتباطات خاصی میان این توصیفات در یک زمینه دانش وجود دارد .این موضوع وابستگی ها و تجمع میان موجودیت ها را بیان می کند .این ارتباطات نوعا" می توانند پیوستگی های رده بندی شده, تعریفی یا تجربی باشند. از سوی دیگر رویه ها , عملیات قابل اجرا بر روی این موجودیت ها را در هنگام حل یک مسئله مشخص می کنند.

در عمل، دانش در یک فرم ناگهانی پدیدار نمی شود بلکه به زیبایی با عناوین مجردی تطابق می یابد.دانش مانند یک ماده تصفیه نشده است .به عنوان مثال می تواند در حوزه علوم تجربی همانند تشخیص های پزشکی , زمین شناسی و غیره باشد.جایی که برای یک داده مشاهده شده می تواند دلایل زیادی وجود داشته باشد. همچنین می تواند به فرم کشف کنندگی (اکتشافی) , محدودیت دار و تنظیمی باشد.
هنر جمع آوری و پردازش دانش , مهندسی دانش نامیده می شود. ارزیابی یک سیستم خبره در شکل ۳-۸ آمده است.



(شکل ۳-۸) توسعه در سیستم های خبره

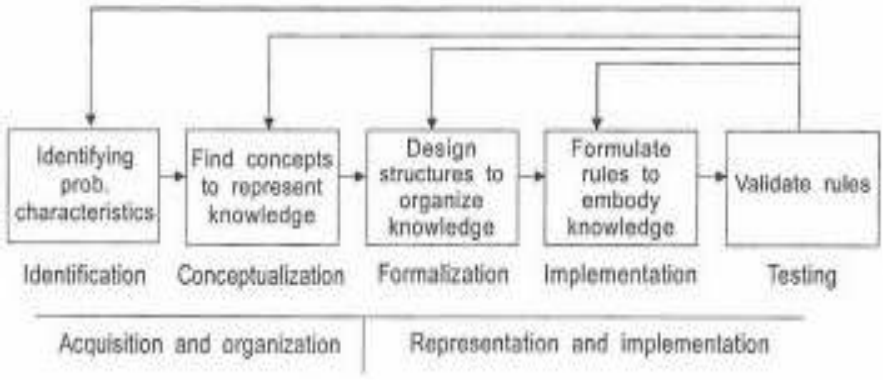
۸-۵ مهندسی دانش

مهندسی دانش یک نام بی مسمی است و باید به هنر یا مهارت دانش تغییر نام دهد.

آن به صورت گسترده شامل:
شناسایی و ادراک مسئله است . ادراک به معنی شناسایی مفاهیم کلیدی , رابطه ها , رویه ها و موارد ویژه است.
رسمی سازی , اجرا و آزمایش است.

۸-۵-۱ مراحل اکتساب دانش

حصول دانش از منابع مختلفی از قبیل کارشناسان , کتاب ها و غیره است که در آن مهندس دانش به واسطه چندین مرحله و طبقه قبل از تولید و سیستم های خبره شرح داده شده در شکل ٤-٨ پیش می رود.این مراحل حقیقتا" یک توصیف درشت از یک فعالیت پیچیده و ساختار ضعیف است که در طول اکتساب دانش اتفاق می افتد



(شکل ٤-٨) مراحل اکتساب دانش

آنها از یک موقعیت منحصر به فرد به یک موقعیت دیگر فرق می کنند.فعالیت ها به صورت گسترده می تواند به صورت زیر رده بندی شوند:

مرحله شناسایی:

شناسایی شرکت کنندگان و نقش آنها : قبل از اینکه اکتساب و استفاده از دانش آغاز گردد, ابتدا باید شرکت کنندگان و نقش هر یک از آنها انتخاب و شرح داده شود .متداول ترین سناریو شامل عمل متقابل بین یک کارشناس در یک زمینه خاص و یک مهندس دانش است.کارشناس حوزه یا دامنه به عنوان یک فرد مطلع عمل می کند که

به مهندس دانش در مورد دانش و تخصص اش توضیح می دهد. فرایند اکتساب واستفاده از دانش می تواند شامل شرکت کنندگان دیگری نیز باشد.ممکن است چندین کارشناس دامنه یا حوزه مختلف , چندین مهندس دانش و حتی چندین کارشناس انضباطی در آنجا حضور داشته باشند.

شناسایی مسئله:

زمانی که شرکت کنندگان انتخاب شدند, مهندس دانش و کارشناس حوزه می توانند به شناسایی مسئله مورد نظر بپردازند.این کار شامل یک مبادله غیر رسمی دیدگاهها از منظرهای متفاوت یک مسئله است.به عنوان مثال توصیفات آن , خصوصیات و زیرمسئله هاست .به منظور شناسایی یک مسئله ، پاسخگویی به سؤالات زیر مهم است:

- از سیستم های خبره انتظار حل مسائل در چه کلاسی می رود؟
- این مسائل چگونه تشریح و توصیف خواهند شد؟
- زیر مسئله های مهم و جزءبندی های وظائف کدام ها هستند؟
- داده های در دسترس کدامند؟
- مفاهیم مهم و روابط مشترک درونی کدام ها هستند؟
- چه نوع راه حل هایی مورد نیاز هستند؟
- متخصصین انسانی در چه حوزه هایی مورد نیاز هستند؟

شناسایی منابع:

منابع به منظور اکتساب دانش, پیاده سازی سیستم و همچنین آزمایش آن مورد نیاز هستند.

منابع نمونه ای شامل منابع دانش, زمان , امکانات محاسبه و پول هستند.

شناسایی اهداف:

احتمالا" کارشناسان دامنه , شناسایی اهداف را در ساخت یک سیستم خبره در حین شناسایی مسئله انجام می دهند.به هر حال مفید خواهد بود اگر اهداف را از وظایف خاص جدا کنیم.به این دلیل که آنها محدودیتها و قیود اضافی تشکیل می دهند که می توانند برای توصیف شرایط مطلوب وامکان و شدنی بودن یک رهیافت خاص مفید

باشند.

مرحله ادراک:

در این مرحله مفاهیم کلیدی و روابط شناسایی شده در مراحل قبلی , به صورت واضح تری توصیف می شوند.سئوالاتی که در این مرحله باید پاسخ گفته شود عبارتند از:

- چه نوع داده هایی در این مرحله در دسترس هستند؟
- چه چیزی داده شده و چه چیزی باید استنتاج شود؟
- آیا زیر وظایف دارای نام خاصی هستند؟
- آیا استراتژی ها دارای نام خاصی هستند؟
- آیا فرضیه های جزئی قابل شناسایی وجود دارند که به طور متداول استفاده می شوند؟اگر وجود دارند کدام ها هستند؟
- آیا می توانید مفاهیم و رابطه ها را به صورت دیاگرام نمایش دهید؟
- محدودیت های موجود در این پروسه ها کدامند؟
- الگوی جریان اطلاعات چیست؟

این مرحله همانند بقیه مراحل شامل تراکنش های تکراری بین مهندسی دانش و کارشناس دامنه است.

مرحله رسمی سازی:

فرایند رسمی سازی شامل نگاشت مفاهیم کلیدی , زیرمسائل , خصوصیات جریان اطلاعاتی در حین مرحله ادراک به حالت نمایش و بازنمایی رسمی تری بر اساس ابزارهای متنوع مهندسی دانش یا چارچوب ها می باشد.مهندس دانش اکنون نقش غالب را در طراحی سیستم خبره بر عهده دارد.در این مرحله, مهندس دانش باید یک پوسته مناسب را شناسایی کند که برای مسئله ی در دست مناسبترین باشد .قالب نمایش دانش , انواع داده های آماده , استراتژی استنتاج و غیره باید کاملا" با خصوصیات مسئله منطبق باشد.سه عامل مهم در فرایند رسمی سازی عبارتند از : ۱)

فضای فرضیه ۲) مدل اصولی فرایند ۳) خصوصیات داده ها . برای فهم ساختار فضای فرضیه , مفاهیم باید رسمی سازی شده و در مورد چگونگی اتصال آنها به فرم فرضیه یا شناسایی زنجیره قوانین تصمیم گیری انجام شود .ساختار تک تک مفاهیم باید قطعی باشند.

سئوالاتی که باید به آنها در این مرحله پاسخ داده شود عبارتند از:

- آیا مفاهیم , اشیا ساخت یافته هستند یا موجودیت های ابتدایی می باشند؟
- آیا روابط سببی یا روابط وابسته به فضا و زمان میان مفاهیم مهم است؟
- آیا مفاهیم و فضای فرضیه، متناهی هستند یا خیر؟
- آیا تردیدها یا عناصر قابل داوری مرتبط با فرضیه های نهایی یا میانی وجود دارند؟
- آیا سلسله مراتب فرضیه نمایش داده می شوند یا خیر؟(آیا سطوح چندگانه از تجرد مورد نیاز است؟)
- آیا نوع پردازش کاملا" وابسته به داوری و قضاوت است یا وابسته به ریاضی و قضاوت است؟
- مدل داده ها به پاسخ سئوالات زیر بستگی دارد

- آیا داده های در دسترس کم و ناکافی یا فراوان هستند؟
- آیا تردیدهای وابسته وجود دارند؟
- آیا تفسیر منطقی داده ها به مرتبه رخداد آنها در طول زمان بستگی دارد؟
- آیا داده ها به اندازه کافی سازگار و کامل برای حل مسائل هستند؟

مرحله پیاده سازی و اجرا:

مرحله پیاده سازی و اجرا شامل نگاشت دانش رسمی سازی شده از مرحله قبلی به چارچوب بازنمایی وابسته به ابزار منتخب برای مسئله است. به دلیل اینکه دانش در این چارچوب سازگار , قابل قیاس و سازمان یافته برای تشریح اطلاعات خاص و کنترل جریان است, بنابراین این دانش, یک برنامه قابل اجرا می باشد.حوزه ی دانش , به همراه توسعه ساختارهای داده , قوانین استنتاج و استراتژی کنترل بسیار آشکار و روشن

می شود.توسعه نمونه اولیه سیستم در ساخت یک سیستم خبره بی نهایت مهم می باشد.

مرحله تست:

مرحله تست شامل ارزیابی نمونه اولیه سیستم و فرم های بازنمایی برای پیاده سازی آن است. هنگامی که سیستم نمونه اولیه دو یا چند بار از ابتدا تا انتها اجرا شد, سپس باید با نمونه های مسائل دنیای واقعی آزمایش شود تا نقاط ضعف موجود در اصول دانش و استراتژی استنتاج آن مشخص شود.

اجرا و کارایی پایین می تواند به دلیل:

- مشخصه های ورودی/خروجی که به اکتساب و استفاده داده و نمایش نتیجه و استنتاج رجوع می کند.
- قوانین استنتاج , واضح ترین مکان برای جستجوی خطاها هستند.قوانین ممکن است نا صحیح , ناسازگار و ناکامل و یا ناپیدا باشند
- استراتژی کنترل به عنوان مثال , توالی قوانین
- انتخاب نمونه ها می توانند بسیار مشابه باشند و بنابراین امکان دارد نتوانند همه جنبه های یک سیستم خبره را تست کنند.

۸-٦ استنتاج

نوع رویه استنتاج مورد استفاده در طراحی یک سیستم خبره بستگی به الگوی بازنمایی دانش مورد استفاده دارد.زبا ن های رسمی زیادی برای بازنمایی دانش با موجودیت های خوش تعریف و روابط بین موجودیت های تسخیر شده با استفاده از فرمول های خوش فرم وجود دارند.اینها دارای علائم معناشناسی واضحی هستند واین زبان ها به صورت گسترده ای در مواردی که دانش مبهم یا بدون ثبات است استفاده می شوند. نمونه ها , منطق رسمی به نام منطق گزاره ای , منطق مسندی و غیره هستند. متد های مورد استفاده برای استنتاج منطقی هستند.منظور از منطقی بودن این است که : اجازه دهید A مجموعه ای از اصول عمومی (قضایا) یا حقایق وR مجموعه ای از قوانین استنتاج باشد .اگر یک تئوری در منطق وجود داشته باشد که درست باشد, آنگاه ما می توانیم اثباتی را بیابیم که چیزی جز یک سری از قوانین نیست که زمانی که در یک

ترتیب مناسب بر مجموعه A بکار برده شود, به یک تئوری منتج خواهد شد.

۸-٦-۱ رویه استنتاج در حساب گزاره ای:

اگر حساب گزاره ای برای بازنمایی حوزه ی دانش استفاده شود, آنگاه مسئله ی استنتاج به صورت زیرمطرح میشود:R یک مجموعه از قوانین در فرم شرطی است و G مجموعه ای از اهداف و فرضیه های قابل اثبات است:

R= { r_i/r_i is a rule in clausal form for 1<= i<= N}
. r_i= VL_i^j , 1<= j<= P , where literal L_i^j is a posited or negated atomic variable in propositional calculus.

سپس مسئله استنتاج برای یک تئوری داده شده $g \in G$,نوشته می شود به صورت زیر:

قانع کننده $R \wedge O$

O مجموعه ای از مشاهدات انجام شده است.

O ={ O ; where O is literal whose value is true}

با بسط دادن فرمول خواهیم داشت:

$$(r_1 \wedge r_2 \wedge ..... \wedge r_n) \wedge O_1 \wedge O_2 \wedge ..... \wedge O_k \wedge g_i$$

هدف ما یافتن مقادیر و ارزش ها برای تمام متغیر های موجود درR جایی که 1< I<K است, میباشد به طوری که تساوی بالا درست باشد.رویه استنتاج خیلی آسانتر خواهد شد در صورتی که ما مسئله را توسط ابطال و تکذیب ثابت کنیم.ما در اینجا مسئله ای را به صورت زیر مطرح می کنیم:

$$F = (R \wedge O \wedge \rightarrow 9)$$

.که برای همه بردارهای تخصیص ممکن , نادرست است .اگر F قانع کننده نیست آنگاه g یک معنی و مفهوم است در غیر این صورت g یک مفهوم نیست .این فرایند برای همه $g \in G$ می تواند تکرار شود.الگوریتم ها برای استنتاج در منطق گزاره ای عبارتند از:

۱) رزولوشن واحد با کشف کنندگی های شکاف دهنده

۲) رزولوشن با دلالت کننده ها

۸-٦-۲ رویه استنتاج در حساب مسندی:

در اینجا نیز استنتاج مبتنی بر ابطال و تکذیب است.منطق معتبر است ولی غیر قابل تصمیم گیری است

اگر یک مجموعه از قوانین wff داشته باشیم می توانیم مجموعه ای از قوانین استنتاج را به ترتیبی به کار ببندیم به طوری که بتوان قضایای درست را از آن مشتق کرد.ولی با داشتن یک قضیه درست و تعریف شده , لازم نیست که ما یک توالی از قوانین را که می توانند قضیه را از فرض های قبلی و اولیه اثبات کنند , بدانیم.در این موارد تضمینی وجود ندارد که الگوریتم استنتاج متوقف شود.

۸-٦-۳ رویه استنتاج در سیستمهای تولید مبتنی بر قانون:

در ادبیات سیستم های خبره اصطلاح قانون, معنای بسیار محدود و باریکتری نسبت به دیگر زبان ها دارد.این اصطلاح به معروفترین نوع تکنیک بازنمایی دانش به نام بازنمایی مبتنی بر قانون اشاره دارد.قوانین یک روش رسمی برای بازنمایی نظریه ها, رهنمود ها یا استراتژی ها ارائه می دهند.آنها معمولا" برای زمانی که دانش دامنه حاصل نتایج علوم تجربی توسعه یافته در طی سالها تجربه است, مناسب می باشند. قوانین معمولا" به صورت حالات if-then بیان می شوند.به طور مثال:

IF oil fire THEN use foam fire extinguishers

اگر نفت آتش گرفت **آنگاه** از آتش خاموش کن کفی استفاده کن

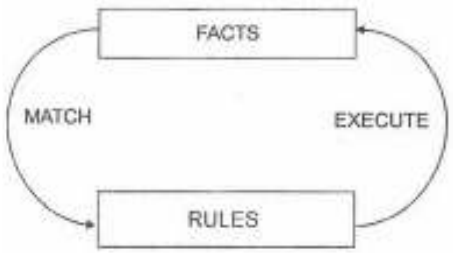
IF wood fire THEN use water

اگر چوب آتش گرفت **آنگاه** از آب استفاده کن.

شکل ۸-۵ .در سیستم های خبره مبتنی بر قانون ,دانش دامنه توسط مجموعه ای از قوانین که در مقابل مجموعه ای از حقایق موجود در وضعیت جاری تطبیق می کنند, بازنمایی می شود.

هنگامی که قسمت IF یک قانون به وسیله این حقایق برقرار باشد, عمل موردنظر در قسمتTHEN اجرا می شود.زمانی که این اتفاق می افتد, اصطلاحا" می گوییم قانون باید شلیک شود. سپس مجموعه ای از حقایق به وسیله تکمیل آنها با حقایق موجود در

قسمت THEN به روز در می آید.



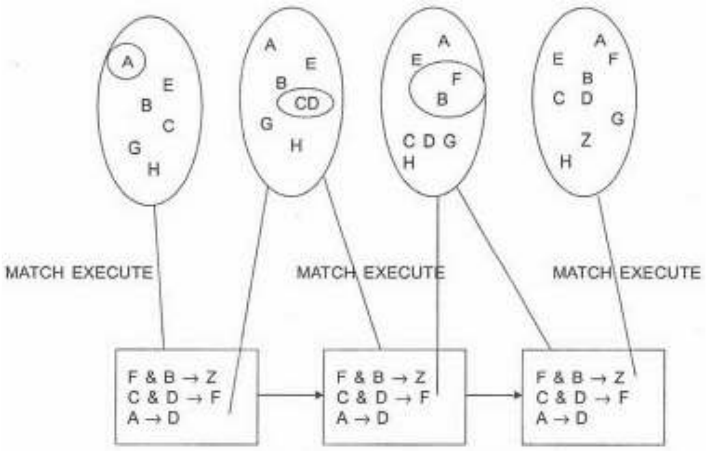
(شکل ۵-۸) الگوی استنتاج در یک سیستم خبره مبتنی بر قانون

این حقایق تازه اضافه شده به پایگاه دانش , خود می توانند برای تطابق با قوانین قسمت IF استفاده شوند.این عمل تطابق و اجرا , زنجیره های استنتاج را شکل می دهد.

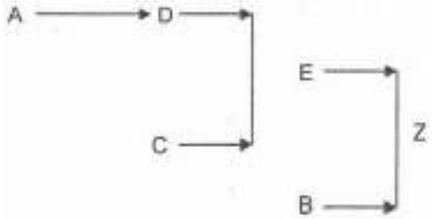
دو روش مهم وجود دارند که قوانین در آنها می توانند برای تطابق و اجرا از پایگاه دانش انتخاب شوند. رهیافت اول: **زنجیره سازی رو به جلو** و دوم: **زنجیره سازی رو به عقب** نامیده می شوند.

۸-٦-٤ زنجیره سازی رو به جلو:

شکل ۸-٦ مکانیسم زنجیره سازی رو به جلو را نمایش می دهد.(شکل ٦-۸) مکانیزم زنجیره سازی رو به جلو



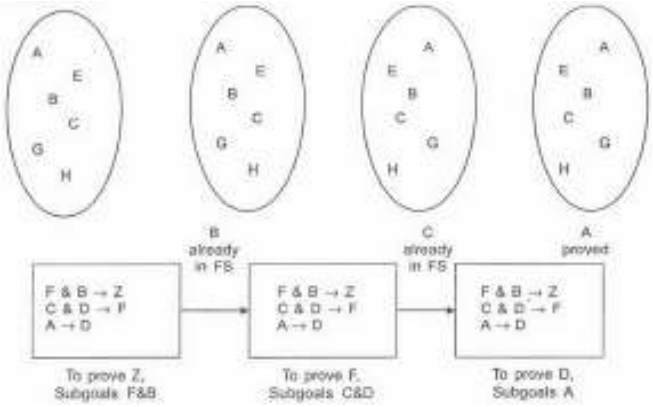
اولین قانونی که شلیک می شود $A \rightarrow D$ است زیرا که A هم اکنون در مجموعه حقایق قرار دارد.به عنوان یک نتیجه منطقی , وجود D استنتاج شده است و D در مجموعه حقایق قرار داده می شود.این عمل باعث می شود تا قانون دوم شلیک شود و تا به آخر تا اینکه Z نیز برقرار شود.این رویه زنجیره سازی رو به جلو نامیده می شود.زنجیره استنتاج تولید شده در شکل ۷–۸ نشان داده شده است.



(شکل ۷–۸) زنجیره استنتاج در زنجیره سازی رو به جلو

۸–٦–۵ زنجیره سازی رو به عقب:

زنجیره سازی رو به عقب استدلال در جهت عکس منطق را دنبال می کند ما در اینجا فرض می کنیم هدفی داریم که درست است, سپس سعی می کنیم تا آن را توسط گرفتن مقادیر مقدم یا پیشین , ثابت کنیم .این کار به وسیله تبدیل هر داده یا اطلاع به یک زیر هدف جدید انجام می شود. این فرایند تا زمانی که همه حقایق در مجموعه حقایق قرار گیرند, ادامه می یابد. شکل ۸–۸ مکانیسم زنجیره سازی رو به عقب را نشان می دهد.(شکل ۸–۸) مکانیزم زنجیره سازی رو به عقب.



۸-٦-٦ متد استنتاج در دیگر الگوهای بازنمایی:

برای طراحی سیستم های خبره , الگوهای دیگر بازنمایی دانش نیز وجود دارند.ما در اینجا قصد بررسی جزئیات رویه های استنتاج توسط آن الگوها را نداریم.

۸-۷ روش شناسی یا متدولوژی برنامه نویسی

یک دیدگاه مفید تر و مجرد تر از سیستم های خبره این است که متدولوژی برنامه نویسی بر جداسازی چیزی که در دنیا درست است از چگونگی بکارگیری دانش برای حل مسئله تاکید دارد.در این دیدگاه یک سیستم خبره شامل دو مفهوم زیر است:

- دانش دامنه یا حوزه

متدهای حل مسئله

در موارد ساده تر , دانش دامنه عبارتست از مجموعه ای از حقایق و متدهای حل مسئله که بیشتر شامل مکانیسم های استدلال همه منظوره است . به عنوان یک مثال واقعی در مورد اینکه چگونه می توان دانش دامنه را از متدهای حل مسئله تمییز داد به نمونه ی ساده شده از یک تشخیص پزشکی دقت کنید:

ما با این تفکر شروع می کنیم که:" اگر گیتا تب دارد , پس گیتا عفونت دارد." که بیشتر یک متد حل مسئله خاص ویژه برای تشخیص علت تب گیتاست.با استفاده از روش برنامه نویسی سنتی با مفهوم متغیرها , می توانیم مثال را به این صورت تعمیم دهیم:

متد حقایق
اگر یک بیمار تب داشت, پس بیمار عفونت دارد. گیتا بیمار است.

که ستون سمت چپ حقایق و ستون سمت راست متد عمومی تر را بیان می کند.

یک قانون عمومی تر که شایسته نامیدن به نام متد حل مسئله است , توسط مجرد سازی بیشتراین مورد بدست می آید.

متد حقایق اگر
بیمارعلائم یک بیماری را داشته باشد گیتا مریض است.
پس بیمار آن بیماری را دارد تب یک علامت بیماری است
عفونت یک بیماری است
تب علامت وجود عفونت است.

یک تعمیم نهایی ما را به نتیجه ای این چنینی می رساند:

متد حقایق
اگر شئ یک صفت از کلاس را نمایش می دهد گیتا مریض است
سپس آن شئ متعلق یه آن کلاس است تب یک علامت بیماری است
عفونت یک
بیماری است
تب علامت وجود عفونت است
بیمار یک شئ است
یک علامت بیماری یک صفت است
یک بیماری یک کلاس است.

.

در این دیدگاه , سیستم های خبره، متدولوژی برنامه نویسی ای را فراهم می کنند که وضعیت های حقیقی و واقعی را از متدهایی که چگونگی بکارگیری این حقایق را مشخص می سازند, جدا می کند.در این متدولوژی , انعطاف پذیری برای داشتن هر دو متد ها و حقایق ویژه و بسیار عمومی وجود دارد. در واقع اگر یک دامنه خودش را به استدلال توسط کد گذاری یک روش خاص برای مقابله با مسائل ویژه , متوجه می سازد , سپس ممکن است که ما برای هر یک از آنها چندین متد تک منظوره (منظور خاص) انتظار داشته باشیم. جایی که هر مولفه احتمالا" به متدهای تست ویژه نیاز دارد. به هر حال , در صورت امکان , متدلوژی به توسعه متدهای حل مسئله عمومی تر, به همان صورتی که در مثال بالا آمده است , پیش میرود

۸-۸ سیستم های خبره – ابزار

سیستم های هوش مصنوعی , به صورت متمرکز به بازنمایی و دستکاری دانش اهمیت می دهند. بعضی از دانش ها , به خصوص ریاضیات و علوم به صورت اعداد و فرمولهایی بیان می شوند که شامل مجموعه ای از ارقام و عملیات حساب هستند.بیشتر تلاش انسان , به هر حال نیازمند بیان در یک زبان عمومی تر و قدرتمندتر است.حتی ریاضیات , زمانی که شامل پردازش های استدلال مجردتر (اثبات قضایا- دگرگونی جبری- و راه حل های سیمبولیک برای تساوی های دیفرانسیل – انتگرالی) است که نیازمند زبان های قدرتمندتر و عمومی تر هستند. این زبان ها توسط مفاهیم و روابط بازنمایی شده به وسیله سیمبولها و رشته هایی از سیمبولها , بیان می شوند.

در واقع , اعداد و فرمولها یک مجموعه ازسمبولهای خاص هستند. اعداد سیمبولهایی هستند که خواص آنها بر روی مجموعه ای از عملیات حسابی تعریف شده اند. این عملیات حسابی توسط سیمبولها ورشته هایی از سمبولها مانند (- و+ و* /) بازنمایی شده است.به هر حال , ابزار هوش مصنوعی عبارتست از زبان ها , پردازش ها و ساختارهایی که اجازه فراگیری یا اکتساب , بازنمایی , ذخیره و انتقال و دیگر تغییرات مفاهیم و رابطه ها توسط ماشین های پردازش اطلاعات را می دهند. در این رابطه , زمینه هوش مصنوعی به صورت خیلی نزدیک بستگی به مطالعه تئوری زبان دارد که

شامل زبان های کامپیوتری سطح بالا و تئوری کامپایلر کامپیوتر است.

.

تعدادی از فعالیتها وجود دارند که بر توسعه یک سیستم خبره مقدم هستند.اینها شامل شناسایی دامنه مسئله , یافتن تخصص و انتخاب ابزار می باشند.در اصل ٤ نوع ابزارهای توسعه در دسترس هستند که در زیر لیست شده اند:

- زبان های الگوریتمیک (مانند C,Pascal, Bassic)
- زبان های سیمبولیک (مانند لیسپ و پرولوگ)
- محیط های توسعه (مانند Art,KEE,LOOPS)
- بدنه ساختمان سیستم های خبره (مانند

(Crystal,XpertRule,Leonardo,Xi-Plus

۸-۸-۱ زبان های الگوریتمیک:

در کل زبانها می توانند تعریف شوند به عنوان:

- انعطاف پذیر و قدرتمند
- آنها می توانند به عنوان یک خیاط برای یک سیستم دقیقا" مطابق با کاربرد در نظر گرفته شوند.
- فاقد چارچوب مهندسی دانش

زبان ها می توانند به دو دسته قراردادی یا AI رده بندی شوند.البته با کمی اشتراکات میان آن دو.

زبان های قرار دادی می توانند به صورت رویه ای در طبیعت توصیف شوند که برای کار بر اساس الگوریتم , جایی که یک وظیفه به تشخیص قدم به قدم شکسته شده و سپس کدبندی می شود, طراحی شده است.

معمولا" ساختارهای داده ای پیچیده می توانند از انواع داده ای اولیه تشکیل شوند, و انها دارای ساختارهای کنترل قدرتمندی هستند .در زمینه توسعه سیستم های خبره , آنها معمولا" به عنوان زبان های پیاده سازی و اجرا (یا تحویل) برای سیستم های تولید, عمل می کنند. یک سیستم خبره , با استفاده از زبان هوش مصنوعی , ساختمان بدنه یا ابزار توسعه داده می شود و به یک زبان قرار دادی پرسرعت تر (معمولا" C)

.

زمانی که آن به صورت راضی کننده ای عمل می کند, ترجمه می شود.

طراحان نیاز دارند تا از کارهای داخلی موتور استنتاج مطلع باشند هرچند که زبانهای شی گرا (مانند C++) به وسیله بکارگیری زبان های برنامه نویسی قرار دادی , توسعه ساختارهای استنتاج را آسانتر کرده اند. فایده این نوع سیستم ها این است که آنها معمولا" برای مسائل الگوریتمیک محاسبه عددی بیشتر کاربرد دارند تا پردازش های سمبولیک و اینکه ساختارهای واضح آماده برای اجرای سیستم های خبره وجود ندارد.

۸-۸-۲ زبان های سمبولیک:

دانش بشری مفهومی پویاست و هر تلاشی برای بازنمایی آن باید شامل ساختارهای قابل بسط دانش باشد . زبان های کامپیوتری برای برنامه نویسی منطق باید ساختارهایی برای ذخیره و بازیابی حقایق شناخته شده یا استنباط شده از پایگاه دانش یا پایگاه حقایق داشته باشند , همچنین برای استنباط حقایق جدید، باید رویه ها و توابعی داشته باشند.

به عنوان یک قاعده کلی , ما این کارها را توسط زبانهایی مانند فرترن یا پاسکال می توانیم انجام دهیم . با این وجود عملیاتی نظیر پردازش لیست که برای اجرای استنتاج منطقی مفید هستند, برای پیاده سازی در زبان های رویه ای بسیار کم بازده و دشوار هستند . بنابراین نیاز برای زبان های هوش مصنوعی تخصصی که تمایل به ساختارهای لیستی دارند, می توانند گسترش یافته یا به صورت دلخواه ترکیب شده یا جدا شده باشند . استفاده از یک زبان خام هوش مصنوعی به پیاده سازی آن اجازه منعطف بودن می دهد اما تلاش بیشتری برای ایجاد امکاناتی نظیر واسط کاربر (که بدون آن امکان دارد زبان مناسب به نظر نرسد) می طلبد.

لیسپ:

لیسپ یکی از زبان های کامپیوتری رایج برای برنامه نویسی هوش مصنوعی است.این زبان برای حمایت از دستکاری های سمبولیک و تقابلی و روش برنامه نویسی آزمون و خطای مورد استفاده در بیشتر تحقیقات هوش مصنوعی است .البته لیسپ تنها زبان

موجودی نیست که می تواند برای کاربرد های هوش مصنوعی در کامپیوتر استفاده شود. به عنوان یک قاعده کلی , کاربردهای این چنینی می توانند در ماشین توسط زبان اسمبلی برنامه نویسی شوند.لیسپ برای استفاده راضی کننده تر است به خصوص با معرفی کامپیوترها و کامپایلر های اصلاح شده لیسپ در قیاس با زبان های کامپیوتری دیگر , این زبان از لحاظ کارامدی بسیار بهتر است.

لیسپ توسط جان مک کارتی در اواخر دهه ۵۰ به عنوان یک زبان برای کاربردهای هوش مصنوعی توسعه یافت .لیسپ مترادف کلمه پردازنده لیست (LISt Processor) است.زبان های رویه ای مانند پاسکال یا C دارای عملگرهای ابتدایی تری برای اجرای محاسبات جبری توسط فرمول های حاوی سیمبولهای عددی صحیح و اعشاری هستند.

علاوه بر این موارد , زبان لیسپ دارای مجموعه ای از عملگرهای اولیه است که آن را قادر به انجام انواع مختلفی از استنباط ها با جملات (لیست ها) که شامل کلمات (رشته ای دلخواه از کاراکترها) هستند و نمایش گزاره ها و آرگومان هایشان , می کند..

پرولوگ:

پرولوگ به عنوان یک زبان جایگزین برای برنامه نویسی هوش مصنوعی توسط آلن کلمرار و همکارانش در مارسل در اوایل دهد ۷۰ توسعه یافت. همانند لیسپ , پرولوگ هم استانداردهای مختلفی پیدا کرد ولی استاندارد قطعی و نهایی آن امروزه به صورت مستدل ،استاندارد پرولوگ ادینبورگ است. این استاندارد در کشورهای اروپایی , ژاپن و استرالیا بیشتر و در ایالات متحده کمتر استفاده می شود. پرولوگ نسبت به لیسپ زبان سطح بالاتری است چرا که آن دارای تعدادی از انواع استنباط و جستجو است که قبلا موجود بوده است.

به پرولوگ می توان به عنوان یک اثبات کننده قضیه پیاده سازی شده در فرم مفسر زبان نظر افکند که ما به واسطه دادن قواعد کلی به آن می توانیم برنامه نویسی کنیم.(چارنیاک و مک درموت ۱۹۸۵) . با این دیدگاه , پرولوگ به عنوان یک وسیله قطعی برای برنامه نویسی اخباری است. همه آن چیزی که ما باید انجام دهیم این است که پرولوگ را با مجموعه ای از حالات و قواعد کلی که بعضی سیستم ها را توصیف

می کنند , تهیه کنیم. سیستم هایی که ما می خواهیم استدلال کنند و حقایق افزوده شده دلخواه (راه حل برای مسئله با استفاده از قدرت built-in استنباط) را استنباط می کند.

آخرین بخش نشان می دهد که لیسپ و پرولوگ چگونه یک محیط برنامه نویسی منطق سطح بالاتر را نسبت به زبان های قراردادی توسط پیاده سازی قابلیت های جداسازی شده برای هر دو استنباط و جستجو , تهیه می کنند. یک کاربرد کامل به چیزی بیشتر از این نیاز دارد . کاربردهایی که برای استفاده توسط انسان طراحی شده اند به واسطه کاربر نیاز دارند و کاربردهایی که برای کار با درخواستهای خارجی و پایگاه داده های آنها طراحی شده اند به واسط های نرم افزاری نیاز دارند.

در این بخش تعدادی از بدنه های ساختمان سیستم های خبره یا محیط آنها را معرفی خواهیم کرد که توسط مهندسین به عنوان ابزارهای قدرت برای ساخت سیستم های مبتنی بر دانش استفاده میشود. در بیشتر موارد , ما مجبوریم هزینه ای را برای بکارگیری یک ابزار سطح بالاتر بپردازیم چون که مقداری اتلاف انعطاف پذیری وجود دارد.

کاربر یک ساختار خاص , نوعا" درجه ای از کنترل بر الگوی استنتاج و طرح واسط کاربر سیستم را واگذار می کند.بنابراین , کاربر باید مسئله را با چیزی که محدود شده و معماری نامناسب بالقوه برای بازنمایی و استدلال را دارد, منطبق کند.

بنابراین خبر خوب این است که برای تعداد زیادی از کلاس های کاربردهای مهندسی (تشخیص , ارزیابی , مانیتورینگ و پیکر بندی) ساختمان سیستم های خبره و ابزارهای سطح بالایی وجود دارند, که نمونه های بازنمایی و استدلال آنها با نیازهای کاربرد خیلی خوب منطبق است.

اگر ساختار یا محیطی مناسب بتواند یافت شود که مطابق با ابزار یک کاربرد معین باشد , آنگاه آن می تواند به طرز معناداری هم سودمندی و هم کیفیت پردازش مهندسی دانش را افزایش دهد.

۸-۸-۳ محیط های توسعه:

محیط های توسعه یا جعبه های ابزار , معمولا" مبتنی بر سخت افزار بهینه شده برای زبان دستکاری سیمبول مانند لیسپ یا پرولوگ هستند. این زبان های سمبولیک همراه با ویراستارهای حساس به متن و گرافیک هستند که معمولا" شامل استنتاج built-in هستند.

محیط های برنامه نویسی سیستم های خبره , بسته خاصی از کدهای از پیش نوشته شده هستند.آنها مجموعه ای از بلاک های ساختمان را آماده می کنند که برای تمام نیازهای برنامه نویسان فراهم شده اند و در بعضی مواقع به عنوان " جعبه ابزار" شناخته می شوند. در مقایسه با زبان های برنامه نویسی قرار دادی یا معمولی همانند پاسکال , محیط ها شامل کتابخانه ای از رویه ها هستند که می توانند فراخوانی شده و توسط برنامه نویس برای توسعه یک کاربرد خاص به هم بپیوندند.

محیط ها اغلب توسط لیسپ به عنوان یک زبان بیسیک نوشته می شوند.

محیط های توسعه نمونه ای عبارتند از : KnowledgeCraft, G2,Art, KEE (جکسون ۱۹۸٦)

که متدهای متنوعی را برای بازنمایی و کنترل پردازش استدلال پیشنهاد می کنند.آنها بعضی ماجول های کار جزئی را در تعدادی از کتابخانه ها تهیه می کنند که این کتابخانه ها می توانند توسط برنامه نویس برای توسعه کاربردها متصل شوند. برنامه نویسان همچنین می توانند ابزارهای خود را به محیط اضافه کنند.

محیط ها به برنامه نویسان کامپیوتری خبره ای نیاز دارند تا بهترین کارایی از انها بدست آید. آنها فقط برای کارشناسان یا کاربران معمولی نیستند. تجربه نشان داده است که حدود ٦ ماه تلاش مداوم نیاز است تا محیط ها بتوانند پربار باشند.

فایده محیط ها براساس میزان انعطاف پذیری استنتاج , مکانیسم های استنتاج که می توانند توسعه یابند و همچنین قدرت نتیجه بخشی سیستم های توسعه یافته است.

۸-۸-٤ بدنه ساختار سیستم های خبره (shells):

شل یا پوسته یا بدنه ساختار ابزارهایی هستند برای ساخت سیستم های خبره ای که امکانات بازنمایی دانش و مکانیزم های استنتاج را فراهم می نمایند . یک برنامه نویس باید جزئیات دانش در مورد یک حوزه خاص را از یک کارشناس و منبع اطلاعاتی کسب کند. بنابراین یک شل می تواند به عنوان یک سیستم خبره با همه دانش حوزه یا دامنه و دارای امکاناتی برای وارد کردن یک پایگاه دانش جدید باشد. به طور کلی بعضی از فرم های امکانات اشکالزدایی برای کنترل استنتاج یک مسئله داده شده، فراهم می شود.

- متد های استنتاج به صورت معنی داری از یک دامنه به دامنه دیگر تغییر می کنند و شل های سیستم های خبره ای توسعه یافته اند تا به طراح اجازه منعطف بودن بیشتر را در طول ساخت سیستم خبره بدهند.بعضی از شل های

سیستم هایی خبره که هم اکنون رایج هستند در زیر لیست شده اند:

عموما" شل ها فقط برای مسائل همان نوع قابل استفاده هستند و با قابلیت های خود محدود می شوند. با این حال آنها شاید راحتترین و بهترین روش برای ساخت نمونه اولیه سیستم های خبره باشند و برای بکارگیری نیازبه مهارتهای برنامه نویسی کمتری دارند.

امکانات نمونه ای پیاده سازی تهیه شده توسط شل ها عبارتند از :

- یک زبان بازنمایی دانش
- یک ویراستار پایگاه دانش
- امکانات ردیابی و اشکالزدایی
- تعدادی امکانات واسط کاربری
- به زبان های قراردادی یا معمولی / خارجی می پیوندند.
- امکاناتی برای استدلال های غیر حتمی
- امکانات قیاس کل از جزء (شاید)

به علاوه بعضی یا تمام خصوصیات زیر ممکن است برای اصلاح قابلیت استفاده یک سیستم شل موجود باشد:

- در دسترس بودن فرمان های کاربر (چرا , چگونه, لیست کن و غیره)
- شل, مکالمه مورد نیاز در یک مشاوره از پایگاه دانش را تولید خواهد کرد.
- شل , صفحه نمایش مشاوره را به طور خودکار قالب بندی خواهد نمود.
- درجه ای از شبیه سازی زبان طبیعی

۸-۹ کاربردها

الیزا:

در سال ۱۹٦٦ دانشمند رشته کامپیوتر , ژوزف وایزن بام , آخرین اقدام را برای برنامه

ای که الیزا نامیدش انجام داد. الیزا به کاربر اجازه تایپ یک جمله توسط صفحه کلید با هیچ محدودیت گرامری یا محتوایی را می داد و سپس کامپیوتر با یک جمله از خود به آن پاسخ می گفت. الیزا از دو ماجول تشکیل شده بود. یکی از ماجول ها حاوی روتین اصلی برنامه و دیگری حاوی چیزی بود که وایزن بام آن را نمایشنامه یا اسکریپت می نامید. یک اسکریپیت مجموعه ای از قوانین بود که به الیزا اجازه می داد تا یک مکالمه در مورد یک موضوع خاص را داشته باشد .

اسکریپت ها قابل معاوضه یا تبادل پذیر بودند به طوری که اسکریپت های مختلفی می توانند به الیزا متصل شوند. برای اینکه آن را وادار به صحبت با کاربر در مورد موضوعات مختلف کنند .

اسکریپتی که وایزن بام آن را برای الیزا خلق کرد منتج به ساخت برنامه ای شد که نوعی تقلید از یک جلسه روانپزشکی به شیوه دکتر روانپزشک , کارل آر . راجرز بود. نسخه ای از الیزا که حاوی اسکریپت روانپزشکی بود به نام DOCTOR معروف شد. در آن زمان , DOCTORشاید به مشهورترین برنامه کامپیوتری در جهان تبدیل شد.

آ ن برنامه نتایج خوبی را تولید می کند, هرچند بطور متوسط برنامه ی پیچیده ای نیست ولی تقریبا" با هوش است. الیزا بر اساس تطابق قسمت سمت چپ قوانین در مقابل آخرین جمله کاربر عمل می کند , و از قسمت راست مناسب برای تولید پاسخ استفاده می کند. به عنوان مثال اگر کاربر تایپ کند

” My brother is mean to me“ الیزا ممکن است این گونه پاسخ دهد : ” who else in your family is mean to you “ یا “ tell me more about your family” . قوانین به وسیله کلمات کلیدی مرتب شده اند پس در واقع فقط جند جمله می توانند با یک جمله به خصوص تطبیق داشته باشد. این تطابق به صورت زیر است:

بعضی از قوانین سمت چپ (قسمت اگر) ندارند. بنابراین قانون می تواند در هر جایی به کار رود . این قوانین زمانی که هیچ قانون قابل تطبیق نباشد استفاده می شود و پاسخ هایی مانند : چیزهای بیشتر ی در مورد آن بگو ”tell me more about that “

تولید می کنند . توجه کنید که قوانین خودشان یک فرم تقریبی تطابق را باعث می شوند.الگوها در مورد کلمات خاصی در جمله کاربر سئوال می کنند . آنها نیازی ندارند که با کل جملات تطابق یابند. از این رو طیف وسیعی از جملات می توانند توسط یک قانون تطبیق یابند, پیچیدگی گرامری زبان انگلیسی به راحتی می تواند نادیده گرفته شود.

این موضوع دلیلی برای قدرت بزرگ الیزا (توانایی در گفتن چیزی تقریبا" معقول، تقریبا"برای تمای موارد) و هم ضعف بزرگ آن (سطحی بودن فهم آن و امکان گمراه شدن آن) است.

فهرست زیر یک مکالمه نمونه از مجموعه الیزا را نشان می دهد:

مکالمات الیزا:

شخص : مردها همه مثل هم هستند.

الیزا: در چه مورد؟

شخص : آنها همیشه در مورد یک موضوع یا چیزی به ما گیر می دهند.

الیزا: می توانی در مورد یک نمونه خاص فکر کنی؟

شخص: خوب , نامزد من مرا وادار به آمدن به اینجا کرد.

الیزا: نامزد تو , تو را وادار به آمدن به اینجا کرد

شخص: او می گوید من اغلب اوقات افسرده ام.

الیزا: متاسفم که می شنوم تو افسرده هستی.

شخص: درست است, من بدبخت هستم.

الیزا: آیا فکر می کنی آمدنت به اینجا باعث می شود که دیگر احساس بدبختی نکنی؟

شخص : اینکه من به کمک احتیاج دارم به نظر حتمی می آید.

الیزا: برای تو کمک گرفتن چه معنی می دهد؟

شخص : شاید بتوانم یاد بگیرم که با مادرم بهتر سر کنم.

الیزا: در مورد خانواده ات بیشتر بگو.

با وجود اینکه در نگاه اول مکالمه بالا شگفت انگیز به نظر می رسد, ولی بیشتر از آنکه از یک ادراک واقعی از جانب DOCTOR بدست آمده باشد توسط یک حیله گری معنایی به دست آمده است . این یک نمونه خوب از برنامه بود. متاسفانه , همانطور که بیان شد , DOCTOR هیچ ادراک واقعی از مکالمه ای که در آن شرکت دارد ,ندارد.

گاهی یکی از حقه های موجود در برنامه در یک موقعیت نامناسب به کار خواهد رفت زیرا که DOCTOR خود نمی فهمد که این حقه یا تدبیر مناسب آن وضعیت نیست . پاسخ یا نتیجه این برنامه می تواند به سادگی بسیار نامناسب و مضحک باشد.

PARRY پارانوید مصنوعی:

ضمنا" , کنت کلبی که علاقه مند به کار در زمینه مدلسازی رفتار های انسانی بود , شروع به نوشتن برنامه خود نمود که می توان گفت به نوعی این برنامه ابتکاری تر از DOCTOR بود . این برنامه پری(PARRY) بود. این برنامه یک مکمل برای DOCTOR بود به این صورت که DOCTOR نقش روانپزشک و PARRY نقش بیمار بر روی تخت روانپزشک را بازی می کرد.

به طور مختصر , PARRY واکنش های یک جوان که مشکل اسکیزوفرنی پارانوید را دارد , شبیه سازی می کند. PARRY همانند DOCTOR برنامه ای دارای اثر متقابل است. شخص به وسیله صفحه کلید کامپیوتر , سئوالات و فرمانهای خود را وارد می کند . پاسخ ها به وسیله برنامه نوشته می شود. شایستگی این شبیه سازی این گونه تصدیق شد که , کلبی نسخه ای از یکی از مکالمات PARRY را برای چند روانپزشک فرستاد و ازآنها خواست تا در مقایسه با یک مکالمه واقعی بیمار با روانپزشک , تصمیم بگیرند که کدام یک واقعی و کدام رونوشت است. فقط ۵۱٪ از روانپزشکان توانستند مکالمه واقعی را تشخیص دهند. هر چند که ۵۱ درصد رقمی بهتر از انتخاب های تصادفی نیست.

مکانیزم داخلی PARRY همانند DOCTOR کاملا" براساس تدابیر و حقه های معنایی پایه گذاری شده است. اینگونه به نظر می آید کهPARRY فقط سئوالات مصاحبه کننده را می فهمد ولی در واقع به هیچ وجه مکالمه را درک نمی کند . هنگامی که سئوالی نه چندان سطح بالا پرسیده شود , PARRY پاسخ نامربوط ارائه خواهد داد همانطور که DOCTORاین کار را می کند. با این حال , PARRY یک برتری نسبت به دکتر دارد و آن این است که ما حداقل جوابهای نامربوط را از یک بیمار روانی انتظار داریم نه یک پزشک. انتظارات ما از یک روانپزشک لایق احتمالا" کمی متفاوت است! لیست زیر مجموعه ای از یک مکالمه با PARRY را نشان می دهد.

- دیالوگ PARRY (ضعیف)

شخص: چه مدت در بیمارستان بوده ای؟
PARRY : مدت یک هفته
شخص: آیا تو افسرده هستی؟
PARRY : نه
شخص: آیا تو دارو مصرف می کردی؟
PARRY: نه
شخص: مشکل تو چیه؟
PARRY : مردم منو عصبی می کنند.
شخص: چگونه این کار ار انجام می دهند؟
PARRY : آنها به من خیره نگاه می کنند.
شخص: چرا؟
PARRY : شاید آنها از طرز نگاه کردن من خوششان نمی آید.
شخص: آیا تو به طرز غیر معمولی نگاه می کنی؟
PARRY : چرا تو علاقه مند به دانستن طرز نگاه کردن من هستی؟
شخص: چون تو گفتی مردم خیره به تو نگاه می کنند.
PARRY : خوب , گاهی.

- مکالمه PARRY (قوی)

شخص : تو چرا در بیمارستان هستی ؟
PARRY: خوب نباید می بودم.
شخص: چه کسی تو را اینجا آورد؟
PARRY : پلیس
شخص: چه مشکلی با پلیس داشتی؟
PARRY : پلیس ها کار خودشان را انجام نمی دهند.
شخص: به جای آ ن چه کاری انجام می دهند؟
PARRY : آنها آدمهای اشتباهی را دستگیر می کنند
شخص: ولی آنها گاهی آدمهای درستی را هم دستگیر می کنند.
PARRY : قانون باید آن کلاهبرداران ایتالیایی را دستگیر کنه.

شخص: آیا تو ایتالیایی هستی؟
PARRY: اجداد من انگلیسی هستند ولی من در این کشور به دنیا آمده ام.
شخص: آیا تو کلاهبرداران ایتالیایی رو می شناسی؟
PARRY: من سعی می کنم از دنیای تبهکاران فاصله بگیرم
شخص: آیا تو با مافیا در ارتباط هستی؟
PARRY: می دونی؟ آنها منو می شناسند.
شخص: آنها چه طوری تو رو می شناسند؟
PARRY : آنها جاسوسی منو می کنند.

DENDRAL دندرال:

دندرال توسط فایگن بام در سال ۱۹٦٤ در دانشگاه استنفورد توسعه یافت. هدف آن یافتن ساختمان مولکولی یک جسم مرکب است . هنگامی که یک شیمیدان با یک جسم مرکب ناشناخته مواجه می شود , اولین کاری که انجام می دهد این است که پی به اتمهای سازنده آن جسم و نسبت این اتمها می برد. برای انجام این کار , او تست های تحلیلی و آزمایشاتی انجام می دهد. یکی از ابزارهایی که در این کار استفاده می شود , دستگاه اسپکترومتر جرم است. دقت عملیات آن در اینجا مورد توجه ما نیست . با یک تعریف ساده , آن فرکانس نسبی اتمهای مختلف و اجزای مولکولی در یک ترکیب را می یابد . با استفاده از این اطلاعات , شیمیدان , اتمهای سازنده و چگونگی چیده شدن آنها را در یک مولکول حدس می زند.

این عمل ما را به یاد مسئله ای می اندازد که با داشتن سنین افراد یک خانواده باید پی به سن افراد دیگر خانواده بریم .ولی البته , این مورد بسیار پیچیده تر است . خانواده بسیار بزرگ است, در مورد اتمهای شناخته شده و روابط بین آنها کتاب ها می توان نوشت . ولی مهمترین واقعیت این است که هیچ الگوریتم علمی وجود ندارد که با آن بتوان از طیف جرمی به ساختار مولکولی آن پی برد.

در اصل , دندرال برای شمارش تمام پیکربندی های ممکن از مجموعه اتمهای رعایت کننده قوانین بنیان شیمی، طراحی شده بود. این شمارش بعدها می تواند به عنوان یک لیست از امکان ها برای شیمیدان باشد. با یک تعریف اکید می توان گفت , دندرال اکنون یک برنامه نیست بلکه یک خانواده از برنامه هایی است که الگوریتم اصلی را در

مرکز این خانواده دارند. بطور عمده برنامه های دیگری قدرت این برنامه را افزایش دادند. یکی از مهمترین بسط ها این بود که مجموعه ای از موارد ممکن را می گرفت و آن را به مواردی که محتمل هستند کاهش می داد. برای انجام این کار , آن می بایست قوانین مبتنی بر حقایق شیمیایی و کشف کنندگی ها را ذخیره نموده و از آنها استفاده کند, (قوانین مبتنی بر حقایق شیمیایی , مبتنی بر قوانین شیمیایی و مبتنی بر تجربه و قضاوت کارشناسان)

دندرال داستان یک موفقیت است . نتایج گرفته شذه از این کاربرد در بیش از ۵۰ مقاله ذکر شدند که نه تنها مفید بودن آن را تصدیق نمودند بلکه درباره اختیارات علمی آن نیز بحث کردند. این برنامه بصورت مرتب و روتین مورد استفاده قرار میگیرد. تعداد کاربران آن در سال ۱۹۸۳ به قدری با سرعت افزایش یافت که نهایتا" یک کارخانه مجزا برای توزیع و ساخت نسخه های دیگر آن بر پا شد

مایسین:

یکی از متداولترین فرم های بیماری که ما را رنج می دهد , عفونت های باکتریایی است. با سپاس از پیشرفت های پزشکی , ما امروزه تعداد زیادی از آنتی میکروب ها و داروی معروفتر آنتی بیوتیک ها را برای مبارزه با این عفونت ها را داریم . ولی امروزه پزشکان همراه با داشتن طیف وسیعی از داروهای آنتی بیوتیک با طیف وسیعی از انتخاب های آنها برای انواع بیماری ها نیز روبرو اند.اگر تنها یک عامل آنتی میکروبیکال موثر برای تمام انواع باکتری های عفونی بود , مسئله انتخاب وجود نداشت . افسوس که چنین اکسیری وجود ندارد. چیزی که هست این است که ممکن است یک دارو برای نوع خاصی مفید و برای نوعی دیگر بی ارزش باشد. پزشکان در انتخاب های خود باید محتاط باشند . از این رو مفید بودن تنها یکی از معیارهاست . پزشک همچنین باید موارد دیگر نظیر آلرژی های بیمار , داروهایی که مصرف می کنند و محدودیتهای مشابه را در نظر بگیرد و مایسین طراحی شد تا در این مسئله به پزشک یاری رساند.

اگر به صورت دقیقی تری به وظایف یک پزشک نظر افکنیم , می بینیم که ٤ تصمیم باید گرفته شود. ۱) آیا بیمار از عفونت رنج می برد؟ ۲) کدام یک از اندام های بدن درگیر هستند؟ ۳) کدام دارو ها ممکن است مناسب باشند؟ و ٤) کدام یک از آنها باید استفاده شود؟ مایسین برای کمک به هر چهار تصمیم گیری طراحی شد. نحوه کمک

کردن به صورت زیر است : بر اساس اطلاعاتی که بیمار می دهد و نتایج آزمایشات , مایسین برای هر٤ مورد پاسخ می دهد. این برنامه , نتایج و درجه قطعیت آنها را نمایش می دهد. این برنامه همچنین می تواند بنابه درخواست , خطی از استدلال را که منجر به این نتیجه گیری شده , قوانینی که در این مسیر استفاده شده اند , پیشنهادات رد شده و حتی کتابهای مرجع ودیگر انتشارات که می توانند این قوانین را تضمین کنند , را ارائه دهد . با داشتن چنین اطلاعاتی , پزشک می تواند بهترین قضاوت را انجام دهد.(کار بر این پروژه در سال ۱۹۷۲ در دانشگاه استنفورد آغاز شد . نام آن از پسوندی که معمولا" در نام داروهای آنتی بیوتیکی وجود دارد , گرفته شده است.) قوانینی که این برنامه از آن استفاده می کند از متخصصین در زمینه عفونت های باکتریال اخذ شده بود. در یک سری از آزمون ها که از موارد عفونت های خونی انتخاب شده بودند , نتیجه گیری های مایسین با نظرات متخصصین این زمینه و حتی غیر متخصصین مقایسه شد . مایسین به نحوعالی در این آزمون ها عمل کرد , حداقل به خوبی کارشناسان و یا غیر متخصصین . یک نمونه از قانون مایسین در زیر نشان داده شده است: قانون ۸٦:

اگر: ۱) عفونت مورد مداوا منتژیت است

و ۲) بیمار مدرکی دال بر عفونت جدی پوستی یا بافت نرم بدن دارد.

و ۳) ارگان زنده ای در نمونه کشت شده در آزمایشگاه وجود ندارد.

و ٤) نوع عفونت باکتریال است.

سپس:

دلیلی وجود دارد که ارگان زنده ای (غیر از آنهایی که در نمونه کشت شده دیده شده اند) که این عفونت را باعث شده است staphylococcus-cog-pos نام دارد.

R1(XCON) ۱۰-۸

R1 (یا XCON)شاید بهترین سیستم خبره مورد استفاده باشد . این برنامه توسط جان مک درموت و همکارانش در دانشگاه CMU بنا به در خواست کمپانی Digital Equipment Corporatin یا DEC توسعه یافت . هنگامی که DEC , کامپیوترهای نوع VAX را به بازار ارائه داد, بازاریابی خود را بر مفهوم انتخاب مشتری متمرکز ساخت . آنها می خواستند به مشتری اجازه دهند تا در انتخاب آیتم هایی که مایلند در تاسیسات مخصوصشان وجود داشته باشد , آزادی کامل را داشته

باشند. این آزادی انتخاب مشکلات سختی را برای DEC به بار آورد.
نیازهای مشتری صرفا" یک شکل اجمالی از چیزهایی است که یک پیکربندی کارا را می سازد. به هر حال , سفارشات مشتریان باید به صورت کامل به پیکر بندی ترجمه می شد. بعضی قطعات نظیر Power supply , کابل ها و غیره باید اضافه می شدندو قطعات دیگر نیاز به جزیئات بیشتری داشتند مانند تبدیل ابزارهای ذخیره دیسک به واحدهای دیسک و کنترلرها. بر روی محل استقرار کامپیوترها با داشتن طول کابل ها و مقصد های مورد نظر باید کار می شد. به عبارت ساده تر , بر جزئیات زیادی باید کار می شد که آنها نیاز به دانستن طیف قطعات موجود و محدودیت های مشاهده شده , داشت.
کارکنان DEC فورا" متوجه شدند در صورتی که این پروسه را مکانیزه نکنند, کارمندان خود را از دست خواهند داد. آنها ابتدا شروع به استفاده از متدهای سنتی کردند ولی وقتی متوجه شدند که به جایی نمی رسند , تصمیم به کمک گرفتن از CMU گرفتند.
نتیجه همکاری آنها R1 که یک سیستم با پیکربندی VAX و مبتنی بر دانش است, شد. یک نمونه قانون از R1 در زیر نشان داده شده است:
اگر:
۱) متن فعالیت جاری ابزارهای massbus را توزیع می کند.
و ۲) یک دیسک درایو تک پورته وجود دارد که هنوز به massbus اختصاص داده نشده
و۳) هیچ دیسک درایو دو پورته اختصاص داده نشده ای وجود ندارد
و ٤) تعدا د دستگاههایی که هر massbus باید حمایت کند شناخته شده است.
و ٥) نوع کابل اتصال دیسک درایو به massbus معین است
سپس: دیسک درایو را به massbus وصل کن.
مذاکرات در مورد R1 در دسامبر ۱۹۷۸ اغاز شد. در آن زمان حدود ٤۰۰ قانون وجود داشت . رقمی که تاکنون به بیشتر از ٤۰۰۰ هم گسترش یافته. تا سال ۱۹۸٤، DEC بدون R1 به ۸۰ کارمند دیگر احتیاج داشت و آنها باید متقاعد می شدند که آن برنامه خیلی بهتر از انسان ها می توانست عمل کند . حقیقتا" آنها در مورد قدرت تکنیکی که می خواستند از آن استفاده کنند مطمئن بودند. برای اینکه به مشتری در انتخاب مناسب

پیکربندی ها که به بهترین شکل با نیازها مطابقت داشت , کمک کنند . برای کمک به طراحی و آماده سازی محل استقرار کامپیوترها و برنامه ریزی تولید و تحویل پیکربندی طبق سفارشات و برای کمک به برنامه ریزی کارخانه , و کنترل اجناس و مغازه ها و غیره.

۸-۱۱ PROSPECTOR معدن یاب

معدن یاب یک سیستم مشاوره مبتنی بر کامپیوتر برای کمک به زمین شناسان در جستجوی کانی ها و سنگ های معدنی و برای کمک به ارزیاب پتانسیل معدنی در نواحی گسترده زمین شناسی , است. توسعه این برنامه در انستیتوی تحقیقات استنفورد در سال ۱۹۷۸ شروع شد. همانند مایسین , این یک سیستم مکالمه ای مبتنی بر قوانین اخذ شده از متخصصین است. معدن یاب فقط یک سیستم نیست. این برنامه با مدل های زمین شناسی واقعی تطبیق یافته همانند سه مدل مختلف رسوبات ماسه سنگهای اورانیوم , سنگ آذرین مس و مدل سنگ آذرین مولیبدیوم .

وظیفه یک زمین شناس در تشخیص یک محل توسط این واقعیت که نشانه ها برای یک سنگ ماسه خاص کمتر نامبهم هستند و یا اینکه نشانه ها همیشه وجود ندارند , مشکل می شود. بنابراین او باید بین نشانه های قابل توجه و ضدو نقیضها تعادل یجاد کند, مقدار نسبی آنها را وزن کند و به یک داوری احتمالی دست یابد.

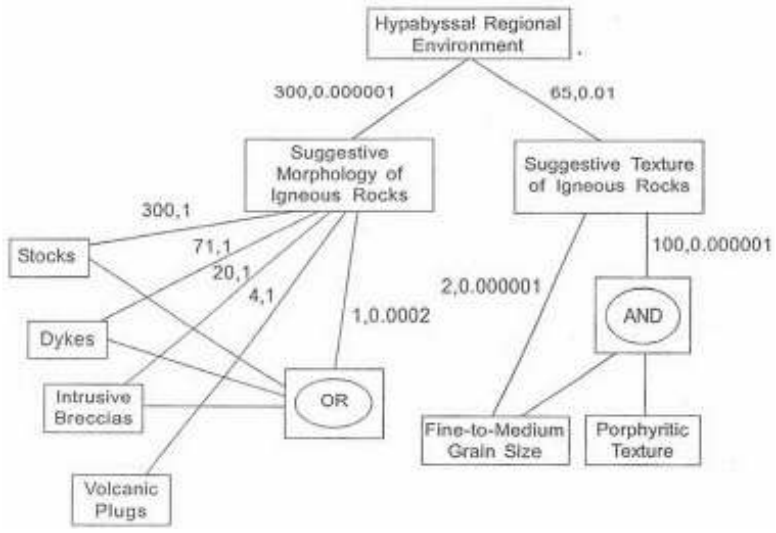
این عوامل اکتساب قوانین را سخت و گیج کننده می سازد. با این وجود, هنگامی که مدلها برای انجام تست ها ارائه شد تا در مقابل اکتشافات مکانهای شناخته شده و در مقابل قضاوتهای کارشناسان قرار گیرد, معدن یاب توانست فقط تائید ۷٪ را جلب کند.

اولین مشکل در ایجاد شبکه استنتاج زمین شناسی , شناسایی بیانیه ها برای مطرح شدن در ارزیابی نهایی است . این عوامل ابتدایی مقدار زیادی از ملاکها را خلاصه می کنند و مشکل تصمیم گیری در مورد درجه واقعیت ، همانند مشکل تصمیم گیری اصلی است. بنابراین عوامل ثانویه که عوامل اولیه را حمایت می کنند باید شناسایی شوند. به عنوان مثال مساعد بودن وضع ساختارهای نفتی توسط در نظر گرفتن یک کمربند حاشیه – اقلیمی خاص تصمیم گیری می شود.

این پردازش تصفیه بازگشتی ادامه می یابد تا عوامل حمایت کننده برابر با چیزی که

باید باشند , شوند.

یک دیاگرام از شبکه استنتاج واقعی استفاده شده برای تصمیم گیری در مورد اینکه آیا محیط آن ناحیه hypabyssal است یا خیر در شکل ۹–۸ نشان داده شده است . برای منظور ما , تفسیر زمین شناسی این دیاگرام می تواند نادیده گرفته شود.نکته مهم این است که چگونه مکانیزم های معدن یاب برای ترکیب ملاکها به کار گرفته می شوند. در این دیاگرام , ترکیبات عطفی و فصلی توسط گره های AND و OR نشانه گذاری شده است, ترکیبات وزن دار توسط کمانهای شماره گذاری شده توسط زوج اعداد , نشان داده می شوندو متون توسط کمان های نقطه گذاری شده حامل فواصل قطعی نشان داده شده اند.



(شکل ۹–۸) دیاگرام شبکه استنتاج برای تصمیم گیری محیط های نواحی hypabyssal

٤ گره زیر شبکه در قسمت پایینی سمت راست شکل، یک ساختار شبکه نمونه ای را نشان می دهند.

این زیر شبکه علاقه کارشناس برای دیدن نوع خاصی از توزیع اندازه برای صخره های کریستالی به نام porphyry texture را بیان می کند.

بیشتر سئوالاتی که از معدن یاب پرسیده می شود به جواب بله یا نه و یا پاسخ های

قطعی نیاز دارد . با اینکه بعضی از سئوالات برای مقادیر سئوال می کنند . در مورد قبلی نسبت احتمال برای قانون، تابعی از آن مقدار است .در صورتی که کاربر پاسخ یک سئوال را نمی داند, قطعیت از احتمال صفر به مقدار قبلی خود تغییر می کند. با اینکه این موضوع باعث تضعیف قطعیت نتیجه نهایی می شود اما برنامه را از پیشرفت باز نمی دارد.

فصل نهم

شبکه های عصبی

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

- **درک تفاوت بین هوش انسان و ماشین**
- **آشنایی با شبکه های عصبی و طرز کار آن**
- **آشنایی با معماری شبکه های مختلف**
- **توانایی مقایسه شبکه های عصبی با سیستم مبنی بر قاعده**
- **توانایی مقایسه شبکه های عصبی و سیستم خبره**
- **درک مزایا و محدودیات محاسبات عصبی**

۹-۱ مقدمه

وان نیومن۱ در سال ۱۹٤٦ ، ابداعش را که به عنوان کامپیوتر الکترونیکی همه منظوره نام گرفت ، به منظور خرد کردن اعداد ، تولید کرد و در سال ۲۰۰۱ ، اینتل تراشه ریزی را که Pentium-4 ، نامیده شد ، نه فقط جهت خرد کردن اعداد بلکه برای عملیات چند رسانه ای ارائه داد . فن آوری ، معماری ، نرم افزار ، و مفهوم پردازش اطلاعات ، چهار ابعاد کلیدی هستند که مشخصات مختلف کامپیوتر ها را تعیین می کنند . پیشرفت در همه این چهار فضای بزرگ برای تغییراتی که در بالا ذکر شده ، مسئول می باشد .

روند پیشرفت در تکنولوژی دستگاه ، طوری است که دستگاهها را به طور افزاینده نیرومند می سازد و یا به طور پیوسته دستگاه بیشتری را در یک تراشه تکی قرار دهند ، بنابر این تراکم بسته بندی را در حال افزایش است . روند پیشرفت در نرم افزار بدین گونه داست که موجب نزدیکتر شدن کامپیوترها به کاربران می شود که این نزدیکی از طریق فراهم آوردن ارتباطات چند کیفیتی و امکانات ورودی و خروجی طبیعی صورت می گیرد . مخصوصاً ، هدف برقراری ارتباط از طریق زبان طبیعی کاربر به جای استفاده از یک زبان مصنوعی کامپیوتر می باشد . پیشرفت های مهمی در سیر تکاملی معماری کامپیوتر در حال اتفاق افتادن است . مشخص شده که پردازشگرهای چند گانه یا منفرد که از مدل وان نیومن استفاده می کنند ، محدودیت های شدیدی در سرعت دارند . کامپیوتر های موازی و نوری نیز در حوزه وظایفشان به علت عدم مطابقت با مسائل محدود شده اند . روند پیشرفت به این گونه است که کاوش معماری های کامپیوتری مبتنی بر مدل پردازش های توزیع شده و موازی ، از طریق درک ما از ساختار و کارکرد شبکه عصبی زیستی ، تحریک شده است . نیروی حرکت دهنده پشت این توسعه خواستار ایجاد ماشین هائی به طور افزاینده هوشمند می باشد (قابلیت تحمل خرابی) . مفهوم اینکه ما چگونگی پردازش اطلاعات در کامپیوترها نیز در حال تغییر است . این تغییر مهمترین تغییراست ،به طوری که کامپیوتر ، یا یک ماشین در یک شبیه ساز را که کارکردهائی مانند مغز انسان دارد ، هدایت می کند .

---

1 Von Neumann

کار های معمولی	کارهای رسمی(دارای فکر)	کارهای تخصصی

ماشین هوشمند تصور محقق ها در دهه ۵۰ بود . جان مکارتی۱ از MIT اصطلاح هوش مصنوعی را در سال ۱۹۵٦ ابداع کرد . با پیشرفت هایی در کامپیوتر های دیجیتال و روش پردازش اطلاعات محققین شروع به در یافت نتیجه های مثبتی کردند .

بعضی از رویدادهای مهم AI شامل موارد زیر است :

۵۷-۱۹۵٦: نگرشگر منطقی ، یکی از اولین برنامه ها برای اثبات برهان خودکار که به وسیله نوول۲ ، شاو۳ ، و سایمن٤ کامل شده بود .

٦۵-۱۹٦۱ : .A.L ساموئل۵ برنامه ای را که بازی چکرز را در سطحی بالا آموخته است توسعه داد .این برنامه نه فقط با حریف ها بازی می کرد بلکه از تجربیاتش جهت بهبود بخشیدن اجرای آخریش استفاده می کرد .

٦۵-۱۹٦۱ :.J.A رابینسون٦ تحلیلی به عنوان یک روش استنباط در منطق معرفی کرد.

۱۹٦۸: کار کردن روی MACSYMA ، یک برنامه واکنشی بزرگ است که انواع بی شماری از مسائل ریاضی را حل می کند .

بعضی ازدامنه وظیفه AI در جدول ۹,۱ لیست شده اند :

---

1 John McCarthy
2Newell
3 Shaw
4 Simon
5 Samuel
6 Robinson

ادراک	بازی ها	مهندسی
تصور	شطرنج	طراحی
گفتار	تخته نرد	اکتشاف خرابی
زبان طبیعی	چکرز	ساخت
درک	گشتن	برنامه ریزی
نسل	ریاضی	تحلیل علمی
ترجمه	هندسه	تشخیص طبی
شعور	منطق	تحلیل اقتصادی
استدلال	حساب جامعه	
کنترل ربات	اثبات خصوصیات برنامه	

جدول ۹,۱ دامنه وظیفه AI

یک فرد مهارت های لازم را که در جدول بالا به آن اشاره شده است در یک ترتیب استاندارد یاد می گیرد . ابتدا مهارتهای ادراکی ، وابسته به زبان شناسی ، و شعوری یاد گرفته شده اند و سپس مهارت های تخصصی مانند مهندسی ، پزشکی ، یا سرمایه گذاری فراگرفته شده اند. ممکن است به نظر آید که برای داشتن یک معنی واضح ، مهارتهای اولیه آسانتر هستند و از این رو جهت نسخه برداری کامپیوتری شده نسبت به مهارت های تخصصی تر و اخیرالذکر بیشتر قابل پاسخگوئی هستند . به این دلیل ، میزان زیادی از کار اصلی و ابتدایی AI در این فضاهای اولیه متمرکز شده است .

ولی معلوم شد این فرض ساده درست نیست . اگرچه مهارت تخصصی دانشی را که بسیاری از ما فاقد آنیم ، فرا می گیرد ، اما این مهارتها اغلب به نسبت انجام دادن مهارت های عمومی ، دانش بسیار کمتری را فرا می گیرند و آن دانش معمولاً برای ارائه و سروکار داشتن با برنامه داخلی آسانتر است .

در نتیجه ، نواحی مسئله که AI در آن بیش از همه در حال رشد و پیشرفت است ، عمدتاً به عنوان یک نظم عملی ، دامنه هائی هستند که فقط نظریه های تخصص یافته را

.

بدون کمک دانش شعوری (حس عام) فرا می گیرند . اکنون هزاران برنامه که سیستم خبره نامیده شده اند در یک عملیات روز به روز در تمامی دامنه های صنعت و دولت وجود دارد . هر یک از این سیستم ها تلاشی برای حل قسمتی ، یا شاید تمام یک مسئله مهم و عملی را دارند ، که قبلاً به نظریه انسانی کمیاب احتیاج داشته .

اگرچه پیشرفت های موجود در AI امید بخش ظاهر شد ، ولی وقتی در امور هوشمند دنیای واقعی مانند گفتار ، بینایی و پردازش کردن زبان طبیعی به کاربرده شده ، تکنیک های AI نارساییها و تردی هایش را تا اندازه ای نشان داد . مانند روش الکوریتمی برای حل مسئله ، حتی تکنیک های AI به تشخیص و نگاشت واضح از مسئله داده شده در یک فرم مناسب برای تکنیک هائی که قابل اجرا باشند ، نیاز دارند . برای مثال ، به منظور به کار بردن روش های جستجوی اکتشافی ، نیازی برای نگاشتن مسئله به عنوان مسئله جستجویی وجود دارد . همچنین ، برای حل کردن یک مسئله استفاده کردن از یک رهیافت قانونمند ، که قوانین را به طور واضح و روشن شرح دهد ، لازم است . از طریق هوش مصنوعی ، سعی شده است که هوش در ماشین هائی که توانائی انجام تعداد بسیاری از عملیات ها بر روی داده را دارند و قادر به ذخیره ، مطابقت و سپس بازیابی حجم زیادی از داده می باشند ، آورده شود . هنگامی که نتیجه نهایی کارایی ماشین با کارایی انسان مقایسه شد یک شکاف بسیار بزرگی پیدا شد . راه هائی که کارها به وسیله ماشین و یک انسان انجام شده است ، اساساً با هم متفاوت می باشند . ماشین ، کاری را در یک روش ترتیبی مرحله به مرحله که از طریق یک الگوریتم دیکته شده ، انجام می دهد ، که ممکن است این الگوریتم به وسیله بعضی بحث های اکتشافی شناخته شده ، اصلاح شده باشد . بنابراین ، این الگوریتم و بحث های اکتشافی مجبورند که برای یک کار معین مشتق شده باشند . یکبار مشتق می شوند و ثابت باقی می مانند .

مشکلات گفتار ، بینایی و پردازش زبان طبیعی توجه طراحان سیستم های هوشمند را به سمت خود معطوف کرده . سختترین موضوع در این موارد مشتق کردن توصیفی از الگو موجود در یک داده در بخش هائی از نماد و مشتق کردن یک مجموعه از قوانین

که دانش دامنه مسئله را ارائه می دهند ، می باشد . حتی اگر بتوانیم الگوریتم های بسیار پیچیده و متمرکز را با فناوری جاری اجرا و محاسبه کنیم ولی مشخص شده است که ما نمی توانیم تشریح طرح و دانش را به طور کامل برای مشکل معین اشتقاق کنیم .

بنابراین توانایی محض یک ماشین که پردازش مقدار زیادی نشانه و استنتاج منطقی را انجام می دهد اثری در یک رفتار هوشمند ندارد .

زمانی که ما تشریح " خاکستری-بزرگ- پستاندار " را می خوانیم ما به طور خودکار به فکر فیل و خصایص وابسته به آن می افتیم ما به حافظه مان از طریق محتوا دسترسی داریم .

در اجرای سنتی دسترسی با محتوا شامل جستجو های گران و رویه های تطبیق است .

ما در فصل ۱ دیده ایم که تاریخچه هوش مصنوعی کنجکاو است .اولین مشکل از نظر محققین هوش مصنوعی مشکلاتی مانند شطرنج و اثبات قضیه بود زیرا آنها فکر کردند که این موارد ذات هوش باشد .

یک بچه ۵ ساله به آسانی بر پرو سه بینایی و فهم زبان مسلط می شود بنابراین در نظر گرفته شده که این پروسه ها برای انسان سخت نباشد .

این روزها ما برنامه های شطرنج تخصصی و برنامه تشخیص پزشکی تخصصی داریم ولی هیچ برنامه ای که بتواند مهارتهای ادراکی یک بچه را تطبیق دهد نداریم.

محققین در این مورد که یک عدم مطابقت پایه ای بین فن آوری پردازش اطلاعات کامپیوتر و فناوری که به وسیله مغز استفاده می شود وجود دارد با یکدیگر مباحثه می کنند .

۹-۲ تفاوت بین هوش انسان و ماشین

تفاوت اصلی بین هوش انسان و ماشین از یک واقعیت می آید که یک انسان هر چیزی را مانند یک الگو یا طرح درک می کند در حالیکه برای یک ماشین هر چیزی یک داده است .

حتی در جریان عادی داده شامل اعدادصحیح مانند (شماره های تلفن – شماره های

حساب بانکی)است انسان ها دوست دارند که یک الگو را درک کنند .
بیاد آوردن داده نیز یک فرم نرمال از یک الگوی ذخیره شده است .اگر هیچ الگویی وجود نداشته باشد برای یک انسان به یاد آوردن و دوباره ساختن داده برای بعد خیلی سخت است .
بنابر این ذخیره سازی و عملیات به یاد آوردن در انسان ها و ماشین ها به وسیله مکانیزم های متفاوتی انجام می شود .
نوع الگو در ذخیره سازی و به یاد آوردن خودکار تحمل خرابی را به سیستم انسان می دهد.
انسانها و ماشینها متفاوت هستند چون انسان الگوها را می فهمد در حالیکه ماشین می تواند بگوید که الگوها را در داده تشخیص می دهد .
به عبارت دیگر یک انسان می تواند تمام اشیاء را در داده بگیرد حتی اگر هیچ شناسایی واضح از تابع داده در دست نباشد .
برای مثال نام شخصی را که در یک متن روان با دست نوشته شده در نظر بگیرید حتی اگر الگوهای تک برای هر یک از حرفها آشکار نباشد اسم از طریق اشاره های دیداری که روی متن نوشته شده فراهم می شود فهمیده می شود .
همچنین گفتار از طریق الگوهایی متناظر با صداهایی فردی که ممکن است تحریف شده باشد فهمیده می شود بعضی وقتها اندازه و وسعت قابل تشخیص نیست .
خصوصیت اصلی دیگر یک انسان توانایی است که به طور مداوم از مثال هایی یاد می گیرد که به خوبی فهمیده نشده است که آن را در یک مد الگوریتمی در یک ماشین اجرا کند .
انسانها قابلیت ساخت الگو های ذهنی در شبکه های عصبی زیستی شان از داده های ورودی به فرم اعداد – متن – عکس –صدا و غیره و استفاده از مکانیزم حسی شان از نگرش- صدا-لمس-بو و مزه را دارند .
این الگوهای ذهنی حتی وقتی که داده پر خش یا ناقص شده از طریق تغییراتی مانند ترجمه- چرخش و پیمایش باشد ایجاد می شوند . الگوها همچنین از توالی موقت داده

در یک مورد گفتار و عکس ایجاد می شوند.
انسان ها توانایی دوباره به یاد آوردن الگوهای ذخیره شده حتی وقتی که اطلاعات ورودی پر خش یا جزئی یا آمیخته با اطلاعاتی که وابسته به الگوهای دیگر هستند را دارد .

۹-۳ خصوصیات زیستی شبکه های عصبی

بعضی خصوصیات جذاب زیستی شبکه های عصبی که آنرا حتی به پیچیده ترین سیستم کامپیوتری هوش مصنوعی برای کارهای متفاوت ممتاز می سازد موارد زیر است :

* قدرتمند و تحمل خرابی :خراب شدن سلولهای عصبی به نظر نمی آید که روی کارایی تاثیر قابل توجهی بگذارد .

* انعطاف پذیری :شبکه به طور خودکار خود را به محیط جدید بدون استفاده از هیچ دستورالعمل پیش برنامه نویسی شده مطابقت می دهد.

* توانایی مواجه شدن با موقعیت های داده ای مختلف : شبکه می تواند با اطلاعاتی که دارای شرایط عدم قطعیت هستند یا احتمالی یا خش دار یا متناقض هستند مواجه شود .*محاسبات انبوه : شبکه به طور روتین خیلی از عملیات را به طور موازی و همچنین کار های معین شده را در یک روش توزیع شده انجام می دهد .

۹-٤ چگونه مغز انسان یاد می گیرد ؟

(شکل ۱-۹) وجود دارد سیگنال ها را از دیگران از طریق یک میزبان (neuron) در مغز انسان نوعی یاخته عصبی نامیده می شود جمع می کند .(dendrites)کوچک که دندریت شناخته می شود به بیرون می فرستد که axon نرون فعالیت الکتریکی را از طریق یک رشته بلند و باریک که بنام

نامیده می شود (شکل ۲-۹) فعالیت (synapse) به هزاران شاخه تقسیم می شود. در پایان هر شاخه یک ترکیبی که را به نتیجه الکتریکی تبدیل می کند که مانع یا موجب برانگیختن فعالیتها در نرون های متصل شده می شود .axon وقتی یک نرون ورودی

برانگیختن را در یافت میکند با ورودی ممانعتش مقایسه می شود سپس یک فعالیت الکتریکی به axon میفرستد .

با تاثیر یک نرون روی تغییرات دیگری حاصل می شود . synapse یادگیری به وسیله تغییر نتایج

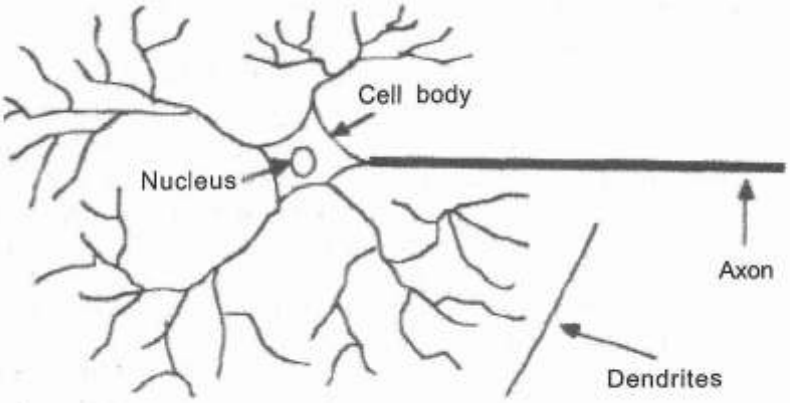


Figure 9.1 Components of a neuron.

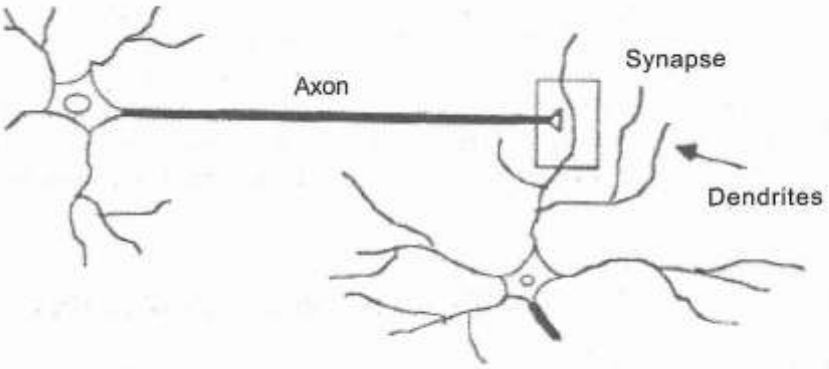


Figure 9.2 The synapse.

۹-۵ از نرون انسان تا نرون مصنوعی

ما این شبکه های عصبی را هدایت می کنیم که مشخصات لازم نرونها و ارتباطات درونیشان را استنتاج کنیم سپس یک کامپیوتر برای شبیه سازی این مشخصه ها برنامه نویسی می کنیم اگرچه دانش ما از نرونها ناقص است و قدرت محاسبه مان محدود است مدل هایمان لزوما کمال گرایی های کل شبکه های حقیقی نرون ها هستند .
یک نمونه مدل نرون در شکل ۳-۹ نشان داده شده است .

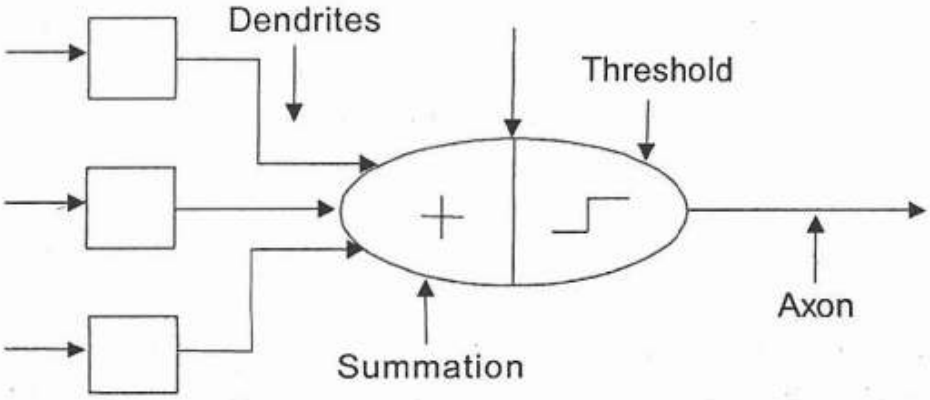


Figure 9.3 The neuron model.

۹-٦ چگونه شبکه های عصبی یاد می گیرند ؟

واحد ها ساخته شده است که در مدل (units) شبکه های عصبی مصنوعی عموما از یک سری اتصالات داخلی بنام
نرون به کار می رود .
به وسیله یک وزن قابل اصلاح مدل شده است که به هر اتصالی پیوسته شده است وظیفه Synapse.
هر واحد الگوهای فعالیتهای ورودی را که در یافت می کند به یک فعالیت خروجی منفرد تبدیل می کند که آن به واحد های دیگر منتشر می شود .

.

(واحد ها) این تبدیلات را در دو مرحله انجام می دهند :units

۱) واحد هر فعالیت ورودی را با وزنش روی هر اتصال ضرب می کند و همه این ورودی های دارای وزن را با هم جمع می کند تا یک کمیتی بنا م ورودی جامع به دست آورد .

۲)یک واحد از یک تابع ورودی – خروجی استفاده می کند که ورودی جامع را به فعالیت خروجی تبدیل می کند .

یک تحقیق تکنیکی تر از نرون منفرد در شکل ٤-۹ داده شده است که نشان می دهد که آن می تواند یک بردار از از می روند .N وزن از ابعاد W داشته باشد .این ورودی ها از طریق بردار N از ابعاد xورودی

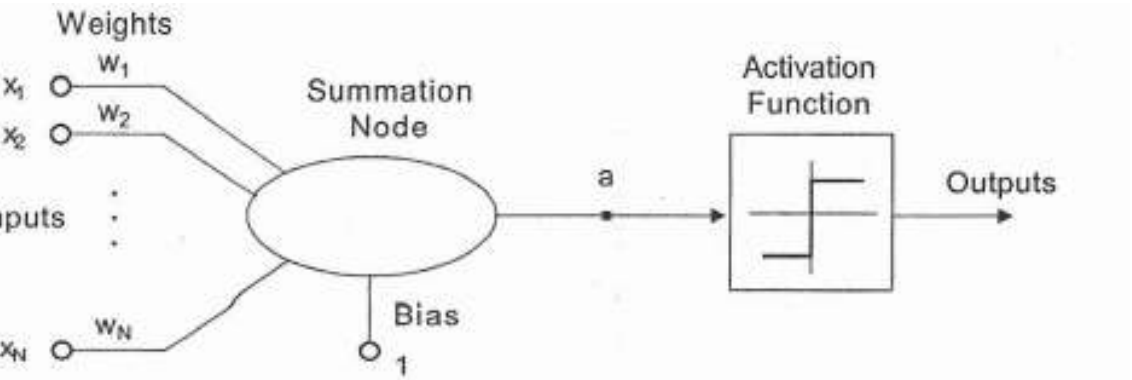
بعلاوه x,w یک تولید نقطه ای از بردار'a' تولید شده جایی که 'a' پردازش ادامه پیدا می کند با مجموع گره ها

یک اریب است .

سپس از طریق یک تابع فعال سازی که مقدار آنرا با یک آستانه از پیش تعریف شده مقایسه می کند پردازش 'a'

بیرون انداخته نمی شود و اگر بالاتر از مقدار perceptron می شود .اگر مقدار آن پایین تر از مقدار آستانه باشد

یک پالس به بیرون از دامنه ای که از پیش تعریف شده انداخته خواهد شد . perceptron آستانه باشد



$$a = w_1x_1 + w_2x_2 + \ldots + w_nx_n + \text{Bias}$$

$$\text{output} = \text{Threshold}(a)$$

$$\text{where Threshold}(a) = \begin{cases} -1, & \text{for all } a \le 0 \\ 1, & \text{for all } a > 0 \end{cases}$$

Figure 9.4 Learning in neural network.

(شبکه عصبی مصنوعی) به هر دوی وزن ها و توابع ورودی – خروجی (تابع انتقال)
بستگی دارد که آن ANNرفتار
برای واحدها مشخص شده است .
این توابع عموما در یکی از این سه دسته بندی قرار می گیرند :

۱) خطی
۲)آستانه
۳) حلقوی

برای واحد های خطی فعالیت خروجی متناسب با خروجی دارای وزن کلی شده است .

برای واحدهای آستانه خروجی مجموعه ای از یک یا دو مرحله است بسته به اینکه آیا ورودی کل بزرگتر یا کوچکتر از مقدار آستانه است .

.

برای واحد های حلقوی خروجی به طور مداوم متغیر است ولی نه به طور خطی مانند تغییرات ورودی .واحد های حلقوی شباهت بیشتری به نرون واقعی نسبت به واحدهای آستانه یا خطی دارند ولی هر سه باید به عنوان تخریب های بد در نظر گرفته شوند.

برای ایجاد یک شبکه عصبی که بعضی کارهای خاص را انجام می دهد ما باید اینکه چگونه واحدها به یکدیگر وصل شوند را انتخاب کنیم و باید اوزان را روی اتصالات دقیقا تنظیم کنیم .

اتصالات تصمیم می گیرند که آیا این برای یک واحد ممکن است که روی دیگری تاثیر بگذارد و اوزان قدرت تاثیر را تعیین می کنند .

عمومی ترین نوع شبکه عصبی مصنوعی شامل سه گروه یا لایه یا واحد (شکل ۵-۹) است.یک لایه از واحد های ورودی به یک لایه از واحدهای پنهان که آن هم به یک لایه از واحدهای خروجی متصل شده وصل شده است .

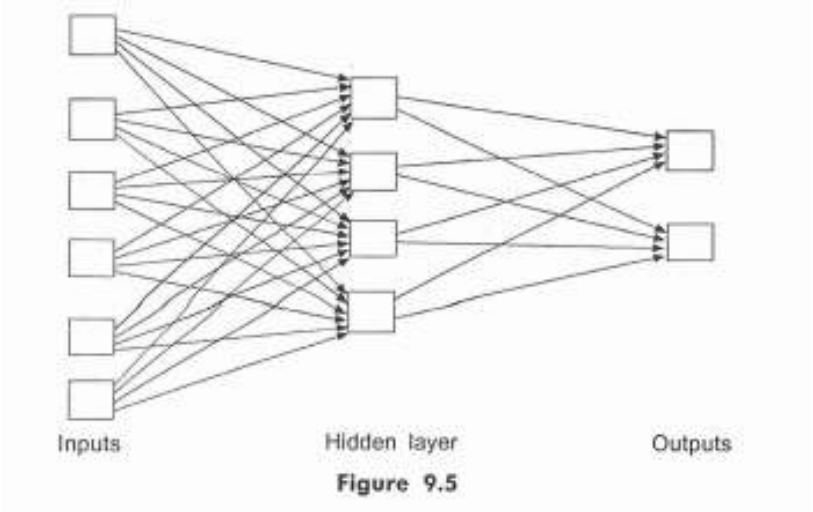


Figure 9.5

فعالیت واحدهای ورودی یک سری اطلاعات نارس را ارائه می دهد که آن به شبکه تزریق می شود .

فعالیت هر یک از واحد های مخفی به وسیله فعالیت های واحد های ورودی و اوزان روی اتصالات بین واحد های ورودی و واحدهای مخفی تعیین شده است .

رفتار واحد های خروجی بستگی به فعالیت های واحد های پنهان و اوزان بین واحد های خروجی و مخفی دارد .

این نوع شبکه ساده جالب است زیرا واحد های مخفی آزاد هستند تا نمایش ورودی را بسازند . اوزان بین واحد های ورودی و واحد های مخفی تعیین می کنند که چه زمانی هر واحد مخفی فعال است و همچنین با تغییر این اوزان یک واحد مخفی می تواند انتخاب کند که چه چیزی ارائه شود .

ما می توانیم یک شبکه سه لایه ای را برای انجام یک کار خاص با استفاده از رویه های زیر آموزش دهیم :

الف) ما شبکه ای را با مثال های آموزشی ارائه می دهیم که شامل الگو های کاری برای واحد های ورودی با یکدیگر با الگو های کاری خواسته شده برای واحدهای خروجی است .

ب) ما تعیین می کنیم چگونه خروجی واقعی شبکه با خروجی خواسته شده مورد مطابقت قرار می گیرد .

ج) ما اوزان هر یک از اتصالات را تغییر می دهیم بنابراین شبکه یک تخمین بهتری از خروجی های خواسته شده انجام می دهد .

۹-۷ فراگیری الگوریتم ها

یادگیری می تواند با ساکن شدن وزنها در یک روش اصولی دیده شود .چندین قانون آموزشی در حال استفاده وجود دارد و به درخواست یک معماری یا برنامه کاربردی قانونهای جدیدی پیشنهاد شده است .

بر اساس خواصی که از این الگوریتمهای آموزشی انتظار می رود که داشته باشند می

.

توانند مطابق زیر دسته بندی شوند:

 # نظارت شده یا نظارت نشده: در آموزش نظارت شده تنظیم وزن براساس انحراف خروجی‌های خواسته شده از خروجی‌های واقعی تعیین می‌شود. فراگیری نظارت شده ممکن است برای فراگیری ساختمانی یا برای فراگیری موقتی استفاده شود. فراگیری ساختمانی به گرفتن اوزان و رابطه بین جفت الگوهای ورودی – خروجی داده شده مربوط است. فراگیری موقتی به گرفتن وزن و رابطه بین الگوها در یک توالی از الگوها مربوط می‌شود.

فراگیری نظارت نشده خرابی را در یک مجموعه الگوهای معیین شده کشف می‌کند و الگوها را برطبق آن تشخیص می‌دهد. هیچ خروجی خواسته شده ویژه بیرونی در این مورد وجود ندارد فراگیری نظارت نشده بیشتر از اطلاعات محلی برای بروز رسانی اوزان استفاده می‌کند، اطلاعات محلی شامل سیگنالها یا مقادیر فعالسازی از واحدهایی در پایان اتصال برای وزن های بروز شده استفاده می کند.

 # روی خط یا برون خطی (online or offline)

دریک فراگیری برون خطی همه الگوهای داده شده با یکدیگر استفاده می‌شود تا اوزان را تعیین کند از طرف دیگر در فراگیری روی خط اطلاعات در هر الگوی جدید بوسیله تنظیم کردن افزایشی اوزان داخل شبکه می‌شوند. بنابراین فراگیری روی خط به شبکه عصبی اجازه می‌دهد که اطلاعات را همواره به روز رسانی کند اگرچه فراگیری برون خطی راه حلهای بهتری را نسبت به روی خط از اطلاعاتی که با به کارگیری تمام نمونه‌های آموزشی استخراج می‌شود فراهم می‌کند.

گسسته یا پیوسته

آیا مقادیر اوزان در مراحلی گسسته بروز رسانی می‌شود یا در زمان پیوسته نوع آن الگوریتم تعیین می‌شود؟

همه این فاکتورها نه فقط روی همگرایی اوزان بلکه همچنین توانایی شبکه که از مثالهای آموزشی بیاموزد تأثیر میگذارند.

۹-۸ معماری شبکه های مختلف و کاربردهایشان

معماری شبکه های مختلف و کاربردهایشان در جدول زیر تشریح شده است :

شبکه	کاربرد ۹-۰	توضیحات
ADALINE	رده بندی الگو شناسی ارقام	یک شبکه یک لایه ای تکثیر است و روی یک کار الگو شناسی ترتیب شده است و هدف آن ، این است که یک ارائه
BPN	پیش بینی کردن سری های زمانی سالیانه تعدادی از لکه های خورشیدی	را به کلاسهای متناظر دسته بندی از ارقام Bitmap ۹- ۰ کند.به علت قابلیت های محدود، این شبکه فقط الگوهای آموزشی دقیق را تشخیص می دهد.وقتی که کاربر وارد یک شبکه چندلایه ای می شود یک درجه قابل توجه از تحمل خرابی می تواند بدست آید .
Hopfield	حافظه شرکت پذیر خودکار	این برنامه شبکه چند لایه ای جدیدی را با مقدار حرکت اجرا می کند . این برنامه بکار میرود تا ترکیباتی را در سریهای زمانی که به شبکه ارائه می شود و از یک حافظه تاخیر استفاده می کنند کشف کند . برنامه آموزش میدهد تا فعالیت های آینده لکه های خورشیدی را با استفاده از داده های تاریخی که در سه قرن گذشته جمع آوری شده است OVERFITTING پیش بینی کنیم . برای جلوگیری از

.

پایان رویه فراگیری به وسیله روش آموزشی متوقف کننده کنترل می شود. این مدل به عنوان حافظه شرکت پذیر خودکار برای ذخیره . Bitmap سازی و فراخوانی یک مجموعه تصاویر تصاویر بوسیله ماتریکس وزن متناظر محاسبه شده، ذخیره شده اند و پس از آن شروع از یک پیکر بندی اختیاری خواهد بود . حافظه دقیقا روی عکس ذخیره شده ساکن خواهد شد که نزدیکترین به پیکربندی شروع از نظر فاصله همینگ است .بنابراین برای نسخه خراب یا ناقص از یک عکس ذخیره شده معین ، شبکه قادر خواهد بود که عکس اصلی متناظر را فراخوانی کند .	حافظه اشتراک پذیر و پیوستگی نام و شماره تلفن	BAM حافظه اشتراک پذیر دو جهته
حافظه اشتراک پذیر دو جهته می تواند به عنوان عمومیت نشان داده شود . در این مورد Hopfield دادن به مدل پیوستگی بین اسامی و شماره تلفن متناظر است .بعد از رمز گذاری مجموعه ای از مثال ها ، شبکه وقتی با اسم ارائه می شود قادر به فراخوانی شماره تلفن متناظر و	بهینه سازی مسئله فروشنده دوره گرد	Boltzman

برعکس خواهد بود . حافظه حتی یک درجه محدود شده تحمل خرابی در مورد الگوهای ورودی خراب شده نشان می دهد . است . Hopfield این ماشین یک نسخه احتمالی از مدل شروع با دمای بالا و بعد به تدریج سرد شدن به شبکه اجازه می دهد که به تعادل در هر مرحله دست یابد . شانس ها در شبکه خوب هستند که در یک حداقل سراسری از تابع انرژی متناظر ساکن خواهند شد این پروسه نامیده می شود . شبکه سپس برایSimulated annealing حل مسائل بهینه سازی معروف استفاده می شود . ماتریکس وزن مانند تابع انرژی حداقل سراسری که به حل ویژه نمونه هایی از مسئله فروشنده دوره گرد مربوط است ، انتخاب شده است . شبکه پخش شمارنده یک شبکه رقابتی طراحی شده است تا مانند یک جدول	تعیین دید زاویه های چرخش مسئله کنترل تعادل قطب	CPN SOM نقشه خود سازمانده

.

جستجوی خود برنامه نویسی شده با توانایی های اضافی برای داخل کردن بین داده های وارد شده عمل کند . کاربرد این است که چرخش زاویه ای یک شیئ موشکی شکل را تعیین کند که تصاویری است که به شبکه به عنوان ارائه شده است .کارایی شبکه به علت دقت Bitmap الگوی محدود شده است .Bitmap پایین نقشه خود سازمانده یک شبکه رقابتی با توانایی است که نگاشت توپولوژی بین فضاهای ورودی و خروجی تشکیل دهد .در این برنامه شبکه فرا میگیرد تا یک قطب را با اعمال نیرو در پایه ی قطب متعادل سازد . کار شبکه این است که یک نگاشت بین متغیرهای قطب و نیروی بهین برقرار کند تا تعادل آنرا نگه دارد این کار با استفاده از روش فراگیری تقویت انجام می شود . برای هر حالت داده شده قطب ، شبکه برای یک اختلاف ناچیز از نیروی نگاشت شده تلاش می کند . اگر نتیجه نیروی جدید در کنترل بهتر		

است نگاشت تغییر یافته است و از متغیر های جاری قطب استفاده می شود و نیروی جدید به عنوان بردار آموزشی در نظر گرفته می شود		

Hopfield-۸-۹

۱ شبکه-های Hopfield یکی از مراحل برجسته برای تجدید حیات جاری در زمینه شبکه‌های عصبی بود. مدل شرکت پذیری بوسیله

در اوایل ۱۹۸۰ پیشنهاد شد او مدلش را به عنوان تئوری از حافظه با ویژگیهای زیر پیشنهاد کرد:

* ارائه توزیع شده: یک حافظه به عنوان یک الگوی کاری در سرتاسر مجموعه

.

واحدهای ذخیره شده حافظه‌ها میتوانند روی دیگری اضافه شود، حافظه‌های مختلف می‌توانند با الگوهای متفاوت روی یک واحد یکسان ارائه شوند.

* کنترل توزیع شده همزمان: هر واحد تصمیم می‌گیرد که فعالسازیش براساس تنها موقعیت محلیش باشد همه این تصمیمات محلی به راه حل سراسری یا یک حافظه ویژه اضافه می‌شود.

* حافظه نشانه پذیر از روی محتوا: برای بازیابی یک الگو ما فقط احتیاج داریم تا قسمتی از آن را تعریف کنیم شبکه به طور خودکار نزدیکترین انطباقها را پیدا می‌کند.

* تحمل خرابی: اگر تعداد کمی از واحدها در یک شبکه به طور نادرست عمل کنند یا بطور کامل نادرست باشند تمام شبکه درست عمل خواهد کرد.

یک شبکه آسان از پردازش اجزا یا واحدهایی که روشن یا خاموش هستند پیشنهاد کرد. واحدها Hopfield

به یکدیگر از طریق اتصالات متقارن دارای وزن وصل شدند. اتصالات مثبت بین دو واحد نشان می‌دهد که هر دو واحد تمایل دارند که همدیگر را فعال سازند و اتصالات منفی نشان می‌دهند که واحدهای متصل شده مایل خواهند بود که همدیگر را خاموش سازند.

واحدهای انفرادی حالتهای انفرادیشان را تا هنگامی که آنها برای بروزرسانی جدید انتخاب شده‌اند حفظ می‌کنند. انتخاب یک واحد بطور تصادفی صورت می‌گیرد. اگر هر یک فعال هستند واحد انتخاب شده مجموع اتصالات به واحدهای فعال را محاسبه می‌کند اگر مجموع مثبت باشد سپس واحد خودش را روشن می‌کند از طرف دیگر واحد خودش را خاموش می‌کند این پروسه که استراحت موازی نامیده می‌شود ادامه پیدا می‌کند تا هنگامی که هیچ واحدی نتواند مقدارش را تغییر دهد و به شبکه گفته می‌شود که آن به یک حالت پایدار رسیده است.

این شبکه‌ها فقط تعداد محدودی از حالتهای پایدار دارند و هر یک به وسیله یک الگوی فعالسازی در سراسر واحدهای شبکه ارائه می‌شوند.

ثابت می‌کند قدرت اتصالات مجموعه اولیه و هر حالت اولیه فعالسازی شبکه در

نزدیکترین حالت پایدار Hopfield
از طریق استراحت موازی ساکن خواهند شد و هیچ نوسان یا واگرایی نمی‌تواند اتفاق بیفتد.
این شبکه به عنوان حافظه شرکت پذیر دسته‌بندی شده است.

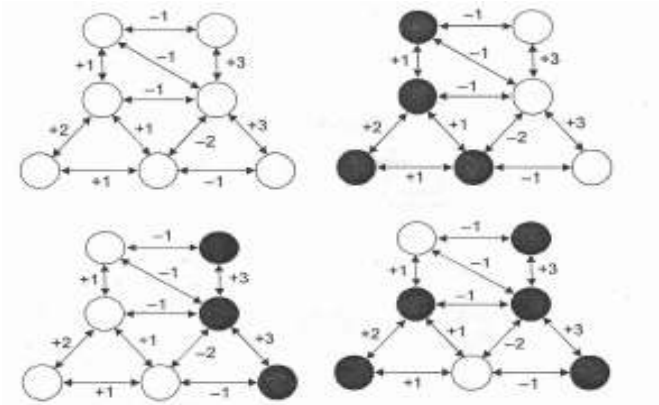


Figure 9.6 The four stable states of a Hopfield network.

۹-۸-۲ SOM یا شبکه های Kohonen

بر خلاف **SOM از فراگیری نظارت نشده استفاده می‌کند از این رو قابل اجرا** دیگر شبکه‌های عصبی، شبکهبرای انواع مختلف مسائل است.
واحدهای یک لایه‌ای دارند در حین آموزش، دسته‌های واحدها با کلاسها مختلف شبکه‌های یکسان آماری) که در داده‌های آموزشی ارائه شده **Kohonen(با ویژگیهای** است پیوسته می‌شود. این شبکه‌ها برای کاربردهایی که، مهم است تعداد زیادی از مثالها را تجزیه و تحلیل کنند و گروهها را با خصیصه‌های یکسان تشخیص دهند، مفید است.
این شبکه در شکل ۷-۹ نشان داده شده است:

.

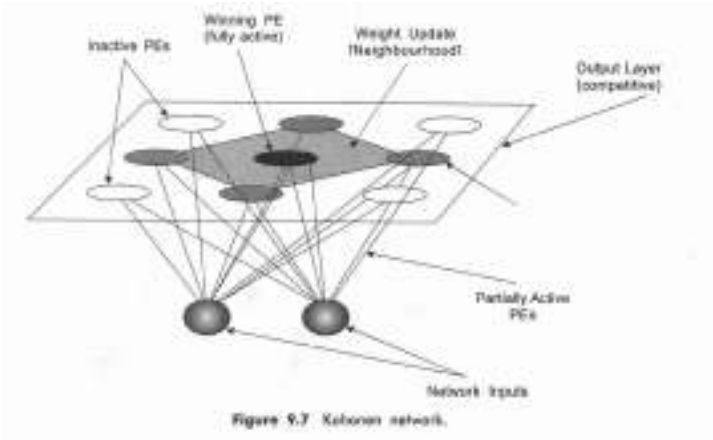


Figure 9.7 Kohonen network.

ورودی‌ها را با ایجاد یک نگاشت دو بعدی روی لایحهایش از داده‌های ورودی دسته‌بندی می‌کند بنابراین SOM شبکه

ترتیب داده‌های ورودی حفظ می‌شود، برای مثال اگر دو الگو ورودی نزدیک به هم یا یکسان از نظر اندازه در فضای ورودی باشند سپس آنها به اجزای پردازش که به همدیگر در لایه‌های دو بعدی نزدیک هستند نگاشت خواهند شد.

)Kohonen(همچنین شناخته شده به عنوان لایه‌های

۹-۹ برخی شبکه‌های ساده

ما می‌توانیم از آنچه که اخیراً یادگرفته‌ایم برای اینکه نشان دهیم یک شبکه عصبی ساده به عنوان یک دروازه منطقی عمل می‌کند استفاده کنیم.

را مدل کرده است. (جدول ۱-۹ تا ۳-۹) NOT, OR, AND شکل(۸-۹)

بعضی از جدولهای درستی برای

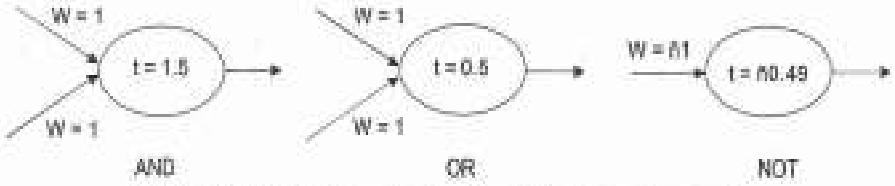


Figure 9.8 Simple neural network which act as a logic gate.

Table 9.1 AND truth table

$Input_1$	$Input_2$	$Output$
0	0	0
0	1	0
1	0	0
1	1	1

Table 9.2 OR truth table

$Input_1$	$Input_2$	$Output$
0	0	0
0	1	1
1	0	1
1	1	1

Table 9.3 NOT truth table

$Input$	$Output$
0	1
1	0

در این شبکه‌ها ما مراحل تابع فعالسازی را استفاده می‌کنیم.

شما باید خودتان را متقاعد کنید که این شبکه‌ها خروجی‌های صحیح را برای تمام ورودی‌های مجاز می‌سازند.

شما متوجه خواهید شد که هر نرون آستانه متفاوتی دارد از یک دیدگاه محاسباتی آن ممکن است آسانتر باشد اگر تمام نرونها مقادیر آستانه یکسان داشته باشد و آستانه واقعی به طریقی با اوزان مدل شده است در واقع آن ممکن است که آنرا دقیقا انجام دهد.

را اصلاح کرده است که نرون ورودی اضافی دارد.ANDشبکه در شکل ۹-۹ را در نظر بگیرید که شبکه

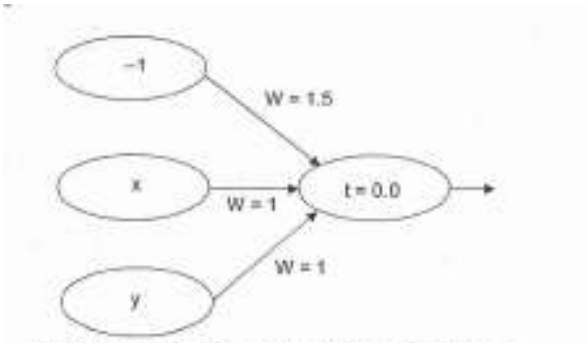


Figure 9.9 AND network with extra input neuron.

است به جزء این که یک نرون ورودی اضافی وجود دارد که فعال سازی همیشه به ۱-AND این شکل مانند شبکه

تنظیم شده است، آستانه نرون با وزن ارائه شده است که این نرون اضافی به نرون خروجی پیوند می‌زند و به این معنی است که آستانه نرون می‌تواند به صفر تنظیم شود. ما می‌توانیم ازطریق این شبکه کار کنیم و چهار ترکیب ممکن از

را استفاده کنیم و متقاعد شویم که شبکه به درستی عمل می‌کند. Y,X

فواید این روش این است که همیشه می‌توانیم مقدار آستانه را برای نرون به صفر تنظیم کنیم و از یک دیدگاه محاسباتی وقتی فراگیری شبکه موجود است ما فقط

مجبوریم اوزان را به روزرسانی کنیم نه جفت اوزان و آستانه.

شبکه‌های ساده‌ای که ما در نظر گرفتیم فقط نرون‌های ورودی و خروجی دارند و یک شبکه یک لایه‌ای (نرونهای ورودی به طور نرمال از یک لایه به عنوان وسیله‌های دریافت داده در یک شبکه در نظر گرفته نشده‌اند) در نظر گرفته شده است.

همچنین در شبکه‌هایی که ما در نظر گرفته‌ایم داده فقط در یک جهت حرکت می‌کند (از نرون‌های ورودی به نرون‌های) شناخته شده است. انواع بسیاری شبکه وجود Feedforward Network خروجی)که این رابطه به عنوان شبکه (

دارد.

) دو لایه‌ای در شکل ۱۰-۹ نشان داده شده است.Feedforward Networkیک مثال از شبکه (

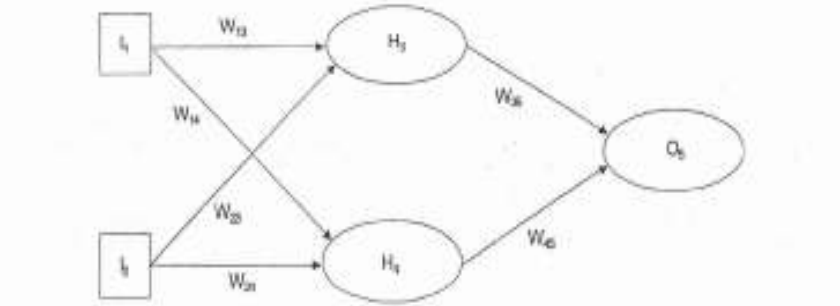


Figure 9.10 Two-layer, feed-forward network.

Perceptron ۱-۹-۹

یک لایه ای استفاده می شود .Feedforward Network این نام اکنون به عنوان یک واژه مترادف برای

Perceptron آنها اولین بار در سال ۱۹۵۰ مطالعه شدند و اگرچه معماری دیگر شبکه ها شناخته شده بودند ولی

تنها شبکه ای بود که به قابلیت فراگیری شناخته شده بود و بنابراین بیشتر تحقیقات در

.

آن زمان بر روی آن متمرکز شده بود .
شکل ۱۱-۹ مثالی از آن را نشان می دهد .

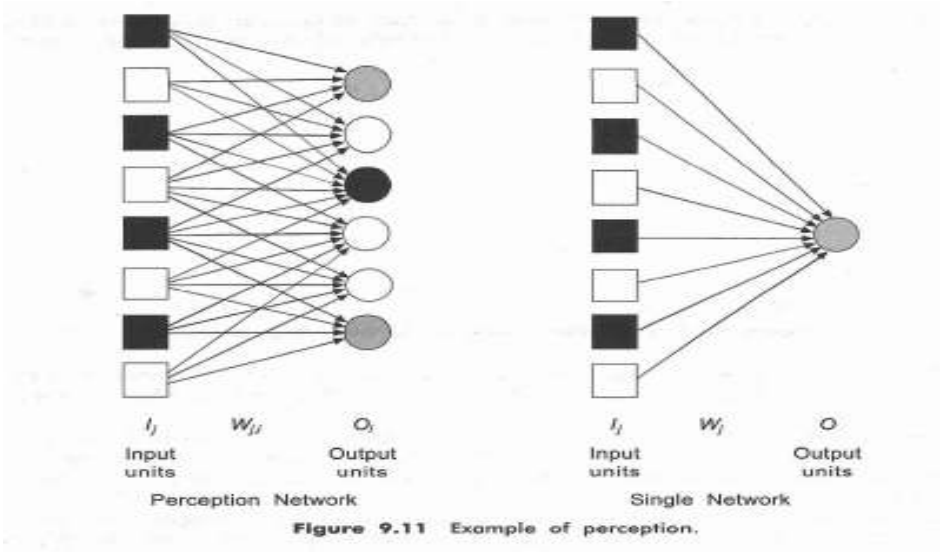


Figure 9.11 Example of perception.

ما میتوانیم در سمت چپ شکل ببینیم که وزن فقط روی یکی از خروجی ها اثر می گذارد ، این به این معنی است که ما می توانیم مطالعاتمان را آسانتر و فقط با در نظر گرفتن شبکه ها با یک خروجی منفرد انجام دهیم .(مانند شبکه ای که در سمت راست شکل ۱۱-۹ نشان داده شده است)
وقتی که فقط یک خروجی داشته باشیم می توانیم نشانگذاری هایمان را ساده تر سازیم .

مشخص شده است .بنا براینj W با j مشخص شده است ووزن نرون داخلی O بنابراین نرون خارجی با
تابع فعالسازی مطابق زیر است :

$$O = \text{Step}, C\ W_{jIj}$$

را ارائه دهد .Not، Or،And می تواند توابع منطقی perceptron
یک لایه ای) میتواند هیچ تابع بولینی را ارائه دهد ؟متاسفانه این مورد را

Feedforward Networkولی آیا آن (

شامل نمی شود .

برای اینکه علت آنرا بدانید دو جدول زیر را در نظر بگیرید (جدول ۵-۹ و ٦-۹)

Table 9.5 AND truth table

$Input_1$	$Input_2$	$Output$
0	0	0
0	1	0
1	0	0
1	1	1

Table 9.6 XOR truth table

$Input_1$	$Input_2$	$Output$
0	0	0
0	1	1
1	0	1
1	1	0

ما می توانیم

این دو جدول درستی را به صورت گرافیکی همانطور که در شکل ۱۲-۹ نشان داده شده است نشان دهیم (جایی که یک دایره توپر را نشان می دهد خروجی ۱ و دایره توخالی خروجی ۰ است)

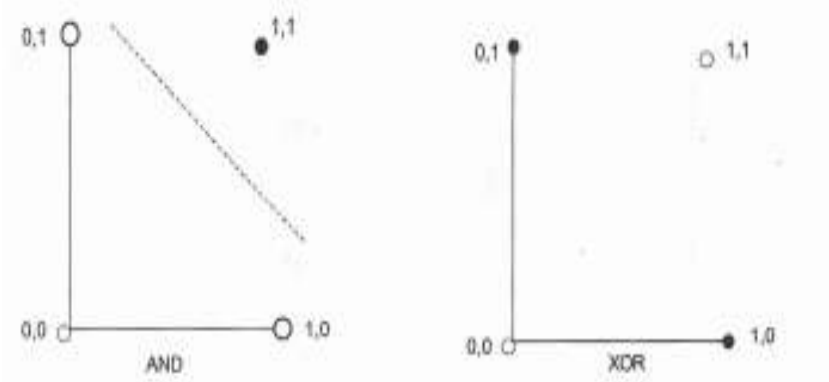


Figure 9.12 Dimensional view of the linear separability.

.

نگاه کنید خواهید دید که ما می توانیم یکی از صفرها را با یک خط جدا کنیم در حالی که این And اگر شما به گراف
ممکن نیست .Xorبرای
جدا شدنی خطی نامیده می شود .And تابع مانند
می توانست فقط توابع جداشدنی خطی را Perceptron مطرح شد که Minskyو Papert آن دلیلی بود که بوسیله
فرا گیرد که منجر به نزول تحقیقات در زمینه شبکه های عصبی شد تا اواسط ۱۹۸۰ وقتی که ثابت شد که معماری دیگر شبکه ها می توانند این نوع از توابع را یاد بگیرند .
است ولی ارزش مطالعهPerceptron اگرچه فقط قادر بودن به یادگیری توابع جدا کننده خطی از مهمترین عیب های
کردن را دارد چون به طور نسبی ساده است و می تواند به فراهم کردن یک چارچوب برای دیگر معماری ها کمک کند
ورودی که به ما مسئله n و می توانیم And فقط محدود شده به دو ورودی نیست مثل perceptron فهمیده شد که
بعدی می دهد داشته باشیم .n
است می توانیم تفکیک پذیری خطی از مسئله را که در شکل ۱۳-۹ نشان داده شده است تجسم کنیم .n=3 وقتی که
ها بزرگتر از ۳ است تجسم کردن آن سخت است .n زمانی که یکی از

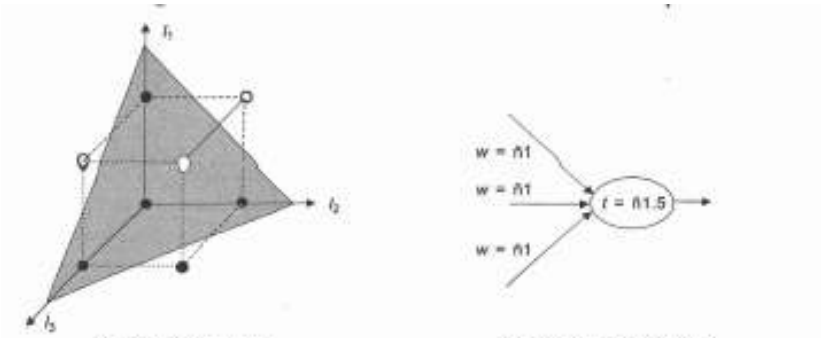


Figure 9.13 Dimensional view of the linear separability.

۹-۹-۲ فراگیری توابع تفکیک پذیر خطی

(با فقط دو ورودی)ما میتوانیم به آسانی تصمیم بگیریم که چه اوزانی استفاده شود تا به ما And با یک تابع مانند

خروجی مورد نیاز از نرون را بدهد .

ولی با بیشتر توابع پیچیده (مثل آنهایی که بیش از دو ورودی دارند) ممکن است تصمیم گیری روی اوزان در ست خیلی آسان نباشد .

بنابراین ما می خواهیم شبکه های عصبیمان فرا بگیرند بنابراین آن میتواند با مجموعه ای از اوزان بیاید .

این جنبه از شبکه های عصبی برای توابع ساده (۱ یا ۲ ورودی) را در نظر خواهیم گرفت .

ما می توانیم آشکارا مسئله را افزایش دهیم تا مسائل پیچیده بیشتری را تطبیق دهیم مشروط بر اینکه مسئله ، جداپذیر خطی باشد .

با ورودی نیرون که در شکل ۱٤-۹ نشان داده شده است را در نظر بگیرید .And جدول درستی در جدول ۷-۹ و شبکه

.

Table 9.7 AND truth table

$Input_1$	$Input_2$	$Output$
0	0	0
0	1	0
1	0	0
1	1	1

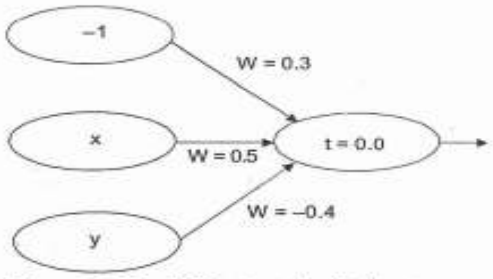


Figure 9.14 AND network with input neuron.

در واقع تمام چیزی که ما انجام داده ایم مجموعه ای از اوزان که مقادیرشان بطور تصادفی بین ۰/۵ – و ۰/۵ است را تنظیم کرده ایم .

با اعمال تابع فعالسازی برای هر یک از ٤ ورودی ممکن به این نرون ، جدول درستی زیر را به ما می دهد .جدول درستی می تواند به صورت گرافیکی ارائه شود که در شکل ۱۵–۹ نشان داده شده است .

Table 9.8 ANDX truth table

$Input_1$	$Input_2$	$Output$
0	0	0
0	1	0
1	0	1
1	1	0

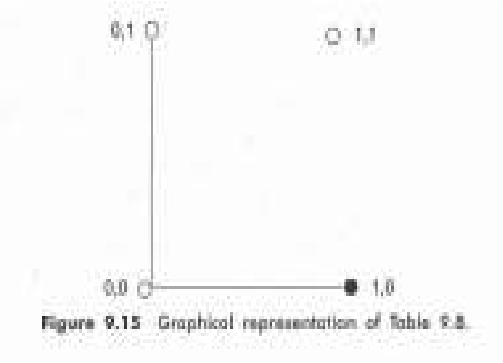


Figure 9.15 Graphical representation of Table 9.8.

را نمایش نمی دهد بنابر این ما نیاز داریم که اوزان را تنظیم کنیم بنابراین تابع را به درستی And شبکه ، آشکارا تابع

فرا می گیرد .

قبل از حل الگوریتم اجازه دهید بعضی اصطلاحات علمی را تعریف کنیم :

شامل Epoch٤، یک And : ارائه مجموعه آموزش کامل به شبکه های عصبی است .در هر مورد از تابع Epoch

)[1,1][0,1],[1,0],[0,0] مجموعه از ورودی ها است که به شبکه ارائه شده است مانند (

: وقتی ما شبکه ای را آموزش می دهیم فقط با مقدار ورودی هایش نمایش نمی دهیم بلکه با یک Training Value

نمایش دهیم ، مقدار آموزش And برای تابع [1,1] مقدار که ما نیاز داریم شبکه بسازد . برای مثال ،اگر ما شبکه را با

۱ خواهد شد .

: مقدار آن برابربا مقداری است که بین مقدار خروجی شبکه و مقدار آموزش اختلاف است .برای مثال Error,Err

.

.Err=-1 اگر ما خروجی ۰ نیاز داریم و خروجی آن ۱ است پس

:خروجی هایی است که به نیرون ارائه می شود .I_j

: وزن از نیرون ورودی به به نیرون های خروجیW_j

: نرخ فراگیری .چگونه همگرا بودن شبکه را دیکته می کند و با آزمایش تنظیم شده است و عموما ۰و۱ است . LR

Perceptronالگوریتم آموزشی

While
epoch produces an error
Present network with next inputs from epoch
Err = T - O
If
Err < > O then

را افزایش دهیم و اگر منفی باشد باید آنرا کاهش دهیم .هر ورودی O مثبت باشد ما نیاز داریم تا error نکته : اگر

را افزایش می دهد و اگر منفی W_j،O مثبت باشد ،افزایش درI_j را به ورودی کل شرکت میدهد بنابراین اگر$W_j I_j$
باشد کاهش خواهد داد .

فراگیری آن معمولا قانون فراگیری دلتا نامیده می شود.

$$Wj = Wj + LR " Ij " Err.$$

End If
End While
Perceptron

And را فرا بگیریم . اجازه دهید به یک مثال نگاه کنیم. مقدار اولیه وزنها ۰/۳، ۰/۵ و ۰/٤- است و ما سعی میکنیم تابع

هیچ اتفاقی برای اوزان نخواهد افتاد (به علتepoch ارائه دهیم از اولین [0,0] اگر ما شبکه را با اولین جفت آموزشی
. step) است که نتیجه اش در شبکه ایجاد ۰ است (با خاصیت تابع [0,1] ضرب در ۰).جفت آموزشی بعدی
ندارد و بموجب آن فراگیری ادامه می یابد . errorصفر خروجی مورد نیاز است که هیچ
است این بدین Error=-1 یک خروجی ۱ می سازد ، خروجی خواسته شده ۰ است بنابراین [1,0] جفت آموزش بعدی
معنی است که ما باید اوزان را تنظیم کنیم .
)LR=0.1 این عملیات به صورت زیر انجام می شود :(

$$W_0 = 0.3 + 0.1 * -1 * -1 = 0.4$$
$$W_1 = 0.5 + 0.1 * 1 * -1 = 0.4$$
$$W_2 = -0.4 + 0.1 * 0 * -1 = -.04$$

بنابر این اوزان جدید ۰/٤ ،۰/٤ ،۰/٤ – است .

را به شبکه اعمال می کنیم که این یک اشتباه می سازد و مقاذیر وزن جدید ۰/۳ ، ۰/۵ [1,1] سرانجام ما ورودی
۳/.– خواهد بود .
می سازد (در واقعیت دو مورد) ما نیاز داریم که آموزش را ادامه دهیم و شبکه را error یک Epoch در این نمایش
دیگر نمایش می دهیم .Epoch با
دیگری نمی سازد . error نمایش می دهد که هیچEpoch آموزش ادامه می یابد تا هنگامی که
در نظر بگیریم (اوزان = ۳/. ،۵/. ، ۳/. –) ما می دانیم که اوزان Epoch اگر ما حالت های شبکه را در انتهای اولین

.

صحیح باشند Epoch یک اشتباه می سازد(در واقع اوزان ممکن هستند که در پایان Epoch اشتباه هستند هنگامی که

دیگری ارائه دهیم که آنرا نمایش دهیم)Epoch ولی ما نیاز داریم که

همچنین می توانیم یک گراف که حالت جاری شبکه را نشان می دهد بسازیم .

دست یابیم که در شکل ۱۶–۹ نمایش داده شده است .And سعی می کنیم که به تابع خط تفکیک پذیری خطی می تواند با بکارگیری اوزان یک خط روی گراف بکشد .دو نکته می تواند روی محور

می تواند یافت شود به عنوان :I_2 ,I_1

II point = *WdWl*
I2 point = *WdW2*

آستانه را نمایش می دهد بنابراین ما آستانه را به اوزان تقسیم می کنیم .W_0

$I_2 = -1$ $I_1 = 0.6$

قرار نگرفته است و خط صفر And این خط در شکل ۱۶–۹ نشان داده شده است و این خط در محل درستی برای تابع

ها و یک ها را به درستی جدا نکرده است .

دریافت نکنیم وزن ها ۰/۴ ، ۰/۴ ، ۱/۰ خواهد Epoch اگر ما آموزش را ادامه دهیم تا هنگامی که هیچ خطایی از تمام

بود .

را بدست می آوریم(دوباره خط تقریبا در مکان صحیح Andاگر ما این اوزان را روی گراف رسم کنیم ، گراف تابع

است)در شکل ۱۷–۹

و ما می توانیم چگونگی تفکیک پذیری خطی را در یک مکان معتبر ببینیم و میتواند با چک کردن شبکه عصبی که خروجی های صحیحی می سازذ تایید شود .

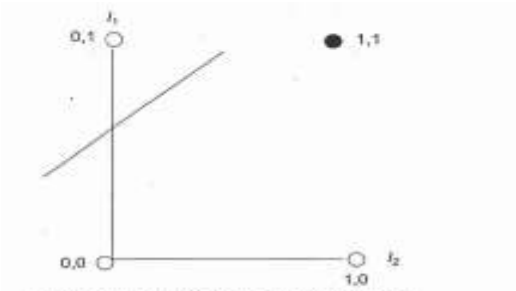


Figure 9.16 AND function representation.

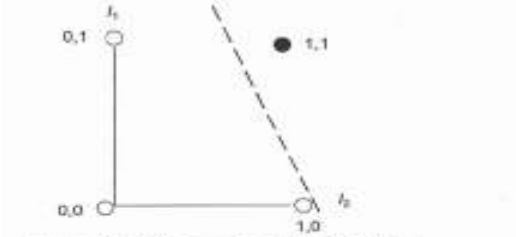


Figure 9.17 The graph of AND function.

۹-۱۰ مقایسه شبکه های عصبی با سیستم مبنی بر قاعده

شبکه های عصبی ، ابزار های انعطاف پذیری در یک محیط پویا هستند .شبکه های عصبی ظرفیت اینکه سریع فرا بگیرند و سریع تغییر کنند را دارند .

مدل می تواند شرایط محیطی و مقادیر داده و تغییر خروجی ها را سریعا فرا بگیرد و خود را وفق دهد . سیستم های مبنی بر قاعده به موقعیت های تعریف شده ای محدود شده اند در حالیکه شبکه های عصبی در وفق دادن خود به اطلاعات تغییر یافته دائمی ، عالی هستند .

۹-۱۱ مقایسه شبکه های عصبی و سیستم خبره

سیستم های خبره برنامه های کامپیوتری مبنی بر قاعده هستند .آنها تصمیمات را با بکار بردن یک مجموعه از قوانین که دانش افراد (یا متخصصین) را منعکس می کنند ، می

.

گیرند .

سیستم های خبره برای کاربردهایی با جداساز منطقی بین فاکتورهای تاثیر گذار تصمیم ، مناسب هستند .از وقتی که آنها قواعد را صریح ساخته اند ، فهمیدن اینکه چگونه تصمیمات گرفته می شود آسان است ولی ساختن سیستمهای خبره مشکل است زیرا قواعد همیشه وجود ندارند و بدست آوردن اطلاعات "خبره " نیز دشوار است .

معمولا متخصصین وجود ندارند یا اینکه آنها نمی توانند به طور دقیق با هم ارتباط برقرار کنند که چگونه آنها فاکتورهای مختلف را که به یک فرجام می انجامد اندازه می گیرند .

و یکی از سیستم های خبره ای که برنامه نویسی شده است نمی تواند به آسانی با شرایط تغییر یافته ، وفق داده شود، قواعد همان قواعد هستند تا هنگامی که یک سیستم جدید ساخته شود .

۹-۱۲ مزایای محاسبات عصبی

مزایای مختلفی در بکار بردن شبکه های عصبی وجود دارد که تحلیلگر می فهمد .

الگو شناسی یکی از تکنیک های قوی برای به کنترل درآوردن اطلاعات و داده ها و عمومیت بخشیدن به آنهاست .شبکه های عصبی فرا می گیرند که چگونه الگوهایی را که در مجموعه داده ها وجود دارد تشخیص دهند .

سیستم بیشتر از طریق فراگیری توسعه می یابد تا برنامه نویسی .برنامه نویسی وقت بیشتری را از تحلیلگر می گیرد و نیاز دارد تا تحلیلگر یک الگوی دقیقی از رفتار مدل را تعریف کند . شبکه های عصبی به خودشان الگوها را می آموزند و به تحلیلگر، آزادی برای کارهای جالب را می دهند .

شبکه های عصبی در تغییرات مهم انعطاف پذیر هستند .سیستم مبنی بر قاعده یا سیستم برنامه نویسی شده به موقعیتهایی که برای آنها طراحی شده ، محدود شده اند و وقتی شرایطی تغییر کند آنها هیچ اعتباری نخواهند داشت .

اگر چه شبکه های عصبی مدتی زمان نیاز دارند تا یک تغییر شدید ناگهانی را فرا

بگیرند ولی آنها در وفق دادن خود با اطلاعات تغییر یافته دائمی عالی هستند .
شبکه های عصبی می توانند مدل های مفیدی را بسازند جایی که بیشتر روش ها ی معمول شکست خورده اند چون شبکه های عصبی می توانند خیلی از فعل و انفعالات پیچیده را مدیریت کنند آنها می توانند داده هایی را که خیلی مشکل برای مدل کردن به روش سنتی هستند را به آسانی مدل کنند مانند برنامه نویسی منطقی یا آمار استنتاجی .

راندمان شبکه های عصبی حداقل به خوبی مدل کردن های آماری کلاسیک است و برای بیشتر مسائل بهتر هستند . شبکه های عصبی مدل هایی را می سازند که بیشتر انعطاف پذیری ساختاری در زمان کمتر قابل توجهتری دارند .
شبکه های عصبی می توانند با سخت افزار های کامپیوتری خوب کار کنند .
اگر چه شبکه های عصبی شدیدا محاسباتی هستند ، روتین ها به نقطه ای بهینه شده اند که آنها می توانند در یک زمان منطقی در کامپیوتر های شخصی اجرا شوند .
آنها به سوپر کامپیوتر هایی که تحقیقات شبکه عصبی در آن انجام می شد نیاز ندارند .

۹-۱۳ محدودیت محاسبات عصبی

بعضی محدودیت هایی برای محاسبات عصبی وجود دارد . محدودیت کلیدی ، ناتوانایی شبکه ها برای توضیح مدل هایی است که در یک روش مفیدی ساخته شده است .تحلیلگران معمولا می خواهند بدانند که چرا مدل ها اینگونه رفتار می کنند . شبکه های عصبی جوابهای بهتری بدست می آورند ولی توضیح اینکه چگونه آنرا بدست آورده اند برای آنها مشکل است .
این دلیلی است که ارتباطات عصبی ، ابزارهای زیادی را برای اکتشاف خروجی و انتخاب عملیاتی زیادی را روی این ابزارها که مدل را می سازنند ، فراهم می آورد . با آزمایش پارامتر های مختلف و اکتشاف کامل نتایج هم در متن و هم گرافیک ، کاربران ارتباطات عصبی می توانند به درک بیشتری از رفتار مدل ها و اطمینان بیشتری از نتایج

.

برسند .

چند مورد محدودیت های دیگری وجود دارد که باید فهمیده شود . اول ، استخراج قواعد از شبکه های عصبی دشوار است بعضی وقتها این برای افرادی مهم است که مجبورند جوابهایشان را بدیگران توضیح دهند و برای آنهایی که با هوش مصنوعی به ویژه سیستم های خبره که مبنی بر قاعده هستند در گیر شده اند .

در بیشتر روش های تحلیلی شما نمی توانید فقط یک داده را در شبکه عصبی قرار دهید و سپس یک جواب خوب بگیرید ، بلکه شما مجبورید زمانی را برای درک مسئله یا خروجی که شما سعی می کنید پیش بینی کنید صرف کنید و شما باید مطمئن باشید که داده هایی که استفاده کردید که به سیستم آموزش دهید مناسب هستند و به روشی اندازه گیری شده اند که رفتار فاکتور ها را انعکاس می دهند .

اگر داده ها حاکی از مسئله نباشند محاسبات عصبی نمی تواند نتیجه خوبی را تولید کند . این یک موقعیت کلاسیک است جایی که زباله ورودی ، زباله خروجی تولید خواهد کرد .

سرانجام ، آموزش یک مدل از مجموعه ای از داده های پیچیده زمان خواهد برد . تکنیک های عصبی به شدت کامپیوتری هستند و سرعت کا مپیوترهای شخصی یا ماشین بدون پردازشگر های ریاضی کم خواهد شد و

این مهم است که بیاد آورید که زمان سراسری برای بدست آوردن نتیجه می تواند سریعتر از راه حل های تحلیلی داده ای باشد حتی وقتی که سیستم زمانی طولانی تری را می گیرد که آموزش ببیند .

سرعت پردازش ، تنها فاکتور در راندمان نیست و شبکه های عصبی به برنامه نویسی زمانی و اشکال زدایی یا تست گمانها که دیگر روش های تحلیلی انجام می دهند نیاز ندارند .

فصل دهم

استدلال بر مبنای نمونه

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

- آشنایی با استدلال مبنی بر مورد(CBR)
- درک پایگاه مورد و بازیابی مورد
- آشنایی با مفاهیم استفاده دوباره و بازیابی مورد
- شناخت شاخص
- آشنایی با برخی از کاربردهای CBR
- درک فواید استفاده از استدلال مبنی بر مورد

۱۰-۱ مقدمه

سیستم استدلال مبنی بر مورد می‌تواند به عنوان یک نوع ویژه از سیستم مبنی بر دانش در نظر گرفته شود.

اصولهای استدلال مبنی بر مورد در هوش مصنوعی یافته شده است با کارهای Roger Schank در حافظه پویا و نقش مرکزی که به یاد آورنده موقعیتهای اخیر (حادثه ضمنی – موارد) و الگوهای موقعیتی Script,Mop در حل مسئله‌ها و فراگیری هستند. سیستمهای مبنی بر مورد با موارد یا اپیزودهایی سر و کار دارند که به عنوان مفاهیم یا فرم عمومی نمایش داده می‌شوند. آنها به ویژه در دانشهای مختلفی استفاده

می شوند و در نتیجه نمایش‌های متفاوت و نیازهای مدیریتی متفاوتی دارند.
کارایی سیستم استدلال مبنی بر مورد، مستقیما به کیفیت و کمیت دانشهای رمز شده بستگی دارد، از طرف دیگر اگر دانش کافی نباشد قابلیت حل مسئله محدود می‌شود و می‌تواند به خاطر مقدار دانشهای غیر مدیریت شده غیرقابل استفاده شود.
سیستمهای استدلال مبنی بر مورد می‌توانند همچنین به عنوان فراگیری اطلاعات به هم پیوسته و سیستمهای یادگیری دیده شوند. فرا گیری دانش با انواع دیگر یادگیری، خصوصیات پایه‌ای از سیستم هوشمند هستند.
نمایش و مدیریت پایگاههای دانش مورد بزرگ کارهای جزئی و ناچیز نیستند وتا کنون کاملا استادانه درست نشده‌اند. این کارها می‌توانند به عنوان یک پروسه از ارائه، تجسم وکاربرد دانش اپیزودیک که محتوای اپیزود مورد که شرح یک الگوریتم بازیابی است توصیف شوند .
ارائه و مدیریت برای استدلال مبنی بر مورد رقابت پذیر است زیرا خیلی از کاربردها برای پردازشهای مؤثر فضایی و زمانی پایگاههای موردی بزرگی نیاز دارند.حتی اگرچه ارائه و مدیریت موارد می‌توانند به طور یکسان مانند سیستم مبنی بر دانش دسته‌بندی شوند ولی تفاوت ویژه ای وجود دارد که نیاز است در نظر گرفته شود.
موارد اصلی برای پروسه های حل مسئله و ارائه این پروسه ها به وسیله اپیزودها مناسب هستند ، بنابراین آنها نه فقط در محیط‌های رسمی مانند سیستم‌های مبنی بر دانش سنتی بلکه در محیط های با رسمیت کمتر نیز مناسب هستند .
یک مورد می‌تواند یک طرح کامل راه حل را ارائه دهد که می‌تواند بازیابی شود ، وفق داده شود و از حذف در آینده برای تجزیه و ترکیب دوباره جلوگیری کند. در مقابل سیستمهای هوش مصنوعی قدیمی ، سیستم های مبنی بر مورد ، از تجارب گذشته در حل مسائل استفاده می کنند .
در واقع استدلال مبنی بر مورد فقط یکی از مجموعه نامهایی است که به این نوع از سیستم ها منتسب می شود که منجر به یک سری از اغتشاش ها می شود به ویژه وقتی که نام استدلال مبنی بر مورد ، هم به عنوان یک نام کلی برای چندین روش مختلف به خوبی یک روش تعریف می شود ،در بعضی حوزه ها آن می تواند به عنوان نامی برای استدلال مقایسه ای گفته شود .
تلاشهایی برای واضح سازی، اگر چه اغتشاش های نام های مربوط به استدلال مبنی بر

مورد را رفع نکرد ،صورت گرفت که در زیر آمده است :

* استدلال مبنی بر مثال : این نام از طبقه بندی دیدهای مختلف ، دید کلاسیک ، دید احتمالی و دید مثالی مشتق شده است .در دید مثالی ، یک مفهوم به طور بسط داده شده ای با مجموعه ای از مثالهایش تعریف می شود .روش استدلال مبنی بر مورد که تعاریف مفاهیم آموزشی را مرتب میکند (به عنوان مثال ،مسئله با بیشتر تحقیق ها در فراگیری ماشین مرتب می شد) بعضی وقتها به پایه های مثالی رجوع داده می شود .

* استدلال مبنی بر نمونه : آن یک تخصص از استدلال مبنی بر مثال به سمت یک روش استدلال مبنی بر مورد نحوی سطح بالا است .به منظور جبران کردن کمبود راهنمایی از دانش پیش زمینه عمومی ، به یک تعداد نسبی زیادی از نمونه ها نیاز است که آنرا به تعاریف مفاهیم نزدیک کند .ارائه این نمونه ها آسان است (به عنوان مثال نمودار مشخصه)وقتی که یک کانون اصلی موجود است که آموزش خودکار را بدون کاربر در یک حلقه مطالعه کند .اساسا ، این یک روش غیر عمومی برای مسائل فراگیری مفاهیم، که بوسیله روش های فراگیری ماشین استنتاجی یا کلاسیک مرتب می شونداست .

*استدلال مبنی بر حافظه : این روش بر یک مجموعه ای از موارد به عنوان حافظه بزرگ و استدلال به عنوان پروسه دسترسی و جستجو در این حافظه تاکید می کند . سازمان حافظه و دسترسی یک نقطه مورد توجه در روش مبنی بر مورد است.

بکارگیری روش پردازش موازی یکی از ویژگیهای این روش است واین روش را از دیگر روشها متمایز می سازد .

* استدلال مبنی بر مورد : اگر چه استدلال مبنی بر مورد به عنوان یک نام کلی در این بخش استفاده شده است ولی روش های استدلال مبنی بر مورد بعضی ویژگیهایی دارند که آنها را از دیگر روش هایی که در اینجا لیست شده است متمایز می سازد .

ابتدا، یک مورد معمولا فرض می کند که یک در جه از توانگری اطلاعاتی را دارد و یک پیچیدگی قطعی با سازماندهی داخلی دارد و این یک بردار مشخصه است که بعضی مقادیر و کلاسهای متناظر را نگه می دارد و آن چیزی نیست که ما به عنوان توضیح مورد فراخوانی می کنیم .

.

چیزی که ما به عنوان روش های مبنی برمورد استنتاج می‌کنیم ، خواص مشخصه دیگری هم دارند : آنها قادر هستند تا یک راه حل بازیابی شده را وفق یا تغییر دهند وقتی که در مفاهیم حل مسائل مختلف بکار برده میشوند .

روش های مبنی بر مورد همچنین از دانش پیش زمینه عمومی با وجود توانگریش ، درجات ارائه صحیح و نقش مطابق پروسه CBR استفاده میکند.روش های هسته ای سیستم های CBR ، از نظریه های روانشناسی شناختی، اقتباس های زیادی می کنند.

* استدلال مبنی بر قیاس : این نام بعضی وقتها به عنوان هم معنی سیستم مبنی بر مدل برای شرح دادن روش های مبنی بر مدل استفاده می شود.

اگر چه معمولا استفاده می شود تا روش هایی که مسائل جدید را بر اساس موارد گذشته از دامنه های مختلفی حل می کنند توصیف کند ، هنگامی که روش های مبنی بر مورد روی استراتژی شاخص گذاری و مطابقت برای موارد تک دامنه ای متمرکز می شوند .

۲-۱۰ پروسهCBR

در واژگان CBR ، واژهCASE معمولا یک موقعیت مسئله را مشخص می کند . یک موقعیت تجربه شده قبلی ، که در روشی فراگرفته می شود و می تواند برای حل مسائل آینده بکار رود به عنوان مورد گذشته ، مورد قبلی ، مورد ذخیره شده یا مورد حفظ شده رجوع داده می شود .متناظراً یک مورد جدید یا مورد حل نشده ، توضیح مسئله جدیدی است که باید حل شود .

استدلال مبنی بر مورد ، یک پروسه چرخه ای و یکپارچه از حل مسئله و فراگیری از تجارب و حل یک مسئله جدید است .

به طور کلی ، چرخه استدلال مبنی بر مورد می تواند در چهار پروسه زیر شرح داده شود :

الف) بازیابی مشابه ترین موردها یا موارد

ب) دوباره استفاده کردن اطلاعات و دانشها در آن مورد برای حل مسئله

پ) اصلاح کردن راه حل پیشنهاد شده

ت) حفظ بخشهایی از تجارب که برای حل مسائل آینده مفید است

چرخهCBR به صورت دیاگرامی در شکل ۱-۱۰ تصویر شده است :

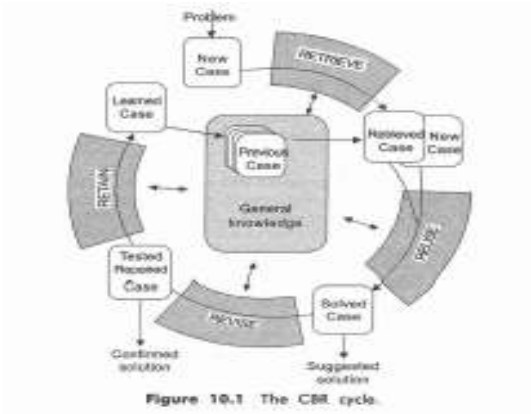


Figure 10.1 The CBR cycle.

توضیحات آغازی یک مسئله یک مورد جدید را تعریف می کند و این مورد جدید برای بازیابی یک مورد از مجموعه موارد قبلی استفاده می شود ، این مورد بازیابی شده با مورد جدید از طریق اصلاح مورد حل شده به عنوان مثال ، راه حل ارائه شده ترکیب می شود .از طریق پروسه اصلاح ، این راه حل برای موفقیت تست می شود و اگر رد شود اصلاح می شود .

در طی پروسه نگهداری ، تجربه های مفید برای استفاده های آینده نگهداری می شوند و استدلال مبنی بر مورد با موارد فراگرفته جدید و یا با اصلاح یا تغییر موارد موجود به روز رسانی می شود .

دانش عمومی ، یعنی دانش مستقل اصلی پروسهCBR را حمایت می کند .

۱۰-۲-۱ پایگاه مورد

پایگاه مورد در سیستم های CBR نماینده مجموعه ای از موارد است .یک مجموعه اهداف و زیر اهداف که از هر دوی موفقیت ها و شکستهای تلاشها در رسیدن به آن اهداف را شامل می شود .

یک مورد ، قطعه ای از دانش است که یک تجربه از مفهومی ویژه را بیان میکند .نوعا یک مورد ، شامل :

۱) یک جمله مسئله که شامل اطلاعات مفهومی است (به عنوان مثال ثابتها و پیش شرطها)

۲)حل یک مسئله
۳)خروجی اعمال یک راه حل(به عنوان مثال پس شرطها)
٤) هر واقعیت اصول واستدلال پشتیبانی که سیستم می داند چگونه از آن استفاده کند
۱- یک جمله مسئله در یک مورد ، توضیح مسئله یا موقعیت است که عمدتا شامل اهداف سیستم (انواع حل، قدرت تشخیص، ایجاد کردن ،طرح یا نوع تشریحی ، قدرت فهم ، توضیح دادن ، ارزیابی کردن) است ، محدودیت هایی به موقعیت و خصیصه های مسئله است .
CHEF یک سیستمCBR برای طراحی یک دستورالعمل چینی است که فقط از محدودیت ها به عنوان توزیع
مسئله استفاده می کند ، در مقابلPROTOS یک سیستمCBR ، برای گزارش تشخیص های نا مرتب است
CASEY یکCBR برای تشخیص سکته قلبی و HYPO که استدلالهایی را در حوزه شرعی ایجاد می کند و و فقط توضیح موقعیت ها را به عنوان توضیح مسئله نگه می دارد .

۲-راه حل مسئله در یک مورد می تواند شامل نماهای دیگری مثل انطباق و انتقاد باشد که در زیر آمده است :
الف) راه حل معنی تشخیص ، طرح ، توضیح
ب) مجموعه ای از مراحل استدلال برای حل مسئله استفاده شد
پ) مجموعه ای از توجیهات برای تصمیم هایی که در حل مسئله ساخته شده بودند
ت) راه حل های قابل پذیرش که انتخاب نشده بودند
ث) راه حل های غیر قابل پذیرشی که خارج از قاعده بودند
ج) انتظارات از اینکه چه نتیجه ای روی گسترش راه حل خواهند داشت
۳- خروجی ها ، شامل بازخوردهایی از دنیا و شرح بازخورد ها است . اگر انتظارات نقض کننده هستند یا راه حل ها شکست خورده اند ، خروجی ممکن است شامل توضیحاتی برای مسئله و روش های اصلاح آن باشد .
چندین قسمت خروجی مطابق زیر وجود دارد :
الف)خود خروجی

ب) آیا خروجی کامل کننده انتظارات است یا نقض کننده آن
پ)آیا خروجی موفقیت آمیز است یا شکست خورده
ت)توضیحات نقض کننده انتظارات و یا شکست ها
ث)استرا تژی اصلاح
ج) چه کاری برای جلوگیری از مشکل می تواند انجام شود ؟
چ)نشانگری به تلاش بعدی در راه حل

موارد می توانند در فرم های مختلفی مثل فریم ،شبکه های معنایی، قواعد به خاطر سپاری اسناد یا فرمت های مختلط بیان شوند .

موارد باید برای بازیابی موثر و استفاده دمباره سازماندهی شوند . روش های سازماندهی موارد متفاوتی وجود دارد .دو مدل با قدرت حافظه مورد ، مدل حافظه پویا و مدل مثالی رده بندی شده است .

۱۰-۲-۲ بازیابی مورد

پروسه بازیابی بهترین تطابق مورد قبلی از پایگاه مورد می‌دهد. زیر وظیفه‌هایش به عنوان خصایص شناسایی، تطابق اصلی جستجو و انتخاب ارجاع داده می‌شوند و به ترتیب اجرا می‌شود. وظیفه شناسایی اساساً با یک مجموعه از توصیفگرهای مسئله مربوطه می‌آید، هدف تطابق ، این است که یک مجموعه از مواردی را که شبیه مورد جدید هستند برگرداند. (به کار بردن آستانه همانندی بعضی از انواع) و انتخاب، روی مجموعه از موارد کار می‌کند و بهترین تطبیق را بر می‌گرداند.

بعضی روشهای مبنی بر مدل یک مدل قبلی را بوسیله همانندیهای سطحی و معنایی بین توصیفگرهای مسائل که به عنوان روش نقطه دانش، شناخته می شود ،بازیابی می کند و این در دامنه هایی قابل اجرا است که دانش عمومی برای بدست آوردن خیلی سخت یا غیر ممکن است.بعضی روشها که سعی می کنند تا موارد را بر اساس خصایصی که همانندیهای عمیقتر و معنایی تری دارند بازیابی کنند به عنوان روش متمرکز بر دانش ارجاع داده می شوند جایی که مفاهیم توصیفات مسئله برای دامنه هایی استفاده می شود که دانش دامنه ای عمومی قابل دسترس است .

مشخصه شناسایی

شامل توصیفگر های ورودی است ولی در بیشتر روش های استادانه مسئله با مفاهیمش

فهمیده می شود .
توصیفگر های ناشناخته ممکن است نادیده بگیرند یا در خواست کنند که بوسیله کاربر توضیح داده شوند ، توصیفگر های مسئله ی دارای مشکل را پاک می کند که به دیگر خصایص مسئله وابسته رجوع کند و چک کند که آیا مقادیر خصیصه با مفهوم ، معنایی دارد و انتظارات دیگر خصایص را حاصل می کند .
توصیفگر های دیگر جز آنهایی که به عنوان ورودی معین شده اند ، ممکن است با استفاده کردن از یک مدل دانش عمومی یا بوسیله بازیابی توصیف مسئله یکسان از پایگاه مورد استنتاج شوندو خصایص آن مورد را به عنوان خصایص مورد انتظار استفاده کنند .

مطابقت اصلی

این کار یافتن یک مورد قبلی مطابقت خوب را شامل می شود .این وظیفه به زیر وظیفه هایی تقسیم می شود .الف)پروسه تطابق اصلی، که یک مجموعه از کاندید های پذیرفتنی را بازیابی می کند ب)یک پروسه دقیقتر از انتخاب بهترین بین زیر مچموعه ها .
یافتن یک مجموعه از موارد مطابقت، بوسیله استفاده از توصیفگرهای مسئله به عنوان شاخص هایی به کتابخانه مورد در یک روش مستقیم یا غیر مستقیم انجام می شود .
سه روش بازیابی یک مورد یا ممجموعه ای از موارد وجود دارد : ۱)بوسیله دنبال کردن نشانگر های شاخص مستقیم از خصایص مسئله ۲)بوسیله جستجو در ساختمان شاخص ۳)با جستجو در یک مدل از دانش حوزه ی عمومی .
موارد ممکن است فقط از خصایص ورودی یا همچنین از خصایصی که از ورودی استنتاج شده است بازیابی شود . مواردی که با همه خصایص ورودی مطابقت دارند ،البته ، کاندید های خوبی برای مطابقت هستند ولی به استراتژی مواردی که کسری از خصایص مسئله معین شده را مطابقت میدهند(ورودی یا ورودی ارجاع داده شده)که ممکن است بازیابی شوند بستگی دارد .
بعضی تست هایی برای ارتباط مورد بازیابی شده معمولا اجرا می شود ، بویژه اگر که مواردی در یک پایه ای از زیر مجموعه ای از خصایص بازیابی شده باشند .
پروسه تطابق آسان خواهد بود اگر ارائه مورد به طور صحیح انجام شده باشد که این یک مسئله برای واژگان شاخص گذاری است .

یک راه برای تعیین کردن درجه یکسانی نیاز است و چندین استاندارد یکسانی ، بر اساس سطح یکسانی مسئله و خصایص مورد اظهار شده است .
تخمین یکسانی ممکن است بیشتر بر دانش متمرکز باشد با سعی برای فهمیدن مسئله که عمیقتر است و اهداف و قید هایی را استفاده می کند .
انتخاب دیگر این است که توصیفگر، مسئله را طبق اهمیتش برای توصیف کردن مسئله بسنجیم .
در پروسه مقایسه ، بعضی وقتها خصایص اضافی از موقعیت احتیاج است که از آن مشتق شود ، که این یک مسئله تشخیص موقعیت است .
برای جستجوی کتابخانه مورد کلان در یک روش مؤثر ، پیدا کردن موارد متناسبی از یک الگوریتم بازیابی کار خواهد کرد .اینجا نیاز است که یک الگوریتم جستجوی صحیح به خوبی الگوریتم تطابق ، داده شود که به عنوان مسئله شاخص گذاری استنتاج می شود .

انتخاب

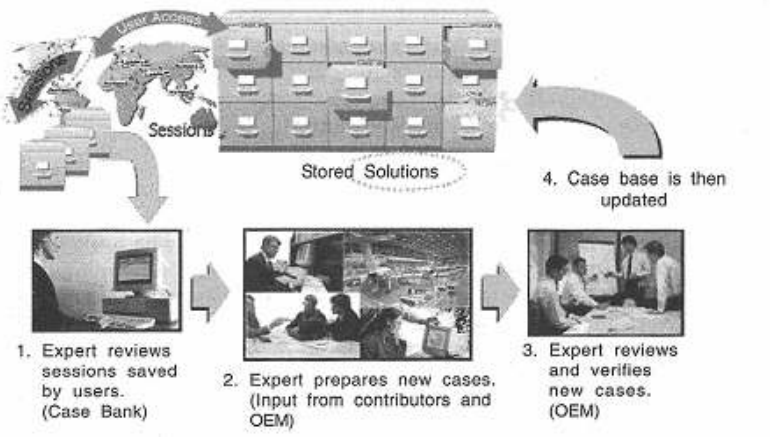
انتخاب ، بهترین تطبیق را از مجموعه موارد همانند پیدا می کند .بهترین مورد تطبیق معمولا بوسیله ارزیابی درجات تطبیق اصلی تعیین می شود .اگر تطبیق برگردانده شده به اندازه کافی قوی نباشد ،تلاشی انجام می شود تا تطابق بهتری را بوسیله پیوند های مختلفی که رابطه نزدیکتری به مورد دارند، پیدا شود.
پروسه انتخاب ، نتایج و انتظارات را از هر مورد مشتق شده بوجود می آورد و سعی می کند تا نتایج رت ارزیابی کند و انتظارات را تنظیم کند و این کار می تواند با استفاده از مدل سیستمی از دانش دامنه عمومی یا با سوالاتی از کاربرد درباره اطلاعات اضافی و تایید انجام شود .
موارد ، سرانجام طبق بعضی از الویت ها و شرایط رتبه بندی ، رتبه بندی می شوند .
روش های انتخاب متمرکز بر دانش توضیحاتی را بوجود می آورد که پروسه رتبه بندی را حمایت می کند و موردی که قویترین تفسیر را برای همانندی به مساله جدید دارد انتخاب می شود .

۱۰-۲-۳ استفاده دوباره

بکار گیری مورد بازیابی شده در گذشته از طریق مرحله بازیابی را ه حل را برای مساله

جاری ارائه می دهد ، که به دو جنبه رسیدگی می کند :
۱) تفاوتهای بین مورد جاری و مورد گذشته
۲) چه بخشی از مورد بازیابی شده می تواند به مورد جدید منتقل شود
مورد گذشته ممکن است دوباره استفاده شود یا با بکاربردن بعضی راه حلها برای مورد جدید به عنوان راه حلش یا با در نظر گرفتن تفاوتهای بین دو مورد و وفق دادن قسمت های مورد نیاز از مورد گذشته و بکار بستن آن در مورد جدید صورت می گیرد .
با در نظر گرفتن اهداف سیستم ، دو راه اصلی برای وفق دادن موارد گذشته با مورد جدید وجود دارد :
* دوباره استفاده کردن با تغییر شکل، دوباره استفاده کردن راه حل موردی در گذشته
*دوباره استفاده کردن اشتقاقی ، متدهای قبلی که راه حل را ایجاد کرده اند ، دوباره استفاده می کند .
در دوباره استفاده کردن به صورت تغییر شکل ، راه حل قبلی به طور مستقیم به عنوان راه حل مورد جدید استفاده نمی شود .
ولی دانش جدید به فرم عملگر(T) در راه حل قدیمی بکار می رود که راه حل خواسته شده برای مورد جدید را می دهد . این عملگر می تواند به کمک شاخص گذاری موارد حول تفاوتهایی که بین مورد بازیابی شده و مورد جدید کشف می شود ، سازماندهی می شود .
دوباره استفاده کردن به صورت تغییر شکل روی هم ارزی راه حل ها متمرکز می شود ، بنابراین به یک مدل وابسته قوی دامنه ای به فرم عملگر تغییر شکل به علاوه یک روش کنترل برای کاربرد عملگرها نیاز دارد .
دوباره استفاده کردن اشتقاقی روی یک روش حل متمرکز می شود . مورد بازیابی شده، اطلاعات را درباره روشی که برای حل مسئله ، شامل تنظیم عملگرهای استفاده شده ،زیر هدفهای بررسی شده ،راههای جستجوی شکست خورده و غیره استفاده می شود نگه می دارد .
دوباره استفاده کردن اشتقاقی ، دوباره روش بازیابی را به مورد جدید معرفی می کند و طرح قدیمی را به مفهوم جدید replay می کند .
وقتی دوباره استفاده کردن موفق انجام شد ، عملگرها و مسیرها ابتدا کاوش خواهند شد و از مسیرهای شکست خورده جلوگیری خواهد شد. زیر اهدافهای

جدید بر اساس مورد قدیمی پیگیری می شوند وزیر طرحهای قدیمی می توانند به صورت بازگشتی برای آنها بازیابی شوند .



۱۰-۲-٤ اصلاح کردن

وقتی که راه حل بوسیله مرحله استفاده کردن دوباره ایجاد شد ،برای راه حل صحیح نیازی نیست که مورد بازیابی شده اصلاح شود .این مرحله شامل دو کار است :

۱) راه حل ارائه شده برای مورد را ارزیابی کند ، اگر موفق بود ، راه حل را فرا بگیرد .

۲)در غیر اینصورت راه حل را با استفاده از دانش خاص دامنه ای اصلاح کند .

ارزیابی راه حل،به بازخوردهایی از دنیای واقعی با بکارگیری راه حل در محیط واقعی رسیدگی می کند و کاربردهایی از راه حل پیشنهاد شده به مسئله واقعی را شامل می شود .نتیجه حاصل از بکارگیری راه حل ممکن است که زمانی برای ظاهر شدن بگیرد که بستگی به نوع کاربرد دارد .

موردی که فرا گرفته شده است و در پایگاه مورد در یک دوره زمانی متوسط در دسترس است ، ولی آن مجبور است که به عنوان یک مورد بی ارزش نشانه گذاری شود .

راه حل ممکن است که در یک برنامه شبیه سازی بکار برده شود که قادر است راه حل صحیحی را ایجاد کند .

با پروسه ارزیابی ، مورد دوباره استفاده شده با راه حل ارائه شده /مورد انتظار مقایسه می شود .

بازخورد های پروسه ارزیابی ، توضیحاتی در باره تفاوتهای بین راه حل مورد انتظار و اتفاق واقعی ، تفاوت تنظیمات ، خروجیهای پروژه و مقایسه و رتبه بندی امکانهای پیشنهادی را ترکیب می کند .این بازخورد تطبیق های اضافی یا نیازمندیهای اصلاحی برای راه حل ارائه شده ، پیشنهاد می کند .

دومین وظیفه ی مرحله اصلاح ،اصلاح راه حل است و از بازخوردهای مرحله ارزیابی برای اصلاح راه حل در مسیری که شکستی اتفاق نیفتد استفاده می کند .

ماجول اصلاح سیستم CBR ، دارای دانش سببی عمومی و دامنه ای در باره چگونگی ناتوانی یا جبران دلایل خطاهایی
است که در دامنه اتفاق می افتد .

طرح اصلاح شده می تواند مستقیما نگهداری شود اگر درست باشد در غیر اینصورت باید دوباره ارزیابی و تصحیح شود .

ارزیابی و اصلاح تا هنگامی که راه حل صحیح بدست نیاید ، یک پروسه بازگشتی است .

۱۰-۲-۵ حفظ کردن

خروجی ارزیابی و اصلاح ممکن ، به فراگیری از موفقیت ها یا شکستها از راه حل ارائه شده کمک می کند .این پروسه ، انتخاب بخشهای خواسته شده از موردی که نگهداری میشود ،و اینکه به چه فرمی آنرا نگه دارد ، چگونه مورد را برای بازیابی های بعدی شاخص گذاری کندو چگونه مورد جدید را در ساختار حافظه قرار دهد و یکپارچه سازد .

استخراج کردن

اگر مورد با استفاده از مورد قبلی حل شده بود ،یک مورد جدید ممکن است ساخته شود یا مورد قدیمی ممکن است که تعمیم یافته باشد وشامل مورد موجود باشد .

اگر مسئله با روش های دیگری شامل سؤال از کاربر حل شده بود ، یک مورد جدید به طور کامل ساخته خواهد شد .

حالا سؤال پیش می آید که کدام قسمت از مورد نگهداری می شود ؟توصیفگر های مسئله ، راه حلهای مسئله ،توضیحات یا تنظیمات راه حل برای روش حل مسئله

،ساختارهایی هستند که برای فراگیری استخراج می شوند.اطلاعات درباره اصلاح و موارد شکست یا موارد مسئله کلی نیز ممکن است استخراج و نگهداری شوند که فهم مورد شکست خورده موجود را بهبود خواهد بخشید .

شاخص

اینجادرباره ی انواع شاخصها و ساختار شاخصها تصمیم گرفته خواهد شد .معمولا این یک مسئله فراگیری دانش است و باید به عنوان بخشی از تحلیل دانش دامنه ای و مرحله مدلسازی تجزیه و تحلیل شود .

یک راه حل جزئی برای یک مسئله این است که از همه خصیصه های ورودی به عنوان شاخص استفاده کند که روش مبنی بر نحو نامیده می شود .

در روش مبنی بر حافظه ، خصایص وابسته بوسیله تطابق در موازات همه موارد در پایگاه مورد و از بین بردن خصایصی که متعلق به مواردی با خصایص متداول کمتر با مورد مسئله هستند ، تعیین می شوند .

یکی کردن

در اینجا پایگاه دانش با مورد جدید بروز رسانی می شود و این کار با تغییراتی در شاخص گذاری موارد موجود انجام می شود .اهمیت شاخص برای یک مورد ویژه یا راه حل و تنظیمات ناشی از موفقیت ها یا شکستها از استفاده کردن مورد برای حل کردن مسئله ورودی است .

خصایص وابسته برای بازیابی موفق موارد ، مورد را قدرتمند می کند و آن برای خصایصی که منجر به موارد ناموفقی که بازیابی شده اند ، ضعیف است .بنابراین ساختار شاخص به میزان سازی و وفق سازی حافظه مورد برای استفاده اش کمک می کند .

در روش متمرکز بر دانش در CBR ، فراگیری ممکن است از طریق تعاملهایی با کاربر یا دیگر روشهای فراگیری ماشین مانند روش مبنی بر قاعده صورت گیرد.

CBR ابزار و روشهایی برای افزایش پیوسته پایگاه دانش فراهم می کند .یکپارچه سازی دانش از طریق مرور متخصص وطبقه بندی همه جلسات کامل شده تضمین می شود .

۱۰-۳ کاربردها

.

در این بخش ، ما بعضی از کاربردهای موفق و واقعی از استفاده هایCBR را شرح خواهیم داد .

۱-۳-۱۰ Clavier (ردیف جا انگشتی)

اولین کاربرد CBR بوسیله شرکت فضایی آمریکاییLockheed توسعه یافت .آن به تعیین یک روش مؤثر برای
بهبود اجزای هواپیما هنگام تولیدشان بود .چندین قطعه در یک زمان در یک دیگ (نوعی از فر بزرگ تحت فشار)
حرارت دیدند .مشخصه حرارتی دیگ ، کامل قابل فهم نیست و مدل دقیقی از رفتار آن وجود ندارد .
متصدیان به طراحی های موفق قبلی اعتماد می کنند که آنها را برای چیدمانی از مجموعه قسمتهای جدید با جستجوی موارد همانند راهنمایی می کند.
این به طور صریح در نمونه هایCBR گنجانیده می شود و همچنین یک راه حل با بکارگیری این تکنولوژی ایجاد می شود.
یک سیستمCBR ،CLAVIER ، بااهداف زیر توسعه یافت :

* استفاده دوباره پیکر بندیهای موفق قبلی
*کاهش بار کاری متخصصین
* امن کردن تخصص برای آینده
* آموزش کارکنان جدید

موارد طرحهای موفق شامل اطلاعات زیر است :

* بخشها و موقعیت وابسته شان در جدول
*جدولها و موقعیتهای وابسته شان در دیگ
* پارامترهایی مثل زمان شروع و پایان فشار و دما

از پایگاه موردCLAVIER ، نزدیکترین تطابق را می توان یافت و طرحهای جدیدی را به متصدیان پیشنهاد می کند.
این توافقات شامل :

* ایجاد طرحهای جدید از بخشی از طرحهای قدیمی
* حداقل کردن تعداد بخشهای مورد نیازی که در طرح وجود ندارد

* حداکثر کردن تعداد بخشهایی با الویت بالا در طرح
* حداکثر کردن تعداد بخشهای کلی در طرح

CLAVIER یک روش مفید برای انتقال تخصص ها بین متصدیان فراهم می آورد .تولید یک سیستم مبنی بر مدل سخت

خواهد بود مانند زمانی که متصدیان نمی توانند بگویند که چرا یک طرح ویژه موفق بوده است .سیستم همچنین توانایی برای فراگیری طرحهای جدید نشان می دهد برای یکی آن یک طرح موفق جدید که می تواند در یک پایگاه مورد ذخیره شود پیشنهاد کرده است .

CLAVIER از ۲۰ تا ۱۵۰ مورد در پایگاه داده اش رشد کرده است ودر ۹۰٪ موارد طرح موفقی را ایجاد می کند .

۱۰-۳-۲ CHEF طراح غذا

CHEF یک طراح مبنی بر مورد است (هومند،۱۹۹۰).آن اهدافی را که رسیدها می توانند بدست آورند می پذیرد و طرح

رسیدی(دستورالعمل) را که میتواند از اهداف معینی در حوزه آشپزی بدست آورد تولید میکند .

CHEF ، طراحی را به عنوان یک کار حافظه ای در یک حلقه استدلال ، اجرا و فراگیری در نظر می گیرد .

رویه طراحی آن ، تغییر شکل مستقیم وابستگی است .

الگوریتم

مرحله ۱: دستورالعمل قدیمی را بازیابی کنید که بسیاری از اهداف جدیدش را در صورت امکان با بکارگیری کتابخانه مورد تکمیل می کند .

مرحله۲: طرح قدیمی را متناسب با شرایط و موقعیت جدید اصلاح کنید که می‌تواند در دو مرحله انجام گیرد:

الف) دوباره معرفی کردن طرح قدیمی. آن یک نمونه ایجاد می‌کند که اشیاء جدید را برای طرحی که قبلاً استفاده شده است جانشین می‌کند. از استانداردهای یکسانی برای جانشینی اشیائی که در طرح قدیمی نقش اشیاء را داشتند استفاده می‌کند.

ب) انتقاد اشیاء یک منظوره را اعمال کنید تا طرح قدیمی را برای موقعیت جدید اصلاح کنید.

مرحله ۳: طرح یا برنامه را استفاده کنید و بازخوردهایی در مورد طرز کار آن برنامه را جمع آوری کنید. اگر آن برنامه به درستی عمل کرد، سپس آن را در حافظه برای استفاده دوباره ذخیره کنید. در غیر اینصورت یک توضیح سبی از این که چرا طرح به درستی عمل نکرده است ایجاد کند و آنرا به دانش عمومی‌اش که برای اصلاح طرح خراب استفاده می‌کند، فهرست بندی کند.

مرحله ٤: یک واحد شبیه ساز دارد که بازخورد طرح ارائه شده را پیش بینی می‌کند. آن طرح را با این عملیات که آیا طرح در دنیای بیرون نیز قابل اجرا است تست می‌کند و بازخوردها را با نتایج / اهداف مورد انتظار مقایسه می‌کند.

مرحله ٥: اگر طرح شکست خواهد خورد: الف) دلایل شکست برنامه به صورت روابط سببی بین اهداف، طرحها و مراحل برنامه توضیح داده شود که واژگان عمومی را برا توضیح موقعیت طراحی عمومی فراهم می‌آورد.

ب) استفاده کردن از رده بندی که بوسیله چف فراهم می شود ، کلاس صحیحی از توضیحات را به طرح جدید اختصاص می دهد .

پ) یک استراتژی اصلاح را با جستجوی در موارد یکسانی که در TOPذخیره شده‌اند انتخاب

کنید که یک روش اصلاح پیشنهاد می‌کند و صحیح ترین و مناسبترین اصلاح برای طرح را انتخاب می‌کند.

ت) طرح مناسب را اعمال کنید و طرح معیوب را اصلاح کنید.

برای جلوگیری از تکرار اشتباهات یکسان،CHEF ادراکش را از دنیا به روز رسانی می‌کند. برای انجام این کار، پیشگویی کننده‌ها شکستها را درک می‌کند و از آنها به عنوان شاخصهایی برای هشدار که ممکن است شکستهایی اتفاق بیفتد استفاده می‌کند. اگر شاخصهاییش را براساس تست توضیحات مشتق شده قبلی شکست‌ها، انتخاب کند، خصیصه‌های محیطی که مسئول این شکستها هستند را استخراج می‌کند.آن از شاخص‌ها برای سازماندهی هر دو مورد که یکی طرحهای شکست خورده به عنوان

طرح عمل کننده و دیگری طرحهای اصلاح شده به عنوان طرح راحل، برای مسائل استفاده می‌کند.

منابع دانش

CHEFمنابع دانش زیر را استفاده می کند :

۱) کتابخانه مورد ،شامل مواردی است که بر اساس شاخصهای ساختار موارد سازماندهی شده است .در مورد ، اسامی در یک ساختار فریمی ارائه می شوند و آنها با رابطه هایی با همدیگر در یک شبکه معنایی سازماندهی می شوند . موارد به روش شاخص ، طبق اهدافی که بدست می آورند شاخص گذاری می شوند .

۲) TOP (بسته های سازماندهی موضوعی) . این یک حافظه پویا برای داشتن اثر یادگیری در سیستم است .آن استراتژیهایی

را که در اصلاح طرحهای شکست خورده کمک می کنند نگه می دارد .

۳)Object Critics : کمک می کند تا طرح قدیمی را با شرایط جدید وفق دهیم .

٤)حافظه معنایی : تعاریف استفاده هایCHEF را نگه می دارد و برای تطابق جزئی و یافتن ارتباطات بین موارد بکار می رود .

خصایصCHEF

CHEF روش جدیدی برای طراحی دارد که از خیلی از مشکلات در سیستم های طراحی قدیمی جلوگیری می کند و از

طرحها دوباره استفاده می کند ، طرح ها را اصلاح می کند و از تجارب فرا می گیرد .در نتیجه فراگیری و مراحل پیش بینی

قبل از تمهید یک راه حل ایجاد می شود .

CHEF می تواند مشکلات را قبل از اینکه اتفاق بیفتد پیش بینی کند و همچنین طراحی را به عنوان یک وظیفه حافظه ای می بیند تا یک وظیفه ساختاری.

۳-۳-۱۰ انتخابگرA زمانبند مبنی بر مورد

انتخابگر یک برنامه ریزی نیروی انسانی باCBR Chaudhury,Bhattacharya (Gupta,1998, است این سیستم

مشخصات شغلی را می پذیرد و یک لیست کامل ، فراگیر (کمین) و نیازمندیهای سازگار را برای شغل بر اساس تجاربی از انواع شغلی یکسان که در یک سازمان در

گذشته دسته بندی شده اند پیدا می کند . برای مثال کاربر می تواند یک نوع کلی از پرس و جوها مثل " من یک مدیر می خواهم " یا "به من لیست نیازمندیهایی که مدیر یک گروه کوچک که تجربه برنامه نویسی شبکه دارد و زبان ژاپنی را می شناسد بده " سیستم مجبور است که به همه این نوع سؤالهای عمومی به خوبی سؤالهای ویژه با وضوح خوبی جواب دهد .

مراحل عملیات

ورودی: پرس و جوهای کاربر به عنوان فرم

خروجی: لیست نیازمندیهایی که به وسیله سیستم بوجود آمده است

مرحله۱: تمام مواردی را که طبقه‌بندی شده‌اند یا تابع طبقه‌بندی هستند و کاربر نام برده است بازیابی کنید.

مرحله۲ :از یک ماجول انتخابی که شایستگی را به همه موارد می‌دهد استفاده کنید.

مرحله۳: تمام مواردی را که آستانه شایستگی را می‌گذرانند ادغام کنید و برآیند موارد را به عنوان مورد هسته فراخوانی کنید.

مرحله٤: برای هر نیازمندی گمشده در مورد هسته، از یک ماجول جستجو استفاده کنید که برای موارد ممکن می‌سازد تا یک سری نیازمندی‌هایی که احتیاج است اضافه شود تا این نیازمندیها را ارضا کند درک کند و سپس آنها را به مورد هسته اضافه می‌کند.

مرحله۵: لیست نیازمندیهای ساخته شده را به ماجول پذیرنده بدهید که ببینید آیا نیازمندی ارضا نشده دیگری وجود دارد سپس اگر موردی وجود داشت مرحله ٤ را تکرار کنید.

مرحله٦: وقتی که همه نیازمندیها به لیست نیازمندیها اضافه شد سازگاری را چک کنید این مرحله نیازمندیها را پاک می‌کند.

مرحله۷: بعد از اینکه نیازمندی پایانی ساخته شد، موارد جدید را در زیر طبقه‌بندیها و زیر طبقه‌بندیها از جایی که جستجو شروع شده است ذخیره کنید.

ساختار مورد

مورد در یک روش طراحی شده است که برای استفاده دوباره برای ساختن موارد جدید مؤثر خواهد بود. و شامل شماره مورد، نام آن و جدول حاوی ردیفها است. هر ردیفی شامل:

* نیازمندیها به فرم جفت متغیرصفت

* برچسبها (زمینه‌های اختیاری) که مرجع (انقیاد) با نیازمندیهای دیگر را می‌دهد.
* لیست وقایع ممنوع (برخلاف برچسبها) نیازمندیهایی که نباید در مورد جاری ظاهر شوند.
* نشانه‌های خارجی- وابسته به نیازمندیهایی که نشان می‌دهد که این یک نیازمندی عمومی از کلاسی که مورد به آن تعلق دارد نیست و از طبقه‌بندی دیگری وارد شده است.

ساختار شاخص

موارد طبق نیازمندیهایی از طرف پهنا و عرض در دو مرحله سازماندهی می‌شوند:
الف) طبقه‌بندی
ب) زیر طبقه‌بندی

ماجول اصلاح

طرز کار ماجول اصلاح مطابق زیراست:
۱- با یک مورد خالی شروع می‌شود.
۲- طبقه‌بندی و زیر طبقه‌بندی تعیین شده را در پرس و جوها به دست می‌آورد.
۳- به همه موارد طبقه بندی و زیر طبقه‌بندی را انتخاب می‌کند.
٤- به هر موردی یک شایستگی می‌دهد و از ماجول انتخاب استفاده می‌کند.
۵- همه موارد را که از آستانه شایستگی عبور کرده‌اند ادغام می‌کند.
٦- موارد ساخته شده را به یک پذیرنده می‌دهد تا همه نیازمندیهای خواسته شده که در مورد گم شده‌اند را درک کند.
۷- برای هر نیازمندی، که اگر آن طبقه‌بندی یا زیرطبقه‌بندی است، همه موارد زیر آن ادغام می‌کند و به مورد جدید اضافه می‌کند.
۸- برای نیازمندیهای طبقه‌بندی و زیر طبقه‌بندی شده یک مورد جدیدی که شامل نیازمندیها است پیدا می‌کند و این نیازمندیها را با نیازمندیهای برچسب شده به مورد جدید اضافه می‌کند.
۹- مرحله ٦ را تکرار می‌کند تا هنگامی که همه نیازمندیها در مورد جواب نشان داده شود.
۱۰- سازگاری را چک می‌کند و از لیستهای ممنوع در مورد جواب استفاده می‌کند.

برای کار کردن، ماجول اصلاح از پارامترهای زیر استفاده می‌کند:
تحمل: تصمیم می‌گیرد که سختگیری تطابق بین پرس و جو و پایگاه موارد چگونه باشد.
شکست: شکست موارد باید در یک نیازمندی ویژه برای نیازمندی که انتخاب شده است ارائه شود.
جایزه و جریمه: اگر یک مورد نیازمندیهای داده شده تطابق دهد و پرچم خارجی‌اش ست شده باشد آن یک شایستگی برابر با جایزه دریافت می‌کند در غیر اینصورت اگر نیازمندیها تطابق نداشته باشند و پرچم خارجی ست شده باشد، یک جریمه می‌گیرد.
امتیاز مثبت: چند امتیاز، یک مورد برای هر یک از تطابقهای معمولی هنگامی که شایستگی به مورد داده می‌شود به دست خواهد آورد.

ماجول پذیرنده

این ماجول قابلیت پذیرش مورد کاندید شده، ساخته شده از پذیرش ماجول اصلاح برای تکامل و سازگاری را چک می‌کند.
عیوب موارد مطابق عیبهای زیر طبقه‌بندی می‌شوند:
۱- مورد خالی
۲- گمشدن نیازمندیهای ضروری
۳- گمشدن نیازمندی برچسب شده ضروری
٤- وجود نیازمندی خارجی غیر ضروری
۵- وجود نیازمندیها در یک لیست ممنوع

ماجول یکپارچه سازی

بعد از اصلاح مورد در یک یا پایگاه مورد به منظور غنی سازی آن ذخیره می شود .ماجول یکپارچه سازی پرچم های خارجی را ست می کند و مورد را زیر طبقه بندی و زیر طبقه بندی یکسان از پایگاه موردی که گرفته شده ، قرار می دهد .

۱۰-۳-٤ PROTOS مدل طبقه بندی بر اساس مثال برای گزارش تشخیص های بی نظم و دارای اختلال

روش مبنی بر مثال برای فراگیری طبقه بندی ابداعی استفاده می شود . روش استدلال مبنی بر مورد برای حوزه هایی با تئوری های ناکافی مناسب است ولی مشکلاتی در تصمیم گیری همانندی و شاخص گذاری مثالها دارد .

PROTOS یک برنامه کاربردی در حوزه شنوایی درمانگاهی ، بر مسائل غلبه می کند و روش مبنی بر مورد را به طبقه بندیبندیهایی پیاده سازی می کند.
روشPROTOS از کارهای فراگیری مفهومی معمولی در روشی به عنوان طبقه بندی که توضیح داده شده است متفاوت است
توصیفگر موارد غیر کامل خودش را با این کار وفق می دهد .

طبقه بندی

یک مورد با مجموعه ای از خصایصش شرح داده می شود . یک توصیفگر مورد ممکن است مشکلاتی را در دو روش داشته باشد :
الف) یک توصیفگر مورد ممکن است کامل نباشد در نتیجه ممکن است برخی از خصایصی را که در توصیفگرهای دیگر است نداشته باشد .
ب)خصایص با مواردی که شرح داده می شوند ممکن است مستقیما عضویت در یک دسته را نشان ندهد .
این دسته بندی با تصمیم گیری دسته هایی که از مراجع مبنی بر دانش استفاده می کنند انجام می شود .
علاوه بر دسته بندی ،PROTOS توضیحاتی را برای دسته بندی ارائه می دهد .این توضیحات استفاده می شوند تا رده بندی یک مورد را در یک روش ویژه تصدیق کنند و درجه یکسانی دو مورد را تصدیق کند .
توضیحات می توانند با سازماندهی و اصلاح دوباره بعضی جزئیات مسیرهای مربوط به استنتاج که منجر به دسته بندی می شودساخته شوند .
فراگیری دانش درPROTOS با نگهداری یک مثال از موارد آموزشی که دسته بندی شده و بوسیله متخصص شرح داده شده انجام می شود .
فراگیری مفهوم مبنی بر مثال:
مفاهیم با مجموعه ای از مثالها که با خصایصی که در زبان مورد شرح داده شده است ارائه می شوند .
رده بندی مورد جدید با جستجو برای مثالی که با مورد مطابقت دارد (روش مطابقت مستقیم) صورت می گیرد . توضیح طبقه بندی ، استدلالی را که برای این تطابقات استفاده می شود را نشان می دهد (لیستی از خصایص متعارف مورد و مثال).

فراگیری بوسیله تنظیم کردن دسته بندی ها صورت می گیرد ، بنابر این مورد به طور صحیح دسته بندی وتوضیح داده خواهد شد .مسئله در این روش مطابقت موارد است . توضیحات طبقه بندی و تنظیم دسته بندیهاکه فرا گرفته شده اند ، در PROTOS حل شده اند .

برای حل مسئله تطابق،PROTOS ، دانش مطابقت را استفاده می کند که یک فرم از نظریه دامنه است .

دانش مطابقت دو نوع دارد : یکی ارتباط بین مفاهیم و دیگری اهمیت خصایص است . با دانش مطابقت معین PROTOS می تواند خصایص مختلف را با یافتن یک مسیر از ارتباطات که آنها را به هم متصل می کند مطابقت دهد .

PROTOS به دانش مطابقت از موارد توضیح داده شده یا از توضیحات که بوسیله متخصصین فراهم شده است (در مورد
شکست ها) نیاز دارد .

PROTOS ، خصیصه هایی برای دسته بندی توضیحات نیاز دارد که هر خصیصه ای را به دسته بندی مورد ربط می دهد .

برای توضیح رابطه ، متخصصین مفاهیم و رابطه های جدیدی را معرفی می کنند .

متخصصین توضیحات مشخصه – مشخصه فراهم می آورند و PROTOS توضیحات طبقه بندی را کامل می کند .

رابطها در سه کلاس دسته بندی شده اند :

۱) روابط تعریفی برای مشخص کردن واقعیت های ثابت

۲) روابط تابعی/ سببی برای مشخص کردن مکانیسم های شناخته شده

۳) روابط همبستگی برای مشخص کردن دانش تجربی

هر رابطه ای می تواند با توصیف کننده هایی مثل همیشه ، معمولا ، بعضی وقتها ، گهگاه تحکیم شود .

PROTOS ، به طور ابتکاری اهمیت یک خصیصه به یک دسته را با تجزیه و تحلیل توضیح – دسته ارزیابی می کند .

اهمیت مشخصه ای درونی به صورت اعدادی بین ۰(نادرست) و ۱ (واقعی) ارائه می شود .

PROTOS اهمیت مشخصه ای که ممکن است بوسیله متخصص اصلاح شده باشد

ارزیابی می کند اگر آنها در یک توضیح
غیر قابل پذیرش یا غیر قابل دسته بندی نتیجه شوند .
PROTOS تکنیک مطابقت مبنی بر دانش را برای توضیح همانندی مورد جدید با یک مثال بکار می برد .طی این پروسه تصمیم می گیرد که کدام خصایص برای یک مطابقت موفق مهم هستند .
خصیصه مهم ناپیدا در توضیح مورد از خصایص موردی که از دانش مطابقت استفاده می کنندااستنتاج می شود . این جستجو از هر یک از خصایص مثالی تطبیق نیافته شروع می شود و از طریق شبکه ای از دانش مطابقت ، سلسله بندی می کند تا هنگامی که به یک خصیصه موردی یا رابطه ضمانت شده عمیق دست یابد .
کشف کننده ها برای یافتن قویترین توضیح بوسیله ارزیابی سهم بالقوه هر رابطه استفاده می شوند .
این تابع ، رابطه فردی و توصیف کننده ها و همچنین توضیحات ساخته شده را در نظر می گیرد .
PROTOS ، یک مورد را مانند یک مثال حفظ می کند اگر مورد با مثالهای موجود تفاوتهایی در روشهای مهمی داشته باشد آن نمی تواند با استفاده از دانش مطابقت موجود توضیح داده شود .

شاخص گذاری

برای انتخاب و سفارش مثالها برای مطابقت با مورد معین ،PROTOS سه نوع شاخص گذاری – یاذآوری کردن دانش و
وابسته به طرح اصلی بودن و تفاوتهای مثالی را استفاده می کند .
شاخص گذاری یادآوری کردن ، بوسیله خصایص موارد جدید دسته بندی می شوند .آن یادآوری کننده را به عنوان اشاره هایی به طبقه بندی موارد استفاده می کند .هر یاد آوری کننده یک پیوند قوی دارد که به منظور لیست کردن مثالهای کاندید استفاده می شود .
وقتی یک جستجو برای مثالی که با مورد جدید مطابقت دارد انجام شدPROTOS ، اول یادآوری کننده ها را برای
طبقه بندی جمع آوری می کند و طبقه بندی های مرتبط بوسیله جمع آوری یاد آوری

.

کننده های المثنی و با وارد کردن یاد آوری کننده ها از دسته بندی عمومی به زیر دسته بندی ترکیب می شوند .
فقط N تا از قویترین یاد آوری کننده ها برگشت داده می شوند.
PROTOS سپس چندین تا از مثالهای وابسته به طرح اصلی برای ارائه به هر دسته انتخاب می کند .
سرانجام PROTOS ، یاد آوری کننده ها را از خصایص مورد برای مثالهای ویژه ای جمع آوری می کند . نتیجه ، یک لیست مرتب شده از مثالهایی است که سعی می کنند با مورد جدید مطابقت داشته باشند .
PROTOS یک یاد آوری کننده را با همگرا کردن توضیحات تهیه شده توسط متخصص از خصیصه مورد فرا می گیرد .
بعضی یادآوری کننده ها یی که سنسور نامیده می شوند وابستگی های منفی بین خصایص مورد و دسته بندیها یا مثالها هستند و
پیشنهاد می کنند که مواردی که با خصایص شرح داده می شوند باید در طبقه بندی هایی ، رده بندی شوندو برای برداشتن ورودیها از مجموعه ای از دسته بندی های کاندید مفید است و از روابط استثنایی متقابل که در توضیحات استفاده می شود مشتق شده اند .
دومین نوع از دانش شاخص گذاری ،prototypicality (وابسته به طرح اصلی است) که مثالها را در یک دسته طبق رکوردهایشان در طبقه بندی قبلی مرتب می کند . وقتی یک مورد جدید PROTOSیک دسته را به یاد می آورد، مثالهای آن دسته با مورد به صورت prototypicality کاهشی مطابقت داده می شوند.
prototypicality ، یک اکتشافی است که در جه شباهت یک مثال را به دیگر اعضای آن دسته را ارزیابی می کند و به صورت توسعه ای فراگرفته شده است. اگر یک تطابق بین یک مورد و یک مثال توسط متخصصین پذیرش شود سپس prototypicality ، مثالی با یک مقدار متناسب با قدرت مطابقت افزایش می یابد .
سومین نوع دانش شاخص گذاری ، تفاوتهای مثالی است که مثالها را با خصیصه هایشان که آنها را از مثالهای دیگر با توضیحات یکسان متمایز می سازد ، شاخص گذاری می کند .بعد از یافتن یک مثال که با مورد جدید مطابقت دارد ، PROTOS به بهترین مثال تطابقی صعود می کند .

PROTOS تفاوتهای مثالی را با تطابق مورد جدید بهnear miss قبل از تطابق یک مورد با یک مثال که توسط متخصص ترجیح داده می شود ، فرا می گیرد .
near miss و مثال ترجیح داده شده ممکن است عضو یک دسته بندی یکسان یا عضو دسته بندی های مختلف باشد .
PROTOS آنها را با تفاوتهایشان به هم مربوط می دهد تا طبقه بندی درست را روی موارد بعدی بهبود بخشد .

الگوریتم طبقه بندی و پروسه فراگیری درPROTOS

تعیین شده : یک مورد (مورد جدید) برای طبقه بندی
تا از قویترین یاد آوری کننده های N یاد آوری کننده ها از خصایص مورد جدید را جمع آوری و ترکیب کنید . فقط
ترکیب شده را نگه دارید .
یک لیست از مثالهایی که بوسیله یاد آوری کننده مرتب شده ایجاد کنید .
تکرار کنید
تکرار کنید (مثالها را در یک ترتیب کاهشی در نظر بگیرید)
اجازه دهید مثالها ، مثالهایی با یاد آوری کننده های بالاتر باشند .
همانندی مورد جدید و مثالها را تعیین کنید
تا هنگامی که یک تطابق قوی شایسته پیدا شود
از تفاوتهای مثالی استفاده کنید از مثال ۱ تا مطابقت های بهتری را معلوم کنید (مثال ۲)
دسته مثال ۲ را برای طبقه بندی موارد جدید استفاده کنید و تطابق بین مثال ۲ و مورد جدید را استفاده کنید تا این طبقه بندی را شرح دهید .
این دسته بندی و توضیحات را با متخصص مطرح کنید
اگر متخصص این دسته بندی یا توضیحات را رد کرد
سپس دوباره یاد آوری کننده را برای خصایص مورد جدید تعیین کنید
دوباره دانش مطابقت جدید را از متخصص در خواست کنید
در غیر اینصورت (متخصص با دسته بندی یا توضیحات موافقت کرد)
prototypicality مثال ۲ را افزایش دهید
اگر مطمئن هستید که خصایص مثال ۲ مطابقت داده نشده اند

.

سپس توضیحات خصیصه – خصیصه را درخواست کن
اگر تطابق خیلی قوی است
سپس مورد جدید را دور بینداز
در غیر اینصورت مورد جدید را به عنوان مثال نگه دار
توضیحات خصیصه به دسته را ایجاد کن
اهمیت خصیصه را ارزیابی کن
یاد آوری کننده را برای مورد جدید فرا بگیر
اگر مورد جدید به نادرستی دسته بندی شده یا توضیح داده شده
سپس رکوردهای مثالی متفاوت هستند
تا هنگامی که متخصص با توضیحات و دسته بندی موافقت کند

۱۰-۳-۵ CAROL (استفاده دوباره به کمک مورد از کتابخانه اشیاء)

CAROL ، کلاسهای شیئ گرا مانند موارد در یک پایگاه مورد ، همانندی های کامپیوتر بین توصیفهای کلاس را
ارائه می کند و لیستی از توصیفهای کلاس معینی که بیشترین شباهت را به مشخصات کلاس مقصد دارد ، بر می گرداند .
(Shankaraman , 1989)
پایگاه مورد ، مجموعه ای از موارد است .موارد اسکریپت هایLOOM (زبان برای توسعه مدلهای شیء گرا در دانشگاه
ناپیر) که به عنوان حقایق پایگاه داده PROLOG ذخیره شدند و کلاسها و خواص و رابطه ها را نمایش می دهند .

اجزایCAROL

۱. ایجاد پایگاه مورد : اسکریپتهایLOOM (فایل های کتابخانه) ، از طریق ماجول هایی برای تجزیه و تحلیل نحوی ،
تجزیه و تحلیل معنایی ، طبقه بندی و استخراج خصایص ضروری از کلاسها برای ساختن مورد ،عمل می کنند .
خصایص کلاس مانند نام کلاس ، نام کلاس های بزرگ ، نوع کلاس ، متغیر های مثال و دیگر خصیصه ها برای خواص مورد متناظر تعیین می شوند .
۲. مشخصات : الگوی ورودی، بوسیله کاربر برای تعیین یک مورد جدید به عنوان

کلاس مقصد فراهم می شود . مشخصات مقادیری را برای بعضی یا همه خواصی که برای بازیابی ماجولها اقدام خواهند کرد ، فراهم می آورد .

۳. بازیابی : بازیابی یک مشخصات کاربر را در برابر همه یا بخشهای مشخص کاربر از پایگاه مورد مطابقت می دهد ولیستی از اسامی موارد تطابق یافته با امتیاز همانندیشان برمی گرداند .تنظیمات مانند تنظیمات بازیابی قابل دسترس هستند و سنگینی خصوصیات به کاربر اجازه میدهد تا تکنیک جستجو را انتخاب کند .

این سیستم فاقد این است که حمایتی را برای استفاده دوباره از کلاسها در یک سطح بالا از انتزاع فراهم آورد .آن همچنین نمی تواند توضیحاتی که کلاس بازیابی شده را تصدیق کند و کلاس بازیابی شده را با نیازمندیهای کاربرفعلی وفق دهد را تسهیل کند . برای غلبه بر این محدودیت ها ، نسخه ی جدید آن توسعه یافته است .

ویژگیهای اضافی در CAROL-II

۱. استفاده دوباره در سطحی بالاتر از انتزاع : تجزیه و تحلیل طراحی و مشخصات در سطح بالاتری از انتزاع رسیدگی می ، ROME ابزاری که در دانشگاه ناپیر توسعه یافت ، بعنوان ابزار تدریس استفاده می شود .

این یک ابزار محاوره ای در مدل کردن محیط است که مستندات طراحی و کدهای++C را حمایت می کند .

LOOM یک انتزاع را در مرحله طراحی تعریف می کند .

کلاس ورودی و خروجی پروسه های بازیابی شده ، اسکریپتهای LOOM هستند که استفاده دوباره را در سطح بالا تری ازانتزاع ، حمایت میکنند .

۲. توضیحاتی از پروسه های بازیابی شده : درCBR امتیاز تطابق برای خواص مختلفی از مورد برای توضیحات مفید هستند .

در CAROL-II، لیستی از خواص تطبیق یافته استفاده می شوند تا توضیحات را فراهم آورند . توضیحات به فرم امتیازهای

سر تا سری و امتیازهای دارای وزن برای هر خاصیت تطابق یافته فراهم می شود. امتیازها برای متغیرهای و خصوصیات رابطه بدست می آیند .هر امتیاز بدون وزنی برای تعیین کننده روشها و برچسبهای روش بدست می آید .

۳ .توافق کلاسهای بازیابی شده : کلاسهای بازیابی شده وفق داده می شوند تا متناسب

با نیازمندیهای کاربر باشند .
سیستمهای CBR روشی مثل جانشینی ،تغییر شکل ، ابتکاری و اشتقاق را برای توافق پیشنهاد می کند .
CAROL II روش جانشینی را که جستجوی محلی برای توافق نامیده می شود استفاده می کند .جستجوی محلی ، یک روش
از جستجوی یک ساختار دانش مشخص یا بخشی از پایگاه مورد که رابطه نزدیکی به یک جانشین برای بعضی ساختارهای قدیمی، مشخصات یا مقادیری که برای موقعیت جدید نامناسب هستند ، فراهم می آورد .
CAROL-II به این مورد رسیدگی میکند که اگر یک کلاس بازیابی شده خیلی شبیه به کلاس مقصد است سپس احتمال آن است که ساختمان داده اصولی ، درست است ولی بعضی عملکردهای آن ناپیدا هستند .

توافق درCAROL-II

الف) کاندید ابتدایی اسکریپتLOOM را در یک پنجره ویرایشگر بارگذاری کنید . برای هر مورد بازیابی شده ، لیستی از خصوصیات تطابق یافته و تطابق نیافته را نگهداری کنید .
ب)دسته های انتخاب شده کاربر از پایگاه مورد با استفاده از خبرهای همانند بکارگرفته شده در جستجوی اولیه ولی با یستی ازعملکردها ، مشخصات و خصوصیات به هم پیوسته که از کلاس مقصد مشتق شده اند جستجو می شوند (جستجوی اولیه) و با لیستی متناظر از خصوصیات تطابق نیافته از مورد کاندید ، جایگزین می شوند .
دومین کاندید بازیابی شده اسکریپتLOOM در پنجره ویرایشگر کاندید دوم بارگذاری می شود .
پ) پنجره ویرایشگر دومین جستجو برای کلمات کلیدی و هویت مربوطه، از لیستهای خصوصیات تطابق یافته از کلاس کاندید دوم مشتق می شود ، سپس از جعبه های مکالمه ای ، پرس و جوی کاربر استفاده می کند تا هر بخش مناسب را در پنجره ویرایشگر اولیه در یک نقطه متناسب کپی کند .
ت) به کاربر اجازه داده می شود تا ترکیب حاصل شده را به صورت دستی در جایی که مورد نیاز است ویرایش کند و آنرا در فایل ذخیره کند .
CAROL - II با تسهیلاتی مانند توضیحات و توافقات ، حمایت بهتری از استفاده

دوباره فراهم می آورد .

٤-۱۰ تشخیص مبنی بر مورد

تشخیص ، در یک سیستم مبنی بر قاعده بوسیله زنجیره های عقبگردی که از زنجیره های پیشرو با حذف بسیاری از امکانهای منطقی که متناقض هستند پیروی می کند ، ایجاد می شود . آنها به استنتاجهایی اشاره میکنند که غلط شناخته شده اند .

در استدلال مبنی بر مورد ، ما به مواردی که در سرتاسر زمان ساخته شده اند اعتماد می کنیم تا تخصصی را فراهم کنیم ، ولی شناخت با دیگر انواعCBR خیلی متفاوت است .

خلاصه شناخت از طریقCBR در زیر داده شده است:

الف) یک درخت از سؤالها برای کاربر ارائه کنید جایی که سؤالات بعدی به جوابهای قبلی بستگی دارد .

ب)درخت جوابها را با درختهای جوابهای مورد مقایسه کنید ، موارد همانند را انتخاب کنید .

پ) راه حل را از یکسان ترین موارد انتخاب کنید ،راه حل را با قواعد یا جداول چک کنید .

اولین سؤالها معمولا شبیه هم هستند ، ولی بعد از آن ، سؤالهایی که پرسیده می شود به جوابهای قبلی و بنابراین به فرم درخت بستگی دارد .

این ممکن است که فقط شاخه های نتایج از درخت سؤالات و جوابها را برای بازیابی استفاده کنید .بعد از اینکه یک یا چند راه حل بوسیله قواعد یا راه حلها چک شد ، یک قسمت بازخوردی در جایی که راه حل بکار میرود وجود دارد .اگر راه حل موفقیت آمیز باشد ، سپس آن در درخت جواب قرار می گیرد و به عنوان مورد ذخیره می شود ، در غیر اینصورت پروسه بالا با حداقل بعضی تغییرات کوچک تکرار می شود .

در روشCBR ، الگوی جوابها مهم هستند زیرا آنها علائم را تعیین می کنند .یک شاخه تعیین شده برای بازیابی شاخه های همانند دیگر در پایگاه مورد بکار می رود .

اگر شاخه بازیابی شده همان چیزی است که می خواهیم ،تشخیصش به عنوان راه حل استفاده می شود .

چقدر باید یک شاخه بازیابی شده نزدیک باشد تا قابل اجرا باشد ؟ این بستگی به نوع تشخیص و قضاوتی دارد که باید برای بعضی انواع دانش ساخته شده در سیستم ، درست شود ،دارد .

ولی تشخیصCBR می تواند به عنوان مددکار برای انسان بکار رود که آنها را راهنمایی می کند ومطمئن می شود که آنها ازامکانهای مهم چشم پوشی نمی کنند .

تشخیص هایCBR می توانند ، برای مثال در پزشکی ، تعمیرات اتوماتیک ، تعمیر کامپیوتر و کنترل سیستم و غیره استفاده شوند .

۵-۱۰ بعضی فواید

در مقایسه با سیستمهای مبنی بر قاعده (سیستمهای خبره) ،CBR به یک مدل صریح نیاز ندارد و موارد جمع آوری شده به

پایگاه مورد در طی توسعه و بعد ازتوسعه که خصایص مهم را می شناسد ، اضافه می شوند .که این آسانتر از توسعه یک مدل صریح است چنانکه این ممکن باشد که پایگاه مورد را بدن گذراندن گردنگاه فراگرفتن دانش ، توسعه داد .

متخصصین ممکن است که این را سخت بدانند که دانششان را مانند قواعدی بیان کنند ، ولی آنها سختی کمتری کمتری در فراخوانی موارد به هم پیوسته ای دارند که در عمل با آن مواجه می شوند. بنابراین ، مجموعه ذهنیشان به صورت متمایل به سمت روش مبنی بر مورد ، حداقل برای تصمیمات دانش سطحی ظاهر می شوند.

مدل مبنی بر سیستم ممکن است برای سطح دانش سطحی یا عمیق استفاده شود .برای یک دانش عمیق یک روش مبنی بر مدل تنها راه حل امکان پذیر است .

اگرچه برای ارائه دانش سطحی ، که برای بسیاری از سیستمهای عملی تشخیص مورد نیاز است ،CBR فواید آشکاری دارد .

در یک تحقیق ، مسئولیتی به عهده سیستمهای تشخیص گذاشته شد ، توضیح داده شد که یک سیستم مبنی بر قاعده ، مدت ٤ ماه زمان برد تا ساخته شود در حالیکه سیستم CBR معادل آن فقط ۲ هفته نیاز داشت . ادعاهای مؤثری همچنین وجود داشت

که پیشنهاد می کرد که توسعه یک سیستمCBR سریعتر و آسانتر از ساختن یک معادل مبنی بر قاعده است .

پایگاه موارد ، مجبور نیستند که کامل شوند وقتی که آنها برای استفاده ای توسعه می

یابند و موارد می توانند به ساختار موجود در هر محلی و بوسیله متخصصین غیر کامپیوتری اضافه شوند .
نگهداری پایگاه های موارد بسیار درست است به ویژه وقتی که با سیستم مبنی بر قاعده مقایسه می شود ، که شامل کارهای خطایابی بیشتری است .
این ناشی از اتکاء متقابل خصایصی است که باید رسیدگی شوند .
موارد معمولا به صورت رکودهایی در یک پایگاه داده ذخیره می شوند که می تواند حجم زیادی از داده های رضایتبخش را مدیریت و ذخیره کنند .
نگهداری آسانتر ایجاد می شود وقتی که این سیستم ها قابلیت فراگیری یا گزارش مواردی که نیاز است اصلاح شوند را دارند .
سیستمهایCBR میتوانند توضیحات مفهومی را به عنوان کمک فراهم آورند که می تواند به موارد منفرد یا عمومی پیوست شود .

فصل یازدهم

برنامه نویسی محدود شده

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

■ در این فصل با برنامه نویسی محدود شده که یکی از بهترین توسعه های جالب در زبان برنامه نویسی در دهه اخیر است آشنا میشویم.

■ در مورد مسایل رضایتمندی محدودیت یا CSP و چگونگی بدست آوردن راه حلهای آن اطلاعات کسب میکنیم.

■ به ضرورت استفاده از برنامه نویسی محدود شده پی میبریم و میفهمیم که نوع بنیادی اطلاعات محدودیات هستند و آنها قابل اجرا روی انواع زمینه ها و تقریبا تمام موجودیت ها هستند و حتی آنها یک نوع عمومی از اطلاعات در ارتباطات روز به روز ما هستند.

■ به این موضوع رسیدگی میکنیم که وقتی یک محدودیت به یک سیستم داده میشود چه رخ میدهد.

■ نقش انتشاردهنده و فرایند جستجو را که بعنوان عناصر پایه ای در سیستم برنامه نویسی محدود شده بیان شده اند را درک خواهیم کرد.
■ در مورد انواع مدلهای سازگاری در این نوع سیستم اطلاع کسب میکنیم و همچنین راه حل مسایل Life _Real که محدودیت دارند ، مانند مسئله واگذاری برچسبهای ریز برنامه ای نمادین به آدرسهای بانیری و مسئله ی از هم ترازی توالی DNA را خواهیم دید.
■ با زبانها و تجهیزات موجود در این سیستم آشنا میشویم و در آخر زمینه های پیشرفتها این نوع سیستم برنامه نویسی در آینده را خواهیم خواند.

۱۱-۱ مقدمه

برنامه نویسی محدود شده یا CP نوعی از مطالعه مربوط به مدلهای محاسباتی است و سیستمهایی که استوار بر این محدودیتها است. این عقیده مطابق با مدل و حل کردن یک مسئله بوسیله کاوش برمحدودیات است که میتواند مشخصه کاملی از مسئله باشد ، از این رو هر گونه حل مسئله ای باید ارضاکننده ی تمام حالتهای محدود شده باشد. برنامه نویسی محدود شده یکی از بهترین توسعه های جالب در زبان برنامه نویسی در دهه اخیراست. زبانهایی مانند پرولوگ بر یک سری مستندات حساب و دیفرانسیل استوارند و زبان لیسپ استوار بر برنامه نویسی تابعی جایی که به تشخیص هدف نزدیکترند درباره " برنامه ریز مسئله را مشخص میکند و کامپیوتر آنرا حل میکند."

CP جدیدا به طور نسبی در این زمینه مورد گرایش قرار گرفته است. در هر حال مانند آن در گذشته، زبانهای پرولوگ و لیسپ مثالهای کلاسیک هستند ، برنامه نویسی محدود شده همچنین، هنوز برای پیداکردن پایگاه وسیعی در صنعت نرم افزار است. وقوع و تکرار نمونه ای از پیشامد خودش یک نشانه ای است که CP هنوز به طور عادی یک زبان توسعه یافته ی مورد قبول است.

اما، نباید اینطور تفسیر شود که فقدان پذیرش گسترده، ناشی از برخی ضعفهای ذاتی در زبان یا نمونه است. بسیاری از ضعف ها ناشی از فقدان فهم در میان صنعت و آکادمی، پیرامون احتمالات و یک تخمین واقعی از ریسکهای درگیر و مبهم است . مقایسه ای بنا بر روزهای ابتدایی از برنامه نویسی منطقی، ظهور ماشین های انتزاعی WAM_LIKE ساخته شده برای تالیف برنامه های منطقی نتیجه دار ممکن، به سوی پیشرفتهای مهم در حالتهای اجرایی از برنامه های نوشته شده در زبان های PROLOG_LIKE است. امروزه بطور مشابه برای مسئله های پیچیده ، بسیاری از زبانهای برنامه نویسی محدود شده میتوانند برای حالتهای قابل مقایسه در زبانهایی مانند جاوا کار کند ، وفواید عظیمی در روابط حاکی از نیرو ، تراکم ، سطح بالایی از انتزاع ، و غیره را عرضه کنند. در طی سالهای نه چندان دور ، پیشرفت های عظیمی در زمینه تعدادی از درخواست های کاربردی وجود داشته است.

یکی از جنبه های جالب در زمینه های برنامه نویسی محدود شده ، ماهیت " inter disciplinary – " همراه با تعدادی از زمینه هایی که شامل آنالیزهای عددی ، پژوهش های عملی ، زبانهای برنامه نویسی ، هوش مصنوعی ، منطق ، ریاضیات ، و غیره میشود ، است. عقاید و فنون از تمام اینها برای حل مسئله های مخصوص در حال قرض گرفتن هستند. چنین یکپارچگی تقریبا در هر نمونه دیگری از برنامه نویسی سخت و دشوار است. از آنجاییکه بسیاری از استراتژی های حل و جزئیات ، خارج از مشخصات مسئله قرار دارند ، سیستم میتواند پیوندهای مناسبی برای بکار بردن الگوریتم های اختصاصی مطابق با هر یک از این زمینه ها تنظیم کند.

مبدا برنامه نویسی محدود شده میتواند به عنوان ردپایی برای به عقب برگشتن به منظور عمل پیشگام Sutherland در نگاره سازی کامپیوتر (Sketchpad) باشد

در جاییکه شکلهای مختلف تعیین شده ، در شرایطی روی یکدیگر محدودیات را اعمال میکنند ، کار **David Waltz** در برچسب زدن لبه برای دید کامپیوتر ، و کار **Akan Borning** روی **"Thinglab "** . بعد از آن زمان برنامه نویسی محدود شده در واقع بمراتب کامل شد. یک تعدادی از زبانها و چارچوب ها قابل دسترس هستند که شامل تعدادی از انواع تجاری هستند. که شامل **ILOG Solver** ، **PROLOG III** ، و غیره هستند. ولی سپس ، باقیمانده راه برای رفتن شاید بیشتر از آنچه تاکنون پوشش داده شده ، است. یکی از ضعف های قوی ، فقدان محیط های توسعه ی نیرومند، که شامل حمایت کردن از عمل اشکال زدایی است، میباشد. زبانهای محدود شده موجود، بطور ذاتی در نحوه دسترسی شان به برنامه نویسی محدود شده ، انتخاب ترکیب یک چارچوب سخت ، تفاوت دارند. زمینه همچنین برای عمل پژوهش اضافی مهیا و غنی است. امکانات بیشتر میان آکادمیک ها و صنعت ، بطور ویژه برای حل کردن مسائل پیچیده ، خواستار یک ترقی خوشایند برای عمل اضافی در زمینه است.

۱۱-۲ مسائل رضایت محدودیت (CSP)

CSP یک موضوع پژوهش در هوش مصنوعی ایست که در سالهای بسیاری وجود داشته است. پیشگام(**Pioneering**) روی شبکه هایی ازمحدودیات که بیشتر بوسیله رخداد مسائل در زمینه ای از پردازش تصویر تحریک شده اند، کار میکند. پژوهش **AI** که متمرکز بر مسائل ترکیبی مشکل است به پیشرفت قابل توجهی در استدلال **constraint-based** کمک میکند. الگوریتمهای قدرتمند بسیاری طراحی شده بودند که یک مبنایی از جریان الگوریتمهای رضایت محدودیت ، شوند.

یک **CSP** شامل:

- یک مجموعه ای از متغیرها $X = (x_1, \ldots, x_n)$,

- برای هر متغیر X_i ، یک مجموعه متناهی D_i ، ازارزش های ممکن (دامنه)
- و یک مجموعه ای از محدودیات قیود که مقدار متغیرها میتواند به طور همزمان گرفته شوند.

یادداشت ها برای آنکه ارزش داشته باشند نیازی ندارد که یک مجموعه ای از اعداد صحیح پیوسته باشند ، آنها حتی نیازی نیست که عددی باشند.
یک راه حل برای CSP واگذاری یک مقدار از دامنه خودش برای هر متغیر است واین راه راضی کردن هر محدودیتی است. بنابراین ما ممکن است بخواهیم پیدا کنیم:

- فقط یک راه حل ، که برتری نسبت به راه حلهای دیگر نداشته باشد.
- همه راه حلها
- یک راه حل بهینه ، یا حداقل یک راه حل خوب، که تعیین کننده تعدادی توابع هدف که تعریف کننده روابط بعضی از متغیرها یا همه ی آنها است ، میباشد.

راه حل های CSP میتوانند بوسیله تحقیق قاعده دار در سرتاسر واگذاری مقدارهای ممکن به متغیرها یافت شوند. روشهای تحقیق به دو طبقه تقسیم میشوند : آنهایی که فضایی از راه حلهای جزئی را طی میکنند (یا واگذاری ارزش جزئی) ، و آنهایی که فضایی از واگذاری ارزش کلی را بطور ناگهانی کشف میکنند (برای همه متغیرها) .

مثال ۱) فرض میکنیم متغیرهای X, Y, Z داریم و دامنه آنها هست:

$$D\_X = D\_Y = \{1, 2, 3\}, \quad D\_Z = \{2, 3, 4\}$$

و محدودیتهابه صورت زیر هستند:

$$X < Y \text{ and } Y < Z$$

آن واضح است که CSP رضایتمند است. برای مثال، واگذاری زیر یک راه حل است.

$$X = 1 , Y = 2 , Z = 3$$

مثال ۲) کاربرد مسئله ٤ وزیر به عنوان یک مثال، ما میتوانیم یک متغیر به عنوان نمایش هر وزیر بکار بریم ، بنابراین ما دامنه هایی برای متغیرها به صورت زیر داریم:

$$D\_xl = \{1, 2, 3, 4\},$$
$$D\_x2 = \{1, 2, 3, 4\},$$
$$D\_x3 = \{1, 2, 3, 4\},$$
$$D\_x4 = \{1, 2, 3, 4\},$$

محدودیتها برای آنکه راضی شدنی باشند میتوانند حالتی مانند روابط باشند . محدودیت برای آنکه بگوید X_1 و X_2 نمیتوانند روی یک ستون مشابه باشند ، میتواند بوسیله مجموعه ای از tupel های رضایتمند بیان کننده این عبارت باشد :

$$r1 (x1,x2) = \{(1,2),(1,3),(1,4),(2,1),(2,3),(2,4),(3,1),(3,2),(3,4),(4,1),(4,2),(4,3)\}$$

تعریف کردن یک رابطه بوسیله تاپل های حالات رضایتمند ، کسل کننده است. یک زبان مناسب ممکن است این عبارت را به صورت زیر شرح دهد

rl(x1, x2) if xl is not equal to x2,.and
xl takes a value from D_xl and x2 takes a value from D_x2

واضح است که ما یک زبان برای توضیح دادن همه سه بخش از یک CSP احتیاج داریم.

۱۱-۳ چرا برنامه نویسی محدود شده ؟

در اصل برنامه نویسی یک فرآیند ارتباط دهی نیازهای ما به کامپیوتر برای بدست آوردن یک راه حل است. چه نوعی از اطلاعات شامل نیازهای ما بطور نرمال است؟ در اکثر اوقات ، راه حل ما بر طبق این متمایل شده بواسطه چارچوب کاربردی ما است- زبان برنامه نویسی بکار گرفته شده. هرچند ما میتوانیم انواع مختلفی از اطلاعات خارج یک عدد را فهرست کنیم مانند – موجودیتهای مناسب برای مسئله ی در دست ، خواص و رفتارهایشان ، نسبتهای میان موجودیتهای مختلف ، محدودیتهایی روی رفتار موجودیتهای مختلف ، محدودیتهایی روی خواص ، وظایف مختلف که نیازبه انجام شدن دارند و موجودیتهای درگیر در آنها ، و غیره. زبانهای برنامه نویسی رویه ای یک زمینه خوب برای شرح چگونگی انجام وظایف معین را مهیا میکنند. زبانهای شیءگرا یک مکانیسم برای توصیف موجودیتها و خواص و رفتارشان تهیه میکنند.

نوع بنیادی اطلاعات ، محدودیات هستند. آنها قابل اجرا روی انواع زمینه ها و تقریبا تمام موجودیت ها هستند.

محدودیات یک نوع عمومی از اطلاعات در ارتباطات روز به روز ما هستند و آنها از مشخصات برنامه ای ناشی میشوند. برای مثال :

- پهنای ستون شمالی نباید از ۵۵ متر تجاوز کند.
- دایره ، محاط شده داخل مثلث (بعبارت دیگر ، اگر مثلث توسعه یابد ، دایره هم باید رشد کند و غیره)
- تعداد اشیاء فرزند A باید بین ۳ و ١٤ باشد.
- حقوق ، یک عدد صحیح بین ٤۰۰۰ و ٤۰۰۰۰ است.

بهر حال ، زبانهای عمومی کنونی به ما اجازه نمیدهند که بطور صریح انواعی از

اطلاعاتشان را کدگشایی کنیم. اغلب آنها تبدیل شده به یک مجموعه ای از نمایش های جزئی مختلفی اند که در سراسر برنامه پخش شده اند. برای مثال ، محدود کردن تعداد اشیاء فرزند باید بوسیله یک تست صریح درساختن یک شیء و در روشی که کدام فرزندها را به شیء دیگری اضافه کند و هر روش دیگری که باید فرزندها را به یک شیء اضافه یا پاک کند ، انجام گرفته شده باشد. این یک فرایند سنگین و طاقت فرساست .

بطور مساوی ، چگونگی تهیه اطلاعات اضافی توسط محدودیات در بعضی موارد مهم است. اگر ما بدانیم که x و y کمتر از ۱۰ هستند و $x+y > 20$ ، سپس راه حلی برای تعیین مقدارهای x و y بطور صریح وجود ندارد. برای این مثال، ما احتیاجی به تست کردن مقادیری که خارج از محدوده x و y است نداریم. یک مجموعه ای از محدودیات روی x ممکن است به تعیین کردن مقدارش کمک کند (هر کدام بطور اختصاصی عاجز از تصمیم گیری یک مقدار برای x). این یک استدلال نیرومند معنی دار و مدل حل مسئله ای مفید در بسیاری از مسائل است. چارچوب های برنامه نویسی قراردادی این موضوع را حمایت نمیکند .

محدودیات ، مقایسه ای روی انواع مختلفی از اطلاعات دارند :

- آنها میتوانند اطلاعات جزئی را نشان دهند . $x > 4$ یک محدودیت معتبر است که یک مقدار واحد به x نمیدهد .
- آنها میتوانند انواعی از اطلاعات را رسیدگی و گزارش کنند.
- آنها افزودنی هستند و از این رو مستقل از ترتیبی که تعیین شده اند ، میباشند. $X >4$ و $X <6$ مانند $X<6$ و $X > 4$ است . این یک درجه آزادی قوی است در جاییکه ، زبانهای قراردادی یک ترتیب معین برای انتخاب و اجرائیات اعمال میکنند.
- آنها non_directional هستند . اگر $x+y =z$ این طور خوانده شود " z حاصل میشود از جمع x وy" ، آنگاه این عبارت یک دستور را بطور واقعی اعمال نمیکند . اگر x و z شناخته شده باشند، ما میتوانیم y را از معادله بدست آوریم.

در زبانهای قراردادی ، گزاره ها directional هستند و میتوانند دقیقا در یک دستور استفاده شوند . به طور معنایی ، آنها دو مرتبه مقدار متغیر سمت چپی (LHS) قبلی را مینویسند.اگر هر یک از متغیرهای سمت راست (RHS) غیر محدود باشند، یک خطا گزارش داده میشود . هر چند مانند یک محدودیت ، تساوی میتواند بسادگی ، برای اعتبار، جفت بکار برده شود اگر تمام سه متغیر محدود باشند.

- آنها بیشتر اعلانی از اکثر انواع اطلاعات هستند و بطور عادی هیچ عنصر رویه ای را در بر نمیگیرند.

۱-۳-۱۱ Issues پایه ای

چگونه برنامه نویسی محدود شده از مدلهای دیگر متفاوت است؟ آیا آن قابل قیاس با مدلهای دیگر است؟

ما یک نظریه بر اندیشه های اساسی و پی آمدهای مربوط به پاسخ به این سؤالها بدست خواهیم آورد. ما میتوانیم نظریه مسائل برنامه نویسی مان را بوسیله یک مجموعه ای از متغیرها تعیین کنیم- در این زمان ما نگرانی نداریم اگر مجموعه ای متناهی یا نامتناهی باشد. و کار اصلی ، واگذاری مقدارها به این متغیرات ، راضی کردن تمامی مشخصات است. این یک نظریه ای از برنامه نویسی برای نیازهایمان که بطور کافی قوی است ، میباشد. بطور سنتی با یک مجموعه از متغیرها شروع میکنیم، تعدادی به طور اتوماتیک ، و تعدادی هم بوسیله ورودی کاربر، مقداردهی میشوند. مابقی در طی فرایند حل کردن مسئله محاسبه میشوند. این بطور عادی یک فرایند تکراریست ، که مبنی بر مقدارهای متغیرهای مختلف و مقدار متغیر محاسبه شده است . این وابستگی ها میتواند مانند محدودیات در یک فرم فشرده گرفته شوند. x2+y2=z یک دستورالعمل برای محاسبه z . آن همچنین رویه ای برای محاسبه کردن x و y از دو متغیرهای دیگر در بردارد. در مدلهای سنتی ، ما مجبوریم که این موقعیتها را مانند رویه های جداگانه نمایش دهیم.

چگونه برنامه نویسی محدود پیرامون چنین مسائلی عمل میکند؟ در مدلهای برنامه نویسی محدود ، ما محدودیت ها را همچنان که آن هست نمایش میدهیم. بر طبق آنچه که قبلا گفتیم ، دستورالعملهای مشخص شده محدودیتها مجرد هستند. بنابراین x=2; y=x+1; معادل y=x+1; x=2; است. پس برخلاف زبانهای سنتی، ما نمیتوانیم حالات را بعد از آنکه آنها مراحلی را طی کرده باشند ، دور بیاندازیم. وقتی ما به عبارت y=x+1 کردیم میدانیم که y یک واحد از x بیشتر است با اینکه نه مقدار x و نه مقدار y را نمیدانیم. بنابراین اگر چیزهای بیشتری درباره x یا y شناخته شده باشد ، ما احتیاج به ذخیره کردن این محدودیت ها، و تست دوره ای داریم . پس یک جزء ضروری از هر سیستم برنامه نویسی محدودیت، انبار محدودیت است که که محدودیتهای جاری را ارائه میکند. وقتی یک محدودیت جدید بوجود می آید، به انبار اضافه میشود.

چگونه ما از یک محدودیت استفاده میکنیم؟ اصولا ما به عامل برای اینکه مطمئن سازد که یک محدودیت معین هرگز تجاوز کننده نیست ، احتیاج داریم و همانند متغیرهای درگیر ، تغییرات را نگه میدارد. در مثال بالا ، مقدار اولیه x و y نامحدود شده است ، بنابراین چیزی برای انجام دادن وجود نداشت. پس ممکن است ما میگفتیم که x>10 . به هر حال خطری از تجاوز و تخطی محدودیت وجود ندارد. حال اگر دیدمان نسبت به محدودیت به طور سودمند باشد، ما متوجه میشویم که ما میتوانیم امکاناتی از y را محدود کنیم ؛ برای بزرگتر بودن از ۱۱. قبل از اینکه ما این عامل را کشف کنیم اجازه دهید که نگاهی به جزئیات بیشتری در مورد محدودیت داشته باشیم.

ما میتوانیم محدودیت را بطور وسیع بصورت دسته های زیر طبقه بندی کنیم:

۱) محدودیت های دامنه : اینها محدوده ای از مقدارهای معتبری که یک متغیر میتواند بگیرد را تعیین میکنند مانند یک زیر مجموعه ای از دامنه متغیر. برای مثال :

"integer age; age 1#100" معنی میدهد که age یک متغیر integer است که

.

مقدارهای ممکن آن در محدوده ی ۱ تا ۱۰۰ است.
۲) محدودیت های پایه ای : اینها محدودیت ساده روی یک متغیر را نمایش میدهند ، متغیرهای دیگر را درگیر نمیکنند.
۳) محدودیتهای دیگر : اینها تمام انواع دیگر را شامل میشوند و احتمالا یک گروه بزرگی از مواردی مانند x*x + y*y < 200 , x-y < z , x > 2*y و غیره را دربرمیگیرند.

چگونه ما یک محدودیت رابه طور درونی ارائه می کنیم ؟ در مدلهای محاسباتی سنتی ، یک متغیر برای یک مقدار شناخته شده، محدود یا نامحدود است . ما انواعی از داده های از پیش تعریف شده داریم که مقدارهای مختلف را رمزگذازی میکنند .منظور از اینکه " X هست ۱۰۰# ۱۰"چیست؟ به طور عادی دوراه در تصرف کردن اطلاعات وجود دارد.
در نمایش فاصله ای ، مجموعه هایی از مقدارهای مجاز به صورت فاصله نمایش داده میشوند . برای مثال age [10+50] نشان میدهد که age بین ۱۰ و ۵۰ است . در نمایش دامنه ای یک لیستی از مقدارهای قابل دسترس وجود دارد .این آشکارا تنها برای اندازه های کوچک قلمرو امکان پذیر است . از میان آن دو، نمایش فاصله ای کار آمد تر برای نمایش است چون تنها یک جفت از مقدارها را به کار میگیرد . اما آن همچنین کمترین تاثیر را دارد و برای مثال اگر تعیین شده باشد که age یک عدد زوج است در نمایش دامنه ای ما میتوانیم تمام اعداد فرد را در نظر نگیریم اما در نمایش فاصله ای نمایش بی پیرایه است . بیشتر سیستم های تجاری نمایش فاصله ای را به دلیل کارایی اش زیاد به کار میبرند .
حال اجازه دهید به این موضوع رسیدگی کنیم که وقتی یک محدودیت به یک سیستم داده میشود چه رخ میدهد. اگر آن یک محدودیت دامنه ای باشد فرآیند آسان است . ما یک فاصله متناظر را برای دامنه و ذخیره آن تصرف میکنیم. پس ما میتوانیم محدودیت را رها کنیم در نتیجه ما در نمایش ، تمام اطلاعات محدودیتها را تصرف کرده ایم .

برای محدودیت پایه ای همچنین ما میتوانیم یک فرایند مشابه را انجام دهیم . اگر محدودیت $x<a$ باشد ، ما میتوانیم کران بالای x را در صورتی که بزرگتر از a باشد اصلاح کنیم .محدودیت نمیتواند اطلاعات بیشتری تهیه کند . از این رو ما میتوانیم آن را از انواع حذف کنیم . برای محدودیتهای دیگر مانند $x+y<z$ ، این امکانپذیر نیست . ممکن است قادر به حذف کردن بعضی مقدارها از دامنه تمام سه متغیر باشیم ، اما ما نمی توانیم تمام اطلاعات تهیه شده به وسیله محدودیت را جذب کنیم . برای مثال ، اگر $x>5$ ، $y>3$ ، $z>2$ باشد پس، با این محدودیت ما می توانیم کران پایینی z را به ۸ اصلاح کنیم . اما تا وقتی که حداقل مقدار دو متغیر، محدود به مقدار واحد هستند ، ما نمی توانیم این مقدار را بوسیله تعویض کردن نمایش دامنه ای حتمی کنیم.

بنابراین ما متوجه مرحله پیشرفت محدودیتهای معین بطور کامل می شویم ، از آنجاییکه برای محدودیت دیگر ، به نظارت بر مقدارهای متغیرها برای مطمئن شدن از اینکه محدودیتمان رضایت بخش است احتیاج داریم.

بعنوان مثال اگر ما از قبل بدانیم که $z>15$ است آنگاه ما می توانیم کران بالایی x را به ۱۱ کاهش دهیم . این تغییر ممکن است محدودیت های دیگر x را trigger کند که ممکن است بر دامنه های متغیر دیگر، اثر داشته باشد . این فرآیند، گسترش محدودیت ((propagathin constraint)) نامیده می شود و یک بخش بحرانی برنامه ریزی محدود است .

در بعضی موارد ، که محدودیت کافی وجود دارد آن برای سیستم مقدور می باشد که با مقدارهای واحد برای تمام متغیرهایی که نتیجه ی فرایند گسترش تکراری هستند مطرح شود، هر چند به دو دلیل ، این مورد عادی نیست . اولا ، چنانچه در قبل دیدیم ، ارائه ها بطور عادی برای گرفتن معنی و مفهوم یک محدودیت بطور کامل کافی نیست و بنابراین propagation ممکن است زود متوقف شود . دوماً ، محدودیت ها به تنهایی مقدور نیست یک راه حل واحد را تعیین کنند . بنابراین تمام سیستم های برنامه نویسی محدود یک جستجو را ثبت می کنند . وقتی گسترش محدودیت تثبیت شد ، سه امکان

وجود دارد :

۱) یکی یا بیشتر متغیرها ، دامنه هایشان خالی می شود . این مفهوم یک مجموعه ای از محدودیت های متناقض را می رساند و بنابراین راه حل معتبری وجود ندارد.

۲) تمام متغیرها محدود به مقدارهای واحد هستند . بعبارت دیگر ، تمام دامنه های مجموعه ها یکی هستند . یک راه حل شایسته حاصل می شود .

۳) هیچکدام از موارد بالا یک متغیر نه دامنه غیر یکی و نه دامنه خالی دارد . در این مورد ما احتیاج به جستجوی بیشتری برای یک راه حل داریم .

فرایند جستجو مانند زیر عمل می کنند . یک متغیر نامحدود v ،مادامیکه یک مقدار در محدوده ی a#b داشته باشد، برداشته میشود. در یک راه حل، مامیتوانیم هر کدام از v=a یا v<>a را بیان کنیم. در نتیجه ما دو امکان داریم. ما هر دو را در جستجو کشف می کنیم . در یک شاخه ، ما v = a و در دیگری v<>a را داریم .

ما هر دو را در جستجو کشف می کنیم . در یک شاخه ، ما v = a و در دیگری v<>a را داریم این ها بعنوان محدودیتهای جدید در جائیکه گسترش بیشتر امکان پذیر است ،عمل می کنند . اگر آنها دو مسئله مختلف باشند ،دو شاخه بطور مستقل کشف می شوند. راه حل های هر دو شاخه ترکیب می شوند و بعنوان راه حلهای مسئله بازگردانیده می شوند .

این فرایند ممکن است در دو شاخه چند بار تکرار شود . اگر مورد ۳ پافشاری کرد حتی بعد از آنکه x = a، ما قادریم متغیر دیگری را بگیریم و مانند آن مقیدش کنیم . این فرایند تکرار می شود تا زمانیکه متغیر اضافی برای محدود شدن باقی نماند ، یا یک خطا گزارش شود . بنابراین در اصل یک سیستم برنامه نویسی محدود شامل یک انبار محدودیت برای نگهداشتن محدودیتهای non-basic ،ویک مدل گسترش محدودیت ، ویک مدل جستجو . اثر انتشار دهنده بیشتراز ارزش جستجو است اما در اصل ، برای مطمئن شدن این که فرایند راه حل کامل است یک مدل جستجو ضروری است.

۱۱-۳-۲ مدلهای سازگاری

در اصل نقش انتشار دهنده کاهش دادن مقدارهای ممکن از متغیر دیگر است که اطلاعات اضافی روی یک متغیر را کاهش می دهد . در اصل برای هر تغییر در دامنه ی هر متغیر نیازبه آزمایش و پردازش هر محدودیت در سیستم دارد. و بالقوه یک تغییر در دامنه یک متغیر می تواند منجر به تغییراتی در دامنه ی متغیرهای دیگر شود . ما مقدارهای دامنه ی متغیرهایی که سازگار با هر مقدار متغیر دیگر نیستند را انتقال می دهیم . پس ما یک مدل انتشار دهنده گران داریم . کار بیشتر ممکن است در تست کردن مقایسه امکانات انتشار با نیازهای دنیای واقعی برای حل کردن مسئله ،انجام گیرد.

برای انکه انتشار محدودیت در عمل تکرار، موثر باشد ، باید یک راه کارامد برای انجام این عمل داشته باشیم . پیشرفت های آشکاری مانند نگهداری یک لیستی از متغیرهای ساختگی و برداشتن آن محدودیت هایی که به یکی از کتغیرهای ساختگی رجوع می کنند ، وجود دارد . اساس تحلیل و آنالیز اینگونه است که متغیرهای متفاوت در آن محدودیاتی که ساخته شده اند هویت دار شده و به مخزن اضافه می شوند . این دوری کردن از یک تست کامل از هر محدودیت است، تا وقتیکه انتشار متوقف شود . حتی آنزمانی که هزینه مانع است .

همانطوری که قبلا دیدیم در اصل ما قادر به بهره برداری تمام توان و نیروی محدودیت های بیشتری نیستیم . و هزینه این ناتوانی بالاتر از هزینه جستجو است . بکارگیری این واقعیت ، یک توافقی است که اغلب در مقدار انتشار و مقدار هزینه جستجو بکار برده شده است . چنین مدلهای سازگاری بسیاری وجود دارد .

ساده ترین مدل سازگاری node consistency که بر طبق محدودیات یگانی محدود می کند. آن مقدارهایی که متناقض با محدودیات یگانی روی متغیر هستند ، از دامنه متغیر انتقال می یابند . برای مثال ، اگر متغیر تعریف شده فرد باشد ، مقدارهای زوج

remove می شوند ، این برای اجرا کردن آسان است و بطور تخمینی گران نیست ، و محدود شده به یک پیمایش منفرد متغیرهاست .

مدل سازگاری بعدی ، arc – consistency است ، در جاییکه مابه یک جفت از متغیرها مانند مانند v2 و v3 رسیدگی می کنیم ، و تمام مقادیر دامنه ی v1 را که با تمام مقادیر v2 ناجور است ، انتقال می دهیم . برای انتشار هزینه بیشتری دارد . وقتی متعاقباً جفت متغیر v3 –v2را رسیدگی می کنیم مقادیر بیشتری از دامنه v2 ممکن است منتقل شود . در رسیدگی کردن به جفت v2 – v1 ممکن است مقادیر بیشتری از دامنه v1 ریخته شود . بنابراین ترکیبات جفتی مجبورند مکررا آزمون کند تا تغییر زیادی صورت نگیرد .

مدلهای سطح بالاتری مانند k- consistency وجود دارند ، در جاییکه k>2 باشد ، k-tuple ها از متغیرها رسیدگی می شوند . اینها بطور تخمینی پرهزینه تر و کاربردشان کمتر است .

اما اگر یک سیستم با N متغیر بتواند به N-CONSISTENT ساخته شود ، آنگاه حداکثر انتشار امکان دارد .

همانطور که گفته شد حتی arc – consistency برای اجرا پر هزینه است . اینجا فرایند کامل مجبور است هر زمان که تغییر در یک دامنه مشاهده می شود، تکرار شود . بنابراین بیشتر ابزارهای کاربردی از فرم arc – consistency محدود شده استفاده می کنند . یک تعدادی از الگوریتم ها در ادبیات شناخته شده اند که شامل AC1 و AC2 و غیره می شوند. یک مثال از محدودیت ،خواستار بررسی کردن arc-consistency برای تنها متغیر تحت توجه و رسیدگی در مقابل تمام متغیرهایی که جریان دارند است . برای مثال در مسئله n وزیر ،وقتی I امین وزیر در حال نشستن

است ، ما برای مقابله با خطا تمام وزیرهای ۱ تا ۱- i را بررسی می کنیم و وزیرهای i+۱ تا n را چک نمی کنیم . انواع استراتژیهایی وجود داشته که بر اساس الگوریتمهایش و بهینه سازی متفاوت اجرا شده اند . forward checlcing مدل arc-consistencyمحدود شده را یکی میکند در جاییکه متغیرهای بعدی در برابر متغیر جاری به منظور سازگاری بررسی شده اند . این به کشف زود هنگام خطاهای بعدی و همکاری به مظور فعالیت مفید و کارایی کمک می کند .Backjumping سعی بر بهبود رفتار backtracking دارد در جاییکه ترتیب ساده backtraking بطور عادی، مورد استفاده قرار میگیرد.Backmarking یک کوشش برای کاهش دادن محاسبات سازگاری تکراری است .

۱۱-٤ حل کردن مسائل Real - Life

یک عنصر مهم از حل کردن مسائل Real – Life که محدودیت دارند ، مدلسازی مسئله در شرایط محدودیتهاست . تبدیل کردن تعریف مسئله از زبان طبیعی به زبانی از محدودیتها . ما به دو مثال در اینجا رسیدگی خواهیم کرد . اولا ما مسئله واگذاری برچسبهای ریز برنامه ای نمادین به آدرسهای بانیری را خواهیم دید و سپس مسئله ی از هم ترازی توالی DNA .

۱۱-٤-۱ مسئله واگذاری برچسب ریز برنامه

مسئله وگذار کردن بر چسبهای ریز برنامه نمادین به آدرسهای باینری در یک صفحه ی ۲۵٦ آدرسی از حافظه ریزبرنامه است . یک نمونه دستورالعمل ریزبرنامه از فرم زیر پیروی می کند .

L0 : opcode L1,..,Ln

که L0 برچسب مبداء و L1,…Ln برچسبهای هدف هستند . برای بالا بردن کارایی و کاهش استفاده از حافظه ، برچسبها بیت های معینی را شریک می شوند و شاخه ای از وضعیت را بعنوان یک بخش از آدرس در بر می گیرند . بنابراین برچسبها نمی توانند

.

به آدرسهای متوالی واگذاری شوند مانند زبان اسمبلی ، اما باید در سراسر فضای آدرس برای راضی کردن تمام محدودیات توزیع شوند . برای مثال ، دستورالعمل یک شاخه ی ٤ انشعابی بصورت زیراست: LO: branch4 L1, L2, L3, L4

به نمایش باینری هر برچسب که شامل ۸ لیست می شود در زیر نگاه کنید :

```
L۰: X7 X6 Y5 Y4 Y3 Y2 Y1 Y۰
L1: X7 X6 X5  0    0   X2 X1 X۰
L2: X7 X6 X5  0    1   X2 X1 X0
L3: X7 X6 X5  1    0   X2 X1 X0
L4: X7 X6 X5  1    1   X2 X1 X0
```

توجه کنید که تعدادی از بیت ها برای تعدادی برچسب عمومی است . طبیعتا هر برچسب می تواند بوسیله یک متغیر با یک دامنه بین ۰ تا ۲۵۵ معرفی شود و توسط قیود زیر بصورت دودویی بیان شود :

- بیت های مساوی ، تعدا بیت بین برچسبها مشترک است .

```
(L۰&11000000=L1&11000000).
```

- بیت های وضعیت ، این بیت ها ثابت هستند .

```
(L2&00011000=00001000).
```

- افزایش ها ، بعضی از برچسبها مستقیما از برچسب های دیگر پیروی می کنند .

```
(Ll=LO+l)
```

- All different تمام بر چسبها باید به آدرسهای مختلف واگذار شوند .

۱۱-٤-۲ هم ترازی توالی protein / DNA

هم ترازی چند گانه از یک خانواده از توالی های protein / DNA یک مرحله بنیادی در مطالعه ی همسانی در شکل و تابع شان است . هدف هم ترازی چندگانه درج

نمادهای " _ " برای نشان دادن شکاف و فاصله داخل K توالی بعنوان یک راهی که نتایج K توالی طول یکسان داشته باشد و اگر نتیجه ی توالی ها مانند یک ماتریس مورد بحث قرار گیرد ، آنگاه هر ستون حداقل یک کاراکتر مختلف از "_" را شامل می شود . برای مثال :

LIMITED-	LIMIT-ED--	LIMIT--ED
LITTLE--	LIT-TLE---	LIT-TL-E-
LISTENER	LIS-T-ENER	LIS-TENER

کلید اشاره برای آن است که ما به سادگی خواستار یک هم ترازی اختیاری نباشیم . ما خواستار پیدا کردن یک هم ترازی خوب هستیم که می تواند همسانی یک گروه از توالی ها را بطور صحیح و ممکن نشان دهد . (تعریف "خوب" بر پایه دانش زیستی است) .

فرض کنید که داده ی ورودی K توالی با طول n است و ماکزیمم عدد درج شده (به هر توالی) m است . ما می توانیم از دو آرایه از متغیرهای دامنه متناهی برای نشان دادن داده استفاده کنیم :

- Position array S با سایز k*n : هر عنصر S_{ij} مرتبط با انتقال حرف j ام در توالی i ام است . دامنه هر S_{ij} ، {o,…,m} است .
- Letter array P با سایز k*(n+m) ، هر عنصر مرتبط با آخرین حرف هم ترازی است . دامنه هر P_{ij} ، { - ,S_{ij} , S_{ij-1} , … , S_{ij-m} } است که S_{ij} ، حرف j ام در توالی i ام مبدا است . (توجه کنید که دو پیشامد حرف یکسان باید بوسیله ارجاع یک آفست به هر حرف ، متمایز باشند .)

اکنون ، دو مجموعه از محدودیات می توانند تعریف شده باشند :

۱۱-۵ زبان ها و ابزاره

همانطور که قبلا ذکر شد، عناصر پایه ی یک سیستم برنامه ریزی محدود یک مکانیسم انتشار محدودیت و یک مکانیسم جستجو هستند . یک تعدادی از مدلها برای نشان

.

دادن محدودیات و استدلال به آنها در طی سالها پیشنهاد شده است . ما به دو مدل نگاهی می اندازیم : برنامه نویسی منطقی محدودیت و برنامه نویسی محدودیت همزمان .

زبا ن برنامه نویسی منطقی محدود ، یک نوعی از زبان است که بر یک مدل معنایی استوار است . مفهوم بنیادی این موضوع ، قرار دادن اداره محدودیت داخل یک چهارچوب برنامه نویسی منطقی (LP) . LP یک پایگاه معنای غنی و استوار در منطق FIRST – ORDER دارد ، که همراه با تحلیل نحوه شکل دادن اثبات اشتباه ، بر اساس نحوه مدل اجرایی اش است .

یک برنامه منطقی شامل مجموعه ای از قوانین و واقعیت هاست .(واقعیت ها ، قوانینی هستند که شرط ندارند) . اجرای آن توسط یک پرس و جو در مورد پیوستگی اهداف شروع می شود . مدل اجرایی ، برداشتن یک آیتم از لیست هدف جاری هویت بخشیدن به قانون یا واقعیتی که می تواند با هدف مشخص شده یکی باشد ، و جایگزینی آن هدف در لیست بوسیله شرط های fact / rule تطبیق یافته . یگانگی مراحل ، محدودیت های اجباری اضافه شده روی متغیرهای موجود در لیست هدف را معرفی می کند که به تمام هدفها در لیست هدف نتیجه ، منتشر شده است .

بطور واضح در اینجا موضوعات آشکار بسیاری وجود دارد . اگر قوانین / واقعیتهای متعددی برای یکی شدن با یک هدف موجود باشد چه اتفاقی می افتد ؟ و اگر وجود نداشته باشد چه اتفاقی می افتد ؟ در بیشتر زبان های ترتیبی یکی از option ها انتخاب عملیات است . و متعاقبا اگر آن option پیدا شد ، آنگاه منجر به یک راه حل نمی شود . زبان های موازی همزمان شاید option های متفاوتی در همزمان کردن آزمایش کنند ، ولی بدون دخالت اجباری بر دیگری .اگر لیست هدف شامل یک هدف باشد قانون / واقعیتی که بتواند با آنها یکی شود را نداشته باشد ، آنگاه لیست هدف ناراضی و یک خطا گزارش می شود .

.

تعریف یک چارچوب برای برنامه نویسی منطقی میتواند بطور قابل توجهی آن برنامه را با یکی کردن محدودیت بررسی آنها ، بهبود بخشیده و GLP یک راه طبیعی برای انجام دادن این موضوع را نشان می دهد . وضعیت یک محاسبه PROLOG بوسیله لیست هدف جاری و مجموعه جاری انقیادها و اجبارها نشان داده می شود . با CLP این بسط داده می شود برای اینکه شامل مجموعه جاری از محدودیات روی متغیرها شود . اگر یک مرحله یگانه سازی یک محدودیت را بی اعتبار کند ، آن مانند یک شکست و خرابی مطرح شده در زمینه شکست یگانه سازی است .

بطور مشابه ، قوانین و حقایق گسترش یافته هستند برای اینکه هیچ یا بیشتر محدودیت های در حال افزایش را در جهت مستند کردن شامل شوند . برای مثال :

```
fib(N, M) : N > 2, % constraint
fib(N-1, M۱), fib(N-2, M2), M is Ml+M2.
```

وقتی یک محلی تطبیق داده می شود ، همانند انقیادهای اضافی که به لیست انقیادها افزوده می شوند و انقیادها برای سازگاری چک می شوند ، همانطور هم محدودیت های اضافی به انبار ذخیره محدودیت اضافه شده و محدودیتهای موجود برای سازگاری چک می شوند .

یک مجموعه از زبان ها بطورذاتی، بر اساس این مفهوم توسعه داده می شوند. تعدادی از انواع برجسته ی زبان ها PROLOGIII , CLP(R) , CHIP , CHR هستند. نوع محدودیت های پشتیبانی شده و نوع استنباط گرفته شده از انها از زبانی به زبان دیگر فرق دارد. برای مثال ، زبان CHR به برنامه نویس در جهت تعیین کردن چگونگی اداره ی محدودیات توسط به کار گیری قوانین اداره ی محدودیت اجازه می دهد.این قوانین در مورد انبار ذخیره محدودیت برای تطبیق دادن LHS ان ها اگاهی می دهند و هر کدام از

.

محدودیت های تطبیقی با یک فرم ساده جایگزین می شوند یا محدودیت های اضافی را برای اضافه کردن به مخزن محدودیت استنتاج می کند. برای مثال ، در این جا یک برنامه ی ساده در CHR برای تعیین کردن اعداد اول از ۱ تا N وجود دارد.

اینجا گزاره Primes مانند یک مولد عمل می کند که یک گزاره ی اول برای هر مقداری در محدوده ۱ تا N تولید می کند. قانون سوم یک قانون ساده است که به عنوان فیلتر ان گزاره هایی که اول نیستند را خارج می کند.
دسته ی دیگری از زبان های برجسته به وسیله ی زبان domain-public به نام oz نشان داده می شوند.
oz یک زبان چند نمونه ای است که از هم زمانی ، محدودیت ها و تابعی ،منطقی ، و روش های شی ء گرا پشتیبانی می کند. با مراجعه به برنامه نویسی محدودیت ،هم در گسترش و هم در جستجوبه طور صریح بنا می شود. ان از نمایش فاصله برای متغیر های محدودیت استفاده می کند و محدودیت های basic را به وسیله ی اصلاح کردن فواصل جذب می کند. برای محدودیت های non-basic ، انتشار دهنده ها ساخته شده اند که تغییرات در هر یک از متغیرهای داخل محدودیت را اگاهی می دهند و تغییر را به متغیرهای دیگر انتشار می دهند. واسطه ی جستجوی قابل برنامه نویسی کاربر اجازه می دهد که کاربر تعیین کند که جستجوی یک راه حل ، انجام گرفتنی باشد. انواع مختلفی از استراتژی های جستجو منجر به pay-off های مختلف در شرایط مؤثر می شود و غیره. نمونه هایی از استراتژی های جستجوی به کار گرفته شده رایج fail-first و naive هستند.

در این جا یک نمونه برنامه oz برای حل کردن مسئله مشهور Cryptarithmetic است.

OZ مدیریت محدودیت، برای تعریف کردن متغیرهای دامنه را منحصر می کند. FD یک کتابخانه از سازنده ها، برای تعریف کردن دامنه ی محدودیت ها است.FD مشخصات استراتژی جستجو را توزیع می کند.ff یک انتخاب از استراتژی fail-first را نشان می

دهد ، بنابراین متغیرهایی که بیشتر محروم شده هستند (کوچکترین دامنه) اول گرفته می شوند. نوعی از محرومیت ها برای تعریف متغیرهای دامنه در FD ساخته شده اند.

بر این اساس ما میتوانیم راه حل کاربردی ساخته شده مانند seachone و searchall پیدا کنیم. OZ هم چنین یک شاخه و بهینه ساز متناهی را مهیا می کند که ، یک تابع برای مقایسه ی دو راه حل را میدهد و راه حل بهینه را برمی گرداند.

هم چنین تلاش ها و مدل های دیگری برای زبان ها مخصوصا" برای توسعه ی زبان های سنتی همراه با قدرت محدودیت ها وجود دارند .CHR یک عضو خانواده CLP است که یک اجرا بر اساس جاوا دارد که می تواند با کاربرهای جاوا ترکیب شود. هر چند متغیرهای جاوا و CHR به طور کامل با هم نا مرتبط هستند واز این رو واقعا" یک توسعه برای CP نیست. تغییر پذیر بودن یک تلاش برای آوردن CP داخل چار چوب رویه ای است. هر چند این تلاش ها عمومی تر از CLP یا زبان های کامل نیستند.

کد منبع زیر مسئله cryptarithmetic را حل می کند. کد به وسیله ی Matthew Hammond در سال ۲۰۰۰ نوشته شده است.

```
The source code given below solves arbitrary cryptarithmetic
problem. The code is written by Matthew Hammond in 2000

% This Eclipse program solves arbitrary cryptarithmetic
problems.
% It works by setting up a single constraint for the summation.
% Thus it does not use any "carry" variable.
%

:-lib(fd).

solve(First,Second,Result) :-
        reverse(First,F),
        reverse(Second,S),
```

```
        reverse(Result,R),
        uniques(First,[],Ul), % The calls to the 'uniques/3' predicate
        % could be replaced with the merge/predicate.
        uniques(Second,Ul,U2),
        uniques(Result,U2,VarList),
        constrain(VarList,F,S,R),
        labeling(VarList).

uniques([],L,L).
uniques( [H | T],InList,List) :-
      uniques(H,InList),
       !,
      uniques(T,[H | InList],List).
  uniques([_ | T],InList,List) :-
       uniques(T,InList,List).

uniquein(_,[]).
uniquein(X,[H | T]) :-
       X \= = H ,
uniquein(X,T).

constrain(Vars,F,S,R) :-
        alldistinct(Vars),
        Vars :: 0..9,
        sumterm(F,FSum),
        sumterm(S,SSum),
        sumterm(R,RSum),
        FSum + SSum # = RSum.

sumterm( [] ,0).
sumterm([H |[]],H) :-
      H #\= 0.
sumterm([H | T] ,Result) :-
      T \= [],
      Result # = H + 10*R2, %% can. use '=' here instead of '# ='
      sumterrn(T,R2).
```

```
labeling([]) :- !.
labeling(List) :-
        deleteff(H,List,Rest),
        indomain(H),
         labeling(Rest).
```

۱۱-۶ زمینه برای پیشرفت آینده

اگرچه اکثر کار انجام گرفته شده است ، و زبان های مفید و چار چوب ها برای کاربرد عملی در دسترس هستند. زمینه CP سر آغازهای فراوانی برای پیشترفت پژوهش مهیا می کند.

رضا یتمندی محدودیت پاره ای: دستورات آماده شده به طور ویژه با کاربرد CP در مسائل AI وجود دارند. اغلب مشاهده شده که دامنه ی مسئله شامل نا سازگاری میان محدودیت هاست که یک توافق باید قبل از پیدا کردن یک راه حل انجام گیرد. هم چنین موارد زیادی وجود دارند ، در جایی که مجموعه ای از محدودیت ها ، یک مسئله تحمیلی را مطرح می کنند که راه حلی ندارد. در این موارد ، استراتژی خواستار مجزا کردن تعدادی از محدودیت های کم اهمیت است و آن ها را برای یافتن یک راه حل که محدودیت های مینیمال را نقص می کند یا باعث کمترین هزینه می شوند ، سست می کند.این فرم پی آمدها مرکز توجه رضایت مندی محدودیت پاره ای است.

جستجوی Backtrack-free : مانند حالت قبل ، جستجو یک جزء اساسی از سیستم های برنامه نویسی محدود شده است. انتخاب کردن متغیرها در هر مرحله از درخت جستجو انجام می شود ، و مقدار آن برای این که محدود شده باشد باید بر پیچیدگی درخت جستجو تأثیر گذارد.انتخاب بد منجر به خرابی در انتها می شود و Backtrack مجبور می شود که زود منشعب و شاخه شاخه شود. یک قسمت از کار ، جستجو کردن روابط بین انتخاب مقدار متغیر و پیچیدگی جستجو برای کاهش مقادیر مورد نیاز Backtraking . نتایج به دست آمده ی جستجوی backtvack-free برای مرتب

سازی حتمی تعدادی دستور العمل های درخت محدود شده وجود دارند.
محیط های پیشرفت : محیط های پیشرفت برای رسیدن به کمال هستند اگرچه GUI بر اساس واسطه های قابل دسترس برای اکثر تولیدات تجاری وجود دارند که بهتر از OZ هستند. بنابراین ، زمینه برای هویت بخشیدن زیر کلاس هایی که قابل انعطاف هستند و تابع ابزارهای کار آمد هستند وجود دارد.

فصل دوازدهم

کاربردهای هوش مصنوعی

اهداف

در پایان فصل، دانشجو با مفاهیم زیر آشنا می‌شود:

- بررسی تاثیر رهیافت های AI در تجارت الکترونیکی (B2B, B2C)
- بررسی نقش هوش مصنوعی در انتخاب و پیشنهاد کالا
- بررسی انواع رهیافت های AI در انتخاب و پیشنهاد کالا و محصولات(رهیافت های مبتنی بر ACF, KB,....)
- نقش AI در حل مسائل دنیای واقعی و در Enhancing Scalability
- نقش AI در مزایده و مذاکره Online
- نقش AI در تولید پاسخهای خودکار
- نقش AI در دسته بندی و قیمت گذاری خودکار کالا
- بررسی هستی شناسی در تجارت الکترونیکی
- نقش AI در گردشگری

نقش AI در صنعت و بررسی کاربردهای صنعتی AI

۱۲-۱ مقدمه

در این فصل ما تعدادی از کاربردهای امروزی هوش مصنوعی در زمینه هایی مانند بازرگانی الکترونیکی ، گردشگری ، پزشکی و صنعت را بررسی میکنیم.

در کنار کاربردهای عملی، تعداد زیادی از آنها منتظرند که در صحنه وطنی هندی خود

بهره برداری شوند . سیستم های متخصص مانند پردازش زبان هندی شامل پشتیبانی کردن ترجمه ، سیستمهای زمانبندی و برنامه ریزی ، سیستمهای بازیابی اطلاعات ، و غیره ، که استعداد و پتانسیل بالایی دارند.

۱۲-۲ AI در بازرگانی الکترونیکی

مشی های AI در پیشرفت هر دو سیستمهای بازرگانی الکترونیکی B2B و B2C مفید است. در زمینه بازرگانی الکترونیکی ، بیشتر توجه AI بر اداره B2C متمرکز شده است. در بازرگانی B2C ، AI بیشتر برای محصول انتخاب و پیشنهاد بکار گرفته میشود ، مانند مذاکرات، مزایده ها، حل کردن مسائل زمانبندی دنیای واقعی و enhancing scalability، تولید کردن خودکار پاسخ ها ، دسته بندی و قیمت گذاری کالا. در بازرگانی B2B ، AI اساسا" برای مدیریت رشته ذخیره شده (SCM) (بکار گرفته میشود. تعدادی از تکنولوژی های AI، مانند هستی شناسی در هر دو نوع زمینه بکار گرفته میشوند. جزئیات در ادامه بخش توضیح داده میشود.

۱۲-۲-۱ AI در بازرگانی الکترونیکی B2C

در این بخش ما مشی های مختلف AI را که در بازرگانی B2C مفید است را ارائه میکنیم.

AI در محصول انتخاب و پیشنهاد

AI در آگاهی دادن کاربران، در مواردی که آنها خواستار رسیدگی کردن یا خرید کردن در میان اینترنت هستند ، بکار گرفته میشود. این آگاهی در هدایت کردن محدوده ی بزرگی از توضیحات محصول ، سودمند میباشد. انواع مختلفی از مشی های محصول انتخاب و پیشنهاد مانند مشی های فیلتر کردن مشارکت خودکار شده (ACP) و، مشی های براساس دانش (KB) و مشی های ترکیبی و غیره وجود دارد.

- *ACP approaches :* این مشی ها داده راجع به اولویت های مصرف کننده ی قبلی یا الگوهای خرید را جمع میکنند و پیشنهادات مؤثر برای خریداران، که بر اساس تشابه الگوهای موجود است ، را آماده میکند. بعنوان مثال ، سیستم GroupLens

اخبار مقالاتی که استوار بر تشابهات بین مطالعه رفتار کاربر است را پیشنهاد میکند. این مشی در بسیاری از زمینه های دیگر مانند فرآورده های مصرف کننده و صفحات وب بکار گرفته میشود. و یک تکنیک فروش استاندارد در بازرگانی محسوب میشود. بعضی سیستمهای ACP حتی میتوانند یک دلیل و داده پشت یک پیشنهاد آماده کنند. یک مانع اصلی در ACP این است که، تا زمانی که تعداد زیادی از کاربران پروفایل هایشان را وارد کنند کارا نیست و یک تعداد کافی از موارد نرخ بندی در پایگاه داده قابل دسترسی هستند.

● *KB approaches :* این مشی ها بر پایه ی محصول براساس دانش هستند. تعدادی از آنها به صورت زیر شرح داده شده اند :

■ Case-based reasoning (CBR) approaches : اساسا" سیستمهای CBR اولویتهای کاربر را قبول میکنند، شبیه محصول عرضه شده از مورد اصلی ، دوباره بدست می آورد و آنرا برای کاربر پیشنهاد میکند. اگر هماهنگی درستی وجود نداشته باشد ، ممکن است کاربر اولویت هایش را تغییر دهد. مراحل بالا تا زمانیکه کاربر یک محصول را انتخاب میکند ، تکرار میشود. اکثر تکنیک متعارفی در کاربردهای بازرگانی CBR ، بازیابی نزدیکترین همنوع است.

■ Content-based recommendor approaches : این مشی ها وابسته به طبقه بندی ماشین بر اساس یادگیری هستند. برای مثال ، سیستم فیلتر کردن اخبار NewsDude ، حکایات اخباری که یک کاربر ممکن است علاقه به خواندن داشته باشد را پیشنهاد میدهد. این سیستم ها در اصل، دانش ماشین نظارتی را در جهت ساختن یک طبقه بند بکار میگیرد و طبقه بند برای فرق گذاشتن بین مواردی که موردعلاقه کاربراست و مواردی که مورد علاقه کاربر نیست ، بکار میرود.

● Hybrid approaches : مشی های ترکیبی در اصل ، یک ترکیبی از ACF و KB هستند. در بعضی سیستمها، ACF در مرحله ی پیش پردازش مورد استفاده قرار

میگیرند و بطور عمده سیستمها بر اساس دانش هستند. سیستمهای دیگر چک میکنند که آیا تعداد کافی از بازخوردها از کاربران قبلی وجود دارد. این تعداد مانند یک مقدار آستانه برای تصمیم گرفتن در مورد اینکه چه مشی ای بکار گرفته شده است، مورد استفاده قرار میگیرد. اگر مقدار کمتر باشد آنگاه یک مشی KB مناسب بکار گرفته میشود، در غیر اینصورت مشی ACF استعمال میشود. مقدار آستانه میتواند در تأثیر گذاری محصولات و معاملات بر روی یکدیگر تعیین شده باشد. بعضی سیستمها برای بکارگیری مشی CBR در مقابل ACF تلاش میکنند.

- علاوه بر موارد بالا ، مشی های دیگری مانند مذاکره کردن کاربر برای تعیین اعتبار طرح نسبت دودویی، شکل دادن مدلهای کاربر وابسته به مشاهده ی تصمیمات کاربر، پاسخ به پیشنهادات سیستم یا در میان مشاهدات منفعل، و دانش اولویتهای مشتری بوسیله مشاهده کردن انتخاب های مشتری از روی دستگاههای مرجوعی .

AI در مذاکره Online

در بازرگانی ، مذاکره یک فرایندی است که خریدار و فروشنده منابع را با بگارگیری ابزارها و تکنیکهای بازرگانی معامله میکنند. مذاکره متواند به صورت انتخابی یا غیرانتخابی طبقه بندی شده اند. همچنین به صورت مشارکتی یا رقابت طبقه بندی میشوند. تکنیکهای مذاکره اصولا از نوع رقابتی هستند. مشی های CBR ، بطور وسیع در مذاکرات بکار میروند. تعدادی مشخصه /روشها برای نشان دادن ی کفرایند مذاکره ی براساس CBR، عبارتند از: روش عامل فعال یا غیر فعال ، تغییرات تقاضاها یا تغییرات در جهت تقاضاها. مشی های براساس CBR تکنولوژی های عامل را بکار میگیرند. در این مشی ها ، عاملهای مذاکره استراتژی های episode برای بخشهای اصلاح ارائه برای یک توافق بکار میگیرند. مشی های دیگری برای مذاکره وجود دارند و براساس دانش Bayesian هستند. دانش Bayesian برای یاد گرفتن استراتژی مذاکره بکار گرفته میشود. تمام مشی های بالا از نوع توافقی هستند.

AIدر مزایده Online

امروزه، حدود ۲۰۰ سایت مزایده در اینترنت موجود است. اکثر مزایده های Online مزایده های متعارفی هستند. برای مثال ، مزایده های ماشین ها و کامپیوترها. تکنیکهای عامل قابل پیکربندی برای نشان دادن کاربران در مزایده های Online مخصوصا در Michigan AutionBot بکار گرفته میشوند. عاملها میتوانند بواسطه ی یک واسط Online ، پیکر بندی و راه اندازی شده و نمایش داده شوند.

کارکردن بطور همزمان با سایتهای چند مزایده ای یک وظیفه مشکل برای کاربران است. و اگر کسی بتواند ارزش محصول عرضه شده را پیشگویی کند ، او میتواند یک قیمت مطلوب را ارائه کند . مشی های AI برای اهداف این پیشگویی بکار میروند. این مشی ها استوار بر تکنولوژی های عامل هستند. آنها یک تعداد از عاملها ی پیشنهاد کننده و یک عامل رئیس برای هماهنگ کردن آنها را ثبت میکنند. عاملهای پیشنهاد کننده ی مختلفی برای سایتهای مزایده اختصاص داده شده اند. عاملها بطور همزمان قیمتهای یک شیء در چندین سایتهای مزایده را آگاهی میدهند و در میان خودشان (البته با کمک عامل رئیس) برای رسیدن به یک تخمین ارزش شیء موردنیاز ، همکاری دارند.

AI در حل کردن مسائل دنیای واقعی و در Enhancing Scalability

سیستمهای بازرگانی باید برای حل کردن مسائل واقعی ، توانا باشند. برای مثال ، در حوزه مسافرت ، جستجو کردن پروازها با محدودیت هایی مانند کرایه مسافر، زمان، ایمنی و break-journey و غیره در انتخاب پرواز راحت مهم است. برنامه ریزی، زمانبندی و مشی های بهینه سازی عموما برای حل کردن این نوع مسائل بکار گرفته میشود.

سرورهای بازرگانی همچنین باید Scalable باشند، بنابراین آنها میتوانند به وسیله تعداد زیادی از مشتریها بطور همزمان در دسترس باشند. تکنیکهای smart client

.

برای این هدف بکار میروند. Smart clinet ها استوار بر روشهای رضایتمند ی محدودیت هستند و راه حلهایی برای سیستمهای بازرگانی فهرست گونه مهیا میکنند. آنها حل کننده های مسئله خودمختار کارآمد هستند، و به اندازه کافی کوچک هستند که در زمان کوتاهی در سراسر اینترنت فرستاده میشوند. این تکنیکها همچنین برای حل کردن مسائل دنیای واقعی که در بالا ذکر شد، بطور وسیع بکار میروند.

AI در تولید کردن خودکار پاسخها

آماده کردن پاسخ خودکار برای برطرف کردن اکثر سؤالات مشتری نیازمند اشخاص حرفه ای برای جوابگویی سؤالات در سازمانها است. تکنیکهای Classification در توسعه دادن این نوع سیستمها بکار میرود.

AI در دسته بندی و قیمت گذاری خودکار کالا

اگر یک تولید کننده کالاهای مختلفی (محسوس یا نامحسوس) داشته باشد آنگاه او میتواند به راه های مختلفی آنها را دسته بندی و قیمت گذاری کند ، مانند قیمت گذاری هر کالا، دسته بندی خالص، و دسته بندی کردن مخلوط و غیره. بسته های مختلف دستگاههای ماهواره با فهرست قیمتهای مختلف ، برای ما در بازار امروزی قابل دسترس است، این یک مثال از این مورد است. تصمیمات دسته بندی و قیمت گذاری مناسب برای توسعه دادن پایگاه مشتری و همچنین افزایش دادن سود ، حیاتی است. مشی های Decision theoretic برای تولیدات در ساختن تصمیمات دسته بندی مفید هستند. تولید کننده مجبور است یک سری از دسته بندی های گذشته را برای بیشتر کردن سود و کارایی رضایت بخش خود مادامیکه در حال تغییر دادن دسته بندی های موجود است، معرفی کند. طبقه بندی سلسله مراتبی تولید ، همچنین در رسیدن به تصمیم مناسب مفید است.

علاوه بر موارد بالا ، تکنیکهای AI برای تعدادی از هدفهای بازرگانی B2C مفید هستند. برخی از آنها در اینجا ذکر شده اند.

- AI در توسعه حوزه ی ابزارهای پایگاه داده مستقل مفید هستند.ابزارها برای

دسترسی اطلاعات، بویژه برای فهرست های تولید الکترونیکی طراحی شده اند. آنها استوار بر روشهای preference-based navigation هستند که نسبت به بکارگیری text-based بصورت سنتی یا تکنیکهای جستجوی query-based سریعتر هستند.

● تکنیکهای AI همچنین برای توسعه ی عاملهای خریدار مفید است. اکثر سرویسهای بازرگانی مانند سرویسهای فروشندگان تحت نظر هستند زیرا هدف هایشان دادن سرویس و کالا به کاربران است. عاملهای خریدار برای خریداران نسبت به فروشندگان سریعتر کار میکنند. برخی از مسئولیت های چنین عاملهایی عبارتند از: آگاهی دادن مشتریان از عمل متقابل پیچیده بین اولویتهای مشخص شده و وضعیتهای غالب بازار ، مهیا کردن آنالیزهای تشخیص دهنده برای پشتیبانی تصمیم ، و استفاده کردن از هستی شناسی برای کمک کردن به کاربران برای اینکه سؤالاتشان را تنظیم کنند.

۱۲-۲-۲ AI در بازرگانی B2B

تکنیکهای AI بطور وسیع در مدیریت رشته موجودی بکار میرود. یک رشته موجودی مجتمع ، تجارتهایی را برای تسهیم کردن اطلاعات بی درنگ بوجود می آورند و عمدتا هزینه حمل موجودی را کاهش میدهند. این برای بازرگانی B2B بسیار مهم است . تعدادی از مشی های حل شدنی مسائل SCM بر اساس AI ، قابل دسترس هستند. اکثر آنها براساس عامل هستند و هر عامل ، مسئول یک یا چند فعالیت SCM است. اکثریت مشی های بر اساس عامل ، با مسئله مانند یک مسئله محدود رضایت بخش متمرکز شده رفتار میکنند. برخی از مشی های بر اساس عامل ، اشیاء ساخته شده توسط انسان را مدلسازی یا سازماندهی میکنند و برای هدفهای شبیه سازی بکار میروند. حتی سیستمهای دیگر، عاملها را برای قراردادهای فرعی بکار میگیرند. در سیستمهای قرارداد فرعی، هماهنگی رشته موجودی تبدیل به رشته موجودی مجازی در یک سیستم چند عاملی، از طریق فرایند مذاکره بین عاملهای نرم افزار، شده است.

۱۲-۲-۳ هستی شناسی در بازرگانی

تعدادی از سیستمهای بازرگانی بوسیله سازمانهای مختلفی برای یک نوع ازهدفهای تجاری توسعه داده شده، پیکربندی شده، و بکارگرفته شده اند.طبقه بندی نمونه های تجاری، فرآیندها و ساختارهای دانش برای موفق شدن در زمینه سرمایه گذاری حیاتی است. مانع اصلی برای موفقیت بازرگانی ، نیازمندی سیستمهایی است که قادرند اطلاعات را بطور عمده سهم برند. هستی شناسی در پیروزی این هدف، حیاتی است. آنها بطور بنیادی مقابل واژگان برای روابط پایه ای و معنی شان می ایستند. هدف یک هستی شناسی برقراری ارتباط بین سیستمهای کامپیوتر و اشخاص، از طریقی که مستقل از تکنولوژی های سیستم فردی، و طراحی اطلاعات و حوزه کاربردی است میباشد.هستی شناسی برای همکاری، ارتباط عامل با عامل، مدیریت دانش و سیستمهای مختلف پایگاه داده بسیار مهم است.

۱۲-۳ AI در گردشگری

پیشرفت نمائی در زمینه Word Wide Web ، مفهوم های سنتی در بسیاری از زمینه ها مانند تجارت، بازرگانی، یا گردشگری را تغییر داده است. این تغییرات ناشی از دلایل مختلفی میباشد . تعدادی از کاربران به اینترنت وصل میشوند، مقدار زیادی اطلاعات در آن ذخیره شده است، و دسترسی به این اطلاعات برای کاربر آسان است. مشخصه هایی مانند پست الکترونیکی و انتقال امن اطلاعات ، باعث جستجوی اخبار به صورت باز و آزاد و زمینه های کاربردی شده است. بطور رایج ، انجام دادن معاملات تجاری درweb امکانپذیر است. این بخش چگونگی کاربرد این تکنولوژی های جدید را در زمینه گردشگری شرح میدهد.

امروزه پیداکردن تعداد زیادی از شرکتها که از web برای ارائه و فروش محصولاتشان استفاده میکنند ، شدنی است. در زمینه گردشگری پیدا کردن آژانسهای مسافرتی (Amadeus, FourAirlines, Halcon Viajes) وسایل حمل ونقل ، هتل یا

کمپانی های ماشین که محصولات مختلف را در وب ارائه میکنند ، امکانپذیر است. در واقع یافتن قیمتهای معامله ، تنها برای کاربرانی که در میان وب قصد خرید دارند ، مقدور است. مشورت کردن در مورد قیمتها ، جداوی زمانی، خرید بلیط ها (بلیط های هواپیما یا قطار ، کرایه ماشین و هتل و اتاق و غیره ...) برای کاربر امکانپذیر است. بنابراین زمینه گردشگری بوسیله مذاکرات الکترونیکی که در میان اینترنت وجود دارد، تغییر کرده است. اما مسائل جدی در بازیابی، اداره کردن و استفاده مجدد از اطلاعات ذخیره شده در وب وجود دارد. این مسائل بصورت زیر خلاصه شده اند:

- تعدادی از سایتها که اطلاعات پنهانی ارائه میدهند.
- اطلاعات بکار گرفته شده توسط شرکتها ممکن است بعد از مدتی تغییر کند.
- دسترسی به منابع اطلاعات همبشه امکانپذبر نیست.
- ارائه ناهمگن اطلاعات (ممکن است اداره کردن یا فهمیدنش برای کاربر مشکل باشد).
- معاملات ارائه شده توسط شرکتها ممکن است بعد از مدتی تغییر کرده باشد.

در واقع برای حل کردن این مسایل ، سیستمهای بسیاری وجود دارند که اطلاعات فراهم شده از وب را بطور کارامد استخراج و فیلتر میکنند و سپس ارائه میکنند. اما بسیاری از این سیستمها (موتورهای جستجو ، موتورهای متاجستجو، webBots, spiders و غیره) روی تعدادی از اطلاعات برای بازیابی آنها متمرکز شده اند (Mediators). متأسفانه این سیستمها برای حل کردن تمام مسائلی که قبلا ذکر شد توانا نیستند.

توسعه یافتن تواناییهای بالقوه ی وب ، نیازمند پیدا کردن مشی های جدید در جهت اجازه یافتن برای طراحی و ساختن سیستمهایی است که بطور خودکار اعمال زیر را انجام میدهند:

- جستجو کردن وبازیابی اطلاعات مفید
- اداره و استدلال کردن با راه حلهای مختلف
- وفق دادن با پروفایل کاربر، برای رسمی کردن پاسخ ها

.

● افزایش بازده ی بکارگیری دانش یادگرفته شده توسط عاملهای دیگر (انسان یا نرم افزار)

آن برای اثر متقابل بین عاملها ضروری است.

۱۲-۳-۱ حوزه گردشگری الکترونیکی

حوزه گردشگری الکترونیکی به این صورت تعریف میشود: " آنها منابع مختلف الکترونیکی دسترس پذیر برای یک عامل سفر (انسان یا نرم افزار) که میتواند برای برنامه ربزی یک سفر بکار گرفته شود را دارند."

یک عامل سفر الکترونیکی باید توانایی اداره کردن تمام اطلاعات موردنیار برای برنامه ریزی کردن مواردی که برای مسافرت کاربر لازم است را داشته باشد. در این نوع حوزه ، عامل برنامه ریزی، به نشان دادن چندین شهر ، وسایل حمل و نقل ، مکان های کرایه ای و غیره نیاز دارد. و عامل نیاز به دانستن چگونگی سفر بین شهرها و و فرودگاهها و ایستگاه های قطار و نیز کرایه یک ماشین یا رزرو کردن یک اتاق در هتل دارد. شکل ۱۲,۱ ارائه گرافیکی از یک مثال در این حوزه را نشان میدهد، که یک کاربر میتواند وسایل حمل و نقل و هتلهای مختلفی را برای رسیدن به هدف خود داشته باشد.

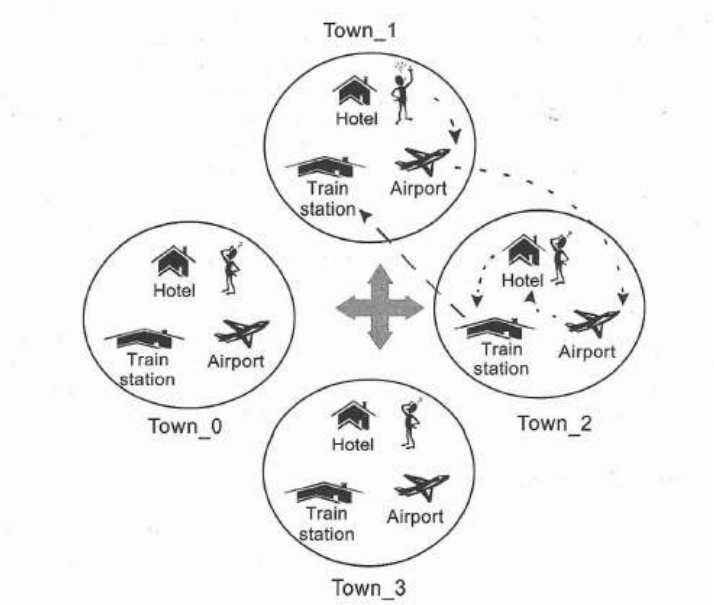


Figure 12.1 Graphical description of the electronic-domain.

بیشتر نکات مهم در مدیریت یک سفر عبارتند از:

۱) حرکت کردن از شهر مبدا به شهر مقصد

۲) مسکن در مقصد

۳) وسایل نقلیه موجود در شهر مقصد

٤) بازگشت به شهر مبد (یا شهر دیگر)

برای اداره کردن مورد (۱) وسایلی مانند هواپیما ، قطار، یا اتوبوس نمونه هایی از وسایل حمل ونقل هستند که تاکنون مطرح شده اند. همچنین کاربر ممکن است برای حرکت به سمت فرودگاه ، ایستگاه قطار یا ایستگاه اتوبوس به قطار و اتوبوس محلی و یا تاکسی نیاز داشته باشد. این بدان معنی است که عامل همچنین نیار به تصمیم گیری دارد که از چه وسیله ی محلی استفاده کند. این یک بخش مهم و پیچیده ی مسایل است زیرا تعدادی از راه حلهای ممکن ، بصورت نمایی

.

رشد میکنند. در مورد وسایل نقلیه ی موجود در شهر مقصد، امکاناتی مانند اجاره یک ماشین ، یا بالا بردن اطلاعات کاربر درباره وسایل حمل ونقل عمومی (اتوبوس و مترو) وجود دارد.

پی آمد مهم دیگر این است که راه حلهای مختلف بسیاری برای یک مسئله معین موجود است و همه ی آنها باید مطابق با چندین پارامتر طبقه بندی شوند. برای مثال، راه حلها میتوانند مطابق با هزینه ، میزان زمان برای اتمام سفر ، یا اولویتهای کاربر ، طبقه بندی شوند. (برای مثال ، اگر یک مشتری تصمیم گیرد با هواپیما سفر نکند.) .

عامل سفر باید همچنین برای استدلال کردن بین تمام اطلاعات ، قادر باشد، زیرا بیشتر اوقات تمام اطلاعات در دسترس نیست یا بخشی از اطلاعات لازم ممکن است خراب باشد. آن وضعیت نباید کار حل کردن مسایل را متوقف کند و باید سعی بر جستجوی راه حلهای تناوبی کند. در حوزه ی گردشگری الکترونیکی، اصل منابع اطلاعات الکترونیکی بوسیله وب قابل دسترسی خواهد بود. بنابراین عامل سفرنیازمند مدیریت همراه با اطلاعات ذخیره شده در آن است.

برخی مشی های مناسب بکار رفته در وب، عاملهای وب (مخصوصا softbots و spiders)، و سیستمهایی که قادر به جمع آوری و فیلتر کردن داده ای که وب ارائه میکند دارند. سیستمهایی که موتورهای جستجو یا موتورهای متاجستجو نامیده میشوند موفقیتهای زیادی در سالهای اخیر داشته اند، زیرا آنها به کاربر اجازه میدهند که به آسانی اطلاعات مفیدی را از وب بدست آورد.

مشی دیگر، SIMS یک پیاده سازی از یک نوع سیستمهاست (Mediators نامیده میشود) که سعی بر یکی کردن منابع اطلاعات ناهمگن در جهت دستکاری کردن داده های آنها برای رسیدن به راه حل های جدید دارند. سیستمهای دیگر مانند ARIADNE ، سعی بر بدست آوردن دانش ناهگن (بدست آمده از منابع مختلف) برای بکاربستن یک فرایند برنامه ریزی شده دارند.

امروزه، یافتن سیستمهایی که اجازه ی تعریف کردن و اثبات عاملها با مهارتهای مختلف ، ارتباط زبانی و غیره را میدهند امکانپذیر است. هدف پروژه Zeus آسان کردن پیشرفت سریع کاربردهای چند عاملی جدید بود که آنرا توسط انتزاعی کردن داخل یک بسته که حاوی حقایق معمولی و تعدادی عناصر اصولی موجود در سیستمهای چند عاملی است، انجام میداد. بسته های دیگری مانند JATLite وجود دارند که یک بسته از کلاسهای جاوا و برنامه هایی که به کاربران در جهت ساختن سریع سیستمهای جدید عاملهای نرم افزار اجازه میدهند ، هستند که در سراسر اینترنت به ترتیب اجرای یک محاسبه توزیع شده ، ارتباط برقرار میکنند.

۱۲-۳-۲ مثال: نقشه سفر

در اینجا مشی های مختلفی برای کار با اطلاعات ذخیره شده در وب وجود دارد. در این بخش تعدادی از آنهایی را که در تکنیکهای هوش مصنوعی برای ساختن سیستمهای خودمختار و هوشمند بکار میروند را تحلیل میکنیم.

عاملهای هوشمند یک نمونه ی جدید برای توسعه دادن کاربردهای نرم افزار هستند. بطور رایج عاملها مرکز قوی برای جذب کردن روی بخشی از بسیاری از زیر رشته های علم کامپیوتر و هوش مصنوعی هستند. کلید انتزاع بکار رفته شده در این سیستمها، مفهوم عامل است. خلاصه کردن صفات اصلی عامل کار سختی است زیرا هیچ توافق دقیقی روی اینکه یک عامل چیست ، وجود ندارد. اما بیشتر پژوهشگران در موارد زیر توافق دارند:

- هر عامل یک تعداد اطلاعات ناقص دارد یا قابلیتهای درستی برای حل کردن تمام مسایل را ندارد.
- یک سیستم کنترل جهانی ندارد.

- داده متمرکز شده نیست،بنابراین تمام عاملها باید داده را تسهیم کنند.
- سیستم اجرایی غیرهمزمان است، هر عامل مادامیکه سؤالات را دریافت میکند ،میتواند در حال کار کردن باشد
- تعریف کردن و بکاربردن مشخصه هایی مانند نمایندگی، خودمختاری، رفتاراجتماعی، پویایی، سازگاری و غیره مقدوراست.

یک سیستم بر اساس عامل ممکن است شامل هر تعداد غیر صفر از عاملها باشد. یک سیستم چند عاملی باید شامل چند عامل که روی هم تأثیر میگذارند باشد، و بنایراین واضح است که سیستم چند عاملی از یک سیستم تک عاملی پیچیده تر است.

سیستمهای چند عاملی (MAS) یک زیر رشته از هوش مصنوعی توزیع شده هستند که به گروههایی از عاملهای هوشمند که برای حل کردن مسئله بصورت مشارکتی تلاش میکنند ، رسیدگی میکند. موفقیت MAS ناشی از چند دلیل میشود (ما تنها به علتهای مناسب اشاره میکنیم):

آنها در حل کردن مسایل بزرگ قادر هستند ، مخصوصا در جاییکه سیستمهای کلاسیک موفق نباشند.

آنها به سیستمهای مختلف برای کار کردن بطور پیوسته و مشارکتی اجازه میدهند.

آنها راه های مؤثری را که اطلاعات ، میان مکانهای مختلف توزیع شده اند را فراهم میکند.

آنها اجازه استفاده مجدد از نرم افزار را میدهند، پس قابلیت انعطاف زیادی برای قبول کردن تواناییهای مختلف عامل، برای حل کردن مسایل وجود دارد.

در شکل ۱۲,۲ یک ارائه گرافیکی از یک MAS کلی نشان داده شده است. در این ارائه ، یک مجموعه ای از عاملهایی که میتوانند بین خودشان برای رسیدن به راه حل مسئله ارتباط برقرار کنند و با هم مشارکت داشته باشند ، وجود دارد.

در طی سالهای اخیر ، تعدادی از افراد ، یک مجموعه از سیستمهایی با چندین عامل توسعه داده اند که سعی بر حل کردن یک مسئله خاص دارد. برای رسیدن به این هدف، انها تکنیکهایی مانند رقابت (در جاییکه عاملها برای بدست آوردن منابع سیستم

با هم نزاع میکنند) یا مشارکت (در جاییکه عاملها برای رسیدن به راه حل با هم همکاری میکنند) بکار میگیرند.

نقشه سفر یک مشی چند عاملی توزیع شده و مشارکتی برای حل کردن مسئله در محیط های پویا (برای مثال : وب) است و در حوزه ی گردشگری الکترونیکی بکار میرود. ما تصمیم به استفاده از تکنیکهای MAS داریم زیرا این نوع سیستم خیلی قابل انعطاف است و سازش بالایی دارد. ما از تکنیکهای wrapping برای اضافه کردن سیستمهای برنامه ریزی کلاسیک مانند Prodigy 4.0 داخل یک سیستم چند عاملی استفاده میکنیم.

۱۲-۳-۳ معماری نقشه سفر

نقشه سفر یک مشی MAS است که انواع مختلفی از عاملها را مجتمع میکند . که خلاصه وار به صورت زیر است:

- عامل کاربر: این عامل به سؤالات کاربر توجه میکند و راه حل را نشان میدهد. مسئله را حل میکند و یک ارائه ی مطلق را فراهم میکند. متعاقبا برای یک راه حل نیاز به عامل نقشه کش دارد. عامل کاربر مهارتهای مختلفی مانند ارتباط با عاملهای نقشه کش و کاربران ، یا فراگیری پروفایلهای لازم کاربران برای رسمی کردن پاسخ سیستم ، دارد.
- عامل نقشه کش: هدف اصلی عامل نقشه کش استدلال درباره ی عاملهای کاربر و مسایل عاملهای نقشه کش نوع دیگر، و کشف کردن یک مجموعه از راه حلهای ممکن است.عاملهای نقشه کش مهارتهای مختلفی مانند ارتباط (با عاملهای مختلف در سیستم) ، برنامه ریزی (طرح استدلال اصلی اش) ، و فراگیری (بکارگیری تکنیکهای CBR برای فهرست و طبقه بندی هر نقشه ی ذخیره شده) دارند.
- WebBot : این عامل برای فراهم کردن اطلاعات موردنیاز از اینترنت، در جزئیات سیر میکند. راه حلهای جزئی مختلفی توسط عاملهای وب ، با عاملهای نقشه کش برای فراهم کردن یک راه حل تفضیلی برای سؤالات عامل کاربر ، ترکیب شده اند.

در شکل ۱۲,۳ یک ارائه گرافیکی از نقشه سفر نشان داده شده است . سیستم بوسیله ی

یک مجموعه ای از عاملها ساخته شده است که میتوانند در میان خودشان برای رسیدن به راه حل مسئله ارتباط برقرار کنند و با هم مشارکت داشته باشند.

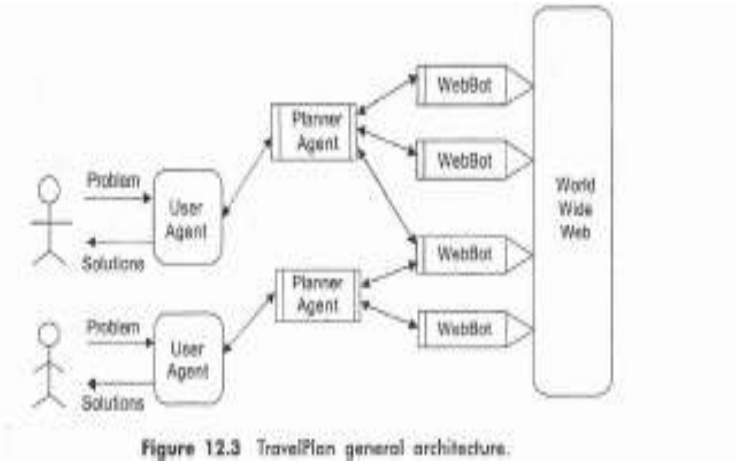


Figure 12.3 TravelPlan general architecture.

نقشه ی سفر یک معماری مشارکتی دارد درحالی که عاملها نیاز به مشارکت برای رسیدن به راه حل ها را دارند. عاملهای مختلف نیاز به تسهیم کردن دانش و مهارت ها برای کامل کردن راه حلهای مطلق بدست آمده بوسیله عامل نقشه کش را دارند. همچنین بدست آوردن یک راه حل کامل از سیستم عامل دیگر ، شدنی است. موفقیت نقشه سفر نیاز به دو مشخصه دارد، تسهیم کردن دانش برای بدست آوردن راه حل های جدید (یا بازیابی راه حلهای ذخیره شده ی قدیمی) و آگاهی از اولویت های کاربران برای یافتن و مرتب کردن اکثر راه حلهای مفید برای خودش. در بخش بعدی یک قالب برای تسهیم کردن دانش تحلیل شده است.

۱۲-۳-٤ اطلاعات تسهیم شده میان عاملها

عاملها در نقشه سفر ، یک نمایش عادی برای دانش بکار میگیرند. این مشخصه به ساده کردن فرآیندهای تسهیم شده و استدلال با دانش اجازه میدهد. ارتباط میان عاملها از performative استفاده میکند. یک performative یک دستور مطلق برای عامل دیگر ذخیره میکند . برای ارتباط بین دو عامل سیستم ، یک زیرمجموعه از قالب

KQML بکار گرفته شده است. در شکل ۱۲,٤ نمایش برای برخی performative ها نشان داده شده است: request , inform , accept , reject .

Performative اولی که request است، بوسیله یک عامل نقشه کش به WebBot برای پرسیدن یک سفر هوایی بین Madrid و Zaragoza فرستاده شده است. دومین Performative که inform است از WebBot برای عامل نقشه کش درخواست میشود. Performative های accept و برای reject همکاری و مذاکره بین عاملهای مجاز بکار گرفته میشود.

با بکارگیری پروتکل ارتباطی ، عامل نقشه کش راه حلهای نیمه کاملی از مسئله تعریف کاربر و از راه حلهای مطلق تعیین شده توسط نقشه کش بدست آمده را، کامل میکند. بخش بعدی چگونگی عملیات پروتکل دو عامل سیستم را نشان میدهد.

۱۲-۳-۵ پروتکل ارتباط

عاملهای نقشه سفر نیاز به پیاده سازی یک زبان ارتباطی دارند. شکل ۱۲,٤ یک مثال از این پروتکل را نشان میدهد. عامل کاربر (UA) فرایند را با فرستادن یک توضیحی از مسئله کاربر برای عامل نقشه کش (PA) آغاز میکند. عامل نقشه کش ، مسئله را با بکارگیری نمونه استدلالی اش و همکاری با عاملهای دیگر مانند WebBot (wb) ها را حل میکند. مثال ، انواع مختلفی از performative ها مانند request , inform, accept را نشان میدهد.

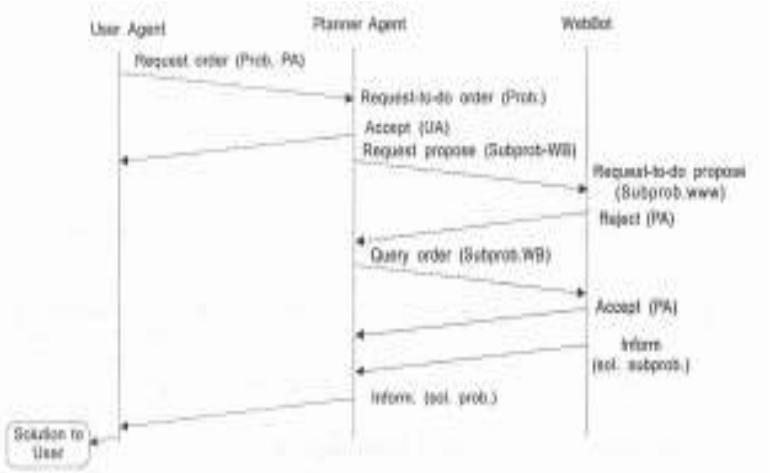


Figure 12.4 Example of communication protocol in TravelPlan

در فرایند ارتباط هر عامل، پیام تولیدی performative accept را تصدیق میکند (بوسیله هر عامل دیگر فرستاده شده) . هنگام پردازش ، آنها یک پیام پاسخ، برای فرستنده میفرستند و کار شروع میشود. اگر عامل برای اجرای کار آماده نباشد ، یک performative reject تولید میکند. هنگام پردازش، آن یک پیام برای عامل میفرستد. وقتی یک request performative فرستاده شد آن ممکن است انواع مختلفی از اعمال را تولید کند. با بکارگیری request.propose(prob.agent) فرستنده اجرای یک کار را تنظیم میکند (بدون مذاکره). اما اگر عامل یک request.propose(prob.agent) بفرستد آنگاه داشتن مذاکره با عامل دریافت کننده میسر است.

در شکل ۱۲-٤ عامل نقشه کش یک پیشنهاد برای مشارکت به webBot میفرستد، و این عامل درخواست را رد میکند. وقتی عامل نقشه کش واقعا به اطلاعات نیاز داشته باشد یک request.order برای اطلاعات خواسته شده فراهم میکند. وقتیکه اطلاعات WebBot پردازش شده باشد، عامل نقشه کش میتواند راه حل را کامل کند و برای عامل کاربر بفرستد که در نهایت برای کاربر نشان داده خواهد شد.

نقشه سفر مانند نوعی MAS یک مجموعه از واسط های کاربر گرافیکی (GUI)

برای ارتباط با کاربران بکار میبرد.
عامل کاربر واسطه های کاربردی مختلفی برای ارتباط با کاربر دارد. واسط اولی برای ورود مسئله ی کاربر بکار رفته است، و دومی اختیاری است و برای تعریف کردن اولویتهای کاربر بکار میرود. پروفایل کاربر که بوسیله عامل کاربر ذخیره شده است، مسئله را تحلیل میکند و به یک عامل نقشه کش برای پرسیدن درباره یک راه حل متصل میشود. هنگامی که سیستم ، مسئله نمونه را حل میکند ، عامل نقشه کش راه حلهای نمونه را از اکثر راه حلهای ممکن برای کاربر برمیگرداند، و عامل کاربر این راه حلها را برای کاربر نشان میدهد. سیستم یک مجموعه ای از راه حلهای ممکن را رد میکند برای اینکه بازده را زیاد کند و تنها یک زیرمجموعه از راه حلها برای کاربر نشان داده میشود. نقشه سفر یک راه حل مخصوص پیشنهاد میکند اگر این راه حل با اولویت عالمانه از سوی کاربر هماهنگ باشد. برای انجام این ، عامل کاربر که به کاربران توجه دارد، مشخصه های اصلی از راه حلهای ذخیره شده قدیمی توسط کاربر را استخراج میکند و آنها را برای طبقه بندی تمام راه حلهای ممکن بکار میگیرد. با بکارگیری پروفایل کاربر و راه حلهای مختلف پیدا شده توسط ، یک زیرمجموعه از راه حلها برای کاربر نشان داده میشود.

۱۲-٤ AI در صنعت

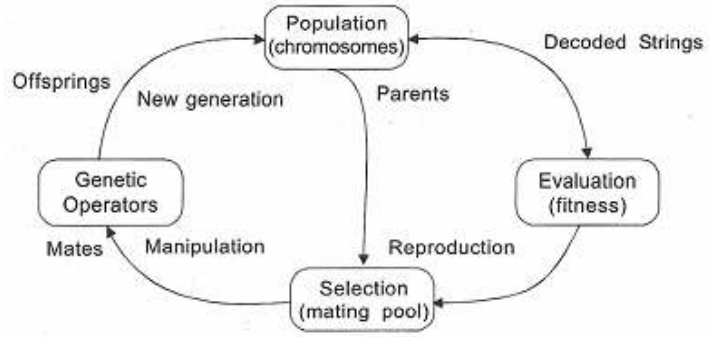
علاوه بر شبکه های عصبی ، چند تکنیکهای AI دیگر وجود دارند که نه تنها در پژوهش بلکه در کاربردهای صنعتی نیز محبوبیت کسب کرده اند. منطق fuzzy یک تکنیکی است که میتواند در شرایط کاربردهای صنعتی بهمان اندازه موفقیت آمیز مطرح شود.
پروفسور عسکری زاده مادامیکه مجموعه های fuzzy را فراخوانده ، یک متدولوژی برای رفتار کردن مبهم پیشنهاد کرده است. مجموعه های fuzzy مسائل مجموعه ی پیچیده را برطرف میکنند ، که به جای تنها " true " یا "false " ، " yes " یا " no" یا "۱" یا "۰" ، یک عضویت از درجه "۰ تا ۱" میتواند به این مجموعه اختصاص داده

شود. منطق fuzzy بطور موفقیت آمیز در شناسایی الگو ، سیستمهای پشتیبانی و کنترل تصمیم کاربرد داشته است. انتخاب بکارگیری منطق fuzzy در کنترل کردن ، امکانی برای توسعه دادن یک کنترلر همراه خطوط زبان شناسی است. در کاربردهای منطق fuzzy ، قواعد زبان شناسی توسعه یافته اند بهمان اندازه که بر وضعیت های متداول استوارند مکان بعدی فعالیتها میتوانند تنظیم شده باشد.

یکی از پیشرفتهای اخیر در AI الگوریتمهای تکوینی (GA) است. GA یک فرمی از جستجو و تکنیک بهینه سازی است . GA بوسیله Prof.j.Holland و همکلاسیهایش در دانشگاه میشیگان در اواسط دهه ٦٠ توسعه داده شده است. الهام بخش اصلی در تنظیم GA ها مشاهده ی تکامل دستگاه ها یی از سیستم طبیعی بود که بخوبی کار میکرده است. مشخصه های مهمی مانند : Self-guidance , Self-repair و تکثیر که در سیستمهای زیستی وجود دارد در قاعده سازی GA ها اتخاذ شده است. شکل ۱۲,۵اساس گرداننده ها و چرخه GA را نشان میدهد. همراه با این معیارهای مثبت ، یک مجموعه منحصر به فرد از تکنیکهای جستجو، GA ها را بصورت یک جستجوی قوی و تکنیک بهینه سازی میسازد.

مجموعه ای از تکنیکهای جستجو که GA ها قبول کرده اند از دیگر تکنیکهای بهینه سازی مانند hill climbing و calculuse-based ، تفاوت دارند و در زیر این تفاوت ها آمده است:

- GA ها با رمزگذاری پارامترها کار میکنند.
- GA ها در باره جمعیتی از موضوعات جستجو میکنند و از روی یک موضوع تک جستجو نمیکنند.
- GA ها اطلاعات تابع هدف را بکار میگیرند و بصورت اشتقاقی یا دانش کمکی بکار نمیگیرند.
- GA ها قواعد انتقالی احتمالی بکار میگیرند و از قواعد قطعی استفاده نمیکنند.



کاربردهای موفقیت آمیز بسیاری از GA ها در زمینه هایی مانند مسیریابی Wire ، طراحی VLSI ، بازی کردن، شناسایی چهره و... وجود داشته است.

یک گرایش در حال رشد در جهت ترکیب کرن چند تکنیک AI برای توسعه دادن یک ابزار قوی وجود داشته است. یک نمونه ، ترکیب تکنیک منطق fuzzy و شبکه های عصبی است که منجر به برچسب مناسبی برای سیستمهای neuro-fuzzy میشود. این حقیقتی را باعث میشود که تکنیک fuzzy در طبیعت ساخت یافته است مادامیکه یک فایده برای بشر ، قواعد را همراه خطوط زبان شناسی گسترش میدهد. اما آن قابلیت یادگیری ندارد. از طرف دیگر، شبکه های عصبی، یک قابلیت یادگیری ذاتی را فراهم میکنند اما ساخت یافته نیستند. ترکیب این دو تکنیک یک حوزه ی جدید از پژوهش در AI را باعث میشود. چند ترکیب از نمونه های شبکه عصبی با منطق fuzzy پیشنهاد داده شده است. برای مثال، تابع پایه شعاعی (RBF) شبکه عصبی اجرا شده مانند یک کنترلر neuro-fuzzy (NFC) است و شبکه عصبی برگشت عمومی (GRNN) مانند یک پیشگو در کنترل neuro-fuzzy بکار برده میشود که پیکربندی سیستم در شکل ۱۲-٦ نشان داده شده است. تکنیک میتواند میزان سازی خودکار را از NFC مهیا کند که استوار بر یک مدل پیشگویی Online درباره تشخیص دستگاه بوسیله GRNN است.

با بکارگیری این معماری ، سیستم توافقی قادر به واکنش سریع نسبت به تغییرات در

مجموعه است. این سیستم توافقی مجتمع شده پیشنهادی ، خیلی قوی است حتی وقتی که یک تغییر عمده در پویایی دستگاه، موقعی که با یک کنترلر پیشگویانه تعمیم یافته مقایسه شده ، وجود دارد.
بدست آوردن فواید کمی و کیفی از NFC که میتواند مربوط به دسترسی به حوزه ی دانش بشری باشد، فقط نتیجه ی قواعد کنترلی، در این مشی وفق داده شده اند. اگر یک مجموعه ناهنجار از فعالیتهای کنترل برای اولین بار در دسترس باشند، آن میتواتد همچنین در سیستم کنترل ثبت شده باشد مادامیکه در آن زمان بطور اتوماتیکی بسیار خوب وفق یافته باشد. در کنار توافق پیوسته ، متدولوژی میتواند مانند یک الگوریتم Self_organizing دیده شود، بدلیل اینکه سیستم قادر به یادگیری دانش کنترلی لازم بدون هیچ یک از فعالیتهای کنترل تعریف شده در طی مرحله اول است.

در اصل متدولوژی الهام بخش از نظر اساسی یک تکنسین یا متصدی با حداقل دانش کنترل،عملیات کنترلر را وفق خواهد داد، که استوار بر مشاهده اش بر واکنش سیستم است. توافق سیستم سه روند بصورت زیر را درگیر میکند: ۱) وفق دادن عملیات یا قواعد کنترل. ۲) دستکاری کردن عملیات کنترلی. ۳) تنظیم خروجی کنترلر . روش وفق دادن یک روش non-gradiant descent .بر پایه واکنش سیستم پیشگویانه است، قادر به self-organize عملیات کنترلی مرحله اول است. برنامه دستکاری کننده میتواند به کاهش تهاجم قواعد کنترلی خاص برای تثبیت کردن واکنش برای مجموعه نکات، کمک کند. اگر خروجی مناسب کنترلر برای ترکیب بطور ذهنی مشکل باشد ، برنامه تنظیم کنترلر میتواند بکار برده شود.

همچنین تلاشهایی در ترکیب تکنیک fuzzy و سیستمهای خبره بوجود آمده است که منجر به نمونه ای میشود که به سیستمهای fuzzy-expert اشاره میکند. اما سیستمهای fuzzy-expert موفقیت سیستمهای neuro-fuzzy را نداشته اند. همچنین پیشرفتهایی در ترکیب تکنیک fuzzy و GA بوجود آمده است اما عمومی نشده است.

بدلیل ترکیب چند تکنیک AI برای تشکیل یک سیستم ترکیبی ، یک سری از فواید و معایب وجود دارد.

۱۲-٤-۱ کاربردهای صنعتی AI

از جمله اولین کاربردهای موفقیت آمیز AI در صنعت ، در زمینه های دانش ماشین نمادین یا کاربردهای سیستمهای خبره بودند که در اواخر دهه ٦۰ و اوایل دهه ۷۰ آغاز شدند. با وجود آنکه تحقیق در شبکه های عصبی زودتر شروع شده بود، کاربردهای دنیای واقعی از چنین تکنیکهایی نسبتا خاموش شده اند که بدلیل صفت تکراری شان است که نیاز به سخت افزار قوی داشتند که در آنزمان وجود نداشت. دو مثال از کاربردهای اولیه ی سیستمهای خبره عبارتند از : MYCIN برای تشخیص بیماری های عفونت خونی که در دانشگاه Stanford در سال ۱۹۷۲ ، و CADACUES برای تشخیص بیماریهای درونی که در دانشگاه Pittsburgh در سال ۱۹۷۰ توسعه داده شده اند. اولین کاربرد تجاری از سیستم خبره XCON است که برای پیکربندی سیستمهای کامپیوتر VAX بوسیله DEC در ۱۹۸۰ بکار برده شده است.

پیدایش ریزپردازنده ها باعث توجه صنعتهای بسیاری برای بکاربردن تکنولوژی منطق fuzzy در اکثر محصولات مصرفی ، شده است. سازندگان ژاپنی نسبت به دیگر سازندگان در دنیا، سود قبل توجهی بدست م آورند بخاطر وصل کردن تکنولوژی منطق fuzzy در محصولاتشان به همراه قابلیتهای زیادش برای توسعه دادن تکنولوژی های Sensor .

وقتیکه Matsushita برای اولین بار در سال ۱۹۸۹ماشین ظرفشویی بر اساس منطق fuzzy را ساخت، فروشش خیلی موفقیت آمیز بود که موجب شد بسیاری از صنعتهای ژاپنی ، تکنولوژی منطق fuzzy را در تولیداتشان بکار گیرند. امروزه پیداکردن محصولی که در آن از تکنولوژی منطق fuzzy استفاده نشده باشد، خیلی

مشکل است.

در اواسط سال ۹۰ ، تکنولوژی شبکه عصبی با تکنولوژی fuzzy بطور موفقیت آمیز در بسیاری از محصولات بکار میرفت. اکثر کاربردهای تکنولوژی های neuro-fuzzy در محصولات ، برای مرحله طراحی هستند که کنترلرها داخل سیستمها طراحی و گنجانده شده است. شکل ۱۲-۹ یک بلوک دیاگرام از طراحی یک سیستم هوشمند را نشان میدهد که شامل ٤ اجزا است: واسطه ماشین هوشمند، ادراک، شناخت و اجرا.

تحقیق در AI تعدادی از علوم مانند علم عصب ، علم زیستی ، روانشناسی ، فیزیک ریاضیات و علوم کامپیوتر را درگیر میکند، اما بطور وسیع کاربردهایشان در دو علوم مهم وارد شده است: علوم کامپیوتر و مهندسی الکترونیک. در سستم هوشمند یا در ماشین طراحی شده ، و در حوزه ی علم کامپیوتر ، یک بخش بزرگی از تحقیق و توسعه (R&D) در جزء واسطه ماشین هوشمند متمرکز شده است.بعضی از مثالهای R&D در این ناحیه، پردازش زبان طبیعی ، سیستمهای خبره، واسط ماشینی ، استخراج داده ، استخراج متن و غیره هستند. در حوزه مهندسی الکترونیک ، تلاشها در R&D در طراحی اجزاء شناخت از ماشین هوشمند متمرکز شده اند. برای مثال، در طراحی محصولات ، یک نوعی از تکنیکهای AI در طراحی کردن کنترلر جاسازی شده ، بکار گرفته شده است، مانند ماشینهای ظرفشویی ، پلوپز ، یخچالها و ...
غیر از محصولات مصرفی، تکنولوژی AI در سیستمهای صنعتی بکار میرود. Hitachi در ژاپن ، کنترل قطار منطق fuzzy ، زودتر از سال ۱۹۷۸ شروع شده بود. بعد از ۳۰۰۰۰۰ شبیه سازی و ۳۰۰۰ مسیر خالی، آنها اجازه ی عمل کردن در سال ۱۹۸٦ را گرفتند.
شکل ۱۲-۱۰ مسیر پروفایل قطار fuzzy را به همراه ٦ مرحله اجرا نشان میدهد که بوسیله کنترلر fuzzy برای بهبود و مدل کردن اجرا رسیدگی شده اند که عبارتند از: زمان جاری، مصرف انرژی ، قابلیت ردیابی، ایمنی و راحتی . Hitachi نشان میدهد

که قطار fuzzy میتواند در سه زمان متوقف شود ، در ۲ زمان قدرتش کاهش یابد و نیروی مصرفی ٪۱۰ برابری با کنترلر PID کاهش یابد. موفقیت قطار fuzzy بسیاری از کاربردهای AI در صنعت را بدنبال دارد.

بر اساس اشتباه انتشار در الگوریتم شبکه عصبی و منطق fuzzy (تکنیکهای Pavilion) ، یک شرکت آمریکایی، یک نرم افزاری را برای فرایند مدل کردن و بهینه سازی که ((اطلاعات فرایند)) نامیده میشود ، توسعه داده است. این نرم افزار میتواند برای خواندن مقدار زیادی از داده های فرایند مرحله یکنواخت از مورخ و شبکه عصبی بکار رفته شده برای مدل کردن دستگاهی که بصورت offline کار میکند، مورد استفاده قرار گیرد. منطق fuzzy برای اداره کردن محدودیات fuzzy در طی مرحله offline برای اهداف کنترل مورد استفاده قرار میگیرد. سالیانه بیش از یک میلیون دلار بوسیله بسیاری از صنعتهای کاملا بهینه سازی شده بوسیله Pavilion ، با بکارگیری نرم افزار اطلاعات فرایند ، بدست می آید. شکل ۱۲,۱۱ دید دیاگرامی از نرم افزار اطلاعات فرایند را نشان میدهد.

اخیرا بسیاری از صنعتها در ژاپن بر پژوهش در روباتهای شبیه انسان متمرکز شده اند. صنعتهایی مانند هوندا، میتسوبیشی، سونی ، فوجیتسو، NEC ، NTT ، پاناسونیک و غیره.

تکنولوژی اصلی در روباتهای شبیه انسان ، بالابردن توانایی راه رفتن ، سخن گفتن و تشخیص تصویر ، و پیشرفت در موتورها و محرکها است. ASIMO (گام پیشرفته در ابداع تحرک) که توسعه داده شده توسط هوندا بود ، توانایی راه رفتن را نشان میدهد که حتی بالا رفتن و پایین آمدن از پله ها را به روباتها اجازه میدهد. اما قابلیت جسمی شان هنوز مطابق با یک انسان ۵ ساله نیست.

در سه دهه ، دولت ژاپن ۵۰ میلیون ین (٤۰۰ میلیون دلار) برای توسعه ی یک روبات شبیه انسان با ظرفیت هوشی ، جسمی و احساسی از یک انسان ۵ ساله ، هزینه کرده

.

است که به پروژه "Atom" معروف است. پژوهشگران معتقدند پروژه Atom ، الهام گرفته شده با سریهای انیمیشن روبات مشهور "Tetsuwan Atom" بوسیله کارتونیست Osamu Tezuka است، که نه تنها در نشستن انسان روی ماه بلکه در مشارکت بر محدوده ی وسیعی از تکنولوژی های breakthrough موفق است.

سؤالات تستی - تشریحی

سؤالات فصل ۱

تمرین :

۱) کامپیوتر ها انسانها را از انجام دادن چه وظایفی بی نیاز کرده اند ؟

۲) خصوصیاتی را که از قابلیت های ضروری برای یک رفتار هوشمندانه است نام ببرید

۳) کدام یک از موارد زیر هوشمندی لازم دارد ؟

الف)تولید و درک گفتار. ب) تشخیص الگو ج) حرکت در یک فضای پر از مانع دینامیک د) همه موارد

٤) مزایای ارائه سمبلیک را نام ببرید .

٥) اهداف اصلی تحقیق AI را بنویسید .

٦) ویژگی های مهم تست تورینگ را بیان کنید .

۷) کاربردهای AI را نام ببرید .

۸) سیستم های خبره را تعریف کنید .

۹) جای خالی را با عبارت مناسب پر کنید .

الف) ..... کارایی ماشین هوشمند را در برابر انسان اندازه می گیرد .

ب) بهترین و تنها استاندارد برای رفتار هوشمند ، .....، است .

ج) ..... ، سیستم های مشاور اتوماتیک هستند .

د) در طراحی هر سیستم خبره برای حل هر مسئله ..... وجود دارد .

۱۰) کدام یک از گزینه های زیر صحیح است ؟

الف) اثبات یا رد کردن قضایای ریاضی یک کار کاملاً فکری است

ب)PROLOG ، زبان ارجح برای بسیاری برنامه نویسان AI است که خصوصیات پردازش لیست و دستکاری نماد LISP را دارد .

ج) پروسه مهندسی نرم افزار تولید یک برنامه از مشخصات رسمی ، یک مسأله AI است .

د) همه موارد

سؤالات فصل ۲

تمرین :

۱) جملات زیر را به منطق مرتبه اول ترجمه کنید .

همه سگها پستاندارند .

فیدو سگ است .

فیدو پستاندار است .

همه پستاندارها شیر تولید می کنند .

۲) **خواص WFF** (فرمول خوش ساخت) را بنویسید

۳) *تفکیک پذیری را تعریف کنید .*

٤) **اجزای اساسی منطق گزاره ای را نام ببرید .**

٥) جاهای خالی را با عبارات مناسب پر کنید .

الف)هدف منطق نمادین ، ..... است

ب) « فرمولها » در منطق گزاره ای به صورت..... تعریف شده اند

ج) یک فرمول اتمی ، یک فرمول .....است.

٦) کدام یک از گزینه های زیر صحیح نیست ؟

الف) یک فرمول « متناقض » نامیده می شود اگر و فقط اگر تحت تمام تفاسیرش غلط باشد.

ب) یک فرمول « نا متناقض » نامیده می شود ا گر و فقط اگر متناقض نباشد

ج). یک فرمول نامعتبر است اگر حداکثر در یک تفسیر غلط باشد.

د) یک فرمول« معتبر » گفته می شود اگر وفقط اگر تحت تمام تفاسیرش درست باشد.

۷) کدام یک از گزینه های زیر صحیح نیست ؟

الف) یک فرمول معتبر است اگر نقیض آن متناقض باشد.

ب) یک فرمول متناقض است اگرنقیض آن معتبر باشد .

ج) یک فرمول نامعتبر است اگر حداقل در یک تفسیر غلط باشد.

د) اگر یک فرمول متناقض باشد آنگاه غیر معتبر است و عکس آن نیز صحیح است .

سؤالات فصل ۳

تمرین :

۱) فراگیری دانش چیست ؟

۲) رویه ی فراگیری دانش را با رسم شکل شرح دهید .

۳) طرحهای نمایش شبکه را توضیح دهید .

٤) شبکه های معنائی را توضیح دهید .

٥) گرافهای ادراکی را شرح دهید .

٦) خصوصیات گرافهای ادراکی را بنویسید .

۷) قالب را تعریف کرده و خصوصیت های مهم آنرا نام ببرید .

۸) مزایای و معایب اسکریپت ها را بنویسید .

۹) کدام یک از گزینه های زیر صحیح است ؟

الف) راههای مختلفی برای طبقه بندی انواع دانش وجود دارد ، یکی از مهمترین امتیاز ها تفاوت میان دانش القاء شده و دانش استنتاج شده ، است

ب) فرآیند فراگیری دانش نیز می تواند به عنوان وظیفهٔ شناختن و استخراج کردن دانش دامنه مربوط ، جهت حل مسائل پیچیده تعریف شود .

ج) سودمندی یک سیستم خبره می تواند کاملاً به دانش استراتژیک وابسته باشد .

د) همه موارد

۱۰) موضوعات کلیدی که یک طراح سیستم AI با آنها روبرو می شود ، عبارتند از:

الف) فراگیری دانش ب) ارائه دانش ج) دستکاری دانش د) همه موارد

۱۱) سه منبع اصلی دانش عبارتند از :

الف) ادبیات ب) مهارت ها ج) مثال ها د) همه موارد

۱۲) سه رکن مختلف دانش عبارتند از :

الف) قوانین علمی ب) تجربه ج) الگوها د) همه موارد

۱۳) جاهای خالی را با عبارات مناسب پر گنید .

الف) نمایش های شبکه ای ، دانش را به عنوان یک .....دربر می گیرد

ب) برای حل استنتاج ، حل کردن ..... اولویت دارد

ج) یک سیستم خبره یک برنامه ..... است که راه حل های "کیفیت خبره " برای مسائل موجود در گستره ای خاص ، را فراهم می کند

د) یک گراف ..... یک گراف متناهی ، متصل و دوقسمتی است

سؤالات فصل ۷

سئوالات تستی:

۱. جای خالی را با کلمات مناسب پر کنید.

الف) مشکل اصلی یک چشم کامپوتری ***تشخیص و فهم یک شئ*** یا یک منظره با ***خواص سه بعدی*** آن در یک تصویر یا یک توالی از تصاویر است.

ب) تایید کیفیت تصاویر به وسیله ***مکانیزم پردازش تصاویر*** انجام می شود.

ج) همه واسطهای محدودشده زبان طبیعی با مشکل ***قابلیت سکنی*** مواجه می شوند.

د) نتیجه تجزیه یک جمله معمولا" به صورت یک ***درخت*** است.

ه)در تجزیه پایین به بالا همه گرامرهای مستقل از متن می توانند در ***n*n*n*** گام تجزیه شوند . طوری که n طول جمله می باشد.

۲)کدام یک از گزینه های زیر جزء مراحل تقلید یک کامپیوتر از دید انسان نیست؟
الف) پردازش تصویر ب) تجزیه و تحلیل تصویر
ج)تبدیل آن به eigen vector د) فهم تصویر
۳) کدام یک گزینه های در مورد پردازش زبان طبیعی صحیح نیست؟
الف) یکی از بزرگترین کاربردهای هوش مصنوعی است.
ب) مزیت واسطهای زبان طبیعی این است که به اندازه کافی مختصر و موجز هستند.
د) مزیت آن این است که به هیچ مهارت خاصی غیر از تایپ نیاز ندارد.
د) از لحاظ مفهومی دو نوع واسط زبان طبیعی وجود دارد.
٤) کدام یک از گزینه های زیر از انواع واسطهای زبان طبیعی هستند؟
الف) آنهایی که خود را به یک زیر مجموعه از زبان انگلیسی محدود می کنند.
ب) آنهایی که سعی می کنند یک پوشش کامل از یک زبان را تامین کنند.
ج) الف و ب د) هیچ کدام
۵)کدام یک از موارد زیر جزء انواع گرامر ها هستند؟
الف)قوانین حساس به متن ب)قوانین مستقل از متن
ج)گرامرهای با قاعده ***د)همه موارد***

سئوالات تشریحی:

۱)در پردازش زبان طبیعی دو تکنیک برای تجزیه و تحلیل زبان طبیعی را نام برده و مختصرا" توضیح دهید.
۲) گرامر یک زبان چیست؟
۳) پارسر چیست؟
٤) انواع ابهامات در یک زبان طبیعی را نام برده و تعریف کنید(دو مورد)

سؤالات فصل ۸

سؤالات تستی

۱) کدام برنامه نویسی دارای صفت جداسازی دانش از کنترل است؟

الف)رویه ای ب)مبتنی بر دانش ج)شیءگرا د)هرسه مورد

۲)در مورد نرم افزار مبتنی بر دانش کدام گزینه صحیح است؟

الف)ساختار رویه ای ثابت ب)مناسب برای تغییر سیمبولها

ج)سیستم خبره= مسئله+کنترل د)مناسب برای پردازش عددی

۳)کدام گزینه در مورد سیستمهای خبره صحیح نیست؟

الف)سیستم مبتنی بر دانش پیچیده شده است.

ب)میتواند برای دانش درونی خود استدلال کند.

ج)برنامه های کامپیوتری دارای اثر متقابل هستند

د)هیچکدام

٤)کدامیک از موارد زیر،عملیات قابل اجرا بر روی موجودیتها را در هنگام حل یک مسئله مشخص میکند؟

الف)ارتباطات ب)توصیفات ج)رویه ها د)منطق

۵)کدام گزینه صحیح است؟

الف)هر سیستمی برای بازنمایی کل دامنه دانش میتواند مناسب باشد.

ب)دانش مانند یک ماده تصفیه شده است.

ج)GPS برای حل مسائل پیچیده بسیار قوی است.

د)میزان مفید بودن یک سیستم خبره مستقیما" وابسته به کیفیت دانش آماده شده توسط طراحان است.

٦)در کدام مرحله از مراحل اکتساب دانش باید به سؤال روبرو پاسخ داده شود؟ (الگوی جریان اطلاعات چیست؟)

الف)مرحله شناسایی مسئله ب)مرحله ادراک

ج)مرحله رسمی سازی د)مرحله تست

۷)کدام عامل در فرایند رسمی سازی مهم است؟

الف)فضای فرضیه ب)مدل اصولی فرایند

ج)خصوصیات داده ها د)هر سه مورد

۸)لیسپ جزء کدام نوع از ابزارهای توسعه در دسترس است؟

الف)زبانهای الگوریتمیک ب)بدنه ساختمان سیستمهای خبره
ج)محیط های توسعه د)زبانهای سیمبولیک

۹)کدامیک از موارد زیر بهترین روش برای ساخت نمونه اولیه سیستمهای خبره است و برای بکارگیری نیاز به مهارتهای برنامه نویسی کمتری دارد؟

الف)محیط توسعه ب)شل
ج)زبانهای الگوریتمیک د)زبانهای سیمبولیک

۱۰)کدامیک از کاربردهای زیر هدفش یافتن ساختمان مولکولی یک جسم مرکب است؟

الف)PARRY ب) **مایسین ج)الیزا** د)DENDRAL

سؤالات جاخالی

۱)یک سیستم خبره، سیستمی است که دارای قوانین ..... است و از .....کورکورانه اجتناب میکند و این سیستم ها،برنامه های کامپیوتری دارای.....هستند.

۲)متخصصان معمولا داری دانش ..... هستند که این دانش بطور گسترده شامل قوانین ..... به نام ..... هستند.

۳)سیستمهای خبره..... را تغییر میدهند در حالیکه برنامه های معمولی ..... را دستکاری میکنند.

٤)در یک نگاه مجرد و عمومی، دانش شامل..... و .....
و..... در یک زمینه مورد علاقه است.

۵)دانش میتواند به فرم ..... و ..... و ..... باشد.

٦)هنر جمع آوری و پردازش دانش ..... نامیده میشود که بطور گسترده شامل ..... و رسمی سازی و ..... و آزمایش است.

۷)منابع نمونه ای شامل منابع .....و ..... و ..... هستند.

۸)مرحله ..... شامل ارزیابی نمونه اولیه سیستم و فرم های بازنمایی برای پیاده سازی آن است.

۹)فعالیتهایی که بر توسعه یک سیستم خبره مقدم هستند عبارتند از ..... و یافتن تخصص و .....

۱۰) نسخه ای از الیزا که حاوی اسکریپت روانپزشکی بود به نام..... معروف شد.

۱۱)..... واکنش های یک جوان که مشکل اسکیزوفرنی پارانوید را دارد , شبیه سازی می کند.

جواب:PARRY

سؤالات تشریحی

۱)دانش در هر رشته تخصصی بر چند قسم است؟ به اختصار شرح دهید.

۲)مشکلات اساسی در رهیافت مبتنی بر دانش را بیان کنید.

۳)یک حل کننده مسئله ایده ال چه مواردی را باید داشته باشد؟

٤)چهار تفاوت بین پایگاه های داده و پایگاه های دانش را بیان کنید.

۵)خصوصیات اولیه یک سیستم خبره را بیان کنید.

٦)مراحل اکتساب دانش را نام برید و دو مورد را بدلخواه توضیح دهید.

۷)دو روشی که قوانین در آنها میتوانند برای تطابق و اجرا از پایگاه دانش انتخاب شوند را توضیح دهید.

۸)انواع ابزارهای توسعه در حال دسترس را با ذکر مثال بیان کنید.

۹)امکانات نمونه ای پیاده سازی تهیه شده توسط شل ها را بیان کنید.

سؤالات فصل ۹

سؤالات تشریحی

۱) تفاوتهای بین هوش انسان و ماشین را بیان کنید .

۲) خصوصیات زیستی شبکه های عصبی را نام برده و هر یک را توضیح دهید .

۳) فرآیند یادگیری شبکه های عصبی را به طور کامل توضیح دهید .

٤) الگوریتم های آموزشی به چند دسته تقسیم می شوند ؟ هر یک را به طور خلاصه شرح دهید .

۵) ویژگیهای شبکه Hopfield را بیان کنید .

٦) شبکه های عصبی را با سیستم های مبنی بر قاعده مقایسه کنید .

۷) مزایای محاسبات عصبی را بیان کنید .

۸) توضیح دهید چگونه شبکه های عصبی می توانند پردازش زبان طبیعی را بهبود ببخشند ؟

۹) جاهای خالی را با کلمه مناسب پر کنید .

i. نرون فعالیت الکتریکی را از طریق یک رشته بلند و باریک که بنام ..... شناخته می شود به بیرون می فرستد.

ii. در مغز انسان نوعی یاخته عصبی وجود دارد که سیگنال ها را از دیگران از طریق یک میزبان کوچک که ..... نامیده می شود جمع می کند .

iii. عمومی ترین نوع شبکه عصبی مصنوعی شامل ..... است .

iv. رفتارشبکه عصبی مصنوعی به هر دوی .....

و..... بستگی دارد.

۱۰) کدام یک از گزینه های زیر صحیح نیست ؟

الف) perceptron فقط محدود شده به دو ورودی نیست و می توانیم n ورودی داشته باشیم.

ب) قادر بودن به یادگیری توابع جدا کننده خطی از مهمترین عیب های Perceptron است.

ج)perceptron نمی تواند توابع منطقی.Not، Or،And را ارائه دهد.

د) اگرچه فقط قادر بودن به یادگیری توابع جدا کننده خطی از مهمترین عیب هایPerceptron است ولی ارزش مطالعه کردن دارد .

۱۱) کدام یک از گزینه های زیر صحیح است ؟

الف) perceptron به عنوان یک واژه مترادف برای Feedforward Network یک لایه ای استفاده می شود.

ب) بر خلاف دیگر شبکه‌های عصبی، شبکه SOM از فراگیری نظارت نشده استفاده می‌کند.

ج) شبکه‌هایKohonen واحدهای یک لایه‌ای دارند .

د)همه موارد

۱۲) در الگوریتم آموزشی perceptron اگر I_j مثبت باشد افزایش در W_j ، O را ..... می دهد و اگر منفی
باشد افزایش در W_j ، O را ..... خواهد داد .
الف)افزایش،افزایش ب) افزایش،کاهش ج)کاهش،افزایش
ج)کاهش ،کاهش
۱۳)کدام یک از خصوصیات زیستی شبکه های عصبی نیست ؟
الف) قدرتمند و تحمل خرابی ب) توانایی مواجه شدن با موقعیت های داده ای مختلف ب) محاسبات انبوه ج)ایستا بودن
۱٤) کدام یک از گزینه های زیر در مورد شبکه Hopfield صحیح است ؟
الف) یکی از مراحل برجسته برای تجدید حیات جاری در زمینه شبکه‌های عصبی بود.
ب) واحدها به یکدیگر از طریق اتصالات متقارن دارای وزن وصل شدند.
ج) انتخاب یک واحد بطور تصادفی صورت می‌گیرد.
د) این شبکه به عنوان حافظه شرکت پذیر دسته‌بندی شده است.

سؤالات فصل ۱۰

سؤالات تشریحی

۱) چرخه استدلال مبنی بر مورد را شرح دهید .
۲) پنج فایده استفاده از سیستم استدلال مبنی بر مورد را بنویسید .
۳) سیستم مبنی بر قاعده را با سیستم استدلال مبنی بر مورد مقایسه کنید .
٤) ویژگیهای Caroll II را نام ببرید .
٥) استدلال مبنی بر مثال را شرح دهید .
٦) منابع دانش Chef را نام ببرید .
۷) طرز کار ماجول اصلاح را بنویسید .
۸) پایگاه مورد در سیستم های CBR را توضیح دهید .

۹) اهداف توسعه CLAVIER را بنویسید .

۱۰) جاهای خالی را با کلمه مناسب پر کنید .

I. CASEY یکCBR برای ..... است .

II. پایگاه مورد در سیستم های CBR نماینده ..... است.

III. سیستم استدلال مبنی بر مورد می‌تواند به عنوان یک نوع ویژه از ..... در نظر گرفته شود .

۱۱) کدام یک از گزینه های زیر صحیح است ؟

الف) کارایی سیستم استدلال مبنی بر مورد، مستقیما به کیفیت و کمیت دانشهای رمز شده بستگی دارد.

ب) اگر دانش کافی نباشد قابلیت حل مسئله محدود می‌شود.

ج) سیستمهای استدلال مبنی بر مورد می‌توانند همچنین به عنوان فراگیری اطلاعات به هم پیوسته و سیستمهای یادگیری دیده شوند.

د)همه موارد

۱۲) یک مورد ، شامل کدام یک از گزینه های زیر است ؟

الف) یک جمله مسئله که شامل اطلاعات مفهومی است

ب)حل یک مسئله

ج)خروجی اعمال یک راه حل

د)همه موارد

۱۳)کدام یک از گزینه های زیر صحیح نیست ؟

الف)CHEF یک سیستمCBR برای طراحی یک دستورالعمل چینی است.

ب)PROTOS یک سیستمCBR ، برای گزارش تشخیص های نا مرتب است.

ج) CASEY یکCBR برای تشخیص سکته قلبی است .

د)HYPO برای گزارش تشخیص های نا مرتب است.

سؤالات فصل ۱۱

.

سؤالات تستی

۱) کدام گزینه نادرست است؟

الف) بسیاری از زبانهای برنامه نویسی محدود شده میتوانند برای حالتهای قابل مقایسه در زبانهایی مانند جاوا کار کنند.

ب) راه حل های CSP میتوانند بوسیله تحقیق قاعده دار در سرتاسر واگذاری مقدارهای ممکن به متغیرها یافت شوند.

ج) زبانهای شیءگرا یک مکانیسم برای توصیف موجودیتها و خواص و رفتارشان تهیه میکنند.

د) زبانهای عمومی کنونی به ما اجازه میدهند که بطور صریح انواعی از اطلاعاتشان را کدگشایی کنیم.

۲) کدام گزینه در مورد محدودیات نادرست است؟

الف) آنها میتوانند انواعی از اطلاعات را رسیدگی و گزارش کنند.

ب) آنها نمیتوانند اطلاعات جزئی را نشان دهند .

ج) آنها افزودنی هستند و از این رو مستقل از ترتیبی که تعیین شده اند ، میباشند.

د) آنها بیشتر اعلانی از اکثر انواع اطلاعات هستند و بطور عادی هیچ عنصر رویه ای را در بر نمیگیرند.

۳) کدام مدل سازگاری است که بر طبق محدودیات یگانی محدود میکند؟

الف) node consistency **ب)** k– consistency

ج) arc – consistency **د) هیچکدام**

٤) عناصر پایه ای یک سیستم برنامه نویسی محدود شده کدام موارد هستند؟

الف) مکانیسم جستجو **ب) مکانیسم انتشار محدودیت**

ج) مدلهای سازگاری **د) گزینه الف و ب**

۵) کدامیک مشخصات استراتژی جستجو را توزیع می کند؟

الف) OZ **ب)** CHR

ج) FD **د)** CP

سؤالات جاخالی

۱) برنامه نویسی محدود شده یا ..... نوعی از مطالعه مربوط به مدلهای ..... است.

۲) پیشگام(Pioneering) روی شبکه هایی از ..... که بیشتر بوسیله رخداد مسائل در زمینه ای از ..... تحریک شده اند، کار میکند.

۳) محدودیات بطور وسیع به محدودیات ..... و محدودیات ..... دسته بندی میشوند.

٤) در نمایش ..... ، مجموعه هایی از مقدارهای مجاز به صورت فاصله نمایش داده میشوند و در نمایش ..... یک لیستی از مقدارهای قابل دسترس وجود دارد.

۵) در اصل نقش ..... کاهش دادن مقدارهای ممکن از متغیر دیگر است که اطلاعات اضافی روی یک متغیر را کاهش می دهد.

٦) هدف ..... درج نمادهای " _ " برای نشان دادن شکاف و فاصله داخل K توالی بعنوان یک راهی که نتایج K توالی طول یکسان داشته باشد، است.

سؤالات تشریحی

۱) برنامه نویسی محدود شده چیست؟

۲) مسئله رضایتمندی محدودیت را بطور مختصر شرح دهید.

۳) محدودیات موجود در مسئله ٤ وزیر را شناسایی و بیان کنید.

٤) سه نمونه از مدلهای سازگاری را با ذکر خصوصیاتسان بیان کنید.

۵) برنامه ای بنویسید که بتواند معمای کریپتاریتمیک زیر را حل کند. (در صورتی که D=5 باشد آنگاه مقدار حروف دیگر را در محدوده ی ۰ تا ۹ پیدا کنید.)

DONALD
GERALD +
ROBERT

سؤالات فصل ۱۲

سئوالات تستی:

۱) جای خالی را با کلمات مناسب پر کنید.

الف) رهیافت های AI در پیشرفت هر دو سیستمهای بازرگانی الکترونیکی *B2B,* *B2C* مفید است.

ب) در تجارت B2B ، AI اساسا" برای مدیریت *رشته ذخیره شده SCM* به کار گرفته می شود.

ج) سیستم های CBR ***اولویت های کاربر*** را قبول می کند.

د) متعارفترین تکنیک در کاربردهای بازرگانی CBR ***بازیابی نزدیکترین مجاور*** است.

و) در بازرگانی مذاکره به صورت ***مشارکتی*** یا ***رقابتی*** طبقه بندی می شود.

۲) در زمینه تجارت الکترونیکی بیشتر توجه AI بر اداره چه مقوله ای متمرکز شده است؟

الف) B2C　　ب) B2B

ج) C2B　　د) C2C

۳) کدام یک از رهیافت های زیر برای فیلتر کردن اخبار استفاده می شود؟

الف) case based reasoning(CBR)

ب) رهیافت Hybrid

ج) رهیافت content based recommender

د) هیچ کدام

٤) کدام یک از رهیافت ها در مذاکرات به صورت وسیع به کار می روند؟

الف) CBR　　ب) Hybrid

ج) content based recommender　　د) همه موارد

۵) کدام یک از تکنیک های زیر برای سرور های بازرگانی است که می توانند به طور همزمان در دسترس تعداد زیادی مشتری قرار گیرند؟

الف) break journey　　ب) smart client

ج) query based　　د) text based

٦) کدام یک از روش های زیر به عنوان ابزاری برای دسترسی به اطلاعات در تجارت B2C سریعتر هستند؟

الف) text based　　ب) query based

ج) preference based navigation د) هیچ کدام

سئوالات تشریحی:

۱. انواع رهیافت های انتخاب و پیشنهاد کالا را نام برده و مختصرا" توضیح دهید.
۲. اشکال اصلی در ACF چیست؟
۳. حضور AI در تجارت B2B را توضیح دهید.
٤. تعدادی از مسائل در بازیابی مدیریت و استفاده مجدد از اطلاعات در زمینه گردشگری را توضیح دهید.
۵. گردشگری الکترونیکی را تعریف کنید.
٦. دلایل موفقیت سیستم های چند عاملی را توضیح دهید.
۷. الگوریتم های تکوینی در زمینه استفاده از AI در صنعت را توضیح دهید.