اصول سیستمهای خبره

Peter J.F. Lucas & Linda C. van der Gaag

برگردان: مهدی مرسلی
انتشارات پندار پارس

انتشارات پندارپارس

دفتر فروش: انقلاب، ابتدای کارگرجنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ **www.pendarepars.com**

تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸همراه: ۰۹۲۱۴۳۷۱۹۶۴ **info@pendarepars.com**

نام کتاب	:	**اصول سیستم‌های خبره**
ناشر	:	**انتشارات پندار پارس**
تالیف	:	Peter J.F. Lucas & Linda C. van der Gaag
برگردان	:	**مهدی مرسلی**
چاپ نخست	:	**اردیبهشت ۹۵**
شمارگان	:	**۵۰۰ نسخه**
طرح جلد	:	**سارا یعسوبی**
چاپ، صحافی	:	**روز**
قیمت	:	**۳۳۰۰۰ تومان** **شابک : ۶-۱۰-۸۲۰۱-۶۰۰-۹۷۸**

***هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد ***

تقدیم به:

پدر و مادر فداکار و همسر مهربانم

پیشگفتار

کتاب اصول سیستم‌های خبره پیتر جی. لوکاس و لیندا سی. وندرگاک یکی از معدود کتاب‌های سیستم‌های خبره است که هیچ کاری را ناتمام رها نکرده است و تلاش کرده به‌جای راهنمایی کردن خواننده برای انجام کار، نمونه‌ای از آن کار را انجام دهد تا خواننده بتواند چگونگی انجام کار را بیاموزد.

در هر فصل از کتاب نخست مطلب به صورت تئوری گفته شده و سپس کاربرد آن با استفاده از نمونه‌های فراوان بررسی شده است. البته ناگفته نماند که نویسندگان به این بسنده نکرده و همراه با بیان هر کدام از روش‌های بازنمایی دانش، برنامه‌ای کاربردی برای روش بازنمایی دانش گفته شده، نوشته‌اند که باعث می شود خواننده با مطالعه کتاب، توان ساخت سیستم خبره را در خود احساس کند. آنچه که حقیر را به برگردان این کتاب برانگیخت، نبود منابع مطالعه متنوع برای درس سیستم‌های خبره بود که باعث می‌شد همه‌ی دانشجویان در بیشتر دانشگاه‌ها تنها با شمار اندکی از مسائل سیستم‌های خبره روبرو شوند و از نمونه کارهای دیگر محروم باشند.

انگیزه‌ی دیگر این بود که کتاب حاضر بسیار کم به مطالب حاشیه‌ای پرداخته و نمونه‌ای از یک کتاب فنی بسیار خوب و کامل است که در آغاز مطالب را به صورت تئوری گفته و سپس با آوردن نمونه مسئله‌هایی تلاش کرده تا مسائل دشوار را قابل فهم نماید و در پایان برنامه‌های حقیقی نوشته شده به دو زبان برنامه نویسی پرولوگ و Lisp را آورده است که بسیار کابردی و مفید است. همچنین در پایان فصل‌های کتاب تمرین‌هایی بسیار هوشمندانه از دانشجویان خواسته شده است که می تواند عمق مطالب آموزش داده شده را دو چندان نماید. در پایان کتاب در دو پیوست جداگانه مقدمه‌هایی کوتاه بر زبان‌های پرولوگ و Lisp آورده شده است که می تواند برای کسانی که با این زبان‌ها آشنا نیستند، مفید باشد.

با وجود اینکه حقیر همه‌ی بخش‌های کتاب را با وسواس بسیار به فارسی برگردان و آماده چاپ نموده بودم، به خاطر افزایش صفحات و قیمت روی جلد کتاب، از چاپ فصل هفتم که معرفی LOOPS، OPS5 و CENTAUR بود، گذشتیم و همچنین فهرست واژه‌گان و فهرست تصاویر نیز سرنوشتی بهتر از فصل هفتم نیافتند.

بر خود لازم می‌دانم به کسانی که در عصر خوانش سرسری مطالب بر روی انواع رسانه‌های الکترونیکی، هنوز هم کتاب می‌خوانند، دست مریزاد گفته و از دور، دست یکایک آنان را بفشارم. همه‌ی خطاهای برگردان و نگارش و ... کتاب حاضر بر عهده بنده می‌باشد و از طریق آدرس پست الکترونیکی mehdi_morsali@yahoo.com جویای نظرات اصلاحی دوستان هستم.

فهرست مطالب

فصل نخست: مقدمه ۱

۱-۱- سیستم‌های خبره و هوش مصنوعی ۲
۱-۲- چند مثال ۳
۱-۳- جداسازی دانش و استنتاج ۶
۱-۴- دامنه یک مسئله ۱۳
تمرین‌ها ۱۷
فصل دوم: منطق و رزولوشن ۱۹
۲-۱- منطق گزاره‌ای ۲۰
۲-۲- منطق مسندی مرتبه نخست ۲۸
۲-۳- فرم کلازی منطق ۳۵
۲-۴- استدلال در منطق ۳۹
۲-۵- رزولوشن و منطق گزاره‌ای ۴۲
۲-۶- رزولوشن و منطق مسندی مرتبه نخست ۴۶
۲-۶-۱- جایگزینی و یکسان‌سازی ۴۶
۲-۶-۲- جایگزینی و یکسان‌سازی در لیسپ ۵۰
۲-۶-۳- رزولوشن ۵۶
۲-۷- استراتژی‌های رزولوشن ۶۰
۲-۷-۱- رزولوشن معنایی ۶۱
۲-۷-۲- رزولوشن SLD: حالت ویژه‌ای از رزولوشن خطی ۶۴
۲-۸- پیاده‌سازی رزولوشن SLD ۷۲
۲-۹- کاربرد منطق در ساخت سیستم خبره ۸۵
۲-۱۰- منطق به عنوان روش بازنمایی ۹۵
تمرین‌ها ۹۷
فصل سوم: قواعد تولید و استنتاج ۱۰۱

۳-۱-بازنمایی دانش در یک سیستم تولید ۱۰۲
۳-۱-۱-متغیرها و حقایق ۱۰۲
۳-۱-۲-شرط‌ها و نتیجه‌ها ۱۰۵
۳-۱-۳-تاپلهای شیء-صفت-مقدار ۱۱۰
۳-۱-۴-قواعد تولید و منطق مسندی مرتبه نخست ۱۱۲
۳-۲-استنتاج در سیستم تولید ۱۱۶
۳-۲-۱-استنتاج بالا به پایین و قواعد تولید ۱۱۹
۳-۲-۲-استنتاج بالا به پایین در پرولوگ ۱۳۰
۳-۲-۳-استنتاج بالا به پایین در لیسپ: ۱۳۹
۳-۲-۴-استنتاج پایین به بالا و قواعد تولید ۱۵۰
۳-۳-بازشناخت الگو و قواعد تولید ۱۶۰
۳-۳-۱-الگوها، حقایق و همتایی ۱۶۰
۳-۳-۲-الگوها و قواعد تولید ۱۶۳
۳-۴-قواعد تولید به‌عنوان یک روش بازنمایی ۱۶۵
تمرین‌ها ۱۶۶
فصل چهارم:فریم‌ها و وراثت ۱۷۱
۴-۱-شبکه‌های معنایی ۱۷۲
۴-۱-۱-راس‌ها و یال‌های برچسب گذاری شده ۱۷۲
۴-۱-۲-وراثت ۱۷۸
۴-۱-۳-شبکه معنایی گسترش یافته ۱۸۲
۴-۲-فریم‌ها و وراثت یگانه ۱۸۴
۴-۲-۱-رده‌بندی فریم درخت مانند ۱۸۵
۴-۲-۲-استثناءها ۱۹۵

۴-۲-۳- وراثت یگانه در پرولوگ ۱۹۹
۴-۲-۴- وراثت یگانه در لیسپ ۲۰۳
۴-۲-۵- وراثت و جنبه‌های صفت ۲۰۸
۴-۳- فریم‌ها و وراثت چندگانه ۲۱۸
۴-۳-۱- زیرنوع‌سازی در رده‌بندی‌های درخت شکل ۲۱۸
۴-۳-۲- وراثت چندگانه مقادیر صفات ۲۲۳
۴-۳-۳- زیرنوع‌سازی در رده‌بندی‌های گراف شکل ۲۳۸
۴-۴- فریم‌ها به‌عنوان یک روش بازنمایی ۲۴۳
تمرین‌ها ۲۴۴
فصل پنجم:استدلال با عدم قطعیت ۲۴۷
۵-۱- قواعد تولید، استنتاج و عدم قطعیت ۲۴۹
۵-۲- نظریه احتمالات ۲۵۵
۵-۲-۱- تابع احتمال ۲۵۶
۵-۲-۲- احتمالات شرطی و قضیه بیس ۲۵۸
۵-۲-۳- کاربرد در سیستم‌های خبره مبتنی بر قواعد ۲۶۰
۵-۳- روش موضوعی بیس ۲۶۵
۵-۳-۱- نرخ همانندی ۲۶۵
۵-۳-۲- توابع ترکیب ۲۶۷
۵-۴- مدل ضریب قطعیت ۲۷۵
۵-۴-۱- اندازه باور و عدم باور ۲۷۵
۵-۴-۲- توابع ترکیب ۲۷۷
۵-۴-۳- تابع ضریب قطعیت ۲۸۰
۵-۵- مدل ضریب قطعیت در پرولوگ ۲۸۳

۵-۵-۱- ضریب قطعیت در حقایق و قواعد ۲۸۴
۵-۵-۲-پیاده‌سازی مدل ضریب قطعیت ۲۸۷
۵-۶- نظریه دمپستر- شافر ۲۹۴
۵-۶-۱-تخصیص احتمال ۲۹۴
۵-۶-۲- قاعده ترکیب دمپستر ۲۹۹
۵-۶-۳-کاربرد در سیستم‌های خبره مبتنی بر قاعده ۳۰۳
۵-۷- مدل‌های شبکه ۳۰۵
۵-۷-۱-بازنمایی دانش در یک شبکه باور ۳۰۵
۵-۷-۲-انتشار مدرک در یک شبکه باور ۳۰۹
۵-۷-۳- مدل شبکه کیم و پرل ۳۱۱
۵-۷-۴- مدل شبکه لاریتزن و اسپیگلهالتر ۳۱۷
تمرینها ۳۲۵
فصل ششم:ابزارهای بازبینی دانش و استنتاج ۳۲۹
۶-۱- واسط کاربر و امکانات توضیح ۳۳۰
۶-۲- واسط کاربر در پرولوگ ۳۳۶
۶-۲-۱- مفسر فرمان (دستور) ۳۳۸
۶-۲-۲-تسهیلات چگونگی ۳۴۰
۶-۲-۳ تسهیلات چرایی منفی ۳۴۲
۶-۲-۴-یک واسط کاربری در لیسپ ۳۴۷
۶-۲-۵-توابع پایه استنتاج ۳۵۰
۶-۲-۶- مفسر فرمان ۳۵۶
۶-۲-۷-تسهیلات چرایی ۳۵۷
۶-۳- مدلهای قاعده ۳۶۴

تمرینها ۳۷۰
پیوست A:مقدمه‌ای بر پرولوگ ۳۷۵
A–۱–برنامه نویسی منطقی ۳۷۷
A–۲–برنامه نویسی در پرولوگ ۳۷۹
A–۲–۱– معانی توصیفی ۳۸۰
A–۲–۲– معانی رویه‌ای و مفسر ۳۸۲
A–۳–مروری بر پرولوگ ۳۹۰
A–۳–۱– خواندن در برنامه ۳۹۰
A–۳–۲– ورودی و خروجی ۳۹۱
A–۳–۳– مسندهای محاسباتی ۳۹۲
A–۳–۴–بررسی جایگزینی‌ها ۳۹۴
A–۳–۵–کنترل پسگرد ۳۹۵
A–۳–۶–دستکاری پایگاه داده ۳۹۸
A–۳–۷– دستکاری ترمها ۳۹۹
پیوست B: مقدمه‌ای بر لیسپ ۴۰۳
B–۱–اصول بنیادین لیسپ ۴۰۵
B–۱–۱–عبارات لیسپ ۴۰۵
B–۱–۲– فرم ۴۰۷
B–۱–۳–انتزاع رویه‌ای در لیسپ ۴۰۹
B–۱–۴–متغیرها و محدوده آنها ۴۱۱
B–۲–مروری بر لیسپ ۴۱۲
B–۲–۱– دستکاری نمادها ۴۱۲
B–۲–۲– مسندها ۴۱۸

B-۲-۳-ساختارهای کنترلی ۴۲۴
B-۲-۴-عبارات لمبدا ۴۲۸
B-۲-۵-اجبار به ارزیابی توسط مفسر لیسپ ۴۲۹
B-۲-۶-تعریف ماکرو و backquote ۴۳۱
B-۲-۷-ساختار ۴۳۲
B-۲-۸-ورودی و خروجی ۴۳۴

فصل ۱

پیش‌گفتار

در طول دهه گذشته، علاقه به نتایج پژوهش‌های مرتبط با هوش مصنوعی با سرعت چشم‌گیری رو به افزایش است. به‌ویژه، *سیستم‌های مبتنی بر دانش*[1] یکی از نخستین عرصه‌هایی است که از نظر تجاری با موفقیت‌هایی همراه بوده و توجه زیادی را به خود جلب کرده است. اصطلاح سیستم‌های مبتنی بر دانش به‌طور معمول برای توصیف سیستم‌های اطلاعاتی ویژه‌ای به کار می‌رود که در آنها بازنمایی نمادین دانش بشر انجام شده و عموما استدلال انسانی همانندسازی شده است. در میان سیستم‌های مبتنی بر دانش موجود، *سیستم‌های خبره*[2] از همه موفق‌تر بوده‌اند. سیستم‌های خبره، سیستم‌هایی هستند که قادر به ارائه راه حل برای یک مسئله در یک زمینه (دامنه[3]) خاص می‌باشند و یا می‌توانند برای حل مسئله توصیه‌هایی ارائه دهند که هم از لحاظ روش کار و هم از لحاظ نتیجه با افراد خبره در آن زمینه قابل مقایسه‌اند. ساختن یک سیستم خبره برای کاربرد در یک زمینه خاص، خود مبحثی مجزاست که با عنوان *مهندسی دانش*[4] شناخته می‌شود.

دامنه مسائلی که سیستم‌های خبره برای حل آنها ساخته می‌شوند، محدود به مسائلی است که دانش انسانی لازم برای حل مسئله در آن زمینه وجود دارد. به عنوان نمونه، از این دست مسائل می‌توان از تشخیص پزشکی بیماری‌ها، مشاوره مالی، طراحی محصولات و . . . نام برد. امروزه سیستم‌های خبره قادر به ارائه راه حل در زمینه‌های محدودی هستند. با این وجود، حتی در دامنه بسیار محدود مسائل نیز سیستم‌های خبره به‌طور معمول به مقدار زیادی دانش نیاز دارند تا کارایی آنها با یک انسان خبره در آن زمینه قابل مقایسه باشد.

در این فصل به بررسی ریشه‌های تاریخی سیستم‌های خبره در زمین رشته برادر یعنی هوش مصنوعی خواهیم پرداخت و به‌طور خلاصه چند مثال کلاسیک را بررسی خواهیم کرد. افزون بر این اصول بنیادین سیستم‌های خبره را معرفی کرده و ارتباط آنها با فصل‌های کتاب را بیان خواهیم نمود. تلاش شده است که بیان اصول بنیادین، دارای عمق قابل توجهی باشد. این فصل با توصیف دامنه مسئله‌ای به پایان خواهد رسید که به‌طور تقریبی همه‌ی مثال‌های کتاب از آن دامنه انتخاب خواهند شد.

---

1 Knowledge-based systems

2 Expert Systems

3 Domain

4 Knowledge Engineering

۱-۱- سیستم‌های خبره و هوش مصنوعی

هرچند کامپیوترهای دیجیتال بر این پایه طراحی شده‌اند تا یک پردازنده اعداد باشند، با این وجود در همان روزهای آغازین خلقتشان، گروه کوچکی از پژوهشگران، آنها را برای کاربردهای غیر عددی به کار گرفتند. سرانجام تلاش این پژوهشگران منجر به چیزی شد که در سمینار کالج دارتموث[۱] سال ۱۹۵۶ هوش مصنوعی[۲] نامیده شد و مشتاق ایجاد سیستم‌هایی بود که می‌توانستند نتایجی تولید کنند که رفتار انسانی برای آن کارها ضروری به نظر می‌رسید.

نخستین حوزه‌های مورد توجه در دهه پنجاه میلادی عبارت بودند از *اثبات قضایا*[۳] و *حل مسائل*[۴]. در هر دو حوزه برنامه‌های کامپیوتری نوشته شده با دو ویژگی مبتنی بودن بر الگوریتم‌های پیچیده که توانایی حل مسائل عمومی را دارند و مستقل بودن از دامنه یک مسئله ویژه شناخته می‌شوند و افزون بر این، قابلیت اجرا برای مثال‌های آغازین و ساده را دارند.

اثبات قضایا حوزه‌ای است که علاقه‌مند است تا با داشتن یک مجموعه از اصول، قضیه‌ها را به صورت خودکار اثبات نماید. قضیه‌ها و اصول با منطق بیان می‌شوند و قواعد استنتاج منطقی بر روی اصول داده شده اعمال می‌شوند تا قضیه‌ها را اثبات نمایند. نخستین برنامه‌ای که توانست به‌طور حقیقی یک قضیه در نظریه اعداد را اثبات نماید، در سال ۱۹۵۴ توسط ام. دیویس[۵] نوشته شد. اما با این وجود تا نیمه‌های دهه شصت میلادی موفقیت بزرگی در اثبات قضایا به دست نیامد. تنها با ابداع یک قاعده استنتاج به نام *رزولوشن*[۶] اثبات قضایا از نظر علمی به یک موضوع جذاب تبدیل شد. پیشرفت‌های بیشتر در این زمینه در دهه هفتاد با اعمال برخی اصلاحات بر روی اصول آغازین رزولوشن به وجود آمد.

پژوهشگران حوزه حل مسئله بر روی توسعه سیستم‌هایی تمرکز کردند که توانایی عمومی حل انواع گوناگون مسائل را داشته باشند. بهترین سیستم شناخته شده GPS[۷] توسط ای. نیویل[۸]، اچ. ای. سیمون[۹] و جی. سی. شاو[۱۰]

1 Dartmouth
2 Artificial intelligent
3 Theorem proving
4 Problem solving
5 M. Davis
6 Resolution
7 General Problem Solver
8 A. Newell
9 H.A. Simon
10 J.C. Shaw

تولید شد. مسئله ورودی با واژه‌های حالت آغازین، حالت پایانی خواسته شده و مجموعه‌ای از حالت‌های گذار ارائه می‌شد تا از یک حالت به حالتی جدید منتقل شود. با روش گفته شده، GPS با استفاده از حالت‌ها و عملیات، مجموعه‌ای از حالت‌های گذار را تولید می‌کرد که در صورت انجام شدن با ترتیب ویژه، حالت آغازین را به حالت پایانی تبدیل می‌کرد. با این حال GPS چندان موفق نبود زیرا نخست اینکه ارائه مسئله پیچیده آغازین با واژه‌هایی که توسط GPS قابل پردازش باشد کار ساده‌ای نبود. دوم اینکه، کارایی GPS بسیار پایین بود. از آنجا که GPS یک حل کننده مسئله عمومی بود، دانش مخصوص برای حل یک مسئله ویژه که منجر به حل مسئله با کارایی بالا می‌شد، نمی‌توانست در اختیار حالت مورد نظر قرار گیرد. GPS در هر گام همه‌ی حالت‌های ممکن را بررسی می‌کرد که منجر به پیچیدگی زمانی نمایی می‌شد. با وجود اینکه موفقیت GPS به‌عنوان یک حل کننده مسئله، بسیار محدود بود اما توانست توجه زیادی را در هوش مصنوعی به سمت سیستم‌های تخصصی‌تر جلب نماید. این تغییر توجه از سیستم‌های عمومی به سیستم‌های تخصصی که در آنها فرآیند استنتاج می‌توانست توسط دانش مربوط به آن مسئله مورد نظارت قرار گیرد، یک موفقیت بزرگ در هوش مصنوعی به شمار می‌آمد.

برای کارهای عملی در بیشتر حوضه‌ها راه حل‌های خوش تعریف در منابع وجود ندارد. دانش یک فرد خبره در یک حوضه خاص به‌طور معمول منجر به یک تعریف روشن و یا یک الگوریتم نامبهم نمی‌شود، بلکه تنها به قواعد و واقعیتهایی منجر می‌شود که با تمرین آموخته می‌شود و ابتکار یا *هیورستیک*[1] نامیده می‌شوند. بنابراین دانش موجود در یک سیستم خبره، وابستگی زیادی به دامنه مسئله دارد. موفقیت یک سیستم خبره بیشتر به توانایی آن سیستم خبره در ارائه دانش و تکنیک‌های هیورستیک و قابل استفاده کردن آنها برای کامپیوتر بستگی دارد. به‌طور کلی سیستم‌های خبره قادر به ارائه راه حل هستند و توصیه‌هایی که ارائه می‌دهند وابسته به دانشی است که در آنها بازنمایی شده است. افزون بر این سیستم‌های خبره می‌توانند دانش تازه به دست آمده را با دانشی که از پیش موجود است به صورت انعطاف پذیری ادغام نمایند.

۱-۲- چند مثال

نخستین سیستم خبره در اواخر دهه شصت ساخته شد، هر چند آغاز تحقیقات گسترده در این حوزه تا دهه هفتاد به تاخیر افتاد. بیشتر سیستم‌های خبره آغازین به حوزه تشخیص پزشکی علاقه‌مند بودند. بهترین سیستم خبره شناخته شده در حوزه پزشکی که در دهـه هفتـاد ساخته شد MYCIN می‌باشد. ساخت این سیستم خبره در دانشگاه استنفورد انجام شد؛ به‌ویژه ای. اچ. شورتلیف[2] نقش مهمی در توسعه این سیستم خبره داشت. سیستم MYCIN می‌توانست به متخصصان در تشخیص و درمان برخی از بیماری‌ها و به‌ویژه مننژیت و عفونت خونی باکتریایی کمک کند. هنگامی‌که یک بیمار علائم بیماریهای عفونی را داشت، به‌طور معمول کشت نمونه خون و

1 Heuristic

2 E.H. Shortliffe

ادرار بیمار انجام می‌شد تا نوع باکتری‌های عامل بیماری مشخص شوند. آماده شدن نتایج کشت آزمایشگاهی به‌طور معمول بین ۲۴ تا ۴۸ ساعت طول می‌کشید. در چنین مواقعی پزشکان باید فرآیند معالجه را پیش از آماده شدن نتایج آزمایشگاهی آغاز می‌کردند؛ وگرنه ممکن بود بیماری پیشرفت کرده و به فوت بیمار بینجامد. MYCIN با دریافت داده‌های به‌دست آمده از بیمار که ممکن است هم ناقص و هم نادقیق باشند، یک تشخیص موقتی از ارگانیسم‌هایی ارائه می‌دهد که ممکن است منجر به بیماری شده باشند. MYCIN بر اساس این تشخیص موقت تعدادی دارو برای متوقف کردن ارگانیسم‌های شناسایی شده، تجویز می‌کند. داروهای تجویز شده به بیمار و نیز داروهایی که بیمار از پیش مصرف می‌کند، ممکن است باهم تداخل دارویی ایجاد کنند. برای پیشگیری از این اتفاق داروهای تجویز شده به بیمار در حساب کاربری او ثبت و نگه‌داری می‌شوند. افزون بر این، MYCIN می‌تواند توضیحاتی درباره تشخیصی که داده و داروهایی که تجویز کرده، ارائه نماید.

MYCIN اثر خود را بر روی سیستم‌های خبره پس از خود گذاشته است. این سیستم خبره و انشعابات آن منبع اثرات زیادی در دانش پزشکی بوده‌اند. MYCIN باعث انگیزش زیادی برای ساخت سیستم‌های خبره مشابه در حوزه‌های دیگر غیر از پزشکی نیز شد.

ساخت INTEREST-1 در آغاز دهه هفتاد آغاز شد. سیستم توسط اچ. ای. پوپل[1] و جی. دی. میرز[2] در دانشگاه پیتسبورگ ساخته شد. در ادامه پژوهش پوپل و میرز نام سیستم را به CADUCEUS تغییر دادند. یکی از ویژگی‌های مهم CADUCEUS مطالعه برای تشخیص بیماری‌های داخلی بود. در پزشکی داخلی چند صد بیماری گوناگون شناسایی شده‌اند. پزشک داخلی نه تنها باید نشانه‌های آزمایشگاهی همه‌ی این بیماری‌ها را در ذهن داشته باشد بلکه باید به تداخل و ترکیب این بیماری‌ها و علائم آنها نیز توجه داشته باشد. تعداد بیماری‌های داخلی بسیار بیشتر از آن است که بتوان نشانه‌های آزمایشگاهی و فیزیکی آنها را یکی یکی در نظر داشت. CADUCEUS روی بیماری‌هایی تمرکز کرده است که نشانه‌های آزمایشگاهی و فیزیکی مانند هم دارند.

در این عرصه رو به پیشرفت، سیستم‌های خبره در حوزه‌های فنی نیز ساخته شدند. یکی از این نوع سیستم‌های خبره که با واژه سیستم‌های خبره نیز مقیدسازی شده است HEURISTIC DENDRAL می‌باشد. پروژه DENDRAL در سال ۱۹۶۵ در دانشگاه استنفورد کلید خورد. این سیستم توسط جی. لدربرگ[3] شیمی‌دان آلی (و برنده جایزه نوبل شیمی) با همراهی ای. اِی فیگنباوم[4] و بی. جی. بوچانان[5] که هر دو پژوهشگران علمی شناخته شده‌ای در زمینه هوش مصنوعی بودند، ساخته شد. سیستم DENDRAL در شیمی آلی و تخمین ساختار مولکولی

1 H. E. Pople
2 J. D. Myers
3 J. LEDERBERG
4 E. A. Feigenbaum
5 B. G. Buchanan

ترکیبـات آلی که از یک نمونه به دسـت می‌آمدند، کاربرد داشت. در تخمین فرمول ساختاری نمونه‌ای مانند C_4H_9OH یا بوتانول و منبع اصلی که بوتانول از آن به وجود آمده است، از اطلاعاتی استفاده می‌شود که از آزمایش‌های فیزیکی و شیمیایی و طیف‌نگاری به دست آمده‌اند. روش استفاده شده *تولید و آزمایش*[1] نامیده می‌شود، چرا که سیستم نخست همه‌ی ساختارهای مولکولی فرضی ممکن را ساخته و سپس به ترتیب با داده‌های مشاهده شده آزمایش می‌کند. الگوریتم اصلی DENDRAL توسط لدربرگ برای تولید همه‌ی ایزومرهای شیمیایی یک ترکیب ارائه شد. DENDRAL زیرسیستمی داشت که به اصطلاح تولید کننده ساختار نامیده می‌شد و الگوریتم DENDRAL را پیاده‌سازی می‌کرد و افزون بر آن محدودیت‌های هیورستیک را بر روی ساختارهای ممکن اعمال می‌کرد که باعث می‌شد تعداد آلترناتیوهایی که باید توسط سیستم ذخیره شوند، کاهش یابد. طیف‌نگاری جرمی برای یافتن فرمول ساختاری صحیح بسیار مفید است. در طیف‌نگاری جرمی، ترکیب در خلاء توسط پرتویی از الکترون‌ها بمباران می‌شود که این عمل باعث می‌شود مولکول به چندین بخش کوچکتر شکسته شود. این بخش‌های کوچک در داخل میدان الکتریکی شتاب می‌گیرند و به تناسب جرم خود در داخل میدان مغناطیسی انحنا پیدا می‌کنند. تکه‌های جدا شده الگویی به‌وجود می‌آورند که طیف نامیده می‌شود و توسط ابزارهایی رسم می‌شود. این طیف تعداد اکسترمم‌های مربوط به جرم هر کدام از تکه‌ها را نمایش می‌دهد. نمودار طیف‌نگاری دربردارنده اطلاعات با ارزشی درباره ساختار شیمیایی آغازین می‌باشد. DENDRAL در توصیف الگوهای نمودار طیف‌نگاری کمک زیادی انجام می‌دهد. زیرسیستم دیگر DENDRAL که *پیش یاب*[2] نامیده می‌شود، نمودارهای طیف‌نگاری لازم برای هر ساختار مولکولی را پیشنهاد می‌کند. هر نمودار طیف‌نگاری بر اساس شباهت با طیف‌نگاری جرمی مقایسه می‌شود و این کار در آخرین بخش سیستم انجام می‌شود که *تابع ارزیابی*[3] نامیده می‌شود. به‌طور معمول بیش از یک ساختار مولکولی با الگوی به دست آمده از طیف‌نگاری همخوانی دارد. بنابراین سیستم به‌طور معمول بیش از یک پاسخ تولید کرده و بر اساس شواهد مرتب می‌کند.

XCON که پیش‌تر R1 نامیده می‌شد، سیستم خبره‌ای است که می‌تواند VAX، PDP11، MicroVAX را که از تولیدات شرکت DEC[4] می‌باشند، پیکربندی کند. DEC به مشتریانش گزینه‌های زیادی از قطعات کامپیوتری را ارائه می‌دهد و مشتری می‌تواند سیستم سفارشی خود را داشته باشد. بر اساس سفارش مشتری ممکن است لازم باشد تا قطعه‌ای جایگزین قطعه‌ای دیگر شود تا سیستم به بیشترین کارایی عملیاتی برسد. مشکل این مورد در ناقص بودن اطلاعات نیست بلکه هدف این است که این تغییر به سرعت انجام شود و از طرفی پیکربندی سیستم

1 Generate-and-Test

2 Predictor

3 Evaluation Function

4 Digital Equipment Corporation

کامپیوتری نیاز به مهارت و تلاش زیادی دارد. در اواخر دهه هفتاد، DEC با همکاری *جی مک درموت*[1] از دانشگاه *کارنگی- ملون*[2] ساخت XCON را آغاز کرد. تا سال ۱۹۸۱سیستم XCON به‌صورت کامل استفاده می‌شد. هم اکنون XCON به XSEL ضمیمه شده است. XSEL سیستمی است که به کارگزاران DEC در آماده‌سازی سفارش‌ها کمک می‌کند.

سیستم‌های خبره گفته شده در بالا نمونه‌های کلاسیک هستند. صرف نظر از میزان موفقیت آنها، شمار زیادی از سیستم‌های خبره تا پایان دهه هفتاد ساخته شدند. از برخی سیستم‌ها، انشعاب‌های مستقیم دیگری به وجود آمد، به‌عنوان نمونه سیستم MYCIN با نام NEOMYCIN از نو طراحی شد که در آن وظایف تشخیص پیچیده‌ای گنجانده شده بود.

HEURISTIC DENDRAL به‌طور استادانه‌ای زیرسیستم‌هایش را برای تولید و آزمایش ساختارهای مولکولی ممکن بهبود داد. افزون بر آن یک سیستم ساخته شد که می‌توانست راه حل‌های ابتکاری را از مثال‌ها بیاموزد. این سیستم METADENDRAL نامیده می‌شد و برای آسان نمودن انتقال دانش برای استفاده در DENDRAL استفاده می‌شد. در متن پیشنهادی برای مطالعه بیشتر در آخر این فصل چند سیستم خبره جدیدتر نیز به‌صورت کوتاه توصیف شده‌اند.

۱-۳- جداسازی دانش و استنتاج

در سال‌های آغازین، سیستم‌های خبره به‌طور معمول با یک زبان برنامه‌سازی سطح بالا نوشته می‌شدند. در بیشتر موارد LISP به عنوان زبان پیاده‌سازی انتخاب می‌شد. وقتی از یک زبان سطح بالا به‌عنوان ابزار تولید یک سیستم خبره استفاده می‌شود باید در ساختار سیستم دقیق شویم زیرا سیستم باید در دامنه مسئله‌ای کار کند که هیچ چیزی درباره آن نمی‌داند. افزون بر آن دانش فرد خبره و الگوریتم‌هایی که باید بر روی دانش اعمال شوند به شدت به یکدیگر تنیده می‌شوند. این شرایط باعث می‌شود که سیستمی که ساخته می‌شود در برابر تغییرات انعطاف پذیر نباشد. دانش فرد خبره طبیعتی پویا دارد: دانش و خبرگی به‌طور مداوم در حال تغییر هستند. آگاهی از این ویژگی باعث به‌وجود آمدن این دیدگاه شد که جداسازی الگوریتم‌ها از دانش بسیار مفید است، هر چند که اجباری بر این جداسازی نیست. این بینش در ساخت سیستم‌های خبره امروزی به شکل معادله زیر نمایش داده می‌شود.

Expert system=Knowledge+ Inference

در نتیجه یک سیستم خبره به‌طور معمول شامل دو بخش اساسی زیر می‌باشد:

1 J. Mc Dermott

2 Carnegie-Mellon

- یک پایگاه دانش که دانش یک دامنه خاص را پوشش می‌دهد.
- یک موتور استنتاج که از الگوریتم‌هایی تشکیل شده است که برای پردازش[1] داده‌های موجود در پایگاه دانش استفاده می‌شود.

امروزه سیستم‌های خبره هنوز هم با زبان‌های سطح بالا نوشته می‌شوند. این سیستم‌ها به‌طور معمول در یک محیط ویژه و محدود ساخته می‌شوند که به آن پوسته سیستم خبره[2] می‌گوییم. EMYCIN[3] نمونه‌ای از چنین محیط‌هایی می‌باشد که از جدا کردن دانش مربوط به بیماری‌های عفونی از MYCIN به دست آمده است. به تازگی چندین ابزار عمومی جدید برای ساختن سیستم‌های خبره تولید شده‌اند که همانند زبان‌های برنامه‌نویسی خاص منظوره هستند. در این ابزارها نیز جداسازی دانش و استنتاج اجباری است. در این کتاب به این سیستم‌ها، *ابزارهای سازنده سیستم‌های خبره*[4] می‌گوییم.

دانش تخصصی مربوط به یک دامنه ویژه با استفاده از یک روش بازنمایی دانش به پایگاه دانش تبدیل می‌شود. در یک پوسته سیستم خبره یا یک ابزار سازنده سیستم خبره، از پیش یک یا چند روش بازنمایی دانش برای کد کردن دانش در نظر گرفته شده‌اند. افزون بر آن یک موتور استنتاج متناسب با آن روش بازنمایی دانش نیز برای پردازش دانش بازنمایی شده، وجود دارد. در ساخت یک سیستم خبره حقیقی دانش مرتبط با دامنه مسئله مورد نظر به دست آمده و به روش بازنمایی مورد نظر بیان شود. جداسازی پایگاه دانش از موتور استنتاج مزایایی دارد که عبارتند از: نخست اینکه هر کدام از موتورهای استنتاج و پایگاه دانش به‌طور مجزا طراحی و عیب‌یابی می‌شوند و در صورت داشتن مشکل هر کدام بدون نیاز به تغییر اساسی در دیگری اصلاح می‌شوند. امتیاز دیگر جداسازی موتور استنتاج از پایگاه دانش این است که پایگاه دانش موجود می‌تواند با یک پایگاه دانش در یک دامنه دیگر تعویض شده و یک سیستم خبره به‌طور کامل متفاوت به دست آید.

ساخت یک سیستم خبره با استفاده از دانش موجود در منابع دانش گوناگون مانند خبرگی انسانی، کتاب‌ها و پایگاه داده‌های گوناگون انجام می‌شود. ساخت یک سیستم خبره کاری است که نیاز به مهارت بالایی دارد. شخصی که این کار را انجام می‌دهد، مهندس دانش نامیده می‌شود. اگر دانش با استفاده از مصاحبه به دست آید که به‌طور معمول چنین است شخص مصاحبه کننده، استخراج کننده دانش نامیده می‌شود. یک بخش از کار مهندس دانش عبارت است از انتخاب یک روش بازنمایی دانش برای کد کردن دانش به شکلی که در کامپیوتر قابل استفاده باشد.

1 Manipulate
2 Expert System Shell
3 Essential MYCIN
4 Expert system builder tools

در ادامه، چکیده‌ای خواهیم دید، از آن چه که قرار است در کتاب گفته شود. ارائه دانش به‌صورتی که در فرآیند حل مسئله قابل استفاده باشد، مدت‌ها یکی از کارهای بی اهمیت در هوش مصنوعی بود. تنها از اواخر دهه هفتاد بود که به‌عنوان یک کار مهم مجزا شناخته شد و یک زمینه تحقیق مجزا به‌عنوان بازنمایی دانش به وجود آمد. پیش از اینکه یک روش بازنمایی دانش برای کد کردن دانش در یک دامنه خاص استفاده شود باید پیش شرط‌هایی داشته باشد:

- توان کافی برای بیان دانش در یک حوزه خاص را داشته باشد.
- اساس معنایی روشنی ارائه دهد، به این صورت که معنی دانش بازنمایی شده در پایگاه دانش به آسانی قابل فهم باشد، به‌ویژه برای کاربر.
- اجازه تفسیر الگوریتمیک را فراهم آورد.
- اجازه بیان و اثبات راه حل‌های پیشنهادی را فراهم کند، یعنی بیان اینکه چرا یک سوال خاص از کاربر پرسیده می‌شود و یا چرا به یک نتیجه خاص رسیده است.

بخشی از این پیش شرط‌ها مربوط به فرم (ساختار) روش بازنمایی دانش می‌شود و برخی دیگر مربوط به معنی[1]. متاسفانه این نتیجه به دست آمده است که روش بازنمایی دانشی وجود ندارد که بتواند همه پیش شرط‌ها را برآورده کند. به‌عبارت دیگر به نظر می‌رسد که توان توصیفی بالای یک روش بازنمای دانش با توان تفسیر آن روش بازنمایی دانش با یکدیگر در تضادند. همانطور که در فصل آینده خواهیم دید با محدود کردن توان تفسیر یک روش بازنمایی دانش (به‌طوری که دانش آن حوزه به‌طور کامل بازنمایی شود) باعث می‌شود که به روش بازنمایی دانشی برسیم که اجازه توصیف کافی را فراهم نمی‌آورد. با پربارتر شدن ایده‌ها در سال‌های اخیر، سه روش بازنمایی دانش توجه بیشتری را به خود جلب کرده‌اند که عبارتند از:

- منطق
- قواعد تولید
- شبکه‌های معنایی و فریم‌ها

در سه فصل مجزا تلاش خواهیم کرد تا به این سوال پاسخ دهیم که دانش چگونه می‌تواند با استفاده از روش بازنمایی دانش، بازنمایی شود؟

برای هر کدام از روش‌های بازنمایی دانش، متدهای ویژه‌ای وجود دارد که به منظور مدیریت دانش مرتبط با آن به کار می‌رود.

1 semantic

به‌دست آوردن اطلاعات جدید از دانش موجود، استنتاج یا استدلال نامیده می‌شود. با توانایی‌های کامپیوترهای دیجیتال آغازین، استدلال خودکار در منطق، به یکی از نخستین موضوع‌های پژوهشی تبدیل شد که نتایج آن نیز عبارت بود از اثبات برخی از قضایای ریاضی توسط کامپیوتر. هرچند در این حوزه نیز تضاد اصلی بین بیان آشکار و کوتاه مورد نیاز منطق برای بازنمایی مسائل ریاضی و تفسیر مناسب آن به زودی روشن شد. الگوریتم‌هایی که کارایی بالایی داشتند با مشکل نبود منطق کاربردی روبرو بودند. در سال ۱۹۶۵، جی. ای. رابینسون[1] یک قاعده استنتاج عمومی به‌دست آورد که با نام اصل رزولوشن[2] شناخته می‌شود و اثبات قضایای ریاضی را شدنی‌تر می‌کند. این اصل، اساس شاخه برنامه‌نویسی منطقی و زبان برنامه‌نویسی پرولوگ بود. در برنامه‌نویسی منطقی، از منطق برای بازنمایی مسئله دلخواه استفاده می‌شود. ویژگی‌های منطقی می‌تواند با مفسری مبتنی بر رزولوشن اجرا شود. تلاش‌های دیگری نیز در دهه شصت توسط افرادی مانند سی. سی. گرین[3] برای استفاده از منطق به‌عنوان روش بازنمایی دانش در حوزه‌هایی غیر از ریاضیات صورت گرفت. آن زمان این سیستم‌های اثبات قضایا با عنوان سیستم‌های پاسخ به سوالات شناخته می‌شدند و امروزه ما آنها را با نام سیستم‌های خبره مبتنی بر منطق آغازین می‌شناسیم. هر چند منطق در سیستم‌های خبره کلاسیک به‌عنوان یک روش بازنمایی دانش برای تخصصی سازی و محدود کردن دامنه مسئله و البته کد کردن دانش انتخاب می‌شد، اما به ندرت پیش می‌آمد که به‌عنوان روش بازنمایی دانش در پیاده‌سازی سیستم خبره استفاده شود (با این وجود سیستم‌های خبره زیادی با استفاده از پرولوگ پیاده‌سازی شده‌اند). از سوی دیگر، همانطور که خواهیم دید فکر کردن به روش‌های دیگر بازنمایی دانش مانند برخی از حالت‌های خاص منطق، به فهمیدن معنی آنها کمک بزرگی می‌کرد و بسیاری از ویژگی‌های ناشناخته آنها را آشکار می‌کرد. آن وقت بود که ما فهمیدیم پایه‌های استوار منطق می‌تواند به مهندس دانش در ساختن سیستم‌های خبره کمک کند، هر چند که از روش‌های دیگری برای پیاده‌سازی حقیقی سیستم خبره استفاده شده باشد. در فصل دوم به بازنمایی دانش به روش منطقی و استدلال منطقی با فرمول‌های منطقی خواهیم پرداخت. این فصل برای ما روشن خواهد کرد که منطق چگونه می‌تواند در ساختن یک سیستم خبره به کار رود.

از دهه شصت تاکنون تلاش‌های زیادی در هوش مصنوعی برای توسعه روش‌های بازنمایی دانش به غیر از منطق شده است که منجر به ابداع قواعد تولید و فریم‌ها و . . . شده است. برای هر کدام از این روش‌ها، متدهای استنتاج ویژه‌ای که استنتاج منطقی را همانندسازی می‌کند به وجود آمده است. به‌طور معمول دو گونه از این استنتاج‌ها از سایر آنها متمایزترند. واژه‌های استنتاج بالا به پایین یا استنتاج هدفمند، که برای نمایش نوعی از استنتاج به‌کار می‌رود که در آن چند هدف آغازین داده شده‌اند و زیرهدف‌ها با استفاده از دانش موجود در پایگاه

1 J. A. Robinson

2 resolution principle

3 C. C. Green

دانش و داده‌های موجود به‌دست می‌آیند. نوع دوم استنتاج، استنتاج پایین به بالا یا استنتاج مبتنی بر استخراج یا اشتقاق[1] داده‌هاست. زمانی‌که از این نوع استنتاج استفاده می‌کنیم، داده‌های جدید از داده‌های موجود مشتق شده و به دانش موجود در پایگاه دانش اضافه می‌شوند. این فرآیند تا زمانی که اشتقاق دیگری برای به‌دست آوردن اطلاعات جدید وجود نداشته باشد، ادامه می‌یابد. تفاوت اساسی بین استنتاج بالا به پایین و استنتاج پایین به بالا در استدلال با قواعد تولید[2] بیشتر به چشم می‌آید، هر چند این دو نوع استنتاج با توجه به زمینه مورد بحث به سادگی از همدیگر تمیز داده می‌شوند. روش بازنمایی دانش بر اساس قواعد تولید و روش‌های استدلال مرتبط با آن موضوع فصل سوم خواهد بود.

فصل چهارم مرتبط با سومین رویکرد اصلی در بازنمایی دانش می‌باشد و شبکه‌های معنایی[3] و فریم‌ها[4]. این روش‌های بازنمایی دانش با ساختار سلسله مراتبی در ذخیره اطلاعات شناسایی می‌شوند. چون شبکه‌های معنایی و فریم‌ها چندین ویژگی مشترک دارند و شبکه‌های معنایی به‌طور معمول از اجداد روش بازنمایی فریم به حساب می‌آیند لذا این روش‌های بازنمایی دانش در یک فصل ارائه می‌شوند. روشی که در شبکه‌های معنایی و فریم برای بازنمایی دانش به‌کار می‌رود، وراثت نامیده می‌شود.

همانطور که گفته شد سیستم‌های خبره برای حل مسائلی از زندگی حقیقی استفاده می‌شوند که راه حل از پیش تعریف شده‌ای در منابع و کتاب‌ها ندارند. به‌طور معمول دانش صریح موجود درباره یک موضوع خاص، ناقص و غیر قطعی است. یک انسان خبره بیشتر می‌تواند با استفاده از دانش با کیفیت نامناسب، راه حل خوبی برای یک مسئله ارائه دهد، بنابراین هدف پژوهشگران سیستم‌های خبره این است که سیستم خبره‌ای بسازند که بتواند با اطلاعات ناقص و غیرقطعی همانند یک فرد خبره برخورد نماید. چندین مدل برای استدلال با عدم قطعیت ارائه شده است که درباره برخی از اینها در فصل پنجم بحث خواهیم کرد.

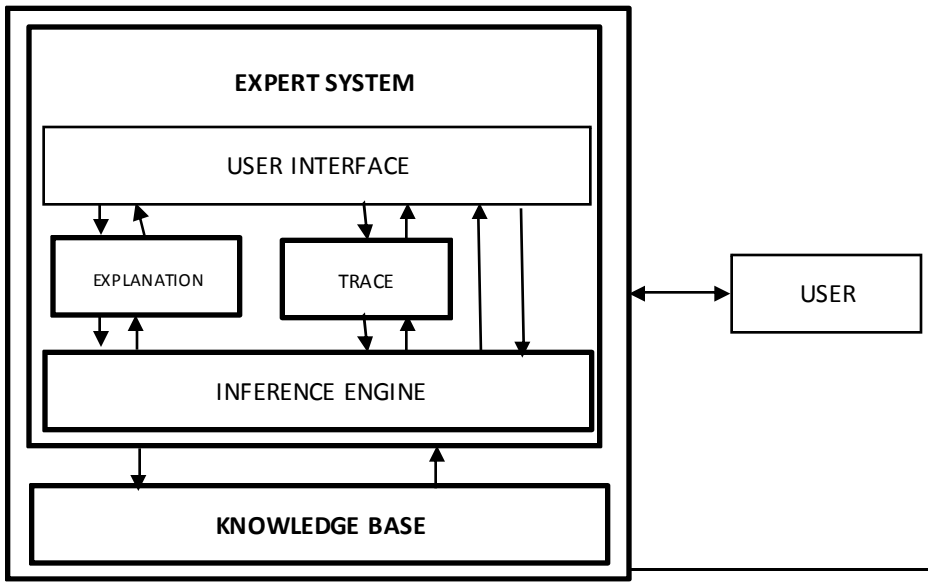
1 Derivation

2 Production Rules

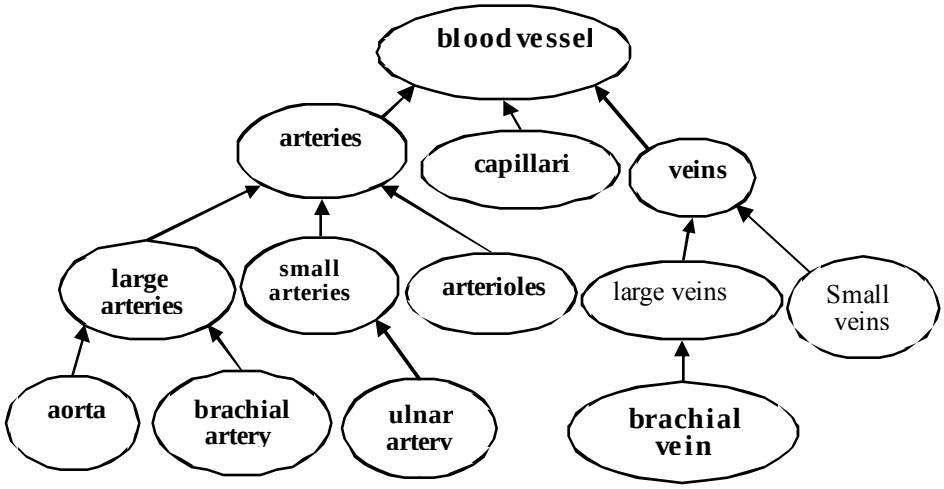
3 Semantic networks

4 Frame

موتور استنتاج پوسته یک سیستم خبره معمولی بخشی از یک به اصطلاح *سیستم هم‌فکری*[1] است. یک سیستم هم‌فکری شامل یک واسط کاربر برای ارتباط با کاربر می‌باشد که به‌طور معمول به صورت جلسات پرسش و پاسخ می‌باشد. افزون براین کاربر سیستم خبره و مهندس دانش نیاز به انواع گوناگونی از تسهیلات دارند که بتوانند کارکرد موتور استنتاج و درستی محتویات پایگاه دانش را بررسی نمایند. تسهیلات توضیح این امکان را فراهم می‌کند تا کاربر بتواند در هر لحظه‌ای از هم‌فکری بتواند از پایگاه دانش بپرسد که چگونه به این نتیجه رسیده است، چرا یک سوال خاص را پرسیده یا چرا به فلان نتیجه مخالف آنچه که گفته است، نرسیده است. با استفاده از *تسهیلات ردیابی*[2] که در سیستم هم‌فکری موجود است، رفتار استنتاجی سیستم می‌تواند به صورت مرحله به مرحله و یک گام در هر مرحله دنبال شود. روشن است که بیشتر این امکانات به درد مهندس دانش می‌خورند و برای فرآیند خطایابی و . . . استفاده می‌شوند و کاربر پایانی به‌طور معمول به این امکانات نیازی ندارد. فصل ششم به این امکانات می‌پردازد. شکل۱-۱ ویژگی‌های اصلی معماری یک سیستم خبره ساخته شده بر اساس پوسته سیستم خبره را نشان می‌دهد.



شکل ۱-۱- معماری عمومی یک سیستم خبره



شکل ۱-۲- دسته‌بندی رگ‌های خونی

پیش‌تر گفتیم که به جز پوسته سیستم‌های خبره ابزارهای بسیار قدرتمند سازنده سیستم‌های خبره نیز وجود دارند که برای ساخت سیستم‌های مبتنی بر دانش و سیستم‌های خبره استفاده می‌شوند. فصل هفتم درباره برخی از این ابزارها صحبت می‌کند. نخست OPS5 که یک زبان برنامه‌نویسی *خاص منظوره*[1] برای طراحی سیستم‌های تولید است مورد بحث قرار می‌گیرد و سپس LOOPS که یک محیط چندمنظوره برای پشتیبانی از برنامه‌نویسی شیء گرا در سیستم‌های خبره می‌باشد، مورد بررسی قرار می‌گیرد. افزون‌بر این در فصل هفتم به CENTAUR خواهیم پرداخت که یک سیستم خبره اختصاصی است که در آن روش‌های استدلال گوناگون و چندین روش بازنمایی باهم ترکیب شده‌اند.

در این کتاب چندین برنامه مورد بحث قرار می‌گیرند که به زبان‌های پرولوگ[2] و لیسپ[3] نوشته شده‌اند و هدفشان بیان چند تکنیک پیاده‌سازی برای ابزارها یا پوسته‌های توسعه سیستم‌های خبره می‌باشد. برای کسانی که با این زبان‌ها آشنا نیستند، پیوست‌هایی برای یادگیری مقدماتی این زبان‌ها ارائه شده است.

1 Special-Purpose

2 PROLOG

3 LISP

۱-۴- دامنه یک مسئله

در هر کدام از سه فصل بعدی، یک روش بازنمایی خاص و متدهای استنتاج مربوط به آن مورد بحث قرار خواهد گرفت. وقتی درباره این روش‌های گوناگون بازنمایی دانش بحث می‌کنیم، ممکن است مثال‌هایی از یک مسئله پزشکی مطرح شود که دامنه این مسئله سیستم گردش خون انسان می‌باشد. به این منظور سیمای کلی مسئله در اینجا ترسیم می‌شود.

سیستم گردش خون شامل قلب و یک شبکه بزرگ از رگ‌های خونی متصل به آن می‌باشد. رگ‌های خونی به سه دسته تقسیم می‌شوند. سرخرگ‌ها، سیاهرگ‌ها و مویرگ‌ها. این دسته‌ها دوباره طبق شکل۲-۱ دسته‌بندی می‌شوند. آئورت، سرخرگ بازویی و سیاهرگ و سرخرگ زند زیرین[1] برخی از این رگ‌های خونی هستند. آئورت خون را به کمک مویرگ‌ها از قلب به بافت‌ها منتقل می‌کند. سرخرگ‌ها به وسیله دیواره ضخیمی که شامل لایه نازکی از سلول‌های عضلانی می‌باشد، شناخته می‌شوند. بیشتر زمان‌ها سرخرگ‌ها حامل خونی می‌باشند که سطح اکسیژن بالایی دارد. برخلاف سرخرگ‌ها، سیاهرگ‌ها خون را از اندام‌ها به قلب باز می‌گردانند. آنها دیواره به‌طور نسبی نازکی دارند که شامل رشته نازکتری از بافت عضلانی به نسبت سرخرگ‌ها می‌باشد اما در عوض رشته‌های ارتباطی بیشتری با اندام‌ها دارند. خون موجود در سیاهرگ‌ها به‌طور معمول از نظر اکسیژن فقیر است.

میانگین فشار در سرخرگ‌ها به نسبت بالاست. برای نمونه میانگین فشار خون در آئورت حدود ۱۰۰میلی‌متر جیوه و میانگین فشار در سرخرگ زند زیرین ۹۰ میلی‌متر جیوه است. در داخل سیاهرگ‌ها فشار خون به‌طور نسبی کمی برقرار است. جدول ۱-۱ چکیده‌ای از اندازه‌های فشار خون برای بخش‌های گوناگون دستگاه گردش خون را نشـان می‌دهـد. یکی از استثناهای دسته‌بندی رگ‌های خونی سرخرگ ریوی می‌باشد. این سرخرگ خون را از قلب به ریه‌ها انتقال می‌دهد و دیواره ضخیمی دارد. با توجه به ویژگی‌های رگ‌ها، این رگ جزء دسته سرخرگ‌ها به شمار می‌آید اما با این وجود خون با اکسیژن کم را انتقال می‌دهد و فشار خون به‌طور نسبی پایینی حدود ۱۳ میلی‌متر جیوه دارد- هر چند این فشار بیشتر از فشار خونی است که در سیاهرگ ریوی جریان دارد.

جدول ۱-۲ قطر رگ‌های خونی متعلق به دسته‌های گوناگون را نشان می‌دهد. درصد خون جاری در هر دسته از رگ‌های خونی در جدول ۱-۳ آمده است. قلب هم ۷ درصد کل خون موجود را در خود جای می‌دهد. اینکه سیستم گردش خون چگونه کار می‌کند به‌طور معمول با یک مثال سیستم مکانیکی - هیدرولیکی بیان می‌شود که از یک پمپ (قلب)، مجموعه‌ای از لوله‌های متصل به هم (رگ‌های خونی) و منبعی که به پمپ متصل است (سیستم اعمال فشار) که از آب پر شده است.

جدول ۱-۱: میانگین فشار خون برای برخی از دسته‌های رگ‌های خونی

1 Ulnar Artery

نوع	میانگین فشار خون(میلی‌متر جیوه)
سرخرگ‌ها	۴۰–۱۰۰
سرخرگ‌های بزرگ	۹۰–۱۰۰
سرخرگ‌های کوچک	۸۰–۹۰
سرخرگ‌های خیلی کوچک	۴۰–۸۰
سیاه‌رگ‌ها	<۱۰

اطلاعات بازنمایی شده در بحث بالا می‌تواند قابل کاربرد در هر کدام از موارد مربوط به سلامتی انسان بالغ باشد، زیرا وجود بیماری در یک فرد خاص ممکن است باعث تغییر در هر کدام از پارامترها شود. برای نمونه در بیماری که دچار تنگی سرخرگ است، فشار خون دیستال (در سمت دور از قلب) آئورت به‌طور تقریبی نزدیک به صفر است.

فشار خون میانگین بیمار، از میانگین فشار خون بیمار در بازه‌های زمانی ثابت به دست می‌آید. اگر فشار خون ثبت شود یک منحنی مانند شکل ۱–۳ به دست می‌آید. فشار خون بین فشار خون حد بالا که (انقباضی) سیستولیک[1] نامیده می‌شود و فشار خون حد پایین که (انبساطی) دیاستولیک[2] نامیده می‌شود، نوسان می‌کند.

جدول ۱–۲–قطر انواع گوناگون رگ‌های خونی

نوع	قطر
سرخرگ‌های بزرگ	۱–۲.۵ سانتی‌متر
سرخرگ‌های کوچک	۰.۴ سانتی‌متر
سرخرگ‌های خیلی کوچک	۳۰میکرومتر
سیاه‌رگ‌های بزرگ	۱.۵–۳ سانتی‌متر

1 systolic
2 Diastolic

سیاه‌رگ‌های کوچک	۰.۵ سانتی‌متر

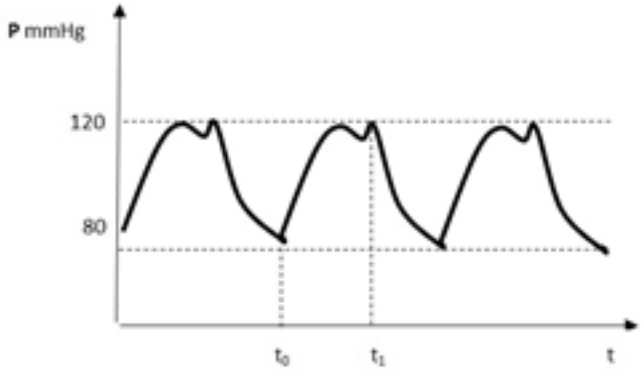
فرض کنید که پزشک فشار خون را در یک بیمار اندازه می‌گیرد. با استفاده از اطلاعات مربوط به بازه‌های زمانی اندازه‌گیری فشار، میانگین فشار خون در بازه زمانی $[t_0,t_1]$ با استفاده از فرمول زیر قابل محاسبه است.

$$\bar{P} = \frac{\int_{t_0}^{t_1} P(t)\,dt}{t_1 - t_0}$$

که $\bar{P}$ میانگین فشار خون، t_0 و t_1 نقاطی در زمان، که P(t) (فشار خون) در آن اندازه‌گیری شده است. در عمل، فشار حد بالایی و حد پایینی با استفاده از دستگاه فشارسنجی به‌دست می‌آید که به بازو بسته می‌شود. افزون‌بر فشار خون در برخی بیماران، بازده قلبی[1] نیز تعیین می‌شود. بازده قلبی مقدار خونی است که در یک دقیقه توسط قلب به آئورت پمپ می‌شود که به‌طور احتمالی با فرمول زیر محاسبه شود:

CO=F.SV

که CO بازده قلبی، F نرخ ضربان قلب و SV[2] حجم خون پمپ شده به آئورت در هر ضربان می‌باشد. F و SV می‌توانند اندازه‌گیری شده و ثبت شوند.



شکل ۱-۳-اندازه‌گیری کننده فشار خون

جدول ۱-۳- درصد کل خون موجود در رگ‌های گوناگون

1 Cardiac output
2 Stroke Volume

نوع	درصد خون
سرخرگ‌های بزرگ	۱۱
سرخرگ‌های کوچک	۷
سرخرگ‌های خیلی کوچک	۲
مویرگ‌ها	۹
سیاه‌رگ‌های بزرگ	۳۹
سیاه‌رگ‌های کوچک	۲۵

نوع دانش مرتبط با سیستم گردش خون انسان که در بالا بازنمایی شـد، دانش عمیق نامیـده می‌شود. دانش عمیق نیاز به دانستن جزئیات ساختاری و عملکرد دامنه مورد بحث دارد. دانش عمیق ممکن است در استدلال تشخیصی[1] مفید باشد. این نوع استدلال در یافتن علت خطا در سیستم‌ها کاربرد دارد. برای نمونه اگر فشار خون یک فرد بیمار پایین باشد، ما از ساختار و عملکرد سیستم گردش خون می‌فهمیم که ممکن است خطا در توانایی بیرون دادن خون کافی توسط قلب باشد و یا خطا در رگ‌های خونی به علت نداشتن مقاومت کافی در برابر جریان خون باشد و یا حجم خون برای پر کردن سیستم کم باشد. در بیماری که از اختلال گردش خون رنج می‌برد، علائم بیماری، نشانه‌ها و داده‌های گردآوری شده از آزمایش‌هایی که به بیمار توصیه شده‌اند، باید مربوط به آن بیماری باشند. به عبارت دیگر، وجود علائم اصلی یک بیماری، نشانه‌ها و نتایج آزمایش‌ها باید در راستای تشخیص بیماری‌های گردش خون استفاده شوند. به‌عنوان نمونه اگر بیماری هنگام راه رفتن دچار گرفتگی پا می‌شود که پس از یک یا دو دقیقه برطرف می‌شود، برای او احتمال گرفتگی سرخرگ قابل تصور است. این نوع از دانش بیشتر دانش سطحی نامیده می‌شود تا بتوان آن را از دانش عمیق تشخیص داد. همان‌طور که می‌بینید این دانش ارتباطی با ساختار و عملکرد سیستم گردش خون ندارد و تنها در تشخیص به کار می‌رود. در عوض وابستگی تجربی نوع خاصی از گرفتگی عضله و برخی بیماری‌های گردش خون به‌عنوان مدرکی برای تشخیص گرفتگی سرخرگ استفاده می‌شود. بسیاری از سیستم‌های خبره تنها از دانش سطحی برخوردارند، زیرا این نوع دانش در عمل برای مدیریت سریع مسائلی که پیش می‌آید، کاربرد دارد. با این وجود استفاده از دانش عمقی به‌طور مکرر باعث توجیه بهتر راه حل ارائه شده توسط سیستم می‌شود. مثال‌های دیگری از کاربرد دانش عمقی و سطحی در فصل‌های آینده خواهد آمد.

1 Diagnostic Reasoning

تشخیص سریع دانش سطحی و عمقی در دامنه مسئله همیشه کار آسانی نیست. به مثال زیر از تشخیص پزشکی توجه کنید. اگر فشار خون سیستولیک یک بیمار بیش از ۱۴۰ میلی‌متر جیوه باشد و در بررسی‌های پزشکی صدای مورمور شنیده شود و یا بزرگ شدگی قلب مطرح باشد، ممکن است بازگشت خون (سوراخ شدگی دریچه آئورتی) علت آن باشد. در این مثال دیدیم که از تعداد محدودی از ساختارهای سیستم گردش خون برای تشخیص استفاده شده است. مثال زیر از قلمرو تشخیص پزشکی، توصیف ما از دامنه مسئله را به پایان می رساند: وقتی که یک بیمار از درد شکمی رنج می‌برد و با استفاده از گوشی صدای مورمور در ناحیه شکم شنیده می‌شود و با معاینه یک توده ضربان‌دار دیده می‌شود، به احتمال زیاد نشانه‌ها و علائم مربوط به آنوریسما[1] (تورم شریان شکمی) باشند.

تمرین‌ها

۱–۱– یکی از پرسش‌هایی که از نخستین روزهای تولد هوش مصنوعی مطرح شد، این بود که آیا ماشین‌ها می‌توانند فکر کنند؟ هنوز هم این پرسش جزء بحث‌های داغ می‌باشد. این پرسش به صورت واضح و روشن توسط تورینگ در مقاله Computing machinery and intelligence بیان شد که در Mind، جلد ۵۹، صفحه ۲۳۶، سال ۱۹۵۰ چاپ شده است. مقاله آلن تورینگ را بخوانید و تلاش کنید، فکر کنید وقتی کسی از شما این سوال را بپرسد، پاسخ شما چه خواهد بود؟

۱–۲– توصیف GPS در بخش ۱–۱ از این فصل را بخوانید. فرآیند خرید را با استفاده از واژه‌های "حالت آغازین"، "حالت پایانی" و "انتقالات"، همان‌طور که در GPS مورد نیاز است، بیان کنید.

۱–۳– یک بخش مهم از سیستم HEURISTIC DENDRAL، زیرسیستم تولید ساختار است که ساختارهای مولکولی باورپذیر را تولید می‌کند. برنامه‌ای در LISP یا PROLOG بنویسید که همه‌ی ساختارهای ترکیبی آلکان مورد نظر را که فرمول آن C_nH_{2n+2} می‌باشد برای $n=1, 2, \ldots, 8$ تولید کند.

۱–۴– زمینه‌های مهندسی دانش و مهندسی نرم‌افزار مشترکات زیادی دارند، هر چند که تفاوت‌هایی نیز دارند. چه شباهت‌ها و تفاوت‌هایی بین آن دو می‌بینید؟

۱–۵– چند مثال از دانش عمقی و سطحی از دامنه مسئله‌ای که با آن آشنا هستید، بیاورید.

۱–۶– چند مورد از مسائلی را بیان کنید که سیستم‌های خبره می‌توانند بخشی از یک کمک حقیقی برای حل آنها باشند.

---

۱ مترجم: به نقل از ویکی پدیا: آنوریسما عبارت است از بزرگ شدن یا بیرون‌زدگی دیواره یک سرخرگ در اثر ضعف دیواره سرخرگ. آنوریسما معمولاً در آئورت یا سرخرگ‌هایی که مغز، پاها، یا دیواره قلب را تغذیه می‌کنند، ایجاد می‌شود.

فصل ۲

منطق و رزولوشن

منطق، یکی از قدیمی‌ترین روش‌های بازنمایی دانش است. روشی که با نحو و معنایی خوش تعریف شناخته می‌شود و تعدادی قواعد استنتاج فراهم می‌کند تا فرمول‌های منطقی را بر اساس فرم آنها پردازش کرده و دانش جدیدی به‌دست آورد. منطق دارای بیشینه‌ای کهن و غنی است که ریشه‌اش به یونان باستان می‌رسد: ریشه منطق به‌طور احتمالی به ارسطو می‌رسد. با این وجود تا قرن بیستم طول کشید تا پایه‌های ریاضیِ منطق مدرن توسط تی. اسکولم[۱] و جی. هربراند[۲] و کی. گورل[۳] و جی. گنتزن[۴] پی‌ریزی شود. کار این ریاضی دانان بزرگ و پرنفوذ، پایه‌های منطق را سال‌ها پیش از تولد علوم کامپیوتر، استوار کرد.

از آغاز دهه پنجاه که کامپیوترهای دیجیتال ابداع شدند تا کنون پژوهش‌های زیادی در زمینه استفاده از منطق برای حل مسائل توسط کامپیوتر انجام شده است. این پژوهش‌ها از دیدگاه‌های گوناگون انجام شده‌اند. برخی از این پژوهش‌ها به مکانیزه کردن اثبات ریاضی علاقه‌مند بودند: تولید با کارایی بالای اثبات ریاضی هدفشان بود. یکی از آنها ام. دیویس[۵] بود که در سال ۱۹۵۴ برنامه‌ای کامپیوتری ساخت که قادر به اثبات چند قضیه از نظریه اعداد بود. بزرگترین موفقیت این برنامه اثبات این بود که جمع دو عدد زوج، عددی زوج است. پژوهش‌های دیگر بیشتر علاقه‌مند به مطالعه در زمینه انسانی روش حل مسئله و به‌ویژه در زمینه هیورستیک بودند. برای این پژوهشگران استدلال ریاضی نقطه شروعی برای مطالعه هیورستیک بود و منطق اساس ریاضیات را تشکیل می‌داد. آنها از منطق به عنوان زبانی آسان برای بازنمایی رسمی استدلال انسانی استفاده کردند. مثال کلاسیک از این رویکرد به اثبات قضایا برنامه‌ای است که توسط ای. نیول[۶]، جی. سی. شاو[۷] و اچ. ای. سیمون[۸] در سال ۱۹۵۵

---

1 T. Skolem
2 J. Herbrand
3 K. Gödel
4 G. Gentzen
5 M. Davis
6 A. Newell
7 J.C. Shaw
8 H.A. Simon

نوشته شد و *ماشین تئوری منطق*[1] نام گرفت. این برنامه قادر به اثبات چند قضیه از کتاب اصول ریاضی ای. ان. وایتهد[2] و بی. راسل[3] بود. در سال ۱۹۶۱ جی. مک کارتی[4] و دیگران به این نتیجه رسیدند که اثبات قضایا می‌تواند می‌تواند برای مسائل غیر ریاضی نیز استفاده شود. این ایده توسط افراد دیگری نیز بسط داده شد. نخستین کار مشهور در زمینه به اصطلاح سیستم‌های پاسخ به سوالات، توسط جی. آر. اسلگل[5] و آخرین کار در این زمینه توسط سی. سی. گرین[6] و بی. رافائل[7] انجام شد.

پس از برخی موفقیت‌های آغازین روشن شد که قواعد استنتاج شناخته شده تا آن زمان برای کاربرد در کامپیوترهای دیجیتال، آن‌گونه که تصور می‌شد، مناسب نیستند. بیشتر پژوهشگران هوش مصنوعی علاقه خود به کاربرد منطق را از دست دادند و به ساخت روش‌های جدید و کارا برای بازنمایی و پردازش اطلاعات پرداختند. پیشرفت سریع با ساخت یک قاعده کارا و انعطاف‌پذیر در سال ۱۹۶۵ به دست آمد و رزولوشن نامیده شد. رزولوشن اجازه کاربرد منطق برای حل خودکار مسائل توسط کامپیوتر را فراهم می‌کرد. این اتفاق باعث شد تا در پایان، اثبات قضایا جایگاه خود در هوش مصنوعی و علوم کامپیوتر را به دست آورد.

منطق می‌تواند به‌طور مستقیم به‌عنوان یک روش بازنمایی دانش در ساخت سیستم‌های خبره استفاده شود، هر چند امروزه در عمل تنها در مقیاس کوچک استفاده می‌شود. اما معنای روشن و واضح منطق باعث شد، این روش بازنمایی دانش به عنوان نقطه شروعی برای درک صحیح از سایر روش‌های بازنمایی دانش استفاده شود. در این فصل نخست با آغاز از منطق گزاره‌ای درباره اینکه دانش چگونه می‌تواند توسط منطق بازنمایی شود، بحث خواهیم کرد. منطق گزاره‌ای با وجود داشتن محدودیت در توصیف، در معرفی برخی مفاهیم بسیار مفید است. منطق مسندی مرتبه نخست که زبان بسیار غنی‌تری برای بازنمایی دانش ارائه می‌دهد، در بخش ۲-۲ مورد بحث قرار خواهد گرفت. بخش اصلی این فصل به جنبه‌های الگوریتمیک کاربرد منطق در سیستم‌های استدلال خودکار و به‌ویژه رزولوشن اختصاص دارد.

۲-۱- منطق گزاره‌ای

منطق گزاره‌ای ممکن است به‌عنوان یک زبان بازنمایی دانش شناخته شود که اجازه توصیف و استدلال با عباراتی را فراهم می‌آورد که می‌توانند true یا false باشند. مثال‌های زیر نمونه‌هایی از چنین عباراتی هستند.

1 Logic Theory Machine
2 A.N. Whitehead
3 B. Russell
4 J. McCarthy
5 J.R. Slagle
6 C.C. Green
7 B. Raphael