

**ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ**

А. Ж. Асамбаев

ЖАСАНДЫ ИНТЕЛЛЕКТ НЕГІЗДЕРІ

оқулық

*Қазақстан Республикасы Білім және
ғылым министрлігі бекіткен*

Алматы, 2011

УДК 004(075.8)
ББК 32.973 я 73

*Павлодар мемлекеттік педагогикалық институтының
ғылыми Кеңесі баспаға ұсынады*

Пікір жазғандар:

Техника ғылымдарының кандидаты, профессор **А. Д. Тастенов**;
Педагогика ғылымдарының кандидаты, профессор **Б. А. Найманов**

А. Ж. Асамбаев

Жасанды интеллект негіздері: Оқулық. Алматы, ЖШС РПБК «Дәуір»
2011 ж. – 136 б.

ISBN 978-601-217-242-3

Оқулықта жасанды интеллект туралы негізгі бастапқы мәліметтер берілген, білімдерді ұсыну және есептерді шешу әдістер туралы негізгі мағлұматтар келтірілген. Білімдерді тауып алу әдістер туралы қысқа деректер, сараптамалық жүйенің құрылымы, жасау кезеңдері туралы жеткілікті мәліметтер берілген, оларды жасау үшін құрал-жабдықтардың жіктеуі көрсетілген.

Оқулық – жоғары оқу орындарының ақпараттық технологиялар, информатика, физика-математика мамандықтарының студенттері мен ұстаздар қауымына және жасанды интеллект әдістерін өз жұмыстарында қолданатын инженерлерге арналған.

ISBN 978-601-217-242-3

© А. Ж. Асамбаев
© ҚР Жоғары оқу орындарының
қауымдастығы, 2011

КІРІСПЕ

Адам мен машинаның бірлескен қызметі өте актуалды. Компьютер адамға кеңесші болу қажет, жағдайды тез талдап іс-әрекет варианттарын кісіге ұсыну керек. Ал адам келтірілген варианттардан керегін таңдайды.

Қазір жасанды интеллект - шешімді жасау мен қабылдау үшін адамның қызметін күшейтетін, өзін-өзі үйрететін құрал-сайман.

Орын ауыстыру, билет сату (яғни тізбекті және дәлдікті орындауы мен дәл нәтижесін алуы қажетті есептер) есептерді шешу үшін формалданған орындаушының басқаруы алгоритмдық пайдаланушы үшін бағдарлама түрде орындалады. Осында пайдаланушының қолында нақты анықталған командалар жүйесі мен орта болады, олар шешілетін есептің және басқарудың сипатына әсер етеді. Бірақ, формалдауға жатпайтын есептерде бар. Осындай есептер типіне эвристикалық есептерді, тиімді басқару есептерді, бейнені айырып тануды, мәтінді машиналық аударуын және т.б-ды жатқызуға болады. Бұл есептер интеллектке жатады, осындай есептердің компьютерлік құралдар көмегімен шешу проблемасы кибернетика бағытының зерттеу ядросын құрастырады, оны *жасанды интеллект* деп атайды.

Жасанды интеллект (ЖИ) – компьютерлік құралдар арқылы жеке ақыл-ой іс-әрекеттерін орындау. ЖИ жеке ғылыми бағыт ретінде XX ғасырдың екінші жартысында пайда болды (бұл көбінесе, кибернетиканың дамуына тәуелді болатын). Басқару талдау, салыстыру, ақпаратты өңдеу, болжамды жасау, жорамалдың дұрыстығын дәлелдеу (яғни интеллектуалды қызметіне жататын операциялар) негізінде шешімді қабылдаумен байланысты болады.

Жасанды интеллект жүйелері – арнайы логикалық жүйелер арқылы компьютерлік бағдарламада жүзеге асырылған адам интеллектің жеке аспектісін ұдайы өндіретін техникалық жүйелер.

Жасанды интеллект жүйелерінің қолдану салалары:

- роботтехникасы;
- сараптамалық жүйелер;
- есептерді әмбебап шешушіні жасау;
- бір тілден екінші тілге аудару, мәтінді рефераттау;
- пайдаланушыны компьютермен ыңғайлы диалогын қамтамасыз ететін интеллектуалды интерфейсты жасау.

Өткен ғасырдың соңғы он жылдығында интеллектуалды жүйелер дамуының келесі маңызды бағдарлары анық көрінді:

1. Шығармашылық үдерістерді еліктейтін жүйелер. Музыкалық шығармаларды жасау, ойын есептерді (шахмат, дойбы, домино) шешу, автоматтандырылған аударма, теоремаларды дәлелдеу, бейнелерді айырып тану, ойлауды еліктеу және т.с.с.

2. Білімдерге негізделген (сараптау жүйесі) ақпараттық жүйелер, яғни жабдықтарды күйге келтіру, тәжірибелі емес пайдаланушыларға кеңес беру, оқыту және т.б.

3. Интеллектуалды ақпараттық жүйелер – заттық салада есептерді шешуге арналған математикалық және алгоритмдық модельдерге негізделген үлкен және өте үлкен бағдарламалар. Олардың мүмкіндігі: басқаруды оңайлату және адамның жұмыс көлемін азайту үшін пайдаланушымен мағыналы сұхбат жүргізу.

4. Роботтық техника. «Интеллектуалдығы» жағынан бірнеше робот буындарын ажыратады. Бірінші буын – алдын ала бекітілген және өзгермейтін бағдарлама бойынша істейтін робот-манипуляторлар (мысалы, станокқа дайындамаларды әперетін). Екінші буын – бейімделген роботтар. Осындай роботтар әртүрлі датчиктермен (бұрышөлшеуіштер, тензометрлер, газ талдаушылар және т.б.) жабдықталған. Роботтардың алғашқы екі түрі - өнеркәсіптік роботтар, олар арнайы ортада (зауыт цехында) жұмыс істеуге арналған. Ұқсас проблемалар іздестіру роботтар да пайда болады, бұл ерекше бағдарламалар Ғаламторда құжаттарды индексациялауға арналған.

Автор

1-БӨЛІМ

ЖАСАНДЫ ИНТЕЛЛЕКТІҢ ТҰЖЫРЫМДАМАЛЫҚ НЕГІЗДЕРІ

1.1. Жасанды интеллект теориясының даму тарихы

Қазіргі ғылымның кез келген бағыттары сияқты жасанды интеллектің даму тарихы өте бай.

Автоматтандырылған технологиялардың кең дамуы (1.1 кестені қараңыз) XX ғасырдың екінші жартысында басталды. Жасанды интеллектің негізгі теориялық қағидалары дәл сол заманда орнықты деп айтуға болады.

Жасанды интеллект теориясында салмақты өзгерістер мен жетістіктерді алдымыздағы он - жиырма жылда күтуге болады.

Кесте 1.1

Ақпараттық технологиялардың даму кезеңдері

Кезең	Шешілетін міндеттер	Ғалымдар және инженерлер	Техника
Машиналардың алдындағы (XVII ғ.)	Есептерді (көбінесе, арифметикалық) автоматтандыру	Г. Лейбниц, Б. Паскаль	Арифмометр, логарифм сызғышы және т.с.с.
Механикалық және механика-электр есептеуіш машиналар (XIX ғ. – XX ғ. ортасы)	Есеп жүргізу (тауарларды есепке алу, халық санағы, ғылым). Хабарларды шифрлеу	Ч. Бэббидж және А. Лавлейс, Г. Голлерит, П. Чебышев	Голлерит машинасы, электржетегі бар арифмометр, «Энигма» шифрлеу машинасы
Алғашқы ЭЕМ (XX ғ. 40 жылдарынан бастап)	Бірталай деректерді енгізу, сақтау, өңдеу	Фон Нейман (теориялық негіздері), Кодд Э. (реляциялы деректер базасы, 60 жылдар).	ENIAC (АҚШ, ядролы бағдарлама үшін есептер). Мейнфреймдер, ЕС ЭВМ (70 жылдары)
Алғашқы басқарушы есептеуіш машиналар (XX ғ. 50 жылдарынан бастап)	Техникалық қарапайым заттардың, күрделі жүйелердің параметрлерінің басқару және бақылау.	Н. Винер (теориялық негіздері), К. Шеннон, Д. А. Поспелов	Икемделген ЭЕМ, космос ракеталарын жүргізу жүйелері (ҚСРО) және т.б.
Басқарушы есептеуіш кешендер (XX ғ. 70 жылдары)	Ірі техникалық жүйелердің ақаулықтарын автоматты түрде диагностикалау. Ірі техникалық жүйелерді басқару мен жұмыс тәртіптерін таңдау	Р. Белман, Л. С. Понтрягин, М. Мински	Есептеуіш өлшеу кешендер СМ ЭВМ (PDP-11). Автоматтандырылған жобалау жүйесі, микро-ЭЕМ
Дербес компьютерлер және үлкен ЭЕМ (XX ғ. 80 жылдарынан бастап)	Көпөлшемді деректерді талдау, болып кеткен оқиғалардың себебін іздеу, қалыптасқан жағдайдан шығу жолдарды іздеу	Аристотель, Дж. Буль, Л. Заде, М. Мински, С. Пейперт, Р.Кини, Х. Райфа, Р. Ковальски, және басқалар.	Микро-ЭЕМ PDP-11, IBM/370, дербес компьютерлер IBM286, 386 және т.б.

1.2. Бағдарламалы құралдардың даму кезеңдері

Әрбір кезеңде (кесте 1.2) ақпараттың ерекше өңдеу технологиясы құрылды (мысалы, деректер базасының технологиясы). Кезеңдердің әрбіреуі өркендеп, келесі жоғарғы деңгейдің технологиясының негізі болып табылады.

Кесте 1.2

Бағдарламалы құралдардың кейбір кезеңдері

Технология	Міндеттер
ЭЕМ және сыртқы құрылғылардың физикалық деңгейлерін бағдарламалы басқару технологиясы	Операциялық жүйелерді жасау Эвристикалық программалау
ЭЕМ физикалық деңгейінің іс-әрекетін программалы-аппаратты оңтайландыру	Операциялық жүйелерді жетілдіру, арнайы утилиталарды жасау
Сақталу деректердің логикалық ұйымдастыру технологиясы	Файлдық деректер жүйесін жасау
Деректер базалар мен есептеу технологиясы	Деректерге қатынау әдістерді және стандартты форматтарды жасау, сандық алгоритмдер
Жасанды интеллект теориясы	Сақталу деректердің салыстырмалы талдауының әмбебап әдістерін жасау. Қайшылықтарды, олардың пайда болу себебін және жою әдістерін іздеу

Ескерту: эвристикалық программалау — белгілі, алдын ала берілген ережелер арқылы іс-әрекеттер стратегиясын жасау.

Әрбір технология өзіндік ерекше ұғымдарды, терминдерді және әдістерді туғызады. Келесі жоғарғы деңгейдің технологиялары оны қолдайды, бірақ жаңа ұғымдар да пайда болып жаңа теорияны құруға әкеледі. Жасанды интеллект үшін бұл білім туралы ұғым.

Сонымен, жасанды интеллект — өзіндік міндеттері бар, ерте технологияға негізделетін ең жаңа ақпараттық технология.

1.3. «Жасанды интеллект» ұғымы

1-ші анықтама. Ақпаратты кіруінің күйіне тәуелді, мақсатқа бағытталған іс-әрекет параметрлерінен басқа, өзінің мінез-құлқын өзгертетін жүйе *интеллектуалды* деп аталады (1.1-сурет). Мінез-құлқының тәсілі ақпаратты кіруінің ағымды күйіне ғана емес, жүйенің алдыңғы күйлеріне де тәуелді болады.



1.1-сурет. «Қара жәшік» сияқты интеллектуалды жүйе

Бірнеше мысал келтірейік.

Әрбір жанды организм — интеллектуалды жүйе. Оның ұзақ зердесі және өзін оқыту қабілеті бар. Қолын күйдіріп алған бала ыстық пешке жоламайды.

Техникалық жүйелер интеллектуалды бола алмайды, яғни олардың реакциясы бірдей оқиғаға айтарлықтай өзгермейді. Құбырдағы газ қысымын автоматты басқару жүйесі қалқалағышты жауып және ашып қана алады (параметрді басқару), бірақ құбырдан қалқалағышты алып тастау шешімді қабылдай алмайды. Егер апатқа келтіретін, газөткізгіштің ішінде қысым өзгеретін болса (мысалы, алдымен кенет көтеріледі, сосын кенет төмендейді), онда автоматты жүйе осы өзгерісті қалыпты деп есептеп, қалқалағыштың қозғауымен бұл жағдайды басқара бастайды. Бірақ, осындай оқиғалар қайталана берсе жүйе үйренбейді. Өйткені, бала параметрлерді (ыстық пеш) ғана емес, мінез-құлық ережелерді де жаттап алды (ыстық пешке жолама).

2 анықтама. Адамның ойлауын компьютерде модельдейтін жүйе *интеллектуалды* деп аталады.

Екінші анықтама 60 жылдары пайда болды, ол кезде адам миындағы процестерді компьютер көмегімен модельдеуге болады деп есептелетін. Миталшықтары – нейрондар – арнайы математикалық әдістермен бағдарламалы түрде сипатталатын. Осымен, компьютерлік бағдарлама адам миының тілімі деп келтірілетін. Бағдарламаның кіруіне кейбір мәліметтер берілетін (жанды организмде электр сигнал беріледі), шығуында алынған нәтижелер үлгі-нұсқамен салыстырылатын. Осыған байланысты, егер нәтижелер үлгі-нұсқауға сәйкес болмаса, есептік коэффициенттер өзгертілетін. Осындай «оқытудың» айналымдар санына тәуелді, бағдарламаның жұмыс нәтижелері адам миының өте кіші тілімінің жұмыс нәтижелеріне жақындайтын.

XX ғасырдың 80 жылдары бұл идея сәтсіздікке ұшырады, бірақ нейронды желілер теориясының пайдалығы басқа тәжірибелік салаларда көрсетілді. Мысалы, жақсы нәтижелер параметрлердің мәндерін болжау есептерде және бейнені айырып тануда алынды.

3 анықтама. Адаммен мағыналы өзара сұхбат жүргізу арқылы, оның интеллектуалды қызметін күшейтетін жүйені *интеллектуалды* деп атайды.

80 жылдардың аяғында әмбебап жасанды ақыл-ойды жасауға болмайтыны және оның қажетсіздігі айқын болды. Адамды алмастыратын емес, оның қызметін толықтыратын тар көлемді интеллектуалды жүйелерді жасау керекті. Адамның бірегей қабілеттерінен басқа бірнеше кемшіліктері де бар. Компьютерде энциклопедиялық жады бар, бір секундта миллиондаған операцияларды жасай алады, бірақ компьютер ой жүгірте алмайды және өзінің қылықтарына жауап бере алмайды.

Адам мен машинаның бірлескен қызметі өте актуалды. Компьютер адамға кеңесші болу қажет, жағдайды тез талдап іс-әрекет варианттарын кісіге ұсыну керек. Ал адам келтірілген варианттардан керегін таңдайды.

Сонымен, қазір жасанды интеллект - шешімді жасау мен қабылдау үшін адамның қызметін күшейтетін, өзін-өзі үйрететін құрал-сайман.

1.3.1. Терминдер және анықтамалар

Интеллектуалды бағдарламаны процедуралық деректер базалар және бағдарламау теория көмегімен ғана жасау мүмкін емес. Егер деректер базасына барлық кездескен жағдайларды енгізетін болсақ, бағдарлама одан интеллектуалды болып қалмайды.

Мысал. 40 жылдардың аяғында бір тілден екінші тілге автоматтандыру аудару міндеті қойылған болатын. Мысалы, орыс тілінен ағылшын тілге аудару тіркестер: Read Only Memory. Мұнда Read — читать (оқу), Only — только (тек қана), Memory — память (зерде). Осы сөйлемді қалай дұрыс аударуға болады?

1. Компьютер көзқарасына қарай — ағылшын сөздерді кездескен ретімен орыс сөздерімен алмастыру қажет, онда аламыз: «Читать только память».

2. Аудармашының көзқарасына қарай — ағылшын тілінің грамматикасын еске алып, сөйлемнің аяғынан бастау керек: «Память только для чтения».

Анық, бастапқы мәліметтерден басқа (біздің жағдайда бұл сөздер «читать, только, память») аудару ережелерді де еске алу қажет. Ал аудару ережелері деген не? Аударушы бағдарламада оларды қалай ескеру керек? Ол үшін жаңа ұғым білімдер пайдаланылады.

Білімдер — жасанды интеллектің негізгі атау сөзі. Қарапайым түрде білімдерді деректер элементтерінің арасындағы арақатынастар деп анықтауға болады. Мысалы, машинаның кіруіне берілген сөйлем: «Арман Әсемді жақсы көреді». Мұндағы Арман мен Әсем — мәліметтер. Жақсы көреді — арақатынас. Арақатынастың симметриясы жоқ, сондықтан машина осыны біліп, мағыналы жауап бере алады: «Әсем Арманды жақсы көрмеуі мүмкін».

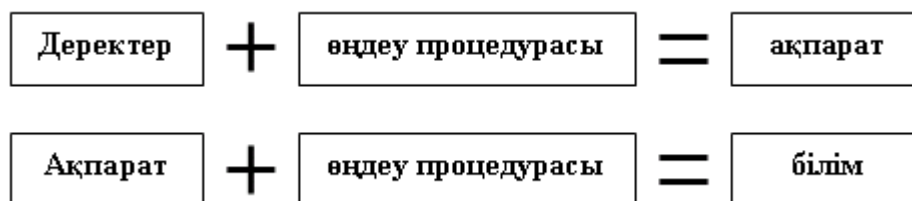
Аудару мысалға қайтадан келейік. Оны графикалық түрде көрсетейік (1.2-сурет). Аудару сызбасы аударудың дұрыс бағытын анықтауға мүмкіндік береді. Ол орыс пен ағылшын варианттарына бірдей сәйкес келеді. Бастапқы сөйлемде болмаған және сыртқы көздерден алынған осындай сызба білім деп саналады.



1.2-сурет. Деректер элементтерінің арасындағы байланыс

Басқа сөзбен айтқанда, білімдер бастапқы мәліметтерге (оларды декларативті ақпарат деп атауға болады) кейбір өңдеу әдістерді қолдау және

сыртқы процедураларды (оны қосылған процедуралар немесе процедуралық ақпарат деп атайды) қосу арқылы алынады. Мәндер арасындағы арақатынастарды анықтау үшін жаңа бағдарламалық құралдар қажет. Жасанды интеллект бағдарламасының жұмысы «білімдер арқылы шешімді» қабылдау немесе «жаңа білімдерді шығару» мен анықталады. Бұл процесс сызба түрінде 1.3-суретте көрсетілген.



1.3-сурет. Білімдердегі тұжырым

Мысалды қарастырайық. Адам сағатқа қарайды. Нені ол көреді? Мәліметтерді? Ақпаратты? Білімді? Адамның іс-әрекетін бақылайық. Көз циферблатқа қарап тұр. Миға электр сигнал түседі. Сосын кісі ойланып түсіну керек (сағат тілінің орнынан) қанша уақыт (яғни алынған мәліметтерді кейбір шкаламен салыстыру). Нәтижесінде декларативті ақпаратты алып, мысалы 08:20, кісі сыртқы білімді қосып (мысалы, өзінің жұмыс тәртібін), кешіккенін (білім) түсінеді және жылдам жүре бастайды (мінез-құлқының параметрлерін өзгертеді, яғни мәліметтерді).

Білімдердің ерекше белгісі – олар бастапқы жүйеде болмайды. Сағат циферблатында «мен кешіктім» деген жазу жоқ. Бұл сөз адамның жұмыс тәртібінде де жоқ. Ақпараттық бірліктерді салыстыру, арасындағы қайшылықтарды табу және шешу нәтижесінде білімдер пайда болады. Яғни, білімдер белсенді, олардың пайда болуы (немесе жетіспеушілігі) кейбір іс-әрекетке келтіреді.

Білімдер үшін келесі қасиеттер сипатты:

- ішкі түсіндірушілік (әрбір ақпараттық бірліктің бірегей аты болу керек және ол бір мәнді анықталу керек);
- құрылымдылық, яғни ақпараттық бірліктердің арасында арақатынастар анықталу қажет (мысалы, «бөлік - бүтін», «тегі - түрі» және т.б.)
- білімдер кейбір кеңістікті жасайды, бұл кеңістік метрлік немесе метрлік емес те болу мүмкін.

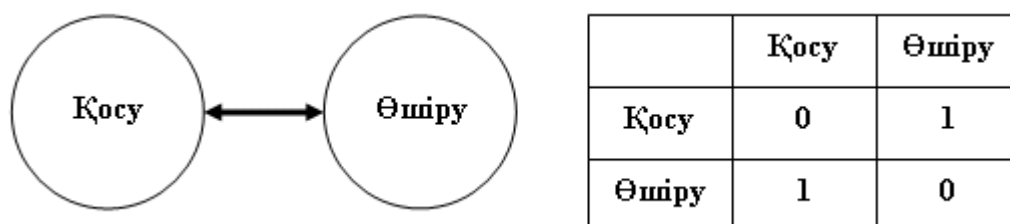
Ақпараттық бірліктер үшін аттарын қою, осы бірліктерді ажырату, арақатынастарды анықтау – күрделі есеп. Қарапайым құрылымы бар нақты объектілер үшін ақпараттық бірліктерге және арақатынастарға бір ғана ұсақтауы болады. Мысалы, «ит - құйрық» жүйесі үшін жалғыз ғана модель ойлап табуға болады «құйрық иттің бөлігі».

Күрделі, ең бірінші абстракты, жүйелер үшін, «бүтіні» және «бөлігі» қайсысы қайда түсінуге оңай емес. Қарастырайық, мысалы «доллар курсы»

жүйені, бұл үшін қайсысы маңызды параметр, қайсысы маңызды емес айқын айтаалмаймыз. Немесе көп мәнді арақатынасты «тию» ($A \rightarrow B$ -ға тиді) екенін әртүрлі жағдайға сәйкес түсіндіріп көріңіз.

Осындай проблемалардың шешуімен жасанды интеллекттің «білімдер инженериясы» деген саласы айналысады.

Күй – екінші маңызды атау сөзі. Әрбір ақпараттық бірліктер және толығымен барлық жүйе белгілі бір күйде болу мүмкін. Мысалы, шам бір күйде бола алады: қосылған - өшірілген. Бұл жағдайда күйлер кеңістігінің қуаты 2 (1.4-сурет қараңыз).



1.4-сурет. Жүйенің күйлер кеңістігі және ауысу кестесі

Шексіз көп күйлер кеңістігі бар жүйелер болады.

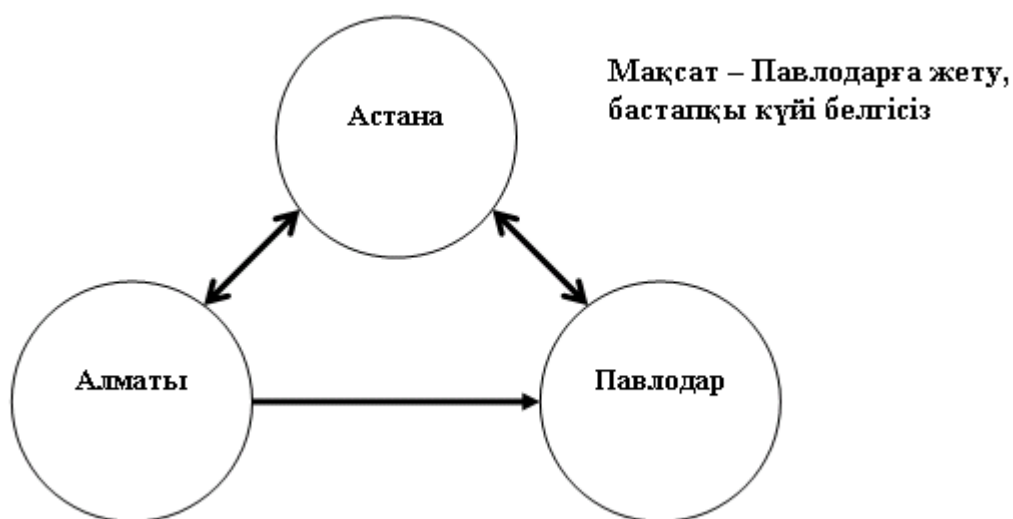
Күйлер кеңістігін граф (1.4-сурет) түрінде немесе ауысу кестесі түрінде көрсетуге ыңғайлы.

Жүйенің бір күйден екінші күйге ауысуында симметриялы қасиеті болу мүмкін. Осындай жағдайда жүйе бастапқы күйге қайту мүмкін (ал компьютерлік бағдарлама – циклденіп қалады).

Мақсат – жасанды интеллект жүйесінің негізгі сипаттамасы. Жасанды интеллект жүйесінің (интеллектуалды бағдарлама) дәстүрлі бағдарламадан айырмашылығы – мақсаттылық қасиетінде. Интеллектуалды бағдарламада алгоритмы жоқ, мақсаты ғана және осы мақсатқа жету ережелері бар. Жасанды интеллект бағдарламаны - «не істеу керек» бағдарламасы деп айтуға болады, ал дәстүрлі бағдарламама – «қалай істеу керек» бағдарлама. Дәстүрлі бағдарламада мақсатқа жету алгоритмы алдымен берілген және сөзсіз орындалады, интеллектуалды бағдарламада мақсатқа жету кезінде ағымды жағдайды еске алып алгоритм құрылады.

Жасанды интеллект жүйесінде істейтін бағдарламашыға дұрыс мақсатты қойып үйрену өте маңызды.

Интеллектуалды бағдарламаларда мақсат ретінде көбінесе, жүйенің берілген күйге ауысу талаптары тапсырылады, ал тиімділік критерий ретінде – шешімнің қадам саны. Әрине, бастапқы күйді көрсетпей мақсатқа жету көбінесе мүмкін емес (1.5-сурет).



1.5-сурет. Циклдары бар күйлер кеңістігі

Білімдер ұсыну моделі – жасанды интеллект модельдерінің негізгі типі. Нақты жүйелер бір ғана классикалық модельге сирек негізделеді. Нақты модельдер бұл – жиірек өзіндік болжалы, шектеулері және шарттылықтары бар классикалық модельдердің буданы. Олардың ішіндегі ең пайдалыларын эвристика деп атайды.

Заттық сала – есептерді шешу үшін жасанды интеллект теориясын қолдайтын адам қызметінің саласы. Мысалы, егер Сіз анализ арқылы ауруға диагноз қоятын сараптамалық жүйені жасасаңыз, онда Сіздің заттық салаңыз - медицина. Егер Сіз кемелерді қақтығысудан сақтайтын интеллектуалды жүйені жасасаңыз, онда Сіздің заттық салаңыз – навигация, және т.б.

Білімдер бойынша инженердің (ЖИ бойынша маман) ең бірінші міндеті әртүрлі жағдайларды шешетін ережелерді жазу (формалдау). Сонымен бірінші кезеңде маңызды ұғымдарды (заттық саланың «аттарын») белгілеу қажет.

Екінші кезеңде берілген заттық саласы үшін маңызды «аттарын» байланыстыратын арақатынастарды және осы «аттары» тұратын күйлерді бөлу керек. Мысалы, кемелерді қақтығысудан сақтайтын жүйе үшін маңызды аттар: «су», «кеме», «қақтығысу» және басқалар. «Су» деген сөздің бірнеше күйлері болады: «биік», «тайыз», «тынық», «долы» және т.б. «Су» мен «кеме» аттарының арасындағы қатынас «батырылу». Осы кезде түсінбеушілік болу мүмкін. «Батырылу» деген не? Ал егер кемең ауа жастығы бар болса? Ал су астындағы қайық болса? Сондықтан осы сияқты сұрақтарға заттық мамандар жауаптарды таба алмау мүмкін. Қатынастың «күші» туралы сұрақтар да маңызды: «қаншалыққа батырылған?», «жүзіп келе жатыр ма, әлде батып кетті?» және т.с.с. Бұл үдерісті «білімдерді табу» деп атайды.

Сонымен, білімдер бойынша инженердің негізгі міндеті - заттық саланы және бағдарламаларды жасау әдістерді анық білу емес, әмбебап

математикалық немесе айқын емес логикалық-лингвистикалық модельдерді құруда.

Заттық саласы қоятын негізгі шектеуі - бұрынғы белгілі бір модельдің қолдауының мүмкін еместігінде.

1.3.2. Жасанды интеллектің қазіргі зерттеу салалары

Өткен ғасырдың соңғы он жылдығында интеллектуалды жүйелер дамуының келесі маңызды бағдарлары анық көрінді:

1. Шығармашылық үдерістерді еліктейтін жүйелер. Музыкалық шығармаларды жасау, ойын есептерді (шахмат, дойбы, домино) шешу, автоматтандырылған аударма, теоремаларды дәлелдеу, бейнелерді айырып тану, ойлауды еліктеу және т.с.с.

2. Білімдерге негізделген (сараптау жүйесі) ақпараттық жүйелер, яғни жабдықтарды күйге келтіру, тәжірибелі емес пайдаланушыларға кеңес беру, оқыту және т.б.

3. Интеллектуалды ақпараттық жүйелер – заттық салада есептерді шешуге арналған математикалық және алгоритмдық модельдерге негізделген үлкен және өте үлкен бағдарламалар. Олардың мүмкіндігі: басқаруды оңайлату және адамның жұмыс көлемін азайту үшін пайдаланушымен мағыналы сұхбат жүргізу.

4. Роботтық техника. «Интеллектуалдығы» жағынан бірнеше робот буындарын ажыратады. Бірінші буын – алдын ала бекітілген және өзгермейтін бағдарлама бойынша істейтін робот-манипуляторлар (мысалы, станокқа дайындамаларды әперетін). Екінші буын – бейімделген роботтар. Осындай роботтар әртүрлі датчиктермен (бұрышөлшеуіштер, тензометрлер, газ талдаушылар және т.б.) жабдықталған. Роботтардың алғашқы екі түрі - өнеркәсіптік роботтар, олар арнайы ортада (зауыт цехында) жұмыс істеуге арналған. Нақты әлемде (мысалы, батып кеткен кемеңі зерттейтін) жұмыс істейтін роботтарға өте қиын: «қайда мен?», «алдымда не тұр?», «өтуге бола ма?» деген сұрақтарға жауап табу керек. Интеллектуалды роботтар өзінің датчиктер көрсеткіштерін және адам бұйрықтарын шын уақыт көлемінде өңдеу керек. Ұқсас проблемалар іздестіру роботтар да пайда болады, бұл ерекше бағдарламалар Ғаламторда құжаттарды индексациялауға арналған.

1.3.3. Жасанды интеллектің қазіргі теориялық мәселелері

1. Білімдерді ұсыну мәселесі

1.1. Нейрон желілер мәселесі

1.2. Көп критериялы шешімді қабылдау

1.3. Шешімді қабылдаудың стохастикалық моделі

1.4. Өте тар заттық салалар үшін модельдерді жасау

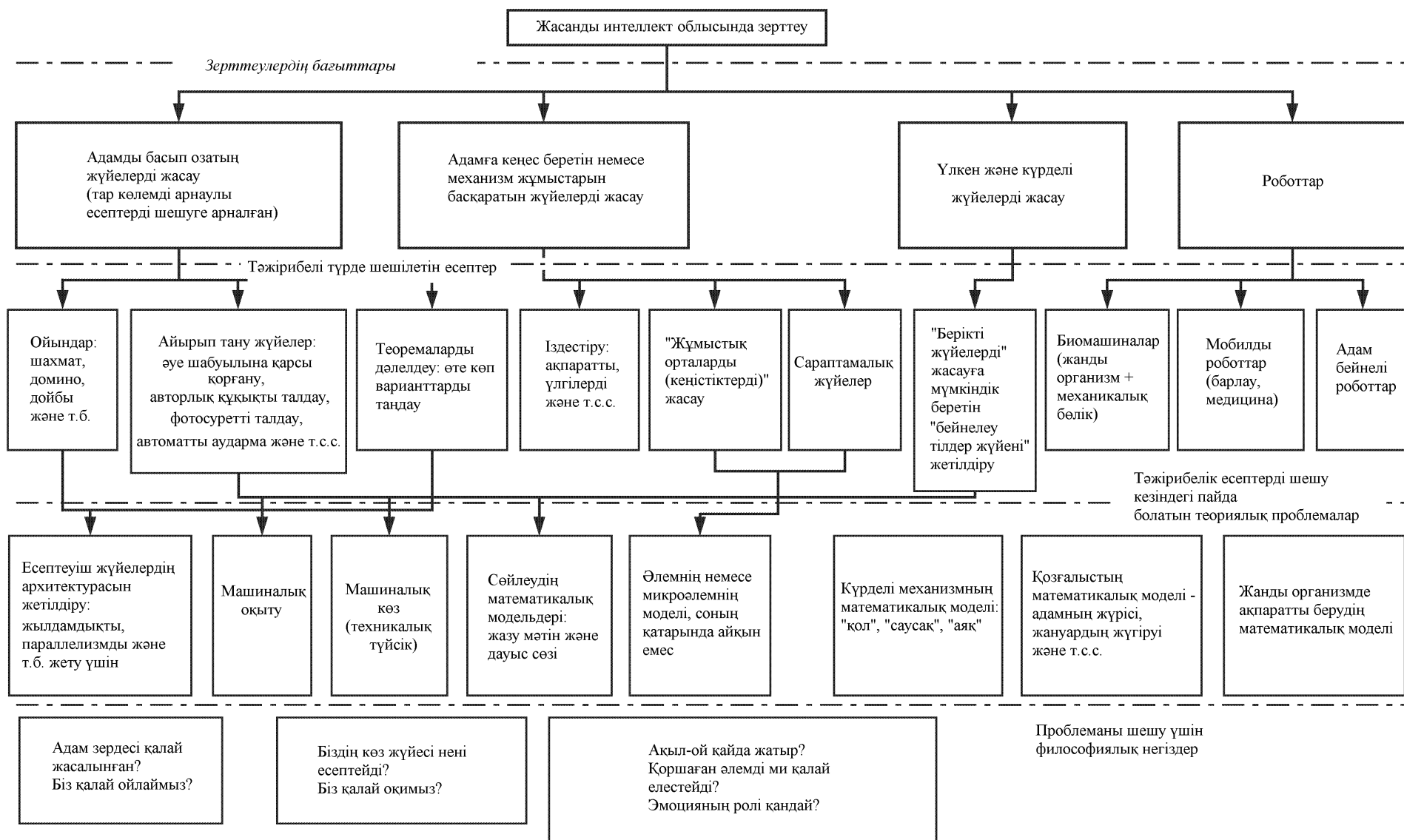
2. Компьютерлік лингвистиканы әзірлеу

- 2.1. Жаңа берікті бағдарламау тілдерді жетілдіру
- 2.2. Табиғи тілге негізделген роботтарды басқару тілін жетілдіру

3. Компьютерлік логика

- 3.1. ЭЕМ-ның архитектурасын жетілдіру

1.7-суретте қарастырылған сұрақтар жинақтау түрінде келтірілген.



1.7-сурет. Қарастырылған сұрақтар жинақталған түрде

2-БӨЛІМ

ЖАСАНДЫ ИНТЕЛЛЕКТІҢ НЕГІЗГІ ТЕОРИЯЛЫҚ МІНДЕТТЕРІ

2.1. Жасанды интеллект — зерттеудің пәнаралық саласы

Академиялық білімдердің әрбір саласында дәстүрлі есептердің жеткілікті тар аясы болады. Осындай есептердің және олардың тұжырымдамаларының тізбесі үсті-үстіне тарлау болып қалыптасады. Зерттеулердің саласы ақырында ондаған жеке есептерге бөлінеді, олардың біреулері есептелу мүмкін, басқаларының шешімдері болмайды.

Жүйелер теориясынан біз «пәнаралық тасымалдаудың» қажеттілігін, яғни әртүрлі ғылымдар есептердің тұжырымдамаларын және оларды шешетін жолдарын бір-бірінен ауысып алып пайдаланатынын білеміз.

Бір мысал келтірейік. Дәстүрлі медицинада арнайы «тамыр кеңейткіш» препараттардың енгізумен тамырдағы қан қысымының нормалау әдісі пайдаланады. Оның қолдануы, дегенмен, әрдайым болу мүмкін емес. Бұдан басқа кеңейтудің белгілі бір шегі бар. Сонымен қатар қысымды түсіру үшін қанның тұтқырлығын азайтуға талаптануға болады, бірақ бұл жағымсыз химиялық өзгерістерге желуі мүмкін.

Ұзақ уақыт бұл есептің шешімі табылмай тұрды. Жоғары айтылған проблемаларды инженер-реологтар білмеді, бірақ өзінің саласындағы (техникалық жүйеде сұйықтың қозғалысы) міндетті: «құбырдың диаметрі өзгермейтін және сұйықтың химиялық құрамы сақталу жағдайда құбырөткізгіштің өткізу қабілетін жоғарылату», – деп қойды.

Нәтижесінде, құбырдың диаметры мен сұйықтың тұтқырлығын өзгертпейтін, ағыстың сипатын (турбулентты тәртіптен ламинарлы тәртіпке) өзгертетін, арнайы қоспаларды пайдалануға негізделген әдіс ұсынылды. Осы әдістің медицинаға «кері тасымалдауы» сапалы эффект берді.

Қазіргі заманда ғалым өзінің зерттеу саласына ғана емес басқа шектес салаларға жататын есептерді білу қажет. Жасанды интеллект үшін бұл концепция өте маңызды, себебі жасанды интеллект – пәнаралық қолданбалы зерттеу саласы. «Қолданбалы» дегеніміз, өйткені жасанды интеллект ұғымы компьютерлік ғылымдардан және маңызды қолданбалы есептерден жеке бола алмайды.

2.2. Жасанды интеллектің дәстүрлі есептер тізбесі

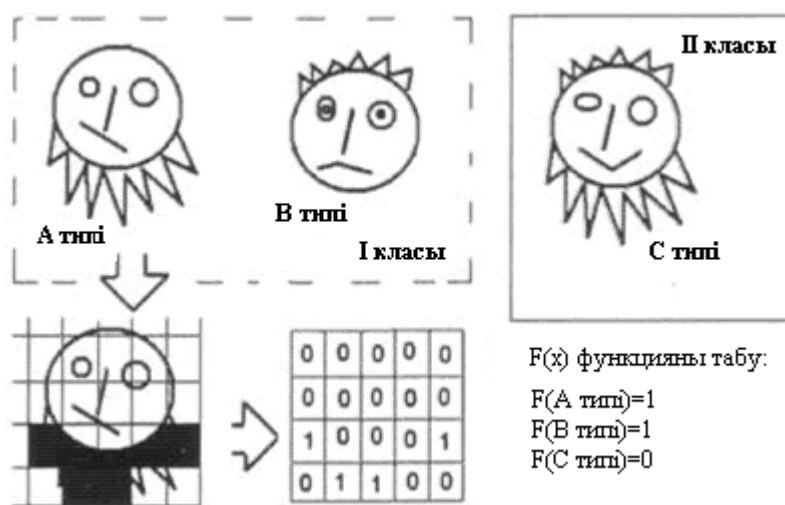
Жасанды интеллектің дәстүрлі есептердің ішінде, әрине жіктеу есебін ерекшелеу керек, яғни кейбір объекті белгілі немесе белгісіз объектілер көптігіне келтіру.

Осындай есеп мысалы, биологияда жаңа биологиялық түрін тапқан кезде болу мүмкін. Ірі айуан кездесе, табылған түрді белгілі бір класқа, мысалы, сүтқоректі класына жатқызуға болады.

Ұсақ организм, мысалы, құрттар үшін, бұл өте күрделі. Қайтадан ашылған түрлер, бірнеше әртүрлі бағыңыңқы түрлерге сипатты, белгілері болады. Сондықтан оларды белгілі бір класқа жатқызуы айқын емес (2.1-сурет).

Мысал: *A* объектіні қарастырайық. Оның белгісі «түс» пен сандық сипаттамасы «қою жасыл» және «қысқа» сипаттамасы бар «табан» белгісі. Және де екі класс белгілі: табаны жоқ қою жасыл объектілер кіретін *C1* класы және ұзың аяқты түрлі түсті объектілері бар *C2* класы. *A* объектіні қандай класқа жатқызу керек?

Басқа жағдай да мүмкін: екі белгісіз *C1* мен *C2* кластар бар. *A* объектісі *C1* класқа жатады, ал *B* объектісі – *C2* класқа. Топтастырудың негізіндегі белгілерді анықтау керек.



2.1-сурет. Топтастыру есебі

2.2.1. Есептердің тым қатал тұжырымдамалары

1. Топтастыру есебі: *C1*, *C2* кластар берілген; жаңадан келген объекті кластардың біреуіне жатқызу.

1.1. Кері есеп: *C1*, *C2* кластарға жататын *A*, *B* объектілер берілген; топтастыру белгілерін іздеу.

2. Іс-әрекет жоспарын жасау есебі

2.1. Граф үстінде жолды іздеу есебі

2.2. Өткен оқиғаларды түсіндіру есебі

2.3. Мүмкін болатын оқиғаларды болжау есебі

3. Баламаларды генерациялау есебі

3.1. Кейбір жолдаудан болатын барлық мүмкін баламалар. Дербес жағдайда, мысалы, кейбір датаға доллар курсы болжау есебі

3.2. Берілген қасиеттері ғана бар баламалар

4. Мүмкін болатын баламалар көптігінен ең жақсы баламаны талғау есеб.

4.1. Кездейсоқ талғау есебі

4.2. Критериалды талғау

5. Есептелетін параметрлері өте көп немесе олар қайшы болатын кейбір техникалық объекті басқару есебі

2.2.2. Жасанды интеллекттің міндетін тұжырымдау ережелері

Әрбір есепті шешу үшін қажетті: біріншіден, нақты бұйым саласы үшін оны дәлелдеу; екіншіден, керекті шешім параметрлерін таңдау (нәтижелердің дәлдігі мен сенімділігін, жұмсалған уақыты және басқалар).

Есепті дәлелдеу – бұл модельге тәуелсіз кезең; есептің қойылуын айқындағанда шешім әдістерін емес, есептің табиғатын еске алу керек. Демек, «не керек» немесе «қандай мақсатқа жету керек» нұсқауларды тарылтып, «қалай істеу» диапазонды кең қалдыру қажет.

Осындай қайшы талаптарды бір мезгілде қанағаттандыру үшін абстракты есепті тұжырымдау керек.

Абстракты есептердің мысалдары: тізімдерді өңдеу, ықтималдықты есептеу, жорамалды негіздеу және т.б.

Есепті қойылуы мысалы: шикі жұмыртқаны биік мұнарадан жерге сындырмай жеткізу. Шешімді тек жаңа әдіспен ғана табу керек. Осындай қарапайым есепті шешкенде, оқушылар көбінесе, жаңа әдісті табуға қиналады. Қойылу қателіктері мақсаттың тұжырымдауына (жұмыртқаны «лақтыру», не «түсіру») және сыртқы жағдайлары мен бастапқы күйлеріне (жұмыртқа жер бетінде жатыр, қатты емес бетке құлауы мүмкін, жұмыртқа қатырылған болу мүмкін және т.с.с.) қатысады. Кейін оқушылар «ең жаман» жағдайларды еске алдық деп түсіндіреді, бірақ есептің қойылуында ол туралы ештеңе айтылмаған.

2.2.3. Есепті шешудің параметрлерін таңдау

Маңыздылық жағынан бірінші параметр бұл шешімнің *дәлдігі*. Физикалық жүйелер үшін айқын және бізмәнді: килограмм, метр және т.с.с. сияқты өлшем бірліктер жүйесі бар Интеллектуалды жүйелер үшін жалпыға танылған әдістеме және өлшем бірліктер жоқ. Көбінесе, дұрыс шешімдердің пайызы туралы немесе алынған шешімдердің тәжірибелік жағдайдан ауытқу дәрежесі туралы айтады. Мысалы, медициналық жүйелер үшін 70%-дан астам дұрыс диагноздар саны жақсы деп есептеледі.

Екінші параметр – *орнықтылық* (сенімділікке қойылған талаптар) - әртүрлі жағдайда сынау кезінде жүйенің мінез-құлығына қатысты. Интеллектуалды жүйелер үшін бірдей кірістік мәліметтерде әртүрлі шығу мәліметтер болу мүмкін. Интеллектуалды жүйе жиірек аумалы болады - бұл оның күшті қасиеті. Мысалы, «Келесі жылдың 1 қаңтарында доллар курсы қандай болады?» – деген сұраққа алынған жауап: 145 теңге. Қайтадан осы

сұрақты қояйық – жауап аламыз: 145 теңге 25 тиын. Айырмасы неліктен шықты, себебі әрбір сұраныста жүйе жүздеген және мыңдаған айқын байланыспайтын фактілерді модельдейді. Осындай болжамның пайдасы – бірдей сұрақты жүздеген рет қайталап, біз оқиғалардың ең ықтимал дамуын түсіне аламыз.

Сенімділік (дәл әрқашан жоспарланған (бастапқы күйлер «шамамен бірдей» болғанда) нәтижеге әкелетін қабілет) пен адекватты (дәл жоспарланған (берілген дәлдігі бойынша) нәтижеге әкелетін қабілет) параметрлерден кейінгі мүмкін болатын шешім сапасының үшінші ең маңызды сипаттамасы – **үнемділік**. Шешім кейбір критерий бойынша немесе кейбір көзқарасы арқылы үнемді болады. Көбінесе, есептің шешу уақыты және есептің шешу құны туралы айтады. Кері вариант та болу мүмкін – қателіктің құны. Есептің шешу құны мысалы, мәліметтерді алудың жиынтық бағасынан тұруы мүмкін.

2.2.4. Шешу (білімдер ұсыну) моделін тандау

Таңдалған шешу параметрлері (ең үлкен дәрежеде) және есептің қойылуы (ең аз дәрежеде) бізге есептің шешу әдісін табуға «көмектеседі». Енді білім бойынша инженер анықтау керек – бұрын шешілген есептердің қайсысына бұл есеп ұқсайды және оның негізгі айырмашылығы неде? «Ұқсайтын» есептің шешу тәжірибесінен ол кейбір жалпы шешу жолын, құрылысын, процедураларын, мәліметтерін немесе «моделін» шеттен алып пайдаланады. Модельдің дұрыс тандауы өте маңызды. Жасанды интеллект теориясында «білімдер ұсыну моделі» (немесе «жағдайға байланысты басқару») деп аталатын бірнеше типтік модельдер қалыптасқан.

Оларды біз төменірек толық қарастырамыз.

Білімдер ұсынудың әрбір моделінде сенімділік бойынша және адекватты дәрежесі бойынша кейбір априорды шектеулері болады. Нақты есеп үшін керекті модель сапасының математикалық сипаттамаларын таңдалған модельдің тәжірибелік шын сипатталарымен салыстыру қажет. Мысал келтірейік. Жасанды интеллекттің ықтималдық моделінде детерминистық модельге қарағанда, адекваттылығы мен үнемділігі жақсырақ. Бірақ ықтималдық басқару жүйесіне ракетаны нысанаға көздеуді Сіз бере алмайсыз («статистика бойынша» осы жүйенің 100-ден 99 ракетасы тура нысанаға түсетін болсада).

Заттық саланың (бұл кейін Сіздің интеллектуалды жүйеңіздің жұмыс істейтін ортасы) әрдайым өзгеру қасиеті бар. Кейбір модельдер «әлем құбылмалық» қасиетін көбірек, кейбіреулері – азырақ еске алады.

3 - БӨЛІМ

ЖАСАНДЫ ИНТЕЛЛЕКТ ӘДІСТЕРІНІҢ ҚОЛДАНУ САЛАЛАРЫ

3.1. Жақсы және нашар құрылымданған заттық облыстары

Жасанды интеллект әдістері аз құрылымданған заттық салалар үшін тиімді, яғни іс-әрекет алгоритмы салалар үшін алдын ала белгілі.

Осындай салалар үшін кіру мәліметтерге айқындықсыз бен саралықсыз сипатта. Сонымен бірге, қабылданатын шешімдер бірмәнді, айқын және саралықты болу керек. Және де осындай шешімдердің тиімділігін болжау мүмкіндігі болған жөн.

Осындай салаларға: медицина, экономикалық менеджмент, күрделі техникалық объектілерді басқару, психология, лингвистика және т.б. жатады.

«Күрделі» жүйелер дегеніміз - іс-әрекет принциптері толығымен белгілі емес жүйелер. Мысалы, «программалау тілі» жүйесі бірінші курс студентіне күрделі болып көрінеді, ал бітіруші үшін бұл – жай үлкен (көп белгілі элементтері бар) жүйе.

Жасанды интеллект әдістерінің тиімді қолдау салаларына не жатпайды? Бұл - жақсы құрылымданған заттық салалар. Оларға біріншіден, дәлі және инженерлік ғылымдар: математика, физика, материалдар кедергісі, геодезия және т.б. кіреді. Себебі, осы ғылымдардың есептерін шешу үшін өзіндік берікті алгоритмдер мен тәсілдер баяғыдан бар.

Айта кетейік, есептеуіш техниканы енгізген кезде осы ескі және тексерілген алгоритмдер мен тәсілдер өзгереді. Сондықтан есептеуіш күрделілігі өсу немесе алгоритмның өзінің күрделенуі мүмкін. Компьютер болмаған дәуірде жасалынған алгоритмдер мен әдістер ең жақсы нәтижені алуға бағытталмай, минимал есептеуіш шығын арқылы нәтижені алуға арналған. Бұл алгоритмдер қолмен істеу есептеуге арналған болатын.

Басқа жақтан, алгоритмдер мен тәсілдердің күрделенуі, пайдаланушының алгоритмды және оны (бағдарламаны) тиімді басқаруының түсінбеуіне келтіру мүмкін. Бұл жағдайда пайдаланушы енгізген немесе деректер қорынан алынған, кіру және басқару мәліметтердің қате немесе жарамайтын мәндері, оптималды шешімді алу үшін автоматты түрде түзетулі керек.

Мысалы: пайдаланушы қағаз парағының ортасында «Жаңа Жылмен құттықтаймын!» сөйлемді басып шығарайын дейді. Бұл үшін ол осы сөйлемнің алдынан 16 «бос орын» белгісін тереді. Интеллектуалды мәтін редакторы пайдаланушының тәжірибесіздігін сезетін болса, сұрақ қояды: «Сіз мәтінді ортаға дәл келтіруге тырысасыз ба?». Пайдаланушының жауабына байланысты редактор не бос орындарды тегістеу белгімен ауыстырады, не нұсқаулықты шығарады, не бәрін өзгеріссіз қалдырады, егер бұл мәтіннің тегістеуі болмаса.

Сонымен, есепті біліп алу, бағдарламаның пайдаланушымен «есті» диалог негізінде жүргізу мүмкін. «Ақылды» бағдарлама адамның қызмет

бабын анықтағаннан кейін оған кәсіптік бағытталған, параметрлердің мәніне қатысты сұрақтар қояды (мысалы, инженер-жобалаушынан температуралар диапазонын сұрауға болады) немесе пайдаланушының жұмыс мақсаты бойынша жалпы тұжырымдамалық сұрақтар қойылуы мүмкін. Мақсатты басқа әдіспен де айқындауға болады, мысалы стохастикалық (оймен шешу әдісі).

Сөйтіп, жасанды интеллект әдістердің қолдануы дәстүрлі жақсы құрылымданған салалар үшін де мүмкін. Мысал ретінде химияны айтуға болады. Prospector деген жасанды интеллект бағдарламасы бар, ол берілген бастапқы заттардан алынған барлық химиялық қоспаларды зерттеуге мүмкіндік береді.

Түрлі заттық салаларда ЖИ әдістердің қолдануының басқа себебі - шешімнің жылдамдығы. Есептің шешіміне әртүрлі талаптар қойылу мүмкін, мысалы, нақты уақытты бұзбау (нақты уақыт тәртібі) талабы, яғни бағдарламаның жұмыс процесін кейбір басқа физикалық немесе программалық процеспен үйлестіру. Осындай есептер аз емес және қазіргі инженерлік тәжірибеде бірінші орынға шығып отыр. Мысалы, жауынгерлік машинаның қаруын басқару есебі. Ешқандай жауынгер бір мезгілде бірнеше нысананы байқай алмайды, нысанаға 10 ракет ата алмайды. Бұл жағдайда адам тек нысананы белгілеу керек. Ал ракетанын жіберуін, нысанаға атуын және жоюын ЖИ жүйесі қамтамасыз етеді.

Ал осындай жүйеге адам керек пе? Қазірге кезеңде – сөзсіз керек. ЖИ жүйелері әлі жеткілікті «ақылды» емес. Басқаратын ЖИ жүйесінің іс-әрекетіне адам жауапты. Бірақ кейбір мамандар осындай сұрақты мағынасыз деп есептейді, олардың пікірі бойынша, сұрақ: «Осындай жүйеде адамның рөлі қандай?» – деп қойылу керек.

Интеллектуалды машиналардың басқа қолдану бағыты - білім беру саласы, мысалы компьютер көмегімен оқыту.

Кейбір салада оқыту, арнайы жобаланған жағдайда, оқу ақпараттың өте көп қайталау циклдерін орындайды. Мысалы, ұшақ қозғалысын басқару, ағылшын сөздерді және сөйлемдерді жаттау және т.с.с. Осындай процесте мұғалімнің бар болуы қажет емес.

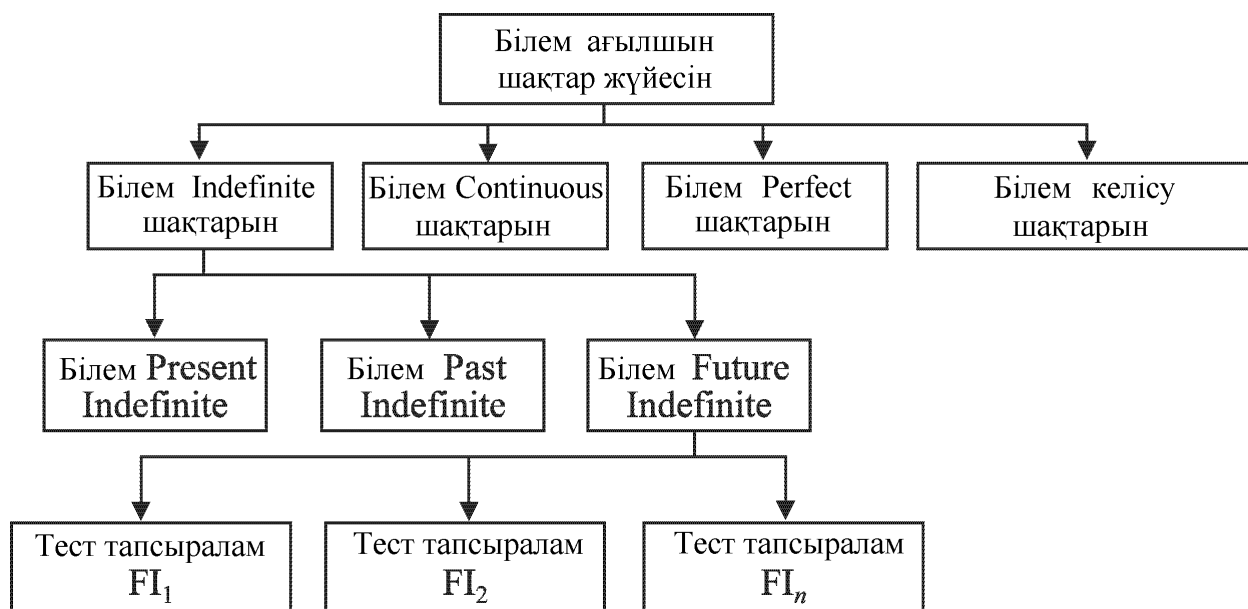
Машина қателіктерді өзі түзете алады, бақылау және үйрету жаттықтыруды бере алады және оқыту әдістемесі бойынша (тәжірибелі мұғалім енгізген арнайы ережелер негізінде) кеңес бере алады. ЭЕМ көмегімен жеке дағдыларға және ептілікке үйрету арзандау және тиімділеу, өйткені ЭЕМ-ға бәрібір неше рет бірдей қателікті түзету керек, ол ешқашан шаршамайды. Одан басқа, тестілеу нәтижесіне негізделіп, оқыту әдістемесінің өзін де жетілдіруге болады. Бұл мүмкіндікті машинасыз технологияда іске асыру өте қиын.

Осындай жүйенің жұмысын мысал көмегімен түсіндірейік. Кезінде ақпараттық білім беру орталығы «Chorin» ұйымдастырылған болатын. Бұл бағдарламалардың және электронды әдістемелік материалдардың арнайы кешені болып, шетел тілге үйретуді «бір оқытушы – көп студент» принципі

бойынша ұйымдастыратын. Студент электронды оқулықтармен, электронды кітапханамен, электронды фонотекамен және тестілеу қорымен кез келген уақытта жұмыс істей алады.

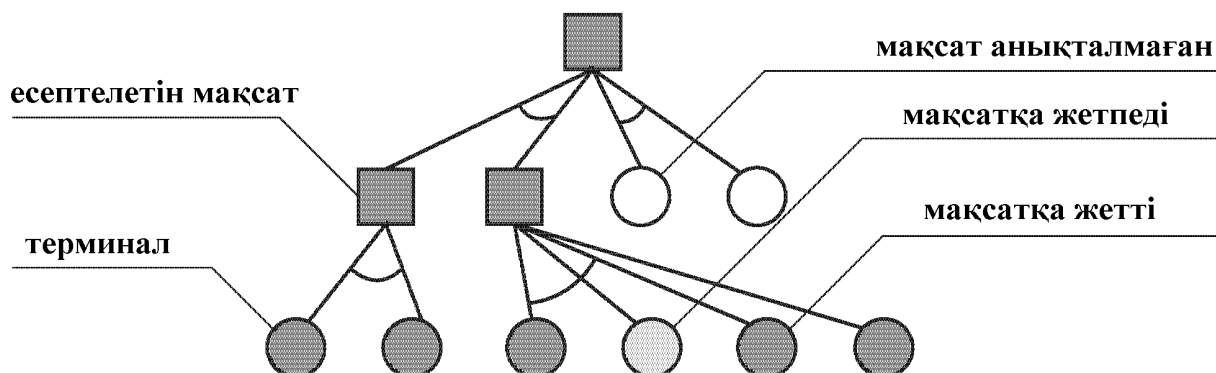
Компьютермен өзіндік жұмыс барысында студентте кейбір қиындықтар пайда болу мүмкін. Оқушы жаттығуларды орындағанда - «оларды ол дұрыс орындап жатыр ма» білмейді.

Егер компьютерлік бағдарлама қателіктерді түзетсе, онда оқушы сұрау «неге осылай» деп сұрауы мүмкін. Ол бұл сұраққа оқытушыдан жауап дереу ала алмайды және оқулықтың қай бөлімінде жауап бар білмейді (немесе оны түсінбейді). Барлық процесс компьютерде өткендіктен, оны деректер қоры (ДҚ) көмегімен толық құжаттауға болады. Осындай ДҚ оқушылар бойынша кейін қарап шығып, оқытушы әрбір студентке жеке кеңес бере алады. Осы жоспарды құру – ДҚ-нан тандауға қарағанда, күрделілеу есеп. «Chopin» жүйесінде бұл арнайы «мақсаттар ағашы» көмегімен жасалынады, яғни оны оқушының моделі деп айтуға болады (3.1-сурет).



3.1-сурет. «Мақсаттар ағашы» мәндерді ұсыну моделі ретінде

Жұмыс кезінде оқушы тестілерді тапсырады, ал ағаштың түйіндері автоматты түрде түрлі түсті боялады (3.2-сурет).



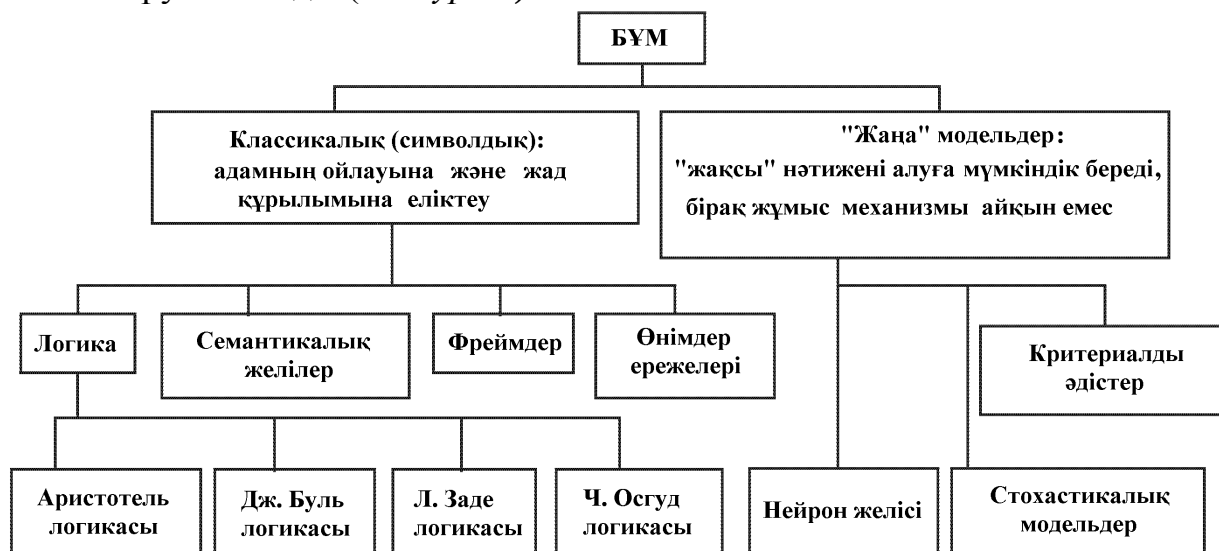
Осы принциптер «үйрету ортасы» жасауымен жүзеге асырылады. Бұл желілік бағдарламалық қамтамасыз етудің ерекше класы, оған оқу ақпараттың орасан зор көлемі, бақылау және диагностика құралдар, кеңес беретін жүйелер, нақты пайдаланушыға бейімделген бағыныңқы жүйелер және тағы басқалар кіреді. Оқу ақпараттың күн сайын артуы «Ғаламтор негізінде оқыту» (Web learning, Web-based learning) деген арнайы аталымның пайда болуына әкелді.

Ақпаратты іздеу жүйелер – ғаламдық ақпараттық қойманы құру негізі. Мысал ретінде «жұмсақ іздестіруді» айтуға болады. Іздестіру жолында (мысалы, Интернеттегі іздестіру машинада) біз «жаңа жылғы торт рецепті» деген сұраныс енгізейік. Шындығында, сұрауға релевантты құжаттар тек «торт» пен «жаңа жыл» бөлімде ғана емес, басқа әртүрлі бөлімдерде де: «пісіру», «мереке дастарқаны» болуы мүмкін. Қазіргі іздестіру машиналар торт, торттар, рецепт және т.б. сөздері бар барлық құжаттарды жай ғана тауып береді.

Кейде медицина мен әскери істе шешімнің жылдамдық пен айқындығы емес, мүмкін болатын шешімдер саны, мысалы, заңды қабылдағанда барлық мүмкін кемтіктерді және қайшылықтарды есептеу; кеткен сарапшылардың тәжірибесін сақтап алу және т.б. маңызды.

3.2. Білімдер ұсыну моделдері

ЖИ жүйелерде қолданылатын нақты модельдер тізбесі кең және оларды топтастыру қиын. Нақты жүйелердегі модельдердің көбі дұбаралы және эвристикалы болады. Ең танымал модельдерді мына түрде топтастыруға болады (3.3-сурет).



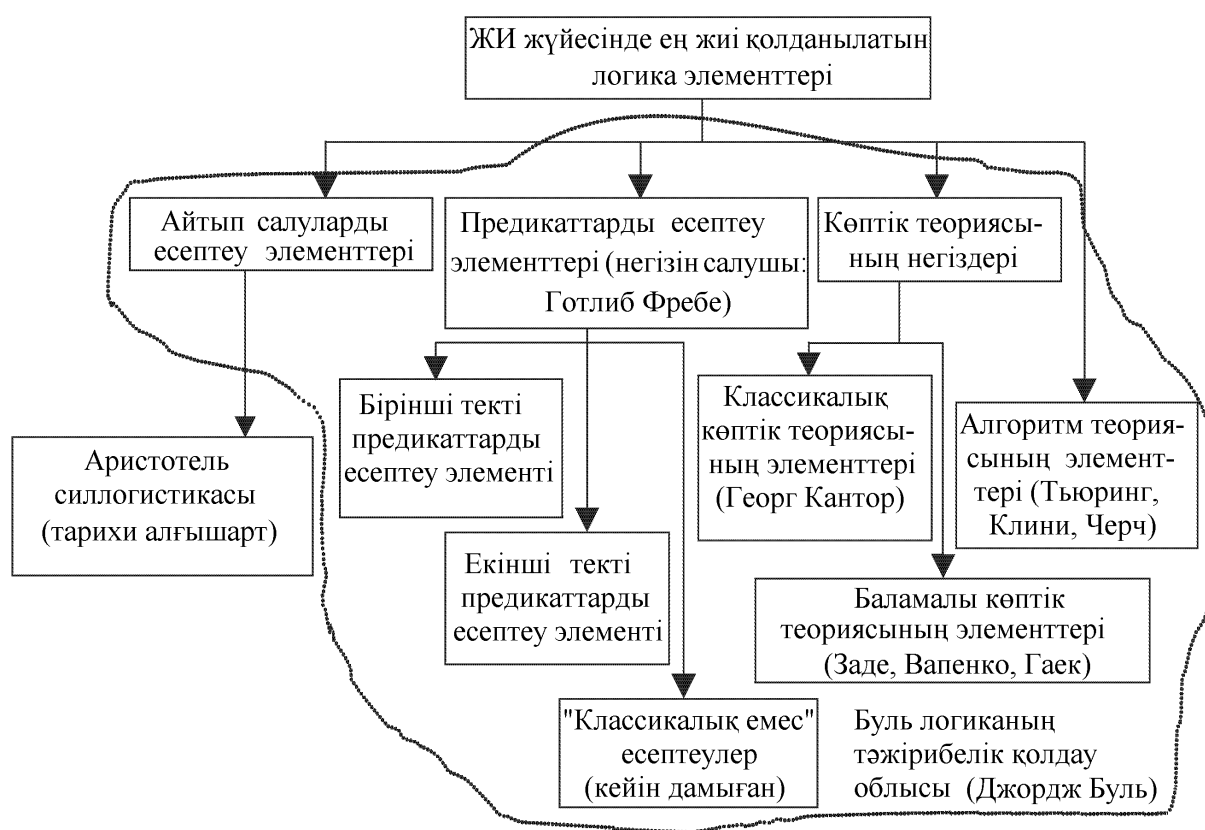
3.3-сурет. Кейбір ең танымалы білімдер ұсыну моделдері

4 - БӨЛІМ

БІЛІМДЕР ҰСЫНУДЫҢ ЛОГИКАЛЫҚ МОДЕЛІ

4.1. Логикалық модельдердің түрлері, жалпы ұғымдар мен анықтамалар

Жасанды интеллектің жаңа теориясында: логика, фреймдер, семантикалық желілер, нейрон желілер және т.б. сияқты бірнеше негізгі модель топтарын пайдалынады. Бүгін логикалық модель өзінің тәжірибелік нәтижесі және нақты техникалық құрылғыларға енгізу дәрежесі бойынша орталық орын алып отыр. Модель өзінен-өзі біртекті емес және әртүрлі оншақты теорияларды өзіне қосады (4.1-сурет).



4.1-сурет. ЖИ жүйелерінде ең жиі қолданылатын логика элементтері

Тарихи жағынан білімдер ұсынудың бірінші моделі классикалық (формалды) Аристотель логикасы, бұл барлық кейінгі логикалық теориялардың негізі деп есептеледі.

Жасанды интеллекте «білімдер ұсынудың модельдері» туралы, ал логикада – «формалды теориялар» туралы айтылады, біздің жағдайымызда мұны бірдей деп есептеуге болады.

Модельдің кейінгі дамуы Кант (категориялар негізінде семантикалық талдау), Буль (логиканы «математикалау»), Фреге және Заде («айқын емес логика» ұғымын енгізді) сынды ғалымдар аттарымен байланысты. Соңғы он

жылдықта «классикалық емес» бағыттар (модальды логика, уақытша логика және т.б.) қарқынды өркендеп жатыр.

Қандай болса да формалды теорияның құрылысы ойшыл процестерді формалды түрде (яғни, формула көмегімен жазу) көрсетуге арналған. Әрине, осындай формуламен құрылған теңдеулер жүйесі қандай болса да әдістермен есептелу қажет. Есептеу нәтижелері нақты әлемдегі бейнелеумен бірдей болу керек. Егер (кейбір нақты әлем үшін) осындай формулаларды және ережелерді тұжырымдап шығара алсақ, онда тәжірибелік түрде жүзеге асырмай біз қандай да іс-әрекеттердің (берілген әлемдегі) салдарын алдын ала болжай аламыз.

Біз тек ақырғы нәтижелерді ғана интерпретациялаймыз (нақты шағын әлемге қарағанда, мағынасын толтырамыз). Олай болса, логикалық шығарудың формалды процесін машинаға тапсыруға болады. Ал адамға - нәтижелерді интерпретациялау және пайдасын бағалау қалады.

Формалды логикалық теорияны жасау үшін әліппені (жазу үшін пайдаланатын символдар көптігі), синтаксис ережелерін (формулаларды жазу ережелері), аксиоматиканы (формулалар көптігінің ерекше тармағы) және шығару ережелерді (формулалар көптігінде қатынастар көптігі) анықтау қажет. Шығару ережелер түрлі бастапқы деректерде логикалық қорытындысының дұрыстығын қамтамасыз етуімен берілу керек. Әліппе мен аксиоматика алынатын қорытындыларының және аралық салдарының ақылға қонатындығын (семантика) қамтамасыз етіп берілу керек. Сонымен, 4.1-суретте көрсетілген логикалық теориялардың негізгі айырмашылығы - аксиомаларды таңдауда және шығару ережелерін анықтауда.

Онан әрі аксиоматика мен логикалық модельдердің шығару ережелерін қарастырамыз.

Әртүрлі микроәлемдерде модельді жүзеге асыру кезінде, түрлі логикалық теориялардың өз артықшылықтары болады. Мысалы, Аристотель силлогистикасының семантикасы табиғи тілге өте жақын, сондықтан формалдау шығарудың нәтижелерін интерпретациялауға оңай. Предикаттарды есептеу негізінде алынатын тұжырымдамалар табиғи тілдің семантикасымен толық сәйкес келмейді, оларды интерпретациялау қиын, бірақ бұл жағдайдағы модельдер жинақтылау және көзге жетерлік болады.

Айқын емес Л.Заде логикасының негізіндегі модельдер шайып кеткен (сирек) ұғымдармен жұмыс істеуге мүмкіндік береді, бірақ осындай нәтижелерді интерпретациялауға да қиын және кейде мүмкін емес.

4.2. Формалды (Аристотель) логикасы

Аристотель атағы пікірлесуді зерттеуден шықты, кейін мұны формалды логика деп атады. Осы жағдайда «пікірлесу» - бұл ұғым. Пікірлесу дегеніміз белгілі, ішкі келісіп алынған бекітулердің (бастамалар) байланысы. Мысалы, екі бастама болсын: «крокодил жасыл» және «крокодил ашулы»;

қорытындысы «жасыл – демек, ашулы». Қорытындыны алу әдісі – бұл пікірлесу болып табылады. Көрініп тұрғандай, дұрыс емес және дұрыс, бірақ мағынасыз пікірлесулер болу мүмкін. Формалды логика – бұл дұрыс ойлау (логикалық қорытынды, пікірлесулер) операциялары мен заңдары туралы ғылым. Логиканың негізгі міндеті – пікірлесу дұрыс әдістерін дұрыс емес әдістерінен ажырату.

Аристотель жорамалдауы бойынша адам ойлауы тіл арқылы өтеді, сондықтан ой сөзбен білдіргенде (айтылса) ғана ой болып шығады. Және сол кезде ол санайтын, сөзбен айтылған ой – бұл тек графикалық жазу, символдар жолы, кейбір грамматикалық дұрыс сөйлем және т.б. Егер грамматика ережелерін бұзбаса, онда сөйлем мағынасын түсінбей-ақ, жауабын (не осы сөйлемге қойылған сұрағын) тек грамматика ережелері негізінде құруға болады. Аристотель логикасының көзқарасынан пікірлесудің дұрыстығы тек осы пікірлесудің логикалық түріне тәуелді және оның мазмұнына тәуелді болмайды, сол себептен оны формалды деп атайды. Ой мазмұны пікірлесу дұрыстығына ешқандай әсер етпейді.

Әлемде әртүрлі тілдер бар. Жазу әдістері (біз үйренген символды түрден графикалық - пиктограммаға дейін) өте ерекше болу мүмкін, кейде бұл тіл ме, көркем бейне ме (4.2-сурет) ажыратуы өте қиын. Өйткені көркем суретті де ақпаратпен алмасуға мүмкіндік беретін бірегей тіл деп айтуға болады.

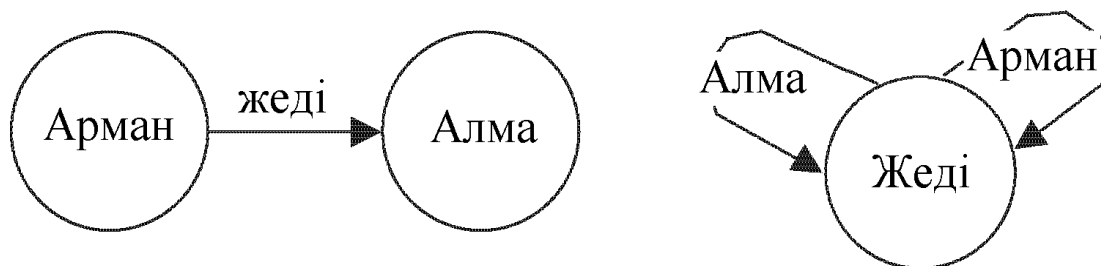
Тіл дегенді қалай анықтауға болады? Классикалық логика жағынан тіл – бұл ішкі мазмұндарды және олардың арасындағы қатынастарды белгілеуге мүмкіндік беретін кейбір жүйе. «Ішкі мазмұн» және «қатынас» ұғымдары логиканың орталық ұғымдары болып табылады. Ішкі мазмұндар деп нақты әлемнің немесе нақты түрде болмайтын абстракты объектілерді айтады, мысалы: «автокөлік», «айдаһар», «ақымақтық». Қатынастар – бұл ішкі мазмұндарды байланыстыру әдісі, мысалы: «жүреді», «ұшады», «бұрын». Қазір бұл терминді былай анықтайды: $J_p = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i \text{ және } P(a_1, a_2, \dots, a_n) = \text{Ақиқат}\}$ көптікте берілген n -орынды предикаттың бағыныңқы көптігі (ақиқат саласы). Логика көзқарасынан тілдегі түрлі сөйлемді: «ішкі мазмұн - қатынас - ішкі мазмұн» түрінде қарастыруға болады. Мысалы, «Арман алма жеді» сөйлемде екі ішкі мазмұн бар («Арман», «алма») және бір қатынас «жеді».



4.2-сурет. Ішкі мазмұндар және олардың арасындағы қатынастар

Дегенмен, қазіргі ЖИ теориясы тұрғыдан кері интерпретация да орынды: бір ішкі мазмұн «жеді» және екі қатынас («Арман» және «алма»). Ішкі мазмұн және қатынас ретінде нені қабылдауға болатыны еркін және шешілетін есептің ерекшелігіне байланысты болады.

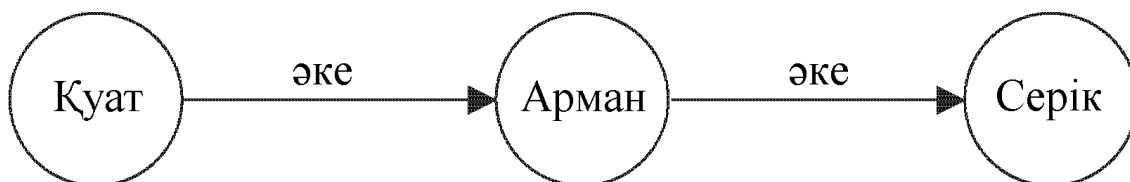
Сөйтіп, Аристотель тіл ішінде ішкі мазмұн мен қатынастарды бөліп шығарды. Бұл ойды графикалық түрде көрсетуге болады (4.3-сурет). Унарлы, бинарлы, тернарлы және n -арлы қатынастарды ажыратады. Мысалы, қатынас «өмір сүреді» унарлы, өмір сүреді (X), көреді (X, Y) – бинарлы, үшбұрыш (1 бұрыш, 2 бұрыш, 3 бұрыш) – тернарлы.



4.3-сурет. Сөйлемнің логикалық сұлбасы

Көрініп тұрғандай, осындай сұлбаға ондаған түрлі сөйлемдер сәйкес болу мүмкін: «Асқар Майраны жақсы көреді», «ит мысықты қуады», «тоғыз жетіден артық» және т.б.

Осы сұлбаның тәжірибелік мағынасы қандай? Ол формалды ережелер арқылы екі берілген сөйлемдерден үшінші мағыналы сөйлемді жасауға мүмкіндік береді. Екі сөйлемді қарастырайық: «Қуат Арманның әкесі» және «Арман Серіктің әкесі» (4.4-сурет).



4.4-сурет. «Әке» деген қатынас транзитивті емес

Осы сұлбадан мынадай қорытынды жасауға болады Қуат Серікке «әке - әке», яғни атасы. Және де 4 алтыдан «кем - кем» деген қорытынды жасауға болады, яғни «көп кем», ал Асқар теннисты «жақсы көреді – жақсы көреді» (өйткені «Асқар Майраны сүйеді» және «Майра теннисты жақсы көреді») деген қорытындыны жасауға болмайды. Сөйлемдер келтірілген сұлбаға кіріп түсе алса да, неге былай шығады? Біздің жағдайда қатынас «сүйеді» транзитивті емес.

Осы сұраққа Аристотель, ал одан кейін атақты философтар Р.Бэкон, Д.Локк және И.Кант көп назар аударған болатын.

Пікірлердің екі түрі болу мүмкін.

- Аналитикалық – тәжірибені өткізу алдында алынатындар (a priori) сенімді пікірлер; субъект пен предикат байланысы қажетті және жалпыға бірдей. Мысал: «дене бұл ұзындығы бар бірдеме». Бірақ, мұнда жаңа білім жоқ, себебі ұзындық деген предикат «дене» ұғымының өзінде болып тұр.
- Синтетикалық - субъект пен предикат байланысы, субъект ұғымында болмайтын (пікірлер a posteriori – тәжірибеден кейін), бірдеңе жаңалықты жорамалдайды. Мысалы, көшенің ұзындығы – 377 м. Пікір бірдеңе жаңалықты хабарлайды, яғни метрлер саның берілген көшенің ұсынысынан аналитикалық түрде шығара алмаймыз; бірақ бұл пікір, тек қажетті және жалпыға бірдей мәні жоқ, жекелеген эмпирикалық фактілерді бейнелейді.

Пікірлер таным мәнін (бастапқы бастамаларда болмаған жаңа сенімді білімдер) алады, олар бір уақытта аналитикалық және синтетикалық болып табылады. Мысал: пікір $789 + 567 = 1356$ бұл қажетті және жалпыға бірдей ақиқат, себебі сандар қосындысы әрқашан дәл сол болып қалады. Пікір нәтижесі – 1356 саны 789 және 567 сандардың ұғымында бар болмайды. Пікір келесі түрмен пайда болады:

- ми барлық берілген сезімдерді уақыт пен кеңістік нысандарда (сезімтал құбылыстар әлемін жасайды) кейбір көрнекі (орныққан пікір) ретке келтіреді;
- сезімтал құбылыстарды түсінудің (парасат категориялары) танымал негізгі әдістері бойынша ақыл-ой түрде байланыстырады.

Пікірдің барлығы төрт нысаны болу мүмкін (S субъектің P предикатпен байланысы), әрбіреуінде үш жағдай бола алады:

1. Предикат субъектің көлемін бейнелейді – мөлшер категориясы.
 - 1.1. Бірлік (жекелеген дана сияқты субъект өз предикат астында болады) – жекелеген пікір.
 - 1.2. Көптік (S тектің бөлігі ретінде P астында болады) – ерекше пікірлер.
 - 1.3. Жалпылық (S тектің толығы ретінде P астында болады) – жалпыға бірдей пікір.
2. Предикат субъект мазмұнына кіретін белгіні білдіреді – сапа категориясы.
 - 2.1. Нақтылық (кейбір белгінің бар болуын мақұлдайды, « A бұл B ») – мақұлдаған пікірлер.
 - 2.2. Мойындамау (кейбір белгінің бар болуы теріске шығарылады, « A бұл B емес») – теріс пікірлер.
 - 2.3. Шек қою (« A B емес» - субъекте осы біреуден басқа қалған белгілері қалады) – шексіз пікірлер.
3. Предикат пен субъект белгілі қалпымен арақатыста болады – қатынас пікірлер.

3.1. Субстанция және тиістілік (предикат субъектке сөзсіз жатады) – шартсыз (үзілді-кесілді) пікірлер.

3.2. Себеп және іс-әрекет (субъект предикаттың шарты ретінде көрсетіледі, «егер A болса, яғни B ») – шартты (гипотетикалық) пікірлер.

3.3. Әрекеттестік немесе қатынаста болу (предикат бірнеше түрге бөлінген болып көрсетіледі, олардың біреуі арқылы онымен субъект байланысады, « A немесе $B1$ немесе $B2$ болады») – бөлетін пікірлер.

4. Предикат пен субъектте байланыс күші бар – модальность категориясы.

4.1. Мүмкіндік (S пен P үйлесімі мүмкін болып көрсетіледі, « A мүмкін B болады») – проблемалы (күмәнді) пікірлер.

4.2. Шындық немесе бар болу (S пен P үйлесімі шынында бар, « A бұл B ») – ассерторикалық (оң, мақұлдаған) пікірлер.

4.3. Қажеттілік (S пен P үйлесімі қажетті болып көрсетіледі, « A B болу керек») – аподиктикалық (міндетті) пікірлер.

Айтылған негізгі ұғымдардан (категориялар): *шама*, *күш* және т.с.с. кейбір басқа жалпы ұғымдар шығарылады. Соның негізінде И.Кант тәжірибелік танымды немесе табиғи ғылымды қамтамасыз ететін түбірлі ақиқаттарды (таза парасаттың негіздемесі) анықтайды. Осыдан қосылу ережелерден басқа жалғастыру іс-әрекеті де қажетті болады.

Жалғастыратын ұсыныстар келесі шарттарға сәйкес болу керек.

1. Өзінің ерекшелігінен бөліп көрсетілген немесе ұстап алынған («аппрегендирандалған»).

2. Олардың жаңа қосылуында қайталанғыштығы болу керек.

3. «Танылу» болу керек. Қайталанған ұсыныстар бұрын ұстап алынғаннан айырмалықты болмау керек. Егер ұсыныстарды ұстап алатын, қайталанғыштайтын және таныйтын субъект бірдей қалып отырса немесе өзіне тең болса, яғни субъект өз ішкі теңдік $МЕН = МЕН$ сақтаса, сонда ғана танылу актісі орын алады.

Формалды өзгертулерді жүзеге асыру кезінде алынатын нәтижелердің саналылығын жоғалтпау үшін қатынастар мен шығару ережелер типтері туралы ғылымды семантикалық категориялар теориясы (немесе логикалық грамматика) деп атайды.

Егер мағыналы сөйлемде олардың біреуін басқамен ауыстырғанда, сөйлем мағынасызға ауыспайтын болса екі өрнек тілдің бірдей семантикалық категориясына жатады. Әрбір дұрыс құрылған тіл өрнегі бір ғана семантикалық категорияға кіреді.

Барлық категориялар негізгі және функторлыққа бөлінеді. Негізгі – бұл есімдер және сөйлемдер (айтылулар). Оңайлату үшін категориялар санын үшеу: есімдер, сөйлемдер (айтылулар) және функторлар (жаңа есімдерді немесе айтылуларды жасау үшін арналған өрнектер) деп санайық.

4.2.1. Есімдер

Есім – бұл жеке бұйымды, ұқсас бұйымдар жиынтығын, қасиеттерді және т.б. белгілейтін тілдің өрнегі. Есімдерді неше бұйымды көрсететіне байланысты жекеленген және жалпыға бөледі. Бос (айдаһар, елес) және бос емес есімдерді ажыратады (аз болғанда бір нақты объектке жібереді).

Есім мазмұны – тек берілген есімімен белгіленген барлық ұйымдарға тиісті осы қасиеттердің жиынтығы. Мысалы, «жайлы пәтер» есімнің мазмұнында біз су құбыры, канализация және телефоны бар адамдар тұру үшін арналған бөлмені түсінеміз. ЖИ теориясында экстенсионал және интенсионал ұғымдары құрылған. Экстенсионал – бұл берілген ұғымға сәйкес нақты фактілер жиынтығы. Интенсионал – бұл ұғымның қасиеттері арқылы кейбір ұғымның анықтауы немесе сипаттауы.

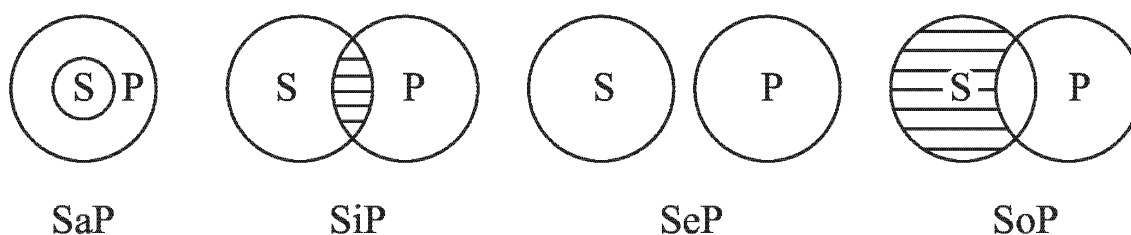
Есім көлемі – есім мазмұнына кіретін белгілері бар бұйымдар жиынтығы немесе класы. «Жайлы пәтер» есім көлеміне кіреді, «қазіргі қалалық үйдегі пәтер», «қала сыртындағы коттедж пәтері» және т.с.с.

Бір мәнді және көп мәнді, дәл және дәл емес, айқын және айқын емес есімдерді ажыратады. Логикада «бір мағыналылық принцип» бар, оған сәйкес коммуникация кезінде бірнеше түрлі мәні бар тіл айтылуларды пайдалануға болмайды. Дәл емес есімдер үшін қандай нәрселерді олар кірістіреді, ал қандайды жоқ (көлем) кейде айту қиын. Айқын емес өзінің есімдер өзінің мазмұнына немесе мағынасына тиісті «шайып кеткен» немесе толық анықталмаған болады. Айқын емес есімдерді қосатын пікірлесулерде кейбір маңызды ерекшелері бар.

(i) Дәлсіздіктің контекстуалды сипаты бар. Бұйымның (процесс, күй) өзін ғана алып, ол туралы айтысуға мағынасыз, өйткені бір жағдайда есім бердемені белгілейді, ал басқа тұрғыда – басқаны белгілейді.

(ii) Дәл емес есімдерді қолдану парадоксалды қорытындыға әкелу мүмкін. Кейде есімдердің дәл еместігін жоюға болады (жас - ер жеткен - 18 жаста).

Есімдер бір-бірімен әртүрлі қатынаста болады. Бір-бірімен салыстыруға келетін түрлі есімдердің көлемдер арасындағы қатынастардан тек біреуі ғана орын алады: маңызы бірдейлік, қиылысу және шығарып тастау (4.5-сурет).



4.5-сурет. Есімдер арасындағы қатынастарды бейнелеу үшін Эйлер диаграммалары

Көлемі толық сәйкес келетін екі есім маңызы бірдей болады. Маңызы бірдейлік бұл мазмұны сәйкес келетін емес.

Мысалы, «жайлы пәтер» және «жайлы бөлме» есімдерді екі айтылған баламалар көптігінде маңызы бірдей деп санауға болады, бірақ олардың мазмұны сәйкес келмейді (себебі офис бөлмесінде тұруға болмайды).

Бөлу – бұл бастапқы есімнің көлеміне кіретін бұйымдарды топқа үлестіру операциясы. Үлестіруден алынған топтар – бөлу мүшелері. Бөлу жүргізетін белгі – бұл бөлу негізі. Бөлуге қойылатын талаптар:

- 1) бөлу тек бір негізі бойынша жүргізіледі;
- 2) бөлу шамаласты (жеткілікті) болу қажет, яғни бөлу мүшелердің қосындысы бөлетін ұғымның көлеміне тең болу керек;
- 3) бөлу мүшелері бір-бірін өзара шығарып тастау қажет;
- 4) бөлу үздіксіз болу керек.

Бөлудің дербес жағдайы дихотомия. Дихотомиялық бөлу белгінің түрлендіру шеткі жағдайына негізделеді. Бір жақтан осы белгісі бар бұйымдар бөліп көрсетіледі, екінші жағынан – белгілері жоқтар.

Топтастыру – көп сатылы тармақталған бөлу. Топтастыру нәтижесі матасқан есімдер жүйесі: бөлінетін есім тек болып табылады, жаңа есімдер – түрлер, түрдің түрлері (түр тармағы) және т.с.с.

Класс – бұйымдар көптігі, олардың әрбіреулеріне, сәйкес ұғымның мазмұнындағы көрсетілген, белгілер тиісті. Осы ұғымды білдіретін сөз немесе сөз тіркесі класс есімі болып қызмет етеді. Кластар қарапайым және күрделі болу мүмкін.

Қарапайым класс – бұл мүшелері осы кластың есімімен берілетін тек бір ортақ белгімен сипатталатын класс. Мысал: ұшақтар, жолаушылар, реактивті қозғалтқыштар және т.с.с. Қарапайым класты одан қарапайымдарға бөлуге болмайды.

Күрделі класс - бұл мүшелері бірден артық ортақ белгісі бар класс. Күрделі класты әрқашан қарапайымдарға бөлуге, ал керек кезде қарапайымдардан қалпына келтіруге болады.

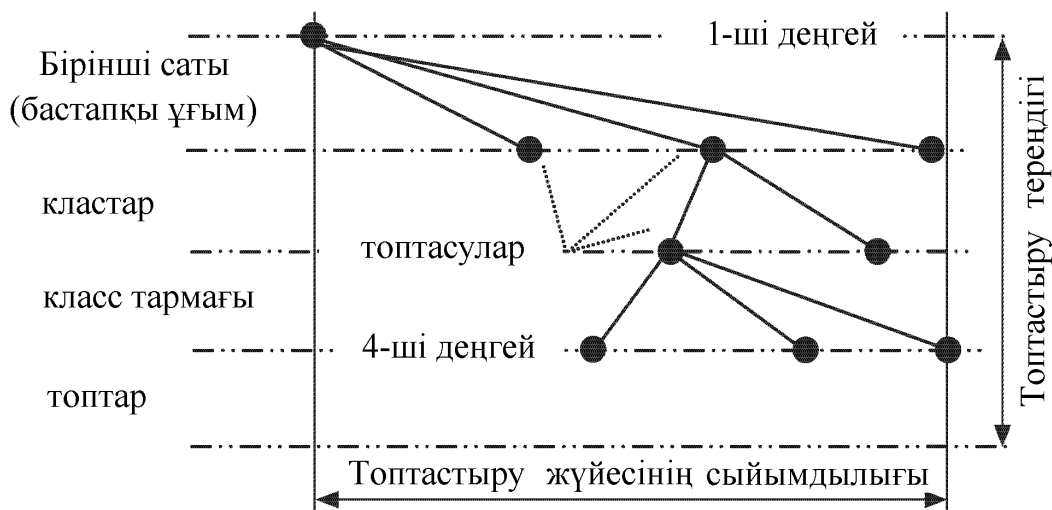
Топтастыру жүйелер (ТЖ) білімдерді құрылымдау және қорыту үшін қолданылады. Бір жағынан, барлық ішкі мазмұндар кейбір белгілер арқылы бірнеше класқа бөлінеді; басқа жағынан, бірге топтастырылады. Байқаушыға кейбір белгілер көптігімен сипатталатын объектілер жиынтығы беріледі. әрбір объект бір немесе одан артық кластарға жатады.

ТЖ көмегімен үш негізгі есеп шешіледі.

- Бейнелерді топтастыру: байқаушы бұрын қойылған ережені қолдану қажет және жаңадан түскен объекті кейбір класқа жатқызу.
- Бейнелерді айырып тану: түрлі кластарға (үйрететін іріктеу) жататын көп объектілерді зерттеу негізінде топтастыру ережесі жасалынады.
- Бейнелерді құрылу: қандай класқа жататынын көрсетпей объектілер байқаушыға беріледі. Байқаушы сәйкес бөлуді өзі құрастырып шығарады.

Топтастырудың жалпы есебі айқындау есебіне баламалы: кейбір формалды тілде (грамматикада) кейбір тізбек мүмкін сөйлем болып ала ма. Кейбір ТЖ декларативті білімдерді ұсыну үшін қолданылады, мысалы, алгоритмдық тілдердің кілтті транслятор кестелерін құру үшін.

Иерархиялық жүйеде (4.6-сурет) топтастырылатын топтар арасында бағыну қатынас қойылған (мысалы, тек - түр). Объектілер көптігі кейбір белгі (бөлу негізі) бойынша ірі топтастыққа бөлінеді, сосын әрбір топтастық келесі топтастыққа және т.б., біртіндеп топтастыру объектіне жеткенше бөлінеді. Жоғары айтылған бөлу ережелерін бұзбау қажет.

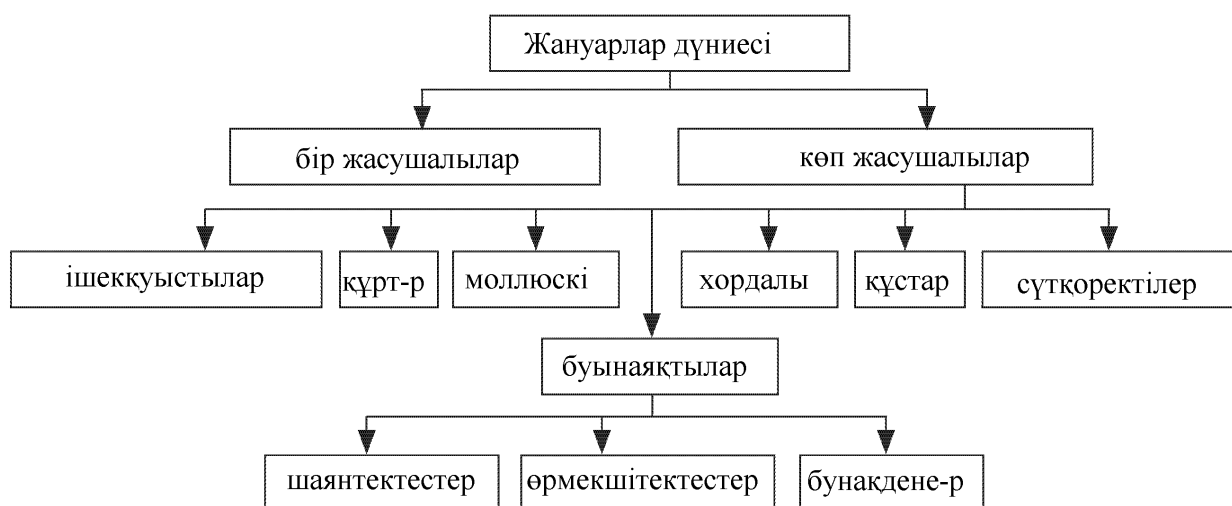


4.6-сурет. Иерархиялық топтастыру

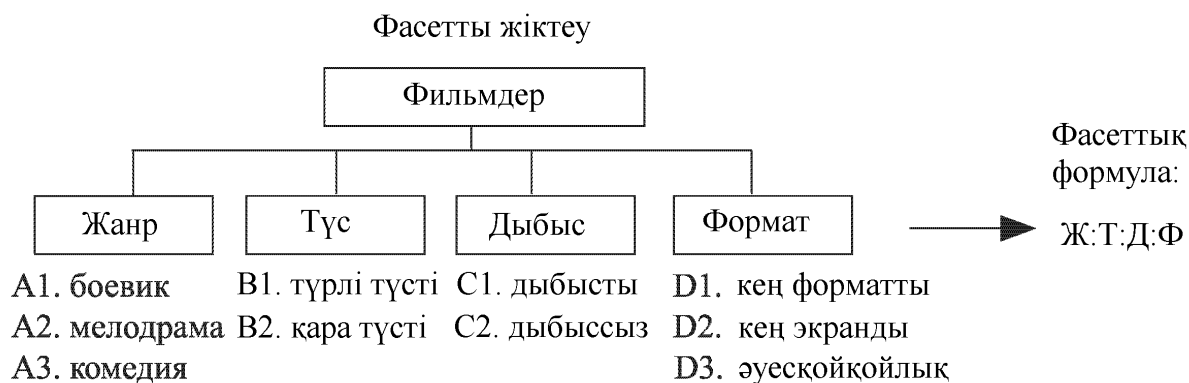
Осылай, мысалы, иттерді тектер бойынша топтастыру кезінде ит ұғымын колли, дог және овчаркаларға бөлу дұрыс емес. Егер біз барлық колли, дог және овчаркаларды бірге жинасақ, «иттер» деген класс алалмаймыз.

Қанша тектер тізімін толтыра бермесек, бәрі бір кейбіреулерін жоғалтамыз. Дұрыс вариант: иттер – бұл «догтар және «догтар емес»», «догтар емес» - бұл «овчаркалар және «овчаркалар емес»» және т.б.

Иерархиялық топтастыру көптеген салаларда қолданылады. Компьютер болмаған кезде, ол есепке алу мен жүйелеудің негізі болып табылатын (мысалы, биологияда, 4.7-сурет). Жануарлардың әртүрлілігін жүйелеу зерттейді. Оның басты міндеті – жануарларды топтар (жіктеу) бойынша үлестіру. Топтастырудың негізгі бірлігі – жануарлар түрі.



Автоматтандырылған жүйелерде қазір тезаурустар көптен қолданылады (4.1- кесте) және фасетты жіктеулер (4.8-сурет).



Иерархиялық топтастыру жүйенің негізгі артықшылығы – үлкен ақпараттық көлем және іздестірудің оңайлығы (қолмен іздеу мүмкіндігі). Кемшіліктері – құрылым иілгішінің аз болуы, түрлі белгілердің еркін үйлесуі бойынша объектілерді агрегаттауға мүмкін емес.

Алфавит-заттық топтастыру – бұл алфавит ретінде орналасқан (телефон анықтамалығы) кластар жүйесі (олардың әрбіреуі заттың немесе фактілердің бір түріне сәйкес). Осындай жіктеудің мысалы ағылшын-қазақ сөздігі. Сөздер алфавит ретіне сәйкес қатаң орналасқан.

Кейде басқа принцип ыңғайлы – әртүрлі әріпке басталатын, бірақ бір жағдайға немесе затқа қатысатын барлық сөздерді қатар қою. Мысалы, «полиция», «аурухана», «сақтандыру» сөздерді «авария» бөліміне орналастыру орынды. Бұл жағдайда заттық жіктеумен жұмыс істейміз.

Тезаурус – құжаттың немесе ақпараттық сұраныстың орталық нүктесі (тақырып). Әдетте, қарапайым кластар есімі болып табылатын, табиғи тілдің кейбір қарапайым сөздердің және сөз тіркестерінің тізбегі түрінде көрсетіледі. Осындай сөздер – *n*-өлшемді тақырып-заттық кеңістігіндегі құжат координаталары. Құжаттық немесе ақпараттық сұраныстың

координаталы индексациялау үшін толық мәнді (қызметтік емес) сөздер пайдалану мүмкін, олар тікелей индексацияланатын мәтіндерден алынады. Осындай сөздер мен тіркелер – кілтті болады. Тезаурусты құру процесі 4.1-кестеде келтірілген.

4.1-кесте

Тезаурусты құру реті

Топтастыру бірлігін атауы	Операциялар
Бастапқы мәтін немесе бастапқы мәтіннің фрагменті	Кілттік сөздерді іздеу (мәтінде белгілеу)
Нормативті сөздік	Кілттік сөздерді бірыңғай морфологиялық нысанға және жазуға келтіру. Синонимия мен полисемияны еске алу, мысалы, көпір (құрылыс), көпір (жүріс бөлігі)
Дескрипторлы сөздік	Жақын кілттік сөздерден класс құрылады, қорытылған есім тандап алынады (дескриптор)
Нормативті анықтамалық-сөздік (ақпараттық іздестіру тезаурус)	Тақырып-заттық саласы туралы мәліметтермен дескрипторлы сөздікті толтыру

Ақпараттық іздестіру тезаурус құжаттарды және ақпараттық сұраныстарды координаталы индексациялау үшін арналған, оларда әліпби ретімен бар дескрипторлар және оған синонимды кілттік сөздер келтірілген, және де дескриптор арасындағы ең маңызды қатынастар көрсетілген. Функцияға байланысты лексикалық жағынан тезаурус бірліктері дескриптор мен шартты синонимға бөлінеді.

Дескриптор – шартты баламалылық класты белгілейтін бірімәнді кілттік сөз немесе код, оған баламалы сөздер және оларға мағынасы бойынша жақын сөздер қосылған. Тезаурустың элементарлы құрылымдық бірлігі – сөздік мақала.

Мысал. Есептеуіш машиналар:

⟨компьютерлер, ЭЕМ, есептеуіш кешендер⟩

⟨дербес, үлкен ЭЕМ, кіші ЭЕМ, супер-ЭЕМ⟩

⟨есептеу техникасы, есептеуіш техника, инженерлік есептеу теориясы⟩

⟨есептер, Интернет, ойындар, экономика⟩.

Осында тезаурус $d_i\langle M_1, M_2, M_3, M_4 \rangle$,

мұндағы d_i – бас дескриптор;

M_1 – әліпби бойынша реттелген берілген бас дескриптордың шартты синонимдар көптігі, онымен бірге шартты баламалылық класты жасайды;

M_2 – реттелген дескрипторлар көптігі, олардың әрбіреуі тегі-түрі бас қатынаспен байланысқан;

M_3 – реттелген дескрипторлар көптігі, олардың әрбіреуі түрі-тегі бас қатынаспен байланысқан;

M_4 – реттелген дескрипторлар көптігі, олардың әрбіреуі бүтін-бөлік, бөлік-бүтін, себеп-салдар және т.б. бас қатынаспен байланысқан.

Топтастыру қиыншылықтардың себебі объективті болады. Қоршаған әлем күрделі, оның ішінде қатты шекаралар және айқын шектелген кластар жоқ. Жаппай құбылмалық суретті күрделендіреді.

Қатты суреттелген шекараны бөліп көрсету кезіндегі проблемаларды шешетін ғылым таксономия деп аталады. Бұйымды бір мәнді топтастыруға қиын болған жағдайда, оны сандар арқылы бағаланған белгілер жиынтығы бойынша кейбір класқа жатқызады. Бұл жағдайда сандық таксономия туралы айтады.

4.2.2. Айтып салулар

Айтып салу (бекіту) – білдіретін мағынасымен бірге алынған және шын немесе жалған болатын грамматикалық дұрыс сөйлем. Қарапайым айтып салу (бөлінбейтін, атомрлы, атом) өз бөлігі ретінде басқа айтып салуларды қоспайды. Күрделі айтып салу қарапайымдардан жасалынады логикалық байлам көмегімен: мойындамау, конъюнкция (және), дизъюнкция (немесе), шығарып тастайтын дизъюнкция. Қарапайым айтып салудың мысалы: «Аюлардың бәрі балды жақсы көреді», күрделі – «Кейбір аюлар балды және (кейбір аюлар) бамбуктың жас өскіндерін жақсы көреді».

Қарапайым айтып салулар құрамдас бөліктерге әртүрлі бөліну мүмкін. Аристотель қарапайым айтып салулардың өзге түрін қарастырған – *үзілді-кесілді* айтып салулар (пікірлер), яғни оларда бекітіледі немесе мойындалмайды қандай болсада белгінің бар болуы.

Мысал: « S бұл P » және « S P емес», мұнда S – бұйым (субъект) есімі, P – белгі (предикат) есімі. «Жер – бұл планета».

Екі немесе одан артық бұйымдар арасындағы қатынастар туралы да айтып салулар қарастырылады. *Мысал:* «Үш бестен кем», «Астана Ақтаудан үлкен». Қатынастар туралы айтып салуларды үзілді-кесілді айтып салуларға келтіруге болмайды.

Үзілді-кесілді айтып салулардың ерекшелігі бұйым мен белгі арасында байланыс жай ғана қойылмайды, яғни «барлықтар», «кейбіреулер» (4.2-кесте). Логиканың даму барысында «предикаттарды есептеу» ұғымға келді – бұл математикалық логиканың арнайы облысы. Бірінші және екінші ретті предикаттарды қарастырып бастады. Теория негізінде арнайы машиналық тілдер жасалынған, мысалы Planner және Prolog.

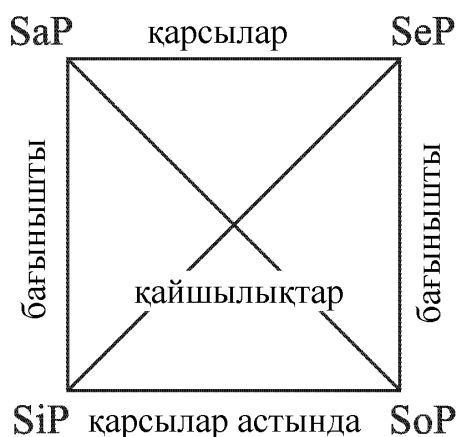
4.2-кесте

Үзілді-кесілді айтып салулардың нысандары

Функторлар	Белгілеу	Интерпретация мысалы
«барлығы ... бұл ...»	a	SaP – барлық сұйықтар серпінді
«кейбіреулері ... бұл ...»	i	SiP – кейбір жануарлар сөйлейді
«барлығы ... бұл емес ...»	e	SeP – барлық дельфиндер балық емес
«кейбіреулері ... бұл емес ...»	o	SoP – кейбір металдар сұйықтар емес

Осы төрт өрнектердің (a , i , e , o) әрбіреуін Аристотель өзіндік мазмұны жоқ, шын немесе жалған қарапайым айтып салулар болатын логикалық тұрақтылар ретінде қарастырды. Сонымен бірге, S пен P есімдер жекеленген немесе бос болмау қажет.

Күрделі айтып салуларды құрастырған кезде, логикалық шығаруды (екі бастамадан салдарын алу) жүзеге асыруда қорытындының шындығын ғана емес, оның саналылығын да ойлау қажет. Саналылығын «логикалық шаршы» көмегімен формалды түрде талдауға болады (4.9-сурет). Логикалық шаршыда ерекше пішіндер (логикалық шығарудың) айыруға болады, оларға бағынып саналы нәтиже аламыз.



Қайшылық айтып салулар – бір уақытта шын және жалған бола алмайды.

Қарсы айтып салулар – бірге жалған бола алады, бірақ бірге шын бола алмайды.

Қарсылар астындағы айтып салулар - бір уақытта жалған бола алмайды, бірақ бір уақытта шын бола алады.

Бағынышты айтып салулар – бағыныңқы айтып салудың шындығынан автоматты түрде бағыныштының шындығы шығады

4.9-сурет. Логикалық шаршы

4.9-сурет шығару ережелерін (формулалар арасындағы қатынастар) береді деп айтуға болады. Сонымен әліпбиді шығару ережелерін, синтаксисті анықтап, біз кейбір логикалық теорияны тапсырдық, берілген жағдайда – айтып салуларды есептеу. Кейін пайда болған предикаттар есептеуінен оның айырмашылығы мынадай – терминдер (S , P) ретінде дәл логикалық айнымалылар болып шығады, ал предикаттарды есептеуде термин орнында бүтін предикат болу мүмкін. Үзілді-кесілді айтып салуларда логикалық шығару индуктивті немесе дедуктивті өту мүмкін.

Дедуктивті ой тұжырымында бастамалар мен тұжырым байланысы логикалық заңға негізделеді. Логикалық қажеттілікпен тұжырым қабылданған бастамалардан шығады, шын бастамалардан әрқашан шын тұжырымға әкеледі. Дедукция үшін жалпы білімнен дербес білімге өту сипатты.

Индуктивті ой тұжырымында бастамалар мен тұжырым байланысы логикалық заңға емес, кейбір формалды сипаты жоқ фактілік немесе психологиялық негіздерге сүйенеді. Индукция ықтималдық (шындыққа ұқсас) тұжырымдар береді. Индукция үшін дербес жағдайлар көптігінен жалпы білімге өту сипаты - қорыту. Дедукция мен индукция арасындағы негізгі айырмашылығы мұндай: дедукция - бұл бір ақиқаттан басқаға логикалық ауысу, ал индукция – сенімді білімнен ықтимал білімге ауысу.

Шын бастамалардан толық шығару әрқашан шын қорытындыға келтіреді. Тәжірибе мен сезуге қарамай, таза ойлау көмегімен жаңа білімді алуға мүмкіндік береді. Егер бастамалардың біреуі ғана жалған болса, онда дұрыс ойлау қорытындысында ақиқатты да, жалғанды да бере алады. Шын

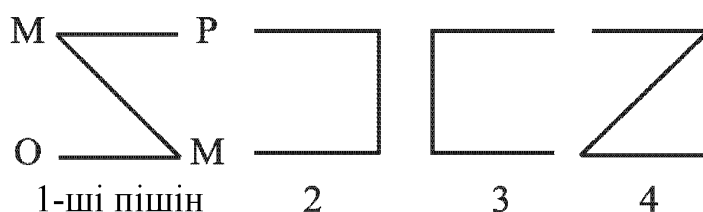
бастамалардан дұрыс емес тұжырым шын да, жалған да қорытындыларға әкелуі мүмкін.

Аристотель үзілді-кесілді силлогизмдарды (немесе жай силлогизм, ал теориясы силлогистика деп аталады) кезінде қарастырған. Бұл дедуктивті ой пікірлер, оларда екі үзілді-кесілді айтылулардан жаңа үзілді-кесілді айтылу шығарылады.

Барлық сұйықтар серпінді
Су сұйық
Су серпінді

Барлық M бұл P
Барлық S бұл M
Барлық S бұл P

Әрбір силлогизмда үш термин болу қажет: кіші (субъект S), үлкен (предикат P) және орта (бастамаларда болатын және қорытындыда болмайтын термин M). Орта терминның орналасуына байланысты төрт силлогизм пішіндерін ажыратады (4.10-сурет).



4.10-сурет. Силлогизм пішіндері

Силлогизм модустары деп пішіндердің өзге түрін айтады, олардың айырмашылығы бастамалар мен қорытынды сипаттарында. Барлығы $4 \times 64 = 256$ әртүрлі бастамалар мен қорытындылар сәйкестігі болу мүмкін (4 пішін, әрбір пішінде 64 модусы бар). Барлық мүмкін болатын силлогизм модустарынан дұрысы тек 24 (әрбір пішінде алтаудан), олардын бесеуі әлсіз – олардағы қорытынды дербес бекітілген немесе дербес қайшы айтылулар. Сонымен, 19 дұрыс силлогизм модусы қалады. Силлогизмның кейбіреулерінде дәстүрлі пайдаланатын есімдер болады (4.3-кесте).

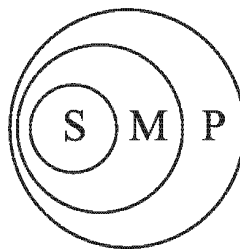
4.3-кесте

Кейбір силлогизмдардың дәстүрлі есімдері

1 пішін	Barbara, celarent, darii, ferlo, barbari, celaront
2 пішін	Cesare, camestres, festino, baroco, cesaro, camestros

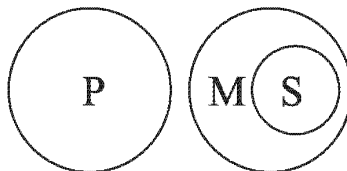
Осы атаулардың әрбіреуінде 3 дауысты әріп бар. Олар модуста бастама мен қорытынды ретінде қандай үзілді-кесілді айтылулар пайдаланатынын көрсетеді. Мысалы, Celarent — үлкен бастама (SeP), кішісі — жалпы мақұлдаған (SaP), қорытынды — жалпы қайшы айтылу (SeP). Силлогизмның дұрыстығын бағалау үшін Эйлер диаграмманың пайдалануы ыңғайлы.

1-мысал. (Barbara): Барлық металдар (M) созылымды (P) + темір (S) бұл металл (M) = темір (S) созылымды (P). Яғни егер барлық M P көлеміне кірсе, онда S P көлеміне кіреді (4.11-сурет).



4.11-сурет

2-мысал. Барлық балықтарда (P) қауырсын (M) болмайды + барлық құстарда (S) қауырсын (M) бар = құстардың (S) ешқайсысы балық (P) емес. Яғни егер барлық құстар M көлемге кірсе, ал M -ның P мен ешқандай ортағы жоқ болса, онда S -та да P мен ешқандай ортағы болмайды (4.12-сурет).



4.12-сурет

Бастамада болмаған ақпарат силлогизмда да болмайды. Қорытынды, бастамадағы ақпаратты тек жазады, бірақ оның ішіндегі болмаған жаңа ақпаратты енгізе алмайды.

Бастамалардың біреуі айқын берілмеген силлогизмды *энтимема* деп атайды. Дұрыстығын бағалау үшін энтимеманы толық силлогизмға қалыптастыру керек.

4.2.3. Дәлелдеу және мәлімдеу процедуралары

Логикалық моделі негізіндегі ЖИ тәжірибелік программалық жүйелері көбінесе, екі есепті шешуге арналған.

1. Логикалық жорамалды дәлелдеу (берілген нақты салада олар теорема деп аталады). Жорамал кейбір логикалық формула арқылы көрсетіледі және оның ақиқаттығын кейбір бастапқы деректер жиынтығында дәлелдеу керек. Ақпараттық жүйенің кіреберісіне формула беріледі, жүйе оны тексереді және егер формула шын болса, жауап береді «Иә». Егер формула жалған болса, онда жүйенің жауабы «Жоқ», бірақ кейбір жағдайда жүйе не «Иә», не «Жоқ» деп айталмайды. Олай болса, осы формуланың мойындамауын (яғни оны мәлімдеу) дәлелдеуге тырысып, сұранысты басқаша тұжырымдайды. Кейбір жағдайда мәлімдеу жалғанды дәлелдеуден әлдеқайда оңай болып шығады.

Мысалы: «барлық аюлар қоңыр» - бұл үшін бір ғана ақ аюды көрсету жеткілікті, яғни «аюлардың бәрі ақ емес» деген теореманы растау.

2. Теорема дәлелдеулерін түсіндіру – пайдаланушыға барлық аралық қорытындыларды беру. Бұл жағдайда жүйе түсіндіреді: «Мен осындай қорытынды жасадым, өйткені ...». Егер 10-15 «себеп-салдар» жұптар туралы әңгіме болса, онда машинаның жұмыс логикасын қадағалауға болады, жұптар 20-30-дан артық болса, машинаның түсініктемесі естен айырылған адамның сөздеріне ұқсай бастайды.

Сөзсіз, логикалық модель негізінде басқа есептерді де шешуге болады, мысалы, кейбір бастамадан барлық мүмкін болатын (дұрысын да, дұрыс еместерін де және саналы еместерін) салдарды шығару есебі.

Айтып салулар логикасын ең анық деп айталмаймыз. ЖИ түрлі бағдарламалық жүйелері әртүрлі дәлелдеу және мәлімдеу процедураларды пайдаланады, ең жалпы теориясын қарастырайық.

Дәлелдеу – кейбір бекітудің ақиқаттығын дедуктивті анықтау процедурасы, ол үшін ақиқаттағы белгілі бекітулер шақырылады және олардан қажетті түрде біріншісі шығады. Дәлелдеу тезистен, негізден (аргумент) және логикалық байланыстан тұрады. Құрылым бойынша дәлелдеуді тура және жанама түрге бөледі (антитезистың қателігін ашып әділдігін анықтайды). Жанама дәлелдеу түрлерін қарастырайық.

1. Мағынасыздыққа келтіру (бекітуден қайшылықты шығару). Мысалы, ежелгі грек философы Протагор: «Кімге қандай ой келсе, сол шындық», деген бекіту айтты. Оған Демокрит: «айтылулардың барлығы шын емес», - деп жауап берді.

2. Бөліп көрсететін жанама дәлелдеу (егер қарастырылатын мүмкіндіктер саны 2-ден артық болса). Егер дәлелденетін қағида барлық қарастырылатын мүмкіндіктер санына кіретін болса, сонда ғана қолданылады.

Мәлімдеу – қойылған тезиске қарсы бағытталған пікірлесу, оның мақсаты тезистың жалғандығын немесе дәлелденбегенін анықтау. Мәлімдеу тәсілдерін қарастырайық.

1. Мәлімденетін бекітуден ақиқатқа қайшы салдарды шығару.

2. Бекітуді мойындамауының шындығын дәлелдеу (бекіту және оның мойындамауы бір уақытта шын бола алмайды). Мысал: аюлар тек қоңыр ғана болады. Тек бір ақ аюды көрсетіп – мәлімдеуді аламыз.

3. Егер тезис негіздемемен бірге қойылса, онда осы негіздемелерді мәлімдеуге болады. Сонымен бірге бірақ, қағида өзі дұрыс болса да, аргументтері әлсіз болу мүмкін.

4. Аргумент пен тезис арасындағы байланыстың өзін мәлімдеуге болады, егер тезистің дәлелден шықпайтығын көрсете алсақ. Бірақ, осыдан не аргументтің, не тезистің қателігі шықпайды.

Дәлелдеудегі қателіктер.

1. Аргументтерге қатысты қателіктер:

1.1. Мазмұнды қателік – жалған аргументтер көмегімен тезисті негіздеу;

1.2. Дәлелдеудегі шеңбер – қағиданы өз-өзінен дәлелдеу, бірақ қағида басқа түрде тұжырымдалған.

2. Тезисті ауыстырып алу – дәлелдеу барысында тезисты пішіне немесе мазмұнына жақын мен ауыстыру.

3. Жоғалған логикалық байланыс.

Логикалық қателіктерді іздеп табу өте күрделі еңбек. Ежелгі заманда софизм мен паралогизмдарды құрастыру және шешу ережелеріне елшілерді арнайы үйрететін.

Софизм — дұрыс болып көрінетін, бірақ әдейі жасырған логикалық қателігі бар, пікірлесу. Жалған қорытындыға шындық көрерлігін жасау үшін қызмет етеді.

Мысалдар: «бірдеңені жоғалтпасаң, сол сенде бар; сен мүйізді жоғалтпадың, сондықтан сенде мүйізін бар», «отырған тұрды, кім тұрды, сол тұр; олай болса, отырған тұр», «бұл ит сенікі, ол әке, ендеше сенің әкең ит».

Паралогизм — пікірлесудегі әдейі жасалмаған қателік.

4.3. Формалды логиканың математикалық жүзеге асыруы

XIX ғасырда ағылшын математигі Джордж Буль логикалық белгілеу жүйесін жақсартты.

Ол ақиқатты 1 деп, ал жалғанды 0 деп жазуға ұсынды. Басқа сөзбен, айтылуды айтылу деп санауға болады егер оның айнымалылары нақтылау алынса, сонымен бірге не «Ақиқат», не «Жалған» мәнімен. «Белгісіз» деген нақтылау айтылмайды. Буль тек қатынастарды «+» бірлесуді, «*» қиылысуды шығарып тастауды қарастырды.

Жақсартылған Буль жазу жүйесі абстракты логикалық есепті теңдеулер жүйесі түрінде қоюға мүмкіндік берді:

$$X_1 + X_2 = 1$$

$$X_2 * X_3 = 0.$$

Кім шоколадты жеп қойды? Мысық (X_1) немесе («+») күшік (X_2)? Егер иесі (X_3) жемесе, ал сол уақытта күшікпен ойнаса, онда ол мысық.

Буль арнайы белгілер енгізді (4.4, 4.5- кестелер). Қазір осы бағыт «буль алгебрасы» деп аталады, бірақ оның бір қатар маңызды қағидалары Дж.Бульге дейін ашылған болатын.

Предикаттарды есептеу – кейбір ортаны модельдеуге және қандай болса да жорамалдарды тексеруге арналған кейбір аксиоматикалық жүйе. Жорамалдар, сонымен бірге, кейбір объектілерде кейбір қасиеттерінің бар болуын немесе болмауын бекітеді және логикалық формула түрінде беріледі. Жорамалдың негіздеуі, сонымен, логикалық формуланың шығаруына және орындалуына баға беруге апарады.

Жорамалды жиі «теорема» деп атайды, осыдан термин «теоремаларды автоматты түрде дәлелдеу» шығады. Бұл кәдімгі рекурсивті логикалық шығару, көбінесе, қарар көмегімен. Осындай шығарудың нәтижесі «Иә» (егер дұрыс теңдік алынса) немесе «Жоқ» (егер бос формула алынса) болып шығады.

4.4-кесте

Буль логиканың кейбір белгілеулері

Операция	Белгілеу
Конъюнкция	«және», « $\wedge$ », « $\&$ »
Дизъюнкция	«немесе», « $\vee$ », « $+$ »
Баламалылық	« A сонда және тек сонда ғана, B болғанда», « $\leftrightarrow$ », « $\equiv$ »
Импликация	«егер A , онда B . A тартады B », « $\rightarrow$ » (кейде « $\supset$ »)
Мойындамау	«емес», « $\neg$ »

4.5-кесте

Негізгі логикалық операциялардың шындық кестесі

X	Y	$\neg X$	$X \wedge Y$	$X \vee Y$	$X \rightarrow Y$	$X \leftrightarrow Y$
Ж	Ж	А	Ж	Ж	А	А
Ж	А	А	Ж	А	А	Ж
А	Ж	Ж	Ж	А	Ж	Ж
А	А	Ж	А	А	А	А

Қайшы есептеу – мойындамаулары мен бірге осы есептеуде шығарылатын формулалары бар есептеу.

Егер кез келген, бірақ осы формулалардың (мәндер жиынтығы) бірдей моделі α формуланың моделі болып шықса, онда α *формуласы* $\alpha_1, \alpha_2, \dots, \alpha_n$ - нан *шығарылатын* болып табылады. Шығарылу символы: « $|$ –», деп жазылады.

Формула моделі – заттық ортаның мәндерімен нақтылынатын формула. Әрине, бірдей абстракты логикалық формула кейбір ортаның моделін құру үшін негіз болып шығуы мүмкін. Сондықтан абсолютты және салыстырмалы орындалу туралы, бірдейлік (барлық модельдер үшін) және салыстырмалы (нақты модель үшін ғана орындалады) шындық туралы айтады.

Формула семантикасы – формалды логиканың формуларын мағыналы интерпретациялау, яғни формуладағы айнымалыларды орта объектілердің қасиеттерімен салғастыру. Айтылулар логикасының семантикасы және табиғи тілдегі сөз орамы тура келеді.

Орындалатын формула – өзінің аргументтер жиынтығының үстінде «Ақиқат» мәнін қабылдайды.

Орындалмайтын формула - өзінің барлық аргументтер жиынтығында «Жалған» мәнін қабылдайды.

Жалпы маңызды формула (тавтология, белгіленеді $\models$) – өзінің барлық аргументтер жиынтығында абсолютты (яғни барлық модельде) шындық болатын формула.

Элементарды (атомарды) айтылулар – қарапайым айтылулар, мысалы: «таң атады», «үй ақ» және т.с.с.

Конституента (толық конъюнкция) – элементар конъюнкция, оған ортаның күйін анықтайтын айнымалылардың әрқайсысы бір рет кіреді. Мысалы, орта «соқыр жаңбыр» - жаңбыр жауып тұр және күннің көзі жарқырап тұр ($x_1 \wedge x_2$), немесе «дауыл» - қатты жел және «түк көрінбейді» ($x_3 \wedge \neg x_4$).

Квантор – толық объектілер көптігіне қандай болса да қасиеттерді қосып жазу әдісі: $\forall$ (ортақтық кванторы), $\exists$ (бар болу кванторы). Квантор белгісі жақшада жазылған айнымалылар байланысқан болады; бірде-бір квантор байланыстырмайтын айнымалылар еркін болады. Кванторлардың бар болуы, және аргумент ретінде айнымалылармен және айтылулармен операциялар жасау мүмкіндігі – бұл предикаттарды есептеудің айтылуларды есептеуден негізгі айырмашылықтың біреуі. Бірінші реттік предикаттарды есептеудің екінші реттік предикаттарды есептеуден айырмашылығы мынадай: бірінші реттік предикаттарды есептеуде біз кванторларды айнымалылар үстіне қоямыз, ал екінші реттік предикаттарды есептеуде біз кванторларды қатынастар (яғни предикаттар) үстіне орналастырамыз. Мысалы, бірінші реттік логикада біз «Кейбір R қатынаста қандай X, Y, Z және т.б. айнымалылар бар?» деген сұранысты тұжырымдай аламыз. Дәл осындай мысал 10-бөлімде келтірілген. Екінші реттік есептеуде мұндай сұраныс мүмкін: «Кейбір X, Y және т.б. айнымалылар қандай қатынастарға кіреді?». Осы мысалға тиісті алынған жауап «likes» болар еді. Логикалық модельге негізделген қазіргі ЖИ жүйелер бірінші реттік предикаттарды есептеуді жүзеге асырады және тек жеке жүйелер (Prolog секілді) екінші реттік предикаттар есептеуді орындайды.

4.3.1. Автоматты түрде теоремаларды дәлелдеу әдістері (предикаттарды есептеу)

Кейбір, ең маңызды аксиомалар.

Коммутативтілік: $\alpha_1 \& \alpha_2 \equiv \alpha_2 \& \alpha_1$, $\alpha_1 \vee \alpha_2 \equiv \alpha_2 \vee \alpha_1$.

Дистрибутивтік: $\alpha_1 \& (\alpha_2 \vee \alpha_3) \equiv (\alpha_1 \& \alpha_2) \vee (\alpha_1 \& \alpha_3)$.

Ассоциативтік: $\alpha_1 \& (\alpha_2 \& \alpha_3) \equiv (\alpha_1 \& \alpha_2) \& \alpha_3$, $\alpha_1 \vee (\alpha_2 \vee \alpha_3) \equiv (\alpha_1 \vee \alpha_2) \vee \alpha_3$.

Де Морган заңдары:

$$\neg(\alpha_1 \& \alpha_2) \equiv (\neg\alpha_1) \vee (\neg\alpha_2),$$

$$\neg(\alpha_1 \vee \alpha_2) \equiv (\neg\alpha_1) \& (\neg\alpha_2),$$

$$\neg(\neg\alpha_1) \equiv \alpha_1.$$

Айтылуларды есептеудің (АЕ) классикалық аксиомалары:

$$\alpha \supset (\beta \supset \alpha),$$

$$(\alpha \supset \beta) \supset (\alpha \supset (\beta \supset \gamma)) \supset (\alpha \supset \gamma),$$

$$(\alpha \wedge \beta) \supset \alpha,$$

$$\begin{aligned}
&(\alpha \wedge \beta) \supset \beta, \\
&\alpha \supset (\alpha \vee \beta), \\
&\beta \supset (\alpha \vee \beta), \\
&\alpha \supset (\beta \supset (\alpha \vee \beta)), \\
&(\alpha \supset \gamma) \supset ((\beta \supset \gamma) \supset ((\alpha \vee \beta) \supset \gamma)), \\
&(\alpha \supset \beta) \supset ((\alpha \supset \neg \beta) \supset (\neg \alpha)), \\
&\neg \neg \alpha \equiv \alpha.
\end{aligned}$$

Шығару ережелері.

1. Импликация жағдайының шындығынан және импликация өзінің шындығынан импликация салдарының шындығы шығады (модус поненс): $\alpha, \alpha \supset \beta \vdash \beta$. *Modus ponens* (лат.) — «бөліп алу ережесі».

2. $\alpha(p)$ формуладан p орнына P -ны ауыстырудан $\alpha(P)$ формула шығарылады (ауыстыру ережесі): $\alpha(p) \vdash \alpha(P)$.

4.3.2. Л.Заде айқын емес көптіктер теориясының элементтері

Жоғары айтылғандай, кез келген пікірлесу не шындық, не жалған, яғни жүйе. «Майра $\rightarrow$ Асқар» екі мүмкін болатын күйлердің біреуінде ғана болу мүмкін: не «Майра Асқарды жақсы көреді», не «Майра Асқарды жақсы көрмейді».

Элементар логика микросызбаларын жасаушылар ең азы үш мүмкін болатын күйлерді ойлап тапты. Шынында да, егер 1 күй (жақсы көреді) кернеудің жоғарғы деңгейімен кодталса, ал 0 күйі (жақсы көрмейді) төмен деңгейімен кодталса, онда не болады, егер берілген сызбада кернеу сірә болмаса, не оның деңгейі 0 мен 1 тура ортасында болса? Осы үшінші күйді былай интерпретациялауға болады: «Майра Асқарға қалай қарайтыны белгісіз» немесе «Асқарға қалай қарайтынын Майра білмейді». Ал егер кернеу деңгейі «жақсы көрмейді»-ге қарағанда «жақсы көреді»-ге жақын болса ше?

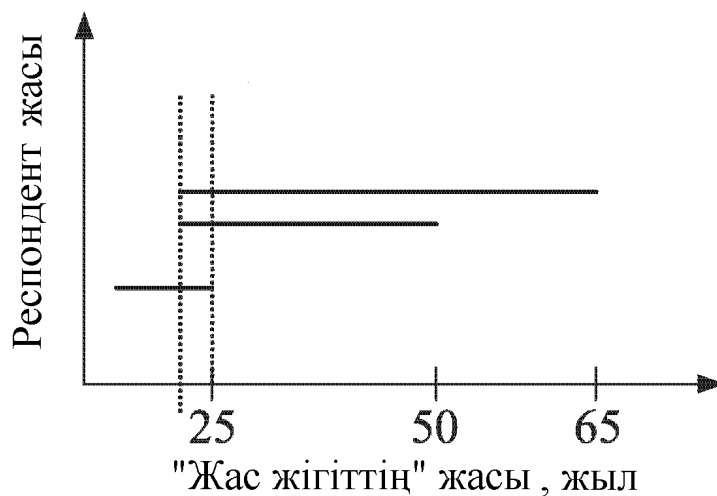
Ұзақ уақыт осындай жағдайды құрылғы жұмысының қателігі деп есептеген. Бірақ, Л.Заде (Lotfi A. Zadeh) 1965 жылы АҚШ-та шыққан өзінің жұмысында бұл қателік өте пайдалы деп көрсетті. Осы «қателік» машинаның жұмыс логикасын адамның пікірлесу логикасына жақындауға мүмкіндік береді және машинаны «күдіктенуге», «варианттарды әр жағынан бағалауға» үйретуге мүмкіндік береді. Ол үшін жүйенің мүмкін болатын күйлерін белгілеп, дәл осы күйде тұратын сенімділік дәрежесін бағалау қажет.

Айқын емес көптік мына түрдегі жұптармен беріледі $x_1 | \mu_X(x_1)$, мұндағы x_1 - айқын емес X көптіктің элементі, $\mu_X(x_1)$ - x_1 элементтің айқын емес X көптікке тиесілік дәрежесі. $\mu_X(x_1)$ мәні $[0...1]$ интервалда өзгереді. Мысалы, кейбір айқын емес көптік A мына түрдегі өрнекпен беріледі

$$A = \{(x_1|0.1), (x_2|1), (x_3|0.6)\},$$

$x_1 = 17$ (жас), $x_2 = 24$, $x_3 = 49$
 немесе бірден $\{(17|0.1), (24|1), (49|0.6)\}$.

Көптік А (4.13-сурет) мына сұраққа жауап берген кезде мағыналы: «Жас жігіт неше жаста?» Түрлі буынға жататын адамдар осы сұраққа әртүрлі жауап береді. Жасөспірім үшін «жас жігіт» - бұл «жасы кіші» деген кісі, олардан жасы үлкен адамдарға – «жасы көп емес» және т.б. Зейнеткерлер он алтыдағы жас жігітті «жасөспірім» категорияға, ал 30 жастағы оқытушыны – «едәуір жас адам» деген категорияға жатқызар еді. Көптің ой-пікрілері бірдей болар еді, егер «жас жігіт 18-25 жаста» болса. Осындай айқын еместік нақты инженерлік есептерде әрқашан орын алады, себебі шеткі ұғымдар «шындық - 1» және «жалған - 0» теорияда ғана болады.



4.13-сурет. Лингвистикалық функция

Қазіргі сараптамалық жүйелер классикалық (Аристотель) логиканың орнына Заде логикусын жиі пайдалана бастады.

5 - БӨЛІМ

БІЛІМДЕР ҰСЫНУДЫҢ ӨНІМДІК МОДЕЛІ

5.1. Заттық саланы ережелер мен фактілер көмегімен бейнелеу

Жоғарыда көрсетілген логикалық шығарудың принциптерін жүзеге асыру үшін, автоматтандырылған жүйеде арнайы машиналы-бағытталған тіл қажет. Логикалық есептерді сипаттау үшін ең қарапайым және тиімділігі машиналы-бағытталған тілдерінің бірі өнімдер ережелері болады.

«Өнім» мына өрнек түрінде жазылады:

егер A (шарт), онда B (іс-әрекет), кейінгі шарт C .
ядро

Кейбір білімдер қорынан іздеуді жүзеге асыратын сөйлем-үлгіні *шарт* деп, ал іздеу сәтті орындалғанда қимылдарды *іс-әрекет* деп түсіндіреді. Іс-әрекеттер аралық (әрі қарай шарт болып есептеледі) және жүйенің жұмысын аяқтайтын мақсатты (терминалды) болу мүмкін. Шарттан кейінгі шарт қимылдарды жүзеге асырғаннан кейін орындауға қажетті іс-әрекеттер мен процедураларды сипаттайды. Мысалы, дүкенде кейбір нәрсені сатып алғаннан кейін, тауарлар тізімдемесінде осы типті нәрсенің санын бірге азайту керек.

Сонымен, өнімдік жүйесі білімдер қоры (өнімдер ережелері) және шығару машинасы болып табылады – «үлгі бойынша салғастыру» арнайы бағдарламасы. Пайдаланған өнімдер мен шығару ережелерінің бәріне тәуелді әртүрлі өнімдер жүйелері алынады.

Өнімдер ядросын әртүрлі негіздер бойынша топтастыруға болады. Біріншіден, детерминды және детерминды емес ядроларды ажыратады. Детерминды ядроның белсенділігін және шартты орындауын оятуда шарттың оң жағы (іс-әрекет) міндетті түрде (сенімділік дәрежесі) орындалады, ал детерминды емес – ықтималдық арқылы.

Егер A , онда B мүмкіндігі α . (классикалық логикада $\alpha = 1$ не $\alpha = 0$).

Егер A , онда B сенімділік коэффициентпен α (детерминды өнім).

Өнімдер де бірімәнді және баламалы болу мүмкін. Баламалы ережелер үшін ядроның оң жағында «баламалы таңдау мүмкіндіктер» көрсетіледі, олар «таңдау салмақтармен — $\alpha_1, \alpha_2, \dots, \alpha_n$ » бағаланады.

Егер A , онда «жиірек» B_1 жасау керек, «сирек» B_2 (ықтималдық бағалар).

$$A \text{ бағалар } \left\{ \begin{array}{lll} \hat{A}_1 & \text{ñáíîä³ë³ê ïáí} & \alpha_1; \\ \hat{A}_2 & \text{ñáíîä³ë³ê ïáí} & \alpha_2; \\ \hat{A}_n & \text{ñáíîä³ë³ê ïáí} & \alpha_n. \end{array} \right.$$

$\alpha_1, \alpha_2, \dots, \alpha_n$ ретінде ықтималдық бағалар, лингвистикалық бағалар, сараптамалық бағалар және т.с.с. пайдаланылады.

Болжам процедураларды жеке ажыратады, оларда A -ны жандандыруда күтетін салдар сипатталады, мысалы: егер A , онда p ықтималдықпен B -ны күтуге болады.

Өнімдік білімдер қоры (БҚ) фактілер мен ережелер жиынтығынан тұрады. Ережелердің іріктеп алуын басқаратын бағдарлама *шығару машина* деп аталады. Көбінесе, шығару тура (деректерден мақсатты іздеуге) немесе кері (мақсаттан оны растау үшін - деректерге) болады. *Деректер* – бұл бастапқы фактілер, солардың негізінде шығару машина қосылады, яғни қорынан ережелерді іріктейтін бағдарлама.

БҚ абстракты мысалы (5.1-сурет).

«Фактілер» бөлімі:

жол (A, B) ;

жол (B, C) ;

жол (B, D) ;

жол (C, D) .

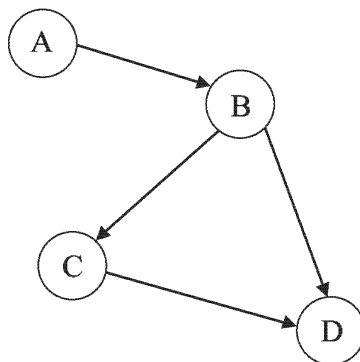
«Ережелер» бөлімі:

тура жол (X, Y) , егер жол (X, Y) немесе жол (Y, X) ;

транзит жолы (X, Y) , егер жол (X, Z) немесе жол (Z, Y) ;

жол (X, Y) бар, егер тура жол (X, Y) немесе транзит жолы (X, Y) .

Осындай БҚ көмегімен шешілетін есептің мысалы: « A мен D пункттері арасында жолдың бар болуын анықтаңыз».



5.1-сурет. Абстракты жолдар картасы:
есімдер арасындағы қатынастардың графикалық түсіндірмесі

Шешім барысы (5.2-сурет).

1. Шығару машинаның (әрі қарай ШМ) кіреберісіне пайдаланушы мақсатты береді:

« (A, D) жол бар».

2. ШМ «Жол бар» деген мәтін жолын БҚ сақталынған файлдан іздейді.

3. ШМ «Жол бар» мақсатты енгізеді, мақсаттар стекке және «Тура жол» мақсатты іздей бастайды.

4. ШМ «Тура жол» мақсатты енгізеді, мақсаттар стекке және «Жол» мақсатты іздей бастайды.

5. ШМ барлық мәтіндік жолдарды «Жол (A, B)», «Жол (B, C)» және т.б. карап шығады және іздеген мақсаттың «Жол (A, D)» жоғын анықтайды.

6. ШМ стектен «Жол», «Тура жол» және «Жол бар» деген мақсаттарды итеріп тастайды. «Жол бар» мақсатты ашады және «Транзитті жол» мақсатты іздеуге кіріседі, алдымен, «Жол бар» мақсатты стекке кіргізіп.

7. «Транзитті жол» мақсаты қанағаттандырылады, өйткені шынында да « $A — B — D$ » жолы бар.

Стек деп деректерді сақтау үшін құрылым аталады, деректер элементтерін LIFO (бірінші болып кірді – соңғы болып шықты) принципі бойынша қабылдайды және береді.

"Жол бар (A, D)"	"Тура жол (A, D)"	"Жол"	"Транзитті жол (A, D)"
	"Жол бар (A, D)"	"Тура жол (A, D)"	"Жол бар (A, D)"
		"Жол бар (A, D)"	
1 қадам	3 қадам	4 қадам	6 қадам

5.2-сурет. Логикалық есепті шешу кезіндегі стек жұмысы

Бұл мысалда «кері шығару» деп аталатын шешу көрсетілген – бар болған ережелер мен деректер көмегімен кейбір мақсат расталған.

Тура шығару орын алар еді, егер берілген картада барлық мүмкін болатын маршруттарды көрсету қажет болса, бұл жағдайда мақсат мұндай болу мүмкін «Жол (X, Y)».

Көрініп тұр, алгоритм жағынан шешімнің іздестіруі (адамның ойлауын еліктеу) циклдағы бағыныңқы жолды іздеуге ұқсайды. ЭЕМ уақыты мен ресурстарын үнемдеу үшін осы іздестірудің тиімді басқаруы өте маңызды. Күйлер кеңістігінің графикалық бейнелеуі осы есепті бірталай оңайлатады (5.3-сурет).

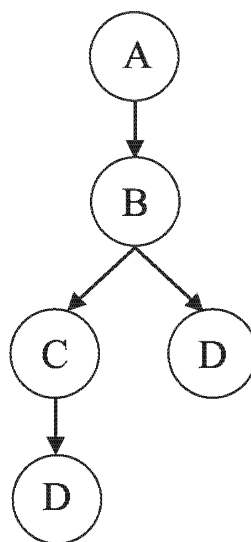
Екі шешім болу мүмкін « $A — B — D$ » және « $A — B — C — D$ ». Осындай жазу «есептің шешу жолы» деп аталады. Бірінші шешім екіншіден сөзсіз қысқа, бірақ « A » пунктте болып, біз оны білмейміз!

« A » шынды «ағаш түбірі», «мақсатты шың» немесе «ғаламдық мақсат» деп атайды.

« B » мен « C » шындар – есептелетін (ашылатын, аралық) шындар деп аталады.

« D » шыңы – терминалды, яғни аяқталатын.

Шындарды қосатын қабырғалардың мағынасы - қосылған процедуралар, оларды келесі күйге (шыңға) ауысу үшін орындау қажет.



5.3-сурет. Абстракты жолдар картасы:
есімдер арасындағы қатынастардың графикалық түсіндірмесі

Қосылған процедуралардың қолдану процесін *шыңдарды тудыру* немесе *варианттарды іріктеп алу* деп атайды. Жаңа шыңды тудырған кезде ескі шыңға сілтегіш міндетті түрде есте қалдырылады. Іріктеп алудың аяғында осы сілтегіштердің жиынтығы есептің шешу жолын жасайды. Бұл жол орындалған қосылған процедуралар аттарымен бірге жазылады, мысалы:

$A = (r1(r2(r3)))$, мұнда r — граф қабырғалары; немесе $A(B(C(D)))$.

Әрбір мүмкін болатын мақсаттар ағаштың «бұтағын» терминалды фактілерге дейін немесе берілген тереңдікке дейін ашуға байланысты, соқыр іріктеп алудың (бұл үдерісте мақсатты шыңның орналасуы ашудың ретіне әсер етпейді) екі негізгі стратегиясын ажыратады: «ені бойынша» және «тереңдік бойынша».

Бірінші жағдайда шыңдар тудырылған ретке сәйкес ашылады (*breadth — first process*). Екінші жағдайда әрбір қадамда алдымен, соңғы құрылған шың ашылады (*depth — first process*).

Егер күйлер кеңістігінің графы және мақсаттың орналасуы туралы пікір алуға мүмкіндік беретін заттық саласы бойынша кейбір қосымша ақпарат бар болса, онда эвристикалық («ашуға қызмет ететін») іздестіру туралы айтады. Бұрыңғы тәжірибеге негізделетін эвристикалық ақпарат ең перспективалы бағыттарда іздестіруді орындауға мүмкіндік береді.

Граф туралы айтқанда, оның тек ең қарапайым типін қарастырамыз — «ағаш» типті граф. *Ағаш* — бұл әрбір шыңының алдында тек бір ғана шыңы (аналық шың) болатын граф. Түбір-шыңнан басқа, оның алдында шыңдар жоқ.

5.2. Ені бойынша толық іріктеп алу әдісі

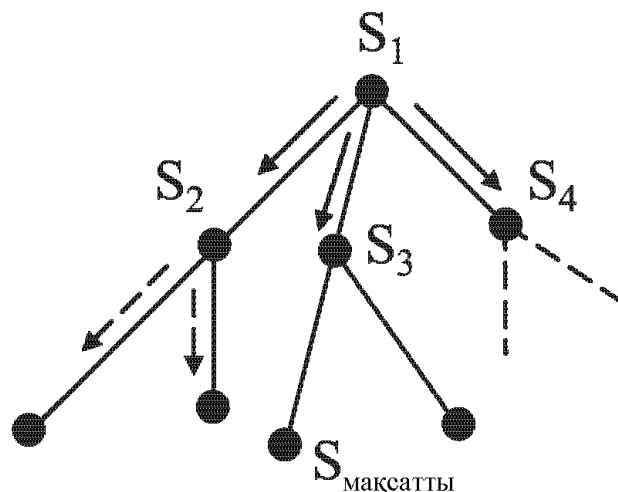
Шындар қандай ретімен құрылса, сондай ретімен ашылады (5.4-сурет). Негізгі алгоритм келесі әрекеттердің орындауынан тұрады.

1. Бастапқы шың ашылғанша, бірдей (немесе әртүрлі, шартқа қарағанда) операторды қолдап оны аша береді. Осы кезде бірінші деңгейдің шындары $S_2, S_3 \dots$ пайда болады. Олар өзінің ретімен ашылады және екінші деңгейдің шындарын жасайды және т.б.

2. Жаңа шындардан түбірге жүргізетін сілтегіштер (шартты есімдер, әріптер, цифрлер, оператор аттары, қашықтықтар, бағасы, салмақ және т.б.) қойылады.

3. Алынған шындардың арасында мақсатты бар болғаны тексеріледі. Егер бар болса, онда сәйкес операторға негізделіп шешім жасалынады. Егер мақсатты шындар жоқ болса, онда бірінші тудырылған шың қарастырылады және оған да сол алгоритм қолданады.

Осыдан кейін алынған шындардың арасында мақсаттыны тапқанша екінші және т.б. ауысып отырады.



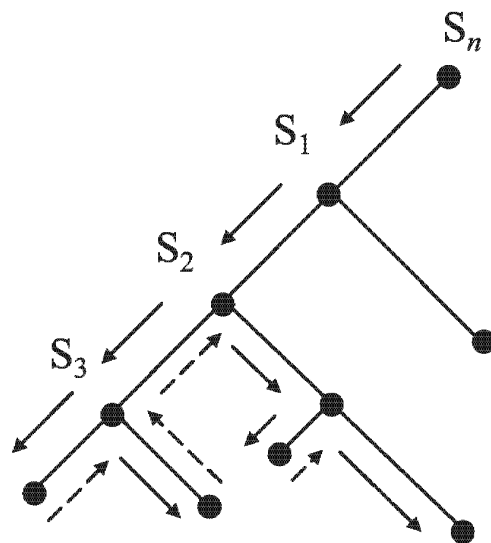
5.4-сурет. Ені бойынша іріктеп алу ағаштың фрагменті

Ені бойынша толық іріктеп алу мақсатты шыңның табуына кепіл береді, себебі іріктеп алу толық жүргізіледі. Мақсатқа жетудің бірнеше жолдары болу мүмкін. Бұл жағдайда ең қысқа (ең арзан, ең оңай, ең жылдам және т.с.с.) жолды таңдауға болады. Бірақ, іздестіру графы шексіз бола қалса, онда алгоритм өз жұмысын аяқтай алмайды. Классикалық мысал – «лабиринтты іздестіру» есебі, оны бірінші шешкен К.Шеннон. Мұнда түрлі әрекет варианттар кеңістігі үлкен емес: «оңға бұрылу», «солға», «алға» және шешім кейбір берілген тереңдікте (әдетте кішкентай) жатады.

5.3. Тереңдік бойынша толық іріктеп алу әдісі

Тереңдік бойынша іріктеп алуда (5.5-сурет) алдымен, соңғы құрылған шыңдарды ашу керек. Бірінші, ендеше соңғы да ашылатын шың бұл түбір шыңы. Процесс әрқашанда шыңдардың ең сол жақ тармағы бойынша өтеді. Іріктеп алуды қалай болсада шектеу үшін, іріктеп алу ағашта шыңның тереңдігі деген ұғым енгізіледі. Ағаштың түбір тереңдігі нөлге тең, ал әрбіреу келесі шыңның тереңдігі оның алдындағы шыңның тереңдігіне тең бір плюс. Ең үлкен тереңдік осы мезетте ашуға қажетті болатын шыңда. Егер жасалынған жол пайдасыз болса (яғни берілген тереңдікте мақсатты шың ашылмаса), онда ашылғанның алдында болған шыңға қайтып келу керек және оған тағы да, мақсатты шыңды алғанша, ашу операциясына әркеттену қажет. Бұл процесс (бірнеше қадамға қайтып келу) артқа шегіну немесе бэктрекинг деп аталады.

Артқа шегіну сілтегіштер көмегімен жүзеге асырылады. Берілген шекаралық тереңдікке жеткеннен кейін, осы шекарадан аспайтын ең үлкен тереңдікті шың ашылады. Бұл «кері қадағалауы бар бағдарламалау» (*back – track programming*) деп аталады. Тереңдік бойынша іріктеп алудың сұлбасы 5.5-суретте көрсетілген.



5.5-сурет. Бэктрекинг нұсқауы бар тереңдік бойынша іріктеу ағашы

Тереңдік бойынша іріктеп алудың алгоритмы келесі:

1. Бастапқы күйге сәйкес бастапқы шың ашылады.
2. Бастапқы шыңның ашу нәтижесінде алынған бірінші шың ашылады. Сілтегіш қойылады.
3. Егер ол ашылса, онда келесі болып қайтадан тудырылған шың ашылады. Егер шың ашылмаса, онда процесс алдындағы шыңға қайтып келеді.
4. Мақсатты шыңды алғаннан кейін ашу процесі аяқталады және сілтегіштер арқылы түбірге келтіретін жол құрылады. Доғаларға сәйкес операторлар есептің шешімін жасайды.

5. Егер берілген ашу тереңдігі үшін мақсатты шың табылмаса, онда барлық процесс қайталанады, ал жаңа шың ретінде алдындағы кезеңде алынғанның ең сол жақ шыңы қарастырылады.

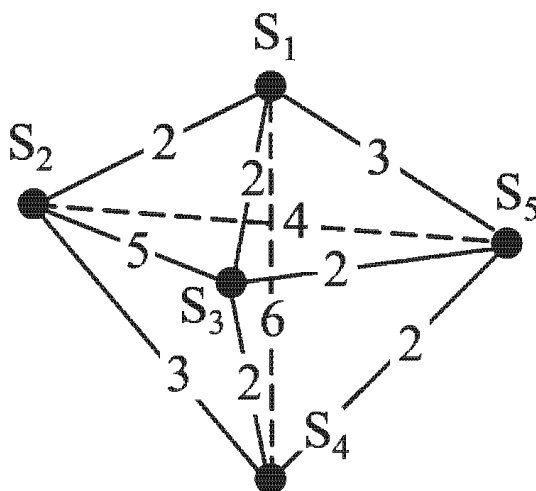
5.4. Күйлер кеңістігінде іздестірудің эвристикалық әдістері

Толық іріктеп алудың әдістері есептің шешімін кепілдейді, егер ол бар болса, ал бірнеше шешім болса, онда ең тиімділігін растайды. Бірақ, тәжірибеде бұл әдістер тек үлкен емес күйлер графтары үшін пайдаланады.

Нақты жағдайлар үшін бұрынғы тәжірибеге сүйенетін немесе теориялық шығармаларға негізделген қосымша ақпарат көбінесе, пайдаланады. Осындай ақпарат *эвристикалық*, ал ережеге ұйымдастырылған – *эвристикалық ережелер* немесе *эвристика* деп аталады.

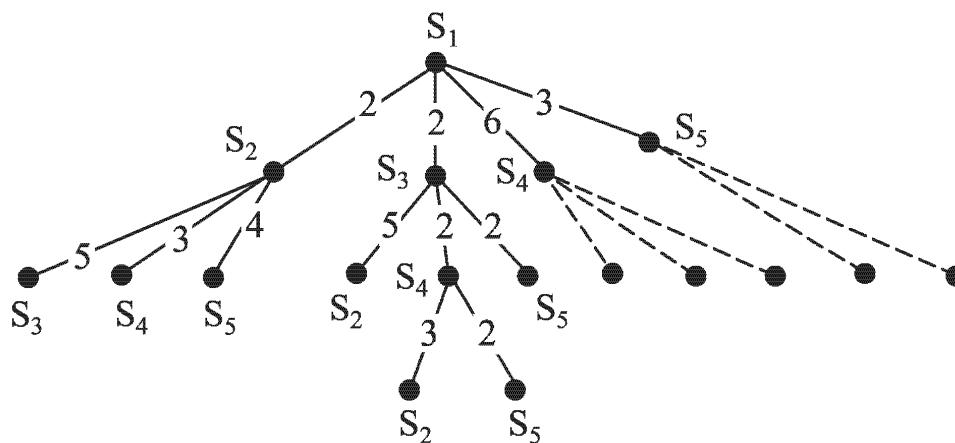
Эвристикалық ақпараттың сипаты арнайы және тек берілген есепке ғана қолданады. Эвристикалық ақпарат іріктеп алуды реттелгенге айналдырады. Мысалы, «Беллманның динамикалық бағдарламау әдісі», «бұтақтар мен шекара әдісі» және т.б.

Мысал ретінде коммивояжер туралы танымалы есепті қарастырайық. Көшпелі сатушы N қаланың бәрінде бір рет болып шығып, бастапқы қалаға қайтып келуі тиіс. Маршрут ұзындығы минималды болу керек (5.6-сурет).



5.6-сурет. Коммивояжер туралы есептің күйлер графы

Күйлер кеңістігінің фрагменты 5.7-суретте келтірілген. Ені бойынша іріктеуді бастаймыз және бірден бірінші деңгейде әртүрлі ұзындығы бар мүмкін болатын жолдар аламыз: 2, 2, 6, 3. Егер «әрбір қадамда минимал ұзындығы жолды таңдау» эвристикадан бастасақ, онда келесі қадамдарды жасау керек $S_1 \rightarrow S_2$ және $S_2 \rightarrow S_3$, сосын $S_3 \rightarrow S_4$ немесе $S_3 \rightarrow S_5$, сосын $S_4 \rightarrow S_5$ және т.б.



5.7-сурет. Коммивояжер туралы есеп үшін іріктеу ағаштың фрагменті

Коммивояжердің барлық мүмкін жолдарын қосатын толық іріктеу графында $(N - 1)!$ варианты болады. Егер кері жолдарды есептемесек, онда $(N - 1)!/2$. Осындай типке жататын нақты есептерде (мысалы, пейджинг таратқыштың орналасуы, банкоматтар орналасуы және т.с.с.) N әдетте, бірнеше ондаған объектілерге тең болады.

Эвристикалық алгоритмдер бір ғана дұрыс (оптимальды) шешімді іздеу үшін емес, ал кейбір критерийге сәйкес көбінесе, бірінші шешімді іздеу үшін қолданылады. Мысалы, адам дүкенге барғанда, ең арзан сүтті алу деген мақсат өзіне қоймайды, ол белгілі сапасынан төмен емес және кейбір бағадан қымбат еместі сатып алғысы келеді және осыған 5 минуттен артық уақыт жұмсамайды.

5.5. Бағыныңқы есепке бөліп есепті шешу әдістері

Есептің өзін және оның шешу әдісін түсіну үшін графикалық ұсыныс өте маңызды.

Жоғарыда көрсеткендерге сәйкес, графтар қарапайым болып табылады, өйткені олар тек заттық саласының объектілер арасындағы өзара байланысты бейнелейді және іс-әрекеттер арасындағы өзара байланысты көрсетпейді. Басқаша айтқанда, жоғарыда қарастырылған графтарда доғалар (қосылған процедуралар) арасында ешқандай қатынастар берілмеген.

Осындай қатынастар болу мүмкін, мысалы, тізбектілік және бір уақыттылық қатынастары.

5.6. Есепті ЖӘНЕ-НЕМЕСЕ граф түрінде ұсыну

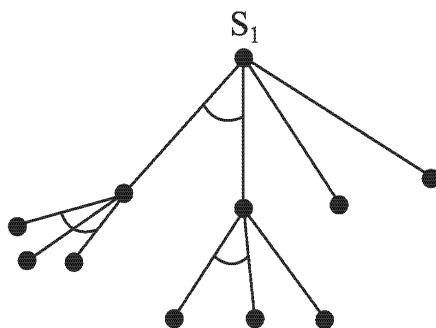
Бөлген кезде алынған бағыныңқы есептердің арасында олардың шешімдерінің ұйғарушылық (бір уақыттылық) қатынастары («ЖӘНЕ» қатынасы), немесе баламалық қатынастары («НЕМЕСЕ» қатынасы) болу мүмкін (5.8-сурет).

«ЖӘНЕ» типті қатынастарды граф қабырғаларын байланыстыратын доғамен (кейде екілік) белгілейді. Бір қалпына келтіру үшін қосымша (жалған) шыңдарды енгізуге болады.

Сонымен, бастапқы есеп баламалы «HEMЕСE»-сипаты бар бағыныңқы есептер арқылы беріледі, ал бағыныңқы есептердің өзі «ЖӘНЕ» типті қатынастармен беріледі. Балама шыңдар («HEMЕСE»-шыңдар деп аталады.

«ЖӘНЕ-HEMЕСE» граф үстінде іздестірудің негізгі мақсаты - S_i шыңның шешілетінін көрсету. Шың *шешілетін* деп аталады, егер келесі шарттардың біреуі орындалса:

- 1) S_i шыңы ақырғы (терминалды) болады;
- 2) S_i -дан кейінгі шыңдар «HEMЕСE» типті шыңы болып табылады және олардың аз болғанда біреуі шешілетін болады;
- 3) S_i -дан кейінгі шыңдар «ЖӘНЕ» типті шыңы болып табылады және олардың барлықтары шешілетін болады.



5.8-сурет. «ЖӘНЕ-HEMЕСE» ағаштың фрагменті

Шешуші граф – бұл шешілетін шыңнан тұратын және түбірі бастапқы шыңда орналасқан бағыныңқы граф.

Егер графтың «ЖӘНЕ-HEMЕСE» шыңнан кейін басқа шыңдар болмаса, онда осындай шыңы *шешілмейтін* деп аталады.

Жаңа шыңдарды тудыру (есептің редукциясы) жалпылама операторды (яғни, мүмкін болатын көпшіліктен қандай болса да оператор; мысалы былай белгілейік $g \in G$) қолдану арқылы орындалады. Есептің сиппатамасына g оператордың қолдануы графтың (редукция графы) барлық «ЖӘНЕ-HEMЕСE» құрылымын тудырады.

5.7. Өнімдер жүйесін басқару

Өнімдер ядроларының қолдану шарттарын орындау кезіндегі басты мәселе – осы жағдайда белсенді болатын өнімді таңдау. Осы есептің шешуі өнімдер жүйесін басқару жүйесіне тапсырылады.

Егер АЖ параллельды архитектурасы бар ЭЕМ жүзеге асырылса, онда дайын өнімдерден біреуі емес, параллельды бұтақтарға сәйкес берілген жағдайда бір уақытта орындалатын бірнеше өнімдер таңдалады.

Шешудің екі: орталықтандырылған және орталықтандырылған емес әдісі мүмкін. Біріншісінде актуализация туралы шешім арнайы басқару жүйесімен, екіншісінде, осы мезеттегі құралатын жағдайға байланысты қабылданады.

5.8. Өнімдік модельдің артықшылықтары мен кемшіліктері

Өнімдік моделі басқаларға қарағанда, өнеркәсіптік сараптамалық жүйелерде жиірек қолданылады. Оның көрнекілігі, жоғары модульдіктігі, қосымшылар мен өзгерістерді оңай енгізуі және логикалық шығару механизмның қарапайымдылығы әзірлеушілерді қызықтырады. Өнімдік тұрғыны жүзеге асыратын көп программалық құралдар бар:

- Prolog, Lisp (Джон Маккартни (АҚШ) 1961 ж. жасаған. LISP — List processing language — тізімдерді өңдеуге арналған тіл), LOGO (LISP негізінде Сеймур Пайперт (АҚШ, М.І.Т.) 1969 ж. жасаған), OPS тілдері;

- «қабықшалы» немес «бос» СЖ, мысалы EMYCIN, EXSYS, ESISP, ЭКСПЕРТ;

- құрал-сайманды жүйелер: ПИЭС, СПЭИС және т.б.

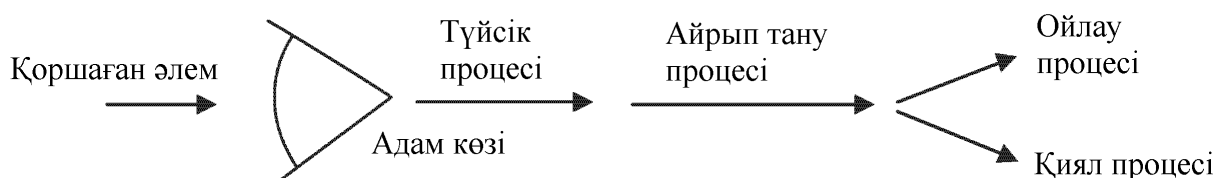
Өнімдік модельдердің екі маңызды кемшілігі бар. Өнімдер саны тым көп болғанда, өнімдер жүйенің қайшылық еместігін тексеруге өте күрделі болады. Сондықтан, жаңа өнімдерді қосқанда, оларды тексеруге көп уақыт кетеді. Жүйе детерминды емес (орындалатын өнімнің таңдауы бір мағыналы емес), жүйенің дұрыс жұмыс істегенін тексеру оңай емес. Егер АЖ өнімдер саны мыңдағаннан асатын болса, онда өнімдер жүйесінің дұрыс жұмыс істеуіне ешкім жауап бере алмайды.

6 - БӨЛІМ

БІЛІМДЕР ҰСЫНУ ҮШІН ФРЕЙМДЕР

Жасанды интеллектті зерттеудің ең басты проблемасы - білімдер ұсыну проблемасы (яғни жадты ұйымдастыру проблемасы: жадынан басқа өзгерту механизмдер қажет, мысалы ойлау, қиял және т.б.), бұл бейнені айырып тану есептерден басталып, табиғи тілді түсіну есептерге дейін жалғасады. Әлемнің машиналық моделін құру күрделілігі осы әлемнің әртүрлігінде. Психологиялық негізделген және тәжірибелік бағалы модельдердің біреуін американ ғалымы Марвин Мински ұсынды. 1974 жылы оның «Білімдер ұсыну үшін фреймдер» кітабы шықты (6.1-сурет).

Фрейм (*frame* (ағыл.) — кадр, рамка, қаңқа) дегеніміз – стереотипті ситуацияны ұсыну үшін деректер құрылымы. Әрбір фрейммен түрлі ақпарат ассоциацияланады. Оның бір бөлігі берілген фреймді қалай пайдалануға болатынын, басқа бөлігі – оның орындауы неге келтіретінін, үшінші бөлігі – егер болжаулар орындалмаса не істеу керек екенін көрсетеді.



6.1-сурет. Фрейм теориясының көзқарасынан психика

Мәтіндерді талдау үшін жасалынған және психикалық іс-әрекеттің тек жеке кезеңдерін ғана қамтитын семантикалық желілерге қарағанда, фрейм теориясы жеткілікті толық деп саналады. Бұл теориясында ойлау процесінен (мұны Аристотель силлогистика мен семантикалық желілер де қарастырады) басқа түйсік, айырып тану, ойлау мен қиял (ойлаудың ең жоғары түрі ретінде) процестері де қарастырылады. Фрейм теориясының бастауы гештальт (психология) салада сірә жатыр деп айтуға болады.

Ойлаудың параллельдігі (нейрон саны тым үлкен болғандықтан адам миындағы процестердің «табиғи» параллельдігі) туралы қазіргі тезисті М.Мински бір мәнді емес деп санайды. Ол жоғары деңгейдің процесі параллельды деп есептейді. Осындай процестерге ол қиял процесін жатқызады. Ойлау мен айырып тану процестерді тізбекті деп санайды. Білімдер ұсынуға логикалық көзқарасты ол априорды шектелген және тек оқу есептерге ғана қолдауға болады деп санайды.

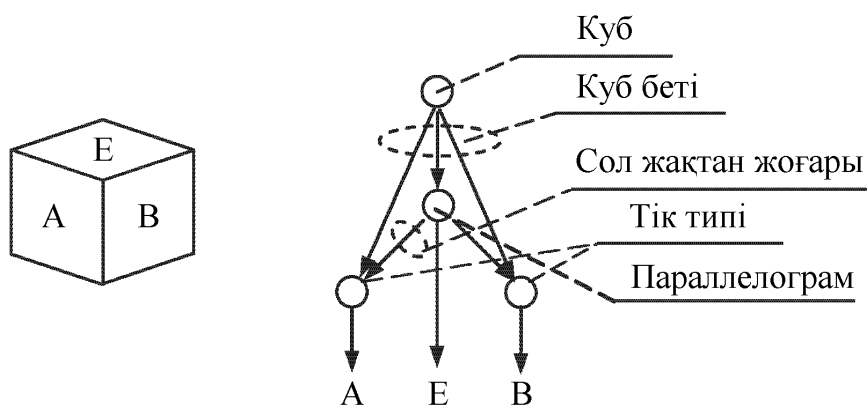
Өзінің теориясының жасауын М.Мински түйсік және кеңістіктік сахналарды (куб, бөлме, бөлмедегі бұйымдар) айырып тану процестердің қарастырудан бастады. Сосын есепті күрделендіру үшін қозғалыс (бөлме бойынша роботтын) қосты. Ол үшін уақыт пен сахналар динамикасын, яғни байқаушы қозғалған кезде бұйымдардың геометриялық бейнелерінің өзгерісін еске алу керек болды. Нақты уақытты бұзбау немесе оған жақындау

қажеттілігі «күтуді» (білімнің кейбір элементін айырып танудан алдын ала анықтау) енгізуге әкелді. Зерттеу барысында М.Мински мына тұжырымға келді: болжамға қарамастан айырып тану процесі (адамда) өте ұзақ уақыт алады, ол объекті («күту») көріну алдында басталады және айқын аяқталмайды. Бұл процесс өзінің табиғаты бойынша тізбекті, себебі микро жағдайларды егжей-тегжейлі зерттеуді және оларды басқа микро жағдайлармен келісуді қажет етеді. Сосын, ол өзінің теориясын растау кезінде, қайталанған айырып тану процесін талқылады – жаңа ғана өткен бөлмеге роботтың қайтып келуі. Осында М.Мински есіктің сыртына бір минутқа шыққан адам әбден басқа бөлмені тапқаны туралы мысал келтірді.

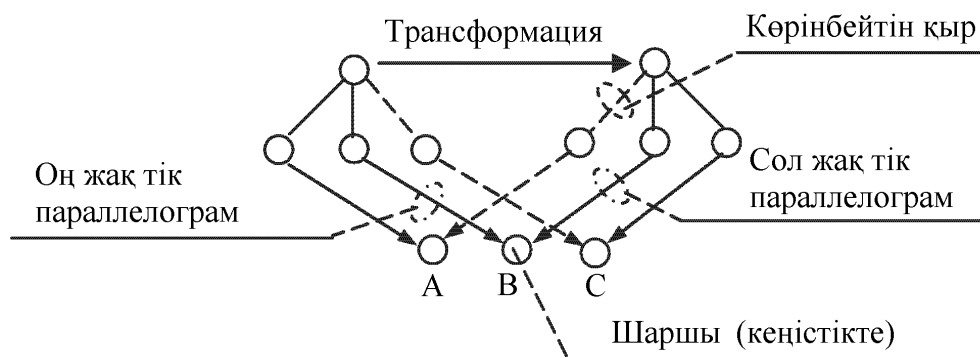
Айырып тану, ойлау және қиял процестерді М.Мински фреймдерге арналған типтік операциялар ретінде көрсетті: кейбір фреймді өзіне сәйкес желіден таңдау, оны толтыру (бос мәндері де болу мүмкін) және жаңа фреймге ауысу (кейбір байланыс көмегімен). Адамға ұқсайтын роботтардың көзі туралы проблемамен жұмыс істеген кезде, ол мұндай тұжырымға келді: адамның санасында және соқыр сезімінде (түйсігінде) кейбір «басты» фреймдердің жиынтығы бар. Кеңістіктік сахналарды айырып тану үшін, ол «көкжиек сызығы», «үсті», «асты», «оң жағы» және «сол жағы» бар «бөлмені» немесе абстракты «қорапты» анықтады. Осындай қорапты оң жағындағы қабырға сол жақта болып қалатынан қорықпай айналдыруға болады (кейбір көзқарас жағынан, яғни кейбір көкжиек сызығы жөнінде).

Білімдерді фрейм арқылы ұсынуы, байқаушының позициясы өзгергенде, бұйымдар тұрған орнын қайтадан есептеуден бас тартуға болады. Мысалы, баланың кеңістіктік ойлауы (фреймдер жүйесі) 8-9 жасқа таман толық қалыптасады.

6.2-, 6.3-, 6.4-суреттерде М.Минскидің өзіндік фреймдері көрсетілген.

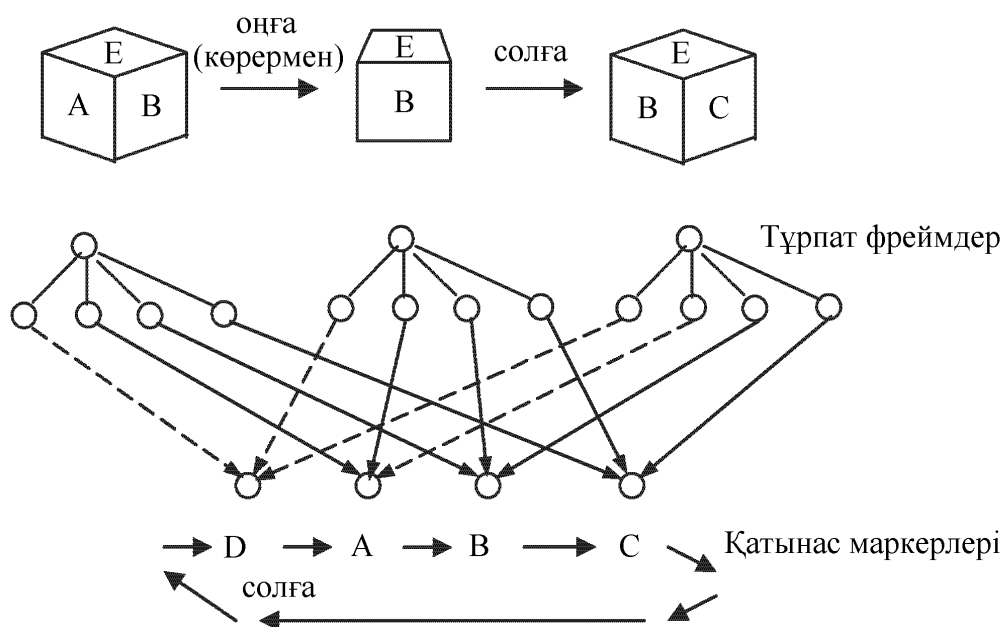


6.2-сурет. «Куб» фреймы



6.3-сурет. Фреймдердің трансформация процесі

Байқаушының (объектіге қарағанда) орны ауыстыру кезінде фреймдердің трансформациясы пайда болады (6.4-сурет). Байқаушының әртүрлі позициясына сәйкес жақын фреймдер тобы желі жасайды.



6.4-сурет. Фреймдер желі

«Фреймды түйіннен және олардың арасындағы байланыстан тұратын желі түрінде көрсетуге болады. Фреймның жоғарғы деңгейлері айқын анықталған. Төменгі деңгейлерде көп ерекше *терминал-шыңдар* немесе ұяшықтар бар, олар *сипатты мысалдармен* немесе *мәліметтермен* толтырылу қажет. Әрбір терминалмен *тапсырмаларға* сәйкес болатын шарттар қойылуы мүмкін. Қарапайым шарттар *маркер* арқылы анықталады. Одан күрделі шарттар әртүрлі терминалдық шыңдарға кіретін ұғымдар арасындағы қарым-қатынастармен анықталады. Семантикалық бір-біріне жақын фрейм топтары *фреймдер жүйеге* бірлеседі».

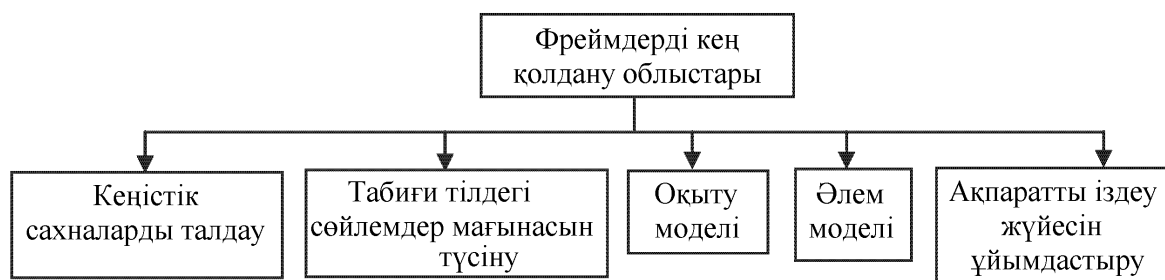
Сонымен, фреймдер жүйесі қалай жұмыс істейді? Адам кейбір бұйымға қарайды және саналы немесе санасыз түрде өзіне белгілі фреймдерден ең жақынын тандап алады. Бұл *келісу* процесі деп аталады. Келісу бірнеше

сатылардан тұрады. Егер қанағатты нәтиже алынса, келісу тоқтатылады және фреймның терминалдары шектеуге сәйкес міндеттермен толтырылады. Қарапайым уақиғада, жағдайға дәл тиісті фрейм табылады, нақты жағдайда болатын терминалдар және фреймдармен салғастырылады.

Егер бұл мүмкін болмаса, онда *ақтауды* сынап көруге болады, яғни түсіндіру - фрейм келеді, бірақ объектінің кемшіліктері бар, мысалы, сынуы бар. Егер фрейм «қайтадан келмесе», *кеңесу* жүргізіледі, яғни не істеу керек айқын нұсқаулар беріледі. Және соңғы сәтсіздік уақиғада, *түйіндеме* жасалынады – фреймнен бас тарту және бас тартудың себебін анық көрсету.

Келісу процесі аяқталғаннан кейін адамда бұйым туралы кейбір абстракты елес пайда болады. Фрейм түйіндерінің көбі әлі анықталмаған. Өзінің жұмысын *нақтылау* процесі – субфреймдер анықталады, тапсырмалар айқындалады. Нақтылау кезінде жеңілмейтін қиындықтар кездесу мүмкін, онда *өзгеріс* процесі белсенділеді. Бұл екі бұрынғы фреймға негізделген жаңа фреймды тудыру мүмкін. Нақтылау процесі барысында фреймның терминалдары толығымен толтырылады және егер фреймның қайталанған пайдалануында қайшылықтар шақырылса, *жаңарту* процесі белсенділеді. Осы жолмен алынған фреймның белгілі бір құны болады. Соған байланысты ол пайдаланғаннан кейін не жойып тасталу мүмкін, не бағалы үлгі ретінде ұзақ мерзімді жадқа салыну мүмкін. Соңғы процесі *оқыту* деп аталады.

Әртүрлі міндеттер қолдануда адамда әрбір уақыт мезгілінде, әртүрлі ғаламдық фреймдер жүйелері болады (6.5-сурет). Олар адамға кеңістікте бағдарлауға, қажетті ақпаратты іздеуге, поэзияны түсінуге және т.б. мүмкіндік береді.



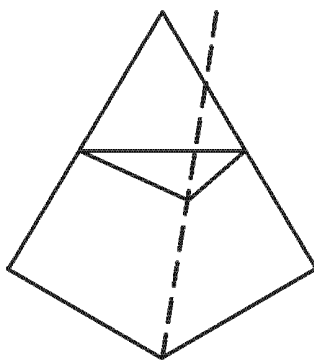
6.5-сурет. Фреймдерді кең қолдану салалары

Кейбір көзқарас арқылы орындалатын және келісуді жетуге бағытталған бір фреймнен басқа фреймге ауысу (трансформация) *басқару процесі* деп аталады. Теориялық тұрғыдан басқару орталықтандырылған болу мүмкін, мұнда кейбір суперфрейм (орталық фрейм) субфреймды тапсырмалармен толтыру процесті басқарады. Керісінше, теориялық тұрғыдан басқару орталықсыздандырылған да болу мүмкін, алдымен субфреймның терминалдары өзіндік түрде толтыруға әрекеттенеді. Бұл процестердің біреуі де басқару есебінің шешімі болып саналмайды, өйткені оның аяқталуы осы екі процестердің өзара әрекеттесуін қажет етеді. Шынында, егер Сіз кейбір объекті «кіре беріс» деп айырып танысаңыз, онда

Сізге оны «үй» деген субфреймге қосу керек. Бірақ, «үй» фрейм ішінде ықтимал болатын әртүрлі кіре берістерді айырып тану мүмкін емес.

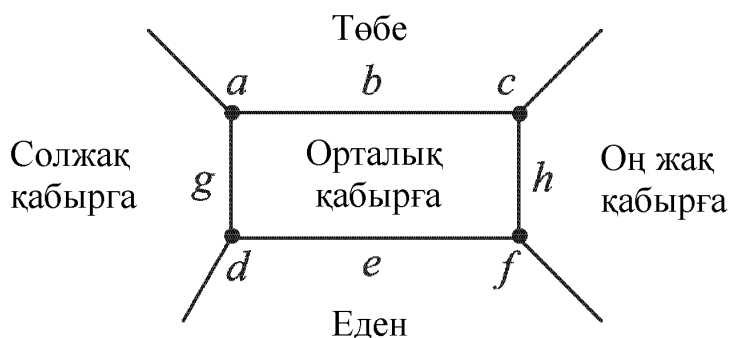
6.1. Кеңістік сахналарды талдау

Фотографиялық қағазда, робот көзінің жарық диодты матрицада және т.с.с. алынған бейнелерді талдаған кезде, объектілер арасындағы арақашықтықты өлшеумен байланысты, бірнеше ерекше проблемалар пайда болады: перспектива проблемасы (перспективаны бұрмалау, жабу және қаптау), 2-өлшемді не 3-өлшемді кеңістіктің өлшемін талдау проблемасы, объекті фоннан ажырату проблемасы. Адамдар осындай есептерді жақсы шешеді (6.6-сурет). Суретте пирамида көрсетілген деп ойлаймыз, бірақ олай емес. Бұл жағдайда келісу механизмі жұмыс істеді – «пирамида сынған» деп ойлаймыз.



6.6-сурет. «Сынған пирамида?»

Басқа маңызды сұрақтар: «Сурет үш өлшемді ме?», «Оны екі өлшемді жиынтығы ретінде қарастыруға бола ма?», «Суретті қарау процесі неден басталады». М.Мински бұл сұрақтарды «бөлме» фреймның мысалында қарастырды (6.7-сурет).



6.7-сурет. «Бөлме» фреймның жоғарғы деңгейінің құрылымы

Бөлмеге кіруімен адам «көкжиек сызығын» іздейді, өйткені соған байланысты ол кейін «сол жақ», «оң жақ», «жоғарғы», «төменгіні» анықтай алады, мұндай ережелер бойынша:

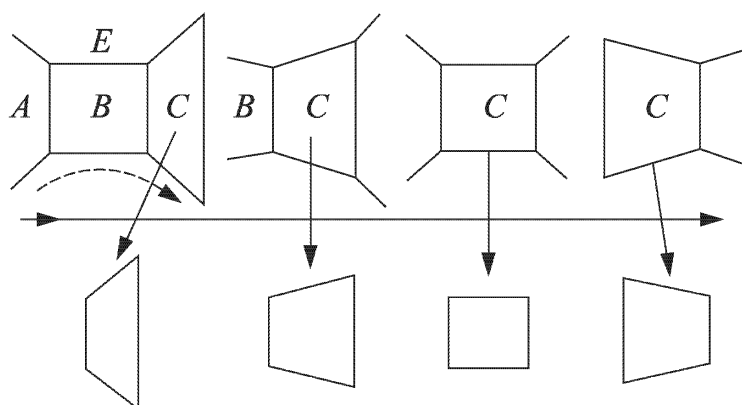
- шекара горизонтал сызық емес;
- шекара көз деңгейінен төмен орналасады;
- шекара еден мен қабырғаны бөлмейді.

Онан әрі қабырғалардың қарауы жеке жүргізіледі, бірақ қажет болғанда «бөлме» фреймы белсенділеді (6.8-, 6.9- суреттер). Төртбұрышты қарастырған кезде, ол қабырғада ма, не қабырғаға перпендикуляр ма түсініксіз, сондықтан «бөлме» фреймы қажет. Бұл не? Ашық терезе ме? Қабырғадағы суретпе (мұны перспектива талдауы деп айтуға болады)?



6.8-сурет. Ашық желдеткіш немесе қисық терезе?

Осы сұраққа жауапты трансформация көмегімен – басқа көзқарасқа ауысу ретімен табуға болады. Байқаушы өз орнын ауыстырады және қабырғалар проекциясы өзгереді (6.9-сурет). Сонда белгісіз объект қабырғалардың біреуінің өзгеріс заңына бағынады, демек ол оған параллель (мысалы, үстінде жатыр).



6.9-сурет. Бөлменің ішінде роботтың қозғалысы

Енді бөлмені нәрселермен толтырайық. Қаптау проблемасы пайда болады – бөлме бойынша қозғалған кезде объектілер геометриясының өзгерісі тек проекция заңдарына бағынбайды. Жаңа бұйымдар әрдайым көзге түсе бастайды және біз қарайтын элементтерді жауып қаптайды. Қаптау проблеманың шешуі қиялды модельдеуге мүмкіндік береді.

6.2. Сөйлемнің мағынасын түсіну

Логикалық модель мағынаны талдауға мүмкіндік бермейді. Фрейм теориясы сөйлемдердің формасы (синтаксис) мен мағынасын (семантика) модельдеуге мүмкіндік береді. Сөйлемнің мағынасын талдау (автоматты аударма, мәтінде қателіктерді іздеу және т.с.с.) кезіндегі негізгі проблема бұл «тұйықталу» проблемасы. Жай сөйлемдердің қысқа тобын талдаған кезде грамматиканы талдау міндеті көбінесе сөйлемнің мағынасын талдау міндетіне сәйкес келеді. Бірақ, сөйлемдер санының өсуімен айырмасы анық көрінеді: «Мақсат Секеннің туған күніне сыйлық сатып алды, бірақ онда ондай бар деп ойлағандықтан, ол оны қайтадан дүкенге апарудан сескенеді». Секенде не бар? Нені дүкенге апару керек? Осындай «онда», «оны» деген мәтіндерді тек керек орынға қойып сұрақтарға түсінікті жауап табалмаймыз. Ол үшін «терең құрылым» қажет, сонда есімдік «онда» «Секенге» сәйкес, ал есімдік «оны» - «сыйлыққа» сәйкес болады.

«Тілдік сезіну» деп аталатын қабілетсіз (яғни, жаңа сөздердің мағынасын олардың дыбыстау немесе сөйлем құрамы бойынша болжап білу) шет тілді оқып білу мүмкін емес. Кейбір адамдар, мысалы аудармашы, сөзді бірнеше сөйлемнен алдын ала (мағынасы бойынша) таба алады. Автоматты аударма үшін көп деңгейлі фреймдер желісін пайдаланады, олар беттік, синтаксистік, семантикалық, тематикалық және хабарлы фреймдерден тұрады.

Сөзді тыңдау және түсіну міндеттері өзінің табиғи негізінде әртүрлі. Бірінші жағдайда кейбір фреймның терминалдарының пассивты толтыруы, екінші жағдайда, мәнін түсінуі (келісу – активті форма) болады. Тұйықталуы түрліше келеді. Активті түрде кейбір сценарий ретінде жүргізіледі (6.1-кесте).

6.1-кесте

Фрейм сценарийі

Сценарий: мейрамханаға бару	Рольдер: келуші, официант, шеф, кассир Мақсат: тамақты алу
I сахна. Кіру	Бос үстелі бар орынға көз салу Қайда отыруды таңдау Үстелге барып отыру
II сахна. Тапсырыс	Менюні алу Менюні оқу Нені алуды шешу Официантқа тапсырыс беру
III сахна. Тамақ	Тамақты алу Тамақты жеу
IV сахна. Шығу	Есепті сұрау, чекті алу Кассирге бару Ақшаны төлеу Мейрамханадан шығу

Келтірілген сценарий бойынша «мейрамханаға бару» графтың түрін жасауға қиын емес.

Әрбір сценарийде іс-әрекеттерді орындау құралдар жағдайына байланысты өзгеру мүмкін. Мысалы, II сахнада тапсырысты ауызша, жазбаша немесе қол сермеу көмегімен де (егер тілді білмесеңіз) істеуге болады. IV сахнада ақшаны кассирге, официантка немесе «Менің есебіме жазыңыз» деген сөзбен де төлеуге болады.

Және де іс-әрекеттердің тізбегі бұзылу мүмкін. Осындай бұзылудың ең азы үш жағдайы кездесу мүмкін. Бірінші – іс-әрекеттердің тізбегінің басқа сценариймен үзілуі. Басқа екеуі кедергі және қателік деп аталады. Осы кедергі мен қателіктер әрбір элементар іс-әрекеттерден кейін пайда болу мүмкін, сондықтан сценарийге «егер былай болса...» сияқты сұрақтар енгізіледі. Жағымды жауапты тапқанда кедергі мен қателіктерді жоятын жаңа іс-әрекеттер сценарийге енгізіледі. Мысалы, II сахнада, егер официант келушіні байқамай жүрсе, онда ол оны шақырады немесе оған көз айырмай қарайды.

Сонымен сценарий бұл жай оқиғалар тізбегі емес, оны каузалды (ағылшын сөзінен *cause* — себеп) іс-әрекеттер тізбегі деп айтуға болады. Осындай сценарий ықтимал болатын жолдарға тармақталу мүмкін, олар ерекше сипатты нүктелерде (элементар іс-әрекеттер) үйлеседі. Мейрамхана сценарийі үшін «тамақ ішу» және «ақша төлеу» осындай іс-әрекеттер. Сценарийді қашан пайдалануға болатынын білу үшін тақырыптар қажет. Бұл тақырыптар берілген сценарийге көңіл аудару үшін күй-жайларды анықтайды.

6.3. Фрейм моделін тәжірибелік жүзеге асыру

Тәжірибеде фреймдерге негізделген модель объекті-бағытталған бағдарламауда және объекті деректер қоры теориясында ең кең тараған. Қазіргі заманның бағдарламау тілдерде пайдалынатын фреймдердің негізгі қасиеттері: инкапсуляция, мұралану және объектілердің полиморфизмы.

Инкапсуляция — объект ішіндегі мәліметтер мен әдістердің бірлігі.

Мұралану — объектінің өзінің бір ата-тегінің ішіндегі анықталған мәліметтер мен әдістермен пайдалану қабілеті.

Полиморфизм — түрлі уақытта объектінің өзін-өзі әртүрлі ұстау (не өзінің типі сияқты, не қай-қайсысы ата-тегінің типі сияқты) қабілеті.

Білімдер қорында сақталынатын үлгі-фреймдерді (түп тұлғалар, абстракты кластар) және нақты жағдайды бейнелеу үшін жасалынатын дана-фреймдерді (объектілер) ажыратады. Керек фреймды оңай табу үшін оларды типтерге бөледі:

фрейм-құрылымдар (сканер матрицасы, үй құрылысы);

фрейм-рольдер (оператор, сатып алушы, директор);

фрейм-сценарийлер (мейрамханаға бару, туған күні).

Фреймдерді кесте түрінде жазады (6.2-, 6.3-, 6.4- кестелерді қараңыз).

6.2-кесте

Фрейм аты: абстракты техникалық объект			
Слот аты	Слот мәні	Қосылған процедура	Слот типі
Объект типі	Техникалық	Жоқ	АКО
Жағдай	$X = 20; Y = 20; Z = 50$	Қозғалу	Сол сияқты
Міндет	Кеңістікте массаны алмасу	Жоқ	Сол сияқты

6.3-кесте

Фрейм аты: абстракты автомобиль			
Слот аты	Слот мәні	Қосылған процедура	Слот типі
Ата-тегі	Абстракты техникалық объект	Жоқ	АКО
Жағдай	$X = 20; Y = 20; Z = 50$	Жүру	Сол сияқты
Міндет	Кеңістікте массаны алмасу	Жоқ	Сол сияқты
Жылдамдықты шектеу	120 км/сағ	Жоқ	Сол сияқты

6.4-кесте

Фрейм аты: абстракты жеңіл автомобиль			
Слот аты	Слот мәні	Қосылған процедура	Слот типі
Ата-тегі	Абстрактный автомобиль	Жоқ	АКО
Жағдай	$X = 20; Y = 20; Z = 50$	Жүру	Сол сияқты
Міндет	Жолаушыларды алмасу	Тиеу	Сол сияқты
Жылдамдықты шектеу	120 км/сағ	Жоқ	Сол сияқты

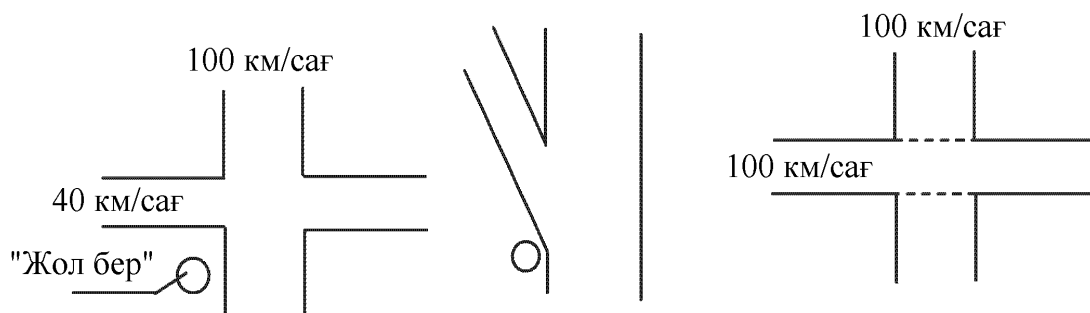
Фрейм желісінде «абстракты жеңіл автомобиль» ұғымы иерархияның жоғарғы деңгейінде тұратын «абстракты техникалық объект» пен «абстракты автомобиль» фреймдердің мұрагерлік қасиеттерін қабылдайды. «Кеңістікте автомобиль қозғала ала ма?» деген сұраққа жауабы: «Иә», өйткені бұл ата-тек класының қасиеті.

Интеллектуалды есептерді шешу үшін объекті бағдарламаудың қолдану саласы өте кең. Белгілі шығындарды жұмсап, модельді қандай да интеллектуалды есепті шешуге арнауға болады. Модельдің тиімді қолдануының ең айқын саласы – күрделі детерминды емес процестер, яғни өте көп түрлі компоненттердің, объектілердің, шарттардың өзара әрекеттесуіне тәуелді шығатындар. Осындай жағдайлардың фреймды моделі негізінде модельдеу ерекшелігі мынадай – процесті дәйекті (егер өнімдер ережелерді пайдалансақ) жүргізбей, «нақты уақытта жүргізуге» болады.

Мысал ретінде қауіпсіз жол қиылысты жобалау туралы есепті қарастырайық: «Қаланың сыртында орналасқан екі жол қиылысының геометриясы мен өлшемін жобалау қажет, мұнда автокөліктердің жылда ең

ықтимал қақтығысу саны берілгеннен аспау керек. Қиылысатын жолдың параметрлері – ені, жабыны, рұқсатты және нақты жылдамдық тәртібі, автокөліктер ағыны туралы мәліметтер және басқалар - белгілі».

Инженер-жобаушылар қиылыстың үш вариантын ұсынды (6.10-сурет). Бірінші жағдайда классикалық қиылыс салынады. Қозғалыс қауіпсіздігі жүргізушінің өз қолында. Жылдамдық тәртібін және «Жол бер» белгіні бұзбағанда, қақтығыс тек техниканың ақаулығы немесе ойланбаған кедергілер (мысалы, жолдағы сиырлар) себебінен болу мүмкін. Априори жобалаушылар осы шешімнің қауіптілігін «қанағаттанарлық» деп бағалады.



6.10-сурет. Жол қиылысының варианттары

Екінші жағдайда жанасу бұрышты дұрыс таңдау және үдеу жолы болу арқылы жүргізушілер керекті жағдайда «тура қақтығыстан кете» алады. Онан әрі трасса бойынша сақиналы шешілу салыну мүмкін. Жобаушылар оның да қауіпсіздік дәрежесіне «қанағаттанарлық» баға қойды, ал шешімнің құны «жоғары».

Үшінші вариант бір жолдың үстінен көпір салуды болжайды. Жобалаушылар соңғы вариантты «ең қауіпсіз, бірақ қымбат» деп бағалады және тәуір көрінетін деп есептеді.

Компьютерде модельдік сынау келесі қадамдардан тұрады:

1. БҚ-нан керекті фрейм-үлгілерді таңдау: «сиыр», «ат», «тойота», «мерседес», «опель» және т.б.

2. Қиылыстың берілген геометриясына көрсетілген фреймдер салынды, мына шарттарға сәйкес:

2.1. қозғалыс тәртібін бұзбау / бұзу;

2.2. ауа райы жайлы / жайлы емес;

2.3. транспорт құралдардың техникалық күйі жақсы / нашар.

3. Фреймдерге «модельдеуді бастау» деген хабарлама жіберіледі. Әрбір фрейм кейбір амал ретінде істей бастайды: көліктер кейбір жылдамдықпен қозғалады, сиырлар жолдан кенет жүгіріп өте бастайды. Ұзақ модельдеуден кейін жағдай белгілі бір тұрақты күйге келеді. Соңғы кезеңде модельден статикалық сипаттамаларын түсіріп алады: тәулік бойы апаттар саны, нақты өткізу қабілеті және т.б.

Келтірілген статикалық сынау көрсетті – жол тәртібі ережелерін бұзбаса және техникалық күйі өте жақсы болса, қауіпсіздіктің ең жоғары сипаттамалары 3 вариантта болады.

Жол тәртібі ережелерін бұзбаса және техникалық күйі нашар болса, қауіпсіздіктің ең жоғары сипаттамалары 1 вариантта болады.

2 варианттың (1 вариантпен салыстырғанда) жоғары өткізу қабілеті бар, бірақ қозғалыс қауіпсіздігі жылдамдыққа қатты тәуелді, 60 км/сағ шектен асқанда 2 вариант (1 вариантпен салыстырғанда) көп апаттарға келтіреді.

Берілген жерде автокөліктің негізгі паркінің салынған кезеңі 8 жылдан асқанын еске алатын болсақ, онда салуға 2 вариантты ұсыну керек.

Фреймды модельдің бір қатар сөзсіз артықшылықтары бар. Біріншіден, бұл теориялық жақсы негізделген тереңдік модель. Оның тәжірибелік жүзеге асыруы — микроәлемдерді құру. Осыған байланысты фреймды модель жүйенің өлшемдігі бойынша қолдану шегін жылжыта алады.

Фреймды модель нақты әлемнің объектілер иерархиясын жақсы жүйелендіре алады. Модель көп бағдарламау тілдерде және білімдерді ұсыну арнайы тілдерде (FRL — Frame Representation Language) жүзеге асырылған, бір қатар танымалы сараптамалық жүйелерде ойдағыдай пайдаланылған: ANALYST, МОДИС және т.б.

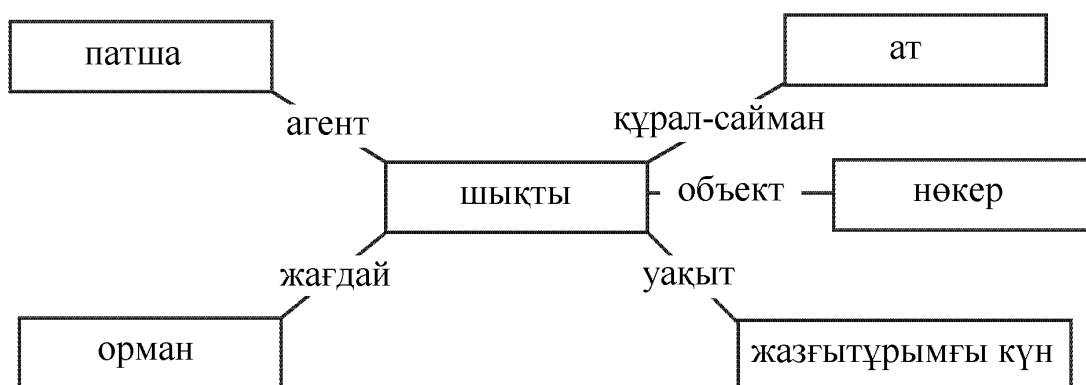
Модельдің кемшіліктеріне: екі объект арасында үлкен деректермен алмасуда пайда болатын кейбір қиыншылықтар жатады. Айта кетейік, М. Мински суреттеген классикалық фреймде ешқандай алмасу болмайды.

7 - БӨЛІМ

БІЛІМДЕРДІ ҰСЫНУ ҮШІН СЕМАНТИКАЛЫҚ ЖЕЛІЛЕР

«Семантикалық (мағыналық) желі» деген ұғым 1968-1969 жылдары Р.Куиллиан (Ross Quillian) жұмыстарында пайда болды. Бұл үшін доғалармен (графтар) қосылған түйіндері бар графикалық сызбалар пайдаланылады. Семантикалық желілер теориясында граф-ағаштар мен қатар, циклдық және толық байланысқан графтар қарастырылады. Одан басқа, мүмкін болатын арақатынастар типтері жеке және ерекше түрде зерттелінеді (дәстүрлі: АКО — a kind of, ISA, has part және кейбір жаңалар).

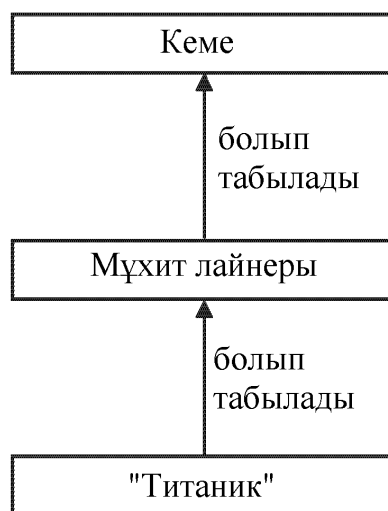
Алдымен семантикалық желілер табиғи тілдерді талдау үшін және адамның ақыл-есінің психологиялық моделін құру үшін жасалынған болатын. Бұл кезеңде сөйлемде бейнебір «орталық тақырып» бар деп есептелетін, осыны пайдаланып машина сөйлемнің мағынасын (семантикасын) «түсіну» мүмкін (7.1-сурет).



7.1-сурет. «Жазғытұрымғы күні патша нөкерін ертіп, орманға атпен серуендеуге шықты»

Семантика – белгілер мен объектілер арасындағы қатынастарды қоятын ғылым, яғни белгілердің мағынасын анықтайтын ғылым. *Семантикалық желі* – бұл бағытталған граф, оның шыңдары – ұғымдар, ал доғалары – олардың арасындағы қатынастар.

Адамның ақыл-есінің құрылысын және логикалық шығарудың ұйымдастыруының теориялық қағидаларын иерархиялық желіге негіздеп, Р.Куиллиан А.Колинзбен (Allan Collins) бірге 1969-1970 жылдары тәжірибелік түрде тексеріп шығарды. Осындай желі үшін мынадай қатынастар сипатты: класс – класс элементі; қасиет – мәні. Егер «кеме», «мұхит лайнеры» және басқа түйіндерде ішкі құрылысы жоқ деп есептесек, онда 7.2-суреттегі желіні иерархиялық емес, қарапайым желі деп атауға болады. Класс элементінің мысалдары 7.2 - 7.4-суреттерде көрсетілген.



7.2-сурет. «Болып табылады» деген қатынасты пайдаланатын иерархиялық желі

Модельдің мұнан былай дамуы Р.Симмонс (Robert Simmons), С.Шапиро (Stuart C. Shapiro), Хендрикс (Hendrix) және Р.Шенк (Roger C. Schank) аттарымен байланысқан.

Мысалы, Р.Симмонс желілерінде қатал иерархия жоқ, өйткені ол басқа есепті шешуге тырысқан: етістіктерді «өрістету/толықтыру» арқылы сөйлемнің мәнін түсіну. Яғни етістіктен өзіндік «қосымша желіні» (берілген етістікті «пайдалану варианттары» бар желі) құрастыру арқылы. Осыған сәйкес түйіндер жіктеуі де табылған: Агент, Реципиент (receive – алу), Объект.

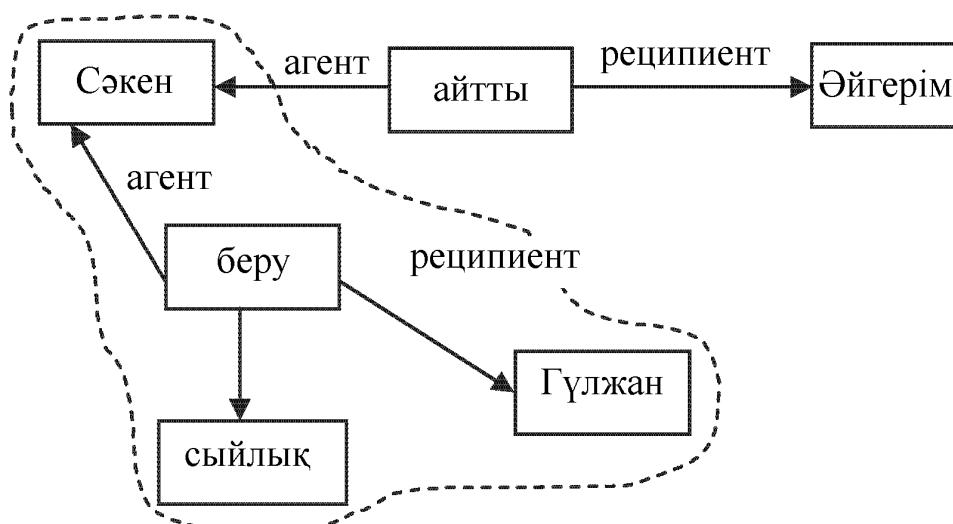
Беру

Агент: (Яго, бала, Отелло)

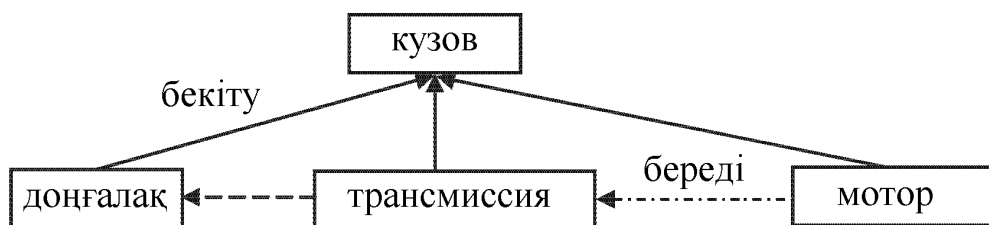
Реципиент: (Дездемона, Родриго)

Объект: (алма, гүл, орамал)

Семантикалық желідегі түйіндер әдетте, объектілерге, тұжырымдамаларға, оқиғаларға немесе ұғымдарға сай болады. Доғалар, білімдерді ұсыну әдісіне тәуелді әртүрлі әдістер арқылы анықталу мүмкін. Желінің кезкелген фрагменті, мысалы, бір шыңысы, екі шыңысы және оларды қосатын доға, *бағыныңқы желі* деп аталады (7.3-, 7.4-суреттер). Семантикалық желідегі логикалық шығару (шешімді іздестіру) - бұл кейбір шарттарға қанағаттандырылған бағыныңқы желіні табу немесе құрастыру. Осы процесті формалдау үшін семантикалық желілердің типтеуін енгізеді, сосын нақты түрлі желілер үшін (желіге кіретін қатынастырдың математикалық қасиеттерін талдау негізінде) шешу әдістерін әзірлейді.



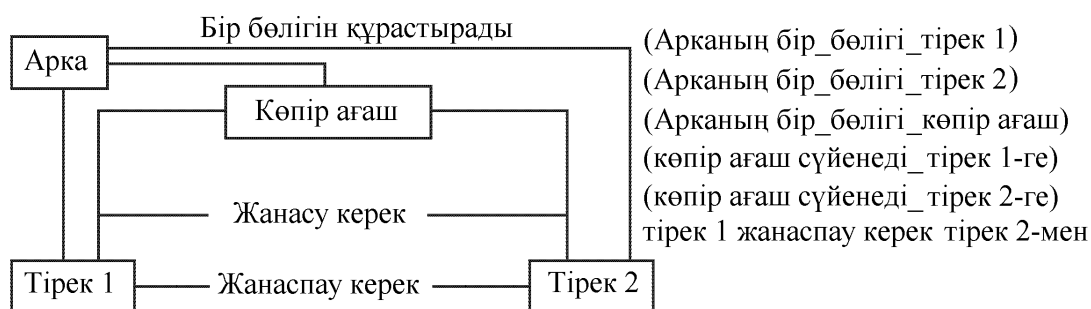
7.3-сурет. Күрделі сөйлемді: «Гүлжанға сыйлығын бергенін, Сәкен Әйгерімге айтты» семантикалық желі түрінде көрсету; бағыныңқы желі «Сәкен Гүлжанға сыйлық берді»



7.4-сурет. Семантикалық желі «автомобиль» және бағыныңқы желі «мотор доңғалақтарға күш береді»

Шыңдар мен доғаларға қойылатын шектеулерге тәуелді желілердің әртүрлі типтері пайда болады. Егер желі шыңдарының ішкі құрылымы болмаса, онда осындай желілер *қарапайым желілер* деп аталады. Егер шыңдардың ішкі құрылымы бар болса, онда желі *иерархиялық* деп аталады. Динамикалық семантикалық желілер (сценарийлер) – оқиғалары бар желілер. Семантикалық желінің қарапайым желінен негізгі айырмашылығы – желіні бағыныңқы желілерге бөлу және шыңдар арасында ғана емес, бағыныңқы желілер арасында да қатынастарды анықтау мүмкіндігі. Желіде болатын әртүрлі бағыныңқы желілер, бағыныңқы желілердің ағашы түрінде реттелу мүмкін, оның шыңдары - бағыныңқы желі, ал доғалары – көріну қатынастары. Бағыныңқы желі ұғымы математикадағы жақшалар ұғымына ұқсас, көріну ұғымы – жақшаның ішінде және сыртында болатын айнымалылар ұғымына ұқсас.

Семантикалық желі көмегімен «Арка» деген техникалық объектіні сипаттайық – бір бірімен жанаспайтын екі тіректерге сүйенетін көпір ағаш (7.5-сурет). Формалды түрде бұл желіні былайша бейнелеуге болады:



7.5-сурет. «Арка» ұғымының семантикалық желісі

Сонымен, желінің сипаттамасының өнімдік моделіндегі ағаштың сипаттауынан ешқандай айырмашылығы жоқ. Желі мен ағаштарды бейнелеу үшін бірдей машиналық тілдер пайдаланады. Бірақ, желілерде шығаруы сөзсіз күрделі, себебі көп қатынастар типтері әлі зерттелмеген және формалды алгоритмы олар үшін жоқ.

Желі үстінде логикалық шығаруға жету өте қиын, өйткені заттық саланың кеңістіктері жиі метрлік емес болады, яғни келесі аксиомалардың біреуіне қанағаттандырылмаған:

1. $\rho(a, b) = 0$ тек $a = b$ болғанда,
2. $\rho(a, b) = \rho(b, a)$,
3. $\rho(a, b) < \rho(a, c) + \rho(b, c)$,

мұндағы ρ - өлшем (оңай түсіну үшін, өлшем элементтер арасындағы арақашықтыққа ұқсас деп алуға болады).

Семантикалық желілердің мүмкін болатын өте пайдалы қасиеті - *транзитивтілік*. Транзитивтілік: әрбір $a, b \in R$ үшін (мұнда $a < b$ және $b < c$) $a < c$ арақатынас әділетті. Транзитивтілікті бұзатын мысал: A - B -ның ұлы, C - B -ның ұлы, бірақ C - A -ның ұлы емес.

Мысал. «Куин Мери мұхит лайнеры болып табылады» және «Әрбір мұхит лайнеры кеме болып табылады» деген сөйлемдерді қарастырайық (7.2-сурет). «Болып табылады» доғалар пайдалыналады. Түйіндерді байланыстыратын доғалардың қасиеттерін білгендіктен («болып табылады» қатынас ISA — транзитивті), желіден үшінші мақұлдау шығаруға болады: «Куин Мери кеме болып табылады»,

«Болып табылады» және басқа («бөлігі бар» сияқты) қатынастар желіде мұралану иерархия қасиетін орнатады. Бұл желідегі төменгі деңгейдің элементтері жоғарғы деңгейдің элементтерінің қасиеттерін мұралану мүмкіндігін көрсетеді. Сондықтан жад үнемделеді, себебі ұқсас түйіндер туралы ақпаратты әрбір түйінде қайталау қажетсіз. Оның орнына ол желінің бір орталық түйінде орналасу мүмкін. Мысалы, кемеңі ұсынатын семантикалық желі үшін, оның мынадай қозғалтқыш қондырғысы, корпусы, бу қазандары орны сияқты бөліктері иерархияның төменгі деңгейінде қайталанбай, бір рет қана кеме деңгейінде қосылған. Бұл, жүздеген кемемен және олардың бөліктерімен жұмыс істегенде, жадтың орасан зор көлемін үнемдейді. Сосын, желінің доғаларымен белгіленген қатынастар мағынасы

туралы білімдерді пайдаланып, желі бойынша іздестіруді жүзеге асыруға болады. Мысалы, «Куин Мериде бу қазандары орны бар» деректі орнату үшін.

Семантикалық желілердің бірнеше топтамасын енгізуге болады. Мысалы, қатынас типтерінің саны бойынша:

- біртекті (бір ғана қатынас типі бар);
- біртекті емес (әртүрлі қатынас типтері бар).

Қатынас типтері бойынша:

- бинарлы (қатынастар екі объекті байланыстырады);
- *n*-арлы (екі ұғымдардан артық байланыстыратын арнайы қатынастар, мысалы «үшбұрыш»).

Семантикалық желілерде ең жиі келесі қатынастар пайдалынады:

- «бөлік - бүтін» типті байланыстар («класс – бағыныңқы класс», «элемент – көптік» және т.с.с.);
- функционалды байланыстар (әдетте «жасайды», «ықпал етеді»,... етістіктермен анықталатын);
- сандық (артық, кем, тең,...);
- кеңістік (алыс, жақын, артында, астында, үстінде,...);
- уақытша (бұрын, кейін, барысында,...);
- атрибутивті байланыстар (қасиеті бар, мәні бар,...);
- логикалық байланыстар (және, немесе, жоқ) және басқалар.

Семантикалық желілер пәндік сала туралы білімді ұсынудың ең сәтті әдісі болып табылады. Күрделі грамматикалық сөйлемдерді ұсыну үшін табиғи тіл бойынша ғылыми жұмыстарда пайдалынады.

Модельдің тәжірибелік қосымшаларын қарастырайық:

1. Патрик Уинстон (Patrick Winston) кейбір объектінің «концепін» автоматты жасау үшін программа жасады (7.6-сурет). Ол «концепт» идеясың арқа сияқты физикалық құрылымды бейнелеу мысалында көрсетті (7.5-сурет). «Концепт» шынында «фрейм» ұғымына (М. Мински енгізген) өте жақын, бірақ ахуалдың өзгеру динамикасы (хабарлама мен алмасу) еске алынбайды.

2. Джейм Карбонел (Jaime Carbonell) Оңтүстік Американың географиясың зерттеуге арналған SCHOLAR оқыту-сараптамалық жүйесін жасады. Осындай жүйелерден кейін қазіргі географиялық ақпаратты жүйелер шықты.

3. Хайдерн PLNLP (Programming Language for Natural Language Processing) жасады – табиғи тілді өңдеу үшін бағдарламау тілі (екі сөйлемнің ұқсастығын талдау есебі, 7.3-сурет; дербес жағдай – бір сөйлем басқа сөйлемнің бөлігі болып табылады). PLNLP келесі ережелер бойынша жұмыс істейді:

- декодтау ережелер көмегімен сызықты тіл тізбегінің синтаксистік талдауы жүргізіледі және желі салынады;
- кодтау ережелер көмегімен желі сканерленеді, тіл тізбегі немесе өзгертілген басқа желі тудырады.

өзге түрі уақыт	жалпы	ерекше
тұрақты	класс ↑ енгізеді өзге түрі	түрінің мәні ← объект ↑ тарту бөлігі бар
уақытша	қасиет ↑ жасайды включает в себя	қасиеті бар ← ахуал ↑ жасайды бөлігі бар
өтпелі	өзгеріс ↑ себептілік	түрінің мәні ← оқиға ↑ алдында болу

7.6-сурет. Үш күйі бар екі өлшемді концепт (мүмкін болатын вариант)

4. Арнайы NET желілі тіл жасалынды.

5. Білімдерді ұсыну тілі ретінде семантикалық желілерді пайдалынатын кең белгілі сараптамалық жүйелер жасалынды - PROSPECTOR, CASNET, TORUS.

Модельдің негізгі кемшілігі – шығаруды іздестіру күрделілігі; семантикалық желі үстінде жақсы математикалық модельдер тек бірнеше қатынастар типтері үшін құрастырылған.

8 - БӨЛІМ

БІЛІМДЕРДІ ҰСЫНУ ЖАҢА МОДЕЛЬДЕРІ

Білімдерді ұсыну негізгі модельдер, бағалы есептер тізімі және оларды шешу әдістер өткен ғасырдың 80 жылдары ұйымдастырылды.

90 жылдардың басында жасанды интеллектің есептері мен әдістерінің философиялық түсінуі біраз өзгерді, бұл бірден жаңа модельдердің пайда болуына әкелді: іздестіруді сипаттау критериалды тілі, нейрон желі, ықтималдық модельдер. Бірақ, айта кетейік, кейбір мамандар оларды толық модель деп санамайды, себебі бұл модельдер өзінің қорытындысын негіздей алмайды. Мұнда логикалық пен өнімдік модельде болатын және білімдерді ұсыну, жаңа модельде болмайтын бектрекинг механизмы еске алынады. Сараптамалық жүйеде өзінің қорытындысын түсіндіретін бағыныңқы жүйесі болу қажет. Сондай-ақ, критериалды тілдің пайда болуына кем дегенде, жүздеген жыл бар, сондықтан оларды жаңа деп айту дұрыс емес.

Сонымен, «білімдерді ұсыну үшін модельге» жаңа көзқарас – бұл «жақсы шешімдерді» тудыруға мүмкіндік беретін кейбір модель. Және ол адамның ойлау әдісін бе, не өзінің қорытындыларын түсіндіре ме, аса маңызды емес.

8.1. Критериалды әдістер

Тәжірибеде интеллектуалды, айқын емес модельдердің қолдануы баламаны бағалау және ең жақсысын таңдау үшін қажет. Олардың қолдану қажеттілігі баламаның көп санына, еске алатын параметрлеріне себепті; мысалы, «қолайлы, әдеттегі және ең нашар».

Бұл жағдайда барлық «иә» мен «қарсы» еске алынып немесе кездейсоқ түрде (мысалы, аналогия арқылы), таңдау ұғынып жүзеге асырылу мүмкін. Айта кетейік, үлкен жүйелер үшін сарапшы ұсынып беретін логикалық негіздемелер кездейсоқ таңдаудан аса сенімді емес, ал кейбір жағдайда сенімі төмен болады. Бұл алдымен көп фактілерді, себептерді, мақсаттарды адам жадында сақтай алмайтынына байланысты, ал қазіргі есептер (мысалы, экономикалық) осыны талап етеді.

Мысалы, банк қызметкері тұрақсыз жағдайдағы кәсіпорынды несиелеу туралы шешім қабылдайды. Егер несие берілмесе және кәсіпорын өз күшімен қиыншылықтан шықса, онда банк пайда алалмайды. Егер де кәсіпорын ойсырап қалса, банк шығындарға ұшырайды. Егер несие үлкен кепілге берілсе, кәсіпорын алудан бас тартады. Сонымен қызметкердің интуициясы және бухгалтерлік құжаттары шешімді қабылдауға жеткілікті ме?

Сол сияқты есептерді сипаттау және шешу үшін арнайы модель жасалынған – таңдауды бейнелеу тілдері. Таңдау есептер әр алуанды және оларды шешу әдістері де әртүрлі. Алдымен барлық таңдау есептер үшін жалпы ұғымдарды енгізейік.

8.1.1. Таңдау есебі

ЖИ қолданылатын әдістердің ең ескісі критериалды баға беру, дәрежелеу және кейінгі таңдау әдісі.

Ең қарапайым қойылуда бұл әдіс екі баламаны (бинарлы қатынас) салыстыруға ұсынады. Мысалы, $3 < 5$, $6 \in \{3,5,6\}$ және т.б. осындай қатынастар Аристотельге дейін белгілі болатын. Біздің жағдайымызда салыстыру бір параметр (критерий) бойынша жүргізіледі. Келесі мысалда одан күрделі жағдайды көрсетейік.

Берілген: балама $A(3,3) = A(a_1, a_2)$ және балама $B(5,5) = B(b_1, b_2)$. Оларды байланыстыратын R қатынасты анықтаңыз, яғни $A R B$.

Егер дәстүрлі көзқарасты ұстасақ, онда $B > A$ өйткені $b_1 > a_1$ және $b_2 > a_2$. Бірақ, мұндай шешім даусыз емес, себебі a_1, a_2, b_1, b_2 деген не және $\{a_1, b_1\}$ мен $\{a_2, b_2\}$ көптіктерде қатал ретті қатынастарды («<», «>») қоюға балама айқын емес. A мен B баламалар мәні «Омаров» және «Оспанов» сәйкес болсын. $\{a_1, b_1\}$ және $\{a_2, b_2\}$ параметрлер – 1 және 2 семестрдегі қанағаттанарлықсыз бағалар саны. Осылардың қайсысы үздік студент? Әрине, $A = B$ — онда екеуі де нашар оқиды. Егер де параметрдің мағынасын «күрделі сызбадағы қателіктер саны» деп санасақ, онда анық $A > B$ (A B -дан жақсы). Көрініп тұрғандай, шешім көзқарасқа («өлшеудің нолі») және R қатынастың түріне тәуелді.

Күрделі жағдай – екіден артық баламаны салыстыру (екі екіден), немесе екіден артық аргументтерді байланыстыратын қатынасты анықтау. Бұл жағдайда R қатынасы тернарды болу мүмкін (мысалы, R – үшбұрыш a, b, c қабырғаларды байланыстырады) немесе n -арлы (отбасы). n саны шекті, есептеулі немесе континуалды болу мүмкін. Мысалы, Сізге «демалыс үшін турбаза таңдауыңыз» керек, мұндағы критерийлер «комфорт деңгейі» және «баға». Егер Сіз таңдау үшін Алатау базалар анықтамалығын пайдалансаңыз, балама көптігі шектеулі болады. Егер де «жабайы туризмды» ұнатсаңыз, ал «база» ретінде «кемпинг үшін орындарды» қарастырсаңыз – онда континуалды (яғни $[0..1]$ кесіндіге сәйкес сандар көптігі) болады. Шынында, екі түрлі кемпинг арасында ғажап орын әрқашан табуға болады. Және де әрбір кемпингты тым бағалаудың қажеті жоқ, Сізге кездескен қандай болса да, ертең одан күштісі кездесу мүмкін. Кездейсоқ таңдау (кездейсоқ шаманың кейбір үлестіру заңы) – баламалар континуалды көптігі үшін әбден есті шешім.

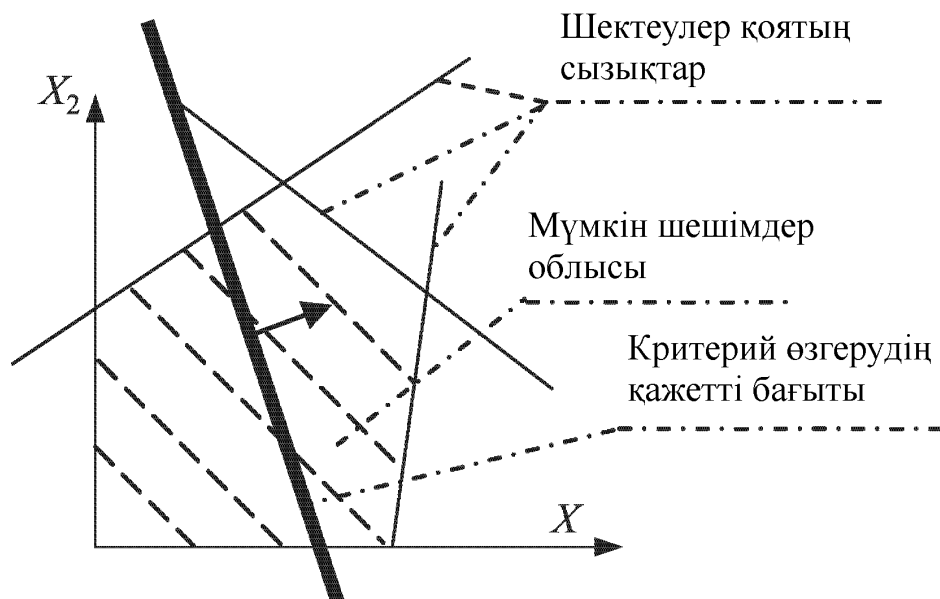
Сонымен, критериалды таңдау есебі сипатталады: көп өлшемді параметрлер кеңістігінің бар болуымен; осы кеңістікте («идеал нүктелер» немесе нүктене беру мүмкін болмаса «дәмелену деңгейлер») қажетті салаларды бейнелеуі мен; таңдалған баламаның қажетті салаға жақындығын анықтайтын өлшеммен (таңдау критерийі); шекаралық мәндер жиынтығымен (яғни баламалар шынында да әртүрлі не көрсете алатын шекті айырмашылықтар); және де күйлер кеңістігінде ең пайдалы қозғалыс

бағытты анықтау үшін қолданылатын математикалық әдістермен (8.1-, 8.2-, 8.3-суреттер).

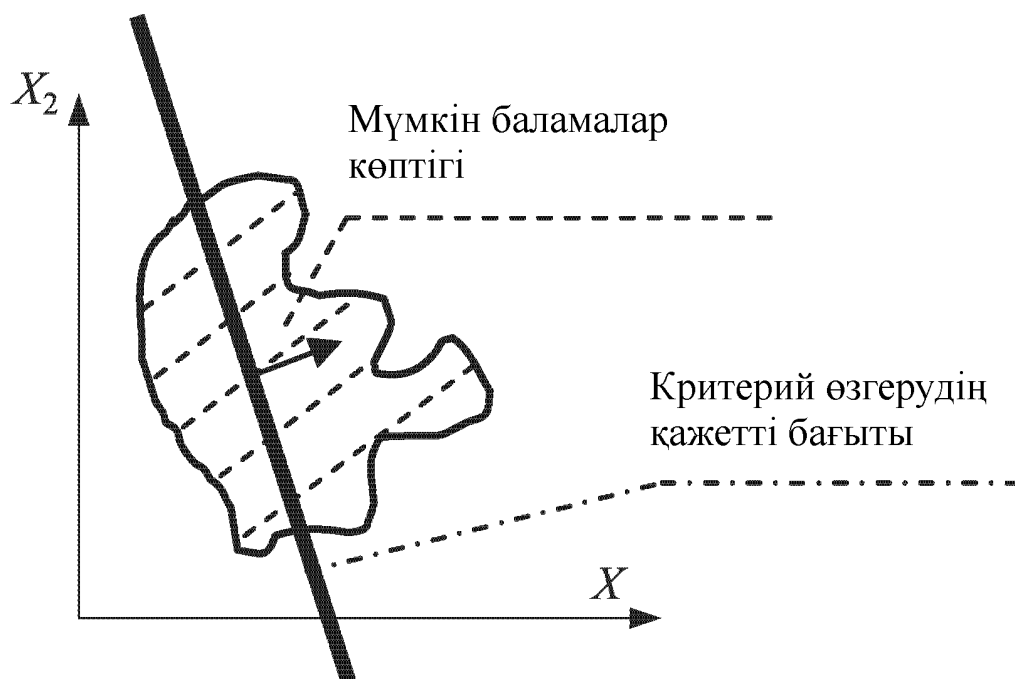
Жоғарыда келтіргенді басқа сөзбен айтқанда: n тәуелсіз айнымалылардан кейбір функция $f(x_1, x_2, \dots, x_n)$ бар. 8.1-кестеде келтірілген шектеулердің біреуі болғанда $\max \{f(x_1, x_2, \dots, x_n)\}$ табу керек. Немесе $\min$, егер «қашықтау өлшемі» берілсе.

Мұнда барлық индекстер бүтін сандар 1, 2, 3 ...

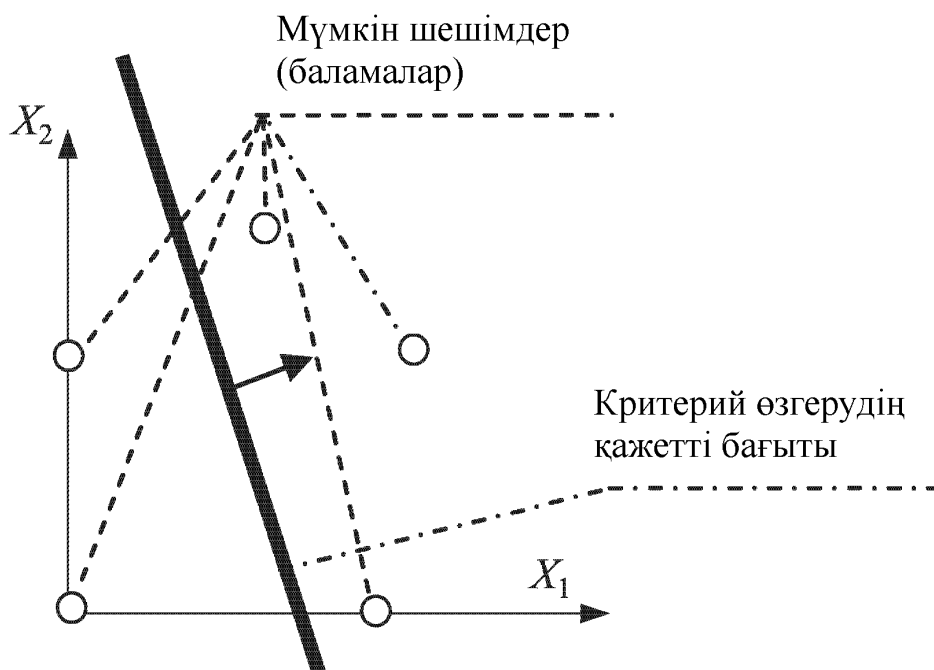
Осыны кейбір дербес геометриялық мысалдармен екі параметрлер x_1 және x_2 көптігінде де көрнекті көрсетуге болады (8.1-, 8.2-, 8.3- суреттер).



8.1-сурет. Үздіксіз сызықты шектеулер көптігіндегі сызықтық критерий



8.2-сурет. Сызықты емес шектелген баламалардың дөңес емес үздіксіз көптігіндегі сызықтық критерий



8.3-сурет. Дискретті баламалар көптігіндегі сызықтық критерий

8.1-кесте

Критериалды есептердегі шектеулер түрлері

x_i параметрлер мәндерінің үздіксіз көптігінде сызықтық шектеулер	x_i параметрлер мәндерінің үздіксіз көптігінде сызықтық емес шектеулер	x_i параметрлер мәндерінің дискретті көптігінде сызықтық не сызықтық емес шектеулер
$C_0 \geq X_1 \geq C_1$ $C_2 \geq X_2 \geq C_3$ $\dots$ $C_{n+2} \geq X_n \geq C_{n+3}$ $a_{1m}x_1 + a_{2m}x_2 + \dots + a_{nm}x_n \leq b_1$ $\dots$ $a_{1m}x_1 + a_{2m}x_2 + \dots + a_{nm}x_n \leq b_m$	$C_0 \geq X_1 \geq C_1$ $C_2 \geq X_2 \geq C_3$ $\dots$ $C_{n+2} \geq X_n \geq C_{n+3}$ $a_{1m}x_1^{k1} + a_{2m}x_2^{k2} + \dots + a_{nm}x_n^{kn} \leq b_1$ $\dots$ $a_{1m}x_1^{k1} + a_{2m}x_2^{k2} + \dots + a_{nm}x_n^{kn} \leq b_m$	$X_1 \in \{x_{11}, x_{12}, \dots, x_{1s}\}$ $\dots$ $X_n \in \{x_{n1}, x_{n2}, \dots, x_{nd}\}$ $\dots$ Сызықтық немесе сызықтық емес шектеулер

Шешімді критериалды қабылдау есептер сызықтық немесе сызықтық емес математикалық бағдарламауға жататыны анық көрініп тұр. Қай есеп болмасын дербес жағдайы, мысалы, бүтін санды бағдарламау есебі болып шығу мүмкін. Онда ең жақсы баламаны табу керек, оның x_1 және x_2 бағалары бүтін сандар.

Жалпы жағдайда критериалды таңдау (шешім қабылдау) әдісі келесі қадамдардан тұрады:

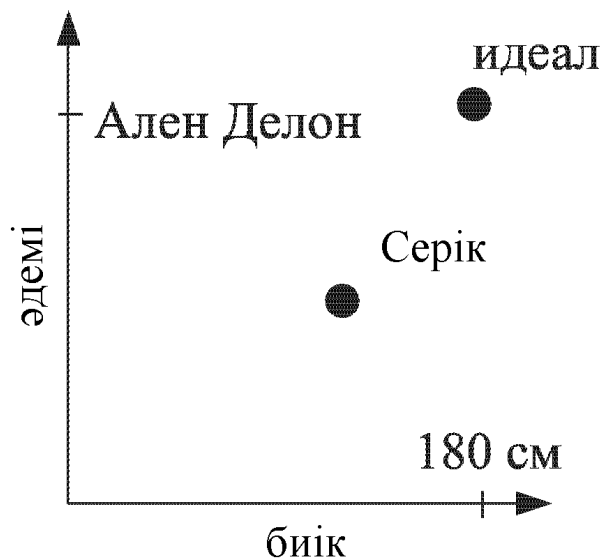
1. Өте жалпы, концептуалды есепті қою: «Не істеу керек?» немесе «Аяғында не керек және қазір не бар?». Яғни шешім қабылдаудың мақсатын (мақсаттарын) анықтау.

2. Баламаларды генерациялау. Бұл кезеңде сараптамалық деп аталатын әдістер қолданылады: «ми шабуыл», «синектика», «өнертабыс матрицалары» және т.б.

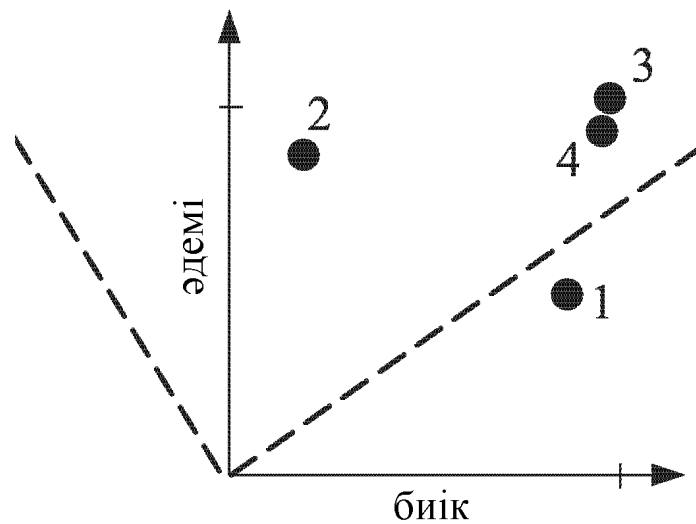
3. Мақсатқа жету дәрежесін бағалайтын критерийлерді таңдау.
 - 3.1. Көзқарасты таңдау: «идеалға ұмтылу» немесе «шығындарды минимизациялау».
 - 3.2. Әрбір критерий үшін бағалау функцияның түрін таңдау.
 - 3.3. Әрбір критерий үшін «шекаралық мәндерді» анықтау («жақсы – жаман» шкаласы).
4. Тандалған критерийлер бойынша баламаларды бағалау.
 - 4.1. Әрбір баламаның параметрлер мәндерін анықтау.
 - 4.2. Алдыңғы тарихты есепке алу (көп реттік таңдау үшін).
 - 4.3. Критерий мәнін есептеу.
5. Мақсатқа (мақсаттарға) жету дәрежесін бағалаудан алынған баламаны дәрежелілеу.
6. Ең нашар баламаларды лақтырып тастау.
7. Басым болмайтын көптіктен (мысалы, кездейсоқ түрімен) бір баламаны таңдау немесе 1-ші пунктке қайту.

Тағы да бір маңызды мәселе барлық қарастырылған функциялардың дөңестік (батыңқылық) сипаттамасы және мүмкін мәндер көптігінің тұйықтығы. Бұл анықтамалар функционалды талдауда беріледі.

Критериалды есепті шешудің қадамдарына қайтып келейік. 1 қадам бірінші бөлімдерде толық бейнеленген. Екінші пункт ұлан-байтақ, оның қарауы ЖИ курсына кірмейді. Критерийлер таңдауды (3 қадам) өнер деп айтуға болады. Барлық критериалды әдістерді (шешімді қабылдау) «идеал нүкте» және «ымыралы бағдарламау» әдістерге бөлуге болады. Бірінші жағдайда барлық баламалар өзінің белгілерінің көп өлшемді кеңістікте қарастырылады, сонымен бірге «идеал нүкте» бар болады (8.4-сурет). Әрбір баламаның бағалауы идеал нүктеге дейін аралығы деп алынады (ыңғай үшін идеал нүктені координат басына қояды). Кімде бұл аралық минималды, сол жеңеді. Кейде, керісінше, «жақындық өлшемін» емес «қашықтау өлшемін» қарастырады. Бұл жағдайда, мысалы, «абсолютты қажет емес нүктені» жасауға және оған дейін (әрбір баламадан) аралықты анықтауға болады. Әрине, ең жақсы балама «жаман» нүктеден ең алыста болады. «Жақсы - жаман», «күшті - әлсіз» және т.с.с. ұғымдарды енгізуге мүмкін беретін көп өлшемді кеңістіктерді «семантикалық метрикасы бар кеңістіктер» немесе «семантикалық кеңістіктер» деп атайды. Кейде «Осгуд кеңістігі» деп аталады, өйткені алғаш рет осындай кеңістіктер Чарльз Осгуд пен Суси жазған «Мәндерді өлшеу» кітабында қарастырылған болатын (1947 ж., АҚШ). Кітапта түсіну және бейнелеу үшін ағылшын тіліндегі құрылыстар мен психологиялық құрылымдар қарастырылған. Сондықтан лингвисттер де, психологтар да және ЖИ мамандары оны «біздің кітабіміз» деп есептейді. Ч.Осгуд бұқаралық процестерді зерттеген. Мысалы, кейбір студенттер тобында математикадан бақылау жұмысының нәтижесі бойынша Осгуд кеңістігін құруға болады.



8.4-сурет. Идеал нүкте және нақты балама (Майраның көзқарасы)



8.5-сурет. Басым болатын және салыстырмасыз баламалар

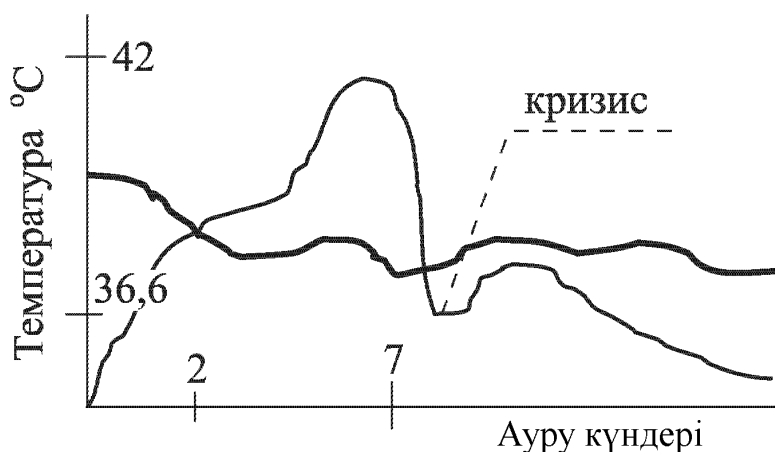
Сөйтіп, шешім қабылдайтын кейбір кеңістікті құру (8.5-сурет) мынаны ойлайды: мақсаттарды (қажетті күйді, идеал нүктені), кеңістік қасиеттерінің өзгеру бағыттары және де өлшем түрі мен шекаралық мәндерін анықтау. Шетелдік ұғымдарда мақсатқа сәйкес ұғым «goal», ал қасиеттерінің өзгеру бағытына — ұғым «objectives», бірақ олар синоним емес.

Қарапайым жағдайда өлшем ретінде жай айырма (айырмалар қосындысы) пайдалану мүмкін. Мысалы, «Серік идеал өлшемге $(180-170) + (\text{өте әдемі} - \text{сүйкімді})$ сәйкес емес». Осыдан көрініп тұр - қарапайым өлшемді енгізу түрлі өлшеуіш шкаланың бар болу проблемасына әкеледі. Шкалалар мүмкін типтер саны математикалық өлшеу теориясында жақсы зерттелген. Геометриялық көзқарастан Евклид өлшемін (аралық) қарастыру керек $\Delta = \{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 + (f - f_1)^2\}^{1/2}$. Мысал әдейі 4-өлшемді кеңістік үшін берілген, яғни дәреже көрсеткіші 2 немесе 3-өлшемді кеңістіктерде де өзгермейтінің көрсету үшін. Егер екінші дәрежелердің орнына біз p дәрежесін пайдалансақ, онда $p \rightarrow 0$ болғанда классикалық

баллдық бағаны аламыз, мұнда нақты күйдің идеал күйден әрбір ауытқуы 1 деп есептеледі; $p \rightarrow \infty$ болғанда, керісінше, кіші ауытқулар (1-ден кем) тегістеледі, ал едәуір ауытқулар өседі. Нақты есеп үшін дәрежені дұрыс таңдаудан, біз «шешімді жақсы таратып әкететін» өлшемді аламыз, яғни басым болатын баламалар (3 балама) көптігін және басым болмайтын (бір-біріне салыстырмалы) баламалар (1 және 2) көптігін жасақтауға мүмкіндік береді. Өлшем дұрыс таңдалмаса 4 баламасы бар проблеманы аламыз, ол 3 ден жаманырақ, бірақ қаншаға? 4 баламаны қабылдау үшін осы ауытқу жеткілікті ме?

Өйткенмен қандай да өлшем болғанда 1-мен 2-нің арасында таңдау керек болады. Бірақ қалай? Геометриялық бұл осьті бұру және ось үстіне нүктелерді проекциялау әдіс арқылы мүмкін. Айқын көрініп тұр 1 балама 2 ден жақсырақ және де анағұрлым (абсцисса осіне проекцияны қараңыз). Математикалық жағынан бұру операциясы кейбір коэффициентке (параметр салмағы) көбейтуге эквивалентты. Осындай өлшемдер өлшенген деп аталады: $\Delta = \{w_1(x - x_1)^p + w_2(y - y_1)^p + w_3(z - z_1)^p + w_4(f - f_1)^p\}^{1/p}$, әрине $\sum w_i = 1$. Салмақтар жиынтығын әлі «сарапшының артық көру құрылымы» деп атайды, ал өлшемнің нақты түрін – түйіншек (свертка). Катал түрде дәл осы формуланы: $\Delta = \{w_1(x - x_1)^p/a_1 + w_2(y - y_1)^p/a_2 + w_3(z - z_1)^p/a_3 + w_4(f - f_1)^p/a_4\}^{1/p}$ деп жазу керек. Мұнда a_n коэффициенттер шкаланы нормалдау үшін қызмет етеді. Бірақ, бұл формула да әмбебап емес, өйткені кейбір шкалалар үшін «бірыңғай» масштабтан басқа «есеп басын жылжыту» қажет болады.

Егер шешімнің көп ретті қабылдауы кездессе одан да күрделі жағдай болады. Уақытты еске алып бір параметр бойынша медициналық диагнозды қою есептің мысалын қарастырайық (8.6-сурет).



8.6-сурет. Тыныс жолдары мен өкпе қабындының үздіксіз температуралық кисықтары

Өкпенің талаурауы (тым тонудан және әлсіреуден басталды деп ойлайық) жалпы жағдайда температураның күрт көтерілумен және көңіл күйі нашарлаумен сипатталады. Егер еш нәрсе істемесек, бір аптаның ішінде температура көтеріледі және көңіл күйі нашарлайды, сосын дағдарыс басталады, яғни температура күрт көтеріледі және сонан соң төмендейді. Антибиотиктер болмағанда ауру адам дағдарыстан аман-есен шығып

жазылады, әйтпесе, өліп қалады. Өкпе қабынғанда температуралық қисық өзгеше жүреді – пациенттің күйі тұрақты, температура көтерілмеген немесе тіпті төмендеген. Диагнозды екінші күні қою керек. Әрине, бірінші күнді еске алмай мұны істеуге болмайды. Егер бірінші күні аурудың температурасы аса қатты көтерілмесе, онда тыныс жолдары, әйтпесе – өкпе қабынғаны. Бірақ, ақырғы шешімге екі нүкте жеткілікті ме? Ал нешеу жеткілікті? Егер нүктелер «дәл модельді қисыққа жатпаса» не істеу керек? Берілген мысалда тек бір параметр ғана еске алынады, ал нақты диагностика үшін ең азы 7-8 параметр қажетті.

Қойылған сұрақтарға бізмәнді жауаптарды табу оңай емес. Кездесетін әдістер негізінде шешімді бір рет қабылдауға тиіседі. Қазіргі заманда медициналық диагностика есептері критериалды әдістер көмегімен сирек қарастырылады. Жиі нейрон желілерді пайдаланады, себебі олар анағұрлым тиімділеу (мысалы, салмақтарды есептеу үшін). Бірақ, жоғарыда айтылған проблемалар нейрон желісіне де қатысты. Мысалы, біз салмақтарды есептеп, пациент пен модельдік қисықтар арасындағы 14 күн аралығында өлшенген айырманы анықтадық. Баламаның ешбіреуі басым болатындай шықпау мүмкін, өйткені аурудың бастауы бір модельге ұқсайды, ал аяқтауы басқа модельге ұқсас болады. Онда салмақтарды әрбір күн үшін енгізіп, графиктерді «сатыларға» (бастауы, барысы, жазылуы) бөлеміз. Нәтижесінде түсініксіз құру логикасы бар тым үлкен суперкритерий алыну мүмкін.

Тәжірибеден шығады тиімді диагностика жасау үшін 7-8 маңызды параметр жеткілікті. Егер параметрлер санын көбейтсек, диагностиканың дәлдігі аса өзгермейді. Сондықтан, тек маңызды параметрлерді таңдау қажет. Сонымен, критериалды есеп – бұл салмақтарды негізді таңдау есебі, мұнда маңызды параметрлер «үлкен» салмақты алады.

Критериалды әдістердің даму салдарының бірі «бөлетін кеңістік / гипержазықтық / функция» деген ұғым. Бөлетін гипержазықтық (шешуші ереже) – бұл кейбір геометриялық интерпретациясы. Яғни осы функция зерттелетін нүкте $A(a_1, a_2, a_3, \dots, a_n)$ кейбір салаға жатама жатпай ма деген сұраққа бізмәнді жауап бере алады (8.3-сурет). Осы функцияның түрі бөлінетін салалардың конфигурациясына және өзара орналасуына қатты тәуелді. Егер салалар арасында кейбір түзуді жүргізе алсақ және барлық нүктелер оның екі жағында қалатындай болса, онда «сызықтық бөліну» туралы айтады. Алғашқы перцептрондар (нейрон желінің болашақ үлгісі) осы принципке негізделген болатын.

Егер салалардың біреуі дөңес, ал екіншісі басыңқы болса, онда есеп күрделі болып шығады (8.5-сурет). Одан әрі күрделіледі, егер салалар қиылысатын болса, мысалы екі бұлт. Бір бұлт – орманның өртенуі, екіншісі - өзен бетінен булану. Арасындағы шекарасын табу керек. Әрине, осындай есепті бір ғана бөлетін түзу көмегімен шешу мүмкін емес. Ол үшін кейбір қисық қажетті, не жанамалар негізінде оның кесектік аппроксимациясы қажетті. Берілген жағдайда бөлетін функцияның түрі екінші, үшінші (4-дейін) ретті сызықтық емес функция болып шығады. Осындай

гипержазықтық негізінде айырып тану туралы негізгі математикалық теоремалар нейрон желілер теориясында М. Мински және С. Пейрепт дәлелдеген.

Енді, қарастырылған сұрақтарға жататын, бір қатар маңызды кезендерді қайталайық, келтірілген жолға сүйеніп келесіні еске алайық:

- баламалар көптігі шектеулі, есептеулі және континуалды болу керек;
- баламаны бағалау бірден бірнеше критерий (әрбіреуі көп өлшемді бола алады, мысалы, активті және реактивті бөліктен тұратын толық электр кедергісі, өлшеммен және түсімен сипатталатын медициналық симптом «бөртпе» және т.б.) бойынша жүзеге асырылу мүмкін. Критерийлерде сандық және сапалық сипаты бар және олар түрлі типті шкалада өлшенеді;
- таңдау тәртібі бір реттік немесе қайталанатын (мысалы, тәжірибе арқылы үйретуді жүргізу, яғни салмақтарды қайта есептеу) болу мүмкін;
- таңдаудың салдары алдын ала белгілі болу мүмкін (айқындылық жағдайдағы таңдау) немесе ықтималдық сипатты болу мүмкін (тәуекел кезіндегі таңдау) немесе бір мәнді емес шығысы болу мүмкін (белгісіздік жағдайдағы таңдау).

Атап айтылған варианттардың әртүрлі тіркестері әртүрлі математикалық таңдау есептерге әкеледі. Осындай сұрақтарға арналған математиканың арнайы бөлімдері бар, мысалы «ойын теориясы», «операцияларды зерттеу», және т.б.

8.1.2. Өлшеуіш шкалалар

Баламаларды салыстыру объектің сипаттамаларын өлшеу негізінде өтіп жатады. Өлшеу – бұл алгоритмдық операция, ол объектінің, процестің, құбылыстың берілген байқалған күйіне сәйкес: сан, нөмір немесе символ сияқты белгі қояды.

Өлшеу кезінде ең маңызды мәселе өлшеуіш шкаланың типі. Өлшеу теориясында 6 негізгі шкалалар типін ажыратады (қуаттың өсу ретімен): номиналды, реттік, интервалды, циклдық, қатынастар, абсолютты.

Номиналды шкала (атаулар шкаласы, таптасу шкаласы) – бұл кез-келген екі күйі туралы тек айырып тануға болама не болмай ма деген объектілерді қарастыруға мүмкіндік береді. Және түрлі күйлерге сәйкес түрлі белгілер қояды, ал айырып алмайтын күйлерге – бірдей белгілер қоятын өлшеу алгоритмдер.

1. $\text{Не } A = B, \text{ не } A \neq B.$
2. $\text{Егер } A = B, \text{ онда } B = A.$
3. $\text{Егер } A = B \text{ және } B = C, \text{ онда } A = C.$

Мұнда = символы эквиваленттік қатынасты белгілейді; егер де A мен B – сандар болса, онда бұл теңдеу белгісі.

Кластарды белгілеу үшін шкалада сөздер, символдар, цифрлер, олардың түрлі комбинациялар (мысалы, түстер шкаласы – қызыл, қызғылт

сары, сары және т.б.) пайдаланылады; олардың еру реті маңызды емес (8.7-сурет). Егер біз түске толқын ұзындығын салғастырсақ, онда бұл номиналды шкала болмайды. Онан әрі «квантификатор» ұғымды пайдаланамыз.



8.7-сурет. Эквивалентты номиналды шкалалар

Номиналды шкаладағы белгілермен, олар сандық болса да, арифметикалық операцияларды, тек қана олардың сәйкес келуін тексеру операцияларды жүргізуге болмайды. Сәйкес келуін тексеру операциялардың нәтижесімен бірге одан күрделі өзгертулерді орындауға болады: сәйкес келудің санын санау, салыстырмалы жиіліктерді есептеу және салыстыру, статистикалық процедураларды орындау және т.б.

Мысал. Кейбір эксперименттің басында біз сынаушыларды нөмерлейміз. Аяғында алынған нәтиже бойынша сынаушыларды реттейміз. Жалпы жағдайда, басындағы нөмер мен және соңындағы рангпен ешқандай өзара байланыс болмайды.

Ал үй ішіндегі пәтер нөмерлерінің шкаласы кейбір заңдылықтарды анықтауға мүмкіндік береді. Мысалы, 80 нөмерді 40 нөмерге бөлу мағынасыз, бірақ, 80-нен 40-ты алсақ біз пәтерлердің өзара орналасу туралы қорытынды істей аламыз. Сондықтан, осындай шкаланы номиналдыға да реттікке де жатқызуға болады.

Номиналдық шкаланы күшейтсек, біз реттік шкаланы (ординалды, рангтік) аламыз, ол белгілі бір түрмен кластарды салыстыруға, квантификаторлардың ілесу ретін көрсетуге мүмкіндік береді. Берілген жағдайда квантификатор — бұл шкаладағы бөлік.

Шкаладағы квантификаторлар сәйкес келуіне байланысты күшті (қарапайым) ретті, әлсіз ретті және жарым-жарты ретті шкалаларды ажыратады. Номиналды шкаланың аксималарына қосымша келесі аксиомалар орындалу қажет:

Күшті рет:	Әлсіз рет:	Жарым-жарты рет:
егер $A > B$, онда $B < A$;	$A \leq B$ немесе $A \geq B$;	өзара
егер $A > B$ және $B > C$,	егер $A \geq B$ және $B \geq C$,	салыстырылмайтын қос
онда $A > C$.	онда $A \geq C$.	кластар бар, яғни
		не $A \leq B$, не $B \leq A$.

Жарым-жарты ретті шкалалар социологиялық зерттеулерде жиі пайда болады. Мысалы, сатып алушы сұранысты зерттегенде, екі әр текті тауарлардың қайсысы өзіне ұнайтынын субъект айталмайды.

Реттін қатынасы салыстыратын кластар арасындағы қашықтық туралы ештеме айталмайды (8.5-сурет қараңыз). Реттік экспериментал деректерді цифрлермен көрсетілсе де, сан ретінде қарастыруға болмайды, олармен

амалдар жүргізілмейді. Мысалы, орта мәнің есептеуге, мәнді мәнге немесе коэффициентке бөлуге жарамайды.



8.8-сурет. Қайрат баланың ұнату реттік шкаласы

Эксперимент арқылы анықталған: Қайрат тирге бару үшін, алдымен, балмұздақтан, сосын қампиттен де бас тартады. Бірақ, тир үшін ол екі қорап қампиттен немесе төрт балмұздақтан бас тартатынын анық айталмаймыз.

Барлық психологиялық шкалалар реттік болады. Егер кейбір психологиялық тестін нәтижелері екі субъект үшін a_1 мен a_2 тең болса, онда бірінші субъект екіншіге қарағанда, кейбір қызметке «азды-көпті қабілетті» немесе «қабілеттері бірдей» деп айтуға болады. Бірақ, бірінші субъект екіншіден « a_1/a_2 есе қабілетті» деп айтуға болмайды. Керісінше, егер бір атлет 100 кг штангыны көтерсе, ал екіншісі – 200 кг көтерсе, онда «екінші атлет біріншіден, 2 есе күшті» деп айту әділ, бірақ мұнда шкала типі басқа.

«Сандармен жұмыс істеу» үшін ұмтылыс шкаланың интервалды (бұл шкалада екі түрлі квантификатор арасындағы қашықтық белгілі) типіне дейін күшеюіне келтіреді. Қашықтықты шкаланың ұзындығы бойынша бірдей өлшем бірлікте көрсетеді. Интервал ұзындығы шкаладағы орналасу орнына тәуелді емес. Арифметикалық операциялар квантификатор мәнімен емес, интервалдар ұзындықтарымен ғана өткізіледі, себебі осындай шкалада «0» жалғыз емес.

Мысал. Суды 9° -тан 18°C -ға дейін қыздырды. Судың температурасы екі есе өсті деп айтуға бола ма? Фаренгейт шкаласы бойынша дәл осындай өлшеулер мұндай нәтиже береді $37^{\circ} - 42^{\circ}\text{F}$. (Фаренгейт пен Цельсий шкалалар арасындағы байланыс мына формуламен беріледі $F = 5/9^{\circ}\text{C} + 32$).

Кейбір шамаларда, өзінің физикалық табиғаты бойынша, абсолют нөлі болмайды немесе есептеу басын қоюға еркіндік береді. Сондықтан олар интервалды шкалада өлшенеді: температура, уақыт, жер биіктігі. Статистикада орталық моменттерде (соның ішінде дисперсия) объективті физикалық мағынасы бар, ал бастапқы моменттер (соның ішінде орта мәні) есептеу басымен қатар салыстырмалы болады.

Аурудың күй-жағдайын бақылайтын интеллектуалды бағдарламаның кодындағы кәдімгі қателік мысалын келтірейік. Дене температурасының «елеулі өзгеруіне» бағдарлама әсер тигізу керек. Норма $36,6^{\circ}\text{C}$ болғанда 40°C ахуал пайда болды. Бағдарлама салыстырмалы ауытқуды есептеді: $(1 - 36,6/40) = 1 - 0,915 = 0,085$; ауытқу «айқын кішкене» болғандықтан, ешқандай іс-әрекет орындалмады.

Дұрыс есептеулер мынадай болар еді:

$$1 - |40 - 36,6|/|36,6 - 43| = 1 - 0,45 = 0,55.$$

Мұндағы 43°C — тірі адам денесінің шекті температурасы.

Қатынастар шкаласындағы квантификатор мәндерімен «сандар секілді» арифметикалық амалдар жүргізуге болады. Және де аддитивность аксиомаға қанағаттандырылу (қосымша) керек:

$$\begin{aligned} &\text{егер } A = P \text{ және } B > 0, \text{ онда } A + B > P; \\ &\quad A + B = B + A; \\ &\text{егер } A = P \text{ және } B = Q, \text{ онда } A + B = P + Q; \\ &\quad (A + B) + C = A + (B + C). \end{aligned}$$

Қатынастар шкаласына сәйкес болатын шамалар: ұзындық, салмақ, ақша.

Кейбір өлшеуіш аспаптарда шкала тұйықталған (айырмалық, циклдық, кезеңдік шкала). Бұл жағдайда кейбір кезеңі $y = x + nb$, $n = 0, 1, 2, \dots$ бар интервалды шкала туралы әнгіме.

Ақырғы және ең «күшті» шкала абсолютті. Оның абсолютті нөлі және абсолютті бірлігі бар. Ол жалғыз және бірегей. Абсолют шкаланың басқалар мен салыстырғанда, маңызды ерекшелігі – оның бірлігінің дерексіздігі (өлшемсіздігі) және абсолюттігі. Айтылған ерекшелігі абсолют шкаланың көрсеткіштеріне, басқа шкалаларға қарағанда, түрлі операцияларды жүргізуге және осы көрсеткіштерді логарифм аргументі мен дәреже көрсеткіші ретінде қолдануға мүмкіндік береді. Осындай қажеттілік, мысалы, психологиялық эксперименттерді өңдеу кезінде жиі кездеседі. Зерттелетін шамалар арасындағы өзара байланысты анықтау үшін қарапайым корреляция қолданылады. Осындай есеп тек нормалды заң бойынша үлестірілген бастапқы деректерге ғана рұқсат. Нақты жағдайда (мысалы, log-нормалды үлестіру) деректер логарифмды пайдаланады, себебі олар нормалды түрде сөзсіз үлестіріледі. Мынадай шкала сандық ось болып табылады.

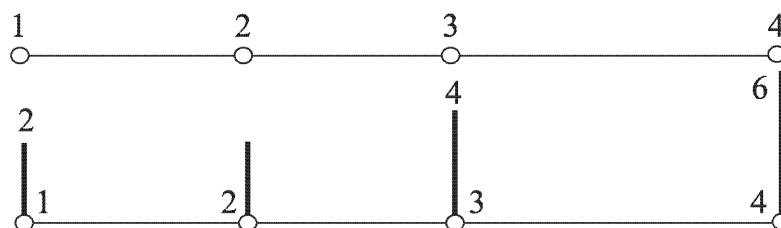
8.2-кестеде қарастырылған шкалалар туралы негізгі мәліметтер жиналған. Шкала неғұрлым күшті болса, зерттелетін объект, құбылыс, процесс туралы өлшеу арқылы соғұрлым көп мәліметтер алуға болады. Кейде бастапқы байқауларды әлсіз шкалада жасайды, сосын қандай да болса жорамал енгізіп, одан күшті шкалаға ауысады. Өңдеу негізінде алынған зерттеу нәтижелері тағы өзгертіледі, енді шкаланың үшінші типі үшін, әдетте олардың көрнекілігін қамтамасыз ету. Мысалы, 7-8 параметр бойынша шешімді қабылдау нәтижесінде үшөлшемді бейнені салады.

Өлшеуіш шкалалар типі

Шкала атауы	Анықтайтын қатынас	Шкаланың эквивалентті өзгертуі	Деректер үстінен мүмкін операциялар	Деректердің екінші реттік өңдеуі
Номиналды	Эквиваленттік	Атаулардың орнын ауыстыру	Кронекер символды есептеу	Салыстырмалы жиіліктерді және олармен операцияларды есептеу
Реттік	Эквиваленттік, ұнатушылық	Бір қалыпты (ретті өзгермейтін)	Кронекер символды және рангты есептеу	Салыстырмалы жиіліктерді, олармен операцияларды және таңдаулы квантильді есептеу
Интервалды	Эквиваленттік, ұнатушылық, интервал қатынастарын сақтау	Сызықтық өзгерту $y = ax + b$, $a > 0, b \in R$	Кронекер символды, рангты және интервалдарды (байқаулар айырмасы) есептеу	Интервал үстінен арифметикалық амалдар
Циклдық	Эквиваленттік, ұнатушылық, интервал қатынастарын сақтау, мерзімдік	Ығысу $y = x + nb$, $b = \text{const}, n = 0, 1, 2, \dots$	Интервалды үшін сияқты	Интервалды үшін сияқты
Қатынастар	Эквиваленттік, ұнатушылық, интервал қатынастарын сақтау, мерзімдік, екі мәннің қатынастарын сақтау	Созылу $y = ax, a > 0$,	Барлық арифметикалық операциялар	Әрбіреу ыңғайлы өңдеу
Абсолют	Эквиваленттік, ұнатушылық, интервал қатынастарын сақтау, мерзімдік, екі мәннің қатынастарын сақтау, абсолют және өлшемсіз бірлік, абсолют ноль	Бірегей шкала	Барлық арифметикалық операциялар; дәреже көрсеткіші, логарифмның негізі мен аргументі ретінде пайдалану	Әрбіреу қажетті өңдеу

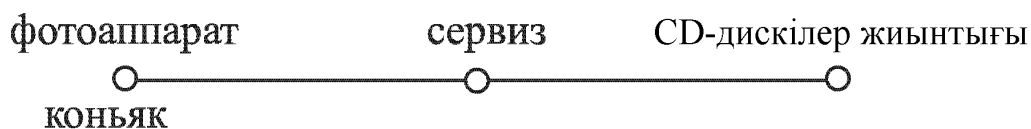
Шкалалар типі туралы бір күлкілі мысал келтірейік. Майра мен Бота Қайраттың тұған күніне жиналып жатыр. Қандай сыйлық ұсынамыз?

1. Варианттар генерацияланады: 1 – фотоаппарат, 2- коньяк, 3 – сервис, 4 – CD-дискілер жиынтығы. Ұсыну барысында баламалар бағаланбайды және сыналмайды, әйтпесе, «біз ештеңені ойлап таппаймыз». Баламалар жинағы номиналды шкаланы құрайды, оның үстіндегі ұнатулар белгісіз (8.9-сурет). Майра мен Бота сарапшыларды (Қайратты жақсы білетін) сұрастырады. Жоғары айтылғаннан Қайрат нені ең жақсы көреді? Номиналды шкалада жиіліктерді өлшейді. 1 мен 2 баламаға екі адам дауыс берді, үшінші баламаға – төрт адам және соңғысына – алтау.



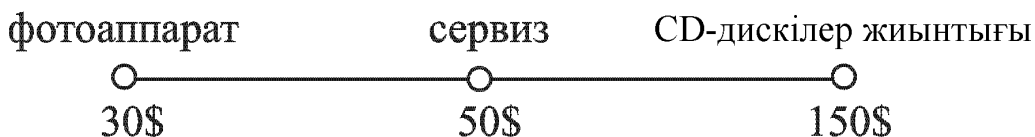
8.9-сурет. Номиналды шкаладағы қарапайым жиілік талдау

2. Майра мен Бота Қайраттың ұнатуларының реттік шкаласын салады (8.10-сурет).



8.10-сурет. Әлсіз ретті шкаласы

3. Майра мен Бота шығындар туралы ойлайды – олар варианттарды кейбір валютада бағалайды. Фотоаппарат жақсы коньяктан арзан, сондықтан реттік шкала «күшті» болып түседі, ал ақша белгілерін енгізгеннен кейін оның типі салыстырмалыққа өзгереді (8.11-сурет).



8.11-сурет. Қатынастар шкаласы

4. Енді қыздарда екі шкала бар, біреуі Қайраттың ұнатуларын сипаттайды, екіншісі - Майра мен Ботаның ұнатулары (8.12-сурет). Қайраттың көзқарасынан варианттар бағалауы кейбір абстракты бірлікпен көрсетіледі (шкала «0-ге келтірілген» - бағалардан 2 алынған). Қатал айтқанда, «0, 2 және 4» квантификаторларға сан мағынасын қосып жазуға болмайды, бірақ біз мысалды оңайлату үшін осындай жорамал істейміз. Ол үшін интервалды шкаланы салыстырмалыққа келтіреміз.



8.12-сурет. Шкаланы «0»-ге келтіру

5. Қайраттың көзқарасынан дискілер жиынтығы сервиздан 2 есе артық көрінеді. Майра мен Бота сервиз CD жиынтығынан 3 есе артық көрінеді деп ойлайды. Әрине, фотоаппарат 5 есе артық көрінеді, бірақ ол Қайратқа ұнамауы мүмкін (8.13-сурет).



8.13-сурет. Қайраттың және қыздардың ұнатуларын салғастыру

Сонымен, қыздар 50\$ баламаға келеді, ал Қайраттың бағасы – 2.

8.1.3. Шешімді қабылдаудың кейбір әдістемелері

Ең бірінші айту керек, таңдау бір критериймен (немесе бірге оралған көптік критерийлермен), не критерий жинағымен (бірге келтірмейтін) ескертіледі. Таңдау нәтижесі «жақсырақ», «берілген деңгейге сәйкес», «кейбір берілген деңгейден кем емес» болуы мүмкін. Таңдау нәтижесі айқын анықталған немесе айқын емес болуы мүмкін. Егер кейбір көзқарастан «анық жақсы» баламаны белгілеп алсақ, онда оны басым болатын деп атайды. Қалған варианттарды «анық нашар баламалар» деп атайды. Ал егер бірнеше варианттар арасында не жақсысын, не нашарын көрсете алмасақ (мысалы, $q(X_1) = (3,5)$, $q(X_2) = (2,6)$ – екі критерий бойынша екі баламаның бағалары), салыстырылмайтын (басым болмайтын) баламалар көптігі туралы немесе Парето көптігі туралы айтады.

Критериалды таңдауды жүзеге асырудың қалыптасқан әдістемелері «техника» немесе «бағдарламау әдістері» деп аталады. Мысалы, мақсатты бағдарламау және келісімді бағдарламау туралы айтады.

Критериалды әдістер көмегімен қабылданатын шешімдер тиімділігі заттық салаға тәуелді. Мысалы, экология-экономикалық есептер, объектілерді реттеу есептер және басқалар.

Басты критерий әдісті карап шығайық. Егер баламалар (олардың арасынан ең жақсысын таңдау керек) бір немесе бірнеше (олардың арасында «ең маңыздысы» біреу бар) критериймен бағаланса, онда «ең жақсырақ» принципі бойынша таңдауды жүзеге асыру қажет:

$$x_{\text{opt}} = \arg[\max q(x)], \quad x \in X.$$

Мұнда X – қарастырылатын баламалар көптігі, $q(x)$ – критерий мәні.

Егер де критерийлер арасында ең маңызды біреу (немесе бірге оралатын топ) бар болса, онда басты критерийдің максимизациясы туралы айтады, қосымша критерийлер ол берген деңгейлерде қалады (шартты максимизация):

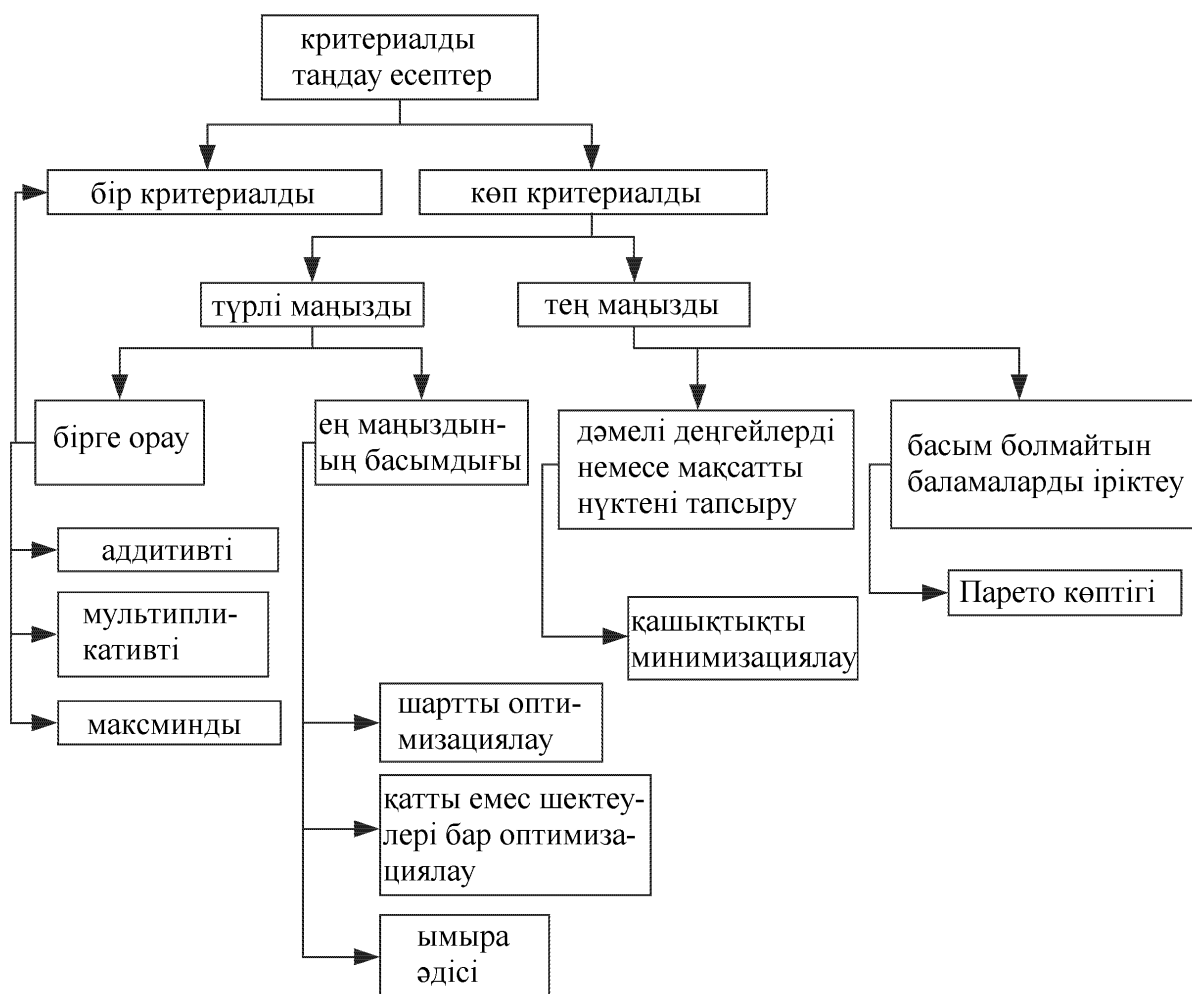
$$x_{\text{opt}} = \arg[\max q_0(x) \mid q_i(x) \leq C_i, i = 1, 2, 3, \dots, p], \quad x \in X.$$

Жол беру әдісінде (шартты максимизацияның өзге түрі) біз кейбір үлкен емес шаманы анықтаймыз, оған басты критерийдің орнына жол береміз. Осындай жол беру арқылы екінші дәрежелі критерийлер маңызы бойынша реттелу мүмкін.

Берілген қасиеті бар баламаларды іздеу түрлі баламаны таңдау үшін қолданылады, олардың дербес критерийлері мәні берілген диапазонда жатады. Дербес жағдайда бос баламаны алу мүмкін.

Шартты максимизация немесе берілген қасиеті бар баламаларды іздеу кезінде «дәмелену деңгейлер» туралы жиі айтады. Бұл деңгейлер, критерий мәнінің өзгеру диапазоны бізге оқиғалардың: «ең нашар», «ең жақсы», «ымыралы» және т.с.с. даму варианттарын тапсырады. Әрине, егер түрлі критерийлердің дәмелену деңгейлері қиылысатын болса, онда біз идеал нүктені аламыз. Геометриялық жағынан дәмелену деңгейлері картадағы тең деңгейлі сызықтарына өте ұқсайды.

Төменде кейбір ең жиі қолданылатын әдістердің жалпы сызбасы келтірілген (8.14-сурет).



8.14-сурет. Кейбір ең маңызды критериалды әдістер

8.1.4. Шешім көпкритериалды қабылдауының оқу мысалы

Критериалды әдістер көмегімен медициналық диагнозды қою мысалды қарастырайық. Барлық деректер нақты аурулардан алынған.

Пациент артынан байқаулар мәліметтері бар болсын (8.3-кесте), ол сары аурумен аурады деп болайық (не ауырмайды). Кестенің бағандарында зертханалық зерттеулер нәтижелері жазылған, мысалы, кан мен зәр талдауы және т.с.с. Бірінші ұшық – ауруды байқау күннің нөмірі. Атаулардың ашуы берілмейді, себебі ЖИ жағынан есепті қоюға ол ешнәрсе қоспайды (және де білімдер бойынша инженер медицинаны білмейді). Тек өлшеуіш шкаланың типі қызықты. Қабылдайтын мәндеріне сәйкес, типті өзіміз анықтауымыз керек.

8.3-кесте

Диагностика үшін бастапқы деректер

3	0,00	95,00	50,00	1,50	0,50	1,00	1,00	1,00	1,90	50,40
---	------	-------	-------	------	------	------	------	------	------	-------

Сау адамдарды байқаудан сол параметр бойынша 8.4-кесте алынған.

8.4-кесте

3	0,00	95,00	50,00	1,50	0,50	1,00	1,00	1,00	1,90	50,40
---	------	-------	-------	------	------	------	------	------	------	-------

Сары аурудың өзге түрімен ауратын пациенттерді байқау келесі нәтижелер берді.

8.5-кесте

Паренхиматозды сару ауру

3	0,00	10,52	28,00	1,30	2,00	4,69	0,00	0,00	1,00	0,00
---	------	-------	-------	------	------	------	------	------	------	------

8.6-кесте

Механикалық сару ауру

3	1,00	18,48	10,80	0,48	0,40	19,18	1,00	1,00	0,00	0,00
---	------	-------	-------	------	------	-------	------	------	------	------

8.7-кесте

Гемолитикалық сару ауру

3	0,00	4,80	30,00	0,30	0,40	3,60	0,00	0,00	0,00	1,00
---	------	------	-------	------	------	------	------	------	------	------

Медициналық әдебиеттерді оқып білу және сарапшылармен ақылдасу келтірілген параметрлер үшін «маңызды ауытқуларды» анықтауға мүмкіндік берді.

8.8-кесте

Гемолитикалық сару ауру

3	0,10	0,30	6,00	0,01	0,01	0,00	белгісіз	белгісіз	белгісіз	белгісіз
---	------	------	------	------	------	------	----------	----------	----------	----------

8.9-кесте

Денсаулық

3	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,00
---	------	------	------	------	------	------	------	------	------	------

8.10-кесте

Механикалық сару ауру

3	0,10	8,00	2,00	0,03	0,01	3,20	белгісіз	белгісіз	белгісіз	белгісіз
---	------	------	------	------	------	------	----------	----------	----------	----------

8.11-кесте

Паренхиматозды сару ауру

3	0,10	0,40	5,00	0,12	0,20	0,40	белгісіз	белгісіз	белгісіз	белгісіз
---	------	------	------	------	------	------	----------	----------	----------	----------

Қашықтау өлшем (ауру моделден нақты баламадан) ретінде мына функцияны таңдаймыз

$$D = \sum [w_i (X_{i\text{нәтижесі}} - \tilde{O}_{i\text{әрекеті}}) / (2 \cdot X_{i\text{нәтижесі}} \cdot \tilde{a}_{i\text{әрекеті}})],$$

мұндағы $i = 1, \dots, 10$ – параметр нөмірі; w_i – i -индексы бар параметр салмағы.

«Бөліну функцияның» нақты түрі шешімнің дұрыстығына қатты әсер етеді. Әрбір есепке ол жеке іріктеп алынады және күрделі, көлемді түрде болады. D функция әрқашан бірнеше түрлі әдіс арқылы анықталады:

$$D_{\text{(пациент - паренхиматозды сары ауру)}} = 366,7;$$

$$D_{\text{(пациент - механикалық сары ауру)}} = 290,2 \text{ (ең жақын балама)};$$

$$D_{\text{(пациент - гемолитикалық сары ауру)}} = 470,5;$$

$$D_{\text{(пациент - денсаулық)}} = 418.$$

Салмақтарды таңдау да маңызды сұрақ. Берілген есепте салмақтар тең деп алынған, ол тәжірибеде сирек кездеседі. Егер есепті қайталайтын болсақ, онда «0-ге бөлу» немесе «белгісіз шамаға бөлу» деген оқиғалар кездесу мүмкін. Осындай проблеманы шешу - білімдер бойынша инженердің жұмысы. Мысалы, «0» орнына кейбір кіші мән қойылады (берілген мысалда 0,1). Екінші оқиға («белгісіз шамаға бөлу») нашарлау. Егер осы «белгісіз шаманы» анықтай алмасақ, оны кейбір айнымалымен ауыстырады. D функцияның мәні «сан/белгісіз» түрде алынады. Осындай бағалар бойынша да шешім қабылдауға болады.

Берілген мысалда пациент шынында да механикалық сары аурумен ауырады.

8.2. Ықтималдық әдістер

Баламалардың жақындау өлшемі ретінде кең ыңғай кластар ықтималдық бағаларды пайдаланады. Қазіргі уақытта әр шешімдердің дұрыстығын, ықтимал шығысын бағалауға мүмкіндік беретін әдістер тәуекел теориясында зерттелінеді. Теорияның сипаты қолданбалы, сондықтан әнгіме заттық саладағы тәуекел туралы жүргізіледі. Математикалық жағынан бұл кәдімгі ықтимал теориясы. Тәуекел теориясының мағынасы математикалық емес, нақты заттық салада өлшеудің ерекшелігіне және деректерді ықтималдық өңдеуіне байланысты алгоритмдық процедуралар.

Жоғарыда келтірілген медициналық диагнозы бойынша ықтимал теориясының әдістер қолдануының қарапайым мысалын қарастырайық. «Маңызды ереже» ретінде Байес формуласын алайық:

$$P(A/B) = [P(A) \cdot P(B/A)] / [\sum (P(A) \cdot P(B/A))].$$

Шартты ықтималдық формуласы егер « B » оқиғасы пайда болса, « A » оқиғаның ықтималдығын есептей алады, яғни тәжірибе алдында (априорды) бар болған $P(A)$ мен $P(B/A)$ ықтималдықтар негізінде алынған $P(A/B)$. Формула бөліміндегі қосындылау толық оқиғалар тобында барлық жорамал бойынша жүргізіледі.

1-мысал. Температура көтерілуі байқалғанда, пациент ангинамен ауыратын ықтималдығын есептеу керек. Егер пациент ангинамен ауырмаса, онда ол «ангина емес» пен ауырады, мысалы сау.

$$P(A_i / \bar{O}) =$$

$$= \frac{P(A_i) \cdot P(\bar{O} / A_i)}{P(A_i) \cdot P(\bar{O} / A_i) + P(\bar{A}_i) \cdot P(\bar{O} / \bar{A}_i)}$$

Априорды ықтималдықтар $P(\text{Ангина})$ және $P(\text{Температура/Ангина})$ осы жер үшін медициналық статистиканы талдаудан алыну мүмкін. Егер, емхананың жазулары бойынша, 10000 аурулардан 7000-ы ангинамен ауырып шықса, онда априорды ықтималдық $P(\text{Ангина}) \approx 0,7$. Және медициналық әдебиеттен табуға болады $P(\text{Температура/Ангина}) \approx 0,8$.

$P(\text{Ангина емес}) \approx 0,3$. Біраз $P(\text{Температура/Ангина емес})$ -ны табуға қиындау болады, қала берсе оны сараптама жасау арқылы бағалаймыз. Осы жолмен бөлім үшін де бағаларды аламыз.

2-мысал. «В» оқиға - бұл бір оқиға емес, бір уақытта болатын толық оқиғалар тобы. Яғни «температура», «тері түсінің өзгеруі», «бүйір аурады» деген симптомдар бар болғанда, пациент паренхиматозды сары аурумен ауыратының ықтималдығы қандай (ол не үш сары аурулардың біреуімен ауырады, не сау)? Оңайлатып айтсақ, синдром ықтималдығы жеке симптом ықтималдықтар көбейтіндісіне тең.

Сонымен, критериялды мысалдағы деректер үшін аламыз:

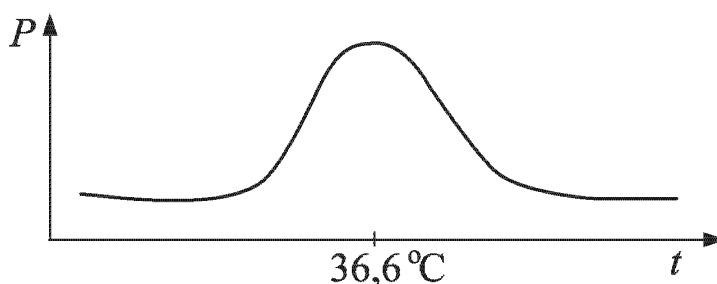
$$P(A_1) = P(A_2) = P(A_3) = P(A_4),$$

яғни аурулардың бәрін тең ықтималды деп жорамалдаймыз (A_i – аурулар аты: паренхиматозды, механикалық және гемолитикалық сары аурулар, денсаулық).

$$P(B_1, B_2, \dots, B_{10}/A_1) = P(B_1/A_1) \cdot P(B_2/A_1) \cdot \dots \cdot P(B_{10}/A_1),$$

сірә, бұл «тым азайтылған» баға.

$P(B_1/A_1), P(B_2/A_1), \dots$ ықтималдықтар - бұл B_1, B_2 және т.б. симптом үшін «осы аурудың қалыпты» мәнінен ауытқып кету ықтималдығын түсіндіреді. Мысалы, денсаулыққа қалыпты $36,6^\circ \text{C}$ мәнінен «температура» параметрдің ауытқуы 42°C дейін, береді $P(42^\circ \text{C}/\text{сау}) \rightarrow 0$, ал ауытқу $P(35,5^\circ \text{C}/\text{сау}) \approx 0,8$ (сау болғаны ықтимал). 0,8 деген баға осы аймақ үшін медициналық статистиканың талдауынан алыну мүмкін. Мысалы, симптом мәндері кейбір орта мәні айнала нормалды түрде үлестірілген деп жорамалдауға болады (8.15-сурет).



8.15-сурет. Бастапқы мәндер нормалды үлестірілу мүмкін

Қарастырылған симптомдар үшін (3000 астам салынған үлестірулерге негізделіп) табылған бағалар 1 – механикалық сары ауру, 2 – паренхиматозды, 3 – гемолитикалық, 4 – сау) және 8.12-кестеге түсірілген.

8.12-кесте

Байес формуласына қойылатын мәндер

	$P(0,0 / A_i)$	$P(95,0 / A_i)$	$P(50,0 / A_i)$	$P(1,5 / A_i)$	$P(0,50 / A_i)$
$i = 1$	0,60	0,90	0,20	0,05	0,05
$i = 2$	0,05	0,10	0,90	0,85	0,85
$i = 3$	0,01	0,01	1	0,02	0,02
$i = 4$	1,00	1	1	0,50	0,70

$P(1,00 / A_i)$	$P(1,00 / A_i)$	$P(1,00 / A_i)$	$P(1,90 / A_i)$	$P(50,4 / A_i)$	$P \cdot 10^{-6}$
0,95	0,60	0,90	0,20	0,01	0,277
0,10	0,15	0,03	0,90	0,30	0,395
0,01	0,01	0,01	0,10	0,60	0
0,50	0,01	0,01	0,01	0,01	0,00175

$$P(A_1) = (0,25 \cdot 0,277 \cdot 10^{-6}) / (0,25 \cdot 0,277 \cdot 10^{-6} + 0,25 \cdot 0,395 \cdot 10^{-6} + 0,25 \cdot 0,0 \cdot 10^{-6} + 0,25 \cdot 0,00175 \cdot 10^{-6}) \approx 0,41$$

$$P(A_2) \approx 0,58$$

$$P(A_3) \approx 0,0$$

$$P(A_4) \approx 0,002$$

8.3. Нейрон желілері

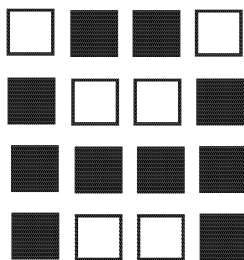
Бүгін ең белсенді түрде дамитын ЖИ бағыттарының бірі нейрон желілері – оны критериалдық, ықтималдық және логикалық әдістердің дубара моделі деп айтуға болады. Нейрон желісін өнімдер ережелері негізінде де жасалуы мүмкін. Бірақ, өнімдік жүйесі жеткілікті үлкен желіні жасай алмайды. Нейрон желісі бұл белгілі жағдайда «егер/онда» ережелердің «сандық жазбасы». 80 жылдары әзірленген VI буын ЭЕМ-ді жапондар «нейрокомпьютер» немесе «транспьютер» деп атады, жай айтқанда – параллель ЭЕМ.

Теорияның дамуы У. Макклох (W. McCulloch) және У. Питтс (W. Pitts) жұмыстарынан басталды, олар бас мидың іс-әрекетін қарастырған (1943 ж.). 1957 жылы психолог Фрэнк Розенблатт (Корнелл, АҚШ) «перцептрон» деп аталатын электрмеханикалық құрылғыны ойлап тапты. Ол ұлу көзін және оның ми мен өзара әрекеттесуін модельдеді. Модель әліппе әріптерін айырып таныйтын, бірақ жазу түріне әсерленгіш болатын. Перцептрон үшін A , $\bar{A}$ және A әріптер – үш әртүрлі белгілер.

1969 жылы классикалық жұмыс «Перцептрондар» шықты, онда М.Мински мен С.Пейперт (М.И.Т.) перцептрон көмегімен шамалы айырып тану туралы бір қатар теоремаларды дәлелдеді. Сосын қазіргі нейрожелілер

теориясының дамуы басталды, ал классикалық перцептрондар қолданудан шықты. Перцептронның бір тарихи мысалын көрсетейік.

Перцептрон мысалы. «әріп А» және «А әріп емес» бейнелерді айырып таныйтын перцептронды жасайық (8.16-сурет). Бірінші қадамда w_{ij} салмақтарды 1-ге тең деп, x_{ij} элементер мәндерінің көптігін екі элементтен $[0,1]$ тұрады деп алайық. Формула ретінде тандаймыз: $F(X, W) = \sum x_{ij} \cdot w_{ij}$, шекті мәні 10. Перцептрон түрі мынадай $\sum x_{ij} \cdot w_{ij} \geq 10$. $\sum x_{ij} \cdot w_{ij} \leq 16$ болғаны анық. Үлгі көмегімен үйретейік (8.16-суретте қараңыз). Аламыз $w_{11} = 0$, $w_{12} = 1$, $w_{13} = 1$, $w_{14} = 0$, $w_{21} = 1$ және т.б.

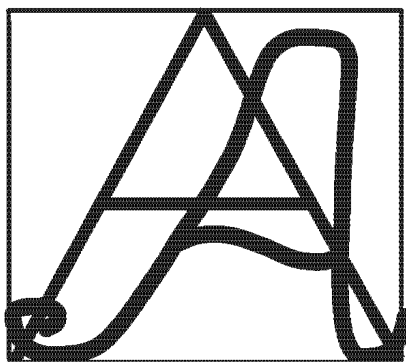


8.16-сурет. «А» әріптің растрлы ұсынуы

Енді айырып тану тәртібіне өтейік және мысалы, «R» әріпті көрсетіп көрейік. Оның бейнесінде де нүкте (1,1) қара (яғни 1 сияқты кодталады). Теңсіздік $\sum x_{ij} \cdot w_{ij} \geq 10$ орындалатындықтан, «R» әріпі «А» сияқты айырып танылды! Егер қатаң теңдік қоятын болсақ, онда «А», немесе «A» немесе «a» және т.б. айырып тануын жүзеге асыра алмаймыз. Перцептронды «A»-ны айырып тануға үйрету үшін, сірә w_{12} салмақтарды қайта есептеп, шектік мәнін азайту қажет болар. М. Мински мен С. Пейперттың классикалық кітабы дәл осы сұрақтарға арналған.

Мысалға қатысты келесі жолдар істеу керек.

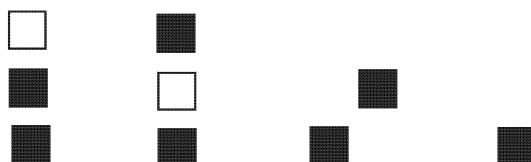
1. «А» әріпті бірімәнді идентификациялау үшін қажетті минимал нүктелер санын анықтау. Ол үшін қандай әліппемен (символдар жиынтығы) жұмыс істеуді анықтау керек болады (8.17-сурет).



8.17-сурет. Әріп әртүрлі жазылу мүмкін

2. Тек айырып тану символға сәйкес ерекшеліктерді анықтау. Мысалы, «А» әріптің міндетті ерекшелігі тұйықталған жоғарғы бөлігінің бар болуы. Ал «негізіндегі бұралаңның» бар болуы қажет емес.

3. 2 пункттің әрбір бекітуіне n -орынды $f(x_{sp}, x_{gh}, x_{nm}, \dots)$ түрі бар предикатты салыстырып, алғашқы екі пунктті логикалық бейнелеу. Предикат мәні 1 (ақиқат), егер ізделіп отырған элемент табылса және керісінше 0. Мұндағы индекстер – түрлі бүтін сандар, сонымен бірге бірінші индекс – тор ені, екінші – биіктігі. Солай «А» әріпті айырып тану үшін бізге бір орынды, екі орынды және үш орынды предикаттар керек болады (8.18-сурет).



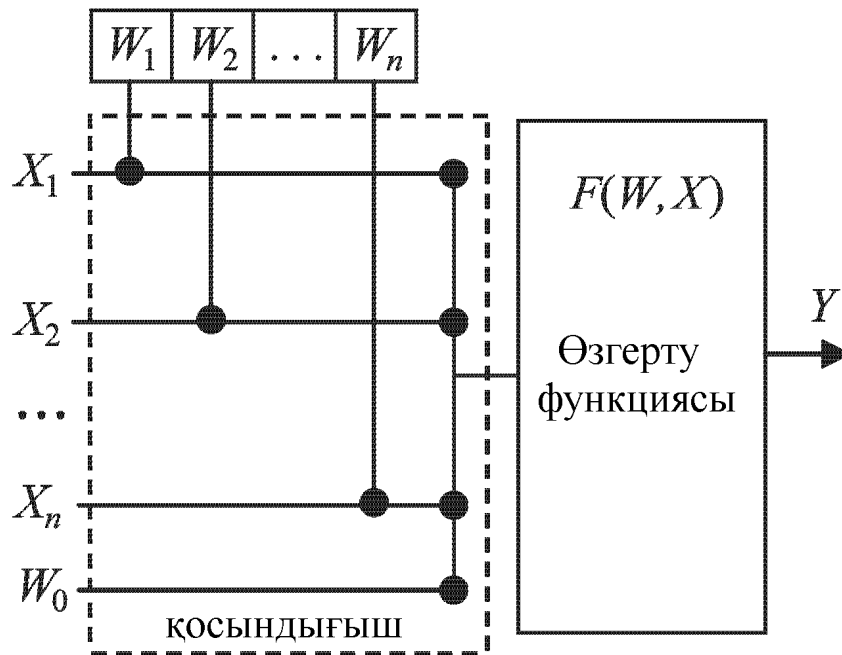
8.18-сурет. Екі үш орынды, бір екі орынды және бір орынды предикаттар

4. Перцептронның өзін бейнелеу, ол енді ұзынырақ және түсініксіз түрде болады:

$$\sum f(x_i) \cdot w_i + \sum f(x_j, x_k) \cdot w_s + \sum f(x_j, x_k, x_l) \cdot w_h \leq Q.$$

Мұнда индексы бар w әріптермен предикат салмағы белгіленген, ал қосындының ішінде бір орынды, екі орынды және үш орынды предикаттар тұр. Шекті мәні Q сәйкестіктің минимал қажетті санын тапсырады. Мысалы, бірінші ретті предикаттар саны (8.16-суретті қараңыз) 10-нан кем болмау керек, екінші – 6-дан, үшінші – 4-тен кем емес. Барлығы Q , ең жақсы жағдайда, 20 болу мүмкін. Басқа предикаттарды бейнелеп шығарсақ, басқа Q алар едік. Перцептрон таза түрде көбінесе оқу есептер үшін пайдаланылады.

Медицинада нейрон (8.19-сурет) деп бас ми қабығының талшығы аталады. Арнайы байланыс арқылы келіп түсетін электр сигнал әсерінен қоздырылып, нейрон өзіндік сигналын генерациялайды және айналадағыларға береді және т.б. Бір уақыттан кейін тежелу процесі басталады – жуықта қоздырылған нейрондардың электр белсенділігі басылады, ал «демалған» нейрондар қайтадан қоздырылады.



8.19-сурет. Сызықты емес өзгертуші

Нейрон желісін алдымен жуық математикалық теңдеулер жүйесі ретінде қарастыру керек. Желі көмегімен нақты есепті шешу кезінде не жаңа желіні жобалау кезінде, алынған жүйенің математикалық негіздеуі туралы ойлау керек.

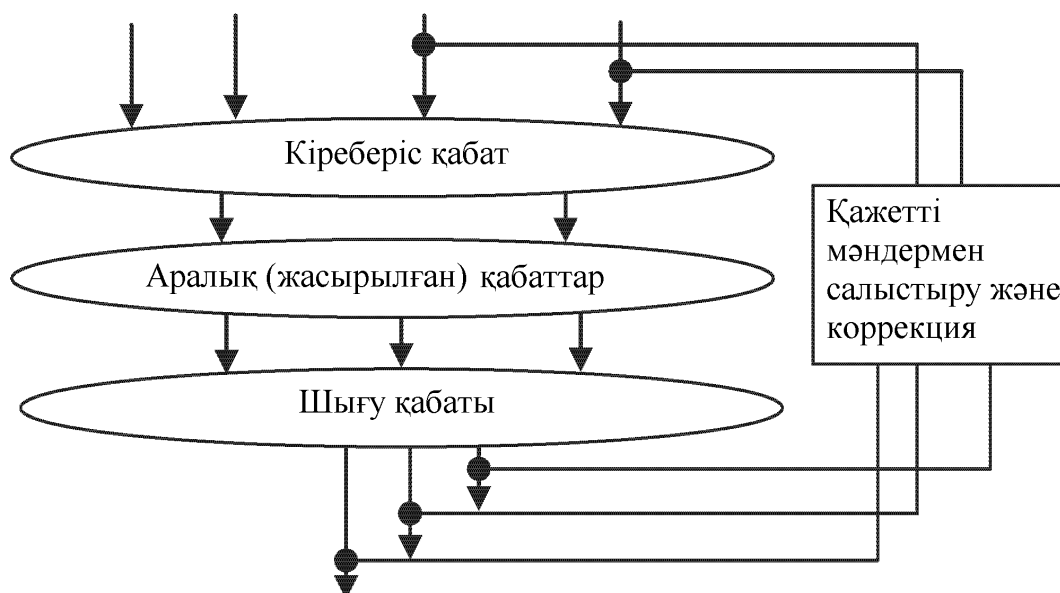
Математикалық көзқарастан нейрон – бұл сызықты емес өзгертуші, кіреберісіне кейбір мән (қоздырушы сигнал) берілген, ал шығысынан кірген мен күрделі немесе ықтималды түрде байланысқан басқа мән (қоздыру шамасы) алынған:

$$Y(X, W) = w_0 + \sum f(w_i \cdot x_i), \quad i = 1, 2, \dots, n;$$

мұндағы x_i – ақпараттық кіру мәндері; w_i – салмақтар, сонымен бірге w_0 – ығысу.

Бұл сызбаны былай істейді. Қосындығышта нейрон алған қоздыру есептеледі ($w_0 + \sum f(w_i \cdot x_i)$). Осы шаманың негізінде өзгерту функциясы нейронның өз қоздыруын есептейді - Y . Өзгерту функцияның нақты түрі шешілетін есептің өзгешелігін еске алып, таңдап алынады. Егер Y тек 0 немесе 1 мәнін алатын функцияны қабылдасақ, онда, әрине, өнімдер ережелерін аламыз.

Өзгерту функцияның пайдалы түрлері аса көп емес. Нейрон көптігін (миллиондаған болу мүмкін, ал өнімдер ережелерінде мыңдаған ғана болады) желіге біріктірген кезде «қабаттарды» ажыратады (8.20-сурет), олардың арасындағы байланыс бір бағытты немесе «кері таратуы» бар болу мүмкін. Өнімдік модельдегі ережелерді іріктеп алуды басқару кезінде «дайын өнімдер» қатарымен ұқсастық айқын.



8.20-сурет. Көпқабатты нейрон желісі

Желіні үйрету дегеніміз бұл w_i салмақтарды есептеудің итеративті процесі берілген дәлдігімен желі үстінде шығу мәндерін алуға мүмкіндік береді. Осындай салмақтарды есептеу үшін, кіру мен оларға сәйкес шығу мәндерінің сынақ жиынтықтары болу қажет. Осындай жиынтықтар басқа модельдерде де пайдаланылады және үйрететін іріктеу деп аталады. Желіні үйрететін іріктеу үстінде алдын ала оқытуды керек қылмайтын оқыту алгоритмдар бар, олар нақты есепті шешу процесінде тура үйренеді. Осындайға, мысалы, өзін-өзі ұйымдасқан Кохонен желісі (картасы) жатады.

Салмақтар мәндерінің өзгеруі кейбір берліген қадам арқылы, не кездейсоқ шаманың кейбір үлестіру заңы бойынша өту мүмкін. Шешімді критериялды қабылдау кезіндегі жол беру әдісімен ұқсастығы айқын көрініп тұр. Өзгеру салмақтар кеңістігінде градиент бағыты бойынша өтіп жатады. Осы кезде екі оқиға болу мүмкін, бірнеше есеп қадамдарынан кейін желі тұрақты күйге (w_i салмақтар қадам бойынша енді өзгермейді) келеді, не «циклдан» шықпай қалады.

Тәжірибелік қолдану бір қабатты желілерде де бар, мысалы, толық байланысты Хопфилд желісі, онда барлық нейрондар бір бірімен байланысқан.

Теорияда кері таратуы (қабаттар арасындағы сигналды) бар нейрон желісі қалаған функцияны қандай да дәлдікпен аппроксимациялауға мүмкіндік береді. Егер тәуелсіз X кірулердің саны бірнеше мыңнан аспайтын болса, тәжірибеде бұл мүмкін.

Егер шығу мәндерінің векторы кіру вектордың сығылған немесе бұрмаланған бейнесі болып алынса, онда автоассоциативті нейрон желі туралы, кері жағдайда – гетероассоциативті туралы айтады.

ЖИ саласында нейроинформатика бойынша зерттеулер бүгін ең озық деп саналады. Нейрон желілері тәжірибелік тұрғыда экономикалық, әскери, өндірістік жүйелерде кең қолданылады. Аппараттық жағынан нейрон желілері арнайы микросызба түрінде жүзеге асырылады. Мысалы, осындай

микросызбалар әуе қарсылы қорғаныс жүйелерде пайдалынады, олар қашықтығы 300 км астам «радар экранындағы белгісіз объекті» танып айыру есепті шешеді. Бірақ, нейрон желілерінің мүмкіндіктері математикалық дәлелденбеген. Осы себептен, мысалы әскери қызметшілері, осындай көздеу жүйеге толық сене алмайды.

9 - БӨЛІМ

САРАПТАМАЛЫҚ ЖҮЙЕЛЕР

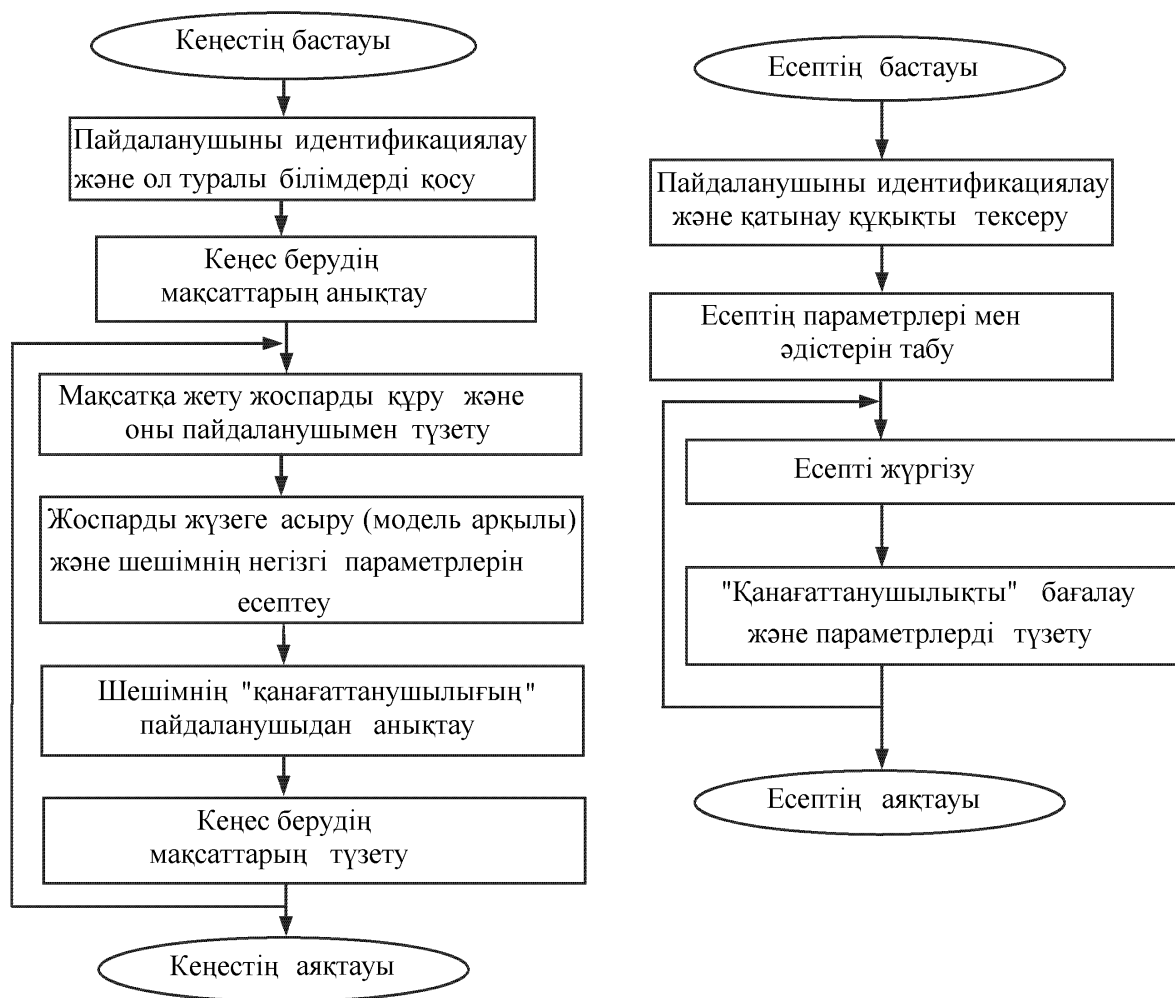
9.1. «Сараптамалық жүйе» анықтамасы

Жасанды интеллект әдістерінің кең тәжірибелік мәлімі «Сараптамалық жүйе» деп аталатын ерекше бағдарлама жасауымен өмірге келді.

Осындай бағдарламалар өткен ғасырдың 60 жылдары бірінші пайда болып, белгілі бір заттық саласы үшін жасанды «ақыл-ойды» жасауға арналған болатын. Мысалы, MYCIN бағдарламасы гематология саласында «дәрігер» ретінде ойланған, CASNET – көз аурулар саласында; ELIZA бағдарламасы «психолог» рөлін атқаруға арналған болатын және т.б.

60-70 жылдары сараптамалық жүйелердің көбі ең жақсы сарапшылардың (кейбір заттық салада) тәжірибесін жинауға мүмкіндік беретін бағдарламалық механизмі болып ойланатын. Сол кезеңдегі жасақталған деректер қорылары бұл рөлді атқаруға келмеген, өйткені олардың іздестіру тілі аса икемді емес; одан басқа ДҚ арналған іздестіру тілінің рекурсивтігі (яғни өзін өзі шақыру мүмкіндігі) болмаған.

Сараптамалық жүйенің қазіргі түсінуі – бұл тар мамандандырылған бағдарламалық кешен. Ол не өте тез стандартты шешімдерді (көбінесе, техникалық объектілерді басқару үшін) қабылдауға, не пайдаланушымен сұхбат негізінде, кейбір шешімнің таңдауына көмектесуге мүмкіндік береді (пайдаланушы ұсынатын варианттарға баға беру жолымен және өзінің шешу варианттарын ұсынуы мен оларды түзету (9.1-сурет)).



9.1-сурет. Сараптамалық жүйенің пайдалануымен әрекеттесу технологиялары және дәстүрлі сызба

9.2. Сараптамалық жүйелермен жұмыс істеу технологиясы

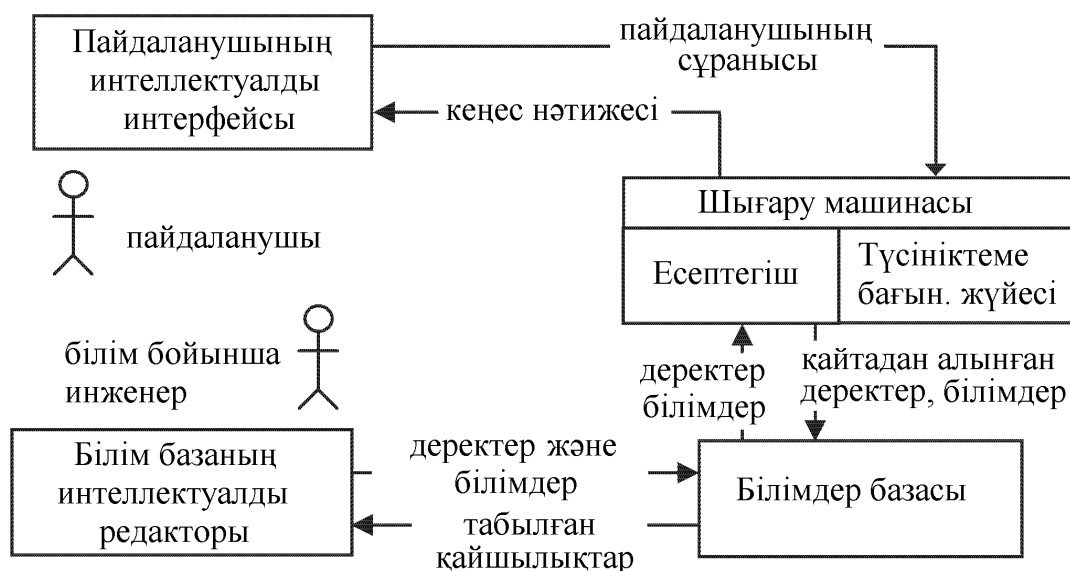
Сараптамалық жүйенің көмегімен қолданбалы есептің шешуі бұл әрқашан мақсатқа бейімділік итеративті процесс (9.1-сурет), себебі оның белгілі бір көзқарасына сәйкес ең жақсы шешімді табу мақсаты бар.

Егер сараптамалық жүйені қолданбай (басқа бағдарламалық құралдар арқылы) шешімді табатын болсақ, онда бұл бір реттік есеп болып табылады.

Сараптамалық жүйемен «сараптамалық емес» жүйелерінің арасындағы негізгі айырмашылығы – біріншісі әрбір итерациясында заттық салада жаңа алгоритм пайдаланады, бұл алдағы қадамның алгоритмның параметрлерімен ғана емес, құрылымымен де ерекшеленеді. Яғни қойылған мақсатқа жету кезінде мақсат және оның параметрлері де өзгеру мүмкін. 9.1-суретте сараптамалық жүйесі және оларсыз жұмыс кезеңдерінің салыстырмалы сызбалары көрсетілген.

9.3. Сараптамалық жүйелердің құрылымы

9.2-суретте абстракты сараптамалық жүйенің жалпы құрылымы берілген. Нақты сараптамалық жүйелер осындай құрылымға көбінесе, толық сай бола алмайды, себебі нақты сараптамалық жүйеде, әрбір заттық саласы үшін сипатты, қосымша блоктар өте көп болады. Сонда да жұмыстың жалпы технологиясы және СЖ негізгі құрылым элементтерінің міндеті түсінікті болу мүмкін.



9.2-сурет. Абстракты сараптамалық жүйенің жалпы құрылымы

Сараптамалық жүйенің басқа бағдарламалардан басты құрылымдық айырмашылығы – бұл білімдер қорының бар болуы және оқып үйрену мен өзін үйрету қабілетінде. Оның нақты түрі таңдалған ұсыну моделіне қатты тәуелді, бірақ жалпы түрде оған әрқашан фактілі және алгоритмді бөліктер кіреді.

Енді өнімдер ережелері жүйесі үшін біз фактілер және ережелер (логикалық, көбінесе дедуктивты қорытынды) туралы айтып отырамыз.

Мысал. Айтып салу: крокодил (табандары, тістері, құйрығы, түсі).

Ереже. крокодил (X, Y, Z, U), егер X = «қысқа», Y = «үлкен», Z = «ұзын», U = «жасыл».

Білімдерді ұсынудың фреймды моделіне негізделген, жасанды интеллект жүйесі үшін фактілер мен әдістер (қосылған процедуралар) туралы әңгіме жүргізіледі.

Мысал. Фрейм «экрандағы сызық»:

- басының координатасы 100, 100;
- аяғының координатасы 200, 200;
- көрші сызықтың координатасы белгісіз.

Әдістер:

- орта арқылы саздылы айналу;

- жоқ болу;
- көріну.

Бұл жағдайда сұрақ шығу мүмкін: «Экранда салынған сызықтың қандай интеллектуалдығы бар?». Жауабы мұндай: жасанды интеллектте фреймдер көбінесе, кейбір процесті немесе құбылысты модельдеу құралы ретінде пайдалынады, сондықтан экрандағы сызық тірі организмнің және т.б. моделі болуы мүмкін. Айналу кезінде ол басқа сызықтармен соғысады, нәтижесінде, мысалы «көрші сызықтың координатасы» алаңы толтырылады және т.с.с. (жүйе үйренеді).

Желілік моделі үшін жалпы жағдайда білімдер қоры: жүйенің күйін (мысалы, желі түйінділеріндегі кейбір параметрдің мәнін), таңдалған белсенді бағыныңқы желіні және шығару ережелерді ұстайды (оларға кіреді желі ішіндегі қатынастар сипаттамасы).

Шығару машинасы – БҚ алгоритмдік бөлігін өзектіктіру ретіне және әдісіне жауапты бағдарлама. Өртүрлі білімдерді ұсыну моделі үшін ол, әрине өртүрлі.

Логикалық моделі үшін оның түрі қарапайым тізбекті автомат болып табылады. Бұл модельде әрбір уақыт сайын барлық логикалық теңдеулердің іріктеуі толық өткізіледі.

Өнімдер жүйесі (логикалық шығару машина) үшін оның жағдайға тәуелді, детерминдық немесе ықтималдық бағаға негізделген артық көру жүйесі бар. Бұл модельде кезекті өзектіктіретін ереженің таңдауы бір мәнді болмайды.

Фреймды моделі үшін шығару машинасы екі бөліктен тұрады. Жалпы бөлігі фреймдер жүйесінде өтетін оқиғалар барлық фреймдерге бір уақытта немесе тізбекті түрде белгілі болсын дегеніне жауапты. Екінші бөлігі – бұл фреймдер өзінің оқиғаларын өңдеуші. Олар болған оқиға туралы алынған хабарға қандай әдістің қосу қажетін шешеді.

Түсініктеме бағыныңқы жүйесі – бұл сараптамалық жүйенің маңызды элементтерінің біреуі, ол оның тәжірибелік пайдалану тиімділігін анықтайды. Түсініктеме бағыныңқы жүйесін жүзеге асыруда есептегіш қолдайтын процедураларды пайдаландықтан (бірақ, басқа ретте), шығару машина есептегіштен және түсіндіру бағыныңқы жүйесінен тұрады деп санауға болады. Машина не адам алған шешім өзі емес, ал шешімнің жолы – процесі жиі бағалы болады. Түсініктеме бағыныңқы жүйесі – бұл «Неге осылай?» немесе «Неге осылай, неге былай?» сұраққа жауап алатын бағдарламалық механизм.

Мұндай қабілет (түсініктеме бағыныңқы жүйесінің бар болуы) интеллектуалды жүйелердің барлығында бола алмайды. Бұл бағдарламау күрделігіне және білімдерді ұсыну моделінің кемшіліктеріне байланысты. Түсініктеме бағыныңқы жүйесі тек өнімдер ережелеріне және семантикалық желіге негізделген жүйелерде болады.

Мысал. Интеллектуалды жүйеге қойылған пайдаланушының (интерфейс арқылы шығару машинаға) сұранысы: «Ертеңге доллар курсы қандай болады?».

Жүйе диалогы.

1. Саяси тұрақтылығы қандай? Оператордың жауабы: «Жоғары».
2. Экономикалық болжам қандай? Оператордың жауабы: «Қолайлы».
3. Биржада белсенділік қандай? Оператордың жауабы: «Төмен».

Жүйенің жауабы: «Болжалды курс 145,7 тнг.»

Жүйе түсініктемесі.

1 ереже: Саяси тұрақтылығы жоғары және болжам қолайлы болғандықтан, 1 параметр = тұрақтылық.

2 ереже: 1 параметр тұрақты және биржадағы белсенділік төмен болғандықтан, 2 параметр = өсуі төмен.

3 ереже: 2 параметр = өсуі төмен болғандықтан, курс = бүгінгі курс + кішкене кездейсоқ ауытқуы.

Білімдер қорының *интеллектуалды редакторы* – бұл білімдер қорынындағы фактілер мен ережелерді қосу, жою және өзгертуге мүмкіндік беретін бағдарлама. «*Интеллектуалды*» дегеніміз, білімдер қорының тұтастығын, дұрыстығын және қайшылықсыздығын қамтамасыз ететін қабілет.

Жеке алынған білімдер элементінің мағынасы жоқ, сондықтан білімдер қорына қандайда операцияны (жазыларды қосу, жою, өзгерту) жасау үшін бүкіл білімдер қорының тұтастығын, дұрыстығын және қайшылықсыздығын берілген тереңдікке тексеру қажет.

Мұны мысал арқылы түсіндірейік. Телефон анықтамалық деген кесте бар болсын. Бастапқы кілт «Тегі, аты + телефон» (9.1-кесте).

9.1-кесте

Тегі, аты	Телефон	Мекенжай
Аманов С.	31-05-85	Абай к., 45-60
Сарсенов Қ.	65-78-41	Ауэзов к., 71-23
...	...	...
Тастенов А.	45-17-80	Желтоқсан к., 12-84

Енді жаңа жазуды енгізейік «Аманов С. / 31-05-86 / Абай к., 45-60». Егер енгізуде қателік (мысалы, телефон 31-05-85) шықса ДҚБЖ (СУБД) қабыл алынбайды – себебі бастапқы кілт бұзылған. Ал егер он әртүрлі тектерді терген кезде қателісіп отырсақ (мысалы, бәрін бір телефонға жазайық «31-05-85»), онда ешқандай хабар алмаймыз, бір пәтерде 11 адам тұрмайды ғой! Білімдер қорына жаңа мәлімет қосқанда, «қосылған» деп аталатын процедура белсенділі болу керек, ол жүйедегі бар болған ақпаратпен салыстырып, енгізілген ақпараттың дұрыстығын тексереді.

«Берілген тереңдік» дегенді толығырақ қарастырайық. Себебі, өзінің табиғаты бойынша білімдердің рекурсивтігі («ішке салуы») бар болады. Білімдер қорында болғанда, білім элементтері таң ғаларлық және адамға түсініксіз мағыналық тізбектер жасайды. Қандайда қосылған жаңа элемент

олардың арасында «көпір» сияқты бола алады, сол себептен одан үлкен жаңа тізбектердің пайда болуына әкеледі. Осындай тізбектің шетінде заттық сала туралы кейбір жаңа қорытынды болады. Бұл жаңа қорытынды бұрынғы қорытындымен қайшы болып шығу мүмкін. Осындай қайшылықтар интеллектуалды жүйенің негізгі жұмыс тәртібі, себебі олардың пайда болуы фактілер мен ережелерді тексеруге және дәлдеуге мәжбүр етеді. Және де жүйеге енгізген жаңа білімнің (көпірдің) дұрыстығын және дәлдігін бағалауға келтіреді.

Мысал 1. Жүйеде жазулар бар болсын:

«Қайрат Майраның әкесі»;

«Майра Болаттың шешесі».

Сіз «Болат Төкеннің немересі» жазуды енгізуге тырысасыз. Жүйе 2 тереңдікте қайшылықты табады, өйткені бар болған мәлімет бойынша «Болат Қайраттың немересі»! Жүйе дәлдеуді қажет етеді: «бұл сол Болат па». Егер олар әртүрлі болса, онда жүйе білімдер қорын өзгертеді:

«Қайрат Майраның әкесі»;

«Майра Болат Амановтың шешесі»;

«Болат Қасенов Төкеннің немересі».

Немесе қателікті түзетеді:

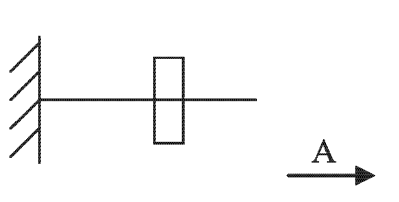
«Қайрат Майраның әкесі»;

«Майра Болаттың шешесі»;

«Болат Қайраттың немересі».

2-мысал. Сызбада көрсетілген (9.3-сурет) техникалық объект бар.

Қатты бітеуде гайкасы бар штанга орнатылған. Сіз «Апаттан кейін «А» бетінде цилиндрлық пішіні бар бірнеше фрагменттер табылған, мүмкін – үзілген гайкалар» деген жазуды енгізуге тырысасыз.



9.3-сурет. Сараптамалық жүйеде бейнеленген объект сызбасы

Жүйеде сызба шамамен былайша сипатталған:

қосылған (қатты бітеу №1, штанга №1);

қосылған (штанга №1, гайка №1);

кішкентай (гайка №1);

алты бұрышты (гайка №1);

цилиндрлық (штанга №1);

ұзын (штанга №1);

іші қуыс (штанга №1);

қасында (штанга №1, «А»).

Жаңа жазу қайшылықты шақырады.

1. «Бір қираған гайканың ұсақ фрагменттері аса көп болмау керек». Дұрыс па?
2. Әйтпесе «фрагменттер қираған штанганікі».
3. Әйтпесе «фрагменттер сырттан түскен».

Қорытындылары сөзсіз шығады деп ойлап қалуға болады, бірақ олай емес. Мысалы, автомобиль туралы әңгіме болсын. Майын ауыстырған кезде бөлшектердің қандай фрагменттері табылғаның айталасыз ба?

Қаралған мысалдарда қатынастардың транзитивті қасиеті бар болды.

Пайдаланушының интеллектуалды интерфейсы – бұл оптималды түрде мақсатқа жету үшін пайдаланушымен сұхбатты ұйымдастыруға жауап беретін бағдарлама.

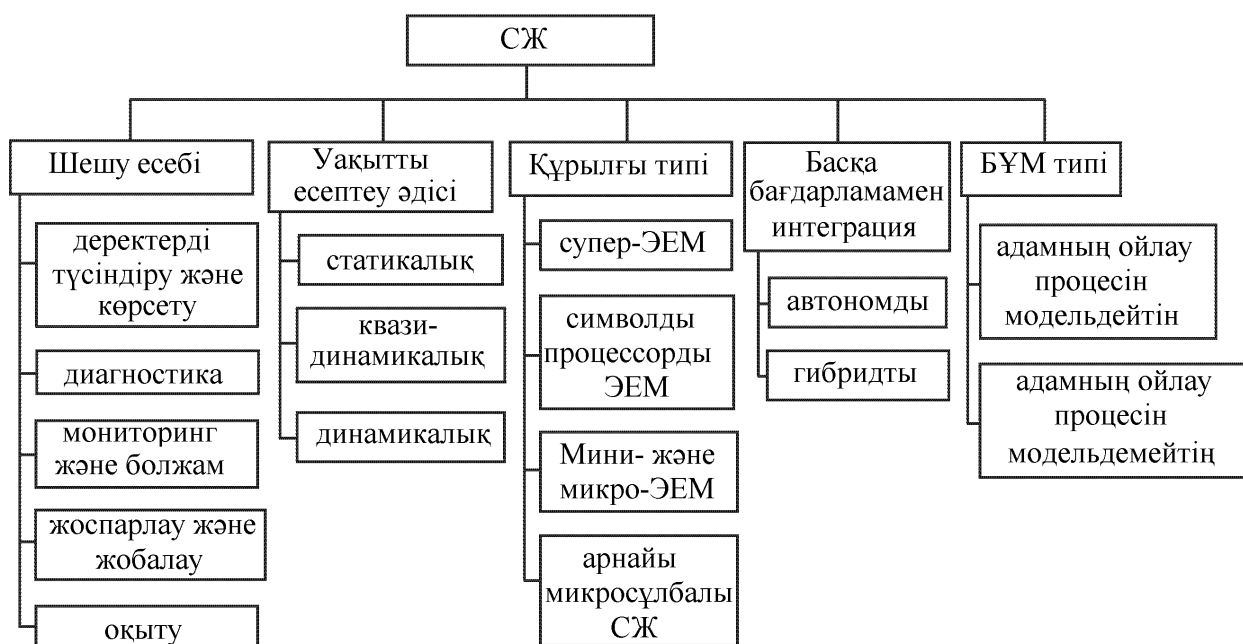
Сұхбат барысында жүйе пайдаланушыға сараптамалық жүйемен кеңесудің мақсатын дәлдеуге немесе өзгертуге көмектеседі, сондықтан ол интерфейс «интеллектуалды» деп аталады. Осындай қажеттілік болу керек, өйткені сараптамалық жүйе көмегімен модельденетін қазіргі объектілер, процестер, құбылыстар өте үлкен және күрделі болғандықтан, пайдаланушы олардың аттарын (атауларын) да параметрлерін де есте сақтай алмайды, сондықтан сұранысты сауатты қою және жауапты алу мүмкін емес.

9.4. Сараптамалық жүйелердің жіктеуі

Сараптамалық жүйелерді әртүрлі негізі бойынша топтастыруға болады: пайдалынатын білімдер ұсыну моделінің типі бойынша, қолдану саласы, міндеті, тиімділік көрсеткіштері бойынша және басқа (9.2- кесте, 9.4-сурет).

Өзінің қорытындыларын дәйекті дәлдеуге мүмкіндік беретін модель *тереңдік* деп аталады. Тереңдік білімдерге: абстракциялар, бейнелер, аналогиялар жатады. Оларда заттық сала құрылымының ұғынуы, жеке ұғымдардың міндеті мен өзара байланысы, белгілі заңдар және теориялық негіздер ұсынылады.

Шалағай (сыртқы) білімдер көбінесе, зерттеу объектінің сыртқы шығуына жатады. Олар кейбір (көбінесе, өте ықтималды) жорамалдарды жасауға мүмкіндік береді. Шалағай білімдерге негізделген қорытындылар, тереңдік білімдер көмегімен алынған қорытындылар сияқты, берікті және бағалы болу мүмкін, бірақ олар дәйекті емес. Шалағай білімдер, әдетте, заттық салалар ұғымдарының арасындағы эмпириялық ассоциацияларға және себепті салдар қатынастарға тиеді. Көп жағдайда жасанды интеллектің моделі шалағай болып туады, сосын өзінің «пайдалығын» дәлелдегеннен кейін математикалық негіздемесін, интерпретация әдісін және модель жұмысының нәтижелер түсіндірмесін алады.



9.4-сурет. Сараптамалық жүйенің жіктеу варианттары

Мысал. Телефонды пайдаланған кезде біз шалағай білімдерге негізделеміз, яғни трубканы алып нөмірді теріп керекті абонетпен қосылуға болады. Сол кезде телефон иесіне телефонды байланыс құрылымы және телефон аппараттың сызбасы туралы терең білімдердің қажеті жоқ.

9.2-кесте

Білімдер ұсыну моделінің типі бойынша СЖ жіктеуі

Білімдер ұсыну моделі	Білімдер			
	Тереңдік бойынша жіктеу		Қаттылық бойынша жіктеу	
	тереңдік	шалағай	жұмсақ	қатты
Логика	-	+	-	+
Өнімдер	-	+	+	+
Фреймдер	+	-	+	+
Семантикалық желілер	+	-	+	+
Объекті бағытталған тілдер	+	-	+	+
Стохастикалық модельдер	Бұл модельдер ойлау процесті модельдемейді			
Критериялды таңдау тілдер				
Нейронды желілер				

Сараптамалық жүйелердің көбінде, қазір шалағай білімдер пайдалынады. Тереңдік ұсыныстардың енгізуі үлкен қуатты БҚ жасауға мүмкіндік береді, өйткені тереңдік білімдер қатты шалағай білімдерге қарағанда икемділеу және адаптивті келеді. Классикалық мысал – медицина, мұнда тәжірибелі дәрігерлердің тереңдік білімдері бір аурудың әртүрлі емдеу әдістерін тудыруға мүмкіндік ашады (аурудың жағдайына, оның жасына, дәрінің бар болуына және т.б. байланысты). Ал жас немесе тәжірибесі төмен дәрігер шалағай моделі бойынша жұмыс істейді: «Егер жөтел – онда жөтелден таблетка ішу керек, егер ангина – онда эритромицин ішу керек» және т.с.с.

Тереңдік білімдер заттық саланың бастапқы ұғымдарын кейбір абстракты құрылымға жинақтау арқылы пайда болады.

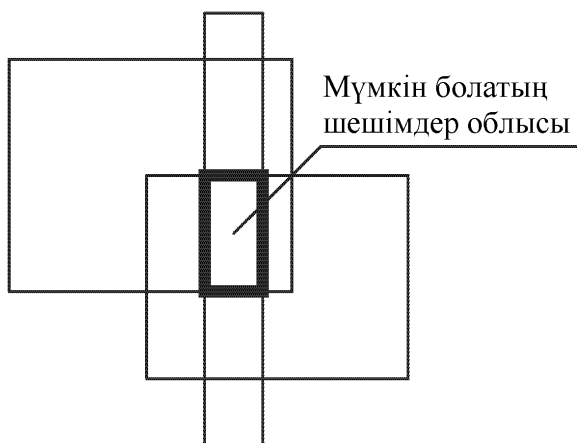
Білімдердің «қаттылығы» және «жұмсақтығы» берілген бастапқы жағдайларда бірімді, айқын ұсыныстарды алу мүмкіндікті немесе көптік, бұлдыр шешімдер мен ұсыныстарды алуды көрсетеді. Осы жіктеуге сәйкес заттық саланың өзін де қатты мен жұмсаққа жатқызуға болады. СЖ дамуының қазіргі тенденциясы - қатты шалағай модельдерден жұмсақ тереңдік модельдерге ауысу.

9.4.1. Шешілетін міндеті бойынша жіктеу

Интерпретациялау және көзбен шолу сараптамалық жүйелер көбінесе, экономикалық жүйелерде қолданылады. Себебі, кәсіпорынды басқару барысында пайдалынатын есептеуіш коэффициенттерді талдау да, есте сақтауға да өте қиын. Ал коэффициенттер мәндерінің талдауы аса күрделі емес, өйткені кейбір коэффициенті мәні берілген шектен аспау керек және олар бір-бірімен өзара байланысты болады.

Осындай жағдайда мәліметтерді көрнекті түрде ұсыну өте маңызды, яғни көзбен шолу. Ең қарапайым вариант – диаграммалар (діңгекті, дөңгелекті, нүктелі және т.б.) жасау.

Бірақ, егер коэффициенттер белгілі тәуелділікте болса, одан күрделіден көзбен шолуды пайдалану қажет (9.5-сурет). Егер 9.5-сурет үш баған цифрлерден тұрса, онда шешімді тез қабылдауға қиын болар еді.



9.5-сурет. Бірнеше шектеулерге қанағаттанған шешімді таңдау

Көзбен шолу әрқашан бола бермейді. Ең қарапайым вариант – кеңістіктің өлшемдігі үштен артық болса. Мысалы, цифрлердің төрт бағаны болсын, бірақ егер бағандардың арасында өзара тәуелділері болса, онда бұл төрт өлшемді кеңістік емес. Тәжірибеде бастапқы мәліметтердің көбі «тәуелсіз мәнді» болмайды.

Бұл жағдайда цифрлер бағандарды толық түсініктемемен ауыстыру қажет, мысалы, мәтіндік түрде. Түсініктемелер қысқа болу керек және олардың ішінде себеп пен салдар, байланыстар, мүмкін шешімдер туралы

қорытындылыр болу қажет. Интерпретациялау есептерді шешу үшін математикалық және статистикалық мәліметтерді талдау теориясы бар.

Ең бірінші жасалынған сараптамалық жүйелер - диагностика жасайтын жүйелер. Мониторинг жүргізетін СЖ бірге, олар техникалық объектілердің үздіксіз диагностикасын жасау үшін арналған, мысалы, ракеталарды іске қосуда және ұшу барысында. Жалпы айтқанда, осындай СЖ объекті кейбір класқа келтіру және ақаулықтың орын мен түрін табуды орындайды. Ақаулық – бұл нормадан ауытқу. Сондықтан осындай баян ету техникалық жүйедегі жабдықтың ақаулығын, тірі организмнің ауруын және әртүрлі табиғи аномалияларды бірыңғай теориялық көзқараспен қарауға мүмкіндік береді.

Мониторинг жүргізетін СЖ ерекшелігі - жүздеген не мыңдаған датчиктерден үздіксіз түсіп тұратын көп параметрлерді еске алуда. Осындай жағдайда уақыттың нақты масштабында көбінесе, жұмыс істеу керек, сонда дабыл сигналын дәл кезінде беруге болады.

Жоспарлау және жобалау СЖ көзбен шолу және интерпретация жүйелермен көп ортағы бар. Мысалы, жаңа автокөліктің электржабдықтарының сызбасын талдап, осындай жүйе сымдардың механикалық бұзылу орындарын көрсете алады. Осы есепті шешу үшін арнайы «когнитивті» графика, варианттарды таңдау әдістері және ерекше әдістемелер пайдаланылады, оған тәжірибелі инженерлер жүйені үйретеді. Қазіргі СЖ мұны сценарийлер моделі көмегімен орындайды. Егер жоба экономика саласында жүзеге асырылса, онда сызбалар орнына біз бизнес-жоспарды қолдаймыз.

Болжам жасайтын СЖ берілген жағдайдан ықтимал салдарды шығарады. Болжамдайтын жүйеде әдетте, параметрлік динамикалық модель пайдалынады, онда параметрлердің мәндері берілген жағдайға «сыналастырылады».

Оқыту СЖ қандай да пәнді оқып үйрену кезінде ЭЕМ көмегімен қателіктерді анықтайды және дұрыс шешімдерді көрсетеді. Олар оқушы туралы білімдерді жинайды (сипатты қателерін, нашар мағлұматын) және оларды жоятын сәйкес құраларды табады.

Жалпы жағдайда білімдерге негізделген жүйелерді екі топқа: *талдау* есептерді шешетін және *синтез* есептерді шешетін жүйелерге бөлуге болады. Олардың арасындағы негізгі айырмашылығы мынадай - талдау есептерде шешімдердің көптігін атап өтіп, жүйеге қосуға болады, ал синтез есептерде шешімдердің көптігі потенциалды және компонент немесе проблема шешімдерінен құрылады. Талдау есептеріне: мәліметтерді интерпретациялау, диагностика; синтез есептері: жобалау, жоспарлау жатады.

Құрастырылған есептер: оқыту, мониторинг, болжам жасау. Автономды СЖ пайдаланушымен кеңес беру тәртібінде тек «сарапшылық» есептер үшін жұмыс істейді, оларды шешу үшін мәліметтерді өңдеу дәстүрлі әдістерді (есептеу, модельдеу және т.б.) қолдамауға болады.

Гибридты СЖ бұл бағдарламалық кешен, олар стандартты қолданбалы бағдарламалардың пакеттерін (мысалы, математикалық статистика, сызықты бағдарламалау немесе мәліметтер қорын басқару жүйелер) құрастырады. Бірақ, осындай жүйелерді жасау өте күрделі.

Сараптамалық жүйелердің жасауының қазіргі күйі мұндай: диагностика есептері, интерпретация, жобалау —> мониторинг, оқыту; статикалық —> динамикалық; автономды —> гибридты.

Статикалық СЖ, есепті шешу кезінде білімдер қоры және мәліметтер өзгермейтін, заттық салада жасалынады. Олар тұрақты болады. Мысалы, автокөліктің ақаулықтарың диагностикалау.

Квазидинамикалық СЖ, кейбір қойылған уақыт барысында өзгертін жағдайды интерпретациялайды (түсіндіреді). Мысалы, микробиологиялық СЖ. Зертханалық өлшемдер технологиялық процесс барысында 4-5 сағатта бір рет алынады және алынған көрсеткіштердің динамикасы талданады.

Динамикалық СЖ, есепті шешу кезінде өзгертін мәліметтермен, кейде нақты уақыт тәртібінде жұмыс істейді. Мысалы, икемді өндірістік жүйелер, аурухана палатасында мониторинг және тағы басқалар.

Кейбір жақсы танымал СЖ 9.3-кестеде көрсетілген.

9.3-кесте

Кейбір жақсы танымал интеллектуалды жүйелер

Жасау орыны	Атауы	Міндеті	Жұмыс істеу принципі	Ерекшеліктері
Стэнфорд университеті, 1965-1983	DENDRAL	Химиялық құрылымдарды тану	Ережелер	Қазіргі түрі GENOA деп аталады
Стэнфорд университеті, 1972-1980	MYCIN	Қан ауруларды емдеуде кеңес беру	Өнімдер ережелері (500-дей). Дұрыс диагнозы - 70%	LISP тілі
Стэнфорд университеті, 1977-1980	CENTAUR	Тыныс бұзуын талдау	Гибридты модель: өнімдер ережелері және фреймдер	Фреймды және өнімді әдістемелерді салыстыру үшін жасалынған. Екі бөліктен тұрады
Стэнфорд университеті, 1975-1984	MOLGEN	Молекулярлы биологияда зерттеу	Фреймды модель	Ең күрделі жүйелердің біреуі
Корпорация DEC, 1985	XCON	Компьютердің кескін үйлесімін таңдауға көмектеседі	Бірнеше мыңдаған ережелер. Тура қорытынды	Қазір де пайдаланылады, өйткені компьютердің бөлшектер саны өте көп
Рэдиан корпорейшн, АҚШ	WILLARD	Ауа райын болжау	Өнімдік модель	Ауа райы туралы көп жылғы мәліметтерге негізделген ережелер
Intelligent Applications Ltd.	AMETHYST	Айналмалы бөлшектері бар машиналардың	Өнімдік модель	Дірілдеу датчиктармен интерфейс

		ақаулықтарын диагностикалау		
Compaq	SMART	Тәжірибелі емес клиенттердің телефон арқылы сұрақтарына жауап беруге көмектеседі	Өнеге моделіне негізделген	Жағдайларды (өнегелер) жүйеге келтіру арнайы жүйесі бар
Португалия, гидроэлектрстанция	CONTROL OF PLANT (Alarm system)	Автоматты хабар беру жүйеден ақпаратты талдайды (100 астам әртүрлі қауіп-қатерлер)	Өнімдік модель + арнайы математикалық модель	Нақты уақыт жүйесі.
GTE Labs inc.	COMPASS	Телефон жабдықтарында ақаулықтарды анықтау	Өнімдік модель + өнеге	Телефон станциясының жабдықтарына қосылған.

9.5. Сараптамалық жүйелерді жасау технологиясы

9.5.1. Құрастырудың негізгі кезеңдері

Дәстүрлі бағдарламалық қамсыздандыру типтерін жасау технологиясы: талаптарды талдау, жобалау, бағдарламалау және күйге келтіру, тестілеу кезеңдерінен тұрады. Интеллектуалды жүйені жасау технологиясын жүзеге асыруында бір қатар ерекшеліктері болады (9.4-кесте).

Кестеден көрініп тұр – бірнеше жаңа кезеңдер пайда болады. Барлық кезеңдерді толық қарастырайық.

1. Талаптарды талдау арқасында негізгі ішкі мәндер және олардың арасындағы мүмкін болатын өзара байланыстар анықталады. Бұл кезеңде кәдімгі сұраныстар және оларды қолдану шектері ескертіледі. Мысалы, тұрғындарға қызмет ету орталығының деректер қорына негізделіп, СЖ туысқандық байланысты талдап, «туыс ағайындарды» табу білу керек.

2. Заттық саланың талдауы барлық мүмкін ішкі мәндер мен олардың арасындағы арақатынастарды қарыстыруды, заттық саланың шектеуін және ішкі мәндер мен арақатынастардың мүмкін болатын типтерін математикалық бейнелеуін орындайды. Мысалы, тұрғындарға қызмет ету орталығының деректер қорында мүмкін болатын ішкі мәндер: «еркек», «әйел», «үй», «пәтер», «көше». Арақатынастар: «күйеу», «зайып», «қызы», «ұл», «іні», «күйеу бала» және т.б. Нақтылау дәрежесі СЖ мүмкіндіктерін анықтайды.

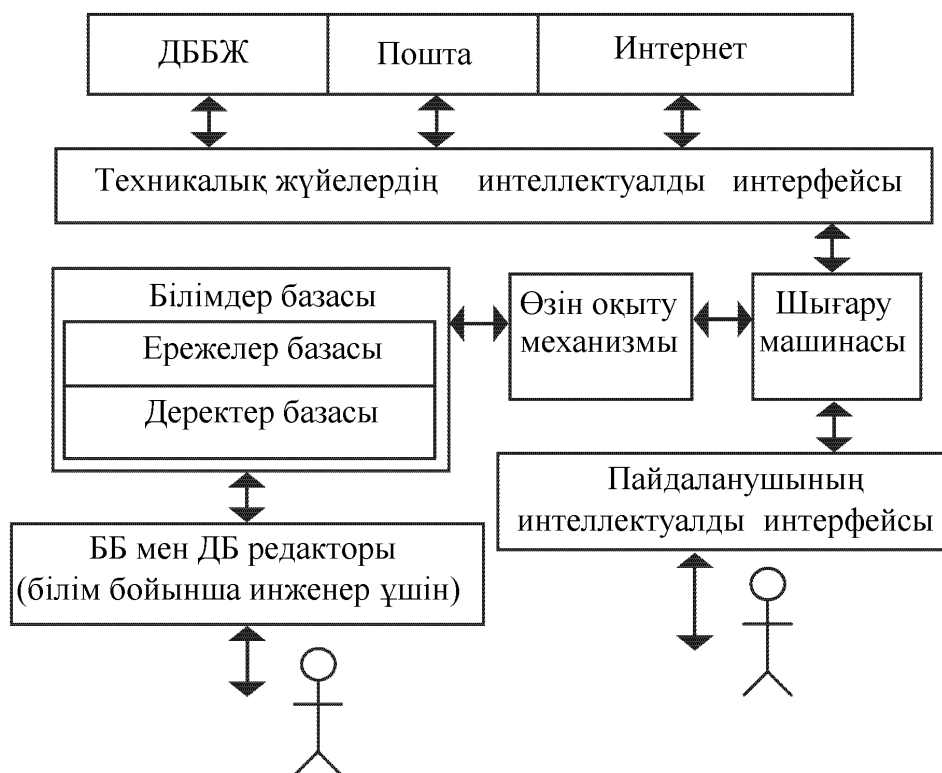
9.4-кесте

Сараптамалық жүйенің жасау кезеңдері

Кезең атауы	Шешілетін есептер	Мерзім
1. Талаптарды талдау	Проблеманы таңдау; СЖ жұмыс тәртібін анықтау; СЖ жұмыс нәтижелерінің адекваттылығына және дәлдігіне қойылатын талаптарды анықтау	2 – 4 апта
2. Заттық саланы талдау	Шешілетін есепке тікелей қатысатын заттық саланың мәнін бөліп шығару; шешілетін есепке тікелей қатысатын	Жарты жылға дейін

	фактілер мен ережелерді алдын ала бөліп шығару	
3. Жобалау 3.1. түп тұлғаны жасау; 3.2. баға беру	Білімдер ұсыну моделін алдын ала таңдау; логикалық құрылымын алдын ала жасау; программалық құрылымын алдын ала жасау; алдын ала кодтау және сынау (қысқа мерзімде); жобалауға немесе талаптарды талдауға қайту	4 – 8 апта
4. Бағдарламалау және күйге келтіру 4.1. кодтау; 4.2. өнеркәсіптігіе келтіру; 4.3. СЖ түйісу	Бағдарламалау тілді таңдау; кодтау; интерфейсты, көмек қызметін және құжаттаманы жасау; кәсіпорынның технологиялық процесіне СЖ енгізу; білімдер қорын толтыру жолымен алдын ала оқыту	4 – 8 апта
5. Оқыту	Нақты есепті шешу тәртібінде СЖ жұмыс істеуі (білім бойынша инженердың көз салуымен)	Жарты жылға дейін
6. Эталонмен салыстыру	Эксперимент жүргізу және жоба мақсатына жету дәрежесін анықтау	1 апта
7. СЖ қолдау	Логикалық және бағдарламалық қателіктерді іздеу, жаңа версияларды шығару, өзгерген пайдалану шарттарына бейімделу	5 жылға дейін

3. СЖ жобалауы СЖ құрылымының жасауын шақырады. СЖ жобалауының мәнді ерекшелігі – құрылымның таңдап алынған білімдерді ұсыну моделіне тәуелсіздігінде. Мысалы, негізгі өнімдік модельдің таңдауы автоматты түрде ережелер редакторын; фреймдық модельдің таңдауы – объектілер редакторын; ықтималдық модельдің таңдауы – ықтимал редакторын пайдалануды білдіреді. Барлық үш жағдайда білімдер қорының редакторы қажет, бұл СЖ кейбір абстракты құрылымын көрсетуге мүмкіндік береді (9.6-сурет)



Сонымен СЖ жұмыс механизмін келесі қадамдармен бейнелеуге болады:

1. Білім бойынша инженер білімдер қорын (БҚ) толтырды (СЖ оқытты);
2. Пайдаланушы жүйеге сұрақ қойды;
3. Интеллектуалды интерфейс сұрақты дәлелдеді және шығару машина үшін мақсатты тұжырымдады;
4. БҚ алынған білімдер негізінде, қойылған мақсатқа жету үшін машина талаптанды;
5. Мәліметтер кемді болғандықтан мақсатқа жетпеді;
6. Техникалық жүйенің интеллектуалды интерфейсы сәйкес мәліметтер көзімен байланысты;
7. Шығару машина мақсатқа жетуді қайтадан талаптанды;
8. Қайшы қорытындылар алғандықтан мақсатқа жетпеді;
9. Пайда болған қайшыларды шешу үшін өз өзін оқыту механизм қосылды;
10. Өз-өзін оқыту механизмнің жұмыс барысында алынған жаңа білімдер, БҚ редакторына түсті;
11. Білім бойынша инженер жаңа білімдер мен фактілерді қабылдады не қабылдамады;
12. Жаңа білімдер мен фактілер ДҚ және БҚ енгізілді;
13. 4 пунктке қайту.

СЖ басқа бағдарламалық құралдардан басты айырмашылығы – бұл білімдер қорының бар болуы, оның ішінде білімдер кейбір білімдерді ұсыну тілде (БҰТ) жазулар жиынтығы түрінде сақталынады. БҰТ мамандарға (сараптамалық жүйені әзірлеуші) түсінікті нысанда білімдер қорын оңай өзгертуге және толықтыруға мүмкіндік береді. Соңғы уақытқа дейін СЖ жасауда орталық проблемасы әртүрлі БҰТ болатын. Қазір білімді ұсынудың ондаған тілі немесе моделі бар. Ең үлкен тараған келесі модельдер: өнімдер, семантикалық желі, фреймдер, 1 ретті предикаттарды есептеу, объект-бағытталған тілдер және т.б. Модельдің таңдауы нақты заттық салада білімдердің құрылымымен анықталады. Білімдер қорын жасау және БҰТ таңдау алдында құрылымды анықтау жүргізіледі. Білімдер элементінің негіздеу және нақтылау заттық саланың мамандары (сарапшы) тікелей көмегімен жүзеге асырылады. Бұл процесс *білімдерді үзінді алу* (шығару) деп аталады. Білімдерді шығару мен және құрылымдау мен айналысатын сараптамалық жүйені әзірлеушілерді *білімдер бойынша инженерлер* деп атайды.

9.5.2. Кейбір тәжірибелік ұсыныстар

Сараптамалық жүйені әзірлеу дәстүрлі бағдарламаны жасаудан бірнеше рет қиын. Сондықтан:

- есепті дәстүрлі әдістермен шешуге болатынын;

- есепті шешу қажеттілігі жиі пайда болмайтынын дәлелдеу қажет.

Тәжірибелі инженер осы бекітулерді қандай да проблемаға қатысты көрсете алады.

Ал сіздің қарсылыстарыңыз:

- кірістік мәліметтерде шу мен қайшылықтарды автоматты түрде айырып тану қажеттілігі;

- көп сан параметрлерді еске алу;

- сарапшы қолдайтын шешімді қабылдау әдістердің (эвристика) қарапайымдығы сияқты дәлелдерді келтіруі мүмкін.

Бірақ, оларға сенбеңіз. Егер шудың автоматты айырып тануы шынында оңай жүзеге асырылса, 2 мен 3 есептерді шешу мүмкін емес.

Дұрыс келіс – басынан өте қысаң есепті таңдау. Оның шешімі тұнық, түсінікті және бірнеше белгілі әдістерге сәйкес.

Сосын Сіздің бағдарламаңызды қолдайтын пайдаланушыны табу керек. Оны бірдемеге үйрету керек.

Енді Сізбен жұмыс істейтін сарапшыны табу керек. Бірақ, Сіз емес сарапшы Сізді іздеу керек.

СЖ жасауының келесі кезеңінде түп тұлғаны немесе түп тұлғалық жүйені құру қажет. Осы кезде минималды шығын ережесін сақтау керек. Түп тұлғаны жасауға және енгізуге көп уақыт пен қаражатты жұмсамау керек.

Түп тұлғалық жүйе фактілер мен байланыстардың кодтау дұрыстығын тексеруге арналған СЖ қысқартылған версиясы болып табылады. Оның көлемі – бірнеше ондаған ережелер, фреймдер немесе мысалдар.

Сосын, егер құптауға болатын түп тұлға жасалынса, жүйенің негізгі жұмыс тәртіптерін толық жетілдіру қажет. Ол үшін қосымша фактілер мен ережелерді енгізу керек, яғни өнеркәсіптікке дейін пысықтау. Бастапқы кодтар, әдетте, қайтадан жазылады. Егер бұрынғы кезеңдерде «бос СЖ» пайдаланса, онда кейбір қажетті модульдерді қосып жазады (9.7-сурет).



9.7-сурет. СЖ құрастыру мысал циклы

10 - БӨЛІМ

САРАПТАМАЛЫҚ ЖҮЙЕЛЕРДІ ЖАСАУДЫҢ АСПАПТЫҚ ҚҰРАЛДАРЫ

10.1. Жасанды интеллект үшін бағдарламау тілдер

Қазіргі уақытта ЖИ негізгі тілі болып табылады LISP, Prolog және C++. Алғашқы екеуі – қысаң мамандандырылған тілдер, ЖИ үшін арнайы әзірленген. Соңғысы – жалпы арнаулы аспаптық тіл, ол ең кең тараған кәсіптік тілі деп саналады.

LISP пен Prolog-ң қолдануы есептің шешу логикасына зейін қоюға мүмкіндік береді, ал C++ қолдануы – бағдарламаның жоғары жұмыс жылдамдығына жету және стандарты емес өңдеу процедураларын жүзеге асыруға мүмкіндік береді. Қазіргі есептеуіш техниканың сипаттамалары өте жоғары, сондықтан C++ қолдануы тек интерфейс пен тыс-бағдарламаларды жасау үшін пайдалы.

Бос сараптамалық жүйе (басқаша, тыс-бағдарлама немесе матрица) – бұл шығару механизмі, фактілерді енгізу және редакциялау құралдары бар бағдарлама. Матрицаны өзіндік фактілермен толтырып, ережелерді редакциялап, ол зерттелетін проблема туралы қорытындыны жасай бастайды.

ЖИ тілдер қатарында SmallTalk, Forth, LOGO, РЕФАЛ, ПЛЭНЕР (Planner) айта кетуге болады.

10.2. Prolog туралы қысқа мәліметтер

Prolog (логикалық программалау) тілі 1974 жылы Марсель (Франция) университетінде А.Колмаэро (Colmerauer Alan) жасаған. Кейбір мамандар Prolog-ты оқу тілі деп санайды, ал бірінші орынға LISP тілін шығарады. Әрине, LISP тілінде бағдарлама өте қысқа, бірақ Prolog-бағдарламалар түсініктілеу. Prolog-ң бірнеше түрлі: Turbo Prolog, Visual Prolog және т.б. версиялары бар.

Prolog-бағдарламаның барлық айнымалылары үлкен әріптен жазылады: X, Maіra. Егер айнымалының мәні қажетті болмаса, онда *бос айнымалы* «__» пайдаланылады. Мәні анықталмаған айнымалы *еркін* деп аталады. Басқа мәліметтер төменгі кестелерде көрсетілген.

10.1-кесте

Constants	Константаларды жариялау
[GLOBAL] domains	Стандартты емес немесе құрамалы мәліметтер типін жариялау секциясы. Болмау да мүмкін
[GLOBAL] database	Ішкі мәліметтер қорымен жұмыс істеу үшін міндетті емес предикаттарды жариялау секциясы
[GLOBAL] predicates	Предикаттарды жариялау секциясы
Clauses	Ережелер мен фактілерді жариялау секциясы
Goal	Ішкі мақсатты жариялау секциясы. Болмау да мүмкін

10.2-кесте**Мәліметтер типі**

Symbol	жүйенің ішкі символдар кестесіне енгізілген жол
String	ұзындығы 64 Кбайт-қа дейін символдар тізбегі
Char	1-байтты символдар
Integer	2-байтты таңбасы бар бүтін сандар
Real	8-байтты қалқыма нүктесі бар сандар
Ref	мәліметтер қорының сілтеме сандары
Reg	микропроцессордың регистры: AX, BX, CX, DX, SI, DI, DS және ES
File	файл

10.3-кесте**Операциялар**

Арифметикалық	+, -, *, /, mod, div
Реляционды	>, <, =, >=, <=, <>, ><
Математикалық функциялар	sin, cos, tan, arctan, ln, log, exp, sqrt, round, trunc, abs
Логикалық	and («»), not, or, ! (кесіп тастау)

10.4-кесте**Енгізу – шығару**

Write	экранға шығару
Read	консольден оқу

Параметрлер:

%d – кәдімгі ондық сан (chars және integers)
 %u – таңбасы жоқ бүтін сан (chars және integers)
 %R – мәліметтер қорының сілтеме саны (database reference numbers)
 %X – ұзын 16-ретті сан (strings, database reference numb)
 %x - 16-ретті сан (chars және integers)
 %s – ағындар (symbols және strings)
 %c - символ (chars және integers)
 %g – ең қысқа форматта жазылған нақты сандар (default for reals)
 %e – экспоненциалды түрде жазылған нақты сандар
 %f – бекітілген нүктесі бар форматта жазылған нақты сандар
 %lf – (fixed reals)
 \n – жана жол
 \t - табуляциялау
 \nnn – nnn коды бар символ

10.5-кесте**Turbo Prolog-ң стандартты предикаттары**

Стандартты предикаттар алып түседі				
Енгізу / шығару	Файлдық жүйе	Экранмен жұмыс істеу	Терезелі жүйе	Жолдарды өңдеу
Типтерді өзгерту	МҚ жұмыс істеу	Графикамен жұмыс істеу	ОЖ жұмыс істеу	Төмен деңгейлі

Бағдарламаның мысалы (Turbo Prolog тілінде):

```
domains
    person, activity = symbol
predicates
    likes (person, activity)
clauses
    likes (ellen, tennis)
    likes (john, football)
    likes (torn, baseball)
    likes (bill, X) if likes (torn, X)
goal
    likes (X,Y)
```

Жұмыс нәтижесі (айнымалылар мәндерінің қатынастар есімі бойынша айқындауы орын алды):

```
X = ellen      Y = tennis
X = john Y = football
X = torn Y = baseball
X = bill Y = baseball
```

10.3. Lisp туралы қысқа мәліметтер

1961 жылы профессор Джон Маккартнидың тобымен жасалынған (Стэнфорд, АҚШ). LISP (List processing) қысқартуы «тізімдерді өңдеу тілі» деп аударылады. 1970-1980 жылдары бұтақ тәріздес құрылымға негізделген есептерді шешу үшін, мысалы, лабиринтті іздестіру және «генетикалық программалау» қолданған (Стэнфорд, проф. Джон Коз). Көп версиялардың ішінде ең танымалы – COMMON LISP. 1990 жылдардан кейін қолданудан шықты.

Геометриялық пішіндерді сипаттауына өте ыңғайлы болғандықтан, LISP тілі қайтадан пайдалана басталды. Қазір автоматтандырылған жобалау жүйелерде кеңінен пайдалынады, мысалы AutoLisp (AutoCAD жүйесі).

Lisp-бағдарламасында қатты құрылымы жоқ. Ол Lisp интерпретатордың кіреберісіне дәйекті келіп түсетін s–айтылулардың тізбегі болып табылады (яғни берілген грамматикадағы символды айтылулар).

10.6-кесте

Мәліметтер типі

Бүтін сан	Мысалы, 4	
Нақты сан	Егер нүктесі бар болса – 4.5	
Символдық атом	Айнымалы деген ұғымның аналогы. Айнымалының мәні сөзсіз айнымалының өзіне тең	s(symbolic) деп аталатын айтылулар
Тізім	Белгіленеді (), мысалы (A, B, C)	
Ішкі құрылған функция	Тізім төменде берілген	
Примитив	Тек AutoLisp үшін	

Кейбір функциялар үшін, мысалы CAR және CDR, қысқартылған вариант бар:

C-R, мұнда символдың орнына D немесе A символдарды қоюға болады.

Мысалы:

$(CAAAR\ X) \Leftrightarrow (CAR(CAR(CAR\ X)))$ немесе $(CADR\ X) \Leftrightarrow (C\ AR(CDR\ X))$

Бағдарламаның мысалы (μ-LISP тілінде):

Факториалды есептеу:

```
(defun factorial (n)
  (cond ((=n 1) 1)
        (t (*n (factorial (-n 1))))))
```

10.7-кесте

Кейбір негізгі функциялар

Атауы	Қысқа сипаттамасы
QUOTE A	Функция өзінің аргументін бағаламайды және оны нәтиже ретінде шығарып береді
+, -, *, /	Арифметикалық функциялар (тек сандар үшін)
SETQ және SET	Меншіктену функциялар
CAR A	Тізім болатын өзінің аргументін (A) бағалайды және мән ретінде осы тізімнің бірінші элементін шығарып береді
CDR A	Тізім болатын өзінің аргументін (A) бағалайды және мән ретінде осы тізімді шығарып береді, бірақ бірінші элементсіз
CONS A B	Аргументтерді бағалайды, бірінші аргументтің мәні түрлі s-айтылуы болу мүмкін, ал екіншінің мәні тізім болу керек. Функцияның нәтижесі басы A және құйрығы B тізім болып табылады
APPEND A B	A мен B мәндері тізімдер болсын: (A...A) және (B...B), онда APPEND функцияның мәні тізім болады: (A...A B...B)
NULL X	Аргументті бағалайды және T-ны шығарады, егер оның мәні NIL болса, ал керісінше, NIL шығарады. Функция NOT аналогы болып табылады.
ATOM X	Аргументті бағалайды және T-ны шығарады, егер оның мәні атом болса, ал керісінше NIL шығарады.
NUMBERP X	Өзінің аргументін бағалайды және T-ны шығарады, егер аргументтің мәні сан болса, ал керісінше, NIL шығарады.
LISTP X	Өзінің аргументін бағалайды және T-ны шығарады, егер аргументтің мәні тізім болса, ал керісінше, NIL шығарады. ATOM функциясына қарама-қарсы болып табылады.
EQ A B	Екі атомды салыстыру функция
EQUAL A B	Түрлі s-айтыларды салыстыру
COND A A	Шарттарды тексеретін функциялар
DEFUN	Жаңа функцияны анықтайтын функция

ГЛОССАРИЙ

Абсурд – мағынасыздық
Актуальность - өзектік
База знаний – білімдер қоры
Визуализация – көзбен шолу
Воображение – қиял
Восприятия – түйсік
Выборка - іріктеу
Высказывания – айтып салулар, айтылулар
Высказывание категорическое – үзілді-кесілді айтылу
Отделение - бөліп алу
Группировка – топтасу
Детализация – нақтылау
Доказательство – дәлелдеу
Запрос – сұраныс, сұрау салу
Извлечение – үзінді алу
Интерпретируемость - ішкі түсіндірушілік
Исключения – шығарып тастау
Исчисление высказываний – айтып салуларды есептеу
Модель представления знаний - білімдер ұсыну моделі
Модель прецедентов - өнеге моделі
Мышление – ойлау
Наследование – мұралану
Обоснованный – дәйекті
Объединения - бірлестіктер
Опровержение – мәлімдеу
Отношения - қатынастар
Отрицание – мойындамау
Пересечения - қиылысулар
Поверхностный – шалағай, сыртқы
Повествовательный – хабарлай айтылған, хабарлы
Подкласс - класс тармағы
Подсеть – бағыныңқы желі
Подсистема объяснений - түсініктеме бағыныңқы жүйесі
Подсознание - соқыр сезім, түйсік
Поиск – іздестіру, іздеу
Положение - қағида
Постусловие – кейінгі шарт
Посылка - бастама
Правила продукций - өнімдер ережелері
Предметная область - заттық сала
Предок – ата-тек
Представление знаний - білімдерді ұсыну

Принадлежность - тиістілік
Продукционная модель - өнімдік моделі
Прототип – түп тұлға
Ранжирование – дәрежеліу
Распознавание образов - бейнені айырып тану
Рассуждение – пікірлесу
Резолюция- қарар
Решатель - есептегіш
Сознание – сана
Структурированность - құрылымдылық
Суждение - пікір
Сущности - ішкі мазмұндар
Точка зрения – көзқарас, тұрғы
Узкоспециализированный - тар көлемді
Умственный - ақыл-ой
Экспертная система - сараптамалық жүйе
Язык представления знаний (ЯПЗ) - білімдерді ұсыну тілі (БҰТ)

Пайдаланылган әдебиеттер

1. Братко И. Программирование на языке Пролог для искусственного интеллекта: Пер. с англ. - М.: «Мир», 1990. - 560 с.
2. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. - СПб.: «Питер», 2000. - 384 с.
3. Девятков В. В. Системы искусственного интеллекта. - М.: Изд-во МГТУ им. Баумана, 2001. - 352 с.
4. Лорьер Ж.-Л. Системы искусственного интеллекта. Пер. с франц. - М.: Мир, 1991. - 568 с.
5. Люгер Д. Искусственный интеллект: стратегии методы решения сложных проблем. – М.: «Вильямс», 2003. – 864 с.
6. Марселлус Д. Программирование экспертных систем на Турбо Прологе: Пер. с англ. - М.: «Финансы и статистика», 1994. – 256 с.
7. Нейлор К. Как построить свою экспертную систему: Пер. с англ. - М.: Энергоатомиздат, 1991. - 286 с.
8. Потапов А.С. Технологии искусственного интеллекта – СПб: СПбГУ ИТМО, 2010. – 218 с.
9. Рассел, Стюарт, Норвиг, Питер. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. - М.: Издательский дом "Вильямс", 2006. - 1408 с.
10. Смолин Д.В. Введение в искусственный интеллект: конспект лекций. - М.: Физматлит, 2004. - 208 с.
11. Сойер Б., Фостер Д.Л. Программирование экспертных систем на Паскале. - М.: «Финансы и статистика», 1990. - 191 с.
12. Стобо Д.Ж. Язык программирования Пролог: Пер. с англ. - М.: «Радио и связь», 1993. – 368 с.
13. Тейз А., Грибомон П., Луи Ж. Логический подход к искусственному интеллекту: От классической логики к логическому программированию. - М.: «Мир», 1990. - 432 с.
14. Шампандар А. Искусственный интеллект в компьютерных играх. - М.: Издательский дом "Вильямс", 2007. - 768 с.
15. Элти Дж., Кумбе М. Экспертные системы: концепции и примеры. — М.: «Финансы и статистика», 1987. - 192 с.

ЖАСАНДЫ ИНТЕЛЛЕКТ НЕГІЗДЕРІ

МАЗМҰНЫ

Кіріспе 3

1-БӨЛІМ

ЖАСАНДЫ ИНТЕЛЛЕКТІҢ ТҰЖЫРЫМДАМАЛЫҚ НЕГІЗДЕРІ 5

- 1.1. Жасанды интеллект теориясының даму тарихы 5
- 1.2. Бағдарламалы құралдардың даму кезеңдері 5
- 1.3. «Жасанды интеллект» ұғымы 6
 - 1.3.1. Терминдер және анықтамалар 8
 - 1.3.2. Жасанды интеллекттің қазіргі зерттеу салалары 12
 - 1.3.3. Жасанды интеллекттің қазіргі теориялық проблемалары 12

2-БӨЛІМ

ЖАСАНДЫ ИНТЕЛЛЕКТІҢ НЕГІЗГІ ТЕОРИЯЛЫҚ МІНДЕТТЕРІ 15

- 2.1. Жасанды интеллект — зерттеудің пәнаралық саласы 15
- 2.2. Жасанды интеллекттің дәстүрлі есептер тізбесі 15
 - 2.2.1. Есептердің тым қатал тұжырымдамалары 16
 - 2.2.2. Жасанды интеллекттің міндетін тұжырымдау ережелері 17
 - 2.2.3. Есепті шешудің параметрлерін таңдау 17
 - 2.2.4. Шешу (білімдер ұсыну) моделін таңдау 18

3-БӨЛІМ

ЖАСАНДЫ ИНТЕЛЛЕКТ ӘДІСТЕРІНІҢ ҚОЛДАНУ САЛАЛАРЫ 19

- 3.1. Жақсы және нашар құрылымданған заттық салалар 19
- 3.2. Білімдер ұсыну модельдері 22

4-БӨЛІМ

БІЛІМДЕР ҰСЫНУДЫҢ ЛОГИКАЛЫҚ МОДЕЛІ 23

- 4.1. Логикалық модельдердің түрлері, жалпы ұғымдар мен анықтамалар 23
- 4.2. Формалды (Аристотель) логикасы 24
 - 4.2.1. Есімдер 29
 - 4.2.2. Айтып салулар 34
 - 4.2.3. Дәлелдеу және мәлімдеу процедуралары 37
- 4.3. Формалды логиканың математикалық жүзеге асыруы 39
 - 4.3.1. Автоматты түрде теоремаларды дәлелдеу әдістері (предикаттарды есептеу) 41
 - 4.3.2. Л.Заде айқын емес көптіктер теориясының элементтері 42

5-БӨЛІМ

БІЛІМДЕР ҰСЫНУДЫҢ ӨНІМДІК МОДЕЛІ 44

- 5.1. Заттық саланы ережелер мен фактілер көмегімен бейнелеу 44
- 5.2. Ені бойынша толық іріктеп алу әдісі 48
- 5.3. Тереңдік бойынша толық іріктеп алу әдісі 49
- 5.4. Күйлер кеңістігінде іздестірудің эвристикалық әдістері 50
- 5.5. Бағыныңқы есепке бөліп есепті шешу әдістері 51
- 5.6. Есепті «ЖӨНЕ-НЕМЕСЕ» граф түрінде ұсыну 51
- 5.7. Өнімдер жүйесін басқару 52
- 5.8. Өнімдік модельдің артықшылықтары мен кемшіліктері 53

6-БӨЛІМ

БІЛІМДЕР ҰСЫНУ ҮШІН ФРЕЙМДЕР 54

- 6.1. Кеңістік сахналарды талдау 58
- 6.2. Сөйлемнің мағынасын түсіну 60
- 6.3. Фрейм моделін тәжірибелік жүзеге асыру 61

7-БӨЛІМ

БІЛІМДЕР ҰСЫНУ ҮШІН СЕМАНТИКАЛЫҚ ЖЕЛІЛЕР 65

8-БӨЛІМ

БІЛІМДЕРДІ ҰСЫНУ ЖАҢА МОДЕЛЬДЕРІ 71

8.1. Критериалды әдістер 71

8.1.1. Таңдау есебі 72

8.1.2. Өлшеуіш шкалалар 79

8.1.3. Шешімді қабылдаудың кейбір әдістемелері 85

8.1.4. Шешім көпкритериалды қабылдауының оқу мысалы 87

8.2. Ықтималдық әдістер 89

8.3. Нейрон желілері 91

9-БӨЛІМ

САРАПТАМАЛЫҚ ЖҮЙЕЛЕР 97

9.1. «Сараптамалық жүйе» анықтамасы 97

9.2. Сараптамалық жүйелермен жұмыс істеу технологиясы 98

9.3. Сараптамалық жүйелердің құрылымы 98

9.4. Сараптамалық жүйелердің жіктеуі 103

9.4.1. Шешілетін міндеті бойынша жіктеу 105

9.5. Сараптамалық жүйелерді жасау технологиясы 108

9.5.1. Құрастырудың негізгі кезеңдері 108

9.5.2. Кейбір тәжірибелік ұсыныстар 110

10-БӨЛІМ

САРАПТАМАЛЫҚ ЖҮЙЕЛЕРДІ ЖАСАУДЫҢ АСПАПТЫҚ ҚҰРАЛДАРЫ 112

10.1. Жасанды интеллект үшін бағдарламау тілдер 112

10.2. Prolog туралы қысқа мәліметтер 112

10.3. Lisp туралы қысқа мәліметтер 114

Глоссарий 116

Пайдаланылған әдебиеттер 118