О. І. Гороховський

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

Міністерство освіти і науки України
Вінницький національний технічний університет

О. І. Гороховський

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

Навчальний посібник

Вінниця
ВНТУ
2010

УДК 004.89(075)
ББК 32.813я73
Г77

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 2 від 25.09.2008 р.)

Рецензенти:
В. М. Лисогор, доктор технічних наук, професор
В. А. Лужецький, доктор технічних наук, професор
В. І. Месюра, кандидат технічних наук, доцент

Гороховський, О. І.
Г77 Інтелектуальні системи : навчальний посібник / О. І. Гороховський. – Вінниця : ВНТУ, 2010. - 194 с.

В посібнику розглянуто основні поняття інженерії знань, розпізнавання зорових і мовних образів та побудови експертних систем.

Посібник розроблено відповідно до плану кафедри та програми до дисципліни "Інтелектуальні системи".

УДК 004.89(075)
ББК 32.813я73

ЗМІСТ

ВВЕДЕННЯ

Виникнення проблеми інтелектуалізації обчислювальних машин обумовлено з однієї сторони розвитком досліджень в напрямку «штучний інтелект» (ШІ), з іншого боку - швидким розвитком обчислювальної техніки і постійно зростаючими потребами її різноманітних застосувань.

Початок досліджень в галузі штучного інтелекту (кінець 50-х років минулого сторіччя) пов'язують із роботами Ньюелла, Саймона і Шоу, що досліджували процеси розв’язання різноманітних задач. Результатами їхніх робіт з’явилися такі програми, як ЛОГІК-ТЕОРЕТИК, призначена для доказу теорем у численні висловлювань, і ЗАГАЛЬНИЙ ВИРІШУВАЧ ЗАДАЧ [1-3].

Ці роботи поклали початок першому етапу досліджень в галузі штучного інтелекту, пов'язаному з розробкою програм, що вирішують задачі на основі застосування різноманітних евристичних методів.

Евристичний метод розв’язання задачі при цьому розглядався як властивий людському мисленню «узагалі», для якого характерне виникнення «здогадок» про шлях розв’язання задачі з наступною перевіркою їх. Йому протиставлявся використовуваний в ЕОМ алгоритмічний метод, що інтерпретувався як механічне здійснення заданої послідовності кроків, яка детерміновано приводить до правильної відповіді. Трактування евристичних методів розв’язання задач як сугубо людської діяльності й обумовили появу і подальше поширення терміна *штучний інтелект* (*artificial intelligence*) [4].

Так, при описі своїх програм Ньюелл і Саймон наводили як докази, які підтверджують, що їхні програми моделюють людське мислення, результати порівняння записів доказів теорем у вигляді програм із записами міркування «мислячої вголос» людини. На початку 70-х років вони опублікували багато даних подібного роду і запропонували загальну методику упорядкування програм, що моделюють мислення.

Приблизно в той час, коли роботи Ньюелла і Саймона стали привертати до себе увагу, у Массачусетському технологічному інституті, Стендфордському університеті і Стендфордському дослідницькому інституті також сформувалися дослідницькі групи в галузі ШІ. На противагу раннім роботам Ньюелла і Саймона ці дослідження більше відносились до формальних математичних уявлень. Способи розв’язання задач у цих дослідженнях розвивалися на основі розширення математичної і символічної логіки. Моделюванню же людського мислення надавалось другорядне значення. До дослідників цього напрямку можна віднести таких відомих в галузі ШІ вчених, як Мінський, Маккарті, Слейгл, Рафаель, Бобрі, Бенерджі й ін. [5-6].

На подальші дослідження в галузі ШІ значно вплинула поява методу резолюцій, запропонованого Робінсоном, заснованого на доказі теорем у логіці предикатів і що є, принаймні, теоретично вичерпним методом

доказу (хоча використання цього методу для розв’язання реальних задач пов'язано з більшими, іноді практично непереборними труднощами).

Методологічне значення робіт Робінсона й інших аналогічних робіт полягало в тому, що головна увага в дослідженнях з ШІ перемістилася з розробки методів відтворення в ЕОМ людського мислення на розробку машинно-орієнтованих методів розв’язання задач [7].

При цьому означення терміна «штучний інтелект» зазнало істотної зміни. Ціллю досліджень, проведених у напрямку ШІ, стало не моделювання способів мислення людини, а розробка програм, спроможних вирішувати «людські задачі». Так, один із значних дослідників ШІ того часу Р. Бенерджі в 1969 р. писав: «Галузь досліджень, що звичайно називається штучним інтелектом, мабуть, можна уявити як сукупність методів і засобів аналізу і конструювання машин, спроможних виконувати завдання, із якими донедавна могла справитися тільки людина. При цьому за швидкістю й ефективністю машини повинні бути порівнянні з людиною» .

Функціональний підхід до спрямованості досліджень зі штучного інтелекту зберігся, в основному, дотепер, хоча ще і зараз ряд вчених, особливо психологи, намагаються оцінювати результати робіт з ШІ з позицій їхньої відповідності людському мисленню [7-10].

Дослідницьким полігоном для розвитку методів ШІ на першому етапі були всілякі ігри, головоломки, математичні задачі. Деякі з цих задач стали класичними в літературі про штучний інтелект (задача про мавпу і банани, місіонерів та людожерів, Ханойську вежу, гра в 15 і ін.).

Вибір таких задач для досліджень обумовлювався простотою і ясністю проблемного середовища (середовище, у якому розгортається розв’язання задачі), її відносно малою громіздкістю, можливістю достатньо легкого добору і навіть штучного конструювання «під метод». У той же час такі середовища підходили для моделювання досить складних процесів розв’язання і дослідження різних стратегій розв’язання з відносно невеликими витратами як людських, так і машинних ресурсів.

Головний розквіт такого роду досліджень припадає на кінець 60-х років, після чого стали робитися перші спроби застосування розроблених методів для задач, розв'язуваних не в штучних, а в реальних проблемних середовищах. Проте такі спроби наштовхнулися на великі труднощі, обумовлені, головним чином, необхідністю моделювання зовнішнього світу. Ці труднощі були пов'язані з проблемами опису знань про зовнішній світ, організації їх збереження і достатньо ефективного пошуку, введення в пам'ять ЕОМ нових знань і усунення застарілих (у тому числі автоматичного їхнього вилучення із середовища), перевірки повноти і непротиріччя знань і т.п. Показані проблеми ще і сьогодні далекі від повного розв’язання, проте вже в той час ставало все більш зрозумілим, що саме їхнє розв’язання є ключем до створення ефективних систем штучного

інтелекту.

Необхідність дослідження систем штучного інтелекту при їхньому функціонуванні в реальному світі привела до постановки задачі створення інтегральних роботів (*integrated robots*). При розробці проектів таких роботів використання терміна «штучний інтелект» стало звучати більш обумовлено, тому що в них вирішувалися не окремі задачі ШІ, а досліджувався і реалізовувався необхідний спектр «інтелектуальних» функцій, таких як організація цілеспрямованої поведінки, сприйняття інформації (*the information*) про зовнішнє середовище, формування дій, навчання, спілкування з людиною й іншими роботами.

Для формування цілеспрямованої поведінки, тобто формування програми розв’язання деякої зовнішньої відносно робота задачі, інтегральний робот повинен мати необхідний комплекс знань про реальний світ, у якому він функціонує, що значно перевершує знання, відображені, власне, у програмі функціонування. Ці знання повинні бути закладені в роботі у виді моделі зовнішнього світу або, точніше, моделі проблемного середовища, тобто тієї частини зовнішнього світу, що істотна для розв’язання задач, які поставлені перед роботом. Модель проблемного середовища (*model of problem environment*) інтегрального робота - це сукупність взаємозалежних відомостей, необхідних і достатніх для розв’язання відповідного класу задач, у тому числі і відомостей про можливі способи впливу на середовище і зміни, які вони викликають у ній. У систему знань робота повинні бути закладені алгоритми, що дозволяють відтворювати «уявні» перетворення середовища і будувати на цій основі план розв’язання чергової задачі, а також алгоритми, що забезпечують виконання даного плану і контрольне порівняння очікуваних і дійсних результатів запланованих дій [8, 14-16].

Проведення робіт, пов'язаних із створенням інтегральних роботів, можна вважати другим етапом досліджень з штучного інтелекту.

У Стендфордскьому університеті, Стендфордскьому дослідницькому інституті і деяких інших місцях були розроблені експериментальні роботи, що функціонують у лабораторних умовах. Проведення цих експериментів показало необхідність вирішення кардинальних питань, пов'язаних із проблемою подання знань про середовище функціонування, і одночасно недостатню дослідженість таких проблем, як зорове сприйняття, побудова складних планів поведінки в динамічних середовищах, спілкування з роботами на природній мові.

Ці проблеми були більш-менш ясно сформульовані і поставлені перед дослідниками в середині 70-х років, пов'язаних із початком третього етапу досліджень систем ШІ. Його характерною рисою з’явився зсув центру уваги дослідників із створення автономно функціонуючих систем, самостійно (або в умовах обмеженого спілкування з людиною) розв’язуючих у реальному середовищі поставлені перед ними задачі, до

створення людино-машинних систем, що інтегрують у єдине ціле інтелект людини і здібності обчислювальних машин для досягнення загальної цілі – розв’язання задачі, поставленої перед інтегральною людино-машинною системою прийняття рішень [17, 18].

Такий зсув обумовлювався двома причинами:

1) на цей час з'ясувалося, що навіть прості на перший погляд задачі, що виникають перед інтегральним роботом при його функціонуванні в реальному світі (наприклад, прямування по пересіченій місцевості, розпізнавання об'єктів на складному фоні з природним освітленням, організація складної поведінки і т.п.), не можуть бути вирішені методами, розробленими для експериментальних задач у спеціально сформованих проблемних середовищах;
2) стало ясно, що поєнання доповнюючих одна одну можливостей людини й ЕОМ дозволяє «обминути гострі кути» шляхом перекладання на людину тих функцій, що поки ще недоступні для ЕОМ. Обчислювальна машина, із своєї сторони, здатна опрацьовувати великі обсяги інформації з використанням регулярних методів, багаторазово переглядати різноманітні шляхи розв’язання, запропоновані людиною, надавати їй всіляку довідкову інформацію.

На перший план висувалася не розробка окремих методів машинного розв’язання задач, а розробка методів і засобів, що забезпечують тісну взаємодію людини й обчислювальної системи протягом усього процесу розв’язання задачі з можливістю оперативного внесення людиною змін у ході цього процесу [19, 20].

Розвиток досліджень з ШІ в даному напрямку обумовлювався також різким зростанням виробництва засобів обчислювальної техніки і таким самим різким їхнім здешевленням, що робить їх потенційно доступними для більш широких кіл користувачів. Проте ця доступність для більшості реальних користувачів так і залишалася «потенційною», оскільки потребувала для реалізації оволодіння більшими обсягами спеціальних знань із використання ЕОМ.

Все це, разом узяте, і привело до того, що в даний час під інтелектуалізацією ЕОМ розуміється в основному розвиток можливостей обчислювальних машин у напрямку забезпечення спільного з користувачем розв’язання задач, спрощення процесу спілкування людини й ЕОМ у ході розв’язання, постійного розширення частки машини в спільній з людиною діяльності із розв’язання задачі. При цьому значна увага приділяється також і підвищенню здатності обчислювальної машини до самостійного (в автоматичному режимі) розв’язання важкоформалізовуваних задач [21-23].

1 НОВА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ

1.1 Необхідність нової технології розв’язання задач на ЕОМ

Для обчислювальної техніки 60—70-ті роки минулого сторіччя були періодом бурхливого розвитку різноманітних систем обробки інформації в різноманітних сферах застосування ЕОМ. У цей час були створені і випробувані методи і засоби реалізації таких систем, удосконалена технологія їхніх застосувань. Ще більш інтенсивно в цей самий період розвивалися засоби обчислювальної техніки. Різко підвищилися продуктивність, надійність ЕОМ, істотно збільшилися обсяги оперативної і зовнішньої пам'яті, розширилися номенклатура периферійних пристроїв, функціональні можливості системного програмного забезпечення. Все це відбувалося в умовах безупинної мініатюризації технічних засобів ЕОМ і зниження їхньої вартості.

Характерним результатом цих процесів стало створення на початку 80-х років персональних ЕОМ (ПЕОМ). В даний час ПЕОМ мають такі габарити, що вільно розміщаються на невеликому письмовому столі, при цьому її технічні характеристики і функціональні можливості багато в чому перевершують характеристики і можливості великих ЕОМ 60—70-х років. Інакше кажучи, обчислювальна техніка стала набагато доступнішою для користувачів, істотно розширилася сфера її потенційного застосування. Там, де колись використання ЕОМ було занадто дорогим і, отже, неефективним, тепер воно стає доцільним і бажаним. Більш того, як і для будь-якого промислового продукту, для ЕОМ напрямок і інтенсивність розвитку визначаються тим, наскільки широко, різноманітно й ефективно вона буде застосовуватися.

Проте до початку 80-х років розвиток засобів обчислювальної техніки і її застосування усе більше стримується традиційною технологією розв’язання задач на ЕОМ. Для пояснення цього явища необхідний докладний розгляд організації розв’язання задачі на ЕОМ і життєвого циклу програмного забезпечення.

Процес розв’язання задачі починається її початковою постановкою кінцевим користувачем (користувачем, що використовує результати її розв’язання). У переважній більшості випадків задача виникає в процесі цілеспрямованої діяльності користувача в галузі його професійних інтересів.

Необхідною умовою здійснення цієї діяльності й адекватного формулювання постановки задачі є наявність у користувача професійних знань, використовуючи які він може аналізувати події, що відбуваються в галузі його діяльності, прогнозувати розвиток тих або інших ситуацій, знаходити необхідні розв’язання.

Володіючи професійними знаннями, кінцевий користувач, за означенням, у загальному випадку не має ніяких (принаймні таких, що

могли б йому надати практичну допомогу) знань про способи використання обчислювальних засобів для розв'язання його задачі. Посередниками між ним і ЕОМ виступають системний аналітик і програміст.

Основними знаннями системного аналітика є знання про методи постановки різноманітних задач для розв'язання на ЕОМ. Крім того, він повинен мати деякі знання про галузь професійної діяльності користувача, що забезпечують можливість проведення з ним діалогу, і уявляти, хоча б на узагальненому рівні, процес складання програми. У процесі діалогу з користувачем системний аналітик вибирає адекватний задачі метод, визначає необхідну обраним методом організацію даних і будує схему розв'язання задачі на ЕОМ. При цьому, у випадку якщо задача ставиться до інформаційно-пошукового типу, головну увагу він повинен приділяти саме організації даних. Вже після побудови схеми розв'язання відбувається майже повна «відчуженість» кінцевого користувача від подальшого процесу розв'язання задачі.

Етап розробки програми, здійснюваний програмістом або самостійно, або в діалозі із системним аналітиком, проходить без участі кінцевого користувача. Програміст повинен мати знання, достатні для побудови програм, що реалізують ті або інші схеми розв'язання задачі, а також деякі знання про методи розв'язання задач на ЕОМ, достатні для діалогу із системним аналітиком. Їхній діалог спрямований тільки на розробку й оптимізацію програми. Похибки, неточності, нераціональні розв'язання, допущені при постановці задачі і при побудові схеми її розв'язання, практично не можуть бути виявлені. При цьому іноді процес оптимізації може приводити до внесення таких змін у схему розв'язання задачі, що несуттєві з погляду системного аналітика, але неприпустимих із погляду користувача.

Налагодження програми здійснюється програмістом, причому одержуваний готовий продукт уже цілком «відчужений» від ініціатора його створення - кінцевого користувача.

Проте при традиційній технології внесення змін потребує в загальному випадку нового залучення програмістів і системних аналітиків, що повинні внести корективи в програму, що зазвичай являє собою більш складний процес, ніж розробка нової програми. На все це може знадобитися значний час, особливо якщо програміст і аналітик обслуговують декількох користувачі. При цьому можливості зміни програми значно обмежуються в першу чергу організацією прикладного програмного забезпечення в цілому, у другу - його настроюванням на конкретне застосування. Варто підкреслити також, що необхідність змін, що потребують такої перебудови, виникає не тільки і не стільки в межах розв'язання конкретної задачі. У більшості випадків вона є наслідком зміни виробничої, економічної і т. п. ситуацій, зміною поглядів групи

користувачів у процесі експлуатації системи, нарешті (і дуже часто), результатом неправильного розуміння творцями відповідних вимог кінцевих користувачів.

Як правило, прикладне програмне забезпечення компонується з незалежних програмних систем, кожна з яких призначена для розв’язання певного класу задач. Очевидні техніко-економічні розуміння змушують будувати ці програмні системи як універсальні в рамках розв'язуваних ними задач. Так, якщо програмна система призначена для розв’язання задач оперативного керування підприємством із дискретним характером виробництва, то вона повинна бути прийнятною і для годинникового, і для підшипникового, і для електромеханічного заводів. Для того, щоб врахувати специфіку конкретного застосування, при проектуванні програмної системи передбачаються і засоби її настроювання (адаптації). Залежно від складності настроювання відповідні засоби орієнтуються на застосування кінцевим користувачем або програмістом.

Після розробки й оформлення програмної системи як виробу вона надходить до користувачів. Проте подальша робота з нею із ряду причин не обмежується застосуванням для розв’язання прикладних задач.

Як показує досвід, складна програмна система звичайно містить похибки, не виявлені на стадіях налагодження і тестування. Подібні похибки можуть бути внесені на будь-якій стадії створення системи. Зокрема, нерідко бувають випадки, коли виявляються похибки проектування системи, обумовлені тим, що розробник неправильно зрозумів постановку задачі або вимоги користувачів, або не передбачив необхідні засоби адаптації системи до умов конкретного застосування. Ще частіше зустрічаються технічні похибки реалізації, що виявляються в невідповідності програмного продукту експлуатаційної документації або нероботоздатності системи при деяких варіантах її застосування. Виправлення подібних похибок, інформація про які надходить до розробника від користувачів, і розвиток функціональних можливостей системи є предметом діяльності розроблювача, яку називають *супроводом програмної системи.* Якщо теоретично і можна припустити існування «безпомилкової» програмної системи (на практиці таке трапляється вкрай рідко), то передбачити всі наступні зміни постановки задачі, вимог до її розв’язання в більшості випадків просто неможливо, тому що вони змінюються в процесі застосування системи і почасти в зв'язку з її застосуванням. Ці зміни визначаються, головним чином, природним розвитком предметної галузі, умов, у яких задача вирішується. Звідси і випливає необхідність постійного розширення функціональних можливостей системи.

Діяльність розробника із виправлення похибок у програмній системі і розширення її функціональних можливостей правильно було б називати централізованим супроводом системи. Поряд із ним завжди існує

необхідність локального супроводу кожного користувача. Головна задача тут полягає в адаптації системи до умов конкретного застосування, які змінюються. При цьому використовуються або вхідні в систему засоби настроювання, або (коли вони не можуть допомогти) додаткові програми, що розроблюються супровідними програмістами.

Таким чином, супровід прикладної програмної системи виконується протягом усього її життєвого циклу. Процес супроводу в традиційній технології потребує, принаймні, такої ж кількості ресурсів, як і розробка програми. Зокрема, подвоюється число фахівців із програмного забезпечення, що обслуговують потреби користувачів. У умовах неухильного розширення сфери застосування ЕОМ потреби в таких фахівцях не можуть бути реально задоволені. Все сказане вище й обумовлює необхідність зміни технології використання ЕОМ.

1.2 Головні ідеї нової технології розв'язання задач на ЕОМ

Описану кризову ситуацію в застосуванні ЕОМ можна подолати тільки шляхом залучення користувачів до процесів розв'язання задач, супроводи програмної системи і, можливо, навіть розробки *прикладного програмного забезпечення* (ППЗ - *the applied software*).

Проте це потребує корінної зміни принципу організації ППЗ і методів його використання при вирішенні задач, що склалися в рамках традиційної технології.

Насамперед, необхідно будувати програмні системи таким чином, щоб радикально спростити процеси їхньої експлуатації і супроводу. Для того, щоб глибше зрозуміти характер ускладнень користувача при його взаємодії з ЕОМ, потрібно підкреслити, що програмна система в традиційній технології звичайно ґрунтується на формальній моделі розв'язання задачі. Залежно від задачі це може бути будь-яка модель дослідження операцій, чисельний метод розв'язання прикладного математичного аналізу, деяка модель даних і т.п. При цьому, як правило, множина понять і терміни, у яких формулюється й описується задача для застосування програми, мінімальна і пов'язана з математичною моделлю, а не з конкретною галуззю її застосування. Таке становище обумовлене прагненням до універсалізації програмних систем, тобто до можливості значного поширення кожної такої системи в різних предметних галузях. У той же час кожна предметна галузь характеризується системою змістовних понять, якими оперує користувач при розв'язанні задачі.

Таким чином, у традиційній технології обробки даних системи понять предметної галузі і формальної моделі, покладеної в основу програми, як правило не збігаються.

Це розходження і є головною причиною ускладнень, що виникають при взаємодії користувача з ЕОМ у процесі розв'язання задачі.

Для застосування програми користувач повинен перекласти

постановку задачі, виражену в системі понять предметної галузі, у постановку, виражену в системі понять формальної моделі.

Переклад з однієї системи понять у другу називається інтерпретацією. При одержанні результатів розв’язання задачі користувач також повинний виконати інтерпретацію, обернену першій.

Процес інтерпретації пов'язаний із рядом істотних об'єктивних і суб'єктивних труднощів, що збільшуються із зростанням обсягу, складності й універсальності програмної системи.

Методи вирішення проблеми інтерпретації в рамках традиційної технології можна умовно розділити на приватні і системні.

Приватні методи покращують процес взаємодії користувача з ЕОМ лише в окремих аспектах. До них можна віднести, наприклад, ефективні методи документування програмних систем, різноманітного роду системи допомоги користувачеві (help-системи) і т.п.

Головна концепція систематичних методів спрощення процесу взаємодії складається в тому, щоб одночасно забезпечити:

- взаємодію на основі системи понять предметної галузі для жорстко обмеженого кола прикладних задач;
- оперативне розширення цього кола задач супровідним програмістом.

У традиційній технології систематичні методи підтримуються використанням різноманітного роду діалогових систем з архітектурою «меню» і засобів генерації таких систем, а також деяких сучасних мов програмування. Такі системи одержали значне поширення і швидко розвиваються. Проте усім їм органічно притаманний один недолік - необхідність регулярного супроводу програми для кожного кінцевого користувача. Нова інформаційна технологія (*new information technology*) ставить своєю ціллю забезпечення простоти процесу взаємодії користувача з ЕОМ із винятком необхідності регулярного супроводу.

Основна ідея нової інформаційної технології, покликаної забезпечити вирішення проблеми, складається в тому, щоб розглядати систему понять предметної галузі і відповідність між нею і системою понять формальної моделі як вихідну інформацію для розв’язання прикладних задач.

У ідеалі подібний підхід повинен забезпечити користувачеві можливість самостійної зміни системи понять предметної галузі, визначення нових понять через «відомі» системі. Користувач одержує можливість формулювання свого бачення предметної галузі, виділення в ній об'єктів і взаємозв'язків, істотних для розв’язання задачі і зручних для міркування в процесі розв’язання. Описаний підхід до проблеми протилежний традиційному підходові, що розглядає задачу спрощення взаємодії користувача з ЕОМ як важливу, але усе ж другорядну.

Система програмних і апаратних засобів, що забезпечують для

кінцевого користувача, який не має спеціальної підготовки в галузі обчислювальної техніки, використання ЕОМ для розв'язання задач, що виникають у сфері його професійної діяльності або без посередників-програмістів, або з незначною їхньою допомогою, називається *інтелектуальним інтерфейсом* *(the intellectual interface)***. Процес же впровадження засобів інтелектуального інтерфейсу в обчислювальну техніку можна визначити як *інтелектуалізацію* ЕОМ.**

При цьому терміни «інтелектуальний» і «інтелектуалізація» не припускають обов'язкового означення поняття «інтелект», а тільки підкреслюють, з одного боку, нетривіальність функцій, виконуваних інтелектуальним інтерфейсом у загальній системі обчислювальних засобів, з іншого боку - те, що виконання цих функцій до останнього часу було прерогативою людини.

Функціонування засобів інтелектуального інтерфейсу опирається на розвинуті методи роботи зі знаннями: їхнє подання, збереження, перетворення і т.п.

Під терміном *знання (knowledge*) при цьому розуміється вся сукупність інформації, необхідної для розв'язання задачі, що включає в тому числі інформацію:

1) про систему понять предметної галузі, у якій вирішуються задачі;

2) про систему понять формальних моделей, на основі яких вирішуються задачі;

3) про відповідність систем понять, згаданих вище;

4) про поточний стан предметної галузі;

5) про методи розв'язання задач.

При цьому система знань повинна бути організована в ЕОМ таким чином, щоб забезпечити взаємодію обчислювальної системи (ОС) із користувачем у системі понять і термінів предметної галузі.

Знання про предметну галузь, організовані на основі тих або інших методів і засобів подання знань, називається моделлю предметної галузі (МПГ - *model of a subject domain*).

Накопичений досвід робіт з подання знань і створення окремих підсистем інтелектуального інтерфейсу дозволяє підтверджувати, що послідовне проведення нового підходу до проблеми організації взаємодії користувача з ЕОМ істотно впливає на усі види робіт зі створення, супроводу й експлуатації програмних засобів. Таким чином, справедливо говорити про виникнення дійсно нової технології розв'язання задач на ЕОМ - нової інформаційної технології.

При новій технології залежно від типу розв'язуваної задачі повинні забезпечуватися адекватні форми подання усіх видів інформації, якою обмінюються користувач і обчислювальна система.

Це означає, що засоби підтримки нової технології повинні забезпечувати подання інформації в природному для користувача вигляді -

тексти на природній мові, намальовані від руки зображення (у тому числі таблиці, графіки, рукописні символи), усні мовні повідомлення, різноманітні математичні вирази, а також інші подання професійними діалектами користувача. Надалі необхідне і забезпечення можливості формулювання комбінованих повідомлень, що поєднують різноманітні форми подання інформації.

При використанні нової інформаційної технології кінцевий користувач повинен затрачати мінімальний час на навчання. Саме навчання в основному повинно зводитися до оволодіння роботою з терміналом: «як дізнатися», «що натиснути», «куди сказати», «куди подивитися» - і ознайомлення з можливостями системи. При цьому користувач у випадку ускладнень повинен одержувати допомогу від системи, що дозволяє подолати виниклі труднощі. Варто підкреслити, що при цьому мова йде про «випадкового» кінцевого користувача. Для кінцевого користувача, що регулярно використовує ЕОМ у своїй діяльності, повинна бути розроблена більш широка система навчання, тобто і при новій інформаційній технології повинен зберігатися принцип: «Чим більше знаєш, тим більше можеш одержати (*the more know, the more will have*)».

Основна вимога до обчислювальної системи, що визначає нову інформаційну технологію розв'язання задач із використанням ЕОМ і узагальнює інші вимоги, - перетворення машини в зручного партнера кінцевого користувача при вирішенні задач, що виникають у ході його професійної діяльності. Зручність програмного продукту для користувача в загальному випадку є більш важливою якістю, ніж продуктивність ЕОМ.

Інтелектуальний інтерфейс, що забезпечує в складі єдиної людино-машинної системи прийняття рішення безпосередню взаємодію кінцевого користувача й ЕОМ при розв'язанні задачі, повинен виконувати три групи функцій:

- забезпечення для користувача можливості постановки задачі для ЕОМ шляхом повідомлення тільки її умов без задання програми розв'язання. При цьому повинна зберігатися можливість детальної розбивки задачі на послідовність підзадач, що забезпечує, у свою чергу, можливість непрямого вказання шляху розв'язання задачі;
- забезпечення для користувача можливості самостійного формування операційного середовища (*operational environment*) розв'язання задачі з використанням тільки термінів і понять з галузі професійної діяльності користувача;
- забезпечення для користувача природних для нього форм подання інформації, якою він обмінюється з ОС у процесі розв'язання задачі, а також вибір прийнятних для нього способів організації діалогу при цьому обміні;
- можливість зміни за бажанням користувача структури діалогу, при

цьому спектр можливих змін повинен передбачати діалог типу «меню», різноманітні форми анкетно-форматного типу, обмін довільними (не регламентованими заздалегідь) повідомленнями.

З непідготовленості користувача, його «випадкового» характеру випливає вимога забезпечення роботи ОС в умовах помилок, що припускаються користувачем у повідомленнях. Ця ж непідготовленість потребує включення до складу засобів інтелектуального інтерфейсу, що забезпечують процес спілкування, засобів, що пояснюють користувачу його помилки і незрозумілі місця в ході розв’язання задачі.

1.3 Місце інтелектуального інтерфейсу в обчислювальній системі

Для кращого розуміння місця, яке інтелектуальний інтерфейс повинен займати в загальній системі апаратних (*hardware*) і програмних (*software*) засобів ЕОМ, зробимо ретроспективний аналіз розвитку обчислювальної техніки. Такий аналіз дозволяє стверджувати, що обчислювальна техніка постійно розвивається в двох напрямках.

Перший напрямок пов'язаний із поліпшенням параметрів існуючих засобів, орієнтованих на підвищення ефективності виконання ними своїх функцій. Його можна назвати розвитком по горизонталі (*development on a horizontal*).

Другий напрямок визначають зміни в технології обробки інформації, що приводять до поліпшення використання ЕОМ. Розвиток у цьому напрямку відбувається в головному стрибкоподібно і пов'язаний з появою якісно нових апаратно-програмних засобів, що доповнюють існуючі засоби. Його можна назвати розвитком по вертикалі (*development on a vertical*).

Розвиток по вертикалі накладає свій відбиток на організацію обчислювальної машини, надаючи їй багаторівневий ієрархічний характер.

У сучасних ЕОМ, що використовують традиційну технологію обробки інформації, можна виділити такі рівні: мікропрограмний, традиційний машинний, операційної системи, асемблерний і проблемно-орієнтованих мов. Виділення цих рівнів ґрунтується на трактуванні обчислювальної машини як інтегрованого набору алгоритмів і структур даних, здатного зберігати і виконувати програми. Якщо при цьому обчислювальна машина побудована на реальних елементах, то вона трактується як реальна або апаратна. Якщо ж вона побудована за допомогою програм, виконуваних на деякій іншій обчислювальній машині, то вона є віртуальною машиною, що програмно моделюється. Віртуальна машина визначається структурою даних і алгоритмів, використовуваних у процесі виконання програми, за умови реалізації деякої мови, якою написана програма.

Засоби інтелектуального інтерфейсу утворять новий рівень - *рівень кінцевого користувача,* що визначає якісно новий стрибок у розвитку

технології обробки інформації на ЕОМ, що пояснюється принциповими відмінностями цього рівня від усіх попередніх.

Перша відмінність полягає в тому, що засоби рівня кінцевого користувача орієнтуються на пристосування до користувачів, що мають різну, у тому числі найнижчу, кваліфікацію. Засоби ж попередніх рівнів у головному були спрямовані на пристосування користувача до цих засобів, вимагаючи від нього придбання спеціальної кваліфікації для їхнього використання.

Друга відмінність полягає в тому, що засоби рівня кінцевого користувача повинні забезпечувати весь процес розв'язання задачі, починаючи з виникнення інформаційної потреби користувача, у той час як традиційні засоби в основному були спрямовані на забезпечення процесу виконання програми.

Третя відмінність полягає в тому, що засоби нового рівня повинні забезпечувати спільне розв'язання задачі кінцевим користувачем і ЕОМ у тісній взаємодії і при постійній зміні ролей у процесі цієї взаємодії. Засоби ж традиційних рівнів практично забезпечували такий процес тільки для програмістів на етапі налагодження і виконання програми.

Необхідно підкреслити, що самий по собі рівень кінцевого користувача не розширює можливостей засобів рівнів, наведених нижче, а тільки забезпечує для кінцевого користувача оперативний доступ до них без додаткових посередників.

Якщо ці можливості бідні, то будуть бідні і можливості системи обробки інформації в цілому, навіть якщо вона має у своєму складі інтелектуальний інтерфейс.

Під кінцевим користувачем при цьому потрібно підрозумівати не якогось усередненого кінцевого користувача, а спектр різноманітних користувачів, кожний із який, у свою чергу, висуває спектр різноманітних вимог до обчислювальної системи.

Основна особливість цих спектрів - неможливість їхньої апріорної розбивки на чітко розділені класи. Кожний кінцевий користувач являє собою деяку індивідуальність по своїй кваліфікації (як у своїй професійній галузі, так і в галузі використання засобів обчислювальної техніки), запитам, навичкам, відношенню до ЕОМ і т.п.

Практично неможливо персонально для кожного користувача побудувати обчислювальну систему, орієнтовану на задоволення саме його вимог, тому використовують два шляхи розв'язання даної проблеми.

Перший шлях - створення спеціалізованих систем, орієнтованих на задоволення вимог окремих класів користувачів. Забезпечуючи успіх в окремих поодиноких випадках, цей шлях не дає загального розв'язання проблеми.

Другий шлях спрямований на створення базових засобів обчислювальної техніки, орієнтованих на розв'язання широкого кола

задач, але тих, що адаптуються тими або іншими способами до специфічних вимог окремих груп користувачів (у межі - до потреб індивідуального користувача).

Виконання кожної з інтелектуальних функцій забезпечується відповідним комплексом таких засобів. Включення в конфігурацію відповідної обчислювальної системи таких комплексів дозволяє забезпечувати інтелектуальний інтерфейс із кінцевим користувачем при розв'язанні певного класу задач. При цьому кінцевому користувачу надається можливість формування для себе персонального операційного середовища, що найбільш повно відповідає його професійним потребам і індивідуальним навичкам.

Нова технологія обробки інформації, що охоплює весь процес розв'язання задачі (від моменту виникнення в кінцевого користувача деякої інформаційної потреби до її задоволення) і передбачає особисту участь кінцевого користувача на всіх етапах розв'язання, висуває якісно нові вимоги як до загальної організації процесу розв'язання задачі, так і до функцій окремих ланок людино-машинної системи (ЛМС). Це пояснюється тим, що:

• *у традиційній технології розв'язання задачі з використанням ЕОМ функціонування обчислювальної системи зводилося до правильного виконання заданих послідовностей інструкції й апріорно визначених правил переходу від однієї послідовності до іншої.*

Одержуваний результат є функцією виконуваних операцій, що містяться в заданій програмі, і вихідних даних, що містяться в умовах задачі. При цьому практично всі операції здійснюються тільки обчислювальною системою.

• *При новій технології обробки інформації обчислювальній системі повинна задаватися тільки постановка задачі у вигляді опису необхідного результату й умов його одержання, а послідовність операцій, за допомогою виконання якої він досягається, повинна визначатися задачею, яка вирішується системою.*

При цьому обчислювальна система в складі людино-машинної системи перетворюється з пасивної ланки в *активну цілеспрямовану систему, метою якої є одержання необхідного в постановці задачі знання, що досягається автоматичним формуванням системою і виконанням оптимальної у певному сенсі послідовності обчислювальних, логічних і пошукових операцій над наявними знаннями.* При цьому в дану послідовність можуть включатися послідовності операцій, які виконуються користувачем як ланкою ЛМС.

1.4 Організація обчислювального процесу в новій технології

Нова організація процесу обробки інформації в обчислювальній системі є черговим кроком у постійній еволюції програмного

забезпечення, тісно пов'язаній з двома формами подання знань в ЕОМ - процедурною і декларативною (описовою), відносною вагою процедурних і декларативних знань у відображенні проблемного середовища, їхніми взаємозв’язками і роллю в процесі обробки інформації.

Будь-яка програма являє собою деяку процедуру - послідовність операцій, здійснюваних над певними структурами даних. Структури даних є декларативним знанням, що несе тільки функцію відображення середовища. Елементи знання завжди мають певну інтерпретацію в проблемному середовищі, тобто відображають її окремі об’єкти.

Будь-яка процедура орієнтована на роботу з певним типом декларативних знань. Результатом роботи завжди є нове декларативне знання. Необхідні кінцевому користувачу знання також подаються у декларативному вигляді.

У процесі еволюції програмного забезпечення змінювалися способи організації даних, причому їхня структура постійно розвивалась та ускладнювалась, змінювався ступінь взаємопов’язаності між даними та обробляючими програмами, поступово переоцінювалась роль даних і програм у процесі обробки інформації на ЕОМ. На зміну поданню даних у вигляді ізольованих одиничних елементів (слів) прийшло подання у вигляді векторів, масивів, списків, файлів. Максимум цього розвитку припав на абстрактні типи даних, що забезпечують для користувача вибір найбільш адекватної для розв’язання задачі структури даних.

Одночасно змінився і погляд на взаємовідносини даних та програм, що їх оброблюють, - від тлумачення даних як простого придатка до програми до розуміння певних сукупностей даних як деяких полів для роботи відповідних процедур, а потім до переходу до баз даних, у складі яких дані утворюють цілісну сукупність декларативних знань, що відображає проблемне середовище, і незалежну від процедур, що їх оброблюють.

Ряд змін відбувся і відносно оцінки ролі програм і даних у процесі обробки інформації на ЕОМ. У цьому зв'язку можна говорити про *підхід, орієнтований на процес, і підхід, орієнтований на дані.*

При першому підході, традиційному для обчислювальної техніки, процес обробки інформації при розв’язанні задачі визначається як виконання програми, при другому - як одержання необхідних даних, що і виступає в цьому випадку як єдина ціль процесу обробки.

Різниця цих підходів полягає не тільки в різних способах опису інформації. Різні підходи обумовлюють різні способи контролю та різні організації схеми керування процесом розв’язування задачі. У традиційній обчислювальній техніці поняття «процес» є фундаментальним. *Процес - це виконання програми.* У кожний момент часу процес знаходиться у певному стані, що характеризується ступенем його виконання. При цьому стан містить у собі всю інформацію, необхідну для припинення процесу і його

наступного поновлення, і містить, щонайменше, таку інформацію:

а) програму;

б) індикацію команди, що повинна виконуватись наступною;

в) значення всіх програмних змінних і даних;

г) стан усіх пристроїв введення – виведення, що використовуються.

В міру протікання процесу його стан змінюється. Наприклад, змінюється індикація команди, змінні набувають нових значень. У цілому параметри, що змінюються, характеризуються вектором стану процесу, за яким здійснюється контроль процесу обробки інформації та керування ним.

При другому підході контроль та керування процесом обробки інформації в обчислювальній системі здійснюються за станом всієї сукупності даних, тобто наявності або відсутності в цій сукупності тих або інших даних.

Нова технологія обробки інформації продовжує і розвиває підхід, орієнтований на дані, тому що кінцевою метою процесу розв’язання задачі є одержання необхідного знання (дані ж - окремий випадок надання декларативних знань), а хід процесу контролюється й керується тільки з погляду досягнення цього результату.

Отже, можна сформулювати вимоги до загальної організації процесу обробки інформації в обчислювальній системі при новій технології.

1. Розв’язання будь-якої задачі повинно розглядатися як надання споживачу за його запитом потрібного знання, що задовольняє деякі специфікації на знання, що міститься в запиті.

   Як «споживач» знання і «замовник» на нього може виступати або кінцевий користувач (у тому числі при початковій постановці задачі), або будь-яка програма, що обробляє знання.

2. Необхідність для «споживача» у тому або іншому знанні (інформаційна потреба) визначається як відсутність у споживача інформації, необхідної для розв’язання підзадачі у складі загальної задачі. Інформаційна потреба одного споживача (програми або користувача) є основою для формулювання підзадачі для інших програм (а також і користувача в складі ЛМС), які потенційно спроможні вирішити цю підзадачу, тобто одержати необхідне знання.

   ЛМС у цілому трактується як структура, що складається з компонентів, кожний з який виконує дві функції: розв’язання деякої задачі за запитом, який надійшов ззовні, і формулювання запиту до інших компонентів на відсутнє для рішення цієї задачі знання.

   Задача подається як множина підзадач, які розв'язуються окремими компонентами ЛМС, а весь процес розв’язання задачі розпадається на множину процесів задоволення запитів

різноманітних компонентів. Кінцевий користувач трактується як деякий кінцевий споживач знань, інформаційні потреби якого ініціюють роботу всіх інших програм.

3. Хід розв’язання задачі в будь-який момент часу оцінюється за станом системи знань, в якій процедурна частина залишається постійною, а декларативна - постійно змінюється. Невідповідність реального стану необхідному визначає внутрішню активність обчислювальної системи і викликає спрямовану перебудову процесу обробки інформації.

1.5 Інтелектуальні системи й галузі їх застосування

Інтелектуалізація *систем обробки інформації* (СОІ - *systems of processing of the information*) обумовлена необхідністю застосування їх у більш різноманітних сферах: науці, техніці, виробництві, економіці, медицині, навчальному процесі, побуті та багатьох інших галузях життєдіяльності людини. При цьому, по-перше, необхідно забезпечити можливості застосування засобів обробки інформації користувачами, що підготовлені слабко або взагалі не мають підготовки, по-друге, на практиці виникають все нові фізичні, технологічні процеси, контроль і керування якими повинно здійснюватися інтелектуальними системами (без участі людини) в умовах значної невизначеності відносно особливостей зовнішнього середовища.

Таким чином, можна виділити дві істотні галузі функціонування інтелектуальних систем (ІС):

- людина - СОІ;
- технічні системи - СОІ.

Кожна з цих галузей застосування ІС має свої особливості, обумовлені характером взаємодіючих компонентів і розв'язуваних задач.

1.5.1 Людина - СОІ

Характер взаємодії людини з середовищем, з технічними системами визначається його творчими інтелектуальними можливостями. Людський мозок - унікальна система сприйняття, переробки і генерації нової інформації. Його поведінка в будь-якій ситуації, характер реакції на найнесподіваніші зміни навколишнього середовища вражаючі. Але у розв’язанні дуже багатьох задач, особливо якщо потоки інформації дуже великі або при обробці інформації ставляться обмеження за часом, людина не може змагатись з сучасною технікою. Природно, що в таких умовах цілком зрозуміло бажання та є нагальна потреба перекласти значну частину функцій із сприйняття і переробки інформації на техніку, що має властивості, які граничать із розумною поведінкою.

Для надання засобам обробки інформації "інтелектуальних здібностей" необхідно в першу чергу забезпечити їх комунікативними

засобами сприйняття інформації, що подається текстами, написаними від руки або, у крайньому випадку, надрукованими яким-небудь способом, кресленнями, графіками, малюнками, тобто необхідно забезпечити СОІ технічним зором. Не менш важливо забезпечити СОІ можливістю сприймати звичайну мову людини, тобто забезпечити її технічним слухом.

Проте мова йде не тільки про сприйняття інформації технічним зором та слухом, але й, що найважливіше, про "розуміння" штучною системою суті надрукованого або сказаного, тобто мова йде про дійсну інтелектуалізацію комунікативних засобів СОІ. Ця задача значно складніша простого сприйняття такою системою зорових і слухових сигналів, що сьогодні вирішити не так вже і важко.

Але головним призначенням інтелектуальної СОІ не є просте сприйняття і "розуміння" суті сприйнятих сигналів. Її функція повинна бути значно складнішою і значимішою - обробка інформації, розв’язання задач, що стоять перед людиною. Отже, про інтелектуальність СОІ можна говорити тоді, коли в ній є можливість "розуміти" суть розв'язуваної задачі. А це значить, що інтелектуальна система повинна здійснювати аналіз інформації, що надходить, виділяти компоненти задачі (вихідні дані, шукані результати) і планувати свою діяльність за рішенням цієї задачі відповідно до мети, яку поставила людина. При цьому необхідно мати на увазі, що людина ставить глобальну мету, а проміжна мета повинна формуватися, трансформуватися самою СОІ, так само як і стратегії досягнення цієї мети і розв’язання всієї задачі.

Такі властивості, такі інтелектуальні можливості СОІ досяжні тоді, коли, по-перше, у пам'яті системи будуть закладені знання про будь-яку предметну галузь, до якої відноситься розв'язувана задача, по-друге, коли задачу, мету її розв’язання людина зможе сформулювати на природній - нехай і фаховій - мові, тобто мові, що притаманна предметній галузі задачі. А це значить, що перед інтелектуальною СОІ ставиться задача "розуміти" фахові мови предметних галузей.

Таким чином, усе сказане дозволяє виділити ті головні характеристики, ті мінімально необхідні властивості, які повинні мати СОІ, щоб їх можна було вважати інтелектуальними.

1. Інтелектуальні системи повинні забезпечуватися "органами почуттів", головними з який є технічний зір і слух, що дозволяють їм сприймати інформацію в тому вигляді, який характерний для людських засобів комунікації.

2. Інформація повинна сприйматися ІС не пасивно, не як потік акустичних або візуальних сигналів, а як семантично значимий потік даних, що розуміється.

3. У пам'яті ІС повинний знаходитись банк знань із різноманітних предметних галузей, що дозволяє людині ставити перед такою системою різні задачі в звичній для неї формі і глобальні цілі розв’язання цих задач.

4. У ІС повинні існувати засоби, що дозволяють, по-перше, сприймати й аналізувати суть задачі і мету її розв'язання, по-друге, формулювати підцілі і синтезувати план розв'язання задачі.

5. Звісно ж, у ІС повинні бути засоби, що дозволяли б видавати результати обробки інформації в зручній для людини формі: друкарський текст, креслення, малюнки, мова (акустичний висновок) і інші.

1.5.2 Технічні системи - СОІ

Взаємодія людини із системою, що має інтелектуальні "здібності" будь-якого досяжного ступеня, характеризується тим, що інтелектуальна компонента властива обом контактуючим сторонам. І якщо в якийсь момент у ІС можливості виявляться вичерпаними, то людина бере на себе відповідну частину зобов'язань, забезпечуючи нормальне функціонування симбіозу людина - СОІ.

Цілком інша картина, інші обставини виникають при функціонуванні симбіозу технічна система - СОІ. Будь-яка технічна система призначена для перетворення або передачі (транспортування) речовини або енергії. Її функціонування здійснюється в умовах деякого середовища, яке, як правило, має деструктивний вплив, порушуючи певний процес.

Доти, доки технічні системи були прості, доки швидкості процесів у них були порівнянні зі швидкістю реалізації людиною проблеми контролю і керування такими процесами і технічними системами гостро не стояли. З цими задачами цілком успішно справлялася людина або досить прості пристрої контролю і керування. Різке збільшення швидкостей переміщення, поява нових матеріалів і їхнє застосування в різноманітних конструкціях машин і механізмів, ускладнення технологічних процесів і удосконалення технологічного устаткування, мікромініатюризація, космічна техніка, біоінженерія вимагають зовсім інших підходів до організації процесу контролю і керування. Виявляється необхідним вирішувати проблеми аналізу багатьох джерел інформації, які характеризують параметри і характеристики процесів і технічних систем, де протікають ці процеси. При цьому час аналізування в більшості випадків зникаючи малий, оскільки процеси швидкоплинні і вихід їх із нормального режиму може призвести до дуже важких наслідків катастрофічного характеру. Прикрих прикладів тому чимало. Другий момент - найвища точність, прикладом якої може служити виробництво інтегральних схем, де розміри знаходяться в межах сотих і тисячних часток мікрометра. Ще однією ілюстрацією своєрідності сучасних систем і процесів в них може служити, наприклад, процес керування місяцеходом або марсоходом. Людина взагалі не може оперативно брати участь у процесі керування таким автоматом, оскільки інформація про поточний стан автомата і навколишнього середовища доходить до земного пункту за хвилини і десятки хвилин. За цей час поточний стан настільки змінюється,

що впливи, які посилаються будуть носити тільки деструктивний характер.

Отже, виникає потреба оснащення таких (і їм подібних) апаратів інтелектуальними системами керування, що змогли б, по-перше, сприймати різноманітну інформацію про навколишнє середовище і її взаємодію з апаратом, по-друге, аналізуючи цю інформацію, формувати цілі і здійснювати розв’язання складних задач із вироблення керуючих впливів.

Зі сказаного очевидно, що така система повинна забезпечуватись технічним зором, спроможним сприймати об'ємні сцени, виділяти в цих сценах значимі для даної ситуації фрагменти, аналізувати всю сцену, включаючи в її апарат, виділені фрагменти навколишнього середовища і приймати рішення, що сприяють нормальному функціонуванню апарата. Цілком ясно, що ці функції відповідають інтелектуальній поведінці, інтелектуальному процесу керування складною системою.

У цих умовах можна знайти сьогодні багато подібних ситуацій. Роботи, що встановлюють, наприклад, елементи на друкарську плату, працюють у не менш складних умовах, викликаних високою точністю позиціонування елементів щодо місця розташування плати й елемента. При цьому необхідно забезпечити високу швидкість технологічного процесу і належну якість та надійність. І тут необхідні "інтелектуальні здібності" в системі керування роботом.

Саме ж головне у тому, що в симбіозі технічна система – СОІ інтелектуальні "здібності" має тільки підсистема обробки інформації. Якщо вона вичерпає свої можливості із сприйняття інформації, її переробки і вироблення керуючих впливів, то система в цілому утрачає функціональну повноту і роботоздатність. Вивести її з такого стану може третя "компонента" - людина.

Таким чином, властивості, характеристики, можливості ІС, поєднаних із сучасними складними технічними системами, повинні відповідати вимогам ще більш сучасних інтелектуальних систем і включити в себе не тільки технічний зір та слух, але й інші органи чуття. Наприклад, захоплювачі роботів повинні забезпечуватись чутливими елементами дотику, які спроможні сприймати не тільки загальну силу стиску, але і деякі характеристики поверхні предмета. У інших випадках було б доцільно оснащувати ІС "технічним нюхом" для функціонування в середовищі з газовим складом, що змінюється, та інше. І в усіх випадках інтелектуальні системи повинні мати у своєму складі банки знань і програмне забезпечення, що дозволяє аналізувати інформацію, яка надходить від "органів чуття", формувати адекватні цій інформації задачі і цілі їх розв’язання стосовно глобальних цілей і задач, сформульованих людиною.

1.5.3 Проблеми розпізнавання образів

Основна група ІС повинна мати здатність до розпізнавання образів, тобто властивість активно сприймати явища зовнішнього середовища, класифікувати і ідентифікувати ці явища (процеси, об'єкти) відповідно до деякої цільової настанови. Такими об'єктами можуть бути зображення найрізноманітнішого характеру (письмові знаки, силуетні, контурні, текстурні зображення об'єктів, напівтонові фотографії або телевізійні кадри складних просторових сцен) акустичні сигнали, серед яких можна виділити, наприклад, людську мову. Не менш складними об'єктами є сукупності сигналів, що характеризують стан складних систем, серед яких найскладнішою системою є людський організм. Очевидно, не так вже і важко виявити множину інших процесів, явищ і об'єктів різноманітних модальностей, що потребують при їхньому аналізі застосування принципів розпізнавання.

В усій цій сукупності значне місце займають зорові і слухові образи, що забезпечують потік інформації до ІС через технічні зір та слух. Це особливо стосується технічного зору для роботів, що забезпечує їхнє функціонування в слабко упорядкованому середовищі. У цьому випадку мова йде про аналіз і розпізнавання об'ємних сцен і об’єктів у цих сценах. Процеси аналізу сцен і розпізнавання об'єктів у таких умовах значно складніші, оскільки той самий об’єкт у межах сцени може приймати різні положення, змінюючи свої властивості і характеристики; він може затінюватися іншими об'єктами, що додає у його образ "чужі" йому властивості, маскує його до ступеня невпізнанності і багато чого іншого.

1.5.3.1 Задачі в розпізнаванні образів

Будь-який об'єкт (явище, ситуація, процес) характеризується різноманітними властивостями, параметрами, ознаками і прикметами. У кожному конкретному випадку вибираються й аналізуються ті факти, ті властивості, що найбільшою мірою характеризують об'єкт відносно конкретної цілі.

Сукупність певних властивостей і їх співвідношень на об'єкті формують образ об'єкта. Очевидно, що множини об'єктів, які мають однакові властивості і співвідношення цих властивостей, повинні групуватися в сукупність, яку прийнято називати класом.

Тут необхідно чітко розуміти цей факт, що клас не обов'язково є множиною абсолютно ідентичних, в усьому рівних, невідрізняємих об’єктів. У цьому легко переконатися на такому прикладі, як клас засобів пасажирського транспорту. Сюди можна віднести тролейбус, автобус, велосипед, літак, а як контраст - кінь або верблюд. Цілком ясно, що до класу засобів пасажирського транспорту ці об'єкти віднесені за певною ознакою - спроможністю переміщати пасажира в просторі. За будь-якими іншими ознаками ці об’єкти відрізняються суттєво й до одного класу, як

правило, входити не можуть. Обмовка "як правило" тут цілком доречна, адже кінь і верблюд, наприклад, можуть бути віднесені до якогось одного класу за ознакою "мають по чотири ноги і є тваринами", а тролейбус, автобус і велосипед - "мають пневматичні колеса як головний двигун". Ці компоненти в теорії і практиці розпізнавання образів мають дуже важливе значення, оскільки визначають характер і тип задачі і, як наслідок, методи і підходи до її розв'язання.

Ідентифікація (identification) - один із найбільш характерних і поширених типів задач. Суть подібної задачі у такому. Заздалегідь виділяється деяка обмежена множина класів об'єктів

$$A=A_1, A_2, \ldots, A_M.$$

Кожний A_i клас цієї множини подається *еталоном* $V_i = v_{i1}, v_{i2}, \ldots, v_{ik}, \ldots, v_{in}\ldots$ При цьому вважається, що кожний еталон містить строго n ознак v_{ik}. Реальні об'єкти, що підлягають розпізнаванню і називаються *реалізацією класу* A_i, подаються описом $S_i = s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots, s_{in}\ldots$

Кількість еталонів, як правило, дорівнює кількості класів або може бути трохи більшою (наприклад, для літери „А" необхідно мінімум два еталони „А" і „а"), а кількість реалізацій нескінченна навіть для одного класу. Характерно і те, що кожна ознака *sik* реалізації може набувати будь-яких значення в межах $sik_{\min} \le sik \le sik_{\max}$.

Задача полягає в тому, щоб знайти таке вирішальне правило або міру близькості D, що дозволило б порівнювати будь-яку реалізацію S_i з множиною еталонів V і виносити рішення про належність реалізації до одного з класів A_i. Це значить, що при порівнянні реалізації S_j із будь-яким еталоном V_i

$$D\ (V_i, S_j) = max, \quad i \neq j, \tag{1.1}$$

а при порівнянні реалізації зі своїм еталоном

$$D = (V_i, S_i) = min. \tag{1.2}$$

Це і буде критерієм належності реалізації до конкретного класу, тобто розпізнаванням або ідентифікацією. Суть міри близькості D може бути різною. У самому простому випадку її можна трактувати як відстань між двома точками V_i і S_j у n - вимірному евклідовому просторі. Тоді кожний елемент *vin*, *sjk* еталонного опису і реалізації є значеннями координат цього простору, а процес порівняння являє собою операцію обчислення відстані за відомою залежностю

$$\mathbf{D} = \sqrt{\sum_{k=1}^{n} \left(v_{ik} - s_{jk}\right)^2} \ . \tag{1.3}$$

Можна використовувати як міру близькості і квадрат відстані, тобто обчислювати величину

$$\mathbf{D2} = \sum_{k=1}^{n} (v_{in} - \mathrm{S_{in}})^2 , \qquad (1.4)$$

що спрощує процес обчислення. Адже в даному випадку важливо одержати не конкретну величину, а критерій, що відповідає умовам (1.1) і (1.2). У ідеальному випадку порівняння реалізації S_i із своїм еталоном V_i за будь-якою залежністю для міри близькості повинно дати значення критерію рівне нулю.

Але вся складність полягає в тому, що будь-який елемент *sjk* реалізації S_j може набувати значення в досить широких межах. Це викликано впливом перешкод. Тоді будь-яка міра, чи (1.3), чи (1.4) не дає при порівнянні реалізації S_i із своїм еталоном V_i умови (1.2), тобто значення нуля. Мало того, це значення може виявитися більшим, ніж можливе значення при порівнянні S_i реалізації з чужим еталоном V_j. Отже, процес розпізнавання буде помилковим. Виникає похибка першого роду, тобто пропускання «своєї» реалізації.

Можна вважати, що будь-яка реалізація S_j буде мати такі значення компонент *sjn*, що на чужому еталоні V_i дасть дуже мале значення міри близькості і буде прийняте рішення про те, що це «своя» реалізація. І знову допущено похибку, яку прийнято називати похибкою другого роду.

У кожному конкретному випадку намагаються різноманітними способами мінімізувати будь-яку з похибок. Наприклад, в обороні країни важливо не пропустити "свою" ціль, тобто ворожу ракету. Тут необхідно мінімізувати похибку першого роду.

Проблемам теорії і практики розпізнавання присвячений спеціальний розділ, де більш докладно розглядаються ці питання. Але зі сказаного можна зробити цілком певні висновки, окреслити коло задач, що виникають у цій проблематиці.

1.5.3.2 Вихідна інформація

Однією з серйозних і важких проблем у теорії і практиці розпізнавання є одержання необхідних ознак (параметрів, властивостей) об'єктів, що розпізнаються. Питання упирається в першу чергу в можливість технічної реалізації пристроїв, що сприймають саме необхідну інформацію. Як приклад роздивимося акустичні сигнали мови людини. Поки єдиним технічним "вухом" є мікрофон, на виході якого можна одержати величину, адекватну характеру механічного тиску на мембрану (або інший елемент) цього мікрофона і нічого більше.

Далі необхідно всілякими хитрощами витягати корисну інформацію, що і здійснюється завдяки особливостям мовного сигналу. Це якийсь коливальний процес - зміна в часі величини тиску на мембрану, що можна зобразити у вигляді деякої часової залежності (рис.1.1)

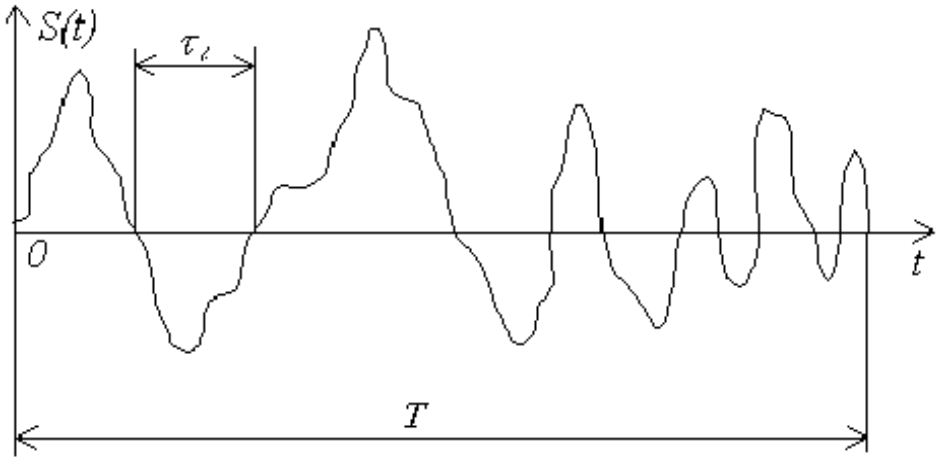


Рисунок 1.1 - Приклад часового відрізка мовного сигналу

Аналітичну залежність *I = f(t)* записати в явному вигляді неможливо, отже, необхідно працювати з деяким об'єктом, що подає множину даних - часовий ряд. А для цього в системі, що розпізнає, необхідно одержати даний часовий ряд методом аналого-цифрового перетворення. Але виникає інша проблема. Необхідно вибрати крок дискретизації за часом, тобто $\Delta t = ti+1 - ti = const$, вибір якого ґрунтується на певних критеріях, наприклад, теоремі Котельникова, вираженій такою залежністю [1]:

$$I(t) = \sum_{k=-\infty}^{\infty} I(k\Delta t)\frac{\sin[w_c(t-k\Delta t)]}{w_c(t-k\Delta t)},$$

де $\Delta t = \frac{\pi}{w_c} = \frac{1}{2f_c}$.

По-перше, у цій залежності підсумовування здійснюється за нерівністю $-\infty \le k \le \infty$, що в принципі неможливо, оскільки мовний сигнал існує на кінцевому відрізку часу, і знати його передісторію ($k = -\infty$) і постісторію ($k = \infty$) нам не дано. По-друге, Δt залежить від частоти w_c, що обмежує реальний спектр сигналу, а він має - у силу своєї скінченності в часі - дуже широкий спектр, що вносить суттєву невизначеність.

На практиці доводиться значно збільшувати «запас за Котельниковим», тобто вибирати Δt значно меншим, ніж це необхідно фактично. У іншому випадку буде внесено таку похибку, яка значно спотворить сигнал.

Але і це ще не усе. Будь-який коливальний процес характеризується рядом параметрів, серед яких найбільш часто використовуються частота (спектр частот), фаза, амплітуда. Ці параметри легше виявити, коли в

сигналі немає постійної складової. Технічно розв’язати задачу усунення постійної складової не так вже і важко. Але завжди виникають сумніви щодо істинності такого усунення. Цілком можливо, що фактичне розташування осі *t* знаходиться трохи вище або нижче, а отримане розташування обумовлене особливостями самого сигналу або апаратури, яка усувала постійну складову.

Найскладніша задача при обробці вихідного сигналу - виявлення в ньому корисної інформації - пов'язана з перешкодами. Разом із корисним сигналом на мікрофон діє шум навколишнього середовища, крім того, вимірювальний пристрій, підсилювачі, аналого-цифрові перетворювачі, канали передачі сигналу самі "шумлять". У найкращому випадку будь-який відлік являє собою суміш *I(t)* етапів попередньої обробки інформації:

$$I(t) = f(t) + Ut,$$

де *f(t)* - дійсне значення сигналу, а *Ut* - перешкода.

Виявити в потоку даних корисну інформацію означає в даному випадку усунення перешкод. Вся складність цієї операції полягає в тому, що тут не можна використовувати статистичні методи, оскільки є одноактний вимір послідовності. Для таких послідовностей є досить прості прийоми фільтрації шуму [2], що здійснюють згладжування ряду в цифровій формі. Найпростіший фільтр подається залежністю:

$$\hat{I}_k = \frac{1}{4}(I_{k-1} + 2I_k + I_{k+1}), \qquad (1.5)$$

де $\hat{I}_k$ - згладжене значення *k*-того відліку ряду даних;

I_k - вихідне значення *k*-того відліку ряду даних.

Цілком очевидно, що відфільтрований сигнал більш плавний, без різких викидів (що є характерним для суцільної кривої) і зламів.

Проте плавність цього сигналу зовсім не говорить про те, що будь-які отримані відліки $\hat{I}_k$ відповідають істинному значенню сигналу *f(t)*. Адже який-небудь «викид» може бути характерним нюансом акустичного сигналу і нести в собі дуже корисну інформацію. Тому застосування етапів попередньої обробки інформації повинно бути зваженим і обдуманим.

1.5.3.3 Ознаки

Попередня обробка вихідної інформації дозволяє приступити до виявлення різноманітних ознак, значення яких одержати на вихідному об'єкті, що розпізнається, за допомогою простих вимірів неможливо. Ця задача носить творчий характер і саме її вдале розв’язання дозволяє в значній мірі вирішити і проблему розпізнавання об'єктів даного характеру.

Незважаючи на фрагментарність викладених у цьому параграфі питань, можна зробити ряд чітких висновків про характер головних задач, які повинні розв’язувати розпізнавальні системи, про головні етапи

одержання й обробки інформації для винесення рішення щодо тих проблем, котрі необхідно вирішити при створенні цих систем.

1. Етап одержання вихідної інформації залежить від характера об'єктів, особливостей їх параметрів, ознак і характеристик. Ці особливості визначають засоби одержання інформації, можливість використання наявних датчиків або необхідність створення нових. Ефективність цього етапу обумовлює характер обробки інформації на таких етапах і розпізнавання в цілому.

2. Етап попередньої обробки інформації й одержання ознак об'єктів, які розпізнаються, обумовлюється тим, що, по-перше, вихідну інформацію можна одержати доступними в даних умовах засобами, що залежить від розвитку науки і техніки, по-друге, у будь-якому випадку вихідна інформація надходить разом з усілякими шумами і похибками.

Таким чином, на цьому етапі здійснюється процес фільтрації і різноманітні перетворення покликані виділити з потоку вихідної інформації найбільш корисні ознаки. При цьому необхідно мати на увазі, що методи обробки інформації і виділення ознак завжди супроводжуються втратою корисної інформації. І однією з головних проблем здійснення цього етапу є вибір або розробка таких методів, що забезпечували б мінімум втрат і високої ефективності одержуваних ознак.

3. Етап власне розпізнавання являє собою порівняльний аналіз ознак реалізації з еталонними ознаками й ухвалення рішення про належність реалізації до одного з обмеженої множини класів. Головними проблемами в створенні засобів, що дозволяють реалізувати ефективно і надійно цей етап, є розв’язання задачі оцінки інформативності і достатності ознак, які утворять еталони; вибір або розробка методів порівняльного аналізу ознак, тобто вирішального правила; підходи до оцінки можливих похибок розпізнавання і встановлення припустимих меж цих похибок.

Ці проблеми часто ускладнюються вимогами до швидкодії процесу розпізнавання. Якщо мова іде, наприклад, про розпізнавання слів, які вимовляє людина, команд, то швидкодія процесу їхньої ідентифікації повинна бути адекватна нормальному темпу усного мовлення. У зчитувальних автоматах необхідно одержати швидкодію, що дозволяє розпізнавати сотні і тисячі знаків у секунду. У інших випадках вимоги можуть носити інший характер, що стосується достовірності і ряду інших аспектів, які потребують інших підходів і інших систем.

1.5.4 Експертні системи

Експертна система (*expert system*) є системою штучного інтелекту, що використовує знання з порівняно вузької предметної галузі для розв’язання виникаючих у ній задач, причому так, як це робила б експерт-людина, тобто в процесі діалогу з зацікавленою особою, що надає

необхідні відомості з конкретного питання. Необхідно підкреслити, що це саме одне з множини наявних означень, із яким можна погоджуватися або ні. Важливо інше. Мова йде про інтелектуальну систему, призначену для розв’язання задач в умовах значної невизначеності щодо вихідних даних цієї задачі.

Для більшої ясності зазвичай прийнято наводити приклад з галузі медицини в частині постановки діагнозу хворому. Дійсно, хвора людина не може ясно і чітко описати свій стан, дати вичерпні й однозначні відомості про характер недуги. Розповідь його носить дуже туманний, приблизний і суперечливий характер. Основу для діагнозу складають дані аналізів, явні ознаки хвороби, наприклад, температура, зміни в крові і ряд інших.

Але підвищення температури властиве дуже багатьом захворюванням, як і зміни в крові. Це відноситься і до інших об'єктивних показників. Отже, ці ознаки не можуть слугувати єдиною підставою для однозначної постановки діагнозу. В цій ситуації використовується метод експертизи, здійснюваної групою фахівців високого класу. Потрібно саме високий клас фахівця, оскільки тільки в цьому випадку можна опиратися на великі і глибокі знання про предмет експертизи.

Таким чином, експертна система повинна постачатися обсягом знань, достатніх для розв’язання широкого кола задач, що виникають у конкретній предметній галузі. Це одна із фундаментальних ознак, що відрізняють експертну систему від інших інтелектуальних систем. У закордонній літературі еквівалентом поняття науки в галузі експертних систем є термін "*інженерія знань*" *(knowledge engineering*). Цілком зрозуміло, що знання істотно відрізняються від сукупності яких-небудь параметрів про процес, явище, об'єкт. У кращому випадку цей набір ознак, параметрів являє собою упорядковану множину, порядок у якій установлюється на підставі об'єктивних критеріїв.

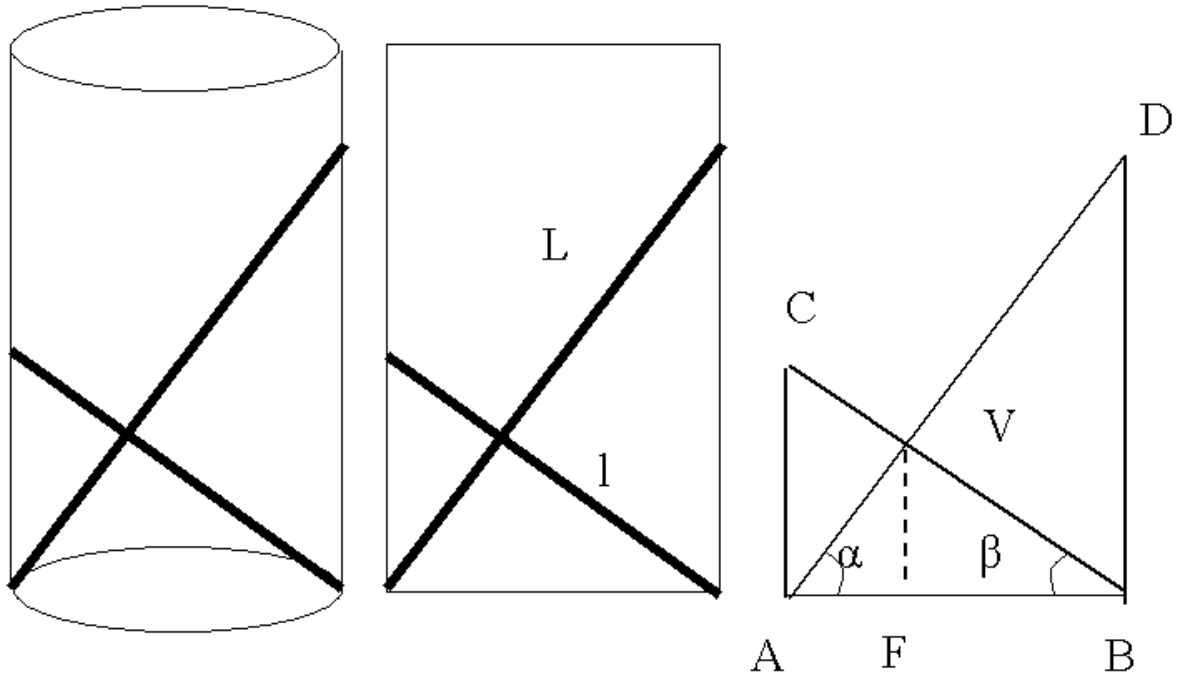
Розглянемо приклад, що дозволить усвідомити істотні відмінності знань від набору даних і роль знань у розв’язанні задачі. Як такий приклад використаємо задачу "жерців бога Ра", що задавали претендентам на посаду жерця. Суть її ось в чому.

Визначити діаметр колодязя, якщо відомо, що глибина води в ньому складає одну міру (наприклад, 1 лікоть, метр і т.п.). Якщо опустити в колодязь дві перехресні жердини, одна довжиною дві, а інша три міри, сперти кінці в протилежні точки краю дна, то місце їхнього перехрещування виявиться на рівні поверхні води. Схематично це зображено на рис. 1.2, а).

Вже в умові є ряд недомовок, для надолуження яких у розв’язуючого задачу повинні бути певні знання. Мова йде, наприклад, про те, що колодязь круглий. Про це говорить слово "діаметр", але прямого твердження немає. Потрібно зіставити слово "діаметр" із твердженням (знанням) про те, що діаметри бувають у круглих об'єктів. Якщо цих знань

немає, то вся наступна інформація позбавлена будь-якого сенсу.

Але важливіше інше. Для розв’язання задачі необхідно перейти до абстрактного об'єкта, позбувшись непотрібних відомостей. Насамперед, необхідно жердини замінити пересічними прямими, шорсткуваті і криві стінки колодязя уявити у вигляді ідеального циліндра з рівним круглим дном, забути про воду і розглядати відстань від точки перетинання прямих до дна. Це своєрідний перший етап абстрагування, що потребує цілком конкретних знань, правил, припустимих умовностей.



а) б) в)

Рисунок 1.2 - До задачі „жерців бога Ра”

Далі необхідно знати, що через дві пересічні прямі можна провести площину, а в перерізі цієї площини з циліндром-колодязем утвориться прямокутник (рис.1.2, б). Всі ці знання витягаються з "скарбнички" знань розв’язуючого задачу. Немає таких знань - не буде і розв’язання.

Продовжуючи процес абстрагування, необхідно перейти до схеми (рис.1.2, в) двох прямокутних трикутників *ABC* і *ABD*, маючи на увазі, що *AB=x=AF+FB*; *CB=l*; *DA=L* і *VF=h*, виразити шукану величину через залежність

$$x = \frac{h}{tg\alpha} + \frac{h}{tg\beta};$$

замінити $tg\alpha = BD/x$ і $tg\beta = AC/x$ виразами

$$tg\alpha = \frac{\sqrt{L^2 - x}^2}{x}, tg\beta = \frac{\sqrt{l^2 - x}}{x}$$

і отримати вираз

$$x = x* \; h* \left(\frac{1}{\sqrt{L^2 - x^2}} + \frac{1}{\sqrt{l^2 - x^2}} \right),$$

який дозволяє скористатися ітеративним процесом обчислення шуканої величини у вигляді такої процедури:

$$x_{i+1} = \frac{x_i}{\frac{1}{\sqrt{L^2 - x_i^2}} + \frac{1}{\sqrt{l^2 - x_i^2}}}.$$

З усього цього видно, яку лавину знань викликала вимога розв’язати задачу, в умовах якої вихідні дані подані дуже мізерною інформацією. Д. Пойа помітив, що ідея розв’язання задачі «може з'являтися поступово або вона може виникнути раптом в одну мить після, здавалося б, безуспішних спроб і тривалих сумнівів. Важко розраховувати на вдалу ідею, маючи слабкі знання з предмету, і неможливо знайти таку ідею, не маючи ніяких пізнань».

Приклад показує, що знання про предметну галузь, у якій породжується задача, являють собою складну структуру взаємозв'язку і взаємозумовленості фактів, параметри, характеристик. Це їхній взаємозв'язок і взаємозумовленість і дозволяє збудувати ланцюжок цілком конкретних операцій і висновків (тверджень, доказів), що призводять до деякого розв’язання виникаючої задачі.

Таким чином, створення експертних систем пов'язано з можливістю виявлення мінімально необхідної базової сукупності знань у предметній галузі припущених задач, створення умов поповнення, уточнення і зміни будь-яких фрагментів подання знань у пам'яті експертної системи і маніпуляції ними в процесі розв’язання задачі.

1.5.5 Подання знань

Класичні СОІ донедавна, і ще і зараз, будувалися і будуються на принципах фон Неймана. Вхідні дані і програми їх обробки подані в пам'яті ЕОМ у вигляді безликих двійкових кодів. Детермінований процес передбачає послідовну вибірку кодів команд і їхнє виконання, засноване фактично на трьох базових функціях алгебри логіки - *І*, *АБО*, *НІ*. Ніякої інтерпретації семантики вихідних даних, проміжних результатів або шуканої інформації в таких системах немає і бути не може.

Побудувати інтелектуальну експертну систему на подібному принципі не можна. Необхідно знайти спосіб відобразити, подати в пам'яті СОІ взаємозв'язок між інформацією, що надходить ззовні і необхідними знаннями, наявними в банку знань цієї СОІ.

Яким же чином повинні співвідноситися потоки вихідних даних і знання? До деякої міри прояснити це питання допоможе розглянутий приклад. З зовнішніх даних прямо не випливає, що колодязь круглий, проте, якщо в базі знань СОІ є інформація про те, що круглі предмети

характеризуються діаметром і що ніякі інші предмети (за формою) діаметра не мають, то твердження "знайти діаметр колодязя" достатньо для винесення судження "колодязь круглий". На рис. 1.3 ця ситуація ілюструється прикладом опису двох геометричних фігур.

Правда, прямого виразу - "діаметр мають тільки круглі фігури" - немає, але відповідна аналізуюча програма СОІ подібне твердження зможе синтезувати на базі сукупної характеристики об'єкта, йдучи по шляху, позначеному пунктирною лінією. Якщо поставити запитання - "що являє собою чотирикутник із довільними кутами", то за схемою можна одержати відповідь: "трапеція" (якщо йти по пунктирних лініях). Можна, очевидно, розв’язувати задачу типу: "чим характеризується трапеція?" або "що являє собою коло?".

Цілком імовірно, що приклад не дасть ніякого інструмента для подання знань в експертних системах. Але він дозволяє проілюструвати ті труднощі, що виникають у процесі формування знань, вибору найбільш загальних, фундаментальних відомостей про об'єкт, що дозволили б вирішувати більшість задач, які відносяться до даної предметної галузі.

Таким чином, перед творцем експертної системи виникає два головних питання:

1. Що повинно ввійти в банк знань? Відповідь на це питання окреслює предметну галузь і обмежує обсяг класу задач, які може розв’язувати експертна система. Так, у розглянутому прикладі можна включити знання про прямокутники, квадрати, ромби, трикутники і багатокутники, овали, еліпси і взагалі про всілякі геометричні фігури. Звичайно, це питання вирішується з урахуванням цілі створення експертної системи;
2. Як подавати знання? Це питання істотніше першого, по-перше, тому що ми повністю не знаємо того, як знання взагалі подаються в пам'яті людини; по-друге, принципи подання інформації в пам'яті обчислювальних систем засновані на класичному методі двійкового кодування з використанням не менш класичної пам'яті, її комірок;

   Практика розв’язання всіляких задач із використанням обчислювальної техніки (та й без неї) дозволяє сформулювати ряд позицій відносно взаємозв'язку фактів, подій, параметрів, властивостей об'єктів і їх складових, утворюючих те, що можна назвати знаннями, отже, і те, як їх подати в пам'яті експертної системи;

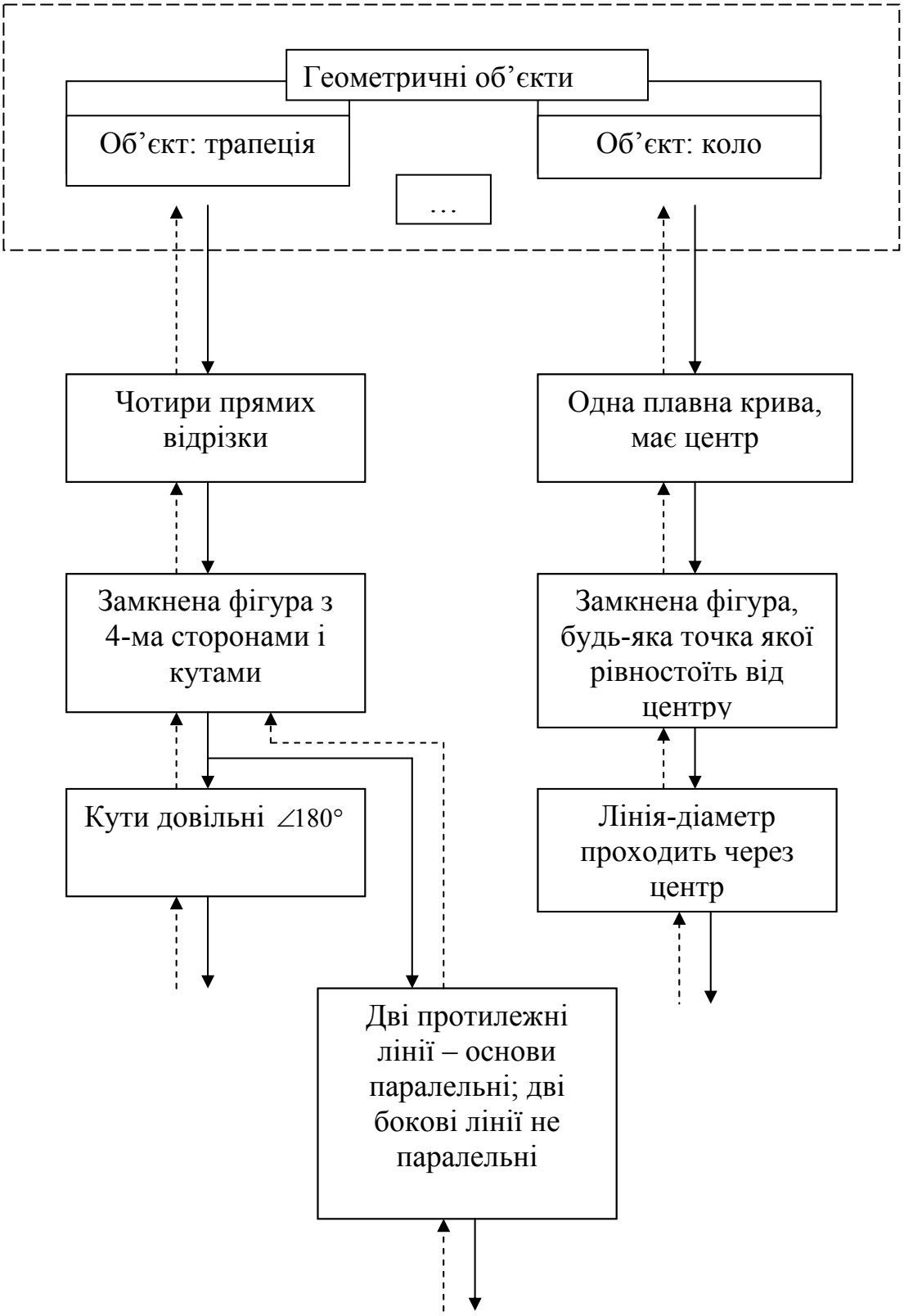


Рисунок 1.3 - До питання про подання знань

Знання являють собою структуровану сукупність інформації про явище, процеси, об'єкти, в якій чітко виявляється співпідпорядкованість елементів, параметрів, ознак і їхній взаємозв'язок. Це дозволяє розчленувати складне явище, об'єкт на сукупність найпростіших

складових, співпідпорядковуючи частину цілому, вигляд - роду на будь-якій ієрархічній сходинці такої декомпозиції. При цьому мова йде не тільки про просторову, але і тимчасову супідрядність - раніше, спільно, пізніше;

Знання не є «складом фактів» у пам'яті системи. Вони повинні носити характер рухливих, «активних» структур, що ініціюють вибір необхідного фрагменту, синтез, якщо його в знаннях немає, вилучення, якщо він суперечливий.

Достатньо потужною базою логічних моделей подання знань є *числення предикатів (calculation of predicates)*, що дозволяють виражати логічні зв'язки між різноманітними поняттями і твердженнями. Не будемо, проте, займатися докладним викладом теоретичних основ числень предикатів і принципів подання знань на цій базі, оскільки це зайняло б значну частину обсягу посібника. Проте приклад формування і запису предиката розглянемо.

Нехай $X=\{x_1,x_2,\ldots,x_i,\ldots,x_n\}$ – деяка множина різних геометричних фігур. Використовуючи елементи логіки предикатів можна записати, наприклад, таке твердження: «існують такі замкнуті геометричні фігури, які утворені чотирма прямими відрізками». Для цього задамо предикати: $A(x)$ - «геометрична фігура x замкнена»; $B(x)$ - «геометрична фігура x утворена чотирма відрізками». Тоді складовий предикат $A(x) \wedge B(x)$ стверджує, що «геометрична фігура x замкнена й утворена чотирма прямими відрізками», а вираз $(\exists x \in X)(A(x) \wedge B(x))$ і є твердження про існування ($\exists$ - квантор існування) замкнених геометричних фігур x множини X, що утворені чотирма прямими відрізками, тобто чотирикутники.

Проте числення предикатів являє собою жорстку формальну систему, у якій не можна здійснити необхідної рухливості і структурування знань. Цей недолік усувається створенням ієрархічної системи, в якій на першому рівні використовується логіка предикатів, а на другому - спеціальні засоби; при необхідності, перебудовують за деякими правилами вихідну множину елементів, характер виразів (формул), аксіоми і правила висновку, що спочатку були задані для першого ієрархічного рівня.

Розширення сфери застосування експертних систем потребує розробки адекватних методів подання знань. Серед ряду розробок цікаві і продуктивні семантичні моделі, теоретичною базою яких є мережі-графи, у яких кожній вершині приписуються сутності об'єктів, процесів, явищ, а дуги характеризують відношення між цими сутностями.

Дуже велика увага приділяється розробці й удосконалюванню **фрейм-моделей** (*frame* - остов, кістяк, каркас), відправним моментом для створення яких, як вказує автор [7], служить уявлення про погляд людини на будь-який об'єкт, явище, ситуацію. «…Людина, намагаючись пізнати

нову для себе ситуацію або по-новому глянути на вже звичні речі, вибирає зі своєї пам'яті деяку структуру даних (уява), названу нами фреймом…» На підтвердження цього можна навести ситуацію входу людини в незнайому кімнату. Він заздалегідь припускає побачити стіл, стільці, книжкову шафу, крісла, телевізор і т.д. Але ніколи він не очікує наткнутися на телевізор у вхідних дверях або зустріти в кімнаті живого слона. Його уявлення будуть відрізнятися від реальності, але предмети цієї реальності швидко знайдуть місця в створеній попередньо моделі. Головним структурним елементом фрейма є **слот** (*slot* - слід), що містить ім'я і значення поняття. При цьому як значення поняття може виступати інший фрейм, тобто фреймова структура може бути ієрархічною, що важливо при створенні фрейма складного явища або об'єкта. Наприклад, для фреймового опису задачі «жерців бога Ра» буде потрібно перерахувати слоти: *<колодязь>*, *<жердина L>*, *<жердина l>*, *<вода>*, а як значення повинні виступити: *<діаметр колодязя>*, *<довжина жердини>*, *<глибина води>*. Важливо, що у фреймі можна відображати поняття просторово-тимчасових відношень у семантичному аспекті, тобто у фреймі можна закласти відповіді на питання «що», «де», «коли», «як» і інші.

Цілком природно, що при будь-якому інструменті подання знань необхідні відповідні засоби їхньої активізації і використання, тобто мови подання знань, що були б адекватними обраному способові (моделі) і відповідній СОІ, її структурі, внутрішній мові. Такі мови розроблені й удосконалюються, як і способи подання знань.

1.5.6 Інтелектуальний інтерфейс

Взаємодія людини із СОІ будь-якого ступеня інтелектуальності - від класичної ЕОМ і до високоінтелектуальних експертних систем потребує адекватного інтерфейсу (від *inter* - між і *face* - стояти обличчям), тобто сукупність засобів, що забезпечують взаємодію двох систем. Узагальнена схема інтелектуальної СОІ подана на рис. 1.4.

Ступінь інтелектуальності інтерфейсу істотно впливає на характер "співдружності" користувача і СОІ. Слабка інтелектуальність інтерфейсу потребує кропіткої підготовки користувача, вивчення ним мов програмування високого рівня, мов керування завданнями й особливостей операційної системи і багато іншого.

Проблему підвищення швидкості запровадження вихідної інформації (користувацьких програм, даних про задачу, текстових документів) можна вирішити за допомогою читаючих автоматів, що виключають ручні операції з перекодування текстів програм і таблиць даних, які забезпечують істотне усунення диспропорції між швидкістю обробки інформації (мільйони і мільярди операцій у секунду) і її введенням із клавіатури (36 знаків у секунду). Цій цілі повинен служити й пристрій

технічного слуху, що дозволить оперативно взаємодіяти із СОІ в процесі введення інформації, а особливо в ході її обробки.

Інтерфейс повинний забезпечити можливість введення інформації з графіків, креслень, малюнків і фотографій. Крім того, важливо створити передумови діалогу між людиною-користувачем і СОІ на звичному інженерному, економічному, медичному і будь-якому іншому "жаргоні". При цьому вимоги інтерфейсу до користувача повинні бути мінімальними. Мало того, він повинен "підказувати" людині форму вираження думки, вказувати на помилки і їхній характер, якщо таких користувач допустився. Ця "доброзичливість" обов'язкова при будь-якому ступені підготовки користувача.

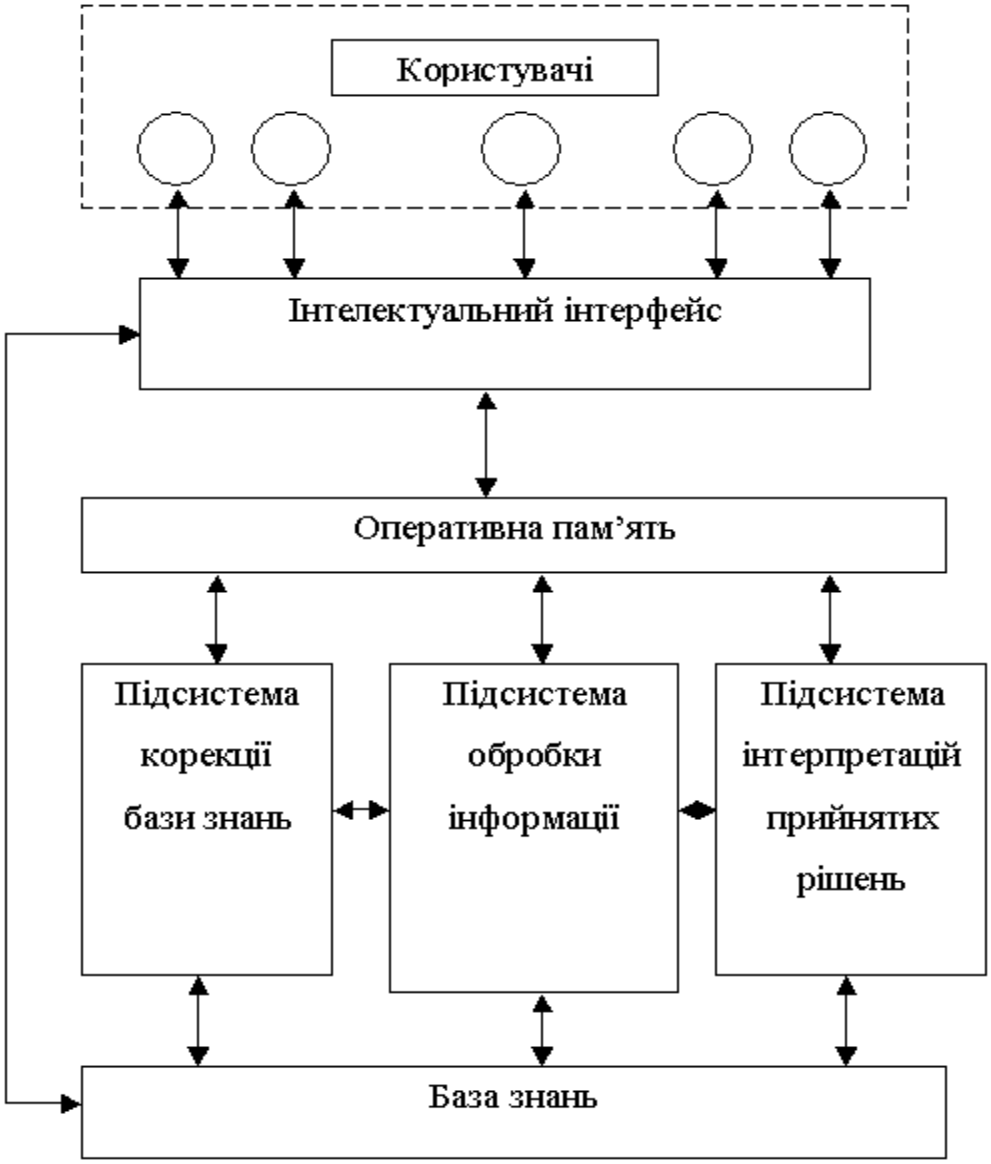


Рисунок 1.4 - Узагальнена схема СОІ

Таким чином, інтелектуальний інтерфейс повинен являти собою своєрідну експертну систему, що працює в темпі, характерному для

людини. Цілком природно, що подібні властивості потребують і відповідної бази знань. Саме тому на узагальненій схемі інтерфейс безпосередньо пов'язаний із базою знань (блоки 1 і 6) усієї системи. Дуже важливо забезпечити в інтерфейсі можливість взаємодії із СОІ багатьох користувачів одночасно. При цьому робочі місця користувачів можуть знаходитися на достатньому віддаленні від СОІ. І це важливо. Значимість інтелектуального інтерфейсу в експлуатації СОІ настільки велика, що у всіх розвинених країнах ставиться задача реалізації "інтелектуальних здатностей" не програмним методом, а апаратним. І це повинно бути реалізоване вже в ЕОМ п'ятого покоління.

Заключні зауваження

Інтелектуальні системи стали на порядок денний науки і практики в зв'язку з безупинним збільшенням потоків інформації, які людині доводиться переробляти в процесі своєї життєдіяльності. І справа стосується не тільки властивої людині "допитливості", але в багатьох випадках питання стоїть про життя і смерть. Тільки швидка і якісна обробка даних часом рятує від катастрофи військового, економічного або екологічного характеру.

Саме тому виникає потреба створювати системи технічного зору, слуху, дотику і нюху. Мало того, дуже багатими виявляються потоки інформації, що несуть сигнали, недоступні сприйняттю людини: інфрачервоні й ультрафіолетові випромінювання, ультра- і інфразвуки, радіоактивне випромінювання, радіохвилі всіх діапазонів і багато що інше. Ефективність обробки подібних сигналів тим вища, чим вищі інтелектуальні можливості СОІ.

Питання для самоконтролю

1. Які СОІ можна віднести до інтелектуальних?
2. Чим повинні відрізнятися ІС, призначені для роботи з людиною або технічною системою?
3. У чому сутність розпізнавання образів?
4. Що прийнято розуміти під класом об'єктів у проблемі розпізнавання образів?
5. Що являє собою задача ідентифікації?
6. Що таке "міра близькості"?
7. Що означають поняття "помилка першого роду" і "помилка другого роду"?
8. Чим, на вашу думку, відрізняються знання від набору даних?
9. Чим відрізняються логічні моделі подання знань від фреймових?
10. Які задачі повинен розв’язувати інтелектуальний інтерфейс у СОІ?

2 ЕЛЕМЕНТИ ТЕОРІЇ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

Штучний інтелект (ШІ) – це наука про концепції, що дозволяють ЕОМ робити такі речі, які для людей виглядають розумними. Але що ж являє собою інтелект людини? Чи є це здатність міркувати? Чи є це здатність засвоювати і використовувати знання? Чи є це здатність оперувати й обмінюватися ідеями? Безсумнівно, усі ці здібності являють собою частину того, що є інтелектом. Насправді дати означення в звичайному сенсі цього слова, очевидно, неможливо, тому що інтелект – це сплав багатьох навичок в галузі обробки і подання інформації.

Центральні задачі ШІ полягають у тому, що б зробити ЕОМ більш корисними і щоб зрозуміти принципи, що лежать в основі інтелекту. Оскільки одна з задач полягає в тому, щоб зробити ЕОМ більш корисними вченим і інженерам, що спеціалізуються в обчислювальній техніці, необхідно знати, яким чином ШІ може допомогти їм у вирішенні важких проблем.

2.1 Дані та знання

Подання знань — це вираження деякою формальною мовою, що називається *мовою подання знань* (МПЗ), властивостей різних об'єктів і закономірностей, важливих для розв’язування прикладних задач і організації взаємодії користувача з ЕОМ. Це можуть бути об’єкти і закономірності предметної галузі, обчислювального середовища і т.д. Той факт, що мова, якою записуються знання, є формальною, забезпечує однозначність інтерпретації записаного, так само як формальна мова конструкторського креслення забезпечує тотожність конструкцій, вироблених на основі того самого креслення навіть різними людьми.

Сукупність знань, що зберігаються в ОС і необхідних для розв’язання комплексу прикладних задач кінцевим користувачем, називається *системою* знань. Зведення про те, які знання закладені в систему, можуть знадобитися користувачу (і він повинен мати можливість їх отримати), однак у першу чергу організовані знання необхідні обчислювальній системі для того, щоб підтримувати процес взаємодії з користувачем і розв’язувати необхідні задачі. Іншими словами, знання забезпечують функціонування системи.

Обчислювальна система використовує систему знань, виконуючи над нею різноманітні дії (операції), такі як пошук необхідних зведень, їхня модифікація, інтерпретація знань, отримання з наявних знань нових і т.п. Алгоритми виконання цих операцій істотно залежать від особливостей мови подання знань і від того, яким чином система знань реалізується.

Оскільки система знань коштовна не сама по собі, а саме можливостями її використання, оскільки використовувати цю систему можна лише виконуючи над нею ті чи інші операції й оскільки алгоритміка цих операцій визначається особливостями МПЗ, будь-який сучасний метод

подання знань являє собою сукупність взаємозалежних засобів формального опису знань і оперування (маніпулювання) цими описами.

Відповідно МПЗ повинна забезпечувати можливості не тільки формального запису необхідних знань, але і маніпулювання цими записами. При цьому для будь-якого методу подання знань природно виділяються основні (базові) операції (примітиви), з яких при необхідності можуть формуватися більш складні процедури.

Довгий час роботи з подання знань проводилися паралельно в двох напрямках досліджень: в галузі штучного інтелекту й у технології баз даних. Результати, отримані в згаданих напрямках досліджень, істотно доповнюють один одного. Більшість дослідників обох напрямків вважають інтеграцію цих результатів однією з найважливіших перспектив розвитку робіт із подання знань [24, 25].

Інформація, з якою мають справу ЕОМ, розділяється на *процедурну* і *декларативну*. Процедурна інформація зосереджена в *програмах*, що виконуються в процесі розв’язання задач, декларативна інформація - у *даних*, з якими ці програми працюють. Стандартною формою подання інформації в ЕОМ є *машинне слово*, що складається з певного для даного типу ЕОМ числа двійкових розрядів - *бітів*. Машинне слово для подання даних і машинне слово для подання команд, що утворюють програму, можуть мати однакове або різне число розрядів. Останнім часом для подання даних і команд використовуються однакові за числом розрядів машинні слова. Однак у ряді випадків машинні слова розбиваються на групи по вісім двійкових розрядів, що називаються *байтами*.

Особливості знань:

1. *Внутрішня інтерпретація*. Кожна інформаційна одиниця повинна мати унікальне ім'я, за яким ІС знаходить її, а також відповідає на запити, у яких це ім'я згадане. Коли дані, що зберігаються в пам'яті, були позбавлені імен, то була відсутня можливість їхньої ідентифікації системою. Дані могла ідентифікувати лише програма, що "витягає" їх з пам'яті за вказівкою програміста, який написав програму. Що ховається за тим або іншим двійковим кодом машинного слова, системі було невідомо.

Якщо, наприклад, у пам'ять ЕОМ потрібно було записати відомості про співробітників установи, подані в табл. 2.1, то без внутрішньої інтерпретації в пам'ять ЕОМ була б занесена сукупність з чотирьох машинних слів, які відповідають рядкам цієї таблиці. При цьому інформація про те, якими групами двійкових розрядів у цих машинних словах закодовані відомості про фахівців, у системи відсутня. Вона відома лише програмістові, що використовує дані табл. 2.1 для розв’язання виникаючих у нього задач. Система не в змозі відповісти на питання типу "Що тобі відомо про Петрова?" або "Чи є серед фахівців сантехнік?" [26].

При переході до знань у пам'ять ЕОМ вводиться інформація про деяку протоструктуру інформаційних одиниць. У розглянутому прикладі

вона являє собою спеціальне машинне слово, де зазначено, у яких розрядах зберігаються відомості про прізвища, роки народження, спеціальності і стажі. При цьому повинні бути задані спеціальні словники, у яких перераховані наявні в пам'яті системи прізвища, роки народження, спеціальності і тривалості стажу. Всі ці атрибути можуть відігравати роль імен для тих машинних слів, що відповідають рядкам таблиці. За ними можна здійснювати пошук потрібної інформації. Кожен рядок таблиці буде екземпляром протоструктури. В даний час СУБД забезпечують реалізацію внутрішньої інтерпретації всіх інформаційних одиниць, що зберігаються в базі даних.

Таблиця 2.1 – Відомості про співробітників

Прізвище	**Рік народження**	**Спеціальність**	**Стаж, число років**
Попов	1965	Слюсар	5
Сидоров	1946	Токар	20
Іванов	1925	Токар	30
Петров	1937	Сантехнік	25

2. *Структурованість*. Інформаційні одиниці повинні мати гнучку структуру. Для них повинен виконуватися "принцип матрьошки", тобто рекурсивна вкладеність одних інформаційних одиниць в інші. Кожна інформаційна одиниця може бути включена до складу будь-якої іншої, і з кожної інформаційної одиниці можна виділити деякі складові її інформаційні одиниці. Іншими словами, повинна існувати можливість довільного встановлення між окремими інформаційними одиницями відношень типу "частина - ціле", "рід - вид" або "елемент - клас".

3. *Зв'язність*. В інформаційній базі між інформаційними одиницями повинна бути передбачена можливість установлення зв'язків різного типу. Насамперед ці зв'язки можуть характеризувати відношення між інформаційними одиницями. Семантика відносин може носити декларативний або процедурний характер. Наприклад, дві або більше інформаційні одиниці можуть бути пов'язані відношенням "одночасно", дві інформаційні одиниці - відношенням "причина - наслідок" або відношенням "бути поруч". Наведені відношення характеризують декларативні знання. Якщо між двома інформаційними одиницями встановлено відношення "аргумент - функція", то воно характеризує процедурне знання, пов'язане з обчисленням певних функцій. Далі будемо розрізняти відношення структуризації, функціональні відношення, каузальні відношення і семантичні відношення. За допомогою перших задаються ієрархії інформаційних одиниць, другі несуть процедурну інформацію, що дозволяє знаходити (обчислювати) одні інформаційні одиниці через інші, треті задають причинно-наслідкові зв'язки, четверті

відповідають всім іншим відношенням.

Між інформаційними одиницями можуть встановлюватися й інші зв'язки, наприклад, що визначають порядок вибору інформаційних одиниць з пам'яті або вказують на те, що дві інформаційні одиниці несумісні одна з одною в одному описі.

Перераховані три особливості знань дозволяють запровадити загальну модель подання знань, яку можна назвати семантичною мережею, що являє собою ієрархічну мережу, у вершинах якої знаходяться інформаційні одиниці. Ці одиниці мають індивідуальні імена. Дуги семантичної мережі відповідають різним зв'язкам між інформаційними одиницями. При цьому ієрархічні зв'язки визначаються відношеннями структуризації, а неієрархічні зв'язки - відношеннями інших типів.

4. *Семантична метрика*. На безлічі інформаційних одиниць у деяких випадках корисно задавати відношення, що характеризує ситуаційну близькість інформаційних одиниць, тобто силу асоціативного зв'язку між інформаційними одиницями. Його можна було б назвати відношенням релевантності для інформаційних одиниць. Таке відношення дає можливість виділяти в інформаційній базі деякі типові ситуації (наприклад, "покупка", "регулювання руху на перехресті"). Відношення релевантності при роботі з інформаційними одиницями дозволяє знаходити знання, близькі до вже знайдених.

5. Активність. З моменту появи ЕОМ і поділу використовуваних у ній інформаційних одиниць на дані і команди створилася ситуація, при якій дані – пасивні, а команди – активні. Усі процеси, що протікають в ЕОМ, ініціюються командами, а дані використовуються цими командами лише в разі потреби. Для ІС ця ситуація не прийнятна. Як і в людини, у ІС актуалізації тих або інших дій сприяють знання, що є в системі. Таким чином, виконання програм у ІС повинно ініціюватися поточним станом інформаційної бази. Поява в базі фактів або описів подій, установлення зв'язків може стати джерелом активності системи.

2.1.1 Подання даних і знань

Існують два типи методів подання знань (ПЗ):

1. Формальні моделі ПЗ;

2. Неформальні (семантичні, реляційні) моделі ПЗ.

Очевидно, що усі методи подання знань, які розглянуто вище, включаючи продукції (це система правил, на яких заснована продукційна модель подання знань), відносяться до неформальних моделей. На відміну від формальних моделей, в основі яких лежить строга математична теорія, неформальні моделі такої теорії не дотримують. Кожна неформальна модель годиться тільки для конкретної предметної галузі і тому не має універсальності, яка властива моделям формальним. Логічний висновок - основна операція в системах штучного інтелекту (СШІ) - у формальних

системах строгий і коректний, оскільки підлягає твердим аксіоматичним правилам. Висновок у неформальних системах багато в чому визначається самим дослідником, що і відповідає за його коректність.

Кожному з методів ПЗ відповідає свій спосіб опису знань.

2.1.1.1 Логічні моделі

В основі моделей такого типу лежить формальна система, що задається четвіркою виду: $M = <T, P, A, B>$. Множина T є множиною базових елементів різної природи, наприклад, слів з деякого обмеженого словника, деталей дитячого конструктора, що входять до складу деякого набору і т.п. Важливо, що для множини T існує деякий спосіб визначення належності або неналежності довільного елемента до цієї множини. Процедура такої перевірки може бути будь-якою, але за кінцеве число кроків вона повинна давати позитивну або негативну відповідь на питання, чи є х елементом множини T. Позначимо цю процедуру $П(T)$.

Множина P є множиною *синтаксичних правил*. З їхньою допомогою з елементів T утворять *синтаксично правильні сукупності*. Наприклад, зі слів обмеженого словника будуються синтаксично правильні фрази, з деталей дитячого конструктора за допомогою гайок і болтів збираються нові конструкції. Декларується існування процедури $П(P)$, за допомогою якої за кінцеве число кроків можна одержати відповідь на питання, чи є сукупність X синтаксично правильною.

У множині синтаксично правильних сукупностей виділяється деяка підмножина A. Елементи A називаються *аксіомами*. Як і для інших складових формальної системи, повинна існувати процедура $П(A)$, за допомогою якої для будь-якої синтаксично правильної сукупності можна одержати відповідь на питання про належність її до множини A.

Множина B є множиною *правил висновку*. Застосовуючи їх до елементів A, можна одержувати нові синтаксично правильні сукупності, до яких знову можна застосовувати правила з B. Так формується *множина виведених* у даній формальній системі *сукупностей*. Якщо є процедура $П(B)$, за допомогою якої можна визначити для будь-якої синтаксично правильної сукупності, чи є вона виведеною, то відповідна формальна система називається *розв'язною*. Це показує, що саме правило висновку є найбільш складною складовою формальної системи.

Для знань, що входять у базу знань, можна вважати, що множина A утворює всі інформаційні одиниці, які введені в базу знань ззовні, а за допомогою правил висновку з них виводяться нові *похідні знання*. Іншими словами, формальна система являє собою генератор породження нових знань, що утворять множину *виведених* у даній системі *знань*. Ця властивість логічних моделей робить їх придатними для використання в базах знань. Вона дозволяє зберігати в базі лише ті знання, що утворюють множину A, а всі інші знання одержувати з них за правилами висновку.

2.1.1.2 Мережеві моделі

В основі моделей цього типу лежить конструкція, названа раніше семантичною мережею. Мережеві моделі формально можна задати у вигляді $H = <I, C_1, C_2, ..., C_n, Г>$. Тут I є множиною інформаційних одиниць; $C_1, C_2, ..., C_n$ - множина типів зв'язків між інформаційними одиницями. Відображення Г задає між інформаційними одиницями, що входять у I, зв'язки з заданого набору типів зв'язків.

Залежно від типів зв'язків, використовуваних у моделі, розрізняють *класифікувальні* мережі, *функціональні* мережі і сценарії. У класифікувальних мережах використовуються відносини структуризації. Такі мережі дозволяють у базах знань уводити різні ієрархічні відношення між інформаційними одиницями. Функціональні мережі характеризуються наявністю функціональних відношень. Їх часто називають *обчислювальними моделями*, тому що вони дозволяють описувати процедури "обчислень" одних інформаційних одиниць через інші. У сценаріях використовуються каузальні відношення, а також відношення типів "засіб - результат", "знаряддя - дія" і т.п. Якщо в мережевій моделі допускаються зв'язки різного типу, то її звичайно називають семантичною мережею.

2.1.1.3 Продукційні моделі

У моделях цього типу використовуються деякі елементи логічних і мережевих моделей. З логічних моделей запозичена ідея правил висновку, що називаються *продукціями*, а з мережевих моделей - опис знань у вигляді семантичної мережі. У результаті застосування правил висновку до фрагментів мережевого опису відбувається трансформація семантичної мережі за рахунок зміни її фрагментів, нарощування мережі і виключення з неї непотрібних фрагментів. Таким чином, у продукційних моделях процедурна інформація явно виділена й описується іншими засобами, ніж декларативна інформація. Замість логічного висновку, характерного для логічних моделей, у продукційних моделях з'являється *висновок на знаннях*.

2.1.1.4 Фреймові моделі

На відміну від моделей інших типів у фреймових моделях фіксується тверда структура інформаційних одиниць, що називається *протофреймом*. У загальному виді вона виглядає в такий спосіб:

(Ім'я фрейма:

Ім'я слота 1(значення слота 1)

Ім'я слота 2(значення слота 2)

Ім'я слота К (значення слота К)).

Значенням *слота* може бути практично що завгодно (числа або математичні співвідношення, тексти природною мовою або програми,

правила висновку або посилання на інші слоти даного фрейма або інших фреймів). Як значення слота може виступати набір слотів більш низького рівня, що дозволяє у фреймових подання реалізувати "принцип матрьошки".

При конкретизації фрейму йому і слотам присвоюються конкретні імена і відбувається заповнення слотів. Таким чином, із протофреймів виходять *фрейми - екземпляри.* Перехід від вихідного протофрейма до фрейму - екземпляру може бути багатокроковим, за рахунок поступового уточнення значень слотів [24, 25].

Наприклад, структура табл. 2.1, записана у вигляді протофрейма, має вигляд

(Список працівників:

Прізвище (значення слота 1);

Рік народження (значення слота 2);

Спеціальність (значення слота 3);

Стаж (значення слота 4)).

Якщо як значення слотів використовувати дані табл. 2.1, то вийде фрейм - екземпляр

(Список працівників:

Прізвище (Попов - Сидоров - Іванов - Петров);

Рік народження (1965 - 1946 - 1925 - 1937);

Спеціальність (слюсар - токар - токар - сантехнік);

Стаж (5 - 20 - 30 - 25)).

Зв'язки між фреймами задаються значеннями спеціального слота з ім'ям "Зв'язок". Частина фахівців з ІС вважає, що немає необхідності спеціально виділяти фреймові моделі в поданні знань, тому що в них об'єднані всі основні особливості моделей інших типів.

2.1.2 Формальні моделі подання знань

Система ШІ в певному змісті моделює інтелектуальну діяльність людини і, зокрема, - логіку її міркувань. У грубо спрощеній формі наші логічні побудови при цьому зводяться до такої схеми: з однієї або декількох посилок (які вважаються правдивими) варто зробити "логічно правильний" висновок (висновок, наслідок). Очевидно, для цього необхідно, щоб і посилки, і висновок були подані зрозумілою мовою, що адекватно відбиває предметну галузь, з якої робимо висновок. У звичайному житті – це наша природна мова спілкування, у математиці, наприклад, це мова певних формул і т.п. Наявність же мови припускає, по-перше, наявність алфавіту (словника), що відображає в символьній формі весь набір базових понять (елементів), з якими доведеться мати справу і, по-друге, набір синтаксичних правил, на основі яких, користуючись алфавітом, можна побудувати певні вираження.

Логічні вирази, побудовані в даній мові, можуть бути щирими або помилковими. Деякі з цих виразів є завжди правдивими. Оголошуються *аксіомами* (або *постулатами*). Вони складають ту базову систему посилок, виходячи з якої і користуючись певними правилами висновку, можна одержати висновки у виді нових виразів, що також є правдивими.

Якщо перераховані умови виконуються, то говорять, що система задовольняє вимоги *формальної теорії.* Її так і називають *формальною системою* (ФС). Система, побудована на основі формальної теорії, називається також *аксіоматичною системою.*

Формальна теорія повинна, таким чином, задовольняти таке означення:

усяка формальна теорія *F = (A, V, W, R)*, що визначає деяку аксіоматичну систему, характеризується:

- наявністю алфавіту (словника), *A*,
- множиною синтаксичних правил, *V*,
- множиною аксіом, що лежать в основі теорії, *W*,
- множиною правил висновку, *R*.

Вирахування висловлювань (ВВ) і вирахування предикатів (ВП) є класичними прикладами аксіоматичних систем. Ці ФС добре досліджені і мають прекрасно розроблені моделі логічного висновку - головної метапроцедури в інтелектуальних системах. Тому усе, що може і гарантує кожна з цих систем, гарантується і для прикладних ФС як моделей конкретних предметних галузей. Зокрема, це гарантії несуперечності висновку, алгоритмічної можливості розв'язання (для вирахування висловлень) і напівроз’язності (для вирахувань предикатів першого порядку).

ФС мають і недоліки, що змушують шукати інші форми подання. Головний недолік - це "закритість" ФС, їхня негнучкість. Модифікація і розширення тут завжди пов'язані з перебудовою усієї ФС, що для практичних систем складно і трудомістко. У них дуже складно враховувати зміни, що відбуваються. Тому ФС як моделі подання знань використовуються в тих предметних галузях, які добре локалізуються і мало залежать від зовнішніх факторів.

Також одним зі способів подання знань є мова математичної логіки, що дозволяє формально описувати поняття предметної галузі і зв'язку між ними.

На відміну від природної мови, що є дуже складною, мова логіки предикатів використовує тільки такі конструкції природної мови, що легко формалізуються.

Тобто логіка предикатів - це мовна система, що оперує з пропозиціями на природній мові в межах синтаксичних правил цієї мови.

Мова логіки предикатів використовує слова, що описують:

– поняття й об'єкти досліджуваної предметної галузі;

– властивості цих об'єктів і понять, а також їхню поведінку і відношення між ними.

У термінах логіки предикатів перший тип слів називається термами, а другий - предикатами.

Терми являють собою засоби для позначення цікавлячих нас індивідуумів, а предикати виражають відношення між індивідуумами (які позначаються за допомогою термів).

Логічна модель - це безліч пропозицій, що виражають різні логічні властивості іменованих відношень.

При логічному програмуванні користувач описує предметну галузь сукупністю пропозицій у вигляді логічних формул, а ЕОМ, маніпулюючи цими пропозиціями, будує необхідний для розв’язання задач висновок.

2.2 Бази знань і принципи їх формування

Однакове число розрядів у машинних словах для команд і даних дозволяє розглядати їх в ЕОМ у якості однакових інформаційних одиниць і виконувати операції над командами як над даними. Вміст пам'яті утворює *інформаційну базу* [26, 27].

У більшості існуючих ЕОМ можливе добування інформації з будь-якої підмножини розрядів машинного слова аж до одного біта. У багатьох ЕОМ можна з’єднувати два або більше машинних слова в слово з більшою довжиною. Однак машинне слово є основною характеристикою інформаційної бази, тому що його довжина така, що кожне машинне слово зберігається в одній стандартній комірці пам'яті з індивідуальним ім'ям - адресою комірки. За цим іменем відбувається добування інформаційних одиниць з пам'яті ЕОМ і запису їх у неї.

Паралельно з розвитком структури ЕОМ відбувався розвиток інформаційних структур для подання даних. З’явилися способи опису даних у виді векторів і матриць, виникли спискові структури, ієрархічні структури. В даний час у мовах програмування високого рівня використовуються *абстрактні типи даних*, структура яких задається програмістом. Поява *баз даних* (БД) знаменувала собою ще один крок на шляху організації роботи з декларативною інформацією. У базах даних можуть одночасно зберігатися великі обсяги інформації, а спеціальні засоби, що утворюють *систему управління базами даних* (СУБД), дозволяють ефективно маніпулювати даними, при необхідності витягати їх з бази даних і записувати їх у потрібному порядку в базу.

В міру розвитку досліджень в галузі ІС виникла концепція знань, що об'єднали в собі багато рис процедурної і декларативної інформації.

В ЕОМ *знання* так само, як і *дані*, відображаються в знаковій формі - у вигляді формул, тексту, файлів, інформаційних масивів і т.п. Тому можна сказати, що знання - це особливим чином організовані дані. Але це було б занадто вузьке розуміння. А тим часом, у системах ШІ знання є основним

об'єктом формування, обробки і дослідження. *База знань*, нарівні з базою даних, - необхідна складова програмного комплексу ШІ. Машини, що реалізують алгоритми ШІ, називаються *машинами, заснованими на знаннях*, а підрозділ теорії ШІ, пов'язаний з побудовою експертних систем - *інженерією знань.*

Інженерія знань - це галузь інформаційної технології, ціль якої - накопичувати і застосовувати знання не як об'єкт обробки їх людиною, а як об'єкт для обробки на комп'ютері. Для цього необхідно проаналізувати знання й особливості їхньої обробки людиною і комп'ютером, а також розробити їхнє машинне подання. На жаль точного і незаперечного означення, що собою являють знання, дотепер не дано. Але проте ціль інженерії знань - забезпечити використання знань у комп'ютерних системах на більш високому рівні, чим дотепер - актуальна. Але варто помітити, що можливість використання знань здійсненна тільки тоді, коли ці знання існують, що цілком з'ясовно. Технологія нагромадження і підсумовування знань йде пліч-о-пліч з технологією використання знань, вони взаємно доповнюють один одного і ведуть до створення однієї технології, технології обробки знань.

Щоб підготуватися до наступного обговорення програм, побудованих на основі знань, розглянемо більш формалізовані ті принципи, на яких базуються подання і керування. Дослідження, що мають пряме відношення до експертних систем, були проведені протягом останніх 20 років у межах наукового напрямку, який одержав назву "штучний інтелект" і безпосередньо пов'язаного з розробкою комп'ютерних програм для автоматизації діяльності, що вимагає людського інтелекту. Так, наприклад, були розроблені програми для гри в шахи чи для виявлення дефектних деталей двигунів. Однак передісторія цієї дисципліни відноситься до однієї з перших академічних дисциплін - логіки.

Логіка має справу головним чином з виявленням обґрунтованості тверджень, тобто з методами, що дозволяють довести, чи можна даний висновок обґрунтовано вивести виходячи з відомих фактів. Більш того, логіка безпосередньо пов'язана з програмуванням, оскільки будь-яка програма, власне кажучи, являє собою набір квазілогічних тверджень, що певним чином обробляються з метою одержання деякого висновку. У рамках логіки для вираження "твердження істинне" існує точне, конкретне значення; твердження вважається істинним, якщо (і тільки якщо) стосовні до нього припущення усі істинні, при цьому висновок самого твердження також істинний.

Для ухвалення рішення про прийнятність будь-якого конкретного твердження необхідно зробити перевірку. У рамках логіки такий метод зводиться до порівняння цікавлячого нас тексту з абстрактними моделями твердження в пошуках придатного. Моделі твердження трактуються як

"форми" і збираються з абстрагованих послідовностей фактів і правил, обґрунтованість яких була раніше доведена математично (чи "формально"). Звернемося до приклада. Припустимо, дане правило:

"ЯКЩО Сміт є фахівцем з ЕОМ, ТО Сміт - оптиміст"

Допустимо такий простий факт:

"Сміт є фахівцем з ЕОМ",

тоді може показатися природним висновок

"Сміт - оптиміст".

Більш формально виразимо це, застосувавши логічну модель:

ЯКЩО P, ТО Q, P ОТЖЕ Q,

де *P* и *Q* означає відповідно два речення: "Сміт є фахівцем з ЕОМ" і "Сміт - оптиміст". Знайшовши збіг, ми можемо стверджувати, що судження має деяку прийнятну логічну структуру, і отже, можна сказати, що висновок "Сміт - оптиміст" справедливо. Повністю твердження виглядає в такий спосіб:

ЯКЩО Сміт є фахівцем з ЕОМ, ТО Сміт - оптиміст
Сміт є фахівцем з ЕОМ
ОТЖЕ
Сміт – оптиміст.

З цього приводу потрібно помітити, що речення, які замінили наведені в "формі" букви, називаються "змістом" твердження. На обґрунтованості приклада ніяк не позначається, чи має зміст твердження якийсь зміст у реальному світі. Наприклад, таке твердження справедливе, хоча й абсурдне:

ЯКЩО Сміт є фахівцем з ЕОМ, ТО Сміт спить
Сміт є фахівцем з ЕОМ
ОТЖЕ
Сміт спить.

Використана в наведених прикладах проста "форма" твердження вважається в логіці однією з основних, і їй дана спеціальна назва латинською мовою "modus ponendo ponens" – назва першої форми гіпотетичного силогізму, що виражається такою формулою: якщо *A* є *B*, то *C* є *D* (скорочено "modus ponens"). Вона тісно пов'язана з конкретним підходом, що використовує продукційні системи для обробки знань. Взаємозв'язок полягає в такому. Правило "ЯКЩО P, ТО Q" відповідає одному з правил продукції; одиничне висловлювання P відповідає деякому факту, зафіксованому в базі даних системи, що при виявленні збігу для частини правила ЯКЩО фіксує твердження Q як новий факт. Більш того, повний цикл обчислень, проведених за допомогою простої продукційної

системи, відповідає багаторазовому застосуванню твердження "modus ponens"; породжене твердження, у свою чергу, як факт фіксується в базі даних. Отже, подібна логіка має здатність як породжувати, так і оцінювати твердження.

Здатність логіки до породження (чи "логічному висновку") нової інформації на підставі старої становить певний інтерес в аспекті програмування як засіб для керованого породження логічного висновку. Як буде показано нижче, у логіці розроблені добре чіткі і зрозумілі формалізми для подання фактів і правил маніпулювання ними.

Проблематика технології баз даних не обмежується питаннями подання знань.

2.3 Формалізація та інтерпретація знань

Важливим етапом при створенні БЗ є етап здобуття знань. На цьому етапі різноманітний набір фактів про деякий предмет повинен бути поданий у вигляді деякої узагальненої структури. Однією з них є структура, що одержала назву «дерево рішень».

Це один з найпростіших способів подання фактів і його застосування обмежене. Разом з тим, використання дерева рішень може бути ефективно там, де знання представляються у виді правил.

Даний підхід буде розглянутий з єдиною метою: показати, як знання про конкретну предметну галузь можуть бути формалізовані до рівня структури БЗ деякої експертної системи.

Структура дерева рішень ілюструє відношення, які повинні бути встановлені між правилами в добре організованій БЗ. Даний підхід можна реалізувати в системі „Мікроексперт” для IBM PC і багатьох інших, більш сучасних оболонках ЕС [28-30].

2.3.1 Формалізація задачі

Уявімо собі, що ми присутні при бесіді, коли фахівця в галузі ботаніки по телефону просять визначити тип деякої рослини. Через те що він не бачить конкретний екземпляр, а запитувач не є фахівцем в галузі ботаніки, консультуючий задає ряд питань, щоб одержати відомості, необхідні для розв’язання задачі [31, 32].

Один з варіантів такої консультації може бути поданий у графічній формі (рис. 2.1).

Результатом такої консультації буде висновок: «Ґрунтуючись на Вашій відповіді, можна припустити, що тип рослини – дерево».

Однак дана діаграма ілюструє тільки один з можливих варіантів питань і відповідей (тобто експертизи). Аналогічним образом можна подати і всі інші варіанти відповідей, і хід консультації.

Рисунок 2.1 - Телефонна консультація

Що ж дозволяє фахівцю провести таку консультацію і визначити як послідовність питань, так і їхній зміст залежно від відповідей опитуваного? Відповідь одна: його знання в конкретній предметній галузі, у якій він є фахівцем (експертом).

Базуючись на знаннях експерта графічно діаграму всіх можливих результатів даної консультації можна подати у вигляді рис. 2.2.

Це графічне зображення моделі даних називається «деревом рішення», що поєднує всі галузі пошуку типу «невідома рослина».

Але якщо консультація ЕС повинна бути більш глибокою і визначати, наприклад, клас рослини, то в цьому випадку для «типу рослини – дерево» повинне бути побудоване своє «дерево рішень», що після одержання знань від фахівця можна зобразити у вигляді рис. 2.3.

Як видно, нова частина буде «піддеревом» вихідного «дерева рішень».

Існує кілька причин, через які усе «дерево рішень» розбивається на секції:

- «дерево рішень» швидко стає довгим і ненаочним;

- розподіл «дерева рішень» на секції спрощує запам'ятовування мети, що переслідується в процесі здобуття знань.

Коли «піддерево» створено, заключна його частина може бути скопійована в корінь знову створюваної галузі «дерева рішень», і для неї на основі знань, одержуваних від експерта, може бути побудоване своє «піддерево рішень» (рис. 2.4) і т.д.

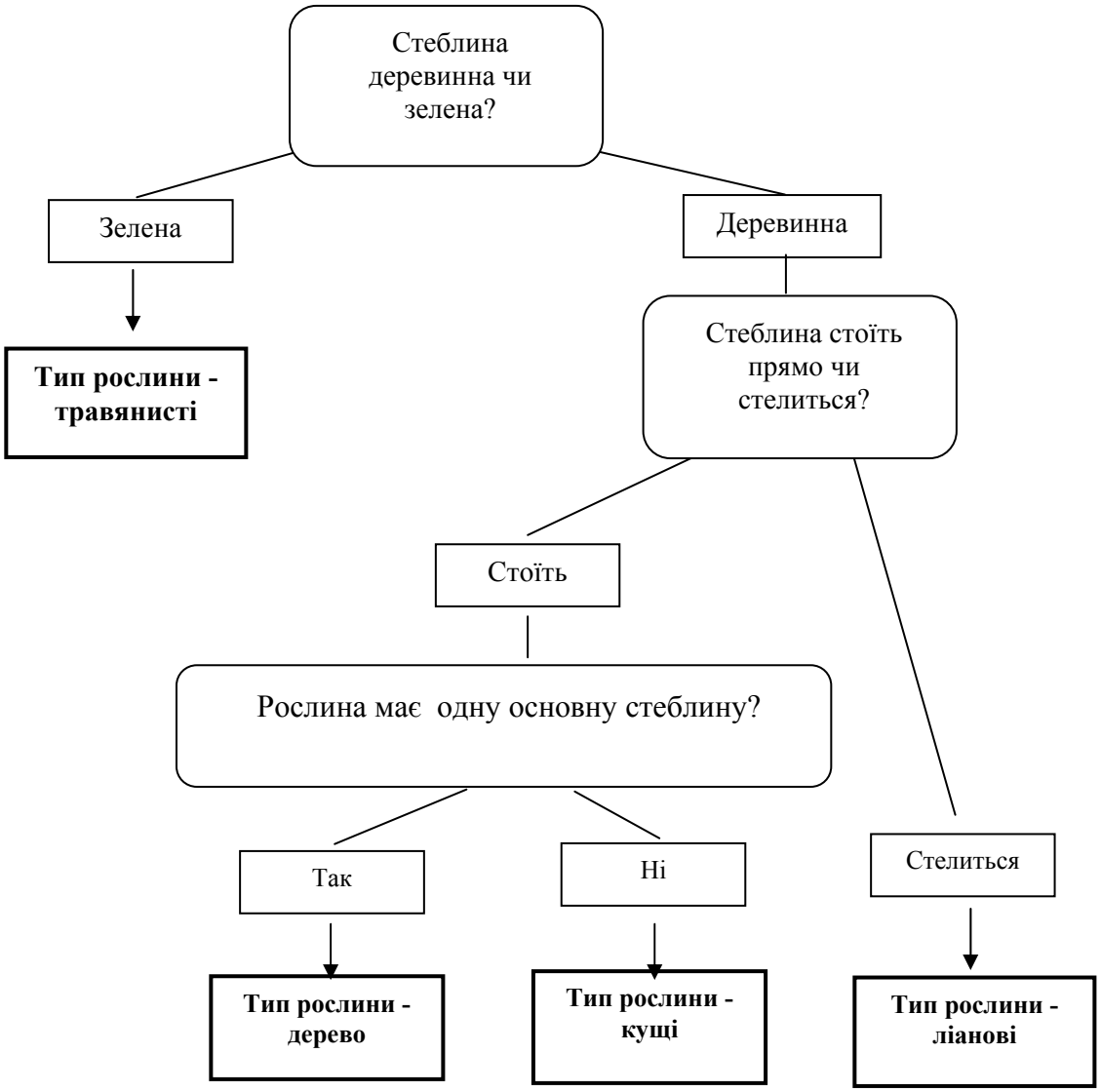


Рисунок 2.2 - Дерево розв’язання задачі

Діаграми, наведені на рис. 2.2÷2.4 - це модель незакінченої ботанічної БЗ, що вирішує тільки вузьку частину загальної задачі.

На основі викладеного можна зробити висновок, що при розробці моделі БЗ будь-якої предметної галузі на основі «дерева рішень» необхідно:

- загальну задачу розбити на ряд підзадач;
- для кожної з підзадач розробити своє «дерево рішень» (це спростить створення і налагодження БЗ).

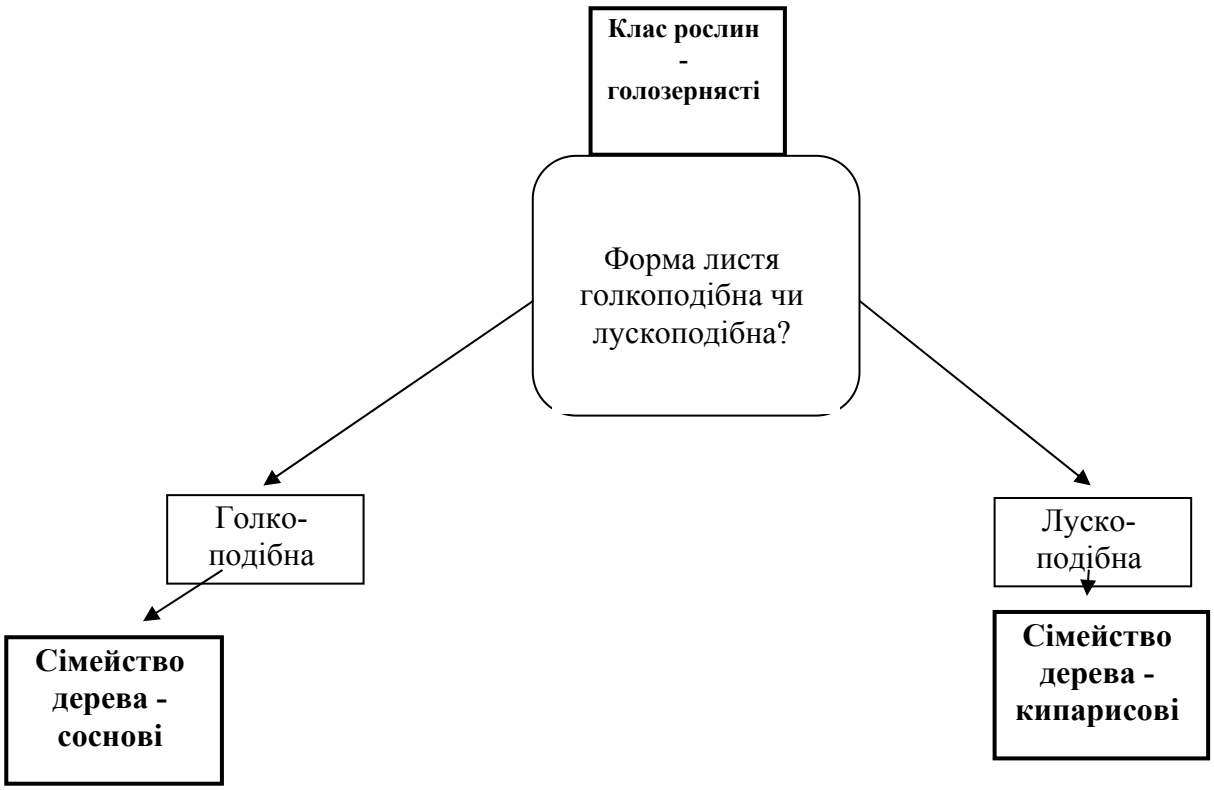


Рисунок 2.3 - «Піддерево-1» розв’язання задачі

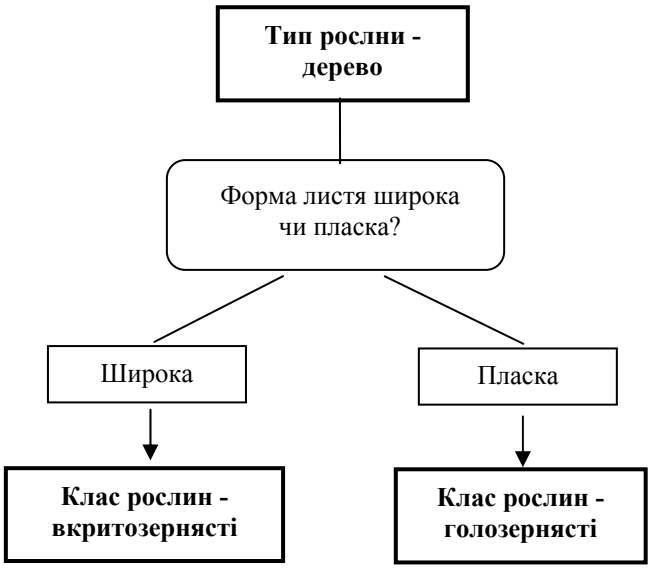


Рисунок 2.4 - «Піддерево-2» розв’язання задачі

2.3.2 Інтеграція методів представлення знань

У практиці побудови інтелектуальних систем усі частіше виникає потреба у використанні переваг описаних методів подання знань і компенсації їхніх недоліків.

Сучасний етап розвитку інструментальних засобів створення інтелектуальних систем характеризується інтеграцією — забезпеченням можливостей спільного використання різних методів подання знань у рамках єдиного програмного продукту. Основою для такої інтеграції послужила концепція об’єктно-орієнтованого програмування.

В об’єктно-орієнтованому програмуванні програма подається набором активних об'єктів — комплексів, що складаються зі структури даних і сукупності процедур. Об’єкти здатні видавати (іншим об'єктам) і приймати (від інших об'єктів) повідомлення, а також виконувати ті чи інші маніпуляції зі структурами даних відповідно до прийнятих повідомлень. Структурно об'єкти можуть розглядатися як фрейми, постачені асоційованими процедурами. Виділяються процедури двох типів: *методи* і *демони.* Кожен метод призначений для обробки повідомлень певного типу й ініціюється тоді і тільки тоді, коли об’єкт одержує повідомлення цього типу. Демони асоціюються з елементами структури даних об'єкта, що називаються активними значеннями. При певних операціях доступу до активних значень і їхньої модифікації ініціюються демони, що викликають ті чи інші дії, логічно пов'язані з виконуваною дією.

При об’єктно-орієнтованому підході можна реалізувати не тільки процедурні методи подання знань і методи, засновані на семантичних мережах і фреймах, але і логічні. Більш того, оскільки логічні твердження (формули) розглядаються при такому підході як об'єкти, з'являється можливість класифікації і структурування наборів формул, що, принаймні частково, компенсує недоліки логічних методів подання знань.

Таким чином, об’єктно-орієнтований підхід варто розглядати як одну з перспективних інструментальних основ для побудови інтелектуальних систем.

2.3.3 Методи пошуку рішень в просторі станів

Традиційними методами пошуку рішень в ІС вважаються: методи пошуку в просторі станів на основі різних евристичних алгоритмів, методи пошуку на основі предикатів (метод резолюції і ін.), пошук рішень в продуктивних системах, пошук рішень в семантичних мережах і т.д. Розглянемо ці методи більш детально.

Методи пошуку рішень в просторі станів почнемо розглядати з простої задачі про місіонерів і людоїдів. Три місіонери і три людоїди знаходяться на лівому березі річки і їм потрібно переправитися на правий берег, проте у них є тільки один човен, в який можуть сісти лише 2 людини. Тому необхідно визначити план, дотримуючись якого і курсуючи кілька разів туди і назад, можна переправити всіх шістьох. Проте якщо на будь-якому березі річки число місіонерів буде менше, ніж число людоїдів, то місіонери будуть з'їдені. Рішення ухвалюють місіонери, людоїди їх виконують.

Основою методу є такі етапи.

1. Визначається кінцеве число станів, один з станів береться за початковий і один або декілька станів визначаються як шукане (кінцеве або термінальне). Позначимо стан *S* трійкою *S=(x,y,z)*, де *x* і *y* - число місіонерів і людоїдів на лівому березі, *z= {L,R}* - положення човна на лівому *(L)* або правому *(R)* берегах. Отже, початковий стан S_0*=(3,3, L)* і кінцевий (термінальний) стан S_k*=(0,0, R)*.

2. Задані правила переходу між групами станів. Введемо поняття дії *M:[u, v]w*, де *u* - число місіонерів в човні, *v* - число людоїдів в човні, *w* - напрям руху човна (*R* або *L*).

3. Для кожного стану задані певні умови допустимості (оцінки) станів: *x≥y*; *3-x≥3-y*; *u+v≤2*.

4. Після цього з поточного (початкового) стану будуються переходи в нові стани, показані на рис. 2.5. Два нові стани слід зразу ж викреслити, оскільки вони ведуть до порушення умов допустимості (місіонери будуть з'їдені).

5. При кожному переході в новий стан проводиться оцінка на допустимість станів і якщо при використанні правила переходу для поточного стану виходить неприпустимий стан, то проводиться повернення до того попереднього стану, з якого було досягнуто цей поточний стан. Ця процедура одержала назву бектрекінг (bac tracing або BACKTRACK).

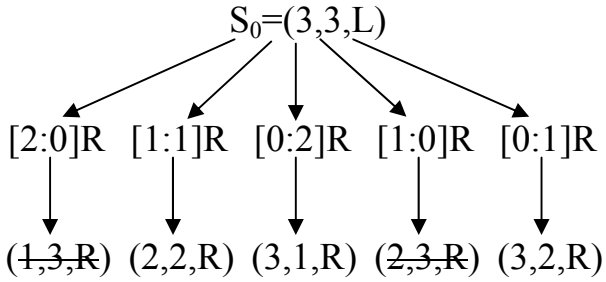


Рисунок 2.5 - Переходи з початкового стану

Тепер ми можемо проаналізувати повністю алгоритм найпростішого пошуку рішень в проблемному просторі, описаний групами станів і переходами між станами на рис. 2.6. Розв'язання задачі виділено жирними стрілками. Такий метод пошуку $S_0 => S_k$ називається прямим методом пошуку. Пошук $S_k => S_0$ називають зворотним пошуком. Пошук в двох напрямах одночасно називають двонаправленим пошуком.

Як вже згадувалося, фундаментальним поняттям в методах пошуку в ІС є ідея рекурсії і процедура *BACKTRACK*. Як приклад багаторівневого повернення розглянемо задачу розміщення на дошці 8×8 восьми ферзів так, щоб вони не змогли "з'їсти" один одного.

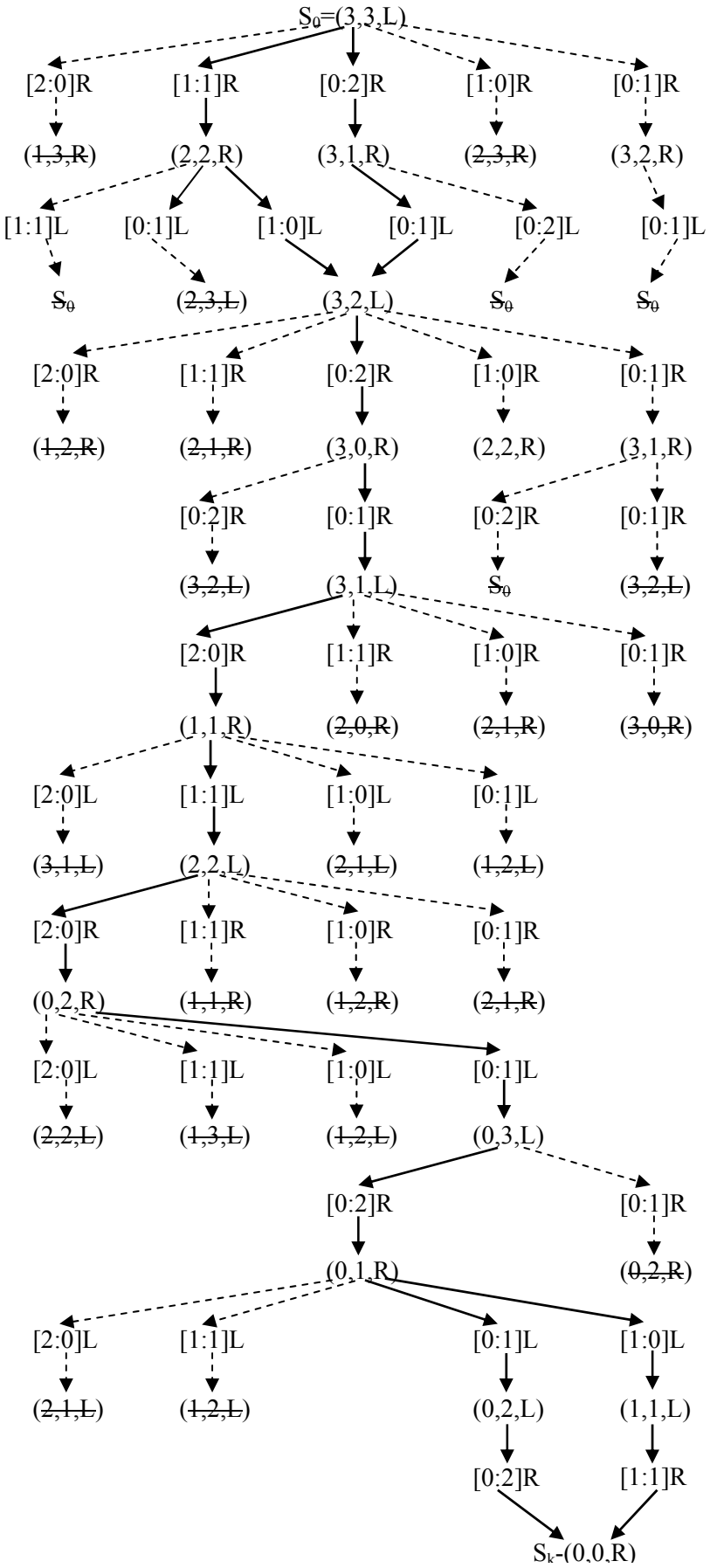


Рисунок 2.6 - Метод пошуку в просторі станів

Припустимо, ми знаходимося на кроці розміщення ферзя в 6 ряду і бачимо, що це неможливо. Процедура *BACKTRACK* намагається перемістити ферзя, в 5 рядку і в 6 рядку знову невдача. Тільки повернення до 4 рядка і знаходження в ньому нового варіанта розміщення приведе до розв'язання задачі. Читач сам може завершити розв'язання цієї задачі на основі процедури *BACKTRACK*.

x							
		x					
				x			
	x						
			x				

2.3.3.1 Алгоритми евристичного пошуку

У розглянутих прикладах пошуку рішень число станів невелике, тому перебір всіх можливих станів не викликав затруднень. Проте при значному числі станів час пошуку зростає експоненціально, і в цьому випадку можуть допомогти алгоритми евристичного пошуку, які мають високу вірогідність правильного вибору розв'язання. Розглянемо деякі з цих алгоритмів.

Алгоритм найшвидшого спуску по дереву рішень

Приклад побудови більш вузького дерева розглянемо на прикладі задачі про комівояжера. Торговець повинен побувати в кожному з 5 міст, позначених на карті (рис. 2.7).

Задача полягає в тому, щоб, починаючи з міста *A*, знайти мінімальний шлях, який проходить через всю решту міст тільки один раз і що приводить назад в *A*. Ідея методу виключно проста - з кожного міста йдемо в найближче, де ми ще не були. Розв'язання задачі показано на рис. 2.8.

Такий алгоритм пошуку рішення одержав назву алгоритму *найшвидшого спуску* (в деяких випадках - *найшвидшого підйому*).

Алгоритм оцінних (штрафних) функцій

Вміло підібрані *оцінні функції* (в деяких джерелах - *штрафні функції*) можуть значно скоротити повний перебір і призвести до розв'язання досить швидко в складних задачах. В нашій задачі про людоїдів і місіонерів як найпростішу цільову функцію при виборі чергового стану можна узяти число людоїдів і місіонерів, що знаходяться

не "на місці" порівняно з їх розташуванням в описі цільового стану. Наприклад, значення цієї функції $f=x+y$ для початкового стану $f_0=6$, а значення для цільового стану $f_1=0$.

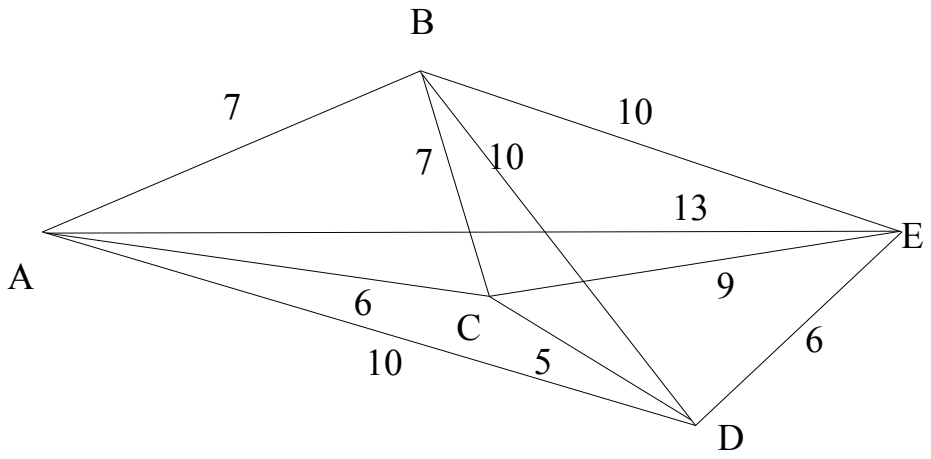


Рисунок 2.7 – До задачі про комівояжера

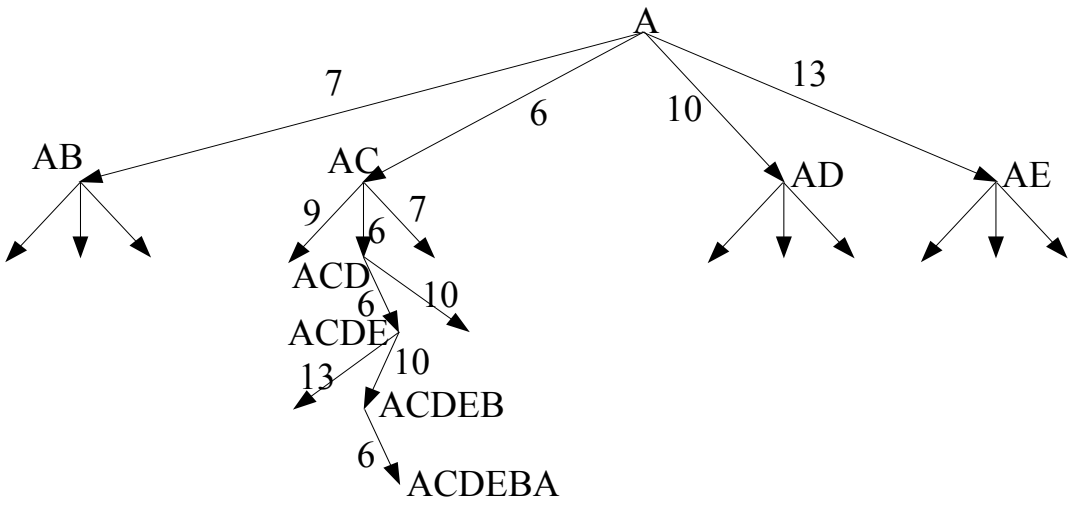


Рисунок 2.8 – Розв’язання задачі про комівояжера

Евристичні процедури пошуку на графі прямують до того, щоб мінімізувати деяку комбінацію вартості шляху до мети і вартості пошуку. Для задачі про людоїдів введемо оцінну функцію:

$$f(n)= d(n)+ w(n)$$

де $d(n)$ - глибина вершини n на дереві пошуку і $w(n)$ - кількість місіонерів і людоїдів, що знаходяться не на потрібному місці. Евристика полягає у виборі мінімального значення $f(n)$. Головним в евристичних процедурах є вибір оцінної функції.

Розглянемо питання про порівняльні характеристики оцінних цільових функцій на прикладі функцій для гри в "8" ("квачі"). Гра в "8" полягає в знаходженні мінімального числа перестановок при переході з початкового стану в кінцевий (термінальний, цільовий).

2	8	3
1	6	4
7	*	5

1	2	3
8	0	4
7	6	5

Розглянемо дві оцінні функції:

$$h_1(n)\& = Q(n);$$

$$h_2(n)\& = P(n) + 3S(n),$$

де $Q(n)$ - число фішок не на місці; $P(n)$ - сума відстаней кожної фішки від місця в її цільовій вершині; $S(n)$ - облік послідовності нецентральних фішок (штраф $+2$, якщо за фішкою стоїть не та, яка повинна бути в правильній послідовності; штраф $+1$ за фішку в центрі; штраф 0 в решті випадків).

Порівняння цих оцінних функцій наведено в таблиці 2.2.

Таблиця 2.2 - Порівняння оцінних функцій

Оцінна функція h	Вартість (довжина) шляху L	Число вершин, відкритих при знаходженні шляху N	Трудомісткість обчислень, необхідних для підрахунку h S	Примітки
h_1 S_0 S_1	5 >18	13 100-8!(=40320)	8	Пошук в ширину
h_2 S_0 S_1	5 18	11 43	8*2+8+1+1	Пошук в глибину

На основі порівняння цих двох оцінних функцій можна зробити висновки.

- Основу алгоритму пошуку становить вибір шляху з мінімальною оцінною функцією.
- Пошук по ширині, що дає функція h_1, гарантує, що який-небудь шлях до мети буде знайдений. При пошуку ушир вершини розкриваються в тому самому порядку, у якому вони породжуються.
- Пошук у глибину керується евристичним компонентом $3S(n)$ у функції h_2 і при вдалому виборі оцінної функції дозволяє знайти рішення по найкоротшому шляху (за мінімальним числом вершин, що розкриваються). Пошук у глибину тим і характеризується, що в ньому першою розкривається та вершина, що була побудована останньою.
- Ефективність пошуку зростає, якщо при невеликих глибинах він направляється в основному вглиб евристичним компонентом, а при зростанні глибини він більше схожий на пошук по ширині, щоб гарантувати, що який-небудь шлях до мети буде знайдений. Ефективність пошуку можна визначити як $E=K/L*N*S$, де K і S (трудомісткість) - залежать від оцінної функції, L - довжина шляху, N - число вершин, відкритих при знаходженні шляху. Якщо домовитися, що для оптимального шляху $E=1$, то $K=L^0*N^0*S^0$.

Алгоритм мінімакса

У 1945 році Оскар Моргенштерн і Джон фон Нейман запропонували метод *мінімакса*, що знайшов широке використання в теорії ігор. Припустимо, що супротивник використовує оцінну функцію (ОФ), яка збігається з нашою ОФ. Вибір ходу з нашої сторони визначається максимальним значенням ОФ для поточної позиції. Супротивник прагне зробити хід, який мінімізує ОФ. Тому цей метод і одержав назву мінімакса. На рис. 2.9 наведено приклад аналізу дерева ходів за допомогою методу мінімакса (обраний шлях рішення помічений жирною лінією).

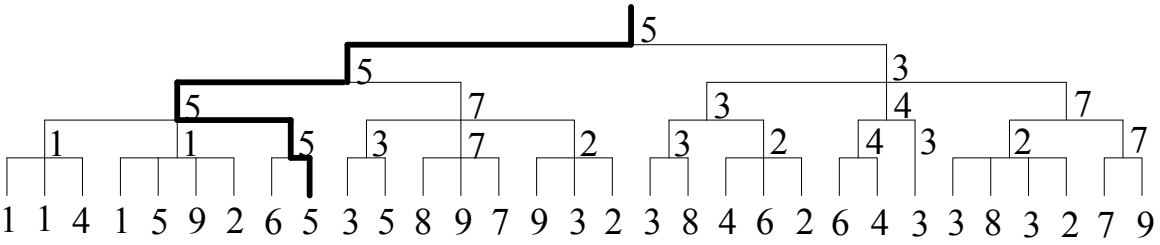


Рисунок 2.9 - Дерево ходів

Розвиваючи метод мінімакса, призначимо вірогідність для виконуваних дій в задачі про місіонерів і людоїдів:

P([2 : 0]R) = 0; 8; P([1 : 1]R) = 0; 5;

P([0 : 2]R) = 0; 9;

P([1 : 0]R) = 0; 3; P([0 : 1]R) = 0; 3.

Інтуїтивно зрозуміло, що посилати одного людоїда або одного місіонера менш ефективно, ніж двох осіб, особливо на початкових етапах. На кожному рівні ми вибиратимемо стан за критерієм P_i. Навіть такий простий підхід дозволить уникнути частини тупикових станів в процесі пошуку і скоротити час порівняно з повним перебором. До речі, цей підхід достатньо поширений в експертних продукційних системах.

Альфа-бета процедура

Теоретично, – це еквівалентна мінімаксу процедура, за допомогою якої завжди отримуємо такий самий результат, але помітно швидше, оскільки цілі частини дерева виключаються без проведення аналізу. В основі цієї процедури лежить ідея Дж. Маккарті про використання двох змінних, позначених α і β (1961 рік).

Основна ідея методу полягає в порівнянні якнайкращих оцінок, одержаних для повністю вивчених гілок, з якнайкращими передбачуваними оцінками для тих, що залишилися. Можна показати, що за певних умов деякі обчислення є зайвими. Розглянемо ідею відсікання на прикладі рис. 2.10. Припустимо, позиція *A* повністю проаналізована і знайдено значення її оцінки α. Припустимо, що один хід з позиції *Y* приводить до позиції *Z*, оцінка якої за методом мінімакса дорівнює *z*.

Припустимо, що $z \leq \alpha$. Після аналізу вузла Z, коли справедливе співвідношення $y \leq z \leq \alpha \leq s$, гілки дерева, що виходять з вузла Y, можуть бути відкинуті (альфа-відсікання).

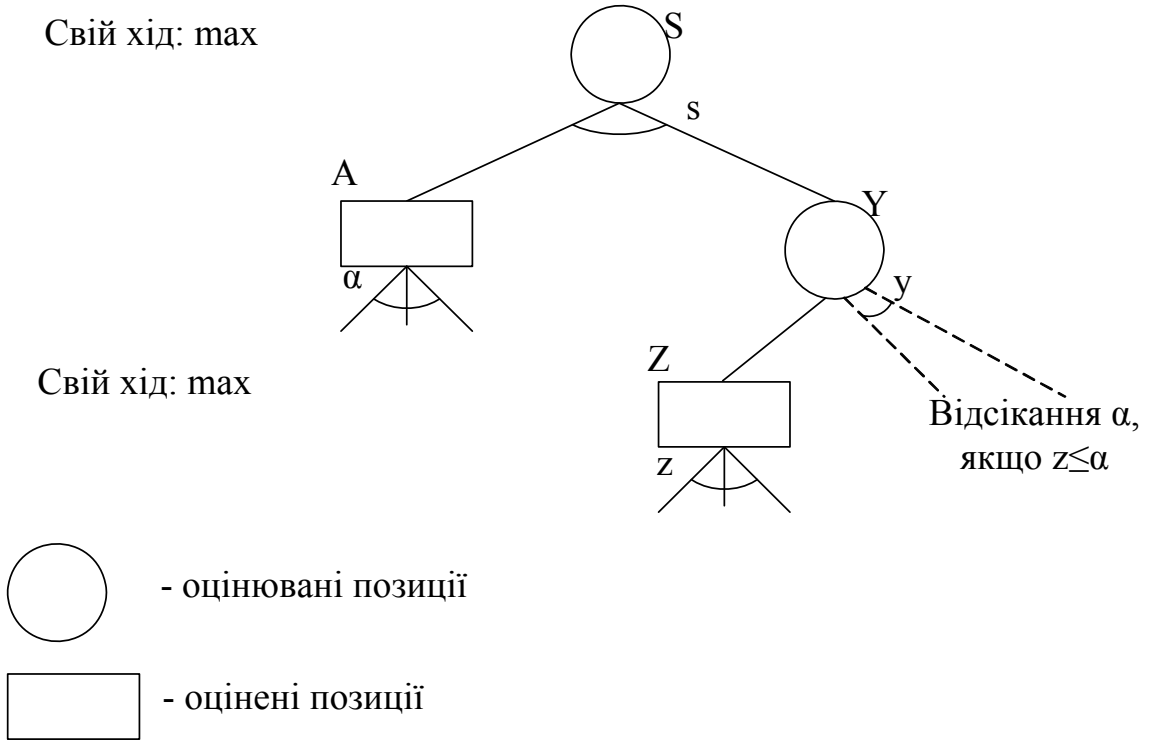


Рисунок 2.10 - Відсікання

Якщо ми схочемо опуститися до вузла Z, що лежить на рівні довільної глибини, який належить тій самій стороні, що і рівень S, то необхідно враховувати мінімальне значення оцінки β, яка отримується на ходах супротивника.

Відсікання типу β можна виконати всякий раз, коли оцінка позиції, яка виникає після ходу супротивника, перевищує значення β. Алгоритм пошуку будується так, що оцінки своїх ходів і ходів супротивника порівнюються при аналізі дерева з величинами α і β, відповідно. На початку обчислень цим величинам присвоюються значення $+\infty$ і $-\infty$, а потім, у міру просування до коріння дерева, знаходиться оцінка початкової позиції і найкращий хід для одного з супротивників.

Правила обчислення α і β в процесі пошуку рекомендуються такі:

1. Біля *MAX*- вершини значення α дорівнює найбільшому в даний момент значенню серед остаточних повернених значень для її дочірніх вершин;
2. Біля *MIN*- вершини значення β дорівнює найменшому в даний момент значенню серед остаточних повернених значень для її дочірніх вершин.

Правила припинення пошуку:

1. Можна не проводити пошуку на піддереві, що росте зі всякої *MIN*-вершини, біля якої значення β не перевищує значення α всіх її батьківських *MAX* вершин;
2. Можна не проводити пошуку на піддереві, що росте зі всякої *MAX*-вершини, біля якої значення α не менше значення β у всіх її батьківських *MIN*- вершин.

На рис. 2.11 показані α-β відсікання для конкретного прикладу. Таким чином, α-β-алгоритм дає той самий результат, що і метод мінімакса, але виконується швидше.

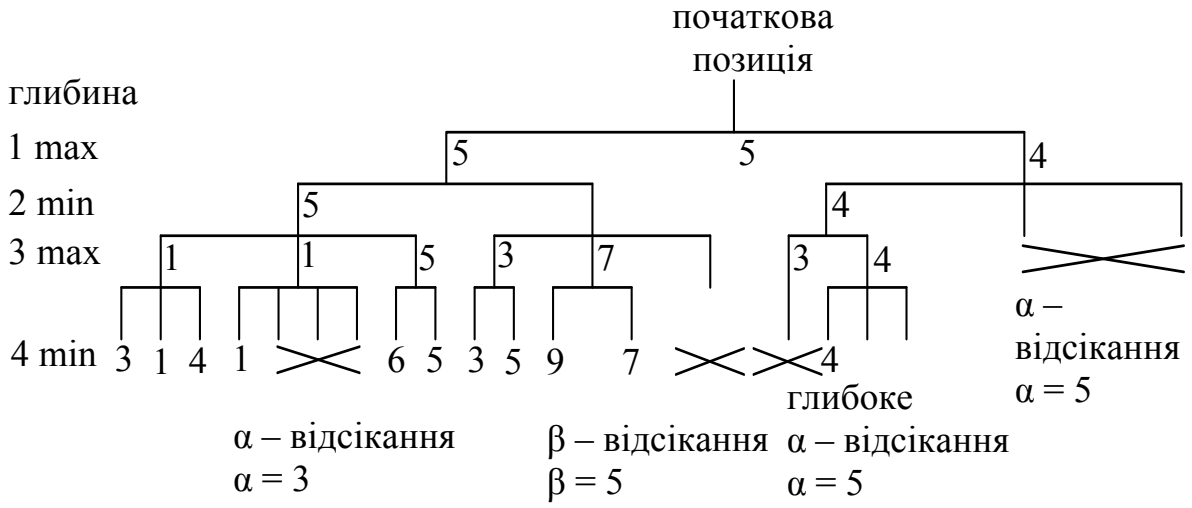


Рисунок 2.11 - α-β відсікання для конкретного прикладу

Використовування алгоритмів евристичного пошуку для пошуку на графі І, АБО виграшної стратегії в складніших задачах і іграх (шашки, шахи) нереально. За деякими оцінками ігрове дерево гри в шашки містить 10^{40} вершин, в шахах – 10^{120} вершин. Якщо при грі в шашки для однієї вершини потрібно 1/3 наносекунди, то всього ігрового часу буде потрібно 10^{21} століття. В таких випадках вводяться штучні умови зупинення, засновані на таких чинниках, як найбільша допустима глибина вершин в дереві пошуку або обмеження на певний час і об'єм пам'яті.

Багато з розглянутих вище ідей були використані А. Ньюеллом, Дж. Шоу і Г. Саймоном в їх програмі GPS. Процес роботи GPS загалом відтворює методи розв’язання задач, використовувані людиною: висуваються підцілі, що наближаються до рішення; застосовується евристичний метод (один, інший і т. д.), поки не буде одержано рішення. Спроби припиняються, якщо одержати рішення не вдається.

Програма STRIPS (STanford Research Institut Problem Solver) виробляє відповідний порядок дій робота залежно від поставленої мети. Програма здатна навчатися на досвіді розв’язання попередніх задач. Велика частина ігрових програм також навчається в процесі роботи. Наприклад, знаменита шашкова програма Самюеля, що виграла в 1974 році у чемпіона світу, “заучувала напам’ять” виграні партії і

узагальнювала їх для витягання користі з минулого досвіду. Програма HACHER Зуссмана, що управляє поведінкою робота, навчалася також і на помилках.

2.3.3.2 Методи пошуку рішень на основі вирахування предикатів

Семантика вирахування предикатів забезпечує основу для формалізації логічного висновку. Можливість логічно виводити нові правильні вирази з набору істинних тверджень дуже важлива. Логічно виведені твердження коректні, вони сумісні зі всіма попередніми інтерпретаціями первинного набору виразів. Обговоримо вищесказане неформально і потім введемо необхідну формалізацію.

У вирахуванні висловів основним об'єктом є змінний вислів (предикат), істинність або помилковість якого залежить від значень вхідних в нього змінних. Так, істинність предиката "*x* був фізиком" залежить від значення змінної *x*. Якщо *x* - П. Капіца, то предикат істинний, якщо *x* - М. Лермонтов, то він помилковий. Мовою вирахування предикатів, твердження $\forall x(P(x) \supset Q(x))$ читається так: "для будь-кого *x* якщо *P(x)*, то має місце і *Q(x)"*. Іноді його записують і так: $\forall x(P(x) \rightarrow Q(x))$. Виділена підмножина тотожна підмножині істинних формул (або правильно побудованих формул - ППФ), істинність яких не залежить від істинності вхідних в них висловів, називається аксіомами.

У вирахуванні предикатів є безліч правил висновку. Як приклад наведемо класичне правило відділення або modus ponens :

$$(A, A \rightarrow B) / B,$$

яке читається так "якщо істинна формула *A* і істинно, що із *A* виходить *B*, то істинна і формула *B*". Формули, що знаходяться над межею, називаються посилками висновку, а під межею - висновком. Це правило висновку формалізує основний закон дедуктивних систем: з істинних посилок завжди виходять істинні висновки. Аксіоми і правила виведення вирахування предикатів першого порядку задають основу формальної дедуктивної системи, в якій відбувається формалізація схеми міркувань в логічному програмуванні. Можна згадати і інші правила висновку.

Modus tollendo tollens : Якщо з *A* йде *B* і *B* помилкове, то і *A* помилкове.

Modus ponendo tollens : Якщо *A* і *B* не можуть одночасно бути істинними і *A* істинне, то *B* помилкове.

Modus tollendo ponens : Якщо або *A*, або *B* є істинним і *A* неістинне, то *B* істинне.

Задача, яку розв'язуємо, подається у вигляді тверджень (аксіом) f_1, $F_2 ... F_n$ вирахування предикатів першого порядку. Мета задачі *B* також записується у вигляді твердження, справедливість якого потрібно встановити або спростувати на підставі аксіом і правил виведення формальної системи. Тоді розв'язання задачі (досягнення мети задачі)

зводиться до з'ясування логічного проходження (виведення) цільової формули B із заданої безлічі формул (аксіом) $f_1, F_2... F_n$. Таке з'ясування рівносильне доказу загальнозначущості (тотожно-істинності) формули

$$f_1 \& F_2 \& ... \& F_n \rightarrow B$$

або нездійсненності (тотожно-неправдивості) формули

$$f_1 \& F_2 \& ... \& F_n \& \neg B.$$

З практичних міркувань зручніше використовувати доказ від супротивного, тобто доводити нездійсненність формули. На доказі від супротивного засновано і головне правило висновку, що використовується в логічному програмуванні, - *принцип резолюції*. Робінсон відкрив більш сильніше правило висновку, ніж modus ponens, яке він назвав принципом резолюції (або правилом резолюції). При використанні принципу резолюції формули вирахування предикатів за допомогою нескладних перетворень приводяться до так званої диз'юнктивної форми, тобто подаються у вигляді набору диз'юнктів. При цьому під диз'юнктом розуміється диз'юнкція літералів, кожний з яких є або предикатом, або запереченням предиката.

Наведемо приклад диз'юнкта:

$$\forall x \, (P(x, c_1) \supset Q(x, \rightarrow c_2)).$$

Нехай P - предикат поважати, c_1 - Ключевський, Q - предикат знати, c_2 - історія. Тепер даний диз'юнкт відображає факт "кожний, хто знає історію, поважає Ключевського".

Наведемо ще один приклад диз'юнкта:

$$\forall x \, (P(x, c_1) \& P(x, c_2)).$$

Нехай P - предикат знати, c_1 - фізика, $c2$ - історія. Даний диз'юнкт відображає запит "хто знає фізику і історію одночасно".

Таким чином, умови розв'язуваних задач (факти) і цільові твердження задач (запити) можна виразити в диз'юнктивній формі логіки предикатів першого порядку. В диз'юнктах квантори загальності $\forall, \exists$, звичайно опускаються, а зв'язки $\neg$, замінюються на імплікацію.

Повернемося до принципу резолюції. Головна ідея цього правила висновку полягає в перевірці того, чи містить безліч диз'юнктів R порожній (помилковий) диз'юнкт. Звичайно резолюція застосовується з прямим або зворотним методом міркування. Прямий метод з посилок A, $A \rightarrow B$ виводить висновок B (правило modus ponens). Основний недолік прямого методу полягає в його неспрямованості: повторне використання методу приводить до різкого зростання проміжних висновків, не пов'язаних з цільовим висновком. Зворотний висновок є направленим: з бажаного висновку B і тих самих посилок він виводить новий підцільовий висновок A. Кожний крок висновку в цьому випадку пов'язаний завжди із раніше поставленою метою. Істотний недолік методу резолюції полягає у формуванні на кожному кроці виведення безлічі резольвент - нових диз'юнктів, більшість з яких виявляється зайвими. У зв'язку з цим

розроблені різні модифікації принципу резолюції, що використовують більш ефективні стратегії пошуку і різного роду обмеження на вигляд початкових диз'юнктів. В цьому значенні найвдалішою і популярною є система ПРОЛОГ, яка використовує спеціальні види диз'юнктів, званих диз'юнктами Хорна.

Процес доказу методом резолюції (від зворотного) складається з таких етапів:

1. Пропозиції або аксіоми приводяться до диз'юнктивної нормальної форми;

2. До набору аксіом додається заперечення доводжуваного твердження в диз'юнктивній формі;

3. Виконується сумісний дозвіл цих диз'юнктів, внаслідок чого виходять нові, засновані на них, диз'юнктивні вирази (резольвенти);

4. Генерується порожній вираз, що означає протиріччя.

5. Підстановки, використані для отримання порожнього виразу, свідчать про те, що заперечення істинне.

Розглянемо приклади використання методів пошуку рішень на основі вирахування предикатів. Приклад "цікаве життя" запозичено з [2]. Отже, задані твердження 1-4 в лівому стовпці таблиці 2.3. Потрібно відповісти на запитання: "Чи існує людина, що живе цікавим життям?" У вигляді предикатів ці твердження записані в другому стовпці таблиці. Передбачається, що $\forall X(smart(X) = \neg stupid(X))$ і $\forall Y(wealthy(Y) = \neg poor(Y))$. В третьому стовпці таблиці записані диз'юнкти.

Таблиця 2.3 - Цікаве життя

Твердження і висновок	Предикати	Пропозиції(диз'юнкти)
1. Всі небідні і розумні люди щасливі	$\exists X(\neg poor(X) \wedge smart(X) \rightarrow happy(X))$	$poor(X) \wedge \neg smart(X)$& happy(X)
2. Людина, яка читає книги, - недурна	$\forall$ Y (read(Y) → smart(Y))	¬read(Y)& smart(Y)
3. Джон уміє читати і є самостійної людиною	$read(John) \wedge \neg poor(John)$	3a read(John) 3b ¬poor(John)
4. Щасливі люди живуть цікавим життям	$\forall Z$ (happy(Z)→ exciting(Z))	¬happy(Z)&exciting(Z)
5. Висновок: Чи існує людина, що живе цікавим життям?	$\exists W(exciting(W))$	exciting(W)
6. Заперечення висновку	$\neg \exists W(exciting(W))$	¬exciting(W)

Заперечення висновку має вигляд (рядок 6): $\neg \exists W(exciting(W))$

Один з можливих доказів (їх більше одного) дає таку послідовність резольвент:

1. ¬happy(Z) резольвента 6 і 4;

2. $poor(X) \wedge \neg smart(X)$ резольвента 7 і 1;

3. poor(Y) ∧ ¬read(Y) резольвента 8 і 2;

4. ¬read(John) резольвента 9 і 3b;

5. NIL резольвента 10 і 3a.

Символ *NIL* означає, що база даних виразів містить протиріччя і тому наше припущення, що не існує людини, яка живе цікавим життям, неправильне.

У методі резолюції порядок комбінації диз'юнктивних виразів не встановлювався. Значить, для великих задач спостерігатиметься експоненціальне зростання числа можливих комбінацій. Тому в процедурах резолюції велике значення мають також евристики пошуку і різні стратегії. Одна з найпростіших і зрозумілих стратегій - стратегія переваги одиничного виразу, яка гарантує, що резольвента буде меншою, ніж найбільший батьківський вираз. Адже як результат ми повинні одержати вираз, що не містить літералів взагалі.

Серед інших стратегій (пошук в ширину (breadth-first), стратегія "множини підтримки", стратегія лінійної вхідної форми) стратегія "множини підтримки" показує відмінні результати при пошуку у великих просторах диз'юнктивних виразів [2]. Суть стратегії така. Для деякого набору початкових диз'юнктивних виразів S можна вказати підмножину T, звану множиною підтримки. Для реалізації цієї стратегії необхідно, щоб одна з резольвент в кожному спростуванні мала предка з множини підтримки. Можна довести, що коли S - нездійсненний набір диз'юнктів, а S-T - здійснимий, то стратегія множини підтримки є повною в значенні спростування. З іншими стратегіями для пошуку методом резолюції у великих просторах диз'юнктивних виразів читач може познайомитися в спеціальній літературі.

Дослідження, пов'язані з доведенням теорем і розробкою алгоритмів спростування резолюції, привели до розвитку мови логічного програмування PROLOG (*Programming in Logic*). PROLOG заснований на теорії предикатів першого порядку. Логічна програма - це набір специфікацій в рамках формальної логіки. Не дивлячись на те, що в даний час питома вага мов LISP і PROLOG знизилася і при розв'язанні задач ШІ все більше використовуються C, C++ і Java, проте багато задач і розробка нових методів розв'язання задач ШІ продовжують опиратися на мови LISP і PROLOG. Розглянемо одну з таких задач - задачу планування послідовності дій і її розв'язання на основі теорії предикатів.

2.3.3.3 Задачі планування послідовності дій

Багатьох результатів в галузі ШІ досягнуто при розв'язанні "задач для робота". Однією з таких простих в постановці і інтуїтивно зрозумілих задач є задача планування послідовності дій або задача побудови планів.

У наших міркуваннях будуть використані приклади традиційної робототехніки (сучасна робототехніка багато в чому ґрунтується на

реактивному управлінні, а не на плануванні). Пункти плану визначають атомарні дії для робота. Проте при опису плану немає необхідності опускатися до мікрорівня і говорити про датчики, крокові двигуни і т.п. Розглянемо ряд предикатів, необхідних для роботи планувальника з світу блоків. Є деякий робот, що є рухомою рукою, здатною брати і переміщувати кубики (рис. 2.12). Рука робота може виконувати такі завдання (*U, V, W, X, Y, Z* - змінні):

goto(X,Y,Z) – перейти в місцеположення *X,Y,Z;*

pickup(W) – узяти блок *W* і тримати його*;*

putdown(W) – опустити блок *W* в деякій точці*;*

stack(U,V) – помістити блок *U* на верхню грань блоку *V;*

unstuck(U,V) – прибрати блок *U* з верхньої грані блоку *V*.

Стан світу описуються такою безліччю предикатів і відношень між ними.

on(X,Y) – блок *X* знаходиться на верхній грані блоку *Y;*

clear(X) – верхня грань блоку *X* порожня*;*

gripping(X) – захват робота утримує блок *X;*

gripping( )– захват робота порожній*;*

ontable(W) – блок *W* знаходиться на столі.

Предметна галузь з світу кубиків зображена на рис. 2.12 у вигляді початкового і цільового стану для розв'язання задачі планування. Вимагається побудувати послідовність дій робота, яка веде (при її реалізації) до досягнення цільового стану.

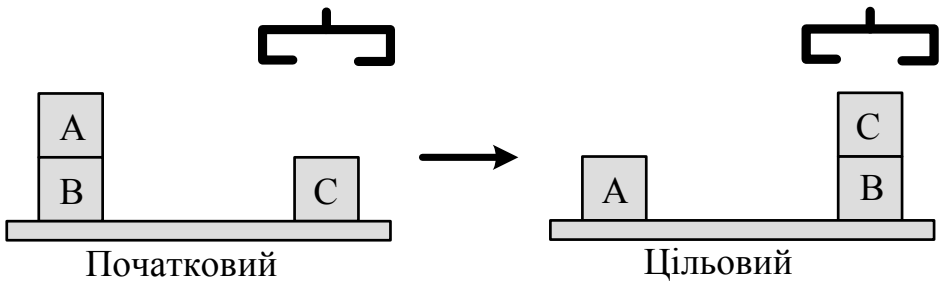


Рисунок 2.12 - Початковий і цільовий стан задачі з світу кубиків

Стану світу кубиків запишемо у вигляді предикатів. Початковий стан можна описати таким чином:

start = [handempty, ontable(b)
ontable(c), on(a,b), clear(c)
clear(a)],

де: handempty означає, що рука робота Роббі порожня.

Цільовий стан записується так.

goal = [handempty, ontable(a)
ontable(b), on(c,b), clear(a)

clear(c)]

Тепер запишемо правила, що впливають на стан і що приводять до нових станів.

(∀ X) (pickup(X) → (gripping(X) ←
(gripping() ∧ clear(X) ∧ ontable(X))))

(∀ X) (putdown(X) → ((gripping() ∧
ontable(X) ∧ clear(X)) ← gripping(X)))

(∀ X) (∀ Y) (stack(X,Y) →
((on(X,Y) ∧ gripping() ∧ clear(X)) ←
(clear(Y) ∧ gripping(X))))

(∀ X) (∀ Y) (unstack(X,Y) →
((clear(Y) ∧ gripping(X)) ←
(on(X,Y) ∧ clear(X) ∧ gripping()))

Перш ніж використовувати ці правила, необхідно згадати про проблему меж. При виконанні деякої дії можуть змінюватися інші предикати і для цього можуть використовуватися аксіоми меж - правила, що визначають інваріантні предикати. Одне з рішень цієї проблеми запропоноване в системі STRIPS.

На початку 1970-х років в Стенфордському дослідницькому інституті (Stanford Research Institute Planning System) була створена система STRIPS для управління роботом. В STRIPS чотири оператори pickup, putdown, stack, unstuck описуються трійками елементів. Перший елемент трійки - безліч передумов (П), які задовольняють світ до використання оператора. Другий елемент трійки - список доповнень (Д), які є результатом використання оператора. Третій елемент трійки - список викреслювань (В), що складається з виразів, які видаляються з опису стану після використання оператора.

Ведучи міркування для даного прикладу від початкового стану, ми приходимо до пошуку в просторі станів. Необхідна послідовність дій (план досягнення мети) буде такою.

unstuck(A, B), putdown(A), pickup(C), stack(C, B)

Для великих графів (сотні станів) пошук слід проводити з використанням оціночних функцій. Більш докладно про роботи із планування, у тому числі сучасні публікації із адаптивного планування, можна прочитати в літературі [7], [47-50].

Як висновок в даному розділі слід сказати, що планування досягнення мети можна розглядати як пошук в просторі станів. Для знаходження шляху з початкового стану до цільового (плану послідовності дій робота) можуть застосовуватися методи пошуку в просторі станів з використанням вирахування предикатів.

2.3.3.4 Пошук рішень в системах продукцій

Пошук рішень в системах продукцій натрапляє на проблеми вибору правил з конфліктної множини. Різні варіанти рішення цієї проблеми розглянемо на прикладі ECO CLIPS. Правила в ЕС, окрім чинника упевненості експерта, мають пріоритет виконання (*salience*). Конфліктна множина (КМ) - це список всіх правил, що мають умови, які задоволені при деякому поточному стані списку фактів і об'єктів і які були ще не виконані. Як наголошувалося раніше, конфліктна множина – це найпростіша база цілей. Коли активізується нове правило з певним пріоритетом, воно поміщається в список правил КМ нижче за всі правила з високим пріоритетом і вище за всі правила з меншим пріоритетом. Правила з вищим пріоритетом виконуються в першу чергу. Серед правил з однаковим пріоритетом використовується певна стратегія.

CLIPS підтримує сім стратегій розв’язання конфліктів.

Стратегія глибини (*depth strategy*) є стратегією за замовчуванням (default strategy) в CLIPS. Тільки що активізоване правило поміщається над всіма правилами з таким же пріоритетом. Це дозволяє реалізувати пошук вглиб.

Стратегія ширини (*breadth strategy*) - тільки що активізоване правило поміщається після всіх правил з таким же пріоритетом. Це, у свою чергу, реалізує пошук по ширині.

LEX стратегія - активація правила, виконана більш новими зразками (фактами), розташовується перед активацією, здійсненою більш пізніми зразками. Наприклад, як це вказано в таблиці 2.4.

MEA стратегія - сортування зразків не проводиться, а здійснюється тільки впорядкування правил по перших зразках, як це показано в стовпці 3 таблиці 2.4.

Таблиця 2.4 - Результати вживання LEX і MEA стратегій

Початковий набір правил	Правила, відсортовані LEX	Правила, відсортовані MEA
rule-6: f-1,f-4	rule-6: f-4,f-1	rule-2: f-3,f-1
rule-5: f-1,f-2,f-3	rule-5: f-3,f-2,f-1	rule-3: f-2,f-1
rule-1: f-1,f-2,f-3	rule-1: f-3,f-2,f-1	rule-6: f-1,f-4
rule-2: f-3,f-1	rule-2: f-3,f-1	rule-5: f-1,f-2,f-3
rule-4: f-1,f-2	rule-4: f-2,f-1	rule-1: f-1,f-2,f-3
rule-3: f-2,f-1	rule-3: f-2,f-1	rule-4: f-1,f-2

Стратегія спрощення (*simplicity strategy*) - серед всіх правил з однаковим пріоритетом тільки що активізоване правило розташовується вище за всі правила з рівною або більшою визначеністю (specificity). Визначеність правила задається кількістю зіставлень в лівій частині правил плюс кількість викликів функцій. Логічні функції не збільшують визначеність правила.

Стратегія ускладнення (*complexity strategy*) - серед всіх правил з однаковим пріоритетом тільки що активізоване правило розташовується вище за всі правила з рівною або більшою визначеністю.

Випадкова стратегія (*random strategy*) - кожній активації призначається випадкове число, яке використовується для визначення знаходження серед активацій з певним пріоритетом.

Підхід на основі стратегій пошуку рішень в продукційних ЕС відомий досить давно. Вельми популярна на початку 90-х років ЕСО GURU (ІНТЕР-ЕКСПЕРТ) також використовувала подібні механізми управління стратегіями пошуку. Можливість зміни стратегії в ході розв’язання задачі програмним чином і накопичення досвіду, які стратегії дають кращі результати для певних класів задач, дозволяє одержати ефективні механізми пошуку рішень в СПЗ на основі продукцій.

Завершуючи даний розділ, слід зазначити, що існують різні методи пошуку рішень в семантичних мережах, наприклад, метод обходу семантичної мережі - мультипарсинг. Даний метод оригінальний тим, що дозволяє паралельно "вести" по графу декілька маркерів і, тим самим, розпаралелювати процес пошуку інформації в семантичній мережі, що збільшує швидкість пошуку. Ці методи використовуються, як правило, при поданні тексту у вигляді об'єктно-орієнтованої семантичної мережі і в даному посібнику не розглядаються.

Питання для самоконтролю

1. Що розуміється під терміном "представлення знань"?
2. Що називається системою знань?
3. На яки типи розділяють інформацію при зберіганні знань в ЕОМ?
4. Які є особливості знань?
5. Яка додаткова інформація вводиться у пам'ять ЕОМ при переході від даних до знань?
6. Які існують методи подання знань?
7. Які методи подання знань, відносяться до неформальних моделей?
8. Яка система лежить в основі логічної моделі подання знань?
9. Як здійснюється подання знань із застосуванням фреймів?
10. Що моделює система ШІ?
11. Яка основна ідея фреймового підходу до подання знань?
12. Бази знань і принципи їх формування.
13. Формалізація та інтерпретація знань.
14. Поясніть метод подання знань у вигляді «дерева рішень».
15. Що таке інтеграція методів подання знань?
16. Що означають макрооператори?

3 РОЗПІЗНАВАННЯ В СИСТЕМАХ ШТУЧНОГО ІНТЕЛЕКТУ

Інтерес до проблеми розпізнавання образів (РО - *recognition of images*) виник у зв'язку з дослідженнями процесів розпізнавання і класифікації, реалізованими в живих організмах, у тому числі у сфері інтелектуальної діяльності людини. Ці дослідження стосувалися розпізнавання і (або) класифікації об'єктів, явищ (ситуацій), процесів і спроб моделювання процесу розпізнавання на ЕОМ. Було з'ясовано, що практичне значення проблеми розпізнавання образів надзвичайно велике і далеко виходить за рамки чисто дослідницьких задач. Справа в тому, що багато задач сучасної техніки і природознавства зводяться до розпізнавання об'єктів, явищ або процесів.

Розпізнавання – це віднесення конкретного об'єкта (реалізації), поданого значеннями його властивостей (ознак), до одного з фіксованого переліку образів (класів) за певним вирішальним правилом відповідно до поставленої мети [33, 34].

3.1 Основні принципи розв'язання задачі розпізнавання

Серед безлічі цікавих галузей використання розпізнавання зорових образів слід відзначити такі:

- розпізнавання відбитків пальців;
- розпізнавання за райдужною оболонкою ока;
- розпізнавання машинобудівних креслень;
- визначення реальних координат об'єктів;
- розпізнавання машинописних і рукописних текстів.

Практичне значення задачі машинного читання друкованих і рукописних текстів визначається необхідністю подання, збереження і використання в електронному вигляді величезної кількості накопиченої текстової інформації. Крім того, велике значення має оперативне введення в інформаційні і керуючі системи інформації з машиночитних бланків, що містять як надруковані, так і рукописні тексти. У зв'язку з цим розглянемо принципи і підхід до розпізнавання друкованих і рукописних текстів.

Для розв'язання даної задачі використовуються такі основні принципи.

1. *Принцип цілісності* (*principle of integrity*) - розпізнаваний об'єкт розглядається як єдине ціле, що складається зі структурних частин, пов'язаних між собою просторовими відносинами.

2. *Принцип двонаправленості* (*principle of a two-orientation*) - створення моделі ведеться від зображення до моделі і від моделі до зображення.

3. *Принцип передбачення* (*principle of a prediction*) полягає у формуванні гіпотези про зміст зображення. Гіпотеза виникає при взаємодії процесу "униз", що розвертається на основі моделі середовища, моделі

поточної ситуації і поточного результату сприйняття, і процесу "вгору", заснованого на безпосередньому грубому *ознаковому* сприйнятті.

4. *Принцип цілеспрямованості* (*principle of purposefulness*), що включає сегментацію зображення і спільну інтерпретацію його частин.

5. *Принцип "не нашкодь"* (*principle "to not do much harm"*) - нічого не робити до розпізнавання і поза розпізнаванням, тобто без "розуміння".

6. *Принцип максимального використання моделі проблемного середовища* (*principle of maximal use of model of problem environment*).

Зазначені принципи реалізовані в пакеті програм "Графить", у програмах FineReader-рукопис і FormReader - для розпізнавання рукописних символів і, частково, у програмі FineReader для розпізнавання друкованих текстів. Складова програми FormReader, що призначена для читання рукописних текстів була випущена в 1998 році одночасно із системою ABBYY FineReader 4.0. Ця програма може читати всі рукописні малі і великі символи, допускає обмежені злиття символів між собою і з графічними лініями і забезпечує підтримку 10 мов. Основне застосування програми - розпізнавання і введення інформації з машиночитних бланків.

У системі ABBYY FormReader при розпізнаванні рукописних текстів використовуються структурний, растровий, ознаковий, диференціальний і лінгвістичний рівні розпізнавання. Для більш докладного освоєння підходів до розпізнавання машинописних і рукописних текстів у системі ABBYY FormReader читачеві рекомендується безпосередньо ознайомитися з роботою А. Шамиса, при цьому вважається, що є певного рівня знання основ машинної графіки.

Відзначимо особливості задачі зорового сприйняття роботів порівняно з традиційними задачами розпізнавання образів і машинної обробки зображень:

- необхідність побудови комплексного опису середовища на основі обліку значної апріорної інформації (моделі проблемного середовища) на відміну від традиційної задачі виділення фіксованих ознак або виміру окремих параметрів;
- необхідність аналізу тривимірних сцен не тільки в плані аналізу тривимірних об’єктів за їхніми плоскими проекціями, але й у плані визначення об’ємних просторових відношень;
- необхідність аналізу зображень, що включають одночасно декілька розташованих об’єктів (у загальному випадку довільної форми) на відміну від традиційної задачі, коли для розпізнавання пред’являється, як правило, один об’єкт;
- необхідність аналізувати реальне динамічне середовище, а не статичні зображення;
- відсутність постійної фіксованої задачі і необхідність оперативно розв’язувати виникаючі в процесі справи задачі;
- необхідність стежити за змінами в середовищі, які можуть

породжувати нові оперативні задачі;

– необхідність організації системного процесу взаємодії в реальному часі декількох підсистем робота ("око-мозок", "око-мозок-рука").

Із розвитком комп'ютерних систем стає усе більш очевидним, що використання цих систем набагато розшириться, якщо стане можливим використання людської мови при роботі безпосередньо з комп'ютером, і, зокрема, стане можливим керування машиною звичайним голосом у реальному часі, а також введення і виведення інформації у виді звичайної людської мови.

3.2 Параметри, ознаки, характеристики об'єктів

Образ - об'єкт, явище або процес, над яким буде здійснена операція розпізнавання або класифікації.

Ознака - проста або складна характеристики об'єкта, яку можна виміряти.

Клас - група об'єктів, відібраних за однією або декількома ознаками.

Словник ознак - список всіх можливих ознак, що описують групу образів.

Інваріантна ознака - ознака, незмінна при перетвореннях на рецепторній матриці.

Наприклад, ознака якого-небудь образу - його площа. Ця ознака інваріантна до повороту зображення; до паралельного перенесення зображення; до надпорогової зміни кольору або освітленості [35, 36].

Розглянемо деякі властивості ознак зорових образів.

Ознаки: площа - велика;
.....
мала;
компактність - велика;
.....
мала;
опуклість - є;
немає;
зв'язність -1
- 2
- 3
.....
До;
і т.д.

Таких ознак може бути досить багато і їх число залежить від різновидів і числа сенсорів, від алгоритмів кодування, від кількісної оцінки величини конкретної ознаки в даному образі.

Ознаки можуть бути взаємозв'язаними або частково

взаємозв’язаними (наприклад площа - компактність), а можуть бути незалежними (наприклад, площа - зв’язність).

Зміна величини ознаки може бути різною:

- градуальною - 1 см^2, 1,5 см^2, 28 см^2;
- дискретною - зв’язність =1, 2, 3,… ;
- бінарною - світло – тьма, 0 або 1.

Тобто вектор опису якого-небудь образу може мати координати, виражені різними типами даних.

Простір образів - простір, осі якого є осями ознак, а точки простору утворюють безліч образів відповідних класів.

Рецепторами або рецепторними клітинами організму називаються клітини, чутливі до специфічної дії зовнішнього середовища.

Модальністю дії називатимемо конкретний специфічний вид дії.

Приклади дії різних модальностей: зорова, дотикова (тактильна), звукова (акустична), смакова і т.д.

У технічних системах також є елементи, які виконують рецепторні функції. Ці елементи називаються сенсорами або датчиками.

Сенсори, як і рецептори, можуть бути різними за складністю, швидкістю, точністю і іншими параметрами функціонування.

3.3 Статистичні методи розпізнавання

Ті методи, що мають і детерміністське, і статистичне трактування, будуть розглянуті двічі у відповідних розділах курсу. Це стосується, зокрема, методу потенційних функцій, методів найближчих сусідів і інших [37-39].

3.3.1 Побудова вирішальних правил

Для побудови вирішальних правил потрібна навчальна вибірка. Навчальна вибірка – це безліч об’єктів, заданих значеннями ознак і належність яких до того або іншого класу вірогідно відома "учителеві" і повідомляється вчителем "навчальній" системі. За навчальною вибіркою система будує вирішальні правила. Якість вирішальних правил оцінюється за контрольною (екзаменаційною) вибіркою, у яку входять об’єкти, задані значеннями ознак, і належність яких тому або іншому образу відома тільки вчителеві. Надаючи навчальній системі для контрольного розпізнавання об’єкти екзаменаційної вибірки, вчитель у змозі дати оцінку імовірностей помилок розпізнавання, тобто оцінити якість навчання. До навчальних та контрольної вибірок висуваються певні вимоги. Наприклад, важливо, щоб об’єкти екзаменаційної вибірки не входили в навчальну вибірку (іноді, щоправда, ця вимога порушується, якщо загальний обсяг вибірок малий і збільшити його або неможливо, або надзвичайно складно).

Навчальні й екзаменаційна вибірки повинні досить повно представляти генеральну сукупність (гіпотетична безліч усіх можливих

об’єктів кожного образу). Наприклад, при навчанні системи медичній діагностиці в навчальних і контрольній вибірках повинні бути представлені пацієнти різних статевозрілих груп, з різними анатомічними і фізіологічними особливостями, що супроводжуються захворюваннями і т.д. При соціологічних дослідженнях це називають репрезентативністю вибірки.

Отже, для побудови вирішальних правил системі надаються об’єкти, які входять у навчальну вибірку.

3.3.2 Методи порівняння з еталонами

Для кожного класу за навчальною вибіркою будується еталон, що має значення ознак

$$\bar{x}^0 = \left\{x_1^0, x_2^0, \ldots, x_N^0\right\},$$

де $x_i^0 = \frac{1}{K}\sum_{k=1}^{K} x_{ik}$;

K – кількість об’єктів даного образа в навчальній вибірці;

i – номер ознаки.

Власне кажучи, *еталон* – це усереднений по навчальній вибірці абстрактний об’єкт (рис. 3.1). Абстрактним ми його називаємо тому, що він може не збігатися не тільки з жодним об’єктом навчальної вибірки, але і з жодним об’єктом генеральної сукупності.

Розпізнавання здійснюється в такий спосіб. На вхід системи надходить об’єкт $\bar{x}^*$, належність якого до того або іншого образу системі невідома. Від цього об’єкта виміряються відстані до еталонів всіх образів, і $\bar{x}^*$ система відносить до того образу, відстань до еталона якого мінімальна. Відстань виміряється в тій метриці, яка введена для розв’язання конкретної задачі розпізнавання.

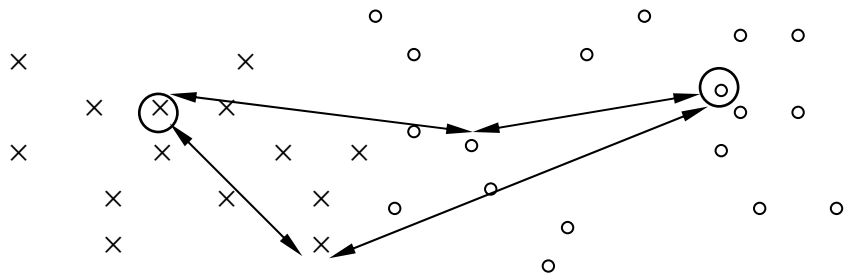


Рисунок 3.1 - Вирішальне правило "Мінімум відстані до еталона класу":

 – еталон першого класу;

 – еталон другого класу

3.3.2.1 Метод еталонів, що дробляться

Процес навчання полягає в такому. На першому етапі в навчальній вибірці "охоплюють" всі об’єкти кожного класу гіперсферою якомога

меншого радіуса. Зробити це можна, наприклад, так. Будується еталон кожного класу. Обчислюється відстань від еталона до всіх об'єктів даного класу, що входять у навчальну вибірку. Вибирається максимальна з цих відстаней $r_{\max}$. Будується гіперсфера з центром в еталоні і радіусом $R = r_{\max} + \varepsilon$. Вона охоплює всі об'єкти даного класу. Така процедура проводиться для всіх класів (образів). На рис. 3.2 наведено приклад двох образів у двовимірному ознаковому просторі.

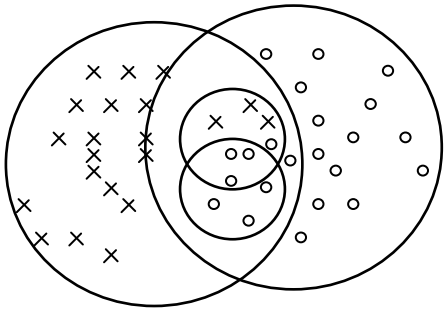


Рисунок 3.2 - Вирішальне правило типу "Метод еталонів, що дробляться"

Якщо гіперсфери різних образів перекриваються й в зоні перекриття виявляються об'єкти більш ніж одного образу, то для них будуються гіперсфери другого рівня, потім третього і т.д. доти, доки зони не виявляться неперетинними або в зоні перетинання будуть наявні об'єкти тільки одного образу.

Розпізнавання здійснюється в такий спосіб. Визначається місцезнаходження об'єкта відносно гіперсфер першого рівня. При потраплянні об'єкта в гіперсферу, що відповідає одному і тільки одному образові, процедура розпізнавання припиняється. Якщо ж об'єкт виявився в зоні перекриття гіперсфер, що при навчанні містили об'єкти більш ніж одного образу, то переходимо до гіперсфер другого рівня і проводимо дії такі ж, як для гіперсфер першого рівня. Цей процес продовжується доти, доки належність невідомого об'єкта тому чи іншому образу не визначиться однозначно. Правда, ця подія може і не настати. Зокрема, невідомий об'єкт може не потрапити в жодну з гіперсфер будь-якого рівня. У цих випадках "учитель" повинний включити у вирішальні правила відповідні дії. Наприклад, система може або відмовитися від рішення про однозначне віднесення об'єкта до якогось образу, або використовувати критерій мінімуму відстані до еталонів даного або попереднього рівня і т.п. Який з цих прийомів ефективніший, сказати важко, тому що метод еталонів, що дробляться, носить в основному емпіричний характер.

3.3.2.2 Лінійні вирішальні правила

Сама назва говорить про те, що границя, яка розділяє в ознаковому просторі зони різних образів, описується лінійною функцією (рис. 3.3)

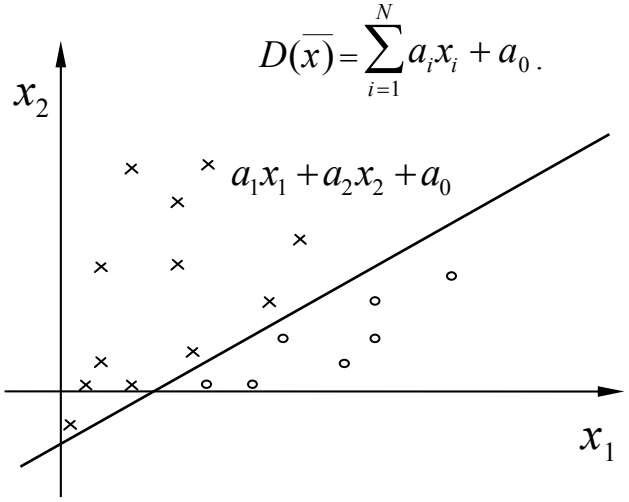


Рисунок 3.3 - Лінійне вирішальне правило для розпізнавання двох образів

Існують різні методи побудови лінійних вирішальних правил. Розглянемо один з них, реалізований у 50-х роках Розенблатом, у пристроях розпізнавання зображень, названих персептронами (рис. 3.4).

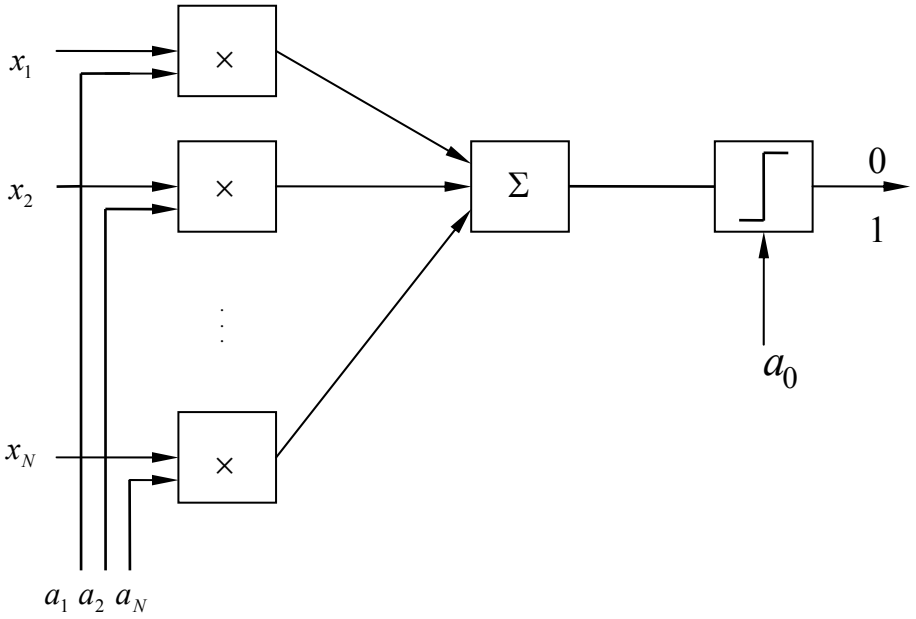


Рисунок 3.4 - Спрощена схема одношарового персептрона

Нехай

$$D(\bar{x}^*) = \bar{a}^T \bar{x}^* = \sum_{i=0}^{N} a_i x_i^* \begin{cases} > 0, \text{ якщо } \bar{x}^* \in s_1, \\ < 0, \text{ якщо } \bar{x}^* \in s_2, \end{cases}$$

де $\bar{x}^*$ – деякий об'єкт одного з образів, $x_0 = 1$.

Вибір $\bar{a}$ здійснюється покроковим чином. Поточне значення $\bar{a}$ заміняється новим $\bar{a}'$ після пред'явлення персептрону чергового об'єкта навчальної вибірки. Це корегування виконується за таким правилом:

1. $\bar{a}' = \bar{a}$, якщо $\bar{x}^* \in s_1$ й $\bar{a}^T \bar{x}^* > 0$ або якщо $\bar{x}^* \in s_2$ і $\bar{a}^T \bar{x}^* < 0$;

2. $\bar{a}' = \bar{a} + c\bar{x}^*$, якщо $\bar{x}^* \in s_1$ і $\bar{a}^T \bar{x}^* \le 0$, $c > 0$;

3. $\bar{a}' = \bar{a} - c\bar{x}^*$, якщо $\bar{x}^* \in s_2$ і $\bar{a}^T \bar{x}^* \ge 0$.

Це правило цілком логічне. Якщо черговий об'єкт системою класифікований правильно, то немає основ змінювати $\bar{a}$. У випадку (2) $\bar{a}$ варто змінити так, щоб збільшити $\bar{a}^T \bar{x}^*$. Запропоноване правило задовольняє цю вимогу. Дійсно,

$$\bar{a}'^T \bar{x}^* = \bar{a}^T \bar{x}^* + c\bar{x}^{*T} \bar{x}^* > \bar{a}^T \bar{x}^*.$$

Відповідно у випадку (3) $\bar{a}'^T \bar{x}^* < \bar{a}^T \bar{x}^*$.

Важливе значення має вибір c. Можна, зокрема, вибрати $c = const$. При цьому показано, що якщо навчальні вибірки двох образів лінійно роздільні, то описана покрокова процедура сходиться, тобто будуть знайдені значення $\bar{a}$, при яких

$$D(\bar{x}^*) > 0, \text{ якщо } \bar{x}^* \in s_1,$$

$$D(\bar{x}^*) < 0, \text{ якщо } \bar{x}^* \in s_2.$$

Якщо ж вибірки лінійно нероздільні, то збіжність відсутня й оцінку $\bar{a}$, мінімізуючи число неправильних розпізнавань, знаходять методом стохастичної апроксимації.

3.3.2.3 Метод найближчих сусідів

Навчання в даному випадку складається із запам'ятовування всіх об'єктів навчальної вибірки. Якщо системі пред'явлений нерозпізнаний об'єкт $\bar{x}^*$, то вона відносить цей об'єкт до того образа (рис. 3.5), чий "представник" виявився ближче усіх до $\bar{x}^*$

Це правило найближчого сусіда. Правило k найближчих сусідів полягає в тому, що будується гіперсфера обсягу V з центром в $\bar{x}^*$ Розпізнавання здійснюється за більшістю "представників" якого-небудь образа, що з'явився усередині гіперсфери. Тут тонкість полягає в тому, щоб правильно (розумно) вибрати обсяг гіперсфери V, який повинний бути досить великим, щоб у гіперсферу потрапило відносно велике число "представників" різних образів, і досить маленьким, щоб не згладити нюанси, поділяючі образи границі. Метод найближчих сусідів має той недолік, що вимагає збереження всієї навчальної вибірки, а не її узагальненого опису. Зате він дає гарні результати на контрольних

випробуваннях, особливо при великих кількостях об’єктів, пред’явлених для навчання.

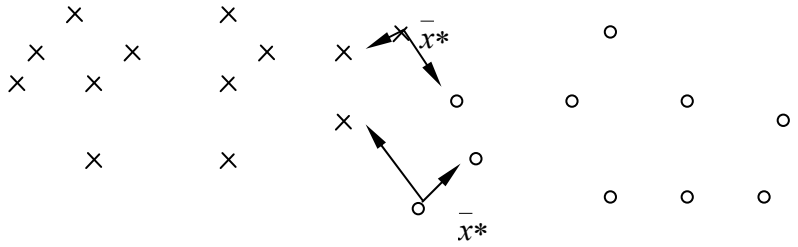


Рисунок 3.5 - Вирішальне правило "Мінімум відстані до найближчого сусіда"

Для скорочення числа об’єктів, що запам’ятовуються, можна застосовувати комбіновані вирішальні правила, наприклад поєднання методу еталонів, що дробляться, і методу найближчих сусідів.

У цьому випадку запам’ятовуванню підлягають ті об’єкти, що потрапили в зону перекриття гіперсфер якого-небудь рівня. Метод найближчих сусідів застосовується лише для тих розпізнаваних об’єктів, що потрапили в дану зону перекриття. Іншими словами, запам’ятовуванню підлягають не всі об’єкти навчальної вибірки, а тільки ті, котрі знаходяться поблизу розділяючою образи границі.

3.3.2.4 Метод потенційних функцій

Назва методу деякою мірою пов’язана з такою аналогією (для простоти будемо вважати, що розпізнається два образи). Уявимо собі, що об’єкти є точками $\overline{x_j}$ деякого простору X. У ці точки будемо поміщати заряди $+q_j$, якщо об’єкт належить образові s_1, і $-q_j$, якщо об’єкт належить образові s_2 (рис. 3.6).

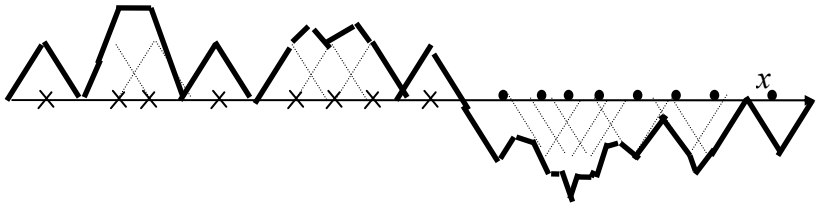


Рисунок 3.6 - Ілюстрація синтезу потенційної функції в процесі навчання:

..... – потенційна функція, породжувана одиночним об’єктом;

——— – сумарна потенційна функція, породжена навчальною послідовністю

Функцію, що описує розподіл електростатичного потенціалу в такому полі, можна використовувати як вирішальне правило (або для його

побудови). Якщо потенціал точки $\bar{x}$, створюваний одиничним зарядом, що знаходиться в $\overline{x_j}$, дорівнює $K(\bar{x},\overline{x_j})$, то загальний потенціал у $\bar{x}$, створюваний n зарядами, дорівнює

$$g(\bar{x}) = \sum_{j=1}^{n} q_j K(\bar{x},\overline{x_j}).$$

$K(\bar{x},\overline{x_j})$ – потенціальна функція. Вона, як у фізиці, спадає із зростанням евклідової відстані між $\bar{x}$ і $\overline{x_j}$. Найчастіше як потенціальна використовується функція, що має максимум при $\bar{x} = \overline{x_j}$ і монотонно спадає до нуля при $\|\bar{x} - \overline{x_j}\| \to \infty$.

Розпізнавання може здійснюватися в такий спосіб. У точці $\bar{x}$, де знаходиться непізнаний об'єкт, обчислюється потенціал $g(\bar{x})$. Якщо він виявляється позитивним, то об'єкт відносять до образу s_1. Якщо негативним – до образу s_2.

При великому обсязі навчальної вибірки ці обчислення досить громіздкі, і найчастіше вигідніше обчислювати не $g(\bar{x})$, а оцінювати поділяючу класи (образи) границю або апроксимувати потенційне поле.

Вибір виду потенційних функцій – справа непроста. Наприклад, якщо вони дуже швидко зменшуються із зростанням відстані, то можна домогтися безпомилкового поділу навчальних вибірок. Однак при цьому виникають певні неприємності при розпізнаванні непізнаних об'єктів (знижується вірогідність прийнятого рішення, зростає зона невизначеності). При занадто "положистих" потенційних функціях може необґрунтовано збільшитися кількість помилок розпізнавання, у тому числі і на навчальних об'єктах. Певні рекомендації в цьому відношенні можна одержати, розглядаючи метод потенційних функцій зі статистичних позицій (відновлення щільності розподілу імовірностей $p(\bar{x})$ або поділяючої границі за вибіркою з використанням процедури типу стохастичної апроксимації). Це питання виходить за рамки даного курсу.

3.3.2.5 Кластерний аналіз

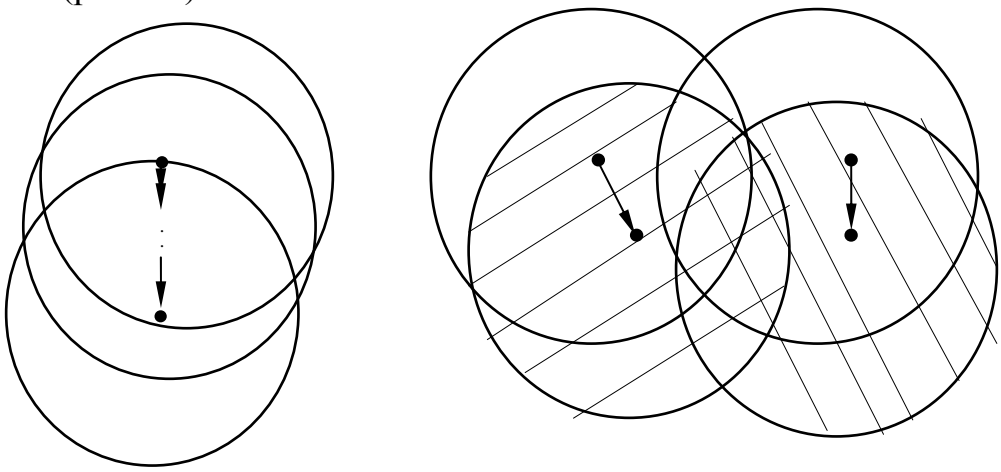
Як уже раніше відзначалося, кластерний аналіз (самонавчання, навчання без учителя, таксономія) застосовується при автоматичному формуванні переліку образів за навчальною вибіркою. Всі об'єкти цієї вибірки пред'являються системі без вказання, якому образові вони належать. Подібного роду задачі розв'язує, наприклад, людина в процесі природно-наукового пізнання навколишнього світу (класифікації рослин, тварин). Цей досвід доцільно використовувати при створенні відповідних алгоритмів.

В основі кластерного аналізу лежить гіпотеза компактності (*hypothesis of compactness*). Передбачається, що навчальна вибірка в ознаковому просторі складається з набору згустків (подібно до галактик у

Всесвіті). Задача системи – виявити і формалізовано описати ці згустки. Геометрична інтерпретація гіпотези компактності полягає в такому.

Об’єкти, що відносяться до одного таксона, розташовані близько один до одного порівняно з об’єктами, що відносяться до різних таксонів. "Близькість" можна розуміти ширше, ніж при геометричній інтерпретації. Наприклад, закономірність, що описує взаємозв’язок об’єктів одного таксона, відрізняється від такої в інших таксонах, як це має місце в лінгвістичних методах.

Обмежимося розглядом геометричної інтерпретації. Зупинимося на алгоритмі FOREL (рис.3.7).



а) б)

Рисунок 3.7 - Ілюстрація алгоритму FOREL:

а – процес переміщення формального елемента по безлічі об’єктів (точок);

б – ілюстрація залежності результатів таксономії за алгоритмом FOREL від початкової точки переміщення формального елемента

Будується гіперсфера радіуса $R_1 = R_{\max}$, де $R_{\max}$ відповідає гіперсфері, що охоплює всі об’єкти (точки) навчальної вибірки. При цьому число таксонів Q буде дорівнювати одиниці. Потім будується гіперсфера радіуса $R_2 = R_1 - \Delta R$ з центром у довільній точці вибірки. За безліччю точок, що потрапили усередину гіперсфери (формального елемента), визначається середнє значення координат (еталон) і в нього переміщується центр гіперсфери. Якщо це переміщення істотне, то помітно зміниться безліч точок, що потрапили усередину гіперсфери, а отже, і їхнє середнє значення координат. Знову переміщуємо центр гіперсфери в це нове середнє значення і т.д. доти, доки гіперсфера не зупиниться або зациклиться. Тоді всі точки, що потрапили усередину цієї гіперсфери, виключаються з розгляду і процес повторюється на точках, що залишилися. Це продовжується доти, доки не будуть вичерпані всі точки. Результат таксономії: набір гіперсфер (формальних елементів) радіуса R_1 з центрами,

визначеними в результаті вищеописаної процедури. Назвемо це циклом з формальними елементами радіуса R_1.

У наступному циклі використовуються гіперсфери радіуса $R_2 = R_1 - \Delta R = R_{\max} - 2\Delta R$. Тут з'являється параметр ΔR, обумовлений дослідником найчастіше підбором у пошуках компромісу: збільшення ΔR веде до зростання швидкості збіжності обчислювальної процедури, але при цьому зростає ризик утрати тонкостей таксономічної структури безлічі точок (об'єктів). Природно очікувати, що зі зменшенням радіуса гіперсфер кількість виділених таксонів буде збільшуватися. Якщо в ознаковому просторі навчальна вибірка складається з Q компактних згустків, що далеко знаходяться один від одного, то починаючи з деякого радіуса $R_i = R_{i-1} - \Delta R$ кілька циклів з радіусами формальних елементів $R_{i+1}, R_{i+2}, \ldots$ будуть завершуватися при однаковому числі Q таксонів. Наявність такої "полички" у послідовності циклів розумно пов'язувати з об'єктивним існуванням Q згустків, яким у відповідність ставлять таксони. Наявність декількох "поличок" може свідчити про ієрархії таксонів. На рисунку 3.8 зображено безліч точок, що мають дворівневу таксономічну структуру.

При усій своїй наочності і інтерпретації результатів алгоритм FOREL має істотний недолік: результати таксономії в більшості випадків залежать від початкового вибору центра гіперсфери радіуса R_1. Існують різні прийоми, що зменшують цю залежність, але радикального рішення цієї задачі немає.

Прикладом використання людських критеріїв при розв'язанні задач таксономії служить алгоритм KRAB. Ці критерії відпрацьовані на двовимірному ознаковому просторі в ході таксономії, здійснюваної людиною, і застосовані в алгоритмі, що функціонує з об'єктами довільної розмірності.

Фактори, виявлені при "людській" таксономії, можна сформулювати в такий спосіб:

– усередині таксонів об'єкти повинні бути якнайближче один до одного (узагальнений показник ρ);

– таксони повинні якнайдалі відстояти один від одного (узагальнений показник d);

– у таксонах кількість об'єктів повинна бути, за можливості, однаковою, тобто їхнє розходження в різних таксонах потрібно мінімізувати (узагальнений показник h);

– усередині таксонів не повинно бути великих стрибків щільності точок, тобто кількості точок на одиницю об'єму (узагальнений показник λ).

Якщо вдасться вдало підібрати способи виміру ρ, d, h і λ, те можна домогтися гарного збігу "людської" і автоматичної таксономії. В алгоритмі KRAB використовується такий підхід.

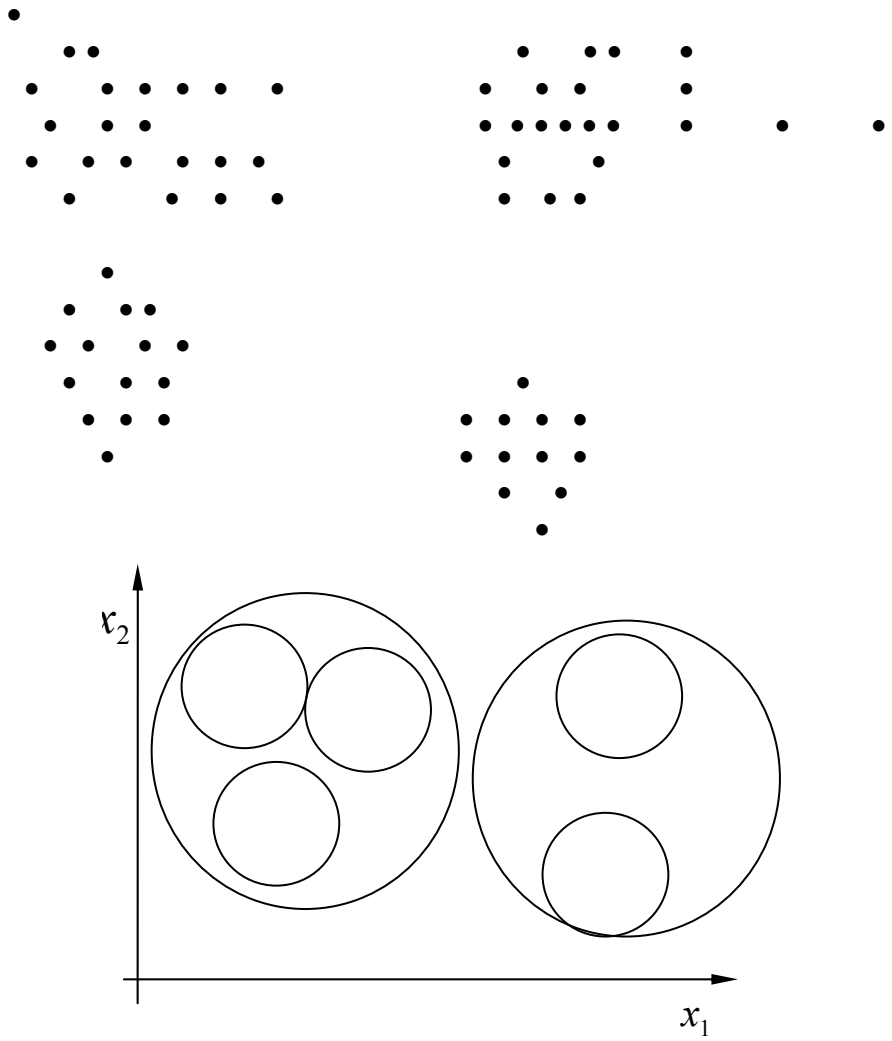


Рисунок 3.8 - Ієрархічна (дворівнева) таксономія

Усі точки навчальної вибірки поєднуються в граф, у якому вони є вершинами. Цей граф повинен мати мінімальну сумарну довжину ребер, що з'єднують усі вершини, і не містити петель (рис. 3.9). Такий граф називають ННШ-графом (ННШ – найкоротший незамкнутий шлях).

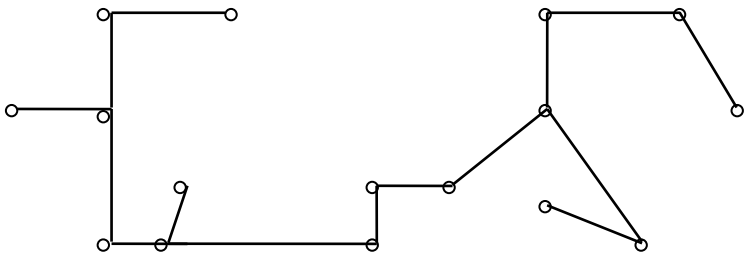


Рисунок 3.9 - Ілюстрація алгоритму KRAB

Міра близькості об'єктів в одному таксоні – це середня довжина ребра

$$\rho_q = \frac{1}{\left(m_q - 1\right)} \sum_{i=1}^{m_q - 1} \alpha_i ,$$

де m_q – число об'єктів у таксоні s_q,

α_i – довжина i -го ребра.

Усереднена по всіх таксонах міра близькості точок $\rho = \frac{1}{Q}\sum_{q=1}^{Q}\rho_q$.

Середня довжина ребер, що з'єднують таксони, $\alpha = \frac{1}{Q-1}\sum_{q=1}^{Q-1}\alpha_q$.

Міра локальної неоднорідності визначається в такий спосіб.

Якщо довжина i -го ребра α_i, а довжина найкоротшого ребра, що примикає до нього, $\beta_{\min}$, то чим менше $\lambda_i = \frac{\beta_{\min}}{\alpha_i}$, тим більша різниця у довжинах ребер, тим більше основ вважати, що по α_i ребру з довжиною пройде границя між таксонами. Узагальнюючий критерій

$$\lambda = \frac{1}{M-1}\sum_{i=1}^{M-1}\lambda_i,$$

де $M = \sum_{q=1}^{Q} m_q$ – загальне число точок у навчальній вибірці.

Визначимо величину h в такий спосіб:

$$h = \prod_{q=1}^{Q}\frac{m_q}{M}.$$

Можна показати, що при фіксованому Q максимум h досягається при $m_1 = m_2 = \ldots = m_q = \ldots m_Q$.

3.3.3 Статистичні методи

Говорячи про статистичні методи розпізнавання, ми припускаємо встановлення зв'язку між віднесенням об'єкта до того або іншого класу (образу) і імовірністю помилки при розв'язанні цієї задачі. У ряді випадків це зводиться до визначення апостеріорної імовірності належності об'єкта образові s_1 за умови, що ознаки цього об'єкта набули значення $x_1, x_2, \ldots, x_N$. Почнемо з байєсовського вирішального правила. За формулою Байєса

$$p(s_i / x_1,\ldots,x_N) = \frac{P_0(s_i)p(x_1,\ldots,x_N / s_i)}{\sum_{j-1}^{M} P_0(s_j)p(x_1,\ldots,x_N / s_j)}.$$

Тут $P_0(s_i)$ – апріорна імовірність пред'явлення до розпізнавання об'єкта i -го образа:

$$P_0(s_i) > 0,\ \sum_{i=1}^{M} P_0(s_i) = 1.$$

Для кожного s_i

$$\int_{x_1}\ldots\int_{x_N} p(x_1,\ldots,x_N / s_i)dx_1\ldots dx_N = 1,$$

при ознаках з безперервною шкалою вимірів

$$\sum_{j_1=1}^{x_1} \dots \sum_{j_N=1}^{x_N} p(x_1^{j_1}, x_2^{j_2}, \dots, x_N^{j_N} / s_i) = 1,$$

при ознаках з дискретною шкалою вимірів

$$p(x_1, \dots, x_N / s_i) \geq 0.$$

При безперервних значеннях ознак $p(x_1, \dots, x_N / s_i)$ являє собою функцію щільності імовірностей, при дискретних – розподіл імовірностей.

Розподіли, що описують різні класи, як правило, "перетинаються", тобто маються такі значення ознак $x_1, \dots, x_N$, при яких

$$p(x_1, \dots, x_N / s_i) \cdot p(x_1, \dots, x_N / s_j) \rangle 0.$$

У таких випадках помилки розпізнавання неминучі. Природно, нецікаві випадки, коли ці класи (образи) в обраній системі ознак $(\overline{X}_1, \dots, \overline{X}_N)$ нерозрізнені (при однакових апріорних імовірностях рішення можна вибирати випадковим віднесенням об'єкта до одного з класів рівноможливим чином).

У загальному випадку потрібно прагнути вибрати вирішальні правила так, щоб мінімізувати ризик втрат при розпізнаванні.

Ризик втрат визначається двома компонентами: імовірністю помилок розпізнавання і величиною "штрафу" за ці помилки (втратами). Матриця помилок розпізнавання:

	s_1	s_2	s_3	...	s_M
s_1	p_{11}	p_{12}	p_{13}		p_{1M}
s_2	p_{21}	p_{22}	p_{23}		p_{2M}
s_M	p_{M1}	p_{M2}	p_{M3}		p_{MM},

де P_{ii} – імовірність правильного розпізнавання;

P_{ij} – імовірність помилкового віднесення об'єкта i-го образу до j-го $(i \neq j)$.

Матриця втрат

	s_1	s_2	s_3	...	s_M
s_1	u_{11}	u_{12}	u_{13}		u_{1M}
s_2	u_{21}	u_{22}	u_{23}		u_{2M}
s_M	u_{M1}	u_{M2}	u_{M3}		u_{MM},

де u_{ii} – "премія" за правильне розпізнавання;

u_{ij} – "штраф" за помилкове віднесення об'єкта i-го образу до j-го $(i \neq j)$.

Необхідно побудувати вирішальне правило так, щоб забезпечити мінімум математичного сподівання втрат (мінімум середнього ризику). Таке правило називається байєсовським. Розіб'ємо ознаковий простір X на M непересічних зон $v_j (j = 1,2,..., M)$, кожна з яких відповідає певному образу.

Середній ризик при потраплянні реалізацій q-го образу в зоні інших образів дорівнює

$$r_q = \sum_{j=1}^{M} u_{qj} \int_{v_j} p(x_1, x_2, ..., x_N / s_q) dx_1 ... dx_N , \; q \neq j .$$

Тут передбачається, що усі компоненти X мають неперервну шкалу вимірів (у даному випадку це непринципово).

Величину r_q можна назвати умовним середнім ризиком (за умови, що зроблено помилку при розпізнаванні об'єкта q-го образу). Загальний (безумовний) середній ризик визначається величиною $R = \sum_{q=1}^{M} P_0(s_q) r_q .$

Вирішальні правила (способи розбивки X на $v_j, j = 1, \; K, M$) утворять безліч D. Найкращим (байєсовським) вирішальним правилом є те, що забезпечує мінімальний середній ризик $R_Б = \min_D R_D$, де R_D – середній ризик при застосуванні одного з вирішальних правил, що входять в D.

Розглянемо спрощений випадок. Нехай $u_{ii} = 0$, а $u_{ij} = 1$ $(i \neq j)$. У такому випадку байєсовське вирішальне правило забезпечує мінімум імовірності (середньої кількості) помилок розпізнавання. Нехай $M = 2$. Імовірність помилки першого роду (об'єкт одного образу віднесений до другого образу)

$$p_{12} = \int_{v_2} p(\bar{x} / s_1) d\bar{x} ,$$

де $\bar{x} = \{x_1, ..., x_N\}$ – імовірність помилки другого роду

$$p_{21} = \int_{v_1} p(\bar{x} / s_2) d\bar{x} .$$

Середні помилки

$$P_{ош} = P_0(s_1) p_{12} + P_0(s_2) p_{21} .$$

Тому що $\int_X p(\bar{x} / s_i) d\bar{x} = 1$, то $p_{12} = 1 - \int_{v_1} p(\bar{x} / s_1) d\bar{x}$ і $P_{ош} = P_0(s_1) + \int_{v_1} \left[P_0(s_2) p(\bar{x} / s_2) - P_0(s_1) p(\bar{x} / s_1) \right] d\bar{x}$. Ясно, що $P_{ош}$ буде мати мінімум у тому випадку, коли підінтегральний вираз в зоні v_1 буде строго негативним, тобто в v_1 $P_0(s_1) p(\bar{x} / s_1) > P_0(s_2) p(\bar{x} / s_2)$ В зоні v_2 повинна виконуватися протилежна нерівність. Це і є байєсовське вирішальне

правило для розглянутого випадку. Воно може бути записане інакше: $\frac{p(\bar{x}/s_1)}{p(\bar{x}/s_2)} > \frac{P_0(s_2)}{P_0(s_1)}$; величина $p(\bar{x}/s_i)$, розглянута як функція від s_i, називається правдоподібністю s_i при даному $\bar{x}$, а $\frac{p(\bar{x}/s_1)}{p(\bar{x}/s_2)}$ – відношенням правдоподібності. Таким чином, байєсовське вирішальне правило можна сформулювати як рекомендацію вибирати рішення s_1 у випадку, коли відношення правдоподібності перевищує певне граничне значення, що не залежить від спостережуваного $\bar{x}$.

Без спеціального розгляду вкажемо, що якщо число розпізнаваних класів більше двох ($S > 2$), рішення на користь класу (образу) s_j береться в зоні v_j, у якій для усіх $i \neq j$ $P_0(s_j)p(\bar{x}/s_j) > P_0(s_i)p(\bar{x}/s_i)$.

Іноді при невисокій точності оцінки апостеріорної імовірності (малих обсягах навчальної вибірки) використовують так звані рандомізовані вирішальні правила. Вони полягають у тому, що невідомий об'єкт відносять до того або іншого образу не за максимумом апостеріорної імовірності, а випадковим чином, відповідно до апостеріорних імовірностей цих образів $p(s_i/\bar{x}*)$. Реалізувати це можна, наприклад, способом, зображеним на рисунку 3.10.

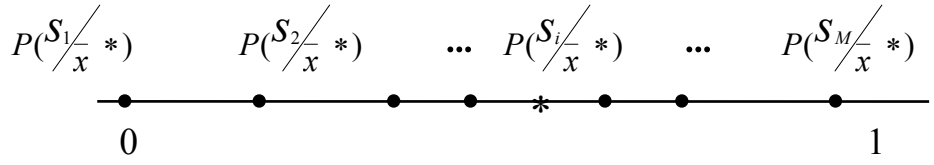


Рисунок 3.10 - Ілюстрація рандомізированого вирішального правила

Після обчислення апостеріорних імовірностей належності невідомого об'єкта з параметрами $\bar{x}*$ до кожного з образів s_i, $i = 1,\ldots,S$, відрізок прямої, довжиною одиниця, розбивають на S інтервалів з довжинами, чисельно рівними $p(s_i/\bar{x}*)$, і кожному інтервалові ставлять у відповідність цей образ. Потім за допомогою датчика випадкових (псевдовипадкових) чисел, рівномірно розподілених на $[0,1]$, генерують число, визначають інтервал, у який воно потрапило, і відносять розпізнаваний об'єкт до того образу, якому відповідає даний інтервал.

Зрозуміло, що таке вирішальне правило не може бути краще байєсовського, але при великих значеннях відношення правдоподібності ненабагато йому поступається, а в реалізації може виявитися досить простим (наприклад, метод найближчого сусіда, про що мова йтиме пізніше).

Байєсовське вирішальне правило реалізується в комп'ютерах, в основному, двома способами.

1. Пряме обчислення апостеріорних імовірностей

$$p(s_i / \bar{x}*) = \frac{P_0(s_i)\, p(\bar{x}*/s_i)}{\sum_{j=1}^{M} P_0(s_j)\, p(\bar{x}*/s_j)},$$

де $\bar{x}* = \{x_1*, x_2*, \ldots, x_N*\}$ – вектор значень параметрів розпізнаваного об'єкта і вибір максимуму. Рішення приймається на користь того образу, для якого $p(s_i / \bar{x}*)$ максимальне. Іншими словами, байєсовське вирішальне правило реалізується розв'язанням задачі $\arg\left\{\max_i p(s_i / \bar{x}*)\right\}$.

Якщо піти на подальше узагальнення і допустити наявність матриці втрат загального вигляду, то умовний ризик можна визначити за формулою $p(s_i / \bar{x}*)u_{ii} - \sum_{j=1}^{M} p(s_j / \bar{x}*)u_{ij}$, $i \neq j$. Тут перший член означає "заохочення" за правильне розпізнавання, а другий – "покарання" за помилку. Байєсовське вирішальне правило в даному випадку складається в розв'язанні задачі

$$\arg\left\{\max_i \left[p(s_i/\bar{x}*)u_{ii} - \sum_{j=1}^{M} p(s_j / \bar{x}*)u_{ij} \right]\right\}, i \neq j.$$

2. "Топографічне" визначення зони v_i, у яку потрапив вектор $\bar{x}*$ значень ознак, що описують розпізнаваний об'єкт.

Такий підхід використовують у тих випадках, коли опис зон v_i досить компактний, а процедура визначення зони, у яку потрапив $\bar{x}*$, проста. Іншими словами, даний підхід природно використовувати, коли в обчислювальному відношенні він ефективніший (простіший) за пряме обчислення апостеріорних імовірностей.

Так, наприклад (доведення наводити не будемо), якщо класів два, їхні апріорні імовірності однакові, $p(\bar{x} / s_1)$ і $p(\bar{x} / s_2)$ – нормальні розподіли з однаковими коваріаційними матрицями (відрізняються тільки векторами середніх), то байєсовська поділяльна границя – гіперплощина. Запам'ятовується вона значеннями коефіцієнтів лінійного рівняння. При розпізнаванні якого-небудь об'єкта в рівняння підставляють значення ознак $\bar{x}*$ цього об'єкта і за знаком (плюс або мінус) одержуваного рішення відносять об'єкт до s_1 або s_2 (рисунок 3.11).

Якщо в класів s_1 і s_2 коваріаційні матриці $p(\bar{x} / s_1)$ і $p(\bar{x} / s_2)$ не тільки однакові, але і діагональні, то байєсовським рішенням є віднесення об'єкта до того класу, евклідова відстань до еталона якого мінімальна (рис. 3.12).

Таким чином, можна переконатися в тому, що деякі вирішальні правила, які раніше були розглянуті як емпіричні (детерміновані, евристичні), мають цілком чітке статистичне трактування. Більш того, у ряді конкретних випадків вони є статистично оптимальними. Список

подібних прикладів ми продовжимо при подальшому розгляді статистичних методів розпізнавання.

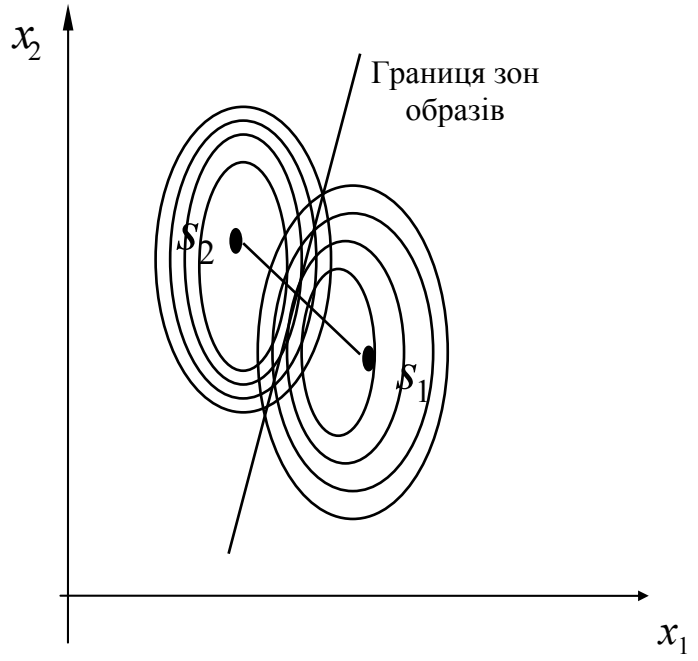


Рисунок 3.11 - Байєсовське вирішальне правило для нормально розподілених ознак з рівними коваріаційними матрицями

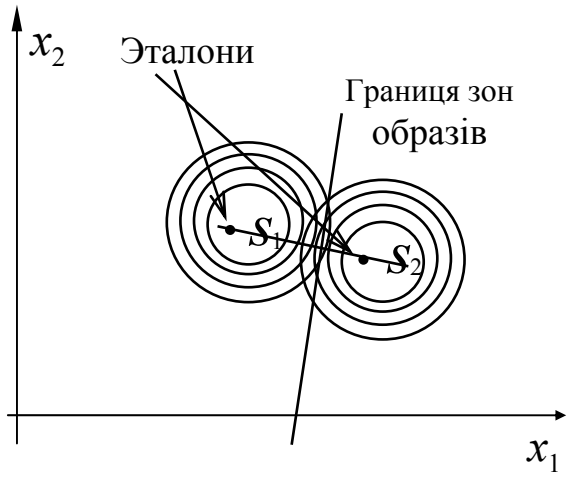


Рисунок 3.12 - Байєсовське вирішальне правило для нормально розподілених ознак з рівними діагональними коваріаційними матрицями (елементи діагоналей однакові)

Тепер перейдемо до методів оцінки розподілів значень ознак класів. Знання $p(\bar{x} / s_i)$ є найбільш універсальною інформацією для розв'язання задач розпізнавання статистичними методами. Цю інформацію можна одержати двояким чином:

1. заздалегідь визначити (оцінити) $p(\bar{x} / s_i)$ для всіх $\bar{x} \in X$ і $i = 1,2,\ldots, M$;

2. визначати $p(\bar{x}*/s_i)$ при кожнім акті розпізнавання конкретного об'єкта, ознаки якого мають значення $\bar{x}*$.

Кожний з цих підходів має свої переваги і недоліки, що залежать від числа ознак, обсягу навчальної вибірки, наявності апріорної інформації і т.п.

Почнемо з локального варіанта (визначення $p(\bar{x}*/s_i)$ в околиці розпізнаваного об'єкта).

3.3.3.1 Метод k_n найближчих сусідів

Тут ідея полягає в тому, що навколо розпізнаваного об'єкта $\bar{x}$ будується осередок об'єму V. При цьому невідомий об'єкт відноситься до того образу, число навчальних представників якого в побудованому осередку виявилося більшістю. Якщо використовувати статистичну термінологію, то число об'єктів образу s_i, що потрапили в даний осередок, характеризує оцінку усередненої за обсягом V щільності імовірності $p(\bar{x}/s_i)$.

Для оцінки усереднених $p(\bar{x}/s_i)$ потрібно вирішити питання про співвідношення між обсягом V осередку і кількістю об'єктів, що потрапили в цей осередок, того або іншого класу (образу). Цілком розумно вважати, що чим менше V, тим більш точно буде охарактеризована $p(\bar{x}/s_i)$. Але при цьому менше об'єктів потрапить у цікавлячий нас осередок, а отже, тим менша вірогідність оцінки $p(\bar{x}/s_i)$. При надмірному збільшенні V зростає вірогідність оцінки $p(\bar{x}/s_i)$, але губляться тонкості її опису через усереднення по занадто великому об'єму, що може призвести до негативних наслідків (збільшенню імовірності помилок розпізнавання). При невеликому об'ємі навчальної вибірки V доцільно брати гранично великим, але забезпечити при цьому, щоб усередині осередку щільності $p(\bar{x}/s_i)$ мало змінювалися. Тоді їхнє усереднення по великому об'єму не дуже небезпечно. Таким чином, цілком може статися, що обсяг осередку, доречний для одного значення $\bar{x}$, може зовсім не годитися для інших випадків.

Пропонується такий порядок дій (поки що належність об'єкта тому або іншому образу враховувати не будемо).

Для того щоб оцінити $p(\bar{x})$ на підставі навчальної вибірки, що містить n об'єктів, центруємо осередок навколо $\bar{x}$ і збільшуємо її об'єм доти, доки вона не вмістить k_n об'єктів, де k_n є деяка функція від n. Ці k_n об'єктів будуть найближчими сусідами $\bar{x}$. Імовірність P потрапляння вектора $\bar{x}$ в зону R визначається виразом $P = \int_R p(\bar{x}')d\bar{x}'$.

Це згладжений (усереднений) варіант щільності розподілу $p(\bar{x})$. Якщо узяти вибірку з n об'єктів (простим випадковим вибором з

генеральної сукупності), то k з них виявиться усередині зони R. Імовірність потрапляння k з n об'єктів в R описується біноміальним законом, що має різко виражений максимум біля середнього значення nP. При цьому k/n є непоганою оцінкою для P.

Якщо тепер допустити, що зона R настільки мала, що $p(\bar{x})$ усередині неї змінюється незначно, то

$$\int_R p(\bar{x}')d\bar{x}' \approx p(\bar{x})V,$$

де V – об'єм зони R, $\bar{x}$ – точка усередині R.

Тоді $P \approx p(\bar{x})V$. Але $P \approx \frac{k}{n}$, отже, $p(\bar{x}) \approx \frac{k/n}{V}$.

Отже, оцінкою $p_n(\bar{x})$ щільності $p(\bar{x})$ є величина

$$p_n(\bar{x}) = \frac{k_n/n}{V_n}. \tag{3.1}$$

Без доведення наведемо твердження, що умови

$$\lim_{n\to\infty} k_n = \infty \text{ і } \lim_{n\to\infty} \frac{k_n}{n} = 0 \tag{3.2}$$

є необхідними і достатніми для збіжності $p_n(\bar{x})$ до $p(\bar{x})$ за ймовірністю у всіх точках, де щільність $p(\bar{x})$ неперервна.

Цю умову задовольняє, наприклад, $k_n = \sqrt{n}$.

Тепер будемо враховувати належність об'єктів до того або іншого образу і спробуємо оцінити апостеріорні імовірності образів $p(s_i/\bar{x})$.

Припустимо, що ми розміщаємо осередок об'ємом V навколо $\bar{x}$ і захоплюємо вибірку з кількістю об'єктів k_n, k_{ni} з яких належать образу s_i. Тоді відповідно до формули (3.1) оцінкою спільної імовірності $p(\bar{x},s_i)$ буде величина

$$p_n(\bar{x},s_i) = \frac{k_{ni}/n}{V_n},$$

а

$$P_n(s_i/\bar{x}) = \frac{p_n(\bar{x},s_i)}{\sum_{j=1}^{M} p_n(\bar{x},s_j)} = \frac{k_{ni}}{k_n}.$$

Таким чином, апостеріорна імовірність $P_n(s_i/\bar{x})$ оцінюється як частка вибірки в осередку, що відноситься до s_i Щоб звести рівень помилки до мінімуму, потрібно об'єкт із координатами $\bar{x}$ віднести до класу (образу), кількість об'єктів навчальної вибірки якого в осередку максимальна. При $n \to \infty$ таке правило є байєсовським, тобто забезпечує теоретичний мінімум імовірності помилок розпізнавання (зрозуміло, при цьому повинні виконуватися умови (3.2)).

3.3.3.2 Правило найближчого сусіда

Нехай $X_n = \left\{\overline{x_1}, \overline{x_2}, ..., \overline{x_n}\right\}$ – безліч об'єктів навчальної послідовності, тобто приналежність кожного з них до того або іншого образу напевне відома. Нехай також $\bar{x}* \in X_n$ є об'єктом, найближчим до розпізнаваного $\bar{x} \notin X_n$. Нагадаємо, що при цьому правило найближчого сусіда для класифікації $\bar{x}$ полягає в тому, що $\bar{x}$ відносять до того класу (образу), якому належить $\bar{x}*$. Природно, таке віднесення носить випадковий характер. Імовірність того, що $\bar{x}$ буде віднесений до s_i, є апостеріорна імовірність $P(s_i / \bar{x}*)$. Якщо n дуже велике, то цілком можна допустити, що $\bar{x}$ розташовано досить близько до $\bar{x}*$, настільки близько, що $P(s_i / \bar{x}*) \approx P(s_i/\bar{x})$. А це є не що інше, як рандомізоване вирішальне правило: $\bar{x}$ відносять до s_i з імовірністю $P(s_i / \bar{x})$. Байєсовське вирішальне правило засноване на виборі максимальної апостеріорної імовірності, тобто $\bar{x}$ відносять до s_j у тому випадку, якщо

$$P(s_j / \bar{x}) = \max_i P(s_i / \bar{x}).$$

Звідси видно, що якщо $P(s_j / \bar{x})$ близько до одиниці, то правило найближчого сусіда дає рішення, яке у більшості випадків збігається з байєсовським. Нагадаємо, що ці міркування мають достатні підстави лише при дуже великих n (об'ємах навчальної вибірки). Такі умови на практиці зустрічаються не часто, але дозволяють зрозуміти статистичний зміст правила найближчого сусіда.

Параметричне оцінювання розподілів реалізується в тих випадках, коли відомий вигляд розподілів $p(\bar{x} / s_i)$ і за навчальною вибіркою необхідно лише оцінити значення параметрів цих розподілів. Апріорне знання вигляду $p(\bar{x} / s_i)$ на практиці зустрічається нечасто, однак, з огляду на зручність даного підходу, інший раз роблять допущення, наприклад, про те, що $p(\bar{x} / s_i)$ - нормальний закон. Такого роду допущення далеко не завжди мають переконливі підстави, але проте використовуються, якщо результати навчання приводять до прийнятних помилок розпізнавання.

Отже, навчання зводиться до оцінки значень параметрів $p(\bar{x} / s_i)$ при заздалегідь відомому вигляді цих розподілів. Особливе місце серед розподілів займає нормальний закон. Це пов'язано з тим, що, як відомо з математичної статистики, якщо випадкова величина породжена впливом досить великого числа випадкових факторів з довільними законами розподілу і серед цих впливів немає явно домінуючого, то цікавляча нас величина має нормальний закон розподілу. Для одновимірного випадку

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(m-x)^2}{2\sigma^2}}$$

(для простоти надалі будемо розглядати одновимірний випадок).

Параметрами цього розподілу є дві величини: m – математичне сподівання, σ^2 – дисперсія. Їх і потрібно оцінити за вибіркою. Одним з найбільш простих є метод моментів. Він застосовний для розподілів $p(\bar{x})$, що залежать від L параметрів, які мають L кінцевих перших моментів, що можуть бути виражені як явні функції $\varphi_l(a_1,...,a_L)$ параметрів $a_1,...,a_L$. Тоді, обчисливши по вибірці $\overline{x_1},\overline{x_2},...,\overline{x_n}$ L перших її моментів і прирівнявши їх до $\varphi_l(\cdot)$, одержимо систему рівнянь

$$\widetilde{\varphi}_l(a_1,...,a_L)=\frac{1}{n}\sum_{i=1}^{n}x_i^l=\varphi_l(a_1,...,a_l)=\int p(x,a_1,...,a_L)x^l dx,$$

з якої визначаються оцінки $\widetilde{a}_1,...,\widetilde{a}_L$.

Для одновимірного нормального закону

$$\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}e^{-\frac{(m-x)^2}{2\sigma^2}}xdx=m,\ \ \widetilde{m}=\frac{1}{n}\sum_{i=1}^{n}x_1\ .$$

$$\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}e^{-\frac{(m-x)^2}{2\sigma^2}}x^2dx=m^2+\sigma^2,\ \ \widetilde{\sigma}^2=\frac{1}{n}\sum_{i=1}^{n}x_i^2-\widetilde{m}^2\ .$$

3.3.3.3 Метод максимуму правдоподібності

Функція правдоподібності, уведена Фішером, виглядає в такий спосіб:

$$G(x_1,x_2,...,x_n;a)=p(x_1;a)p(x_2;a)...p(x_n;a)=\prod_{i=1}^{n}p(x_i/a),$$

де a – невідомий параметр.

Як оцінку параметра a потрібно вибрати таку величину, при якій $G(\cdot)$ досягає максимуму. Оскільки $G(\cdot)$ досягає максимуму при тому самому a, що і $\ln G(\cdot)$, то пошук необхідного значення оцінки a полягає в розв'язанні рівняння правдоподібності $\frac{d\ln G(\cdot)}{da}=0$. При цьому всі корені $a=const$ варто відкинути, а залишити тільки ті, котрі залежать від $x_1,...,x_n$.

Оцінка параметра розподілу є випадковою величиною, що має математичне сподівання і "розсіювання" довкола нього. Оцінка називається ефективною, якщо її "розсіювання" навколо свого математичного сподівання мінімально.

Справедлива така теорема (наводиться без доведення). Якщо існує для a ефективна оцінка $\widetilde{a}$, то рівняння правдоподібності має єдиний розв'язок. Цей розв'язок при $n\to\infty$ прямує до значення a.

Усе це справедливо і при декількох невідомих параметрах. Наприклад, для одновимірного нормального закону

$$\ln G = \sum_{i=1}^{n}\left[\ln\frac{1}{\sqrt{2\pi}\sigma} - \frac{(m-x_i)^2}{2\sigma^2}\right],$$

$$\frac{\partial \ln G}{\partial m} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - m) = 0,$$

звідси $\tilde{m} = \frac{1}{n}\sum_{i=1}^{n} x_i$ при $\sigma^2 \neq 0$.

$$\frac{\partial \ln G}{\partial \sigma^2} = \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i - m)^2 - \frac{n}{2\sigma^2} = 0,$$

звідси $\tilde{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(\tilde{m} - x_i)^2$.

Оцінка називається незміщеною, якщо математичне сподівання $\mu(\tilde{a})$ оцінки $\tilde{a}$ дорівнює a. Оцінка $\hat{m} = \frac{1}{n}\sum_{i=1}^{n} x_i$ є незміщеною. Дійсно, оскільки $x_1,\ldots,x_n$ – простий випадковий вибір з генеральної сукупності, то $\mu(x_1) = \mu(x_2) = \ldots = m$ і $\mu(\hat{m}) = \mu\left\{\frac{1}{n}\sum_{i=1}^{n} x_i\right\} = \frac{1}{n}\sum_{i=1}^{n}\mu(x_i) = m$.

З'ясуємо, чи є $\tilde{\sigma}^2$, отримана методом максимуму правдоподібності (або методом моментів), незміщеною. Легко переконатися, що

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{m})^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \hat{m}^2.$$

Отже,

$$\tilde{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \frac{1}{n^2}(\sum_{i=1}^{n} x_i)^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \frac{1}{n^2}\sum_{i=1}^{n} x_i^2 - \frac{2}{n^2}\sum_{i<j} x_i x_j =$$

$$= \frac{n-1}{n^2}\sum_{i=1}^{n} x_i^{\,2} - \frac{2}{n^2}\sum_{i<j} x_i x_j.$$

Знайдемо математичне сподівання цієї величини:

$$\mu(\tilde{\sigma}^2) = \frac{n-1}{n^2}\sum_{i=1}^{n}\mu(x_i^2) - \frac{2}{n^2}\sum_{i<j}\mu(x_i x_j).$$

Оскільки дисперсія $\tilde{\sigma}^2$ не залежить від значення m, то виберемо $m = 0$. Тоді

$$\mu(x_i^2) = \mu(\overset{0}{x}_i^2) = \sigma^2,\ \sum_{i=1}^{n}\mu(x_i^2) = \sigma^2 n,\ \mu(x_i x_j) = \mu(\overset{0}{x}_i\,\overset{0}{x}_j) = k_{ij} = 0,$$

де $\overset{0}{x}_i = x_i - m$, k_{ij} – коефіцієнт кореляції між $\overset{0}{x}_i$ і $\overset{0}{x}_j$ (у даному випадку він дорівнює нулю, тому що x_i і x_j не залежать один від одного).

Отже, $\mu(\tilde{\sigma}^2) = \frac{n-1}{n}\sigma^2$. Звідси видно, що оцінка $\tilde{\sigma}^2$ не є незміщеною, її математичне сподівання трохи менше, ніж σ^2. Для ліквідації даного зсуву

необхідно помножити $\tilde{\sigma}^2$ на $\frac{n}{n-1}$. У результаті одержимо незміщену оцінку

$$\tilde{\sigma}_*^2 = \frac{n}{n-1}\tilde{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \tilde{m})^2.$$

При статистично незалежних ознаках істотно спрощується розв'язання задач розпізнавання. Зокрема, при оцінюванні розподілів $p(\bar{x}/s_i)$ замість багатовимірних щільностей імовірності досить оцінити N одновимірних щільностей $p(x_j/s_i), j = 1,2,\ldots,N.$

У зв'язку з цим розглянуті нами приклади одновимірних розподілів не тільки носять ілюстративний характер, але можуть безпосередньо використовуватися при розв'язанні практичних задач, якщо є переконливі підстави вважати ознаки, що характеризують об'єкти розпізнавання, статистично незалежними. При цьому

$$p(\bar{x}/s_i) = \prod_{j=1}^{N} p(x_j/s_i)$$

і формула Байєса, використовувана для обчислення апостеріорної імовірності належності об'єкта з ознаками $\bar{x}*$ образу s_i, набуває вигляду

$$P(s_i/\overline{x*}) = \frac{P_0(s_i)\prod_{j=1}^{N} p(x_j*/s_i)}{\sum_{k=1}^{M} P_0(s_k)\prod_{j=1}^{N} p(x_j*/s_k)}.$$

Зустрічаються практичні додатки теорії розпізнавання, коли ознаки вважають статистично незалежними без вагомих на те причин, але знаючи, що насправді ознаки (хоча б частина з них) взаємозалежні. Це робиться для спрощення процедур навчання і розпізнавання на шкоду "якості" (імовірності помилок), якщо цей збиток можна визнати прийнятним.

Особливо помітне спрощення процедури розпізнавання за методом Байєса, якщо ознаки набувають двійкових значень. У цьому випадку навчання складається в побудові такої таблиці (таблиця 3.1):

Якщо N не дуже велике, то віднесення невідомого об'єкта до того або іншого образу настільки спрощується, що найчастіше немає необхідності використовувати комп'ютер, досить калькулятора, а в крайньому випадку можна виконати розрахунок вручну. Тільки необхідно мати заздалегідь підготовлену таблицю зі значеннями p_{ij} (точніше, з їхніми оцінками). Якщо в невідомого об'єкта виявлена наявність тих або інших ознак, то для кожного з образів у відповідному рядку таблиці вибираються ті p_{ij}, котрі пов'язані з цими ознаками, і перемножуються. Об'єкт відносять до того класу, добуток для якого вийшов максимальним. При цьому, звичайно, здійснюється множення на апріорні імовірності, а нормування

апостеріорних імовірностей можна не здійснювати, тому що воно не впливає на результат вибору максимуму за s_j.

Таблиця 3.1 – Побудова таблиці навчання

		x_1	x_2	x_3	…	x_N
$P_0(s_1)$	s_1	p_{11}	p_{12}	p_{13}	…	p_{1N}
$P_0(s_2)$	s_2	p_{21}	p_{22}	p_{23}	…	p_{2N}
$P_0(s_3)$	s_3	p_{31}	p_{32}	p_{33}	…	p_{3N}
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
$P_0(s_M)$	s_M	p_{M1}	p_{M2}	p_{M3}	…	p_{MN}

Тут $p_{ij} = p(x_i / s_j)$, $x_i = 0,1$.

Якщо ознаки дискретні, але багатозначні, то до двійкових значень неважко перейти шляхом спеціального двійкового кодування.

Розпізнавання при невідомих апріорних імовірностях образів $P_0(s_i)$ можуть бути невідомі з багатьох причин, зокрема, якщо вони є невідомими функціями часу або яких-небудь неконтрольованих обставин, умов. У цьому випадку використовувати байєсовське вирішальне правило неможливе. Замість ризику втрат (в окремому випадку середньої імовірності помилок розпізнавання) доводиться мати справу з вектором (в окремому випадку $\overline{P_{ош}} = \left\{p_{ij}\right\}, \quad i \neq j$).

3.3.3.4 Мінімаксний критерій

Задача ставиться в такий спосіб: із усіх можливих наборів $P_0(s_i)$ при умовах $P_0(s_i) > 0$ і $\sum_{i=1}^{M} P_0(s_i) = 1$ необхідно вибрати такий (і надалі його використовувати при розпізнаванні), при якому максимальна компонента вектора $\overline{P_{ош}} = \left\{p_{ij}\right\}, \quad i \neq j$, мінімальна.

Алгоритмічно одним з найбільш простих є метод Монте-Карло. Випадковим чином n раз вибираються вектори $\overline{P_0} = \left\{P_0(s_1),\ldots,P_0(s_M)\right\}$. Той вектор, при якому максимальний компонент $\overline{P_{ош}}$ набуває найменшого значення, береться для використання. Чим більше n, тим вища імовірність "влучення" у найближчу околицю оптимального вектора $\overline{P_0}$. Можливий, звичайно, і повний перебір варіантів, але він прийнятний лише при не дуже великому числі можливих $\overline{P_0}$. У деяких часткових задачах може бути реалізований аналітичний підхід до пошуку $\overline{P_0}$. Розглянемо випадок із двома образами $M = 2$ (рисунок 3.13).

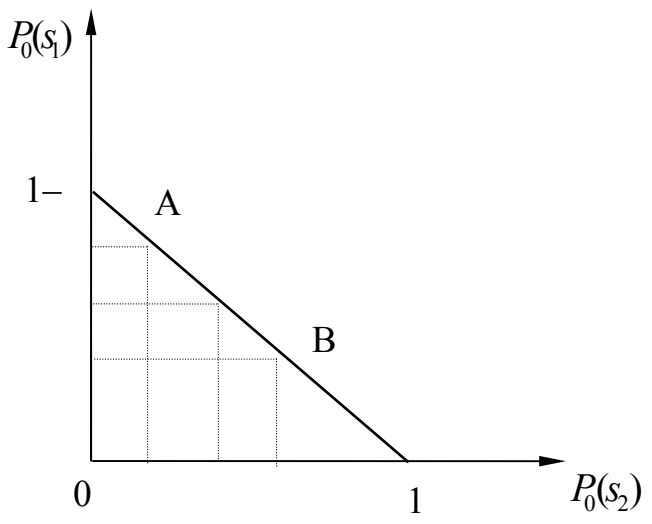


Рисунок 3.13 - Зона розв'язання задачі визначення $P_0(s_1)$ за мінімаксним критерієм

Розв'язання мінімаксної задачі лежить на відрізку прямої (A,B). Позначимо через v_1 зону об'єктів першого образу, а через v_2 – другого. Ясно, що $P_0(s_2) = 1 - P_0(s_1)$. При цьому середня імовірність помилок розпізнавання визначається величиною $p_{ош} = P_0(s_1)\int\limits_{v_2} p(\bar{x}/s_1)d\bar{x} + \left[1 - P_0(s_1)\right]\int\limits_{v_1} p(\bar{x}/s_2)d\bar{x}$. Побудуємо графік $p_{ош} = f\left[P_0(s_1)\right]$.

Очевидно, що $p_{ош} = 0$ при $P_0(s_1) = 0$ і $P_0(s_1)$=1. Між ними знаходяться значення $P_0(s_1)$, при яких $p_{ош} > 0$, у тому числі максимальне її значення. Вважатимемо, що ми вибрали $P_0(s_1)=b$. Тоді $p_{ош}$ як функція дійсного (але невідомого) значення $P_0(s_1)$ лежить на прямій, дотичній до $p_{ош} = f\left[P_0(s_1)\right]$ у точці, що відповідає $P_0(s_1)=b$. При цьому якщо дійсне значення $P_0(s_1)$ лежить зліва від точки b (наприклад, $P_0(s_1)=d$), то фактична середня помилка розпізнавання (P_d') виявиться меншою, ніж прогнозована при $P_0(s_1)=b$. Зате, якщо дійсне значення $P_0(s_1)=e$, то фактична середня помилка (P_e'') виявиться значно більшою за прогнозовану. Аналогічні міркування можна навести і для правого схилу кривої $p_{ош} = f\left[P_0(s_1)\right]$, поклавши, наприклад, $P_0(s_1)=c$. Лише вибравши $P_0(s_1)=a$, що відповідає максимуму кривої $p_{ош} = f\left[P_0(s_1)\right]$, ми гарантуємо, що $p_{ош}$ не перевершить $p_{\max}$, яким би не було дійсне значення $P_0(s_1)$ (рис. 3.14).

Розглянемо аналітичну постановку задачі пошуку мінімаксного рішення (при цьому варто мати на увазі, що p_{12} і p_{21} залежать від $P_0(s_1)$, оскільки вони є функціями від v_1 і v_2, а останні залежать від апріорних імовірностей образів).

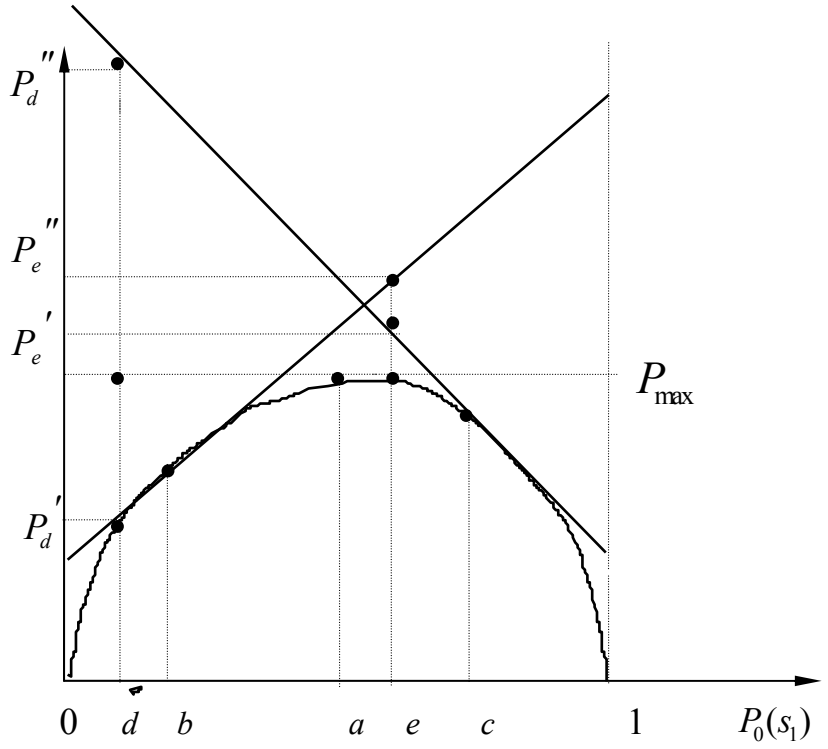


Рисунок 3.14 - Залежність імовірності помилки розпізнавання від $P_0(s_1)$

Позначимо $P_0(s_1)$ через z. Необхідно знайти таке значення z, при якому

$$\frac{dp_{ош}}{dz} = p_{12}(z) - p_{21}(z) + p_{21}'(z) + z\left[p_{12}'(z) - p_{21}'(z)\right] = 0,$$

де $p_{12} = \int_{v_2} p(\bar{x} / s_1)d\bar{x}$, $p_{21} = \int_{v_1} p(\bar{x} / s_2)d\bar{x}$.

З цього рівняння видно, що знайти аналітичне його розв'язання досить непросто. По-перше, необхідно записати в явному вигляді залежність p_{12} і p_{21} від $P_0(s_1)$, а по-друге, рівняння $\frac{dp_{ош}}{dz}$ повинно мати аналітичний розв'язок. У найпростіших випадках це можливо, але найпростіші випадки, на жаль, украй рідко зустрічаються на практиці.

3.3.3.5 Критерій Неймана-Пірсона

Нехай $N = 1, S = 2.$ Задамося допустимим значенням імовірності p_{12}, після чого будемо відшуковувати таку границю x_0 між образами, при якій досягається мінімум p_{21} .

Нехай потрібно $p_{12} \le A$. Потрібно розв'язати задачу перебування

$$x_0 = \arg(\min_x p_{21}) = \arg\left\{\min_x \int_{-\infty}^{x} p(y / s_2)dy\right\}.$$

Ясно, що розв'язок x_1 задовольняє умову $\int_{x_0}^{\infty} p(x/s_1)dx = A.$ Дійсно, при $x_0' > x_0$ p_{21} зростає, тобто стає не мінімально можливим, а при $x_0'' < x_0$ порушується обмеження $p_{12} \le A$. Як і в мінімаксному методі, знайти аналітично x_0 вдається лише в найпростіших випадках.

3.3.4 Послідовні процедури розпізнавання

Якщо в раніше розглянутих методах розпізнавання прийняття рішення про належність об'єкта тому або іншому образу здійснювалося відразу за всією сукупністю ознак, то в даному розділі ми обговоримо випадок послідовного їхнього вимірювання і використання.

Нехай $X = \{x_1, x_2, ..., x_N\}$. Спочатку в об'єкта вимірюється x_1 і на підставі цієї інформації вирішується питання про віднесення цього об'єкта до одного з образів. Якщо це можна зробити з достатнім ступенем упевненості, то інші ознаки не вимірюються і процедура розпізнавання закінчується. Якщо ж такої впевненості немає, то вимірюється ознака x_2 і рішення приймається за двома ознаками: x_1 і x_2. Далі процедура або припиняється, або виміряються ознака x_3, і так доти, доки не буде прийняте рішення про віднесення об'єкта до якого-небудь образу або будуть вичерпані усі N ознак.

Такі процедури надзвичайно важливі в тих випадках, коли вимірювання кожної з ознак вимагає істотних витрат ресурсів (матеріальних, часових і ін.).

Нехай $S = 2$ і відомі $p(x_1/s_i)$, $p(x_1,x_2/s_i)$, ... , $p(x_1,...,x_N/s_i)$, де $i = 1, 2.$ Помітимо, що коли відомо розподіл $p(x_1,...,x_N/s_i)$, то відомі і всі розподіли меншої розмірності (так звані маргінальні розподіли). Наприклад,

$$p(x_1,x_2/s_i) = \int_{-\infty}^{+\infty} ... \int_{-\infty}^{+\infty} p(x_1,...,x_N/s_i)dx_3dx_4...dx_N.$$

Нехай обміряно $n < N$ ознак. Будуємо відношення правдоподібності $\lambda_n = \frac{p(x_1,...,x_n/s_1)}{p(x_1,...,x_n/s_2)}$. Якщо $\lambda_n \ge A$, то об'єкт відносимо до образу s_1, якщо $\lambda_n \le B$, то до образу s_2. Якщо ж $A > \lambda_n > B$, то вимірюється ознака x_{n+1} й обчислюється відношення правдоподібності λ_{n+1} і т.д.

Зрозуміло, що пороги A і B пов'язані з припустимими імовірностями помилок розпізнавання. Домагаючись виконання нерівності $\lambda_n \ge A$, ми прагнемо до того, щоб імовірність правильного віднесення об'єкта першого образу до s_1 була в A раз більша, ніж помилкове віднесення об'єкта другого образу до s_1, тобто

$$\int_{v_1} p(x_1,...,x_n/s_1)dx_1...dx_n = A\int_{v_1} p(x_1,...,x_n/s_2)dx_1...dx_n \text{ або } p_{11} = Ap_{21}.$$

Оскільки $p_{11} = 1 - p_{12}$, то $A = \dfrac{1 - p_{12}}{p_{21}}$ (верхній поріг). Аналогічні міркування проводимо для визначення B. Домагаючись виконання нерівності $\lambda_n \le B$, ми прагнемо до того, щоб імовірність правильного віднесення об'єкта другого образу до s_2 була в $\dfrac{1}{B}$ раз більша, ніж неправильного віднесення об'єкта першого образу до s_2, тобто

$$\frac{1}{B}\int_{v_2} p(x_1,\dots,x_n / s_1)dx_1\dots dx_n = \int_{v_2} p(x_1,\dots,x_n / s_2)dx_1\dots dx_n ,$$

$$\frac{1}{B}p_{12} = p_{22} = 1 - p_{21},$$

$$B = \frac{p_{12}}{1 - p_{21}} \text{ (нижній поріг).}$$

У послідовній процедурі вимірювання ознак дуже корисною властивістю цих ознак є їхня статистична незалежність. Тоді $\lambda_n = \dfrac{\prod_{i=1}^{n} p(x_i / s_1)}{\prod_{i=1}^{n} p(x_i / s_2)}$ і немає необхідності в збереженні (а головне – у побудові) багатовимірних розподілів. До того ж є можливість оптимізувати порядок проходження вимірюваних ознак. Якщо їх ранжувати у порядку зменшення класифікаційної інформативності (кількість розпізнавальної інформації) і послідовну процедуру організувати відповідно до цього ранжування, можна зменшити в середньому кількість вимірюваних ознак.

Якщо образів два ($S = 2$) і кількість ознак необмежена, то послідовна процедура з імовірністю 1 закінчується за скінченне число кроків. Доведено також, що при заданих p_{12} і p_{21} розглянута процедура при однаковій інформативності різних ознак дасть мінімум середнього числа кроків. Для $S = 2$ послідовну процедуру увів Вальд і назвав її послідовним критерієм відношення імовірностей (ПКВІ).

Для $S > 2$ оптимальність процедури не доведена.

При відомих апріорних імовірностях можна реалізувати байєсовську послідовну процедуру, а якщо відомі витрати на вимірювання ознак і матриця штрафів за неправильне розпізнавання, то послідовну процедуру можна зупинити за мінімумом середнього ризику. Суть тут полягає в порівнянні втрат, викликаних помилками розпізнавання при припиненні процедури, і очікуваних втрат після наступного виміру плюс витрати на цей вимір. Така задача розв'язується методом динамічного програмування, якщо послідовні виміри статистично незалежні [40-42].

3.3.5 Апроксимаційний метод оцінки розподілів за вибіркою

Цей підхід розглядаємо окремо тому, що за своїми властивостями він досить універсальний. Крім оцінки (відновлення) розподілів, він дозволяє одночасно розв'язувати задачу таксономії, оптимізованого керування послідовною процедурою вимірювання ознак, навіть якщо вони статистично залежні, полегшує оцінку інформативності ознак і розв'язання деяких задач аналізу експериментальних даних.

В основі методу лежить припущення про те, що невідомий розподіл значень ознак кожного образу добре апроксимується поєднанням базових розподілів досить простого і заздалегідь відомого вигляду

$$p(\overline{x}) = \sum_{q=1}^{Q} \mu_q \varphi(\overline{x} / \overline{a_q}),$$

де $\overline{a_q}$ – безліч (вектор) параметрів q-го базового розподілу, $\mu_q > 0$ – вагові коефіцієнти, що задовольняють умову $\sum_{q=1}^{Q} \mu_q = 1$. Щоб не захаращувати формулу, у ній опущений індекс номера образу, що описується даним розподілом значень ознак.

Подання невідомого розподілу у вигляді ряду використовується, наприклад, у методі потенційних функцій. Однак там $\varphi_q(\overline{x})$ є елементами повної ортогональної системи функцій. З одного боку, це добре, тому що обчислення Q перших значень μ_q гарантує мінімум середньоквадратичного відхилення відновленого розподілу від дійсного. Але з іншого боку, $\varphi_q(x)$ не є розподілами, можуть бути знакозмінними, що при $Q < \infty$, як правило, призводить до неприйнятного ефекту, коли для деяких значень $\overline{x}$ відновлений розподіл $p(\overline{x})$ є негативним. Спроби уникнути цього ефекту далеко не завжди є успішними. Цього недоліку позбавлений розглянутий підхід, коли шуканий розподіл поданий "сумішшю" базових. При цьому $\varphi(\overline{x} / \overline{a_q})$ називають розподілами компонентів "суміші", а $\overline{a_q}, \mu_q$ і Q – її параметрами. У цього підходу при очевидних достоїнствах є недоліки. Зокрема, розв'язання задачі оцінки параметрів суміші є, багатоекстремумним, і немає гарантій, що знайдене рішення знаходиться в глобальному екстремумі, якщо виключити неприйнятний на практиці повний перебір варіантів розбивки суміші на компоненти.

Такі поняття, як "суміш", "компонента" звичайно використовуються при розв'язанні задач таксономії, але це не є перешкодою для опису у вигляді суміші досить загального виду розподілу значень ознак того або іншого образу. Апроксимаційний метод є ніби проміжним між параметричним і непараметричним оцінюванням розподілів. Дійсно, за вибіркою доводиться оцінювати значення параметрів $\overline{a_q}, \mu_q$ і Q, і в той же

час вид закону розподілу заздалегідь невідомий, на нього накладені лише найзагальніші обмеження. Наприклад, якщо компонента – нормальний закон, то щільність імовірності $p(\bar{x})$ не повинна перетворюватись на нуль при всіх $\bar{x}$ і бути досить гладкою. Якщо компонента – біноміальний закон, то $p(\bar{x})$ може бути практично будь-яким дискретним розподілом.

Разом з тим параметри суміші визначити класичними методами параметричного оцінювання (наприклад, методом моментів або максимуму функції правдоподібності) не є можливим за рідкими винятками (частинними випадками). У зв'язку з цим доцільно звернутися до методів, використовуваних при розв'язанні задач таксономії.

Як компоненту зручно використовувати біноміальні закони для дискретних ознак і нормальні щільності імовірностей для неперервних ознак, тому що властивості і теорія цих розподілів добре вивчені. До того ж вони, як показують практичні додатки, як компоненти досить адекватно описують досить широкий клас розподілів.

Нормальний закон, як відомо, характеризується вектором середніх значень ознак і матрицею коваріацій. Особливе місце в ряді задач займають нормальні закони з діагональними коваріаційними матрицями (у компонентах ознаки статистично незалежні). При цьому вдається оптимізувати послідовну процедуру виміру ознак, навіть якщо у відновленому розподілі $p(\bar{x})$ ознаки залежні.

Для полегшення розуміння апроксимаційного методу будемо розглядати спрощений варіант, а саме: одновимірні розподіли.

Отже,

$$p(x) = \sum_{q=1}^{Q} \mu_q \varphi(x / \overline{a_q}),$$

де $\varphi(x / \overline{a_q}) = \begin{cases} N(x / m_q, \sigma_q) = \dfrac{1}{\sqrt{2\pi}\sigma_q} \exp\left\{ -\dfrac{(x - m_q)^2}{2\sigma_q^2} \right\}, & \text{– для неперервних ознак,} \\ B_i(x_j / n, p_q) = C_n^{x_j} p_q^{x_j} (1 - p_q)^{n - x_j}, & \text{– для дискретних ознак,} \end{cases}$

$\overline{a_q}$ – вектор параметрів базового розподілу (компоненти),

μ_q – ваговий коефіцієнт q-ї компоненти,

m_q – математичне сподівання q-ї нормальної компоненти,

σ_q – середньоквадратичне відхилення q-ї нормальної компоненти,

p_q – параметр q-ї біноміальної компоненти,

$n+1$ – число градацій дискретної ознаки,

$x_j = 0, 1, 2, \ldots, n,$

$C_n^{x_j}$ – число сполучень з n за x_j.

При досить великому n замість біноміального закону можна використовувати нормальний.

Тепер потрібно оцінити значення параметрів. Якщо розглядати поєднання нормальних законів, то слід зазначити, що метод максимуму правдоподібності неможливо застосувати, коли всі параметри суміші невідомі. У такому випадку можна скористатися розумно організованими ітераційними процедурами. Розглянемо одну з них.

Для початку зафіксуємо Q, тобто будемо вважати його заданим. Кожному об'єктові x_i вибірки поставимо у відповідність апостеріорну імовірність α_{iq} належності його q-ї компоненти суміші:

$$\alpha_{iq} = \frac{\mu_q \varphi(x_i / \overline{a_q})}{\sum_{j=1}^{Q} \mu_j \varphi(x_i / \overline{a_j})}.$$

Одразу видно, що для усіх $i = 1, 2, \ldots, N$, виконуються умови $0 \le \alpha_{iq} \le 1$ і $\sum_{q=1}^{Q} \alpha_{iq} = 1$. Якщо відомі α_{iq} для усіх $i = 1,\ 2,\ \ldots, N$, то можна визначити μ_q методом максимуму правдоподібності $\mu_q = \frac{1}{N} \sum_{i=1}^{N} \alpha_{iq}$.

Функцію правдоподібності q-ї компоненти суміші визначимо в такий спосіб $l(\overline{a_q}) = \sum_{i=1}^{N} \alpha_{iq} \log \varphi(x_i / \overline{a_q})$, і оцінки максимальної правдоподібності $\overline{a_q}$ можна одержати з рівняння $\frac{\partial l(\overline{a_q})}{\partial \overline{a_q}} = 0.$

Таким чином, знаючи $\overline{a_q}$ і μ_q, можна визначити α_{iq}, і навпаки, знаючи α_{iq}, можна визначити $\overline{a_q}$ і $\mu_q = 0$, тобто параметри суміші. Але ні те, ні інше невідомо. У зв'язку з цим скористаємося такою процедурою послідовних наближень:

$$A^0 \Rightarrow \left\{\alpha_{iq}^0\right\} \Rightarrow A^1 \Rightarrow \left\{\alpha_{iq}^1\right\} \Rightarrow \ldots A^t \Rightarrow \left\{\alpha_{iq}^t\right\} \Rightarrow A^{t+1} \Rightarrow \left\{\alpha_{iq}^{t+1}\right\} \Rightarrow \ldots,$$

де $A = \left\{\mu_q, \overline{a_q}\right\}$, A^0 – довільно задані початкові значення параметрів суміші, верхній індекс – номер ітерації в послідовній процедурі обчислень.

Відомо, що ця процедура є збіжною і при $t \to \infty$ межею служать оцінки невідомих параметрів A суміші, що дають максимум функції правдоподібності

$$L(A) = \sum_{i=1}^{N} \log \sum_{q=1}^{Q} \mu_q \varphi(x_i / \overline{a_q}),$$

причому $L(A^{t+1}) \ge L(A^t)$ і $\left\{L(A^{t+1}) - L(A^t)\right\}$ прямує до нуля при $t \to \infty$.

Для одномірного нормального закону

$$l(\overline{a_q}) = \sum_{i=1}^{N} \alpha_{iq} \ln\left\{ \frac{1}{\sqrt{2\pi}\sigma_q} \exp\left[-\frac{(x_i - m_q)^2}{2\sigma_q^2} \right] \right\}.$$

Вирішуючи рівняння $\frac{\partial l(\overline{a_q})}{\partial m_q} = 0$ і $\frac{\partial l(\overline{a_q})}{\partial \sigma_q^2} = 0$, одержимо для $(t+1)$-го кроку $m_q(t+1) = \sum_{i=1}^{N} \alpha_{iq}^t x_i \Big/ \sum_{n=1}^{N} \alpha_{nq}^t$, $\sigma_q^2(t+1) = \sum_{i=1}^{N} \alpha_{iq}^t \left[x_i - m_q(t+1) \right]^2 \Big/ \sum_{n=1}^{N} \alpha_{nq}^t$, $q = 1,2,...,Q$.

Після завершення послідовної процедури обчислюються μ_q. Відповідні обчислювальні формули для багатовимірних нормальних розподілів можна знайти в роботі [33].

Одержувані в результаті розглянутої послідовної процедури значення параметрів є оцінками максимальної правдоподібності як щодо кожного компонента, так і відносно суміші в цілому [43-45].

Якщо $L(A)$ має кілька максимумів, то ітераційний процес залежно від заданих початкових значень A^0 сходиться до одного з них, не обов'язково глобального. Перебороти цей недолік, властивий практично всім методам оцінок параметрів багатовимірних розподілів (принаймні, сумішей) досить складно. Зокрема, можна повторити послідовну процедуру кілька разів при різних A^0 і вибрати найкраще з рішень. Вибір різних A^0 здійснюють або випадковим чином, або за допомогою різного роду спрямованих процедур. Швидкість збіжності $L(A)$ до максимуму тим вища, чим більше рознесені компоненти в ознаковому просторі і чим ближче обрані A^0 до значень A, що відповідають максимуму функції правдоподібності.

Ми розглянули метод оцінки параметрів суміші при фіксованому числі компонентів Q. Але в апроксимаційному методі і Q варто визначити, оптимізуючи який-небудь критерій. Пропонується такий підхід. Оцінюються послідовно параметри $A = \left\{ \mu_q, \overline{a_q} \right\}$ при $Q = 1,\ 2,\ 3,\ldots$. З отриманого ряду Q_1, Q_2, Q_3,... вибирається, в певному сенсі краще значення Q_{opt}.

Скористаємося мірою невизначеності К. Шеннона $H\left[p(\bar{x}) \right]$ для пошуку Q_{opt}:

$$H\left[p(\bar{x}) \right] = E\left[\log p(\bar{x}) \right] = -\int_X p(\bar{x}) \log p(\bar{x}) d\bar{x},$$

де $H\left[p(\bar{x}) \right]$ – ентропія розподілу $p(\bar{x})$,

$E[\cdot]$ – знак математичного сподівання,

$p(\bar{x})$ – щільність імовірності значень неперервних ознак.

При послідовному збільшенні значень Q мають місце дві тенденції:

1. Зменшення ентропії за рахунок поділу вибірки на частини зі зменшуваним розкидом значень величин $\bar{x}$, що спостерігаються, усередині підвибірок (компонент суміші);
2. Збільшення ентропії за рахунок зменшення об'єму підвибірок і пов'язаним з цим збільшенням статистик, що характеризують розкид значень $\bar{x}$.

Наявність цих двох тенденцій обумовлює існування Q_{opt} за критерієм найменшого значення диференціальної ентропії (рис. 3.15). Щоб використовувати на практиці цей критерій, необхідно в явному вигляді виразити оцінку компонентів суміші через об'єм підвибірки, що формує цей компонент. Такі співвідношення отримані для нормальних і біноміальних розподілів. Для простоти розглянемо одновимірний нормальний розподіл. Скористаємося його байєсовською оцінкою

$$\tilde{\varphi}_з(x) = \iint\limits_{(A)(B)} \varphi(x/m,\sigma) f(m,\sigma/m,\sigma) dm d\sigma,$$

де A – зона визначення m,

B – зона визначення σ,

m,σ – вибіркові оцінки m і σ.

Відповідно до формули Байєса

$$f(m,\sigma/m,\sigma) = \frac{P_0(m,\sigma) f(m,\sigma/m,\sigma)}{\iint\limits_{(A)(B)} P_0(m,\sigma) f(m,\sigma/m,\sigma) dm d\sigma}.$$

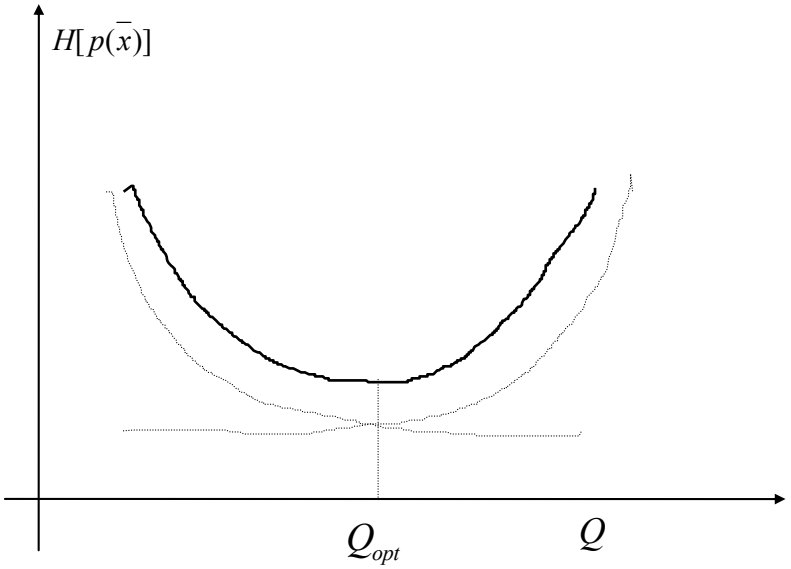


Рисунок 3.15 - Ілюстрація тенденцій, що формують Q_{opt}

Якщо апріорний розподіл $P_0(m,\sigma)$ невідомий, то доцільно використовувати рівномірний розподіл по всій зоні (A,B).

Відмітимо, що розподіл не є гаусовим, але асимптотично збігається до нього (при $n \to \infty$). Середнє значення $\tilde{\varphi}_Б(x)$ дорівнює середньому значенню, визначеному по вибірці, а дисперсія дорівнює вибірковій дисперсії σ^2, помноженій на коефіцієнт $\beta = \dfrac{N^2 - 1}{N(N-3)}$.

Можна показати, що $\tilde{\varphi}_Б(x)$ гарно апроксимується нормальним законом з математичним сподіванням, рівним m, – вибірковому середньому, і дисперсією, рівною вибірковій дисперсії σ^2, помноженій на β. Як видно з формули, β прямує до одиниці при $n \to \infty$, зростає зі зменшенням N і накладає обмеження на об'єм вибірки $N > 3$ (рис. 3.16).

Таким чином, удалося в явній формі виразити залежність параметрів компонентів суміші від об'єму підвибірок, що, у свою чергу, дозволяє реалізувати процедуру пошуку Q_{opt}.

Обсяг підвибірки для *q-ї* компоненти суміші визначається за формулою $N_q = \mu_q N$.

Отже, розглянуто варіант оцінки параметрів суміші. Він не є статистично строго обґрунтованим, але всі обчислювальні процедури опираються на критерії, прийняті в математичній статистиці. Численні практичні додатки апроксимаційного методу в різних предметних галузях показали його ефективність і не суперечать жодному з допущень, викладених у даному розділі.

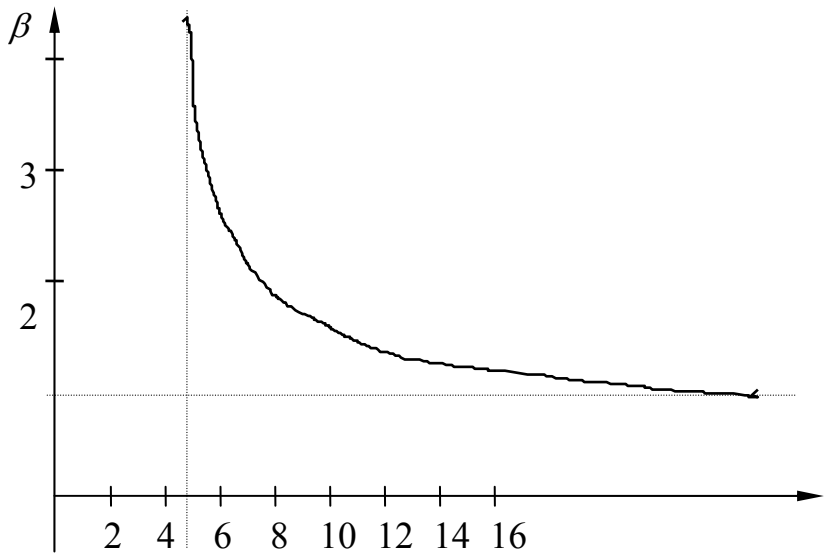


Рисунок 3.16 - Залежність поправкового коефіцієнта β від об'єму вибірки N

3.3.6 Таксономія

Статистичних методів розв'язання задач таксономії є досить багато. Зупинимося тільки на одному, не применшуючи значення або ефективності інших методів.

Він безпосередньо пов'язаний з апроксимаційним методом розпізнавання. Дійсно, відновлення невідомого розподілу за вибіркою у вигляді суміші базових розподілів є, власне кажучи, розв'язанням задачі таксономії з певними вимогами (обмеженнями), пропонованими до опису кожного з таксонів.

Ці вимоги полягають у тому, що значення ознак об'єктів, які входять до одного таксона, мають розподіл імовірностей заданого вигляду. У розглянутому випадку це нормальні або біноміальні розподіли. У ряді випадків це обмеження можна обійти. Зокрема, якщо задано число таксонів Q, а $Q_{opt} > Q$, то деякі таксони варто об'єднати в один, котрий буде мати розподіл значень ознак, який уже відрізняється від нормального або біноміального (рис. 3. 17).

Природно, що поєднувати в один таксон необхідно ті компоненти суміші, що найменше рознесені в ознаковому просторі. Мірою рознесеності компонент може служити, наприклад, міра Кульбака

$$I(i,j) = \int_X \varphi_i(\bar{x}/\bar{a_i}) \log \frac{\varphi_i(\bar{x}/\bar{a_i})}{\varphi_j(\bar{x}/\bar{a_j})} d\bar{x} + \int_X \varphi_j(\bar{x}/\bar{a_j}) \log \frac{\varphi_j(\bar{x}/\bar{a_j})}{\varphi_i(\bar{x}/\bar{a_i})} d\bar{x} =$$

$$= \int_X \left[\varphi_i(\bar{x}/\bar{a_i}) - \varphi_j(\bar{x}/\bar{a_j})\right] \log \frac{\varphi_i(\bar{x}/\bar{a_i})}{\varphi_j(\bar{x}/\bar{a_j})} d\bar{x}.$$

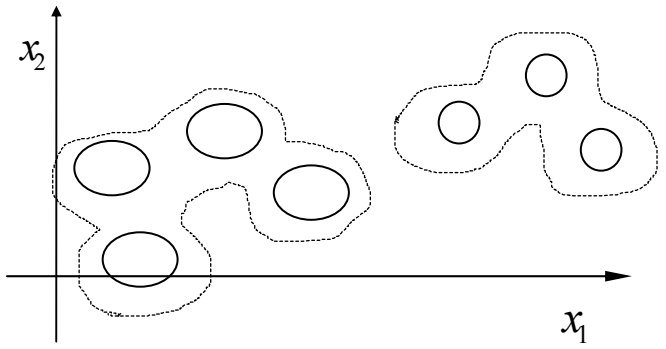


Рисунок 3.17 - Об'єднання компонент суміші в таксони ($Q_{opt} = 7, \quad Q = 2$)

Слід зазначити, що ця міра застосовна лише в тому випадку, коли є підмножина значень X, на якій $\varphi_i(\bar{x}/\bar{a_i}) = 0$, а $\varphi_j(\bar{x}/\bar{a_j}) \neq 0$. Зокрема, ця вимога виконується для нормально розподілених значень ознак компонент суміші.

Якщо говорити про зв'язок викладеного статистичного підходу до таксономії з раніше розглянутими детерміністськими методами, то можна помітити таке [46-49].

Алгоритм ФОРЕЛЬ близький за своєю суттю до апроксимації розподілу суміші нормальних щільностей імовірностей значень ознак, причому матриці коваріацій компонент суміші діагональних елементів цих

матриць рівні між собою, розподіли компонент відрізняються одна від одної тільки векторами середніх значень. Однак на однаковий результат таксономії навіть у цьому випадку можна розраховувати лише при великій рознесеності компонент суміші. Об'єднання декількох сумішей в один таксон за методикою близьке до емпіричного алгоритму KRAB 2. Ці два підходи взаємно доповнюють один одного. Коли вибірка мала і статистичні методи незастосовні або малоефективні, доцільно використовувати алгоритм KRAB, FOREL, KRAB 2. При великому об'ємі вибірки ефективнішими стають статистичні методи, у тому числі об'єднання компонент суміші в таксони.

3.4 Оцінка інформативності ознак

Оцінка інформативності ознак необхідна для їхнього добору при розв'язанні задач розпізнавання. Сама процедура добору практично не залежить від способу виміру інформативності. Важливо лише, щоб цей спосіб був однаковий для всіх ознак (груп ознак), що входять у вихідну їхню множину і, що беруть участь у процедурі добору. Оскільки процедури добору були розглянуті в розділі 3.3, присвяченому детерміністським методам розпізнавання, обговоримо тільки статистичні методи оцінки інформативності [50, 51].

При розв'язанні задач розпізнавання вирішальним критерієм є ризик втрат і як окремий випадок – імовірність помилок розпізнавання. Для використання цього критерію необхідно для кожної ознаки (групи ознак) провести навчання і контроль, що є досить громіздким процесом, особливо при великих обсягах вибірок. Саме це і характерне для статистичних методів. Добре, якщо навчання полягає в побудові розподілів значень ознак для кожного образу $p(\bar{x}/s_i)$. Тоді, якщо нам удалося побудувати $p(\bar{x}/s_i)$ у вихідному ознаковому просторі, розподіл за якою-небудь ознакою (групою ознак) виходить як проекція $p(\bar{x}/s_i)$ на відповідну вісь (у відповідний підпростір) вихідного ознакового простору (маргінальні розподіли). У цьому випадку повторних навчань проводити не потрібно, варто лише оцінити імовірність помилок розпізнавання. Це можна здійснити різними способами. Розглянемо деякі з них.

Якщо є навчальна і контрольна вибірки, то перша з них використовується для побудови $p(\bar{x}/s_i)$, а друга – для оцінки імовірності помилок розпізнавання. Недоліками цього підходу є громіздкість розрахунків, оскільки доводиться велике число раз здійснювати розпізнавання об'єктів, і необхідність наявності двох вибірок: навчальної і контрольної, до кожної з яких висуваються жорсткі вимоги до їх об'ємів. Сформувати на практиці вибірку великого об'єму є, як правило, складною задачею, а дві незалежні вибірки – тим більше.

Можна піти іншим шляхом, а саме: усю вибірку використовувати для навчання (побудови $p(\bar{x}/s_i)$), а контрольну вибірку генерувати датчиком випадкових векторів відповідно до $p(\bar{x}/s_i)$. Такий підхід поліпшує точність побудови $p(\bar{x}/s_i)$ порівняно з попереднім варіантом, але має інші недоліки. Зокрема, крім великого числа актів розпізнавання потрібно згенерувати відповідне число необхідних для цього псевдооб'єктів, що само по собі пов'язане з певними витратами обчислювальних ресурсів, особливо якщо розподіли $p(\bar{x}/s_i)$ мають складний вигляд.

У зв'язку з цим становлять інтерес інші міри інформативності ознак, що обчислюються з меншими витратами обчислювальних ресурсів, ніж оцінка імовірності помилок розпізнавання. Такі міри можуть бути не пов'язані з імовірностями помилок, але для вибору найбільш інформативної підсистеми ознак це не настільки істотно, тому що в даному випадку важливе не абсолютне значення ризику втрат, а порівняльна цінність різних ознак (груп ознак). Зміст критеріїв класифікаційної інформативності, як і при детерміністському підході, складається в кількісній мірі "рознесеності" розподілів значень ознак різних образів. Зокрема, у математичній статистиці використовуються оцінки верхньої помилки класифікації Чернова (для двох класів), пов'язані з нею відстані Бхатачарія, Махаланобіса. Для ілюстрації наведемо вирази відстані Махаланобіса для двох нормальних розподілів, що відрізняються тільки векторами середніх $\overline{m_1}$ і $\overline{m_2}$:

$$d = (\overline{m_1} - \overline{m_2})^T \sum^{-1} (\overline{m_1} - \overline{m_2}),$$

де $\sum$ – матриця коваріацій,

T – транспонування матриці,

-1 – звертання матриці.

В одновимірному випадку $d = \dfrac{(m_1 - m_2)^2}{\sigma^2}$, звідки видно, що d тим більший, чим далі один від одного m_1 й m_2 і компактніший розподіл (менше σ).

Трохи докладніше розглянемо інформаційну міру Кульбака відносно безперервної шкали значень ознак.

Визначимо в такий спосіб середню інформацію в просторі X для розрізнення на користь s_1 проти s_2:

$$I(1{:}2) = \int_X p(\bar{x}/s_1) \log \frac{p(\bar{x}/s_1)}{p(\bar{x}/s_2)} d\bar{x}.$$

При цьому передбачається, що немає зон, де $p(\bar{x}/s_{1\cdot}) = 0$, а $p(\bar{x}/s_2) \neq 0$, і навпаки.

Аналогічно $I(2{:}1) = \int_X p(\bar{x}/s_2)\log\frac{p(\bar{x}/s_2)}{p(\bar{x}/s_1)}d\bar{x}.$

Назвемо розбіжністю величину

$$J(1,2) = I(1{:}2) + I(2{:}1) = \int_X \left[p(\bar{x}/s_1) - p(\bar{x}/s_2)\right]\log\frac{p(\bar{x}/s_1)}{p(\bar{x}/s_2)}d\bar{x}.$$

Чим розбіжність більша, тим вища класифікаційна інформативність ознак.

Очевидно, що при $p(\bar{x}/s_{1'}) \equiv p(\bar{x}/s_2)$ $J(1,2) = 0.$ В інших випадках $J(1,2) > 0.$ Дійсно, якщо $X = X_1 + X_2$, де в зоні X_1 справедливо $p(\bar{x}/s_{1'}) > p(\bar{x}/s_2)$, а в X_2 – $p(\bar{x}/s_{1'}) < p(\bar{x}/s_2)$, то $J(1,2) = J(1,2,X_1) + J(1,2,X_2),$ причому $J(1,2,X_1) > 0$ і $J(1,2,X_2) > 0$.

Легко переконатися, що коли ознаки (ознаки простору) X і Y незалежні, то $J(1,2,X,Y) = J(1,2,X) + +J(1,2,Y).$

Як приклад обчислимо розбіжність двох нормальних одновимірних розподілів з однаковими дисперсіями і різними середніми:

$$p(x/s_1) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-m_1)^2}{2\sigma^2}}, \quad p(x/s_2) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-m_2)^2}{2\sigma^2}}.$$

Виявляється, що в цьому конкретному випадку розбіжність дорівнює відстані Махаланобіса $d = J(1,2) = \frac{(m_1 - m_2)^2}{\sigma^2}$.

3.5 Ієрархічні системи розпізнавання

При розпізнаванні складних об’єктів (слів усного мовлення; назв міст, рік, озер і ін. на географічних картах; складних зображень, що є комбінацією якихось геометричних примітивів) доцільно використовувати ієрархічні процедури розпізнавання. Як випливає із самої назви, особливістю розпізнавальних ієрархічних процедур, є їх багаторівневість. На нижньому рівні розпізнаються елементарні образи (примітиви), на більш високих рівнях – складені образи. Природно, число рівнів може бути різним. Для визначеності будемо розглядати дворівневу систему розпізнавання.

Можна виділити два види дворівневої системи розпізнавання. Перший характеризується наявністю природної тимчасової або просторової послідовності образів першого рівня, що надходять для розпізнавання на другий рівень. Наприклад, це може бути послідовність фонем при розпізнаванні усних слів або послідовність букв при розпізнаванні назв на географічній карті. Тут структурні зв’язки між образами першого рівня гранично спрощені й описуються порядком проходження. В другому виді дворівневих систем розпізнавання структурні зв’язки між образами першого рівня більш складні. Наприклад,

якщо букви розпізнаються дворівневою системою, то образи першого рівня (відрізки прямих, дуг) зв'язані один з одним на площині за деякими правилами, більш складними, ніж просте проходження. Саме цей варіант буде розглянуто пізніше, коли мова піде про лінгвістичні методи розпізнавання.

У даному розділі ми зупинимося на випадку, коли на другу ступінь розпізнавання надходить природна послідовність образів першого рівня.

Отже, на другу ступінь надходить не сам об'єкт, а результати розпізнавання його елементів на першій ступіні $\left\{s_1^1, s_2^1, ..., s_K^1\right\}$. Довжина K таких послідовностей у загальному випадку різна. Алфавіт образів S^2 другого рівня розпізнавання являє собою безліч послідовностей образів першого рівня і може бути істотно більшим за алфавіт образів першого рівня. Наприклад, з 32 букв російського алфавіту можуть складатися десятки тисяч розпізнаваних слів.

Якби розпізнавання образів на першому рівні було безпомилковим, то розпізнавання на другому рівні зводилося б до вибору з алфавіту другого рівня тієї послідовності, що збіглася з послідовністю, отриманою на виході першого рівня розпізнавальної системи. Однак на практиці при розпізнаванні неминучі помилки, у тому числі й в ієрархічних системах, а в останніх – на різних рівнях розпізнавання. Якщо на першому рівні допущені помилки, то на вхід другого рівня може надійти послідовність, що не збігається з жодним з образів, які входять до алфавіту другого рівня, проте якесь рішення приймати необхідно. Можливий, наприклад, такий детерміністський варіант. З алфавіту S^2 вибираються ті послідовності, що містять стільки ж елементів, скільки їх міститься в пропонованій до розпізнавання послідовності. Потім розпізнавана послідовність накладається на відібрані з S^2 послідовності і підраховується число незбіжних елементів. Віднесення до того або іншого образу здійснюється за мінімумом числа незбіжних елементів. Такий підхід є у певній мірі аналогом раніше розглянутого методу мінімуму відстані до еталона, тільки метрики при цьому використовуються різні.

Не треба думати, що помилки розпізнавання, допущені на першому рівні, спотворюють лише той або інший елемент послідовності, не впливаючи на загальне їхнє число. На практиці ж (зокрема, при розпізнаванні усних слів) може спотворюватися і число елементів у послідовності: з'являються зайві помилкові елементи або пропускаються об'єктивно наявні. На цей випадок є досить ефективні алгоритми розпізнавання, реалізовані на другій ступіні ієрархічних систем. Їхнє вивчення виходить за рамки дійсного курсу.

Розглянемо статистичний підхід до розпізнавання в дворівневій ієрархічній системі. Нехай на першому рівні розпізнаються образи s_{ij}^1, де

i – номер позиції в послідовності $\left\{s^1_{1j}, s^1_{2k}, s^1_{3l}, \ldots\right\}$, виданій першою ступінню для розпізнавання на другу ступінь; j – номер конкуруючого образа першої ступіні на i -й позиції в послідовності.

Справа в тому, що в розпізнавальних ієрархічних системах доцільно на проміжних ступінях не приймати остаточне рішення про належність об'єкта до того або іншого образу, а видавати набір варіантів з їх апостеріорними імовірностями. Цей набір повинен бути таким, щоб правильне рішення входило в нього з імовірністю, близькою до одиниці. Якщо всі конкуруючі рішення всіх позицій вважати вершинами графа, увести формально початкову A_0 і кінцеву A_K вершини, з'єднати вершини дугами, як це показано на рисунку 3.18, то виходить орієнтований граф без циклів. Образу s_i^2 другого рівня розпізнавання відповідає певний шлях у графі. Стовщеними дугами позначений шлях, що відповідає послідовності $\left\{s^1_{12}, s^1_{21}, s^1_{31}, s^1_{43}, s^1_{51}, s^1_{63}\right\} = s_i^2$. В четвертому стовпці конкуруючих образів першої ступіні є p^1 – порожня вершина, що відповідає відсутності елементів її стовпця в послідовності. Наявність таких вершин дозволяє видаляти з послідовності "помилкові" елементи, що з'явилися в результаті помилок розпізнавання на першій ступіні. Зрозуміло, при цьому вершині p^1 повинна бути співвіднесена відповідна апостеріорна імовірність.

На побудований граф "накладаються" образи (послідовності) з алфавіту другої ступіні, і той з них, що має максимальну апостеріорну імовірність $p_a^2(s_i^2) = P_0^2(s_i^2)p(s^1_{1j}, s^1_{2k}, s^1_{3l}, \ldots, \overline{/x})$, береться як розв'язок на верхній ступіні.

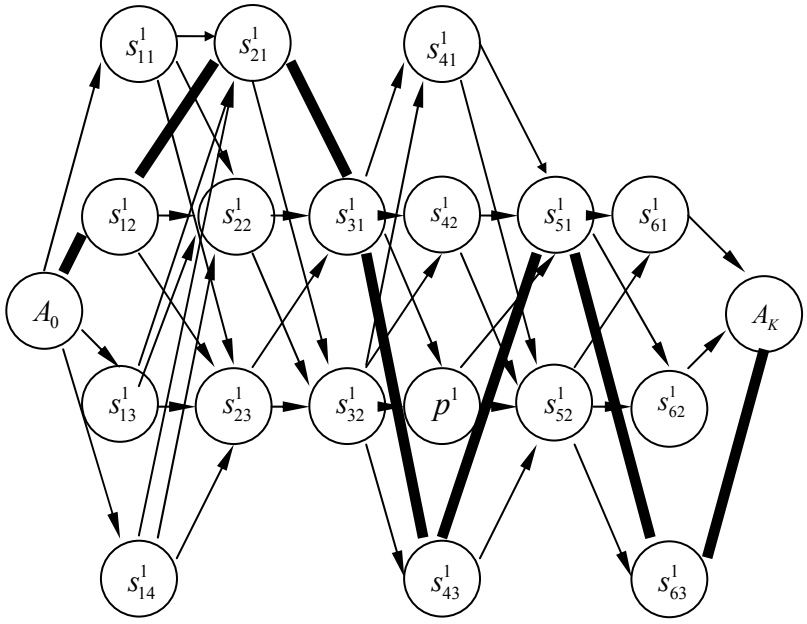


Рисунок 3.18 - Орієнтований граф, що ілюструє процес розпізнавання в двоступінчастій системі

Тут $P_0^2(s_i^2)$ – апріорна імовірність образу $s_i^2 \in S^2$, $\bar{x}$ – вектор параметрів, що характеризують об'єкти на вході першої ступіні розпізнавання.

Якщо говорити про послідовності букв, то кожна з них розпізнається незалежно від інших, тому $p(s_{1j}^1, s_{2k}^1, s_{3l}^1, .../\bar{x}) = p(s_{1j}^1 / \bar{x}) p(s_{2k}^1 / \bar{x}) p(s_{3l}^1 / \bar{x}) ...$

Описана процедура застосовна в тих випадках, коли фіксований алфавіт S_2. При обробці текстів ця вимога найчастіше не виконується. Так, наприклад, виходить при введенні тексту зі сканера і перетворенні зображення сторінки з текстом у текстовий файл. Оскільки тексти, що вводяться, мають довільний зміст, то зафіксувати словник навряд чи можливо, та й немає в тому особливої необхідності. Адже користувачеві потрібна лише послідовність розпізнаних букв, розділових знаків і пропусків. Іншими словами, можна було б обмежитися тільки першою ступінню розпізнавання. Однак результати такого розпізнавання вимагають значного редакторського виправлення, тому що мають місце помилки віднесення вхідних об'єктів до того або іншого образу. Наприклад, при імовірності неправильного розпізнавання 5×10^{-3} на одну машинописну сторінку текстового файлу буде припадати в середньому 10-12 граматичних помилок. Досить жалюгідний рівень грамотності. Його можна підвищити хоча б частковим моделюванням другої ступіні розпізнавання. Наприклад, не маючи фіксованого алфавіту слів, можна зафіксувати алфавіт двобуквених, трибуквених і т.д. послідовностей. Кількість таких послідовностей для кожної мови фіксована, а їхні апріорні імовірності (принаймні для двох букв) слабко залежать від словникового складу. На рис. 3.18 зображено граф, який можна використовувати для двобуквених послідовностей. Використання більшого числа букв веде до істотного ускладнення графа без зміни принципу розрахунків, тому ми обмежимося розглядом двобуквеного варіанта.

Якби система розпізнавала слова, що входять у фіксований алфавіт S^2, то записати апріорну імовірність слова $\{s_{12}^1, s_{21}^1, s_{31}^1, s_{43}^1, s_{51}^1, s_{63}^1\}$ можна було б у такий спосіб:

$$P_0(s_i^2) = p_0(s_{12}^1) p_0(s_{21}^1 / s_{12}^1) p_0(s_{31}^1 / s_{12}^1, s_{21}^1) p_0(s_{43}^1 / s_{12}^1, s_{21}^1, s_{31}^1) \times$$

$$\times p_0(s_{51}^1 / s_{12}^1, s_{21}^1, s_{31}^1, s_{43}^1) p_0(s_{63}^1 / s_{12}^1, s_{21}^1, s_{31}^1, s_{43}^1, s_{51}^1).$$

Якщо обмежитися використанням апріорних імовірностей тільки двобуквеного сполучень, то

$$P_0(s_i^2) \approx p_0(s_{12}^1) p_0(s_{21}^1 / s_{12}^1) p_0(s_{31}^1 / s_{21}^1) p_0(s_{43}^1 / s_{31}^1) p_0(s_{51}^1 / s_{43}^1) \times$$

$$\times p_0(s_{63}^1 / s_{51}^1).$$

Замість добутку можна оперувати сумою, якщо перейти від імовірностей до їх логарифмів.

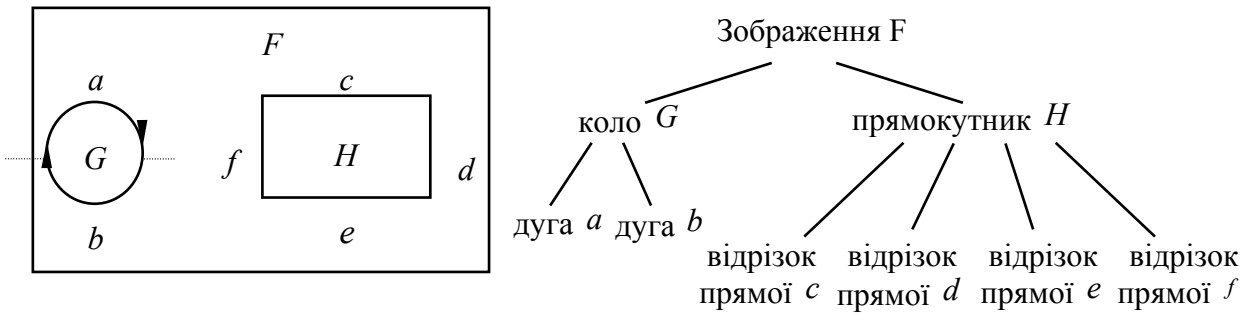
Отже, якщо кожній дузі приписати довжину, рівну $\log\left[p(s_{ij}^1 / s_{i-1,k}^1)p(s_{ij}^1 / \bar{x})\right]$, то найбільш ймовірній послідовності букв буде відповідати шлях максимальної довжини на графі. Цей шлях неважко знайти методами, відомими в теорії графів.

Ефективність такого підходу для виправлення помилок першої ступіні розпізнавання за рахунок мовної надмірності підтверджена практичними випробуваннями. Наприклад, при розпізнаванні послідовності букв розглянутий алгоритм точно знайде, а в більшості випадків і виправить такі помилки, як "голосна –м'який знак", "пропуск – м'який знак", "м'який знак – е" і ряд інших.

3.6 Структурно-лінгвістичні методи

При структурному підході об'єкти описуються не безліччю числових значень ознак $\bar{x}$, а структурою об'єкта. На рисунку 3.19 подано зображення й опис його ієрархічної структури.

Ієрархія припускає опис складних об'єктів за допомогою більш простих підоб'єктів. Ті, у свою чергу, можуть бути описані за допомогою підоб'єктів наступного рівня і т.д. Цей підхід заснований на аналогії між структурою об'єктів і синтаксисом мов. Він прийнятний тоді, коли найпростіші підоб'єкти виділити і розпізнавати легше, ніж зображення (об'єкт) у цілому. Правила композиції найпростіших (непохідних) елементів при описі об'єкта в цілому називають граматикою мови опису об'єктів. Розпізнавання об'єкта полягає в розпізнаванні непохідних його елементів і синтаксичному аналізі (граматичному розборі) "пропозиції", що описує даний об'єкт.



а) б)

Рисунок 3.19 - Зображення (а) і його ієрархічний структурний опис (б)

Перевага лінгвістичного підходу виявляється в тому випадку, що вдається велику кількість складних об'єктів подавати за допомогою невеликого набору непохідних елементів і граматичних правил (наприклад, розпізнавання усних слів за послідовністю фонем). На рис. 3.20 подано приклад опису об'єкта (а) за допомогою операції

композиції "складання ланцюжка" з непохідних елементів (б): $a+a+a+b+b+c+c+c+d+d$.

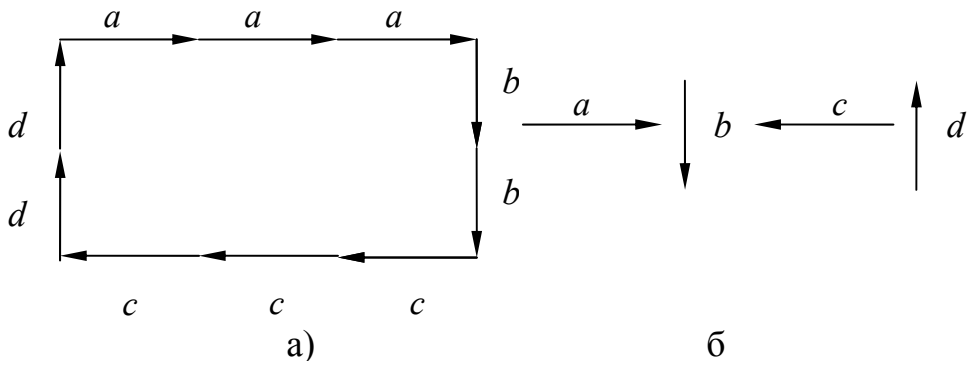


Рисунок 3.20 - Прямокутник (а) і його непохідні елементи (б)

На рисунку 3.21 наведено складніший приклад структурного опису цифри „9”.

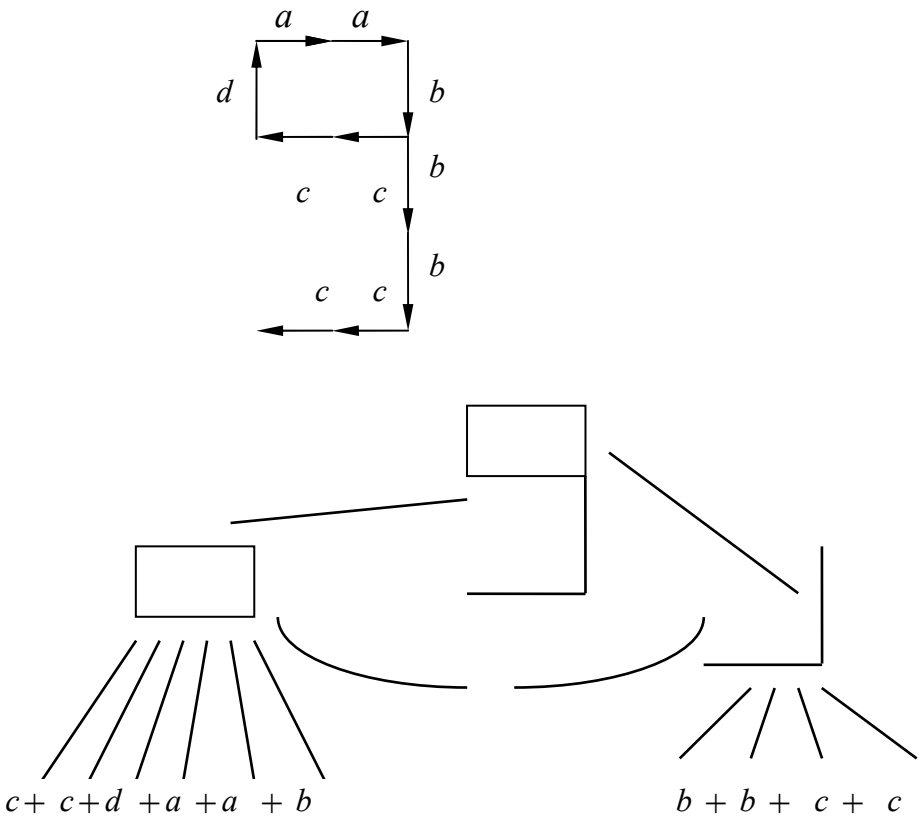
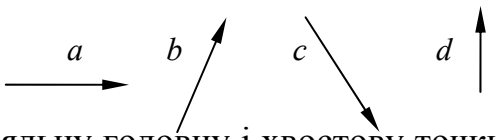


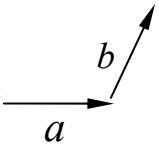
Рисунок 3.21 - Зображення цифри 9 і його структурний опис

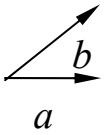
Граматика мови опису об’єктів формується на етапі навчання на основі навчальної вибірки. Теоретичною базою даного підходу є теорія формальних мов і породжувальних граматик, що їхньою основою.

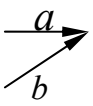
Як приклад наведемо фрагменти мови опису зображень PDL (*Picture Description Language*). Визначено непохідні елементи,

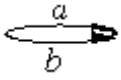


що мають розрізняльну головну і хвостову точки, а також чотири бінарних оператори з'єднання елементів у ланцюжки:

$a + b \Rightarrow$  $\Rightarrow$ головна точка a примикає до хвостової точки b;

$a \times b \Rightarrow$  $\Rightarrow$ хвостова точка a примикає до хвостової точки b;

$a - b \Rightarrow$  $\Rightarrow$ головна точка a примикає до головної точки b;

$a * b \Rightarrow$  $\Rightarrow$ головна точка a примикає до головної точки b і ($\wedge$) хвостова точка a примикає до хвостової точки b.

На рисунку 3.22 наведено вираз мовою PDL, що описує букву A.

Більш докладно цей підхід розглядати не будемо. Відзначимо лише, що найчастіше структурний підхід комбінується з раніше вже розглянутими. Так, усні слова розпізнають за послідовністю фонем на основі структурного методу, а фонеми виділяють і розпізнають у багатовимірному ознаковому просторі X за допомогою тих або інших вирішальних правил.

$(b + (\ (\ (b + c) \times a) + c)\ )$

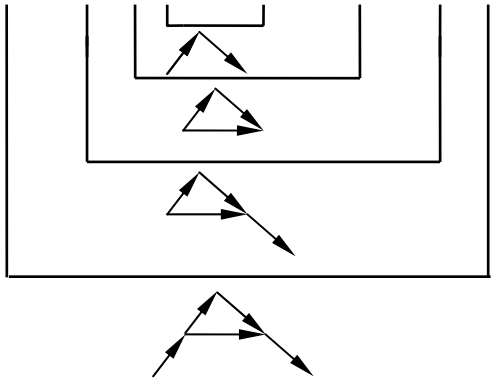


Рисунок 3.22 - Структурний опис літери А мовою PDL

3.6.1 Виділення непохідних елементів

Однозначно визначається нижній рівень декомпозиції вхідного зображення, що визначається заданням цілочислової сітки дискретизації. У цьому випадку елементом, усередину якого вже не проникає опис, є елемент дискретизації, що з'єднаний із сусідніми дискретами зв'язками чотирьох типів:

- вертикальний S_1;
- горизонтальний S_2;
- лівопохилий S_3;
- правопохилий S_4.

Модель зображення при цьому має вигляд

$$MZ=<M,S>,$$

де $M=\{0,1\}$ - множина елементів дискретизації;

S_1, S_2, S_3, S_4 - множина зв'язків, які обумовлюють структуру.

Однак, у більшості випадків, для опису структури реальних об'єктів такий ступінь деталізації є надлишковим. Тому здійснюється "груба" розбивка вхідного образу з використанням як елементів множини *M* сукупностей пов'язаних між собою елементів дискретизації. У найпростішому випадку – це два сусідніх (по вертикалі чи по горизонталі) дискрети. Тоді множина елементів задається сполученням двох сусідніх дискретів у стовпці чи рядку:

$$g0 \rightarrow F(i,j), F(i+1,j) \Rightarrow 00;$$
$$g1 \rightarrow F(i,j), F(i+1,j) \Rightarrow 01;$$
$$g2 \rightarrow F(i,j), F(i+1,j) \Rightarrow 10;$$
$$g3 \rightarrow F(i,j), F(i+1,j) \Rightarrow 11.$$

і містить тільки чотири елементи. Характер зв'язку між виділеними елементами подає інформацію про структуру описуваного зорового образу. Незалежність формування елементів множини *M* призводить до того, що відсутній стохастичний характер їхнього проходження. Тобто, будь-якому елементу множини *M* може передувати будь-який елемент із цієї ж множини. Аналогічне твердження справедливе і для наступних елементів.

Легко помітити, що при використанні як дрібні неподільні частини образу (непохідні елементи) пари сусідніх дискретів для виявлення структурних дрібних утворень – лівих і правих треків, вертикалі і горизонталі – необхідно робити аналіз двох сусідніх пар елементів, тому що структурна інформація вкладена в зв'язках.

Тому логічний перехід на більш високий рівень деталізації [4] вихідних образів - використання як непохідний елемент чотирьох сусідніх елементів дискретизації. При цьому множина *M* буде складатися із

шістнадцяти елементів. Номер кожного з них відповідає двійковому коду станів чотирьох сусідніх елементів дискретизації, причому розряди розташовані в такому порядку:

$$F[i,j],F[i+1,j+1], F[i,j+1], F[i+1,j].$$

Позначивши кожен з можливих станів непохідного елемента деяким символом, можна скористатися їхньою конкатенацією в двох сусідніх рядках дискретного опису зображення як основною операцією попереднього аналізу структури [12]. На рис. 3.23 зображена множина непохідних елементів.

У процесі послідовного перегляду двох сусідніх рядків дискретизованого вхідного зображення по характеру проходження друг за другом виділених елементів формується морфологічний опис частини образа у вигляді двох сусідніх рядків незалежно від складності й апріорної семантики. Оскільки у всьому образі аналізуються пари сусідніх рядків конкатенаційно, те результати такого аналізу дають повний морфологічний опис структури всього образа.

У силу особливостей обраних непохідних елементів і встановленої процедури аналізу спостерігається стохастичний характер проходження четвірок суміжних дискретів образа, що може бути представлений регулярним ланцюгом Маркова. Черговість проходження їх подана в табл. 3.2.

Таблиця 3.2 - Проходження непохідних елементів

Попередній елемент	**Наступні елементи**			
0	**0**	**2**	**4**	**6**
1	**0**	**2**	**4**	**6**
2	**8**	**10**	**12**	**14**
3	**8**	**10**	**12**	**14**
4	**1**	**3**	**5**	**7**
5	**1**	**3**	**5**	**7**
6	**9**	**11**	**13**	**15**
7	**9**	**11**	**13**	**15**
8	**0**	**2**	**4**	**6**
9	**0**	**2**	**4**	**6**
10	**8**	**10**	**12**	**14**
11	**8**	**10**	**12**	**14**
12	**1**	**3**	**5**	**7**
13	**1**	**3**	**5**	**7**
14	**9**	**11**	**13**	**15**
15	**9**	**11**	**13**	**15**

Це дозволяє стверджувати, що при будь-яких припустимих перетвореннях образу, викликаних різними неконтрольованими

зрушеннями його на рецепторній сітківці в процесі дискретизації вихідного зображення, може бути отриманий морфологічний опис з точністю до гомоморфізму. Такий опис дає можливість виявити найпростіші структури - комбінації чотирьох сусідніх елементів дискретизації, що є вихідними частинами більш складних утворень - ліній і різних вузлів [11].

Стартові елементи:

4 S_0

2 S_1

6 S_2

Фінішні елементи:

1 f_0

8 f_1

9 f_2

Горизонталі:

5 h_0

10 h_1

15 h_2

Треки ліві:

3 α_0

7 α_1

11 α_2

Треки праві:

12 β_0

13 β_1

14 β_2

Порожній елемент:

0 z

Рисунок 3.23 - Непохідні елементи у вигляди четвірки сусідніх елементів

3.6.2 Розробка алгоритму формування опису структури

При встановленому порядку перегляду зображення зліва направо, будь-яка геометрична конструкція, що складається з одиничних елементів,

у межах двох сусідніх рядків може починатися тільки одним з таких елементів: *2, 4, 6*, що у силу своєї природи надалі будемо називати стартовими і позначати відповідно як *S1, S0, S2*. Слід зазначити чудовий факт, що стартовий елемент *S1* характерний тільки для правопохилих конструкцій, елемент *S0* - для лівопохилих, і елемент *S2* - для вертикальних. Відповідно до встановленого характеру формування непохідних елементів стартові елементи *S1* можуть бути (див. табл. 3.2) тільки після елементів *0, 1, 8, 9*.

Елемент *0* за домовленістю вважається належним фону і позначається як *Z*. Аналіз показує, що елементи *1, 8, 9* є кінцевими елементами одиничних конструкцій. Причому знову виявляється характерна залежність відповідності цих елементів лініям певної орієнтації: непохідний елемент *1* відповідає правопохилим лініям, елемент *8* – лівопохилим і елемент *9* - вертикальним. Виділені фінішні елементи *1, 8, 9* позначимо відповідно *f0, f1 і f2*.

Горизонтальні лінії характеризуються наявністю непохідних елементів *5* і *10,* що позначаються як *h0* і *h1*, а називаються сполучними. Використовуючи виділені стартові *Si*, фінішні *fi* і сполучні *hi* елементи можна повністю описати характер поведінки контурної лінії зображення будь-якого об'єкта, що еквівалентно опису структури зображення.

Для встановлення взаємозв'язку між іншими непохідними елементами й укладеною в них структурною інформацією припустимо, що на деякому кроці послідовного перегляду двох сусідніх рядків фіксується наявність елемента *S0*. Тоді на наступному кроці, згідно з табл. 3.2, можлива лише одна з чотирьох ситуацій:

- якщо фіксується елемент *f0*, то аналізована конструкція є або випадковою перешкодою, або початком нового структурного елемента;
- якщо фіксується елемент *h0*, то конструкція аналогічна описаній чи вище є ознакою горизонтальної лінії;
- якщо фіксується елемент *3*, то це дає ознаку наявності лівопохилого треку *α0*;
- якщо фіксується елемент *7*, то це дає ознаку наявності лівопохилого треку *α1*.

У випадку, коли на деякому кроці фіксується стартовий елемент *S1*, то на наступному кроці також можлива лише одна з чотирьох ситуацій:

- комбінація *S1*, *f1* дає ознаку закінчення структурного елемента;
- наявність елемента *h1* свідчить про закінчення структурного елемента або є ознакою горизонтальної лінії;
- якщо фіксується елемент *12*, то це дає ознаку наявності правопохилого треку *β1*;

– якщо фіксується наявність елемента *14*, то це також дає ознаку наявності правопохилого треку *β2*.

І, нарешті, якщо конструкція починається стартовим елементом *S2*, то фіксація наступних чотирьох елементів приводить до утворення таких ознак:

– наявність елемента *f2* свідчить про закінчення конструкції типу "вертикаль";
– наявність елемента *15* свідчить про продовження конструкції і позначається як *h2*;
– наявність елемента *11* дає ознаку лівопохилого треку *A2*;
– наявність елемента *13* ознаку правопохилого треку *β1*.

Таким чином, елементи *3, 7, 11* позначені як *α0, α1, α2*, є ознаками лівих треків, а непохідні елементи *12, 13, 14*, позначені як *β0, β1, β2*, характерні для геометричних конструкцій типу "трек правий".

При подальшому наростанні конструкції, на наступних кроках аналізу, формуються більш узагальнені ознаки типу стовщених треків різних вузлів [9]. Товщину треку визначає кількість елементів *h2*, поміщених між парами елементів *α1* і *α2*, *β1* і *β2*. Оскільки в структурному плані конструкції *α1, α2 і α1, h2, h2, h2, α2* несуть однакове значеннєве навантаження, то облік послідовності..., *h2, h2, h2*,... є надлишковим, через те, що в даному випадку ці елементи є просто сполучними між крайніми і не порушують регулярність виділеної конструкції. Аналогічне твердження справедливе і для пари елементів *β1* і *β2*.

При описі в термінах обраних непохідних елементів розглянутих конструкцій звертає на себе увагу факт, що у всіх описах характерним є наявність послідовності (*α0* V *α2*), *h1*(*n*), (*β0* V *β2*), (*n=1,2*...), що відбиває порушення регулярності опису образу.

У випадку конструкцій такого типу будемо позначати як "вузли верхні другого порядку" (*ВВ2Т* і *ВВ2П*) з диференціацією їх на точкові (*Т*) і протяжні (*П*). При досягненні деякого граничного значення кількістю сполучних елементів *h1* вузол вважається протяжним, у іншому випадку – точковим.

У випадку конструкцій, в описі яких характерними є послідовності непохідних елементів (*β0* V *β1*), *h0*(*n*), (*α0* V *α1*), (*n=1,2*...), геометричні конструкції такого типу будемо позначати як "вузли нижні другого порядку" (*ВН2Т* і *ВН2П*) з аналоговою диференціацією на точкові і протяжні.

При наявності в межах опису однієї геометричної конструкції декількох характеристик послідовностей непохідних елементів типу (*α0* V *α2*), *h1*(*n*), (*α2,β2*)*k*, *h1*(*n*), (*β0* V *β2*) чи (*β0* V *β1*), *h0*(*n*), (*α1,β1*)*k*, *h0*(*n*),(*α0* V *α1*) формуються ознаки структурних конструкцій вузлів відповідно верхніх і нижніх порядку *P=K+2*. Залежно від кількості сполучних елементів *h1* і *h0* вони також підрозділяються на точкові і

протяжні (*ВВРТ*, *ВВРП*, *ВНРТ*, *ВНРП*). У межах двох сусідніх рядків зображення можуть бути і більш складні структури, ніж вузли порядку *Р*. Такі структури є послідовними комбінаціями структур верхніх і нижніх вузлів. Встановити які-небудь закономірності в таких структурах важко. Це і дозволяє виділити їх у підклас помилкових вузлів. Необхідно відзначити, що такі структури з'являються, в основному, у результаті різного роду перекручувань і перешкод при одержанні зображення об'єкта чи при фізичних вимірах.

Окремі послідовності елементів ..., *β2,α2,...*і *α1,β1,...* описують структурні елементи, що відносяться до таких ознак, як вузли вироджені верхні (*ВВВ*) та вузли вироджені нижні (*ВВН*).

Як показав проведений аналіз черговості проходження один за одним виділених непохідних елементів, будь-яке бінарне зображення можна подати у вигляді скінченного набору структурних елементів. Тоді в моделі зображення множина буде складатися з таких структурних елементів:

- початок структурного елемента (*П*);
- кінець структурного елемента (*К*);
- трек правий (*ТП*), трек лівий (*ТЛ*);
- вертикаль (*В*), горизонталь (*Г*);
- верхній вироджений (*ВВВ*) і нижній вироджений (*ВВН*) вузли;
- вузли верхні і нижні порядку *Р*(*ВВРТ, ВВРП, ВНРТ, ВНРП*), де *Р=2, 3, 4,...* .

3.6.3 Розробка алгоритму виділення структурних ознак

Існує ряд методів, що дозволяють надійно виділяти зазначені структурні елементи [1,9]. Однак жорсткі вимоги роботи в реальному масштабі часу призводять до непридатності цих алгоритмів у системах ідентифікації. Виходом з цієї ситуації є застосування інших, більш досконалих алгоритмів, заснованих на порядковому методі аналізу зображень.

Запровадимо деякі означення.

- Координатами елемента дискретизованого зображення називається пара *(i,j)(i=1,m; j=1,n)*, де *m* - кількість елементів розкладання по вертикалі; *n* - кількість елементів по горизонталі.
- Два елементи зображення *F(i,j)* і *F(k,l)* називаються сусідніми, якщо виконуються дві умови: *ii-ki<1* і *ii-li<1*.
- Два одиничних елементи зображення називаються зв'язковими, якщо з одного можна потрапити в інший, проходячи тільки від сусіднього елемента до сусіднього.

Відношення зв’язності є відношенням еквівалентності на множині одиничних елементів зображення, отже, останнє розбивається на непересічні класи, що надалі будемо називати зв'язними компонентами

зображення.

•Усяка послідовність одиничних елементів у межах одного рядка називається доріжкою.

•Лівою гранню доріжки називається крайній лівий одиничний елемент доріжки, що граничить з нульовим елементом у даному рядку. Аналогічний одиничний елемент у правому кінці доріжки називається правою гранню.

Неважко переконатися, що інформації про кількість зв'язних доріжок у межах одного структурного елемента і взаємне розташування правих і лівих граней досить для точної класифікації елементів множини *M* [1].

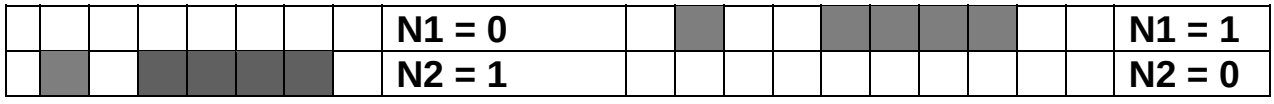
Довжина і положення структурного елемента щодо вертикальної осі визначається шляхом диз'юнкції елементів двох сусідніх рядків *L(j)=F(k,j)VF(k+1,j)*. Координати лівої і правої граней доріжки, що утворилася, визначають початок і кінець елемента.

Кількість доріжок у кожному рядку, у межах одного структурного елемента, визначається шляхом підрахунку числа лівих і правих граней, що виділяються кон'юнкцією сусідніх у рядку елементів:

- $\overline{F(i,j)}\& F(i,j+1)$ для лівих граней;

- $F(i,j)\&\overline{F(i,j+1)}$ для правих граней.

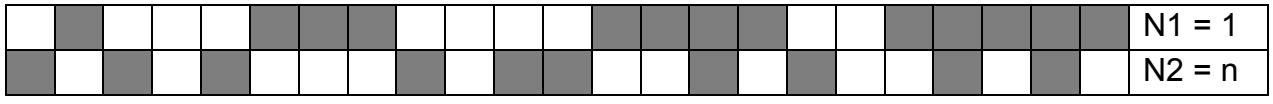
Якщо виділений структурний елемент складається тільки з однієї доріжки в нижньому рядку (*NB=0, NH=1*), то він відноситься до підкласу "початок" (рис. 3.24, а). При фіксації однієї доріжки тільки у верхньому рядку виділений елемент є представником підкласу "кінець" (рис. 3.24, а).



а) б)

Рисунок 3.24 – Приклади структурних ознак „Початок" (а) і „Кінець" (б)

Комбінація *NB=1*, *NH=P* характерна для підкласу елементів "вузол верхній порядку *P*" (рис. 3.25, а). Значення *P* визначається кількістю доріжок у нижньому рядку. Для елементів типу "вузол нижній порядку *P*" характерна комбінація *NB=P*, *NH=1* (рис. 3.25, б).



а)

б)

Рисунок 3.25 – Приклади структурних ознак „Вузол верхній" (а) і „Вузол нижній" (б)

При *NB>1* і *NH>1* виділений елемент є помилковим вузлом. Диференціація вузлів на точкові і протяжні здійснюється по довжині нульових доріжок, розташованих між двома одиничними.

Для інших структурних елементів *NB=1* і *NH=1*. Тому для їхньої класифікації необхідна додаткова інформація про взаємне розташування доріжок у сусідніх рядках. Елементи типу "трек лівий" характеризуються тим, що в них верхня доріжка зміщена вправо відносно нижньої. Цей факт реєструється одиничним значенням предиката

$$Q = \overline{F(k,n)} \vee \overline{F(k+1,m)}.$$

В елементах "трек правий" верхня доріжка зміщена вліво відносно нижньої, що визначається одиничним значенням предиката

$$Q = \overline{F(k+1,n)} \vee \overline{F(k,m)}.$$

Елемент типу "вузол вироджений верхній" характеризується наявністю нижньої доріжки, грані якої розташовані між гранями верхньої доріжки, і реєструється одиничним значенням предиката

$$Q = \overline{F(k+1,n)} \,\&\, \overline{F(k+1,m)}.$$

Відмінною рисою нижніх вироджених вузлів є розташування граней верхньої доріжки між гранями нижньої. Логічне виділення цих елементів проводиться за одиничним значенням предиката

$$Q = \overline{F(k,n)} \,\&\, \overline{F(k,m)}\,.$$

Структурні елементи "вертикаль" і "горизонталь" характеризуються повним збігом доріжки L з доріжками у верхньому і нижньому рядках, що реєструється одиничним значенням предиката

$$Q = F(k,n) \& F(k+1,n) \& F(k,m) \& F(k+1,m).$$

Слід зазначити, що перевищення довжиною доріжки *L* деякого граничного значення визначаємо його характером зображення, елементи "вертикаль", "трек лівий", трек правий", "вузол вироджений верхній" і "вузол вироджений правий" трансформуються в елемент "горизонталь".

Розроблений алгоритм виділення структурних ознак подано на рис. 3.26.

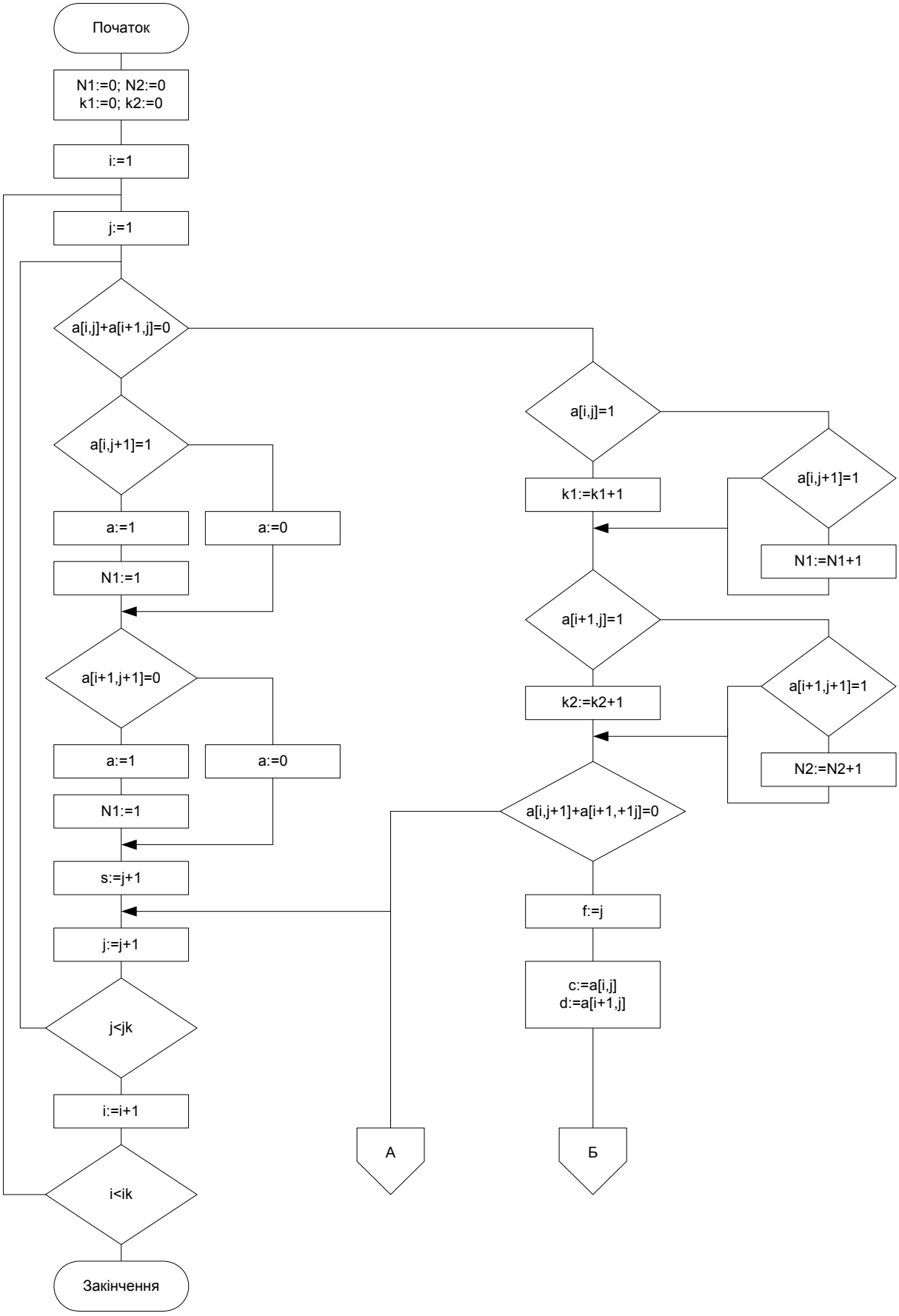


Рисунок 3.26 – Алгоритм виділення структурних ознак

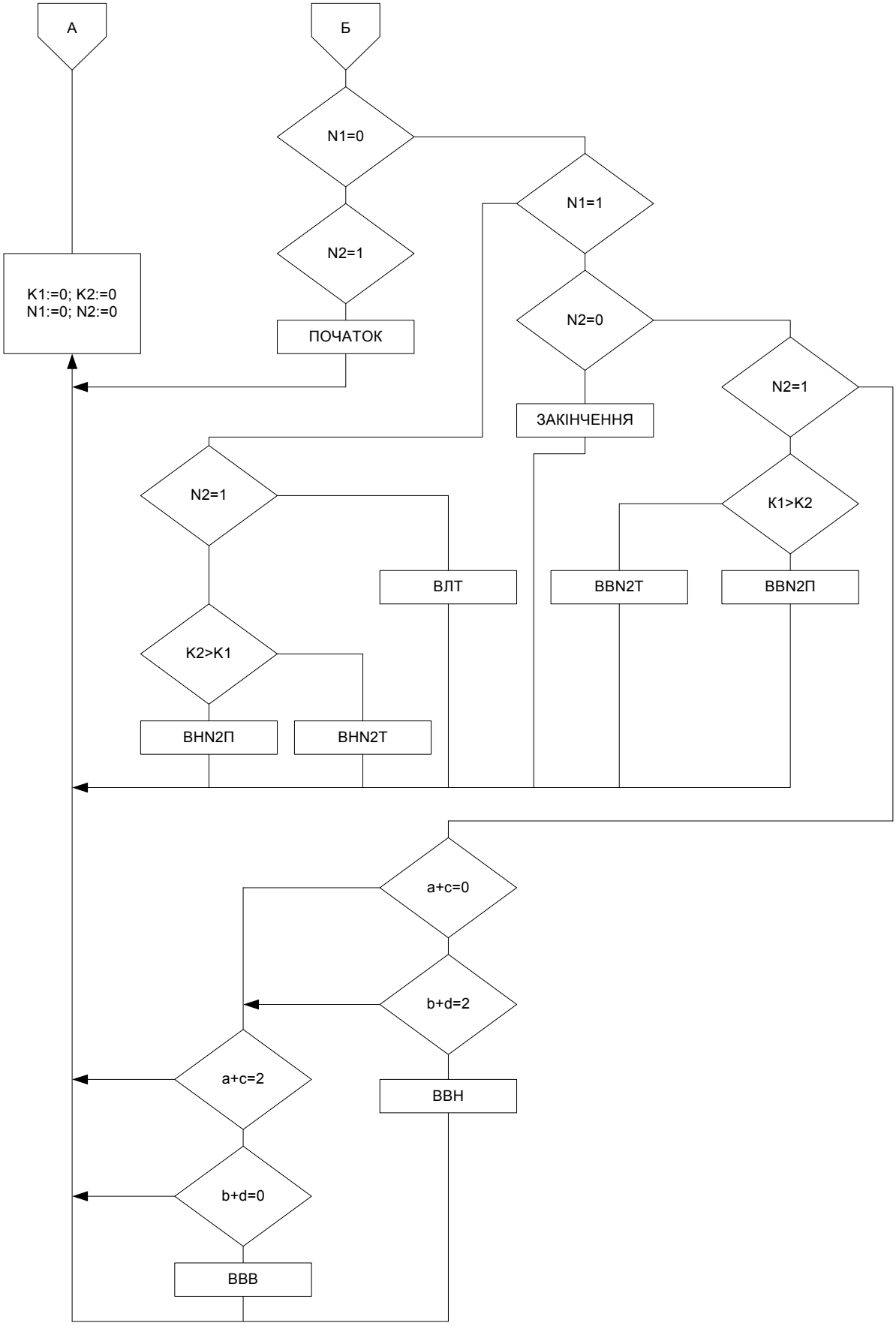


Рисунок 3.26 – аркуш 2

При використанні запропонованої процедури порядкового аналізу для формування структурного опису конфігурацій необхідно виконувати логічний аналіз черговості проходження один за одним виділених елементів у сусідніх рядках. Наприклад, послідовність *Н, ТЛ, ТЛ, ТЛ, ТП, ТП, К* характерна для елементів, виділених у підклас "вузол лівий", а послідовність *Н, ТП, ТП, ТЛ, ТЛ, ТЛ, К* - для елементів "вузол правий". Виділення цих елементів шляхом аналізу вже сформованого структурного опису призводить до додаткових часових (час аналізу опису) і апаратурних (організація аналізу) витрат, неприйнятних для ефективних систем обробки відеоінформації.

Тому логічно організувати другий аналогічний канал порядкового формування опису з тією лише різницею, що сканування зображення в ньому здійснюється не по горизонталі, як у першому, а по вертикалі. При паралельній організації двох каналів за час сканування зображення формується повний структурний опис, незалежно від його характеру і складності.

Оскільки процес виділення структурних елементів здійснюється шляхом аналізу двох сусідніх рядків з накладенням, то за інформацією в поточному рядку опису можна з деякою вірогідністю передбачати структуру образу в наступному рядку. Цю надмірність можна використовувати для підвищення точності опису перекручених перешкодами зображень.

Для цього необхідно досліджувати порядок проходження один за одним структурних елементів у межах трьох сусідніх рядків. При аналізі опису зображення звертає на себе увагу порушення монотонності опису "треків лівих" через похибки дискретизації (*ТЛ*, *В*, *ТЛ*) і дії перешкод (*ТЛ, ТЛ, ТП, ТЛ*). Аналіз елементів, що оточують місце порушення монотонності опису, показує, що воно викликано не структурою образу, а дією перешкод. Отже, в отриманому описі помилково отриманий елемент "трек правий" і "вертикаль" необхідно замінити на елемент "трек лівий".

Попередність елементів "початок" вузлам нижнім і післяслідування елементів "кінець" за вузлами верхніми також є наслідком появи на зображенні помилкових нулів і одиниць. У цьому випадку корекція здійснюється шляхом зменшення порядку вузла згідно з формулою

$$Pk=Pн-k,$$

де Pk - порядок вузла після корекції;

$Pн$ - порядок вузла до корекції;

k - кількість елементів "початок" чи "кінець".

Після проведення зазначеної своєрідної структурної корекції отриманого опису вхідного зображення з метою скорочення потоку для подальшої обробки здійснюється заміна монотонної послідовності однойменних елементів одним відповідним елементом.

Зроблений аналіз основних властивостей непохідних елементів у вигляді четвірки суміжних елементів дискретизації привів до виділення кінцевої множини структурних елементів, у термінах яких описується зображення будь-якої складності.

Запропонована модель бінарного зображення є універсальною: з неї виводяться відомі моделі структурного опису образів. Завдяки цьому алгоритми опису різних об'єктів, розроблені в рамках окремих моделей, можуть бути використані при розробці алгоритмів, описаних на запропонованій моделі. Таке визначення моделі є і конструктивним: оперуючи даними просторових вимірів (яскравості) можна виділити гіпотетичні структурні елементи опису зображень.

Питання для самоконтролю

1. Поясніть практичне значення проблеми розпізнавання образів.
2. Наведіть принципові відмінності розпізнавання машинописних і рукописних текстів.
3. У чому полягає принцип "не нашкодь"?
4. Які рівні розпізнавання застосовуються у системі ABBYY FormReader?
5. Назвіть основні поняття, що застосовуються при визначенні параметрів, ознак, характеристик об’єктів розпізнавання.
6. В чому суть статистичних і дискримінантних методів розпізнавання?
7. У чому полягає метод еталонів, що дробляться?
8. У чому полягає ідея методу найближчих сусідів?
9. Поясніть правило найближчого сусіда.
10. Як виглядає функція правдоподібності, уведена Фішером?
11. У яких випадках використовувати байєсовське вирішальне правило неможливо?
12. Який метод розпізнавання за своїми властивостями є досить універсальним?
13. Як можна здійснити оцінку імовірності помилок розпізнавання?
14. Які процедури застосовуються при розпізнаванні складних об’єктів?
15. Назвіть переваги лінгвістичного підходу до розпізнавання.
16. Як називають особливість людини відносити незнайомий або знайомий об’єкт до певної групи об’єктів (класу)?
17. Які основні і додаткові етапи виділяються при розв’язанні задач, ядро яких складають процеси, пов’язані з класифікацією?
18. Що таке нейрон?
19. Поясніть загальну фізіологічну модель сприйняття.
20. Які на Ваш погляд основні задачі розпізнавання?

4 СИСТЕМИ РОЗПІЗНАВАННЯ МОВИ

Із розвитком комп'ютерних систем, стає усе більш очевидним, що використання цих систем набагато розшириться, якщо стане можливим використання людської мови при роботі безпосередньо з комп'ютером, і зокрема стане можливим керування машиною звичайним голосом у реальному часі, а також введення і виведення інформації у вигляді звичайної людської мови.

4.1 Використання характеристик мовного сигналу для розпізнавання

Існуючі сьогодні системи розпізнавання мови ґрунтуються на зборі всієї доступної (часом навіть надлишкової) інформації, необхідної для розпізнавання слів. Дослідники вважають, що в такий спосіб задача розпізнавання зразка мови, заснована на якості сигналу, підданого змінам, буде достатньою для розпізнавання, однак у даний час навіть при розпізнаванні невеликих повідомлень нормальної мови поки неможливо після одержання різноманітних реальних сигналів здійснити пряму трансформацію в лінгвістичні символи, що є бажаним результатом.

Замість цього проводиться процес, першим кроком якого є первісне трансформування інформації, що вводиться, для скорочення оброблюваного обсягу так, щоб її можна було піддати комп'ютерному аналізу. Прикладом є "техніка зіставлення відрізків", що дозволяє скоротити інформацію, що вводиться, з 50 000 до 800 бітів на секунду. Наступним етапом є спектральне подання мови, що виконується шляхом перетворення Фур'є. Результат перетворення Фур'є дозволяє не тільки стиснути інформацію, але і дає можливість сконцентруватися на важливих аспектах мови, що інтенсивно вивчалися в сфері експериментальної фонетики.

Хоча спектральне подання мови дуже корисне, необхідно пам’ятати, що досліджуваний сигнал досить різноманітний. Різноманітність виникає з багатьох причин, включаючи:

- розходження людських голосів;
- гучність мови;
- варіації у вимові;
- нормальне варіювання руху артикуляторів (мови, губ, щелеп).

Для усунення негативного ефекту впливу варіювання голосового тракту на процес розпізнавання мови було використано безліч методів. По-перше розглядалася характеристика простору траєкторії артикуляторних органів, включаючи голосні. Найбільш вдалі форми трансформації, використаної для скорочення розходжень, були вперше представлені Сакоя та Чибо і називалися динамічними перекручуваннями (*dynamic time warping*). Техніка динамічного перекручування використовується для тимчасового витягування і скорочення відстані між перекрученим

спектральним поданням і шаблоном того, що говорить. Використання даної техніки дало поліпшення точного розпізнавання (~20-30%). Метод динамічного перекручування використовують практично всі комерційно доступні системи розпізнавання, що показують високу точність повідомлення при використанні. Техніка динамічного перекручування подана на рисунку 4.1.

Спочатку сигнал перетворюється в спектральне подання, де визначається нечисленний, але високоінформативний набір параметрів. Потім визначаються кінцеві вихідні параметри для варіювання голосу (слід зазначити, що дана задача не є тривіальною) і виробляється нормалізація для складання шкали параметрів, а також для визначення ситуаційного рівня мови. Вищеописані змінені параметри використовуються потім для створення шаблону. Шаблон заноситься в словник, що характеризує проголошення звуків при передачі голосової інформації. Далі в процесі розпізнавання нових мовних зразків (що уже піддавались нормалізації своїх параметрів), ці зразки порівнюються, використовуючи динамічне перекручування, із зразками, що вже є в словнику і мають схожі метричні параметри. В даний час цей метод вивчається і доповнюється.

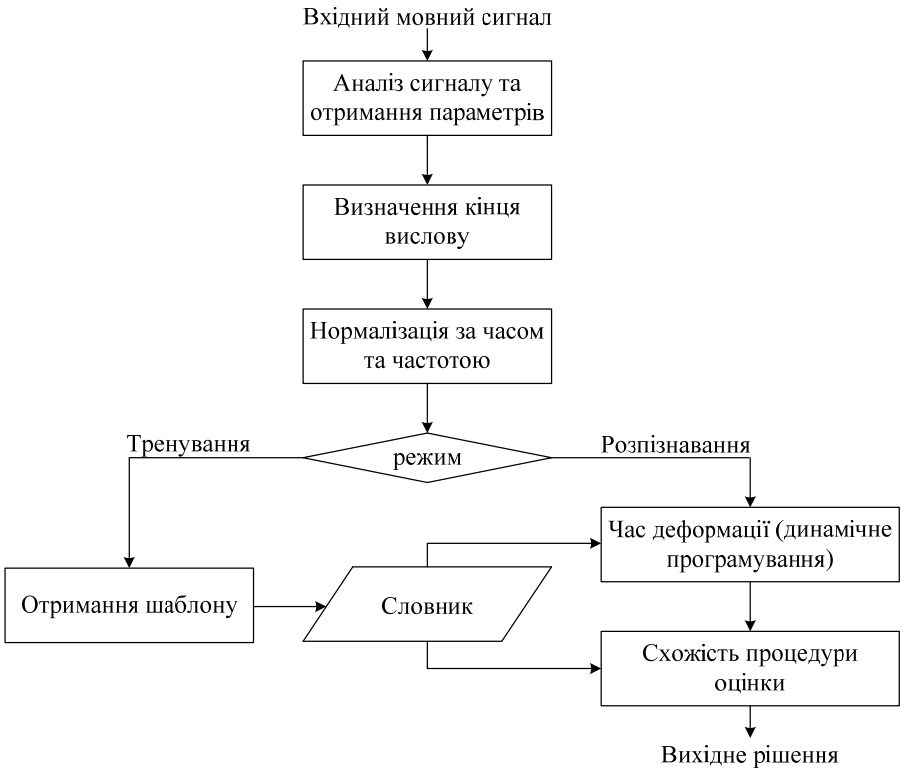


Рисунок 4.1 – Алгоритм динамічного перекручування

Очевидно, що спектральне подання мови дозволяє характеризувати особливості голосового тракту людини і спосіб використання його для розмови. Самий звичайний спосіб моделювання специфічних ефектів "модель-джерело" - використання фільтрів. Мовний апарат моделюється з

використанням джерел, що викликають резонанс, який веде до пікових точок інтенсивності звуку в сусідстві з окремими частотами, називаними формантами. При проголошенні звуків вібрація голосових зв'язок є джерелом порушення, і ці короткі імпульси викликають резонанс між голосовими зв'язками і губами. Оскільки при промовлянні щелепи, губи, зуби й альвеолярний апарат рухаються, розмір і місце цих резонансів змінюються, даючи можливість відтворення особливих параметрів звуків.

Можливо побудувати дуже точну модель, також прямо змоделювати рух артикуляторів фізіологічно реальним шляхом. Використання цих моделей привело до розуміння шляху, по якому проходить мовний сигнал. Але оскільки спостереження над артикуляторами є складними, наявні недоліки. Хоча природа вокального тракту значно впливає на вихідний сигнал мови, це не єдине обмеження, яке необхідно брати до уваги, тому що контроль над мускулами звукового тракту обумовлений сигналами моторного кортекса мозку. Можливо всі аспекти впливу акустичної структури контролюють сигнали і форму звукового виходу мови (хоча це не може бути доведено із систематичної точки зору).

Аспекти впливу акустичної структури містять у собі:

- природу сегментів індивідуального звуку (голосні/приголосні);
- структуру складу;
- структуру морфем (префікси, корені, суфікси);
- лексикон;
- рівень синтаксису фраз і пропозицій;
- довгострокові обмеження мови (*long-term discourse constraints*).

Розглядається вплив обмежень і спосіб їхнього впливу на відтворення сигналу мови. Необхідно також взяти до уваги той факт, що людський апарат сприйняття також повинен бути змодельований, він сам по собі накладає на процес сприйняття додаткові обмеження. Недавно процес сприйняття був вивчений за допомогою методу сигнального придушення барабанних перетинок через порушення нервових клітин, що утворюють приблизно 30 тисяч нервових закінчень слухового нерва. Але вивчення нервових закінчень здатне тільки прояснити формування простих синтетичних голосних. Перед дослідниками постав новий головний напрямок в галузі вивчення процесу відтворення мови, пов’язаний з інтеграцією усієї фізіології сприйняття людини. В даний момент з’являються деякі моделі явищ, які відбуваються в вусі, і не без підстав можна чекати подальшого поліпшення розуміння процесу розпізнавання мови через більш повне розуміння характеристик цього впливу.

Що стосується рівня артикуляторного контролю, першим рівнем є індивідуальний фонетичний сегмент, інакше кажучи, - фонема. У багатьох природних мовах їх приблизно 40. Але їхній набір суттєво відрізняється. Тому, наприклад, англійські голосні можуть бути носовими, навіть

ненавмисно, у той час як у французькому нормалізація голосних є фонетичним контрастом, і тому впливає на значення вимовленого. У французькій мові носова коартикуляція домінує в голосних, істотно впливає на сприйняття фонем і, отже, має головний зміст значення. Хоча всі, хто говорять, мають однаковий голосовий апарат, використання його різне. Так наприклад, використання кінчика язика або клацання, як у деяких африканських мовах. Ясно, що природа артикуляційних рухів має сильний вплив на метод відтворення мови. Ці обмеження завжди активно використовуються в практичних системах.

На наступному рівні лінгвістичної структури фонетичні сегменти згруповані в приголосні/голосні, а отже й у склади. Далі, залежно від ролі фонетичного сегмента усередині цих складів, їхня реалізація може бути сильно змінена. Так наприклад, початковий приголосний у складі може бути реалізований як абсолютно відмінний від кінцевої позиції. Приголосні дуже міцно пов'язуються між собою, що знову ж впливає на наступні обмеження. Наприклад, в англійській, якщо початкова група приголосних складається з трьох фонем, перша фонема повинна бути /*s*/, наступною фонемою повинен бути невимовний приголосний, третьою - або /*r*/ або /*l*/, як наприклад, у слові /*scrape*/ або /*split*/.

Люди, які розмовляють рідною мовою, уникають цих обмежень або можуть активно їх використовувати під час процесу сприйняття. З вище-наведених прикладів очевидно, що хоча й існують жорсткі обмеження, які впливають на слухача, але їхня сила не є вирішальною під час проголошення промови. Тобто будь-яке моделювання процесу сприйняття може бути активним і може надати велику допомогу в розумінні головного змісту.

Інший приклад, що показує необхідність застосування сфальцьованого пошуку, може бути поданий у сприйнятті кінцевого приголосного. Серед багатьох ключових слів для розпізнавання кінцевого приголосного існує спектральна природа шуму, відтвореного при звільненні кінцевої перемички і переходу резонансу другої форманти в голосний, що випливає за цією перемичкою. Багато дослідників вивчали ці впливи, і результати їхніх досліджень показали, що обмежуючий вплив обох вищеописаних характеристик на сприйняття варіюється природою наступного голосного, і отже, могутня стратегія розпізнавання повинна мати деякі знання про тверду позицію голосного перед кінцевим приголосним до того, як буде зроблено саме розпізнавання кінцевого приголосного. Кінцеві приголосні дають яскравий приклад досить цікавого комплексу фонетики, використовуваного для лінгвістичного забарвлення. Наприклад, при розгляді слів *rapid* і *rabid* виявляється 16 фонетичних розходжень.

Крім сегментного і складового рівнів існують обмежені впливи через структуру морфем, що є мінімальними синтаксичними одиницями мови.

Вони містять у собі додатки, корені, суфікси. Можна собі уявити, що це синтаксис на складовому і на морфемному рівнях, також як і нормально розпізнаний синтаксис, що характеризується способом, у якому англійські слова поєднуються у фрази і пропозиції. Можна подати дані обмеження як наслідки розгляду граматики поза контекстом. У цьому виді обмежень багато "гучних" варіацій сегментів мови, що так само відносяться і до ієрархічних синтаксичних обмежень.

Додаткові обмеження на природу входження нової лексики в мову можуть бути рівнем слова. Багато досліджень знайшли, що характеристика слів при введенні розбивки на 5 твердих класів фонетичних сегментів може бути скорочена до мінімуму, часто маючи єдине у своєму роді розпізнавання. Далі значно підсилюється ефект порядку двох букв і фонетичних сегментів з тих пір, як у вивченні англійських і французьких словників було виявлено, що більше 90% слів мали єдине значення і тільки 0,5% мали дві і більше альтернатив. На фонемному рівні було виявлено, що всі слова в англійському словнику з 20 тисяч слів мали одне значення через безладні фонемні пари. Цей приклад допомагає показати, що усе ще існує обмежуючий вплив на лексичному рівні, що ще не визначено в сучасних системах розпізнавання мови. Природно, що дослідження в цій галузі продовжуються.

Крім рівня слів синтаксис має додатковий обмежувальний вплив. Його вплив на послідовний порядок слів часто характеризується в системах фактором, що у свою чергу характеризує кількість можливих слів, які можуть випливати за попереднім словом у процесі проголошення. Синтаксис також має обмежувальні впливи на просодичні елементи, такі як наголос, наприклад у випадку, коли наголос слів у *incline* і *survey* варіюється залежно від частини мови. Можливо для того, щоб охарактеризувати наголос у слові, потрібно взяти до уваги не тільки індивідуальне слово, але й вищенаведені додаткові обмеження синтаксису.

Далі, крім синтаксичного рівня обмеження домінують над семантикою, прагматикою і мовою, що погано усвідомлюється людьми, однак має дуже важливе значення для процесу розпізнавання.

Незважаючи на складність опису характеристик джерел різних обмежень, немаловажну роль відіграють сучасні системи впливу, що представлені всіма можливими варіантами проголошення звуків. Наприклад, система HARPI університету Carnegie-Mellon University є системою, у якій відтворення звуку описується як шлях через комплексну мережу. У цьому способі обмеження структури складу, слова і синтаксису пов'язані однією структурою. Структура контролю, використовувана для пошуку, є адаптацією динамічної програмної техніки. Кращий підхід був запропонований моделями використання ланцюгів Маркова. Ці моделі використовувалися як єдина структура, де можливості можуть бути точно вивчені експериментальним шляхом.

Закодовані подання спектральної трансформації відтворення мови використовуються для побудови найправильнішого шляху через мережу, і нещодавно були отримані дуже гарні результати. Дуже важливо підкреслити використання такого формально - структурного підходу, що сприяє автоматичному визначенню класів символів через структурування і параметризацію.

При іншому підході бази даних і пов'язані з ними процеси обробки використовуються контроль структури. Цей підхід був вивчений системою HEARSAJ 2, що була розроблена в інституті Carnegie-Mellon University, і системою HWIM (*hear what I mean*). У цих системах комплексна структура даних, що містить всю інформацію про відтворення звуків, вивчається з погляду конкретних обмежень. Але як вище зазначено, кожне з цих обмежень має особливу внутрішню модель, і повний аналіз не може бути зроблений. Для проведення аналізу в цілому структура даних повинна мати взаємодію між різними процесами, а також засоби для інтеграції. Незважаючи на те, що структура містить у собі достатньо різних джерел знань і її внесок у розуміння мови узагальнений, вона також має велику кількість ступенів свободи, які можуть бути використані для ретельного системного відтворення.

На відміну від цього, техніка, заснована на ланцюгах Маркова, має математичну підтримку. Щоб мати можливість сфальцьованого дослідження обмежень взаємодії й інтеграції в контексті, необхідно застосовувати обидві системи. Ті системи, що описують обмеження взаємодії, сфальцьовані багато в чому на відтворенні знань, і вони відносно слабоконтрольовані, а системам з математичною підтримкою, що у свою чергу мають чудову техніку для встановлення параметрів і оптимізації вивчення, не вистачає використання комплексної структури даних, необхідних для характеристики обмежень високого рівня, таких як синтаксис. Обидва напрямки в даний момент знаходяться в процесі розвитку.

На закінчення необхідно зробити акцент на вплив виробничої технології на ці системи. Технологія інтеграції не є великою проблемою для систем розпізнавання мови, навпаки, це є архітектурою цих систем, включаючи спосіб подання обмежень. Необхідно провести грандіозні експерименти і знайти нові способи, які необхідні для обмежувального впливу взаємодії.

4.2 Мова як об'єкт аналізу

Забезпечення взаємодії з ЕОМ природною мовою (ПМ) є найважливішою задачею досліджень в галузі штучного інтелекту. Бази даних, пакети прикладних програм і експертні системи, засновані на ШІ, вимагають оснащення їх гнучким інтерфейсом для численних користувачів, що не бажають спілкуватися з комп'ютером штучною

мовою. У той час як багато фундаментальних проблем в галузі обробки ПМ (*Natural Language Processing, NLP*) ще не вирішені, прикладні системи можуть оснащуватися інтерфейсом, який розуміє ПМ при певних обмеженнях.

Існують два види і дві концепції обробки природної мови:

- для окремих голосових команд;
- для ведення інтерактивного діалогу.

Обробка природної мови - це формулювання і дослідження комп'ютерно-ефективних механізмів для забезпечення комунікації з ЕОМ природною мовою. Об'єктами досліджень є:

- власне природні мови;
- використання ПМ як у комунікації між людьми, так і в комунікації людини з ЕОМ.

Задача досліджень - створення комп'ютерно-ефективних моделей комунікації природною мовою. Саме така постановка задачі відрізняє NLP від задач традиційної лінгвістики й інших дисциплін, що вивчають ПМ, і дозволяє віднести її до галузі ШІ. Проблемою NLP займаються дві дисципліни: *лінгвістика* і *когнітивна психологія*.

Традиційно лінгвісти займалися створенням формальних, загальних, структурних моделей ПМ, і тому віддавали перевагу тим з них, що дозволяли витягати найбільше мовних закономірностей і робити узагальнення. Практично ніякої уваги не приділялося питанню про придатність моделей з погляду комп'ютерної ефективності їхнього застосування. Таким чином, виявилося, що лінгвістичні моделі, характеризуючи власне мову, не розглядали механізми її породження і розпізнавання. Гарним прикладом тому служить породжуюча граматика Хомського, що виявилася абсолютно непридатною на практиці як основа для комп'ютерного розпізнавання ПМ.

Задачею ж когнітивної психології є моделювання не структури мови, а її використання. Фахівці в цій галузі також не надавали великого значення питанню комп'ютерної ефективності.

Розрізняються *загальна* і *прикладна* NLP. Задачею загальної NLP є розробка моделей використання мови людиною, що є при цьому комп'ютерно-ефективною. Основою для цього є загальне розуміння текстів, як розглядається в роботах Чарняка, Шенка, Карбонелла й ін. Безсумнівно, загальна NLP вимагає величезних знань про реальний світ, і велика частина робіт зосереджена на поданні таких знань і їхньому застосуванні при розпізнаванні повідомлення, що надходить, ПМ. На сьогоднішній день ШІ ще не досяг того рівня розвитку, коли для розв'язання подібних задач у великому обсязі використовувалися б знання про реальний світ, і існуючі системи можна називати лише експериментальними, оскільки вони працюють з обмеженою кількістю ретельно відібраних шаблонів ПМ.

Прикладна NLP займається звичайно не моделюванням, а безпосередньо можливістю комунікації людини з ЕОМ ПМ. У цьому випадку не так важливо, як введена фраза буде зрозуміла з погляду знань про реальний світ, а важливий витяг інформації про те, чим і як ЕОМ може бути корисною користувачеві (прикладом може служити інтерфейс експертних систем). Крім розуміння ПМ, у таких системах важливо також і розпізнавання помилок і їхнє корегування.

Основною проблемою NLP є мовна неоднозначність. Існують різні види неоднозначності:

- *синтаксична* (структурна) неоднозначність: у фразі *Time flies like an arrow* для ЕОМ неясно, чи йде мова про час, що летить, чи про комах, тобто чи є слово *flies* дієсловом або іменником;
- *значеннєва* неоднозначність: у фразі *The man went to the bank to get some money and jumped in* слово *bank* може означати як *банк*, так і *берег*;
- *відмінкова* неоднозначність: прийменник *in* у реченнях *He ran the mile in four minutes/He ran the mile in the Olympics* позначає або час, або місце, тобто подано зовсім різні відношення;
- *референціальна* неоднозначність: для системи, що не має знань про реальний світ, буде важко визначити, з яким словом - *table* або *cake* - співвідноситься займенник *it* у фразі *I took the cake from the table and ate it*;
- *літерація* (Literalness): у діалозі *Can you open the door? — I feel cold* і прохання, і відповідь виражені нестандартним способом. В інших обставинах на питання може бути отримана пряма відповідь *yes/no*, але в даному випадку в питанні імпліцитно виражене прохання відкрити двері.

Центральна проблема як для загальної, так і для прикладної NLP – розв’язання такого роду неоднозначностей - вирішується за допомогою перекладу зовнішнього подання ПМ в якусь внутрішню структуру. Для загальної NLP таке перетворення вимагає набору знань про реальний світ. Так, для аналізу фрази *Jack took the bread from the supermarket shelf, paid for it, and left* і для коректної відповіді на такі питання, як *What did Jack pay for?*, *What did Jack leave?* і *Did Jack have the bread with him when he left?* необхідні знання про супермаркети, процеси купівлі, продажу і деякі інші.

Прикладні системи NLP мають перевагу перед загальними, тому що працюють у вузьких предметних галузях. Наприклад, в системі, використовуваній продавцями в магазинах із продажу комп'ютерів, не потрібно ”роздумувати” над неоднозначністю слова *terminals* у питанні *How many terminals are there in the order?*.

Проте, створення систем, що мають можливість спілкування на ПМ в широких галузях, можливо, хоча поки результати далекі від задовільних.

4.3 Технології аналізу природної мови

Під технологією аналізу ПМ маємо на увазі переклад деякого виразу ПМ у внутрішнє подання. Фактично всі системи аналізу ПМ можуть бути

розподілені на такі категорії:

- підбір шаблона (*Pattern Matching*),
- синтаксичний аналіз,
- семантичні граматики,
- аналіз за допомогою відмінкових фреймів,
- “чекай і дивися” (*Wait And See*),
- словниковий експерт (*Word Expert*), конекціоністський,
- “ковзний” (*Skimming*) аналіз.

Нижче піде мова про деякі найпоширеніші методи.

4.3.1 Підбір шаблона

Суть даного підходу полягає в інтерпретації введення в цілому, а не в інтерпретації змісту і структури його окремих складових на більш низькому рівні. При використанні цього методу відбувається порівняння вже наявних у системі шаблонів-зразків з текстом, що надійшов на вхід. Звичайно шаблони подані у вигляді простого списку відповідностей між класами висловлювань і інтерпретаціями. Іноді вони доповнені семантичними елементами або іншими компонентами більш високого рівня. За таким принципом працює система *Еліза*, що імітує діалог із психотерапевтом. У дійсності система нічого не розуміє, а лише підтримує діалог, порівнюючи репліки пацієнта із шаблонами і присвоєними їм відповідними репліками, такими, як:

X завжди Y:	Ви можете навести який-небудь приклад?
	Коли?
	Дійсно, завжди?
X Ви Y мене:	Чому Ви думаєте, що я Y Вас?
	Вам подобається думати, що я Y Вас, чи не правда?
	Уявіть, що я Y Вас. Що Ви про це думаєте?

В результаті *Еліза* здатна вести такий діалог:

Пацієнт:	Люди **завжди** пильно дивляться на мене.	підбір за 1-м шаблоном з врахуванням *завжди*
Еліза:	Ви можете навести який-небудь приклад?	перша відповідь за 1-м шаблоном
Пацієнт:	Наприклад, зараз ти **дивишся** на мене.	підбір за 2-м шаблоном, Y=*дивитися*
Еліза:	Чому Ви думаєте, що я *дивлюся* на Вас?	перша відповідь за 2-м шаблоном
Пацієнт:	Тому що ти **ненавидиш** мене.	підбір за 2-м шаблоном, Y=*ненавидіти*
Еліза:	Вам подобається думати, що я *ненавиджу* Вас, чи не правда?	друга відповідь за 2-м шаблоном
	і т.д.	

4.3.2 Синтаксичний аналіз

При використанні синтаксичного аналізу відбувається інтерпретація окремих частин висловлювання, а не усього висловлювання в цілому. Звичайно спочатку здійснюється повний синтаксичний аналіз, а потім будується внутрішнє подання введеного тексту, або виконується інтерпретація.

Дерева аналізу і вільно-контекстні граматики. Більшість способів синтаксичного аналізу реалізовано у вигляді дерев. Один з найпростіших різновидів – вільно-контекстна граматика, що складається з правил типу *S=NP+VP* або *VP=V+NP* і вважають, що ліва частина правила може бути замінена на праву без врахування контексту. Вільно-контекстна граматика широко використовується в машинних мовах, і з її допомогою створені високоефективні методи аналізу. Недолік цього методу – відсутність заборони на граматично неправильні фрази, де, наприклад, підмет не погоджений із присудком у числі. Для рішення цієї проблеми необхідна наявність двох окремих, паралельно працюючих граматик: одна - для однини, інша - для множини. Крім того, необхідна своя граматика для пасивних пропозицій і т.д. Семантично неправильна пропозиція може породити величезну кількість варіантів розбору, з яких один буде перетворений у семантичний запис. Усе це робить кількість правил величезним і, у свою чергу, вільно-контекстні граматики непридатними для NLP.

Трансформаційна граматика. Трансформаційна граматика була створена з врахуванням згаданих вище недоліків і більш раціонального використання правил ПМ, але виявилася непридатною для NLP. Трансформаційна граматика створювалася Хомським як породжуюча, що зробило дуже важкою зворотну дію, тобто аналіз.

Розширена мережа переходів. Розширена мережа переходів була розроблена Бобровим (Bobrow), Фрейзером (Fraser) і багато в чому Вудсом (Woods) як продовження ідей синтаксичного аналізу і вільно-контекстних граматик зокрема. Вона являє собою вузли і направлені стрілки, “розширені” (тобто доповнені) біля тестів (правил), на підставі яких вибирається шлях для подальшого аналізу. Проміжні результати записуються в комірки (регістри). Нижче (рис. 4.2) наводиться приклад такої мережі, що дозволяє аналізувати прості речення всіх типів (включаючи пасив), що складаються з підмета, присудка і прямого доповнення, таких, як *The rabbit nibbles the carrot (Кролик гризе моркву)*.

Позначення біля стрілок означають номер тесту, а також ознаки, які аналогічні застосовуваним у вільно-контекстних граматиках (*NP*), або конкретні слова (by). Тести написані мовою LISP і являють собою правила типу *якщо умова = істина, то присвоїти аналізованому слову ознаку X і записати його у відповідну комірку.*

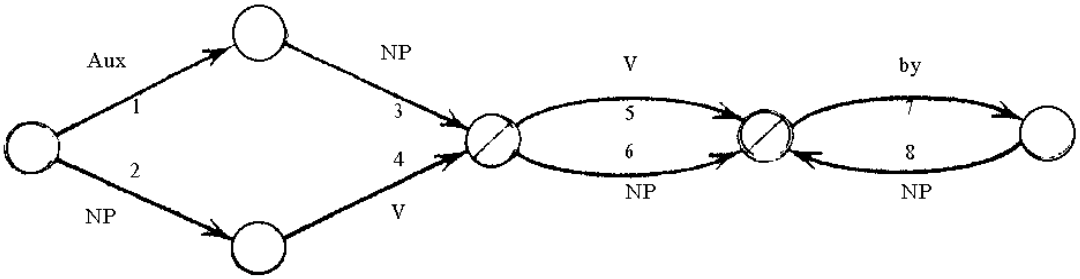


Рисунок 4.2 - Розширена мережа переходів

Розберемо алгоритм роботи мережі на вищенаведеному прикладі. Аналіз починається зліва, тобто з першого слова в реченні. Словосполучення *the rabbit* проходить тест, який з'ясовує, що воно не є допоміжним дієсловом (*Aux*, стрілка 1), а є іменною групою (*NP*, стрілка 2). Тому *the rabbit* заноситься в комірку *Subj,* і речення одержує ознаку *TypeDeclarative*, тобто *розповідне*, і система переходить до другого вузла. Тут додатковий тест не потрібен, оскільки він відсутній у списку тестів, записаних LISP. Отже, слово, що розташоване після *the rabbit* - тобто *nibbles* - дієслово-присудок (позначення *V* на стрільці), і *nibbles* записується в комірку з ім'ям *V*. Перекреслений вузол означає, що в ньому аналіз речення може в принципі закінчитися. Але в нашому прикладі є ще і доповнення *the carrot*, тобто аналіз продовжується по стрілці 6 (вибір між стрілками 5 і 6 здійснюється знову за допомогою спеціального тесту), словосполучення *the carrot* заноситься в комірку з ім'ям *Obj*. На цьому аналіз закінчується (останній вузол був би використаний у випадку аналізу такого пасивного речення, як *The carrot was nibbled by the rabbit*). Таким чином, у результаті заповнено регістри (комірки) *Subj*, *Type*, *V* і *Obj*, використовуючи які можна одержати певне подання (наприклад, дерево).

Розширена мережа переходів має свої недоліки:

- немодульність;
- складність при модифікації, що викликає непередбачені побічні ефекти;
- крихкість (коли єдина неграматичність у реченні унеможливлює подальший правильний аналіз);
- неефективність при переборі з поверненнями, тому що помилки на проміжних стадіях аналізу не зберігаються;
- неефективність з погляду змісту, коли за допомогою отриманого синтаксичного подання виявляється неможливим створити правильне семантичне подання.

Семантичні граматики. Аналіз ПМ, заснований на використанні семантичних граматик, дуже схожий на синтаксичний, з тією різницею, що замість синтаксичних категорій використовуються семантичні. Природно,

семантичні граматики працюють у вузьких предметних галузях. Прикладом служить система *Ladder*, вбудована в базу даних американських судів. Її граматика містить записи типу:

S <present> the <attribute> of <ship>

<present> what is|[can you] tell me

<ship> the <shipname>|<classname> class ship

Така граматика дозволяє аналізувати такі запити, як *Can you tell me the class of the Enterprise?* (*Enterprise* - назва корабля). У даній системі аналізатор складає на основі запиту користувача запит мовою бази даних.

Недоліки семантичних граматик полягають у тому, що, по-перше, необхідна розробка окремої граматики для кожної предметної галузі, а по-друге, вони дуже швидко збільшуються в розмірах. Способи виправлення цих недоліків - використання синтаксичного аналізу перед семантичним, застосування семантичних граматик тільки в рамках реляційних баз даних з абстрагуванням від загальномовних проблем і комбінація декількох методів (включаючи власне семантичну граматику).

4.4 Формування образа мовних препаратів

Акустичні сигнали відносяться до одного з різновидів носіїв інформації, що характерне для багатьох коливальних процесів. Найбільш цікава і найбільш важка для аналізу мова людини. Сучасні теоретичні і технічні засоби аналізу коливань є зовсім неприйнятними для мовних сигналів. Досконалі спектроаналізатори дозволяють фактично розкласти мовний сигнал на гармонічні складові з високою точністю. Демодулятори дозволяють виявити особливості зміни амплітуди у вихідному сигналі. Можна зіставити фази всіх гармонічних складових. Але вирішення проблеми виявлення ознак, які б дозволяли ідентифікувати слово, сказане двома людьми, поки що залишається проблемою, яка чекає свого вирішення [53].

Розглянемо один своєрідний підхід, що хоча не вирішує проблеми, але дозволяє намітити деякі кроки в цьому напрямку. Будемо вважати, що в сигналі відсутня постійна складова і виконано процес усунення завад. Тобто, сигнал можна зобразити у вигляді деякого складного коливання, як показано на рисунку 4.3.

Класичні методи, як уже зазначалося, не дають можливості одержувати інваріантні ознаки. Але в 30-х роках минулого сторіччя у телефонію прийшли методи кліпування і вокодерного зв'язку. Суть кліпування полягає в заміні складних за формою півхвиль реального сигналу на прямокутні імпульси. Тоді замість коливання, поданого на рис. 4.3, з’явиться сигнал імпульсного характеру, який показано на рис. 4.4.

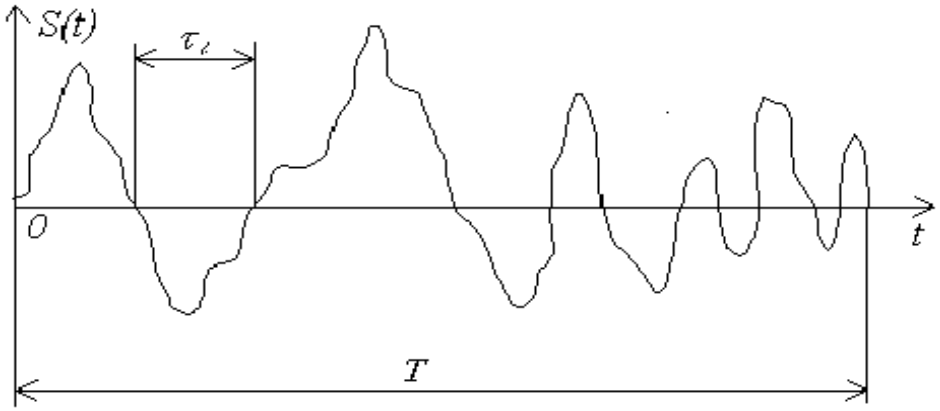


Рисунок 4.3 - Вихідний акустичний сигнал

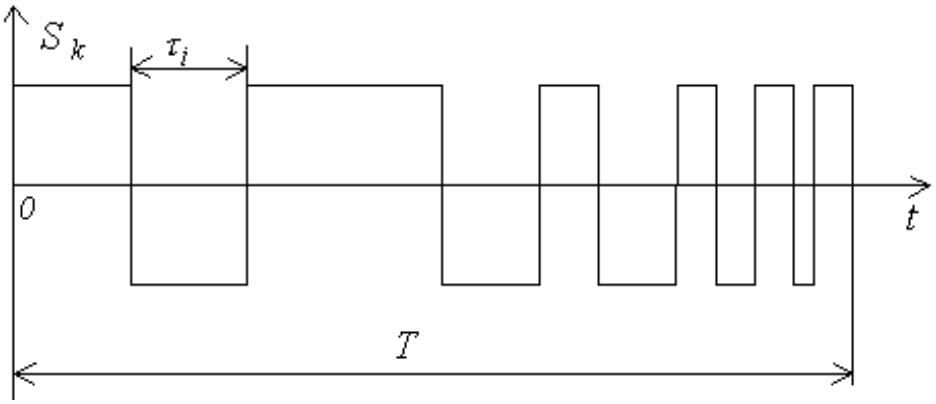


Рисунок 4.4 - Кліпований сигнал

Найнезвичайнішим і поки що непоясненим є те, що такий кліпований сигнал несе в собі досить розбірливу мову. Губляться тембр голосу, різні темброві особливості саме даної людини, але те, що сказано, виявляється досить зрозумілим, хоча і з деяким “металічним” відтінком, шипінням і т.д.

Що ж загального між цими сигналами (на рис.4.3 і 4.4)? Насамперед те, що позитивній півхвилі одного відповідає позитивний імпульс іншого. Але ця спільність досить умовна. Найбільш істотним спільним параметром є абсолютна рівність усіх τ_i, що визначають характер зміни частоти по всій довжині відрізка $0 \div T$.

Цілком справедливо побудувати своєрідний графік зміни частоти на усьому відрізку. При цьому по осі абсцис будемо відкладати поточний номер відрізка i, а по осі ординат можна відкладати τ_i або $2\tau_i$, або значення частоти в локальній зоні сигналу $f_i = \dfrac{1}{2\tau_i}$

Але найдоцільніше використовувати величину $ln(f_i)$. У будь-якому випадку різниця буде в зміні масштабу і тільки. Таку графічну форму

можна назвати частотним портретом коливання, оскільки вона (рис. 4.5) дає наочне уявлення про поведінку частоти на усьому відрізку $0 \div T$.

Точки сполучені ломаною лінією для наочності. У проміжках між двома сусідніми точками просто нічого немає. Але це ”нічого” теж можна перетворити в щось дуже корисне. Припустимо, що на відрізку $0 \div T$ зображена чиста синусоїда. Тоді відрізки τ_i будуть абсолютно однаковими, а її частотний портрет перетвориться в пряму лінію, паралельну осі абсцис.

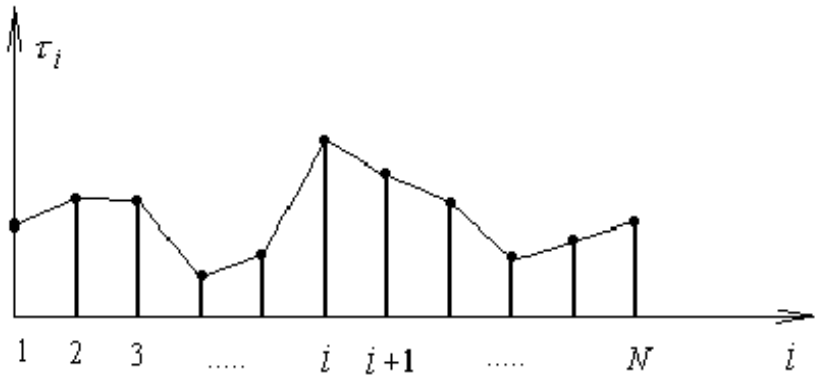


Рисунок 4.5 - Частотний портрет вихідного сигналу

Тепер є привід порівняти реальне коливання з ідеальним (синусоїдою). І зробити це можна, визначаючи тангенс кута α або сам кут і назвати цю величину дефектом частоти, що зображено на рисунку 4.6. При цьому важливо завжди застосовувати по осі абцис один і той самий масштаб. У цьому випадку будь-які зміни частотного діапазону, у якому вимовляється слово (а в кожної людини свій частотний діапазон - високий, низький, бас, фальцет і т.п.), будуть змінювати положення кривої частотного портрета, але дефект частоти буде незмінним.

Дефект частоти доцільно обчисляти як тангенс кута α, а не сам кут. Справа в тому, що $tg\alpha$ дуже легко знаходити, якщо Δi дорівнює одиниці. У цьому випадку різниця $\tau_{i+1} - \tau_i$ і є тангенсом кута, тобто для одержання послідовності дефектів частоти на усьому відрізку $0 \div T$ достатньо визначити відповідні різниці, що і зроблено у частотного портрета (рис. 4.6) і подано на рис. 4.7. Графік досить контрастно показує характер відхилення частоти в кожний момент часу краще, ніж графік частотного портрета.

Таким чином, знайдено досить інваріантну ознаку, за якою можна здійснювати порівняння коливань, використовуючи міру наближення. Елементами-координатами вектора будуть значення дефектів частоти у вигляді тангенсів кута α_i .

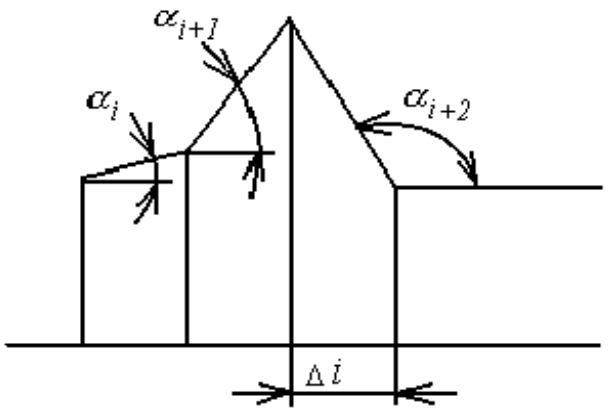


Рисунок 4.6 - Дефект частоти вихідного сигналу через кут α

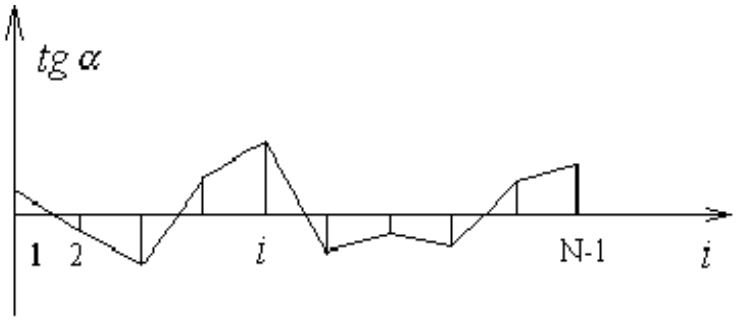


Рисунок 4.7 - Дефект частоти вихідного сигналу через тангенс кута α

4.5 Методи розпізнавання мовних сигналів

4.5.1 Методи розпізнавання, які використовують принцип декомпозиції мовного препарату на фонеми

Існуючі технології розпізнавання мови не мають поки що достатніх можливостей для їхнього широкого використання, але на даному етапі досліджень проводиться інтенсивний пошук можливостей застосування коротких багатозначних слів (процедур) для полегшення розуміння.

Розпізнавання мови в теперішній час знайшло реальне застосування в житті, напевно, тільки в тих випадках, коли словник, що використовується, скорочений до 10 слів, наприклад, при обробці номерів кредитних карток і інших кодів доступу, що базуються на комп'ютерах системах, які обробляють дані для передачі по телефону. Тобто невідкладна задача - розпізнавання принаймні 20 тисяч слів природної мови - залишається поки що недосяжною. Ці можливості поки що недоступні для широкого комерційного використання. Але ряд компаній самотужки намагається використовувати вже існуючі в даній області науки знання.

Для успішного розпізнавання мови потрібно розв’язати такі задачі:

- опрацювання словника (фонемний склад);

- опрацювання синтаксису;
- скорочення мови (включаючи можливе використання жорстких сценаріїв);
- вибір диктора (включаючи вік, стать, рідну мову і діалект);
- тренування дикторів;
- вибір особливого типу мікрофона (беручи до уваги спрямованість і місце розташування мікрофона);
- умови роботи системи й одержання результату з вказанням помилок.

Традиційно процес сприйняття мови поділяється на декілька етапів. На першому проводиться **дискретизація** неперервного мовного сигналу, переробленого в електричну форму.

Звичайно частота дискретизації складає 10÷11 кГц, розрядність - 8 біт, що вважається оптимальним для роботи зі словниками невеличкого обсягу (10÷1000 слів) і відповідає якості передачі мови телефонного каналу (3Гц ÷ 3.4 кГц). Зрозуміти, що збільшення обсягу активного словника повинно супроводжуватися підвищенням частоти дискретизації й у деяких випадках - підвищенням розрядності.

На другому етапі дискретний мовний сигнал **очищується від шуму** і **перетворюється в більш компактну форму**. Стискання здійснюється шляхом обчислення через кожні 10 *мс* деякого набору числових параметрів (звісно, не більше 16), із мінімальними втратами інформації, що описує даний мовний сигнал. Склад набору залежить від особливостей реалізації системи. Починаючи з 70-х років найпопулярнішим засобом (практичними стандартом) побудови стислого параметричного опису стало *лінійно-предикативне кодування* (ЛПК), в основі якого лежить достатньо завершена лінійна модель голосового тракту. На другому місці за популярністю знаходиться, мабуть, *спектральний опис*, отриманий за допомогою дискретного перетворення Фур'є.

Дуже гарні результати, проте, можуть бути досягнуті і при використанні інших засобів, часто менш вимогливих у обчислювальних ресурсів, наприклад, при *кліпуванні*. У цьому випадку реєструється кількість змін знака амплітуди мовного сигналу і часові інтервали між ними. Отримана як результат послідовність значень являє собою оцінку довжини періодів зберігання знака амплітудою. Цей спосіб досить повно подає відмінності між вимовленими звуками. На такому способі обробки заснована, наприклад, система розпізнавання мови, розроблена наприкінці 80-х років у НДІ рахункового машинобудування (Москва).

Тимчасовий (10 *мс*) інтервал обчислення був визначений і обґрунтований експериментально ще на зорі розвитку технології автоматичного розпізнавання мови. На цьому інтервалі дискретний випадковий процес, що подає оцифрований мовний сигнал, вважається стаціонарним, тобто, на такому часовому інтервалі параметри голосового

тракту значно не змінюються.

Наступний етап - **розпізнавання**. Еталони вимови, що зберігаються в пам'яті комп'ютера, по черзі порівнюються з поточною ділянкою послідовності 10 *мс* векторів, що описують вихідний мовний сигнал. Залежно від ступеня збігу обирається кращий варіант і формується гіпотеза про зміст вислову. При цьому виникає дуже істотна проблема - необхідність нормалізації сигналу за часом. Темп мови, тривалість вимови окремих слів і звуків навіть для одного диктора варіюється в дуже широких межах. Таким чином, можливі значні розбіжності між окремими ділянками еталона, що зберігається, і теоретично збіжним із ним вихідним сигналом за рахунок їхньої тимчасової розбіжності. Достатньо ефективно вирішувати дану проблему дозволяє розроблений у 70-х роках алгоритм динамічного програмування і його різновид (алгоритм Вітербі).

Особливістю таких алгоритмів є можливість динамічного стискання і розтягування сигналу по часовій осі безпосередньо в процесі порівняння з еталоном. З початку 80-х років усе більш широке застосування знаходять марківськи моделі, що дозволяють на основі багаторівневого імовірнісного підходу до опису сигналу здійснювати часову нормалізацію і прогнозування найімовірніших продовжень вислову, що значно прискорює процес перебору еталонів і підвищує надійність розпізнавання.

4.5.2 Методи машинного розпізнавання мови

Радянський учений Л. Л. М'ясников ще в 1941 р. запропонував схему динамічного аналізатора. Він же в 1943 р. сформулював принцип квантування сигналів у часі. Аналізатор М'ясникова розрахований на розпізнавання деяких звуків.

Алгоритм його роботи полягає в тому, що для пари фонем, що розпізнаються, наприклад **/а/** і **/про/**, можна підібрати таку пару фільтрів ***Фк*** і ***Фп***, сигнали виходу котрих після випрямлення і порівняння задовольняють умови:

$$А_к - А_п > 0,$$
$$О_к - О_п < 0.$$

Може виявитися також, що для якоїсь третьої фонеми (у даному випадку для фонеми **/я/)** буде справедливо

$$Я_к - Я_п = 0.$$

Таким чином, знак сигналу на виході розпізнавального пристрою (плюс, мінус або нуль) однозначно визначає, яка з трьох фонем була відтворена перед мікрофоном. Розпізнавання не залежить від інтенсивності, тембру й інтонації звуків, оскільки проводиться відносне порівняння. Додаючи нові пари фільтрів можна розширити алфавіт звуків, що розпізнаються.

У результаті інших досліджень були запропоновані інші методи розпізнавання звуків мови, засновані на аналізі їхніх акустичних ознак. Наприклад, швейцарський учений Дрейфус-Грааф у 1950 р. сконструював фонограф - автомат для графічного розпізнавання звуків мови. У цьому приладі звукові сполучення, сприйняті мікрофоном і посилені, надходили на гребінку із шести смугових фільтрів, що перекривають діапазон від 200 до 3000 Гц.

Виділені фільтром сигнали згладжувалися і подавалися на відхиляючі котушки, які переміщали перо, що записує, яке дотикалось до паперової стрічки, що рухається. У результаті цього на стрічці утворювалися діаграми вхідних звуків, що нагадують давні доалфавітні китайські і єгипетські піктографічні знаки. Звуки розпізнавалися за характером діаграм. Ознаками, що виділяються сонографом і характеризують окремі фонеми, були енергетичні рівні сигналу в шести частотних піддіапазонах.

4.5.3 Метод спектральночасового аналізу

Дослідження із розпізнавання окремих звуків мови (фонем), що розпочались тридцять років тому, поклали початок методу спектральночасового аналізу, що і в даний час є основою побудови систем розпізнавання.

Людська мова - змінний у часі, багатовимірний процес. Досліджувати її можна лише на основі динамічного аналізу, тобто аналізу змін мовних параметрів у часі.

З фізичної (акустичної) точки зору можливе подання мовного сигналу у вигляді амплітудно-частотної, частотно-часової або амплітудно-часової залежностей. Для проведення аналізу використовуються всі ці спектри.

Амплітудно-частотна залежність відображає тільки миттєву картину сигналу. Для часового аналізу потрібна послідовність таких картин, одержуваних через дуже короткі (порівняно з тривалістю самого сигналу) проміжки часу.

Робиться це звичайно за допомогою гребінки з перекриваючих один одного смугових фільтрів, що охоплюють у сукупності весь частотний діапазон мови (метод паралельного аналізу), або вузькосмугового фільтра, який швидко перебудовується, і після кожної перебудови опитується спеціальним електронним комутатором (метод послідовного аналізу). Кількість смугових фільтрів (або число перебудов вузькосмугового фільтра) вибирається залежно від умов аналізу в широких межах (від 5 до 40).

Для дослідження мовних процесів у реальному масштабі часу звичайно застосовується метод паралельного аналізу, а для детального, більш повільного аналізу - послідовний метод.

Миттєві спектри не можуть самі по собі слугувати джерелами відмітних ознак мовного сигналу, тому що вони містять лише миттєві значення інтенсивності або частоти звука, що носять до того ж випадковий характер внаслідок мінливості природної мови.

Для одержання відмітних ознак використовують цілий ряд (послідовність) миттєвих спектрів, що йдуть один за одним, піддаючи їх обробленню в різноманітних обмежувальних, перетворювальних схемах розпізнавального пристрою.

У результаті такого оброблення виділяються ознаки, що у певному поєднанні дозволяють найбільш надійно і швидко відрізнити даний мовний сегмент або звук від інших сегментів і звуків.

До цих ознак відносяться:

- частота першої, другої і третьої формант;
- протяжність формантних ділянок;
- нахил згинаючої формантної ділянки;
- рівні максимальної і мінімальної енергії;
- різниця рівнів позитивної і негативної частин спектра;
- миттєва потужність сигналу;
- тривалість ділянок певної інтенсивності;
- тривалість нестаціонарних (перехідних) ділянок;
- крутизна наростання сигналу;
- частоти (щільності розподілу) переходів через нуль у тональній і шумовій складових сигналу;
- висота (частота) основного тону;
- тривалість паузи (змички) - глухої і тональної;
- частоти нечутних компонентів (ультра- і інфразвуків).

Використовуються і деякі похідні характеристики перерахованих ознак:

- відношення формантних моментів (F2 - F1) до (F3 - F2);
- різниця і добуток рівнів енергії двох частотних каналів;
- похідна за часом зміни формантного максимуму.

У процесі розпізнання мовного сигналу автоматичний пристрій повинен визначити, які з заданих ознак містяться в сигналі, у якому поєднанні вони з'являються.

Великий набір використовуваних ознак розширює можливості системи розпізнавання, підвищує точність розпізнання, але потребує автоматизації процесів виділення й опрацювання ознак (через їхню громіздкість). Це ускладнює систему.

Обмежений набір ознак дозволяє вирішувати тільки часткові, більш прості задачі. Слід також мати на увазі, що складність схем і пристроїв, які виділяють різні ознаки, неоднакова. Наприклад, пристрій, що визначає тональність сигналів, є відносно простим, а виявлення крутизни

наростання сигналу потребує застосування складних схем.

У зв’язку з цим для різних задач розпізнавання спектрально-часовий аналіз виконують у різному обсязі - відповідно до встановленого набору відмітних ознак.

За характером ознак звуки мови приблизно можна розділити на такі класи:

1. *Тональні* звуки (голосні і деякі приголосні) - характеризуються наявністю основного тону, інтенсивністю і тривалістю звучання, формантними параметрами (частотами резонансів, амплітудами, шириною формантної смуги).
2. *Шумові* звуки (частина приголосних) - характеризуються наявністю суцільного шумового спектру, тривалістю.
3. *Вибухові* звуки (частина приголосних) - характеризуються крутизною наростання сигналу, максимальним і середнім рівнями потужності.
4. *Паузи* (моменти мовчання) - характеризуються тривалістю, миттєвою потужністю сигналу.

Просодичні особливості мови характеризуються тривалістю голосних і деяких приголосних, змінами частоти основного тону й інтенсивності сигналів.

До цього часу мова йшла про спектрально-часовий аналіз мовних сигналів в умовах відсутності зовнішніх звукових завад. У дійсності такі завади (природного й індустріального походження) існують завжди. При аналізі необхідно виключити їх з сигналів. Робиться це різноманітними засобами, заснованими на вирахуванні спектрів.

Надалі ми будемо звертатися до методу спектрально-часового аналізу мовних сигналів у його різних варіантах.

Як діаметрально протилежні приклади застосування цього методу, що характеризують його великі можливості, наведемо систему, що аналізує мовні сигнали в реальному масштабі часу за розподілом щільності переходів сигналу через нуль, і результати експериментів з аналізу шумів дихальних шляхів із метою об'єктивної діагностики захворювань.

У розпізнавальній мовній системі сигнал, що надходить, розділяється на низькочастотну і високочастотну складові, перетворюється в двополюсні прямокутні імпульси різної тривалості; перетворюється в аналого-цифровому пристрої в цифрові послідовності і вводиться в ЕЦОМ. Інтервали між переходами сигналу через нуль за кожною із складових протягом кожного короткого проміжку часу (порядку 20 мс) класифікуються за тривалістю на декілька груп.

В результаті такого сортування і підрахунку інтервалів у різних групах (каналах) обох складових сигналу накопичується інформація про число інтервалів різної тривалості. Розподіл тривалості у всіх каналах у сукупності характеризує певний звук.

Отримані послідовності інтервалів зіставляють із набором еталонних послідовностей і за найкращим збігом розпізнають звук, що аналізується.

Окремим словам у координатній системі двох складових сигналу відповідає графічно виражений розподіл тривалості інтервалів, що мають характерні ознаки. За цими ознаками розпізнаються слова - назви десяти цифр.

Успіхи в галузі спектрально-часового аналізу мовних сигналів наштовхнули дослідників-медиків на думку про застосування цього методу для вивчення близьких до звукоутворення процесів - дихальних шумів.

Відомо, що існуючі в медицині методи діагнозу патологічних шумів дихальних шляхів досить суб'єктивні, тому що ґрунтуються на особистій оцінці лікарем результатів прослуховування хворого, його огляду і рентгеноскопічного дослідження.

Як об'єктивний метод спробували застосувати спектрально-часовий аналіз. В результаті статистичного опрацювання отриманих даних були виявлені залежності між видом захворювання і характером спектра дихальних шумів для чотирьох видів захворювань дихальних шляхів.

4.5.4 Метод двійкового відбору (бінарної селекції)

У основі методу лежить виявлення в звуках мови елементарних розпізнавальних (диференціальних) ознак шляхом сортування звуків за принципом двійкового відбору і розпізнавання звука за сумою виявлених у ньому ознак.

Система диференціальних ознак введена в застосування для досліджень мови вченими Якобсоном, Фантом, Трубецьким, Цемелем і ін. Відповідно до цієї системи інформація, що міститься в мовному повідомленні, виражається мінімальними розходженнями - протиставленнями між елементами мови. За ними слухач робить ряд рішень, кожне з який робить елемент мови несхожим на інші.

Таким рішенням може бути вибір між двома полярними властивостями однієї категорії (наприклад, низькотональний - високотональний) або між наявністю і відсутністю якої-небудь якості (наприклад, дзвінкий - глухий).

У основу протиставлень покладені як акустичні, так і артикуляційні категорії.

Вибір між двома протилежностями називають диференціальною ознакою. Їй приписується первинна розпізнавальна суть мови. Самі по собі диференціальні ознаки не мають смислового значення. Їхня роль полягає в тому, що вони сигналізують про відмінність даного звука від інших, якщо він несе протилежну ознаку.

Мала кількість диференціальних ознак (у будь-якій сучасній мові їх не більше дванадцяти), що використовуються для побудови мовних образів, спонукала дослідників направити свої зусилля на створення

розпізнавальних пристроїв, заснованих на розрізненні окремих фонем за «низками» властивих їм диференціальних ознак.

Передбачалося, що такі пристрої будуть досить простими і достатньо універсальними.

Перед тим, як звернутися до конкретних прикладів подібних пристроїв, наведемо повний перелік диференціальних ознак (за Якобсоном і Халле).

У одному ранньому пристрої, що розпізнає прості мовні сигнали методом двійкового відбору за диференціальними ознаками, звуки послідовно розділяються на групи, а ті у свою чергу - на окремі фонеми залежно від наявності або відсутності тієї або іншої ознаки. Ознаки ідентифікуються (встановлюються) за акустичними характеристиками.

Поділ на дзвінкі і глухі проводиться при наявності або відсутності основного тону і за частотою переходів через нуль, поділ на шумові і нешумові - за рівнем енергії в зоні першої форманти. Сортування голосних на високі і низькі проводиться за різницею середніх рівнів енергії в діапазонах 800 ÷1200 і 2400 ÷ 3700 Гц.

Глухі приголосні поділяються на вибухові і фрикативні за рівнем енергії початкової ділянки сигналу. Подібним чином побудований поділ і в інших групах. Для сортування використані електронні схеми.

Пристрій показав гарні результати при розв’язанні обмежених задач. Наприклад, у коротких словах, що вимовлялись 21 диктором, точність розпізнавання голосних склала 94%, що можна порівняти з можливостями людини.

У іншому експериментальному пристрої, що використовує ЦОМ, методом двійкового відбору розпізнаються голосні і більш складні дифтонги.

Аналіз проводиться в реальному масштабі часу. Мовні сигнали діляться на короткі сегменти, що вводяться в пристрій у спектральному вигляді за допомогою 35 смугових фільтрів.

Кожний сегмент кожного частотного каналу перетворюється в цифрову форму. Послідовності цифр обробляються ЦОМ.

Голосні і дифтонги розпізнаються за чотирма диференціальними ознаками:

1. Високотональні – низькотональні;
2. Компактні – дифузійні;
3. Бемольні – прості;
4. Напружені - ненапружені.

Фізичними (акустичними) виразниками цих ознак є:

1. Висока 2-а форманта - низька 2-а форманта;
2. Висока 1-а форманта - низька 1-а форманта;
3. Поріг F1+F2 - поріг F1 - F2;
4. Велика тривалість - мала тривалість.

Пристрій забезпечує точність розпізнавання голосних, що дорівнює 92%, і дещо меншу точність для дифтонгів. Образами, що розпізнаються є односкладові слова. При цьому:

Програма розпізнавання кожного сегмента передбачає:

- поділ на тональні і нетональні (відсутність енергії на частоті більшій 350 Гц свідчить про відсутність дзвінкості; наявність складових, більших ніж 4000 Гц, указує на турбулентність;
- відділення голосних від приголосних (за силою звука);
- розпізнання голосних (за положенням формант на частотній шкалі).

Як очевидно з прикладів, метод двійкового відбору дає гарні результати при розпізнаванні окремих простих слів, що містять голосні звуки, які характеризуються одночасно декількома диференціальними ознаками. Розпізнавання приголосних виявляється важчим через меншу кількість і мінливість ознак.

Що стосується більш протяжних мовних конструкцій - багатоскладових слів і фраз, то розпізнавання їх зазначеним методом викликає ускладнення. Перше полягає в тому, що в поширених і злитих мовних образах дуже складно виявити ознаку звука, що вказує на відсутність якої-небудь якості (наприклад, дзвінкості).

Повна відсутність ознаки звичайно не спостерігається, можна виявити лише різний ступінь її пригнічення. А це потребує встановлення при розпізнаванні певного порогу, за яким ознака або присвоюється, або відхиляється. В умовах широкої мінливості мови будь-який такий поріг набуває не загального, а часткового характеру.

Інша трудність полягає в неоднорідності одних і тих самих фонем у суцільній мові, а отже, і в мінливості їхніх диференціальних ознак.

Ці складності певною мірою усуваються, якщо при розпізнаванні поділяти мовні сигнали не на фонеми, а на більш короткі одиниці - сегменти. Такий підхід, а також сполучення методу двійкового відбору з іншими методами розпізнавання дають досить непогані результати.

4.5.5 Метод формантного аналізу

Формантний аналіз можна розглядати як особливий вид спектрально-часового аналізу. У цьому випадку задача розпізнавання полягає у виявленні характеристик ділянок спектра, що утворюють так звані форматні зони.

Мовний тракт являє собою багаторезонансну систему. Тому спектр сигналу на її виході є результатом накладення один на одного великого числа затухаючих гармонічних коливань, при одержанні якого утворюється множина амплітудних максимумів.

Для аналізу істотне значення мають перші три максимуми (перші три форманти: F1, F2, F3). Частотні значення цих максимумів, а також ширина

форматних ділянок і крутизна обвідної спектра в зоні максимуму є характерними ознаками голосних і деяких приголосних звуків, за цими ознаками можливе їхнє розпізнавання.

Наприклад, от як відрізняється за частотою розташування першої форманти в голосних звуках **/У/**, **/ПРО/**,**/А/**: У-200-400, О-400-700 і А- 700-1100 Гц.

Для частини приголосних формантні ознаки недостатні для ідентифікації.

Виявлення формантних частот проводиться різними способами. Одним із перших був застосований спосіб підрахунку числа переходів сигналу через нуль. Частоти F1 і F2 у цьому випадку приблизно визначалися відповідно середньою щільністю переходів через нуль і її похідною за часом. Інший спосіб полягає в знаходженні формантної частоти з співвідношення формантних моментів для ділянки спектра, що розглядається.

При аналізі сигналів у реальному масштабі часу проводиться безпосередній вимір максимумів миттєвого амплітудного спектра різними способами, зокрема за нульовим нахилом згинальної формантного піка.

Цей засіб, запропонований Д. Фланаганом, полягає в такому.

За допомогою набору смугових фільтрів, детекторів і інтеграторів, підключених до комутатора, одержують миттєві амплітудно-частотні спектри сигналу, що йдуть один за одним з частотою 100 Гц.

Значення частот цієї спектральної послідовності зчитуються, запам’ятовуються, диференціюються і перетворюються в двійкову цифрову послідовність. Цифри, що відзначають частотні максимуми, є маркірувальними імпульсами, що визначають рівні максимальної напруги в аналогових схемах. У результаті на виході пристрою одержують закономірності зміни перших трьох формантних частот у часі. За характером цих змін розпізнають окремі звуки.

Застосовуючи ЕЦОМ, формантний аналіз сигналів виконують способом підгонки синтезованих у машині спектрів до аналізованих миттєвих спектрів. Синтез здійснюється або на основі розкладання гармонічної складової сигналу в ряд Фур'є, або шляхом використання параметрів моделі артикуляції (передатної функції). При найкращому збігу синтезований спектр використовується для ідентифікації сигналу.

Спосіб підгонки в принципі придатний для розпізнавання як голосних, так і приголосних, але модель, що підбирається для приголосних звуків виявляється більш складною.

Можливе комбінування способу визначення формантних частот за піковим значенням з засобом синтезованих спектрів.

Ширина формантної смуги звичайно визначається за частотною згинальною між двома її точками, що лежать по обидві сторони максимуму на рівні, що складає половину максимального значення

частоти.

Розвитком методу формантного аналізу є виділення ознак, близьких до формантних, із застосуванням кліпування мовного сигналу.

4.5.6 Метод використання лінгвістичної інформації

Раніше вже було сказано, що із ускладненням задач автоматичного розпізнавання акустичної інформації, що міститься в мовному сигналі, виявляється недостатньо для розпізнання мовних послідовностей і доводиться вводити в пристрій розпізнавання ті чи інші лінгвістичні правила.

Однією з перших і найбільш простих спроб застосувати цей метод стало створення пристрою для розпізнавання окремих слів, у якому лінгвістична інформація врахована у формі ймовірностей проходження одного звука за іншим.

У акустичній частині пристрою проводиться аналіз сигналів на основі їхніх спектрально-часових характеристик і порівняння отриманих даних з еталонними спектральними картинами. За результатами порівняння приймається попереднє рішення про ідентифікацію звуків.

Логічна частина пристрою визначає ймовірність появи кожного звука в ряду інших звуків (наприклад, яка ймовірність того, що після **/е/** піде **/м/** і за найбільшим значенням ймовірності встановлює звук, що розпізнається.

Обидва рішення порівнюються, і при їхньому збігу видається сигнал на друк для відтворення розпізнаного звука.

Досліди, проведені з подібними пристроями, показали, що хоча використання лінгвістичної інформації не покращує розпізнавання окремих звуків, зате точність розпізнання простих слів збільшується приблизно вдвічі.

4.5.7 Методи розпізнавання мови за допомогою аналізу її зображення

Один із таких методів запропонований у СРСР В. С. Файном і В. Н. Сорокіним. У його основі лежить аналіз відеограм мови, що одержуються за допомогою спектрографа.

Відеограма є тривимірним зображенням динамічного спектра мови. Перевага її порівняно з аналоговим або цифровому поданням мовного сигналу полягає в тому, що вона є краще організованим і більш завадостійким образом.

Те саме слово, вимовлене різними людьми і подане у формі відеограм, має велику схожість у розумінні можливості перетворити різні його відбитки один в одного топологічним способом, тобто надати цим відбиткам те саме значення в обраному для аналізу просторі. Цей ефект пояснюють топологічною однаковістю голосових трактів різних людей при

вимовлянні ними того самого слова.

На відеограмі мовного сигналу по вертикальній осі відкладаються значення частоти сигналу, по горизонтальній - час, а ступінь щільності зображення (його колір від блідо-сірого до глибоко чорного) характеризує інтенсивність сигналу.

Для розпізнавання відеограм застосовується неперервно-груповий метод аналізу, що дає уявлення про аналізований мовний образ у вигляді груп математичних множин, що описують класи зображень в обраному просторі. Застосування цього методу полегшує опис образу фонеми, тому що дає можливість досліджувати мовні сегменти з урахуванням впливу на них акустичного оточення. Це дає особливо гарний ефект при розпізнаванні глухих вибухових звуків, позбавлених формант, дуже слабких енергетично, але значно впливаючих на оточуючі звуки.

Для виділення інформації з відеограми використовують параметри формантних зон, які виражаються для елементарних ділянок відеограми характерними прямими лініями і точками в частотно-часовій площині. Такі лінії, обом кінцям яких присвоюються цифрові індекси, і окремі позначені цифрами точки показують, як розташовані на площині ділянки формантні максимуми, а особлива цифра, крім того, фіксує максимальні енергетичні рівні.

Розташовані одна за одною ділянки відеограми шляхом застосування зазначеного вище методу перетворюються в послідовність цифрових матриць, що вводяться в ЕЦОМ. Машина в кожній матриці аналізує координати всіх характерних точок, обчислює центри ваги абсцис і ординат скупчень однакових індексів і на цій основі розбиває матрицю на клітинки, що містять характерні графи-символи мовних образів.

Графи являють собою сполучення двох-трьох точок зображення, що поміщаються у тій чи іншій клітинці.

Вертикальні ряди клітинок містять інформацію: про номери формант - для тональних ділянок, про високочастотність або низькочастотність - для шумових ділянок і про дзвінкість або глухість - для ділянок змичок (пауз). Горизонтальні ряди клітин дають уявлення про чергування стаціонарних (тональних) ділянок, шумових ділянок і змичок.

На останньому етапі розпізнавання за допомогою описаного алгоритму проводиться порівняння аналізованого динамічного спектра з еталонним і встановлюється їхня відповідність, тобто ідентифікується мовний сигнал, що надійшов.

Інший метод аналізу зображень мови розроблений Ліверингтоном в Англії. У основі методу лежить аналіз голографічних зображень мовних образів. Такі зображення одержують шляхом перетворення в лазерному промені відеограм мови в голограми.

Для голографічного запису кожної мовної послідовності використовується певний, відмінний від інших, кут нахилу еталонного

лазерного променя.

Отримана голограма образу, що розпізнається, розташовується за еталонною голограмою, і обидві вони просвічуються розширеним колімірованим лазерним променем. Якщо обидва образи ідентичні, відбувається відновлення зображення аналізованої голограми, проникність її стає найбільшою і це виражається яскравою плямою на контрольному екрані. Якщо ж образи не збігаються, то пляма не з'являється.

Техніка зіставлення образів може полягати, зокрема, у почерговому прокручуванні плівки з усіма голографічними образами, що розпізнаються перед заданими еталонами.

Голограми слів розпізнаються гірше, ніж голограми окремих фонем.

Перевагами методу є більш простий порівняно з електронним методом аналіз і можливість швидкої зміни еталонів при зміні дикторів.

4.5 8 Метод фазового аналізу

Існуючі в мовних сигналах фазово-амплітудні залежності через їхню складність до останнього часу не використовувалися [52].

Проте досвід створення сучасних радіолокаційних і гідроакустичних систем показав, що у фазових характеристиках міститься багато цінної інформації. Це дало поштовх до розробки фазових методів розпізнавання мови. Передбачається, що з їх допомогою вдасться більш успішно перебороти складності, що полягають у близькості спектрів ряду звуків і їх випадковій мінливості.

Один із таких методів заснований на визначенні статистичних властивостей траєкторії зображуваної точки, на фазовій площині, тобто на одержанні й ідентифікації «фазових портретів» звуків мови.

4.6 Структурні методи аналізу та ідентифікації мовних препаратів

Аналіз за допомогою відмінкових фреймів. Зі створенням відмінкових фреймів пов’язаний великий стрибок у розвитку NLP. Вони набули популярності після роботи Філлмора “Справа про падіж”. На сьогоднішній день відмінкові фрейми - один з найчастіше використовуваних методів NLP, тому що він є найбільш комп’ютерно-ефективним при аналізі як знизу вверх (від складових до цілого), так і зверху вниз (від цілого до складових).

Відмінковий фрейм складається з заголовка і набору ролей (відмінків), пов’язаних певним чином із заголовком. Фрейм для комп’ютерного аналізу відрізняється від звичайного фрейму тим, що відношення між заголовком і ролями визначається семантично, а не синтаксично, тому що в принципі одному і тому ж слову можуть приписуватися різні ролі, наприклад, іменник може бути як інструментом дії, так і його об’єктом.

Загальна структура фрейму така:

[Заголовне дієслово

[відмінковий фрейм

агент: <активний агент, що виконує дію>

об'єкт: <об’єкт, над яким відбувається дія>

інструмент: <інструмент, використовуваний при здійсненні дії>

реципієнт: <одержувач дії - часто непряме доповнення>

напрямок: <ціль (звичайно фізичної) дії>

місце: <місце, де відбувається дія>

бенефіціант: <сутність, в інтересах якої відбувається дія>

коагент: <другий агент, що допомагає робити дію>

]]

Наприклад, для фрази *Іван дав м'яч Каті* відмінковий фрейм виглядає так:

[Давати

[відмінковий фрейм

агент: Іван

об'єкт: м'яч

реципієнт: Катя]

[грам

час: минулий

застава: акт]

]

Існують обов’язкові, необов’язкові і заборонені відмінки. Так, для дієслова *розбити* обов'язковим буде відмінок *об’єкт* - без нього висловів буде незакінченим. *Місце* і *коагент* будуть у даному прикладі необов’язковими відмінками, а *напрямок* і *реципієнт* - забороненими.

Часто в NLP буває корисним використовувати семантичне подання в якомога більш канонічній формі. Найбільш відомим способом такої репрезентації є метод **концептуальних залежностей**, розроблений Шенком для дієслів дії. Він полягає в тому, що кожна дія подана у вигляді однієї або більше найпростіших дій.

Наприклад, для речень *Іван дав м'яч Каті (1)* і *Катя взяла м'яч в Івана (2),* що розрізняються синтаксично, але обидва містять акт передачі, можуть бути побудовані такі репрезентації з використанням найпростішої дії *Atrans*, що застосовується в граматиці концептуальних залежностей:

(1)

[Atrans

відн.: володіння

агент: Іван

об'єкт: м'яч

джерело: Іван

реципієнт: Катя]

(2)

[Atrans

відн.: володіння

агент: Катя

об'єкт: м'яч

джерело: Іван

реципієнт: Катя]

За допомогою такого подання легко виявляються подібності і розходження фраз. Для полегшення аналізу також використовується розподіл ролі на лексичний маркер і заповнювач. Так, для ролі *об'єкт* може бути встановлений маркер *пряме доповнення*, для ролі *джерело* - маркер виду *<з>=із|від|...*

У загальному вигляді аналіз тексту за допомогою відмінкових фреймів складається з таких кроків:

- використовуючи існуючі фрейми, підібрати підхожий для заголовка. Якщо такого нема, текст не може бути проаналізований;
- повернути в систему підхожий фрейм із відповідним заголовком-дієсловом;
- спробувати провести аналіз по всіх обов'язкових відмінках. Якщо один або більше обов'язкових заповнювачів відмінків не знайдені, повернути в систему код помилки. Такий випадок може означати наявність еліпсиса, неправильний вибір фрейма, неправильно введений текст або недолік граматики. Наступні кроки використовуються вже для аналізу і виправлення таких ситуацій;
- провести аналіз по всіх необов'язкових відмінках;
- якщо після цього у введеному тексті залишилися непроаналізовані елементи, видати повідомлення про помилку, яка пов'язана з неправильним введенням, недостатністю даного аналізу або необхідністю провести інший, більш гнучкий аналіз.

Переваги використання відмінкових фреймів такі:

- сполучення двох стратегій аналізу (зверху вниз і знизу вверх);
- комбінування синтаксису і семантики;
- зручність при використанні модульних програм.

Стійкість аналізу. Певні труднощі при аналізі виникають із варіативністю того самого запиту. Наприклад, на вхід системи, що керує зарахуванням і перерозподілом учнів на курсах різних спеціальностей, може надійти запит типу *Переведіть Петрова, якщо це можливо, з математики на, скажімо, економіку*.

Найлегше такі труднощі переборюються при використанні відмінкових фреймів. Правило, сформульоване Карбонеллом і Хейзом, говорить: "Варто пропускати невідомі введені елементи доти, доки не буде знайдений відмінковий маркер; пропущені елементи варто аналізувати з урахуванням незаповнених відмінків, використовуючи тільки семантику".

Діалог. Разом із проблемою розпізнавання тексту існує і проблема підтримки інтерактивного діалогу. При цьому виникають додаткові особливості, характерні для діалогів, а саме:

- анафора (тобто використання займенників замість їхніх анафоричних антецедентів – самостійних частин мови);
- еліпсис;

- екстраграматичні речення (пропуск артиклів, помилки, вживання вигуків і т.п.);
- металінгвістичні речення (тобто спроба виправлення введеного раніше).

Крім того, користувачі систем із природно-мовним інтерфейсом намагаються висловлюватись якомога коротше, що в ряді випадків також утрудняє аналіз.

Використання відмінкових фреймів, а саме злиття поточного фрейма з попереднім, забезпечує відновлення еліпсиса.

Таким чином, процес розробки систем, що забезпечують розуміння ПМ, вимагає створення механізмів, відмінних від традиційних способів подання ПМ, а системи з природно-мовними інтерфейсами застосовуються тільки у вузьких предметних галузях.

Питання для самоконтролю

1. Які проблеми варто вирішити для успішного розпізнавання мови?
2. На чому ґрунтуються сучасні системи розпізнавання мови?
3. Які методи використовуються для усунення негативного ефекту впливу варіювання голосового тракту на процес розпізнавання мови?
4. У чому полягає додатковий обмежувальний вплив синтаксису?
5. У чому полягає відмінність методів розпізнавання мови, заснованої на ланцюгах Маркова?
6. Які концепції обробки природної мови Ви знаєте?
7. Що є об'єктами і задачами досліджень при обробці природної мови?
8. У чому полягає основна проблема обробки природної мови?
9. На які категорії можуть бути розподілені системи аналізу ПМ?
10. Розкрийте зміст підходу «Підбір шаблона».
11. Які недоліки має розширена мережа переходів?
12. У чому полягають недоліки семантичних граматик?
13. Назвіть способи одержання частотного портрета.
14. На які класи і за яким критерієм можна розділи звуки мови?
15. Поясніть основний зміст форматного аналізу.
16. Які методи розпізнавання мови за допомогою аналізу її зображення Ви знаєте?

5 ЕКСПЕРТНІ СИСТЕМИ

Комп’ютерна система, що заснована на знаннях, основними структурними елементами якої є бази знань і механізм логічних висновків називається експертною системою (ЕС), є одним з етапів на шляху реалізації досягнень у штучному інтелекті.

5.1 Призначення ЕС і основні вимоги до них

Експертні системи - це комп'ютерні програми, створені для виконання тих видів діяльності, що під силу людині-експерту. Вони працюють таким чином, що імітують образ дій людини-експерта, істотно відрізняються від точних, добре аргументованих алгоритмів і не схожі на математичні процедури більшості традиційних розробок [54-56].

Якщо при традиційному процедурному програмуванні комп'ютеру необхідно повідомити що і як він повинен робити, то загальним для експертних систем є те, що вони мають справу зі складними проблемами:

- які недостатньо добре розуміються або вивчені;
- для яких немає чітко заданих алгоритмічних рішень;
- які можуть бути досліджені за допомогою механізму символічних міркувань.

Специфіка ЕС полягає в тому, що вони використовують:

- механізм автоматичного висновку;
- «слабкі методи», такі як пошук чи евристика.

Основними вимогами до ЕС є:

- використання знань, пов’язане з конкретною предметною галуззю;
- придбання знань від експерта;
- визначення реальної і достатньо складної задачі;
- наділення системи здібностями експерта.

Експерти - це кваліфіковані фахівці у своїх галузях діяльності – фінансисти, економісти, лікарі, адвокати і т.д., що мають загальні якості:

- мають величезний багаж знань про конкретну предметну галузь;
- мають великий досвід роботи в цій галузі;
- уміють точно сформулювати і правильно розв’язати задачу.

ЕС покликані замінити фахівців у конкретній предметній галузі, тобто дозволити розв’язати задачу без експерта [56, 57].

5.2 Спрощена структура ЕС

Спрощена базова структура ЕС має вигляд, зображений на рис. 5.1. Для успішного виконання функцій, покладених на ЕС, необхідні:

- механізм подання знань про конкретну предметну галузь і керування ними;
- механізм, який на підставі наявних у базі знань (БЗ) здатен робити висновки;

- інтерфейс для одержання і модифікації знань експерта, а також для правильної передачі відповідей користувачу (інтерфейс користувача);
- механізм одержання знань від експерта, підтримки БЗ і при необхідності її доповнення (модуль надбання знань);
- механізм, який не тільки здатен робити висновки, але і надавати різні коментарі до цього висновку і пояснювати його мотиви (модуль порад і пояснень).

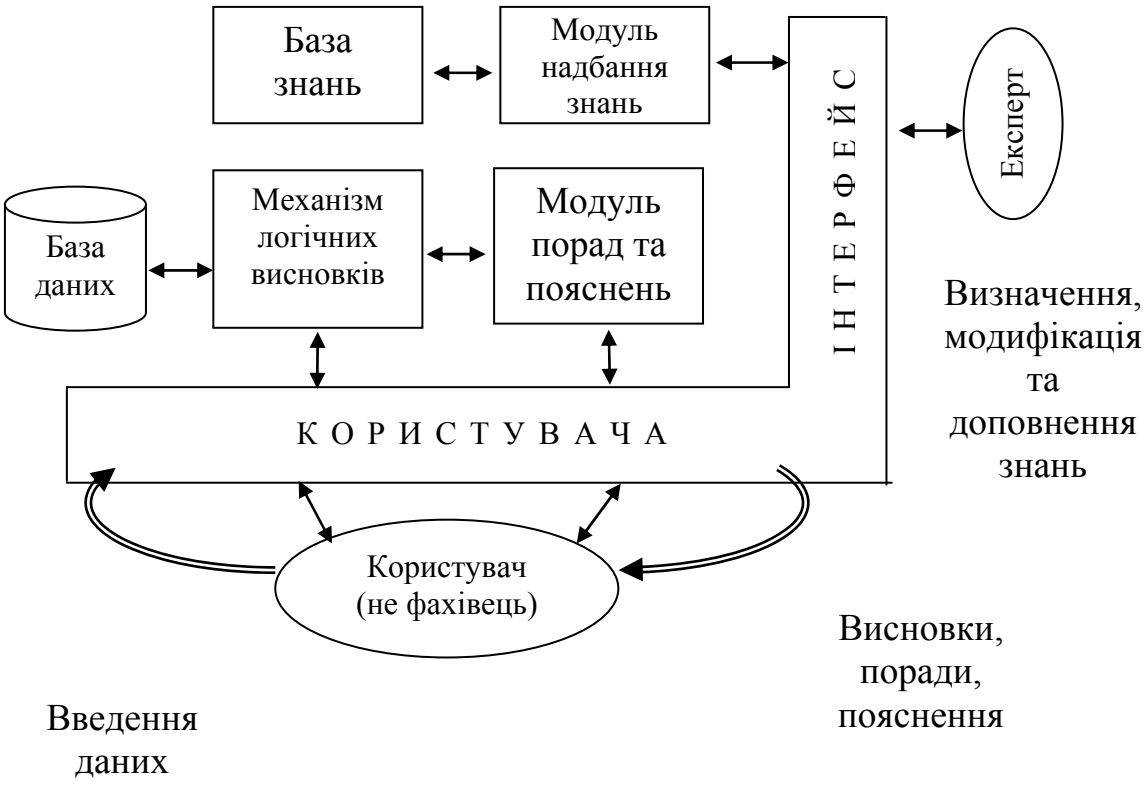


Рисунок 5.1 - Базова структура експертної системи

Варто особливо підкреслити важливість механізму пояснень у складі ЕС:

- без нього користувачу важко буде зрозуміти висновок, отриманий при консультації чи вирішенні будь-якого питання;
- цей механізм важливий для експерта, він дозволяє визначити, як працює система і з'ясувати, як використовуються надані ним знання.

Мова подання знань, що використовується для розробки ЕС, називається **мовою розробки ЕС**, а система програмного забезпечення, що включає зазначені вище функції, називається **інструментом для розробки ЕС** чи **оболонкою ЕС**.

5.3 База знань як елемент експертної системи

База знань містить факти і правила. Факти – це фрази без умов, вони містять твердження, що завжди абсолютно правильні. Правила містять твердження, істинність яких залежить від деяких умов, що утворюють тіло правила [58-61].

Факти містять короткострокову інформацію в тому значенні, що вони можуть змінюватися, наприклад, під час консультації.

Правила являють собою довгострокову інформацію про те, як породжувати нові факти чи гіпотези з того, що зараз відомо.

Чим такий підхід відрізняється від звичайної методики використання БД? Основна відмінність полягає в тому, що БЗ має великі «творчі» можливості. Факти в БД звичайно пасивні: вони або там є, або їх немає. БЗ, з іншого боку, активно намагається поповнити відсутню інформацію.

5.4 Необхідні умови подання знань

Однією з основних проблем, характерних для ЕС, є проблема подання знань. Це пояснюється тим, що форма подання знань впливає на характеристики і властивості системи.

Подання знань зображене на рис. 5.2. Для можливості оперування знаннями з реального світу за допомогою ПК необхідно здійснити їхнє моделювання (за аналогією з побудовою концептуальних і логічних моделей БД). При цьому необхідно відрізняти знання, призначені для обробки комп’ютером, від знань, що використовуються людиною.

При проектуванні моделі подання знань варто враховувати такі фактори, як:

- однорідність подання;
- простота розуміння.

Однорідність подання приводить до спрощення механізму керування логічним висновком і керуванням знаннями.

Простота розуміння припускає доступність розуміння подання знань і експертам, і користувачам системи. У іншому випадку ускладнюється надбання знань і їхня оцінка.

Однак виконати ці вимоги в однаковій мірі як для простих, так і складних задач досить важко. В даний час для подання знань використовують такі види моделей:

- модель на базі логіки;
- продукційна модель;
- модель семантичної мережі;
- модель, заснована на використанні фреймів і ін.

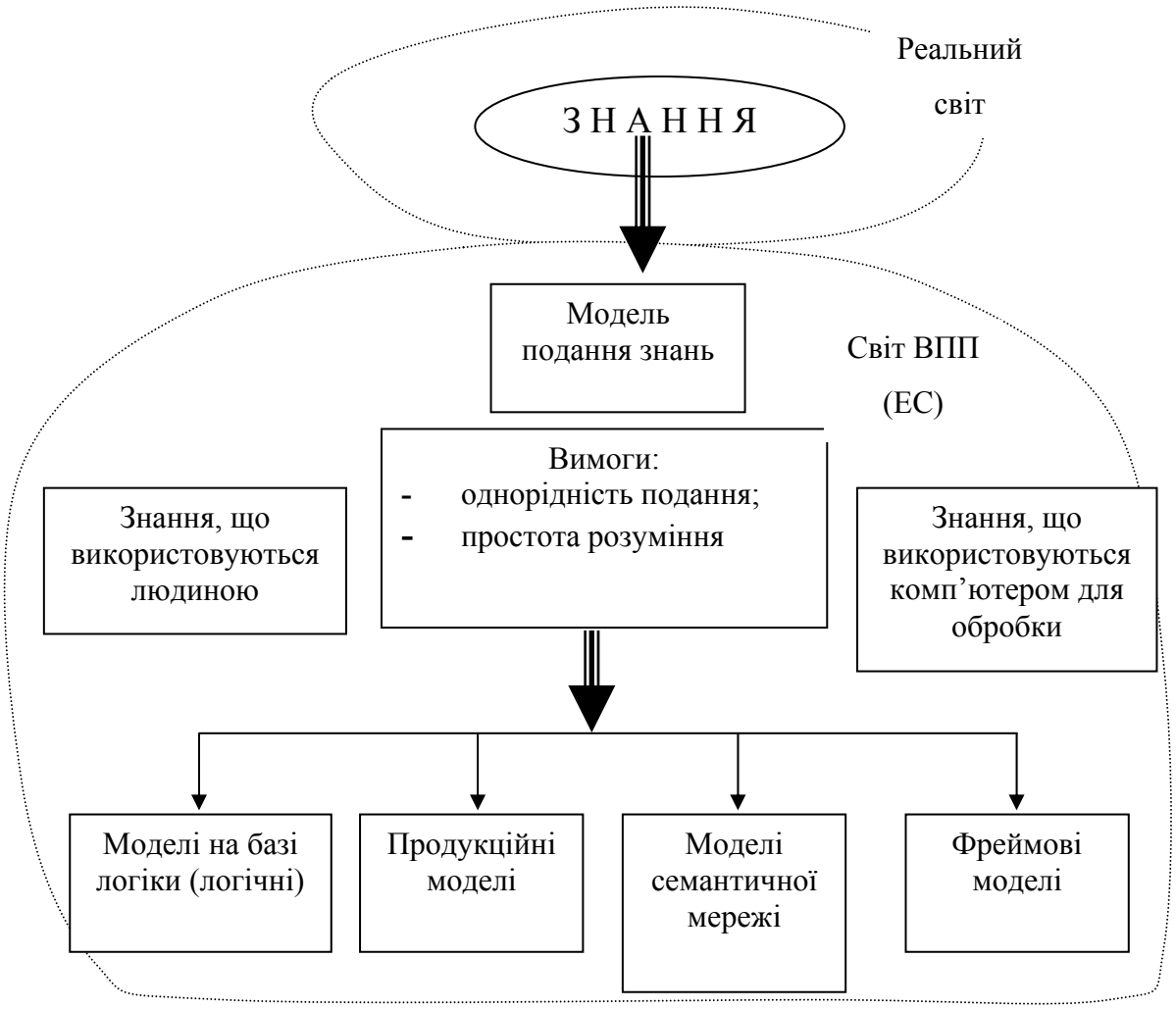


Рисунок 5.2 - Подання знань

Ілюстрацією логічної моделі є наведений вище приклад.

Основна ідея логічного підходу полягає в тому, щоб розглядати всю систему знань, необхідну для розв'язання прикладних задач, як сукупність фактів (тверджень).

Факти подаються як формули в деякій логіці (першого чи вищого порядку, багатозначній, нечіткій чи ін.) Система знань відображається сукупністю таких формул і, подана в ЕОМ, вона утворює БЗ.

Формули неподільні і при модифікації БЗ можуть лише додаватися чи видалятися. Логічні методи забезпечують розвинений апарат виведення нових фактів з тих, котрі явно подані в БЗ. Основним примітивом маніпуляції знаннями є операція виведення.

5.5 Принципи організації та функціонування ЕС

Типова статична ЕС складається з таких основних компонентів (рис. 5.3):

- вирішувача (інтерпретатора);
- робочої пам’яті (РП), яку називають також базою даних (БД);
- бази знань (БЗ);
- компонентів придбання знань;
- пояснювального компонента;
- діалогового компонента.

База даних (робоча пам'ять) призначена для збереження вихідних і проміжних даних розв’язуваної в поточний момент задачі.

База знань (БЗ) у ЕС призначена для збереження довгострокових даних, що описують розглянуту область (а не поточних даних), і правил, що описують доцільні перетворення даних цієї області.

Вирішувач використовуючи вихідні дані з робочої пам’яті і знання з БЗ, формує таку послідовність правил, що, будучи застосованими до вихідних даних, приводять до розв’язання задачі.

Компонент придбання знань автоматизує процес наповнення ЕС знаннями, здійснюваний користувачем-експертом.

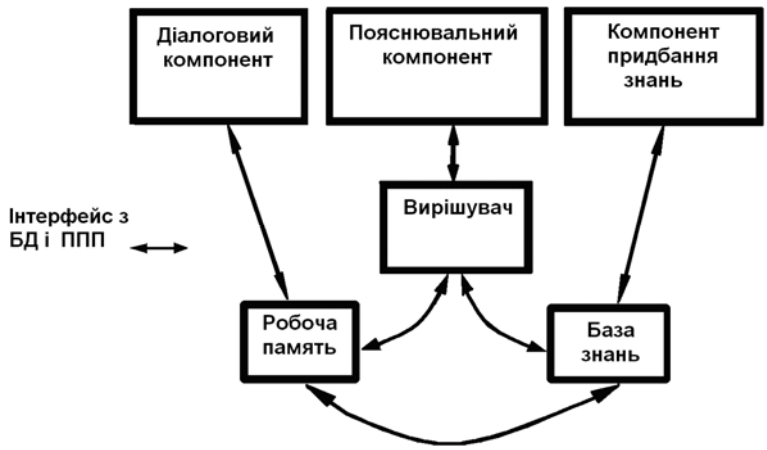


Рисунок 5.3 - Типова статична ЕС

Пояснювальний ***компонент*** пояснює, як система одержала розв’язок задачі (чи чому вона не одержала розв’язка) і які знання вона при цьому використовувала.

Діалоговий компонент орієнтований на організацію спілкування з користувачем.

У розробці ЕС беруть участь представники таких спеціальностей:

- експерт у проблемній галузі, задачі якої буде розв’язувати ЕС;
- інженер із знань - фахівець з розробки ЕС;
- програміст з розробки інструментальних засобів (ІЗ), призначених для прискорення розробки ЕС.

Експерт визначає знання (дані і правила), що характеризують проблемну галузь, забезпечує повноту і правильність введених у ЕС знань.

Інженер із знань допомагає експерту виявляти і групувати знання, необхідні для роботи ЕС; здійснює вибір того ІЗ, що найбільше підходить

для даної проблемної галузі, і визначає спосіб подання знань у цьому ІЗ; виділяє і програмує (традиційними засобами) стандартні функції (типові для даної проблемної галузі), що будуть використовуватися в правилах, які вводяться експертом.

Програміст розробляє ІЗ (якщо ІЗ розробляється заново), що містить всі основні компоненти ЕС, і здійснює його поєднання з тим середовищем, у якому воно буде використано.

Експертна система працює в двох режимах: режимі придбання знань і в режимі розв'язання.

У режимі придбання знань спілкування з ЕС здійснює експерт. У цьому режимі експерт наповнює систему знаннями, що дозволяють ЕС у режимі розв'язання самостійно розв'язувати задачі з проблемної галузі. Експерт описує проблемну галузь у вигляді сукупності даних і правил. Дані визначають об'єкти, їхні характеристики і значення, що існують в галузі експертизи. Правила визначають способи маніпулювання даними, характерні для розглянутої галузі.

Режиму придбання знань у традиційному підході до розробки програм відповідають етапи алгоритмізації, програмування і налагодження, виконувані програмістом. На відміну від традиційного підходу у випадку ЕС розробку програм здійснює не програміст, а експерт, що не володіє програмуванням.

У режимі консультації спілкування з ЕС здійснює кінцевий користувач, якого цікавить результат та спосіб його одержання. Залежно від призначення ЕС користувач може не бути фахівцем у даній проблемній галузі (у цьому випадку він звертається до ЕС за результатом, не вміючи одержати його сам), чи бути фахівцем. У режимі консультації дані про задачу користувача після обробки їхнім діалоговим компонентом надходять у робочу пам'ять. Вирішувач на основі вхідних даних з робочої пам'яті, загальних даних про проблемну галузь і правил із БЗ формує розв'язок задачі. ЕС при розв'язанні задачі не тільки виконує запропоновану послідовність операції, але і попередньо формує її. Якщо реакція системи не зрозуміла користувачу, то він може зажадати пояснення.

Структуру, наведену на рисунку 5.4, називають ***структурою динамічної ЕС***. ЕС даного типу використовуються там, де можна не враховувати змін навколишнього світу, що відбуваються за час розв'язання задачі. Перші ЕС, що одержали практичне використання, були статичними.

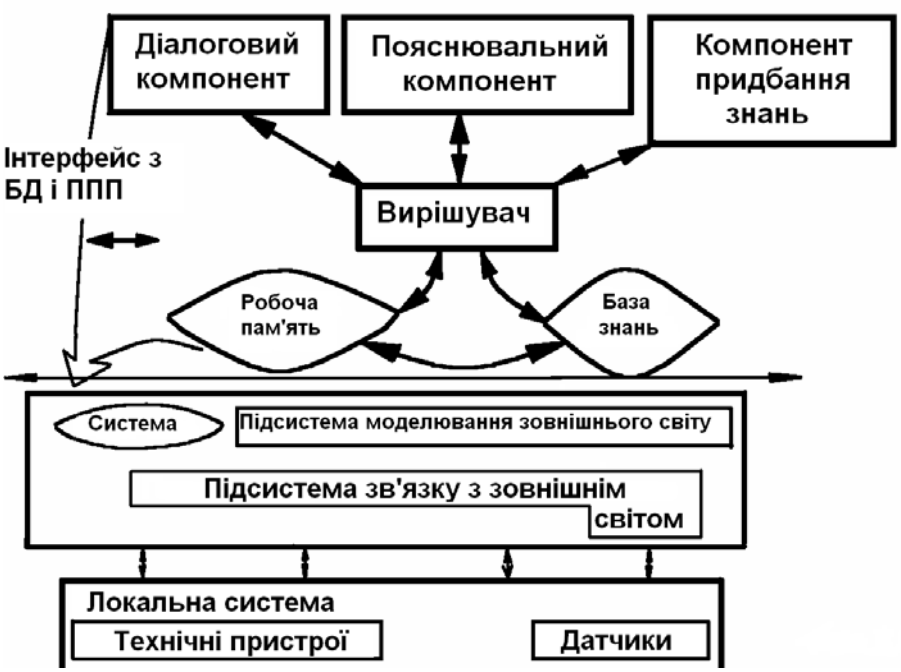


Рисунок 5.4 - Структура динамічної ЕС

На рисунку 5.4 показано, що в архітектуру динамічної ЕС порівняно зі статичною ЕС уводяться два компоненти: *підсистема моделювання* зовнішнього світу і *підсистема зв'язку* з зовнішнім світом. Остання здійснює зв'язок з зовнішнім світом через систему датчиків і контролерів. Крім того, традиційні компоненти статичної ЕС відчувають вплив істотних змін, щоб відбити тимчасову логіку подій, які відбуваються в реальному світі.

5.5.1 Етапи розробки експертних систем

Щоб розробка ЕС була можливою, необхідне одночасне виконання принаймні таких вимог:

1) існують експерти в даній галузі, що розв’язують задачу значно краще, ніж починаючі фахівці;

2) експерти сходяться в оцінці пропонованого рішення, інакше не можна буде оцінити якість розробленої ЕС;

3) експерти здатні подати і пояснити використовувані ними методи, інакше важко розраховувати на те, що знання експертів будуть "витягнуті" і вкладені в ЕС;

4) розв’язання задачі вимагає тільки міркувань, а не дій;

5) задача не повинна бути занадто важкою (тобто її розв’язок повинен займати в експерта небагато часу);

6) задача хоча і не повинна бути виражена у формальному вигляді, але все-таки повинна відноситися до досить "зрозумілої" та структурованої галузі, тобто повинні бути виділені основні поняття, відношення і відомі (хоча б експерту) способи одержання розв’язку задачі;

7) розв’язок задачі не повинен значною мірою використовувати "здоровий глузд" (тобто широкий спектр загальних зведень про світ і про спосіб його функціонування, що знає і уміє використовувати будь-яка нормальна людина), тому що подібні знання поки не вдається (у достатній кількості) вкласти в системи штучного інтелекту.

Застосування ЕС може бути виправдано одним з таких факторів:

- розв’язання задачі принесе значний ефект, наприклад економічний;
- використання людини-експерта неможливо або через недостатню кількість експертів, або через необхідність виконувати експертизу одночасно в різних місцях;
- при передачі інформації експерту відбувається неприпустима втрата часу чи інформації.

У ході робіт зі створення ЕС склалася певна технологія їхньої розробки, що включає шість таких етапів (рис. 5.5): ідентифікацію, концептуалізацію, формалізацію, виконання, тестування, досвідчену експлуатацію. На етапі ідентифікації визначаються задачі, що підлягають розв’язанню, виявляється мета розробки, визначаються експерти і типи користувачів.

На етапі концептуалізації проводиться змістовний аналіз проблемної області, виявляються використовувані поняття і їхні взаємозв'язки, визначаються методи розв’язання задач.

На етапі формалізації вибираються ІЗ і визначаються способи подання усіх видів знань, формалізуються основні поняття, визначаються способи інтерпретації знань, моделюється робота системи, оцінюється адекватність цілям системи зафіксованих понять, методів рішень, засобів подання і маніпулювання знаннями.

На етапі виконання здійснюється наповнення експертом бази знань. У зв'язку з тим, що основою ЕС є знання, даний етап є найбільш важливим і найбільш трудомістким етапом розробки ЕС. Процес придбання знань розділяють на витяг знань з експерта, організацію знань, що забезпечує ефективну роботу системи, і подання знань у вигляді, зрозумілому ЕС.

5.5.2 Подання знань в експертних системах

Прагнення виділити організацію знань у самостійну задачу викликано, зокрема, тим, що ця задача виникає для будь-якої мови подання і способи розв’язання цієї задачі є однаковими (або подібними) поза залежністю від використовуваного формалізму [62-64].

Отже, у коло питань, розв’язуваних при поданні знань, будемо включати такі:

- визначення складу знань, що подаються;
- організацію знань;
- подання знань, тобто визначення моделі подання.

Склад знань ЕС визначається такими факторами:

- проблемним середовищем;
- архітектурою експертної системи;
- потребами і цілями користувачів;
- мовою спілкування.

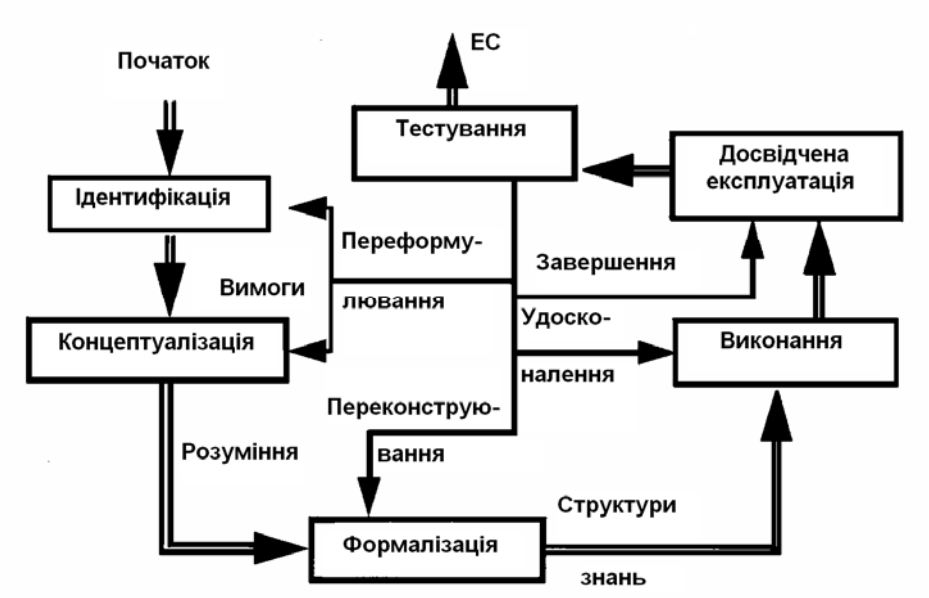


Рисунок 5.5 - Технологія розробки ЕС

Відповідно до загальної схеми статичної експертної системи для її функціонування необхідні такі знання:

- знання про процес розв’язання задачі (тобто керуючі знання), використовувані інтерпретатором;

- знання про мову спілкування і способи організації діалогу, використовувані лінгвістичним процесором (діалоговим компонентом);

- знання про способи подання і модифікації знань, використовувані компонентом придбання знань;

- підтримуючі структурні і керуючі знання, використовувані пояснювальним компонентом.

Для динамічної ЕС, крім того, необхідні такі знання:

1) знання про методи взаємодії з зовнішнім оточенням;

2) знання про модель зовнішнього світу.

Залежність складу знань від вимог користувача виявляється в такому:

- які задачі (із загального набору задач) і з якими даними бажає розв’язувати користувач;

- які кращі способи і методи розв’язання;

- при яких обмеженнях на кількість результатів і способи їхнього одержання повинна бути розв’язана задача;

- які вимоги до мови спілкування й організації діалогу;

- який ступінь спільності (конкретності) знань про проблемну галузь, доступний користувачу;

- які цілі користувачів.

Склад знань про мову спілкування залежить як від мови спілкування, так і від необхідного рівня розуміння.

5.5.3 Рівні подання і рівні детальності

Для того щоб експертна система могла керувати процесом пошуку рішення, була здатна здобувати нові знання і пояснювати свої дії, вона повинна вміти не тільки використовувати свої знання, але і мати здатність розуміти і досліджувати їх, тобто експертна система повинна мати знання про те, як подані її знання про проблемне середовище. Якщо знання про проблемне середовище назвати знаннями нульового рівня подання, то перший рівень подання містить метазнання, тобто знання про те, як подані у внутрішньому світі системи знання нульового рівня. Перший рівень містить знання про те, які засоби використовуються для подання знань нульового рівня. Знання першого рівня відіграють істотну роль при керуванні процесом розв’язання, при придбанні і поясненні дій системи. У зв'язку з тим, що знання першого рівня не містять посилань на знання нульового рівня, знання першого рівня незалежні від проблемного середовища.

Число рівнів подання може бути більше двох. Другий рівень подання містить зведення про знання першого рівня, тобто знання про подання базових понять першого рівня. Поділ знань за рівнями подання забезпечує розширення галузі застосовності системи.

5.6 Організація знань у робочій системі

Робоча пам'ять (РП) експертних систем призначена для збереження даних. Дані в робочій пам'яті можуть бути однорідні чи розділятися на рівні за типами даних.

У сучасних експертних системах дані в робочій пам'яті розглядаються як ізольовані чи як пов'язані. У першому випадку робоча пам'ять складається з множини простих елементів, а в другому - з одного чи декількох (при декількох рівнях у РП) складних елементів. При цьому складний елемент відповідає множині простих, об'єднаних у єдину сутність. Теоретично обидва підходи забезпечують повноту, але використання ізольованих елементів у складних предметних областях приводить до втрати ефективності.

Дані в РП у найпростішому випадку є константами та змінними. При цьому змінні можуть трактуватися як характеристики деякого об'єкта, а константи - як значення відповідних характеристик.

Якщо РП складається зі складних елементів, то зв'язок між окремими об'єктами вказується явно, наприклад заданням семантичних відношень.

При цьому кожен об'єкт може мати свою внутрішню структуру. Для прискорення пошуку і зіставлення дані в РП можуть бути пов'язані не тільки логічно, але й асоціативно.

5.6.1 Організація знань у базі даних

Зв’язність знань є основним способом, що забезпечує прискорення пошуку необхідних знань. Більшість фахівців прийшли до висновку, що знання варто організовувати навколо найбільш важливих об'єктів (сутностей) предметної області. Усі знання, що характеризують деяку сутність, пов'язуються і подаються у вигляді окремого об'єкта. При подібній організації знань, якщо системі потрібна була інформація про деяку сутність, то вона шукає об'єкт, що описує цю сутність, а потім вже усередині об'єкта відшукує інформацію про дану сутність. В об'єктах доцільно виділяти два типи зв'язків між елементами: зовнішні і внутрішні. Внутрішні зв'язки поєднують елементи в єдиний об'єкт і призначені для вираження структури об'єкта. Зовнішні зв'язки відбивають взаємозалежності, що існують між об'єктами в області експертизи.

Перебування бажаних об'єктів у загальному випадку доречно розглядати як двохетапний процес. На першому етапі, що відповідає процесу вибору за асоціативними зв'язками, відбувається попередній вибір у базі знань потенційних кандидатів на роль бажаних об'єктів. На другому етапі шляхом виконання операції зіставлення потенційних кандидатів з описами кандидатів здійснюється остаточний вибір шуканих об'єктів.

Операція зіставлення може використовуватися не тільки як засіб вибору потрібного об'єкта з множини кандидатів; вона може бути використана для класифікації, підтвердження, декомпозиції і корекції. Для ідентифікації невідомого об'єкта він може бути зіставлений з деякими відомими зразками.

Операції зіставлення дуже різноманітні. Звичайно виділяють такі їхні форми: синтаксичні, параметричні, семантичні і примусове зіставлення. У випадку синтаксичного зіставлення співвідносять форми (зразки), а не зміст об'єктів. Результат синтаксичного зіставлення є бінарним: зразки зіставляються чи не зіставляються. У параметричному зіставленні вводиться параметр, що визначає ступінь зіставлення. У випадку семантичного зіставлення співвідносяться не зразки об'єктів, а їхні функції. У випадку примусового зіставлення, один зразок, що зіставляється, розглядається з погляду іншого. На відміну від інших типів зіставлення тут завжди може бути отриманий позитивний результат. Питання полягає в силі примушення. Примушення можуть виконувати спеціальні процедури, що пов'язуються з об'єктами. Якщо ці процедури не в змозі здійснити зіставлення, то система повідомляє, що успіх може бути досягнутий тільки в тому випадку, коли певні частини розглянутих сутностей можна вважати зіставними.

5.6.2 Методи пошуку рішень в експертних системах

Особливості предметної галузі з погляду методів розв’язання можна характеризувати такими параметрами:

- розмір, що визначає об’єм простору, у якому потрібно шукати рішення;

- змінюваність області, характеризує ступінь змінюваності області в часі і просторі (тут будемо виділяти статичні і динамічні області);

- повнота моделі, що описує область, характеризує адекватність моделі, використовуваної для опису даної області;

- визначеність даних про розв'язувану задачу, характеризує ступінь точності і повноти даних. Точність є показником того, що предметна область з погляду розв'язуваних задач описана точними даними; під повнотою даних розуміється достатність вхідних даних для однозначного рішення задачі.

Існуючі методи розв’язання задач, використовувані в експертних системах, можна класифікувати в такий спосіб:

- методи пошуку в одному просторі - методи, призначені для використання в таких умовах: області невеликої розмірності, повнота моделі, точні і повні дані;

- методи пошуку в ієрархічних просторах - методи, призначені для роботи в областях великої розмірності;

- методи пошуку при неточних і неповних даних ;

- методи пошуку, що використовують кілька моделей, призначені для роботи з областями, для адекватного опису яких однієї моделі недостатньо.

Передбачається, що перераховані методи при необхідності повинні поєднуватися для того, щоб дозволити розв’язувати задачі, складність яких зростає одночасно по декількох параметрах.

5.7 Функціональна модель розв'язання задач експертними системами

Розглянемо методику формалізації експертних знань на прикладі створення експертних діагностичних систем (ЕДС).

Метою створення ЕДС є визначення стану об'єкта діагностування (ОД) і наявних у ньому несправностей.

Станами ОД можуть бути: справний, несправний, роботоздатний. Несправностями, наприклад, радіоелектронних ОД є обрив зв'язку, замикання провідників, неправильне функціонування елементів і т.д.

Число несправностей може бути досить великим (кілька тисяч). В ОД може бути одночасно кілька несправностей. У цьому випадку говорять, що несправності кратні.

Введемо такі означення. Різні несправності ОД виявляються в зовнішньому середовищі інформаційними параметрами. Сукупність

значень інформаційних параметрів визначає «інформаційний образ» (ІО) несправності ОД. ІО може бути повним, тобто містити всю необхідну інформацію для постановки діагнозу, або, відповідно, неповним. У випадку неповного ІО постановка діагнозу носить імовірнісний характер.

Основою для побудови ефективних ЕДС є знання експерта для постановки діагнозу, записані у вигляді інформаційних образів, і система подання знань, що вбудовується в інформаційні системи забезпечення функціонування і контролю ОД, які інтегруються з відповідною технічною апаратурою.

Для опису своїх знань експерт за допомогою інженера із знань повинен виконати таке.

1. Виділити множини усіх несправностей ОД, які повинна розрізняти ЕДС.

2. Виділити множини інформативних параметрів, значення яких дозволяють розрізнити кожну несправність ОД і поставити діагноз з деякою імовірністю.

3. Для обраних параметрів варто виділити інформативні значення або інформативні діапазони значень, що можуть бути як кількісними, так і якісними. Наприклад, точні кількісні значення можуть бути записані: затримка 25 нс, затримка 30 нс і т.д. Кількісний діапазон значень може бути записаний: затримка 25-40 нс, 40-50 нс, 50 нс і більше. Якісний діапазон значень може бути записаний: індикаторна лампа світиться яскраво, світиться слабко, не світиться.

Для більш зручного подальшого використання якісний діапазон значень може бути закодований, наприклад, у такий спосіб:

- світиться яскраво `P1 = +++` (або `P1 = 3`),
- світиться слабко `P1 = ++` (або `P1 = 2`),
- не світиться `P1 = +` (або `P1 = 1`).

Процедура одержання інформації щодо кожного з параметрів визначається індивідуально в кожній конкретній системі діагностування. Ця процедура може полягати в автоматичному вимірі параметрів у ЕДС, у ручному вимірюванні параметра за допомогою приладів, якісному визначенні параметра, наприклад, світиться слабко і т.д.

4. Процедура створення повного або неповного ІО кожної несправності в алфавіті значень інформаційних параметрів може бути визначена в такий спосіб. Складаються діагностичні правила, що визначають ймовірний діагноз на основі різних сполучень діапазонів значень обраних параметрів ОД. Правила можуть бути записані в різній формі. Нижче наведена форма запису правил у вигляді таблиці 5.1.

Таблиця 5.1 - Діагностичні правила

Номер	P1	P2	P3	Діагноз	Імовірність діагнозу	Примітки
1		+++		Несправний блок A1	0.95	
2	12-15	+		Несправний блок A2	0.80	

Для запису правил з урахуванням змін за часом варто ввести ще один параметр P0 - час (ще один стовпець у таблиці). У цьому випадку діагноз може ставитися на основі декількох рядків таблиці, а в графі Примітки можуть бути зазначені використані тести. Діагностична таблиця в цьому випадку подана в таблиці 5.2.

Таблиця 5.2 - Динамічні діагностичні правила

Номер	P0	P1	P2	P3	Діагноз	Імовірність діагнозу	Примітки
1	12:00	+	+	+			тест T1
2	12:15	++	++	+	Несправний блок A3	0.90	

Для запису послідовності проведення тестових процедур і задання обмежень (якщо вони є), на їхнє проведення може бути запропонований аналогічний механізм. Механізм запису послідовності проведення тестових процедур у вигляді правил реалізується, наприклад, у такий спосіб:

ЯКЩО: P2 = 1

ТО: тест = T1, T3, T7

де `T1, T3, T7` - тестові процедури, що подаються на ОД при активізації відповідної продукції.

У сучасних ЕДС застосовуються різні стратегії пошуку рішення і постановки діагнозу, що дозволяють визначити необхідні послідовності тестових процедур. Однак пріоритет у ЕС віддається насамперед знанням і досвіду, а лише потім логічному висновку.

Дана методика буде застосована в наступній лекції при створенні експертної системи керування технологічним процесом.

5.8 Приклади експертних систем

Для початку зробимо короткий екскурс в історію створення ранніх і найбільш відомих ЕС. У більшості цих ЕС як СПЗ використовувалися

системи продукцій (правила) і прямий ланцюг міркувань. Медична ЕС MYCIN розроблена в Стенфордському університеті в середині 70-х років для діагностики і лікування інфекційних захворювань крові. MYCIN у даний час використовується для навчання лікарів. ЕС DENDRAL розроблена в Стенфордському університеті в середині 60-х років для визначення топологічних структур органічних молекул.

Система виводить молекулярну структуру хімічних речовин за даними мас-спектрометрії і ядерного магнітного резонансу молекул. ЕС PROSPECTOR розроблена в Стенфордському університеті в 1974-1983 роках для оцінки геологами потенційної рудоносності району.

Система містить більше 1000 правил і реалізована на INTERLISP. Програма порівнює спостереження геологів з моделями різного роду покладів руд. Програма втягує геолога в діалог для витягу додаткової інформації. У 1984 році вона точно пророчила існування молібденового родовища, оціненого в багатомільйонну суму.

Розглянемо експертну систему діагностування (ЕСД) цифрових і цифро-аналогових пристроїв, у якій використовувалися системи продукцій і фрейми, а також прямий і зворотний ланцюги міркувань одночасно. Як об'єкт діагностування (ОД) у ЕСД можуть використовуватися цифрові пристрої (ЦП), БІС, цифро-аналогові пристрої.

На рис. 5.6 показано, що така ЕСД працює разом з автоматизованою системою контролю і діагностування (АКД), що подає в динаміці впливи на ОД (десятки, сотні і тисячі впливів у секунду), аналізує вихідні реакції і дає висновок: придатний або не придатний. У випадку, якщо реакція ОД не відповідає еталонним значенням, то підключається заснована на знаннях підсистема діагностування. ЕСД робить запит значення сигналів у певних контрольних точках і веде оператора за схемою ОД, рекомендуючи йому зробити виміри у певних контрольних точках або підтвердити проміжний діагноз, і в результаті приводить його до місця несправності. Вихідними даними для роботи ЕСД є результати машинного моделювання ОД на етапі проектування. Ці результати моделювання передаються в ЕСД на магнітних носіях у вигляді тисяч продукційних правил. Рух по контрольних точках здійснюється на основі моделі, записаної у вигляді мережі фреймів для ОД.

Така ЕСД не була б інтелектуальною системою, якби вона не накопичувала досвід. Вона запам'ятовує знайдену несправність для даного типу ОД. Наступного разу при діагностуванні несправності ОД цього типу вона пропонує перевірити спочатку цю несправність, якщо реакція ОД говорить про те, що така несправність можлива. Так роблять досвідчені майстри радіоелектронної апаратури (РЕА), що знають «слабкі» місця в конкретних типах РЕА і перевіряють їх у першу чергу. ЕСД накопичує імовірнісні знання про конкретні несправності з метою їхнього використання при логічному висновку. При русі по дереві пошуку рішень

на черговому кроці використовується критерій - максимум відношення імовірності (коефіцієнта впевненості) постановки діагнозу до трудомісткості розпізнавання несправності. Коефіцієнти впевненості автоматично коректуються під час роботи ЕСД при кожному підтвердженні або непідтвердженні діагнозу для конкретних ситуацій діагностування. Трудомісткості елементарних перевірок спочатку задаються експертом, а потім автоматично коректуються в процесі роботи ЕСД.

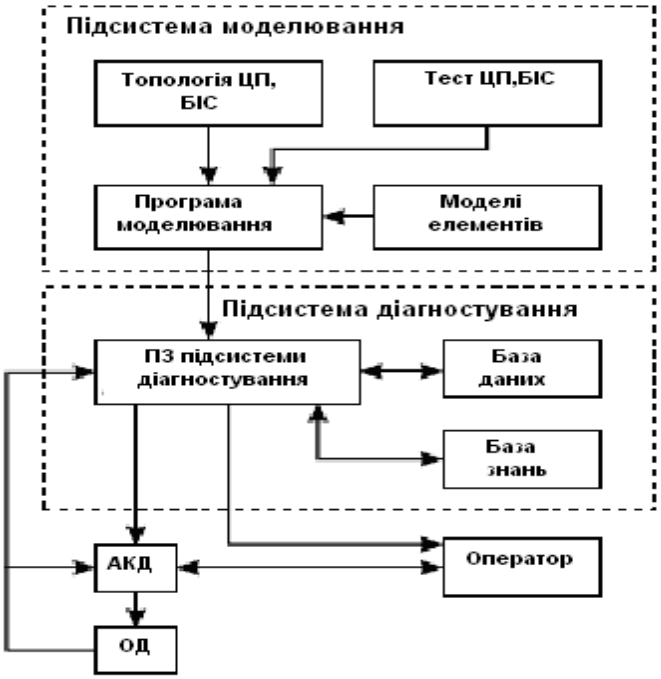


Рисунок 5.6 - Загальна структура експертної системи діагностування

ЕСД не була реалізована у вигляді ІРС з економічних міркувань. Невелика серійність апаратури, що перевіряється, недостатня уніфікація і дешева робоча сила перешкодили реалізувати повністю автоматичне діагностування.

Серед сучасних комерційних систем хочеться виділити експертну систему - оболонку G2 американської фірми Gensym (USA) як неперевершену експертну комерційну систему для роботи з динамічними об'єктами. Робота в реальному часі з малим часом відповіді часто необхідна при аналізі ситуацій у корпоративних інформаційних мережах, на атомних реакторах, у космічних польотах і безлічі інших задач. У цих задачах необхідно приймати рішення протягом мілісекунд із моменту виникнення критичної ситуації. ЕС G2, призначена для розв’язання таких задач, відрізняється від більшості динамічних ЕС такими характерними властивостями, як:

- робота в реальному часі з розпаралеленням процесів міркувань;

- структурований природно-мовний інтерфейс із керуванням через меню й автоматичною перевіркою синтаксису;
- зворотний і прямий висновок, використання метазнань, сканування і фокусування;
- інтеграція підсистеми моделювання з динамічними моделями для різних класів об'єктів;
- структурування БЗ, спадкування властивостей, розуміння зв'язків між об'єктами;
- бібліотеки знань є ASCII-файлами і легко переносяться на будь-які платформи і типи ЕОМ;
- розвинутий редактор для супроводу БЗ без програмування, засобу трасування і налагодження БЗ;
- керування доступом за допомогою механізмів авторизації користувача і забезпечення бажаного погляду на додаток;
- гнучкий інтерфейс оператора, що включає графіки, діаграми, кнопки, піктограми і т.п.;
- інтеграція з іншими додатками (по TCP/IP) і базами даних, можливість вилученої і багатокористувацької роботи.

Як приклад швидкодіючої системи для відстеження стану корпоративної інформаційної мережі (КІС) можна навести засновану на знаннях систему моніторингу OMEGAMON фірми Candle (IBM з 2004 р.). OMEGAMON - типовий представник сучасних експертних мультиагентних динамічних систем, що працюють у реальному часі. OMEGAMON дозволяє за лічені хвилини ввести і налагодити правила моніторингу позаштатних ситуацій для об'єктів КІС. Правило (situation) записується як продукція. Логічний висновок у такій ЕС реалізований за допомогою механізму policy, що забезпечує побудову ланцюгів логічного висновку на основі situations. На рис. 5.7 наведено один з інтерфейсів для заповнення БЗ у ЕС OMEGAMON. На цьому рисунку показана ситуація, що визначає критичну кількість повідомлень у чергах транспортної системи IBM WebSphere MQ (MQSeries).

На рисунку 5.8 показано основні компоненти системи OMEGAMON:

- сервер збору інформації від агентів CandleManagementServer (CMS);
- сервер відображення результатів, оповіщення користувачів і настроювання моніторингу КІС CandleNetPortal Server (CNP) зі своїми клієнтами;
- Candle Management Workstation (CMW) - робоча станція адміністратора OMEGAMON;
- Managed Systems - комп'ютери КІС, на яких працюють агенти.

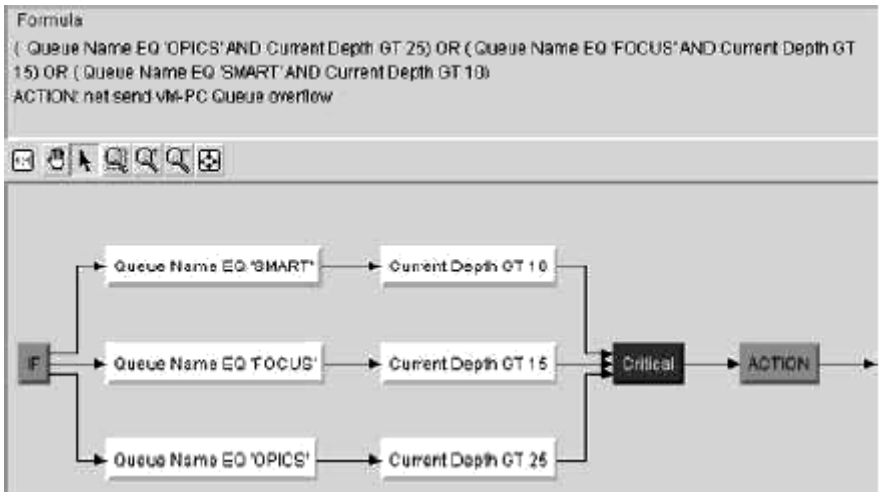


Рисунок 5.7 - Інтерфейс OMEGAMON для заповнення БЗ

Агенти OMEGAMON працюють на контрольованих системах (Managed Systems) як першокласні шпигуни: вони непомітні з погляду використання CPU і оперативні при моніторингу з погляду часу постачання своїх повідомлень у центр (CMS). Вони фіксують критичну ситуацію і забезпечують реакцію (ACTION) менше, ніж за 1 секунду. Усе визначається тим інтервалом моніторингу, що задається експертом на основі своїх інтуїтивних знань. Як ACTION при визначенні ситуацій можна використовувати різні типи дій: посилку поштових повідомлень і sms фахівцям супроводу, посилку інформації в інші системи, виконання системних команд і т.д.

Кількість об'єктів моніторингу (комп'ютерів КІС) може сягати декількох сотень, і на кожному об'єкті може бути кілька сотень контрольованих параметрів. Кількість платформ (типів операційних систем), на яких працюють агенти, перевищує 30, починаючи від OS/390,OS/400, далі різні UNIX-платформи (HP_UX, AIX, Solaris) і закінчуючи Windows. На одному сервері може працювати кілька агентів, наприклад, для моніторингу WebSphere MQ (MQSeries), WebSphere Application Server, DB-2 і HP_UNIX одночасно.

Сервери CMS і CNP-servers можуть працювати на одному виділеному сервері, як правило, на базі операційної системи Windows. Настроювання ситуацій (situations) і механізмів логічного висновку (policy) виконуються на робочому комп'ютері адміністратора через CNP-client. Для тільки що створеної ситуації ви натискаєте кнопку Apply і моментально бачите відображення ACTION через CNP-client, через пошту і т.д.

Варто підкреслити, що заснована на знаннях система моніторингу OMEGAMON - це досить ефективна система керування обчислювальними ресурсами, надійний і незамінний помічник у пошуках рішень з оперативного усунення критичних і важких для діагностування ситуацій,

при аналізі інформаційних потоків, аналізі продуктивності і настроюванні КІС.

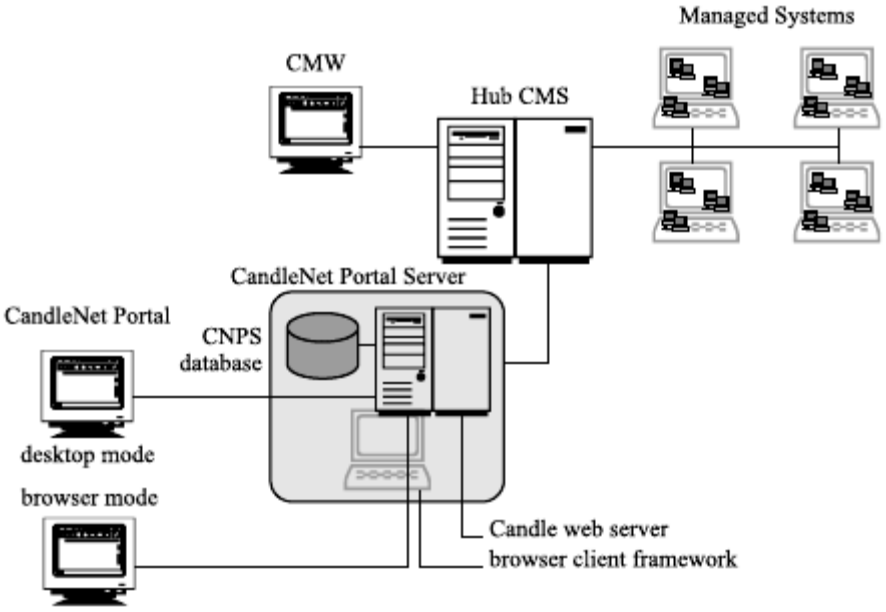


Рисунок 5.8 - Структурна схема ЕС OMEGAMON

5.9 Принципи формування бази даних і знань

Придбання знань реалізується за допомогою двох функцій: одержання інформації ззовні і її систематизації. При цьому залежно від здатності системи навчання до логічних висновків можливі різні форми придбання знань, а також різні форми одержуваної інформації. Форма подання знань для їхнього використання визначається усередині системи, тому форма інформації, яку вона може приймати, залежить від того, які здібності має система для формалізації інформації до рівня знань. Якщо система, що навчається, зовсім позбавлена такої здатності, то людина повинна заздалегідь підготувати все, аж до формалізації інформації, тобто чим вищі здатності машини до логічних висновків, тим менші навантаження на людину.

5.9.1 Методи придбання знань

Функції, необхідні системі, що навчається, для придбання знань, розрізняються залежно від конфігурації системи. Надалі при розгляді систем інженерії знань передбачається, що існує система з конфігурацією, показаною на рисунку 5.9, що включає базу знань і механізм логічних висновків, який використовує ці знання при розв’язанні задач. Якщо база знань поповнюється знаннями про стандартну форму їхнього подання, то цими знаннями також можна скористатися. Отже, від функцій навчання потрібно перетворення отриманої ззовні інформації в знання і поповнення ними бази знань.

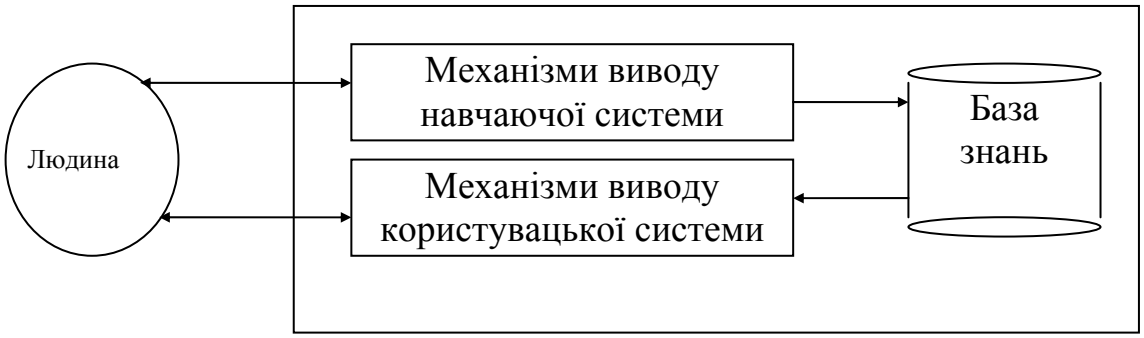


Рисунок 5.9 - Базова структура систем обробки знань

Можна запропонувати таку класифікацію систем придбання знань, яка буде опиратися на здатність системи до сприйняття знань у різних форматах, що якісно різняться між собою і за здатністю до формалізації (рисунок 5.10).

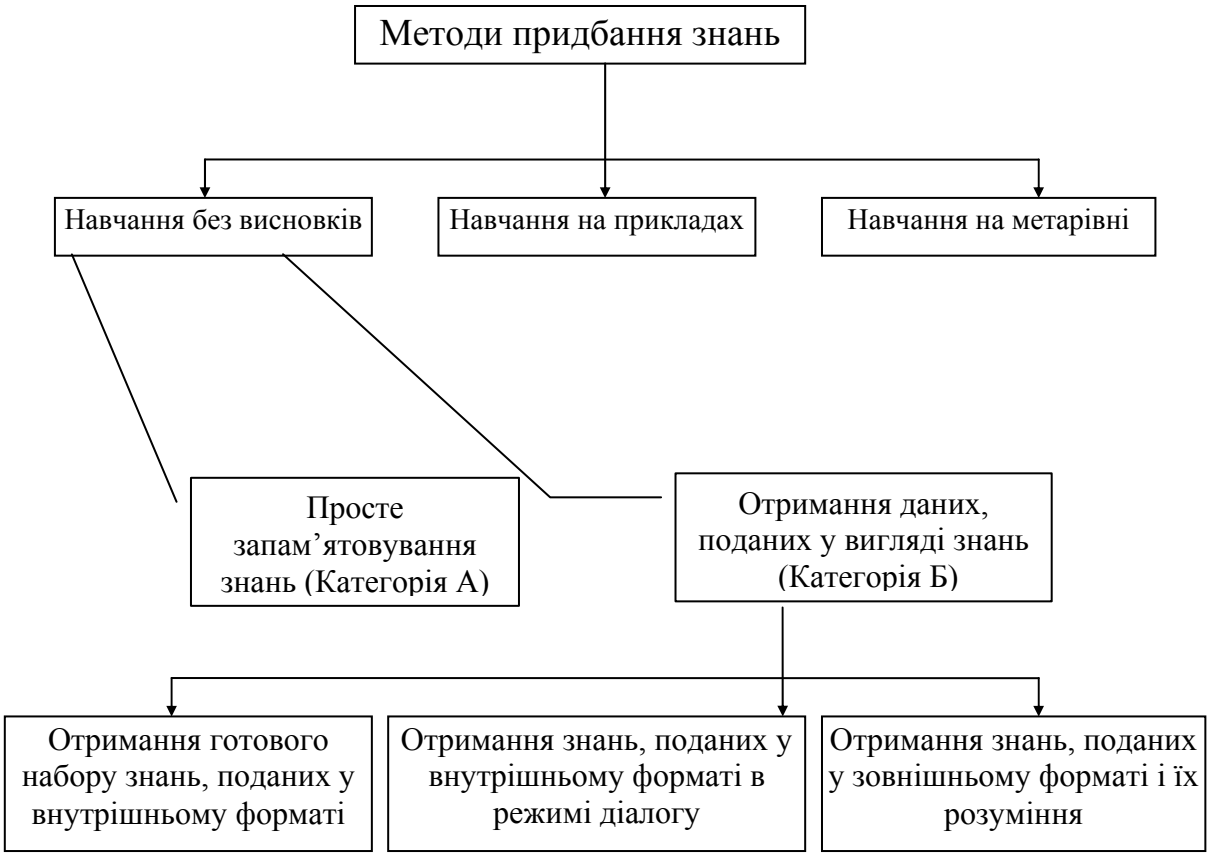


Рисунок 5.10 - Класифікація методів придбання знань.

5.9.2 Методи роботи зі знаннями

Категорію А можна назвати навчанням без висновків або механічним запам'ятовуванням, це простий процес одержання інформації, при якому необов'язкові функції висновків, а отримана інформація у

вигляді програм або даних використовується для розв’язання задач у незмінному вигляді. Іншими словами, це спосіб одержання інформації, характерний для існуючих комп'ютерів.

Категорія Б - це одержання інформації ззовні, представленої у формі знань, тобто у формі, яку можна використовувати для висновків. Системі, що навчається, необхідно мати функцію перетворення вхідної інформації у формат, зручний для подальшого використання і включення в базу знань.

5.9.2.1 Навчання без висновків

Придбання знань на цьому етапі відбувається в найбільш простій формі: це знання, попередньо підготовлені людиною у внутрішньому форматі, якими є більшість спеціальних знань, споконвічно заданих в експертних системах. У випадку прикладних систем інженерії знань необхідно перетворити спеціальні знання з якої-небудь області в машинний формат, але для цього потрібний посередник, який добре знає як проблемну область, так і інженерію знань. Таких посередників називають інженерами із знань. У загальному випадку для заміни функції посередника можна використовувати і спеціальні підпрограми. Тобто необхідно мати функції висновків досить високого рівня, але можна обмежитися і висновками на порівняно низькому рівні, а інше довірити людині — у цьому і полягає придбання знань у діалозі. Прикладом служить добре відома система TEIRESIAS. Це система-консультант в області медицини, розроблена на базі системи MYCIN. Фахівці в проблемній області є викладачами системи, що навчається, а учень — система інженерії знань — вивчає відповіді на поставлені задачі і коректує ті правила в базі знань, що раніше приводили до помилок. Для підготовки знань в експертній системі необхідні допоміжні засоби типу редактори знань, причому в процесі придбання знань у діалозі не тільки редагуються окремі правила і факти, але і заповнюються недоліки існуючих правил, тобто ведеться редагування бази знань.

Якщо знання задані в зовнішньому форматі, наприклад природною мовою, то варто перетворити них у внутрішній формат. Для цього необхідно розуміти зовнішнє подання, тобто природна мова, графічні дані і т.п. Фактично придбання знань і їхнє розуміння тісно пов'язані. Проблема розуміння зводиться не тільки до перетворення структури пропозицій — необхідно одержати формат, зручний для застосування.

Аналогічна проблема — перетворення у внутрішній формат порад, підказок щодо розв’язання задач, що називається «операціоналізацією» знань. У цьому полягає центральна проблема штучного інтелекту; вона, зокрема, вивчає перетворення порад, підказок, поданих у термінах проблемної області, у процедури. Наприклад, система UNDERSTAND виконує операціоналізацію подання задачі про ханойську вежу

англійською мовою шляхом побудови відповідних станів і операцій.

5.9.2.2 Придбання знань на мета-рівні

Вище було розглянуте навчання на об'єктному рівні, а ще більш складна проблема - придбання знань на мета-рівні, тобто знань, основою яких є інформація з керування розв’язанням задач з використанням знань на об'єктному рівні. Для знань на мета-рівні поки не встановлені ні форми подання і використання, ні зв'язок зі знаннями на об'єктному рівні, ні інша техніка їхньої систематизації. Оскільки не визначена форма їхнього подання з погляду використання, то важко говорити про придбання знань на мета-рівні. Проте з цією проблемою пов'язано багато надій в інженерії знань.

5.9.2.3 Придбання знань із прикладів

Метод придбання знань із прикладів відрізняється від попереднього методу, тим, що тут виконується збір окремих фактів, їхнє перетворення й узагальнення, а тільки потім вони будуть використані як знання. І відповідно від рівня складності системи висновку в системі будуть виникати різні за ступенем спільності і складності знання. Необхідно також згадати про те, що цей метод придбання знань майже не знайшов практичного застосування, це може бути пов'язано з тим, що вхідна інформація являє собою несистематизований набір даних і для їхньої обробки необхідна наявність у системі великого об’єму знань з конкретної галузі.

Порівняно з попереднім методом придбання знань, цей метод має великий ступінь свободи і відповідно необхідно описати загальні положення цього принципу.

1. Мови подання. Навчання по прикладах — це процес збору окремих фактів, їхнє узагальнення і систематизація, тому необхідна уніфікована мова подання прикладів і загальних правил. Ці правила, будучи результатом навчання, повинні стати об'єктами для використання знань, тому й утворять мову подання знань. І навпаки, мова подання знань повинна враховувати і визначати зазначені вище умови придбання знань.

2. Способи опису об'єктів. У випадку навчання по прикладах з описів окремих об'єктів створюються ще більш загальні описи об'єктів деякого класу, при цьому виникає важлива проблема: як описати даний клас об'єктів. У повному класі деяких об'єктів варто визначити менший клас об'єктів, що мають загальну властивість (об'єкти тільки в цьому класі мають задану властивість), але в дійсності простіше визначити список об'єктів і переконатися, що всі об'єкти в ньому мають загальну властивість. Для деякого типу задач можна ефективно використовувати помилкові приклади або контрприклади, які переконливо показують, що дані об'єкти не входять у цей клас. Ілюстрацією застосування контрприкладів може

служити поняття «майже те».

3. Правила узагальнення. Для збору окремих прикладів і створення загальних правил необхідні правила узагальнення. Запропоновано кілька способів їхнього опису: заміна постійних атрибутів мови на змінні, виключення описів з обмеженим застосуванням і т.п. Очевидно, що ці способи тісно пов'язані з мовою подання знань.

4. Керування навчанням. У процесі навчання на прикладах можна застосовувати різні стратегії структуризації інформації і необхідно керувати цим процесом у відповідь на вхідні дані. Існують два класичних методи: метод «вверх», при якому послідовно вибираються і структуруються окремі повідомлення, і метод «униз», при якому спочатку висувається гіпотеза, а потім вона коректується в міру надходження інформації. На практиці ці методи комбінуються, хоча керування навчанням з максимальним ефектом не така вже й проста проблема.

При вивченні методу придбання знань на прикладах можна виділити такий ряд методів:

1. Параметричне навчання;
2. Навчання за індукцією;
3. Навчання за аналогією.

Параметричне навчання. Найбільш проста форма навчання по прикладах або спостереженнях полягає у визначенні загального виду правила, яке повинно стати результатом висновку, і наступного корегування вхідних у це правило параметрів залежно від даних. При цьому використовують психологічні моделі навчання, системи керування навчанням і інші методи.

Прикладом системи, що навчається, цієї категорії в галузі штучного інтелекту є система Meta-Dentral. Ця система виводить нові правила шляхом корекції правил продукції у процесі навчання або на основі вихідних даних, параметричне навчання в ній подане в трохи специфічному виді, але усе-таки вона відноситься до зазначеної вище категорії, оскільки в системі задана основна структура знань, що коректується послідовно за окремими даними.

Яскравим прикладом застосування цього методу придбання знань можуть також служити системи розпізнавання образів. У них ясно видно основний принцип цього методу - у ході навчання нейронна мережа автоматично за визначеними заздалегідь законами коректує ваги зв'язків між елементами і значеннями самих елементів.

Метод навчання за індукцією. Серед усіх форм навчання необхідно особливо виділити навчання на основі висновків за індукцією - це навчання з використанням висновків високого рівня, як і при навчанні за аналогією. У процесі цього навчання шляхом узагальнення сукупності наявних даних виводяться загальні правила. Можливе навчання з викладачем, коли вхідні дані задає людина, яка спостерігає за станом

системи, що навчається, і навчання без викладача, коли дані надходять у систему випадково. І в тому, і в іншому випадку висновки можуть бути різними. Вони мають і різний ступінь складності залежно від того, чи задаються тільки коректні дані або в тому числі і некоректні дані і т.п. Так чи інакше, навчання цієї категорії включає відкриття нових правил, побудову теорій, створення структур і інші дії, причому модель теорії або структури, які варто створити, заздалегідь не задаються, тому їх необхідно розробити так, щоб можна було пояснити всі правильні дані і контрприклади.

Індуктивні висновки можливі у випадку, коли подання результату висновку частково визначається з поданням вхідної інформації. Останнім часом звертають на себе увагу програми генерації програм за зразком з використанням індуктивних висновків.

Як уже було сказано, індуктивний висновок — це висновок із заданих даних пояснюючого їх загального правила. Наприклад, нехай відомо, що є деякий багаточлен від однієї змінної. Давайте подивимося, як виводиться *f(x)*, якщо послідовно задаються як дані пар значень (0, *f(0)*), (1, *f(1)*), ... Спочатку задається (0, 1), і природно, що їх зміст виведе постійну функцію *f(x)*=1. Потім задається (1, 1), ця пара задовольняє запропоновану функцію *f(x)*=1. Отже в цей момент немає необхідності змінювати висновок.

Нарешті, задається (2, 3), що погано погоджується з нашим висновком, тому відмовимося від нього і після декількох спроб і помилок виведемо нову функцію f(x)==x^2-x+1, що задовольняє всі задані дотепер факти (0, 1), (1, 1), (2, 3). Далі ми переконаємося, що ця ж функція задовольняє факти (3, 7), (4, 13), (5, 21) ..., тому немає необхідності змінювати цей висновок. Таким чином, з послідовності пар, змінна - функція можна вивести багаточлен другого ступеня. Грубо говорячи, такий метод висновку можна назвати індуктивним.

Як видно з цього прикладу, при висновку, в кожен момент часу порозуміваються всі дані, отримані до цього моменту. Зрозуміло, дані, отримані пізніше, уже можуть і не задовольняти цей висновок. У таких випадках доводиться змінювати висновок. Отже, у загальному випадку індуктивний висновок – це необмежено довгий процес. І це не дивно, якщо згадати процес освоєння людиною мов, процес удосконалювання програмного забезпечення і т.п.

Для точного визначення індуктивного висновку необхідно уточнити:

1) множину правил-об'єктів висновку,
2) метод подання правил,
3) спосіб показу прикладів,
4) метод висновку,
5) критерій правильності висновку.

Як правила-об'єкти можна розглядати головним чином індуктивні

функції, формальні мови, програми і т.п. Крім того, ці правила можуть бути подані у вигляді машини Тьюрінга для обчислення функцій, граматики мов, операторів Прологу й іншим способом. Машина Тьюрінга - це математична модель комп'ютера, вона в принципі може вважатися програмою. У випадку, коли об'єктом висновку є формальна мова, вона сама визначає правила, а його граматика — метод подання правил, тому говорять про граматичний висновок.

Для показу прикладів функції *f* можна використовувати послідовність пар *(x,f(x))* вхідних і вихідних значень так, як зазначено вище, послідовність дій машини Тьюрінга, що обчислює й інші дані. Задання машині висновків пари вхідних і вихідних значень *(x, f(x))* функції *f* відповідає заданню системі автоматичного синтезу програм вхідних значень *x* і вихідних значень *f(x)*, що повинні бути отримані програмою обчислення *f* у відповідь на *x.* У цьому сенсі автоматичний синтез програм по прикладах також можна вважати індуктивним висновком функції *f.* Формальні мови — це безліч слів; тому, наприклад, для мови *L* можна розглядати слова типу слів, належних і неналежних цій мові. Перші назвемо позитивними, а другі — негативними даними. Іншими словами, є два способи показу прикладів формальної мови: за допомогою позитивних і негативних даних. Коли об'єктом служать самі програми, тоді те ж саме можна говорити про функції мови Лісп, але для Прологу показ прикладів здійснюється у вигляді фактів. Наприклад, (3>4, істина), (2<=1, неправда). У цьому випадку позитивним даним відповідають дані з атрибутом «істина», а негативним — дані з атрибутом «неправда».

Висновок реалізується завдяки необмеженому повторенню основного процесу: запит вхідних даних → припущення → вихідні дані.

Іншими словами, при висновку послідовно одержують приклади як вхідні дані, обчислюють припущення на даний момент і видають результат обчислень. Припущення в кожен момент часу засновано на обмеженому числі прикладів, отриманих дотепер, тому звичайно як метод висновку використовують машину Тьюрінга, що обчислює припущення за обмеженим числом прикладів. Таку машину назвемо машиною висновків.

З огляду на те, що індуктивний висновок, як уже було відзначено, це необмежено триваючий процес, критерієм правильності висновку, як правило, вважають поняття ідентифікації в межах. Це поняття введене Глодом, воно використовується майже завжди в теорії індуктивних висновків. Говорять, що машина висновку ***M*** ідентифікує в межах правило ***R*,** якщо при показі прикладів ***До*** послідовність вихідних даних, що генеруються ***M,*** збігається до деякого подання ***m***, а саме: усі вихідні дані, починаючи з деякого моменту часу, збігаються з ***m***, при цьому ***m*** називають правильним поданням ***K***. Крім того, говорять, що безліч правил ***Г*** дозволяє зробити індуктивний висновок, якщо існує деяка машина висновків ***M,*** що ідентифікує в межі будь-яке правило ***До*** з множини ***Г***.

Зверніть увагу на те, що слова «дозволяє зробити індуктивний висновок» не мають змісту для єдиного правила, а відносяться тільки до множини правил.

Навчання за аналогією. Придбання нових понять можливо шляхом перетворення існуючих знань, схожих на ті, котрі збираються одержати. Це важлива функція, яку називають навчанням на основі висновків за аналогією або просто навчанням за аналогією. У нашому житті багато прикладів, коли нові поняття або технічні прийоми здобуваються за допомогою аналогії.

Висновки за аналогією - один з важливих об'єктів дослідження штучного інтелекту, найбільш цікаві результати тут отримані П. Уінстоном. Він використовує висновки за аналогією, ґрунтуючись на такій гіпотезі: «Якщо дві ситуації подібні за декількома ознаками, то вони подібні і ще за однією ознакою». Подібність двох ситуацій розпізнається шляхом виявлення найкращих збігів за найбільш важливими ознаками.

Це метод висновків, при яких виявляється подоба між декількома заданими об'єктами; завдяки переносові фактів і знань, справедливих для одних об'єктів, на основі цієї подоби до зовсім інших об'єктів або визначається спосіб розв’язання задач, або пророкуються невідомі факти і знання. Отже, коли людина зіштовхується з невідомою задачею, вона спочатку використовує цей природний метод висновку.

5.9.2.4 Напрямок дослідження аналогії

Одна з найважливіших проблем інженерії знань — придбання знань. Під придбанням тут розуміється одержання знань у вигляді, придатному для їхнього використання комп'ютерами, тому багато дослідників указують, що ключем до знань є теорія і методологія машинного навчання. У загальному випадку машинне навчання включає придбання нових декларативних знань, систематизацію і збереження нових знань, а також виявлення нових фактів. Серед зазначених форм навчання аналогія, про яку буде йти далі мова, пов’язана, зокрема, із проблемою машинного виявлення нових фактів.

Під новими фактами розуміються факти, що дедуктивно не виводяться з деяких існуючих знань. Одержання нових знань також розглядалося вище у відношенні до індуктивного висновку. У загальному випадку при індуктивних висновках за заданими даними створюється гіпотеза, їх пояснююча, а за допомогою дедукції з цієї гіпотези можна вивести нові факти. З іншого боку, за аналогією нові факти пророкуються шляхом використання деяких перетворень уже відомих знань.

Індукція й аналогія вкрай необхідні при обробці інтелектуальної інформації, і тому бажано викласти основи їхнього спільного застосування. Шапіро ввів строгу формалізацію індуктивних висновків у частині

висновку моделей з використанням логіки предикатів першого порядку; у теорії індуктивних висновків є помітні успіхи.

З метою огляду досліджень аналогії, проведених дотепер, виділимо два типи аналогії: для розв'язання задач і для пророкувань. Аналогія першого типу застосовується головним чином для підвищення ефективності розв'язання задач, що, узагалі говорячи, можна вирішити і без аналогії. Наприклад, завдяки використанню рішень аналогічних задач в областях програмування і доведенню теорем можна прийти до висновків про програми або доведення. З іншого боку, використовуючи аналогію для пророкувань, завдяки перетворенню знань на основі подоби між об'єктами можна зробити висновок про те, що, можливо, справедливі нові факти. Наприклад, якщо об'єктами аналогії є якась система аксіом, то знаннями можуть бути теореми, справедливі в цій системі. При цьому, використовуючи схожість між системами аксіом, можна перетворити теорему в одній із систем у логічну формулу для іншої системи і зробити висновок про те, що ця формула є теоремою. Іншими словами, аналогія використовується і для розв'язання деяких строго сформульованих задач і для пророкувань, а також для придбання не заданої раніше інформації.

Прикладом використання методу придбань знань за аналогією може служити система доведення теорем. При цьому загальна схема виведення виглядає в такий спосіб, як показано на рис. 5.11.

Рисунок 5.11 - Стратегія абстрагування

Питання для самоконтролю

1. Що таке експерті системи?
2. Яка специфіка експертних систем як програмної системи?
3. Наведіть спрощену структуру експертних систем.
4. База знань як елемент експертної системи.
5. Що таке факти?
6. Які необхідні умови подання знань?
7. Яка структура експертних систем?
8. Які є етапи розробки експертних систем?

9. Що являє собою процес подання знань у експертних системах?
10. Які рівні подання і рівні детальності для керування процесом пошуку рішення?
11. Організація знань у робочій системі і у базі даних.
12. Які методи пошуку рішень в експертних системах?
13. Яка методика формалізації експертних знань?
14. Наведіть приклади експертних задач.
15. Які є методи придбання знань?
16. Базова структура системи обробки знань.
17. Класифікація методів придбання знань.
18. Які є методи роботи зі знаннями?
19. Навчання без висновків.
20. Придбання знань на мета-рівні.
21. Які загальні положення придбання знань із прикладів?
22. Що таке навчання на прикладах?
23. На які методи можна поділити, при вивчені методу на прикладах?
24. Параметричне навчання.
25. Метод навчання за індукцією.
26. Навчання за аналогією.
27. Наведіть схему стратегії абстрагування.

ЛІТЕРАТУРА

1. Хант Э. Искусственный интеллект: В 2 ч. М.: Мир, 1978. – Ч. 2: Распознавание. образов. - С. 63-244.
2. Розенблатт Ф. Принципы нейродинамики (перцептрон и теория механизмов мозга). - М.: Мир, 1965. – 480 с.
3. Ивахненко А. Г. Долгосрочное прогнозирование и управление сложными системами. - К.: Техника, 1975. – 312 с.
4. Искусственный интеллект: Справочник - В 3-х книгах. - М.: Мир, 1990.
5. Лорьер Ж.-Л. Системы искусственного интеллекта. - М.: Мир, 1991. - 342 с.
6. У. Росс Эшби Конструкция мозга. Происхождение адаптивного поведения. - М.: Издательство иностранной литературы, 1962. - 392 с.
7. Физиология человека: Учебник для институтов физической культуры. / Под ред. Н. В. Зимикина. - Изд. 5-е. - М.: Физкультура и спорт, 1975. - 496 с.
8. А. В. Тимофеев Роботы и искусственный интеллект. - М.: Наука 1978. - 192 с.
9. П. Уинстон Искусственный интеллект. - М.: Мир, 1980.- 220 с.
10. Э. Хант Искусственный интеллект. - М.: Мир, 1978. – 558 с.
11. Реальность и прогнозы искусственного интеллекта / Под ред. В. Л. Стефанюка. - М.: Мир,1987. – 247 с.
12. Рубашкин В. Ш. Представление и анализ смысла в интеллектуальных системах. - М.: Наука, 1989. - 190 с.
13. Богданов Н. И. Основы построения интеллектуальных технических систем: Уч. пособие. - К.: УМКВО, 1988. - 84 с.
14. Глибовець М. М., Олецький О. В. Штучний інтелект: Підручник для студентів вищих навчальних закладів, які навчаються за спеціальностями “Комп’ютерні науки” та “Прикладна математика”. – К.: Вид.дім “Академія”, 2002. – 366 с.
15. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. – М.: Изд. дом “Вильямс”, 2003. – 864 с.
16. Бондарев В. Н., Аде Ф. Г. Искусственный интеллект: Учебное пособие для вузов. – Севастополь: Изд-во СевНТУ, 2002. – 615 с.
17. Искусственный интеллект: Справочник – В 3-х томах. – М.: Радио и связь, 1990.
18. Логический подход к искусственному интеллекту / Под ред. Г. П. Гаврилова. — М.: Мир, 1990.
19. Лорьер Ж. Л. Системы искусственного интеллекта. — М.: Мир, 1991.
20. Люгер Дж. Ф. Искусственный интеллект : стратегии и методы решения сложных проблем. — М.: Издат. дом “Вильямс”, 2003. — 865 с.

21. Нильсон Н. Принципы искусственного интеллекта. — М.: Радио и связь, 1985.

22. Эндрю А. Искусственный интеллект. — М.: Мир, 1985.

23. Кузин Л. Т. Основы кибернетики: Учебное пособие – В 2 т. / Энергия. - М.: 1979. – Т.2. - С.122-184.

24. Минский М. Фреймы для представления знаний: Пер. с англ. / Под ред. Ф. М. Куланова.- М.: Энергия, 1979.- 151 с.

25. Кузнецов В. Е. Представление в ЭВМ неформальных процедур. - М.: Наука 1989. - 158 с.

26. Месюра В. І., Ваховська Л. М. Основи проектування систем штучного інтелекту: Навч. посібник, МОН України. – Вінниця, ВДТУ, 2000. – 96 с.

27. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.

28. Гаек П., Гавранек Т. Автоматическое образование гипотез: математические основы общей теории — М.: Наука, 1983. — 280 с.

29. Гладун В. П. Процессы формирования новых знаний.— София, 1994. - 192 с.

30. Загоруйко Н. Г. Прикладные методы анализа данных знаний. — Новосибирск: Изд-во Ин-та математики, 1999. - 270 с.

31. Осуга С. Обработка знаний — М.: Мир, 1989. - 294 с.

32. Поспелов Д. А. Моделирование рассуждений. — М.: Радио и связь, 1989.

33. Горелик А. Л., Гуревич И. Б., Скрипкин В. А. Современное состояние проблемы распознавания. - М.: Радио и связь, 1985. - 160 с.

34. Фор А. Восприятие и распознавание образов. - М.: Машиностроение, 1989. - 272 с.

35. Браверман Э. М., Мучник И. Б. Структурные методы обработки эмпирических данных. - М.: Наука, 1983. - 464 с.

36. Васильев В. И. Проблемы обучения распознаванию образов. Принципы, алгоритмы реализация. - К.: Вища школа, 1989.

37. Верхаген К. и др. Распознавание образов. Состояние и перспективы. - М.: Радио и связь,.1985. - 104 с.

38. Вишняков Ю. М., Кодачигов В. И., Родзин С. И. Учебно-методическое пособие для самостоятельной работы по курсам "Системы искусственного интеллекта", "Методы распознавания образов". - Таганрог: Изд-во ТРТУ, 1999. - 132 с.

39. Мелихов А. Н., Карелин В. П. Методы распознавания изоморфизма и изоморфного вложения четких и нечетких графов. - Таганрог: Изд-во ТРТУ, 1995. - 90 с.

40. Орлов В. А. Граф-схемы алгоритмов распознавания. - М.: Наука, 1982. - 120 с.

41. Распознавание. Классификация. Прогноз: Вып. 2 / Под. ред. Журавлева Ю. И. - М.: Наука, 1989.

42. Растригин Л. А., Эренштеин Р. Х. Метод коллективного распознавания. - М.: Энергоатомиздат, 1981. - 79 с.

43. Реброва М.П. Автоматическая классификация в системах обработки информации. Поиск документов. - М.: Радио и связь, 1983. - 96 с.

44. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. - М.: Издательский дом "Вильямс", 2004. - 928 с.

45. Айфичер Э., Джервис Б. Цифровая обработка сигналов: практический подход. - М.: Издательский дом "Вильямс", 2004. - 992 с.

46. Основы цифровой обработки сигналов: Курс лекций. / Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б., Гук И. И. - СПб.: БХВ-Петербург, 2003. - 608 с.

47. Никулин Е. И. Компьютерная геометрия и алгоритмы машинной графики - СПб.: БХВ-Петербург, 2003. - 560 с.

48. Уэлстид С. Фракталы и вейвлеты для сжатия изображений в действии. - М.: Издательство "Триумф", 2003. - 320 с.

49. Сизиков В. С. Математические методы обработки результатов измерений: Учебник для вузов. - СПб.: Политехника, 2001. - 240 с.

50. Васильев В. И., Шевченко А. И. Искусственный интеллект: Проблема обучения опознаванию образов. — Донецк: изд. ДонГИИ, 1997. - 224 с.

51. Ту Дж., Гонсалес Р. Принципы распознавания образов. — М.: Мир, 1978. - 412 с.

52. Карпов О. Н. Технология построения устройств распознавания речи. - Днепропетровск: Изд-во ДНУ, 2001. - 184 с.

53. Данилюк Ю. С., Клименко Ю. Н. Анализ сложных колебаний по их "частотным параметрам" // Электронная техника: Сер.10: Микроэлектронные устройства, Вып. 1(67), 1988. - С. 32.

54. К. Тейлор. Как построить свою экспертную систему. // М.: Энергоатомиздат, 1991. - 287 с.

55. Экспертные системы. Принципы работы и примеры: Пер. с англ. / А. Брунинг, П. Джоис, Ф. Коке и др. / под ред. Р. Форсайта. - М.: Радио и связь, 1987. - 224 с.

56. Попов Э. В. Экспертные системы. Решение неформализованых задач в диалоге с ЭВМ. - М.: Наука, 1987. - 288 с.

57. Минский М. Фреймы для представления знаний: Пер. с англ. / Под ред. Ф. М. Кулакова. - М.: Энергия, 1979. - 151 с.

58. Экспертные системы. Принципы работы и примеры. / Под ред. Р. Форсайта. - М.: Радио и связь, 1987. - С. 224.

59. Джексон П. Введение в экспертные системы. - М.: Изд. дом "Вильямс", 2001. – 624 с.

60. Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин, Е. Ю. Головина, А. А. Загорянский, М. В. Фомина. — М.: Физматлит, 2004. - 704 с.

61. Выявление экспертных знаний / О. И. Ларичев, А. И. Мечитов и др.— М.: Наука, 1989.

62. Осипов Г. С. Приобретение знаний интеллектуальными системами. - М.: Наука, 1997.

63. Приобретение знаний / Под ред. С. Осуги., Ю. Саэки. - М.: Мир, 1990. - 304 с.

64. Уотерман Д. Руководство по экспертным системам. — М.: Мир, 1989.

ГЛОСАРІЙ

artificial intelligence - штучний інтелект

the applied software - прикладне програмне забезпечення

the intellectual interface -інтелектуальний інтерфейс

knowledge - знання

systems of processing of the information - системи обробки інформації

identification - ідентифікація

expert system - експертна система

Knowledge engineering - інженерія знань

calculation of predicates - числення предикатів

frame - остов, кістяк, каркас

slot - слід

inter - між

face - стояти обличчям

salience - стратегія глибини

breadth strategy - стратегія ширини

simplicity strategy - стратегія спрощення

complexity strategy - стратегія ускладнення

random strategy - випадкова стратегія

recognition of images - розпізнавання зображень

the information - інформація

model of problem environment - модель проблемного середовища

integrated robots - інтегральні роботи

new information technology - нова інформаційна технологія

model of a subject domain - модель предметної галузі

the more know, the more will have - чим більше знаєш, тим більше можеш одержати

operational environment - операційне середовище

hardware - апаратні засоби

software - програмні засоби

development on a horizontal - розвиток по горизонталі

development on a vertical -розвиток по вертикалі

principle of integrity - принцип цілістності

principle of a two-orientation - принцип двонаправленості

principle of purposefulness - принцип цілеспрямованості

principle of a prediction - принцип передбачення

principle "to not do much harm" - принцип "не нашкодь"

principle of maximal use of model of problem environment - принцип максимального використання моделі проблемного середовища

hypothesis of compactness - гіпотеза компактності

Picture Description Language - мова опису зображень

Навчальне видання

Гороховський Олександр Іванович

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

Навчальний посібник

Редактор Т. Старічек

Оригінал – макет підготовлено О. Гороховським

Підписано до друку
Формат 29,7×42¼ . Папір офсетний.
Гарнітура Times New Roman.
Друк різографічний. Ум. друк. арк.
Наклад прим. Зам №

Вінницький національний технічний університет,
науково-методичний відділ ВНТУ.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, к. 2201.
Тел. (0432) 59-87-36.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к.114.
Тел. (0432) 59-81-59.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.