

32 975875
265

В. Г. ДОЛЯ



Комп'ютерні системи
штучного інтелекту



ВІДКРИТИЙ МІЖНАРОДНИЙ УНІВЕРСИТЕТ
РОЗВИТКУ ЛЮДИНИ «УКРАЇНА»

В. Г. ДОЛЯ

КОМП'ЮТЕРНІ СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ

Рекомендовано Міністерством освіти і науки України
як навчальний посібник для студентів
вищих навчальних закладів

НБ ПНУС



768881

Київ
2011

УДК 004.383.8(075.8)

ББК 32.973я73

Д65

Гриф надано Міністерством освіти і науки України
(Лист № 1/11-5541 від 16.07.2009 р.)

Рецензенти: **В. В. Петров**, доктор технічних наук, член-кореспондент
НАНУ, директор Інституту проблем реєстрації інформації НАНУ

С. С. Забара, доктор технічних наук, професор, декан факультету комп'ютерних технологій Університету «Україна»

В. Г. Тарасенко, доктор технічних наук, професор, завідувач кафедри спеціалізованих комп'ютерних систем Національного технічного університету України («НТУ КПП»)

Доля В. Г.

Д65 Комп'ютерні системи штучного інтелекту: навч. посіб. — К.:
Ун-т «Україна», 2011. — 296 с.
ISBN 978-966-388-321-2.

Підручник містить основи побудови, функціонування та практичного використання експертних, нейрокомп'ютерних, робототехнічних, мультиагентних, нечітких та інших перспективних комп'ютерних систем штучного інтелекту. Наведені приклади практичного використання інтелектуальних систем у бізнес-застосуваннях, е-комерції, системах керування та в інших сферах економіки й виробництва. Подано описання моделі подання знань та перспективні технології інженерії знань, а також методи й алгоритми розв'язання інтелектуальних задач.

Підручник призначений для навчання студентів вищих навчальних закладів за спеціальностями «Комп'ютерні науки», «Комп'ютерні системи та мережі», «Інформаційні керівні системи та технології», «Програмне забезпечення автоматизованих систем». Систематизоване викладення матеріалу підручника може бути корисним для науковців, які займаються створенням та практичним застосуванням інтелектуальних систем в Україні.

імені Василя Стефаника

код 02125266

НАУКОВА БІБЛІОТЕКА

76 83 81

Ім. №

ISBN 978-966-388-321-2

УДК 004.383.8(075.8)

ББК 32.973я73

© Доля В. Г., 2011

© Університет «Україна», 2011

ЗМІСТ

Скорочення	9
Передмова	10
Вступ	14
Загальна характеристика проблем створення систем штучного інтелекту	14
Коротка історія створення й розвитку систем штучного інтелекту	15
Розділ 1. Інтелект людини	19
1.1. Загальні положення	19
1.2. Функціональні особливості людського мозку	19
1.3. Ієрархія інтелектуальних властивостей людини	21
Розділ 2. Штучний інтелект у комп'ютерній системі	23
2.1. Основні визначення	23
2.2. Основні особливості штучного інтелекту	24
2.3. Основні напрями розвитку штучного інтелекту	25
Розділ 3. Принципи побудови та функціонування комп'ютерних систем штучного інтелекту	28
3.1. Основні характеристики інтелектуальних систем	28
3.2. Типи сучасних інтелектуальних систем	30
3.3. Структура та функції перспективної інтелектуальної системи	32
3.3.1. Підсистема розв'язання задач на базі наявних знань	32
3.3.2. Підсистема інтелектуального інтерфейсу	35
3.3.3. Підсистема збору і первинного опрацювання інформації	37
3.3.4. Підсистема керування виконавчими пристроями	38
3.4. Склад розробників інтелектуальної системи	38
3.5. Технологія створення інтелектуальної системи	40
3.5.1. Ідентифікація	40
3.5.2. Концептуалізація	41
3.5.3. Формалізація	42
3.5.4. Реалізація	42

3.5.5. Налагодження й тестування	43
3.5.6. Дослідна експлуатація та вдосконалення систем штучного інтелекту	44
3.6. Предметні й проблемні області застосування інтелектуальних систем	44
Розділ 4. Основні аспекти інженерії знань	47
4.1. Загальні відомості	47
4.2. Основні види знань	48
4.3. Основні властивості знань	49
4.3.1. Внутрішня інтерпретація	49
4.3.2. Структурованість	49
4.3.3. Зв'язковість	49
4.3.4. Семантична метрика	50
4.3.5. Активність	50
4.4. Методи (добування) знань	51
4.4.1. Основні положення	51
4.4.2. Текстологічні методи добування знань	52
4.4.3. Пасивні комутативні методи	53
4.4.4. Активні комунікативні методи	54
4.4.5. Групові комутативні методи	56
4.4.6. Методи добування латентних (глибинних) знань	57
4.4.7. Особливості процесу добування знань	60
4.5. Комп'ютерне добування знань	61
4.6. Технологія добування знань <i>Data Mining</i>	63
4.7. Системи керування знаннями	64
Розділ 5. Бази знань	68
5.1. Модель предметної області	68
5.2. Технологія створення бази знань	70
5.3. Основні функціональні елементи бази знань	71
5.4. Структура бази знань	73
5.5. Онтології	74
Розділ 6. Моделі подання знань	80
6.1. Основні відомості	80
6.2. Логічні моделі подання знань	81
6.2.1. Основні положення	81
6.2.2. Основні елементи логіки предикатів	82
6.2.3. Формування логічної бази знань	84
6.3. Семантичні моделі подання знань	85
6.4. Фреймові моделі подання знань	88
6.4.1. Фрейми	88

6.4.2. Слоти	89
6.4.3. Функціональні властивості фреймової бази знань	90
6.4.4. Розв'язання задач у фреймових структурах	92
6.5. Продукційні моделі подання знань	93
6.5.1. Продукції	93
6.5.2. Функціональні властивості продукційної бази знань	95
6.6. Подання інформації та пошук знань в <i>Internet</i>	98
Розділ 7. Створення нових знань в інтелектуальних системах	98
7.1. Механізм виведення знань	98
7.2. Мета та локальні цілі	99
7.3. Механізм вибору знань	100
7.4. Рівні інтелектуальності розв'язуваних задач	101
7.5. Машина створення нових знань	102
7.5.1. Формування робочої множини фактів і правил	103
7.5.2. Зіставлення фактів і правил	103
7.5.3. Вирішення конфліктних ситуацій	104
7.5.4. Спрацювання та дія правил	105
7.6. Стратегії керування створенням нових знань	105
7.6.1. Прямая (висхідна) стратегія створення нових знань	105
7.6.2. Зворотна (спадна) стратегія розв'язання задач	106
Розділ 8. Методи розв'язання інтелектуальних задач	109
8.1. Метод резолюцій	109
8.2. Методи розпізнавання образів	111
8.2.1. Ідентифікація	111
8.2.2. Класифікація	112
8.2.3. Кластерний аналіз	114
8.2.4. Методи розв'язання задач розпізнавання образів	115
8.3. Генетичні алгоритми	118
8.3.1. Основні визначення	118
8.3.2. Принципи побудови й функціонування генетичних алгоритмів	119
8.3.3. Генетичні оператори	122
8.3.4. Функціональна схема генетичного алгоритму	124
8.3.5. Приклади практичного застосування генетичних алгоритмів	126
8.3.6. Особливості застосування генетичних алгоритмів	128
8.4. Методи групового урахування аргументів	129
8.5. Графові моделі та методи розв'язання задач на графах	132
8.5.1. Основні характеристики графів	132

8.5.2. Приклади класичних задач на графах	136
8.5.3. Алгоритми пошуку розв'язку задач на графах	138
8.6. Прецедентні методи	143
8.7. Гібридні алгоритми	145
8.8. Методи конструювання процесу розв'язання задач	146
8.8.1. Вихідна стратегія конструювання	146
8.8.2. Спадна стратегія конструювання	147
Розділ 9. Експертні системи	149
9.1. Статичні експертні системи	149
9.2. Динамічні експертні системи	151
9.3. Особливості створення та експлуатації експертних систем	153
9.4. Інструментальні засоби створення експертних систем	156
9.5. Області застосування експертних систем	159
9.6. Різновиди класичних експертних систем	161
9.6.1. Медична діагностична ЕС «Mycin»	161
9.6.2. ЕС хімічних досліджень «Dendral»	165
9.6.3. ЕС терапії онкологічних захворювань «Oncosin»	166
9.6.4. ЕС проектування конфігурацій комп'ютерів «R1/XCon»	169
9.6.5. Прецедентні ЕС комп'ютерного навчання	171
9.6.6. ЕС навчання й контролю знань	173
Розділ 10. Нейрокомп'ютерні системи	177
10.1. Біологічна нейронна мережа	178
10.2. Штучні нейрони	180
10.3. Принцип побудови та функціонування штучних нейронних мереж	181
10.4. Методи навчання нейронних мереж	183
10.4.1. Навчання «з учителем»	183
10.4.2. Навчання «без учителя»	184
10.5. Типи нейронних мереж	186
10.5.1. Персептрон Розенблатта	186
10.5.2. Багатошаровий персептрон. Алгоритм зворотного поширення похибки	188
10.5.3. Алгоритм навчання Хебба	192
10.5.4. Нейромережа та карта Кохонена	193
10.5.5. Нейромережа Хопфілда	195
10.5.6. Нейромережа Хеммінга	197

10.5.7. Зірки Гроссберга	199
10.5.8. Нейромережа зустрічного поширення	200
10.5.9. Когнітрон Фукушими	201
10.5.10. Неокогнітрон Фукушими	203
10.5.11. Нейронні мережі з адаптивним резонансом. Проблема стабільності-пластичності нейромереж	205
10.6. Проблемні області застосування нейрокомп'ютерних систем	210
10.7. Практичні застосування нейрокомп'ютерних систем	212
10.8. Особливості створення нейрокомп'ютерних систем	216
Розділ 11. Робототехнічні інтелектуальні системи	217
11.1. Основні положення	217
11.2. Узагальнена структура робототехнічної системи	219
11.3. Функції підсистем робототехнічної системи	220
11.3.1. Збирання та перетворення інформації	220
11.3.2. Первинне опрацювання й подання інформації	222
11.3.3. Вироблення й прийняття рішень	224
11.3.4. Підсистема зв'язку й керування	227
11.3.5. Виконавчі пристрої й механізми	228
11.4. Колективні робототехнічні системи	229
11.5. Області практичного застосування робототехнічних систем	230
Розділ 12. Інтелектуальні системи прийняття рішень	234
12.1. Основні положення	234
12.2. Структура та функції системи прийняття рішень	235
12.3. Області практичного застосування інтелектуальних систем прийняття рішень	237
Розділ 13. Мультиагентні інтелектуальні системи	239
13.1. Структура й функції мультиагентної системи	239
13.2. Функціональні властивості інтелектуальних агентів	241
13.3. Мультиагентні <i>internet</i> -системи	243
13.4. Практичні розроблення мультиагентних систем	246
Розділ 14. Нечіткі інтелектуальні системи	248
14.1. Основні поняття нечіткої логіки	248
14.2. Структура та функції нечіткої системи	250
14.3. Практичні розроблення нечітких систем	251
14.3.1. Нечітка робототехнічна система	251
14.3.2. Нечітка система керування роботою світлофора	254
14.4. Області застосування нечітких інтелектуальних систем	257

Додаток 1	259
------------------------	-----

Контрольні запитання та завдання	259
Розділ 1. Інтелект людини	259
Розділ 2. Штучний інтелект у комп'ютерній системі	259
Розділ 3. Принципи побудови та функціонування комп'ютерних систем штучного інтелекту	260
Розділ 4. Основні аспекти інженерії знань	261
Розділ 5. Бази знань	262
Розділ 6. Моделі подання знань	263
Розділ 7. Створення нових знань в інтелектуальних системах	264
Розділ 8. Методи розв'язання інтелектуальних задач	265
Розділ 9. Експертні системи	266
Розділ 10. Нейрокомп'ютерні системи	267
Розділ 11. Робототехнічні інтелектуальні системи	268
Розділ 12. Інтелектуальні системи прийняття рішень	269
Розділ 13. Мультиагентні інтелектуальні системи	269
Розділ 14. Нечіткі інтелектуальні системи	270

Додаток 2	271
------------------------	-----

Словник термінів	271
-------------------------------	-----

Література	274
Розділ 1. Інтелект людини	274
Розділ 2. Штучний інтелект у комп'ютерній системі	275
Розділ 3. Принципи побудови та функціонування комп'ютерних систем штучного інтелекту	277
Розділ 4. Основні аспекти інженерії знань	278
Розділ 5. Бази знань	280
Розділ 6. Моделі подання знань	281
Розділ 7. Створення нових знань в інтелектуальних системах	283
Розділ 8. Методи вирішення інтелектуальних задач	284
Розділ 9. Експертні системи	286
Розділ 10. Нейрокомп'ютерні системи	288
Розділ 11. Робототехнічні інтелектуальні системи	291
Розділ 12. Інтелектуальні системи прийняття рішень	292
Розділ 13. Мультиагентні інтелектуальні системи	292
Розділ 14. Нечіткі інтелектуальні системи	294

Скорочення

ШІ – Штучний інтелект.
 КСШІ – Комп'ютерна система штучного інтелекту.
 ПрО – Предметна (проблемна) область КСШІ.
 БЗ – База знань.
 ОнЛ – Онтологія.
 ЕС – Експертна система.
 НКС – Нейрокомп'ютерна система.
 РТІС – Робототехнічна інтелектуальна система.
 ІСПР – Інтелектуальна система прийняття рішень.
 МАС – Мультиагентна система.
 НІС – Нечітка інтелектуальна система.
 WWW (*World Wide Web*) (*Internet*) – Всесвітня гіпертекстова мережа комп'ютерних файлів.
 HTTP (*Hyper Text Transfer Protocol*) – Програмний засіб (протокол) передачі файлів через Internet.
 HTML (*Hyper Text Markup Language*) – мова опису документів у Internet.

ПЕРЕДМОВА

Штучний інтелект (ШІ) нині є престижним, актуальним і перспективним науковим напрямом сучасної інформатики. Як і в кожній новій сфері науки, серед дослідників ШІ є різні погляди на головне призначення цього напрямку. Хиткі межі ШІ, що відокремлюють цю науку від досліджень у галузях кібернетики, створення нових поколінь комп'ютерів і систем програмування, штучних органів чуття (зір, слух, дотик) від робіт в областях математичної логіки, психології, лінгвістики та інших наукових напрямів. Однак основним науковим напрямком досліджень у сфері ШІ сучасні фахівці вважають вивчення *структури й механізмів роботи мозку людини для розкриття й практичного використання таємниць людського мислення*.

Поряд із фундаментальними дослідженнями ШІ в економічно розвинених країнах світу (головним чином – у США та Японії) нині широким фронтом ведуться роботи зі створення *інтелектуальних систем* різного призначення з використанням *методів і засобів ШІ*. У межах цих систем розробляються принципово нові методи *опрацювання інформації та розв'язання інтелектуальних задач*. Ідеї ШІ є концептуальними практично для всіх сучасних проєктів створення *інтелектуальних комп'ютерів* нових поколінь. Результати, досягнуті в області ШІ, дають змогу знаходити ефективні розв'язання багатьох соціально-економічних, виробничих, екологічних та інших задач, які не можуть бути вирішені *обчислювальними засобами звичайних комп'ютерних систем*.

З огляду на світові тенденції розвитку ШІ, навчальна дисципліна «*Комп'ютерні системи штучного інтелекту*» посідає особливе місце у формуванні науково-технічного світогляду майбутнього фахівця у сфері інформатики. Сучасний рівень теоретичного розуміння і практичного використання *інтелектуальних систем* у світовій комп'ютерно-інформаційній індустрії потребує від майбутнього інженера професійних знань у цій галузі. Досягнення останніх років у сфері створення *експертних, нейрокомп'ютерних, робототехнічних, мультиагентних та інших інтелектуальних систем* свідчать про ефективне використання результатів теоретичних досліджень ШІ у конкретних практичних застосуваннях.

В основу цього підручника покладено результати *системного аналізу публікацій* з даної проблематики, перелік яких наведено у списку «*Літератури*». Підручник містить *принципи, методи й за-*

соби створення, функціонування та практичного застосування сучасних і перспективних комп'ютерних систем штучного інтелекту, що можуть бути використані під час створення подібних систем в Україні. Головним завданням підручника є надання навчальної методичної допомоги майбутнім фахівцям із питань проєктування й створення *нових поколінь інтелектуальних систем* у різних сферах людської діяльності. Останніми роками ця тематика посідає особливе місце у традиційних курсах дисциплін, пов'язаних із професійною підготовкою студентів вищих навчальних закладів.

Необхідність написання цього підручника багато в чому зумовлена недостатнім рівнем висвітлення цієї проблематики у *вітчизняних публікаціях*, присвячених, здебільшого, *теоретичним дослідженням* в області ШІ. У цьому підручнику головну увагу приділено *практичним проблемам* зі створення й застосування *комп'ютерних систем ШІ*.

Підручник складається із 14 розділів, літератури та додатків.

У «*Вступі*» наведено загальну характеристику проблеми і коротку історію створення й розвитку інтелектуальних систем.

У розділі 1 аналізується поняття «*Інтелект людини*» з погляду подання його у *комп'ютері*. Розглядаються функціональні *особливості людського мозку та основні інтелектуальні властивості людини*.

У розділі 2 поняття «*Інтелект людини*» досліджується з позицій використання його у *комп'ютерній системі*. ШІ трактується як *комп'ютерна (програмно-апаратна) модель інтелекту людини*. Описуються основні *особливості і напрями розвитку ШІ*.

Розділ 3 присвячений *принципам побудови й функціонування сучасних інтелектуальних систем*. У ньому описано основні *характеристики, структуру та функції типової інтелектуальної системи*. Розглянуто *типи систем ШІ, склад розробників, технологію створення та області практичного застосування*.

У розділі 4 подано основні аспекти *інженерії знань*, яка є підґрунтям технології створення інтелектуальних систем. Розглядаються основні *типи та властивості знань*. Описано *методи здобування знань із різних інформаційних джерел*. Розглянуто основні аспекти *комп'ютерного добування знань, технології Data Mining та систем керування знаннями*.

У розділі 5 подано основні *характеристики, структура та процес створення баз знань* інтелектуальних систем. Наведено відомості про *онтологічні (концептуальні) бази знань та їх практичне застосування*.

У розділі 6 подано основні характеристики *логічних, семантичних, фреймових та продукційних моделей подання знань*, які мають найбільше застосування в сучасних системах ШІ.

Розділ 7 присвячено проблемі *виводу на знаннях у системах III* під час *розв'язання інтелектуальних задач*. Описано *механізм виводу на знаннях*, принцип роботи *машини виводу*, *засоби реалізації процесу виводу та стратегії керування виводом на знаннях*.

У розділі 8 подано *алгоритми вирішення інтелектуальних задач у системах III*. Наведено основні характеристики й описання алгоритмів *розпізнавання образів* під час вирішення задач *ідентифікації, класифікації та кластерного аналізу*. Розглянуто принципи функціонування та області практичної реалізації *генетичних алгоритмів* і методів *групового обліку аргументів*. Описано *прецедентні методи, алгоритми подання й вирішення задач на графах* і методи *конструювання процесу вирішення інтелектуальних задач*.

У розділах 9–14 подано основні *типи сучасних систем III*.

Розділ 9 присвячено *експертним системам*, які набули найбільшого поширення в економіці й промисловості розвинутих країн. Розглянуто основні *види та особливості технології, інструментальні засоби створення та області практичного застосування експертних систем*.

У розділі 10 описано принципи *побудови, функціонування й навчання нейрокомп'ютерних систем*. Наведено структурні схеми основних *типів нейронних мереж* й *алгоритми їхнього функціонування*, зокрема *персептрон Розенблатта, багатошаровий персептрон, нейромережі Кохонена, Хопфілда, Хеммінга, Гроссберга, нейромережу зустрічного поширення, когнітрон і неокогнітрон Фукушими, нейронні мережі з адаптивним резонансом*. Розглянуто області *практичного застосування та перспективи розвитку нейрокомп'ютерних систем*.

У розділі 11 розглянуто *структуру, принципи функціонування та області застосування автономних і колективних робототехнічних інтелектуальних систем*. Описано способи *планування маршрутів руху роботів у замкненому просторі*.

У розділі 12 подано принципи *побудови й функціонування інтелектуальних систем прийняття рішень*. Наведено опис основних *підсистем та області практичного застосування цих систем*.

У розділі 13 описано *структуру, функції та практичні розроблення мультиагентних систем*. Розглянуто *різновиди й властивості інтелектуальних агентів, які функціонують, зокрема у системі Internet*.

У розділі 14 подано *нечіткі інтелектуальні системи*. Розглянуто основні поняття *нечіткої логіки* й *типи операцій із нечіткими множинами*. Описано *алгоритми функціонування, конкретні розроблення та області практичного застосування нечітких систем*.

У *список літератури* включено роботи, присвячені питанням *створення, функціонування та практичного застосування інтелектуальних систем різноманітного призначення*. Перелік робіт складено окремо з кожного із вищезазначених *тематичних розділів у хронологічному порядку*. На жаль, навести перелік усіх наукових публікацій із *фундаментальних проблем III* в межах цього підручника неможливо.

Додатки до підручника містять запитання, вправи й теми для обговорення (за розділами), словник термінів та список використаних скорочень.

Підручник призначений для навчання студентів ВНЗ за спеціальностями *«Комп'ютерні системи та мережі», «Інформаційні керівні системи та технології», «Програмне забезпечення автоматизованих систем»*. Систематизований матеріал підручника може бути корисним для науковців, які займаються створенням та практичним застосуванням інтелектуальних систем в Україні.

Загальна характеристика проблеми створення систем штучного інтелекту

Останнім часом значно зросли обсяги інформації, що спричинило додаткові складнощі в її опрацюванні. Знання людства подвоюються за нашого часу приблизно кожні 5 років, хоча на початку XIX століття цей період становив 50 років. Обсяг нагромадженої людством інформації вже становить більш ніж 10^{14} біт. Мозок людини здатний сприймати інформацію зі швидкістю до 1 біт/с, тим часом, як потік нової інформації спрямовується сьогодні на людину зі швидкістю до 20 біт/с.

Ця проблема так званого *інформаційного вибуху* вирішується нині за допомогою комп'ютерних інформаційних систем, здатних здійснювати *оперативний аналіз, ефективне опрацювання та надійне збереження* значних обсягів інформації. Такі системи створюються з використанням наукових ідей *кібернетики*, що досліджує інформаційні властивості *складних об'єктів різної структури*. Ключовою проблемою *кібернетики* є розроблення і дослідження *систем оптимального керування* складними (промисловими, економічними, соціальними тощо.) об'єктами, які можуть бути описані *точною чи наближеною математичною моделлю*. У цих системах використовуються *алгоритмічні методи розв'язання задач*, що реалізуються у *комп'ютері* як *механічне здійснення заданої послідовності обчислень*, яке приводить до розв'язання задачі. Становлення *кібернетики* і розвиток *класичних (традиційних) методів оптимального керування* пов'язане з іменами таких відомих учених, як *Н. Вінер, Р. Белман, А. Ляпунов, Р. Калман, В. Глушков* та ін.

Однак багато задач керування об'єктами, процесами і ситуаціями навколишнього світу пов'язані, здебільшого, із життєдіяльністю людини та його поведінням в екстремальних (форс-мажорних) умовах, *важко піддаються математичному моделюванню*. Прикладом такої задачі може слугувати оперативне прийняття рішень щодо керування космічним польотом та порятунку життя космонавтів під час *позаштатної ситуації* на борті космічного корабля. Таку ситуацію не можна описати в *математичних термінах*, щоб потім розв'язати цю задачу на комп'ютері.

Важко формалізувати предметну область *медицини*, де лікарі мають справу з найрізноманітнішою інформацією про стан здо-

ров'я пацієнтів, результати лабораторних аналізів, історії хвороб, зміни у стані хворих, діагностику захворювань та можливі способи лікування. Лікаряю дуже важко досягнути такий величезний потік інформації, не забути й врахувати найрізноманітніші чинники під час прийняття рішень. Значна кількість інформації, *неможливість її формального (математичного) опису* для наступного використання *комп'ютерних засобів* призводить до того, що вчасно не виявляються, наприклад, до 25% ракових захворювань.

Для вирішення подібних проблем в економічно розвинутих країнах світу сьогодні широко впроваджуються комп'ютерні системи штучного інтелекту (*КСШІ*), основним призначенням яких є розв'язання інтелектуальних задач у різних областях людської діяльності, які важко або неможливо формалізувати. Використання *КСШІ* дає змогу вирішувати медичні, виробничо-економічні та інші завдання в різноманітних предметних (проблемних) областях, де утруднена математична формалізація об'єктів, процесів, явищ, ситуацій.

Коротка історія створення й розвитку систем штучного інтелекту

Перші *КСШІ* на *ранньому етапі* їхнього розвитку являли собою «іграшкові» програмні засоби, за допомогою яких з'являлися перші спроби розв'язувати найпростіші *інтелектуальні задачі*, пов'язані з *відтворенням (імітацією) окремих творчих процесів*. Дослідницьким полігоном для творців *КСШІ* були різноманітні ігри, головоломки, математичні задачі. У перших роботах із *КСШІ* описано програми розв'язання найпростіших *ігрових інтелектуальних задач* із використанням *лабіринтової моделі* пошуку рішення. Такими задачами були «хрестики-нулики», «морський бій», шашки, нарди, шахи, в яких заздалегідь відомий алгоритм пошуку та вибору можливих альтернативних варіантів розв'язання поставленої задачі.

Багато ігрових задач стали класичними (задачі про *місіонерів і людожерів, мавп і банани, Ханойську вежу, задачу комівояжера, гру в «15»* тощо). Вибір таких задач обумовлювався простотою і ясністю предметної області, її малою розмірністю, можливістю досить легкого підбору й конструювання алгоритму вирішення задачі.

На початку 60-х років минулого століття було створено першу *нейронну систему розпізнавання образів — перцептрон Розенблатта*. Його підґрунтя — спрощена *модель структури мозку* як множини *штучних нейронів (фотоелементів)*, з'єднаних між собою відповідно до наявної (у той час) схеми взаємодії клітин людини.

На розвиток ранніх *КСШ* великий вплив зробили *Дж. Робінсон* і його колеги, що створили *формально-логічні* моделі мислення, які базуються на *методах формальної логіки*. Вони розробили *інтелектуальні програмні засоби «Логік-теоретик»* і *«Загальний вирішувач задач»*, що давали змогу, зокрема, автоматично *доводити шкільні теореми*. В основу цих систем було покладено *метод резолюцій*, який використовує для доказу теорем *логіку предикатів*. На підставі методу *резолюцій А. Кольмере* створив мову логічного програмування *«Prolog»*.

Значну увагу дослідники приділяли тоді розробленню так званих *евристичних (інтуїтивних)* моделей мислення для вирішення інтелектуальних задач різних типів. Суттєвий внесок у створення цих моделей зробили *Дж. Мак-Карті* — автор мови *«Lisp»* (першої мови *ШІ*), *М. Мінський* — автор *фреймової моделі* подання знань, *Д. Мічі*, *П. Уінстон*, *П. Хейес*, *А. Ендрю*, *Д. Хант* та інші дослідники *ШІ*. Система *ситуаційного керування*, яку розробив *Д. Поспелов*, використовувала моделі представлення *ситуацій*, в основу яких було покладено мову символічного опрацювання інформації *«Рефал»*.

На основі розроблених моделей мислення стали здійснюватися перші спроби практичного застосування *КСШ* в різноманітних предметних областях.

Системи комп'ютерного перекладу мов використовували *логічні моделі* співвіднесення *словоформ* однієї мови зі *словоформами* іншої. Під час розроблення цих систем виник новий науковий напрям — *комп'ютерний морфологічний та синтаксичний аналіз і синтез мов*. Уперше відбувся поділ знань на *декларативну* (словник, лексика, морфологія, синтаксис) і *процедурну* (програмну) частини, які могли змінюватися незалежно одна від одної.

Системи комп'ютерного реферування текстів здійснювали пошук інформації у великих інформаційних сховищах; визначали відповідність знайденого тексту *запиту користувача*, виконували семантичний аналіз змісту великих фрагментів тексту й складання його *реферату*, аналізуючи зв'язки *надфразової єдності*. При розробленні цих систем виник новий науковий напрям — *комп'ютерна лінгвістика*, який використовують під час створення сучасних засобів спілкування людини з комп'ютером на підставі *природно-мовних текстів*. Водночас виник новий спосіб пошуку інформації *за зразком*, який широко використовується у *базах знань* сучасних *КСШ*.

Системи відтворення текстів і музики використовували процедуру синтезу текстів і систему правил (за аналогією з формуванням реферату), що породжують структуру зв'язного тексту, комбінуючи слова з невеликого словника. За допомогою перших програм, які складали тексти, їхні творці не прагнули створити високомистецькі здобутки. Потрібно було тільки продемонструвати *принципову*

можливість створення нових текстів за допомогою комп'ютера. Тому при створенні програм приділяли мало уваги *семантиці* текстів, маючи на меті лише створення зв'язного тексту. Так з'явилися програми, що складають *«абстракціоністські»* вірші чи *«заумну»* прозу. Програми відтворення музики використовують принципи *евристичного програмування* й *експертні знання* в галузі музичної творчості. Програми будуються на принципах *евристичного* добору потрібної ноти із множини нот, які генеруються випадковим чином.

Наступний етап досліджень ознаменувався появою і подальшим удосконаленням *інтелектуальних засобів людино-машинної взаємодії* при апаратній реалізації *КСШ*. Центр уваги дослідників змістився з автономно функціонуючих систем, що самостійно вирішують поставлені перед ними задачі, до створення *людино-машинних систем*, які інтегрують у єдине ціле інтелект людини й можливості комп'ютерної техніки для досягнення загальної мети. На перший план було висунуто розроблення методів і засобів, що забезпечують тісну взаємодію людини й комп'ютера з можливістю оперативного внесення людиною змін упродовж усього процесу розв'язання задачі. Розвиток досліджень у цьому напрямі обумовлювався різким зростанням виробництва і здешевленням засобів комп'ютерної техніки, що робило їх потенційно доступними для широких кіл користувачів. У межах цих досліджень у США було створено перші експериментальні *інтегральні роботи*, які функціонували в лабораторних умовах.

Унаслідок проведення зазначених вище досліджень і робіт дослідники усвідомили те, що ефективність створюваних *КСШ* істотно залежить від *знань*, якими вони володіють. Виникла гостра необхідність вирішення таких проблем, як адекватне подання знань, зорове сприйняття, моделювання поведінки в динамічних середовищах, спілкування із системою природною (людською) мовою тощо.

До кінця *70-х років* минулого століття фахівці *ШІ* зуміли виробити загальну систему понять і створили основи теорії й методологію застосування цієї теорії до створення *комп'ютерних КСШ* в різних галузях промисловості. Водночас було розроблено перші проблемно-орієнтовані *мови програмування*, *моделі подання знань*, засоби для побудови й супроводу *баз знань*, засоби маніпулювання знаннями тощо. У коло досліджень у галузі подання знань входили принципи роботи *пам'яті* людини, питання видобутку інформації з *пам'яті*, розпізнавання та відновлення *знамен* тощо. З'явилися комп'ютерні застосування, які могли знаходити *логічні зв'язки* між *знаннями* чи іншому етапі розв'язання інтелектуальної задачі.

Суттєвий прорив у розв'язанні проблем створення *КСШ* відбувся в США, коли на зміну пошукам універсального алгоритму

вирішення задач прийшла ідея моделювати конкретні знання фахівців-експертів. У зв'язку з цим сформувався самостійний напрям КСШІ, що одержав назву «експертні системи», які здатні виробляти рішення на рівні експертів. У США було створено перші експертні системи «Mycin» — у медицині і «Dendral» — у хімії. Ці системи відрізнялися від традиційних систем оброблення даних символьним способом подання інформації, символьним виводом на знаннях і евристичним пошуком рішення задач.

На початку 80-х років минулого століття було оголошено глобальну програму розвитку нових технологій «Esprit» (Європейський Союз), яка вміщувала проблематику створення КСШІ та інтелектуальних комп'ютерів. Водночас Японія оголосила про початок створення проекту інтелектуальних комп'ютерів V і VI поколінь, заснованих на роботі зі знаннями. Проект було розраховано на 10 років і він поєднував кращих молодих фахівців найбільших японських комп'ютерних корпорацій, для яких було створено спеціальний інститут «ICOT». До середини 90-х років японська асоціація зі штучного інтелекту нараховувала 40 тис. фахівців. Принципи побудови КСШІ й інтелектуальних комп'ютерів стали підґрунтям оборонної програми США «Strategic Computer Initiative».

Починаючи із середини 80-х років минулого століття, в усьому світі різко зросли асигнування на створення інтелектуальних систем. Якщо в 1980 році вклади у дослідження КСШІ в усьому світі становили близько \$70 млн, то в 1985 р. ця цифра становила вже \$350 млн. У 1987 році тільки США витратили на дослідження КСШІ \$2,5 млрд, а Японія — \$1,5 млрд. Нині ці витрати в усьому світі вже перевищують \$40 млрд на рік.

В Україні здійснювалися фундаментальні дослідження з моделювання мислення та психіки людини (М. М. Амосов). У межах вирішення проблем розпізнавання образів і моделювання складних об'єктів було розроблено принципово нові методи індуктивного моделювання і самонавчання КСШІ, зокрема — методу групового обліку аргументів, що з успіхом використовується сьогодні в країнах ближнього та далекого зарубіжжя (О. Г. Івахненко).

Українські вчені зробили значний вклад у фундаментальні дослідження з проблем створення інтелектуальних робототехнічних систем (А. В. Тимофєєв), нейронних мереж і нейрокомп'ютерів (Кукуль - Є. М.), інтелектуальних систем прийняття рішень та мультиагентних систем (В. П. Гладун), «образного комп'ютера», здатного оперувати з візуальними і мовними образами (Т. К. Вінцюк), у вирішення проблеми побудови та функціонування баз знань (О. О. Бакаєв, В. І. Гриценко), у розроблення моделей подання знань у КСШІ (Ф. І. Андон) та ін.

Розділ 1

ІНТЕЛЕКТ ЛЮДИНИ

1.1. Загальні положення

У визначення дисципліни «КСШІ» входить поняття «Інтелект» («Intelligence»). Воно походить від латинського «Intellectus», що означає розум людини. Мова йде про природний (біологічний) інтелект людини, вироблений людством упродовж мільйонів років еволюції. Людина вважається інтелектуальною від природи у зв'язку зі здатністю людського мозку ставити й вирішувати інтелектуальні задачі, що виникають у процесах життєдіяльності й виживання людини у складних умовах навколишнього світу [1.1–1.16].

Інтелект являє собою сплав багатьох навичок людини, пов'язаних із пошуком і опрацюванням інформації. Тому дати цьому поняттю таке визначення, щоб воно задовольняло всіх дослідників, досить важко.

У 2000 році редакція журналу «Штучний інтелект» проводила дискусію з визначення поняття «природний інтелект» у телеконференціях системи Internet. Було висловлено різноманітні погляди і запропоновано понад 100 визначень цього поняття. Серед них найрозповсюдженішими є:

- розумові й психологічні здібності людини;
- здатності до: сприйняття інформації, вирішенню задач, оцінки ситуацій, узагальнення й асоціації, творчості;
- розум, уміння мислити;
- інструмент свідомості;
- ступінь складності мислення;
- механізм опрацювання сигналів;
- репродуктивна інформаційна система;
- міра людського в людині тощо.

Інтелектом називають також здатність людини прогнозувати стан зовнішнього середовища, досягати успіху під час пошуку однієї з багатьох можливих цілей у різноманітні зовнішнього світу, уміння перетворювати виниклу думку у визначену реакцію, що веде до поставленої мети тощо.

1.2. Функціональні особливості людського мозку

На сьогодні фізіологи встановили, що основою людського мозку є велика кількість взаємодіючих нервових клітин — нейронів, пов'язаних

між собою міжнейронними з'єднаннями. Кожен нейрон має до 30 тисяч зв'язків із іншими нейронами. Міжнейронних зв'язків у мозку понад 10^{14} . Мозок може виконувати до 10 білйонів операцій за секунду.

Основною особливістю людського мозку є його *напівкульна асиметрія* (рис. 1.1) [1.4–1.10]. Наша *ліва півкуля* володіє *свідомим логічним мисленням*, яке людина використовує під час вирішення повсякденних насущних задач, пов'язаних із її існуванням. Мислення ж *правої півкулі* є рефлексорним, інтуїтивним і спирається на *підсвідомі почуттєві* процеси. Людина часто не може сформулювати свої знання й оперує поняттями, про які не має чіткого уявлення (наприклад, «Я не можу пояснити це, але мені здається, що...»)... Людина часто робить висновки за *аналогією* та *асоціацією*. Її *права півкуля* мислить на підсвідомому рівні *почуттєвих образів*, які вона використовує, здебільшого, у своїй *творчій* діяльності. Без *підсвідомого мислення* неможливе будь-яке відкриття, оскільки підсвідомість породжує гіпотези, що підтверджуються (чи спростовуються) потім логічним аналізом чи експериментом.

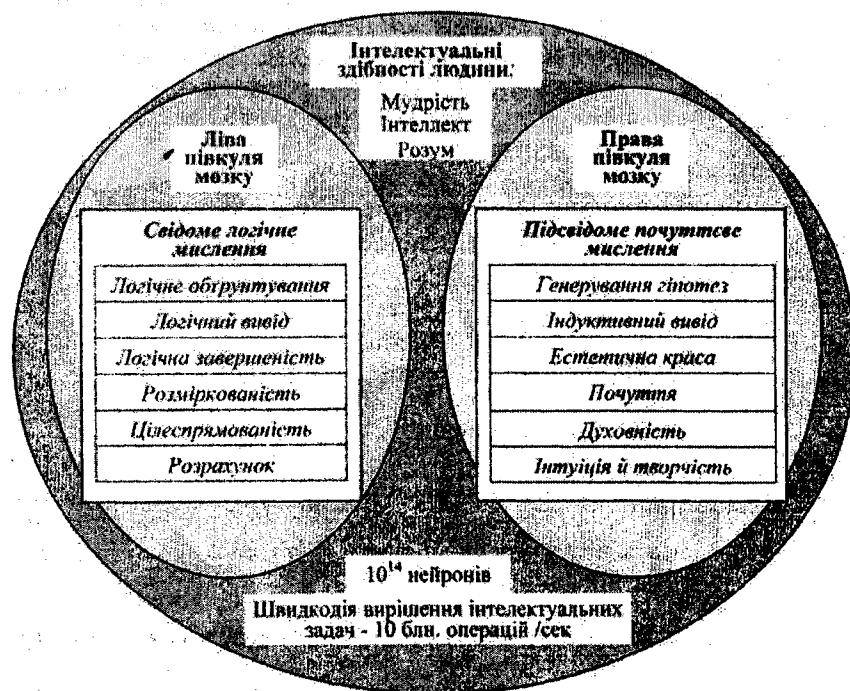


Рис.1.1 — Спрощена структурна схема мозку людини

Слід зазначити, що незважаючи на досягнення вчених-фізіологів, мозок продовжує зберігати свої таємниці й дотепер. Відомо лише орієнтовне розташування *зон мозку*, які відповідають за ту чи іншу функцію. Найскладніші переплетення зв'язків кори головного мозку поки що не піддаються розшифруванню. Дотепер невідомо мотивація й механізми усвідомленої діяльності людини. Неповною мірою вивчені фізіологічні основи функціонування мозку під час процесу мислення. Фізіологам поки не вдалося точно встановити місце зосередження у мізку інформації, що запам'ятовується, і спосіб її опрацювання.

1.3. Ієрархія інтелектуальних властивостей людини

Незважаючи на розбіжність поглядів і підходів, сьогодні більшість дослідників і фахівців вважають, що сукупність *інтелектуальних здібностей* людини можна подати як такі *три рівні ієрархії* [1.10–1.16].

1. *Розум* людини здатний у процесі мислення висувати *цілі* (ставити задачі) і знаходити способи їхнього досягнення (*вирішення*) за допомогою *міркувань* та узагальнюючих *висновків*. Основними властивостями (здібностями) розуму людини є:

- *рефлексія* — реакція людини на свої дії та на впливи довкілля;
- *опис ситуації* з характерною для неї *неповнотою й невизначеністю*;
- *упорядкування інформації* за ступенями *істотності*;
- *вибір стратегій*, адекватних *розв'язуваній задачі*;
- *визначення логічних наслідків*;
- *оцінка результатів та засобів їх одержання*;
- *пошук подібності чинників і ситуацій*;
- *генерування припущень, узагальнень тощо*.

Якщо підсумувати ці якості, то можна сказати, що *розум* — це *однобічна спрямованість людини на вирішення виникаючих перед нею проблем*. Наприклад, людина може бути розумною у розв'язанні математичних задач, але мати примітивний гуманітарний розвиток і не бути інтелектуальною.

2. Більш високим рівнем розвитку здібностей людини є її *інтелект*, який визначають такі чинники:

- а) *мислення* людини, тобто здатність людського мозку до постановки різноманітних *інтелектуальних (творчих) задач* та їх вирішення, що ґрунтується на використанні інформації із зовнішнього світу й засвоєних людиною знаннях;

Розділ 2

ШТУЧНИЙ ІНТЕЛЕКТ У КОМП'ЮТЕРНІЙ СИСТЕМІ

- б) *свідомість (самосвідомість)* людини, що дає змогу контролювати свої дії та усвідомлювати себе як *особистість* у навколишньому світі відповідно до досвіду напромаджених людством знань і цінностей;
- в) вироблені людством *духовні цінності* та *ідеали* (мораль, совість, етика, естетика, культура, релігійні засади тощо), які впливають на особистість і суспільство.

Таким чином, *інтелект* — це *багатобічний розум* (розум «у багатьох напрямках»), що здатний до:

- цілісного сприйняття зовнішнього світу;
- пізнавальної цікавості й міркування;
- засвоєння й використання знань, вироблених людством на сучасному етапі розвитку;
- висування цілей і вироблення стратегій розв'язання задач, що постають перед людиною;
- породження й самоорганізації нових (особистих) знань та нагромадження досвіду;
- усвідомлення себе в зовнішньому світі з огляду на нагромаджені духовні цінності та ідеали.

Характерними рисами інтелекту є здатність до навчання, узагальнення, нагромадження досвіду (знань і навичок) та адаптації до умов, що змінюються, у процесі вирішення виникаючих перед людиною проблем. Інтелект є універсальним засобом, за допомогою якого людина здатна вирішувати неформалізовані задачі, для яких немає стандартних, заздалегідь відомих методів. Саме завдяки інтелекту людині підвладні життєво важливі *інтелектуальні* задачі, і вона може легко переходити від вирішення однієї з них на іншу.

3. Вершиною в ієрархії здібностей людини вважається *мудрість* — *здатність людини у складних, непередбачуваних і екстремальних ситуаціях вчасно застосовувати свій інтелект* для вирішення виникаючих суспільних і особистих проблем, максимально використовуючи свій інтелектуальний досвід, нагромаджені духовні цінності й ідеали.

2.1. Основні визначення

Як було зазначено вище, нині більшість дослідників *ІІІ* розглядає як область *інформатики*, орієнтовану на вирішення *творчих задач* людської діяльності за допомогою використання *комп'ютерних засобів* моделювання процесів мислення і пам'яті людини [2.1–2.41]. Наприклад, у довіднику [2.6] дається так визначення: «*ІІІ* — це галузь *інформатики*, що займається розробленням *інтелектуальних комп'ютерних систем*, які володіють можливостями, традиційно пов'язаними з людським розумом, — розуміння мови, здатність до навчання, міркування, вирішення проблем тощо».

Поряд із цим фахівці зі створення *інтелектуальних систем* схильні до вузького трактування поняття *ІІІ*: «Єдиний біологічний об'єкт, здатний *мислити*, — це *людський мозок*. Тому будь-який «*мислячий*» пристрій повинен якимось чином відтворювати процеси, що відбуваються в мозку людини, або відбивати його структуру». У зв'язку з цим поняття *ІІІ* вони визначають як *комп'ютерну програмно-апаратну модель інтелекту людини*, що здатна брати на себе функції мислення, вибирати й приймати оптимальні рішення на основі аналізу впливів навколишнього середовища і попереднього досвіду, навчатися на основі особливої організації пам'яті, робити висновки на основі суджень тощо [2.1–2.41].

Нині є різні способи й методи оцінки рівня (ступеня) *інтелектуальності* систем *ІІІ* [1.11]. Так, за Колмогоровим, будь-яка система, з якою можна достатньо довго обговорювати проблеми науки, літератури й мистецтва, має інтелект. Іншим прикладом поведінкового трактування *ІІІ* може бути відомий *тест Тьюрінга*, зміст якого полягає в наступному. У різних кімнатах знаходяться люди й комп'ютер. Вони не можуть бачити одне одного, але мають можливість обмінюватися інформацією (наприклад, за допомогою електронної пошти). Якщо в процесі гри люди не здогадуються, що один із учасників — комп'ютер, то можна вважати, що він є інтелектуальним.

2.2. Основні особливості штучного інтелекту

Розроблення комп'ютерних інтелектуальних моделей здійснюється на основі нових досягнень у дослідженнях механізмів пізнання, зв'язаних із процесом опрацювання інформації в мозку людини. Починаючи із 60-х років минулого століття дослідники *ШІ* вели інтенсивні пошуки моделей і методів, які б давали змогу копіювати, імітувати й моделювати внутрішні процеси, що відбуваються у людини, а також процеси, пов'язані зі сприйняттям людиною зовнішнього світу [2.1–2.41]. Складність цих досліджень полягає в тому, що процеси функціонування мозку і життєдіяльності людини важко піддаються формалізації й моделюванню. Людина активна, тобто діє самостійно, постійно змінюючи послідовність своїх дій. Вона нелінійна, бо може реагувати на однакові зовнішні впливи зовсім по-різному. Людина не керується законами формально-логічними, які використовуються при побудові математичних моделей.

Фундаментальні науки про людину поки що не змогли зробити істотного теоретичного внеску в розвиток *ШІ*, який пояснював би (на достатньому для моделювання рівні), як відбуваються інтелектуальні процеси у людини, як улаштована її пам'ять і як вона пізнає навколишній світ. Тому творці *ШІ* розробляли власні моделі людського мислення, за допомогою яких можна знаходити розв'язання виробничих, економічних, медичних та інших інтелектуальних задач у різних областях людської діяльності, де утруднена математична формалізація процесів і явищ. Характерним прикладом такої задачі є моделювання процесів функціонування мозку й життєдіяльності людини в непередбачуваних та екстремальних ситуаціях, що виникають, наприклад, при позаштатних ситуаціях на борті космічного корабля, і можуть призвести до загибелі космонавтів.

На відміну від математичних моделей, моделі людського мислення мають семантичний, тобто змістовний характер. Вони відображають конкретність певної ситуації й конкретні знання фахівців у певній предметній області. Конкретність подається в описовій (вербальній) формі і не може бути подана у вигляді універсальних математичних моделей.

Основними труднощами при моделюванні мислення людини є півкульна асиметрія людського мозку (розд. 1). Наша ліва півкуля логічна, раціональна, тому свідоме мислення лівої півкулі можна представити у вербальній формі і потім змоделювати на комп'ютері. Але права півкуля мислить на рівні почуттєвих образів, тому підсвідоме мислення, яке людина використовує у своїй творчій діяльності, важко піддається моделюванню.

У зв'язку з цим, однією з найважчих проблем, що виникають при створенні *ШІ*, є побудова моделей (виявлення *метапроцедур*) підсвідомого мислення правої півкулі й ув'язування їх із *метапроцедурами* свідомого мислення лівої півкулі. Створені при цьому моделі мислення повинні враховувати спільну роботу обох півкуль, оскільки тільки за цієї умови виникає феномен людського мислення й логічного поводження в життєвих ситуаціях.

Аналізуючи можливість моделювання інтелектуальних здібностей людини (розд. 1), більшість фахівців у області *ШІ* роблять такі висновки [2.1–2.41]:

- штучний «розум» можливий;
- штучний інтелект можливий як комп'ютерна модель мислення людини, оскільки дві інші складові його інтелекту, — *духовність* і *самосвідомість* — важко піддаються моделюванню в практичних застосуваннях;
- штучна мудрість — проблематична.

Частина дослідників думає, однак, що про реалізацію *ШІ* можна буде говорити, коли комп'ютер почне розв'язувати задачі, непосильні для людини, причому зробить за допомогою застосування знайденого ним нового способу мислення.

Оскільки в *ШІ* моделюється інтелект людини, то задачі, що розв'язуються за допомогою засобів *ШІ*, називають інтелектуальними. Зважаючи на це, треба підкреслити, що традиційні обчислювальні задачі, для яких існують математичні методи розв'язання, не є інтелектуальними. Прикладами таких задач можуть слугувати розв'язання систем лінійних алгебраїчних рівнянь, чисельне інтегрування диференціальних рівнянь тощо. Для розв'язання подібних задач є стандартні алгоритми, яка являють собою заздалегідь визначену послідовність елементарних операцій, що може бути легко реалізована як звичайна комп'ютерна програма.

2.3. Основні напрями розвитку штучного інтелекту

У 60-х роках минулого століття відбувся поділ досліджень *ШІ* на два основних напрямки.

1. *Нейробіонічний напрям ШІ* орієнтований на програмно-апаратне моделювання структури людського мозку й створення на цій основі принципово нових технологій і систем опрацювання інформації. Вирішення проблеми розроблення та створення штучного аналога мозку *нейрокібернетики* шукають шляхом створення й реалізації просторової структури так званих *нейронних мереж* (розд. 10) [2.31–2.34].

Принциповим питанням для цього напрямку є з'ясування співвідношення між *будовою* мозку і нервовою системою людини та їх *функцій*. При цьому моделювання здійснюється на кількох рівнях: молекулярному, клітинному, на рівні окремого органа, організму в цілому, і далі — на рівні соціальної групи. Таким чином, класичний нейробіонічний підхід полягає у послідовному ускладненні моделей через просування від елементарних форм до більш складних форм моделювання.

Основним досягненням у сфері нейробіонічних досліджень *III* стало створення *нейрокомп'ютера* — інтелектуального комп'ютера *VI* покоління. Його основу становить *нейромережа* як безліч *мікропроцесорів*, яка здатна вирішувати достатньо складні інтелектуальні задачі (розд. 10).

2. *Кібернетичний напрям III* пов'язаний зі створенням та використанням так званих *евристичних* (*емпіричних, інтуїтивних*) *моделей мислення* (розд. 4–9) [2.1–2.30]. Основою цих моделей є *евристики*, що являють собою правила або *рекомендації*, що не мають строгого теоретичного (математичного) обґрунтування. Їх установлюють досвідчені фахівці (експерти) і використовують під час *вербалізації* (неформального словесно-змістовного опису) відповідної предметної (проблемної) області. *Евристичне моделювання* властиве людському мисленню, для якого характерне створення гіпотез (про шлях розв'язання задачі) з їх наступним експериментальним перевірками.

Прихильники *кібернетичного напрямку в III* мотивували свій підхід тим, що людина не повинна сліпо слідувати природі у своїх наукових і технологічних пошуках. Наприклад, очевидний успіх колеса, якого немає в природі, або літака, що не махає крилами, як птах.

Кібернетичний підхід є класичним для кібернетики з одним із її базових понять — «*чорною скринькою*», яка являє собою пристрій або програмний модуль із невідомою внутрішньою структурою та змістом. Відомі лише (доступні для виміру) вихідні реакції цього об'єкта на вхідні впливи. Цієї інформації достатньо для побудови моделі досліджуваного об'єкта, що в даному випадку і є «*чорна скринька*».

Крім основної тенденції, орієнтованої на створення і використання інтелектуальних систем і комп'ютерів (у межах *нейробіонічного та кібернетичного напрямів*), нині в економічно розвинутих країнах світу (насамперед, у США і Японії) проводяться *фундаментальні наукові дослідження* в області *III*. Основними напрямками теоретичних досліджень є [2.26–2.41]:

- розроблення методів і моделей формалізації та подання знань;
- створення способів поповнення, узагальнення й класифікації знань;

- створення процедур ефективного пошуку, вибору й оцінювання знань;
- організація спілкування з комп'ютером мовою, близькою до природної (створення діалогових процедур, питально-відповідних систем, систем аналізу й синтезу мови тощо);
- організація спілкування з комп'ютером по зоровому каналу (аналіз і синтез зображень, формалізація зорових уявлень тощо);
- розроблення схем людських міркувань і алгоритмів висновку на знаннях;
- планування доцільної діяльності й автоматичний синтез програм із готових модулів;
- розроблення засобів обґрунтування й пояснення одержуваних рішень тощо.

Об'єктами *наукових досліджень* є структура й механізми роботи мозку людини, а кінцева мета полягає в *розкритті таємниць мислення*. Неодміними етапами досліджень у цьому напрямі є побудова й експериментальне дослідження моделей мислення з використанням психофізіологічної інформації, висування нових гіпотез щодо механізмів інтелектуальної діяльності, удосконалення моделей подання знань тощо.

Нові області застосування методів і засобів *III* постійно висвітлюються у працях спеціалізованих конференцій, що публікуються на сторінках журналів *AI Magazine*, *Communications of the Association for Computer Machinery*, *Artificial Intelligence* та ін. [2.35–2.37]. Значний обсяг інформації з проблем *III* міститься в *Internet* на сайті *Journal of Artificial Intelligence Research* [2.38], на сервері *American Association for Artificial Intelligence* [2.39], а також у сховищах університетських лабораторій США, що займаються проблемами *III* [2.40, 2.41].

Розділ 3

ПРИНЦИПИ ПОБУДОВИ ТА ФУНКЦІОНУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

3.1. Основні характеристики інтелектуальних систем

На відміну від традиційних обчислювальних систем, *інтелектуальні системи* мають справу з такими об'єктами (процесами, явищами, ситуаціями), для яких важко чи неможливо побудувати математичну модель, необхідну для вирішення задач у комп'ютері (див. Вступ). В інтелектуальних системах використовуються *евристичні моделі людського мислення*, застосовувані в *евристичних алгоритмах* вирішення інтелектуальних задач (розд. 2.4–2.8).

Евристикою називають неформальне правило, що впливає з людського досвіду і не має строгого (теоретичного, математичного) обґрунтування і використовується в інтелектуальній системі для подання інформації.

З урахуванням зазначених чинників *комп'ютерну СИ* (КСІ) можна охарактеризувати як комплекс взаємозалежних інтелектуальних комп'ютерних засобів, призначених для вирішення інтелектуальних задач у певній предметній області на підставі використання моделей людського мислення, спілкування, пам'яті [3.1–3.25]. *Інтелектуальною задачею* називають проблему, що виникає перед людиною у складній життєвій ситуації.

КСІ здатна:

- 1) сприймати поточну інформацію про зовнішній світ і свій внутрішній стан;
- 2) аналізувати сформовану ситуацію і здійснювати доцільні дії, спрямовані на розв'язання поставленої задачі;
- 3) вирішувати інтелектуальні задачі, що важко формалізуються;
- 4) безпосередньо впливати (за потреби) на об'єкти зовнішнього світу за допомогою відповідних виконавчих механізмів і пристроїв;
- 5) самонавчатися, нагромаджувати нові знання в процесі вирішення інтелектуальних задач і адаптуватися до умов предметної (проблемної) області, в якій вона працює.

КСІ може бути пов'язана із системою *Internet* і одержувати звідти необхідні знання світового рівня.

КСІ належить до класу *відкритих систем*, основними властивостями яких є:

- 1) *розширюваність (масштабованість)* — можливість зміни складу системи в процесі її функціонування;
- 2) *мобільність (переносимість)* — можливість переносу програмного забезпечення системи на різні апаратно-програмні платформи;
- 3) *інтероперабельність* — здатність до взаємодії з іншими системами;
- 4) *дружній інтерфейс* із користувачем і простота спілкування із системою.

КСІ істотно відрізняються від традиційних обчислювальних систем принципом роботи й іншими характеристиками (табл. 3.1).

Таблиця 3.1 — Порівняльні характеристики обчислювальних та інтелектуальних систем

Характеристики	Обчислювальні системи	Інтелектуальні системи
Вид опрацьовуваної інформації	Дані	Знання
Тип розв'язуваних задач	Математичні	Інтелектуальні
Модель подання інформації	Математична	Евристична
Спосіб вирішення задач	Програмний	Висновок на знаннях
Спосіб опрацювання інформації	Числовий	Символьний
Одержувані рішення задач	Оптимальні	Евристичні

Основними властивостями інтелектуальності КСІ, якими вони відрізняються від традиційних комп'ютерних систем, є [3.1–3.25]:

- 1) *символьне (евристичне) подання інформації* у вигляді, близькому до людського розуміння (розд. 4–6);
- 2) *символьне опрацювання інформації* у формі, близькій до людського мислення (розд. 7);
- 3) *здатність розв'язувати інтелектуальні задачі, що важко формалізуються або не формалізуються*, за допомогою використання *евристичних моделей мислення й алгоритмів виводу на знаннях* (розд. 7, 8);
- 4) *здатність до самонавчання*, тобто до автоматичного поповнення й одержання нових знань на основі нагромадженого системою досвіду аналізу й вирішення інтелектуальних задач користувачів;

5) здатність до *адаптації (приспосованості)* системи до об'єктивних змін *предметної (проблемної) області* функціонування системи;

6) розвинуті *комунікативні здібності*, що дає змогу інтенсивний *діалог із людиною* (у процесі постановки й вирішення *інтелектуальних задач*) *природною мовою, близькою до людського спілкування*;

7) широке застосування *інтелектуальних комп'ютерних засобів когнітивної графіки*, що використовує колір, анімацію, відео, аудіо та інші засоби *інтелектуального інтерфейсу для наочного* (інтегрованого) подання властивостей досліджуваного об'єкта, процесу чи явища.

Основними вимогами до КСШ при вирішенні *інтелектуальних задач* є такі:

- 1) *мінімальний час* вирішення задач (одержання результатів);
- 2) *мінімальна трудомісткість* процесу вирішення;
- 3) *мінімальна вартість* опрацювання інформації під час розв'язання задач;
- 4) вироблення рішення в умовах неповної інформації про розв'язувану задачу тощо.

У наступних розділах розглянуті основи побудови й функціонування різних типів КСШ, наведені їхні основні характеристики та області практичного застосування.

3.2. Типи сучасних інтелектуальних систем

Сьогодні найбільшого *практичного застосування* набули такі *типи інтелектуальних систем* [3.1–3.25].

Експертна система (ЕС) (розд. 9) створюється за допомогою *експерта-професіонала* в тій *предметній (проблемній) області*, в якій працює система. Знання експерта закладаються в *базу знань* системи (розд. 5), яка використовується під час вирішення задач користувачів.

Нейрокомп'ютерна система (розд. 10) являє собою *нейромережу* у вигляді безлічі *мікропроцесорів*, з'єднаних між собою відповідно до моделі взаємодії нейронних клітин мозку людини (розд. 2). Основою цієї системи *нейрокомп'ютер* — аналог *перцептрона* на сучасному етапі розвитку інтелектуальних систем цього типу.

Робототехнічна інтелектуальна система (розд. 11) являє собою комп'ютерно-технологічний комплекс, оснащений інтелектуальним комп'ютером, який імітує людські дії та поведінку в навколишньому світі. Система призначена для *автоматизації людської праці у шкідливих та небезпечних умовах зовнішнього середовища*.

Інтелектуальна система прийняття рішень (розд. 12) здійснює *планування і прийняття рішень* щодо керування об'єктами і процесами у складних (екстремальних) умовах за допомогою застосування моделей людського мислення і поведінки.

Інтелектуальна інструментальна система (WorkBench Tools) являє собою програмно-апаратний комплекс, призначений для *проектування, створення й експлуатаційного супроводу інтелектуальних систем* різноманітного призначення (див. вище). Нині найширше використовують *інструментальні системи* створення *експертних систем* (розд. 9.4).

Поряд із розглянутими КСШ в економічно розвинених країнах світу (головним чином, у США і Японії) інтенсивно розробляються *перспективні інтелектуальні системи*. Вони поки що не набули такого широкого *практичного застосування* як попередні КСШ і перебувають у стадії становлення і розвитку. До *перспективних* належать такі *інтелектуальні системи*.

Мультиагентна система (розд. 13) являє собою комп'ютерну систему, яка складається з так званих *інтелектуальних агентів*, кожний з яких є *програмно-виконавчим пристроєм*, що вирішує спеціальну інформаційну задачу. Ці системи призначені для оперативного одержання інформації з різних інформаційних систем і мереж (головним чином, з *Internet*) та розв'язання інтелектуальних інформаційних задач спеціального призначення.

Нечітка інтелектуальна система (розд. 14) розв'язує інтелектуальні задачі за допомогою так званих *м'яких обчислень*, основу яких становлять методи *нечіткої логіки*, що оперують із *недостовірною й неповною* інформацією про досліджувані об'єкти, процеси і явища навколишнього світу.

Генетична інтелектуальна система створюється на концепціях *еволюції (природного добору)* і використовує механізм *генетичного спадкування* у природі. В основу функціонування таких систем покладені *генетичні алгоритми* рішення інтелектуальних задач (розд. 8.2, 8.3).

Онтологічна система являє собою ієрархічну сукупність *онтологій* — баз знань *спеціального призначення* (розд. 5.2.). Система призначена для пошуку і видобування так званих *концептуальних знань* (про різні області людської діяльності), що можуть надходити з різних систем і мереж, головним чином — з *Internet*.

Система керування знаннями призначена для пошуку й опрацювання знань в окремо взятій *корпорації* (фірмі, компанії, організації). Підґрунтям такої системи є *корпоративні знання (корпоративна пам'ять)*, яка, подібно до *людської пам'яті*, дає змогу використовувати попередній досвід роботи корпорації та не допускати повторення помилок і дублювання знань (розд. 4.3).

3.3. Структура та функції перспективної інтелектуальної системи

У структурі перспективної КСШІ можна виділити такі **функціональні підсистеми** (рис. 3.1) [3.1–3.25] (розд. 9–12).

3.3.1. Підсистема розв'язання на базі наявних знань

Основу КСШІ становить підсистема *інтелектуальних програмно-апаратних комп'ютерних засобів*, призначених для *вирішення інтелектуальних задач та вироблення рішень* щодо впливу на об'єкти зовнішнього середовища і робочі органи КСШІ. До складу підсистеми входять наступні *програмно-апаратні пристрої* (рис. 3.1).

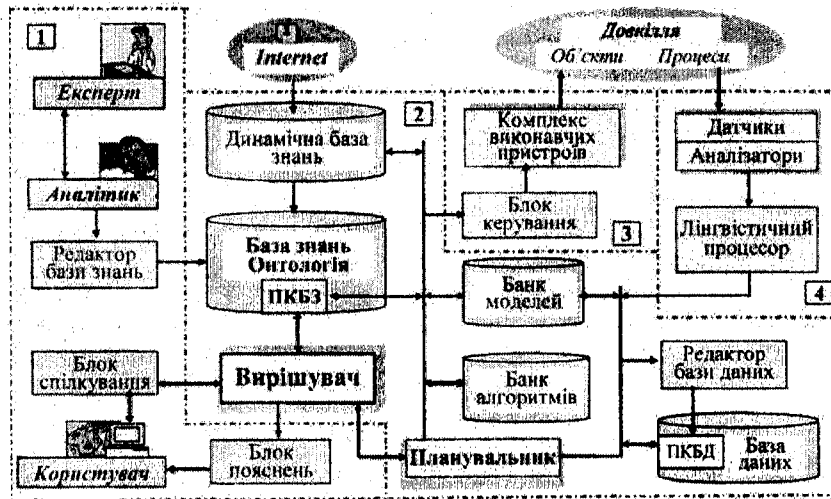


Рис. 3.1 — Узагальнена структурна схема перспективної системи штучного інтелекту

1. Підсистема інтелектуального інтерфейсу.
2. Підсистема виводу на знаннях та рішення задач.
3. Підсистема керування виконавчими пристроями.
4. Підсистема збору та первинного опрацювання інформації.

1) **Розв'язувач** (інтелектуальний комп'ютер) здійснює безпосереднє рішення інтелектуальних задач, використовуючи моделі людського мислення, інтелектуальні механізми опрацювання символічної інформації, моделі подання знань та інтелектуальні засоби виводу на знаннях (розд. 6–8).

Вирішувач виконує такі функції:

- формує (ставить) розв'язувану задачу на основі аналізу вхідної інформації, що міститься в запиті користувача або надходить у КСШІ з інших джерел зовнішнього середовища;
- перетворює розв'язувану задачу у внутрішнє подання системи;
- вибирає (з банку моделей) відповідні моделі мислення й подання знань, необхідні для розв'язання поставленої задачі;
- формує (знаходить у банку алгоритмів) найпридатніший (ефективний) алгоритм розв'язання поставленої задачі;
- зіставляє зміст (факти і правила) розв'язуваної задачі зі знаннями про предметну (проблемну) область, що містяться у базі знань (розд. 5);
- вибирає з бази знань і заносить у свою робочу пам'ять знання, необхідні для розв'язання задачі;
- вирішує поставлену задачу шляхом спільного використання обраних моделей мислення, подання знань і алгоритму розв'язання задачі;
- додає в базу знань нові знання, якщо вони є;
- результат вирішення задачі видає у блок прийняття рішень КСШІ або користувачу через інтерфейс.

Уся поточна інформація, одержувана в процесі вирішення задачі, зберігається в **робочій пам'яті** (пам'яті станів) вирішувача і містить у собі:

- вихідні дані про задачу (запит користувача, відомості про зовнішню ситуацію тощо);
- знання, моделі й алгоритми, необхідні для вирішення задачі;
- зміст діалогового спілкування з користувачем;
- пояснення результатів рішення задачі, виданих за вимогою користувача;
- власні коментарі вирішувача з приводу розв'язуваної задачі й інші зведення.

Робоча пам'ять вирішувача може бути розміщена в одному з розділів бази даних системи (див. нижче).

Перспективні КСШІ мають у своєму складі **паралельний вирішувач**, що складається з кількох вирішувальних процесорів, які паралельно (одночасно) вирішують кілька задач у мультипрограмному режимі. Поряд із традиційними програмними засобами логіки предикатів, семантичних мереж, фреймів, продукційних та інших моделей (розд. 6), паралельний комплексний (універсальний) вирішувач здатний оперувати одночасно з кількома моделями подання знань і мовами програмування, здійснювати м'які обчислення з використанням методів нечіткої логіки (розд. 14), застосовувати для вирішення задач еволюційні, прецедентні та інші алгоритми (розд. 8).

У разі потреби розпізнавання конфігурацій об'єктів складної структури КСШ може бути обладнана *нейрокомп'ютером*, у якому реалізується *нейромережевий* підхід до вирішення інтелектуальних задач (розд. 10). Використання комбінованих програмних засобів підвищує ефективність роботи вирішувача і забезпечує необхідну якість вирішення задач і прийняття рішень.

Основною функціональною характеристикою вирішувача є його *продуктивність*, обумовлена швидкістю вирішення задачі і часом між запитом користувача і наступною відповіддю системи. Для оцінювання продуктивності вирішувача використовується *одиниця виміру швидкості опрацювання* символічної інформації *LIPS* — один логічний висновок у секунду, що відповідає 100–1 000 командам звичайного комп'ютера. Продуктивність сучасних вирішувачів коливається в межах 200¹–500 *KLips*.

2. *База знань (основна)* зберігає знання про *предметну (проблемну) область*, у якій працює КСШ, і забезпечує вирішувач інформацією, необхідною для вирішення задач (розд. 5.1).

3. *Програма керування базою знань (ПКБЗ)* керує функціонуванням бази знань, здійснює вибір правил із бази знань для вирішувача, а також внесення в базу знань *нових знань*, що надходять (видобуваються) із різних джерел інформації, серед них — отриманих у діалоговому процесі спілкування з користувачами при розв'язанні конкретних задач. Функції ПКБЗ в окремих КСШ виконує *планувальник* системи.

4. *Динамічна база знань* зберігає, нагромаджує й поновлює *оперативну* інформацію, одержувану із зовнішніх джерел інформації (головним чином — із *Internet*) для вирішення задач у режимі *реального часу*, взаємодіє з *основною базою знань* під керуванням власної ПКБЗ чи *планувальника* системи. Може входити до складу *основної бази знань* як її окрема *підобласть*.

5. *Онтологія (онтологічна система)* являє собою сховище *інтегрованих концептуальних знань про зовнішній світ* (розд. 5.2). В онтології зберігаються знання (*концепти*) із *багатьох предметних областей*, які надходять, головним чином, із системи *Internet* і використовуються для вирішення задач *цієї предметної області*, в якій працює КСШ. Може входити до складу *основної бази знань* як окрема *підобласть знань*.

6. *Банк моделей* містить моделі подання знань, інформаційні моделі зовнішнього середовища і внутрішнього стану КСШ, які забезпечують інформаційну підтримку вирішення задач і функціонування КСШ (розд. 6). Може входити до складу *основної бази знань* як окремі *розділи (банку)*.

7. *Банк алгоритмів* призначений для зберігання *алгоритмів* розв'язання задач і функціонування КСШ (розд. 7, 8). Може входити до складу *основної бази знань* як окремі *розділи (банку)*.

8. *Операційна система* визначає порядок функціонування системи і розподіляє інформаційні потоки між її функціональними компонентами. У КСШ використовуються *спеціалізовані* операційні системи, орієнтовані на роботу в *інтелектуальному середовищі* програмування.

9. *Планувальник (диспетчер)* здійснює безпосереднє керування процесом вирішення задач та їх реалізацією. Зазвичай входить до складу *операційної системи*. Функції планувальника в багатьох (зокрема — в експертних) системах виконує *розв'язувач*.

10. *База даних (БД)* зберігає загальні характеристики (зокрема — *словник*) предметної області, а також інформацію про власні робочі параметри й можливості КСШ (розд. 5.4). У БД може розміщуватися *оперативна робоча пам'ять вирішувача*, необхідна для збереження проміжних результатів розв'язуваної задачі. БД може входити до складу *основної бази знань* як окремі структурні (розд. 5.4).

Слід зазначити, що термін «БД» тут збігається за назвою, але не за змістом, із терміном, використовуваним у традиційних СУБД та інформаційно-пошукових системах.

11. *Програма керування БД (ПКБД)* взаємодіє з ПКБЗ і за її командою надає вирішувачу відповідну інформацію, необхідну для вирішення задач. Функції ПКБД у багатьох системах виконує ПКБЗ.

3.3.2. Підсистема інтелектуального інтерфейсу

Підсистема здійснює інформаційну підтримку *інтерактивної діалогової взаємодії людини з КСШ* в процесі введення-виведення інформації та розв'язання задач. Основною *властивістю інтелектуальності* інтерфейсу є його здатність здійснювати діалог із користувачем природною мовою, максимально наближеною до *людського спілкування*. Крім того, до складу підсистеми входять програмно-апаратні засоби *когнітивної графіки* та інші засоби *інтелектуального інтерфейсу*, що забезпечують ефективну взаємодію користувача з КСШ.

Тут і далі під терміном «*користувач*» розуміють всі суб'єкти (інженер по знаннях, експерт, програміст, зовнішній користувач та ін.), хто може входити в діалоговий режим із КСШ.

До складу *підсистеми інтелектуального інтерфейсу* входять такі програмно-апаратні пристрої (див. рис. 3.1):

1. **Блок придбання знань** здійснює зв'язок (інтерфейс) ПКБЗ, ПКБД, динамічної бази знань, онтологічної системи, банків моделей і алгоритмів з інформаційними джерелами зовнішнього світу і виконує такі функції:

- одержання знань від досвідчених фахівців (експертів), із публікацій, електронних документів системи Internet та інших джерел інформації;
- перетворення отриманої інформації у потрібну форму для введення у бази знань і онтології КСШІ;
- додавання нових знань у бази знань і онтології;
- модифікацію й приведення наявних знань до такого вигляду, який дає змогу користувачу застосовувати їх у процесі роботи КСШІ;
- перевіряння знань, що вводяться і модифікуються, на несуперечність із наявними знаннями.

2. **Блок спілкування** здійснює безпосередню діалогову взаємодію користувача з КСШІ в процесі введення-виведення інформації й вирішення задач. Він виконує такі функції:

- надає можливість людині-непрограмісту ставити і вирішувати інтелектуальні задачі на *природній (людській) мові*;
- дає змогу користувачу *корегувати сценарій (механізм) взаємодії* із системою чи створювати власний сценарій спілкування;
- забезпечує *дружній інтерфейс* із користувачем, максимально наближений до людського спілкування;
- дає змогу користувачу вводити будь-яку інформацію у будь-який момент діалогу;
- урахує ступінь упевненості користувача у вірогідності інформації, що вводиться;
- висуває гіпотези про можливі варіанти вирішення задачі, а потім ставить користувачу запитання, щоб довести або спростувати їх;
- може відкидати свої гіпотези і вибирати нові на підставі відповідей користувача на свої запитання;
- перефразовує свої запитання, якщо вони незрозумілі користувачу;
- шукає суперечності у відповідях користувача, вказує на них і дає змогу користувачу змінювати і старі, і нові відповіді, щоб домогтися їхньої несуперечності.

Під час роботи блок спілкування може звертатися до користувача за необхідною додатковою інформацією і видавати на його вимогу проміжні дані розв'язуваної задачі. Таким чином, користувач бере безпосередню участь у процесі її розв'язання.

Особливістю інтелектуального інтерфейсу є його можливість давати відповіді на заздалегідь непередбачувані і неформалізовані запитання користувача. Блок спілкування здатний також розпізна-

вати ситуацію непорозуміння між користувачем і системою і відповідним чином на неї реагувати. Наприклад, збою системи не відбувається, навіть якщо користувач ставить безглузде запитання.

3. **Блок пояснень** видає інформацію користувачеві (на його вимогу) про шлях вирішення його задачі (виконання *запиту*) і пропонує свої висновки разом зі зведенням розуміння, на підставі яких КСШІ прийшла до цих висновків. Якщо користувача не цікавить повне рішення, то блок пояснень вибирає з його рішення тільки ключові моменти з урахуванням їх важливості та рівня знань користувача.

Здатність системи пояснити користувачеві, чому прийняте саме таке рішення та внаслідок яких міркувань система дійшла до того чи іншого висновку, називають *прозорістю* системи. Одержавши вичерпну інформацію про хід вирішення своєї задачі, користувач може з довірою поставитися до отриманого результату.

У блоці пояснень широко використовуються засоби *інтерактивної когнітивної графіки*, що дають змогу подавати хід вирішення задачі як графічні образи на екрані дисплея, що забезпечують *прозорість* процесу розв'язання задачі, і може бути пояснений користувачу на якісному рівні.

Наприклад, зміна кольору зображення (у разі зміни поточного стану процесу вирішення) і використання анімації (переміщення або обертання образу на екрані дисплея) дає змогу користувачу зрозуміти й оцінити логіку вирішення своєї задачі, не вивчаючи моделі та алгоритми, а розглядаючи лише графічне зображення процесу рішення на екрані.

4. **Інтелектуальний редактор бази знань** містить засоби когнітивної графіки й інтелектуального інтерфейсу і дає змогу *інженеру по знаннях (аналітику)* КСШІ створювати і редагувати *базу знань* у діалоговому режимі. Може входити до складу блоку *придбання знань*.

5. **Інтелектуальний редактор бази даних** дає змогу інженеру-аналітику створювати базу даних у діалоговому режимі. Містить вкладені меню, шаблони, підказки та інші сервісні засоби когнітивної графіки, що полегшують роботу з базою даних. Функції *редактора бази даних* може виконувати *редактор бази знань*.

3.3.3. Підсистема збору і первинного опрацювання інформації

Підсистема забезпечує одержання і первинне перетворення даних вимірів і спостережень, що надходять по *каналах сприйняття* інформації про *навколишнє середовище* і *внутрішній стан* КСШІ (див. рис. 3.1). Підсистема характерна для *робототехнічних систем* і містить у собі такі основні *апаратно-програмні пристрої* (розд. 11):

- *датчики й аналізатори* забезпечують одержання і первинне перетворення (*моделювання*) інформації (даних вимірів і спостережень), що надходить із довкілля;
- *лінгвістичний процесор* здійснює *сприйняття (аналіз)* і *генерацію (синтез)* символічної інформації. У процесі *аналізу* реалізується перетворення «сигнал — текст», а при *синтезі* виконується зворотне перетворення «текст — сигнал».

3.3.4. Підсистема керування виконавчими пристроями

Підсистема здійснює перетворення *отриманих системою рішень у команди (інструкції)*, які по каналах керування передаються до виконавчих пристроїв і механізмів КСШ (див. рис. 3.1). Підсистема характерна для *робототехнічних систем* (розд. 11), але може бути присутня і в інших КСШ.

До складу підсистеми входить *блок керування*, що перетворює отримані системою рішення в команди, які по каналах керування передаються до *виконавчих пристроїв і механізмів*. Останні реалізують вироблені системою керуючі впливи на об'єкти і процеси зовнішнього середовища.

Необхідно зазначити, що подана в даному розділі структурна схема КСШ є *узагальненою*, оскільки конкретні типи КСШ можуть не мати тих чи інших компонентів, зазначених на рис. 3.1. Наприклад, *статичні експертні системи* (розд. 9.1) не мають онтологій, виконавчих пристроїв і зв'язку із довкіллям і взаємодіють лише з користувачами.

3.4. Склад розробників інтелектуальної системи

Для вирішення проблем проектування, створення й експлуатації інтелектуальних систем необхідні високоєрудовані фахівці, що мають сукупні знання в області КСШ і когнітивної графіки, досвід розроблення сучасних засобів комп'ютерної техніки, систем програмування і неформальних (евристичних) методів подання й опрацювання інформації. У створенні й експлуатації КСШ беруть участь такі фахівці [3.1–3.25] (розд. 4, 5, 9).

1. *Головним фахівцем* зі створення й експлуатації КСШ є *інженер по знаннях (інженер знань)*, якого називають також *аналітиком, інженером-інтерпретатором, когнітологом*. *Інженер по знаннях* (далі для стислості — *аналітик*) виконує такі функції:

- очолює роботи з проектування й розроблення КСШ (якщо вона створюється заново);

- відповідає за добування, формалізацію, структурування і якість інформації, що вводиться до бази знань системи (розд. 4.2);
- разом із програмістом (див. нижче) вибирає (розробляє) інструментальні засоби подання інформації, мови програмування й середовище функціонування системи;
- забезпечує підтримку системи в робочому стані в процесі її експлуатації.

2. *Експерт* — професіонал у даній *предметній (проблемній) області*, який за допомогою *аналітика* здійснює наповнення бази знань і контролює правильність введення в неї інформації (розд. 4.2). При створенні КСШ потрібен експерт, який:

- знає, як вирішуються задачі (проблеми) системи;
- здатний пояснити аналітику способи вирішення задач;
- має у своєму розпорядженні достатній час для роботи із системою;
- має спонукальні мотиви до активної участі у створенні системи.

У процесі розроблення і наступного розвитку КСШ аналітик і експерт працюють разом. Аналітик повинен одержати знання від експерта, *структурувати* їх і подати в *ієрархічному* вигляді, «зрозумілому» для системи. Процес добування знань складний і тривалий, тому що експерт часто не усвідомлює, якими знаннями він користується або не може їх *вербалізувати* (змістовно виразити) (розд. 4.2).

3. *Програміст* відповідає за вибір (розроблення) основних частин *системного* програмного забезпечення КСШ. З наявних нині засобів програмування і подання інформації програміст повинен *вибрати* найефективніш, що є дуже непростою й трудомісткою проблемою.

Програміст працює під керівництвом *аналітика* і виконує такі функції:

- вибирає (розробляє) моделі подання знань і мови програмування, найпридатніші для цієї системи;
- здійснює вибір інструментальних засобів, необхідних для створення й функціонування системи;
- визначає й програмує стандартні функції програмного забезпечення системи;
- здійснює сполучення програмного забезпечення із середовищем функціонування системи тощо.

Слід зазначити, що в розробленні КСШ програміст відіграє допоміжну роль, на відміну від *традиційних* обчислювальних систем, де *системний програміст* є ключовою фігурою в розробленні як *системного*, так і *прикладного* програмного забезпечення.

У процесі створення КСШ можуть брати участь кілька експертів, інженерів по знаннях і програмістів. *Головний інженер по знаннях* може залучати інших експертів для того, щоб переконатися:

- у правильності свого розуміння *основного експерта*;

- у збігу поглядів різних експертів щодо якості пропонуваного рішення зі створення системи;
- у повноцінності тестів, що визначають можливості створюваної системи тощо.

3.5. Технологія створення інтелектуальної системи

На сучасному етапі промислова технологія створення КСШІ складається з таких етапів (рис. 3.2) [3.1–3.25] (розд. 5, 9–12).



3.5.1. Ідентифікація

На етапі *ідентифікації* здійснюється формування вимог до КСШІ та осмислення тих задач, які вона буде вирішувати. На цьому етапі складається *вербальний* (змістовний) опис цілей, принципів побудови, основних задач КСШІ, зокрема визначаються:

- основні характеристики створюваної системи;
- обсяг фінансування й термін розроблення системи;
- *склад розроблювачів* системи й форми їхніх взаємин;
- наявні *ресурси* для створення системи та джерела інформації;
- сфера застосування системи та класи *розв'язуваних задач*;
- типи *користувачів* та області їхніх інформаційних потреб;
- *критерії ефективності* роботи системи тощо.

Правильний вибір *цілей і задач КСШІ* являє собою найбільш критичну частину розроблення системи, оскільки в процесі її створення можна «загрузнути у болоті» нерозв'язуваних задач. Невдало визначена мета може призвести до створення системи, що буде непрацездатною, збитковою чи неприйнятною для користувача.

На цьому етапі аналітик (інженер по знаннях) і експерт працюють у тісному контакті. Вони складають неформальний опис предметної області, уточнюють терміни й ключові поняття, що вводяться до бази знань. Експерт корегує опис предметної області, пояснює, як вирішувати її задачі та які міркування лежать в основі того чи іншого рішення. Після кількох циклів уточнення й корегування аналітик і експерт одержують остаточний опис предметної області (розд. 5.1).

На етапі *ідентифікації* використовується принцип *кооперативного проектування*, відповідно до якого поряд із аналітиками, експертами й програмістами в процесі створення системи беруть участь *кінцеві користувачі*. Вони мають неформальне розуміння розв'язуваних задач і можуть більш якісно проаналізувати опис предметної області.

3.5.2. Концептуалізація

На цьому *етапі* розробляються *концепції* (основні положення, ідеї) *реалізації* тих *цілей і задач КСШІ*, що були визначені на етапі *ідентифікації*. Проводиться змістовний аналіз предметної області і визначаються (виявляються):

- основні напрями вирішення загальної проблеми створення системи;
- загальні задачі проблеми;
- типи взаємозв'язків між об'єктами, явищами та процесами предметної області;
- можливі гіпотези, стратегії й методи вирішення задач;
- види і склад використовуваних знань, ступінь їх повноти, вірогідності, погодженості й надмірності;
- ключові поняття, відносини й характеристики (*словник*) предметної області;
- вхідна й вихідна інформація системи;
- обмеження, що накладаються на процес вирішення задач і функціонування системи;
- тести для перевіряння роботи системи тощо.

Етап концептуалізації завершується створенням *моделі предметної області* (розд. 5.1).

Для визначення наведених вище характеристик складається детальний *протокол концептуальних рішень* зі створення КСШІ, що

узгоджується з усіма учасниками її розроблення. Адекватним засобом для реалізації цих рішень є *відеодіаграми*, що створюються на екрані дисплея і використовуються для проектування, супроводу і документування, а також для організації взаємодії між учасниками процесу створення системи. На відеодіаграмах відображається структура системи та її складові частини (підсистеми), а також оператори, основні блоки виміру й керування, об'єкти зовнішнього оточення й основні потоки інформації.

3.5.3. Формалізація

На цьому *етані* всі ключові поняття і відносини, установлені в процесі *концептуалізації*, описуються деякою *формальною мовою*, яка вибирається (аналітиком і програмістом) із числа наявних або створюється заново. Здійснюється *структуризація* опрацьовуваної інформації у базі знань КСШ і визначаються (вибираються або розробляються):

- інструментальні засоби створення системи;
- ефективні алгоритми вирішення задач;
- адекватні моделі подання інформації (знань);
- програмне середовище функціонування системи;
- мови програмування та інші інформаційні засоби.

Підсумком виконання етапу формалізації є визначення складу засобів і способів подання, інтерпретації й маніпулювання знаннями, а також формальний опис усього процесу функціонування КСШ.

3.5.4. Реалізація

На цьому *етані* створюються один або кілька *дослідних зразків КСШ* (в експертних системах — *прототипів*, які являють собою *початкові версії* майбутньої системи (розд. 9.3). Виконується *первинне наповнення* бази знань і налаштування механізму виводу на знаннях із використанням обраних інструментальних засобів.

На етапі *реалізації* здійснюється *перевіряння* правильності обраних ідей і *принципів побудови системи*, а також методів і способів подання інформації у базі знань. Перевіряється *відповідність* розроблених *зразків* пред'явленим до них *вимогам* і виконуються такі роботи:

- тестування зразків;
- доопрацювання й перевіряння працездатності розширених версій зразків;

- вибір із розроблених зразків *головного зразка* й іспит його в промислових умовах;
- прив'язка *головного зразка* до реального робочого середовища;
- доведення *головного зразка* до стану досвідченої КСШ.

У разі потреби виконується часткове або повне перепрограмування програмного забезпечення системи.

3.5.5. Налагодження й тестування

На цьому *етані* здійснюється перевіряння *відповідності* розробленої системи (її *робочого головного зразка*) *вимогам* технічного завдання на її створення. Перевіряння виконують за допомогою *тестування* функцій системи у всіх можливих режимах її роботи. Є такі основні методи тестування:

- а) *тестування* ситуацій, що можуть виникнути під час розв'язання задач користувачів;
- б) *логічне тестування бази знань* для виявлення в ній надлишкових, циклічних, конфліктних, пропущених і пересічних правил, неузгоджених умов тощо;
- в) *концептуальне тестування* загальної структури системи й урахування в ній усіх аспектів розв'язуваних задач.

Особлива увага приділяється підбору тестових прикладів. У них повинні знайти відображення такі випадки:

- неправильно сформульовані запити користувачів;
- присутність невизначеності в запитах користувачів;
- доступність для користувача знань системи;
- доступність для користувача пояснень, які видає система;
- суперечливість і неповнота знань;
- погодженість діалогової взаємодії користувачів із системою та інші ситуації.

На цьому етапі розробники системи залучають до тестування користувачів різної кваліфікації, що можуть звертатися до системи з різноманітними запитами. Кінцевий користувач також може взяти участь у тестуванні. Вони аналізують здатність системи налаштуватися на рівень кваліфікації та визначати потреби користувача під час діалогу, виявляти й усувати причини його невдач, помилок у роботі тощо.

За результатами етапу може здійснюватися *вдосконалення головного зразка* системи шляхом повернення до етапу реалізації чи *реструктуризації* бази знань та її програмного забезпечення системи (див. рис. 3.2).

3.5.6. Дослідна експлуатація та вдосконалення систем штучного інтелекту

На цьому етапі визначається *придатність* КСШІ для кінцевого користувача за допомогою розв'язання різних можливих задач на *робочих місцях користувачів замовника*. Придатність системи для користувача визначається *зручністю* роботи з нею та її *корисністю*.

Під *зручністю роботи* маються на увазі наступні чинники:

- *природність взаємодії* із системою, тобто спілкування у звичному, (такому, що не стомлює користувача) діалозі із системою;
- *гнучкість* системи, тобто здатність налаштовуватися на різних користувачів, які мають неоднаковий ступінь підготовки до роботи із системою;
- *стійкість до помилок*, тобто здатність системи не виходити з ладу у разі помилкових дій користувача.

Під *корисністю* системи розуміється здатність системи *вирішувати поставлені задачі*, задовольняти потреби користувача, а також виявляти й усувати (під час діалогу) причини збоїв і невдач у роботі системи.

У процесі розроблення КСШІ майже завжди здійснюється її модифікація. Найпростішим її видом є *вдосконалення промислового зразка*, тобто повернення до етапу *реалізації* (див. рис. 3.2). Більш серйозним видом модифікації може бути необхідність *реструктуризації* бази знань і програмного забезпечення системи. Цей вид модифікації необхідний у тому випадку, якщо виявиться, що бажане функціонування системи не досягнуто. Здійснюється повернення на етап *формалізації* для *перепроєктування* розробленої системи.

Якщо ж виникли проблеми функціонування системи ще більш серйозні, то може знадобитися зміна ідеології розроблення системи в цілому. У цьому випадку доводиться повертатися на етапи *ідентифікації* і *концептуалізації* для переформулювання вимог до системи й основних рішень щодо її побудови й функціонування. Повторюється виконання усіх розглянутих етапів, що істотно збільшує вартість розроблення й терміни створення системи.

3.6. Предметні й проблемні області застосування інтелектуальних систем

Як було зазначено в попередніх розділах, сучасні КСШІ можуть розв'язувати *інтелектуальні задачі* в різних областях людської діяльності, серед яких можна виділити такі *проблемні області* [3.1–3.25].

1. *Розпізнавання образів* (об'єктів, процесів, явищ, ситуацій тощо) є історично першим і найрозповсюдженим типом інтелектуальних

задач (розд. 8.2). Завдання *розпізнавання* полягає в тому, щоб серед *безлічі наявних (подібних) образів* знайти (*виявити*) *шуканий образ* за деякими його *властивостями (атрибутами)*, якими він відрізняється від інших образів. Основними різновидами задач *розпізнавання образів* є *ідентифікація, класифікація і кластерний аналіз* (розд. 8.2).

Методи *розпізнавання образів* найчастіше застосовують у таких *предметних (проблемних) областях*:

- *промислове виробництво* — виявлення й класифікація *дефектів* у технічних системах і промислових виробках (розд. 9.6);
- *медицина* — розпізнавання й класифікація захворювань (розд. 9.6);
- *робототехніка* — розпізнавання образів зовнішнього світу й команд оператора, що вводить інформацію в систему голосом (розд. 11);
- *охоронні системи* — ідентифікація особистості та визначення можливості її доступу на територію, що охороняється, тощо.

2. *Керування* діяльністю фірм, організацій і підприємств, а також складними технічними комплексами, об'єктами й процесами. Серед них:

- регулювання *фінансової діяльності* підприємств і організацій (розд. 8.2);
- керування польотами *космічних кораблів* і апаратів;
- керування *корпоративними знаннями*, нагромадженими у фірмах, організаціях та на підприємствах (розд. 4.3);
- керування *повітряним рухом* і транспортними перевезеннями;
- керування *мережами* розподілу електроенергії тощо.

3. *Планування* — складання плану роботи досліджуваних об'єктів і визначення послідовності дій для досягнення поставленої мети. Об'єктами планування можуть бути:

- промислові *замовлення й інвестиції* (розд. 8.2);
- *режими роботи* організацій і підприємств;
- наукові й медичні *експериментальні дослідження* тощо.

4. *Проектування* — визначення конфігурацій фізичних об'єктів відповідно до заданих обмежень, критеріїв ефективності та підготовки відповідних специфікацій (креслень, пояснювальних записок тощо) на створення об'єктів із заданими властивостями. Ці задачі вирішуються, наприклад, при *проектуванні нових промислових об'єктів, космічних станцій, мереж енергопостачання, комп'ютерних апаратно-програмних засобів* тощо (розд. 9.8).

5. *Прогнозування* можливих наслідків розвитку подій на основі аналізу інформації про стан об'єктів, процесів, явищ у поточному і попередніх періодах часу. Сучасні КСШІ здатні:

4.1. Загальні відомості

Як було зазначено у попередніх розділах, основною властивістю інтелекту людини є здатність мозку вирішувати *інтелектуальні задачі* шляхом цілеспрямованого використання *знань*, які здобуває людина у процесі придбання досвіду й адаптації до різноманітних життєвих обставин [1.1–1.16]. За аналогією з людиною, розв'язання задач у КСШ також здійснюється на основі *комп'ютерних знань*, що відображують *закономірності й властивості* (закони, принципи, зв'язки, правила, ознаки тощо) даної *предметної (проблемної) області (ПрО)*, в якій функціонує система. Прикладами *предметних областей* можуть бути бізнес, виробництво, медицина, космос тощо (розд. 9.4). *Проблемною областю* називають *сукупність (клас) задач (проблем)*, що вирішує КСШ в цій *предметній області* (розпізнавання образів, прийняття рішень, моніторинг, проектування, прогнозування тощо) (розд. 3.6 і 9.4).

Знання в КСШ є комп'ютерною інформацією, яка описує (відображає):

- об'єкти, процеси, явища, події, дії, ситуації ПрО і зв'язку між ними;
- проблеми й задачі, які необхідно вирішувати в даній ПрО;
- *методи й засоби вирішення цих проблем і задач*;
- *умови функціонування системи, що визначають вибір методів та засобів вирішення задач*.

Способи одержання й використання *знань* у КСШ по своїй суті наближаються до людського сприйняття й подання інформації.

У межах робіт зі створення КСШ нині швидкими темпами розвивається *інженерна технологія інтелектуального опрацювання інформації*, що одержала назву «*інженерія знань*» («*Knowledge Engineering*») [2.1–2.41, 3.1–3.25, 4.1–4.31]. Основним призначенням цієї технології в області КСШ є розроблення й використання *методів і засобів одержання (добування) і подання знань* для створення *бази знань КСШ – комп'ютерної пам'яті інтелектуальної системи* (розд. 3.3, 5, 6).

Технологія *інженерії знань* передбачає виконання такої послідовності робіт зі створення *бази знань КСШ*:

- визначення видів і складу знань, необхідних для роботи КСШ в цій ПрО;

- прогнозувати *розвиток економіки*,
- аналізувати *кон'юнктуру ринку* і розробляти план капіталовкладень на перспективу (розд. 8.2);
- оцінювати *майбутній урожай*;
- прогнозувати *воєнні дії*;
- *прогнозувати погоду тощо*.

6. *Інтерпретація* – одержання погоджених і коректних висновків на підставі багатоваріантного аналізу поточної інформації та результатів спостережень. До числа практичних задач, розв'язуваних у процесі *інтерпретації*, можна віднести:

- виявлення *місць розташування та ідентифікацію* океанських судів за даними акустичних систем спостереження й результатами аерокосмічного сканування;
- виявлення *покладів корисних копалин*;
- визначення *властивостей особистості* за результатами психодіагностичного тестування тощо.

7. *Моніторинг* – здійснення неперервного контролю за станом об'єктів у реальному масштабі часу та сигналізація про вихід тих чи інших режимних параметрів об'єктів за припустимі межі. Об'єктами моніторингу можуть бути *атомні реактори, нафтохімічні заводи, електростанції, стан хворих* у період хвороби й реабілітації та ін.

8. *Діагностика* – виявлення відхилень від норми або співвіднесення досліджуваного об'єкта з відомим класом об'єктів. Таке трактування задачі дає змогу з єдиних позицій розглядати і несправності в технічних системах, і захворювання живих організмів, і різноманітні природні аномалії. До задач *діагностики* належать:

- *технічна діагностика* несправностей та відмовлень технічних систем, автотранспорту, комп'ютерних апаратно-програмних засобів тощо;
- виявлення *причин відмовлень* технічних пристроїв і устаткування;
- *медична діагностика* захворювань (розд. 9.6) тощо.

9. *Навчання* тому чи іншому *виду діяльності, предмету* або *дисципліні* з використанням *тьюторів* – інтелектуальних комп'ютерних засобів, які працюють в інтерактивному режимі зворотного зв'язку з користувачем, діагностують помилки і підказують правильні рішення (розд. 9.6). Когнітивна графіка тьюторів дає змогу проводити навчання за допомогою зорових образів і мультфільмів, у створенні яких беруть участь і тьютор, і той, хто навчається.

Слід зазначити, що більшість КСШ можуть вирішувати кілька видів інтелектуальних задач одночасно. Наприклад, *система керування* може вирішувати задачі контролю, діагностики, прогнозування й планування роботи фірми, підприємства або організації.

- вибір методів і засобів *одержання* (здобування) знань;
- вибір (за потреби — розроблення) методів і засобів *подання інформації* у базі знань;
- (здобування) знань із різних джерел інформації;
- *вербальний* (словесно-змістовний) опис моделі *ПрО природною* (людською) мовою;
- перетворення *вербального опису ПрО* у форму, необхідну для введення в *базу знань* системи;
- *формування* (структурування) *базу знань* на основі обраних *моделей* подання знань (розд. 5, 6).

4.2. Основні види знань

У різних типах КСПІ (розд. 3.2) можуть використовуватися такі *види знань* [2.1—2.41, 3.1—3.25, 4.1—4.31]:

- *позаособисті знання* містяться в публікаціях (книгах, статтях, журналах, підручниках, методиках), усних повідомленнях фахівців, електронних архівних документах, на сайтах системи *Internet* тощо;
- *особисті знання* є особистим надбанням фахівців (*експертів*) у певній *ПрО* і відображують їхній індивідуальний професійний досвід;
- *поверхневі знання* описують причинно-наслідкові відносини між окремими об'єктами, явищами, процесами й ситуаціями певної *ПрО*;
- *глибинні знання* відображають детальну структуру і глибинну природу процесів, що відбуваються у *ПрО* (наприклад, знання фізіологів і лікарів про причини та види хвороб і методи їх лікування);
- *декларативні знання* описуються мовою подання знань, наближеною до природної й зрозумілою користувачам, що не є фахівцями в цій *ПрО*;
- *процедурні знання* «розчинені» у методах, алгоритмах і програмних засобах вирішення задач і зазвичай приховані від користувача;
- *нечіткі* (*fuzzy*) знання описують якісні характеристики *ПрО* («невеликий», «високий», «прохолодний» тощо) і тому є неточними (розмитими, м'якими) знаннями, вірогідність яких можна оцінити деякою проміжною величиною між «істиною» і «неправдою» (розд. 14);
- *прецедентні знання* відображують міру схожості властивостей об'єктів, процесів і явищ даної *ПрО*;
- *стереотипні знання* відображують відомі стандартні ситуації *ПрО* з використанням так званих сценаріїв, які являють собою послідовності дій або процедур, що описують процеси і явища певної *ПрО*;
- *метазнання* (знання про знання) описують структуру бази знань і визначають порядок і правила застосування знань у сформованій (конфліктній) ситуації при розв'язанні поставленої задачі.

Усі розглянуті вище види знань можуть бути:

- *неповними*, тобто недостатніми для розв'язання поставлених задач;
- *недостовірними*, отриманими з ненадійних джерел;
- *помилковими* — у зв'язку з некомпетентністю фахівців (експертів) у певній *ПрО*.

4.3. Основні властивості знань

4.3.1. Внутрішня інтерпретація

Стандартною формою подання інформації у комп'ютерних системах є *інформаційна одиниця (ІО)*, яка являє собою *машинне слово*, що складається з визначеного числа двійкових розрядів — *бітів*. У процесі еволюції й удосконалення інтелектуальних комп'ютерних засобів відбувалося поступове ускладнення *структури ІО* за схемою: двійкове подання даних — *вектори й матриці* — *списки* — *абстрактні типи даних*.

У сучасних *базах знань* кожна *ІО* (факт або правило) має *оболонку* (*протоструктуру*), в якій міститься *унікальне ім'я* (*адреса*) *ІО*. За цією адресою система знаходить протоструктуру та відповідає на запити, в яких це ім'я згадане. За допомогою цих імен можна зв'язувати *ІО* відповідними відносинами і робити посилання на інформацію, що зберігається в інших *ІО*. В оболонці *ІО* є також спеціальні поля, в яких зберігається додаткова службова й допоміжна інформація. Крім того, в базі знань системи є спеціальні *словники*, в яких містяться всі ці атрибути *ІО*. За їх допомогою можна здійснювати оперативний пошук потрібної інформації для вирішення інтелектуальних задач.

4.3.2. Структурованість

Знання мають гнучку структуру, організовану за *принципом «матрьошки»*, яка забезпечує *рекурсивну вкладеність* одних *ІО* в інші. Кожна *ІО* може бути вміщена до складу іншої, а з кожної *ІО* можна виокремити її складові фрагменти. *Структурування знань* дає змогу швидко знайти необхідну інформацію у базі знань і тим самим забезпечує ефективну роботу системи.

4.3.3. Зв'язковість

У базі знань КСПІ є можливість установаження *зв'язків і відносин* між *ІО* типу «*частина — ціле*», «*рід — вид*», «*елемент — клас*» тощо. Ці зв'язки та відносини можуть мати *декларативний* або *процедурний*

характер. Наприклад, дві або декілька ІО можуть бути пов'язані відносинами «одночасно», «причина — наслідок» тощо. Такі відносини характеризують *декларативні* знання. Якщо між двома ІО встановлено відношення «аргумент — функція», то воно характеризує *процедурне* знання, пов'язане з обчисленням визначених функцій. Між ІО можуть встановлюватися й інші зв'язки, що визначають, наприклад, порядок вибору ІО чи вказують на те, що дві ІО несумісні одна з одною.

4.3.4. Семантична метрика

На множині ІО задаються відносини, які характеризують *ситуаційну близькість (релевантність)* ІО, тобто силу *асоціативного зв'язку* між ІО. Відносини релевантності при роботі з ІО дають змогу знаходити знання, близькі до вже знайдених знань. На сьогодні розрізняють:

- *відносини структуризації*, за допомогою яких задається структурна ієрархія ІО;
- *функціональні відносини*, які несуть процедурну інформацію, що дає змогу знаходити (обчислювати) одні ІО через інші;
- *каузальні відносини*, що задають причинно-наслідкові зв'язки між ІО.

Указані особливості знань дають змогу створити загальну *семантичну модель* подання знань. Вона являє собою *ієрархічну мережу*, у вершинах якої знаходяться ІО, позначені індивідуальними іменами. Дуги семантичної мережі відповідають різним зв'язкам між ІО (розд. 6.4).

4.3.5. Активність

Знання людини є *активними*, оскільки сам факт знання з огляду на його взаємодію з іншими знаннями, що зберігаються в пам'яті, породжують *цілі й підцілі* дій людини. Те, що людина пам'ятає й знає, безпосередньо впливає на те, *що* вона думає і *як* робить.

Знання людини пов'язані між собою позитивними й негативними відносинами. Залежно від того, які структури утворюються із цих зв'язків, виникає *дисонанс* визначеної сили, тобто у свідомості людини з'являється занепокоєння, імпульс до якихось вчинків, прагнення щось змінити, щоб досягти внутрішнього комфорту — *консонансу*.

З моменту появи комп'ютера і поділу використовуваних у ньому ІО на *дані й команди* створилася ситуація, за якої дані *пасивні*, а команди *активні*. Усі процеси, що відбуваються в комп'ютері, ініціювалися командами, а дані використовувалися цими командами лише в разі потреби. В інтелектуальній же системі, як і в людини, актуалізація тих чи інших дій може ініціюватися *динамікою зміни*

стану знань чи появою нових знань, які можуть стати джерелом активності системи. Нова інформація діє на базу знань як *хвиля збурення*, що показує системі, які ще процедури необхідно виконати, щоб вирішити поставлену задачу.

4.4. Методи придбання (добування) знань

4.4.1. Основні положення

Придбанням знань називають процес *одержання знань* із різних інформаційних джерел і перетворення їх у форму, необхідну для подання у *базі знань* КСШІ [4.1—4.31]. Основним *джерелом інформації*, що використовується під час створення інтелектуальних систем, є *особистий* професійний досвід *експертів*, що працюють у даній Про (розд. 3.4). Іншими, не менш важливими джерелами інформації можуть бути інформаційні сховища *позаособових знань* (див. вище), нагромаджених людством, що містяться, насамперед, у публікаціях та електронних документах системи *Internet*.

Головну роль у *процесі придбання знань* виконує *аналітик* (*інженер по знаннях*) (розд. 3.4), який повинен:

- опанувати *позаособові знання* про дану Про;
- освоїти теоретичні основи спілкування і придбати навички інтерв'юєра;
- визначити склад експертів та методи їх тестування;
- вибрати методи добування знань;
- підготувати ситуації спілкування з експертами тощо.

Аналітик повинен ретельно підготуватися до процесу добування знань. Він повинен бути здоровим, спокійним, упевненим, вселяти довіру, бути щирим, дружелюбним, веселим, підтримувати інтерес до спілкування.

Аналітику необхідно глибоко познайомитися зі спеціальною літературою щодо Про, щоб не ставити *безглузвих питань*. Він повинен уміти слухати та налаштовуватися на роль *учня*, а не *екзаменатора*, розбиратися в моделях *когнітивної психології* (яка досліджує манери поведінки, стилі наукового мислення тощо) і способах подання знань, щоб зі знань експерта вирізнити їх чіткі структури.

Процес взаємодії *аналітика з експертом* містить кілька етапів.

1. *Підготовчий етап*. Для успіху спілкування обидва учасники повинні ретельно підготуватися до діалогу чи гри. Бажано, щоб експерт був не тільки компетентним фахівцем, а й зацікавленим (морально або матеріально) у досягненні кінцевої мети — побудові

КСШ. Він повинен бути доброзичливим до аналітика і вміти пояснювати свої знання. Найкращим є випадок, коли експерт має досвід викладацької роботи.

2. *Встановлення «загального коду».* Для організації лінгвістичної взаємодії учасники повинні визначити головні поняття досліджуваної ПрО, виробити словникову основу й рівень деталізації знань, взаємозв'язку між знаннями тощо.

3. *Гносеологічний етап* є визначальним у взаємодії аналітика й експерта. На цьому етапі відбувається з'ясування закономірностей, властивих ПрО, умов вірогідності й істинності знань, їхнє структурування тощо. У процесі аналізу діалогу або гри *вербалізуються* (виражаються у словесно-змістовній формі) і формалізуються знання експерта. У цьому разі в нього самого часто з'являються нові знання.

У процесі добування знань спочатку одержують від експерта *поверхневі* знання, поступово переходячи до *глибинних* структур і більш абстрактних знань. Сукупність отриманих знань повинна мати:

- властивість *унікальності*;
- відсутність *надмірності*;
- *повноту*, тобто досить повний опис ПрО;
- *вірогідність*, тобто адекватне відображення дійсності;
- *несуперечність*, тобто відсутність конфліктних тверджень тощо.

4.4.2. Текстологічні методи добування знань

Як було зазначено, побудова моделі ПрО і бази знань КСШ починається з вивчення *позаособистісних* знань, що добуваються аналітиком із різних джерел інформації за допомогою так званих *текстологічних методів* добування. Для розуміння й виділення змісту тексту аналітик здійснює *декомпозицію* (розчленування) тексту на сукупність фрагментів (*компонентів*), основними з яких є *спостереження, поняття, суб'єктивні погляди, запозичення, загальні місця*. Аналіз цих компонентів і становить сутність *процесу розуміння* тексту, основними моментами якого є:

- висування *попередньої гіпотези* про зміст тексту;
- визначення *ключових значень* (понять) *спеціальної* (професійної) термінології;
- формування *загальної гіпотези* про зміст (знання) тексту;
- уточнення значень *ключових термінів* і *понять* відповідно до загальної гіпотези;
- виявлення *семантичної структури* тексту шляхом установлення зв'язків між ключовими термінами і поняттями;
- формування й уточнення *основної гіпотези* про зміст *знань* у тексті.

Алгоритм добування знань із тексту містить у собі:

- складання базового списку літератури для ознайомлення із цією ПрО;
- вибір тексту для добування знань і перше (швидке) знайомство з текстом;
- консультація з фахівцями чи залучення довідкової літератури для оволодіння *спеціальною* (професійною) термінологією;
- формування гіпотези про макроструктуру тексту;
- визначення (виписування) ключових слів і понять (компресія тексту);
- розроблення макроструктури зв'язків між ключовими поняттями (наприклад, у вигляді графа або реферату «стислого» тексту);
- формування *поля знань* на основі макроструктури тексту.

4.4.3. Пасивні комутативні методи

Спостереження

У процесі *спостереження* аналітик знаходиться поруч із експертом під час його професійної діяльності чи її *імітації*. Під час підготовки до сеансу добування знань аналітик пояснює експерту мету спостережень і просить його максимально коментувати свої дії. Характерною рисою методу є невторчання аналітика в роботу експерта. Під час сеансу спостереження аналітик записує (з використанням, наприклад, відеопаратури) усі дії експерта, його висловлювання й пояснення.

Спочатку аналітик спостерігає за *реальним* процесом роботи експерта, щоб глибше зрозуміти ПрО і виявити її особливості. Потім здійснюється *імітація* діяльності експерта, коли процес його роботи запускається спеціально для аналітика. Імітація проводиться також у випадках, коли спостереження за реальним процесом з певних причин неможливі. Наприклад, професійна стика лікаря-психіатра може не допускати присутності сторонніх осіб під час прийому хворих.

Сеанси спостережень потребують від аналітика серйозного попереднього знайомства з ПрО, оволодіння технікою стенографії, хронометражу та інших навичок спілкування. Протоколи спостережень розшифровуються аналітиком у процесі домашньої роботи, а потім обговорюються з експертом.

Метод *спостережень* застосовується, зазвичай, у сукупності з іншими методами, розглянутими нижче. При використанні кожного із цих методів складаються *протоколи* сеансів добування знань, які аналізуються потім під час побудови моделі ПрО (розд. 5).

Протоколювання «думок уголос»

Цей метод відрізняється від спостережень тим, що експерту пропонується не просто коментувати свої дії та рішення, а й пояснити, як вони були отримані. Передача знань виконується в процесі досить тривалих співбесід між аналітиком і експертом, який повинен чітко сформулювати наявний у нього досвід.

Експерт повинен продемонструвати весь ланцюжок своїх міркувань, що протоколює аналітик (за допомогою відповідної апаратури), як так званий *вербальний звіт*. Водночас експерт може блиснути своєю ерудицією і продемонструвати глибину своїх знань. Тому від аналітика вимагаються ті самі уміння, що й у методі спостережень.

Розшифрування отриманих протоколів аналітик здійснює самостійно з корекцією на наступних сеансах добування знань.

4.4.4. Активні комунікативні методи

Анкетування

До числа *активних комунікативних* методів добування знань належить *анкетування*, коли аналітик заздалегідь складає опитувальник чи *анкету*, розмножує їх і використовує для опитування експертів. Аналітик уголос ставить запитання експерту і сам заповнює анкету. Анкету може заповнювати й експерт після попереднього інструктування аналітиком.

Існують такі основні універсальні рекомендації зі складання анкет:

- анкета не повинна бути монотонною й одноманітною;
- анкета повинна бути пристосована до мови експертів;
- запитання в анкеті повинні йти в логічній послідовності;
- мова анкети повинна бути ясною, зрозумілою та ввічливою.

Інтерв'ю

Одним із найрозповсюдженіших *активних комунікативних* методів добування знань є *інтерв'ю*, під час якого аналітик ставить експерту серію *заздалегідь підготовлених запитань*. На відміну від анкетування, інтерв'ю дає змогу аналітику опускати окремі запитання (залежно від ситуації), ставити нові, змінювати темп і урізноманітнювати характер спілкування з експертом.

Під час інтерв'ю, зазвичай, застосовуються такі основні *типи запитань*, які заздалегідь готує аналітик:

- *відкрите запитання* визначає тему чи предмет, надаючи експерту повну свободу за формою і змістом відповіді;

- *закрите запитання* пропонується разом із набором можливих відповідей, із яких експерт вибирає найбільш придатну (на його думку) відповідь;

- *пряме запитання* безпосередньо вказує на предмет чи тему, що цікавить аналітика;

- *непряме запитання* стосується визначеного предмета чи теми;

- *особисте запитання* стосується особистого індивідуального досвіду експерта;

- *безособове запитання* спрямоване на виявлення найбільш відомих і загальноприйнятих закономірностей Про;

- *зондувальне запитання* спрямовує відповідь експерта в потрібний бік;

- *контрольне запитання* використовується для перевіряння вірогідності та об'єктивності повідомлень експерта;

- *нейтральне запитання* підкреслює неупереджене ставлення аналітика до обговорюваної теми;

- *навідне запитання* пропонує експерту взяти до уваги позицію аналітика.

Поряд із указаними типами запитань існують:

- *вербальні (усні) запитання*, що виникають у процесі інтерв'ю,
- *контактні запитання*, які налагоджують взаєморозуміння між аналітиком і експертом;

- *буферні запитання*, що розмежовують окремі теми інтерв'ю;

- *провокуючі (несподівані) запитання*, які ставлять, щоб отримати спонтанні, непередбачені відповіді.

Указані типи запитань складаються з урахуванням таких вимог, як *зрозумілість, лаконічність, термінологічна чіткість, логічна послідовність, немонотонність, етика, ввічливість*, які істотно впливають на якість інтерв'ю. Слід зазначити, що кілька безглузвих запитань можуть цілком розчарувати експерта і відбити в нього бажання подальшого співробітництва.

Діалог

Процес добування знань може відбуватися в режимі *вільного діалогу* аналітика з експертом, у якому відсутні запитальник і твердий регламентований план інтерв'ю, а діалог ведеться у формі *бесіди*. Проте зовні вільна і легка форма діалогу потребує досить високої професійної та психологічної підготовки аналітика, що допомагає йому стати драматургом чи сценаристом майбутніх діалогових сеансів.

Діалог припускає:

- унікальність кожного із партнерів та їхню принципову рівність;
- розбіжності та оригінальність точок зору партнерів;

- очікування відповіді партнера та очікування своєї;
- взаємну доповненість позицій учасників діалогу.

У вільному діалозі важливо вибрати правильний темп і ритм бесіди. Діалог повинен відбуватися без великих пауз, оскільки експерт може відриватися для поточних справ, але і без гонки, інакше партнери швидко стомляться від напруженої бесіди. Правильне чередування темпів, напруги й розрядки в діалозі суттєво впливає на результат добування знань.

4.4.5. Групові комутативні методи

«Круглий стіл»

Метод «*круглого столу*» передбачає участь *кількох рівноправних експертів* в обговоренні зазначеної проблеми ПрО. При цьому аналітик має провести додаткову організаційну (місце, час, обстановка тощо) і психологічну підготовку «*круглого столу*». Він повинен уміти підтримувати дискусію, гасити конфлікти, мати почуття гумору тощо.

Більшість рекомендацій із використання попередніх методів добування знань (інтерв'ю, діалог тощо) може бути застосована й у методі «*круглого столу*». Однак у ньому є і відмінні особливості, пов'язані зі специфікою поведінки людини у групі.

Спочатку учасники висловлюються у певному порядку, а потім переходять до вільної дискусії, метою якої є обговорення й дослідження спірних гіпотез ПрО з різних точок зору. Більшість експертів, зазвичай, намагаються продемонструвати власну компетентність, щоб справити враження на своїх колег. Тому на «*круглий стіл*» запрошують представників різних наукових напрямів і різних поколінь, щоб уникнути одержання однобічних суб'єктивних рішень. Під час дискусії аналітик повинен простежити, щоб занадто емоційні й говірки експерти не підмінили тему і щоб критика позицій один одного була обґрунтованою. Для цього він повинен бути достатньо суродованим в обговорюваних проблемах ПрО.

«Мозковий штурм»

Одним із найбільш розповсюджених групових методів добування знань є «*мозковий штурм*» («*мозговая атака*»), в якому беруть участь кілька експертів. Зміст штурму полягає в розкріпаченні й активізації творчого мислення учасників у процесі дискусії. Відомо, що побоювання критики заважає творчому мисленню, тому основною ідеєю *штурму* є відокремлення процесу *генерування* ідей від процесу їх *аналізу* й *оцінки*.

Учасникам штурму пропонується висловлювати будь-які ідеї щодо вирішення обговорюваної проблеми. Критика заборонена. Результати висловлень аналізує й оцінює незалежна група експертів, які не брали участі у штурмі. Зазвичай лише 10–15% висловлених ідей виявляються прийнятними, але серед них трапляються й дуже оригінальні рішення.

Аналітик повинен підібрати активну групу експертів («генераторів» ідей), вільно володіти аудиторією, не затискати «поганих» ідей (вони можуть стати каталізатором правильних ідей), вчасно ставити доречні запитання, «підігриваючи» аудиторію і зупиняючи надмірно багатослівних експертів.

Експертні ігри

Грою називають експеримент, у якому експертам для обговорення пропонується *ситуація*, що склалася в ПрО. Скориставшись своїм життєвим досвідом, загальними та професійними знаннями, експерти повинні виробити розв'язання задач, пов'язаних із наявною ситуацією. Потім ці рішення аналізуються незалежною комісією, що розкриває закономірності мислення учасників експерименту. Гра розвиває логічне мислення експертів і здатність швидко приймати рішення в надзвичайних ситуаціях.

При *індивідуальній грі* з експертом грає аналітик, який бере на себе роль *учня* й у присутності експерта виконує його роботу. Експерт у цьому разі виступає в ролі *вчителя* і виправляє помилки аналітика.

Групові рольові ігри передбачають участь у грі кількох експертів. До такої гри заздалегідь складають сценарій, розподіляють ролі, до кожної з яких готують *портрет-опис* і розробляють метод оцінювання рішень гравців. Аналітику надається цілковита свобода у виборі форм проведення гри.

Експерти можуть конкурувати невеликими групами. Елемент змагальності між ними поживляє гру. Кожна група намагається швидше знайти оптимальне рішення обговорюваної проблеми, яке б вимагало мінімальних ресурсів і витрат.

4.4.6. Методи добування латентних (глибинних) знань

Більшість розглянутих вище методів добування знань виявляють лише *поверхневі* знання експерта, не торкаючись їхньої *глибинної* структури. Однак таких знань може виявитися недостатньо для функціонування сучасних КСШ. Добування *глибинних* знань усе

частіше здійснюється з використанням методів *психосемантики*, яка виникла на перетині кількох родинних наукових напрямів, основними з яких є:

- *когнітивна психологія*, яка моделює механізми пізнання людиною зовнішнього світу;
- *психолінгвістика*, що займається структурним аналізом процесів мислення й мови людини;
- *психологія сприйняття* людиною навколишнього світу;
- дослідження *індивідуальної свідомості* людини тощо.

Психосемантика орієнтована на виявлення «прихованих» знань людини, які експерт може не усвідомлювати, оскільки ці знання містяться у *глибинних (латентних)* структурах людської свідомості.

Метод реконструкції суб'єктивного семантичного простору

Суть методу полягає у графічному поданні *структури латентних знань* експерта у багатомірному *семантичному просторі* і подальшому його дослідженні. *Точками* (векторами) цього простору є окремі *поняття (ознаки)* ПрО, а координатами — так звані *фактори* (концепти), що визначаються у процесі попереднього *факторного* або *кластерного* аналізу окремих груп понять. Відстань між точками відображує *семантичну близькість* відповідних понять. Розмірність простору й число досліджуваних понять можуть варіюватися залежно від досвіду та професійної підготовки експертів. Використання *моделі семантичного простору* дає змогу перейти від мови *описів* безлічі *понять* ПрО до більш місткої *метамови концептуалізації* з високим рівнем абстракції.

Кожний із експертів, що бере участь у розробленні бази знань КСШ, будує свою *суб'єктивну модель семантичного простору*, яка відображує досвід і характер діяльності людини. Потім ці моделі аналізуються (зіставляються) незалежними фахівцями, які визначають *ступінь погодженості семантичного простору* і виявляють нові *латентні знання* експертів.

Метод багатомірного шкалювання

Для добування *латентних знань* експерта нині все частіше використовується один із методів математичної статистики, *багатомірне шкалювання*. Цей метод призначений для виявлення *парної подібності понять* у латентних знаннях експерта і встановлення *зв'язків* між ними. У цьому разі поняття подаються у вигляді крапок *координатного простору латентних перемінних*.

В основу цього методу покладена інтерактивна процедура *суб'єктивного шкалювання*, за якою експерту пропонується оцінити *подібність* різних елементів його знань за допомогою деякої *граду-*

йованої шкали (наприклад, від 0 до 9 чи від -1 до $+1$). *Подібність між поняттями* (концептами) визначається відповідно до заздалегідь установленної *міри близькості* або *міри відмінності*.

Після зазначеної процедури отримана інформація опрацьовується за допомогою методів математичної статистики і здійснюється геометрична інтерпретація отриманих результатів шляхом подання *дистанційних параметрів подібності* латентних знань експерта в семантичному багатомірному (евклідовому) просторі.

Метод репертуарних решіток

Репертуарною решіткою називають *матрицю*, яка заповнюється аналітиком і експертом під час їх співбесіди або спільного вивчення ПрО. *Стовпці* матриці вміщують виділені групи *понять (елементів)* ПрО. *Рядки* матриці містять так звані *конструкти*, що являють собою *альтернативні протилежні (біполярні) ознаки, властивості, відносини, параметри* й інші характеристики *елементів (понять)*.

Конструкт відображує одну із властивостей мислення людини, коли вона щось стверджує, то водночас і щось заперечує. Наприклад, *елемент «сторони світу» репертуарної решітки* можна описати двома біполярними конструктами: «північ-південь» та «схід-захід». Інакше кажучи, північний напрямок протилежний південному тощо. *Конструкти* заздалегідь встановлюються аналітиком або виявляються разом із експертом за допомогою спеціальних процедур їх одержання.

Репертуарна решітка не завжди є матрицею у традиційному уявленні, оскільки її рядки (конструкти) можуть мати різну довжину, а на перетинанні рядків і стовпців не обов'язково стоять числа. Визначення «*репертуарна*» означає, що елементи являють собою узагальнені *інструкції (репертуари)*, які відображують визначену область знань і семантично пов'язані між собою аналогічно репертуару *ролей* у п'єсі чи спектаклі. Змінюючи репертуар елементів, можна «настроювати» репертуарну решітку на виявлення конструктивних різних рівнів спільності.

Конструкти мають так званий *діапазон придатності*, обмежений областю уявлень експерта про елементи даної ПрО. Осмислений результат процедури добування знань може бути отриманий лише в тому випадку, якщо елементи репертуарної решітки будуть потрапляти в діапазон придатності конструктивних експерта.

Одним із найбільш розповсюджених способів виявлення конструктивних у процесі добування знань є *метод мінімального контексту (метод триад)*, за якого всі елементи розбиваються на *групи* (зазвичай, на *три*). Експерту пропонується назвати яку-небудь ознаку

(якість), за якою одна частина елементів групи (зазвичай — *два елементи*) подібні між собою і, отже, відрізняються від іншої частини елементів цієї групи (тобто від *третього* елемента). Відповідь на це запитання і являє собою протилежний полюс конструкта. Наприклад, аналізуючи три поняття (*тріаду*) «диван, крісло, стілець», можна виявити *біполярний конструкт* «м'які меблі (диван і крісло) — тверді меблі (стілець)».

Аналіз репертуарних решіток дає змогу побудувати цілісну структуру конструктів експерта, визначити спрямованість і величину зв'язків між конструктами і в остаточному підсумку — виявити найважливіші характеристики й параметри латентних знань експерта. Аналіз серії репертуарних решіток, заповнюваних одним і тим самим експертом у різні періоди часу, дає змогу простежити динаміку зміни його знань у системі власних суб'єктивних оцінок. Поряд із цим нині розроблені методи порівняння кількох (найчастіше — двох) репертуарних решіток, які складають різні експерти. Порівняльний аналіз таких решіток допомагає досягти угоди й розуміння між експертами, що мають різні точки зору.

4.4.7. Особливості процесу добування знань

Багато дослідників і фахівців розглядають *процес добування знань* як одне з головних «вузьких місць» технології створення інтелектуальних систем. Найскладнішими для будь-якої людини є намагання пояснити свій розумовий процес, і, зазвичай, вона неспроможна це зробити. Крім того, частина знань людини зберігається в її пам'яті в *невербальній* формі (наприклад, процес «як зав'язувати шнурки») і практично не піддається словесному описанню.

Є низка об'єктивних обставин, які утруднюють процес добування знань, зумовлюючи тим самим його досить низьку продуктивність. До числа таких труднощів належать:

1. Аналітик зазвичай не є професіоналом у даній Про, тому він і запрошує експертів. Але перш ніж приступити до їхнього опитування, аналітик повинен витратити досить багато часу на ознайомлення зі специфікою й термінологією Про, інакше процес опитування буде непродуктивним.

2. Для подання специфічних знань про Про аналітик повинен підібрати придатну методiku визначення понять і структурування знань, що саме по собі є непростим завданням.

3. Більшість експертів, успішно використовуючи у повсякденній діяльності свої знання, зазнають великих труднощів при спробі їх сформулювати й подати у системному структурованому вигляді.

4. Знання експерта не завжди можуть бути описані вербально (словесно) і можуть мати умоглядний чи наочний характер.

5. Знання експерта найчастіше не можуть бути сформульовані в термінах моделі Про. Так, експерту у фінансовій області може бути відомо, що визначені події можуть стати причиною росту чи зниження котирувань на фондовій біржі, але він нічого не зможе сказати з певністю про механізми, що приводять до такого ефекту, або про кількісну оцінку впливу цих чинників.

6. Знаючи, наприклад, що з *A* випливає *B*, експерт часто не підозрює, що це правило містить у собі цілу низку проміжних логічних міркувань.

7. Фахівці у вузькій області, як правило, користуються власним жаргоном, який важко перевести на звичайну «людську» мову. Тому потрібно досить багато часу для взаємного уточнення використаних понять.

8. Багато експертів дуже ревно ставляться до свого унікального досвіду, не схильні його розголошувати і можуть перешкоджати передачі своїх знань іншим.

9. Експерт може бути не схильним до спілкування або надто зайнятим, щоб гаяти час на тривалі співбесіди з аналітиком. Експерт високого класу, як правило, не має проблем із пропозиціями роботи в тій області, з якою він добре знайомий. Тому він схильний працювати, а не розмовляти про методи своєї роботи.

4.5. Комп'ютерне добування знань

Незадовільні результати співбесід із експертами спонукали дослідників КСШ *автоматизувати* процес добування знань через організацію взаємодії експерта з *комп'ютером*. На сьогодні все ширше застосовується *автоматизоване (комп'ютерне) добування знань* з використанням *засобів комп'ютерного навчання*, яке стає розвинутою і перспективною областю інженерії знань [4.5, 4.8, 4.13–4.15].

До складу *підсистеми комп'ютерного навчання* входять програмні засоби реалізації *метазнань*, які дають змогу здійснювати:

- *інтерактивне* добування знань експерта й автоматичне перетворення їх у форму, необхідну для введення у базу знань;
- групову роботу кількох експертів;
- конструювання різних ієрархічних структур знань;
- тестування, поповнення та корекцію бази знань тощо.

Підсистема комп'ютерного навчання може функціонувати в таких *режимах*:

- в *автоматичному* режимі підсистема цілком бере на себе керування процесом добування знань;
- у режимі *асистента* підсистема дає відповідні рекомендації експерту з приводу його можливих дій, але він може на них не зважати;
- у режимі *спостереження* підсистема лише стежить за діями експерта й автоматично зберігає необхідну інформацію.

У кожному із цих режимів експерт вибирає ступінь детальності підказок і пояснень, які дає підсистема («цілком», «на середньому рівні», «коротко»).

Інтерактивні програмні засоби комп'ютерного навчання дають змогу добувати знання у експерта в процесі його діалогу з комп'ютером за допомогою сучасних комп'ютерних засобів *когнітивної графіки*. У процесі комп'ютерного навчання здійснюється *формування знань* (породження гіпотез) на підставі навчальних вибірок, навчання за аналогією та інших інтелектуальних засобів придбання знань, які дають змогу автоматично встановлювати причинно-наслідкові емпіричні зв'язки між евристиками бази знань.

Програма комп'ютерного навчання використовує метод *структурованого інтерв'ю* для формування нових понять і правил. У цьому разі до створюваної бази знань КСШ, зазвичай заздалегідь, заносяться *метазнання* про те, які типи і види знань необхідні для розв'язання задач (тобто для висновку на знаннях) і як розпізнати їх в інформації, яку повідомляє експерт.

Комп'ютер «розмовляє» з експертом як аналітик, що прагне зрозуміти, як концептуально мають бути організовані знання експерта, необхідні для функціонування КСШ. Програма намагається одержати інформацію про типи концептуальних знань (понять), про природу, напрям і кількісну оцінку величини асоціативного зв'язку між поняттями. Програма аналізує поточний стан бази знань і пропонує експерту переглянути й відредагувати окремі її фрагменти, що стосуються питання, поставленого програмою. Діалог із експертом ведеться або з використанням питань-підказок, або за допомогою віконних меню.

У разі невдалої спроби (в режимі консультації або тестування) програма пропонує експерту вказати й можливі причини. Програма переходить у режим очікування і готується до сприйняття нової інформації, яку буде вводити експерт для усунення невдачі. Крім того, програма самостійно намагається одержати необхідну інформацію, використовуючи свої попередні запитання й відповіді експерта.

Слід зазначити, що засоби комп'ютерного добування знань не вирішують повною мірою таких проблем, як неповнота, неточність і помилковість знань, одержання глибинних і невербальних знань,

суб'єктивність досвіду і знань експерта. Ці проблеми можуть бути вирішені лише за комплексного використання відповідних методів добування знань, які було розглянуто в попередніх розділах.

4.6. Технологія добування знань *Data Mining*

Як було зазначено у *Вступі*, обсяги *електронної* інформації, використовуваної в різних сферах життя сучасного суспільства, стрімко зростають. Інформаційні ресурси *Internet* складають на сьогодні вже *екзабайти «сирих»* неструктурованих даних, які містять *глибинний (прихований) шар знань*. Відшукати ці знання за допомогою розглянутих вище методів добування найчастіше не вдається, і тому дуже цінна для користувачів КСШ інформація залишається незатребованою.

З огляду на це, останнім часом значна увага приділяється розробленню *нових методів пошуку й опрацювання знань* у великих інформаційних сховищах, насамперед — у середовищі *Internet*. Виник новий напрям в опрацюванні текстової інформації — «*глибинний аналіз текстів*» *Text Mining*. Підґрунтям цього напрямку є спеціальна технологія добування знань *Data Mining*, яка призначена для виявлення *глибинних (прихованих) знань у базах «сирих» даних* і трактується фахівцями КСШ як «*розкопка знань*», «*інтелектуальний аналіз даних*» тощо [4.18—4.26].



Рис. 4.1 — Мультидисциплінарні області технології *Data Mining*

Data Mining є *мультидисциплінарною технологією*, яка виникла і розвивається на базі методів КСШ, прикладної статистики, теорії баз даних та інших галузей інформатики (рис. 4.1). Нині є багато версій *Data Mining*, які інтегрують у собі інтелектуальні засоби кількох дисциплін, одна з яких є основною (*ключовою*).

Технологія *Data Mining* базується на використанні *шаблонів* (так званих *паттернів*), які відображують *закономірності* багатоаспектних

взаємин між фрагментами баз даних. Ці шаблони відзначають *неочевидні (несподівані) регулярності* в даних, що містять *приховані (глибинні) знання*. Шаблонами можуть бути, наприклад, такі формулювання задач: «Які чинники визначають можливість нещасних випадків?», «Які схеми покупок характерні для шахрайства із кредитними картками?» тощо.

Є такі стандартні *типи закономірностей (шаблонів)*, які технологія *Data Mining* дозволяє виявити в текстовій інформації:

- *асоціація* взаємозалежних подій;
- *послідовність* низки пов'язаних у часі подій;
- *класифікація* ознак, що характеризують групу, до якої належить об'єкт;
- *кластеризація*, за якої групи ознак заздалегідь не задані;
- *прогнозування* знань за допомогою шаблонів (наприклад, у вигляді тимчасових рядів), що адекватно відбивають динаміку зміни цільових показників текстової інформації.

Однією з основних сфер застосування технології *Data Mining* є предметно-орієнтовані *системи технічного аналізу*, що набули найбільшого поширення в області *бізнес-застосувань*. Програмне забезпечення таких систем містить у собі комплекс методів *прогнозу динаміки цін* і вибору оптимальної структури інвестиційного *портфеля*, заснованих на *емпіричних моделях динаміки фінансового ринку*.

На сьогодні системи *Data Mining* часто використовують також як інструментальні засоби для проведення унікальних наукових досліджень у генетиці, медицині, хімії та інших Про. За їх допомогою вирішують задачі прогнозування, класифікації, розпізнавання образів, сегментації баз даних, добування із баз даних прихованих знань, інтерпретації даних, установлення асоціацій у базах даних та інші інтелектуальні задачі.

4.7. Системи керування знаннями

Проблема *керування знаннями (Knowledge Management)* виникла у великих корпораціях (компаніях, фірмах, організаціях, підприємствах) (далі для стислості — *корпораціях*), де проблеми опрацювання інформації набули особливої гостроти і стали критичними, оскільки тут буває нагромаджено стільки інформації, що співробітники корпорації не в змозі її опрацювати. Найчастіше одна частина корпорації може повторювати роботу іншої її частини, оскільки буває неможливо знайти і використати знання, що знаходяться в різних місцях. Ресурси знань, зазвичай, розосереджені по всій корпорації

і можуть знаходитися у базах знань, базах даних, картотеках, у окремих фахівців тощо.

Для вирішення зазначеної проблеми у великих корпораціях нині створюються *інтелектуальні системи керування знаннями (ІСКЗ)*, призначені для пошуку, аналізу, опрацювання, використання і поширення *знань* [4.27—4.30]. За аналогією з людським мисленням, в основу функціонування ІСКЗ покладено *механізм спрощення* пошуку необхідних знань. Його дію можна порівняти з дією керівника установи, що розподіляє роботу відповідно до її важливості. Секретар, друкарка, клерк та інші співробітники зайняті своїми справами. Якщо ж якусь роботу потрібно закінчити терміново, керівник повинен вирішити, хто і коли буде нею займатися. Якщо не зробити цього, то одні співробітники будуть сидіти без діла, очікуючи, поки інші завершать свою частину роботи, тоді як хтось буде витрачати дорогоцінний час, виконуючи нетермінову роботу. Можна виконати значний обсяг роботи, але вона виявиться непотрібною, якщо не буде досягнуто кінцевої мети.

Без *механізму спрощення* людський мозок буде паралізовано так само, як установу без керівника. Людині при розв'язанні кожної задачі потрібно буде перевіряти усі відомі їй правила. Людський мозок, подібно установі з повним штатом співробітників різних професій, зберігає всі знання, необхідні для ухвалення рішення, але якщо необхідну інформацію не вибрати вчасно, то всі знання будуть марні. Механізм спрощення спрямовує думку людини в потрібне русло і дає змогу швидко знаходити знання, необхідні для досягнення поставленої мети.

Знання, що не використовуються і не поновлюються, згодом стають застарілими й непотрібними. Знання подібні до грошей, які повинні бути оборотним капіталом, інакше вони знецінюються. Використовувані знання (гроші), що поширюються, здобуваються й обмінюються, генерують нові знання (гроші).

Необхідність створення ІСКЗ у тій чи іншій корпорації диктується такими обставинами:

- працівники корпорації витрачають занадто багато часу на пошук необхідної інформації;
- досвід провідних і найкваліфікованіших співробітників використовують тільки вони самі;
- цінна інформація схована у великій кількості документів, доступ до яких утруднений чи неможливий;
- «дорогі» помилки в роботі корпорації повторюються через недостатню інформованість співробітників та ігнорування ними минулого (позитивного й негативного) досвіду їхньої роботи тощо.

Підґрунтям ІСКЗ є так звана *корпоративна пам'ять*, яка являє собою базу знань *корпорації* (розд. 5.1) і, за аналогією з людською пам'яттю, дає змогу використовувати попередній досвід роботи й уникати повторення помилок.

Основними інформаційними *джерелами (ресурсами)*, що використовуються під час створення та функціонування *корпоративної пам'яті*, є:

- наявні інформаційні засоби самої корпорації (бази знань, бази даних, програмне забезпечення, картотеки, довідники, патенти, публікації, документи, інструкції, посібники, відео- і аудіозаписи, креслення, зведення про замовників і конкурентів, поточні дані про виконувані розроблення тощо);
- особисті знання фахівців;
- інформаційні можливості системи *Internet*;
- корпоративні мережі та системи групової підтримки інших корпорацій тощо.

Корпоративна пам'ять нагромаджує інформацію, що надходить із різних джерел, робить її доступною для всіх співробітників корпорації і забезпечує збереження знань фахівців, що вибувають, звільняються, йдуть на пенсію та ін.

Основними функціями ІСКЗ є:

- збір і систематичне відновлення одержуваної інформації в корпоративній базі знань;
- організація взаємодії корпоративної пам'яті з програмно-апаратними засобами системи в процесі виконання запитів користувачів;
- забезпечення співробітників корпорації інформацією із запиту (пасивна форма) або в режимі реального часу (активна форма).

Процес опрацювання інформації в ІСКЗ містить такі етапи:

- *добування* інформації з різних джерел (розд. 4.2);
- *структурування* інформації, у процесі якої повинні бути визначені основні поняття, вироблена структура подання інформації з максимальною наочністю, простотою зміни й доповнення;
- *формалізацію* інформації, тобто подання структурованої інформації на мовах програмування і комп'ютерного опрацювання;
- *корегування* інформації за допомогою відновлення або видалення («чищення») застарілої інформації.

ІСКЗ є інтелектуальним помічником користувача. Вона реалізує *активний підхід* до вирішення проблеми поширення й опрацювання знань, за цього підходу поряд із відповідями на запити користувача ІСКЗ автоматично надає йому додаткову інформацію, необхідну для розв'язання конкретної задачі.

У складі ІСКЗ є *підсистема пояснень*, яка дає змогу безпосередньо відповідати на запитання типу «*Чому?*» або «*Чому ні?*». У традиційній базі даних користувач повинен самотійно шукати потрібну інформацію для відповіді на такі запитання. Водночас користувач ігнорує велику частину інформації, потенційно необхідну йому для відповіді на наступні запитання.

Нині і ІСКЗ викликають значний інтерес у промислових корпораціях, які усвідомлюють високий прикладний потенціал системи і її корпоративної пам'яті. Створення ІСКЗ вирішує проблему оперативного опрацювання корпоративної інформації та одержання нових знань, що, в остаточному підсумку, забезпечує перевагу перед конкурентами. Однак більшість наявних проєктів обмежуються на сьогодні лише стадією *прототипу* ІСКЗ, оскільки корпорації намагаються уникати витрат і ризику вкладення капіталу в нові технології, що ще не набули значного поширення.

Розділ 5

БАЗИ ЗНАНЬ

5.1. Модель предметної області

Під час інтелектуальної діяльності людина створює у своїй свідомості *модель зовнішнього світу (довкілля)*, в якій відображаються і запам'ятовуються *образи предметів* (об'єктів) тієї чи іншої ПрО, їхні властивості та зв'язки між ними. Формування людиною *моделі довкілля* відбувається в процесі *навчання й адаптації*, до різноманітних життєвих обставин [1.1–1.16].

За аналогією з людською моделлю *довкілля*, в КСШ також створюється *комп'ютерна модель ПрО*, необхідна для функціонування системи, яка являє собою структуровану сукупність *елементарних моделей (образів) об'єктів* цієї ПрО, їхніх *властивостей* і *зв'язків* між ними [2.1–2.41, 3.1–3.25, 4.1–4.31, 5.1–5.24].

Об'єктом будемо називати (тут і далі) досліджуваний *предмет* даної ПрО (фізичний об'єкт, суб'єкт, процес, явище, подія, дія, ситуація тощо.), а *образом* — комп'ютерну *модель* цього *об'єкта*.

Модель ПрО містить у собі такі складові частини:

- *об'єктна модель* описує структуру ПрО як сукупність об'єктів;
- *функціональна модель* відбиває дії й перетворення над об'єктами;
- *поведінкова модель* розглядає взаємодію об'єктів ПрО у часі.

Модель ПрО створюється на основі *евристичних моделей* людського мислення, в яких використовуються такі *елементарні логічні форми* (розд. 2, 6).

1. *Поняття* (сутність) відображає (описує) найістотніші (необхідні й достатні) *ознаки* даного *об'єкта* ПрО. *Поняття* має *зміст* — число ознак описуваного предмета й *обсяг* — сукупність структурних елементів об'єкта. Залежно від *змісту*, поняття поділяються на *позитивні* й *негативні*, *відносні* й *безвідносні*, *порівнювані* і *непорівнювані*. Порівнювані поняття можуть бути *сумісними* й *несумісними* (див. рис. 5.1).

Є такі *види операцій* з поняттями:

- *порівняння* — установлення схожості або розбіжності понять;
- *аналіз* — розбиття поняття на складові частини;
- *синтез* — створення поняття зі складових частин (ознак, властивостей, відносин);
- *абстрагування* — виявлення в понятті визначеної групи ознак і абстрагування їх від інших ознак;

- *узагальнення* — об'єднання понять в однорідні групи на основі загальних ознак;
- *індукція* — дослідження об'єктів для формування правила створення поняття.

2. *Судження (твердження, вираження, висловлення)* являє собою речення з інформацією про *стан* одного чи кількох досліджуваних *об'єктів* і містить у собі (розд. 6.2):

- *суб'єкт* — логічний *підмет*, що відображає сукупність розглянутих об'єктів;
- *предикат* — логічний *присудок*, що визначає істинність або *хибність* судження;
- *квантор* спільності, що вказує на істинність судження для *всіх* або *деяких* об'єктів.

Судження може бути *простим* або *складним* і складається відповідно з *одного* чи *кількох* *речень*.

3. *Міркування* є логічним процесом одержання (*виводу*) з одного чи декількох *суджень* A_1, A_2, A_n *нового судження* B :

$$A_1, A_2, A_n \rightarrow B \quad (5.1)$$

де *початкові* судження A_1, A_2, A_n називаються *посиланнями*, а *нове* судження B — *висновком (результатом)*; \rightarrow є знаком *імплікації* (розд. 6.2).

Ліву частину формули (5.1) називають *антедентом*, праву — *консеквентом*.

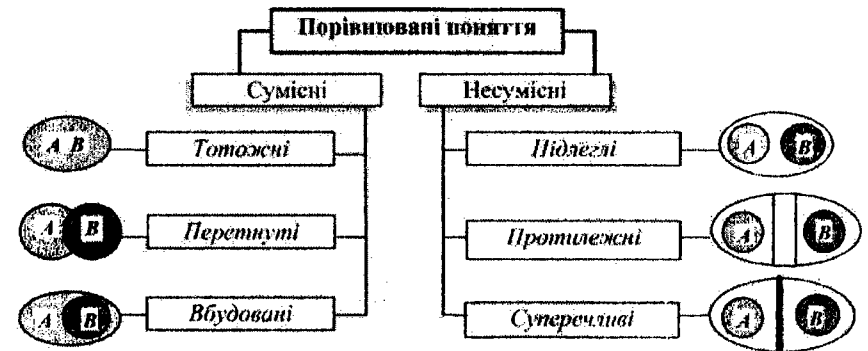


Рис. 5.1 — Відношення порівнювальних понять

Є такі *типи процесів міркувань*:

- *дедуктивні* — від загального до часткового;
- *індуктивні* — від часткового до загального;
- *за аналогією* — від часткового до часткового.

Міркування є основним інструментальним засобом логічного створення нових знань, який являє собою багатоетапний перехід від посилань до висновків і далі — від отриманих висновків (у вигляді нових посилань) — до нових висновків (розд. 7).

5.2. Технологія створення бази знань

База знань (БЗ) КСШІ створюється на основі моделі ПрО і являє собою сховище знань, у якому нагромаджується вся наявна інформація про властивості й закономірності даної ПрО (розд. 3.3.1). БЗ моделює пам'ять людини і є найважливішою складовою частиною (комп'ютерним «мозком») КСШІ. Для БЗ характерні порівняно невеликий обсяг і велика питома вартість інформації [3.1–3.25, 4.1–4.31, 5.1–5.24].

Процес створення БЗ містить у собі такі етапи.

1. Текстуальний аналіз усіх протоколів сеансів добування знань, отриманих у процесі їхнього придбання. Під час аналізу складається словник ключових понять і термінів, які позначають об'єкти, явища, процеси, дії, ситуації, ознаки та інші образи ПрО.

2. «Просіювання» словника термінів для визначення так званих концептів, тобто узагальнених понять, кожне з яких відображає визначений клас образів відповідно до їхніх специфічних ознак. Наприклад, концепт «автомобіль» поєднує безліч різних марок автомашин (образів), що мають двигун, чотири колеса, інші одиниці вузлів (підвіски, гальма тощо) й призначених для перевезення людей і вантажів. Кожен концепт має набір атрибутів (властивостей, характеристик). Концепти визначаються на етапі концептуалізації КСШІ (розд. 3.5) і подаються у БЗ як факти та інші інформаційні одиниці.

3. Виявлення зв'язків між концептами (поняттями), для чого часто використовуються так звані сценарії, що відображають процедури (ситуації, дії), в яких беруть участь визначені об'єкти ПрО. Сценарії містять у собі сцени (фрагменти), між якими встановлюються часові й просторові зв'язки. Усередині фрагмента можуть бути функціональні, ситуативні, асоціативні та інші види зв'язків (розд. 6).

4. Структурування понять для встановлення числа рівнів знань і метапонять високого рівня. Прикладами метапонять можуть слугувати концепт «рік», (що включає в себе поняття «весна», «літо», «осінь», «зима») чи концепт «доба» («ранок», «день», «вечір», «ніч»).

5. Побудова піраміди (ієрархічної структури) знань за рівнями їхньої абстракції (спільності).

6. Визначення відносин між знаннями усередині кожного рівня і між рівнями ієрархії. Відносини подаються у БЗ як правила та метаправила (див. нижче).

7. Складання ланцюжків міркувань, необхідних для вирішення задач ПрО і поєднуючих усі сформовані поняття й відносини в єдину ієрархічну динамічну структуру, що й становить основу БЗ КСШІ.

Слід зазначити, що процес створення БЗ має неформальний характер і часто потребує великої винахідливості, що залежить від професійних здібностей розробників КСШІ. Вони повинні ретельно продумати концептуальну структуру БЗ і вибрати найкращий варіант її побудови із кількох можливих..

В ідеалі БЗ повинна бути структурованою, несуперечливою, повною і текстологічно (синтаксично й граматично) коректною. Особливістю БЗ є динамічно змінюваний склад знань за рахунок постійного їх оновлення, зміни та розвитку у процесі створення й експлуатації КСШІ. Тому в міру збільшення обсягу й ускладнення БЗ за рахунок доповнення нових знань і розширення функцій КСШІ досягти такого ідеального стану БЗ стає все складніше.

Процес безпосереднього створення й супроводу БЗ здійснюється аналітиком за допомогою інтелектуального редактора БЗ, який містить програмно-апаратні засоби діалогового інтерфейсу користувача, когнітивної графіки і текстологічного аналізу знань, що вводяться в БЗ (розд. 3.3). Редактор БЗ дає змогу аналітику створювати і корегувати БЗ у режимі інтелектуальної взаємодії із системою, під час якої обговорюються й уточнюються наявні й здобуті системою знання. При цьому редактор автоматично перевіряє текстологічну коректність і взаємну несуперечність знань, що вводяться у БЗ.

5.3. Основні функціональні елементи бази знань

Як було зазначено вище, БЗ КСШІ являє собою структуровану сукупність елементарних логічних форм, що утворюють модель даної ПрО (розд. 5.1). Основними функціональними елементами БЗ є наступні евристичні форми людського (логічного) мислення [3.1–3.25, 4.1–4.31, 5.1–5.24].

1. Факт являє собою твердження (судження), яке характеризує (описує) відношення між двома чи кількома об'єктами (поняттями) ПрО. Факти поділяються на константи, перемінні й наслідки. Факти можуть бути істинними, помилковими або правдоподібними, коли істинність факту допускається з деяким ступенем ймовірності (упевненості).

Наприклад, у ПрО «Метеорологія» (поняттями якої є «атмосферний тиск», «барометр», «небо», «хмари», «дощ» тощо) можуть бути наявними такі факти:

- *константи*: «барометр вимірює тиск», «дощ утворюється у хмарах» тощо;
- *перемінні (робочі факти)*, що можуть вводитися в систему в процесі розв'язання задачі (розд. 7.2): «Небо вкрите хмарами», «Барометр падає» тощо;
- *наслідки*, одержувані системою на основі інших (робочих) фактів шляхом виводу на знаннях: «Незабаром піде дощ».

2. Правило є логічною формою міркування і являє собою взаємозалежну сукупність кількох тверджень (фактів), які описують причинно-наслідковий зв'язок між властивостями об'єктів або відображують закономірності даної ПрО. Правило має вигляд (5.1):

$$P_1 \& P_2 \& \dots \& P_m \quad Q_1 \& Q_2 \dots \& Q_n, \quad (5.2)$$

де P_i ($i = 1 \dots m$) — твердження-умови; Q_j ($j = 1 \dots n$) — твердження-наслідки; $\&$ — логічний знак «і» (кон'юнкція); \rightarrow — логічний знак імплікації (розд. 6.2).

Правило інтерпретується таким чином:

- *умови (посилань)* P_i ($i = 1 \dots m$) породжують наслідки (дії) Q_j ($j = 1 \dots n$).

У прикладі з ПрО «Метеорологія» (див. вище) правило (5.2) можна подати у вигляді (розд. 6.5):

$$\{ \text{Якщо «Небо вкрите хмарами» і «Барометр падає»} \\ \text{то «Незабаром піде дощ»} \} \quad (5.3)$$

3. *Метаправила* являють собою підтримуючі й керуючі правила застосування правил, основним призначенням яких є:

- *організація (структурування)* знань у БЗ і робочій пам'яті вирішувача;
- виявлення й організація нових знань;
- обґрунтування доцільності занесення нових знань у БЗ;
- вибір із БЗ правил, релевантних (відповідних) поставленій задачі;
- виявлення суперечливих (конфліктних) і помилкових знань у БЗ;
- адаптація БЗ до змінювання ПрО;
- інформаційна підтримка блоку пояснень КСШІ.

Метаправила істотно відрізняються від звичайних правил, оскільки вони не беруть безпосередньої участі у процесі вирішення задач, але використовуються на метарівні організації процесу вирішення задачі й пояснення отриманих результатів.

Стосовно до ПрО «Метеорологія» (див. вище) *метаправило* може бути подано, наприклад, у такому вигляді:

{ Якщо в запиті користувача присутня інформація «Небо вкрите хмарами», то необхідно занести в робочу пам'ять вирішувача правило (5.3) та інші правила, які вміщують ключові поняття «небо» і «хмари» }. (5.4)

5.4. Структура бази знань

БЗ являє собою *структуровану інформаційну мережу*, яка визнає, що з чим пов'язано, що на що впливає, що є причиною, а що — наслідком, що являє собою частину, а що — ціле, тощо. Таку мережу треба вміти економно записувати в БЗ, вибирати з неї потрібні знання, знаходити місце для запису нових знань тощо. Комп'ютерний «мозок» не так важливо наповнити, як добре влаштувати і вміло організувати [3.1–3.25, 4.1–4.31, 5.1–5.24].

Знання, що заносяться у БЗ, попередньо *структуруються* за аналогією з людськими знаннями, які у людини мов би «розкладені на полицях» у *когнітивних структурах мозку*. На одній «полиці» зберігаються знання про те, як треба готувати їжу, на іншій — як добратися до роботи тощо. Поділ знань на окремі сфери дає змогу людині заощаджувати час на пошук потрібних знань.

Подібний ієрархічний поділ за *рівнями знань* здійснюється й у БЗ. Число *рівнів* визначається специфікою розв'язування задач, обсягом знань і способом їх подання. Як правило, виокремлюється не менше трьох рівнів, що відображують загальну, логічну й фізичну організацію знань. Зазвичай, зміна знань на одному з них не зачіпає інші рівні. Водночас наявність різних рівнів запобігає поширенню змін з одного на інші.

Для найефективнішого опрацювання інформації пам'ять БЗ розбивається на окремі *незалежні підобласті (банки) знань* для того, щоб зміни в одній підобласті не призводили до істотних змін в інших підобластях знань. У цьому разі вирішення складної задачі можна звести до кількох простіших підзадач, а для вирішення кожної з них достатньо знань лише з однієї підобласті.

Основними *підобластями знань* БЗ є:

- *модель ПрО* (зовнішнього світу), тобто сукупність знань (фактів, правил, метаправил тощо), що описують дану ПрО (розд. 5.2);
- *алгоритми (правила) виводу на знаннях* (розд. 7.2);
- сукупність *метапроцедур (метаправил)* над правилами виводу;
- знання про способи *взаємодії КСШІ з довкіллям* та інша інформація, необхідна для рішення інтелектуальних задач.

Поряд із зазначеними підобластями знань БЗ містить також знання про *власні* характеристики та можливості КСШ. До числа таких знань належать:

- можливості, принципи роботи й розвитку системи;
- основні характеристики ПрО (та її моделі), у якій працює система;
- характеристики вирішувача та інших програмно-апаратних засобів системи;
- характеристики моделей подання й модифікації знань;
- опис алгоритмів вирішення задач;
- форми та способи організації діалогового інтерфейсу з користувачами;
- інформація про мову діалогового спілкування з користувачами;
- класифікація (види) користувачів за рівнями компетенції та підготовленості до роботи із системою;
- структурований перелік питань, які система може ставити користувачам;
- агрегований перелік відповідей (пояснень) на питання користувачів за період експлуатації системи;
- відомості про фахівців, що брали участь у створенні КСШ тощо.

Знання про *власні* характеристики КСШ можуть бути розміщені також у відповідній підобласті БЗ або в окремо сформованій базі даних (розд. 3.3).

5.5. Онтології

Онтологія являє собою базу концептуальних знань про об'єкти довкілля (розд. 5.1) і зв'язки між ними. Основу онтології становить взаємозалежна й погоджена *таксономія* (ієрархічна класифікація) *концептів*, їхніх *властивостей* (характеристик) і *правил* опрацювання [5.15–5.24]. Поряд із концептами даної ПрО, які формуються на етапі *концептуалізації* (розд. 3.5), онтологія містить також концептуальні знання про *інші* (родинні) ПрО, що можуть бути використані при вирішенні задач у даній КСШ.

Формальну модель онтології (ОнЛ) у даній ПрО можна подати у вигляді:

$$\text{ОнЛ} \equiv \langle \text{П}, \text{О}, \Phi \rangle$$

де: П — множина *понять* об'єктів ПрО; О — множина *відносин* між *поняттями*; Φ — множина *функцій інтерпретації понять* і *відносин*.

Наприклад, *концептуальна модель ОнЛ*, що відображає структуру *корпорації* (її основні структурні підрозділи й зв'язки між ними), містить у собі *концепти* двох класів: «*співробітник*» і «*підрозділ*».

Формальна модель такої ОнЛ складається з таких елементів:

- *поняття* $\text{П} \equiv \{\text{«генеральний директор»}, \text{«директор»}, \text{«заст. директора»}, \text{«виробництво»}, \text{«відділення»}, \text{«ділянка»}, \text{«контактний телефон»}, \text{«e-mail»}, \text{«адреса» тощо}\};$
- *відносини* $\text{О} \equiv \{\text{«підкоряється»}, \text{«є частиною»}, \text{«обіймає посаду»}, \text{«знаходиться за адресою» тощо}\};$
- *функції інтерпретації* $\Phi \equiv \{\text{«Якщо } A \text{ є частиною } B \text{» і «} B \text{ є частиною } C \text{», то «} A \text{ є частиною } C \text{»}\}, \{\text{Якщо } X \text{ є начальником } Y, \text{ то } Y \text{ підкоряється } X\}.$

Залежно від здатності виконувати так званий *онтологічний аналіз* своїх *станів* у процесі функціонування КСШ, ОнЛ поділяються на такі категорії:

- *статична ОнЛ* не виконує онтологічний аналіз своїх станів;
- *динамічна ОнЛ* визначає стани, що виникають у процесі вирішення задач, і виробляє способи перетворення одних станів у інші;
- *епістемічна ОнЛ* виробляє метазнання, які керують процесом переходу з одного стану в інший.

Останнім часом у межах робіт із КСШ створюються автономні *онтологічні системи*, що являють собою ієрархічні *структури* взаємозалежних ОнЛ, які містять у собі такі *концептуальні бази знань* [5.15–5.24].

1. *МетаОнЛ* оперує загальними (філософськими) *концептами* й *відносинами метарівня* (типу «час», «простір», «причина», «наслідок», «властивість», «значення» тощо), які не залежать від конкретної ПрО.

2. *Предметна ОнЛ* містить у собі:

- *поняття*, що описують об'єкти конкретної ПрО;
- *семантичні відносини* між поняттями (об'єктами) ПрО;
- безліч *декларативних і процедурних інтерпретацій* понять і відносин;

3. *ОнЛ задач* містить набір *функцій інтерпретації* різних типів *задач*, що розв'язуються у даній ПрО.

Усі зазначені моделі ОнЛ містять *концепти* (поняття, класи), *властивості* концептів (атрибути, ролі), *відносини* між концептами (залежності, функції) і додаткові *обмеження*, пов'язані з їх формуванням.

«*Життєвий цикл*» створення й функціонування *онтологічної системи* передбачає виконання таких робіт.

- а) *керування проектом*, що передбачає:
 - планування й організацію процесу розв'язання задач;
 - облік тимчасових і ресурсних витрат, необхідних для створення й функціонування ОнЛ;
 - моніторинг процесу розв'язання задач;
 - контроль якості виконуваних робіт;

б) розробка ОнЛ містить у собі:

- специфікацію, яка визначає мету створення ОнЛ, її передбачуване використання та потенційних користувачів;
- концептуалізацію, яка забезпечує структурування предметних знань у вигляді концептуальної моделі;
- формалізацію, за якої концептуальна модель трансформується у формалізовану;
- реалізацію, у процесі якої формалізована модель програмується відповідною мовою подання знань;

в) інформаційна підтримка ОнЛ передбачає:

- придбання нових знань про дану ПрО;
- оцінку ОнЛ, програмного забезпечення і відповідної документації;
- інтеграцію з наявними ОнЛ;
- документування (опис) ОнЛ;
- керування конфігураціями, необхідне для архівації документації, програмного забезпечення, коду ОнЛ і контролю за змінами ПрО.

До складу онтологічної системи входять:

- глосарій термінів, який містить всі ключові поняття (концепти, атрибути, дії тощо), які описують дану ПрО;
- дерева класифікації концептів, які ідентифікують основні таксономії ПрО;
- діаграми бінарних відносин, що визначають відносини між концептами однієї чи кількох онтологій.

Для кожного дерева класифікації концептів розробляються:

- словник концептів ПрО, який вміщує екземпляри концептів та їхні атрибути, а також відносини між концептами;
- таблиця бінарних відносин, у якій фіксуються імена відносин, концептів-джерел і цільових концептів;
- таблиця атрибутів екземплярів концептів, яка містить імена, типи значень, одиниці, діапазони й точність вимірів атрибутів, правила висновку атрибутів та інші характеристики;
- таблиця атрибутів для кожного класу зі словника концептів;
- таблиця логічних аксіом, у якій концепти визначаються як логічні вирази (аксіоми), що містять імена, природно-мовний опис, атрибути концептів тощо;
- таблиця констант, яка містить імена, значення, одиниці вимірів констант, їх природно-мовний опис і атрибути;
- таблиці формул містять імена формул, атрибути, виведені за допомогою тієї чи іншої формули, природно-мовний опис, точність і обмеження на використання формул; для кожної формули складається окрема таблиця;
- дерева класифікації атрибутів представлені у вигляді графів взаємозв'язку атрибутів, констант і формул;

- таблиця екземплярів складається для кожного входу у словник концептів і містить імена, атрибути та значення екземплярів атрибутів.

Сьогодні онтологічні технології широко використовуються в системах комп'ютерного добування знань (розд. 4.5), у мульти-агентних системах (розд. 13), в інформаційних системах пошуку знань у середовищі Internet та інших інтелектуальних системах. До числа основних практичних розроблень у цій області можна віднести такі онтологічні системи [5.15–5.24].

МетаОнЛ «CYCL» являє собою семантичну структуру концептів, зв'язків між ними й аксіом. ОнЛ вміщує 10^6 концептів і 10^5 аксіом. Для подання знань у межах цього проекту розроблена спеціальна мова CYCL.

МетаОнЛ системи «Generalized Upper Model» організована як ієрархія концептів (понад 250 понять) і зв'язків між концептами. МетаОнЛ орієнтована на опрацювання англійської, німецької та італійської мов.

Предметна ОнЛ системи «TOVE» («Toronto Virtual Enterprise Project») призначена для виконання запитів користувачів в області реінжинірингу бізнес-процесів із використанням фреймової моделі корпорації (розд. 6.5) і дедуктивного виводу на знаннях (розд. 7.2).

ОнЛ задач системи «Plinius» призначена для напіваавтоматичного добування знань із текстів в області хімії. Концепти цієї ОнЛ відповідають фреймам-ролям, які грають об'єкти ПрО в процесі виконання тих чи інших дій.

ОнЛ проекту «(KA)²» («Knowledge Annotation Initiative of the Knowledge Acquisition Community») здійснює інтелектуальний пошук знань у середовищі Internet і підтримує три основні напрями досліджень:

- онтологічний інжиніринг;
- анотацію Web-сторінок;
- організацію відповідей на запити користувачів на Web-сторінках.

У межах проекту «(KA)²» розроблені ОнЛ організації, напрями досліджень, проекту, особистості, публікацій, подій, дослідницьких груп і результатів досліджень, розміщені на Web-сторінках.

Однією з основних частин проекту «(KA)²» є онтологічна система «Ontobroker», до складу якої входять:

- інтерфейс формування запитів, орієнтований на фреймове подання онтологій, у межах якого визначені традиційні поняття екземплярів, класів, атрибутів та значень;
- машина створення (розд. 7.2) відповідей, яка складається із транслятора мови подання знань і обчислювача виразів мови логічного програмування;

• машина доступу до *internet*-ресурсів «*Webcrawler*» («Черв'як»), яка здійснює формування анотацій Web-сторінок із використанням онтоотегів (HTML-тегів нового типу).

ОнЛ проекту «*SHOE*» («*Simple HTML Ontology Extensions*») містить ієрархію класів (категорій) термінів (концептів), множину відносин між категоріями і множину правил виводу з використанням фраз Хорна (розд. 6.2). ОнЛ «*SHOE*» має змогу вносити у Web-сторінки додаткової семантичної інформації за рахунок її інтеграції з ОнЛ інших ПрО. Це дозволяє одержувати більш детальні анотації Web-документів і релевантні відповіді на запити користувачів Internet.

Концепція мови представлення знань ОнЛ «*SHOE*» за своєю суттю близька до ОнЛ проекту «(K4)²» і полягає в суттєвому розширенні мови HTML за рахунок введення нових спеціальних *тегів* для специфікації онтологій. Для автоматизації процесів анотування Web-документів розроблений спеціальний *анотатор знань*, до складу якого входять блоки *екземплярів*, *онтологій* і *тверджень*. Користувач може додавати, редагувати й видаляти будь-які елементи цих блоків.

ОнЛ системи «*OntoSeek*» базується на використанні модифікацій так званих *концептуальних графів* для подання запитів користувачів і опису системних ресурсів. Вузли й дуги *концептуального графа* зв'язуються визначеними семантичними відносинами з відповідними лексичними (лінгвістичними) концептами ОнЛ. Задача *контекстного ототожнення* зводиться до керованого процесу *зіставлення на графі релевантності* тексту Web-сторінки відповідним фрагментам ОнЛ.

Функціонування системи «*OntoSeek*» здійснюється у такий спосіб. Користувач подає свій запит у вигляді *концептуального графа*, на якому він *позначає* вузли й дуги, що відповідають розв'язуваній задачі. *Кодувальник ресурсів* системи перетворює граф користувача в *позначений концептуальний граф (ПКГ)*, вузлами й дугами якого є лексичні концепти зі *словника термінів* і ОнЛ системи. Після усунення лексичної й семантичної неоднозначності ПКГ надходить у *класифікатор* системи, який зіставляє ПКГ із *концептуальними графами бази знань* системи. При *ототожнюванні* ПКГ з одним із графів формується відповідь користувачу як HTML-звіт про знайдений граф. Архітектурним ядром системи «*OntoSeek*» є *сервер онтологій*, що підтримує інтерфейси доступу додатків, процедури маніпулювання знаннями ОнЛ, а також зв'язок із базою даних *концептуальних графів*. Більшість сучасних ОнЛ створюються в межах глобального проекту *Semantic Web* консорціуму *W3C*, де представлені сучасні технології семантичної опрацювання інформації [5.24]. Основу цього проекту становлять мультиагентні та онтоло-

гічні системи, сучасні засоби подання знань і нові мови ОнЛ. Сьогодні є низка *інструментальних* онтологічних засобів аналізу, редагування, візуалізації, імпорту й експорту ОнЛ різних форматів, їх об'єднання, порівняння тощо.

До таких засобів належать:

- редактори *Ontolingua*, *Protege*, *DOE (Design Ontological Environment)*, *DUET*, *RDF Edit*, *OntoEdit*, *OntoSaurus*, *OilEd*, *WebOnto*, *ODE*, призначені для створення, перевіряння й модифікації ОнЛ;

- інструментальні системи *PROMPT*, *Chimaera*, *OntoMerge*, *OntoMorph*, *Observer*, *Onion*, призначені для аналізу, об'єднання, вирівнювання (пошуку відповідностей між елементами) і відображення (перебування зв'язків між елементами) різних ОнЛ.

Редактори ОнЛ, зазвичай, використовують графічні засоби редагування й перевіряння ОнЛ, де класи понять подані вузлами (вершинами) на графах, а відносини — дугами між вузлами.

Розділ 6

МОДЕЛІ ПОДАННЯ ЗНАНЬ

6.1. Основні відомості

Однією з найважливіших проблем, що виникають при створенні КСШІ, є вибір (розроблення) способів *формалізації й подання знань* у БЗ. Ця проблема надзвичайно актуальна, оскільки ефективність функціонування системи багато в чому визначається повнотою і якістю знань про ту ПрО, в який працює КСШІ.

Усі знання, отримані в процесі добування з різних джерел інформації (розд. 4.4), *формалізуються й подаються* у БЗ мовою, максимально наближеною до природного (людського) розуміння та спілкування. Це необхідно для організації діалогової взаємодії системи з користувачем, який має розуміти мову комп'ютера, водночас не будучи фахівцем в області комп'ютерної техніки.

Модель подання знань (МПЗ) описує (природною мовою) знання про об'єкти і зв'язки між ними, про ієрархію понять, динаміку змін відносин між об'єктами та інші властивості ПрО. Процес розроблення й застосування МПЗ у КСШІ містить [2.1–2.41, 3.1–3.25, 4.1–4.31, 5.1–5.24, 6.1–6.20]:

- аналіз, структурування, нагромадження та збереження знань у БЗ;
- введення нових знань у БЗ і об'єднання їх із наявними знаннями;
- пошук необхідних знань для вирішення поставленої задачі;
- видалення з БЗ надлишкових і застарілих знань;
- перевіряння несуперечності, нагромаджених у БЗ знань тощо.

На відміну від *математичної* моделі, МПЗ є *евристичною моделлю* (розд. 2) і має *семантичний (змістовний)* характер. Використання *евристичних* моделей, поданих природною (людською) мовою, дає змогу суттєво прискорити процеси введення інформації, взаємодію із системою, розв'язання задач і тим самим забезпечити ефективну роботу КСШІ.

МПЗ може бути *універсальною*, пристосованою для різних ПрО, або *спеціалізованою*, що відображає конкретні знання фахівців у даній ПрО і придатною тільки для неї.

БЗ *ранніх КСШІ* будувалися на основі *формальних логіко-лінгвістичних МПЗ* із використанням мови логіки предикатів, що дає змогу подавати знання у формі, одночасно близькій і до звичайної (людської) мови, і до мови програмування.

На сьогодні у КСШІ використовують *семантичні, фреймові, продукційні* та інші види МПЗ. У наступних розділах розглянуті основні властивості й характерні риси зазначених моделей.

На практиці у багатьох КСШІ використовують *комбінації мережових (семантичних чи фреймових) і продукційних* моделей подання знань. Найбільш популярними сьогодні є *продукційно-фреймові* моделі. У таких моделях *декларативні* знання описуються *мережовою* моделлю, а *процедурні* знання — *продукційною* МПЗ. У цьому випадку говорять про *продукційний вивід на знаннях* у семантичній чи фреймовій мережі (розд. 7).

Слід зазначити, що з погляду користувача в КСШІ бажано використовувати *єдину МПЗ*. Це спрощує введення знань і взаємодію із системою, полегшує створення й супровід системи, прискорює процедуру висновку на знаннях. Проте, однаковість може привести до труднощів у разі спроби втиснути різні за своєю природою знання в один і той самий формалізм. Таким чином, у питанні про подання знань є «золота середина» між крайнощами — повною однаковістю й вузькоспеціалізованим формалізмом МПЗ.

6.2. Логічні моделі подання знань

6.2.1. Основні положення

Як було зазначено в розділі 6.1, *логічні (логіко-лінгвістичні) моделі* були одними з перших комп'ютерних засобів подання знань в історії розвитку інтелектуальних систем. Підґрунтям цих моделей була *логіка предикатів*, що є основним засобом *формалізації лівопівкульного мислення людини* (розд. 2) [2.1–2.11, 2.15–2.19, 3.5–3.11, 6.2–6.4, 6.16].

Відповідно до *логіки предикатів* *логічна модель подання знань* являє собою структурований набір відповідних *речень (суджень) природної (людської) мови* (розд. 5.1), поданих у вигляді *логічних формул числення предикатів* першого порядку.

Предикатом називається *логічна функція*, яка залежить від однієї чи декількох *аргументів (перемінних)* і приймає одне із двох значень: «істина» чи «хибність». *Аргумент* предиката називається *термом*. Число *термів* визначає так звану *арність* предиката. *Унарний предикат* залежить від однієї перемінної, наприклад: «*птах* (*x*)», де *x* — *терм*, що визначає конкретну назву того чи іншого птаха. Три *твердження*: «*7 + 5 = 12*» «*8 + 9 = 17*» «*4 + 5 = 10*» відповідають *бінарному предикату* «*плюс* (*x, y, z*)». Перші дві підстановки конкретних значень замість *x, y, z* породжують істинне твердження, третя — помилкове.

Предикати, що мають істинне значення, називають *аксіомами* (*постулатами*). Елементарною аксіомою є літерал (атом, атомарна формула), що являє собою елементарне твердження, яке складається із предиката і зв'язаних з ним термів:

$$P(t_1, t_2, \dots, t_n), \quad (6.1)$$

де P — позначення предиката; t_i ($i = 1 \dots n$) — терми.

Літерал називається *негативним*, якщо він стоїть під знаком заперечення, або *позитивним* — в іншому разі.

6.2.2. Основні елементи логіки предикатів

У численні предикатів використовуються такі *позначки логічних елементів*:

- константні терми: c_1, c_2, \dots ;
- перемінні терми: x_1, x_2, \dots ;
- функціональні літери: f_1, f_2, \dots ;
- предикатні літери: p_1, p_2, \dots ;
- логічні символи: $\rightarrow, \leftrightarrow, \&, \vee, \sim, \forall, \exists$;
- спеціальний символ, що позначає «суперечність».

У табл. 6.1 наведені основні логічні оператори, що використовуються в численні предикатів.

Таблиця 6.1 — Логічні оператори

Оператор	Запис	Інтерпретація
Кон'юнкція («і»)	$a \& b$	« a і b »
Диз'юнкція («або»)	$a \vee b$	« a або b »
Заперечення («не»)	$\sim a$	«не a »
Імплікація (« \rightarrow »)	$a \rightarrow b$	«якщо a то b »
Тотожність (« \leftrightarrow ») (еквівалентність, рівність)	$a \leftrightarrow b$	« a еквівалентно b »
Помилковість	a	«помилковість a »

За допомогою логічних операторів (табл. 6.1) з елементарних тверджень (літералів) можуть бути побудовані складні низки логічних речень (правил).

З кожним предикатом може бути зв'язаний *квантор*, що визначає, за яких умов предикат перетворюється в істинне твердження.

Серед кванторів числення предикатів можна видокремити *квантор узагальнення* (*all* — для усіх) і *квантор існування* (*exist* — існує). Наприклад:

- запис $\forall x : P(x)$ означає, що $P(x)$ істинно для всіх x ;
- запис $\exists x : P(x)$ означає, що існує x , для якого $P(x)$ істинно.

Диз'юнкція деякої сукупності літералів називається *фразою* (диз'юнктом). Диз'юнкція, серед літералів якої є лише один позитивний літерал, а всі інші — негативні, називається *фразою Хорна*. Наприклад, диз'юнкція

$$\sim A \vee \sim B \vee \sim C \vee D,$$

у якій усі літерали, окрім D , є негативними, являє собою *фразу Хорна*. Вона еквівалентна логічним формулам:

$$\sim (A \vee B \vee C) \vee D \text{ і } A, B, C \rightarrow D,$$

тобто фрази Хорна за своєю суттю є *імплікаціями*.

У табл. 6.2 наведені можливі перетворення логічних формул у рівнозначні, що дають змогу будувати твердження у вигляді *фраз*, у тому числі — *фраз Хорна*.

Таблиця 6.2 — Перетворення логічних формул

Перетворення	Формула	Рівнозначні формули
Імплікація	$a \rightarrow b$	$\sim a \vee b$
Тотожність	$a \leftrightarrow b$	$a \rightarrow b \& b \rightarrow a$ $a \& b (\sim a \& \sim b)$
Переміщення заперечення	$\sim (a \& b)$	$\sim (a \vee \sim b)$ $\sim a \vee \sim b$
Переміщення заперечення	$\sim \forall (a, b)$	$\exists (a, \sim b)$
Перетворення кванторів	$\sim \exists (a, b)$	$\forall (a, \sim b)$
Перенесення операцій	$(a \& b) \vee c$	$(a \vee b) \& (b \vee c)$
	$a \vee (b \& c)$	$(a \vee b) \& (a \vee c)$

6.2.3. Формування логічної бази знань

БЗ, побудована з використанням логічної моделі подання знань, називається логічною БЗ. Вона складається з упорядкованого набору аксіом (істинних тверджень), що можуть бути або фактами, або правилами (розд. 5.3).

Фактом у логічній БЗ є предикат (6.1), аргументами (термами) якого є константи, наприклад: $P(a, b, c)$. Правило складається із взаємозалежної послідовності кількох тверджень (фактів) у вигляді (5.2).

Формування логічної БЗ здійснюється таким чином (розд. 5.2).

1. Визначаються базові елементи даної ПрО, основними з яких є предметні константи, предметні перемінні і предикати.

2. Аксіоми (правила) бази знань подаються у пренексній (префіксній) нормальній формі, яка має такий вигляд:

$$K_1 x_1 K_2 x_2 \dots K_n x_n M, \quad (6.2)$$

де K_i ($i = 1 \dots n$) — i -й квантор (існування або узагальнення) x_i -го предиката; M — деяка безкванторна кон'юнктивна нормальна форма, тобто кон'юнкція визначеного числа диз'юнктив.

3. Далі із пренексної нормальної форми виключаються квантори існування шляхом введення в аксіоми так званих констант і функцій Сколема, наприклад: $\exists x: P(x) \Rightarrow P(c)$;

$$\forall y \exists x: P(x, y) \Rightarrow \forall y: P(h(y), y),$$

де c і $h(y)$ — константа і функція Сколема.

Квантори узагальнення виключаються автоматично на основі стандартної процедури, що звільняє формулу від кванторів, наприклад:

$$\forall x: P(x) \Rightarrow P(x);$$

$$\forall x \forall y \forall z: P(x, y, z) \Rightarrow P(x, y, z).$$

4. Після виключення кванторів існування й узагальнення необхідно подати аксіоми бази знань у вигляді фраз Хорна, скориставшись логічними перетвореннями табл. 6. 2.

Приклад. Нехай вербальна (словесно-змістовна) (розд. 4.1) модель ПрО «Батьки і діти» містить у собі такі факти:

«Боб є сином Кості»,

«Костя є сином Олега»

і правила: «Якщо x є сином y , то x є нащадком y »;

«Якщо y є сином z , то y є нащадком z »;

«Якщо x є нащадком y і y є нащадком z , то x є нащадком z ».

Побудова логічної БЗ на основі приведеної вербальної моделі ПрО здійснюється в такий спосіб.

1. Вводяться константи: B — Боб, K — Костя, O — Олег і предикати:

$$S(x, y): x \text{ є сином } y;$$

$$P(x, y): x \text{ є нащадком } y.$$

2. З урахуванням уведених констант і предикатів факти логічної БЗ будуть мати вигляд:

$$S(B, K); \quad S(K, O).$$

3. Подамо правила (імплікації) БЗ у пренексній нормальній формі (6.2):

$$\forall x \forall y: (S(x, y) \rightarrow P(x, y));$$

$$\forall x \forall y: (S(y, z) \rightarrow P(y, z));$$

$$\forall x \forall y: (((P(x, y) \& P(y, z)) \rightarrow P(x, z)).$$

4. Вилучаючи квантори узагальнення і враховуючи формули табл. 6.2, подамо факти і правила логічної БЗ у вигляді фраз Хорна:

$$S(B, K) \quad S(K, O)$$

$$\sim S(x, y) \vee P(x, y)$$

$$\sim S(y, z) \vee P(y, z)$$

$$\sim P(x, y) \vee \sim P(y, z) \vee P(x, z)$$

Сформована в такий спосіб логічна БЗ підготовлена до рішення логічних інтелектуальних задач (розд. 8.1, 15.2).

6.3. Семантичні моделі подання знань

Семантична модель знань у КСШІ являє собою орієнтований позначений граф, вершинами якого є поняття, а дугами — семантичні відносини між поняттями. В основу семантичної моделі покладена так звана семантична мережа, первинна версія якої розроблена американським психологом Куїліаном. Термін «семантична» походить від назви науки «семантика», що визначає смисл понять природної людської мови.

Семантична мережа (СМ) являє собою взаємозалежну структуровану сукупність тріад: «перше поняття — бінарне відношення — друге поняття». Взаємозалежна сукупність таких тріад утворює ієрархічну мережу, в якій вершини кожного рівня з'єднані дугами з відповідними вершинами верхнього і нижнього рівнів [3.5—3.25, 6.5, 6.10—6.17].

Поняття в СМ описують конкретні або абстрактні *об'єкти* Про, що характеризуються *доменами*, тобто параметрами і (або) константами. У СМ використовуються такі основні типи *понять*:

- *дія* (подія, явище, процес) — зміна стану об'єкта (об'єктів) Про; обумовлене рухом до поставленої мети;
- *властивості* — якісні характеристики (*домени*) об'єктів і дій;
- *значення* — кількісні оцінки дій і властивостей.

Для опису *дій* у СМ використовується спеціальний тип відносин у вигляді таких *триад*:

- *агент* — діюча особа, що викликає дію;
- *об'єкт* — предмет, на який спрямована дія;
- *адресат* особа, яка використовує результат дії.

Властивостями дій можуть бути час, тривалість, місце дії тощо.

Властивостями *понять* можуть бути колір, розміри, якість тощо.

Основними типами *відносин* у СМ є:

- «*ціле частина*» («клас — підклас», «множина — елемент»);
- «*клас — елемент класу*» («квітка — троянда»);
- «*властивість — значення*» («колір — жовтий»);
- «*приклад елемента класу*» («троянда — чайна»);
- *функціональні* («робить», «впливає»);
- *кількісні* («більше», «менше», «дорівнює»);
- *просторові* («далеко від», «близько від», «за», «під», «над»);
- *тимчасові* («раніше», «пізніше», «протягом»);
- *атрибутивні* («має властивість», «має значення», «є частиною», «належить»);
- *логічні* («і», «або», «не»);
- *лінгвістичні* тощо.

Залежно від типів відносин розрізняють такі *види СМ*:

- *однорідні СМ* — з одним типом відносин;
- *неоднорідні СМ* — з різними типами відносин;
- *бінарні СМ*, в яких відносини зв'язують два поняття;
- *багатозв'язні СМ*, в яких відносини зв'язують більш ніж два поняття.

Взаємозв'язок між елементами СМ здійснюється за допомогою таких *зв'язок*:

- *isa* — належність елемента СМ до певного класу;
- *part of* — належність частини до цілого;
- *is* — елемент є *властивістю*;
- *has* — елемент має властивість або має *відношення* до іншого елемента.

Як приклад на рис. 6.1 наведена *чотирирівнева СМ* Про «Власники автомобілів», до складу якої входять:

- 1) *вершини першого (верхнього) рівня* з *прізвищами власників* (5 *суб'єктів*);
- 2) *вершини другого рівня*, які визначають *властивості (вік) власників*;
- 3) *вершини третього рівня*, що визначають *об'єкти*, зв'язані з *об'єктами першого рівня* (тут — *марки автомобілів*);
- 4) *вершини четвертого (нижнього) рівня*, які визначають *властивості об'єктів третього рівня* (тут — *можливі кольори автомобілів*).

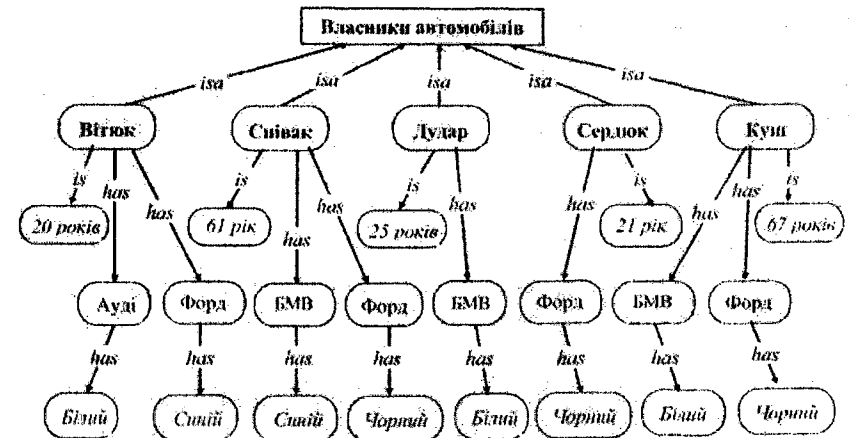


Рис. 6.1 — Приклад фрагмента семантичної мережі

Пошук рішення поставленої задачі з використанням *семантичної БЗ* зводиться до визначення *фрагмента СМ*, що містить у собі відповідь на *запит користувача*. Нехай, наприклад, у цьому випадку необхідно вирішити таку *задачу розпізнавання образу (об'єкта)*: «Знайти у *БЗ* *прізвище молодого власника чорного «Форда»*. Ця задача в *КСШ* вирішується таким чином (розд. 7).

1. У *робочу пам'ять вирішувача* заносяться *запит користувача* та *об'єкти першого рівня* (тобто *прізвища власників автомобілів*), серед яких знаходиться і шуканий об'єкт.

2. Виконується *аналіз запиту користувача* для визначення *базових (ключових) понять* розв'язуваної задачі і відповідних їм рівнів СМ. Такими поняттями тут є: «*молодий*» (другий рівень), «*Форд*» (третій рівень), «*чорний*» (четвертий рівень).

3. Перебираються *поняття другого рівня* і встановлюється, що властивістю «*молодий*» (тут — *менше 30 років*) володіють «*Вітюк*», «*Дудар*» і «*Сердюк*», що залишаються в *робочій пам'яті*, а об'єкти «*Снівак*» і «*Куш*» видаляються з неї.

4. Перебираючи поняття *третього* рівня, можна встановити, що «Форд» мають (із тих, що залишилися на попередньому кроці) «Вітюк» і «Сердюк», які й залишаються в робочій пам'яті, а об'єкт «Дудар» видаляється.

5. Нарешті, після розгляду понять *нижнього* рівня можна встановити, що *чорний* «Форд» має «Сердюк». Це твердження і є результатом розв'язання поставленої задачі.

Основною перевагою СМ є її відповідність сучасним уявленням щодо організації *довгострокової пам'яті людини*. Однак СМ є *пасивними* структурами, що не дають чіткого уявлення про структуру ПрО за її можливих змін. До недоліків СМ належить також складність організації процедури виведення знань. Тому СМ використовуються, зазвичай, у *сполученні* з іншими моделями подання знань, що розглянуті в наступних розділах.

6.4. Фреймові моделі подання знань

6.4.1. Фрейми

Фрейм («*frame*» — «рамка», «каркас») являє собою *таблицю*, в якій містяться основні *характеристики* (*атрибути*, *властивості*, *ознаки*) того чи іншого *об'єкта* даної ПрО. Поняття *фрейма* введено Мінським для відображення стереотипу сприйняття людиною об'єктів навколишнього світу [3.5—3.25, 6.4—6.14].

Таблиця-фрейм має назву, яка є унікальним *ім'ям фрейму*. Рядки *фрейму* (*таблиці*) називаються *слотами* («*slot*» — «щілина»), які мають *імена* та *значення*. Іменами слотів слугують *назви об'єктів*, а значеннями — їх *атрибути*. Атрибути фрейму можуть також бути *фреймами*.

У фрейму є *оболонка з порожніми слотами*, яка називається *протофреймом* (*прототипом*, *зразком*). При створенні конкретної БЗ КСШІ в даній ПрО *протофрейму* привласнюється *ім'я* і його слоти поступово наповнюються. Після наповнення *протофрейму* утворюється так званий *екзофрейм*, тобто *екземпляр* фрейму, який містить конкретну інформацію про об'єкт. Перехід від початкового *протофрейму* до *екзофрейму* зазвичай являє собою багатокроковий процес поступового уточнення значень слотів на кожному кроці.

Розглянемо приклад побудови *фрейму-структури* із фрагмента ПрО «Працівники підприємства» (табл. 6.3).

Таблиця 6.3 — Фрагмент ПрО «Працівники підприємства»

Прізвище	Рік народження	Спеціальність	Стаж (років)
Іванов	1925	Слюсар	30
Петров	1930	Токар	25
Сидоров	1965	Сантехнік	5

Структура *протофрейму*, в якому слоти є *порожніми*, наведена в табл. 6.4.

Цей протофрейм може бути використаний у *різних* ПрО як *оболонка порожнього фрейму-шаблону*. При конкретизації *протофрейму* в конкретній ПрО здійснюється заповнення слотів. У результаті з протофрейму утворюється *екзофрейм* (табл. 6.5), який шляхом поступового уточнення атрибутів перетворюється у *фрейм «Працівники підприємства»*.

Таблиця 6.4 — Протофрейм Таблиця 6.5 — Екзофрейм

Імена слотів	Значення слотів
Прізвище	
Рік народження	
Спеціальність	
Стаж	

Імена слотів	Значення слотів		
Прізвище	Іванов	Петров	Сидоров
Рік народження	1925	1930	1965
Спеціальність	Слюсар	Токар	Сантехнік
Стаж	30	25	5

Розрізняють такі типи *фреймів*:

- фрейми-структури, що використовують для позначення *понять (об'єктів)* даної ПрО (наприклад, «*позика*», «*застава*», «*вексель*»);
- фрейми-ролі («*менеджер*», «*касир*», «*клієнт*»);
- фрейми-сценарії («*банкрутство*», «*збори*», «*іменини*»);
- фрейми-ситуації («*тривога*», «*аварія*», «*режим*») тощо.

6.4.2. Слоти

Слоти є базовими елементами *фрейму*. Як було зазначено, слоти спочатку створюються *порожніми* і заповнюються в процесі *активізації (активації)* і *функціонування* фрейму відповідно до певних умов або директив, якими ці процеси супроводжуються.

Кожен слот містить ім'я, показчик спадкування (розд. 6.4.3) і тип даних, основними з яких є такі:

- *frame* — показчик; *integer* — цілий; *real* — дійсний; *bool* — булевий;
- *text* — текст; *list* — список; *table* — таблиця;
- *lisp* — приєднана процедура; *expression* — вираження тощо.

Значення слоту відповідає одному з указаних типів даних, умови спадкування і може бути ім'ям іншого фрейму, який забезпечує зв'язок (вкладеність) між фреймами. У слотах можуть бути посилання на імена інших слотів у даному фреймі або в інших фреймах. Слоти пов'язані між собою умовними відносинами, порушення яких слугує сигналом помилковості інформації, що вводиться.

Значеннями слотів можуть бути математичні співвідношення, тексти природною мовою, програми, правила виводу на знаннях, посилання на імена інших слотів у даному фреймі або в інших фреймах тощо. Значеннями слоту може бути набір слотів більш низького рівня, що дає змогу у фреймових поданнях реалізувати «принцип матрешки».

Є такі способи одержання слотом значень в екзофреймі:

- за замовчуванням — від *протофрейму*;
- спадкування властивостей *фрейму-батька*;
- за формулою, зазначеною у слоті;
- через приєднану процедуру;
- із діалогу з користувачем;
- із бази знань тощо.

Є *системні* й *користувацькі* слоти фрейму. Значеннями *системних слотів* можуть бути:

- показчик *фрейму-батька*;
- показчик прямого дочірнього фрейму;
- користувач фрейму;
- дата формування фрейму та його останньої модифікації тощо.

Системні слоти використовуються під час *редагування бази знань* і керування процесом *виводу на знаннях*. *Користувацькі* слоти заповнюються користувачем у процесі запиту й вирішення поставленої задачі.

6.4.3. Функціональні властивості фреймової бази знань

Фреймова БЗ являє собою ієрархічну декларативно-процедурну структуру (*мережу*) фреймів і пов'язаних із ними *процедур*, доступ до яких здійснюється безпосередньо із фреймів. *Фреймова БЗ* містить як

інформаційні, так і *процедурні* елементи, які забезпечують перетворення інформації у самих фреймах та їх зв'язок із іншими фреймами.

Значення атрибутів *фреймів верхніх рівнів фреймової БЗ* спільно використовуються всіма *фреймами нижніх рівнів*. У цьому разі інформація, що надходить із зовнішніх джерел Про у фрейми *нижніх рівнів*, має *вищий пріоритет*, ніж дані, успадковані від фреймів *верхніх рівнів*. Така *гнучка* структура *фреймової БЗ* дає змогу систематизувати поняття і додавати нові знання, спрощує процедуру перегляду й виявлення суперечностей у знаннях.

Значеннями слотів фреймової мережі можуть бути *процедури*:

- *спадкування властивостей*,
- *внутрішньої інтерпретації*,
- *установлення зв'язків*,
- *уведення відносин на слотах тощо*.

Зокрема, процедура *спадкування властивостей* дає змогу фреймам запозичати властивості інших фреймів за допомогою *показчиків спадкування* типу «АКО» («A-Kind-Of» — «це»), які показують, яку інформацію про атрибути слотів у фреймі *верхнього рівня* успадковують слоти з такими самими іменами у фреймі *нижнього рівня*.

На *рис. 6.2* наведено приклад *фреймової мережі*, в якій фрейм «Учень» успадковує властивості фреймів «Дитина» і «Людина», що знаходяться на вищому рівні ієрархії (стрілки вказують на фрейми, з яких успадковуються властивості).

Спадкування властивостей може бути *частковим*. Наприклад, *вік «Учня»* не успадковується із фрейма «Дитина», тому що він присутній у власному фреймі.

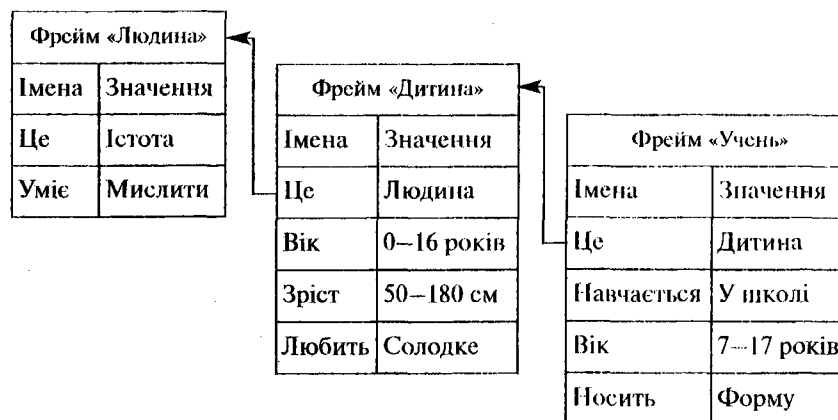


Рис. 6.2 — Фрагмент фреймової мережі з успадкуванням властивостей

6.4.4. Розв'язання задач у фреймових структурах

При розв'язанні задач із використанням фреймової БЗ слотам фреймів попередньо (за замовчуванням) привласнюються значення (атрибути), які відповідають стандартним розв'язкам типових задач, отриманих у процесі експлуатації КСШ. Частина атрибутів визначається за допомогою вбудованих (внутрішніх) процедур. У процесі розв'язання конкретної задачі ці атрибути можуть бути замінені більш достовірними. Про деякі значення атрибутів система запитує користувача, уточнюючи, за потреби, їх значення. У процесі присвоєння слотам достовірних значень здійснюється виклик відповідних процедур.

У фреймових моделях значенням слоту може бути ім'я програми процедурного типу, яка називається *службовою* або *приєднаною процедурою*, що виконується при звертанні до слоту. *Приєднана процедура* може запускатися за повідомленням, переданим із іншого фрейму, тому її можна розглядати як один із механізмів керування процесом виводу на знаннях (розд. 7).

Процесом розв'язання задачі у фреймових структурах керують приховані (віртуальні) процедури, які мають назву *демонів*. Під час роботи системи вони «*сидять у засідці*» й активізуються тільки тоді, коли їх «*просять про допомогу*». Демон запускається при виконанні визначеної сукупності заздалегідь заданих умов. Наприклад, якщо при ідентифікації деякого об'єкта значення слоту не визначене, то демон задає процедуру, що дає змогу визначити або уточнити значення слоту.

Процес розв'язання задачі в КСШ із фреймовою БЗ полягає у зіставленні значень її слотів з атрибутами цілі, які містяться в запиті користувача. Це зіставлення здійснюється у такий спосіб (розд. 7.2).

1. Спочатку в БЗ знаходиться (вибирається) базовий (кореневий) фрейм, який описує найважливіший об'єкт даної Про, що міститься в запиті користувача.

2. Значення слотів базового фрейму зіставляються зі значеннями атрибутів цілі, тобто визначається релевантність базового фрейму запиту. Якщо фрейм підходить, то процес зіставлення закінчується і користувачу видається розв'язок задачі, сформоване шляхом відповідного опрацювання значень слотів знайденого базового фрейму.

3. Якщо знайдений фрейм не релевантний запиту користувача, то здійснюється пошук іншого базового фрейму з аналогічними значеннями слотів.

4. Якщо процес зіставлення закінчується безрезультатно, то задача не має розв'язку.

Семантично близькі фрейми поєднуються в окремі групи (*кластери*), що прискорює процес зіставлення слотів та скорочує час розв'язання задачі в цілому. Так, наприклад, у задачі опрацювання зображень при розпізнаванні образів (розд. 8.1) різні фрейми однієї групи описують його з різних кутів зору. Перехід від одного фрейму до іншого означає переміщення від одного пункту спостереження до іншого.

Перевагою фреймових моделей є їхня гнучкість, наочність і здатність відображення довгострокової пам'яті людини. До недоліків фреймів належить потреба у відносно великій комп'ютерній пам'яті БЗ та відсутність обґрунтованого рівня деталізації під час вибору числа атрибутів (ознак), що характеризують описуваний об'єкт. Рівень розуміння предмета завжди пов'язаний із рівнем його опису, тому і занадто загальний, і занадто докладний опис утруднюють проникнення в суть предмета.

6.5. Продукційні моделі подання знань

6.5.1. Продукції

Значного поширення в інтелектуальних системах набули *продукційні моделі подання знань* як евристичних *продукційних правил* (далі — *продукції*). *Продукція* являє собою упорядковану послідовність наступних операторів (рис. 6.3) [1.3, 3.5—3.25, 4.3, 4.15—4.18, 6.10—6.14, 6.18]:

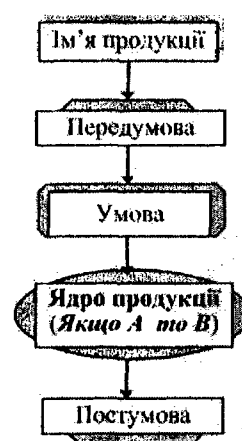


Рис. 6.3 — Структура продукції

1) *передумова* є умовою спрацьовування продукції і являє собою *предикат*, істинність якого визначає можливість виконання продукції;

2) *умова* визначає можливість або заборону використання виконання продукції і встановлює порядок (пріоритет) її застосування;

3) *ядро продукції* становить суть продукційного правила: *Якщо «А», то «В»*;

4) *постумова* визначає умову (рекомендацію) подальшого використання результатів виконання продукції.

Основною частиною продукції є її *ядро*, а інші оператори носять допоміжний характер. Тому *продукцію* звичайно подають у вигляді:

«Якщо (умова), то (дія)».

Тут умова являє собою *твердження*, яке міститься в запиті користувача і застосовується в процесі розв'язання його задачі (розд. 7).

Дія виконується за успішного результату пошуку правил, *релевантних* запиту. Дії можуть бути *проміжними*, що визначають подальший порядок розв'язання задачі, і *термінальними*, які завершують процес розв'язання. Дією може бути, наприклад, виконання наступної підзадачі (операції).

Частковим типом продукції є *імплікація* (розд. 6.2), яка описує процедуру логічного акта: «*посилання — висновок*». Іншими типами продукцій можуть бути:

- в кібернетичних системах: «*ситуація — дія*»;
- в медицині: «*симптом — діагноз*»;
- у кримінальному кодексі: «*диспозиція — санкція*»;
- в інших ПрО: «*попередній — наступний*», «*підстава — гіпотеза*» тощо.

Залежно від умовної ймовірності виконання розрізняють також *детерміновані* та *недетерміновані* продукції.

1. *Детерміновані* продукції можуть бути *однозначними* й *альтернативними*. В *однозначних* продукціях за умови виконання лівої частини «*A*» обов'язково виконується права частина «*B*». В *альтернативних* продукціях у правій частині ядра вказуються альтернативні можливості вибору, які оцінюються спеціальними *вагами вибору*. Такими вагами можуть бути ймовірні, лінгвістичні, експертні та інші оцінки.

2. У *недетермінованих* продукціях при виконанні лівої частини «*A*» не обов'язково виконується права частина «*B*» й інтерпретація ядра виглядає так: «*Якщо A, то можливо B*».

Можливість виконання твердження *B* у *недетермінованих* продукціях визначається певною оцінкою η , яка може бути:

- ймовірною: «*Якщо A, то (B виконається з ймовірністю η)*»;
- лінгвістичною: «*Якщо A, то (можливо B у діапазоні η)*»;
- експертною: «*Якщо A, то (можливо B з частотою появи η)*»;
- прогнозуючою: «*Якщо A, то (можливо B за умови η)*»;

Недетерміновані продукції можуть бути подані також у вигляді:

«*Якщо (A, α) то (B, β)*»,

де α і β — оцінки істинності тверджень, які застосовуються в такий спосіб:

- якщо α і β — *ймовірності*, то продукція визначає *ймовірний вивід*:

«*Якщо (A здійснено з ймовірністю β), то (B можливо з ймовірністю β)*»;

- якщо α і β — *ваги* тверджень, то здійснюється *наближений висновок*:

«*Якщо (A має вагу α), то (B можливо з вагою β)*»;

- якщо α і β — *евристичні оцінки*, то виконується *нечіткий вивід*:

«*Якщо (A має діапазон β), то (B можливо в діапазоні β)*».

6.5.2. Функціональні властивості продукційної бази знань

Продукційна БЗ являє собою ієрархічну структуру, яка складається зі структурованого набору *фактів* і *правил* (розд. 5.3). *Факт* у продукційній БЗ являє собою *твердження (предикат)*, що відображає відношення між двома або кількома об'єктами ПрО, наприклад (розд. 6.2):

«*Боб є сином Кості*»;

«*Костя є сином Олега*».

Продукційне правило являє собою *продукцію*, яка складається з кількох *тверджень (фактів)*, пов'язаних між собою логічними операторами «*і*», «*або*», «*не*», наприклад:

«*Якщо Боб є сином Кості*

і Костя є сином Олега,

то Боб є онуком Олега».

Продукційна модель типової ситуації, пов'язаної, наприклад, із діями людини під час посадки в автобус, може мати такий вигляд:

«*Якщо не (має гроші), то (пішки)*.

Якщо (має гроші) і не (прийшов автобус), то (чекати).

Якщо (прийшов автобус) і не (той маршрут), то (чекати).

Якщо (прийшов автобус) і (той маршрут), то (сідати в автобус)».

Висновки або *постумови* одних продукцій можуть бути *посиланням* для інших продукцій. У цьому випадку із двох таких продукцій, з'єднаних оператором «*і*», можна *вивести третю* продукцію з *посиланням* із першої продукції та *висновком* із другої, тобто:

«*Якщо X, то Y*» і «*Якщо Y, то Z*» «*Якщо X, то Z*».

Так утворюються складні ланцюжки (низки) правил *продукційної БЗ*, які використовуються при розв'язанні задач користувачів.

Продукційні моделі частіше за інші застосовуються в КСШІ у зв'язку з їхньою наочністю, модульністю, легкістю внесення доповнень і змін у БЗ, а також простотою *продукційного виведення*.

6.6. Подання інформації та пошук знань в Internet

Основою *Internet* є розподілена інформаційна мультимедійна система **World Wide Web (WWW)** (далі — *Web*), що являє собою мережу комп'ютерних файлів, з'єднаних між собою асоціативними гіпертекстовими зв'язками (*гіперзв'язками*). У файлах (вузлах) цієї мережі знаходяться *Web-документи*, які містять текстову, графічну, відео, аудіо та інші види інформації. Гіперзв'язки мають асоціативний характер, тому пошук документів у *Web-мережі* здійснюється за їхнім змістом. Гіперзв'язок виконується за допомогою *ключових слів (фраз)*, що містяться в документах, які знаходяться в різних файлах системи *Web*. Механізм пошуку інформації у цій системі спочатку звертається до бази *ключових слів*, а потім на їхній основі обробляє відповідні документи.

Для роботи з документами в системі *Web* застосовуються такі засоби (*механізми*) пошуку, передачі й подання інформації [6.17, 6.19, 6.20].

1. **Ідентифікація інформаційних ресурсів (IP)**, відповідно до якої кожен IP має своє унікальне ім'я й адресу (*URL — Uniform Recourse Locator*), за якою система знаходить IP у разі запиту користувача. За допомогою цих імен можна зв'язувати IP відповідними *гіперзв'язками* і робити *гіперпосилання* на інформацію, що зберігається в різних файлах системи *Web*.

2. **Протокол HTTP (Hyper Text Transfer Protokol)** — програмний засіб, який здійснює доступ до IP, а також одержання й відправлення IP через *Internet*.

3. Ефективний механізм навігації по документах *Web* з використанням засобів *гіпермедіа* й *гіперпосилань*, за допомогою яких користувач може швидко знаходити потрібні документи.

4. Програмні засоби *контекстної допомоги* допомагають користувачу описати *розв'язувану задачу (ситуацію)* природною мовою, а система конкретизує її, задаючи користувачу додаткові запитання в діалоговому режимі. Потім система виконує пошук IP, що належать до даної задачі.

5. Спеціальні мови подання інформації **HTML (Hyper Text Markup Language)** і **XML (Extensible Markup Language)**, які містять ефективні засоби опису, публікації та поширення різних типів документів.

6. Засоби подання графічної інформації (JPEG, GIF), аудіо (MP3), мультимедіа (SMIL, Flash, MPEG) та ін.

Розмітка *HTML-документів* у системі *Web* здійснюється за допомогою спеціальних конструктивів, які називають *тегами*

(*tags*). Розрізняють *теги відкриття* й *закриття* фрагмента тексту, що розмічається. Загальна структура *HTML-документа* може бути подана у такому форматі:

```
<!DOCTYPE HTML PUBLIC (Специфікація версії мови HTML) >  
<HTML> <HEAD> <TITLE> (Заголовок документа) </TITLE>  
<META> (Анотація документа із фіксацією його атрибутів)  
</HEAD>  
<BODY> (Текст документа) </BODY> </HTML>
```

Як було зазначено в розд. 4.6, у складі системи *Internet* сьогодні знаходяться декілька екзабайт «сирих» неструктурованих інформаційних документів, які містять *глибинний (прихований) шар знань*. Відшукати їх за допомогою традиційних методів здобуття знань часто не вдається і тому дуже цінна для користувачів КСШ інформація залишається невикористаною.

З огляду на це, останнім часом велика увага приділяється розробленню *нових методів пошуку й опрацювання знань в Internet*. Сьогодні виник новий напрямок в опрацюванні текстової інформації, основою якого є *технологія здобування знань Data Mining*, що призначена для виявлення *глибинних знань у документах* системи *Web* (розд. 4.6).

Розділ 7

СТВОРЕННЯ НОВИХ ЗНАНЬ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ

7.1. Механізм створення

Як було зазначено в попередніх розділах, КСШ моделює процес мислення людини, зокрема, логіку його міркувань (розд. 5.1). У спрощеній формі логічні міркування людини зводяться до того, щоб із одного чи кількох *посилань* (тобто на підставі наявних знань) зробити логічно правильний *висновок* (тобто одержати нове знання). Одержання людиною нових знань із наявних знань, отриманих на основі досвіду розв'язання життєвих задач, називають *механізмом створення знань*. Основою цього механізму є так званий принцип *modus ponens*, згідно з яким: «Якщо є істинне твердження *A* й існує правило виду «Якщо *A*, то *B*», то твердження *B* теж істинно». Інакше кажучи, якщо істинне *посилання* (*анцедент*), то повинен бути істинним й *висновок* (*консеквент*) (розд. 5.1) [1.1–1.16].

Нехай, наприклад, *Борис* і *Оксана* є батьками *Максима* й *Ольги* і треба визначити, *ким* діти доводяться один одному. Цю задачу людина розв'язує на основі *правила* (*продукції*), яке зберігається в її пам'яті: «Якщо у дітей одні й ті самі батьки, то діти є братами й сестрами». Таким чином, розв'язуючи цю задачу, людина одержує нове знання: «*Максим* і *Ольга* — брат і сестра».

За аналогією з людським мисленням для розв'язання задач у КСШ теж використовуються *інтелектуальні комп'ютерні засоби створення знань*, які здійснюють *цілеспрямований перебір знань* у БЗ для одержання нових знань [2.1–2.41, 3.1–3.25, 5.1–5.24, 6.1–6.20, 7.1–7.28].

Нові знання можуть бути отримані в КСШ різними способами. Можна одержати їх унаслідок логічного аналізу наявних знань, або вивести загальне правило з наявних фактів, чи застосувати ті самі факти і правила для схожих (асоціативних) об'єктів. Нині є такі різновиди методів створення нових знань:

- *дедуктивний метод* передбачає логічний перехід у міркуваннях від загального правила до часткового (одиночного, окремого чи менш загального) факту, на який можна поширити це правило;

- *індуктивний метод* передбачає логічний перехід від часткових (одиночних або окремих) фактів до загального правила, що поширюється на дані факти;

- *створення нових знань за аналогією* (*традукція*) здійснює перехід від часткових фактів (правил) до інших часткових фактів (правил); у цьому разі посилення й висновок повинні бути судженнями (твердженнями) одного і того самого рівня спільності знань.

Слід зазначити, що механізм створення нових знань у людини залишається поки ще недостатньо вивченим. Людським міркуванням властиві неформальність, нечіткість, нелогічність, використання образів, емоцій і почуттів, що надзвичайно ускладнює дослідження й моделювання процесу створення нових знань.

7.2. Мета й локальні цілі

Метою називають кінцевий результат, на який спрямовані розумові процеси людини. Метою може бути, наприклад, вчасне потрапляння на роботу. Але перед цим треба вмитися, одягтися, поспіяти, сісти в машину, не потрапити у «пробку», доїхати до роботи. Здійснення усіх цих *підцілей* приводить до досягнення *основної мети*. Як тільки чергова мета досягнута, перед людиною відразу постають нові цілі. Кожен крок на шляху до *головної мети* має свою *локальну мету*. Мозок завжди зосереджений на меті незалежно від того, чи виконує людина просту фізичну роботу, чи вирішує складну інтелектуальну задачу. Мета змушує людину *мислити*.

За аналогією з людиною, пошук розв'язання поставленої задачі в КСШ здійснюється в інтелектуальному процесі *створення знань*. Сутність цього процесу полягає в тому, щоб на підставі аналізу наявних фактів і правил БЗ одержати (*вивести*) так зване *цільове твердження* або спростувати його. *Цільове твердження* або заздалегідь закладається у БЗ, або формується системою в процесі діалогу з користувачем. Загальна *мета* пошуку, що міститься в *запиті користувача*, розгортається в *підцілі*, а поставлена задача розбивається на *підзадачі* доти, поки не будуть отримані елементарні задачі, рішення яких системі заздалегідь відомі.

Розглянемо, наприклад, *задачу* щодо *проїзду на автомобілі зі Львова в Київ*. Ця *задача* може бути зведена, наприклад, до таких трьох *підзадач* проїзду:

- 1) зі *Львова* у *Рівне*;
- 2) із *Рівного* у *Житомир*;
- 3) із *Житомира* в *Київ*.

Кожну із цих підзадач, у свою чергу, можна проаналізувати й розбити на ще дрібніші підзадачі. У результаті комплексного вирішення цих елементарних підзадач можна одержати розв'язок загальної поставленої задачі.

7.3. Механізм вибору знань

Людський мозок зберігає велике число *фактів* і *правил* їх використання. Коли людина приступає до вирішення навіть найпростішої задачі, то, вибираючи потрібні дії, вона опрацьовує великий обсяг інформації. Наприклад, йдучи на роботу, людина виходить із будинку і прямує до перехрестя. Перш ніж перетнути вулицю, людина аналізує інтенсивність й швидкість руху транспорту, відстань до протилежного тротуару, сигнали світлофора на перехресті. Водночас її мозок опрацьовує враження, які не стосуються переходу вулиці, наприклад, погодні умови, моделі проїжджаючих машин, порода й висота дерев, що ростуть обабіч дороги, вид розташованих неподалік будинків. Людина також думає про місце, куди вона йде, про те, як швидко їй треба там бути, кого вона може зустріти тощо.

Однак, якби людина, перш ніж стати на проїжджу частину, проаналізувала б усі факти і правила, що вміщує її мозок (але які не стосуються її бажання перейти вулицю), то вона простояла б на тротуарі багато часу. Яким же чином людський мозок із величезного різноманіття своїх знань швидко вибирає фрагменти, що підходять тільки до конкретної ситуації? У мозку людини є складна система, яка керує *вибором* правильної реакції на конкретну ситуацію. *Механізм вибору* блокує думки, які не стосуються розв'язуваної в цей момент задачі. Коли людина стикається з якоюсь ситуацією, *механізм вибору* змушує її мозок зосередитися тільки на тих знаннях, які необхідні для досягнення поставленої мети. Для людини факти і правила, що належать до задач, пов'язаних із *роботою*, зовсім не стосуються *поїздки* на роботу.

Аналогічно людському мисленню *механізм створення знань із БЗ* дає змогу вирішувачу КСШ опрацьовувати тільки ту частину інформації, яка необхідна для вирішення конкретно поставленої задачі. Механізм вибору просто ігнорує непотрібну інформацію, що не стосується досягнення поставленої мети.

У зв'язку з цим основною проблемою під час роботи з великою БЗ є проблема *пошуку знань, релевантних (відповідних) розв'язуваний задачі*. У запиті можуть бути відсутні явні вказівки (посилання) на знання, необхідні для її вирішення. Тому необхідний спеціаль-

ний механізм вибору знань із БЗ, який за описами *сутностей (ключових слів)*, що містяться в запиті користувача (тобто в поставленій задачі), зможе знайти у БЗ відповідні знання. Тому *упорядкування (структурування)* знань у БЗ значно прискорює розв'язок поставленої задачі (розд. 5.1).

7.4. Рівні інтелектуальності розв'язуваних задач

Задачі, що розв'язуються в КСШ, мають різні *рівні інтелектуальності*. У зв'язку з цим система повинна володіти відповідними *рівнями розуміння* тих задач, які вона вирішує. Розглянемо їх у порядку *зростання ступеня розуміння*.

1. На *першому (нижньому) рівні* система формує відповіді на поставлені запитання (*запити*) на основі *прямого методу створення* наявних у системі знань. Нехай, наприклад, у БЗ системи закладені такі знання:

- «О восьмій ранку, після сніданку, Макс їде в офіс».
- О сьомій годині вечора він повертається додому».
- Після вечері він відвідує казино».

Тоді на першому рівні розуміння система зобов'язана вміти відповідати правильно на запитання типу: «Коли Макс поїде в офіс?» чи «Що робить Макс після вечері?»

На першому рівні розуміння КСШ здійснює *морфологічний, синтаксичний та семантичний аналіз* запитань і відповідних знань та переводить їх у *внутрішнє подання* в системі.

2. На *другому рівні* до функцій розуміння системи додаються засоби *логічного виводу* інформації, що міститься у знаннях. Такими засобами є *просторово-тимчасові* логічні функції, здатні породити інформацію, що явно не присутня у знаннях. Для попереднього прикладу на другому рівні розуміння можливе формування відповідей на запитання типу: «Що буває раніше: поїздка Максима в офіс чи його вечора?» Тільки побудувавши *часову* структуру знань, система зможе відповісти на подібні запитання.

Для реалізації другого рівня розуміння у БЗ організується *додаткова область знань* (розд. 5.4), де зберігаються закономірності, що належать до *часової* структури подій та їхньої *просторової* організації.

3. На *третьому рівні* до засобів другого рівня додаються знання системи із *проблемної області* у вигляді сценаріїв або *процедур*. На цьому рівні система повинна давати відповіді на запитання типу: «Де буває Макс до восьмої ранку?» або «Звідки Макс повертається додому?» Для цього у БЗ системи повинні бути *сценарії*, які описують процеси «*перебування вдома*», «*перебування в офісі*» та «*перебування в казино*». Таким чином, у механізмі створення нових знань

системи, крім логічного методу створення, повинні бути передбачені засоби *створення нових знань за сценаріями*.

4. На *четвертому рівні* система, поряд з опрацюванням інформації, що знаходиться в БЗ, має здатність до розуміння й опрацювання *додаткових знань*, які надходять у систему по *додаткових каналах* інформації і безпосередньо пов'язані з тією ситуацією, в якій вони породжуються. Таким додатковим каналом інформації може бути, наприклад, *зір*, яким володіють інтелектуальні *робототехнічні* системи (розд. 11).

На *четвертому* рівні розуміння система здатна відповідати на запитання типу: «*Чому Макс відвідує казино?*» або «*Що робить Макс в офісі?*»

5. На *п'ятому рівні* в системі використовуються засоби *діалогового спілкування* з користувачами, експертами та іншими *мовними джерелами* інформації. При розв'язанні задач цього рівня в системі використовуються процедури так званих *мовних актів*, кожний із яких являє собою одиницю *мовного виразу*, який поєднує в собі три дії:

- *локуцію* — висловлення суб'єктом своєї думки;
- *ілокуцію* — дію, що здійснюється за допомогою мовлення (твердження, прохання, наказ);
- *перлокуцію* — додатковий вплив на слухача, щоб умовити, здивувати, звабити тощо.

На *п'ятому рівні* в системі можуть бути використані правила створення, що спираються на знання про даного конкретного суб'єкта спілкування. Наприклад, система може не довіряти користувачу і корегувати знання, що вводяться у БЗ, відповідно до своїх знань про даного суб'єкта й інших користувачів, із якими вона раніше спілкувалася.

Рівень розуміння (інтелектуальності) КСШІ можна оцінювати також її здатністю до *пояснення* отриманого результату розв'язання задачі і його *обґрунтування* (*аргументації*), коли система показує, що результат не суперечить тій сукупності знань, якими вона володіє.

7.5. Машина створення нових знань

Основною програмно-апаратною частиною *вирішувача* КСШІ (розд. 3.3) є *машина створення* (*інтерпретатор правил, блок логічного створення нових знань, логічний блок, логічний планувальник, дедуктивна машина* тощо), призначена для здійснення *процесу створення знань*. В основу функціонування *машини створення* покладено *механізм створення нових знань*, який установлює *порядок вибирання правил* із БЗ та їх подальшого *застосування* (розд. 7.1). Нижче розглянуто принцип роботи й основні функції *машини створення нових знань* (рис. 7.1) [3.1–3.25, 7.1–7.58].

7.5.1. Формування робочої множини фактів і правил

Коли користувач вводить у систему *запит*, *механізм створення нових знань* вибирає відповідний спосіб застосування правил БЗ для розв'язання задачі, що міститься в запиті, й запускає в роботу *машину створення*.

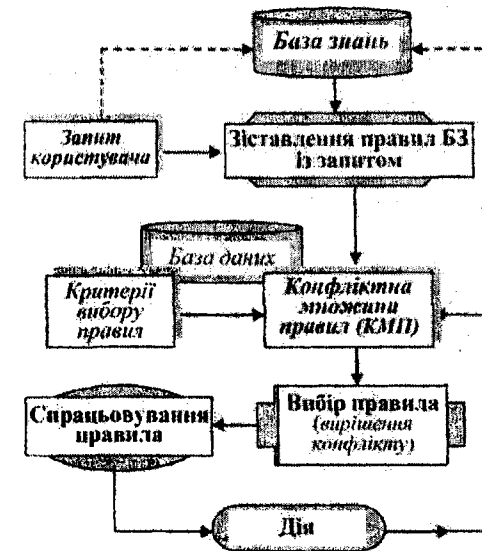


Рис. 7.1 — Схема механізму створення нових знань

Одержавши задачу, машина створення *вибирає* з БЗ і заносить у *робочу пам'ять* вирішувача (базу даних) тільки ті *знання*, які можуть знадобитися при розв'язанні цієї задачі. Для цього машина створення здійснює *аналіз* правил БЗ з метою вибору (*активізації*) тих із них, що відповідають *цілі* і *підцілям* розв'язуваної задачі (розд. 7.1). У результаті аналізу в *робочій пам'яті* вирішувача буде сформована *робоча множина фактів і правил*, необхідна для вирішення поставленої задачі.

Слід зазначити, що машина створення, зазвичай, заносить відповідні факти і правила в робочу пам'ять не одразу, а в процесі розв'язання задачі в діалоговій взаємодії з користувачем.

7.5.2. Зіставлення фактів і правил

Машина створення працює *циклічно*. Основною операцією, що виконується в кожному циклі, є послідовне *зіставлення робочої множини фактів і правил*, сформованих у робочій пам'яті вирішувача, із фактами й правилами, що містяться в запиті користувача. Машина створення задає порядок перегляду альтернативних фактів і правил, розташовуючи їх у певному порядку. Вона бере чергову *підциль* і намагається довести її істинність. Проглядаються всі факти й правила робочої множини, які зіставляються з підциллю. При збігу відповідних тверджень істинність підцилі вважається доведеною і береться наступна підциль.

У процесі зіставлення правил можуть з'явитися нові факти, які машина створення нових знань вносить у *робочу множину фактів*. Наприклад, із фактів «*Небо покрите хмарами*» і «*Барометр падає*», які вже знаходяться в робочій множині, система може вивести новий факт «*Незабаром піде дощ*».

7.5.3. Вирішення конфліктних ситуацій

Машина створення нових знань знаходить усі можливі варіанти розв'язків поставленої задачі. Тому для реалізації (доказу істинності) окремої підцилі вона може вибрати кілька правил, що утворюють так звану *конфліктну множину правил (КМП)*. Оскільки в одному циклі пошуку рішення може спрацювати тільки одне правило, процедуру *вирішення конфлікту (фільтраж)* машина створення виконує, використовуючи відповідні *критерії відбору*, що знаходяться у базі даних системи. Машина створення нових знань вибирає з КМП для реалізації *єдине правило*, найпридатніше для розв'язання поставленої задачі.

У процесі вирішення конфлікту на кожному циклі виконуються такі дії.

1. Із КМП виключаються ті правила, що вже застосовувалися в попередніх циклах процесу вибору правил. Якщо після цього КМП стало порожнім, то цикл завершується.

2. Оскільки в процесі розв'язання задачі в робочій пам'яті з'являються нові правила, то в КМП залишаються «найсвіжіші» елементи, а інші з КМП видаляються. Якщо нове правило є єдиним, то воно і вибирається для застосування, після чого процес вибору в цьому циклі завершується.

3. Якщо в КМП є кілька домінуючих правил із рівним пріоритетом, то перевага надається тому правилу, яке потребує перевіряння найбільшого числа елементів робочої пам'яті.

4. Якщо після здійснення попередніх кроків у КМП усе ж залишилося кілька правил, то до реалізації вибирається будь-яке з них і процес вибору припиняється.

Таким чином, процедура вирішення конфліктів містить аналіз таких показників, як *повторюваність, новизна і специфіка* правил у КМП, які використовуються при розв'язанні поставленої задачі.

Стратегія вирішення конфліктів суттєво впливає на продуктивність роботи КСШ. Тому при виробленні цієї стратегії зазвичай використовують комбінацію різних базових механізмів вирішення конфліктів.

7.5.4. Спрацювання та дія правил

На черговому циклі здійснюється *спрацювання* правила, обраного механізмом створення з конфліктної множини. *Висновок* правила, що спрацювало, запам'ятовується в *робочій пам'яті* вирішувача, а при *одержанні нового знання* заноситься у БЗ системи.

У результаті *спрацювання* здійснюється визначена *дія* виконання певної процедури, перевірка отриманого результату поставленої задачі, зміна критерію вибору конфлікуючих правил, подача звукового сигналу, ініціювання нових дій користувача тощо.

У разі потреби машина створення може здійснити *повернення (back tracking)* до попередньої ситуації, щоб застосувати інше правило пошуку розв'язку поставленої задачі.

Машина створення має здатність функціонувати й знаходити рішення навіть за нестачі вхідної інформації. Якщо, наприклад, для спрацювання чергового правила «*Якщо небо вкрите хмарами і падає барометр, то можливий дощ*» у робочій множині не виявляється факту «*Небо вкрите хмарами*», то машина створення має можливість спитати про це користувача. У разі позитивної відповіді цей факт заноситься в робочу пам'ять вирішувача.

7.6. Стратегії керування створенням нових знань

Одним з основних питань функціонування КСШ є вибір *стратегії керування створенням знань*, яка визначає напрям пошуку розв'язку задачі, спосіб його здійснення, а також порядок застосування та спрацювання правил.

У КСШ застосовуються такі *стратегії керування створенням нових знань* [3.1–3.25, 7.1–7.58]:

а) *прямий метод створення нових знань (від знань до мети)* керується знаннями, коли на основі відомих фактів відшукується розв'язок задачі;

б) *зворотний метод створення нових знань (від мети до знань)* керується метою, коли відшукуються факти, що підтверджують досягнення поставленої мети;

в) *циклічний метод* поєднує в собі *прямий* і *зворотний* методи створення нових знань.

Розглянемо зазначені стратегії більш докладно.

7.6.1. Пряма (висхідна) стратегія створення нових знань

Прямий метод створення знань (стратегія «знизу нагору») полягає в аналізі відомих фактів і послідовному застосуванні до них правил, що породжують нові факти. Цей процес нагромадження нових знань

триває доти, поки не буде створено факт, який визначає мету, тобто *розв'язання поставленої задачі*.

Приклад 7.1. Нехай, наприклад, БЗ містить:

- правило 1:
- Якщо «Небо вкрите хмарами» і «Барометр падає», то «Буде дощ»;
- правило 2:
- Якщо «Буде дощ», то «Треба взяти парасольку».

Припустимо також, що в *робочу пам'ять вирішувача* надійшла інформація (*робоча множина фактів*):

- факт 1: «Небо вкрите хмарами»;
- факт 2: «Барометр падає».

Нехай задачею системи є відповідь на *запит користувача*: «Треба взяти парасольку?»

Алгоритм *прямого методу створення нових знань* у цьому випадку полягає в наступному.

1. Розглядається *правило 1*. Воно *істинно*, тому що обидві його ліві частини присутні в робочій множині. Застосовуючи *правило 1*, додаємо *факт «Буде дощ»* у робочу множину.

2. Розглядається *правило 2*. Воно *істинно*, оскільки його ліва частина знаходиться в робочій множині. Застосовуючи *правило 2*, додаємо до робочої множини факт «Треба взяти парасольку». Цільове твердження виведене і задача вирішена.

У цьому разі припускається, що обидва факти заздалегідь введені в робочу пам'ять. Насправді машина створення нових знань може з'ясовувати істинність факту, що входить до складу опрацьовуваного правила, запитуючи про це користувача тоді, коли вона намагається застосувати це правило.

Таким чином, алгоритм *прямого методу створення нових знань* містить у собі:

- постановку задачі (визначення мети);
- визначення фактів БЗ, що стосуються поставленої мети;
- одержання нових знань, що відповідають фактам цієї задачі;
- одержання цільового результату шляхом створення нових знань.

Прямий метод створення нових знань, тобто процес переходу від знань до цільового результату, називають також *керуванням знаннями*.

7.6.2. Зворотна (спадна) стратегія створення нових знань

Зворотний (спадний) метод створення нових знань (від цілі до фактів) полягає в організації пошуку й витягу з БЗ фактів і правил, які підтверджують істинність *цільового твердження*. Це твердження називають *цільовою гіпотезою* (їх може бути кілька), яку формує

вирішувач на основі вхідної інформації, що міститься в запиті користувача. Механізм створення вибирає з БЗ факти і правила й за допомогою *зіставлення* знаходить серед них ті, що підтверджують *цільову гіпотезу*. Якщо вона виявилася істинною, то вибирається наступна *гіпотеза-підциль*, що деталізує цільову гіпотезу, та відшукуються факти, що підтверджують істинність підлеглої гіпотези. Процес створення нових знань триває до повного вичерпання гіпотез.

Зворотний метод називають також *керуванням цілями*. Він застосовується в тих випадках, коли цілі відомі та їх порівняно небагато.

Приклад 7.5. Істинність цільового твердження (*цілі*) «Треба взяти парасольку» у прикладі 7.1 на основі *зворотного методу створення нових знань* здійснюється в такий спосіб.

1. Розглядається *правило 1*. Оскільки воно не містить цілі у правій частині, то переходимо до *правила 2*.

2. Розглядається *правило 2*. Оскільки воно містить ціль у правій частині, то як *поточна* (проміжна) *підциль* приймається твердження «Буде дощ».

3. *Поточної підцилі* немає в робочій множині. Тому розглядається *правило 1*, що містить підциль у правій частині. Обидві його ліві частини знаходяться в робочій множині, так що це твердження істинне. Застосовуємо *правило 1* і виводимо твердження «Буде дощ», яке було попередньою *поточною підциллю*.

4. Застосовуючи *правило 2*, у лівій частині якого міститься твердження «Буде дощ», одержимо цільове твердження «Треба взяти парасольку».

Таким чином, алгоритм *зворотного методу створення нових знань* полягає в наступному:

1. У робочу пам'ять системи вводиться список *цільових гіпотез* (тверджень), істинність яких треба довести.

2. Машина створення виокремлює у БЗ список правил, *праві частини* яких містять *цільові гіпотези*.

3. На основі *лівих частин* виділених правил формуються *гіпотези-підцилі*.

4. Шляхом зіставлення правил БЗ зі знайденими в п. 3 *гіпотезами* вилучаються з розгляду *хибні підцилі*.

5. Список *цільових гіпотез* доповнюється *істинними підцилями*, які вирішувач одержує від користувача.

6. Правила БЗ зіставляються із *цільовими гіпотезами* і визначаються факти, що підтверджують їх *істинність*.

7. Визначається *істинність цільових гіпотез* через зіставлення їх із правилами БЗ і фактами (твердженнями), отриманими в п. 7.

8. Якщо не всі *цілі* досягнуті (тобто не всі *цільові гіпотези є істинними*), то здійснюється *перехід до п. 2 з вилученням із розгляду досягнутих підцілей*.

9. Після досягнення всіх поставлених цілей машина створення припиняє роботу і надає користувачу список фактів, які визначають розв'язок його задачі. Процедура створення припиняється також, якщо не можна досягти поставлених цілей.

Розглянуті вище алгоритми можуть створити враження, що *прямий* метод створення нових знань простіший та ефективніший, ніж *зворотний*, але це не так. Ефективність тієї чи іншої стратегії створення залежить від характеру задачі й умісту БЗ. Приміром, у системах діагностики частіше застосовується *прямий* метод створення нових знань, тоді як у системах планування ефективнішим виявляється *зворотний* метод.

Для мінімізації часу вирішення задач при використанні розглянутих вище *стратегій керування створенням нових знань* нині розроблені ефективні *способи пошуку і перегляду правил* у структурованій БЗ, основними з яких є:

а) пошук «у глибину», при якому пошук знань у БЗ здійснюється шляхом переходу від загального (*верхнього рівня БЗ*) до детального опису (до нижніх *рівнів БЗ*) розв'язуваної задачі з наступним *поверненням* на верхній рівень у кожному циклі пошуку;

б) пошук «у ширину» являє собою паралельний пошук необхідної інформації на окремому рівні знань БЗ із наступним переходом на інший рівень;

в) *розбивка області БЗ розв'язуваної задачі на підобласті* задля досягнення проміжних цілей на шляху до кінцевої мети;

г) *відсікання* (видалення) областей пошуку, *безперспективних* для успішного розв'язання задачі.

Докладніше зазначені способи пошуку і перегляду наявних знань у БЗ наведені в *розділі 8.5.3* при пошуку *розв'язку задач на графах*.

Слід зазначити, що процес моделювання мислення людини досить складний і може не вкладатися в межі тієї чи іншої стратегії керування виводом на знаннях. Тому для вирішення задач у КСШ часто використовується *комбінований (циклічний) алгоритм* створення нових знань шляхом багаторазового *переключення* з одного режиму керування на інший, що є характерною рисою людського мислення.

Розділ 8

МЕТОДИ РОЗВ'ЯЗАННЯ ІНТЕЛЕКТУАЛЬНИХ ЗАДАЧ

8.1. Метод резолюцій

В основу сучасних *логічних* методів розв'язання інтелектуальних задач покладені формальні конструкції *дедуктивної логіки* з використанням *силогізмів Арістотеля й числення предикатів* першого порядку (розд. 6.2). Використовуючи дедуктивну логіку, із двох чи кількох наявних аксіом можна *вивести* чергове *твердження-наслідок*. Наприклад, на підставі двох тверджень: «Усі люди смертні» і «Арістотель — людина» можна зробити висновок: «Арістотель смертний». Процес одержання *нових тверджень* із наявних знань за допомогою певного набору *правил виводу* називають *логічним виводом на знаннях* (розд. 7).

Основним методом розв'язання інтелектуальних задач із використанням *логічного створення нових знань* є *метод резолюцій* [2.1–2.11, 2.15–2.19, 3.5–3.11, 6.2–6.4, 6.16]. Відповідно до цього методу, із двох наявних у БЗ аксіом (тверджень) можна *вивести нове твердження*, що називається *резольвентою (резолюцією)*. Типовою задачею, що розв'язується в КСШ на підставі *методу резолюцій*, є задача *автоматичного доказу теорем*, у якій необхідно встановити, чи є *теоремою* дане твердження, тобто: чи можна встановити його *істинність*, виходячи з наявних аксіом БЗ.

Доказ теорем із використанням *методу резолюцій* здійснюється у такому порядку.

1. Теорема доводиться *від протилежного*, тобто в робочу пам'ять вирішувача вноситься *заперечення* того твердження, яке потрібно довести. Наприклад, якщо потрібно довести *теорему (факт) P(a)*, то до бази знань вводять *твердження* $\sim P(a)$ (розд. 6.2).

2. Перебираються аксіоми (*фрази Хорна*) робочої пам'яті вирішувача, що відповідають розв'язуваній задачі (запиту користувача) (розд. 7.2). Якщо один і той самий *предикат (літерал)* міститься у складі однієї аксіоми і є *запереченням* в іншій, то ці аксіоми є *резольвентними (резолюючими)*. Унаслідок *об'єднання* цих аксіом і *вилучення* дублюючих (різнознакових) предикатів утвориться *резольвента*. Як приклад, розглянемо дві аксіоми:

$$P(x) \vee \sim Q(y, z),$$

$$Q(y, z) \vee \sim R(y, z),$$

перша з яких містить у *правій* частині *негативний* літерал $\sim Q(y, z)$, а в лівій частині *другої* аксіоми присутній той самий *позитивний* літерал $Q(y, z)$. Ці аксіоми є *резольвентними* і *резольвента* записується у вигляді: $P(x) \sim R(y, z)$.

3. *Перемінна* може бути замінена константою або іншим *термом*. Наприклад, дві аксіоми

$$P(x) \vee \sim Q(a, b),$$

$$Q(x, y) \vee \sim R(x, y)$$

є *резольвентними*. У цьому разі *перемінна* x *зіставляється* з константою a , *перемінна* y — із константою b і *резольвента* записується у вигляді: $P(a) \sim R(a, b)$. У ній *перемінні* замінені константами.

4. Аксіома-факт може *резольвувати* з будь-яким предикатом аксіоми-правила за схемою, наведеною в п. 2. Наприклад, із факту $Q(a, b)$ і правила $P(x, y) \sim Q(x, y)$ утвориться *резольвента-факт* $P(a, b)$.

5. *Різні константи* не *зіставляються* одна з одною, тому аксіоми

$$P(a) \sim Q(a, b) \text{ і } Q(a, c) \sim R(b, c) \text{ не є резольвентними.}$$

6. Якщо в процесі доказу теореми будуть одержані будь-які дві *суперечливі аксіоми* типу $P(a) \text{ і } \sim P(a)$, то їх *резольвентою* є *порожня фраза-суперечність*, яка означає, що припущення про хибність (*заперечення*) теореми невірне. Тому *теорема* є *істинною* і процес доказу на цьому закінчується.

Необхідно зазначити, що всі *резольвентні аксіоми* залишаються у БЗ без змін, а всі *резольвенти*, отримані в процесі доказу теореми, аналізуються редактором БЗ на *несуперечність* і *вносяться* у БЗ, утворюючи *нові знання* про певну ПрО.

Розглянемо приклад використання *методу резолюцій* у ПрО «Батьки і діти», наведеної у розділі 6.2.3. Нехай у цьому прикладі треба довести *істинність твердження*:

$$\text{«Боб є нащадком Олега»}.$$

З урахуванням згаданих у розділі 6.2.3 *констант і предикатів* цю задачу можна сформулювати у такий спосіб: *довести теорему*

$$P(B, O) \quad (8.1)$$

на підставі таких знань (на знаннях) *логічної БЗ*:

$$\begin{aligned} &S(B, K) \quad S(K, O); \\ &\sim S(x, y) \vee P(x, y); \\ &\sim S(y, z) \vee P(y, z); \\ &\sim P(x, y) \vee \sim P(y, z) \vee P(x, z). \end{aligned}$$

Розв'язання цієї задачі здійснюється в такій послідовності.

1. Відповідно до *методу резолюцій* додамо до аксіом БЗ *заперечення теореми* (8.1):

$$P(B, O); \quad (8.2)$$

$$S(B, K); \quad (8.3)$$

$$S(K, O); \quad (8.4)$$

$$S(x, y) \vee P(x, y); \quad (8.5)$$

$$S(y, z) \vee P(y, z); \quad (8.6)$$

$$P(x, y) \vee \sim P(y, z) \vee P(x, z). \quad (8.7)$$

2. Застосовуючи *резолюцію* до тверджень (8.5) і (8.7), одержимо *резольвенту*:

$$S(x, y) \vee \sim P(y, z) \vee P(x, z). \quad (8.8)$$

3. *Зіставляючи* у твердженнях (8.2) і (8.8) *перемінні* x з B і z з O , одержимо:

$$S(B, y) \vee \sim P(y, O). \quad (8.9)$$

4. Із (8.3) і (8.9) виходить:

$$P(K, O). \quad (8.10)$$

5. Із (8.4) і (8.6) одержимо:

$$P(K, O). \quad (8.11)$$

6. Застосовуючи *резолюцію* до тверджень (8.10) і (8.11), одержимо *суперечність*.

Це означає, що *теорема* (8.1) є *істинною*, це і є розв'язком цієї задачі.

8.2. Методи розпізнавання образів

Одним із основних типів інтелектуальних задач є *розпізнавання образів* (розд. 3.6). *Образами* можуть бути *об'єкти* (*процеси, явища, ситуації* тощо) певної ПрО, а також *зображення* (*плоскі й об'ємні*), *сигнали* (*мовні, електричні, оптичні*), *сценарії* тощо. Найбільш розповсюдженими на сьогодні є такі *задачі розпізнавання образів* [2.1–2.41, 3.1–3.25, 8.1–8.15].

8.2.1. Ідентифікація

Задача ідентифікації може бути подана в таких варіантах.

1. БЗ КСШ містить множину *образів* певної ПрО. Кожний образ характеризується набором *атрибутів* (*ознак, властивостей* тощо).

Задача КСШІ полягає у знаходженні певного образу серед множини подібних за окремими його атрибутами, які присутні, наприклад, у запиті користувача.

2. У БЗ КСШІ немає шуканого образу, що присутній у запиті користувача. Потрібно знайти такі образи БЗ, які за сукупністю задовольняють атрибути образу запиту. Така задача характерна, наприклад, для діагностики захворювання (або несправності), де симптоми хвороби (явища) часто пояснюються не однією хворобою (несправністю), а цілим їх «букетом».

3. Із деякої множини образів з відомим пріоритетом атрибутів необхідно знайти найбільш підходящий образ із погляду заданого критерію оптимальності (порівняння). Наприклад, потрібно вибрати придатні ліки для лікування конкретної хвороби, виходячи з вартості, дефіцитності й ефективності наявних лікарських препаратів.

4. На вхід КСШІ подаються рукописне зображення тексту, написаного різними почерками. Система повинна розпізнати букви цього тексту і створити текстовий документ, із яким зможе далі працювати текстовий редактор БЗ.

Одним із основних методів ідентифікації є зіставлення розпізнаного образу з еталоном, сутність якого полягає в наступному. БЗ КСШІ має еталонний набір образів (зразків). Системі задається довільний неідеальний образ і вона повинна його розпізнати й видати на виході зображення зразка, що відповідає вхідному образу. Якщо такого зразка немає в БЗ, то система повинна про це повідомити.

Для розв'язання задач розпізнавання образів розроблений спеціальний метод структурного опису зображень, у якому зображення (образи) замінюються набором ознак, пов'язаних між собою певними співвідношеннями (див. нижче).

8.2.2. Класифікація

Завдання класифікації полягає у віднесенні розпізнаних образів до тих чи інших класів. Образи кожного класу згруповані за певними формальними ознаками або правилами.

Задача класифікації містить у собі:

- аналіз вхідних об'єктів (образів);
- опис образів у вихідному просторі ознак;
- установлення інформативних ознак образів, достатніх для їх розпізнавання;
- перехід від вихідного простору ознак до простору інформативних ознак;
- розв'язання власне задачі розпізнавання образів.

Задача класифікації формулюється таким чином. Задана множина образів $X \equiv \{x_i\}$ ($i = 1...m$), де m — загальне число образів. Кожен i -й образ характеризується множиною ознак $P_i \equiv \{p_{ij}\}$ ($i = 1...m$, $j = 1...n$), де n — загальне число ознак (однакове для всіх образів), за якими здійснюється його розпізнавання. Будь-який i -й образ можна подати у вигляді i -ї точки в n -мірному просторі, координатами якого є значення відповідних ознак. Отже, множина P ознак усіх образів являє собою матрицю розмірності $m \times n$.

Необхідно класифікувати множину образів X , тобто віднести їх до того чи іншого заздалегідь визначеного класу $X_k \equiv \{x_{kij}\}$ ($k = 1...K$, $i = 1...m_k$, $j = 1...n$), де K — число класів ($K \ll m$); m_k — число образів у k -му класі, так що $\sum_k m_k = m$.

Процесу класифікації, зазвичай, передують етапи так званого навчання «з учителем» (розд. 10.4), основними цілями якого є формування типових інформативних ознак розпізнаваних образів. Процес навчання «з учителем» дуже схожий на навчання дитини алфавіту. Показавши дитині зображення літери «А», ми говоримо їй: «Це літера А». Дитина запам'ятовує інформативні ознаки цієї літери, тобто в її пам'яті відбуваються певні зміни в потрібному напрямі. Процес подання літер повторюється знову й знову доти, поки дитина запам'ятає всі 33 літери алфавіту.

Навчання «з учителем» здійснюється за допомогою так званих навчальних вибірок, що складаються з пар «вхідний образ — вихідний сигнал» і являють собою регламентований набір еталонних образів, які належать до розглянутих класів. У процесі навчання на вхід системи подається серія образів, що належать (позитивні образи) або не належать (негативні образи) цьому класу й система їх запам'ятовує. На основі цієї інформації система повинна сформулювати ознаки, за допомогою яких удалося б розділити позитивні й негативні образи. Якщо такі ознаки знайдені, то на їх основі будується вирішальне правило, що визначає клас, до якого належать ті чи інші образи.

На підставі порівняння образів, що подаються на вхід КСШІ, з еталонними образами навчальної вибірки робиться висновок про якість навчання й можливості переходу системи до робочого режиму розпізнавання образів. Навчена система здатна розпізнавати подані на її вхід образи й відносити їх до того чи іншого класу.

8.2.3. Кластерний аналіз

Кластерний аналіз (кластеризація) має на меті *поділ* множини образів на деяке число груп, які називають **кластерами** (*cluster* — група, пучок, скупчення). Образи кожного кластера характеризуються якими-небудь загальними властивостями (ознаками). Число кластерів, зазвичай, заздалегідь невідомо. На відміну від задачі класифікації, розбивка на кластери здійснюється без попереднього навчання системи і визначення числа й типу класів. Тому задачу кластеризації називають *класифікацією «без учителя»*. Множина вхідних образів розбивається на кластери так, щоб забезпечити оптимальне (максимальне чи мінімальне) значення обраного критерію кластеризації (оптимізації).

За аналогією із класифікацією, задача кластеризації полягає в розподілі множини образів X на K кластерів $X_k = \{x_{kij}\}$ ($k = 1 \dots K$, $i = 1 \dots m_k$, $j = 1 \dots n$), де m_k — число образів у k -му кластері, так що $\sum_k m_k = m$. Кожен кластер X_k має центр (ядро) c_k .

Поділ безлічі образів на класи або кластери (далі — класи) здійснюється на підставі заданого критерію класифікації, який повинен задовольняти наступні неформальні вимоги:

- образи різних класів повинні бути *непов'язаними* між собою;
- образи окремого класу можуть бути *пов'язаними*;
- розподіл образів усередині кожного класу повинен бути *рівномірним*.

Критерій класифікації, зазвичай, являє собою *міру близькості* $q(X_k)$ ($k = 1 \dots K$) взаємного розташування класів, яка може бути оцінена відстанями (рис. 8.1):

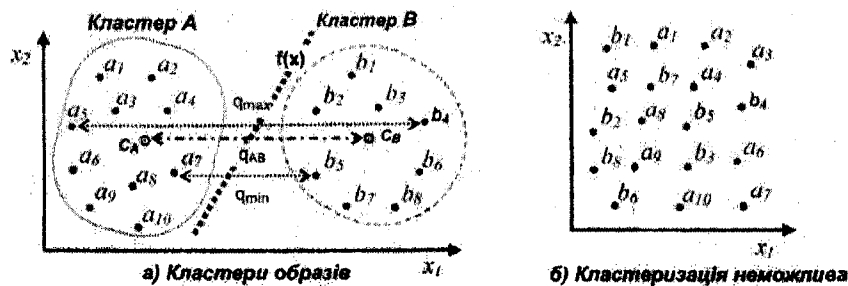


Рис. 8.1 — Схема кластеризації образів

- між *найближчими* сусідніми образами класів q_{min} ;
- між *найдалішими* образами класів q_{max} ;
- між *центрами* класів $q(c_k)$.

У методах розпізнавання образів як *міра близькості* між двома класами A і B приймається, здебільшого, *евклідова відстань* між образами цих класів, поданих у вигляді точок у n -мірному просторі ознак:

$$q(A, B) = \sqrt{\sum_j (a_j - b_j)^2} \quad (j = 1 \dots n) \dots$$

8.2.4. Методи розв'язання задач розпізнавання образів

Методи розв'язання задач *розпізнавання образів* можна поділити на дві великі групи [2.1–2.41, 3.1–3.25, 8.1–8.15].

1. **Дискримінантні методи** використовують описання кожного образу у вигляді точки в n -мірному просторі ознак, властивих даному образу (розд. 8.2.2). До *дискримінантних* методів належать:

а) *детерміністські методи*, в яких рішення щодо належності образу до визначеного класу приймається на основі аналізу геометричних *мір близькості* між точками простору ознак;

б) *ймовірнісні методи Байеса*, в яких ймовірність належності образів до того чи іншого класу визначається на основі ймовірнісних характеристик розпізнаваних образів, поданих як *випадкові вектори* даних.

2. **Структурні методи** ґрунтуються на використанні структури розпізнаваних образів і містять у собі:

в) *синтаксичні (лінгвістичні) методи*, в яких образи описуються певними *словами (фразами)*, а класи образів — деякою *формальною мовою* й розпізнавання образу полягає у віднесенні фрази до тієї чи іншої мови;

г) *логічні методи*, в яких класи образів описуються *предикатами* (розд. 6.2) і розпізнаваний образ (що характеризується набором ознак) може бути віднесений до даного класу, якщо відповідний предикат приймає значення «істина».

Найбільшого поширення в сучасних КСШ набули *дискримінантні методи розпізнавання образів*, розглянуті нижче.

Метод розподільної поверхні

У цьому методі припускається, що множину розпізнаваних об'єктів може бути розділено, наприклад, на два класи — A і B деякою *розподільною поверхню*, яка, зазвичай, являє собою n -мірну *гіперплощину* вигляду (рис. 8.1):

$$f(x) = \sum_{i=1}^n x_i + w_{n+1} \quad (i = 1 \dots n), \quad (8.12)$$

де $x \equiv \{x_i\}$, $(i = 1...n)$ — вектор, який визначає положення розпізнаного образу у вигляді точки в n -мірному просторі ознак; w_i — вагові коефіцієнти (параметри) функції $f(x)$.

Процес навчання системи полягає у визначенні параметрів w_i ($i = 1...n$) функції f і здійснюється у такий спосіб.

1. Задається початковий вектор $w_0 = \{w_{i0}\}$, $(i = 1...n + 1)$.
2. На вхід системи подається послідовність еталонних образів x_k^* ($k = 1...N$), сукупність яких називається навчальною вибіркою довжини N . Заздалегідь відома належність кожного з них до того чи іншого класу образів.
3. Виконується ітераційний процес уточнення параметрів w_i :
 - якщо черговий образ x_k^* належить класу A ($x_k \in A$) і при цьому $f(x_k) < 0$, то

$$w_{k+1} = w_k + c x_k^*, \quad c > 0 \quad (k = 1...N);$$

- якщо черговий образ $x_k^* \in B$ і при цьому $f(x_k) > 0$, то

$$w_{k+1} = w_k - c x_k^*, \quad c > 0, \quad (k = 1...N),$$

де c — малий параметр, що визначається експериментально.

Після побудови розподільної поверхні й визначення параметрів w_i виконується процес розпізнавання (класифікації) образів шляхом підстановки поточної точки x , що відповідає розпізнаваному образу, в рівняння для розподільної поверхні (8.12):

- якщо $f(x) < 0$, то $x \in A$;
- якщо $f(x) > 0$, то $x \in B$.

Слід зазначити, що розподільні функції (8.12) можуть бути також кусково-лінійними й нелінійними. Побудова таких функцій являє собою досить складну проблему, яка тут не розглядатиметься.

Метод Байеса

Метод Байеса належить до ймовірнісних методів розпізнавання образів, що використовують основні положення теорії ймовірностей. Сутність цього методу полягає в такому.

Нехай задана ймовірність $p(x)$ появи розпізнаного образу, що описується вектором x у просторі ознак. Нехай відомі також апіорна ймовірність $p(X_k)$ належності цього образу до класу X_k , а також його умовна щільність розподілу $p(x/X_k)$ ($k = 1...K$), де K — число класів.

На основі зазначених апіорних ймовірнісних характеристик відповідно до методу Байеса можна визначити так звані апостеріорні ймовірності $p(X_k/x)$ належності образу до даного класу:

$$p(X_k/x) = [p(X_k) p(x/X_k)] / p(x) \quad (k = 1...K)...$$

З отриманих величин $p(X_k/x)$ ($k = 1...K$) вибирається максимальна апостеріорна ймовірність $p^*(X_k/x)$ (наприклад, при $k = k^*$), яка і вказує на належність розглянутого образу x до даного класу X_{k^*} .

Метод потенційних функцій

Метод потенційних функцій ґрунтується на ідеї потенціалу електричного заряду, запозичений із курсу фізики. Цей потенціал зменшується в міру віддалення від центра заряду. Відповідно до цього потенційна функція є монотонно спадною функцією в просторі ознак.

За аналогією з методом розподільної поверхні, в методі потенційних функцій також передбачається, що множина розпізнаваних об'єктів може бути розділена, наприклад, на класи A і B лінійною розподільною функцією у вигляді:

$$f(x) = \sum_i \alpha_i \phi_i(x) \quad (i = 1...n),$$

де $x \equiv \{x_i\}$ ($i = 1...n$) — вектор, який визначає положення розпізнаного образу в n -мірному просторі ознак; α_i — вагові коефіцієнти (параметри) функції; $\phi_i(x)$ — система лінійних функцій, які апроксимують розподільчу функцію.

Як і в методі розподільної поверхні, на вхід КСШ подається навчальна вибірка еталонних образів $x_k = x_k^*$ ($k = 1...N$) довжиною N з відомою приналежністю кожного з них до того чи іншого класу.

Вводиться до розгляду позитивна функція у вигляді:

$$\Phi(x, x_k) = \sum_i \alpha_i \phi_i(x) \phi_i(x_k) \quad (k = 1...N),$$

на основі якої будується потенційна функція $V(x, x_k)$, що формується на етапі навчання і задовольняє умовам:

$$V_k(x) = \Phi(x, x_k), \quad \text{якщо } x \in A;$$

$$V_k(x) = -\Phi(x, x_k), \quad \text{якщо } x \in B; \quad (k = 1...N).$$

При розпізнаванні чергового $(k+1)$ -го образу X_{k+1} можливі такі випадки:

а) якщо $V_k(x_{k+1}) > 0$ при $x_{k+1} \in A$ або $V_k(x_{k+1}) < 0$ при $x_{k+1} \in B$, то алгоритм правильно класифікує образ X_{k+1} і значення потенційної функції залишається незмінним: $V_{k+1}(x) = V_k(x)$;

б) якщо $V_k(x_{k+1}) < 0$ при $x_{k+1} \in A$ або $V_k(x_{k+1}) > 0$ при $x_{k+1} \in B$, то виникає помилка класифікації, яка виправляється шляхом відповідної корекції потенційної функції:

$V_{k+1}(x) = V_k(x) + \beta_{k+1} \Phi(x, x_k)$ або $V_{k+1}(x) = V_k(x) - \beta_{k+1} \Phi(x, x_k)$, де β_k — послідовність чисел, які визначаються за умовами:

$$\beta_k \rightarrow 0; \quad \sum_k \beta_k = \infty; \quad \sum_k \beta_k^2 = \infty.$$

Наведені умови задовольняє, наприклад, значення $\beta_k = \gamma/k$, де γ — константа, що визначається експериментально.

На сьогодні відомо кілька алгоритмів потенційних функцій, що відрізняються, головним чином, способом корекції розподільних функцій і вибором величин β_k у процесі розпізнавання образів.

8.3. Генетичні алгоритми

8.3.1. Основні визначення

Останніми роками в КСШІ все ширше використовуються *генетичні алгоритми* розв'язання інтелектуальних задач, концептуальна основа яких розроблена американським ученим Холландом. У його роботах запропоновані так звані *еволюційні алгоритми* розв'язання задач *генної інженерії*, які базуються на концепціях *еволюції* (природного відбору) і *генетики* живих істот [8.1—8.31, 8.34—8.36].

Генетичні алгоритми використовують механізм *генетичного спадкування* у природі, сутність якого полягає в наступному. У кожній клітині будь-якої живої істоти (організму, індивідуума) міститься вся генетична інформація про спадковість цієї істоти, записана як сукупність молекул *дезоксирибонуклеїнової кислоти (ДНК)*, що називається *хромосомою*. Вона має вигляд ланцюжка і може складатися з молекул-*нуклеотидів*, чотирьох типів, що позначаються буквами *A, T, C, G* (*A* — аденін, *T* — тимін, *C* — цитозин, *G* — гуанін) і називаються *генами* хромосоми. Вони *кодують уроджені властивості* істоти (колір очей, спадкові хвороби, тип волосся тощо) і можуть набувати різних значень, які називають *аллелями*. *Порядок розташування* генів у хромосомі визначає *генетичний код (генотип)* істоти. Кілька істот даного генотипу утворюють *популяцію*, а сукупність її спадкоємних властивостей називають *генофондом* популяції.

Відповідно до *еволюційної теорії* розвитку світу кожна істота у природі прагне вижити й залишити після себе численніше потомство. Досягти цієї мети можуть лише *найсильніші істоти й популяції* живого світу. З огляду на це, кожен біологічний вид цілеспрямовано розвивається й змінюється для того, щоб якнайкраще *приспосуватися (адаптуватися)* до навколишнього середовища й вижити.

Про те, наскільки популяція *приспосована* до середовища існування, наскільки благополучно вона розвивається, судять із динаміки зміни її *чисельності*. Живі істоти, які краще пристосовані до навколишнього середовища, мають більше можливостей для виживання й розмноження, ніж менш пристосовані. Індивідуальна пристосованість кожної істоти до навколишнього середовища впливає на генофонд популяції.

Завдяки *генетичному спадкуванню* нащадки успадковують від батьків основні якості переданої генетичної інформації. Тому нащадки *сильних* істот також будуть *краще пристосованими* і після зміни кількох поколінь пристосованість істот даного виду помітно зростає. Таким чином, *еволюція є процесом оптимізації* життєдіяльності суб'єктів, який називають *процесом генетичного пошуку оптимального вирішення задачі виживання*.

8.3.2. Принципи побудови й функціонування генетичних алгоритмів

Генетичний алгоритм (ГА) являє собою *комп'ютерну імітацію механізму еволюції* у природі з використанням *моделей* (аналогів) природного відбору, пристосованості до змінюваних умов середовища й генетичного спадкування нащадками життєво важливих властивостей батьків (табл. 8.1).

Таблиця 8.1 — Структурні елементи генетичного алгоритму

Механізм еволюції	Генетичний алгоритм
Хромосома	Вектор перемінних (символів або чисел)
Ген	Елемент (позиція, локус) хромосоми
Генетичний код	Порядок проходження генів у хромосомі
Істота (індивідуум)	Структурований набір хромосом
Популяція	Упорядкована сукупність істот

Робота ГА полягає у створенні *нових популяцій* істот, що еволюціонують за принципом «*виживає найсильніший*». *Модель природного відбору ГА* визначає, яким чином створюється популяція наступного покоління. У процесі відбору здійснюється цілеспрямований пошук таких *властивостей* істот, які є цінними погляду так званої *фітнес-функції* ГА, яка визначає *ступінь пристосованості (придатності, адаптації)* істот до умов їхнього проживання.

Модель генетичного спадкування ГА — це структурований набір штучних хромосом, кожна з яких являє собою вектор змінних, що складається з послідовності символів або чисел. Оскільки природна хромосома містить у собі ланцюжок нуклеотидів чотирьох типів, вектори змінних ГА також записують у вигляді послідовностей символів, використовуючи дво-, три- або чотирисимвольний алфавіт.

У багатьох ГА використовуються моделі бінарного кодування, де елементи змінних (гени) мають значення «0» чи «1». Такими моделями можуть бути адекватно описані процеси еволюції багатьох природних популяцій. Кожна змінна x_i ($i = 1...n$) у бінарній моделі ГА кодується певним фрагментом хромосоми, який складається із фіксованого числа генів (табл. 8.2). Початкові значення генів хромосоми генеруються в ГА випадковим чином шляхом послідовного присвоєння значень змінних у бінарному коді. Розбивка (декодування) хромосоми на фрагменти здійснюється за допомогою так званої маски картування, яка «накриває» кілька генів і визначає розподіл спадкоємної інформації вздовж хромосоми.

Таблиця 8.2 — декодування хромосоми у вектор бінарних перемінних

Хромосома	1101	1100	1000	1100	0101	1011	0010	0111	0110	1101
Перемінні	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}

Число генів m у масці картування визначає максимальне число M можливих розрядів кодування бінарної інформації у хромосомі. Наприклад, якщо $m = 3$, то $M = 7$; якщо $m = 4$, то $M = 15$ і т. д. У випадку, наведеному в табл. 8.2, маска картування складається із чотирьох генів, а число перемінних у хромосомі дорівнює десяти ($n = 10$).

Бінарне подання хромосоми спричинює певні труднощі при виконанні генетичного пошуку в неперервних просторах великої розмірності. Для розв'язання таких задач розроблено новий тип ГА з натуральним кодуванням (RCGA), в якому хромосоми являють собою вектори натуральних чисел. Застосування натурального кодування підвищує точність розв'язання задач і швидкість обчислень через відсутність операцій кодування й декодування хромосом на кожному кроці ГА.

Перехід від бінарного до натурального кодування здійснюється у такий спосіб. Двоїчні перемінні x_i ($i = 1...n$) хромосоми подаються у десятковому численні за допомогою так званого коду Грея

(табл. 8.3). Якщо десяткові змінні задані в інтервалі $[a_i...b_i]$, то натуральні десяткові значення змінних x_{di} можуть бути визначені таким чином:

$$x_{di} = a_i + n_{di} (b_i - a_i) / M, \quad (i = 1...n),$$

де n_{di} — натуральне ціле десяткове число, яке визначає порядковий номер i -ї двоїчної змінної у коді Грея; M — максимальне десяткове число розрядів кодування, зумовлене числом генів у масці картування (у цьому випадку $M = 15$).

У таблиці 8.3 наведено порядок декодування фрагментів хромосоми у вектор натуральних змінних x . За допомогою коду Грея здійснюється перехід до двоїчно-десяткового коду, від нього — до цілих чисел і далі — до натуральних значень змінних.

Таблиця 8.3 — Подання бінарних перемінних хромосоми

Код Грея	Двоїчний код	Десятковий код	Значення змінних
0000	0000	0	a_i
0001	0001	1	$a_i + \beta_i$
0011	0010	2	$a_i + 2\beta_i$
0010	0011	3	$a_i + 3\beta_i$
0110	0100	4	$a_i + 4\beta_i$
0111	0101	5	$a_i + 5\beta_i$
0101	0110	6	$a_i + 6\beta_i$
0100	0111	7	$a_i + 7\beta_i$
1100	1000	8	$a_i + 8\beta_i$
1101	1001	9	$a_i + 9\beta_i$
1111	1010	10	$a_i + 10\beta_i$
1110	1011	11	$a_i + 11\beta_i$
1010	1100	12	$a_i + 12\beta_i$
1011	1101	13	$a_i + 13\beta_i$
1001	1110	14	$a_i + 14\beta_i$
1000	1111	15	b_i

8.3.3. Генетичні оператори

Модель ГА являє собою упорядкований набір так званих *генетичних операторів*, які моделюють *репродуктивний* процес формування нової популяції зі старої за принципами генетичного спадкування у природі. *Генетичні оператори* описують процеси передачі нащадкам спадкоємних ознак батьківських істот для підтримки достатнього рівня пристосованості потомства до навколишнього середовища. Є такі типи *генетичних операторів*, які моделюють процеси еволюції в природі (рис. 8.2).

Схрещування

ГА імітує еволюцію популяції як циклічний процес відтворення і зміни поколінь. У процесі еволюції відбувається взаємодія *батьківських хромосом* задля утворення *хромосом нащадка*. Основним способом цієї взаємодії є так зване *схрещування*, яке називають також *кроссовером* (cross-over), або *кроссінговером* (crossing-over).

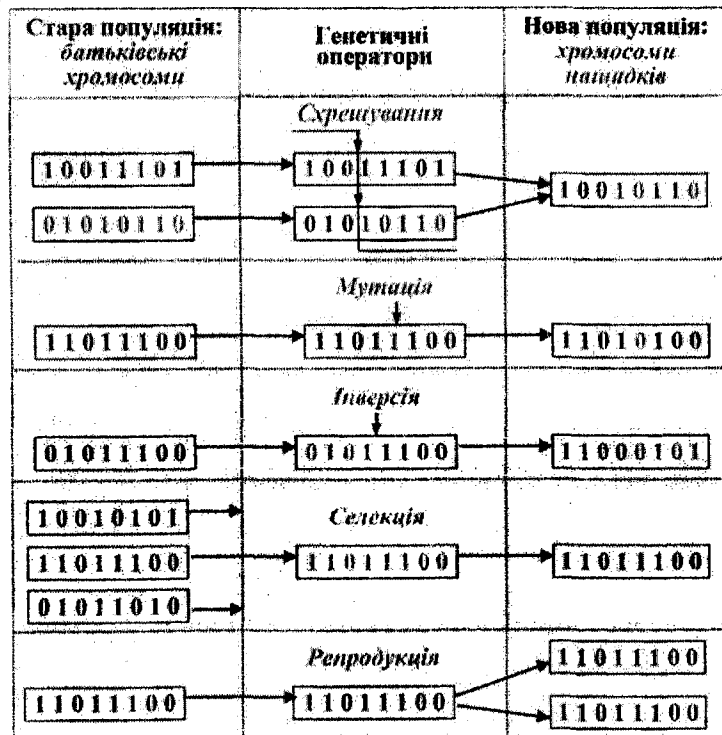


Рис. 8.2 — Типи генетичних операторів

У процесі схрещування хромосоми батьків випадковим чином поділяються на два чи кілька фрагментів, а потім обмінюються цими фрагментами. Місця поділу хромосом на такі фрагменти називають *точками* або *розділовими знаками* оператора схрещування. У практичних застосуваннях найчастіше використовується *одноточкове* й *двоточкове* схрещування.

Для схрещування вибираються дві батьківські істоти: x_{i1} і x_{i2} ($i = 1 \dots n$). Значення нового гена x_i , зазвичай, визначається як лінійна комбінація:

$$x_i = a x_{i1} + b x_{i2} \quad (i = 1 \dots n),$$

де: $a = (1 + c_1 - c_2 (1 + 2c_1))$; $b = (c_2 (1 + 2c_1) - c_1)$;

c_1 і c_2 — випадкові числа, що вибираються з діапазону $[0 \dots 1]$.

При схрещуванні використовуються такі *способи вибору батьківських пар* із істот популяції:

- панміксія* — батьківські пари вибираються *випадково*, причому кожна істота може стати учасником схрещування кількох пар;
- інбридинг* — вибираються істоти, *максимально близькі* одна до одної відповідно до прийнятого *критерію подібності*;
- аутбридинг* — вибираються істоти, *максимально далекі* одна від одної за *критерієм подібності*.

Оскільки в природі схрещуються не всі істоти популяції, то *ймовірність схрещування* пари істот у ГА, зазвичай, коливається в межах 60-90%.

Мутація

При спадкуванні можливі так звані *мутації*, тобто зміни властивостей окремих генів у хромосомах батьків, пов'язані з різними зовнішніми (сприятливими чи шкідливими) впливами навколишнього середовища. Змінені гени передаються нащадку й додають йому *нових властивостей*. Якщо ці властивості корисні, то вони, швидше за все, збережуться у хромосомі нащадка. При цьому відбується стрибкоподібне підвищення *пристосованості* істоти до навколишнього середовища.

Оператор мутації перебирає гени хромосоми й змінює наявний *алельний* стан окремого *мутуючого* гена хромосоми (в деякій випадковій точці) на протилежний. Наприклад, у *бінарному* операторі одиниця змінюється на нуль або навпаки. При натуральному кодуванні застосовується випадкова мутація, за якого ген, що підлягає зміні, набуває випадкового значення з діапазону своєї зміни. Порядковий номер цього гена у хромосомі визначає *ступінь відмінності нащадка від батька*.

Поряд із розглянутою вище *точечною мутацією*, в ГА може використовуватися *групова* (одночасна) *мутація* кількох генів у хромосомі. Відношення *мутуючих генів* до загального числа генів у хромосомі називають *щільністю мутації*, величина якої в ГА може коливатися в межах 1–10%.

Інверсія

Оператор *інверсії* формує хромосому нащадка шляхом перестановки фрагментів батьківської хромосоми. *Ймовірність інверсії* генів окремих істот природної популяції невелика й у ГА вона, зазвичай, становить приблизно 1%.

Селекція

Селекція реалізує процес «виживання найсильнішого», в якому хромосоми відбираються з батьківської популяції у наступну генерацію відповідно до заданої *цільової функції* (критерію відбору). Здійснюється сортування і копіювання декількох кращих істот у нову популяцію, яка потім поповнюється до заданої кількості за допомогою операцій схрещування й мутації.

Репродукція

Під час *репродукції* (копіюванні) із батьківської хромосоми формується одна чи кілька хромосом нащадків. У цьому разі великі шанси потрапити в наступне покоління мають істоти, більш пристосовані до навколишнього середовища.

8.3.4. Функціональна схема генетичного алгоритму

Процес функціонування ГА складається з таких *операцій* (рис. 8.3).

1. Спочатку випадковим чином генерується *початкова* (стартова) популяція, для формування якої найчастіше використовується так звана модель *колеса рулетки*. Поверхня «колеса» поділена на сектори, які відповідають популяціям хромосом із різним ступенем пристосованості. «Кулька», випадково зупинившись в одному із секторів, визначає вибір відповідної хромосоми із цієї популяції.

2. Визначається *ступінь* (індекс) *пристосованості* кожної істоти початкової популяції й обчислюється *середня пристосованість* усієї популяції. Унаслідок застосування оператора *селекції* кожна істота одержує можливість стати батьком із імовірністю, пропорційною її пристосованості до умов існування.

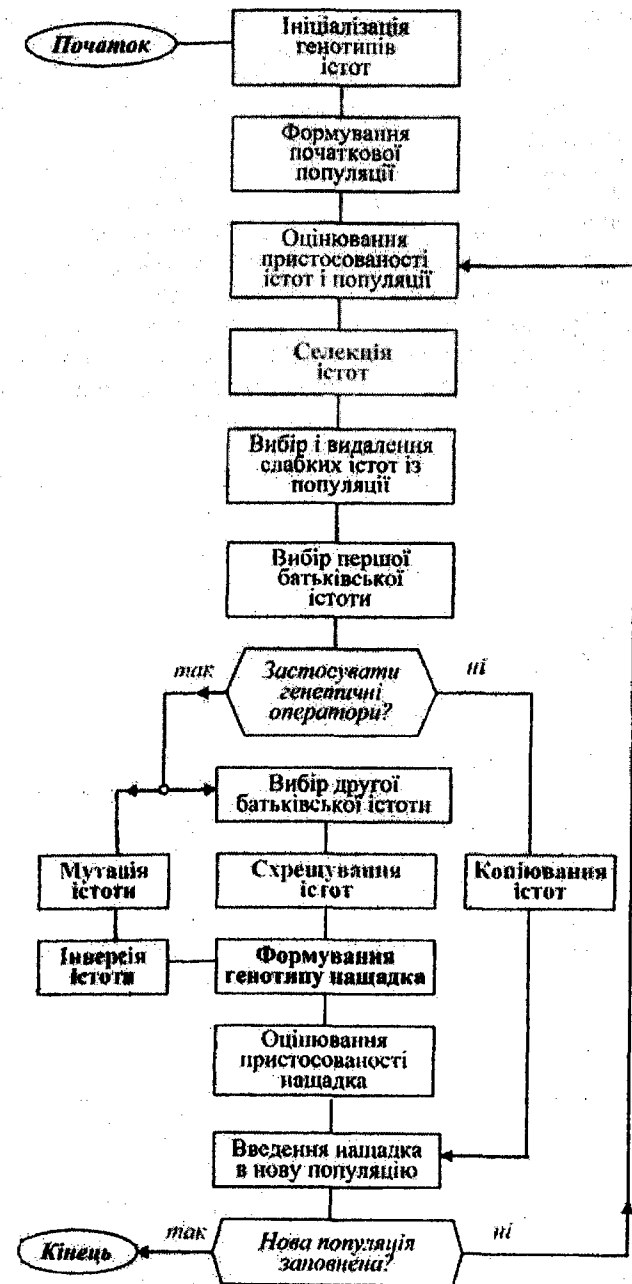


Рис. 8.3 — Схема функціонування генетичного алгоритму

3. Батьківська істота з дуже високою пристосованістю вважається *елітною* й безпосередньо *копіюється* в наступне покоління без змін. Інші батьківські істоти можуть зазнати впливу *генетичних операторів* для створення нащадків із більш високою пристосованістю, ніж їхні батьки.

Істоти з дуже слабкою пристосованістю видаляються з популяції. Використовуючи описані перетворення, можна суттєво збільшити пристосованість популяції нащадків у порівнянні з пристосованістю батьків.

4. Визначається *пристосованість істоти-нащадка*, отримана внаслідок репродуктивних перетворень. Якщо пристосованість істоти вище середньої пристосованості батьків, то ця істота переходить у нове покоління популяції. В іншому разі вона видаляється з популяції.

5. Пункти 2–4 алгоритму повторюються доти, поки нова популяція істот-нащадків не буде цілком *заповненою*.

Робота алгоритму припиняється, якщо протягом заданого числа кроків алгоритму (яке вибирається в межах 10^3 – 10^4) пристосованість істот нащадків залишається незмінною. Результатом роботи алгоритму ГА є формування *популяції нащадків*, яка має *максимальну пристосованість* до умов існування.

8.3.5. Приклади практичного застосування генетичних алгоритмів

Задача розподілу інвестицій

Однією з проблемних областей практичного використання ГА в бізнес-застосуваннях є *розподіл інвестицій*. Хромосомами у цій задачі виступають *інвестиції*, початковою (стартовою) популяцією — *портфель проектів інвестицій*, а генами — обсяги вкладення інвестицій у кожний із проектів (табл. 8.4). Областю зміни окремого гена є мінімальний та максимальний обсяги вкладення в кожний із проектів. Прибутковість окремого проекту розглядається як пристосованість істоти до умов існування. *Цільовою функцією*, що підлягає максимізації, є *сумарний дохід інвестора*.

Таблиця 8.4 — Елементи ГА у задачі інвестування

Генетичний алгоритм	Задача інвестування
Популяція	Портфель інвестицій
Хромосома	Окремий проект (варіант) інвестицій
Гени (перемінні)	Обсяги інвестицій у проекті
Пристосованість істоти	Прибутковість проекту
Цільова функція	Сумарний дохід інвестора

Застосування ГА свідчить, що за малих *сумарних обсягів інвестицій* інвестуються тільки *прибуткові* проекти за *мінімальних* вкладень. Прибутковість проектів у процесі генетичного пошуку зростає, і з'являються більш вигідніші варіанти інвестування. Якщо збільшувати обсяг інвестицій, то буде прибутковим вкладення коштів у значно дорожчі проекти. Однак за подальшого збільшення обсягу інвестицій досягається певний поріг максимальної прибутковості цих проектів і вигіднішим стає інвестування малоприбуткових проектів. Зупиняючи ГА у певний момент і обираючи найкращі проекти, можна одержати оптимальне вирішення задачі, яке забезпечує максимальний дохід інвестора.

Задача аналізу ринкового кошика

Ринковим кошиком називають набір товарів, придбаних оптовим покупцем у межах окремої *транзакції* (ринкової операції, замовлення). Задача *аналізу ринкового кошика* полягає у *виявленні асоціативно пов'язаних ринкових правил* типу: «Якщо відбулася подія *A*, то які події (*B*, *C*, *D* тощо) *виникнуть внаслідок її*». Наприклад, якщо покупець купив принтер (подія *A*), то з якою ймовірністю він купить ще картридж, фотопапір та інші *асоціативно пов'язані* товари. Ця задача належить до типу задач *Data Mining* (розд. 4.5) і орієнтована на дослідження великих інформаційних баз даних для пошуку тенденцій, шаблонів і взаємозв'язків, здатних допомогти в прийнятті тактичних і стратегічних рішень під час здійснення ринкових операцій.

Хромосомою у цій задачі є *вектор цілих чисел* V_i ($i = 1 \dots N$), кожен елемент (ген) якого є *кодом* (ключем) *i*-го товару (товарної групи). *Цільова функція* має вигляд:

$$F = \sum_i k_i T_i \quad (i = 1 \dots N),$$

де: k_i — частота попиту *i*-го товарного набору із числом товарів T_i ; N — число товарів у *шуканому наборі*, які входять до множини T усіх ключів товарів, що містяться у *портфелі замовлень*.

Метою розв'язання задачі є визначення *набору* товарів, які мають *найбільший попит*. Частина позицій початкової популяції в ГА містить заздалегідь відомі товари, а інші позиції заповнюються випадковим чином ключами товарів із множини K . *Селекція* відбувається за пропорційною схемою, відповідно до якої число нащадків пропорційно залежить від значення цільової функції для поточного вектора. *Схрещування* відбувається шляхом обміну значеннями різних позицій двох випадково узятих хромосом (векторів). При здійсненні *мутації* значення деякої позиції хромосоми замінюється

випадково взятим елементом із множини K , критерієм закінчення є відсутність поліпшення цільової функції протягом заданого числа кроків ГА.

Використання ГА при розв'язанні задачі *аналізу ринкового кошика* свідчить, що шуканий набір товарів може бути отриманий за 30–50 ітерацій у випадку попереднього сортування даних про транзакції і за 60–90 ітерацій – у випадку несортованих даних. Ймовірність виконання операцій *схрещування* при цьому коливається в межах 0,4–0,5, *мутації* – 0,05–0,1. Зі збільшенням розмірів аналізованого кошика слід зменшувати зазначені ймовірності.

8.3.6. Особливості застосування генетичних алгоритмів

Головною перевагою ГА є здатність розв'язувати *багатоекстремальні задачі* без накладання умов на вид цільової функції. Наприклад, ГА не потребує безперервності функції мети та її похідних. Крім того, ГА працює за будь-якого (випадкового) початкового наближення. Однак застосування ГА не забезпечує одержання *точного оптимального* розв'язку, яке б відповідало *глобальному екстремуму* цільової функції.

Нині існують ефективні методи розв'язання оптимізаційних задач, представлених, наприклад, у роботах [8.32, 8.33]. Найрозповсюдженішими з них є *градієнтні* методи, що найбільш ефективні при розв'язанні так званих *унімодальних* задач, де цільова функція має єдиний *локальний* (він же – *глобальний*) екстремум. Таким чином, якщо цільова функція має єдиний екстремум у досліджуваній області, то застосування ГА є *проблематичним*, оскільки будь-який градієнтний метод може швидше й простіше знайти розв'язок задачі оптимізації.

ГА доцільно використовувати для розв'язання *багатоекстремальних задач*, де *точність* розв'язку буде *зростати* в міру збільшення часу роботи алгоритму. Для надійності можна повторно запуснути задачу, сподіваючись на сприятливіший розвиток подій. Еволюція неповторна і за нового поєднання випадкових чинників нове рішення може виявитися кращим, ніж попереднє.

Застосування ГА потребує значних зусиль під час настроювання під конкретну задачу, оскільки оптимальні значення його початкових параметрів (розмір популяції, вагові коефіцієнти цільової функції тощо) можуть бути визначені лише шляхом попереднього експериментального дослідження розв'язуваної задачі. Крім того,

у ГА необхідно вибирати (настроювати) *ймовірності застосування генетичних операторів*, які суттєво впливають на збалансованість процесів добирання й формування нових поколінь популяцій.

8.4. Методи групового урахування аргументів

Широко розповсюдженими алгоритмами розв'язання інтелектуальних задач у КСШ є *методи групового урахування аргументів (МГУА)*, які використовують принципи *самоорганізації* еволюційних процесів природного добору у природі [3.23, 8.34–8.36]. В основі МГУА знаходиться так званий *метод багаторядної селекції*, подібний до того, як це відбувається при селекції рослин або тварин.

Наприклад, *багаторядна селекція рослин* здійснюється в такий спосіб. Висівається деяка кількість насіння, що виростає і запилюється, утворюючи складні *успадковані комбінації*. З них вибирається деяка частина рослин із найбільш вираженою *окремою властивістю*, що цікавить селекціонера. Насіння цих рослин збирають і знову висівають для утворення *нових складних комбінацій*. Через кілька *поколінь* (рядів) процес селекції *зупиняють*, оскільки при надмірному продовженні селекції настає *виродження* рослин. Таким чином, є *оптимальне* число *поколінь* і оптимальна кількість *насіння*, яке відбирається в кожному поколінні.

МГУА реалізує *оператор селекції генетичного алгоритму* (розд. 8.3) і призначений, головним чином, для розв'язання задач *ідентифікації моделей* складних об'єктів, процесів і явищ певної ПрО, у якій працює КСШ.

Сутність алгоритмів МГУА полягає в наступному. Нехай необхідно розв'язати *задачу ідентифікації* складного об'єкта з невідомою структурою:

$$y = f(x), \quad (8.13)$$

де: $x \equiv \{x_i\}$, ($i = 1 \dots n$) – n -мірний вектор входів об'єкта, y – вихід об'єкта; f – невідома функція, що підлягає ідентифікації.

Для визначення вигляду функції f на входи об'єкта подається сукупність векторів входних сигналів $x_l \equiv \{x_{il}\}$ ($i = 1 \dots n$, $l = 1 \dots N$) і фіксуються вихідні реакції об'єкта y_l ($l = 1 \dots N$). Окремий вектор x_l називають експериментальною *точкою* в n -мірному просторі *спостережень*. Множина N *точок спостережень* формується заздалегідь і складається із двох частин, які називаються *навчальною* й *перевірочною вибірками* (*последовательностями*).

Реакція об'єкта y на входні сигнали спостережень x_l ($l = 1 \dots N$) може бути подана у вигляді *узагальненого полінома Колмогорова* — Габора:

$$y = a_0 + \sum_{i=1}^N a_{1i} x_i + \sum_{j=1}^N \sum_{k=1}^N a_{1j} x_l x_j + \sum_{l=1}^N \sum_{k=1}^N \sum_{j=1}^N a_{1jk} x_l x_j x_k + \dots, \quad (8.14)$$

де всі коефіцієнти є *невідомими* і підлягають визначенню.

Розв'язання задачі ідентифікації полягає у знаходженні таких коефіцієнтів полінома, які забезпечують *мінімум середньоквадратичної похибки*:

$$e^2 = (1/N) \sum_{i=1}^N [y_i - f(x_i)]^2 \rightarrow \min. \quad (8.15)$$

Отже, поставлена *задача ідентифікації* зводиться до *оптимізаційної задачі відновлення функції* за експериментальними *точками спостережень*.

В алгоритмах МГУА узагальнений поліном Колмогорова — Габора подається як *багаторядна сукупність поліномів* із меншим числом членів. Як приклад розглянемо поліном четвертої степені:

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4, \quad (8.16)$$

який можна подати у вигляді трьох таких *часткових поліномів*:

$$p_1 = b_0 + b_1 x + b_2 x^2; \quad (8.17)$$

$$p_2 = b_3 x^3 + b_4 x^4; \quad (8.18)$$

$$p_3 = c_0 + c_1 p_1 + c_2 p_2. \quad (8.19)$$

Наведені поліноми (8.17)–(8.19) розглядаються в МГУА як *самостійні часткові рівняння регресії*, які в сукупності розв'язують поставлену задачу ідентифікації. При цьому одні й ті самі точки спостережень використовуються для відновлення як *часткових поліномів*, так і всього полінома в цілому.

Для кожної *пари елементів навчальної вибірки* будується *перший ряд часткових описів (опорних моделей)*, тобто формується *перший ряд селекції*:

$$y_1^{(1)} = f(x_1, x_2), y_2^{(1)} = f(x_1, x_3), \dots, y_{M_1}^{(1)} = f(x_{N-1}, x_N), \quad (8.20)$$

де M_1 — *число моделей* першого ряду селекції, дорівнює числу *сполучень* C_N^2 .

Згідно з *методом найменших квадратів (МНК)* визначаються *оцінки коефіцієнтів моделей* (8.20). У прикладі з поліномом четвер-

тої степені (8.16) визначаються оцінки b_s ($s = 0 \dots 4$) коефіцієнтів часткових описів (8.17)–(8.19) для всієї множини моделей (8.20). Найбільше число членів у кожній моделі дорівнює трьом. Отже, для визначення оцінок коефіцієнтів кожної моделі типу (8.20) за методом МНК досить чотирьох точок навчальної вибірки. У цьому разі величини p_1 , p_2 і p_3 приймаються рівними значенню виходу об'єкта y , який відповідає певній навчальній вибірці.

З отриманих моделей першого ряду селекції (8.20) (уже з відомими коефіцієнтами) відповідно до принципу селекції вибирається деяке число M_1^* *кращих моделей*. Для цього на вхід об'єкта подається *перевірочна вибірка* (точки якої не використовувалися у навчальній вибірці) і оцінюється так званий *ступінь регулярності (оптимальності)* кожної μ -ї моделі (8.20) згідно із *критерієм оптимальності* (8.16), тобто

$$e_{\mu}^2 = (1/N^*) \sum_{i=1}^{N^*} [y_i^* - y_{\mu i}]^2 \rightarrow \min, \quad (\mu = 1 \dots M_1) \quad (8.21)$$

де N^* — *число точок у перевірочній вибірці*; y_i^* — *дійсне значення вихідного сигналу об'єкта в i -й точці перевірочної вибірки*; $y_{\mu i}$ — *вихід об'єкта, розрахований за μ -ю моделлю*.

З моделей *першого ряду селекції* за критерієм (8.21) вибирається M_1^* *кращих моделей* і на їх основі будується *другий ряд селекції*:

$$y_1^{(2)} = f(y_1^{(1)}, y_2^{(1)}), y_2^{(2)} = f(y_1^{(1)}, y_3^{(1)}), \dots, \\ y_{M_2}^{(2)} = f(y_{N-1}^{(1)}, y_N^{(1)}), \quad (8.22)$$

де $M_2 = M_1^* < M_1$.

Далі за методом МНК визначаються оцінки коефіцієнтів моделей (8.22) і знову за критерієм оптимальності (8.21) із моделей *другого ряду селекції* вибираються M_2^* *кращих моделей*, де $M_2^* < M_1^*$. Ряди селекції *нарошуються доти*, поки похибка (8.21) *зменшується*. Якщо вона починає падати занадто повільно і зменшення стане меншим від заздалегідь обумовленої малої величини, то процес селекції *завершується*.

Таким чином, МГУА полягає у послідовному переборі часткових моделей об'єкта, які називаються *рядами (поколіннями) селекції*. Кожний частковий опис наступного ряду є *функцією двох аргументів*, що залежать від *функцій попереднього ряду селекції*. За поступового підвищення складності моделей значення критерію селекції проходить через його мінімальне значення. Отже, існує *єдина модель оптимальної складності*, яка й приймається як *модель розглянутого об'єкта керування*.

На даний час відоме велике число алгоритмів МГУА, основними з яких є:

- багаторядні поліноміальні алгоритми з лінійними, квадратичними, коваріаційними та мультиплікативними моделями;
- комбінаторні й ортогоналізовані алгоритми;
- алгоритми з випадковим вибором партнера;
- узагальнені алгоритми з повним перебором трендів;
- нечіткі алгоритми МГУА тощо.

Різні алгоритми МГУА відрізняються один від одного переважно видом часткових опорних моделей (8.8) і (8.10), які в окремих алгоритмах певним чином корегуються (уточнюються) у процесі селекції.

Вибір критерію оптимальності у багатокроковому процесі селекції моделей і визначенні «середовища» розв'язуваної задачі здійснюється вирішувачем КСШІ, який вибирає з БМ (з банку моделей) критерії й моделі, необхідні для розв'язання поставленої задачі.

Крім ідентифікації, алгоритми МГУА застосовуються також при розв'язання інших задач розпізнавання образів, при довгостроковому прогнозуванні процесів і явищ, керуванні складними виробничими процесами та в інших областях людської діяльності.

8.5. Графові моделі та методи розв'язання задач на графах

8.5.1. Основні характеристики графів

Для розв'язання інтелектуальних задач в КСШІ часто використовують *методи пошуку розв'язку на графах* [2.5–2.10, 2.15–2.41, 3.5–3.25, 6.10–6.14, 7.23–7.25]. *Графом* називають геометричну фігуру, що складається з *вершин (вузлів)*, з'єднаних між собою *дугами (ребрами)*. Наприклад, у *графі транспортної мережі* країни населені пункти є *вершинами*, а дороги — *дугами*.

Різновиди та характеристики *вершин* і *дуг* графів наведені в табл. 8.5.

Таблиця 8.5 — Різновиди елементів графів

Елементи графа	Характеристики елементів графа
Суміжні вершини	Дві вершини зв'язані дугою, інцидентною цим вершинам
Кратні дуги	Кілька дуг, що зв'язують суміжні вершини
Вершини вводу даних	Не мають вхідних дуг
Вершини виводу даних	Не мають вихідних дуг

Кінець таблиці 8.5

Елементи графа	Характеристики елементів графа
Термінальні вершини	Вершини вводу-виводу даних
Орієнтована дуга	Дуга, спрямована з початкової у кінцеву вершину
Батьківська вершина	Початкова вершина орієнтованої дуги
Дочірня вершина	Кінцева вершина орієнтованої дуги
Ребро	Дуга у вигляді прямої лінії
Орієнтоване ребро	Ребро у вигляді вектора, спрямованого з початкової у кінцеву вершину
Петля	Дуга з вершинами, що збігаються
Шлях на графі	Послідовність дуг, що зв'язують кілька вершин
Цикл (контур) графа	Замкнутий шлях із початком і кінцем в одній і тій самій вершині
Простий цикл	Не має кратних дуг
Складений цикл	Має кратні дуги
Довжина шляху	Число дуг, що утворюють шлях
Критичний шлях	Шлях на графі, що має максимальну довжину
Вирішальний шлях	Шлях на графі між початковою й цільовою вершинами
Предок	Батьківська вершина шляху
Нашадок	Дочірня вершина шляху

У табл. 8.6 наведені основні *види графів* та їхні характеристики.

Таблиця 8.6 — Основні види й характеристики графів

Вид графа	Характеристики графа
Зв'язний	Будь-які дві вершини графа можна з'єднати
Регулярний	Із кожної вершини графа виходить однакове число дуг
Орієнтований (орграф)	Містить орієнтовані дуги і (або) ребра
Ациклічний (безконтурний)	Не має циклів (контурів)

Кінець таблиці 8.6

Вид графа	Характеристики графа
Циклічний (контурний)	Містить цикли (контури)
Мультиграф	Містить кратні дуги
Псевдограф	Містить петлі й кратні дуги
Підграф	Частина графа, отримана внаслідок вилучення з нього однієї чи кількох вершин
Дерево	Орієнтований граф, кожна батьківська вершина якого зв'язана дугами з кількома дочірніми вершинами
Решітчастий	Орієнтований граф, вершини якого знаходяться у вузлах цілочисельної прямокутної решітки
Зважений	Дуги і (або) вершини графа мають визначені ваги
Позначений	Вершини і (або) дуги графа виділені позначками, індексами або кольором
Гомоморфізм графа	Процес злиття вершин і відповідної зміни дуг графа (згортка)

Формалізовану модель розв'язуваної задачі найчастіше подають у вигляді *дерева графа*, вершинами якого є *стани (ситуації)*, а дугами — *оператори*. Кожна вершина *дерева* має одну попередню *батьківську* вершину. Винятком є *початкова* вершина, яку називають *коренем дерева*, що не має попередніх вершин. Шлях від *кореня* до будь-якої *вершини* дерева є *єдиним*. Якщо в дереві є *тупикові* вершини, то граф є *нерозв'язним*, що означає відсутність рішень у поставленій задачі.

Як приклад на рис. 8.4 наведене *дерево класифікації хвороб* внутрішніх органів людини. *Кореневий вузол* дерева вміщує множину хвороб, а його *вузли-нащадки* відображають групи хвороб, які відповідають певному ураженому органу. Кожний із цих вузлів, у свою чергу, також має своїх *нащадків*, які представляють вужчі групи хвороб. *Термінальними* (вихідними) вузлами дерева є конкретні захворювання.

Стани графа можуть бути подані як символи, вектори, масиви, списки тощо, які описують параметри й властивості розв'язуваної

задачі. Серед них вирізняються *початкові* (зазвичай, один) і *цільові* (кінцеві) стани. Множину всіх можливих станів називають *простором станів* розв'язуваної задачі.

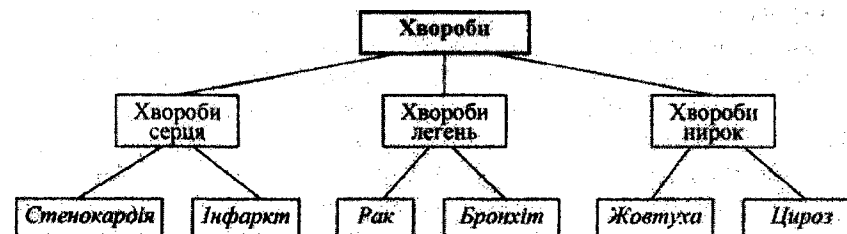


Рис. 8.4 — Дерево класифікації хвороб

Оператори перетворюють одні стани вершин графа в інші. Операторів можна розглядати як функції, визначені на множині станів і такі, що приймають значення із цієї множини.

Якщо, наприклад, система призначена для гри в шахи, то *станами* будуть позиції, що утворюються на шахівниці. Як *початковий* стан може розглядатися позиція, що зафіксована в даний момент гри, а як *цільові* стани — множина нічийних позицій. Множина початкових, проміжних та кінцевих гральних станів становить *простір станів* шахової гри.

Модель розв'язуваної задачі можна навести також у вигляді так званого *«і/або»-графа*, в якому кожна *батьківська* вершина має або *кон'юнктивні* («і»), або *диз'юнктивні* («або») зв'язки (розд. 6.2) зі своїми *дочірніми* вершинами (рис. 8.5).

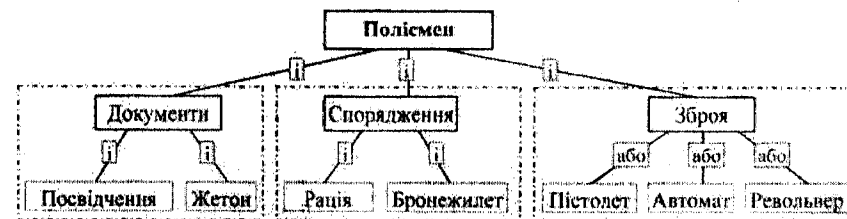


Рис. 8.5 — Дерево «і/або»-графа «Полімен»

Дугам графа, зазвичай, привласнюють певні *ваги*, які відбивають *пріоритетність* дуг під час розв'язання поставленої задачі. Ваги найчастіше визначають *відстань* між вершинами (як на карті доріг) або *ціну* цієї відстані, тобто *умовну вартість* переміщення від вершини до вершини (наприклад, *час* переміщення з однієї вершини на іншу).

8.5.2. Приклади класичних задач на графах

Задача комівояжера

На рис. 8.6 наведено вихідний граф класичної задачі комівояжера, яка полягає в тому, щоб знайти *найкоротший шлях* обходу всіх (у даному разі — *чотирьох*) пунктів, побувавши в кожному з них *один раз*. На графі цієї задачі зазначені *ваги* ребер як *відстані* між пунктами. Обхід починається з пункту *A*.

Ця задача розв'язується в такий спосіб.

1. Описується множина *станів* задачі у вигляді *списків* пунктів, які комівояжер уже відвідав до даного часу. Наприклад, список *ABC* означає, що він уже побував у пункті *B* і знаходиться в пункті *C*. *Дозволеними* є тільки ті *описи станів*, у яких кожен пункт повторюється лише один раз, за винятком пункту *A*, що може знаходитися на початку і в кінці списку.

2. Визначаються *оператори* задачі, кожен із яких описує *рух* до одного з пунктів. Наприклад, оператор № 1 — рух до пункту *A*, оператор № 2 — рух до пункту *B* тощо.

3. Будується формалізована *модель* розв'язуваної задачі у вигляді *дерева графа простору станів* (рис. 8.7). Із графа видно, що є 6 варіантів обходу всіх пунктів. Унизу (в округлених прямокутниках) зазначені довжини шляхів для кожного варіанта.

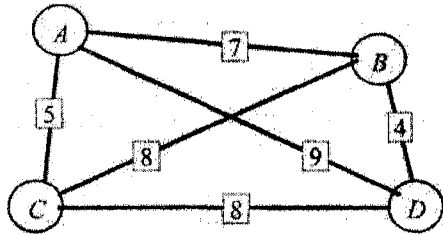


Рис. 8.6 — Вихідний граф задачі комівояжера

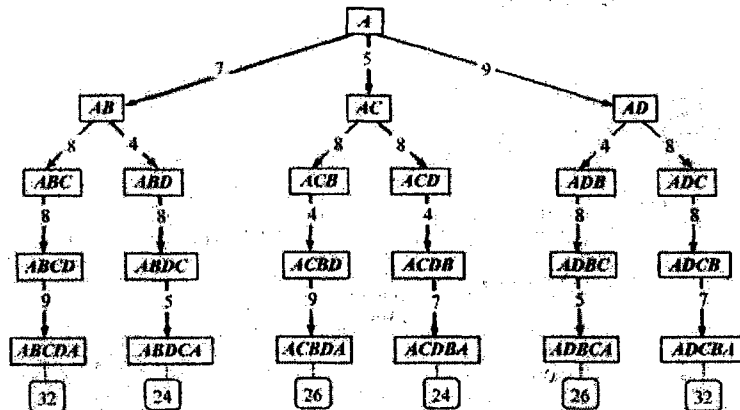


Рис. 8.7 — Дерево графа розв'язку задачі комівояжера

4. Виконується аналіз отриманого графа з погляду заданого *критерію оптимальності*. Таким критерієм у цій задачі є *мінімум довжини шляху* комівояжера при проходженні всіх пунктів. На графі видно, що у цьому випадку є два оптимальних варіанти обходу (виділені напівжирним шрифтом), тобто *два вирішальних шляхи* на графі. Для вибору одного (єдиного) з них необхідно поряд із *мінімумом довжини шляху* ввести ще одну (додаткову) *ознаку оптимальності*.

Задача про «Ханойську вежу»

Класична задача про «Ханойську вежу», що розв'язувалась у *ранніх інтелектуальних системах* (див. *Вступ*), формулюється таким чином (див. рис. 8.8). Є *три стержні* і *три диски* різного розміру: $A < B < C$. Необхідно *перекласти* диски з *першого стержня* на *третій*, не кладучи при цьому *більший диск* на *менший*. На кожному кроці можна *перекладати* тільки один диск.

Ця задача розв'язується подібно до попередньої — комівояжера — у такий спосіб.

1. Описується множина *станів* задачі у вигляді *списків дисків*, що знаходяться на певних *стержнях*.

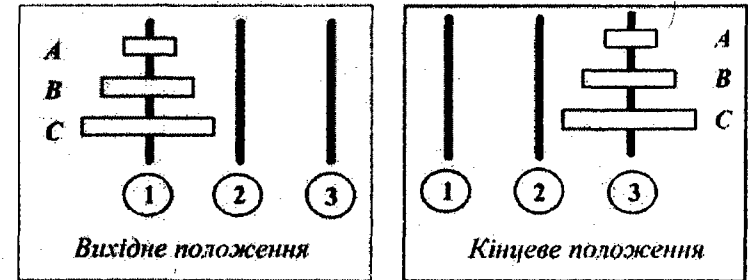


Рис. 8.8 — Постановка задачі про «Ханойську вежу»

Наприклад, список 213 означає, що диск *A* знаходиться на *другому стержні*, диск *B* — на *першому* і диск *C* — на *третьому* стержні.

2. Визначаються *оператори* задачі, кожен з яких описує *перекладання* певного диска з одного стержня на інший.

3. Будується *граф простору станів* із 27 вершинами, кожна з яких відображає одне з положень дисків на стержнях (див. рис. 8.9).

4. Виконується аналіз графа з погляду *мінімально можливого числа перекладань* дисків при їх переміщенні з першого стержня на третій. Один із можливих варіантів *перекладань* показаний *стрілками* як *вирішальний шлях* на графі.

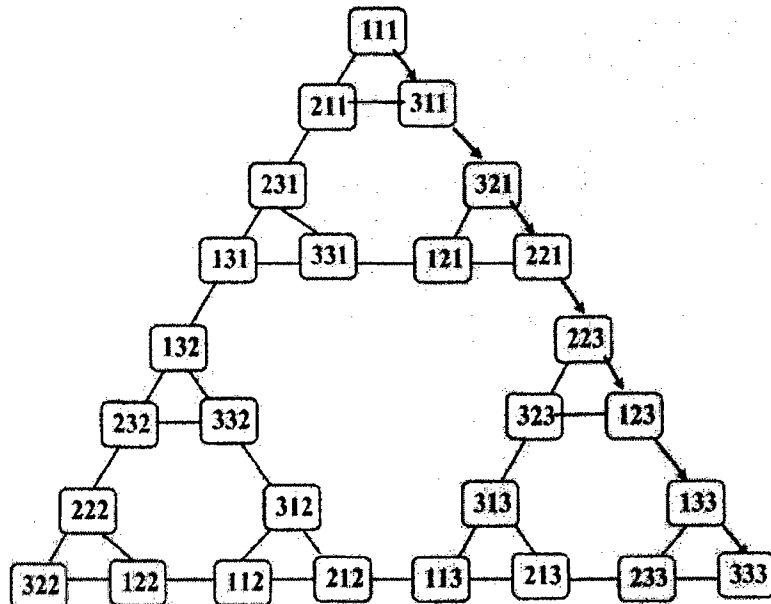


Рис. 8.9 — Дерево графа задачі про «Ханойську вежу»

8.5.3. Алгоритми пошуку розв'язку задач на графах

Процес пошуку рішення задач на графах полягає у знаходженні *вирішального шляху на дереві графа*, який має мінімальну вартість переходу з початкового стану в один із цільових станів. Нижче наведені найвідоміші нині алгоритми пошуку розв'язку задач на графах.

Алгоритм повного перебору варіантів

Класичним алгоритмом знаходження *вирішального шляху на графі* є *алгоритм повного перебору альтернативних варіантів*, який полягає у «сліпому» переборі вершин графа і здійснюється в такий спосіб.

1. Вибирається деяка початкова (батьківська) вершина і до неї застосовуються всі можливі оператори, що породжують дочірні вершини. Цей процес називається *розкриттям вершини*.

2. Для всіх дочірніх вершин встановлюються так звані *покажчики* (стрілки), спрямовані до батьківської вершини, що їх породила. Вони дають змогу знайти шлях назад і здійснити за потреби так званий *back tracking* — повернення з тупика, коли процес пошуку заходить у глухий кут і необхідно повернутися в попередню точку

повернення, щоб знайти інший шлях вирішення задачі. Крім того, покажчики дозволяють повернутися до початкової вершини після того, як задача вирішена, тобто знайдена цільова вершина.

3. Кожна дочірня вершина перевіряється на її належність до цільової вершини (їх може бути кілька). Якщо остання ще не знайдена, то процес розкриття вершин і установки покажчиків триває доти, поки або буде знайдена цільова вершина або будуть вичерпані ресурси пам'яті робочої вирішувача КСШ.

4. Коли цільова вершина знайдена, покажчики проглядаються у зворотному напрямку, тобто від цілі до початку шляху на графі. Таким чином визначається *вирішальна послідовність операторів*, яка відображає шлях розв'язання задачі на графі.

Дерево графа, зазвичай, не задається заздалегідь, а генерується в міру необхідності як окремі гілки графа (підграфів) у процесі розв'язання задачі.

Алгоритм повного «сліпого» перебору може виконуватися шляхом пошуку «у глибину» і «у ширину». При пошуку «у глибину» вводиться оцінка *глибини вершини*, яка дорівнює числу ребер від поточної вершини до кореня дерева. *Вартість* кожного ребра вибирається рівною одиниці. Глибина кореня дерева приймається рівною нулю. Отже, глибина кожної дочірньої вершини на одиницю більша за глибину відповідної батьківської вершини.

При пошуку «у глибину» першою розкривається та з вершин, що має *найбільшу глибину* в дереві графа. При цьому місцезнаходження цілі не впливає на порядок розкриття вершин. Такий підхід може призвести до розгортання процесу розв'язання уздовж деякого *безперспективного* шляху. Тому при пошуку «у глибину» до розгляду вводиться певна *гранична глибина* й у процесі рішення задачі першими розкриваються ті вершини з *найбільшою глибиною*, які не перевищують цієї межі.

При пошуку «у глибину» кожний альтернативний шлях досліджується до кінця, без урахування інших альтернатив. Тому цей алгоритм поганий для «високих» дерев, тому що він легко може «прослизнути» повз потрібну область графа й витратити багато зусиль на дослідження «порожніх» гілок графа.

При пошуку «у ширину» на кожному поточному рівні розкриття вершин досліджуються (на предмет досягнення цілі) всі альтернативні дочірні вершини і тільки після цього здійснюється перехід на наступний рівень. Цей алгоритм може виявитися гірше алгоритму пошуку «у глибину», якщо у графі всі шляхи, що ведуть до цільової вершини, розташовані приблизно на одній і тій самій глибині.

Алгоритми *повного перебору* розрізняються також *напрямом пошуку*. Якщо у просторі станів графа ввести оператори, що переводять *поточні* стани в *попередні*, то пошук можна робити не тільки в *прямому*, а й у *зворотному* напрямках. *Прямий* пошук виконується від початкової до цільової вершини графа і використовується при неявному завданні цілі пошуку. *Зворотний пошук* виконується у зворотному напрямку (від цілі до початкового стану) і використовується за можливості *явного* завдання цілі пошуку. *Комбінований* пошук здійснюється за допомогою переключення з одного режиму на інший у процесі розв'язання задачі, при якому вибирається той напрямок, що звужує область пошуку на графі.

Розглянутий *алгоритм повного («сліпого») перебору варіантів* є найбільш простою формою пошуку розв'язку на графах. Однак цей алгоритм має обмежене застосування в КСШП через проблему так званого *інформаційного вибуху*, за якого *робоча пам'ять* вирішувача у процесі *повного перебору вершин* графа може виявитися переповненою раніше, ніж буде знайдене задовільне рішення задачі.

Алгоритм відмови від безперспективних варіантів (альфа-бета алгоритм)

У цьому алгоритмі використовується *відсікання* частини гілок дерева графа, побудованого на момент розв'язку задачі, для *звільнення простору* робочої пам'яті вирішувача, необхідного для поглиблення перебору. Один із прийомів зменшення перебору полягає в тому, щоб відразу ж після розкриття батьківської вершини залишити для аналізу лише *невелику кількість дочірніх вершин* із *найменшими* значеннями функції *вартості*, а інші дочірні вершини відкинути.

Аналогічний процес пошуку рішення може здійснюватися по *етапах*, що відокремлюються один від одного операціями відсікання галузей дерева, необхідних для звільнення пам'яті. Наприкінці кожного етапу запам'ятовується визначене число відкритих вершин із *найменшими* значеннями функції вартості. Отримані шляхи до цих вершин запам'ятовуються, а інша частина дерева відкидається. Потім процес пошуку рішення продовжується, починаючи уже від цих «кращих» відкритих вершин.

На жаль, у цьому алгоритмі вершини, що відкидаються, можуть виявитися розташованими на кращих шляхах, тому придатність альфа-бета алгоритму для вирішення конкретного класу задач можна визначити тільки експериментальним шляхом на основі досвіду розв'язання задач у певній Про.

Алгоритми пошуку найкоротших шляхів на графах

Прагнення дослідників уникнути *повного перебору альтернативних варіантів* привело до необхідності розроблення *алгоритмів пошуку найкоротших шляхів на графі* шляхом перебору *мінімально можливого* числа вершин графа в процесі розв'язання задачі. Сутність цих алгоритмів полягає в наступному.

З усіх шляхів, що ведуть із даної *батьківської* вершини до *дочірніх* вершин, вибирається *найкоротший* шлях. На цьому шляху виконується *черговий крок* алгоритму, в якому відповідна *дочірна* вершина *розкривається*, утворюючи нові *дочірні* вершини. З отриманих *незакінчених* шляхів (їх число дорівнює кількості дочірніх вершин, що утворилися) вибирається *найкоротший* шлях і робота алгоритму продовжується на *один крок*. Цей процес повторюється, поки не буде досягнута одна з *локальних цільових вершин*.

Отриманий *закінчений (локальний) шлях* запам'ятовується в робочій пам'яті вирішувача. Далі з *незакінчених* шляхів, що *залишилися*, виключаються довші, ніж *закінчений* шлях або рівні йому. Шляхи, що залишилися, обробляються за таким же *алгоритмом* доти, поки їхня довжина стане меншою, ніж *закінчений* шлях. У підсумку або всі *незакінчені* шляхи будуть вилючені, або серед них сформується *закінчений* шлях, більш *короткий*, ніж раніше отримані шляхи. *Останній закінчений* шлях починає відігравати роль *еталона* й процес пошуку триває, поки не буде отримана *глобальна цільова вершина*.

В алгоритмах пошуку *найкоротшого шляху* на графах, зазвичай, використовується *оціночна функція* вигляду [2.9, 2.10, 2.23, 7.23–7.25]:

$$f_i = r_i + p_i,$$

де: r_i — відстань (вартість шляху) між i -ю *поточною* v_i і *початковою* v_0 вершинами;

p_i — передбачувана відстань між *поточною* v_i і *цільовою* v^* вершинами.

Що менше значення оціночної функції f_i , то ближче поточна точка до цільової вершини. За умови $p_i < p_i^*$, де p_i^* — дійсна (невдома) відстань між *поточною* v_i і *цільовою* v^* вершинами, алгоритм гарантує знаходження розв'язку задачі, якщо воно існує.

У процесі пошуку рішення задачі в алгоритмі пошуку *найкоротшого шляху* формуються три *списки* вершин:

- вхідний список X *обраних* (неопрацьованих) вершин, у який спочатку заносяться всі *розкриті дочірні* вершини;
- проміжний список Z *дочірніх* вершин;
- вихідний список Y *оброблених* вершин.

Алгоритм пошуку *найкоротшого шляху* складається з таких кроків.

1. У списку X визначається вершина v_x , для якої виконується умова:

$$f(v_x) < f(v_i) \quad (i = 1 \dots n_x)$$

для всіх n_x вершин списку X . Така вершина v_x переноситься у список Y . Якщо у списку X немає вершин (тобто $n_x = 0$), то процес пошуку на цьому закінчується у зв'язку з відсутністю рішення задачі.

2. Формується список Z дочірніх вершин, у які можливий перехід із вершини v_x . У кожній із них установлюється показник на вершину v_x .

3. Якщо у списку Z присутня цільова вершина v^* , то процес пошуку завершений, оскільки рішення задачі знайдено. Формується результат як *найкоротший шлях* на графі за рахунок зворотного простежування показників від цільової вершини v^* до початкової вершини v_0 .

В іншому разі здійснюється перехід до п. 4 алгоритму.

4. Для кожної вершини v_{zi} ($i = 1 \dots n_z$) списку Z обчислюється оціночна функція $f(v_{zi})$.

5. Якщо вершина v_{zi} не присутня ні у списку X , ні у списку Y , то додати її у список X , приєднавши до неї її оцінку $f(v_{zi})$.

6. Якщо вершина v_{zi} уже присутня у списках X або Y , то порівняти нове значення оціночної функції $f(v_{zi}) = new$ із її колишнім значенням old .

7. Якщо $old < new$, то перейти до п. 1 алгоритму.

8. Якщо $new < old$, то привласнити $f(v_{zi}) = old$. Якщо колишня вершина була у списку Y , то перенести її у список X .

9. Перейти до п. 1 алгоритму.

Алгоритми пошуку *найкоротших шляхів на графі* є найбільш ефективними з розглянутих вище алгоритмів. Однак застосування і цих алгоритмів стикається з певними *труднощами*. Головною з них є необхідність формування оціночної функції, яка адекватно відбивала б «якість» поточного стану. Наприклад, перевага в кількості фігур у шахах не гарантує успіху. Така проста оціночна функція не враховує багатьох особливостей гри, тобто властивостей розв'язуваної задачі. Крім того, алгоритм у процесі пошуку рішення може вийти на «плато», коли жоден із подальших шляхів не веде до «спуску», або потрапити в *локальний мінімум*, із якого можливий тільки «підйом», тобто погіршення значення оціночної функції.

Деякою мірою зазначені труднощі усуваються в *модифікованому алгоритмі* пошуку найкоротших шляхів на графі, який одержав назву *перший крапцій (best-first search)*. Він має відносно крапці властивості й відрізняється від розглянутих вище алгоритмів тим, що

бере до уваги не тільки найближчі стани (тобто *локальну* обстановку), а й «оглядає» велику область простору станів. У разі потреби алгоритм дає змогу *повернутися назад* (тобто здійснити *back tracking*) й піти іншим шляхом, якщо найближчі стани не обіцяють істотного прогресу в досягненні мети. Ця можливість відмовитися від частини пройденого шляху в ім'я глобальної цілі й дозволяє знайти ефективніший шлях.

Нині в алгоритмах пошуку рішення задач на графах використовуються евристичні *метаправила*, які містять попередню інформацію про глобальний характер графа і загальне місцезнаходження цілі. Ця інформація використовується для того, щоб «підштовхнути» пошук до цілі шляхом розкриття, у першу чергу, найперспективніших вершин. Застосування *метаправил* дає змогу враховувати специфіку розв'язуваної задачі й істотно скорочує час розв'язання задач шляхом виключення безперспективних шляхів досягнення мети.

8.6. Прецедентні методи

При вирішенні повсякденних задач людина на підсвідомому рівні завжди прагне заощадити час і сили. І тут на допомогу приходять *пам'ять* і колишній *досвід*, оскільки для людини простіше розпізнати ситуацію і знайти подібну до неї, ніж заново формувати рішення. Пам'ять людини зберігає «*бібліотеку ситуацій*», які зустрічалися в минулому і стосуються нової життєвої задачі. Така бібліотека повинна бути добре *структурована*, щоб у масиві описів ситуацій можна було достатньо швидко розпізнати подібну поточну ситуацію. Крім того, знадобиться також і певний механізм, який дасть змогу *приспосувати* рішення до нової задачі. Такий підхід до рішення інтелектуальних задач називають *прецедентним*.

Мислення людини багато в чому є *асоціативним*. Під асоціацією розуміють певну *міру схожості* або наявність деяких загальних рис у об'єктів, процесів та явищ даної ПрО. Процес *породження асоціацій* і *виводу по асоціації* носить у людини підсвідомий і багато в чому випадковий і неконтрольований характер. Людина вирішує такі задачі шляхом *навчання*, *звикання* до певних ситуацій і *контролю* своєї поведінки. Усі ці здібності деякою мірою повинна мати й КСШІ.

З огляду на ці чинники, нині значного поширення у КСШІ набули *прецедентні методи* вирішення інтелектуальних задач, які одержали назву *Case Based Reasoning (CBR)*. У цих методах властивість асоціації людського мислення використовується для знаходження у БЗ КСШІ *схожих об'єктів, процесів, явищ, ситуацій, які*

називають *прецедентами*. У процесі вирішення задачі вирішувач КСШ *зіставляє запит* користувача із *прецедентами* БЗ й вибирає ті з них, які найбільшою мірою близькі до розв'язуваної задачі [2.1–2.41, 3.5–3.25, 4.1–4.18, 6.10–6.14, 7.1–7.28].

При вирішенні задач на основі прецедентів основним засобом зіставлення зазвичай виступає *фреймова мережа*, в якій *фрейм витягу* оцінює ступінь близькості прецеденту до запиту, а *фрейм модифікації* використовує цю саму інформацію для підстановки одного прецеденту замість іншого. Прецеденти БЗ «конкурують», намагаючись «залучити до себе увагу» фрейму *витягу*, точно так само, як правила, що вибираються із БЗ, конкурують за доступ до вирішувача. *Механізм зіставлення* пов'язує прецедент запиту з певним *кластером* прецедентів розв'язуваної задачі, що дає змогу відрізнити даний прецедент від його «конкурентів».

Кожній властивості прецеденту присвоюється певна *вага* в інтервалі [0...1], відповідно до ступеня «важливості» цієї властивості. Якщо, наприклад, прецеденти означають рахунки користувачів, то залишок на рахунку має суттєве значення і йому варто присвоїти вагу = 1. Прецеденти, в яких більшість властивостей збігаються, заносяться в один і той же розділ (*кластер*) БЗ, унаслідок чого формується *таксономія типів прецедентів*. Це відображується й у структурі фреймової мережі, де *ступінь близькості* прецедентів відповідає близькості їхніх властивостей.

Одним із основних *прецедентних алгоритмів вирішення задач* є *алгоритм зіставлення з найближчим сусідом*, у якому на кожному кроці:

- виконується *порівняння кожної властивості запиту (прецеденту) користувача з такою самою властивістю кожного прецеденту відповідного кластера БЗ*;
- визначається *зважена (агрегована) оцінка g_h ступеня близькості (міра схожості або збігу)* зазначених прецедентів за формулою: $g_h = g \cdot h$, де g — вага властивості; h — ступінь близькості прецедентів;
- із БЗ вибирається *прецедент-переможець із максимальною зваженою (агрегованою) оцінкою g_h* , яка використовується для вирішення задачі користувача (тобто для здійснення *выводу на прецедентах*).

Якщо алгоритм роботи системи припускає і дослідження альтернативних варіантів, то інші прецеденти ранжируються відповідно до отриманих оцінок.

Цей алгоритм дає гарні результати у випадках, коли наявних знань про предметну область достатньо для того, щоб на будь-якій стадії обчислень визначити вірний напрямок подальших дій.

Для адаптації *прецеденту-переможця* БЗ до *цільового прецеденту запиту* використовуються такі підходи:

- *повторна конкретизація перемінних* у прецеденті й присвоєння їм нових значень;
- *уточнення параметрів* прецеденту відповідно до зміни значень його властивостей;

- *пошук* у БЗ алгоритмічного засобу подолання утруднення, що виникло внаслідок заміни одних компонентів рішення задачі іншими.

Зазначені підходи жорстко пов'язані зі способом подання прецедентів у БЗ. Це може бути система фреймів або семантична мережа (розд. 6). У кожному з цих варіантів повинен бути передбачений відповідний спосіб підстановки одних прецедентів замість інших.

8.7. Гібридні алгоритми

Існує певна аналогія між алгоритмами, заснованими на *правилах і прецедентах*. І ті, й інші вибираються із БЗ шляхом *зіставлення* із запитом користувача, причому вибір і ранжування здійснюються на підставі *фонових* знань (наприклад, у вигляді фреймів), які зберігаються в яких-небудь додаткових розділах БЗ. І ті, й інші необхідно якимсь чином *індексувати*, щоб забезпечити ефективний витяг інформації з БЗ.

Однак є й суттєві розбіжності між цими двома класами алгоритмів, основними з яких є такі [7.17]:

- *правила містять перемінні і безпосередньо не описують рішення*, а прецеденти є *константами* і безпосередньо впливають на хід розв'язання задачі;
- *правило вибирається на основі точного зіставлення правила із запитом*, а прецедент вибирається на основі їх часткового зіставлення;
- *застосування правил організоване як послідовність кроків*, які приводять до рішення, а поточний прецедент можна розглядати як наближений варіант вирішення поставленої задачі.

Нині існують *гібридні алгоритми спільного використання правил і прецедентів*, у яких правила мають справу із загальними задачами Про, а прецеденти — з окремими нетиповими ситуаціями. У цих алгоритмах механізм опрацювання прецедентів використовується для аналізу результатів застосування правил [2.1–2.41, 3.5–3.25, 6.10–6.14, 4.1–4.18, 7.1–7.28].

Спочатку для вирішення поточної задачі застосовуються правила, унаслідок чого формується певне рішення. Потім проглядається база прецедентів на предмет виявлення в ній нетипового випадку застосування правил. Використовуючи відповідну міру схожості, можна оцінити ступінь близькості поточної ситуації й прецеденту.

Слід зазначити, що використання прецедентів не рятує розроблювачів БЗ від виснажливої роботи пошуку знань та організації логічного створення нових знань. У багатьох випадках не можна розглядати прецеденти як досить надійну основу для вирішення проблем КСШ. Якщо масштаб розв'язуваної задачі потребує зіставлення сотень і тисяч прецедентів, то наявним методам аналізу прецедентів така задача може бути не під силу. Проте, прецедентний механізм нині розглядається як один із перспективних способів розв'язання інтелектуальних задач.

8.8. Методи конструювання процесу розв'язання задач

В *алгоритмах конструювання* процес розв'язання задачі подається як взаємозалежна сукупність *алгоритмічних компонентів* (програмних модулів) і спеціалізованих *правил вибору та комбінування* цих компонентів [2.1–2.41, 3.5–3.25, 4.1–4.18, 6.10–6.14, 7.1–7.28]. *Компонентами розв'язання* виступають *конструктивні властивості* об'єктів даної ПрО, а *розв'язками* — упорядковані *множини (конструкції)* цих компонентів, які задовольняють певні фізичні обмеження, пов'язані з властивостями об'єктів.

Наприклад, у задачі *діагностики* технічних об'єктів *компонентами рішення* можуть бути окремі *несправності*, а *рішеннями* — *множини несправностей*, що «накривають» усі симптоми неправильного функціонування спостережуваного об'єкта. Одним із основних обмежень у цьому разі може бути формування такої множини можливих несправностей, у якій було б відсутнє дублювання одних і тих самих симптомів.

8.8.1. Висхідна стратегія конструювання

Нехай, наприклад, необхідно розставити меблі в кімнаті. *Мета* вирішення цієї задачі можна сформулювати в такий спосіб: знайти такий *варіант розміщення меблів*, що задовольняв би *обмеження* на геометричні розміри кімнати й меблів, а також враховував би взаємне розташування предметів обстановки (робочий стіл біля вікна, диван напроти телевізора тощо). Почати краще за все з вибору місця для одного-двох *головних предметів*, які зададуть «точки прив'язки» для інших меблів. Потім розставити всі предмети і перевірити, наскільки отриманий варіант (тобто створена *конструкція*) задовольняє задані *вимоги*.

Навряд чи перший варіант буде вдалим. Швидше за все виявиться, що на якомусь етапі розміщення порушуються сформульовані

вимоги. У цьому разі не обов'язково скасовувати все раніше зроблене і повертатися в початок процесу. Зазвичай, можна знайти спосіб, як, зрушивши кілька предметів, «утиснутися» в обмеження. Кращим зі способів такого «підстроювання» є той, як збереже найбільшу *частину* раніше виконаної роботи.

Описаний підхід щодо розв'язання поставленої задачі називається *висхідною стратегією конструювання*, яка полягає в наступному.

1. Рішення задачі починається з *часткового (наближеного) варіанта рішення*, який містить основні (головні) компоненти рішення. Часткові варіанти рішень можуть бути заздалегідь занесені у БЗ або визначені користувачем (на підставі його досвіду й інтуїції) у процесі розв'язання задачі.

2. Якщо компоненти часткового рішення *задовольняють* задані *обмеження* задачі, то задача вирішена. У протилежному випадку поточний варіант рішення розширюється за рахунок додавання нового, найбільш «перспективного» компонента.

3. Якщо новий варіант вирішення «не вписується» в обмеження, те необхідно скорегувати його з *найменшими переробленнями* виконаних етапів і перейти до пункту 2.

Основні труднощі вирішення задач із використанням *висхідної стратегії* конструювання полягають у формуванні множини *часткових рішень (конструкцій)*, оскільки існує занадто багато варіантів таких конструкцій. Компоненти рішення можуть взаємодіяти один із одним і переплітатися будь-яким чином відповідно до можливих причинно-наслідкових зв'язків між ними. Крім того, не можна заздалегідь передбачити, чи підійде дана конструкція для остаточного варіанта рішення, що задовольняє всі обмеження.

З огляду на це, при використанні *висхідної стратегії* конструювання може виявитися, що простір пошуку буде дуже великим і в такий спосіб задача може перейти в розряд нерозв'язуваних. У цьому випадку застосовують *спадну стратегію* конструювання, яка передбачає поетапний процес вирішення задачі на різних рівнях її деталізації.

8.8.2. Спадна стратегія конструювання

Нехай, наприклад, необхідно вирішити задачу планування приміщень будинку на заданій ділянці землі. Цю задачу можна сформулювати на різних рівнях абстракції у термінах:

- *взаємного положення* («кімната А поруч із кімнатою В»);
- *орієнтації* («кімната А на північ від кімнати В»);
- *координат*, що точно вказують положення кімнати А відносно кімнати В.

На найвищому рівні абстракції приймається рішення: яке приміщення з яким повинне межувати (наприклад, кухня і їдальня, ванна і спальня). Це рішення вже скорочує можливість експонентного розширення простору пошуку. На наступному рівні приймається рішення, наприклад, що житлова кімната повинна виходити на південь, а кухня — на північ, щоб у ній було прохолодніше. Це рішення накладає одночасно обмеження і на розташування їдальні. Після того, як буде покінчено із взаємним розташуванням приміщень і їх орієнтацією, будуть потрібні якісь *евристики*, які допомагають виділити місце для всіх приміщень і вирішити виникаючі конфлікти. Наприклад, однією з евристик може бути така: спочатку вибирається місце для головних приміщень, а інша площа розподіляється між підсобними.

Розглянутий підхід до розв'язання задач називається *спадною стратегією конструювання*, яка полягає в наступному.

1. Вихідна задача розбивається на кілька *рівнів підзадач*, які мають різний ступінь деталізації.

2. Спочатку вирішується основна підзадача *верхнього* рівня, яка визначає стратегію розв'язання підзадач нижніх рівнів.

3. Розв'язуються підзадачі нижчих рівнів у спадній послідовності — від верхніх до нижніх. При цьому результат розв'язку підзадачі верхнього рівня розглядається як обмеження (завдання) при розв'язанні підзадач нижніх рівнів.

4. Виконується уточнення деталей розв'язуваної задачі на все більш низьких рівнях абстракції доти, поки вихідна задача не буде цілком розв'язана.

5. Якщо отримане рішення не задовольняє задані обмеження, здійснюється *перегляд рішень*, отриманих на вищих рівнях абстракції, із наступним поверненням до розв'язання підзадач нижніх рівнів.

Вибір часткових варіантів і перегляд рішень, отриманих на різних рівнях абстракції з наступним їх порівнянням, потребує занадто великих обчислювальних витрат. Тому в БЗ систем, орієнтованих на розв'язання задач конструювання, організуються спеціальні розділи, які містять різні *евристики* керування пошуком у просторі вирішення задач. У складі сучасної КСШ може бути кілька БЗ із різними типами *евристичних правил конструювання*. Така архітектура, що одержала назву *системи із дошкою оголошень (blackboard system)*, підтримує ефективний механізм реалізації *стратегій конструювання* [4.4]. Цей механізм забезпечує послідовне уточнення часткових рішень, відновлення інформації після прийняття невдалих рішень, а також дає змогу швидко визначити, в яку саме точку обчислень потрібно повернутися, а не скасовувати по чергові всі попередні операції, щоразу аналізуючи наслідки такого скасування.

Алгоритми конструювання застосовуються у різних КСШ при розв'язанні задач планування, проектування, діагностування тощо.

Експертна система (ЕС) являє собою *інтелектуальну систему*, що моделює *пам'ять і мислення людини-експерта* під час вирішення *інтелектуальних задач*, що знаходяться у його компетенції [3.1–3.25, 9.1–9.38]. *Експерт є фахівцем (професіоналом)* у певній Про (розд. 3.4). Знання експерта використовуються під час створення БЗ системи (розд. 5).

Процес функціонування ЕС здійснюється таким чином. *Користувач*, який бажає розв'язати свою задачу й одержати відповідну інформацію через користувальний інтерфейс, надсилає *запит* до системи, тобто *ставить задачу*. *Вирішувач ЕС*, використовуючи БЗ, розв'язує цю задачу в *діалоговому режимі з користувачем* і видає йому рішення або придатну рекомендацію, пояснюючи хід своїх міркувань за допомогою *блоку пояснень*.

Контактування з ЕС не потребує жодних спеціальних знань — треба тільки натиснути кнопку початкового виклику й чекати, коли на екрані дисплея з'явиться меню, де перелічені всі задачі, при врозв'язанні яких ЕС може надати допомогу. Потім ЕС ставить запитання, а користувач відповідає, натискаючи відповідну клавішу. Якщо користувач не може відповісти, то ЕС може видати додаткову інформацію, що допомагає сформулювати запитання інакше, — аж до докладних інструкцій, підказок або креслень.

9.1. Статичні експертні системи

Статичні ЕС (СЕС) є *локальними КСШ*, що використовуються у тих Про, які істотно не змінюються у процесі експлуатації системи. Тому склад БЗ СЕС залишається практично *незмінним* [3.1–3.10, 9.1–9.24].

На рис. 9.1 наведена структурна схема СЕС, яка є частковим випадком узагальненої структури КСШ (розд. 3.3, рис. 3.1) і містить такі програмно-апаратні пристрої:

- *вирішувач* — інтелектуальний комп'ютер, який моделює мислення людини-експерта;
- БЗ ЕС — *сховище знань* про властивості й закономірності Про (розд. 5);

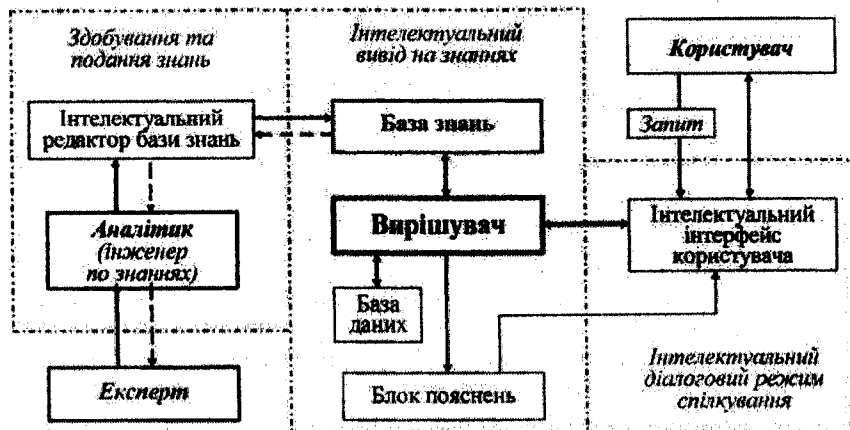


Рис. 9.1 — Структурна схема статичної експертної системи

- *інтелектуальний інтерфейс користувача* (блок спілкування) реалізує діалог користувача з ЕС під час введення інформації, розв'язання задачі й одержання результатів;

- *інтелектуальний редактор БЗ* надає аналітику й експерту можливість створювати БЗ у діалоговому режимі;

- *блок пояснень* видає інформацію про шлях розв'язання поставленої задачі (*запиту*), якщо це цікавить користувача;

- *база даних* (*робоча пам'ять вирішувача*) містить вхідну, проміжну й вихідну інформацію щодо розв'язуваної задачі.

Вирішувач СЕС може працювати в *мультипрограму режимі*, розв'язуючи задачі кількох користувачів. *Віртуальний паралелізм* роботи вирішувача забезпечується *планувальником*, який керує *черговістю* виконання задач, розподіляючи їх у часі так, що складається враження наявності для кожної задачі свого вирішувача. Планувальник конструктивно виконується як складова частина вирішувача.

Нині значного поширення набули *вузькоспеціалізовані* СЕС, БЗ яких містять від 50 до 500 правил. Зазвичай, вони реалізуються на персональних інтелектуальних комп'ютерах й розв'язують нескладні інтелектуальні задачі, що часто зустрічаються. Ці системи називають *інтелектуальними помічниками*, які функціонують на робочих місцях користувачів. Зокрема, компанія «Дюпон» створила тисячі таких СЕС вартістю \$25 тис., кожна з яких дає прибуток \$100 тис.

СЕС може бути апаратно оформлена як валізка і вбудовуватися в панель керування автомобілем, бути в портативному комплекті лікаря тощо. СЕС розрахована на масового споживача. Наприклад, лікар, що практикує в сільській місцевості, може ввести в систему

дані про хворобу й стан пацієнта, його скарги, дієту, характер роботи тощо. У відповідь система видає рекомендації, які відповідають думці кращих експертів-медиків.

СЕС може бути висококваліфікованим помічником інженера, тренера або фахівця іншої професії, якому досить часто доводиться звертатися до *колективного досвіду* своїх колег. СЕС допоможе провести всі необхідні обчислення й викладення, підготувати будь-яку документацію, послугує щоденником, записною книжкою, годинником, будильником тощо.

9.2. Динамічні експертні системи

У другій половині 90-х років минулого століття відбувся справжній прорив у розвитку ЕС, коли різко збільшився обсяг пам'яті комп'ютерних систем, знизилася їхня вартість й були досягнуті нові успіхи в розвитку засобів КСШІ. Це дозволило створити покоління *динамічних ЕС (ДЕС)*, які називають також *партнерськими підсилювачами інтелектуальних здібностей людини* [9.25–9.32].

ДЕС істотно відрізняються від СЕС здатністю *динамічної взаємодії* з користувачами й процесами, що відбуваються в ПрО, а також умінням навчатися й розвиватися, тобто *еволюціонувати* в часі. Наявність таких можливостей дає змогу ДЕС виступати в ролі повноцінного помічника й радника, здатного здійснювати аналіз наявних знань у ПрО, висувати й відкидати гіпотези, оцінювати вірогідність фактів, самостійно поповнювати свої знання, контролювати їхню несуперечність і породжувати нові знання.

Структуру ДЕС можна подати у вигляді *узагальненої структурної схеми КСШІ*, наведеної на рис. 3.1. Поряд із функціональними компонентами СЕС (розд. 9.1), архітектура ДЕС містить у собі апаратно-програмні *засоби одержання й моделювання вхідної інформації* (датчики, аналізатори, контролери) про об'єкти, процеси, явища й ситуації певної ПрО. ДЕС може бути обладнана *виконавчими пристроями*, які реалізують вироблені системою керуючі впливи на об'єкти й процеси зовнішнього середовища. За допомогою *модема* ДЕС може зв'язуватися з *Internet* і одержувати звідти необхідні знання світового рівня.

ДЕС є *розподіленою* ЕС, яка має кілька *кластерних вузлів*, з'єднаних між собою комунікаційними каналами зв'язку. До складу кожного кластерного вузла входить окремий вирішувач, що працює в мультипрограму режимі, і своя окрема БЗ. У процесі функціонування ДЕС кластерні вузли обмінюються інформацією для розв'язку загальної задачі, поставленої перед системою.

Розподілене функціонування ДЕС здійснюється, як правило, із використанням «клієнт-серверної» технології. Розподілений принцип побудови ДЕС дає змогу задіяти при формуванні БЗ кількох віддалених експертів через систему *Internet*.

Характерними рисами й перевагами ДЕС є її спроможність виконувати великий набір функцій, до яких належать:

а) оперативний аналіз і подання неперервного вхідного потоку інформації у вигляді *паралельних процесів обчислень*;

б) структурування *паралельних процесів* і оперативний розподіл вхідної інформації між *паралельними блоками динамічної БЗ* і *робочою пам'яттю ДЕС*;

в) виконання *паралельних обчислень* і *паралельне розв'язання задач* користувачів (*паралельний вивід на знаннях*) з використанням *паралельного вирішувача*;

г) оперативний розподіл внутрішніх інформаційних потоків ДЕС із використанням *високошвидкісного планувальника (диспетчера)*, який забезпечує паралельне вирішення задач за обмежених ресурсів часу й пам'яті системи.

Поряд із цим ДЕС, як типова КСШ, здатна одержувати, нагромаджувати й корегувати знання з відповідної Про, виводити нові знання та пояснювати процес їх одержання.

У зв'язку із зазначеними особливостями основні компоненти ДЕС (головним чином — БЗ) є *динамічними*, оскільки у процесі експлуатації вони зазнають суттєвих змін. *Динамічність ДЕС* визначається *ступенем зміни (еволюції)* її компонентів у процесі функціонування системи. За *рівнем динамічності* сучасні ДЕС поділяються на такі типи.

1. *Пасивні ДЕС* оновлюють свою БЗ, використовуючи вбудований у систему механізм поповнення знань. Цей механізм дає змогу оновлювати й поповнювати БЗ за допомогою аналізу та опрацювання інформації, одержуваної внаслідок діалогового спілкування з користувачами. Знання у БЗ можуть породжуватися також за рахунок використання спеціальних методів, які дозволяють одержувати нові знання шляхом *зіставлення, аналізу й добору* правил у процесі вирішення задач користувачів. До таких методів належить, наприклад, метод *резолюцій* (розд. 8.1).

2. *Квазіактивні ДЕС* змінюють (поповнюють) свою БЗ інформацією, що надходить у систему через визначені періоди (*кванти*) часу. До таких ДЕС належать, наприклад, мікробіологічні ЕС, що роблять періодичні (один раз на кілька годин) лабораторні виміри й аналізують динаміку отриманих показників у порівнянні з попередніми даними.

3. *Активні ДЕС* працюють у сполученні з датчиками об'єктів Про в режимі *реального часу* з неперервною інтерпретацією (опрацюванням) інформації, що надходить у систему. До таких ДЕС належать, наприклад, ЕС моніторингу космічних польотів, ЕС керування виробничими комплексами, ЕС моніторингу лікарняних процесів та інші системи. На частку *активних ДЕС* припадає до 70% усіх наявних на сьогодні ЕС.

Особливе місце в сімействі ДЕС посідають *експертні робочі станції наукових досліджень*, призначені для надання допомоги фахівцям-дослідникам високої кваліфікації і створювані за допомогою інструментальних засобів «G2», «Leonardo», «ART», «KEE» тощо (розд. 9.4). Вбудовані пакети дослідницьких програм дають змогу фахівцю використовувати ДЕС для вирішення задач науково-дослідного характеру. Такі ДЕС мають розвинуту динамічну БЗ, яка вміщує *десятки тисяч правил*, і велике сховище даних, зв'язане з *Internet*, із яких дослідник може одержувати необхідну інформацію світового рівня. Розвинута когнітивна графіка допомагає унаочнювати результати вирішення задач і оцінювати знайдені закономірності.

Поряд із зазначеними типами ДЕС нині інтенсивно розвиваються *прецедентні ДЕС* (розд. 9.6). Перші версії цих ДЕС було створено у другій половині 90-х років минулого століття спочатку в США (фірма «American Research»), а через деякий час — у Росії (НПО «Еталон»). *Прецедентні ДЕС* мають суттєві відмінності, порівняно зі своїми попередниками, в тому, що до складу їх БЗ поряд із правилами входить так звана *база прецедентів*, у якій зберігається інформація про успішний чи неуспішний досвід функціонування системи в певній Про (розд. 8.6).

На сьогодні ДЕС є найбільш розповсюдженими КСШ, що набули стратегічного значення у розвитку науки й виробництва, у керуванні складними об'єктами та проектуванні складних систем, у справах військового будівництва та інших Про.

9.3. Особливості створення та експлуатації експертних систем

Процес створення будь-якої КСШ (зокрема і ЕС) містить у собі *етапи* розроблення, розглянуті у розділі 3.5. Поряд із цим процеси створення й експлуатації ЕС мають свої характерні риси, які й будуть розглянуті у цьому розділі.

Для розроблення ЕС необхідно одночасне виконання таких вимог [9.1—9.38]:

- існування *експертів* у певній ПрО, які знають її набагато краще, ніж звичайні фахівці в області КСПІ;
- експерти зацікавлені в роботі й готові співпрацювати при створенні ЕС;
- експерти однотайні щодо оцінки пропонованого варіанта створення ЕС, в іншому разі оцінити якість розробленої ЕС буде неможливо;
- експерти здатні *вербалізувати* (описати природною мовою) й пояснити свої знання, інакше важко розраховувати на те, що вони будуть витягнуті й вкладені у БЗ ЕС;
- процес створення БЗ ЕС не повинен бути занадто тривалим і може займати у експерта кілька днів, а не місяців;
- задачі, що розв'язуються в ЕС, повинні бути досить важливими й актуальними, оскільки нема сенсу створювати ЕС для вирішення задач, які виникають рідко або можуть бути вирішені людиною зі звичайною кваліфікацією;
- необхідно чітко обмежувати коло розв'язуваних задач, тобто обсяг БЗ повинний бути мінімально можливим, щоб уникнути «комбінаторного вибуху» інформації, необхідної для вирішення поставлених задач;
- розв'язувані задачі повинні бути досить «прозорими» і структурованими, тобто повинні бути виділені основні поняття, відносини і відомі (хоча б експертам) способи вирішення задач ЕС;
- повинна забезпечуватися можливість поступового нарощування системи, тобто БЗ повинна легко розширюватися й коректуватися відповідно до динаміки зміни ПрО;
- повинно бути достотно вихідних даних для перевіряння працездатності ЕС у певній ПрО, щоб розроблювачі могли переконатися в досяжності заданого рівня її функціонування.

При проектуванні ЕС, зазвичай, використовуються два підходи.

1. *Оригінальний підхід* припускає розроблення інформаційної системи з «чистого аркуша». Він реалізується на основі застосування САПР чи CASE-технології і при змінах у системі вимагає щораз нової генерації її програмного забезпечення.

2. *Типове проектування* використовує адаптацію типових розроблень на основі застосування систем компонентного (монтажного) проектування. У цьому разі зміни у системі спричиняє лише зміна конфігурації програмного забезпечення й у поодиноких випадках — його доопрацювання за допомогою CASE-технології.

Проектними ресурсами під час створення ЕС є джерела знань, час розроблення, обчислювальні засоби й обсяг фінансування. Для експерта джерелом знань є його попередній досвід, а для аналітика —

публікації, відомі приклади рішення аналогічних задач, методи подання знань і маніпулювання ними, програмні інструментальні засоби тощо.

При створенні ЕС виникає низка *ускладнень*, пов'язаних із тим, що замовник не завжди може точно сформулювати свої вимоги до розроблюваної системи. Можливе також виникнення труднощів суто психологічного порядку. Наприклад, експерт при створенні БЗ може не передавати свої знання, побоюючись, що згодом його замінять комп'ютером.

Термін розроблення й *упровадження* ЕС становить, як правило, не менш року за трудомісткості до 5 людино-років. Визначення *обсягу фінансування* суттєво впливає на процес розроблення. Наприклад, за недостатнього фінансування перевага може бути надана адаптації наявної ЕС, а не розробленню нової оригінальної системи.

З огляду на значну трудомісткість, створення ЕС вважається не вигідним для якоїсь однієї ПрО. Тому спочатку створюється *порожня ЕС*, у якій БЗ не заповнена. Таку ЕС називають *оболонкою ЕС*. Вона заповнюється пізніше, при виборі конкретної ПрО. *Оболонка* сучасної ЕС являє собою набір різноманітних програмних засобів у вигляді *шаблонів*, які називають *шеллами* (*shells*). Шаблони дають змогу легко переходити від однієї до іншої ПрО.

Відповідно до технології створення ЕС процес її проектування й розроблення здійснюється таким чином. *Організація-замовник* (*кінцевий користувач*) розробляє *технічне завдання* на розроблення ЕС із необхідними характеристиками й передає його *організації-виконавцю*. Ця організація, у свою чергу, розробляє *оболонку* ЕС із *порожньою БЗ*, яка потім передається *аналітику замовника* для заповнення БЗ.

При розробленні ЕС використовується так звана *концепція швидкого прототипу*, яка полягає в тому, що розроблювачі спочатку створюють не саму ЕС, а її *прототип*, який є *початковою версією* створюваної ЕС. БЗ *прототипу* ЕС містить, зазвичай, кілька десятків правил, тоді як БЗ діючої ЕС може вміщувати кілька тисяч (в окремих випадках — десятки тисяч) правил.

Прототип створюється для перевіряння правильності ідей, моделей подання знань та методів розв'язку задач, покладених в основу функціонування ЕС. Прототип повинен підтвердити їх придатність для одержання якісних і ефективних рішень задач певної ПрО. Основною вимогою до прототипу є *мінімум часу і трудомісткості* розроблення за максимального розпаралелювання процесу нагромадження і перетворення знань, виконуваного експертом, із вибором програмних засобів, який здійснюється *аналітиком і програмістом*.

У разі успіху *аналітик* за допомогою *експерта* розробляє *наступну версію прототипу* з розширеними функціональними можливостями. У міру збільшення знань і розширення функцій прототип може досягти такого стану, коли він успішно розв'язує всі задачі даної ПрО. Після цього може бути прийняте рішення про остаточне *перепрограмування* прототипу на мовах високого рівня для поліпшення таких її характеристик, як збільшення швидкодії, зменшення обсягу використовуваної пам'яті, підвищення ергономічних параметрів інтерфейсу тощо.

У разі невдачі може знадобитися розроблення нового прототипу або розроблювачі можуть зробити висновки про непридатність ЕС для використання у даній ПрО. Також може знадобитися створення не одного, а *кількох прототипів*, що ґрунтуються на різних концепціях побудови ЕС.

Відповідно до вище викладеного, *технологія* створення ЕС містить у собі такі етапи розроблення прототипів та системи в цілому.

1. *Демонстраційний прототип* підтверджує (спростовує) правильність вибору програмно-апаратних засобів виводу на знаннях та можливість вирішення основної частини задач даної ПрО. БЗ цього прототипу містить 50–100 правил. Час розроблення — до трьох місяців.

2. *Дослідницький прототип* розв'язує основні задачі ПрО у режимі всебічного тестування. БЗ містить 200–500 правил. Термін розроблення — до 1 року.

3. *Діючий прототип* вирішує задачі ПрО у режимі дослідної експлуатації. БЗ містить 1 000–5 000 правил. Термін розроблення — до 2 років.

4. *Промислова ЕС* забезпечує вирішення задач ПрО у режимі промислової експлуатації. БЗ містить 3 000–10 000 правил. Термін розроблення — до 3 років.

5. *Комерційна ЕС (оболонка)* призначена для продажу іншим споживачам. Являє собою *промислову ЕС* із заповненою або порожньою БЗ. Має інструментальні засоби (шаблони) для заповнення БЗ, розвинуті комунікативні можливості, засоби навчання, когнітивної графіки та інші інтелектуальні комп'ютерні засоби. Термін розроблення — до 3 років.

9.4. Інструментальні засоби створення експертних систем

Комплекс апаратно-програмних засобів, що призначений для створення й удосконалення КСШ, називають *інтелектуальною інструментальною системою (ІС)* [9.7, 9.9, 9.27, 9.33–9.38]. Най-

більшого практичного застосування серед КСШ нині набули *ЕС* (розд. 9.5). Тому більшість наявних ІС використовується, головним чином, для створення ЕС. На сьогодні до числа поширених ІС належать «G2», «Leonardo», «Nexpert Object», «ProKappa», «ART*Enterprise», «Level 5 Object» та інші інструментальні системи.

Як було зазначено в розд. 9.3, розроблення ЕС починається з побудови її *оболонки*, що являє собою набір шаблонів, які дають змогу легко переходити від однієї ПрО до іншої. У сучасних ІС використовується широкий набір інструментальних засобів, який називають *середовищем* розроблення або *повним середовищем*, яке використовується на всіх етапах побудови й експлуатації ЕС, зокрема при створенні оболонок, прототипів, промислових і комерційних систем.

До складу типової ІС входять *підсистеми*:

- автоматизації проектування й програмування;
- документування, налагодження й супроводження системи;
- керування процесом створення КСШ тощо.

Основу функціонування більшості сучасних ІС становить так звана *CASE-технологія* (Computer Aided Software Engineering), яка підтримує всі стадії *життєвого циклу* КСШ [9.7, 9.9, 9.27, 9.33–9.38]. Програмні засоби *CASE-технології* поділяються на дві групи:

• *CASE-ToolKits* («Інструментальні скриньки») — пакети розроблювача (технологічні пакети програм), призначені для підтримки розроблення *однієї стадії* створення КСШ чи для вирішення одного типу прикладних задач;

• *CASE-WorkBenches-системи* («Верстати для виробництва програм») — технологічні лінії виробництва програм, які підтримують усі стадії життєвого циклу КСШ — від проектування до передачі в експлуатацію й супроводження системи.

Функціональна структурна схема *інструментальної CASE-технології* наведена на рис. 9.2.

Сучасні ІС розрізняються, головним чином, типами використовуваних *моделей подання знань* та програмним *середовищем* функціонування системи. Розглянемо основні характеристики найвідоміших нині ІС [9.33–9.38].

1. Широко розповсюдженим типом ІС є система «G2», створена фірмою «Gensym Corp». ІС «G2» безпосередньо керує процесом створення КСШ тощо:

- спостерігає за множиною (до кількох тисяч) одночасно змінюваних параметрів процесу керування;
- обробляє зміни параметрів у режимі реального часу;
- перевіряє позаштатні ситуації на керованому процесі;
- приймає рішення в режимі асистування оператору, в автоматичному режимі тощо.

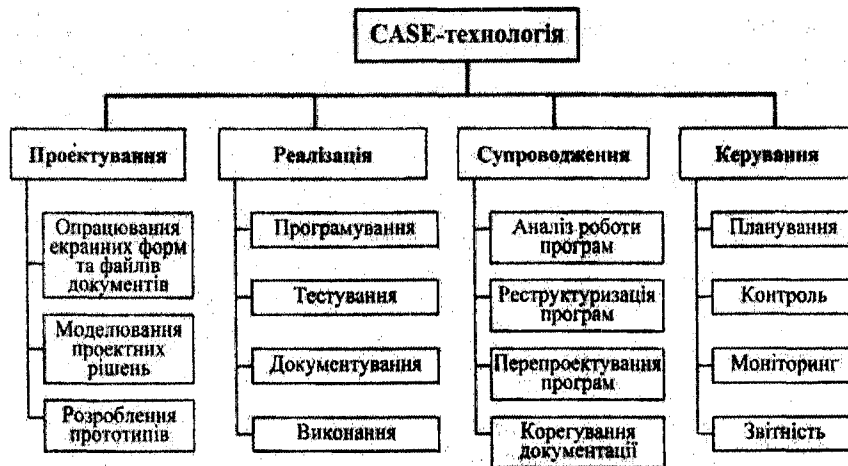


Рис. 9.2 — Функціональна структурна схема CASE-технології

Програмні засоби ПС «G2» забезпечують:

- оперативний вивід на знаннях із посиланнями на минулу інформацію;
- оперативне розпаралелювання і виконання операцій;
- одержання інформації щодо поводження керованого процесу в часі;
- динамічне моделювання знань у процесі вирішення задач;
- доступність даних у режимі «on-line»;
- сканування датчиків для визначення ситуацій, які потребують негайного втручання у процес керування, тощо.

ПС «G2» реалізована на всіх сучасних обчислювальних платформах, зокрема на робочих станціях «Sun», «HP9000», «RS/6000» та персональних комп'ютерах з операційною системою «Windows NT».

2. Розповсюдженим видом ПС є системи типу «ART» («Automated Reasoning Tool»), у яких використовуються *продукційні* моделі подання *процедурних правил* і *фреймоподібні* структури для подання *декларативних* знань. Система забезпечує автоматичну підтримку *множинного спадкування* властивостей фреймів.

Інтерфейс ARTStudio системи ART містить БЗ, підпрограму відладника програм, підсистему меню і підказок, а також графічний інтерфейс ARTist (ART Image Synthesis Tool) із віконним редактором. Цей інтерфейс дає змогу створювати меню, керувати вікнами користувацького інтерфейсу й створювати вікна.

Програмування алгоритмів і середовища функціонування ПС «ART» здійснюється з використанням мов «Lisp» і «C++».

3. ПС «KEE» («Knowledge Engineering Equipment») являє собою інтегрований програмно-апаратний комплекс з універсальним користувацьким інтерфейсом. Програмні засоби системи «KEE» орієнтовані на функціонування у «Lisp»-середовищі й містять:

- *фреймові* структури подання знань зі спадкуванням властивостей;
- *продукційні* правила подання знань;
- *правила логічного виводу* на знаннях;
- *об'єктно-орієнтовані* моделі знань тощо.

Фреймові структури системи «KEE» мають у своєму складі два типи *слотів*:

- *індивідуальні (власні)* слоти, що використовуються для опису атрибутів та властивостей об'єкта (класу об'єктів);
- *колективні* слоти, які описують базові (*родові*) властивості об'єктів даного класу.

ПС «KEE» найефективніша для вирішення тих задач, у яких процес аргументації може трансформуватися, виконуватися й керуватися за допомогою *фреймових* структур, які дають змогу встановлювати кілька видів залежностей між об'єктами.

До складу програмних засобів ПС «KEE» входить логічний блок «TellandAsk», який забезпечує ефективну можливість відновлення й автоматичного перевіряння інформації. Користувацький інтерфейс системи оснащений графічними засобами «KEE_Pictures» і «ActiveImages», призначеними для побудови графічних подань, прив'язаних до фреймових слотів. Користувачу надається графічний редактор правил для початкового введення й корекції правил у процесі налагодження, а також засоби графічного трасування виводу рішень, що дають змогу аналітику легше орієнтуватися у взаємодії правил БЗ.

9.5. Области застосування експертних систем

ЕС використовуються в різних областях людської діяльності, основними з яких є наступні *Про* (у дужках зазначені *назви ЕС*) [9.1–9.38]:

1. Бізнес:

- прогнозування розвитку економіки («Econ»);
- вибір стратегії виходу фірми із кризової ситуації («Crysis»);
- вибір інвестора та страхової компанії («Choice»);
- формування портфеля інвестицій, оцінка фінансових ризиків та оподатковування («Rad», «Rune» та ін.).

2. Виробництво:

- планування роботи підприємств і організацій («Project Assistant»);

- виробництво комп'ютерів та інтегральних схем («*R1/Xcon*», «*DAA*», «*NASL*», «*QO*», «*Cadhelp*», «*Crib*», «*Syn*» тощо) (розд. 9.6);
- технічна діагностика несправностей та відмовлень устаткування («*Intelligence Ware*», «*Plant Diagnostics*», «*Forest*» тощо).

3. Талузі промисловості:

- планування промислових замовлень («*Isis*»);
- контроль стану атомних реакторів («*Reactor*»);
- моніторинг роботи електростанцій та керування мережами розподілу електроенергії («*Cnprint*», «*Alarm Analyser*»);
- контроль за режимами установок на хімічному заводі («*Falcon*»);
- керування повітряним рухом і транспортними перевезеннями («*Airplan*»);
- планування режимів роботи робототехнічних систем («*Strips*»);
- прогнозування воєнних дій («*Analyst*», «*Battl*», «*Tatr*» тощо).

4. Медицина: діагностика захворювань і встановлення зв'язків між порушеннями функціонування організму та їх можливих причин; серед них:

- хіміотерапевтичне лікування ракових захворювань («*Oncocin*»);
- діагностика менінгіту та бактеріальних інфекцій («*Mycin*») (розд. 9.6);
- діагностика й терапія коронарних судин («*Angi*»);
- визначення концентрації гамма-глобуліну в крові («*SPE*»);
- визначення властивостей особистості за результатами психодіагностичного тестування («*Автантест*», «*Microlusher*»);
- медичний контроль за станом хворих («*Aramis*», «*Neurex*») тощо.

5. Наукові дослідження: одержання погоджених і коректних висновків на підставі багатоваріантного аналізу результатів спостережень:

- планування експериментів («*Molgen*»);
 - хімічні дослідження («*Dendral*») (розд. 9.6);
 - виявлення покладів корисних копалин («*Prospector*»);
 - виявлення місць розташування й ідентифікація океанських суден за даними акустичних систем спостереження та результатами аерокосмічного сканування («*Hasp / Siap*»);
 - прогноз погоди («*Willard*»);
 - оцінка майбутнього врожаю («*Plant*») тощо.
6. Освіта (ЕС-тьютори):
- навчання й контроль знань («*Vipes*», «*Guidon*») (розд. 9.6);
 - навчання мови «*Lisp*» («*Учитель Lisp*»);
 - навчання мови «*Pascal*» («*Proust*») тощо.

9.6. Різновиди класичних експертних систем

9.6.1. Медична діагностична ЕС «Mycin»

ЕС «Mycin» є однією з найперших систем діагностики й спостереження за станом хворого при менінгіті та бактеріальних інфекціях. Її першу версію було розроблено у Стенфордському університеті в середині 70-х років минулого століття й вона призначалася для асистування користувачу-лікаря, який не є вузьким фахівцем в області застосування антибіотиків при лікуванні захворювань крові [9.1]. Надалі система неодноразово модифікувалася й оновлювалася (див. нижче). Нині ця система діагностує на рівні професійного лікаря-фахівця.

Базова версія ЕС «Mycin» містить такі компоненти:

- а) БЗ системи містить множину *нечітких продукційних правил* (розд. 6.5, 14), перетворених в оператори мови «*Lisp*»;
- б) *динамічна база даних* містить інформацію про *пацієнтів* та їхні захворювання;
- в) *консультуюча програма* ставить запитання користувачу, видає поради щодо лікування хвороби, використовуючи інформацію про пацієнта та знання БЗ;
- г) *пояснююча програма* відповідає на запитання й видає користувачу інформацію про те, на чому ґрунтуються рекомендації чи висновки, сформульовані системою; при цьому програма здійснює трасування процесу вироблення рекомендацій.
- д) *програма сприйняття знань* слугує для відновлення знань у БЗ у процесі її експлуатації.

Продукційні правила у БЗ записуються у такому, наприклад, вигляді (розд. 6.5): {Якщо (мікроорганізм має грамнегативне забарвлення) і (має форму палички) і (є аеробним), то (з імовірністю 0,8 можна припускати, що він належить до класу Е)}. Число 0,8 називається *коефіцієнтом упевненості* або *рівнем відповідності* (*tally*) правила, тобто мірою правдоподібності висновку, зробленого на підставі сформульованих умов.

Крім правил, у БЗ системи також зберігаються *факти* й *визначення*. Для їх збереження використовуються різні структурні форми:

- списки мікроорганізмів;
- таблиці знань із клінічною інформацією про мікроорганізми;
- форми класифікації клінічних властивостей мікроорганізмів тощо.

Інформація про пацієнтів у БЗ зберігається у вигляді *контекстного дерева «і / або»-графа*, який містить у собі культури мікроорганізмів

та відповідні призначення, пов'язані з прийомом пацієнтами відповідних лікарських засобів. Механізм логічного виведення знань у системі використовує *зворотний вивід* на «і / або»-графі, процес міркувань йде від того, що потрібно довести, до фактів, на яких ґрунтується доказ (розд. 7.3). Мета досягається шляхом розбивки її на локальні цілі.

Пошук розв'язку задачі відбувається в ієрархічно упорядкованому просторі станів через зіставлення й спрацьовування правил. При спрацьовуванні продукційного правила здійснюється *дія*, яка може бути або *висновком*, або *рекомендацією*. Прикладом висновку може бути віднесення даного мікроорганізму до певного класу, а прикладом рекомендації — перелік лікувальних процедур.

Після завершення процесу вирішення задачі система визначає (рекомендує) курс лікування. Вибір терапії у разі бактеріального зараження складається із чотирьох етапів:

- з'ясування виду зараження у пацієнта;
- визначення класу мікроорганізмів, які викликали даний вид зараження;
- визначення набору лікарських препаратів («антимікробних агентів»), призначених для знищення бактерій й перешкоджання їх росту;
- вибір найбільш ефективного лікарського препарату для лікування хвороби.

При виробленні системою рекомендацій може виявитися, що для лікування пацієнта підійде не один лікарський засіб, а кілька. У цьому випадку визначається *результуючий коефіцієнт упевненості* K_y застосування, наприклад, двох лікарських препаратів, який може бути обчислений за однією із формул:

$$K_y = x + y - xy \quad (\text{при } x > 0 \text{ і } y > 0);$$

$$K_y = x + y + xy \quad (\text{при } x < 0 \text{ і } y < 0);$$

$$K_y = (x + y) / (1 - \min(|x|, |y|)) \\ (\text{при } (x > 0 \text{ і } y < 0) \text{ або } (x < 0 \text{ і } y > 0)),$$

де x і y — коефіцієнти впевненості двох однакових висновків, отриманих при застосуванні різних правил; тут $|x|$ означає абсолютне значення x .

Якщо обидві гіпотези підтверджують висновок, то коефіцієнт упевненості їхньої комбінації зростає за абсолютною величиною. Якщо ж одна гіпотеза підтверджує висновок, а інша його спростовує, то наявність знаменника у відповідній формулі згладжує цей ефект.

Оболонка «EMycin»

Оболонка «EMycin» («Empty Mycin») являє собою ЕС «Mycin» із порожньою БЗ (розд. 9.3). Ця оболонка зберігає всі функціональні можливості ЕС «Mycin», яку можна наповнювати знаннями з тієї чи іншої ПрО. На думку розроблювачів «EMycin», ця оболонка може слугувати «кістяком» для створення консультаційних ЕС у багатьох ПрО, оскільки має у своєму розпорядженні набір інструментальних програмних засобів, які полегшують задачу проектувальника конкретної ЕС.

Деякі програмні засоби, уперше розроблені для «EMycin», надалі стали типовими для більшості оболонок ЕС. Серед таких засобів слід зазначити *мову подання правил*, аналогічну мові «Algol». Ця мова, з одного боку, більш зрозуміла, ніж «Lisp», а з іншого — більш структурована, ніж діалект звичайної англійської мови, який було використано в ЕС «Mycin». Крім того, у «EMycin» використана *кластерна схема застосування правил*, що дає змогу розбити правила на *кластери* й вибирати той чи інший кластер відповідно до запиту користувача.

Оболонка «EMycin» виявилася особливо зручною при розв'язанні таких задач, як діагностика захворювань або несправностей, для яких характерна ненадійність вхідної інформації (симптомів, ознак, результатів лабораторних аналізів тощо) та обмеженість простору пошуку рішень, що містить можливі діагнози захворювань пацієнтів.

ЕС «NeoMycin»

Наприкінці 70-х років минулого століття у Стенфорді на базі раніше створених систем було розроблено удосконалену ЕС «NeoMycin», в якій використовувався більш абстрактний підхід до вирішення медичних проблем, ніж у ЕС «Mycin». Значну увагу розробники приділили знанням, які використовують лікарі-практики під час рутинної процедури діагностики та *когнітивного моделювання* процесу діагностування, притаманного людському мисленню.

Цей новий підхід полягає у наданні користувачу інформації про *стратегію керування виведенням знань* і методах вирішення задач, а не просто на перерахуванні правил, що активізуються у процесі роботи системи. З огляду на це, лікар-користувач одержав доступ до інформації про *стратегію діагностування*, яку система використовує у процесі функціонування.

Зворотний вивід на знаннях, використаний у ЕС «Mycin», замінено у ЕС «NeoMycin» виконанням *метаправил*, які кервують процесом

виводу для досягнення абстрактних цілей. Наприклад, метаправило «*Сформулювати питання*» дає вказівку правилам предметного рівня поставити конкретні запитання користувачу. Замість того, щоб формувати судження про окремі фрагменти інформації, метаправила визначають загальну стратегію маніпулювання знаннями. Упорядкована множина таких метаправил утворює в системі *метарівневу архітектуру*, за якої розв'язувані задачі та підзадачі відповідають цілям і підцілям на *метарівні БЗ* (розд. 5.3).

На додаток до правил, що запитують конкретні дані у користувача, у «*NeoMycin*» є також правила, які керують процесом його опитування. Ці правила цитуються при формуванні пояснення, чому користувачу було поставлено конкретне запитання.

Практичні аспекти реалізації ЕС «*Mycin*»

При використанні ранніх версій ЕС «*Mycin*» було отримано напрочуд добрі результати. Команда з п'яти висококваліфікованих експертів в області діагностики інфекційних захворювань підтвердила правильність 72% рекомендацій, зроблених системою, що стосувалися 15 реальних захворювань. Головною проблемою у цьому разі виявилася не точність діагнозу, а відсутність правил, які б давали згоду судити про діагностику захворювань.

Пізніше було організовано іспити удосконаленої версії «*Mycin*», призначеної для діагностики бактеремії та менінгіту. Рекомендації системи в 10 клінічних випадках порівнювалися з висновками 8 провідних медиків Стенфордського університету і рядових лікарів. Вони оцінювали рейтинг 10 рекомендацій про курс лікування в кожному з клінічних випадків.

Результати іспитів показали, що відмінності між рекомендаціями «*Mycin*» і висновками провідних спеціалістів Стенфорда виявилися порівняно невеликими, а стосовно рядових лікарів система виявилася навіть на значно вищому професійному рівні.

Однак із ряду причин ЕС «*Mycin*» надалі не було використано в реальній лікарській практиці й за всієї своєї практичної спрямованості вона залишилася лише експериментальною дослідницькою системою, не розрахованою на комерційне застосування. Основними із цих причин були такі:

- упровадження системи потребувало придбання дуже дорогої (як на ті часи) комп'ютерної техніки, що не могла собі дозволити більшість лікувальних установ;
- обмежений обсяг БЗ системи, що вмішувала близько 400 правил, виявився недостатнім для реального впровадження в медичну практику;

• лікарі-практики не мали особливого бажання працювати на комп'ютері з недосконалим інтерфейсом.

Проте, на основі ЕС «*Mycin*» було створено нові покоління діагностичних ЕС, які використовуються в сучасній лікувальній практиці.

9.6.2. ЕС хімічних досліджень «*Dendral*»

ЕС «*Dendral*» було розроблено в межах однойменного проєкту у Стенфордському університеті на початку 70-х років минулого століття [9.2]. ЕС розв'язує задачу визначення молекулярної структури органічних сполук невідомих хімічних речовин (*зразків*). Вихідною інформацією для вирішення цієї задачі є показання мас-спектрометра, що бомбардує хімічний зразок потоком електронів. Унаслідок цього відбувається перебудова структури зразка і його фрагментів. Зміна структури молекул зразка фіксується мас-спектрометром. Складність рішення зазначеної задачі полягає в існуванні безлічі варіантів поділу складної молекули на фрагменти при її бомбардуванні.

ЕС «*Dendral*» формує опис хімічної структури зразка як сукупність молекул і атомів його речовини. Вхідною інформацією системи є формула молекули речовини та набір обмежень, що накладаються на можливі зв'язки між атомами. Результатом роботи системи є список усіх можливих комбінацій атомів у структурі молекули з урахуванням заданих обмежень.

До складу програмного забезпечення ЕС «*Dendral*» входять програми, які допомагають користувачу відкидати одні гіпотези і структурувати інші, використовуючи знання про зв'язки показань мас-спектрометра зі структурою молекул хімічної сполуки. Одна програма відсіває ті гіпотези-кандидати, що припускають варіанти фрагментації, які не збігаються з даними, отриманими від мас-спектрометра. Інша програма структурує гіпотези, що залишилися, відповідно до того, яку частину піків мас-спектрограми, передбачених цією гіпотезою, було дійсно виявлено в отриманих експериментальних даних. У результаті ЕС формує гіпотези, що прогнозують наявність чи відсутність визначених властивостей мас-спектрограми, а потім ці гіпотези зіставляються з результатами експериментів.

Подальший напрям розвитку ЕС «*Dendral*» полягав у тому, щоб допомогти хіміку виявляти взаємозв'язки між варіантами фрагментації молекул у процесі одержання мас-спектра і структурних характеристик компонентів молекули хімічного зразка. Працюючи спільно, ЕС і хімік вирішують, які дані про структуру компонентів зразка становлять інтерес, а потім відшукують спектрометричний процес, що відображає появу таких даних.

Унаслідок здійснення відповідних досліджень було створено *ЕС «Meta-Dendral»* [9.2], що формує *метаправила*, які використовуються потім у *ЕС «Dendral»* для визначення молекулярної структури невідомої органічної сполуки. Метаправила зв'язують структуру зразка з його мас-спектрограмою. *ЕС* тестує ці правила і за потреби модифікує їх так само, як це зробив би хімік. Для навчання *ЕС «Meta-Dendral»* використовується набір молекул із відомими структурою та мас-спектром. Для цього набору молекул формуються пари «структура-спектр», які входять до навчальної вибірки.

Правила, сформовані системою «Meta-Dendral», порівнювалися з відомими аналогами й аналізувалися досвідченими фахівцями в області спектрометрії органічних сполук. *ЕС* успішно «відкрила» відомі і знайшла нові правила. Здатність сформованих правил прогнозувати вид спектра невідомих їй з'єднань вразила фахівців. Однак ні «Dendral», ні «Meta-Dendral» не стали продуктами *e-комерції*, хоча багато ідей, породжених під час роботи над цими проектами, знайшло широке застосування в комп'ютерній хімії.

9.6.3. *ЕС* терапії онкологічних захворювань «Oncosin»

ЕС «Oncosin» призначена для формування планів лікування хворих онкологічними захворюваннями [9.5]. Курси лікування встановлюються так званими *протоколами* — шаблонами планів лікування, в яких визначаються (специфікуються) необхідні лабораторні аналізи, лікувальні засоби хіміотерапії та, за потреби, курси радіаційної терапії.

При розробленні *ЕС «Oncosin»* однією з найскладніших задач виявилася проблема *здобування знань*. Введення інформації, необхідної для створення *протоколів лікування раку* лімфатичних вузлів, зайняло близько двох років, експерти витратили майже 800 годин робочого часу. Формування наступних наборів протоколів у процесі розвитку системи займало, як правило, кілька місяців. При цьому було виявлено, що ефективність процесу здобування знань багато в чому залежить від того, наскільки успішно аналітик виконує роль перекладача в процесі передавання знань від експертів до програми.

Бажання позбутися цієї залежності надихнуло розроблювачів *ЕС «Oncosin»* на створення програми «*Oral*», яка автоматизує процес придбання знань за рахунок використання *когнітивної графіки* і *мов візуального програмування*. Графічний інтерфейс дає змогу користувачу створювати *пктограми*, які являють собою елементи плану, і формувати з них графічні структури. Розставляючи такі

елементи на екрані комп'ютера та вказуючи зв'язки між ними, користувач формує мнемонічну схему керування потоками інформації, яка потім зчитується й перетворюється у внутрішню програму системи за допомогою відповідної мови програмування. На наступних етапах такі програми перетворюються в *таблиці кінцевих станів*, добре відомі фахівцям в області теорії обчислювальних машин. Для будь-якого поточного стану системи така таблиця дає змогу визначити, в який новий стан перейде система, одержавши визначений набір вхідних сигналів, та який набір вихідних сигналів у цьому разі буде сформований.

Таким чином, у *ЕС «Oncosin»* використовуються такі моделі подання знань:

- *ієрархія протоколів* та їх компонентів;
- *правила, створені* у вигляді фреймів, які формують висновки про значення медичних параметрів у процесі уточнення плану лікування захворювання;
- *таблиці кінцевих станів*, що визначають порядок призначення послідовності терапевтичних компонентів курсу лікування.

Використовуючи *базу протоколів*, *ЕС «Oncosin»* формує рекомендації щодо курсу лікування. Система спочатку вибирає придатний протокол, а потім конкретизує його, тобто призначає конкретні медикаменти, терміни їх прийому тощо.

Включення в систему нового протоколу спричиняє формування ієрархії компонентів, зв'язування їх із придатними породжуючими правилами та заповнення нової таблиці кінцевих станів. Формування елементів нового протоколу здійснюється в процесі «співбесіди» з експертом за допомогою засобів графічного інтерфейсу. При цьому отримані знання перетворюються спочатку в проміжну форму подання, а потім транслуються у формат, що використовується в системі. На останній стадії формуються відповідні правила.

Основними компонентами (*сутностями*) моделі *Про ЕС «Oncosin»* є відповідні *медикаменти* курсу лікування, що утворюють частину *статичної онтології Про*. Між сутностями є *відносини*, які зв'язують атрибути альтернативних медикаментів. Так, медикаменти можуть бути частиною хіміотерапії, а хіміотерапія може бути частиною протоколу.

Для уточнення плану прийому медикаментів при заданих відносинах між компонентами здійснюється звертання до переліку планів. Тим самим модель *Про* системи дає змогу сконцентрувати основну увагу на задачах, а не на використовуваних методах пошуку їхнього рішення. Однак іноді може виникнути потреба у зміні планів для окремих пацієнтів, наприклад, зміна дозування або

заміна одного препарату іншим. Такі зміни здійснюються за рахунок використання *динамічної онтології* ПрО (розд. 5.5).

Основною особливістю моделі ПрО є можливість модифікації призначеного плану лікування за допомогою *предикатів* (розд. 6.2), які характеризують прояви у пацієнта певних симптомів (наприклад, токсикозів) на деякі препарати. Сукупність цих предикатів утворює частину так званої *епістемічної онтології* ПрО системи.

Оскільки плани курсу лікування передбачають певний розклад прийому призначених пацієнту препаратів, то *процедурні знання* щодо способів реалізації протоколів становлять суттєву частину моделі ПрО. Ці знання дають змогу формувати таблиці кінцевих станів, що утворюють іншу частину *епістемічної онтології* ПрО. Для опису процедурних знань розроблена спеціальна мова програмування у вигляді *пиктограм*, які подають окремі елементи плану лікування на екрані дисплея. Ця мова дозволяє експерту представляти досить складні процедури, що пов'язані з реалізацією плану лікування.

Сутності й відносини між ними вводяться в систему за допомогою *екранних формулярів*, у які користувач вибирає елементи з меню. Заповнені формуляри зчитуються й перетворюються системою у *фреймову структуру*. Окремі поля формуляра утворюють слоти фреймів, які автоматично зв'язуються з іншими об'єктами фреймової структури. Наприклад, медикаменти зв'язуються з відповідними об'єктами курсів хіміотерапії тощо.

Терапевтичні компоненти операцій курсу лікування також вводяться шляхом заповнення екранних формулярів. У цьому випадку формуляр являє собою порожній *шаблон* плану, в якому представлені поля для створення розкладу прийому препаратів. Меню можливих дій вміщує такі операції, як зміна дозування, тимчасове припинення прийому ліків тощо. Список можливих дій обмежений, що дає змогу експерту досить легко вводити потрібну послідовність операцій. Використовуючи програму «*Oral*», експерт може сформувати новий протокол упродовж декількох днів. Процес перекладу введеної експертом інформації у вирази, що можуть оброблятися системою, схований від користувача.

Досвід, набутий під час розроблення ЕС «*Oncosin*», було використано при створенні сімейства ЕС «*Protege*» [9.22]. Остання версія цієї системи «*Protege-II*» являє собою комплект інструментальних засобів, які полегшують створення онтології ПрО і формування програм придбання знань (подібних до програми «*Oral*») для різноманітних застосувань.

9.6.4. ЕС проектування конфігурацій комп'ютерів «R1/XCon»

ЕС «*R1*», створена фірмою «*Digital Equipment Corp.*» («*DEC*», США), призначена для інформаційної підтримки розроблення *конфігурацій інтелектуальних систем* на базі *комп'ютерного сімейства VAX*.

Комерційна версія ЕС «*R1*», яку розробив університет Карнегі-Меллон разом із корпорацією «*DEC*», одержала найменування «*XCon*» [9.21]. Ця система була однією з найперших успішних спроб застосування ЕС у промисловості. Першими практичними застосуваннями ЕС «*R1/XCon*» було розроблення конфігурацій обчислювальних комплексів *VAX-11/780* і *VAX-11/750*, а також наступних модифікацій комп'ютерної продукції фірми «*DEC*».

Типовий обчислювальний комплекс *сімейства VAX* вміщує до 100 компонентів, головними з яких є центральний процесор, пристрій керування оперативною пам'яттю та блоки керування інтерфейсом, підключені до єдиного пристрою синхронізації. Шинні інтерфейси підтримують інформаційний обмін із віддаленими периферійними пристроями. Такий набір компонентів дає змогу будувати комп'ютерні системи різної конфігурації.

Одержавши замовлення зі специфікацією характеристик створюваного комп'ютерного комплексу, ЕС «*R1/XCon*» приймає рішення про те, які пристрої потрібно включити до складу комплексу і як їх об'єднати у єдину систему. Для того, щоб визначити, чи відповідає та чи інша конфігурація комплексу тим характеристикам, що подані у замовленні, ЕС повинна мати знання про можливості й характеристики всіх компонентів та зв'язків (відносин) між ними. Не менш складна задача *оптимального компоновання* комплексу з обраного набору компонентів, оскільки при її вирішенні потрібно брати до уваги безліч *обмежень* на взаємне розташування компонентів у структурі комплексу.

БЗ ЕС «*R1/XCon*» містить у собі структуровану сукупність приблизно 10 тис. *продукційних правил* такого, наприклад, вигляду:

{Якщо (поточною активною дією є збільшення кількості пристроїв, що підключаються до шини «Ш1») і (відоме число пристроїв, яке можна підключити до розширювача «Ш1») і (є однопортовий НМД, не підключений до «Ш1») і (відомий тип кабелю, яким повинний бути зв'язаний НМД із «Ш1») то (підключити НМД до «Ш1») }.

Правила БЗ поділені на три категорії:

- *правила застосування операторів* формування й модифікації часткових конфігурацій створюваного комплексу;

- *правила визначення послідовності дій на вибір того чи іншого варіанта конфігурації;*

- *правила накопичення інформації, які організують взаємодію з БД і забезпечують виконання операцій «за замовленням» інших правил.*

БД компонентів системи вміщує інформацію про класи, типи й сукупність характеристик компонентів, які входять до складу створюваних комп'ютерних комплексів.

Динамічна *робоча пам'ять* системи містить внутрішнє (машинне) подання поточної задачі конфігурування, результати проміжних операцій щодо визначення часткових конфігурацій, а також символічні структури, які визначають поточний контекст розв'язуваної задачі. Ці структури допомагають розбити задачу конфігурування на підзадачі, які відповідають активізації визначених груп правил. Підзадачі можна організувати в ієрархічну структуру з тимчасовими відносинами між ними й у такий спосіб визначити порядок (план) вирішення задачі конфігурування створюваного комплексу.

Алгоритм розв'язку задач у *ЕС «R1/XCon»* складається з таких кроків:

- 1) формується елемент керування для поточної задачі і видаляються всі застарілі елементи керування для тих задач, що вже завершені;

- 2) визначаються можливі варіанти конфігурації створюваного комплексу і вилучаються неприйнятні варіанти;

- 3) видаляються зайві оператори відповідно до заданого критерію добору (пріоритету);

- 4) виконується попарне порівняння конкуруючих операторів і приймається рішення — який із них залишити;

- 5) на основі аналізу попередніх кроків із решти операторів вибирається єдиний;

- 6) обраний оператор застосовується до поточного стану задачі, що вирішується, і формується розширення часткової конфігурації створюваного комплексу;

- 7) перевіряється, чи не досягнута сформульована мета зі створення заданої конфігурації; якщо досягнута — процес завершується, в іншому випадку цикл повторюється.

Для реалізації зазначеного алгоритму в системі використана мова *OPS5*.

Як показує досвід роботи *ЕС «R1/XCon»*, процес розширення можливостей системи триває доти, поки не завершиться її життєвий цикл. Про комп'ютерних систем перебуває у постійному русі, оскільки весь час з'являються нові компоненти (пристрої) із новими властивостями, що по-різному сполучаються одини із одним та

з раніше розробленими компонентами. Тому і БЗ, і БД, і робоча пам'ять системи у процесі експлуатації перебуває в стані постійної динамічної зміни (розд. 9.2).

На сьогодні фірма «*DEC*» розробила нову інтерактивну *ЕС «XSEL»*, призначену для надання допомоги покупцям під час вибору *ЕС* потрібної конфігурації.

9.6.5. Прецедентні ЕС комп'ютерного навчання

Процес розроблення нових поколінь *ЕС* був пов'язаний із необхідністю створення *прецедентних систем, зокрема ЕС комп'ютерного навчання (CAI — computer-aided instruction)*. Як було зазначено в розділі 8.6, до складу *прецедентної ЕС*, крім БЗ, входить і база *прецедентів (БП)*, яка поряд із *прецедентами* містить також набір *стратегій розв'язання задач* певної Про. До числа таких *прецедентних систем* належать *ЕС «HYPO»* і «*САТО*», які було створено для навчання студентів — юристів із *методики ведення судових справ* в області *юриспруденції*. Розглянемо основні принципи побудови й функціонування цих систем [9.16].

Для юриста одним із головних компонентів знання законодавства є знайомство з раніше розслідуваними подібними справами — *прецедентами*. Інша частина юридичного знання полягає в умінні знайти в законодавстві аргументи за чи проти визначених обставин у конкретній справі. У процесі обговорення справи в судовому засіданні потрібно вміти формулювати власні аргументи і спростовувати аргументи протилежної сторони. Кваліфікований юрист повинен вміти виступати і як захисник, і як прокурор та передбачати стратегію поведінки обох сторін у судовому процесі. Важливою частиною судових суперечок є також уміння формувати *стратегії застосування* тих чи інших статей закону щодо обставин розглядуваної справи..

Розслідування і міркування в юриспруденції спрямовуються *аргументацією*, тобто аргументами, що виражають протилежні інтереси, за допомогою яких сторони процесу намагаються схилити на свою сторону суддю чи присяжних, переконати їх у тому, що саме пропонується інтерпретація закону і фактів є коректною в цьому випадку. Отже, є деякий аспект ігрового суперництва — сторони роблять визначені *ходи* у боротьбі за перевагу. Поведінку сторін можна розглядати як планування наступної *триходової комбінації* у грі:

- одна сторона за допомогою свого набору *прецедентів* «просуває» свою позицію у грі;

- протилежна сторона висуває інший набір прецедентів для представлення своїх аргументів;

- перша сторона висуває нові розуміння, які париують деякою мірою аргументи протилежної сторони і т. д.

У зв'язку з цим, в основу прецедентних ЕС було покладено *модель процесу суперечки (гри) двох сторін* у судовому засіданні, яка містить:

- порівняння поточного випадку з деяким прецедентом для обґрунтування його аналогічності;

- визначення відмінності (*протиставлення*) між поточним випадком і прецедентом;

- пошук контрприкладу, в якому аналогічний прецедент привів би до іншого результату;

- формулювання гіпотетичних прецедентів, що дали б аргументи за і *проти* визначеної позиції;

- комбінування порівнянь і протиставлень в аргумент, який містить оцінку конкуруючих аргументів.

Процес реалізації цієї гри у прецедентній ЕС містить такі операції:

- *аналіз* чинників, властивих поточному випадку;

- *витяг* із БП прецедентів, які відповідають цим чинникам;

- *упорядкування* знайдених прецедентів за ступенем близькості до поточного випадку;

- *вибір* найпридатніших прецедентів як із погляду однієї сторони, так і з погляду іншої;

- *формування* аргументів для *триходової комбінації* (наведеної вище) за кожним пунктом поточної справи;

- *перевіряння* результатів на гіпотетичних випадках.

Прецедентна ЕС допомагає студенту-юристу перевірити гіпотези щодо застосування статей закону під час розгляду судової справи. Робота юриста з ЕС полягає в тому, щоб проаналізувати обставини справи з юридичного погляду і сформувати аргументи на користь обох сторін, тобто як на користь позивача, так і на користь відповідача. Ці аргументи повинні включати список прецедентів, обраних із БП системи, які кожна зі сторін може використовувати, відстоюючи свою точку зору.

До числа перших практичних (комерційних) систем, створених на базі прецедентів, належить ЕС «*CHEF*», що призначалася для формування *кулінарних рецептів*. Вона приймала інформацію про цільові характеристики страви (тип, смакові якості, своєрідність) і з обліком наявних у БП прецедентів (рецептів страв) формувала придатний рецепт її приготування.

Формування рецептів у ЕС «*CHEF*» здійснюється у такий спосіб. Одержавши замовлення, *модуль витягу* ЕС переглядає БП, відшукує в ній рецепти готування *аналогічних* страв і знаходить страву-прецедент, найбільш близьку до цього замовлення. *Модуль модифікації адаптує* (корегує) обраний прецедент відповідно до заданої цільової специфікації. Після виконання всіх корекцій (модифікацій) *модуль збереження* записує новий рецепт у БП.

Нині прецедентні ЕС знаходять усе більше застосування. Так, зокрема, БП прецедентної ЕС поліції Нью-Йорка містить дані про 786 тис. осіб, БП ЕС НПО «Еталон» — 542 тис. осіб. На створення ЕС із таким обсягом БП зазвичай іде не менше 5 років.

Бортовий комп'ютер американського літака-невидимки «*Стелс*» містить прецедентну ЕС на випадок опрацювання позаштатних ситуацій. Професійні моделі фотоапаратів «*Nikon*» містять у собі прецедентну ЕС із 30 тис. знімальних ситуацій. Подібні ЕС створюються сьогодні у різноманітних сферах людської діяльності.

9.6.6. ЕС навчання й контролю знань

Останнім часом у багатьох Про здійснюється широке впровадження *ЕС навчання й контролю знань (ЕС НКЗ)*, у яких використовуються новітні інформаційні технології навчання, засоби комп'ютерної техніки й сучасні методи контролю рівня знань [9.4, 9.7, 9.12, 9.19, 9.30–9.32].

ЕС НКЗ (їх називають також *тьюторами*) мають у своїй БЗ усі необхідні знання для організації процесу навчання. *Тьютори* працюють в *інтерактивному режимі зворотного зв'язку* з користувачем. *Когнітивна графіка* тьюторів дає змогу проводити не тільки текстове навчання, а й навчання за допомогою зорових образів і мультфільмів, у створенні яких бере участь і тьютор, і той, кого навчають.

Особливого значення ЕС НКЗ набувають у *сфері освіти*, де застосовується *модульно-рейтингова система* навчання, основу якої становить *комп'ютерний тестовий контроль* рівня підготовки учнів (*користувачів*).

Основними структурними компонентами ЕС НКЗ є:

- комп'ютерна *оболонка навчання*;
- комп'ютерний блок *тестування*;
- БЗ навчання й *ітестування*;
- блок редагування БЗ;
- візуальний інтелектуальний інтерфейс користувача;
- блок *аналізу* результатів навчання й тестування;
- блок реєстрації виняткових ситуацій, які виникають при роботі ЕС;
- службова база даних (робоча пам'ять) ЕС.

Інформаційну основу ЕС НКЗ становить *БЗ навчання й тестування*, в якій інформація (знання) про досліджувану ПрО представлена як *електронні навчальні посібники*. Крім того, БЗ містить наступні *структуровані розділи* даних:

- специфікації досліджуваних предметів і об'єктів ПрО;
- комп'ютерні тести і навчальні еталони тестового контролю;
- критерії навчання й тестування;
- структуровані групи користувачів;
- результати аналізу навчального процесу й тестування ЕС тощо.

Процес навчання в ЕС НКЗ проходить у досить вільній формі, коли *той, кого навчають* (далі — *користувач ЕС*), сам вибирає теми, предмети й дисципліни навчання, послідовність етапів, темп навчання тощо. Система надає користувачу повну інформацію про досліджувані предмети й допомагає йому побудувати оптимальний індивідуальний графік навчання. ЕС видає рекомендації користувачу протягом усього курсу навчання. Процес навчання зазвичай складається з кількох сеансів, тому в ЕС передбачена можливість збереження результатів попередніх сеансів навчання. Це дає змогу відновити перерване навчання у будь-який момент.

Комп'ютерне тестування дозволяє прискорити процес навчання, уникнути суб'єктивізму при оцінюванні результатів і забезпечити одержання користувачами глибоких знань і вміння оперувати цими знаннями. Результати тестування фіксуються у БЗ і доступні користувачу для перегляду. Він одержує результати після відповідей на всі запитання або після закінчення фіксованого відрізка часу, відведеного на кожен тест. Користувач має можливість проаналізувати динаміку тестування згідно з темами, дисциплінами, рейтингами тощо.

Щоб виявити труднощі, які виникають у процесі навчання, ЕС здійснює поточний *аналіз результатів* навчання, сутність якого полягає в наступному. Для кожного розділу БЗ установлюється набір так званих *вагових чинників*, які визначають ступінь *труднощів*, що перешкоджають засвоєнню користувачем матеріалу даного розділу. При навчанні ЕС обчислює *загальний ваговий чинник*, підсумовуючи їх за кожним розділом БЗ. Якщо під час вивчення матеріалу користувач відчуває утруднення, то він натискає відповідну клавішу на панелі керування комп'ютера й значення відповідного вагового чинника додається до загальної суми. Якщо загальний ваговий чинник перевищує заздалегідь установлений *критичний рівень труднощів* під час вивчення даного розділу БЗ, то ЕС локалізує конкретні пробіли у навчанні користувача, визначає їх діапазон, корегує графік навчання й видає рекомендації щодо подальшого

засвоєння матеріалу. Структурування знань за розділами БЗ дає змогу судити про помилки користувача під час навчання та його індивідуальні особливості, які заважають сприйняттю матеріалу.

Критичним рівням труднощів значення привласнюються на підставі результатів тестування й виконання відповідних досліджень. Тестуючи різні групи користувачів, ЕС визначає ступінь труднощів вивчення певної ПрО. ЕС аналізує результати тестів, виявляє ключові труднощі сприйняття матеріалу, узагальнює дані соціальних та психологічних досліджень і оцінює здібності до навчання різних груп користувачів.

Для перевіряння несуперечності отриманих результатів використовується метод *верифікації* відповідей на запитання шляхом розгляду виникаючих утруднень і зіставлення їх із вихідними даними на підставі механізму логічного виводу на знаннях. Верифікація виконується для тих розділів БЗ, у яких перевищено *критичний* рівень утруднень. Для верифікації, зазвичай, використовуються *продукційні* правила, в яких нові дані про утруднення, одержувані у процесі навчання, зіставляються із загальними ваговими чинниками.

На сьогодні вже існує певний досвід практичної реалізації ЕС НКЗ із використанням сучасних засобів комп'ютерної техніки й технологій програмування. Програмне забезпечення ЕС НКЗ здійснює інструментальну підтримку процесів навчання й контролю знань шляхом тестування користувачів і виконує такі функції:

- редагування БЗ у режимах сортування й вибору знань;
- керування доступом до інформації БЗ;
- формування досліджуваних тем, предметів та дисциплін навчання й тестування;
- формування тестів і навчальних еталонів тестового контролю різних форматів;
- експертно-редакційне перевіряння та корегування тестів;
- корегування специфікацій та еталонів тестового контролю за результатами експериментального перевіряння;
- формування структурованих списків різних типів користувачів;
- реєстрацію й ідентифікацію користувачів;
- аналіз результатів навчання й тестування;
- формування вихідних форм перегляду результатів навчання й тестування для різних груп користувачів тощо.

ЕС НКЗ пропонує користувачу набір *інструментальних засобів* для оперативного створення і редагування БЗ, що здійснюється за допомогою інтелектуального діалогового інтерфейсу ЕС із користувачем. Для організації такого діалогу використовуються засоби *когнітивної графіки* у вигляді «і/або»-дерева графа навчання, який

дає змогу розмістити на екрані комп'ютера великий обсяг інформації у природній формі (розд. 8.5). Поради ЕС і реакція користувача подаються діями з візуальними об'єктами — вершинами й дугами графа з використанням кольору й елементів анімації. Користувач постійно уявляє, як здійснюється процес навчання, які критерії оцінки, як можна досягти тієї самої мети, діючи інакше тощо.

Програмне забезпечення ЕС забезпечує роботу користувача в локальній мережі з можливістю підключення до системи Internet. Реалізований у ЕС механізм ідентифікації з використанням паролів забезпечує конфіденційність доступу користувачів до інформації.

У системі передбачений широкий набір команд створення, позиціонування, видимості елементів, зміни кольору позначок, написів на кнопках позначення тощо. Набір вкладених меню, панелі інструментів, реєстратор навчальних модулів і редактор форм дають змогу створювати й редагувати навчальні і тестові документи, установлювати параметри роботи програм і виконувати багато інших програмних процедур, що в підсумку забезпечує високу ефективність процесів навчання й контролю знань у ЕС НКЗ.

Для функціонування ЕС та подання знань у БЗ розробляються спеціальні мови й моделі програмування. Так, наприклад, у ЕС «Vipes» використана мова «EDL» («Expert Description Language»), яка являє собою сучасну версію мови «Prolog». Синтаксис цієї мови близький до природного людського спілкування й інтуїтивно зрозумілий користувачам, що не є фахівцями у сфері створення КСШ.

Розділ 10

НЕЙРОКОМП'ЮТЕРНІ СИСТЕМИ

КСШ нейробіологічного напрямку, створювані на підставі штучних нейронних мереж, називають **нейрокомп'ютерними системами (нейрокомп'ютерами)**, а сучасну технологію створення таких систем — **нейрокомп'ютизмом**. **Нейронна мережа** являє собою *структуровану сукупність мікропроцесорів*, з'єднаних між собою відповідно до сучасних *моделей структури* та взаємодії нейронних клітин мозку людини (розд. 2) [2.26–2.41, 3.13–3.25, 10.1–10.45].

Нейрокомп'ютерні системи суттєво відрізняються від традиційних обчислювальних систем принципом роботи та іншими характеристиками, наведеними в табл. 10.1.

Таблиця 10.1 — Порівняльні характеристики обчислювальної та нейрокомп'ютерної систем

Предмет порівняння	Обчислювальна система	Нейрокомп'ютерна система
Процесор	Складний	Простий
	Високошвидкісний	Низькошвидкісний
	Один або кілька	Нейронна мережа процесорів
Пам'ять	Відокремлена від процесора	Інтегрована в процесор
	Зосереджена	Розподілена
	Адресна	Асоціативна
Спеціалізація	Математичні задачі	Інтелектуальні задачі
Носій інформації	Програма	Нейронна мережа
Обчислення	Централізовані	Розподілені
	Послідовні	Паралельні
Надійність	Уразливість	Живучість
Середовище функціонування	Спеціалізоване	Комбіноване
	Постійне	Модифіковане

У наступних розділах розглянуто принципи побудови й функціонування сучасних *нейрокомп'ютерних систем*, наведені їх основні характеристики та області практичного застосування.

10.1. Біологічна нейронна мережа

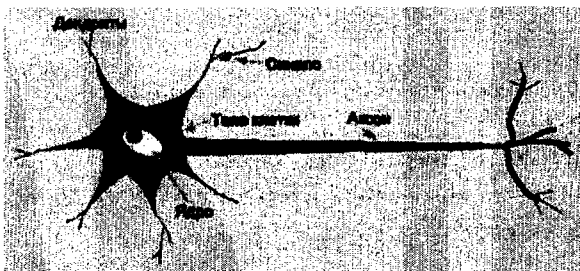


Рис. 10.1 — Структура біологічного нейрона

Як було зазначено в розділі 1, основою людського мозку є біологічна нейронна мережа, яка складається з нейронів, зв'язаних між собою міжнейронними зв'язками — нервовими волокнами мікронної і субмі-

кронної товщини [1.1–1.16]. Кожен нейрон містить безліч гіллястих відростків *двох типів* (рис. 10.1). Відростки першого типу, які називають *дендритами* (за їх подібність до крони розлогого дерева), слугують вхідними каналами для нервових імпульсів від інших нейронів. Ці імпульси надходять у *тіло (сому)* нейрона, яке має розмір від 3 до 100 мікронів.

Дендрити можуть бути *збуджуючими* і *гальмуючими*. Сигнал, що надійшов на збуджуючий вхід, викликає специфічне збудження нейрона. Сигнал, що надходить на гальмуючий вхід, навпаки, знижує збудження нейрона.

Якщо сума сигналів на збуджуючих і гальмуючих входах нейрона перевищує так званий *порог збудження*, то це приводить до формування *імпульсу* на виході нейрона. Цей сигнал поширюється по вивідному відростку другого типу — *аксону*, що може передавати імпульс дендритам інших нейронів через так звані *синапси*, які реагують на силу імпульсу, що надійшов. Таким чином, стан будь-якого нейрона в певний момент залежить від його попереднього стану і від стану нейронів, зв'язаних із його входами.

При проходженні синапса сила імпульсу змінюється у певну кількість разів, яку називають *вагою синапса*. Оскільки ваги синапсів змінюються у часі, то змінюється і стан відповідного нейрона.

Таким чином, нейрони взаємодіють за допомогою короткої серії імпульсів тривалістю кілька *мсек*. Повідомлення від нейрона до нейрона передаються за допомогою *частотно-імпульсної модуляції*. Частота передачі повідомлень не перевищує кількох *сотень герц*,

що в мільйони разів повільніше, ніж швидкість передачі інформації в сучасних комп'ютерних системах.

Тривалість збудження окремого нейрона становить кілька *мсек*, тобто процес спрацьовування нейрона відбувається досить повільно. Однак складну задачу людина вирішує за кілька сотень *мсек*. Оскільки час виконання однієї розумової *операції* у людини становить кілька *мсек*, то процес вирішення задачі складається приблизно зі *100 кроків*. У цьому разі кількість інформації, що посиляється від одного нейрона іншому, не перевищує кілька *біт*. Отже, основна інформація розв'язуваної задачі не передається безпосередньо, а міститься (розподіляється) у зв'язках між нейронами.

Тому дослідники КСШ дійшли висновку, що високу продуктивність опрацювання інформації в мозку людини можна пояснити тільки паралельною роботою безлічі повільних нейронів та великою кількістю взаємних зв'язків між ними, за допомогою яких нейрони постійно обмінюються інформацією. Саме цим пояснюється значне поширення терміна «*масовий паралелізм*» у нейронних мережах.

У людському мозку пам'ять не локалізована в одному місці, як у комп'ютері, а є *розподіленою* і реалізується шляхом посилення чи ослаблення зв'язків між нейронами. Мозок людини реагує не на всі, а тільки на *обрані* зовнішні подразники. Важливою властивістю біологічної пам'яті є її *асоціативність*, тобто можливість *адресації* інформації за її *змістом*. Кожен елемент такої пам'яті може бути знайдений за його змістом та адресою комірки пам'яті.

Кожен нейрон біологічної мережі виступає як елемент збереження інформації та прийняття рішень. Перевагою такої мережі є її *живучість*, оскільки вихід із ладу кількох нейронів не приводить до руйнування мережі в цілому. Біологічна мережа відповідає за напідумки, емоції, здатність навчатися, за сенсомоторику та багато інших функцій людського організму.

Слід зазначити, що незважаючи на досягнення сучасної нейрофізіології та суміжних наук, дотепер не вивчений повною мірою принцип роботи нейронної мережі мозку людини, численні зв'язки якої мають зовні хаотичний характер. Біологічна нейронна мережа працює без процесорів і програм, без керуючих і тактових імпульсів, характерних для роботи комп'ютерних систем. Складні лінгвістичні й розпізнавальні задачі мозок вирішує за кілька ітерацій тривалістю кілька секунд. Сучасний же нейрокомп'ютер «*мислить*» на рівні 5-річної дитини, хоча й вирішує *обчислювальні задачі* набагато швидше, ніж людина. З огляду на це, трансформація можливостей мозку у створювані нейробіонічні конструкції являє собою досить складну наукову проблему.

10.2. Штучні нейрони

З погляду нейробіологічного підходу до побудови КСПШ (розд. 2.3) біологічний нейрон розглядається нині як *елементарний мікропроцесор*, здатний до найпростішого перероблення інформації. Таку модель біологічного нейрона називають *штучним (модельним, формальним) нейроном*. Уперше цю модель запропонували Мак-Каллок і Піттс [10.1], яку в їхню честь було названо *МР-нейроном* (рис. 10.2).

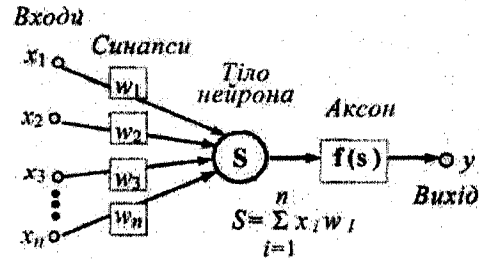


Рис. 10.2 — Схема штучного (модельного) МР-нейрона

Подібно до біологічного нейрона, *МР-нейрон* виконує прийом, елементарне перетворення й передавання інформації іншим нейронам. Подібно до біологічного нейрона, він має *n входів*, один *вихід y* і складається із блоку локальної *пам'яті*, яка зберігає *ваги синапсів* w_i ($i = 1...n$), *суматора S* і блоку нелінійного перетворення $f(s)$.

Вхідні (рецепторні) сигнали x_i ($i = 1...n$), які надходять через синапси з вагами w_i , перетворюються у нейроні у *зважені сигнали* $x_i \cdot w_i$. З них формується *сумарний сигнал*

$$s = \sum_{i=1}^n x_i \cdot w_i \quad (i = 1...n) \dots \quad (10.1)$$

Подібно до біологічного нейрона, штучний нейрон може також перейти у збуджений стан, якщо величина сумарного сигналу s стане більше за деяке граничне значення (*пориг*) s^* .

Блок *нелінійного перетворення* МР-нейрона описується так званою *активаційною функцією* $f(s)$, відповідно до якої нейрон формує *вихідний (ефекторний) сигнал* $y = f(s)$.

Активаційна функція $f(s)$ може мати різний вигляд (рис. 10.3). У штучних нейронних мережах здебільшого використовується *жорстка активаційна функція* (рис. 10.3, а, б), за якої нейрон *стрибкоподібно* переходить із *пасивного* стану (вихідний сигнал «0» або «1») у *збуджений* (вихідний сигнал «+1»). На рис. 10.3 величина визначає *пориг* спрацьовування нейрона, який може бути рівним нулю.

У сучасних нейромережах переважно використовується так звана *логістична активаційна функція*, яку називають також *сигмоїдом*. Останній являє собою функцію *S-образного* вигляду (рис. 10.3, в):

$$f(s) = 1 / (1 + e^{-as}). \quad (10.2)$$

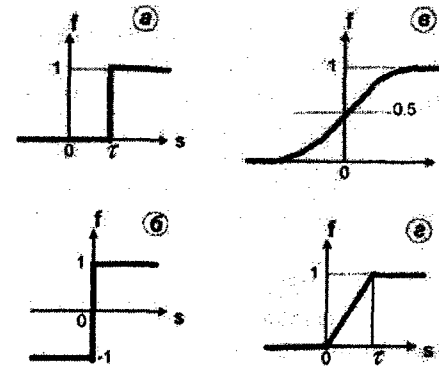


Рис. 10.3 — Активаційні функції

З виразу для *сигмоїдної функції* бачимо, що вона *диференційована* на всій осі абсцис, а її вихідне значення лежить у діапазоні $[0...1]$. При зменшенні параметра α *сигмоїд* стає більш *положистим*, вироджуючись при $\alpha = 0$ у *горизонтальну лінію* на рівні 0,5. При збільшенні α *сигмоїд* наближається за зовнішнім виглядом до *жорсткої активаційної функції* *одиночного стрибка* (рис. 10.3, а).

Сигмоїд має властивість *підсилювати* слабкі сигнали краще, ніж великі, й *запобігає* насиченню від великих сигналів, тому що вони відповідають областям аргументів, де *сигмоїд* має *положистий нахил*. Ці властивості *сигмоїда* широко використовують під час створення й дослідження нейронних мереж.

10.3. Принцип побудови та функціонування штучних нейронних мереж

Структурована *сукупність штучних (модельних) нейронів* у вигляді множини *мікропроцесорів*, з'єднаних між собою міжнейронними зв'язками, являє собою *штучну нейронну мережу (нейромережу)*. В основу побудови *нейромережі* закладена так звана *асоціативна модель* структури людського мозку, яка відображає систему його асоціативних зв'язків, що мають ймовірний характер [10.17].

Найважливіша особливість людського мозку полягає у тому, що один раз навчившись певного процесу, він може правильно діяти за *асоціацією* й у тих ситуаціях, з якими він не стикався під час навчання. Наприклад, ми можемо читати текст, написаний будь-яким почерком, навіть якщо бачимо його вперше. У процесі рішення інтелектуальної (творчої) задачі в нейронній мережі мозку утворюються *специфічно збуджені структури*, що визначають характер асоціативних зв'язків між фрагментами розв'язуваної задачі.

За аналогією з біологічною нейромережею, у *штучній нейромережі* встановлюється *безпосередній асоціативний зв'язок* між її входом і виходом. При поданні на вхід задачі, яку нейромережа раніше вже вирішувала й «запам'ятала», на її виході негайно з'являється

результат цієї задачі, одержуваний практично без будь-яких обчислень. Водночас традиційні (наприклад, експертні) інтелектуальні системи не володіють повною мірою такою здатністю і повинні витрачати час на вирішення однієї і тієї самої задачі, поставленої різними користувачами в різний час роботи системи.

Нейрони здатні запам'ятовувати свій стан і зберігати інформацію. Сукупність інформації, що зберігається в нейронній мережі, називають її *пам'яттю*.

Однією з найважливіших властивостей нейромережі є принцип *паралельного опрацювання інформації*, що надходить у мережу у вигляді сигналів. Паралельне опрацювання сигналів досягається шляхом об'єднання великого числа нейронів у так звані *прошарки* й з'єднання нейронів різних прошарків у різні конфігурації з *пошаровим* дослідженням взаємодії між нейронами.

Нейромережа має явно виражений динамічний характер, оскільки її стан можна розглядати як послідовність миттєвих знімків, у якій кожен новий стан залежить від попереднього циклу збудження нейронів. Для того, щоб нейромережа стала активною, вона повинна одержати певний *вхідний сигнал*. Тому деякі нейрони мережі відіграють роль «сенсорів» і їхня активність залежить від зовнішніх джерел інформації. Потім збудження передається від *вхідних прошарків* нейронів до *внутрішніх (прихованих) прошарків* і в такий спосіб сигнали збудження поширюються вздовж мережі до їх *вихідних прошарків* нейронів.

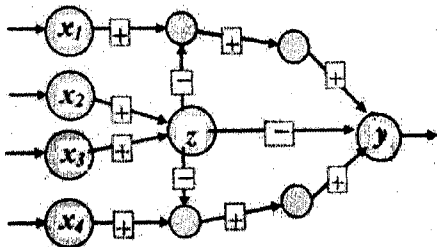


Рис. 10.4 — Фрагмент нейромережі

На рис. 10.4 наведено фрагмент нейромережі, до складу якої входять чотири *вхідних збуджених* (позначені знаком «+») нейронів $x_1 \div x_4$, кілька *прихованих* нейронів і *вихідний* нейрон y . Якщо ваги зв'язків у мережі невідомі, то нейрон y буде збуджений лише тоді, коли будуть збуджені тільки нейрони x_1 і x_4 . Але якщо бу-

дуть збуджені також нейрони x_2 і x_3 , то це призведе до пригнічення збудження прихованих нейронів і вихідного нейрона y *через* гальмуючу дію нейрона z (позначений знаком «-»). Чи буде вихід y збуджений за такого стану нейронів, залежить від сумарної *ваги зв'язків* між нейронами мережі.

Після закінчення процесу проходження сигналів через нейромережу визначаються *стани вихідних нейронів* і реєструються (зчитуються) *рівні їх збудження* (активності). За потреби фіксується також

стан усієї мережі й нейронів із високим рівнем активності. У деяких випадках інтерес може представляти спостереження за процесом переходу мережі у стабільний стан.

10.4. Методи навчання нейронних мереж

Процес функціонування нейронної мережі залежить, головним чином, від величин *синаптичних зв'язків* між її нейронами. Оскільки сумарна вага цих зв'язків суттєво впливає на поширення збудження нейронів мережі, то, змінюючи *вагу*, можна змінювати поведінку мережі у бажаному напрямі.

З огляду на це, основною проблемою, що виникає під час створення нейрокомп'ютерної системи, є визначення *оптимальних* значень *вагових коефіцієнтів синаптичних зв'язків* нейромережі заданої структури, орієнтованої на розв'язання визначеного класу задач. Наприклад, внесок кожної групи нейронів у загальну «оцінку» *розпізнавання* вхідного образу (розд. 8.2) можна встановити шляхом *регулювання ваг зв'язків*, які характеризують ступінь активізації тієї чи іншої ділянки нейромережі. При конструюванні нейронної мережі *ваги зв'язків* можуть бути визначені апіорі або змінюватися під час її роботи.

Процес адаптації нейромережі до розв'язання поставлених перед нею задач шляхом *модифікації й уточнення ваг зв'язків* між нейронами, називають *навчанням нейромережі*. Реалізація процесу настроювання ваг зв'язків у нейромережі, що складається з десятків і сотень тисяч нейронів, являє собою досить складну проблему. Тому на сьогодні створено багатьох різних типів *нейронних архітектур*, які за *типом навчання нейромереж* можна поділити на два великих класи: навчання «з учителем» та «без учителя».

10.4.1. Навчання «з учителем»

Навчання нейромережі «з учителем» здійснюється за допомогою так званої *навчальної вибірки*, що складається з пари «*вхідна ситуація — вихідний сигнал*». Ця пара містить у собі *вхідний образ* (зразок, шаблон, кортеж), що подається на вхід до мережі, та *вихідний (еталонний, цільовий, потрібний) образ*, який бажано одержати на виході мережі (розд. 8.2). *Повний набір навчальних вибірок*, що охоплюють проблемну область нейрокомп'ютерної системи, заздалегідь створюється *експертами* — фахівцями у даній Про, й заноситься в БЗ нейросистеми.

Ціль навчання полягає в тому, щоб при поданні на вхід нейромережі *вхідного образу* (зразка) одержати на її виході *необхідний (еталонний) образ* шляхом підстроювання ваги синаптичних зв'язків між нейронними прошарками мережі.

Наприклад, навчання нейромережі алфавіту відбувається у такий спосіб. Кожну букву алфавіту можна подати як растрове чорно-біле зображення розміром, наприклад, 30×30 пікселів (точок). Тоді нейромережа повинна буде мати $30 \times 30 = 900$ входів і 33 виходів, позначених літерами.

На вхід мережі подається зображення тієї чи іншої літери і мережа повинна її «запам'ятати». Якщо на вході мережі — зображення літери «А», то максимальний рівень вихідного сигналу повинне бути на заданому (бажаному) виході з позначкою «А». Обчислюючи різницю між заданим і дійсним рівнями вихідного сигналу, одержують *вектор похибок* (він складається із 33 чисел), який використовується для корегування ваги зв'язків нейронів мережі згідно з обраним *алгоритмом навчання*. У програмних реалізаціях можна побачити, що в процесі навчання величина похибок поступово зменшується.

Аналогічна процедура виконується для всіх 33 літер. Ту саму літеру, зазвичай, подають мережі кілька разів, тому навчання нагадує повторення вправ у спорті — *тренування*. Після багаторазового подання вхідних образів (у цьому випадку — літер) сумарна вага мережі стабілізується й мережа починає адекватно реагувати на запропоновані їй образи. Коли величина похибок досягає прийнятного *малого рівня*, тренування зупиняють, мережу вважають *натренованою* й готовою для розв'язання практичних задач.

10.4.2. Навчання «без учителя»

Результати спостережень за біологічними нейромережами вказують на наявність *множинних зворотних зв'язків* між нейронами у процесі ітераційної нейродинаміки перебігу біологічних процесів. Подібно до відомого принципу *самоорганізації* біологічної нейромережі, основним методом навчання *штучних нейромереж* є *змагальне навчання*, яке називають навчанням «без учителя» [10.33]. Цей метод ґрунтується на принципі так званого «*латерального гальмування*» нейронів мережі, який полягає в тому, що кожен нейрон певного прошарку одержує *гальмуючий (негативний)* вплив із боку інших нейронів цього прошарку (пропорційно рівню їхнього збудження) й зазнає *збуджуючого (позитивного)* впливу самого на себе.

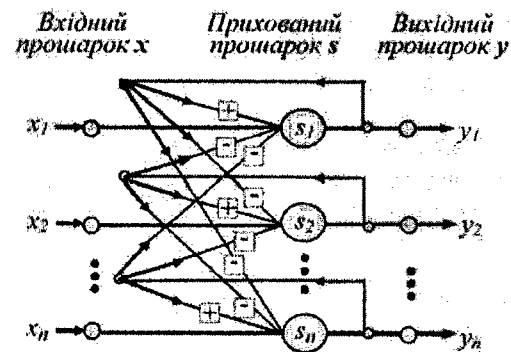


Рис. 10.5 — Фрагмент змагальної нейромережі із «латеральним гальмуванням»

даного нейрона, і він повторно виконує свою функцію. Нейродинаміка в таких мережах є *ітераційною*.

У змагальній мережі ваги *латеральних зв'язків* нормуються таким чином, щоб у кожному нейронному прошарку був збуджений тільки один нейрон з *максимальною активністю збудження*. У цьому разі інші нейрони даного прошарку зазнають *гальмування*.

Таким чином, нейрони кожного прошарку мережі начебто *змагаються* між собою за лідерство й найбільш активний нейрон стає *переможцем*. Він і вказує на той клас, до якого належить вхідний образ. Такий механізм навчання нейромережі одержав назву «*Переможець отримує все*» («Winner Take All» — «WTA»). Принцип *латерального гальмування*, на якому він базується, має біологічне обґрунтування й широко розповсюджений у нейронних мережах живих організмів.

При навчанні «без учителя» нейромережі пред'являються тільки *вхідні образи*, а виходи мережі формуються самостійно. При цьому *вагові коефіцієнти прихованих прошарків* змінюються згідно з алгоритмом, який враховує тільки *вхідні й проміжні* вектори сигналів, що відповідають вхідним образам мережі.

Використання в нейромережах *зворотних зв'язків* суттєво розширює різноманітність типів нейромережових архітектур. Однак зворотні зв'язки можуть призводити до виникнення *нестійкості* нейромереж подібно до тієї, що виникає у *кібернетичних* системах керування складними системами з *позитивним зворотним зв'язком*. У нейронних мережах нестійкість виявляється у *блукуючій зміні станів* нейронів, що призводить до виникнення *нестационарних станів*. Проблема динамічної стійкості довільної нейромережі зі зворотними зв'язками вкрай складна й досі мало вивчена.

Унаслідок навчання нейромережі автоматично будуються *узагальнені правила* або *функції*, які визначають належність образів (ситуацій) певним класам, якими навчена нейромережа буде користуватися при інтерпретації нових виникаючих ситуацій (розд. 8.2). Отримані *правила (функції)* становлять основу *БЗ нейрокомп'ютерної системи*, яка періодично корегується в міру нагромадження досвіду розв'язання практичних задач.

10.5. Типи нейронних мереж

Особливості побудови біологічних нейромереж обумовлюють необхідність використання в нейрокомп'ютерних системах складних *багатошарових* нейромережових структур. При цьому *n-шаровою*, зазвичай, називають нейромережу, яка містить у собі *n прихованих прошарків* (тобто *вхідний* і *вихідний* прошарки мережі в число *n* не входять). Наприклад, *двошарова* мережа має два прихованих прошарки, *тришарова* — три і т. д.

Залежно від способу внутрішнього подання інформації розрізняють *бінарні* й *аналогові* нейромережі. *Бінарні* оперують із *двійковими сигналами*, тобто кожен вхідний чи вихідний сигнал окремого нейрона може приймати дискретні значення «0» або «1». Якщо вихід нейрона дорівнює «0», то він перебуває в *загальмованому* стані, а якщо «1», то — у *збудженому*.

В *аналогових* мережах вхідні й вихідні сигнали нейронів можуть набувати неперервних (проміжних) значень у деяких (заздалегідь установлених) межах, наприклад, $[0...1]$ або $[a...b]$, де *a* і *b* — деякі цілі або натуральні числа.

10.5.1. Персептрон Розенблатта

Однією з найперших *аналогових* нейромереж, здатних до *перцепції (сприйняття)* й формування *реакції* на сприйнятий сигнал, був *персептрон Розенблатта* (рис. 10.6) [10.2, 10.3]. В основу функціонування *персептрона* покладена *асоціативна модель*, яка моделює асоціативні зв'язки між нейронами мозку людини (розд. 10.3). У процесі вирішення задачі у персептроні утворюються специфічно збуджені структури, які визначають характер одержуваного рішення. Зв'язки між нейронами встановлюються у процесі *навчання розпізнавання* вхідних образів (розд. 8.2). *Навчений персептрон* здатний розпізнавати подані на його вхід образи й відносити їх до того чи іншого класу.

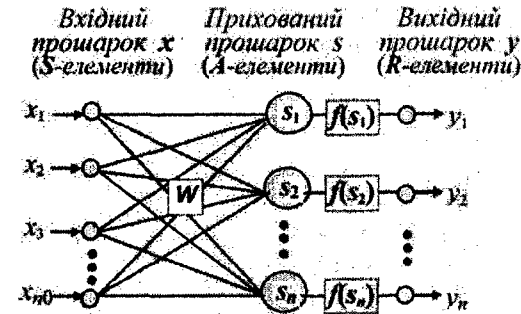


Рис. 10.6 — Персептрон Розенблатта

Персептрон складається зі штучних нейронів *трьох типів* (на рис. 10.4 позначені *кружками*) відповідно до нейронів біологічної (рефлекторної) нейронної мережі. *Вхідні S-елементи* являють собою *сітчатку сенсорних клітин*, які сприймають інформацію із зовнішнього світу. *Сітчатка* формує сенсорні вхідні сигнали *x*, які надходять до синапсів *асоціативних нейронів (A-елементів)* *s*, зв'язаних між собою міжнейронними зв'язками *W*. *Вихідні R-елементи* формують вихідні сигнали *y*, використовуючи *жорстку активаційну функцію* (10.2). Таким чином, персептрон Розенблатта є *одношаровою* нейромережею, оскільки він має *один прихований* прошарок нейронів *s*.

У персептроні Розенблатта використовується *навчання «з учителем»* (розд. 10.4), коли нейромережі пред'являються як вхідні образи (сигнали) *x*, так і вихідні еталонні образи *y**. *Алгоритм навчання* являє собою ітераційний процес, на кожному *k-му* кроці якого здійснюється *підстроювання (уточнення) матриці вагових коефіцієнтів* (далі — *ваг*) зв'язків *Wk* між нейронами мережі, які забезпечують *мінімальні значення похибок* ϵ_{ik} між *еталонними* y_i^* і *фактичними вихідними сигналами* y_{ik} , отриманими внаслідок дії вхідних сигналів x_{jk} ($j = 1...n_0$):

$$\epsilon_{ik} = y_i^* - y_{ik} \quad \min (i = 1...n, k = 1...) \dots \quad (10.3)$$

При цьому *початкові* значення W_0 ваг зв'язків можуть бути довільними.

Алгоритм навчання персептрона складається з таких *ітераційних кроків*.

1. На черговому *k-му* кроці на вхід мережі подається черговий образ у вигляді вектора сигналів x_k . У цьому разі на *виході* мережі (за даних значень ваг зв'язків W_k) з'являється відповідний вектор вихідних сигналів y_k .

2. Обчислюється вектор похибок ϵ_k (10.3). Якщо абсолютні значення похибок стануть меншими за деяку задану малу величину ϵ^* , тобто $|\epsilon_{ik}| < \epsilon^* (i = 1...n)$, то процес навчання на цьому закінчиться і персептрон вважатиметься *навченим*.

3. В іншому разі *уточнюються ваги зв'язків* W_k між нейронами згідно з ітераційною формулою:

$$w_{ijk+1} = w_{ijk} + h_k \epsilon_{ik} f(s_{ik}) x_{jk}, \quad (i = 1 \dots n, j = 1 \dots n_0, k = 1 \dots), \quad (10.4)$$

де: w_{ijk} — вага зв'язку j -го нейрона вхідного прошарку з i -м нейроном прихованого прошарку на k -й ітерації; x_{jk} — сигнал на вході j -го нейрона вхідного прошарку; ϵ_{ik} — i -та похибка (10.3); $f(s_{ik})$ — жорстка сигмоїдальна активаційна функція (10.2) у вигляді одиничного стрибка (див. рис. 10.3, а); s_{ik} — зважений сигнал (10.1) на виході i -го нейрона прихованого прошарку; h_k — параметр, що визначає швидкість навчання ($0 < h_k < 1$), n_0 — число нейронів вхідного прошарку; n — число нейронів прихованого і вихідного прошарку.

4. Отримані ваги зв'язків w_{ijk} устанавлюються в нейромережі і здійснюється перехід до $n1$ даного алгоритму.

Зазначена ітераційна процедура виконується для кожного зразка навчальної вибірки. При цьому параметр h_k зменшується на кожній ітерації на визначену (заздалегідь установлену) малу величину.

У Корнельській авіаційній лабораторії США було розроблено електротехнічну модель перцептрона *MARK-1* [10.2, 10.3]. Вона містила 512 вхідних і 8 вихідних елементів, які можна було з'єднувати в різних комбінаціях. На цьому перцептроні було проведено серію експериментів з розпізнавання літер та геометричних образів. Процес розпізнавання в перцептроні Розенблатта здійснювався з використанням спеціальних технічних засобів — фотоелементів, фотомасок тощо.

10.5.2. Багатошаровий перцептрон.

Алгоритм зворотного поширення похибки

Подальшим розвитком робіт зі створення нових типів нейромереж було вдосконалення одношарового перцептрона Розенблатта й створення на його основі *багатошарових повнозв'язних структур*, у яких нейрони кожного прошарку приймають й опрацьовують сигнали від нейронів попереднього прошарку за схемою «кожний із кожним» [10.4]. Нейромережу такої топології називають *n-шаровим перцептроном*, який містить у собі n прихованих прошарків. На рис. 10.7 представлена *тришарова нейромережа*, яка має вхідний, три прихованих і вихідний прошарки нейронів.

Принцип функціонування *багатошарового перцептрона* полягає у наступному. Кожен нейрон підсумовує сигнали, які надходять до нього від нейронів попереднього прошарку (із вагами, що визнача-

ються станами синапсів), й переходить у збуджений стан, якщо отримана сума вища, ніж певне граничне значення. Приховані прошарки мережі переводять вхідний сигнал у вихідний образ, що визначається нейронами вихідного прошарку. Стан збудження того чи іншого нейрона вихідного прошарку свідчить про належність вхідного образу до відповідної категорії.

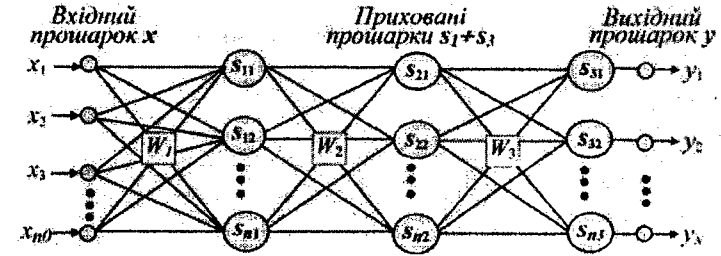


Рис. 10.7 — Схема тришарового перцептрона

Для навчання багатошарової мережі широко використовується так званий *алгоритм зворотного поширення похибки (error back propagation)*, який виконує навчання «з учителем» і дає змогу за вектором похибки *вихідного* прошарку мережі обчислити необхідні поправки для ваг нейронів прихованих прошарків [10.32]. Було встановлено, що похибка на виходах нейромережі суттєво залежить від величини синаптичного зв'язку між нейронами *вихідного* прошарку і відповідними нейронами *попередніх прихованих* прошарків. Таким чином, оцінки похибок для нейронів прихованих прошарків можна визначити, рухаючись від *виходів* мережі до її *входів*, тобто в напрямку, *зворотному* прямому поширенню сигналів у звичайному режимі роботи мережі, що й відображено в назві алгоритму.

Цільовою функцією в алгоритмі зворотного поширення похибки є мінімальне значення квадрата різниці між фактичним і еталонним виходами нейромережі для всіх вхідних образів навчання:

$$E(w) = \sum_p \sum_i (y_{pi}(w) - y_{pi}^*)^2 \quad \min, \quad (p = 1 \dots P_x, i = 1 \dots N_R), \quad (10.5)$$

де: $y_{pi}(w)$ — фактичний стан i -го нейрона *вихідного* прошарку нейромережі при подачі на її входи p -го образу; y_{pi}^* — еталонний (бажаний) стан цього нейрона; w — вектор, компонентами якого є матриці W_r ($r = 1 \dots R$) *ваг зв'язків* між нейронними прошарками мережі; R — загальне число прошарків мережі; P_x — число *образів (вибірок)* навчання; N_R — число нейронів у *вихідному* прошарку R .

Мінімізація цільової функції (10.5) здійснюється з використанням методу градієнтного спуску [8.32, 8.33], в якому підстроювання вагових коефіцієнтів здійснюється за формулою:

$$w_{rijk+1} = w_{rijk} - h_k \frac{\partial E}{\partial w_{rijk}}, \quad (r=1...R, i=1...n_{r-1}, j=1...n_r, k=0...), \quad (10.6)$$

де: w_{rijk} — вага зв'язку i -го нейрона $(r-1)$ -го прошарку з j -м нейроном r -го прошарку на k -й ітерації; h_k — параметр швидкості навчання; $\frac{\partial E}{\partial w_{rijk}}$ — часткова похідна цільової функції (сумарної похибки) по її аргументу, яка визначається таким чином (індекс k далі опустимо):

$$\frac{\partial E}{\partial w_{rij}} = \frac{\partial E}{\partial y_{ri}} \frac{\partial y_{ri}}{\partial s_{ri}} \frac{\partial s_{ri}}{\partial w_{rij}}. \quad (10.7)$$

Тут y_{ri} — вихід j -го нейрона r -го прошарку; s_{ri} — зважена сума вхідних сигналів цього нейрона (див. рис. 10.4):

$$s_{rj} = \sum_i i_j y_{r-1i}, \quad (i=1...n_{r-1}), \quad (10.8)$$

яка є аргументом активаційної функції (10.2):

$$y_{ri} = f(s_{ri}); \quad (10.9)$$

R — число прихованих прошарків нейромережі; n_r — число нейронів у r -м прошарку

Співмножники, що входять у співвідношення (10.7), можна перетворити таким чином. Перший співмножник, визначений для нейронів даного r -го прошарку, можна розкласти на складові для нейронів наступного $(r+1)$ -го прошарку [8.32]:

$$\frac{\partial E}{\partial y_{ri}} = \sum_q \frac{\partial E}{\partial y_{rq}} \frac{\partial y_{rq}}{\partial s_{rq}} w_{r+1iq}, \quad (q=1...n_{r+1})... \quad (10.10)$$

Шляхом введення нової змінної:

$$d_{ri} = \frac{\partial E}{\partial y_{ri}} \frac{\partial y_{ri}}{\partial s_{ri}} \quad (10.11)$$

одержимо рекурентну формулу розрахунку величини d_{ri} для даного r -го прошарку, виходячи з величини d_{r+1i} наступного $(r+1)$ -го прошарку (тобто у напрямку від виходу до входу нейромережі):

$$d_{ri} = \sum_q d_{r+1iq} w_{r+1iq} \frac{\partial y_{ri}}{\partial s_{ri}}, \quad (10.12)$$

у тому числі для вихідного прошарку $r = R$:

$$d_{Ri} = (y_{Ri} - y_i^*) \frac{\partial y_{Ri}}{\partial s_{Ri}}, \quad (i=1...N_R), \quad (10.13)$$

де N_R — число нейронів у вихідному прошарку мережі.

Формули (10.12) і (10.13) є основними в розглянутому алгоритмі зворотного поширення похибки, оскільки вони дають змогу за вектором похибки вихідного прошарку мережі обчислити необхідні поправки для ваг нейронів попередніх прихованих прошарків.

Другий співмножник, що входить у співвідношення (10.7) і (10.11), є похідною активаційної функції (10.9) по аргументу s_{ri} і у випадку сигмоїда (10.2) буде мати вигляд:

$$\frac{\partial y_{ri}}{\partial s_{ri}} = \frac{\partial f(s_{ri})}{\partial s_{ri}} = y_{ri}(1 - y_{ri}). \quad (10.14)$$

Третій співмножник у співвідношенні (10.7) відповідно до рівності (10.8) визначається таким чином:

$$\frac{\partial s_{ri}}{\partial w_{rij}} = y_{r-1i}. \quad (10.15)$$

З урахуванням отриманих співвідношень ітераційний процес (10.6) корекції ваг зв'язків між нейронними прошарками можна подати у вигляді:

$$w_{rijk+1} = w_{rijk} \eta_k \delta_{rik} y_{r-1ik}, \quad (r=1...R, i=1...n_r, j=1...n_{r-1}, k=1...). \quad (10.16)$$

Таким чином, алгоритм навчання нейромережі з використанням процедури зворотного поширення похибки містить у собі послідовність наступних ітерацій.

1. На черговій k -й ітерації на входи мережі подається один із образів у вигляді вектора вхідних сигналів $x_k = y_{0k}$. Визначається відповідний вектор вихідних сигналів y_{Rk} .

Якщо величина сумарної похибки E (10.7) стане меншою за деяку задану малу величину ε^* , то процес навчання на цьому закінчується. В іншому разі здійснюється перехід до п. 2.

2. Обчислюються величини δ_{rik} ($r=R, R-1, \dots, 2, 1$), ($i=1...n_r$) за формулами (10.13) і (10.12).

3. Визначаються (уточнюються) ваги зв'язків W_{rk} ($r=1...R$) між нейронами мережі відповідно до ітераційної процедури (10.16).

4. Отримані ваги зв'язків w_{ijk} установлюються в нейромережі та здійснюється перехід до п. 1 даного алгоритму.

Мережі на кроці 1 поперемінно, у випадковому порядку пред'являються усі тренувальні образи, щоб мережа, образно кажучи, «не забувала» одні образи в міру запам'ятовування інших образів.

Практика показує, що збіжність алгоритму зворотного поширення похибки в загальному випадку порівняно невелика. Це пояснюється відомою хворобою градієнтних методів, у яких локальний напрямок градієнта не завжди збігається із глобальним напрямком до мінімуму. Крім того, підстроювання ваг у цьому алгоритмі виконується незалежно для кожної навчальної вибірки. Тому спроби поліпшення функціонування мережі на даній вибірці можуть призвести до погіршення її роботи на інших (попередніх) вибірках.

10.5.3. Алгоритм навчання Хебба

Одним із найвідоміших методів навчання нейронних мереж «без учителя» є алгоритм навчання Хебба [1.1]. Він заснований на твердженні, що синаптичний зв'язок, що з'єднує два нейрони сусідніх прошарків нейромережі, буде поступово посилюватися у процесі навчання, якщо вони узгоджено будуть зазнавати збудження або гальмування. В алгоритмі Хебба уточнення ваг зв'язків між нейронними прошарками мережі здійснюється за таким правилом:

$$w_{rijk+1} = w_{rijk} + \eta_k y_{r-1ik} y_{rjk}, \quad (10.17)$$

$$(r = 1 \dots R, i = 1 \dots n_{r-1}, j = 1 \dots n_r, k = 1 \dots),$$

де: w_{rijk} — вага зв'язку i -го нейрона $(r-1)$ -го прошарку з j -м нейроном r -го прошарку на k -й ітерації; y_{rik} — вихід i -го нейрона r -го прошарку на k -й ітерації; η_k — параметр швидкості навчання; R — число прихованих прошарків нейромережі; n_r — число нейронів у r -му прошарку.

Існує також диференціальний алгоритм навчання Хебба:

$$w_{rijk+1} = w_{rijk} + \eta_k (y_{r-1ik+1} - y_{r-1ik}) (y_{rjk+1} - y_{rjk}), \quad (10.18)$$

в якому насамперед навчаються (корегуються) зв'язки між тими нейронами, виходи яких найбільше змінюються у бік збудження.

Алгоритм Хебба для навчання мережі окремого образу полягає у наступному.

1. На черговому k -му кроці на вхід мережі подається образ x_k .

2. Для кожного прошарку мережі розраховується зважена сума вхідів кожного нейрона прошарку (10.8), до якої потім застосовується активаційна функція (10.2). Як наслідок, будуть отримані вихідні вектори y_{rk} ($r = 1 \dots R$) усіх прошарків мережі.

3. Обчислюються різниці (похибки) між виходами нейронів кожного прошарку на суміжних k -й і $(k-1)$ -й ітераціях:

$$\epsilon_{rik} = (y_{rik} - y_{rik-1}) \quad (r = 1 \dots R, i = 1 \dots n_r). \quad (10.19)$$

Якщо абсолютні значення похибок ϵ_{ik} (10.19) у всіх прошарках стануть меншими, ніж деяка задана мала величина ϵ^* , тобто $|\epsilon_{ik}| < \epsilon^*$, то процес навчання мережі на цьому закінчиться, а отримані матриці ваг зв'язків W_{rk} ($r = 1 \dots R$), що відповідають даному образу, залишаються в пам'яті нейрокомп'ютера. Інакше здійснюється перехід до п. 4.

4. Визначаються (уточнюються) ваги зв'язків W_{rk} ($r = 1 \dots R$) між нейронами мережі згідно з ітераційними процедурами (10.17) або (10.18).

5. Отримані ваги зв'язків W_{rk} установлюються в нейромережі та здійснюється перехід до п. 1 даного алгоритму.

Розглянута процедура повторюється для кожного вхідного образу (зразка).

10.5.4. Нейромережа та картка Кохонена

Нейромережа Кохонена [10.5, 10.10, 10.25] складається з одного прошарку нейронів (рис. 10.8). Кількість вхідів кожного нейрона дорівнює розмірності вхідного образу. Кількість нейронів визначається тим рівнем деталізації, з яким потрібно виконати кластеризацію вхідних (бібліотечних) образів (розд. 8.2).

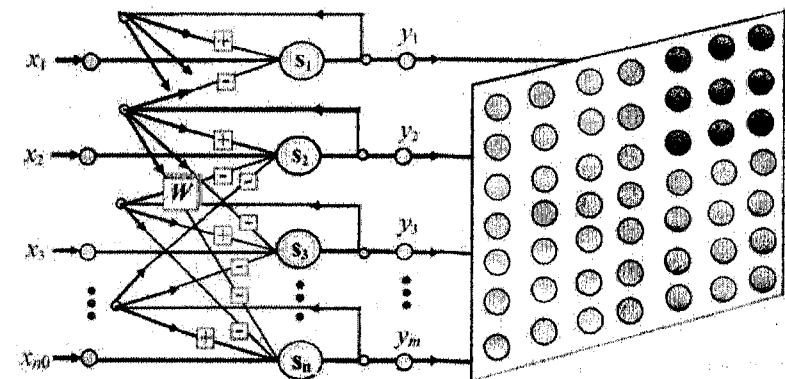


Рис. 10.8 — Схема нейромережі й карти Кохонена

Алгоритм навчання Кохонена використовує принцип «латерального гальмування» (розд. 10.4.2), що реалізується в такий спосіб.

Сигнали збудження при надходженні в мережу вхідного вектора впливають на ті нейрони прошарку, для яких зважена сума входів (10.1) перевищує граничне значення активаційної функції (10.2). Збуджені нейрони утворюють так звану *зону сусідства* прошарку.

Вводиться у розгляд *міра відстані* (за Кохоненом) між нейронами-сусідами, за яку, зазвичай, приймається *середньоквадратичне значення різниці* (похибки) між сумарним вхідним сигналом, що надходить до даного нейрона прошарку (від нейронів попереднього прошарку), і їх зваженими коефіцієнтами (див. *позначки* в (10.17)):

$$d_{rjk} = \sum_i (y_{r-1ijk} - w_{rij}k)^2, \\ (r = 1...R, i = 1...n_{r-1}, j = 1...n_r, k = 1...). \quad (10.20)$$

У поточному прошарку вибирається *нейрон j^** з мінімальною відстанню d_{rjk}^* , для якого виконується так звана *акредитація за Кохоненом*, яка полягає у введенні обраного *нейрона-переможця* у збудження і гальмуванні усіх інших нейронів даного прошарку згідно з принципом «*латерального гальмування*». Таким чином, у даному прошарку вибирається лише *один нейрон*, ваги синапсів якого максимально відповідають вхідному образу, і підстроювання ваг здійснюється тільки для нього. Навчання зводиться до мінімізації різниці між сумарним вхідним сигналом *нейрона-переможця* і вагами його зв'язків із виходами нейронів попереднього прошарку.

Таким чином, навчання мережі окремого образу, згідно з алгоритмом Кохонена, відбувається за такої послідовності.

1. На k -му кроці на вхід до мережі подається *вектор вхідних сигналів x_k* , який відповідає вхідному образу, і сигнали збудження поширюються по всіх прошарках мережі.

2. У кожному r -му прошарку мережі визначається *зона сусідства збуджених нейронів* прошарку, для яких зважена сума входів (10.1) перевищує граничне значення активаційної функції (10.2).

3. Розраховуються *відстані d_{ik}* (10.20) між нейронами-сусідами цього r -го прошарку і визначається *нейрон-переможець j^** із мінімальною відстанню d_{rjk}^* . Якщо ця відстань стане меншою за деяку задану малу величину ϵ (тобто $d_{rjk}^* < \epsilon$), то отримані матриці *ваг зв'язків W_{rk}* ($r = 1...R$), відповідні даному образу, залишаються в пам'яті нейрокомп'ютера. Далі на вхід до нейромережі подається *новий образ* і процес навчання продовжується, починаючи з п. 1 даного алгоритму (при $k = 0$ і $W_{rk} = W_{r0}$).

4. Виконується *акредитація нейрона-переможця j^** шляхом введення його в збудження й одночасне гальмування усіх інших нейронів даного прошарку згідно з принципом «*латерального гальмування*».

5. Здійснюється *модифікація (підстроювання) ваг зв'язків нейрона-переможця j^** із нейронами попереднього прошарку (див. *позначки* в (10.18)):

$$w_{rij}k+1 = w_{rij}k + h_k \cdot (y_{r-1j}k - w_{rij}k), \\ (r = 1...R, i = 1...n_{r-1}, k = 1...). \quad (10...21)$$

6. Отримані ваги зв'язків W_{rk} устанавлюються в нейромережі і здійснюється перехід до п. 1 алгоритму.

Унаслідок роботи алгоритму Кохонена вхідні образи розбиваються на *кластери*, упорядковану сукупність яких можна подати як *двовимірну матрицю*, що називається *картою Кохонена* (див. рис. 10.8). Кожен нейрон мережі Кохонена несе інформацію про свій *кластер вхідних образів*, формуючи для нього *узагальнений* (збірний) образ. Окремому кластеру можуть відповідати кілька нейронів із близькими значеннями векторів ваг, тому вихід із ладу одного нейрона не є критичним для функціонування мережі Кохонена.

При вирішенні практичних задач (наприклад, при розпізнаванні образів) навчена мережа відносить новий пред'явлений образ до одного зі сформованих кластерів, указуючи тим самим *категорію*, до якої він належить. За достатньої кількості нейронів і вдалим параметрів навчання мережа Кохонена може виокремити основні *групи близьких образів*, яким будуть відповідати *близькі карти нейронної активності*. На рис. 10.8 подано приклад *карти Кохонена*, на якій виокремлені окремі кластери з різним ступенем *нейронної активності* (відображені з різним ступенем згущення забарвлення).

На основі мережі Кохонена будуються так звані *самоорганізовані структури* (*self-organizing feature maps*). У таких нейроструктурах після акредитації *нейрона-переможця* навчаються також *нейрони-сусіди*, розташовані у *зоні сусідства*, які обираються для навчання у порядку збільшення відстані до нейрона-переможця. Величина цієї зони на перших кроках алгоритму вибирається достатньо великою, так що спочатку навчаються майже всі нейрони-сусіди, але з часом ця зона зменшується. Таким чином, що ближче закінчення навчання, то точніше визначається група нейронів, які відповідають певному *класу образів*.

10.5.5. Нейромережа Хопфілда

Нейромережа Хопфілда [10.6] призначена для моделювання *асоціативної пам'яті* й складається з *одного прошарку* нейронів, число яких n є одночасно числом *бінарних входів* і *виходів* мережі

(рис. 10.9). Кожен нейрон має один *вхідний синапс*, через який здійснюється введення сигналу. Аксон кожного нейрона зв'язаний з усіма іншими нейронами за схемою «кожний з кожним». Вихідні сигнали, зазвичай, утворюються на аксонах.

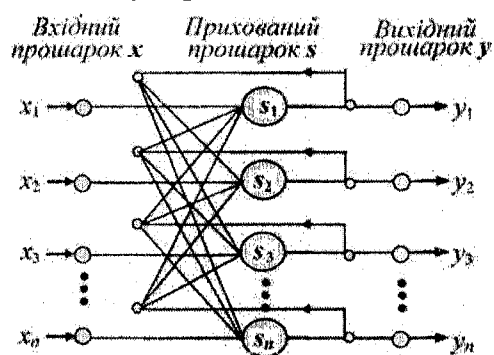


Рис.10.9 — Схема нейромережі Хопфілда

де P — кількість еталонних зразків; n — кількість нейронів у мережі, що дорівнює кількості вхідних і вихідних сигналів. Кожен елемент вектора y_p^* дорівнює або 1 (збуджений стан), або 0 (загальмований стан).

Мережа повинна вміти з довільного *неідеального сигналу* x_p , поданого на її вхід, виокремити («згадати») відповідний p^* -й зразок (якщо такий є). Коли мережа розпізнає який-небудь зразок $p = p^*$ на підставі заданих їй даних, її виходи y_p будуть містити саме його, тобто

$$y_{pj} = y_{pj}^*, (p = 1...P_x, j = 1...n)...$$

В іншому разі вихідний вектор не буде збігатися із жодним зразком. Відобразивши за допомогою засобів когнітивної графіки вихідні сигнали мережі, можна буде побачити картинку, що цілком збігається з одним зі зразків (у разі успіху) або ж «вільну імпровізацію» мережі (у разі невдачі).

На стадії *початкової ініціалізації* мережі Хопфілда ваги зв'язків w_{ij} між нейронами встановлюються таким чином:

$$w_{ij} = \sum_p i^* x_{pj}^* \text{ при } i \neq j; \quad w_{ij} = 0 \text{ при } i = j, \\ (p = 1...P_x, i = 1...n, j = 1...n),$$

де x_{pi}^* і x_{pj}^* — відповідно елементи векторів зразків, адекватні i -му (передсинаптичному) й j -му (постсинаптичному) нейронам мережі.

Задача моделювання асоціативної пам'яті, яку вирішує мережа Хопфілда, формулюється таким чином. Відомий деякий набір зразків y_p^* , поданих як вектори двоїчних сигналів, що вважаються еталонними:

$$y_p^* (y_{pj}^*), (p = 1...P, j = 1...n),$$

де P — кількість еталонних зразків; n — кількість нейронів у мережі, що дорівнює кількості вхідних і вихідних сигналів.

Алгоритм функціонування *мережі Хопфілда* при вирішенні задачі розпізнавання вхідного образу x_p полягає в наступному.

1. На черговому k -му кроці на вхід до мережі подається вектор вхідних сигналів x_k , рівнозначний вектору вихідних сигналів y_{k-1} , отриманому на попередньому $(k-1)$ -му кроці:

$$y_{k-1} = x_k.$$

2. Визначається *зважена сума входів* s_{jk} кожного нейрона:

$$s_{jk} = \sum_i i_{jk-1} y_{ik-1}, (i = 1...n, j = 1...n, k = 1...),$$

до якої застосовується *жорстка активаційна функція* (10.2). Як наслідок, буде отриманий вектор вихідних сигналів y_k на k -му кроці:

$$y_{jk} = f(s_{jk}) (j = 1...n, k = 1...).$$

3. Обчислюються *різниці (похибки)* між бінарними вихідними сигналами на k -му і попередньому $(k-1)$ -му кроках: $e_{jk} = (y_{jk} - y_{jk-1})$, $(j = 1...n)$. Якщо значення похибок e_{jk} дорівнюють нулю, то це означає, що отриманий вихідний вектор y_k відповідає даному зразку y_p^* й ітераційний процес розпізнавання вхідного сигналу x_p на цьому закінчується. Інакше здійснюється перехід до п. 2 даного алгоритму.

Необхідно зазначити, що мережа Хопфілда не завжди може провести розпізнавання й іноді видає на виході неадекватний образ. Це пов'язано із проблемою обмеженості можливостей мережі. Тому число розпізнаваних мережею Хопфілда образів, зазвичай, обмежують величиною $P < 0,15 n$. Крім того, якщо два образи y_1^* і y_2^* дуже схожі, то вони, можливо, будуть викликати в мережі *перехресні асоціації*, тобто пред'явлення на входи мережі вектора $x^* = y_1^*$ призведе до появи на її виходах вектора y_2^* й навпаки.

10.5.6. Нейромережа Хеммінга

Асоціативну пам'ять людини успішно моделює *нейромережа Хеммінга*, яка в порівнянні з мережею Хопфілда характеризується меншими витратами пам'яті та обсягом обчислень. Цю мережу використовують, коли немає необхідності одержувати в явному вигляді еталонний зразок і достатньо, наприклад, одержувати тільки номер цього зразка.

Нейромережа Хеммінга складається із двох прошарків (див. рис. 10.10). Число нейронів у кожному прошарку вибирається рівним числу класів еталонних зразків, так що кожен нейрон «відповідає» за свій власний клас зразків. Нейрони *першого прошарку* безпосередньо з'єднані з вхідними нейронами мережі (утворюючи фіктивний

нульовий прошарок) за схемою «кожний з кожним». Нейрони другого прошарку зв'язані між собою синаптичними негативними (інгібіторними) зворотними зв'язками. Один із синапсів кожного нейрона зв'язаний з його ж аксоном позитивним зворотним зв'язком.

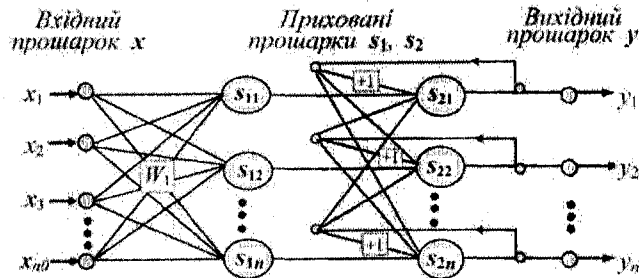


Рис.10.10 — Схема нейромережі Хеммінга

Функціонування мережі полягає у вирішенні задачі про належність невідомого вхідного образу до деякого класу образів (зразків). Ця задача вирішується шляхом обчислення так званих «відстаней Хеммінга» від тестованого образу, до всіх еталонних зразків. У бінарній мережі Хеммінга «відстань» між двома бінарними векторами визначається, як число компонентів, що не збігаються. Використовуючи принцип «латерального гальмування», мережа Хеммінга визначає «найближчий» еталонний зразок, який знаходиться на «мінімальній відстані» від тестованого образу. У результаті буде активізований тільки один вихід мережі, що відповідає цьому зразку.

На стадії ініціалізації мережі Хеммінга вагам зв'язків W_1 першого прошарку привласнюються значення:

$$w_{1pj} = x_{jp} / 2, \quad (j = 1 \dots n, \quad p = 1 \dots P),$$

де x_{jp} — j -й елемент p -го зразка; n — розмірність кожного зразка; P — число зразків.

Ваги w_{2ij} гальмуючих синапсів нейронів другого прошарку вибираються в межах:

$$0 < w_{2ij} < 1/P, \quad (i = 1 \dots n; \quad j = 1 \dots n; \quad i \neq j),$$

де n — число нейронів у другому прошарку та розмірність образів.

Синапс нейрона, зв'язаний позитивним зворотним зв'язком з аксоном того самого нейрона, має вагу +1. Поріг жорсткої активаційної функції (див. рис. 10.3? а) встановлюється рівним $\tau_p = n / 2$.

Алгоритм функціонування мережі Хеммінга при вирішенні задачі розпізнавання образу x_p полягає в такому.

1) На вхід мережі подається вектор вхідних сигналів x_p і розраховуються початкові ($k = 0$) стани нейронів першого прошарку:

$$y_{1j0} = s_{1j} = \sum_i i_{ij} + \tau_j, \quad (i = 1 \dots n, \quad j = 1 \dots n).$$

2. Отриманими значеннями y_{1p0} ініціалізуються виходи нейронів другого прошарку:

$$y_{2j0} = y_{1j0}, \quad (j = 1 \dots n).$$

3) На черговому k -му кроці обчислюються нові стани нейронів другого прошарку:

$$s_{2jk} = y_{1jk-1} - \sum_i i_{ij} y_{2jk-1}, \quad (i = 1 \dots n, \quad j = 1 \dots n).$$

4) Визначаються нові виходи нейронів (аксонів) другого прошарку:

$$y_{2jk} = f(s_{2jk}), \quad (j = 1 \dots n),$$

де $f(s_{2jk})$ — активаційна функція (див. рис. 10.3 а).

5) Обчислюються різниці (похибки) між виходами нейронів другого прошарку на k -му і $(k-1)$ -му кроках: $\epsilon_{jk} = (y_{2jk} - y_{2jk-1})$, $(j = 1 \dots n)$. Якщо значення похибок ϵ_{jk} дорівнюють нулю, то ітераційний процес розпізнавання вхідного образу x_p на цьому закінчується й здійснюється розпізнавання (класифікація) іншого образу, починаючи з п. 1. В іншому разі здійснюється перехід до п. 3 даного алгоритму.

Із приведенного алгоритму видно, що роль першого прошарку нейронів, зазвичай, умовна. Скориставшись на першому кроці значеннями його вагових зв'язків, мережа більше не звертається до нього. Тому перший прошарок у принципі може бути виключений із мережі і замінений на вихідну матрицю вагових коефіцієнтів W_1 .

10.5.7. Зірки Гроссберга

На ранніх етапах розвитку нейрокомп'ютерної техніки поряд із наведеними вище типами нейромереж було розроблено ієрархічні нейромережеві структури — вхідні і вихідні зірки Гроссберга [10.15]. Вхідна зірка Гроссберга G_s (див. рис. 10.11,а) являє собою нейрон, який має n входів x_j з вагами w_{xj} ($i = 1 \dots n$) і один вихід y , що є зваженою сумою входів:

$$y = \sum_i w_{xi} x_i.$$

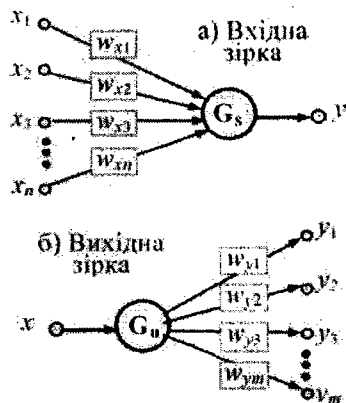


Рис. 10.11 — Зірки Гроссберга

Вихідна зірка Гроссберга G_U (рис. 10.11, б) має один вхід x і кілька виходів y_i з вагами w_{y_i} ($i = 1...m$)... Вона виконує функцію командного нейрона, видаючи на виходах відповідний вектор сигналів y при надходженні вхідного сигналу x . Процес навчання **вихідної зірки Гроссберга** можна подати як аналогічний ітераційний процес:

$$w_{yik+1} = w_{yik} + \eta_k (y_{ik} - w_{yik}), \quad (i = 1...m, k = 1... \infty),$$

де параметр η_k визначається аналогічно попередній процедурі. І ітераційний процес зводиться до *узагальненого образу*, адекватного сукупності навчальних векторів.

10.5.8. Нейромережа зустрічного поширення

Нейромережа *зустрічного поширення* (*counter propagation*) (див. рис. 10.12) складається із *двох прошарків* нейронів — *прошарку Кохонена* S і *прошарку зірок Гроссберга* G [10.33–10.45]. У режимі функціонування (розпізнавання) нейрони прошарку Кохонена працюють за принципом «*латерального гальмування*», визначаючи кластер, до якого належить вхідний образ x_p . Потім *вихідна зірка Гроссберга* за сигналом нейрона-переможця в прошарку Кохонена відтворює на виходах із мережі відповідний образ y_p .

Навчання ваг прошарку W_G Кохонена виконується «без учителя» на підставі *самоорганізації*. Вхідний (аналоговий) вектор x_p спочатку нормується, зберігаючи напрямок. Після виконання однієї ітерації навчання визначається нейрон-переможець, стан збудження якого встановлюється рівним «1», й модифікуються ваги W_G відповідної йому зірки Гроссберга. Темпи навчання нейронів Кохонена

Вхідна зірка навчається видавати сигнал y на виході щоразу, коли на входи надходить певний вектор сигналів x_j . Вхідна зірка пам'ятає «свій» образ, що належить до неї, й ігнорує інші вхідні образи. Процес навчання *вхідної зірки Гроссберга* являє собою таку ітераційну процедуру:

$$w_{xjk+1} = w_{xjk} + \eta_k (x_{jk} - w_{xjk}), \\ (j = 1...n, k = 1... \infty),$$

де параметр *швидкості навчання* η_k вибирається в межах ($0 < \eta_k < 1$), а потім поступово зменшується у процесі навчання.

й Гроссберга повинні бути погоджені. У прошарку Кохонена навчаються ваги всіх нейронів в околиці переможця, яка поступово звужується до одного нейрона.

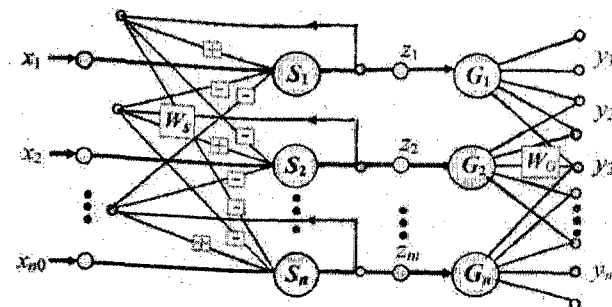


Рис. 10.12 — Схема нейромережі зустрічного поширення

Навчена мережа зустрічного поширення може функціонувати й у режимі *інтерполяції*, коли в прошарку Кохонена залишається не один, а кілька переможців. Тоді рівні їхньої активності пропорційно нормуються, щоб у сумі становити одиницю, а вихідний вектор визначається як сума вихідних векторів кожної з активних зірок Гроссберга. У результаті здійснюється лінійна інтерполяція між значеннями вихідних векторів, що відповідають кільком кластерам.

Однією з особливостей мережі зустрічного поширення є можливість визначення однієї з компонент навчальної вибірки, якщо відома інша її компонента. При пред'явленні тільки вхідної компоненти вибірки здійснюється її пряме розпізнавання на виході мережі. І навпаки, за відомого виходу може бути відновлений відповідний йому вхідний вектор.

Нейромережа зустрічного поширення, зазвичай, використовується для *швидкого моделювання* нейросистем на початкових етапах навчання з подальшим переходом (якщо це потрібно) на складніший, але точніший метод навчання зі зворотним поширенням похибки.

10.5.9. Когнітрон Фукушими

Когнітрон Фукушими являє собою *самоорганізовану багатoshарову* нейронну мережу, структура якої подібна багатoshаровому *персептрону* (див. рис. 10.7). Створення *когнітрона* — це плід спільних зусиль нейрофізіологів, психологів і нейрокібернетиків, які досліджували функціональні *системи сприйняття* людиною зовнішнього світу [10.21–10.29, 10.41–10.45].

Когнітрон складається зі зв'язаних прошарків нейронів, які мають *гальмуючі* та *збуджені* входи. Кожен прошарок когнітрона реалізує свій рівень перетворення інформації. *Вхідний прошарок* чуттєвий до окремих елементів вхідних образів, наприклад, до ліній певної орієнтації або кольору. Наступні прошарки реагують на складніше узагальнені образи. Нейрони вихідного прошарку визначають результат роботи нейромережі — *розпізнавання* певного образу. Для кожного нового образу картина активності вихідного прошарку буде унікальною, однак вона збережеться й у разі пред'явлення перекрученої або зашумленої версії цього образу. Таким чином, робота когнітрона здійснюється з використанням процедур *асоціації* й *узагальнення* опрацьовуваної інформації.

Нейрони двох *сусідніх прошарків* когнітрона зв'язані синаптичними (*пресинаптичними* та *постсинаптичними*) зв'язками. Постсинаптичні нейрони зв'язані не з усіма нейронами пресинаптичного прошарку, а лише з тими, що належать деякій локальній області зв'язків, яка називається *областю конкуренції* нейронів. Ці області близьких один до одного постсинаптичних нейронів перекриваються, тому активність пресинаптичного нейрона впливає на дедалі більше число нейронів наступних прошарків.

Ступінь активності кожного нейрона визначається станом його *збуджених* і *гальмуючих* входів. Пресинаптичні *гальмуючі* нейрони мають ту саму область зв'язків, що й відповідний їм *збуджений* постсинаптичний нейрон. При цьому ваги зв'язків *гальмуючих* нейронів задаються заздалегідь і не змінюються під час навчання. Їх сума дорівнює одиниці, тому активність *гальмового* пресинаптичного нейрона визначається середньою активністю *збуджених* пресинаптичних нейронів у їх локальній області зв'язків.

Сумарний вплив x_{sp} на вхід *збудженого постсинаптичного* нейрона визначається співвідношенням: $x_{sp} = [(1 + s_p) / (1 + s_f)] - 1$, де s_p і s_f — сумарні сигнали на його *збуджених* і *гальмуючих* входах відповідно. Якщо $x_{sp} > 0$, то вихідна активність цього нейрона y_p встановлюється рівною x_{sp} ; інакше вихід y_p приймається рівним нулю. Такі вихідні реакції відповідають реакціям біологічних нейронів, здатних працювати в широкому діапазоні зовнішніх впливів.

Навчання ваг *збуджених* нейронів когнітрона відбувається за принципом *латерального гальмування*, тобто у кожному прошарку модифікуються тільки ваги нейрона (чи нейронів) із максимальним збудженням. Ваги w_i *гальмуючого* нейрона постсинаптичного прошарку модифікуються пропорційно відношенню суми сигналів на *збуджених* входах s_p до сумарного сигналу на *гальмуючих* входах s_f . Така процедура навчання призводить до подальшого росту збудже-

них зв'язків активних нейронів і гальмуванню пасивних нейронів. При цьому ваги кожного з нейронів постсинаптичного прошарку настроюються на деякий образ, який часто подається під час навчання. Нове подання цього образу приведе до високого рівня збудження відповідного нейрона. У разі появи інших образів його активність буде пригнічена при *латеральному гальмуванні*.

Ваги нейрона-переможця, який здійснює латеральне гальмування в постсинаптичному прошарку, не піддаються модифікації. В області конкуренції цього нейрона знаходиться відносно невелике число інших нейронів, тому окремий нейрон-переможець не може здійснити гальмування всього прошарку. Отже, у конкурентній боротьбі можуть виграти кілька нейронів постсинаптичного прошарку, забезпечуючи більш повне й надійне опрацювання інформації у когнітроні.

Когнітрон Фукушими застосовувався автором для оптичного розпізнавання арабських цифр. В експериментах використовувалася 4-шарова мережа з чотирма прошарками нейронів, упорядкованими в матрицю з квадратною областю зв'язків кожного нейрона й областю конкуренції у формі ромба з висотою й шириною 5 нейронів. У результаті було здійснено успішне навчання когнітрона на п'ятьох образах арабських цифр. При цьому було здійснено близько 20 циклів навчання для кожного образу.

До переваг когнітрона можна віднести відповідність його структури та механізмів навчання біологічним моделям, паралельність та ієрархічність опрацювання інформації, розподільність й асоціативність пам'яті тощо. До недоліків когнітрона належить його нездатність до *інваріантного одномоментного розпізнавання образів* при зміні їх положення, орієнтації, розміру та інших деформаціях.

10.5.10. Неокогнітрон Фукушими

Для усунення недоліків когнітрона Фукушими провів дослідження зі створення *неокогнітрона*, який зазнав значних змін і ускладнень у порівнянні з когнітроном, і здатний до *інваріантного розпізнавання образів* [10.42–10.45].

Неокогнітрон складається з ієрархії нейронних прошарків, кожний з яких містить кілька *масивів площин*. Кожен масив являє собою *пару площин* нейронів. *Перша площина* складається з так званих *простих* нейронів, які одержують сигнали від попереднього прошарку й виокремлюють певні образи. Ці образи далі обробляються *складними* нейронами *другої площини*, основна задача яких полягає в тому, щоб зробити виокремлені образи менш залежними від їхнього положення.

Нейрони кожної пари площин навчаються реагувати на певний образ, поданий у певній орієнтації. Для іншого образу чи для нового кута повороту образу потрібна нова пара площин. Таким чином, за значних обсягів інформації неокогнітрон являє собою величезну структуру з великим числом прошарків і площин нейронів.

Прості нейрони чутливі до невеликої області вхідного образу, яка називається *рецептивною* (зв'язковою) областю. Простий нейрон збуджується, якщо в його рецептивній області виникає певний образ. Рецептивні області простих нейронів перекриваються й покривають усе зображення образу. Складні нейрони одержують сигнали від простих нейронів. При цьому для збудження складного нейрона досить одного сигналу від будь-якого простого нейрона. У такий спосіб складний нейрон реєструє певний образ незалежно від його розташування.

Навчання неокогнітрона аналогічно навчанню когнітрона. Змінюються тільки синаптичні ваги простих нейронів. Гальмуючі нейрони замість середньої активності нейронів в області зв'язків використовують *квадратний корінь зі зваженої суми квадратів вхідних сигналів*. Така формула визначення активності гальмуючого нейрона менш чутлива до *розміру* образу. Після вибору простого нейрона-переможця, вага якого буде уточнюватися, він розглядається як представник прошарку і всі інші нейрони прошарку навчаються за тими самими правилами. Таким чином, усі прості нейрони навчаються однаково, видаючи при розпізнаванні одну і ту саму реакцію на однакові образи.

Процес функціонування неокогнітрона відбувається в такий спосіб. Копії вхідного зображення надходять на всі площини простих нейронів першого прошарку. Далі всі площини функціонують паралельно, передаючи інформацію наступному прошарку. В міру поширення інформації від прошарку до прошарку картина нейронної активності стає все менш чутливою до орієнтації, розташування й розміру образу. Для зменшення обсягу опрацьовуваної інформації рецептивні поля нейронів під час переходу від прошарку до прошарку розширюються, а число активних нейронів зменшується.

При досягненні вихідного прошарку, в якому кожна площина містить один нейрон, виникає остаточне інваріантне розпізнавання образу. У вихідному прошарку на кожній площині залишається тільки один нейрон, рецептивне поле якого покриває всі поля образу попереднього прошарку. На результат розпізнавання вказує той нейрон, активність якого виявилася максимальною. У цьому разі істотно різним вхідним зображенням значною мірою будуть відповідати різні результати розпізнавання.

Неокогнітрон успішно проявив себе при розпізнаванні образів. Однак структура цієї нейромережі досить складна і обсяг обчислень дуже великий, тому комп'ютерні моделі неокогнітрона є занадто дорогими для промислового впровадження.

10.5.11. Нейронні мережі з адаптивним резонансом Проблема стабільності-пластичності нейромереж

Характер *сприйняття* людиною зовнішнього світу постійно пов'язаний з вирішенням дилеми — чи є певний образ *новою інформацією* або цей образ є варіантом *старої*, уже знайомої картини. У першому випадку реакція людини на образ повинна відповідати раніше нагромадженому досвіду, тобто повинна бути *пошуково-пізнавальною* з наступним збереженням цього образу в пам'яті. В іншому випадку спеціальне запам'ятовування цього образу не потрібно.

Таким чином, сприйняття людини є *пластичним*, тобто здатним до адаптації стосовно нової інформації, і при цьому воно *стабільне*, тобто не руйнує пам'ять про старі образи. З огляду на це, одним із основних напрямів нейробіологічних досліджень є розв'язання *проблеми стабільності-пластичності* нейронних систем [10.21–10.29, 10.41–10.45].

Розглянуті вище нейромережі не пристосовані до вирішення цієї задачі. Так, наприклад, багатошаровий персептрон, що навчається за методом зворотного поширення похибки, запам'ятовує весь пакет навчальної інформації і в цьому разі образи навчальної вибірки пред'являються у процесі навчання багаторазово. Спроби згодом навчити персептрон нового класу образів призводять до модифікації синаптичних зв'язків із неконтрольованим, у загальному випадку — руйнування структури пам'яті про попередні образи. Таким чином, персептрон не здатний до запам'ятовування нової інформації й необхідно повне перенавчання мережі.

Аналогічна ситуація має місце й у нейромережах Кохонена і Хеммінга, які навчаються на основі самоорганізації. Ці мережі застосовують, головним чином, при вирішенні задач класифікації. Однак вони не спроможні відокремити нові образи від перекручених чи зашумлених версій старих образів.

Характеристика адаптивного резонансу нейромережі

Дослідження в області *стабільності-пластичності* нейромереж призвели до створення *теорії адаптивного резонансу (АРТ)* й побудови *однойменних нейромережеских архітектур нового типу АРТ* на основі цієї теорії [10.41–10.45].

Адаптивним резонансом називають збуджений стан нейромережі АРТ, який виникає внаслідок її адекватного реагування на пред'явлений образ. Нейронні мережі з адаптивним резонансом зберігають *пластичність* при запам'ятовуванні нових образів і *стабільність* при збереженні старої пам'яті. У складі нейромережі АРТ є так званий *детектор новизни*, який містить *тест новизни* й виконує пошук і порівняння пред'явленого образу з набором образів, які знаходяться у пам'яті нейромережі. У разі вдалого пошуку пред'явлений образ класифікується (тобто визначається його клас чи кластер) і одночасно модифікуються (уточнюється) синаптична вага між-нейронних, властивих даному класу, зв'язків у мережі. Таку ситуацію називають виникненням *адаптивного резонансу* в мережі у відповідь на пред'явлений образ. Якщо ж резонансу не виникає, то *детектор новизни* сприймає образ як новий й мережа за допомогою *тесту новизни* формує *новий клас образів*, який надалі буде модифікуватися й уточнюватися під час дослідження інших образів. Для відображення нового класу образів використовується новий, раніше не задіяний нейрон у прошарку розпізнавання. Модифікація ваг нейронів, які не зазнали резонансу, в цьому разі не здійснюється.

Характерною рисою нейромережі з адаптивним резонансом є присутність у ній так званого *шаблону критичних рис* інформації. Його функцією є визначення рис (деталей), поданих у деякому образі, є суттєвими для системи сприйняття. Результат класифікації визначається наявністю специфічних критичних особливостей в образі. Цей механізм у теорії АРТ одержав назву *адаптивної фільтрації* класифікованого образу.

Нейромережа АРТ-1

Історично першою було створено *бінарну нейромережу АРТ-1* [10.15], призначену для класифікації *двійкових образів*. До складу мережі АРТ-1 входять (див. рис. 10.13):

- *прошарок нейронів порівняння S*;
- *прошарок нейронів розпізнавання R*;
- два спеціалізованих нейрони *керування u1* і *u2*;
- нейрон *призупинення активності* мережі *u0*.

Кількість *нейронів n* у кожному прошарку мережі дорівнює *кількості образів n*, що класифікуються.

На *вхід* кожного нейрона s_i ($i = 1...n$) *прошарку порівняння S* надходить *три двійкових сигнали*:

- сигнал x_i від відповідної компоненти вхідного вектора x ;
- сигнал u_1 від нейрона керування u_1 ;
- сигнал зворотного зв'язку u_r із прошарку розпізнавання R .

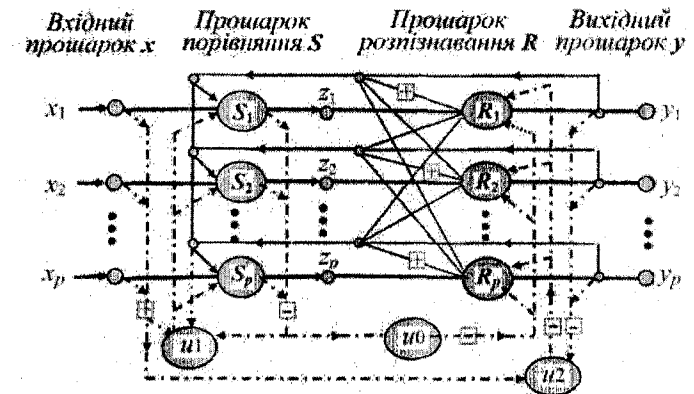


Рис. 10.13 — Схема нейронної мережі АРТ-1

→ Основні сигнали Сигнали керування --->

Для збудження (активації) кожного нейрона у прошарку порівняння S потрібно, щоб на *двох із трьох* його входів сигнали дорівнювали «1». Цю вимогу називають *правилом «двох третин»* (2/3) активації нейрона.

Стан збудження відповідного i -го нейрона r_i ($i = 1...n$) у *прошарку розпізнавання R* свідчить про належність вхідного образу x до відповідної i -ї категорії (класу).

Нейрон керування u_1 виконує функцію *ресстрації* факту надходження на вхід до мережі класифікованого образу. При цьому *вихід* нейрона u_1 набуває значення «1». Задачею нейрона керування u_2 є *пригнічення активності* нейронів прошарку розпізнавання за *невдалої класифікації* образу. Нейрон *призупинення* u_0 здійснює *гальмування активації* нейронів у прошарку розпізнавання, якщо *ступінь подібності* між вхідним вектором x і вихідним вектором z прошарку порівняння S менше деякого заданого значення *параметра подібності*.

Функціонування мережі АРТ-1 у процесі класифікації образів здійснюється таким чином.

Початковий стан мережі. За відсутності вхідного сигналу x виходи нейронів керування u_1 і u_2 , а також нейронів прошарків порівняння S і розпізнавання R спочатку встановлюються в «0». При надходженні на вхід мережі вхідного двійкового образу x виходи нейронів керування u_1 і u_2 набувають значення «1». Вхідний вектор x надходить у прошарок *порівняння S*, який спочатку пропускає його *без зміни*, тобто виходи z нейронів прошарку *порівняння* будуть при цьому в точності збігатися з компонентами вхідного вектора x .

Вихідний вектор z прошарку порівняння надходить на входи нейронів прошарку розпізнавання R , які згідно з принципом *латерального гальмування* в конкурентній боротьбі визначають *нейрон-переможець*, ваги синапсів w_y якого найближчі до ваг вектора сигналів z . У підсумку один елемент y^* вихідного вектора y , що відповідає нейрону-переможцю, набуває значення «1», а інші елементи вектора y дорівнюватимуть «0». Вихідний вектор сигналів y^* нейрона-переможця надходить назад у прошарок порівняння й одночасно встановлює в «0» сигнал керування u_1 .

Фаза порівняння. У прошарку порівняння S вектор сигналів зворотного зв'язку $y_s = y$ порівнюється з компонентами вектора x . Відповідно до правила «*двох третин*» вихід прошарку порівняння z буде тепер містити одиничні елементи тільки в тих позиціях, в яких одиниці є у вхідного вектора x та вектора зворотного зв'язку y_s . Таким чином, вихід прошарку порівняння z тепер уже не збігається в точності зі вхідним вектором x , а містить лише ті елементи x , які відповідають *критичним рисам* класифікованого образу.

Тепер необхідно встановити, чи достатній набір цих критичних рис для остаточного віднесення образу x до категорії нейрона-переможця. Цю функцію здійснює *нейрон призупинення* u_0 , що визначає подібність між векторами x і z . Вихід u_0 нейрона сброса визначається відношенням числа одиничних елементів вектора z до числа одиничних елементів вхідного образу x . Якщо це відношення менше деякого заданого *параметра подібності*, то нейрон u_0 видає *сигнал призупинення* u_0 , який означає, що рівень резонансу образу x із рисами пропонованої категорії недостатній для позитивного висновку про завершення класифікації. Сигнал призупинення u_0 загальмує активацію нейрона-переможця і тим самим почне *фазу пошуку іншого нейрона* в прошарку розпізнавання R .

Якщо ж у результаті порівняння між елементами векторів z і x не буде виявлено відмінностей, то вектор z знову викликає збудження *того ж самого* нейрона-переможця в прошарку розпізнавання, що успішно й *завершить процес класифікації*.

Фаза пошуку. У результаті дії гальмуючого сигналу u_0 нейрона u_0 виходи у нейронів прошарку розпізнавання R устанавляються в «0», а вихід u_1 нейрона керування u_1 набуває значення «1». Знову вихідний вектор сигналів z прошарку порівняння встановиться рівним у точності вектору x , як і на початку роботи мережі. Однак тепер у конкурентній боротьбі у прошарку розпізнавання попередній нейрон-переможець не бере участі й буде знайдений новий кандидат, що відповідає іншій категорії образів. Після цього знову повторюється фаза порівняння.

Ітераційний процес пошуку завершується двома можливими результатами. Якщо буде знайдена одна з категорій образів, подібність якої з вхідним вектором x буде достатньою для успішної класифікації, то процес пошуку закінчується й виконується *навчання* мережі шляхом модифікації ваг синапсів w_y^* збудженого нейрона-переможця y^* , який здійснив класифікацію.

Якщо ж у процесі пошуку жодна з категорій не дала необхідної подібності, то вхідний образ x вважається *новим* для мережі й у прошарку порівняння S і розпізнавання R додаються по одному *новому нейрону*. При цьому ваги синапсів w_y^* нового нейрона прошарку розпізнавання R встановлюються рівними ваговим значенням вектора x , а сигнал нейрона керування u_1 набуває значення «1».

Навчання мережі. На початку функціонування ваги нейронів w_y і *параметр подібності* p_s одержують деякі початкові значення. Величина p_s залежить від вимог, що ставляться до розв'язуваної задачі. За великих значень p_s буде сформована множина категорій із великим ступенем подібності класифікованих образів. За низького рівня p_s мережа сформує невелику кількість категорій із високим ступенем їх спільності.

Процес навчання відбувається «*без учителя*» на основі самоорганізації. Навчання здійснюється для ваг нейрона-переможця у випадку як успішної, так і неуспішної класифікації. Вектор ваг w_y , власне кажучи, є носієм критичних рис категорії, обумовлених нейроном-переможцем.

Елементи вектора w_y при навчанні встановлюються рівними відповідним значенням вектора z . Якщо якийсь із його елементів одержав значення «0», то під час подальшого навчання воно залишиться незмінним. Таким чином, навчання супроводжується *збільшенням числа нульових елементів вектора w_y* , а число *ненульових елементів* визначає сукупність *критичних рис* цієї категорії.

Нейромережі АРТ-2 і АРТ-3

Бінарна нейромережа АРТ-1 пристосована до роботи тільки із *двійковими (бітовими)* сигналами. Цей недолік усунено у *нейромережі АРТ-2*, де можливість роботи як із двійковими, так і з *аналоговими образами*. Основною відмінністю аналогових сигналів від бітових є принципова можливість аналогових векторів бути наближеними один до одного, з будь-якою похибкою, тоді як простір бітових векторів є дискретним. Це обумовлює функціонування нейронів прошарку порівняння, оскільки потрібен тонший і чуттєвіший механізм для виділення *областей резонансу*. Ця задача вирішується в мережі АРТ-2 шляхом переходу до *багатшарової архітектури* з якомога точнішим

настроюванням при переході від прошарку до прошарку. У цьому разі функціонування прошарку розпізнавання принципово не змінюється.

Особливості навчання нейронів мереж АРТ-1 і АРТ-2 не дають змогу використовувати їх як компоненти більш великих ієрархічних нейромереж. Це утруднює подання в мережі ієрархічно організованої інформації, характерної для системи сприйняття людини. Крім того, в АРТ-1 і АРТ-2 недостатній рівень резонансу призводить до генерації сигналу призупинення й наступному повному гальмуванню нейронів прошарку розпізнавання.

Зазначені недоліки усуваються в *нейромережі АРТ-3*, в якій при переході від прошарку до прошарку відбувається *контрастування* вхідних образів й запам'ятовування їх у вигляді дедалі загальніших категорій. При цьому основною задачею кожного нейронного прошарку є стиснення вхідної інформації. Образ входить в адаптивний резонанс між деякою парою прошарків, а потім поширюється на наступні прошарки ієрархії.

У мережі АРТ-3 уведений спеціальний механізм залежності активності синапсів зворотних зв'язків від часу, аналогічний *рефрактерному гальмуванню біологічного нейрона* після передачі збудження. У цьому разі замість повного призупинення сигналу відбувається гальмування синаптичних сигналів зворотного зв'язку й прошарок порівняння одержує вихідний стан збудження для виконання фази пошуку нового резонансу.

Одним із недоліків нейромереж АРТ є *локалізованість* їх пам'яті, яка не є *розподіленою*, як у біологічних системах, оскільки заданій категорії образів відповідає цілком конкретний нейрон прошарку розпізнавання. При зміні образів *губиться пам'ять* про всі категорії. Ця особливість не дає змоги вважати нейромережі з адаптивним резонансом цілком адекватними біологічним нейромережам, пам'ять яких є принципово *розподіленою*.

10.6. Проблемні області застосування нейрокомп'ютерних систем

Нейрокомп'ютерні системи (НКС), створювані на основі штучних нейронних мереж, з успіхом вирішують безліч інтелектуальних задач, які виникають у різних областях людської діяльності (розд. 3.6). Нижче наведені основні *характеристики деяких типових задач, які розв'язуються у НКС* [10.14–10.45].

1. *Ідентифікація* є однією з основних задач НКС, яка формулюється наступним чином (розд. 8.2). У БЗ НКС є *еталонний набір*

образів (зразків), які відображають *об'єкти* певної Про. Системи пред'являється довільний *неідеальний образ* й НКС повинна його *розпізнати*, тобто «згадати» й видати на виході зображення *еталона*, що відповідає вхідному образу. Навчена НКС може з великою ймовірністю правильно реагувати на нові, не пред'явлені їй раніше образи.

Наприклад, при вирішенні *задачі ідентифікації рукописного тексту* на вхід НКС подаються зображення текстових (літерних) *фрагментів*, написаних різними почерками. Система перетворює зображення літер *фрагмента* у *вектор* двійкових або аналогових сигналів й визначає, які це літери. Якщо на окремий вхід системи надійшла *рукописна літера «А»*, то максимальне значення вихідного сигналу досягається на *заданому виході* системи з позначкою «А». У НКС зберігається багато інформації про подібність або відмінність літер алфавіту, тому можна розраховувати на правильну відповідь для будь-якого варіанта вхідного образу.

Якщо на *вхід* системи подане зображення якого-небудь *об'єкта*, то, відобразивши *вихідні дані* НКС за допомогою засобів *когнітивної графіки*, можна буде побачити образ, що цілком збігається (у разі успіху) з одним зі *зразків (еталонів)*, або ж «вільну імпровізацію» системи у разі невдачі.

НКС успішно застосовуються при *ідентифікації сильно зашумлених образів* завадами, а також в умовах неповної інформації про об'єкти.

2. *Класифікація образів* являє собою різновид задачі *розпізнавання* й полягає у віднесенні *розпізнаваних образів* до тих чи інших *класів*, у кожному з яких образи групуються відповідно до визначених формальних *ознак* або *правил* (розд. 8.2). У процесі навчання НКС розпізнавання вхідних ситуацій на її вхід подається серія *позитивних і негативних образів*, які належать або не належать цьому класу.

Характерним прикладом такої задачі розпізнавання є *класифікація* хімічних речовин відповідно до їх властивостей та типів хімічних реакцій. Традиційною областю застосування *НКС-класифікаторів* є фізика високих енергій, де вони розв'язують проблему поділу на класи багатьох експериментально зареєстрованих фізичних явищ та елементарних частинок. До відомих застосувань НКС належать *класифікація мовних діалектів, сигналів електрокардіограм, клітин крові тощо*.

3. *Категоризація* є різновидом алгоритму *класифікації*, яка на основі *узагальнених (концептуальних) властивостей образів* і зв'язків між ними відносить їх до тієї чи іншої *категорії*. Задача *категоризації* менш визначена, ніж задача віднесення образу до того чи іншого класу, оскільки не завжди вдається визначити чіткі межі різних категорій об'єктів. Тому методи *категоризації* оперують,

головним чином, із *нечіткими знаннями* про об'єкти, що розпізнаються (розд. 14) [10.30, 10.40].

Категорія формується на основі *узагальнюючих ознак*, властивих даній сукупності розпізнаваних образів. За збільшення кількості навчальних вибірок несуттєві ознаки згладжуються, а ті, що часто зустрічаються, — підсилюються. У цьому разі відбувається поступове уточнення меж категорій. Добре навчена НКС здатна виокремлювати ознаки з *нових образів*, раніше невідомих системі, й створювати на їх основі *нові категорії образів*.

4. **Кластеризація** є класифікацією образів «без учителя», оскільки в ній відсутня апріорна *таксономія класів* (розд. 8.2). Алгоритм кластеризації здійснює пошук *подібних образів* й розміщує близькі за ознаками образи у певний *кластер* [10.41]. Поряд із розглянутими вище прикладами практичного застосування кластеризація використовується також при розв'язанні проблем здобування знань (розд. 4.4), стискання інформації, дослідження властивостей різних видів знань тощо.

5. **Прогнозування** полягає у прогнозі *майбутнього стану* (поводження) тих чи інших об'єктів даної ПрО на підставі аналізу інформації про стани цих об'єктів у *поточному* і *попередніх* періодах часу. Сучасні НКС здатні прогнозувати розвиток економіки й кон'юнктуру ринку, розробляти план капіталовкладень, прогнозувати погоду тощо (розд. 3.6). НКС широко застосовуються в практиці західних компаній для розв'язання задач *класифікації* промислових підприємств за ступенем їх перспективності й *прогнозування* роботи на основі аналізу взаємозалежних фінансово-економічних показників [10.19, 10.20].

Ефективність розв'язання зазначених задач багато в чому залежить від типу використовуваної нейромережі в НКС. Так, при розв'язанні задач *класифікації* найбільш ефективними є багатошаровий перцептрон і нейромережа Хеммінга. Перцептрон також застосовується в задачах *ідентифікації* й *прогнозування*. При розв'язанні задач *категоризації* будуть потрібні карта Кохонена, архітектура зустрічного поширення або нейромережа з адаптивним резонансом.

10.7. Практичні застосування нейрокомп'ютерних систем

Роботи зі створення НКС ведуться в багатьох країнах світу під егідою *Міжнародного суспільства з нейронних мереж*. Сьогодні розроблення в області НКС займаються більш як 300 компаній, серед

яких такі фірми-гіганти як «IBM», «Intel», «DEC», «Motorolla» та ін. Останнім часом різко зросла кількість НКС, призначених для контролю роботи АЕС, керування процесами безпечного збереження ядерних виробів, прогнозування надійності систем життєзабезпечення на космічних апаратах, літаках та інших об'єктах.

Аналіз сучасних розроблень в області нейрокомп'ютерних технологій дає змогу виокремити такі *перспективні напрями* в області створення й застосування НКС [2.34, 3.17–3.25, 10.7–10.45].

Перспективні нейрокомп'ютери

Головною тенденцією розвитку НКС на сьогодні є створення *оптоелектронних, оптичних, молекулярних нейротехнологій* і створення на їх основі *нейрокомп'ютерів*, здатних до *паралельно-розподіленого* опрацювання інформації, а також до *самонавчання* й *самоорганізації*.

Перспективним підходом до апаратної реалізації НКС є *трансп'ютерна нейротехнологія*. Нині в США та Японії створюються *паралельні трансп'ютерні нейрокомп'ютери* з великою кількістю (десятки й сотні тисяч) процесорів. Однією з основних проблемних областей застосування нейрокомп'ютерів на сьогодні є *розпізнавання образів промислового масштабу*, наприклад, — ідентифікація наземних об'єктів за результатами фотозйомки з космосу.

У 1993 р. німецька фірма «Siemens» розпочала випуск *нейрокомп'ютерів* типу «Synapse», який складається з керуючого *host-комп'ютера* на базі робочої станції «Sun Sparc Station II» і спеціалізованого нейропроцесора. Розрахункова продуктивність нейрокомп'ютера у 8 000 разів вища в порівнянні з *host-станцією*. У нейрокомп'ютері «Synapse» використовується нейромережева мова програмування «nAPL», середовище програмування «C++» та «UNIX»-сумісна операційна система.

У Японії з 1993 року прийнята програма «Real world computing program», основу якої становить створення *надшвидких супернейрокомп'ютерів*, призначених для розпізнавання образів, опрацювання семантичної інформації, керування інформаційними потоками й інтелектуальними роботами, здатними адаптуватися до навколишнього середовища. В Австралії нещодавно створено зразок *комерційного супернейрокомп'ютера*.

Нейрочіпи та нейроплати

Нині спостерігається тенденція до різкого зростання виробництва *базових нейрочіпів* НКС. Одним зі спільних розроблень компанії «Intel» та корпорації «Nestor» є створення спеціалізованих *нейрочіпів*

і *нейроплат* типу «*Snaps*» як приставок до звичайних комп'ютерів. Використання таких мікрочипів дає змогу підвищити швидкодію *гібридного* комп'ютера в сотні й тисячі разів.

Сьогодні 60% кредитних карток у США обробляють НКС прихованого виявлення речовин на базі теплових нейронів. Ідентифікація карток здійснюється за допомогою нейрокомп'ютера, що працює на *цифрових нейročіпах*. Подібна система фірми «*SAIC*» експлуатується у багатьох аеропортах США під час огляду багажу для виявлення наркотиків, вибухових речовин, ядерних та інших небезпечних матеріалів.

Нейročіпи використовуються в охоронних системах із нейромережевими алгоритмами виявлення рухомих об'єктів; а також у системах «електронного ключа» із розпізнаванням відбитків пальців і рисунка райдужної оболонки ока користувачів.

Нейрокомп'ютерне керування динамічними об'єктами

До числа перспективних областей практичного застосування нейрокомп'ютерів нині належать *системи керування*:

- складними маневрами винищувача під час повітряного бою з використанням бортової медичної ЄС для спостереження за станом льотчика;
- двигунною установкою гіперзвукового літака;
- робототехнічними комплексами у реальному масштабі часу;
- адаптивним складним телескопом, що забезпечує високу якість зображення й компенсацію атмосферних збурень;
- електроерозійними верстатами тощо.

Для керування зазначеними динамічними об'єктами створюються супернейрокомп'ютери, здатні оперативно вирішувати задачі аеро-, гідро- та газодинаміки, що описуються надскладними рівняннями математичної фізики.

На сьогодні США і Фінляндія ведуть роботи зі створення й використання нейрокомп'ютерів для керування хімічними реакторами.

Захист інформації в комп'ютерних системах

Нейрокомп'ютери використовуються для запобігання поломкам комп'ютерних систем, виявлення й ідентифікації *комп'ютерних атак* та ефективного захисту інформації в мережних структурах. На відміну від *кібернетичних КСШ*, що працюють на основі заданих правил і не здатні виявляти сценарії атак, НКС аналізують вхідні образи і не ідентифікують їх із закладеними правилами, а погоджують із характеристиками образів, які вони *навчені розпіз-*

навати. Крім того, НКС *навчаються* з часом, що необхідно для відображення нетипових зовнішніх атак на систему й активізації захисних функцій до (і під час) здійснення атаки.

Опрацювання зображень

Однією з основних сфер застосування сучасних НКС є ідентифікація наземних об'єктів за результатами аерофотозйомки з космосу. За допомогою нейрокомп'ютерів здійснюються різні види опрацювання аерокосмічних знімків й траєкторних вимірів, серед яких:

- пошук і розпізнавання об'єктів заданої форми й призначення;
- виявлення рухомих об'єктів;
- стиснення інформації та її відновлення;
- сегментація, контрастування й опрацювання текстур;
- опрацювання інформації у високопродуктивних сканерах тощо.

Опрацювання сигналів

Нині нейрокомп'ютери широко використовуються для опрацювання дискретних та аналогових *сигналів* різноманітної природи. Так, опрацювання *сейсмічної інформації* у нейрокомп'ютері дає змогу з великою точністю визначати координати й потужність землетрусу або ядерного вибуху.

Нейрокомп'ютери застосовуються для опрацювання *гідролокаційних сигналів*, які за своєю структурою близькі до сейсмічної інформації. Нейромережеве *гідролокаційне* опрацювання даних ефективно використовується для розпізнавання типу й координат надводних та підводних цілей.

Нейрокомп'ютери починають активно використовуватися під час опрацювання сейсмічних сигналів у *нафтогазоворозвідці*.

Наукові дослідження

Основним напрямом досліджень в області створення НКС є пошук нових фізичних принципів та середовищ *апаратної реалізації* сучасних типів *нейромережових структур*, орієнтованих на вирішення розглянутих вище практичних задач. Основу новітніх, створюваних тепер типів нейромереж, становлять *лінійні та нелінійні оптичні середовища*, які містять у собі Фур'є-оптику, голограмні засоби, нелінійні фоторефрактивні перетворювачі, оптичні хвилеводи, електронно-оптичні помножувачі тощо. Перспективними є також *хімічні й біологічні середовища* нейромереж із природними автохвильовими властивостями.

10.8. Особливості створення нейрокомп'ютерних систем

Створення НКС пов'язане з певними *труднощами*. Насамперед, пряме моделювання мозку людини потребує значної кількості (більш ніж 10^{14}) штучних нейронів. Реалізація таких нейромереж практично неможлива. Крім того, здатність мозку до підсвідомого мислення й навчання була вироблена протягом багатьох мільйонів років еволюції, тоді як механізми самонавчання та самоорганізації НКС поки що недосконалі і їх можна застосовувати лише обмежено.

Наявні нині технології створення НКС є відносно *дорогими*. Тому спочатку потрібно з'ясувати, чи потрібна взагалі *нейромережа* для вирішення виниклої проблеми. Цілком можливо, що розв'язання цього класу задач може бути отримане шляхом використання експертної чи іншої інтелектуальної системи. У цьому разі застосування НКС, зазвичай, виявляється неефективним.

Вибір структури НКС суттєво залежить від особливостей і складності розв'язуваних у ній задач. За потреби побудови *нової НКС* варто визначити, чи належить розв'язувана задача до одного зі *стандартних типів* нейромережових задач класифікації (категоризації), розпізнавання образів, кластеризації, прогнозування тощо. Для розв'язання цих задач уже існують оптимальні конфігурації практичних нейромереж, розглянутих, зокрема, у даному розділі.

Якщо розв'язувані в НКС задачі не можуть бути зведені до жодного з розглянутих вище типів, то розроблювачу доводиться вирішувати складну проблему синтезу нейросистеми нової конфігурації. Функціональні можливості НКС зростають зі збільшенням числа нейронів, зв'язків між ними та кількістю прошарків нейромережі. Однак у цьому разі суттєво зростає складність технології й вартість виготовлення системи. Введення зворотних зв'язків у нейромережу істотно підвищує ефективність її роботи, однак це може привести до динамічної нестійкості НКС. Тому при вирішенні *нестандартних задач* необхідно всебічно вивчити наявний досвід рішення *традиційних задач* і додатково дослідити ПрО систему.

У цілому проблема побудови такої НКС, яка ефективно вирішувала б усі проблеми у своїй ПрО, являє собою окремий перспективний напрям нейробіонічної науки.

Розділ 11

РОБОТОТЕХНІЧНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

11.1. Основні положення

Одним із основних напрямів розвитку КСШІ нині є створення сучасних *робототехнічних інтелектуальних систем (РТІС)*, які покликані *замінити людину* при виконанні робіт у несприятливих та шкідливих умовах промислового виробництва, у токсичних і радіоактивних середовищах підприємств атомної енергетики, у небезпечних умовах освоєння космосу й океану та в інших ПрО, де є загроза здоров'ю й життєдіяльності людини.

РТІС являє собою комплекс інтелектуальних комп'ютерних засобів і механічного устаткування, призначений для автоматизації (роботизації) людської праці у шкідливих або небезпечних умовах зовнішнього середовища [10.16, 11.1–11.22].

Основною концепцією, яка використовується при створенні РТІС, є *антропоморфний (людиноподібний) підхід* до побудови КСШІ, сутність якого полягає у наступному (рис. 11.1) [1.1–1.16, 11.1–11.22].

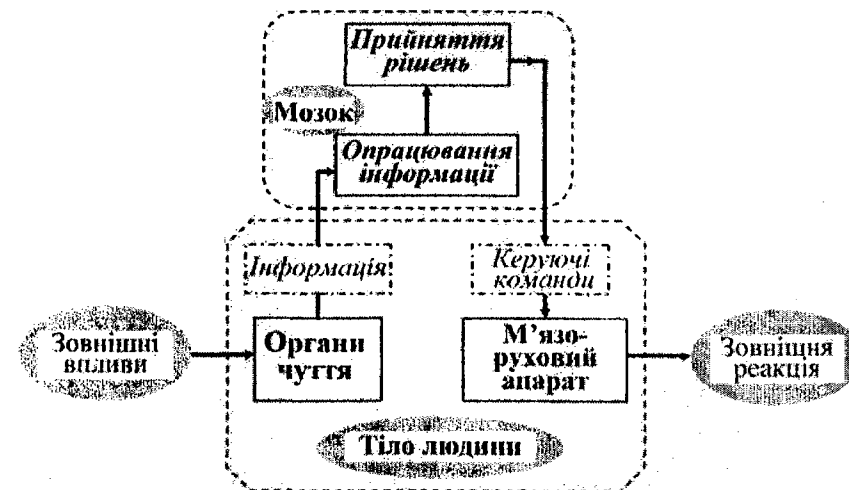


Рис. 11.1 — Узагальнена схема життєдіяльності людини

Людина одержує інформацію з навколишнього світу *п'ятьма органами чуття*, головним чином, *візуальним* (зір), *акустичним* (слух) і *тактильним* (дотик) *каналами сприйняття*. Людське сприйняття імітується в РТІС шляхом заміни очей телекамерами, вух — мікрофонами, дотику — тактильними (контактними) *сенсорами* (рецепторами і датчиками), а складні фізичні дії людини — виконавчими пристроями і механізмами. «Мозком» РТІС слугує *вирішувач* — високопродуктивний *інтелектуальний комп'ютер*.

Антропоморфний підхід до побудови РТІС передбачає:

- створення візуальних, акустичних, тактильних та інших засобів сприйняття системою навколишнього світу;
- *інтеграцію* інформації, одержуваної різними каналами, для створення *інтегрованого* інформаційного середовища функціонування системи;

- вибір та прийняття рішень щодо керування системою у складних (екстремальних) ситуаціях;

- оцінку впливу системи на зовнішнє середовище та інші задачі.

Властивості *антропоморфізму* означають, що РТІС:

- адекватно сприймати й розпізнавати *об'єкти* зовнішнього світу;
- здійснювати моніторинг навколишнього оточення і власного стану;

- формувати внутрішнє уявлення про навколишнє середовище й процеси, що в ньому відбуваються;

- виробляти й приймати рішення з виконання необхідних дій щодо заданих цілей на основі нагромаджених знань і досвіду;

- переміщатися у зовнішньому середовищі, обходячи або долаючи перешкоди;

- здійснювати цілеспрямований пошук потрібних об'єктів у навколишньому середовищі;

- мати адаптивну поведінку, тобто здатність самостійно формувати й виконувати дії залежно від ситуації, що складається;

- виконувати необхідні дії та маніпуляції з об'єктами навколишнього середовища;

- змінювати обстановку навколишнього середовища через маніпулювання її *об'єктами*;

- передавати інформацію людині-оператору (чи інший КСШ) про довкілля та всі свої дії в діалоговому режимі, наближеному до людського спілкування тощо.

Поряд із переліченими вище здібностями РТІС повинна також мати можливості функціонування в несприятливих для людини умовах. Простір (приміщення), в якому розміщена система, може не мати освітлення й система повинна використовувати інфрачервоний діапазон візуального огляду простору. РТІС може працювати

в агресивних, токсичних та радіоактивних середовищах. Температура середовища також може бути небезпечною для людини, атмосфера може бути цілком позбавлена кисню тощо.

11.2. Узагальнена структура робототехнічної системи

Відповідно до вище розглянутої функціональної структури КСШ (розд. 3.3) і принципів побудови РТІС, необхідними структурними компонентами РТІС є такі *підсистеми* (рис. 11.2) [11.1–11.22]:

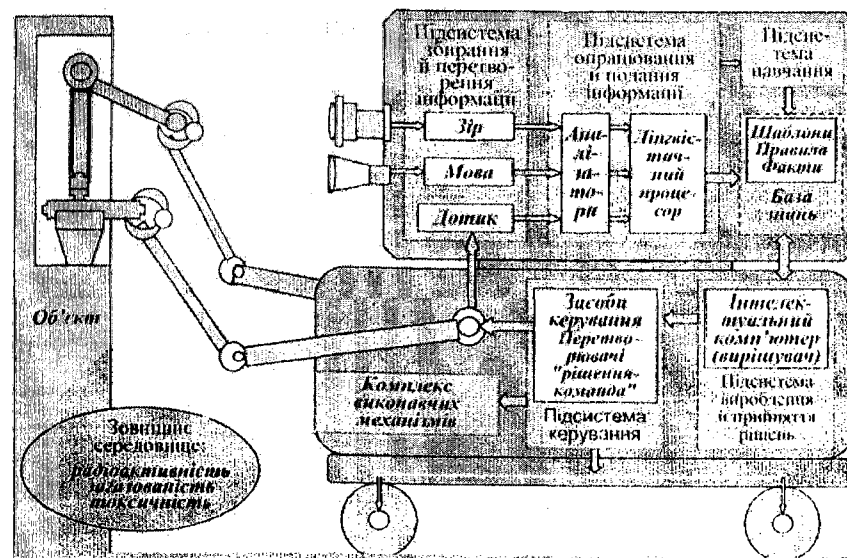


Рис 11.2 — Структурна схема робототехнічної інтелектуальної системи

⇒ Інформація

Керування ⇐

- *підсистема збирання й перетворення інформації* забезпечує одержання і первинне перетворення даних вимірів та спостережень, що надходять у РТІС каналами сприйняття інформації про *навколишнє середовище* і *внутрішній стан* системи;

- *підсистема первинного опрацювання й подання інформації* здійснює опрацювання вхідної (первинної) інформації задля формування *інформаційних моделей*, які адекватно відображують характерні риси й динаміку зміни зовнішнього середовища, а також стан робочих органів системи;

- *підсистема навчання* призначена для *адаптації* РТІС до навколишнього середовища ПрО, в якій вона працює, та до вирішення поставлених перед нею задач (розд. 10.1);

- *база знань* містить інформацію про певну ПрО, в якій працює РТІС (розд. 5);

- *підсистема зв'язку та керування* містить апаратно-програмні засоби *дистанційного зв'язку* людини-оператора з РТІС і вироблення *керуючих впливів та команд*, що по *каналах керування* передаються до виконавчих пристроїв і механізмів системи;

- *підсистема виконавчих пристроїв і механізмів* реалізує керуючі впливи (команди) на об'єкти зовнішнього середовища і внутрішні органи РТІС.

Нижче розглянуті основні характеристики зазначених підсистем РТІС [11.1–11.22].

11.3. Функції підсистем робототехнічної системи

11.3.1. Збирання та перетворення інформації

Ефективність функціонування РТІС залежить багато в чому від можливостей *підсистеми збирання й перетворення інформації*, яка забезпечує систему *оперативними даними* щодо характеристик і стану *об'єктів* зовнішнього середовища, про динамічні зусилля, що виникають у ланках робочих органів, про кінематичні параметри руху виконавчих механізмів та іншими даними. У цій підсистемі здійснюється *первинне перетворення даних*, які надходять у РТІС *візуальним, акустичним, тактильним* та інших *каналах* сприйняття зовнішнього світу і внутрішнього стану системи.

Ця інформація використовується надалі для розв'язання необхідних робототехнічних задач, вироблення й реалізації керуючих впливів на об'єкти зовнішнього середовища й робочі органи РТІС, а також для програмної перебудови системи у випадку непередбачених чи аварійних ситуацій.

Підсистема збирання й перетворення інформації складається, у свою чергу, з кількох *підсистем нижнього рівня*, які містять *візуальні, акустичні і тактильні* апаратно-програмні засоби збирання й перетворення даних вимірів та спостережень.

Візуальна інформація

Візуальна (зорова) інформація формується в *підсистемі машинного зору* РТІС, яка сприймає навколишній світ за допомогою однієї

чи кількох *телекамер* [11.2, 11.7, 11.13]. Одержуване телекамерою *двовимірне зображення* являє собою *матрицю (поле) пікселів (точок)*. Вона відбиває *інтенсивність освітленості* фрагментів так званої *сцени*, що являє собою сукупність *об'єктів*, які знаходяться в деякій обмеженій області простору, яку сприймають телекамери.

Отримана матриця пікселів називається *візуальним (зоровим) образом сцени*. Якщо матриця складається з чорних та білих точок, то такий образ називається *бінарним*. Якщо між чорними та білими точками *інтенсивність освітленості* має кілька відтінків, то це — *напівтоновий образ*.

Наприклад, для формування візуального образу розмірністю 512×512 *пікселів* з інтенсивністю освітленості 256 відтінків потрібно понад $2 \cdot 10^6$ *біт* інформації. Це висуває достатньо високі вимоги до необхідної продуктивності та обсягу пам'яті системи.

При формуванні *стереоскопічних та кольорових* образів вимоги до продуктивності та обсягу пам'яті системи багаторазово зростають. Для одержання *стереоскопічних просторових образів* у РТІС використовуються не менше двох узгоджено працюючих телекамер. За потреби одержання *кольорових образів* визначають освітленість та інтенсивність випромінювання трьох основних кольорів: *червоного, зеленого й синього*. Задля цього телевізійні камери забезпечуються відповідними червоно-зелено-синіми фільтрами для розпізнавання кольорових фрагментів об'єктів даної ПрО.

Акустична інформація

Акустична (звукова) інформація формується в *акустичній підсистемі* РТІС, яка має у своєму складі спеціальне *акустичне устаткування* у вигляді спеціалізованих *мікрофонних пристроїв*, призначених для аналізу й відтворення *акустичної (мовної) інформації* [11.6, 11.8, 11.12].

Формування *акустичної інформації* в РТІС здійснюється в такий спосіб. Коливання тиску повітря, що надходять з *акустичного джерела* (наприклад, під час *мовного повідомлення*), перетворюються мікрофоном у неперервний *електричний сигнал* різної інтенсивності, який являє собою *аналоговий акустичний образ мовного повідомлення*. Цей сигнал надходить в *аналого-цифровий перетворювач мікрофона* [11.4], де перетворюється в сукупність дискретних імпульсів, упорядкована послідовність яких являє собою *дискретний акустичний образ мовного повідомлення*.

Тактильна інформація

Тактильна (контактна) інформація формується в *сенсорній підсистемі* РТІС, яка містить у собі сукупність *сенсорних пристроїв (сенсорів)*,

призначених для сприйняття (прийому) й перетворення даних *тактильних (контактних)* вимірів [11.1, 11.4, 11.5, 11.9–11.1].

Сенсор складається з *рецептора* й *датчика*. *Рецептор* сприймає безпосередній фізичний вплив зовнішнього середовища й виробляє *сигнал* відповідної *природи сприйняття* (просторово-тимчасових зусиль, деформацій, температур тощо.). *Датчик* перетворює сигнал рецептора в *електричний сигнал*, який є *тактильним (сенсорним)* образом *контактного виміру*.

До складу РТІС входять датчики стикання, положення, зусилля, деформації, кутового й лінійного переміщення тощо. Є також потенціометричні, трансформаторні, індуктивні, фотометричні, ультразвукові, тензометричні, акустичні, локаційні, лазерні та інші типи датчиків.

11.3.2. Первинне опрацювання й подання інформації

Аналізатори

Візуальні (зорові), акустичні (мовні, голосові, звукові) і тактильні (сенсорні) образи об'єктів зовнішнього світу і стану робочих органів РТІС, сформовані відповідними апаратно-програмними пристроями, надходять у підсистему первинного опрацювання й подання інформації. Ця підсистема має візуальні, акустичні й тактильні аналізатори, які здійснюють перетворення фізичних образів у відповідні зорові, звукові й контактні інформаційні образи зовнішнього світу й внутрішнього стану системи.

Процес опрацювання (аналізу) окремого фізичного образу в тому чи іншому аналізаторі містить у собі:

- *стискання* інформації з метою скорочення часу опрацювання образу;
- *видалення (фільтрацію)* перекручувань і завад із фізичного сигналу;
- *поділ (сегментацію)* фізичного образу на *образні фрагменти*, які мають різну інформаційну природу (просторово-тимчасову, статичну, динамічну тощо);
- *виділення в образних фрагментах інформаційних елементів* (областей, ліній, вершин, координат взаємного положення тощо) відповідно до їх *інформативних ознак*;
- *формування (синтез) інформаційного образу з інформаційних елементів.*

Унаслідок первинного опрацювання даних у підсистемі формуються *зорові, акустичні й контактні інформаційні образи*, які відображують *характеристики і стани об'єктів зовнішнього світу та робочих органів РТІС.*

Моделі подання інформації

Інформаційні образи, сформовані аналізаторами, структуруються і подаються в РТІС у вигляді двох типів моделей подання інформації, що антропоморфно (див. вище) відбивають процеси функціонування першої та другої сигнальних систем сприйняття людиною зовнішнього світу.

Перша сигнальна система керує поведінкою людини і відповідає за одержання, опрацювання й відображення інформації на *рівні об'єктних образів*. Подібно до цього, у РТІС здійснюється формування *об'єктних (образних) інформаційних моделей*.

Друга сигнальна система людини керує сприйняттям, опрацюванням та відображенням інформації на *мовному рівні*. У РТІС здійснюється подібний процес опрацювання інформації для формування *мовних інформаційних моделей*.

Образні та мовні інформаційні моделі являють собою структуровані сукупності *зорових, акустичних і контактних інформаційних образів* і адекватно відображають характерні риси об'єктів зовнішнього середовища, динаміку її зміни і стан робочих органів РТІС.

Лінгвістичний процесор

Як було зазначено вище (розд. 3.1), у КСШІ здійснюється *символьне опрацювання інформації*. Тому *образні й мовні інформаційні моделі* подаються в РТІС у *символьному (евристичному) вигляді*. Перетворення *фізичних інформаційних образів (сигналів)* у *символьні моделі (і назад)* здійснює *лінгвістичний процесор*, що являє собою апаратно-програмний пристрій, призначений для *сприйняття, розпізнавання, формування, подання та відображення символної (текстової) інформації*.

Лінгвістичний процесор може функціонувати у *двох режимах*, здійснюючи перетворення «сигнал — текст» (аналіз) чи зворотнє перетворення «текст — сигнал» (синтез). У режимі *сприйняття (аналізу)* сформовані аналізаторами *інформаційні образи* надходять у *лінгвістичний процесор*, де перетворюються (кодуються) у *символьні (текстові) образи*, які заносяться потім у БЗ чи в *робочу пам'ять* вирішувача.

У процесі *відображення (генерації, синтезу)* інформації відповідні фрагменти *символьних (текстових) образів*, що знаходяться в *робочій пам'яті* вирішувача чи у БЗ, за командою *планувальника* системи надходять у *лінгвістичний процесор*. Тут вони перетворюються (декоднуються) у відповідні фрагменти *інформаційних образів*, а потім — у *фізичні сигнали*, які по каналах керування передаються

до виконавчих механізмів або відтворюються акустичними (мікрофонними) пристроями як *мовні повідомлення*.

11.3.3. Вироблення й прийняття рішень

Формування *рішень* щодо впливу на об'єкти зовнішнього середовища та робочі органи РТІС здійснюється у *підсистемі вироблення й прийняття рішень*, основу якої становить *інтелектуальний комп'ютер (вирішувач)* (розд. 3.3). Він розв'язує *інтелектуальні робототехнічні задачі* шляхом використання механізмів опрацювання символічної інформації, моделей подання знань та інтелектуальних засобів виводу на знаннях (розд. 4–8).

Функціонування підсистеми (і РТІС у цілому) може здійснюватися у *двох режимах*.

1. В *автоматичному режимі* вирішувач *заздалегідь* «знає» *фізичне завдання*, яке повинна виконати РТІС. Таке завдання може бути сформульовано, наприклад, таким чином: «Знайти в обмеженому просторі об'єкт із заданими властивостями, наблизитися до нього й виконати визначені дії». Оскільки таке завдання виконується в РТІС постійно, то вирішувач розв'язує відповідну *інтелектуальну задачу*, сформовану на етапі навчання системи, але вже з урахуванням нової інформації про стан об'єктів зовнішнього середовища і робочих органів системи.

2. У *ручному режимі* завдання системі видає *людина-оператор*, яка на основі одержуваної інформації про зовнішнє середовище і внутрішній стан системи формує *команди* щодо виконання відповідних робіт (розд. 11.3.4). Одержавши таку команду, вирішувач *формує (ставить)* відповідну *інтелектуальну задачу* й визначає способи її розв'язання.

Наприклад, якщо надійшла команда «Пересунути РТІС у задану точку простору і з'ясувати фізичну природу об'єкта, що в ній знаходиться», то вирішувач «знає», що для цього треба вирішити задачу вибору *оптимального маршруту руху* (за наявності *перешкод*) (див. нижче) і задачу *розпізнавання образів* (розд. 8.2).

На підставі аналізу отриманої команди (*запиту*) вирішувач вибирає з БЗ і заносить у свою робочу пам'ять символічні *шаблони* об'єктів і *дій*, отримані *підсистемою навчання*, а також відповідні інформаційні *моделі й алгоритми* розв'язування поставленої задачі. Потім вирішувач розв'язує поставлену *задачу* на підставі обраних моделей і алгоритмів. Результати вирішення задачі заносяться в робочу пам'ять вирішувача (чи у відповідний розділ БЗ системи) і чекають там команди *планувальника* щодо подальшого їх використання.

Більшість робототехнічних задач пов'язана з необхідністю опрацювання в реальному часі великих обсягів швидкозмінливої, неповної, нечіткої чи невизначеної інформації. До таких задач належить, наприклад, вироблення й прийняття рішень щодо впливу на рухомі об'єкти у складних та небезпечних умовах зовнішнього середовища. Для розв'язання таких задач перспективні РТІС обладнуються *універсальним (гібридним, комплексним) вирішувачем*, що здатний вирішувати задачі шляхом комплексного застосування просторово-тимчасових методів логічного виводу на знаннях, алгоритмів прийняття рішень на розмитих шкалах, планування доцільної діяльності та інших інтелектуальних засобів опрацювання *образної та мовної інформації* [11.15–11.22].

Одним із основних *типів* задач, розв'язуваних у РТІС, є задача *розпізнавання образів* (розд. 8.2). *Символьні образи розпізнаваного об'єкта* чи *мовного повідомлення*, отримані *лінгвістичним процесором*, порівнюються у вирішувачі з відповідними *символьними зразками (шаблонами) БЗ*, які були розроблені на етапі *навчання* РТІС. За максимальної подібності аналізованих *слів* зі *словами-зразками* відбувається їх ототожнення, що і складає сутність процесу *розпізнавання образів*.

Іншим видом задач, що вирішує РТІС у процесі свого функціонування, є визначення *оптимальних маршрутів (траєкторій) руху* в робочій області зовнішнього середовища за наявності в ній різного роду *перешкод*. Для розв'язання таких задач у *перспективних* РТІС застосовують *генетичні алгоритми* (розд. 8.3) планування траєкторій, сутність яких полягає в наступному [11.19–11.22].

Простір (x, y) робочої області S функціонування РТІС розбивається на кінцеве число N елементарних прямокутних *площадок* із цілочисельними координатами (x_p, y_p) ($p = 1 \dots N$). Динаміка зміни (у часі t) ситуації на кожній p -й площадці описується сукупністю M_p *матриць* розмірністю (x_{pk}, y_{pk}) ($k = 1 \dots M_p, p = 1 \dots N$)... Таким чином, модель робочої області S являє собою *матричну мережу*, що складається з $M_i \times N$ *вузлів*. Кожен вузол містить координати *осередків* (елементів матриць) у вигляді індексів x_i і y_i , а також *індекс* (інтервал) *часу* t_i , протягом якого робот знаходиться в i -му осередку.

Величина M_i визначає число *кроків квантування* за часом, що залежить від *динаміки змін* (головним чином — *пересування перешкод*) у робочій області. *Квантування* полягає у фіксуванні *моментів зміни нестационарного* (що змінюється в часі) робочого середовища з подальшим формуванням *осередків матричної мережі*.

Істотами у генетичному алгоритмі слугують маршрути руху роботів по осередках мережі. Сукупність маршрутів на визначеному кроці еволюції утворює популяцію хромосом, кожна з яких являє собою послідовність вузлів, що утворюють маршрут руху окремого робота. Хромосома поділяється на сегменти, які відповідають окремим ділянкам маршруту. Перший вузол сегмента є стартовим, останній — кінцевим. Гени хромосоми відповідають індексам часу t_i й відрізняються один від одного на одиницю. Число генів у хромосомі динамічно змінюється в ітераційному процесі еволюції.

Функцією придатності є зважена сума нормованих значень функцій відповідності у вигляді критеріїв оптимальності, які забезпечують:

- найкоротшу довжину маршрутів руху роботів;
- мінімальний час руху роботів по маршрутах;
- мінімальну довжину маршрутів, що пролягають на «безпечних» відстанях від стаціонарних і рухливих перешкод;
- гладкість траєкторії руху роботів по маршрутах.

Для формування початкової популяції використовується метод випадкового пошуку допустимої траєкторії з використанням моделі «колеса рулетки» (розд. 8.3). Хромосома, що відповідає найбільшому сектору на колесі, має найбільшу придатність (пристосовність) і тим самим визначає оптимальний маршрут руху робота. Модель дає змогу пропускати в наступне покоління хромосоми із середньою придатністю, але з оптимальним значенням однієї з функцій відповідності (наприклад, із вдалим маршрутом обходу робота однієї з перешкод). Це означає, що у популяції з покоління в покоління буде спостерігатися експонентне збільшення кількості істот, що володіють підвищеною придатністю.

Для генерування нового покоління хромосом у РТІС поряд з операторами схрещування й мутації застосовується також операція випрямлення, яка поліпшує структуру хромосоми і полягає в наступному:

- вибирається початковий або кінцевий вузол хромосоми та вузол, що відстоїть від обраного на кілька переходів;
- проводиться пряма, яка з'єднує осередки з вибраними вузлами;
- якщо осередки, через які проходить пряма, вільні від перешкод і роботів, то відповідні проміжні вузли вилучаються зі списку точок траєкторії, а ген часу кожного з наступних вузлів зменшується на величину переходу.

Крім того, у генетичному алгоритмі виконується операція згладжування, яка забезпечує гладкість руху робота за маршрутом. Її

використання дає змогу вилучати ділянки маршруту з різкими поворотами, розворотами в напрямках, зворотних основному напрямку руху тощо.

11.3.4. Підсистема зв'язку й керування

Загальне керування процесом функціонування РТІС здійснює підсистема зв'язку й керування, яка містить у собі такі програмно-апаратні пристрої:

- планувальник (диспетчер), що керує інформаційними потоками системи;
- канали зв'язку й передавання інформації між функціональними компонентами системи;
- пристрої зв'язку з виконавчими механізмами системи;
- пристрої дистанційного зв'язку з віддаленим операторним пунктом керування.

Процес функціонування підсистеми зв'язку й керування РТІС в автоматичному режимі (тобто без втручання людини-оператора) здійснюється в такий спосіб. Дані вимірів і спостережень, отримані (у вигляді фізичних сигналів) у підсистемі первинного опрацювання й подання інформації, планувальник направляє в лінгвістичний процесор, який здійснює перетворення «сигнал-текст» й перетворює фізичні сигнали у відповідні символічні образи зовнішнього середовища та внутрішнього стану системи. Далі планувальник направляє цю інформацію у вирішувач, який формує й розв'язує відповідну інтелектуальну задачу (див. вище).

У ручному режимі загальне керування роботою РТІС виконує людина-оператор, що перебуває у віддаленому пункті керування, обладнаному пристроями й каналами дистанційного зв'язку з РТІС. На підставі одержуваної інформації про зовнішнє середовище і внутрішній стан системи оператор формує команди (накази, інструкції) щодо виконання відповідних робіт. Основними способами взаємодії людини-оператора з РТІС є керування за допомогою тактильних (ручних) апаратно-програмних засобів або використання мовних (голосових) команд у діалоговому режимі, наближеному до людського спілкування. Виконання команд оператора здійснюється в такий спосіб [11.6, 11.8, 11.12].

Тактильна чи мовна команда оператора надходить у відповідні сенсорні або акустичні пристрої (що входять до складу підсистеми первинного опрацювання й подання інформації) і перетворюється там у фізичний сигнал. Цей сигнал направляється в лінгвістичний процесор, що

здійснює перетворення «сигнал-текст», тобто фізичний сигнал перетворюється тут у відповідний символний образ (запит) команди оператора. Отриманий запит надходить у вирішувач, який формує й вирішує відповідну інтелектуальну задачу.

Результати розв'язання задачі, отримані вирішувачем (за будь-якого режиму керування), планувальник знову направляє в лінгвістичний процесор, який здійснює зворотне перетворення «текст-сигнал». Тут символні образи результатів розв'язання задачі перетворюються у відповідні фрагменти інформаційних образів, а потім — у фізичні сигнали (команди), які по каналах керування передаються планувальником до виконавчих пристроїв РТІС.

Спілкування людини з комп'ютером природною мовою (українською, англійською та ін.) не завжди є зручним, оскільки для спілкування з РТІС, створених у різних країнах, необхідні вирішувачі-поліглоти, що розуміють різні мови. З огляду на це, нині створюються спеціальні «рідні» мови спілкування РТІС із людиною. Наприклад, у перспективних РТІС використовуються мови когнітивної графіки (розд. 3.3). Оператор може малювати на екрані дисплея рисунки, писати команди у вигляді умовних знаків і вся ця інформація буде вводиться безпосередньо в робочу пам'ять вирішувача й автоматично виконуватися.

11.3.5. Виконавчі пристрої й механізми

Вироблені системою керуючі впливи на об'єкти зовнішнього середовища реалізує комплекс виконавчих пристроїв та механізмів РТІС. До складу виконавчого устаткування РТІС входять механічні «руки»-маніпулятори й органи переміщення («ноги» робота) у вигляді рухливих шасі (див. рис. 11.2).

За допомогою «рук» (маніпуляторів) РТІС активно впливає на об'єкти зовнішнього середовища. Сучасні маніпулятори мають не менш семи ступенів рухливості, збільшення яких розширює розмірність виконуваних робіт, але ускладнює процес керування роботами.

Рухливі механічні «ноги»-шасі забезпечують виконання таких завдань (задач), як подолання рівнів, переступання через перешкоди, біг по місцевості тощо. У відповідності з цим РТІС може мати гусеничний, колісний чи крокуючий тип переміщення рухливих шасі. Рух у кожному із ступенів рухливості «рук» і «ніг» забезпечується електричними, гідравлічними або пневматичними автоматизованими приводами.

Велику роль для ефективної роботи РТІС грає вибір оптимальної конфігурації виконавчих пристроїв і механізмів, що дозволяє вико-

нувати поставлені задачі з мінімальною витратою енергії. Енерговитрати характеризуються потужністю, що витрачається на виконання корисної роботи, тертя в механізмах, втрати в приводних двигунах, деформацію поверхні зовнішнього середовища тощо.

11.4. Колективні робототехнічні системи

Перспективним напрямом розвитку робототехніки нині є інтегровані РТІС (ІРТІС), що складаються з базового (центрального) робототехнічного комплексу (вузла), що виконує роль оператора — людини (див. вище) та групи (колективу) автономних мобільних роботів, які виконують цілеспрямовані дії з метою розв'язку загальної складної задачі [11.20, 11.21]. Базовий комплекс здійснює дистанційне автономне керування рухом мобільних роботів, з'єднаних каналами зв'язку й узгоджено виконуючих установлені функції. У складі базового комплексу може бути кілька комп'ютерів, телекамер та виконавчих пристроїв. Кожний з мобільних роботів має бортовий комп'ютер.

Однією з основних проблем, що виникають при створенні ІРТІС, є забезпечення необхідної швидкості передавання інформації каналам зв'язку між мобільними роботами і базовим комплексом. Тому автономність керування колективом роботів реалізується шляхом перекладання якомога більше функцій з базового комплексу на бортові системи керування мобільними роботами, що дозволяє знизити вимоги до пропускну здатності каналів зв'язку і скоротити час його роботи.

Нині створюються ІРТІС різної архітектури, які відрізняються одна від одної схемою керування колективами роботів. Централізована схема керування передбачає організацію двонапрямкових каналів обміну інформацією між кожним роботом колективу й центральним комплексом, у якому опрацьовується основна частина всієї інформації. Недоліком такої схеми керування є вихід із ладу всього колективу у разі відмовлення центрального комплексу (вузла).

При децентралізованій архітектурі основну частину інформації опрацьовують мобільні роботи, а центральний комплекс лише видає керуючі команди. Роботи можуть поєднуватися в окремі групи (кластери), які виконують однорідні завдання. У цьому разі керування здійснюється кожним кластером автономно. Недоліком цієї схеми керування є велика ймовірність локальних відмовлень та конфліктних ситуацій, які потребують додаткових керуючих команд для координації дій роботів.

Різновидом *колективних ІРТІС* є *багатоагентні (мультиагентні)* системи, що являють собою системи взаємодіючих *роботів-агентів* (розд. 13). У такій системі є *агент-супервізор*, який здійснює керування *центральною* комплексом *верхнього рівня*, кілька *агентів-координаторів середнього рівня* й *рядових агентів*, оснащених локальними системами керування.

З огляду на нові досягнення в *мікромеханіці* й *мініатюризації* комп'ютерної техніки останнім часом стрімко розвиваються ІРТІС, які містять у собі *колективи міні-* та *мікророботів* [11.17]. Особливостями таких систем (на відміну від *великих ІРТІС*) є малі і надмалі розміри мікророботів, працюючих за мінімальних обчислювальних та енергетичних ресурсів. При побудові таких *мікросистем* використовуються спеціальні методи керування колективом мікророботів, зокрема, — алгоритм *динамічного виокремлення агентів-координаторів* у процесі роботи системи та *хвильовий спосіб* передавання керуючої й сенсорної інформації по каналах зв'язку [11.20].

11.5. Області практичного застосування робототехнічних систем

На сьогодні РТІС широко використовуються в галузях космонавтики, дослідження океану, ядерної енергетики, промислового виробництва та інших сферах людської діяльності. Промислові, маніпуляційні, мобільні й спеціалізовані РТІС широко розповсюджені у всіх сучасних виробничих процесах та технологіях. Загальна кількість роботів, що діють в автомобільній, електротехнічній, хімічній, харчовій, легкій галузях промисловості та в машинобудуванні розвинутих країн світу становить нині більш ніж 2 млн промислових зразків із середньорічним збільшенням до 30%. Кількість спеціалізованих РТІС, призначених для роботи в екстремальних умовах, за останні кілька років збільшилося в сотні разів. Основними сферами практичного застосування сучасних і перспективних РТІС є такі Про.

Космічні дослідження. Однією з основних задач космонавтики є створення РТІС для здійснення фізико-хімічних, біологічних та інших наукових досліджень планет сонячної системи за допомогою аналізу проб ґрунту, збирання зразків порід і вивчення поверхні планет.

Космічна ІРТІС, яка призначена для дослідження планети, складається зі *стаціонарного комп'ютерного комплексу* й *автономних мобільних роботів*, з'єднаних між собою радіоканалами і виконуючих необхідну науково-дослідну роботу (див. вище). Стаціонарний комп-

лекс виконує *паралельне опрацювання інформації*, що надходить із Землі та від мобільних роботів, формує й приймає рішення щодо керування роботами і передає оперативні експериментальні дані на Землю (у Центр керування космічними польотами) каналом далекого космічного зв'язку. Стаціонарний комплекс може розташовуватися на *штучному супутнику* досліджуваної планети, що знаходиться на стаціонарній орбіті над місцем, де працюють мобільні роботи. Кожен із цих роботів обладнаний *бортним вирішувачем*, призначеним для вирішення навігаційних та науково-дослідних задач.

Однією з головних задач космічних досліджень є *виконання робіт у відкритому космосі*. Застосовувані для цих робіт РТІС можуть успішно конкурувати з космонавтами при виконанні таких типових космічних операцій, як *інспекція* космічних кораблів і супутників для оцінювання їх стану та виявлення ушкоджень, обслуговування і ремонт космічних об'єктів (особливо тих, на борті яких є ядерні двигуни або енергоустановки), зняття й заміна експериментальних блоків, приладів і зразків, передача палива і вантажів, порятунків космонавтів та інші роботи.

Дослідження океану. Через обмежені ресурси корисних копалин суші суттєво зростає важливість досліджень океану з його величезним резервом морських ресурсів та засобів для подальшого існування людства. Останнім часом значний інтерес учених викликають тектонічні процеси й дослідження рельєфу дна океану. Найперспективнішим розробленням сучасних технічних засобів підводних наукових досліджень вважаються глибоководні РТІС, що збирають зразки порід, буравлять свердловини, беруть проби ґрунту на дні океану та виконують іншу роботу.

Підводні РТІС можуть бути двох типів: ті, що *рухаються* по дну, й *батискафи*, що плавають (гелікоптери). Вони забезпечуються інтелектуальними комп'ютерами, гідролокаторами, телевізійними камерами, маніпуляторами, звукоуловлювачами, гідрокомпасами та іншим підводним устаткуванням, необхідним для проведення підводних наукових досліджень і робіт в океані.

Промислове виробництво. Нині РТІС широко застосовують у різних сферах промислового виробництва, насамперед — у *гнучких виробничих системах*, де вони можуть бути легко запрограмовані на вирішення різних виробничих задач без реорганізації виробничих ділянок і промислових цехів. Перші промислові РТІС використовували в 60-х роках на автоматичних лініях збирання корпусів автомобілів на заводах фірми США «Дженерал моторс». Їх функціонування здійснюється в такий спосіб. Конвеєр переміщає укріплені на ньому кузови уздовж лінії точкового зварювання, яка складається

із двох десятків автономних роботів, розташованих попарно з кожної сторони лінії та керованих з одного центру системи. Вони по чергово підводять зварювальні пістолети до корпусу автомашини й зварюють деталі. Кінематичні схеми «рук» роботів мають багато ступенів рухливості, тому роботи можуть пересувати зварювальні пістолети, ідучи за складними вигинами корпусу автомобіля.

У Японії створено промисловий інтелектуальний робот, призначений для виконання складально-монтажних робіт із візуальним контролем. Робот керується вирішувачем, який формує необхідний програмний рух, що відпрацьовується електрогідравлічною системою. Захват маніпулятора оснащений тактильними датчиками. У системі візуального сприйняття використовуються дві телевізійні камери, обладнані червоно-зелено-синіми фільтрами для розпізнавання кольору предметів. Робот здатний розпізнавати предмети, обмежені площинами й циліндричними поверхнями за спеціально-го освітлення. Вартість робота становить більш як \$500 тисяч.

У Росії створено *транспортний автономний інтегральний робот (ТАІР)*, який являє собою триколісне шасі, на якому змонтовані сенсорна система й блок керування. До складу устаткування робота входять оптичний далекомір, навігаційна підсистема із двома радіомаяками й компасом, контактні датчики, датчики кутів нахилу візка, таймер та інші пристрої. Основою підсистеми керування робота ТАІР є *бортова нейромережа*, яка реалізує різні алгоритми опрацювання сенсорної інформації, планування поведінки й керування рухом робота.

В Італії створені роботи типу «Робоволк» на комбінованому колісно-крокуючому шасі, призначені для дослідження процесів у вулканах. «Робоволк» було випробувано під час виверження вулкана Етна в 2003 р.

У різних країнах світу створюються роботи вертикального переміщення для детектування вертикально розташованих конструкцій, виявлення вибухових речовин, протипожежних операцій, вирізання отворів у стінах палаючих резервуарів та виконання інших видів робіт.

Розроблено мініатюрні роботи з дистанційним керуванням й різними конфігураціями систем механічного руху. Мініроботи можуть переміщатися в трубах малих діаметрів і здійснювати технічну діагностику (інспекцію) внутрішніх поверхонь труб, виявляти в них сторонні предмети та виконувати інші операції.

Під егідою розвинутих країн світу (головним чином, США і Японії) діє *Міжнародна програма з перспективної робототехніки*

(*International Advanced Robotics Programme — IARP*), покликана сприяти створенню *перспективних РТІС*, які можуть замінити людей у важких, шкідливих і небезпечних умовах роботи. У межах *IARP* визначені основні стратегічні напрями розвитку РТІС, до яких належать:

- інтелектуальні РТІС прийняття рішень;
- *мережеві РТІС*, які працюють в *Internet*, а також у глобальних, корпоративних та інших системах і мережах;
- *адаптивні РТІС*, застосовувані у промисловості, атомній енергетиці, медицині, біоінженерії тощо;
- *міні-, мікро- та нанороботи*, засновані на сучасних досягненнях електроніки, комп'ютерної техніки й мікромеханіки;
- РТІС у будівництві, сільському господарстві, медицині, переробці лісу, у вугільній та інших галузях промисловості;
- персональні роботи для ремонту та обслуговування вокзалів, готелів, клінік тощо.

Ключовими напрямами розвитку *РТІС* є розширення спектра інформації, яку вони сприймають, збільшення обсягу її опрацювання, поліпшення сервісних можливостей спілкування з людиною-оператором для підвищення гнучкості керування об'єктами різної природи й призначення.

Розділ 12

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ПРИЙНЯТТЯ РІШЕНЬ

12.1. Основні положення

Людині у процесі своєї життєдіяльності часто доводиться *приймати рішення* в тій чи іншій *ситуації*, що склалася. *Прийняттям рішень* називають вироблення рекомендацій щодо вибору *оптимального рішення* виниклої проблеми з кількох можливих альтернатив.

Традиційним способом прийняття людиною рішень у складних (екстремальних) ситуаціях є використання *колективного розуму* (знань, умінь) шляхом організації так званої *мозкової атаки*, в якій беруть участь спеціально підібрані й функціонально структуровані групи *фахівців* (експертів) родинних професій, характерних для певної ПрО (розд. 4.4). Її метою є колективне обговорення й дослідження ситуації, що створилася, з різних точок зору. Використовуючи свій життєвий досвід, загальні й професійні знання, фахівці пропонують рішення задач, які допомагають знайти вихід зі складної ситуації. Результати «мозкової атаки» аналізує й оцінює незалежна група експертів, яка приймає остаточне рішення з приводу обговорюваної проблеми.

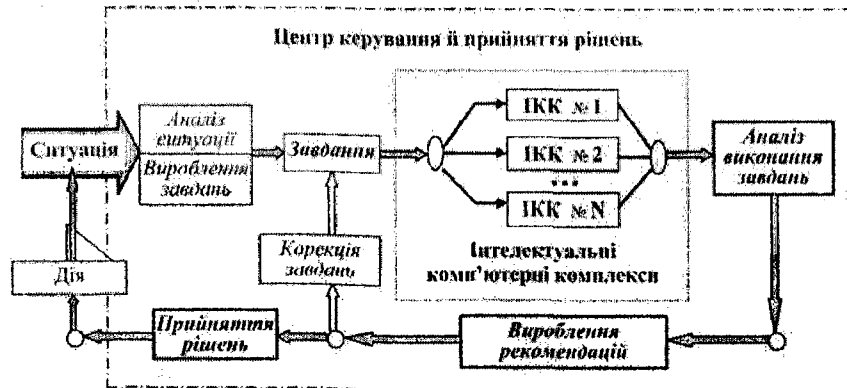


Рис. 12.1 — Узагальнена функціональна схема глобальної інтелектуальної системи прийняття рішень

У процесі вибору оптимального варіанта з кількох можливих альтернатив фахівці, що приймають рішення, здійснюють відповідні *розумові процедури*, у тому числі:

- навчаються на своєму і чужому досвіді;
- поповнюють і заново систематизують свої знання;
- роблять висновки на основі аналізу наявного досвіду та знань;
- пояснюють і обґрунтовують, чому вони дійшли того чи іншого висновку;
- підвищують рівень своєї компетентності тощо.

Системи штучного інтелекту, які *моделюють* подібні розумові процедури задля *оперативного вирішення* інтелектуальних задач, що виникають у різних ПрО, називають *інтелектуальними системами прийняття рішень (ІСПР)*. Головною особливістю цих систем, якою вони відрізняються від інших типів КСШ, є планування цілеспрямованих дій і прийняття рішень у *складних (екстремальних, критичних, форс-мажорних) ситуаціях*, пов'язаних із дефіцитом часу, неповнотою інформації та іншими обставинами, що ускладнюють роботу ІСПР, яка працює в *реальному масштабі часу* [12.1–12.10].

Як і будь-яка інтелектуальна система, ІСПР функціонує у *важко формалізованих* ПрО й використовує *моделі* людського мислення й пам'яті, що ґрунтуються на використанні наявного досвіду, придбаних навичках, знаннях та способах евристичного мислення творців КСШ. ІСПР поєднують у собі людський інтелект, інформаційні технології, інтелектуальне програмне забезпечення, а також досвід та знання людства, нагромаджені до цього часу (розд. 3).

12.2. Структура та функції системи прийняття рішень

На рис. 12.1 наведена *узагальнена функціональна схема* глобальної ІСПР, процес функціонування якої здійснюється таким чином. На підставі аналізу сформованої ситуації, пов'язаної, наприклад, із космічною загрозою, *центр керування й прийняття рішень (ЦКПР)* системи повинен виробити оперативні рішення щодо усунення цієї загрози. У розпорядженні ЦКПР є розвинена мережа *віддалених інтелектуальних комп'ютерних комплексів (ІКК)*, до складу яких входять вирішувачі, сховища знань, банки моделей, алгоритмів та інші апаратно-програмні засоби опрацювання інтелектуальної інформації (розд. 3.3).

Головним завданням системи є формування й реалізація *задач керування* інформаційними потоками та *прийняття рішень* щодо усунення космічної загрози. На базі інформації, отриманої від

космічних та наземних засобів спостереження, ЦКПР розробляє оперативні *завдання*, пов'язані з розв'язанням цих задач. Потім він розподіляє їх між ІКК, які вирішують зазначені задачі й передають у ЦКПР їх результати та рекомендації щодо реагування на ситуацію, яка склалася.

На підставі цих результатів і рекомендацій ЦКПР приймає рішення щодо ліквідації космічної загрози. У разі потреби ЦКПР може виробити нові *скоректовані завдання* і повторити весь цикл вироблення й прийняття рішень.

У процесі роботи ІСПР здійснює неперервне *прийняття рішень у реальному часі* задля вироблення дій, адекватних виниклій ситуації. У процесі свого функціонування ІСПР здійснює такі дії (розд. 3.3):

- на підставі аналізу поточної ситуації формує *первинний опис ситуації* в термінах мови системи;
 - *уточнює* (доповнює) цей опис, зіставляючи його з *моделями БЗ*, і формує *модель ситуації* в термінах знань системи, тобто здійснює процес *розуміння ситуації*;
 - здійснює переклад опису ситуації на внутрішню мову системи;
 - з урахуванням *моделі ситуації* формує відповідні *інтелектуальні задачі* й вибирає *алгоритми* їх вирішення;
 - *вирішує* поставлені задачі шляхом спільного використання моделі ситуації, обраних моделей мислення й подання знань та алгоритмів вирішення задачі;
 - *інтерпретує* отримані рішення задач і формує алгоритм реагування системи на ситуацію, що створилася;
 - впливає на ситуацію за допомогою виконавчих апаратно-програмних пристроїв, що виконують дії, погоджені з поставленою метою.
- Сучасні ІСПР, зазвичай, приймають самостійні рішення щодо вибору та застосування того чи іншого варіанта вирішення проблеми. За потреби ІСПР можуть надавати користувачам можливі варіанти рішень проблеми з оцінюванням наслідків реалізації кожного варіанта.

Основною складовою частиною ІСПР є *підсистема керування моделями (ПУМ)*, призначена для *моделювання зовнішньої ситуації* й вибору відповідних моделей у БЗ. До складу ПУМ входять:

- *банк моделей ситуацій*, що формується на етапі *навчання* системи й містить у собі структуровану множину ситуацій, характерних для даної ПрО;
- *блок (процесор) пошуку*, перебирання й аналізу (*прогону*) моделей;
- *інтелектуальний інтерфейс*, який забезпечує діалоговий режим спілкування користувачів (фахівців) із системою в процесі прийняття рішень.

ПУМ здійснює пошук, генерацію, іспит, реструктуризацію і комбінування моделей, реалізацію зв'язків між ними, ведення довідника моделей тощо.

У ПУМ сучасних ІСПР використовуються *прецедентні* (розд. 8.6), *нейромережеві* (розд. 10), *нечіткі* (розд. 14) та *комбіновані (гібридні) моделі*, що складаються з моделей зазначених типів.

Підсистема еволюційної адаптації ІСПР здійснює пошук оптимального рішення на основі використання генетичних алгоритмів *настроювання моделей і генерації рішень*, зокрема — еволюційних методів групового обліку аргументів (розд. 8).

Підсистема аналізу, оцінювання та вибору альтернативних рішень здійснює:

- імітаційне моделювання альтернативних варіантів рішень;
- дослідження й вибір (оптимізацію) прийнятних імітаційних моделей з використанням методів об'єктно-орієнтованого планування експериментів;
- оцінювання та вибір оптимального варіанта рішення на підставі аналізу результатів імітаційних експериментів з урахуванням заданих критеріїв оптимізації.

У процесі створення ІСПР здійснюється попередній (*передпроектний*) *аналіз прийнятих рішень*, спрямований на максимальне *спрощення дій*, які виконуватимуться в *реальному часі* роботи ІСПР. На *передпроектному* етапі здійснюється розроблення можливих *сценаріїв* розвитку подій, визначення шляхів і методів зменшення впливу несприятливих чинників та ін.

12.3. Області практичного застосування інтелектуальних систем прийняття рішень

Для вирішення *практичних задач прийняття рішень* у США і Японії створюються складні коштовні комплексні ІСПР. Основною сферою їхніх практичних застосувань є *бізнес-процеси* й *фінансово-економічні* сфери, до числа яких належать (розд. 8):

- аналіз кон'юнктури ринку задля вибору стратегії виходу із кризової ситуації;
- формування портфеля інвестицій;
- планування промислових замовлень;
- оцінювання фінансових ризиків;
- вибір інвестора, страхової компанії тощо.

Нині ІСПР успішно застосовуються й в інших ПрО, серед яких можна виокремити:

- *промисловість* — моніторинг (неперервний контроль) і прийняття рішень щодо запобігання та ліквідації наслідків аварій на атомних реакторах, нафтохімічних заводах, електростанціях та інших великих промислових об'єктах;

- *оборону* — вибір стратегії дій під час нападу супротивника або при тероризмі;

- *медицину* — прийняття рішень при наданні швидкої допомоги, у виборі засобів лікування, лікарських препаратів тощо. (розд. 9.6).

ІСПР входить до складу комп'ютерного комплексу ЦКПР (див. рис. 12.1) космічного агентства США NASA, де розроблюють та приймають рішення щодо керування *пльотами космічних кораблів* та порятунку космонавтів у разі позаштатної ситуації на борту корабля.

Перспективним напрямом розвитку ІСПР є використання їх в *інтегрованих робототехнічних системах (ІРТІС)* (розд. 11) як *центрального базового комплексу*, що здійснює *дистанційне керування* функціонуванням автономних мобільних роботів, які узгоджено вирішують загальну задачу розроблення та прийняття рішень щодо впливу на об'єкти навколишнього світу.

Стрімкий розвиток теоретичного апарату та засобів моделювання, комерційний успіх і значне поширення ІСПР в різних галузях економіки і виробництва обумовило їхнє застосування в сучасних *ситуаційних центрах* західних країн, де приймаються ключові політичні рішення й моделюються кризові ситуації в різних країнах світу.

Розділ 13

МУЛЬТИАГЕНТНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

Проблема «*інформаційного вибуху*», пов'язана з необхідністю опрацювання великих обсягів нагромадженої інформації (див. *Вступ*), зумовило появу так званих *мультиагентних систем (МАС)* [13.1–13.35]. Основним призначенням цих систем є оперативний *пошук й витяг інформації* з різних джерел та наступне її опрацювання для подальшого використання в *системах прийняття рішень* (розд. 12).

МАС є *інтелектуальною системою*, що здатна здійснювати оперативний *аналіз, подання, організацію, опрацювання та збереження* інформації. Головними джерелами інформації для МАС є інформаційні ресурси глобальних систем та мереж (головним чином — *Internet*), а також БЗ, онтології, програмне забезпечення інших систем, електронні магазини, публікації тощо.

У МАС використовуються методи й засоби *КСIII*, серед них — моделі мислення й подання знань, механізми опрацювання природно-мовної інформації та інші інтелектуальні комп'ютерні засоби (розд. 5–8).

Основним призначенням існуючих МАС є пошук інформації в *Internet* та її подальше опрацювання. Поряд із цим МАС знаходять також застосування у *системах керування* виробничими процесами, рухом різних видів транспорту, великими інформаційними сховищами, а також в *е-комерції*, системах навчання (тьютори), електронних бібліотеках та в інших Про.

13.1. Структура й функції мультиагентної системи

На рис. 13.1 наведена структурна схема МАС, до складу якої входять такі *програмно-апаратні засоби* [13.1–13.35].

1. Основою МАС є *структурований комплекс інтелектуальних агентів*, кожен з яких являє собою *програмний (виконавчий) модуль*, орієнтований на вирішення певної *інтелектуальної інформаційної задачі*, яку поставив перед ним *користувач (власник)*.

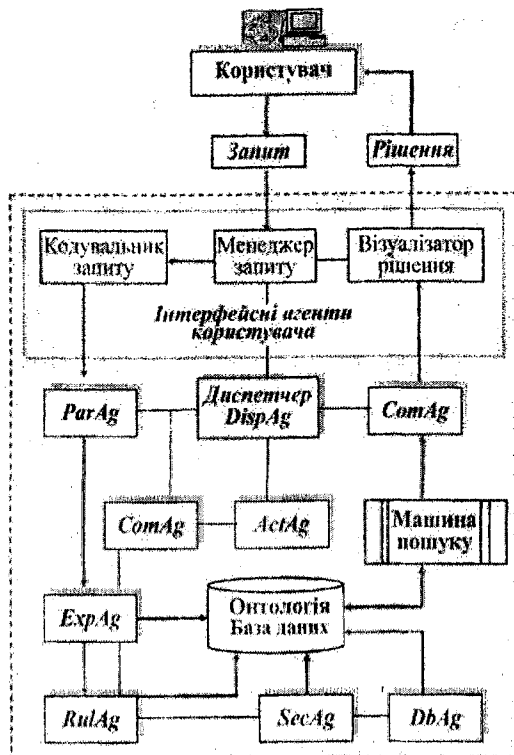


Рис. 13.1. — Структурна схема MAC

— зв'язки між агентами
→ інформаційні потоки

4. **База даних** містить відомості про внутрішні ресурси системи, місцезнаходження й основні характеристики джерел інформації, зв'язки між компонентами системи та іншу службову інформацію.

Основним типом агентів MAC є *інтерфейсні агенти користувача*, до яких належать *менеджер запити*, *кодувальник запити*, *візуалізатор вирішення задачі* та інші агенти. Вони здійснюють безпосередню взаємодію з користувачем, виконують його інструкції та команди, реагують на оцінки користувача щодо вирішення задач, адаптуються до його індивідуальних особливостей тощо.

Поряд із агентами *користувача* у складі MAC функціонують також такі види *агентів* (рис. 13.1):

- *агент-диспетчер (DispAg)* координує роботу всіх агентів MAC;
- *агент дії (ExpAg)* виявляє й описує конкретні ситуації та дії розв'язуваної задачі;

2. **Машина пошуку** інформації здійснює пошук та додавання інформації з різних джерел. Інтерфейс машини пошуку являє собою меню, за допомогою якого користувач може сформулювати *запит* на пошук інформації, використовуючи ключові слова, фрази та логічні зв'язки типу «і», «або», «не».

На сьогодні у MAC використовуються пошукові засоби *AltaVista*, *Exite*, *Infoseek*, *Lycos*, *Webcrawler*, *Yahoo* та ін. [13.25]. Прикладом повнотекстової машини пошуку з використанням морфології російської мови є *Yandex*.

3. **Онтологія** — спеціалізована БЗ, у якій зберігаються *концептуальні знання* Про та функціонує MAC (розд. 5.2).

- *агент БЗ (DbAg)* здійснює *структурування* й *нагромадження знань* у БЗ;
- *агент комунікацій (ComAg)* відповідає за надання агентам каналів зв'язку;
- *агент-спостерігач (ActAg)* сприймає й фіксує дії інших агентів MAC;
- *агент даних (ParAg)* формує базові типи даних, що використовуються для опису параметрів розв'язуваної задачі.

Основними функціями MAC є:

- забезпечення діалогової взаємодії користувача з агентами;
- декомпозиція (поділ) розв'язуваної задачі на підзадачі й розподіл їх між агентами;
- координація спільних планів дій агентів;
- організація синхронної взаємодії між агентами;
- вирішення конфліктів між агентами;
- інформаційна підтримка й відновлення баз даних і знань MAC;
- адаптація до змін навколишнього (мережевого) середовища шляхом програмної перебудови (реконфігурації) агентного співтовариства системи.

До числа основних проблем, із якими стикаються розроблювачі MAC за її практичної реалізації, належать:

- легальність способів переміщення агентів мережею;
- верифікація агентів (зокрема — захист від вірусів);
- дотримання агентами прав приватної власності;
- збереження агентами конфіденційності (таємності) інформації;
- перенаселення мережі агентами;
- сумісність коду агентів із програмно-апаратними засобами системи, якій вони функціонують, тощо.

13.2. Функціональні властивості інтелектуальних агентів

Основними *властивостями* інтелектуальних агентів є [13.4—13.8]:

- наявність базових знань агента про себе й навколишнє програмне середовище;
- *керованість* — виконання операцій відповідно до правил поведінки, установлених користувачем;
- *автономність* — самостійне функціонування без втручання (детальних інструкцій) користувача;
- *активність* — здатність виявляти власну ініціативу, планувати доцільні дії, генерувати цілі й діяти раціональним образом для їх досягнення;

- *самоконтроль* внутрішнього стану та дій;
- *реактивність* — адекватне сприйняття програмного (мережевого) середовища й відповідна реакція на його зміни;
- *співробітництво* — здатність взаємодіяти з іншими агентами для досягнення спільної мети, поставленої перед ними користувачами;
- *адаптивність* — адаптація до змін у МАС й відповідна зміна своєї поведінки залежно від ситуації, що склалася;
- *самонавчання й самовдосконалення* на основі нагромадженого досвіду розв'язуваних задач;
- *спілкування на рівні знань* — можливість взаємодії з користувачем природною мовою, наближеною до людського спілкування;
- *толерантність (інваріантність)* до помилок або невірних дій користувача.

Розглянуті вище агенти можуть бути *стаціонарними* або *мобільними*. Мобільні агенти можуть залишати віддалений клієнтський комп'ютер користувача і переміщатися на сервер для виконання своїх дій, а потім повертатися назад.

Відповідно до сучасних поглядів спеціалістів щодо моделей мислення *агента-людини* інтелектуальні агенти можуть також мати *переконання, бажання, зобов'язання, наміри* й мати такі властивості, як *соціальна поведінка, раціональність, правдивість, доброзичливість* тощо.

Негативним прикладом використання інтелектуальних агентів є *віруси*, упроваджувані в програмне забезпечення тієї чи іншої системи для організації *перебоїв* у її роботі.

Основною характеристикою інтелектуального агента є його *ментальна модель*, яка містить у собі:

- описання можливостей та способів використання агента;
- початкові й поточні задачі (*наміри*) агента;
- можливі результати вирішення задач (*зобов'язань* агента);
- *правила поведінки* агента в тих чи інших ситуаціях, які визначають сукупність його можливих «відгуків» на поточні зміни навколишнього програмного середовища.

Інтелектуальні агенти мають *атрибути*, які містять назву *платформи* (із специфікацією сервісних послуг і властивостей агента), адресу, призначення сервісів агента, ступінь пріоритетів сервісів та їх застосування.

У агентів є *інтерфейси (IAgentMessage)*, через які відбувається спілкування між агентами шляхом приймання-передавання повідомлень, які містять основну й службову інформацію. Для реалізації інтерфейсів розробляються спеціальні *мови* спілкування, до яких належить, наприклад, *IDL (Interface Definition Language)*.

13.3. Мультиагентні internet-системи

Як було зазначено вище (розд. 4), нині в *Internet* нагромадилася значна кількість «сирої» інформації, в якій практично неможливо відшукати необхідні *знання*. З огляду на це, сучасні МАС часто застосовуються для вирішення інформаційних задач у середовищі *Internet* із використанням, зокрема, технології *Data Mining* (розд. 4.6).

MAC-Internet містить у собі комплекс взаємозалежних *Internet-агентів*, які здійснюють пошук, витяг та опрацювання інформації, що розміщена на *Web-сторінках* (див. рис. 13.2). *Internet-агенти*, зазвичай, вбудовуються у браузер, здійснюють сортування електронної пошти й інформують користувача про події та повідомлення, які можуть його зацікавити.

Основними функціями *MAC-Internet* є [13.11–13.18]:

- вирішення задач оперативного *пошуку, витягу й опрацювання інформації* на множині сервісних ресурсів *Internet*;
- оформлення результатів вирішення задач як анотацій документів та звітів (*HTML-файлів*) із наданням їх користувачу та наступним збереженням у БЗ системи;
- класифікація знайденої інформації за напрямками досліджень;
- використання результатів вирішення задач для поліпшення параметрів якості процесу пошуку інформації;
- відновлення БЗ системи відповідно до об'єктивних змін інформаційних ресурсів *Internet*.

Процес функціонування *MAC-Internet* здійснюється у такий спосіб (див. рис. 13.2).

1. *Користувач МАС*, який бажає одержати необхідну інформацію, через *користувацький інтерфейс* надсилає *запит* до системи, тобто *ставить задачу*. Цей запит надходить до *агента користувача*, який підтримує діалогову взаємодію з користувачем і здійснює кодування запиту у внутрішнє подання вхідної мови системи.

2. *Код запиту* сприймається *агентом аналізу запиту*, який вибирає відповідну *онтологію* й інформує про це *агентів-брокерів* та інших агентів системи.

3. Ключові слова запиту користувача, що описують окремі фрагменти ПрО, зіставляються *агентом аналізу* з релевантними концептами онтології. На підставі цієї інформації формується *HTML-подання* запиту, яке направляється *агентам-брокерам* для знаходження релевантних *Web-сторінок*.

4. Одержавши *HTML-запит*, *агент-брокер* вибирає відповідну машину пошуку, яка відшукує *документи, релевантні* запиту користувача. При виявленні відповідної *Web-сторінки* *агент-брокер* дає команду *агенту Internet* на читання документів цієї сторінки.

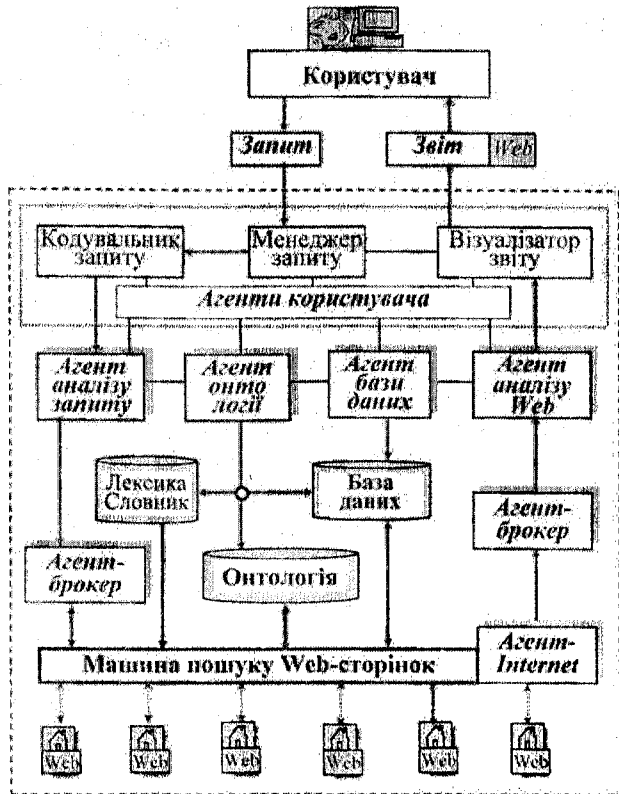


Рис. 13.2 — Структурна схема MAC-Internet

5. Агент Internet (мережний агент, спайдер, бот) підключає, зчитує й індексує відповідну Web-сторінку, яку визначив (знайшов) агент-брокер. Після зчитування Web-сторінки, спайдер надсилає звіт машині пошуку й здійснює повне або часткове індексування сторінки у вигляді заголовків документів. Повідомлення про отримане HTML-подання Web-сторінки спайдер повертає агенту-брокеру, який запам'ятовує отримані Web-сторінки (HTML-файли) і розподіляє їх між агентами аналізу Web-сторінок для подальшого опрацювання.

У разі недоступності сторінки або відсутності на ній необхідної інформації агент-спайдер сповіщає про це агента користувача. Помилково витягнуті документи ігноруються на стадії індексування.

6. Агент аналізу Web-сторінок (агент опрацювання HTML-тексту) здійснює семантичний аналіз Web-сторінки, отриманої від мережних агентів, для визначення її релевантності відповідним

концептам онтології. Він перетворює HTML-текст Web-сторінки у подання, з яким працюють морфологічний та синтаксичний аналізатори. Визначаються типи речень і мовних актів, які в них відображені.

7. Перевіряється відповідність отриманого тексту Web-сторінки так званому онтологічному тесту. Суть його полягає в тому, що кожне речення тексту зіставляється з описами концептів онтології. У разі збігу речення й концепту вибирається відповідна Web-сторінка. Користувачу пред'являються лише ті сторінки, що успішно пройшли цей тест.

8. Агент користувача подає результати пошуку релевантних Web-сторінок, використовуючи текстографічні засоби візуалізації інформації. Результати роботи агента супроводжуються анотаціями змісту знайдених документів, списками ключових слів та іншими даними, які полегшують користувачу одержання інформації, що його цікавить.

На рис. 13.3 наведена схема функціонування internet-агента, який вирішує задачу брокера — постачальника інформації потенційному покупцю про необхідні види товарів. Агент «Брокер» шукає й знаходить відповідні Web-сторінки продавців товарів, обговорює з ними ціни й інформує покупця про можливі варіанти покупок.

Щоб реалізувати ці дії, агент повинен мати можливість взаємодії з користувачем для одержання відповідних завдань і повернення отриманих результатів, орієнтуватися у мережевому середовищі й приймати рішення, необхідні для виконання поставлених перед ним задач. Якщо агент не може самостійно вирішити поставлену задачу, то він звертається до інших агентів, які можуть надати йому відповідні послуги.

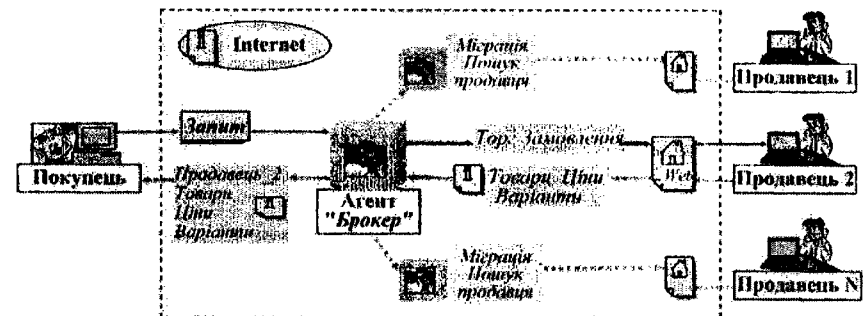


Рис.13.3 — Схема функціонування агента «Брокер»

Пошуковий механізм знаходить вільного агента і передає йому завдання на пошук. Агент приступає до роботи й повертає або зміст

документа, або пояснення, чому даний документ не можна доставити. Агенти функціонують автономно, що дає змогу ізолювати основний процес роботи системи від помилок і проблем із переповненням пам'яті.

13.4. Практичні розроблення мультиагентних систем

В області проектування та створення MAC нині працює багато зарубіжних фірм, організацій, дослідницьких установ, університетів, і їх кількість постійно збільшується [13.11–13.35]. Практичними розробленнями в цій області займаються такі транснаціональні компанії, як «IBM», «Microsoft», «Oracle», «Daimler-Benz», «DEC» та інші промислові фірми. Нижче розглянуті принципи побудови й функціонування деяких сучасних MAC-Internet, створених фірмами США [13.29–13.35].

MAC «Autonomy» фірми «AgentWare» використовує *нейромережевий* метод розпізнавання образів і методи *нечіткої логіки* під час пошуку й опрацювання *internet-інформації*. Запити в систему користувачі подають природною (англійською) мовою. Система аналізує текст запиту й витягає з нього смисловий зміст, який записують у спеціальний *конфігураційний файл*. У цьому разі внутрішнє подання запиту відображується в нейромережі, у вузлах якої розташовуються ключові слова і фрази.

Підґрунтям пошукового алгоритму є так званий *механізм динамічних міркувань*, основними функціями якого є визначення посилань на документи БЗ, релевантні запиту користувача, інформаційне забезпечення агентів та стандартний пошук слів у тексті.

MAC «Webcompass» здійснює пошук інформації в Internet шляхом *індексування тем документів*, поданих на Web-сторінках. Запит у цій системі базується на «прямому» використанні опису ПрО, сформованому користувачем. Для формування запиту досить проіндексувати теми, що цікавлять користувача. На підставі цих позначок система сама сформує запит на пошук релевантної інформації.

Пошук у системі «Webcompass» проводиться на підставі ключових слів і здійснюється одночасно на 35 машинах пошуку, які задаються списком. Цей список можна змінювати й додавати до нього адреси для пошуку нової інформації. Система перевіряє кожне знайдене посилання на доступність, аналізує знайдену інформацію, складає коротке резюме документів, а також визначає ступінь

відповідності Web-сторінок запиту користувача й класифікує знайдені документи.

MAC «MARRY» являє собою *мережу спеціалізованих агентів* у вигляді автономних *Java-програм*, які виконують такі функції:

- *агент користувача (інтерфейсний агент)* асистує користувачу при формуванні запитів й подає результати пошуку як релевантні URL- або Web-сторінки;
- *агент-брокер URL* призначений для «супроводу» списків *interest-адрес*, що доставляються броузерами;
- *агент-брокер HTML* запам'ятовує отримані Web-сторінки й розподіляє їх між *агентами опрацювання тексту* для подальшого аналізу;
- *агент мережі* підключає, зчитує й аналізує відповідну URL- або Web-сторінку, яку визначив *агент користувача*;
- *агент опрацювання тексту* здійснює *семантичний аналіз та опрацювання Web-сторінки для визначення її релевантності відповідним концептам онтології*.

Кожний із зазначених агентів системи:

- має свою власну *мережну адресу (URL)*;
- взаємодіє з іншими агентами за допомогою мови *ACL (Agent Communication Language)*, яка функціонує над *HTTP-протоколом Internet*;
- є й споживачем, і постачальником інформації залежно від того, з якими агентами він спілкується;
- може взаємодіяти з автономними програмними компонентами MAC (Web-броузерами, аналізаторами природної мови, онтологіями тощо);
- має спеціальні знання й можливості виведення текстової інформації Web-сторінки, релевантної певній ПрО.

До складу MAC «MARRY» входять *онтології e-комерції та internet-безпеки клієнтських застосувань*. У процесі вирішення задачі релевантні тексти онтологій зіставляються з реченнями (ключовими словами) запиту користувача, які описують окремі фрагменти ПрО. При цьому *мережеві агенти* вибирають відповідні машини пошуку, а *спеціалізовані агенти* здійснюють попередній аналіз отриманих Web-сторінок шляхом їх перевіряння на відповідність *онтологічному тесту* (див. вище).

НЕЧІТКІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

Багато понять і подань про зовнішній світ, які ми використовуємо в повсякденному житті, не завжди мають точний та повний опис. Такі поняття, як «дорослий», «популярний», «швидкий», «невеликий», «високий», «холодний» тощо, описують якісні характеристики об'єктів і за своєю природою носять відбиток невизначеності.

Для формалізації невизначених знань американський математик Заде розробив спеціальний апарат нечіткої логіки, що дав змогу опрацьовувати ці знання на комп'ютерах [14.1–14.4]. Методи вирішення інтелектуальних задач із використанням нечітких моделей і нечіткого виводу на знаннях одержали назву *м'яких обчислень*. Вони виконуються у нечітких інтелектуальних системах, БЗ яких містять нечіткі (fuzzy) знання [14.1–14.17].

14.1. Основні поняття нечіткої логіки

Нечітка логіка є узагальненням традиційної булевої логіки, яка оперує із двійковими числами, що відповідають поняттям «істина» й «хибність». У нечіткій логіці використовуються й усі проміжні стани між «істиною» та «хибністю», які подаються натуральними числами з інтервалу $[0...1]$.

У нечіткій логіці вводиться поняття *лінгвістичної перемінної*, значеннями якої є поняття *природної (людської) мови*, які називають *термами*. Кожен терм може набувати певних фізичних (числових) значень із деякого інтервалу, що встановлюється у процесі придбання знань із певної Про (розд. 4).

Фізичні інтервали зміни термів перекриваються й утворюють *діапазон зміни лінгвістичної перемінної*. Кожне її значення із цього діапазону характеризується *ступенем належності* до того чи іншого терму, що визначається деяким числом у проміжку $[0...1]$. Упорядкована геометрична послідовність точок, що відображує *ступінь приналежності* термам усіх фізичних значень діапазону, називається *функцією належності* лінгвістичної перемінної. Множину значень, які може приймати *функція належності* в діапазоні її зміни, називають *нечіткою множиною*.

Розглянемо приклад побудови *функції належності віку людини нечіткій множині «молодий»*. У двовимірних координатах по осі абсцис x будемо відкладати значення *лінгвістичної перемінної «вік»*, а по осі ординат — *ступінь належності $P(x)$ людини множині «молодий»* (рис. 14.1).

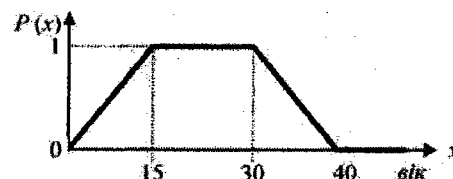


Рис. 14.1 — Функція належності віку «молодий»

Термами лінгвістичної перемінної «вік» можуть бути поняття «дитячий» (менш ніж 15 років), «молодий» (15–30 років), «літній» (30–40 років) і «старий» (більш ніж 40 років). В інтервалі 15–30 років ступінь приналежності віку людини множині «молодий»

дорівнює 1. Після 30 років цей ступінь стає меншим, а після 40 років його значення буде дорівнювати нулю. З'єднавши отриманий горизонтальний відрізок (15–30 років) нахиленими лініями зі значеннями $x = 0$ і $x = 40$ відповідно, одержимо *функцію належності*, що описує нечітку поняття «молодий» для всього діапазону віку людини. Якщо аналогічним чином ввести інші поняття, пов'язані з віком людини («дитячий», «літній», «старий»), то можна одержати формалізовану нечітку множину, яка описує весь період життя людини. Тепер його можна використовувати у БЗ нечіткої інтелектуальної системи, не піклуючись про фізичну природу цього поняття.

Як приклад використання нечіткої логіки можна навести задачу *переміщення важкої довгомірної вантажівки у вузький гараж* із довільної початкової точки. Якщо вирішувати цю задачу традиційним способом, то доведеться обвішати автомобіль різноманітними датчиками й акселерометрами, а потім скласти й розв'язати складну систему диференціальних рівнянь у часткових похідних.

Водночас використання нечіткої логіки принципово спрощує задачу. Використовуючи лише три нечітких параметри (лінгвістичних перемінних) — *швидкість*, *орієнтацію* автомобіля й *відстань* до гаража, можна одержати вичерпний опис будь-якої ситуації цієї задачі. У БЗ системи заносяться нечіткі правила *продукційного типу*: «Якщо до гаража далеко і швидкість невелика і ніс автомобіля спрямований уліво, то візьми вправо». При використанні пакета програм CubiCalc, розробленого на основі нечіткої логіки, для вирішення зазначеної задачі знадобилося описати лише дванадцять ситуацій й тридцять п'ять нечітких правил, кожне з яких не складніше, ніж наведене вище.

Перехід від класичного двійкового числення до м'яких обчислень (з інтервальним поданням чисел між «істиною» та «хибністю»)

потребував розроблення відповідних правил *нечіткої логіки* та *операцій із нечіткими множинами*. На рис. 14.2 подані схематичні зображення основних *нечітких операцій*, до числа яких належать *об'єднання*, *перетинання*, *доповнення* й *концентрація*.

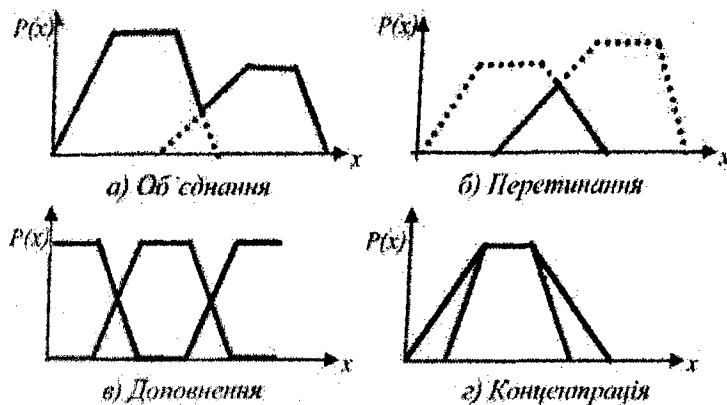


Рис. 14.2 — Операції з нечіткими множинами

14.2. Структура та функції нечіткої системи

Як і будь-яка інтелектуальна система, *нечітка КСШ* функціонує у важко формалізованих ПрО і використовує *моделі людського мислення* й подання знань, механізми логічного виводу на знаннях та інші інтелектуальні апаратно-програмні засоби опрацювання природномовної інформації (розд. 3). Тому структура сучасної *нечіткої КСШ* може бути подана як функціональна схема *перспективної КСШ*, наведеної на рис. 14.3.

Водночас *нечітка система* має низку відмінних рис, пов'язаних із виконанням у ній м'яких обчислень із використанням *нечіткої логіки*. Поряд із традиційними апаратно-програмними засобами КСШ до складу *нечіткої інтелектуальної системи* входять такі функціональні компоненти (див. рис. 14.3):

1. *База нечітких знань* містить структуровану множину моделей, правил нечіткої логіки та лінгвістичних перемінних, які характеризують дану ПрО.

2. *Блок фазифікації (fuzzy block)* перетворює дані вимірів і спостережень, що надходять на вхід до системи, у *нечіткий формат*. Сформована нечітка множина вхідних даних прямує у *вирішувач*.

3. *Нечіткий інтелектуальний вирішувач* розв'язує інтелектуальні задачі на підставі вхідної інформації. Для цього він використовує нечіткі моделі, правила й лінгвістичні перемінні, закладені у *базі*

нечітких знань. Результати вирішення задач мають також *нечіткий* характер.

4. *Блок дефазифікації* перетворює отриману вирішувачем інформацію з нечіткого формату у звичайні фізичні величини, які надходять потім до виконавчих пристроїв системи як сигнали.

5. *Нечіткий блок спілкування* призначений для організації взаємодії користувача з нечітким вирішувачем і містить програмно-апаратні пристрої *фазифікації* й *дефазифікації* відповідної інформації та засоби *мовного спілкування* з користувачем.



Рис. 14.3 — Структурна схема нечіткої інтелектуальної системи

14.3. Практичні розроблення нечітких систем

14.3.1. Нечітка робототехнічна система

Нечіткі КСШ застосовуються у багатьох ПрО, де є потреба в оперуванні *нечіткою* інформацією. Останнім часом методи нечіткої логіки з успіхом використовуються для вирішення *робототехнічних задач*. Інформація про навколишнє середовище, в якій функціонує РТІС, найчастіше носить *нечіткий* характер. Наприклад, система не завжди може точно визначити відстань до шуканого об'єкта й змушена оперувати нечіткими поняттями типу «далеко»

чи «близько». Водночас відстань до об'єкта має суттєве значення для визначення напрямку й швидкості руху до об'єкта за наявності перешкод.

Розглянемо принципи побудови та функціонування нечіткого робота, задачею якого є об'їзд перешкод під час руху до цільового об'єкта.

Для вирішення задач керування роботом вводяться дві вхідні лінгвістичні перемінні: «відстань» (від робота до перешкоди) і «напрямок» — кут між поздовжньою віссю робота й напрямком на перешкоду.

Значеннями лінгвістичної перемінної «відстань» є терми «дуже близько», «близько», «середнє» й «далеко». Значення термів мають такі інтервали зміни: «дуже близько» — 0–40 см, «близько» — 25–100 см, «середнє» — 25–100 см, «далеко» — 25–100 см.

Сукупність фізичних інтервалів зміни термів, що перекриваються, утворює діапазон зміни лінгвістичної перемінної «відстань» (рис. 14.4). Кожне значення із зазначеного діапазону характеризується ступенем його належності до того чи іншого терму. Наприклад, ступінь належності відстані 40 см до терму «дуже близько» дорівнює 0,15, а до терму «близько» — 0,6.

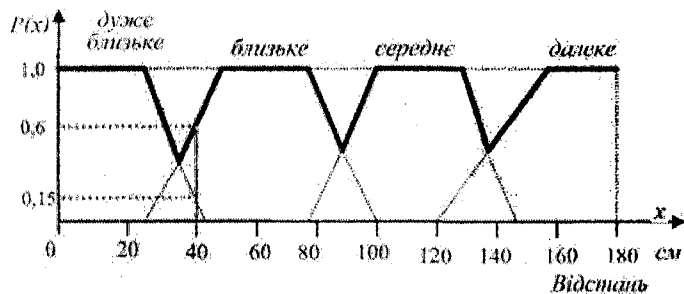


Рис. 14.4 — Функція належності лінгвістичної перемінної «відстань»

Для іншої вхідної лінгвістичної перемінної «Напрямок», що набуває значення в діапазоні від 0 до 360°, визначаються терми: «лівий», «прямий» та «правий».

Вихідною лінгвістичною перемінною є «кут керування» із термами «різко вліво», «вліво», «прямо», «вправо», «різко вправо». Зв'язок між вхідними та вихідними лінгвістичними перемінними можна подати як нечіткі правила функціонування робота, наведені в табл. 14.1.

Кожен запис у цій таблиці відповідає своєму нечіткому продукційному правилу, наприклад:

Якщо дистанція «близько» і напрямок «правий» то кут керування «різко вліво».

Таблиця 14.1 — Нечіткі правила керування роботом

Напрямок	Відстань			
	Дуже близько	Близько	Середньо	Далеко
Правий	Різко вліво	Різко вліво	Вліво	Прямо
Прямий	Різко вліво	Вліво	Вліво	Прямо
Лівий	Різко вправо	Різко вправо	Вправо	Прямо

Розглянуті нечіткі правила табл. 14.1 використовуються, зокрема, в автономному мобільному роботі «MARGE», який здійснює пошук об'єктів, виявляє та об'їжджає перешкоди, здійснює стикування з об'єктами та вирішує інші інтелектуальні задачі (див. фото). Підсистема збирання та первинного опрацювання інформації робота містить у собі дві телевізійні камери технічного зору, оснащені червоно-зелено-синіми фільтрами для розпізнавання кольору предметів, кілька ультразвукових датчиків, які забезпечують загальне поле огляду розміром у 240°, й тактильні сенсори для запобігання зіткнень робота з перешкодами та об'єктами.



База нечітких знань робота містить лінгвістичні перемінні, нечіткі правила та інші нечіткі знання, сформовані на етапі навчання й адаптації робота до функціонування в навколишньому середовищі.

Нечіткий вирішувач-мультиплексор може працювати в режимі пошуку й руху до об'єкта та в режимі об'їзду перешкод. Інформація про об'єкти навколишнього середовища, що надходить із сенсорної підсистеми робота, переводиться у нечіткий формат (фазифікується) й опрацьовується вирішувачем відповідно до правил нечіткої логіки, закладених у базу нечітких знань. Отримані вирішувачем результати дефазифікуються у звичайні фізичні величини, а потім перетворюються в керуючі сигнали, які спрямовуються до виконавчих органів робота.

Режими пошуку об'єкта та об'їзду перешкод реалізуються в роботі за допомогою рульового керування та керування його швидкістю. Правила рульового керування дають змогу роботу рухатися коридором паралельно до стін, об'їжджати перешкоди, вибиратися з напівзамкнених

перешкод й наводиться на шуканий об'єкт за допомогою підсистеми технічного зору. Керування швидкістю здійснюється за допомогою ультразвукових датчиків, які забезпечують бокові й передні огляди простору. Відповідно до нечітких правил робот іде з максимальною швидкістю до шуканого об'єкта, якщо із трьох сторін немає перешкод, і гальмує з появою перешкоди з будь-якої сторони.

До складу нечіткого вирішувача робота входять процесор Z-80, мікроконтролер Motorola 68HC11, три контролери Motorola 68040 та дві відеокарти. Інтелектуальне програмне забезпечення робота орієнтоване на використання в операційних середовищах OS-9, UNIX та MSDOS. Робот «MARGE» завоював перше місце на 11-й національній конференції із ШІ, що відбулася в США в 1993 році.

14.3.2. Нечітка система керування роботою світлофора

Нечіткі КСШІ широко використовуються в системах керування роботою світлофорів, розташованих на перехрестях доріг з інтенсивним рухом автотранспорту. У звичайному світлофорі час циклу роботи в режимах зеленого й червоного світла є фіксованим. Це створює певні труднощі руху транспорту у години «пік», що часто призводить до автомобільних «пробок».

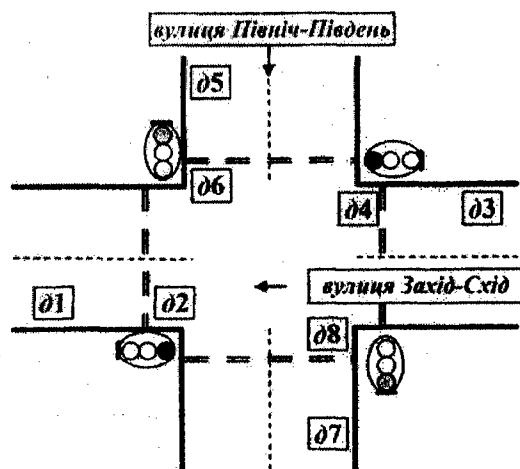


Рис. 14.5 — Схема керування рухом автотранспорту на перехресті

У нечіткому світлофорі час циклу зеленого світла змінюється залежно від кількості машин, що під'їжджають до перехрестя. Для

забезпечення роботи нечіткого світлофора (наприклад, на перехресті вулиць Захід-Схід (ЗС) і Північ-Південь (ПП)) установлюються чотири пари датчиків, які підраховують число машин, що проїхали повз них (рис. 14.5). Світлофор використовує різниці показань датчиків: ($\delta 1 - \delta 2$), ($\delta 3 - \delta 4$), ($\delta 5 - \delta 6$) і ($\delta 7 - \delta 8$). Загальна кількість машин, що не встигли проїхати перехрестя за один цикл зеленого світла світлофора, визначається таким чином:

$$\text{на вулиці ЗС: } M_{ЗС} = (\delta 1 - \delta 2) + (\delta 3 - \delta 4),$$

$$\text{на вулиці ПП: } M_{ПП} = (\delta 5 - \delta 6) + (\delta 7 - \delta 8).$$

Вхідна інформація, необхідна для роботи світлофора, може бути подана як такі *вхідні лінгвістичні перемінні*:

- 1) час циклу зеленого світла світлофора на вулиці ЗС;
- 2) час циклу зеленого світла світлофора на вулиці ПП;
- 3) кількість машин $M_{ЗС}$ на вулиці ЗС;
- 4) кількість машин $M_{ПП}$ на вулиці ПП.

Для кожного виду цих змінних визначаються такі *терми* з відповідними *інтервалами* зміни їхніх значень:

- *час циклу зеленого світла*:
малий ($0 \div 60$), середній ($40 \div 130$), великий ($110 \div 180$) с;
- *кількість машин на вулицях*: *мале* ($1 \div 20$), *середнє* ($15 \div 35$), *велике* ($30 \div 50$).

Вихідною лінгвістичною змінною, яка характеризує роботу світлофора, вважається *зміна часу циклу зеленого світла* залежно від числа машин, що зібралися на обох вулицях. Терми цієї змінної мають наступні значення (с.): *зменшити* ($-20 \div -5$), *не змінювати* ($-10 \div +10$), *збільшити* ($+5 \div +20$).

Зв'язок між вхідними та вихідними лінгвістичними змінними, що описує роботу світлофора з нечіткою логікою, можна подати як *таблицю нечітких правил* 14.2, у якій прийняті такі, наприклад, позначення:

- запис $(+10 \text{ с ЗС})$ означає:
«час циклу зеленого світла на вулиці ЗС збільшити на 10 с»;
- запис (-15 с ПП) означає:
«час циклу зеленого світла на вулиці ПП зменшити на 15 с».

Кожен запис у табл. 14.2 відповідає нечіткому *продукційному* правилу, наприклад:

Якщо (час циклу зеленого світла на вулиці ЗС = середнє)
і (кількість машин на вулиці ЗС = велике),
і (час циклу зеленого світла на вулиці ПП = велике),
і (кількість машин на вулиці ПП = мале),
то (час зеленого світла на вулиці ЗС = збільшити на 10 с),
і (час зеленого світла на вулиці ПП = зменшити на 20 с).

Таблиця 14.2 — Нечіткі правила керування зеленим світлом світлофора

Час циклу зеленого світла на вулиці ЗС (с)	Кількість машин на вулиці ЗС	Час циклу зеленого світла на вулиці ПП (с)								
		Малий (15÷25)			Середній (20÷40)			Великий (35÷60)		
		Число машин на вулиці ПП								
		Мале	Середнє	Велике	Мале	Середнє	Велике	Мале	Середнє	Велике
		Зміна часу циклу зеленого світла світлофора								
Мале (15÷25)	Мале	0	+10с ПП	+15 с ПП	-5 с ПП +5 с ЗС	0	+5 с ПП	-20 с ПП	-10 с ПП	0
	Середнє	+10 с ЗС	+5с ЗС +5с ПП	+10 с ПП	+10 с ЗС -10 с ПП	+5 с ЗС	+5 с ПП	+10 с ЗС -10 с ПП	+5 с ЗС -5 с ПП	0
	Велике	+20 с ЗС	+10с ЗС	+5 с ЗС +5 с ПП	+10 с ЗС -10 с ПП	+5 с ЗС -5 с ПП	+10 с ЗС	+15 с ЗС -15 с ПП	+10 с ЗС -10 с ПП	+15 с ЗС
Середнє (20÷40)	Мале	-5 с ЗС	-5с ЗС +5с ПП	-5 с ЗС +10 с ПП	0	+5 с ПП	-5 с ЗС +5 с ПП	-10 с ПП	-5 с ПП	-5 с ЗС
	Середнє	0	+5с ПП	-5 с ЗС +10 с ПП	+5 с ЗС	0	+5 с ПП	+5 с ЗС -15 с ПП	+5 с ЗС -5 с ПП	0
	Велике	+5 с ЗС	+5с ЗС +5с ПП	+10 с ПП	-10 с ПП	-5 с ПП	0	+10 с ЗС -20 с ПП	+10 с ЗС -10 с ПП	+10 с ЗС
Велике (35÷60)	Мале	-10 с ЗС +10 с ПП	-15с ЗС +15с ПП	-20 с ЗС +20 с ПП	-5 с ЗС +5 с ПП	-10 с ЗС +10 с ПП	-15 с ЗС +15 с ПП	-20 с ЗС -20 с ПП	-20 с ЗС -10 с ПП	-15 с ЗС
	Середнє	-5 с ЗС	-10с ЗС +10с ПП	-15 с ЗС +15 с ПП	0	-5 с ЗС +5 с ПП	-10 с ЗС +10 с ПП	-10 с ЗС -15 с ПП	-10 с ЗС -10 с ПП	-15 с ЗС -5 с ПП
	Велике	0	-5с ЗС +5с ПП	-10 с ЗС +10 с ПП	0	0	-5 с ЗС +5 с ПП	-20 с ПП	-10 с ПП	-5 с ЗВ -5 с ПП

14.4. Области застосування нечітких інтелектуальних систем

Нечіткі КСШІ застосовуються у багатьох областях людської діяльності, де доводиться оперувати з невизначеною й неточною інформацією. До них належать системи керування та прийняття рішень, системна екологія, макроекономіка та інші Про. Нечіткі системи використовуються у сферах фінансового аналізу, медицини, законодавства, юриспруденції тощо.

Перші результати ефективного використання нечітких систем отримали промисловці Японії в дешевих виробках масового ринку — пілососах, відеокамерах, мікрохвильових печах тощо. Піонером у застосуванні нечітких систем у побутових виробках виступила фірма *Matsuhita*. У 1991 році вона анонсувала першу *інтелектуальну пральну машину*, в системі керування якої поєднувалися *нечітка логіка й нейронна мережа*. Автоматично визначаючи нечіткі вхідні чинник (обсяг і якість білизни, рівень забруднення, тип порошку тощо), пральна машина безпомилково вибирала оптимальний режим прання із 3 800 можливих. Після цього застосування нечітких систем у японській побутовій техніці стало повсюдним.

Нині за допомогою нечітких КСШІ реалізуються багато проєктів — від фотокамер, пілососів, пральних машин, автомобілів у споживчому секторі економіки до контролерів потокового виробництва, автоматичних прокатних станів, автобусних розкладів, систем керування, інтелектуальних роботів, безлюдних виробництв, атомних реакторів, аерокосмічних систем і систем військового призначення.

Для вирішення надскладних задач *прогнозування фінансових операцій* у Японії і США створюються дорогі комплексні інтелектуальні системи, до складу яких входять і нечіткі системи. На початку 1990 року японська фінансова корпорація *Yamaichi Securities* поставила собі за мету автоматизувати гру на ринку цінних паперів, для чого залучила до роботи близько 30 фахівців зі штучного інтелекту. У першу версію нечіткої системи ввійшли 600 нечітких правил — утілення досвіду десяти ведучих брокерів корпорації. Перш ніж зважитися на використання нової системи в реальних умовах, її протестували на дворічній вибірці фінансових даних. Система блискуче витримала іспит, спрогнозувавши, зокрема, біржовий крах на токійській біржі в 1988 році. Після цього питання про доцільність застосування нечітких систем у фінансовій сфері вже не піднімалося.

На початку 80-х років у Японії, а потім і в США в секретній обстановці було розгорнуто комплексні роботи з використання

нечіткої логіки в різних оборонних проектах. Одним із застосувань нечітких систем стало створення *нечіткого керуючого мікропроцесора*, за допомогою якого вирішувалася задача керування мобільною зенітною ракетою, яка знищує міжконтинентальні ракети супротивника. Подібні системи давали змогу вирішувати й зворотню задачу, тобто розробляти маневри для ефективного відходу від антиракет.

На сьогодні в Японії функціонує спеціально створена «*Laboratory for International Fuzzy Engineering Research (LIFE)*», яка поєднує 48 компаній, серед яких такі відомі японські фірми, як «*Hitachi*», «*Mitsubishi*», «*NEC*», «*Sharp*», «*Sony*», «*Honda*», «*Mazda*», «*Toyota*». Із закордонних (не японських) учасників «*LIFE*» можна виділити компанії «*IBM*» та «*Fuji Xerox*». До діяльності «*LIFE*» виявляє інтерес космічне агентство США «*NASA*», яке займається застосуванням нечітких систем для здійснення маневрів та комплексного стикування космічних апаратів.

ДОДАТКИ

Додаток 1

Контрольні запитання та завдання

Розділ 1. Інтелект людини

- 1.1. Опишіть особливості побудови й функціонування людського мозку.
- 1.2. Які види мислення відбуваються в людському мозку?
- 1.3. У чому сутність свідомого та підсвідомого мислення людини?
- 1.4. Охарактеризуйте особливості лівопівкульного та правопівкульного мислення людини.
- 1.5. Що таке «інтелект»? Які асоціативні образи викликає у Вас це поняття? Дайте власне трактування цього терміна.
- 1.6. Охарактеризуйте поняття «розум», «інтелект», «мудрість» та дайте їх власне трактування.
- 1.7. Чи існує зв'язок рефлексії та інтуїції з інтелектом людини?
- 1.8. Якими розумовими здібностями (властивостями), на Ваш погляд, повинна володіти інтелектуальна людина?
- 1.9. Чи може бути інтелектуальною людина, яка не має духовних (моральних) якостей?
- 1.10. Як визначити рівень (ступінь) інтелектуальності людини? Що Вам відомо про способи оцінки інтелекту?
- 1.11. Який зміст вкладається в поняття «інтелектуальна задача»?
- 1.12. Назвіть відомі Вам інтелектуальні задачі? У чому полягає їх інтелектуальність?
- 1.13. Чи є інтелектуальними математичні (обчислювальні) задачі?
- 1.14. Дайте порівняльну характеристику інтелектуальних і математичних задач.

Розділ 2. Штучний інтелект у комп'ютерній системі

- 2.1. Який зміст вкладається в поняття «штучний інтелект» (ШІ)? Наведіть власну думку про призначення ШІ.
- 2.2. Дайте характеристику основних напрямів розвитку ШІ.
- 2.3. У чому полягає сутність нейробіонічного та кібернетичного напрямів ШІ? Які задачі вирішуються в межах кожного напрямку ШІ?

2.4. Охарактеризуйте основні задачі фундаментальних наукових досліджень в області ШІ.

2.5. Що являє собою ШІ в комп'ютерній системі?

2.6. Які типи моделей мислення створюються в межах основних напрямів розвитку ШІ?

2.7. Що таке «евристика»? Як вона використовується в КСШІ?

2.8. Чи є, на Ваш погляд, потреба в інтелектуалізації комп'ютерних технологій? Якими можуть бути її соціальні наслідки?

2.9. Що таке комп'ютерна (програмно-апаратна) модель інтелекту?

2.10. У чому полягає відмінність між нейробіонічними та кібернетичними моделями інтелекту?

2.11. Чи всі властивості людського інтелекту можна змоделювати в комп'ютері?

2.12. Охарактеризуйте особливості й труднощі комп'ютерного моделювання інтелекту людини.

2.13. Дайте порівняльну характеристику моделей лівопівкульного та правопівкульного мислення людини.

2.14. Опишіть основні властивості комп'ютерних моделей інтелекту.

2.15. Матеріали яких видань і публікацій в області ШІ Ви використовуєте у своєму навчанні (роботі)?

Розділ 3. Принципи побудови та функціонування комп'ютерних систем штучного інтелекту

3.1. Сформулюйте визначення комп'ютерної системи штучного інтелекту (КСШІ). Які Ваші уявлення про КСШІ?

3.2. Чому і в яких областях людської діяльності виникла проблема створення КСШІ?

3.3. Що означає поняття «Предметна (проблемна) область» (ПрО)?

3.4. Які ПрО називають важкоформалізованими? Наведіть приклади таких ПрО.

3.5. У чому полягає відмінність КСШІ від традиційної комп'ютерної (обчислювальної) системи?

3.6. Опишіть основні проблеми, з якими стикаються творці КСШІ.

3.7. Які види (типи) інтелектуальних систем Вам відомі?

3.8. Які предметні (проблемні) області практичного застосування КСШІ Ви знаєте?

3.9. Які основні задачі вирішує КСШІ?

3.10. Опишіть основні властивості інтелектуальності КСШІ. Дайте їх критичний аналіз.

3.11. Що Вам відомо про способи та методи оцінки рівня (ступеня) інтелектуальності КСШІ?

3.12. Охарактеризуйте загальні вимоги, що пред'являються до КСШІ при вирішенні інтелектуальних задач.

3.13. Опишіть функціональну структуру перспективної КСШІ. Які основні підсистеми входять до її складу?

3.14. Які програмно-апаратні пристрої входять до складу підсистем КСШІ?

3.15. У чому полягають функції вирішувача та бази знань КСШІ?

3.16. До яких програмно-апаратних пристроїв КСШІ можна застосувати вислів: «Мова, близька до людського спілкування»?

3.17. Чи є відмінність між блоком спілкування та блоком пояснень КСШІ?

3.18. Що вкладається у поняття «когнітивна графіка»?

3.19. Фахівці яких професій беруть участь у створенні й експлуатації КСШІ?

3.20. Чим займаються інженер по знаннях, експерт, аналітик, когнітолог, програміст?

3.21. Охарактеризуйте етапи створення КСШІ.

3.22. Назвіть основні конструкторсько-технологічні задачі, розв'язувані на окремих етапах створення КСШІ.

Розділ 4. Основні аспекти інженерії знань

4.1. Який зміст вкладається у поняття «знання» і «дані»? У чому їх принципова відмінність?

4.2. Що означає термін «інженерія знань»? Для чого призначена ця технологія?

4.3. Які джерела знань використовує КСШІ у своїй роботі?

4.4. Назвіть основні види знань і наведіть їх характеристики. Які види знань мають найбільше застосування в КСШІ?

4.5. Що таке глибинні, латентні, приховані знання? Що в них спільного та відмінного?

4.6. Поясніть сутність метазнань. У чому полягає їх відмінність від інших видів знань?

4.7. Дайте порівняльний аналіз нечітких, неточних та невизначених знань?

4.8. Що являють собою прецедентні та стереотипні знання? Що в них спільного та відмінного?

4.9. Що таке оболонка (протоструктура) інформаційної одиниці знання?

4.10. Опишіть основні *властивості* знань. Як вони впливають на ефективність роботи КСШ?

4.11. Поясніть зміст понять «*придбання*», «*добування*», «*витяг*», «*розкопка знань*»? Що в них спільного та відмінного?

4.12. У чому полягає процес *добування знань*? Яка його мета?

4.13. Які *фахівці* беруть участь у процесі здобування знань?

4.14. Чому знання експерта важко формалізувати та ввести до бази знань?

4.15. Опишіть *класи методів витягу знань*. Дайте їх порівняльну характеристику.

4.16. Охарактеризуйте методи витягу *латентних (глибинних)* знань.

4.17. Які, на Ваш погляд, *особливості й труднощі* процесу *добування латентних (глибинних)* знань?

4.18. Поясніть зміст та особливості *комп'ютерного витягу* знань.

4.19. Наведіть основні характеристики й особливості *технології добування знань Data Mining*. Дайте порівняльну характеристику цієї технології та інших способів витягу *глибинних знань*.

4.20. Що розуміється під терміном «*керування знаннями*»? У чому сутність проблеми *керування знаннями*?

4.21. Охарактеризуйте роль і функції *корпоративної пам'яті* інтелектуальної системи *керування знаннями*.

Розділ 5. Бази знань

5.1. Що містить у собі поняття «*модель предметної області*»? Опишіть структурні компоненти цієї моделі.

5.2. У чому полягає сутність *вербалізації (вербальності)*? Як вона використовується в КСШ?

5.3. Охарактеризуйте *елементарні логічні форми моделі предметної області (поняття, сутність, судження, твердження, вираження, пропозиція, висловлення, міркування тощо)*.

5.4. Що являє собою «*база знань*» КСШ? Опишіть *технологію (етапи)* її створення.

5.5. У чому полягає *принципова відмінність бази знань від бази даних*, що використовується в *обчислювальних системах*?

5.6. Чи присутня *база даних* у КСШ і для чого вона призначена?

5.7. Які *проблеми* доводиться вирішувати при створенні *бази знань*?

5.8. Охарактеризуйте основні *функціональні елементи бази знань (факти, правила, метаправила тощо)*.

5.9. Опишіть структуру та функції *бази знань*. Які *підобласті знань* є у *базі знань*?

5.10. Розробіть *приклад бази знань* будь-якої предметної області, що Вам найбільш відома.

5.11. Що таке *онтологія*? Для чого вона використовується в КСШ?

5.12. У чому сутність поняття «*концепт*»? Які знання називаються *концептуальними*?

5.13. Опишіть структуру *формальної моделі онтології*.

5.14. Для чого призначена *онтологічна система*? Опишіть її структуру та функції.

5.15. Виберіть предметну область, яка Вам найбільше знайома, і розробіть для неї *онтологію* в термінах: *ключові поняття, сутності, відносини, предикати, операції*.

5.16. Наведіть *приклад практичних розроблень онтологічних систем*.

Розділ 6. Моделі подання знань

6.1. Які *типи моделей подання знань* Вам відомі? Дайте їх основні характеристики.

6.2. Дайте визначення і наведіть характеристики *логічної моделі подання знань*.

6.3. У чому полягає зміст *предиката, кон'юнкції, диз'юнкції, імплікації, фрази Хорна, кванторів існування та узагальнення*? У яких моделях подання знань вони використовуються?

6.4. Опишіть процес формування *логічної бази знань* із використанням логіки предикатів.

6.5. Подайте за допомогою *логіки предикатів* і *фраз Хорна* такі *твердження*:

«Будь-який чоловік може покохати жінку, яка кохає іншого чоловіка».

«Немає двох філософів, які мали б одну і ту саму думку».

6.6. Що являє собою *семантична модель подання знань*? Чому вона називається *семантичною мережею*?

6.7. Які *типи відносин* використовуються у *семантичній моделі*? Наведіть приклади.

6.8. Дайте визначення *фрейму* та опишіть *фреймову модель подання знань*.

6.9. Яку роль у *фреймовій моделі* грають *протофрейм* і *екзофрейм*?

6.10. Які *типи фреймів* використовуються у *фреймових моделях*?

6.11. Наведіть основні характеристики *слотів* *фреймової моделі*.

6.12. У чому сутність *спадкування* у *фреймовій базі знань*?

6.13. Що таке *демон* і *приєднана процедура* у *фреймових моделях*?

6.14. Як здійснюється *рішення задачі* користувача з використанням *фреймової бази знань*?

6.15. Наведіть основні характеристики *продукційної моделі* подання знань.

6.16. Дайте визначення *продукції* та її складових частин. Які види продукцій Вам відомі?

6.17. Подайте у вигляді *продукції* наступне твердження: «Двигун автомобіля не заводиться за відсутності бензину та несправностей акумуляторної батареї, карбюратора або розподільника запалювання».

6.18. Опишіть функціональні властивості *продукційної бази знань*.

Розділ 7. Створення нових знань в інтелектуальних системах

7.1. Що означає «*створення нових знань*» в інтелектуальній системі?

7.2. У чому полягає сутність *механізму створення нових знань* у людини й у КСШІ?

7.3. Охарактеризуйте відомі вам *різновиди створення нових знань*.

7.4. Що являють собою *цілі й підцілі* пошуку рішення задачі у процесі *створення нових знань*?

7.5. Як реалізується в КСШІ *механізм вибору знань* людиною при розв'язанні інтелектуальних задач?

7.6. Опишіть *рівні інтелектуальності задач*, розв'язуваних у КСШІ.

7.7. Що являє собою *машина виводу створення нових знаків* в інтелектуальній системі? Як працює *машина створення нових знань*?

7.8. Що таке «*конфліктна множина правил*» і як вирішуються конфлікти?

7.9. Дайте визначення й опишіть відомі Вам *стратегії керування створенням нових знань*.

7.10. У чому полягають *пряма (висхідна) й зворотна (спадна) стратегії* створення нових знань?

7.11. Використовуючи *пряму і зворотну стратегії* покажіть, що твердження «*несправний блок харчування*» логічно випливає з наступного набору *фактів і правил продукційної бази знань*:

«Комп'ютер не вмикається», «Напруга мережі в нормі», «Шнур живлення не обірваний»;

«Якщо комп'ютер не вмикається, *то* відсутня напруга в мережі або обірваний шнур живлення, або несправний блок живлення».

7.12. Що являють собою *стратегії пошуку «у глибину» й «у ширину»*, які використовуються у процесі перебору правил у базі знань при розв'язанні задач КСШІ?

Розділ 8. Методи розв'язання інтелектуальних задач

8.1. У чому полягає *метод резолюцій* вирішення логічних задач? Сформулюйте правила *методу резолюцій*. Наведіть приклад розв'язання задачі з використанням методу резолюцій.

8.2. Дано *базу знань*, яка містить факти: «Вовк їсть м'ясо», «Заєць їсть моркву» і правило-імплікація: «Якщо хтось їсть м'ясо, то він є хижаком».

Запишіть ці знання у вигляді *фраз Хорна* і перевірте за допомогою *методу резолюцій*, чи є хижакими вовк і заєць.

8.3. Що означає термін «*розпізнавання образів*»? Які задачі *розпізнавання образів* Вам відомі?

8.4. Наведіть *постановки задач* розпізнавання образів.

8.5. Охарактеризуйте основні *методи розв'язання задач* розпізнавання образів. Наведіть приклад розв'язування такої задачі.

8.6. Охарактеризуйте концептуальні основи *еволюційної теорії природного добору* живих істот у природі.

8.7. Дайте визначення *популяції, істоти, організму, генетичного коду, генотипу, генофонду, хромосоми, гена*.

8.8. Що являє собою *генетичний алгоритм (ГА)* розв'язання інтелектуальних задач? Яким чином подаються в ГА *генетичний код, популяції, істоти, хромосоми, гени*? Як використовуються в ГА *функції пристосовності, придатності, адаптації, фітнес-функції*?

8.9. Для чого призначені *генетичні оператори*? Що таке *кроссовер, кроссинговер, схрещування, мутація, інверсія, селекція, репродукція, копіювання*?

8.10. Опишіть функціональну схему ГА.

8.11. Наведіть *приклади* і вкажіть *особливості* практичного застосування ГА.

8.12. Що являє собою *метод групового урахування аргументів (МГУА)*? Які механізми еволюції покладені в основу МГУА?

8.13. Опишіть *алгоритм перебору часткових моделей* у *багаторядному процесі селекції* МГУА.

8.14. Які види алгоритмів МГУА Вам відомі? У чому полягає їх відмінність?

8.15. У яких предметних (проблемних) областях застосовуються алгоритми МГУА?

8.16. У чому сутність *графових методів* розв'язання інтелектуальних задач?

8.17. Що таке *граф*? Опишіть основні *види (типи) графів*. Наведіть приклад «і»-«або»-графа.

8.18. Охарактеризуйте відомі Вам *алгоритми* розв'язання задач на графах. У чому полягає процес пошуку розв'язку задачі на графі?

8.19. Дайте порівняльний *аналіз алгоритмів* пошуку розв'язку задач на графах.

8.20. Який підхід до вирішення інтелектуальних задач називають *прецедентним*? Яка властивість мислення людини використовується у цьому разі?

8.21. Опишіть відомі Вам *прецедентні алгоритми розв'язання задач*.

8.22. Чи існує аналогія між алгоритмами, заснованими на *правилах і прецедентах*?

8.23. Що являють собою *алгоритми конструювання* процесу розв'язання задач?

8.24. Охарактеризуйте *висхідну та спадну стратегії конструювання*.

Розділ 9. Експертні системи

9.1. Що являє собою *експертна система (ЕС)*? Які фахівці беруть участь у її створенні?

9.2. До якого *напрямку розвитку КСШІ* належать ЕС? Які *моделі мислення* створюються й використовуються в межах цього напрямку?

9.3. Як здійснюються процеси функціонування ЕС та її спілкування з *користувачем*?

9.4. Охарактеризуйте *динамічні ЕС*. У чому їх принципова відмінність від *статичних ЕС*?

9.5. Опишіть *структуру й функції динамічної ЕС*.

9.6. Які *види (типи) динамічних ЕС* вам відомі?

9.7. Опишіть особливості технології створення ЕС.

9.8. У чому сутність *концепції «швидкого прототипу»* при створенні ЕС? Які *види прототипів* Ви знаєте?

9.9. Що таке *«оболонка» ЕС*?

9.10. Що являє собою *інтелектуальна інструментальна система (ІІС)*?

9.11. Які *технології ІІС* використовуються нині для створення ЕС?

9.12. Охарактеризуйте відомі Вам *типи (види) ІІС*.

9.13. У яких сферах людської діяльності *використовуються ЕС*?

9.14. Які *основні задачі* вирішує ЕС? У яких предметних областях ЕС набули найбільшого поширення?

9.15. Наведіть основні характеристики *медичної діагностичної ЕС «Mycin»*.

9.16. Охарактеризуйте основні *версії ЕС хімічних досліджень «Dendral»*.

9.17. Опишіть основні задачі та функції *ЕС терапії онкологічних захворювань «Oncosin»*.

9.18. Для чого призначена і які функції виконує *ЕС проектування конфігурацій комп'ютерів «R1 / XCon»*?

9.19. Які *ЕС навчання й контролю знань* Ви знаєте?

9.20. Охарактеризуйте предметні області *практичного застосування ЕС*. Які приклади практичного використання експертних систем в Україні Вам відомі?

Розділ 10. Нейрокомп'ютерні системи

10.1. Що являють собою *нейрокомп'ютерні системи*? У межах якого *напрямку розвитку КСШІ* вони створюються і які *моделі мислення* використовують?

10.2. Що називають *нейронною мережею, нейрокомп'ютером, нейрокомп'ютингом*?

10.3. Як улаштована *біологічна нейронна мережа*? Що являють собою *дендрити, аксони, синапси*? Які *особливості функціонування біологічної нейромережі*?

10.4. Опишіть *структуру штучного нейрона*? Що називають *активаційною функцією* нейрона? Що таке *сигмоїд*?

10.5. Що являє собою *штучна нейронна мережа*? Як здійснюється процес функціонування нейромережі?

10.6. У чому сутність процесу *навчання* нейронних мереж? Які *види навчання* нейромереж Вам відомі?

10.7. Що таке *змагальне навчання*? У чому полягають принципи *«латерального гальмування»* нейронів мережі та механізм навчання *«Переможець одержує все»*?

10.8. Дайте порівняльну характеристику методик навчання нейромережі *«з учителем»* та *«без учителя»*.

10.9. Які *види (типи) нейромереж* Вам відомі? Що являє собою *багатошарова нейронна мережа*? Які нейронні шари називаються *прихованими*?

10.10. Поясніть роботу *персептрона Розенблатта* і *багатошарового персептрона*.

10.11. У чому сутність *алгоритму зворотного поширення похибки*?

10.12. Дайте характеристику алгоритму навчання *Хейбба*. До якого типу методів навчання він належить?

10.13. Що являють собою *нейромережа і карта Кохонена*? Який вид навчання в них використовується? Чому цей тип нейромережі називають *самоорганізованою структурою*?

10.14. Дайте порівняльну характеристику *нейромереж Хопфілда й Хеммінга*. У чому їх спільність і відмінність?

10.15. Які *види зірок Гроссберга* Ви знаєте? Як вони використовуються в нейромережі *зустрічного поширення*?

10.16. Опишіть *структуру та принцип функціонування когнітрона й неокогнітрона Фукушими*. Що в них спільного та відмінного?

10.17. Що являють собою *нейромережі з адаптивним резонансом*? У чому полягає проблема *стабільності-пластичності* нейромереж?

10.18. Опишіть принцип роботи нейромереж *APT-1*, *APT-2*, *APT-3*. Дайте їм порівняльну оцінку.

10.19. Охарактеризуйте *проблемні області* застосування нейрокомп'ютерних систем. Які класи задач вони вирішують?

10.20. У яких предметних областях знаходять *практичне застосування* нейрокомп'ютерні системи? Наведіть конкретні приклади.

10.21. У чому полягають *особливості* технології створення нейрокомп'ютерних систем?

10.22. Чи відомі Вам приклади практичного використання нейрокомп'ютерних систем *в Україні*?

Розділ 11. Робототехнічні інтелектуальні системи

11.1. Що являють собою *робототехнічні інтелектуальні системи (РТІС)*?

11.2. У чому сутність *антропоморфного підходу* до побудови РТІС?

11.3. Які *підсистеми* є у складі РТІС?

11.4. Як здійснюється процес функціонування *РТІС*? У яких режимах вона може працювати?

11.5. Які апаратно-програмні засоби збирання й перетворення інформації присутні у РТІС?

11.6. Опишіть структуру та функції *підсистеми машинного зору* РТІС.

11.7. Як здійснюється формування *акустичної (звукової, мовної) інформації* у РТІС?

11.8. Охарактеризуйте склад та функції *сенсорної підсистеми* РТІС. Як здійснюється формування *тактильної (контактної) інформації*?

11.9. Які типи *аналізаторів* Вам відомі? Які види інформації вони формують?

11.10. Які *моделі подання інформації* у РТІС відображають функції *першої та другої сигнальних систем* сприйняття людиною зовнішнього світу?

11.11. Що являє собою *лінгвістичний процесор*? Які функції він виконує?

11.12. Опишіть структуру та функції *підсистеми вироблення й прийняття рішень*.

11.13. У чому полягають особливості роботи *вирішувачів* РТІС? Які *робототехнічні задачі* вони вирішують? Наведіть *приклад* такої задачі.

11.14. Яку роль грає *планувальник* РТІС?

11.15. Як здійснюється *керування* процесом функціонування РТІС? Які функції виконують *підсистеми зв'язку та керування*?

11.16. Охарактеризуйте *автоматичний та ручний* режими керування системою. Що являє собою *віддалений операторний пункт керування*?

11.17. Охарактеризуйте склад та призначення *виконавчих пристроїв і механізмів* РТІС.

11.18. Що являють собою *колективні РТІС*? Опишіть функції *базового комплексу й автономних роботів*. У чому полягають характерні риси *колективів міні- та мікророботів*?

11.19. Охарактеризуйте області практичного застосування РТІС. У яких *предметних областях* України вони можуть знайти застосування?

Розділ 12. Інтелектуальні системи прийняття рішень

12.1. Що являють собою *інтелектуальні системи прийняття рішень (ІСПР)*? Наведіть приклади предметних областей, де необхідно створювати ІСПР.

12.2. Які *особливості* предметної (проблемної) області обумовлюють необхідність створення й використання ІСПР?

12.3. У чому полягає сутність *колективного розуму* й «*мозкової атаки*»?

12.4. Які *розумові процедури* містить у собі процес прийняття рішень?

12.5. Як здійснюється процес функціонування ІСПР?

12.6. Які *моделі й методи* прийняття рішень використовуються в ІСПР?

12.7. Опишіть *структуру й функції* ІСПР. Які функції виконують *підсистеми керування моделями, еволюційної адаптації, аналізу та вибору альтернативних рішень*?

12.8. Охарактеризуйте області практичного застосування ІСПР. Чи можуть бути використані ІСПР для виходу із кризової ситуації *в Україні*?

Розділ 13. Мультиагентні інтелектуальні системи

13.1. Що являють собою *мультиагентні системи (МАС)*? У чому їх основне призначення?

13.2. У чому полягають сутність й особливості *мультиагентних internet-систем (МАС-Internet)*?

13.3. Опишіть *структуру й функції* МАС.

13.4. Охарактеризуйте основні *проблеми*, з якими стикаються розробники МАС під час її практичної реалізації.

13.5. Що являють собою *агенти* МАС? Чому вони називаються *інтелектуальними*? Як здійснюється процес *взаємодії агентів*?

13.6. Які типи *агентів* функціонують у складі МАС? Опишіть їхні основні *властивості* та *функції*.

13.7. У чому полягають *активність, самоконтроль, реактивність, адаптивність, толерантність, соціальна поведінка, доброзичливість* агентів?

13.8. Що являють собою та які ролі грають *диспетчер, брокер, менеджер, кодувальник, візуалізатор, агент Internet*?

13.9. Які функції виконують *інтерфейсні* та *мобільні* агенти?

13.10. Що називають *ментальною моделлю* інтелектуального агента?

13.11. Охарактеризуйте області практичного застосування МАС. У яких *предметних областях* застосування МАС найбільш ефективно?

13.12. Наведіть приклади *практичних розроблень* МАС.

Розділ 14. Нечіткі інтелектуальні системи

14.1. Що являють собою *нечіткі інтелектуальні системи (НІС)*? Які основні *концепції* покладені в основу НІС?

14.2. Як здійснюється процес функціонування НІС?

14.3. Охарактеризуйте *структуру* та *функції* НІС. Які *підсистеми (блоки)* входять до складу НІС?

14.4. Що являють собою *блоки фазифікації* та *дефазифікації* у НІС? Яку роль у роботі НІС вони відіграють?

14.5. У чому полягають особливості *нечіткого інтелектуального вирішувача*?

14.6. Які знання називають *нечіткими*? У чому їх відмінність від *невизначених, неточних, неповних* знань?

14.7. Що таке *нечітка логіка, нечіткі моделі, нечіткі множини* та *нечіткий вивід на знаннях*? Які обчислення називають *м'якими*? Які операції з *нечіткими множинами* Ви знаєте?

14.8. Дайте визначення *лінгвістичної перемінної, терму, функції належності*.

14.9. У чому полягають особливості структури та функціонування *бази нечітких знань* і *нечіткого блоку спілкування* щодо аналогічних пристроїв КСШІ?

14.10. Опишіть структуру та функції *нечіткої робототехнічної системи* та *нечіткої системи керування* роботою світлофора.

14.11. Охарактеризуйте області практичного застосування НІС. У яких областях вони набули поширення?

14.12. Чи відомі Вам приклади практичних розроблень НІС в Україні?

Словник термінів

Інтелект — *ступінь (рівень) розумових здібностей (мислення) та духовних якостей людини, який дають їй змогу розв'язувати інтелектуальні задачі.*

Інтелектуальна задача — *творча задача, розв'язувана людиною у процесі її життєдіяльності.*

Мислення — *розумовий процес постановки й розв'язання інтелектуальної задачі.*

Штучний інтелект (наука) — *науковий напрям, орієнтований на створення методів і засобів моделювання інтелекту людини.*

Штучний інтелект (комп'ютерний) — *програмно-апаратна модель інтелекту людини.*

Комп'ютерна система штучного інтелекту (інтелектуальна система) — *комплекс взаємозв'язаних інтелектуальних комп'ютерних засобів, призначений для вирішення інтелектуальних задач у певній предметній (проблемній) області.*

Предметна область — *сфера застосування інтелектуальної системи в певній області людської діяльності (бізнес, виробництво, медицина тощо).*

Проблемна область — *клас інтелектуальних задач (проблем), які розв'язує інтелектуальна система в певній предметній області.*

Клас задач — *структурована сукупність задач, виокремлена й згрупована за певними формальними ознаками або правилами.*

Інженерія знань — *технологія розроблення й використання інтелектуальних методів та засобів добування, витягу, подання й опрацювання знань.*

Знання — *нагромаджена людством евристична інформація щодо закономірностей та властивостей об'єктів (процесів, явищ, ситуацій) зовнішнього світу.*

База знань — *комп'ютерна пам'ять інтелектуальної системи, в якій зберігаються знання про дану предметну (проблемну) область.*

Евристика — *неформальне словесне правило, установлене дослідним шляхом, яке не має строгого (теоретичного, математичного) обґрунтування й використовується в інтелектуальній системі для подання інформації.*

Вербалізація — *евристичний (словесно-змістовний) опис предметної (проблемної) області.*

Асоціативна база знань — комп'ютерна *пам'ять* інтелектуальної системи, в якій *пошук інформації* здійснюється за її *вмістом*.

Асоціативність — *стереотипний зв'язок між різнорідними поняттями, об'єктами, процесами, явищами*.

Онтологія (наука) — *наука про буття*, яка трактує сутність абстрактних *філософських категорій* (простір, час, причина, наслідок, дія, явище тощо).

Онтологія (комп'ютерна) — *база (сховище) концептуальних знань про сукупність об'єктів, процесів та явищ навколишнього світу і зв'язки між ними*.

Концепт — *узагальнене поняття*, яке характеризує певний *клас об'єктів* (процесів, явищ, ситуацій) предметної області згідно з їх специфічними властивостями.

Онтологічна система (комп'ютерна) — *структурований комплекс комп'ютерних онтологій, призначений для пошуку й витягу концептуальних знань із глобальних, корпоративних, локальних систем та мереж, головним чином — із Internet*.

База даних (в інтелектуальній системі) — *сховище загальних відомостей про предметну область, власні можливості системи, розв'язувані задачі, користувачів тощо*.

Об'єкт — *предмет моделювання* (фізичний об'єкт, процес, явище, подія, дія, ситуація тощо) даної предметної області.

Образ — *комп'ютерна модель об'єкта*.

Структурованість — *упорядкування об'єктів (образів) за рівнями ієрархії*.

Ієрархія — *упорядкована сукупність рівнів співвідпорядкованості об'єктів (образів)*.

Аналітик (інженер по знаннях, інженер знань, інженер-інтерпретатор, когнітолог) — *фахівець зі створення й експлуатації інтелектуальної системи*.

Експерт — професіонал у даній *предметній (проблемній) області*, знання якого заносяться до *бази знань* інтелектуальної системи.

Програміст — *фахівець із системного програмного забезпечення інтелектуальної системи, який здійснює вибір (розроблення) моделей подання знань, мов і середовища програмування, найпридатніших для даної системи*.

Користувач — *людина (користувач, аналітик, експерт, програміст, фахівець), що спілкується з інтелектуальною системою для розв'язання своєї задачі*.

Запит користувача — *розв'язувана інтелектуальна задача (завдання), представлена комп'ютерною мовою, наближеною до людського спілкування*.

Вирішувач — *комп'ютер інтелектуальної системи, який моделює мислення людини в процесі розв'язання інтелектуальних задач*.

Інтелектуальний інтерфейс — *програмно-апаратний пристрій, що здійснює діалогове спілкування людини з інтелектуальною системою природною (людською) мовою з використанням інтелектуальних комп'ютерних засобів когнітивної графіки*.

Когнітивність (cognition) — *пізнання, пізнаваність, пізнавальна здатність*.

Когнітивна графіка — *комплекс комп'ютерних засобів інтелектуального інтерфейсу, які використовують колір, анімацію, відео, аудіо та інші засоби мультимедіа й комп'ютерної графіки для наочного (інтегровано-го) подання властивостей досліджуваного об'єкта, процесу або явища*.

Мультимедіа — *інтегроване комп'ютерне середовище, в якій поряд із традиційними засобами взаємодії людини з комп'ютером (алфавітно-цифровий чи графічний дисплей, принтер, клавіатура) використовуються нові інформаційні можливості, зокрема — живий людський голос, музика, кольорові художні та документальні відеоролики, озвучена мультиплікація тощо*.

Гіпермедіа — *інтегроване комп'ютерне середовище, в якому інформаційні засоби мультимедіа об'єднані в гіпертекстову систему*.

Гіпертекстова система — *файлова система, в якій зв'язок між файловими документами здійснюється за допомогою ключових слів або фраз*.

Лінгвістичний процесор — *апаратно-програмний пристрій, який функціонує в режимах «сигнал — текст» (аналіз) і «текст — сигнал» (синтез) і призначений для сприйняття, розпізнавання, формування, подання та відображення символічної (текстової) інформації*.

Діалоговий процесор — *програмно-апаратний пристрій, призначений для опрацювання інтелектуальної інформації, пов'язаної з мовою спілкування, її будовою, граматику, розбором пропозицій і тексту*.

Парадигма — *коротка ключова пропозиція (гасло), що відбиває шляхи для досягнення мети*.

Створення нових знань — *спосіб розв'язання інтелектуальної задачі у комп'ютерній системі*.

Таксономія — *класифікація (система) понять*.

LIPS — *одиниця продуктивності вирішувача інтелектуальної системи (1 логічний вивід у секунду)*.

LISP — *одна з перших мов програмування інтелектуальних систем*.

«Клієнт-сервер» — *архітектура комп'ютерних ресурсів, за якої одна частина застосування (програмного забезпечення розв'язуваної задачі) розміщується на комп'ютері користувача (клієнта), а інша — на сервері використовуваної комп'ютерної системи*.

е-комерція — *серійне виробництво та продаж комп'ютерних засобів*.

ЛІТЕРАТУРА

Розділ 1. Інтелект людини

- 1.1. *Hebb D. O.* The organization of behavior. — N.Y.: John Wiley & Sons, 1949. — 378 p.
- 1.2. *Эшби У. Р.* Конструкция мозга. Происхождение адаптивного поведения. — М.: ИИЛ, 1962. — 392 с.
- 1.3. *Newell A., Simon H. A.* Human problem solving. — Englewood-Cliffs, NJ: Prentice-Hall, 1972. — 384 p.
- 1.4. *Линдсей П., Норман Д.* Переработка информации у человека. — М.: Мир, 1974. — 365 с.
- 1.5. *Прибрам К.* Языки мозга. — М.: Прогресс, 1975. — 315 с.
- 1.6. *Anderson J. R.* Language, memory and thought. — Hillsdale NJ: Lawrence Erlbaum, 1976. — 367 p.
- 1.7. *Schank R. C., Abelson R.* Scripts, plans, goals and understanding. — Hillsdale, NJ: Lawrence Erlbaum, 1977. — 396 p.
- 1.8. *Амосов Н. М.* Алгоритмы разума. — К.: Наукова думка, 1979. — 220 с.
- 1.9. *Гоман Н. Р., Мачинский Н. О.* ЭЭГ — исследование функций организации правого и левого полушарий при решении вербальных и пространственных задач // Журнал высш. нервной деятельности. — 1984. — Т. 34. — № 3. — С. 412–420.
- 1.10. *Бонько В. К., Кулик С. П.* Мышление человека и проблемы компьютеризации. — Минск: Наука і техника, 1992. — 191 с.
- 1.11. *Айзенк Г.* Коэффициент интеллекта. — К.: Гранд, 1994. — 210 с.
- 1.12. *Икколс Дж. Г., Мартин А. Р., Валлас Б. Дж., Фукс П. А.* От нейрона к мозгу. — М.: Едиториал УРСС, 2003. — 672 с.
- 1.13. *Сторож В. В.* Обработка информации у человека // ИИ. — 2003. — № 4. — С. 231–247.
- 1.14. *Сторож В. В.* Алгоритмы, формализация и мышление // ИИ. — 2006. — № 1. — С. 173–187.
- 1.15. *Бондаренко М. Ф., Золкин С. Г., Малокуцко Е. Н.* Анализ взаимосвязей биоритмов головного мозга // ИИ. — 2006. — № 1. — С. 4–11.
- 1.16. *Раменская В. Н.* Инструментарий эмоционального интеллекта // ИИ. — 2007. — № 1. — С. 45–50.

Розділ 2. Штучний інтелект у комп'ютерній системі

- 2.1. *Шеннон К.* Работы по теории информации и кибернетике. — М.: ИИЛ, 1963. — 287 с.

- 2.2. *Feigenbaum E. A., Feldman J.* Computers and thought. — N.Y.: McGraw-Hill, 1963. — 195 p.
- 2.3. *Амосов Н. М.* Моделирование мышления и психики. — К.: Наук.думка, 1965. — 304 с.
- 2.4. *Schank R. C., Colby K. (eds.)* Computer models of thought and language. — N.Y.: Freeman, 1973. — 243 p.
- 2.5. *Уинстон П.* Искусственный интеллект. — М.: Мир, 1980. — 519 с.
- 2.6. *Barr A., Feigenbaum E. A. (eds.)* The handbook of artificial intelligence / Los Altos CA: Morgan Kaufmann. — 1981, V. 1. — 1982, V. 2.
- 2.7. *O'Shea T., Eisenstadt M.* Artificial intelligence: tools, techniques, and applications. — N.Y.: Harper and Row, 1984. — 423 p.
- 2.8. *Эндрю А.* Искусственный интеллект. — М.: Мир, 1985. — 264 с.
- 2.9. *Хант Д.* Искусственный интеллект. — М.: Мир, 1986. — 412 с.
- 2.10. *Нильсон Н.* Принципы искусственного интеллекта. — М.: Радио и связь, 1985. — 373 с.
- 2.11. *Поспелов Г. С., Поспелов Д. А.* Искусственный интеллект — прикладные системы // Математическая кибернетика. — М.: Знание, 1985. — 48 с.
- 2.12. *Поспелов Д. А.* Ситуационное управление. Теория и практика. — М.: Наука, 1986. — 288 с.
- 2.13. *Поспелов Д. А.* Искусственный интеллект: новый этап развития // Вести АН СССР. — 1987. — № 4. — С. 25–47.
- 2.14. *Поспелов Г. С.* Искусственный интеллект — основа новой информационной технологии. — М.: Наука, 1988. — 280 с.
- 2.15. *Поспелов Э. В., Поспелов Д. А.* Справочник по искусственному интеллекту (в трех томах) — М.: Радио и связь, 1990.
- 2.16. *Rich E., Knight K.* Artificial intelligence. — N.Y.: McGraw-Hill, 1991. — 347 p.
- 2.17. *Ефимов Н. Н., Фролов В. С.* Основы информатики. Введение в ИИ. — М.: МГУ, 1991. — 115 с.
- 2.18. *Аверкин А. Н., Гаазе-Рапопорт М. Г., Поспелов Д. А.* Толковый словарь по искусственному интеллекту. — М.: Радио и связь, 1992. — 256 с.
- 2.19. *Winston P. H.* Artificial Intelligence. — Reading, MA: Addison-Wesley, 1992. — 578 p.
- 2.20. *Гринченко Т. А., Стогний А. А.* Машинный интеллект и новые информационные технологии. — К.: Манускрипт, 1993. — 168 с.
- 2.21. *Ямпольский Л. С., Лавров О. А.* Штучний інтелект у плануванні та управлінні виробництвом. — К.: Вища школа, 1995. — 255 с.
- 2.22. *Russell S. J., Norvig P.* Artificial intelligence: a modern approach. — Englewood Cliffs, NJ: Prentice-Hall, 1995. — 543 p.

2.23. *Nillson N.* Artificial intelligence. A new synthesis. — San Franc. : Morgan Kaufmann, 1998. — 514 p.

2.24. *Капитонова Ю. В., Скурихин В. И.* О некоторых тенденциях развития и проблемах искусственного интеллекта // Кибернетика и системный анализ. — 1999. — № 1. — С. 43–50.

2.25. *Спірін О. М.* Початки штучного інтелекту. — Житомир : ЖДПУ, 2001. — 91 с.

2.26. *Глибовець М. М., Олецкий О. В.* Штучний інтелект. — К. : КМ Академія, 2002. — 366 с.

2.27. *Бондарев В. Н., Аде Ф. Г.* Искусственный интелект. — Севастополь: СевНТУ, 2002. — 615 с.

2.28. *Cawsey A.* Kunstliche intelligenz. — Munchen, Boston: Pearson Studium, 2003. — 528 p.

2.29. *Рассел С., Норвиг П.* Искусственный интелект. — М. : Вильямс, 2006. — 1407 с.

2.30. *Костров Б. В., Ручкин В. Н., Фулин В. А.* Основы ИИ. — М. : ДЕСС, 2007. — 192 с.

2.31. *Уоссермен Ф.* Нейрокомпьютерная техника. Теория и практика. — М. : Мир, 1992. — 240 с.

2.32. *Haykin S.* Neural networks: a comprehensive foundation. — N.Y. : MacMillan College Publishing Co., 1994. — 391 p.

2.33. *Горбань А. Н., Дунин-Барковский В. Л., Кирдин А. Н. и др.* Нейроинформатика. — Новосибирск : Наука, СП РАН, 1998. — 296 с.

2.34. *Морозов А. А. и др.* Нейрокомпьютеры и нейротехнологии: накануне нового старта // УСиМ. — 1997. — № 1–3. — С. 93–100.

2.35. Proc. International Joint Conference on Artificial Intelligence (IJCAI).

2.36. Proc. The National Conference on Artificial Intelligence. — American Association for Artificial Intelligence (AAAI).

2.37. Proc. European Conference on Artificial Intelligence (ECAI).

2.38. <http://www.cs.washington.edu/research/jair/home.html>

2.39. <http://www.aaai.org>

2.40. <http://www.cs.cmu.edu/Groups/AI/html/repository.html>

2.41. <http://www.comlab.ox.ac.uk/archive/comp/ai.html>

Розділ 3. Принципи побудови та функціонування комп'ютерних систем штучного інтелекту

3.1. *Payes J. E., Michie D.* (eds.) Intelligent systems — the unprecedented opportunity. — Chichester, UK: Ellis Horwood, 1984. — 475 p.

3.2. *Поспелов Д. А.* Интеллектуальные системы: ожидания и реальность // Природа. — 1988. — № 3. — С. 49–63.

3.3. *Поспелов Д. А.* (ред.) Прикладные системы искусственного интеллекта. — Кишинев : Штиинца, 1991. — 136 с.

3.4. *Поспелов Д. А.* Три шага на пути к официальному признанию // Новости ИИ. — 1997. — № 1. — С. 99–115.

3.5. *Любарский Ю. Я.* Интеллектуальные информационные системы. — М. : Наука, 1990. — 395 с.

3.6. *Финн В. К.* Интеллектуальные информационные системы. — М. : ВИНТИ, 1991. — 192 с.

3.7. *Лорьер Ж.-Л.* Системы искусственного интеллекта. — М. : Мир, 1991. — 568 с.

3.8. *Поляков А. О.* Технология интеллектуальных систем. — СПб. : Санкт-Петербургский ГТУ, 1995. — 242 с.

3.9. *Свириденко С. С.* Информационные технологии в интеллектуальной деятельности. — М. : МНЭПУ, 1995. — 240 с.

3.10. *Змитрович А. И.* Интеллектуальные информационные системы. — Минск: ТетраСистем, 1997. — 368 с.

3.11. *Ситник В. Ф.* та ін. Основи інформаційних систем. — К. : КНЕУ, 1997. — 249 с.

3.12. *Шевченко А. И.* Интеллектуальная система нового поколения // ИИ. — 1997. — № 1. — С. 15–19.

3.13. *Четверигов Г. Г.* Малий тлумачний словник з інтелектуальних систем. — Х. : ХДУТР, 1997. — 64 с.

3.14. *Гаврилов А. В.* Системы искусственного интеллекта. — Новосибирск : НГТУ, 2000. — 48 с.

3.15. *Анісімов А. В., Глибовець М. М., Кравченко М. М.* та ін. Системы штучного інтелекту. — К. : ВПЦ КУ, 2000. — 100 с.

3.16. *Месюра В. І., Ваховська Л. М.* Основы проектирования систем штучного інтелекту. — Вінниця : ВДТУ, 2000. — 95 с.

3.17. *Райков А. Н.* Интеллектуальные информационные системы. — М. : МГУ, 2000. — 96 с.

3.18. *Тельнов Ю. Ф.* Интеллектуальные информационные системы. — М., 2000. — 418 с.

3.19. *Девятков В. В.* Системы искусственного интеллекта. — М. : МГТУ, 2001. — 352 с.

3.20. *Леденева Т. П.* Интеллектуальные информационные системы. — Воронеж : ВГТУ, 2001. — 136 с.

3.21. *Рідкокаша А. А.* та ін. Основы систем штучного інтелекту. — Черкаси : Відлуння—Плюс, 2001. — 231 с.

3.22. *Кушнир Г. А.* Системы искусственного интеллекта. — М. : Маркетинг, 2001. — 34 с.

3.23. *Зайченко Ю. П.* Основы проектирования интеллектуальных систем. — К. : Слово, 2004. — 352 с.

3.24. Плескач В. Л., Рогушина Ю. В., Кустова Н. П. Інформаційні технології та системи. — К. : КНТЕУ, 2004. — 520 с.

3.25. Кузьменко Б. В. Системы штучного интеллекта. — К. : Альтерпрес, 2006. — 140 с.

Розділ 4. Основні аспекти інженерії знань

4.1. Davis R. Meta-rules: Reasoning about control // Artificial Intelligence, 1980, N. 15. — P. 179–222.

4.2. Davis R., Lenat D. Knowledge-based systems in artificial intelligence. — N.Y. : McGraw-Hill, 1980. — 487 p.

4.3. Newell A. The knowledge level // Artificial Intelligence, 1982, N. 18. — P. 87–127.

4.4. Nii H. P. Blackboard systems // AIMagazine, 1986. — Part 1 : V. 7, N. 2, p. 38–53. — Part 2 : V. 7, N. 3, P. 82–106.

4.5. Michalski R. S., Carbonell J. G., Mitchell T. M. Machine learning. — V. II. — Palo Alto, CA : Tioga, 1986. — 532 p.

4.6. Marcus S. (eds.) Automating knowledge acquisition for expert systems. — Boston : Kluiver Academic, 1988. — 456 p.

4.7. Neale I. M. First generation expert systems: review of knowledge acquisition methodologies // Knowledge Engineering Review, 1988, V. 3, N. 2. — P. 105–145.

4.8. Kondratoff Y., Michalski R. S. (eds.) Machine learning an artificial intelligence approach. — V. III. — Los Altos, CA : Morgan Kaufmann, 1990. — 442 p.

4.9. Осуга С., Сазки Ю. (ред.) Приобретение знаний. — М. : Мир, 1990. — 304 с.

4.10. Гаврилова Т. А., Червинская К. Р. Извлечение и структурирование знаний для экспертных систем. — М. : Радио и связь, 1992. — 385 с.

4.11. Buchanan B. G., Wilkins D. C. (eds.) Readings in knowledge acquisition and learning. — Los Altos, CA : Morgan Kaufmann, 1993. — 511 p.

4.12. Gil Y., Paris C. L. Towards method-independent knowledge acquisition // Knowledge Acquisition, 1994, V. 26, N. 2. — P. 163–178.

4.13. Stefik M. Introduction to knowledge systems. — San Francisco, CA : Morgan Kaufmann, 1995. — 423 p.

4.14. Mitchell T. M. Machine learning. — NY : McGraw-Hill, 1997. — 427 p.

4.15. Осипов Г. С. Приобретение знаний интеллектуальными системами. — М. : Наука, 1997. — 367 с.

4.16. Осипов Г. С. Информационные технологии, основанные на знаниях // Новости ИИ. — 1998. — № 1. — С. 7–41.

4.17. Studer R., Benjamins V. R., Fensel D. Knowledge engineering: principles and methods // Data and Knowledge Engineering, 1998, V. 25, N 1-2. — P. 161–198.

4.18. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск : Наука, 1999. — 472 с.

4.19. Ильин Н., Киселев С., Рябышкин В. и др. Технологии извлечения знаний из текстов // Открытые системы. — 2006. — № 6. — С. 48–53.

4.20. Киселев М. В., Соломатин Е. А. Средства добычи знаний в бизнесе и финансах // Открытые системы. — 1997. — № 4. — С. 41–44.

4.21. Дюк В. Д. Data Mining: состояние проблемы, новые решения // Открытые системы. — 1999. — № 3. — С. 12–24.

4.22. Дюк В., Самойленко А. Data Mining : учебный курс. — СПб. : Питер, 2001. — 368 с.

4.23. Han J., Kamber M. Data Mining: concepts and techniques. — San Francisco: Morgan Kaufman. — 2000. — 550 p.

4.24. Коржов В. Н. Data Mining по-русски // Computer World, 2000. — № 34. — С. 54–62.

4.25. Малащук Е. В., Бабин Д. В., Вороной С. М., Кочеткова М. Г. Обзор существующих алгоритмов Data Mining для глубинного анализа текстов и методов извлечения знаний // ИИ. — 2005. — № 4. — С. 619–626.

4.26. Gralmeier J., Rudolph A. Techniques of kluster algorithms in Data Mining // Data Mining and Knowledge Discovery, 2002, V. 6, N. 4. — P. 303–360.

4.27. Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Методы и модели анализа данных: OLAP и Data Mining. — СПб. : Петербург, 2004. — 336 с.

4.28. Nonaka I., Takeuchi I. The knowledge-creating company. — N.Y. : Oxford University Press, 1995. — 280 с.

4.29. Conklin J. Designing organisational memory. Preserving intellectual assets in a knowledge economy / Electronic Publication by Corporate Memory Systems, Inc., 1996. — 213 с.

4.30. Macintosh A. Knowledge as Set Management // Airing. — 1997. — N. 20. — P. 157–169.

4.31. Десуза К. Сценарии в управлении: от запаздывания к упреждению // Открытые системы. — 2006. — № 4. — С. 36–42.

Розділ 5. Бази знань

5.1. Яшин А. М. Базы знаний и экспертные системы. — Л. : ЛПИ, 1990. — 76 с.

- 5.2. Грищенко В. И., Бакаев А. А., Козлов Д. Н. Методы организации и обработки баз знаний. — К. : Наукова думка, 1993. — 150 с.
- 5.3. Нетесин И. Е., Рогозина Ю. В. Методика повышения актуальности распределенной базы знаний // УСиМ. — 1996. — № 4–5. — С. 108–112.
- 5.4. Аббасов А. М., Мамедова М. Г. Методы организации баз знаний с нечеткой реляционной структурой. — Баку : ЭЛМ, 1997. — 472 с.
- 5.5. Дейт К. Дж. Введение в системы баз данных. — К. : Диалектика, 1998. — 784 с.
- 5.6. Микони С. В. Модели и базы знаний. — СПб. : ЛИИЖТ, 2000. — 154 с.
- 5.7. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. — СПб.: Питер, 2001. — 384 с.
- 5.8. Корнеев В. В., Гарев А. Ф., Васютин С. В., Райх В. В. Базы данных. Интеллектуальная обработка информации. — М. : Нолидж, 2001. — 653 с.
- 5.9. Миронов А. С. Статические банки знаний. Математическое и программное обеспечение. — М. : Машиностроение, 2001. — 144 с.
- 5.10. Спирли Э. Корпоративные хранилища данных. Планирование, разработка, реализация. — М. : Вильямс, 2001. — 521 с.
- 5.11. Архипенков С., Голубев Д., Максименко О. Хранилища данных. От концепции до внедрения // Диалог–МИФИ. — 2002. — № 7. — С. 78–104.
- 5.12. Романюк О. Н., Савчук Т. О. Організація баз даних і знань. — Вінниця : Універсум, 2003. — 218 с.
- 5.13. Кунгурцев А. Б., Тыхан И. В. Формирование словаря предметной области // ИИ. — 2006. — № 1. — С. 167–172.
- 5.14. Звенигородский А. С., Будницкий А. В. Структуризация данных из сообщений на естественном языке // ИИ. — 2007. — № 1. — С. 45–50.
- 5.15. Gruber T. R. A translation approach to portable ontologies // Knowledge Acquisition, 1993, V. 5, N. 2. — P. 199–220.
- 5.16. Guarino N., Poli R. The role of ontology in the information technology // International Journal of Human–Computer Studies, 1995, V. 43, N. 5–6. — P. 623–965.
- 5.17. Ushold M., Gruninger M. Ontologies: principles, methods and applications // Knowledge Engineering Review, 1996, V. 11, N. 2. — P. 247–295.
- 5.18. Guarino N., Masolo C., Vetere G. OntoSeek: content based access to the Web // IEEE Intelligent Systems, Mai-June 1999. — P. 184–225.

- 5.19. Плескач В. Л. Онтологии в контексте представления знаний об электронном бизнесе // Экономика: проблемы теории и практики. — 2004. — Т. 3. — С. 697–702.
- 5.20. Овдій О. М., Проскудіна Г. Ю. Онтології у контексті інтеграції інформації: представлення, методи та інструменти побудови // Проблеми програмування. — 2004. — № 4. — С. 353–366.
- 5.21. Овдий О. М., Проскудина Г. Ю. Обзор инструментов инженерии онтологий. — <http://www.elbib.ru/rus/journal/2004/part4/op.html>
- 5.22. Villemin F.–Y. Ontologies–based relevant information retrieval. — <http://www.cnam.fr/f-yv>
- 5.23. KA2 science ontology. — <http://ontobroker.semaanticweb.org/ontos/ka2.html>
- 5.24. W3C The Semantic Web Home Page. — <http://www.w3.org/2001/sw/>

Розділ 6. Моделі подання знань

- 6.1. Post E. L. Formal reductions of the general combinatorial decision problem // American Journal of Mathematics, 1943, N. 65. — P. 197–268.
- 6.2. Robinson J. A. A machine-oriented logic based on the resolution principle // Journal of the Association for Computing Machinery, 1965, 12. — P. 23–41.
- 6.3. Робинсон Дж. Машино-ориентированная логика, основанная на принципе резолюции // Киб. сборник. — 1970. — Вып. 7. — С. 194–218.
- 6.4. Robinson J. A. Logic: form and function. — Edinburgh: University Press, 1979. — 453 p.
- 6.5. Minsky M. (eds.) Semantic information processing. — Cambridge, MA : MIT Press, 1968. — 427 с.
- 6.6. Minsky M. A Framework for representing knowledge / In The Psychology of Computer Vision. — N.Y. : McGraw–Hill, 1975. — P. 211–277.
- 6.7. Минский М. Фреймы для представления знаний. — М. : Энергия, 1979. — 390 с.
- 6.8. Schank R. C., Colby K. (eds.) Computer models of thought and language. — N.Y. : W. H. Freeman, 1973. — 378 p.
- 6.9. Bobrow D. G., Winograd T. An overview of Knowledge Representation Language (KRL) // Cognitive Science, 1977, V. 1, N. 1. — P. 127–174.
- 6.10. Уэно Х. и др. (ред.) Представление и использование знаний. — М. : Мир, 1989. — 342 с.

- 6.11. Кандрашина Е. Ю., Литвинцева А. В., Поспелов Д. А. Представление знаний о времени и пространстве в интеллектуальных системах. — М.: Наука, 1989. — 325 с.
- 6.12. Уэно Х., Исидзука М. (ред.) Представление и использование знаний. — М.: Мир, 1989. — 220 с.
- 6.13. Кисленко Ю. И. Информационное обеспечение интеллектуальных моделей. — К.: УМК ВО, 1992. — 166 с.
- 6.14. Кокорева Л. В., Перевозчикова О. Л., Ющенко К. Л. Диалоговые системы и представление знаний. — К.: Наукова думка, 1992. — 448 с.
- 6.15. Бондаренко М. Ю., Гребенюк В. А., Кайкова Е. Б. и др. Теория многоуровневых семантических сетей. — Х.: ХТУРЭ, 1997. — 76 с.
- 6.16. Андон Ф. И., Яшунин А. Е., Резниченко В. А. Логические модели интеллектуальных информационных систем. — К.: Наукова думка, 1999. — 398 с.
- 6.17. Некрашевич С. П., Божко Д. В. Представление данных в Интернет на основе семантических сетей // ИИ. — 2006. — № 1. — С. 412–420.
- 6.18. Булкин В. И., Шаронова Н. В. Формальное представление знаний в продукционных системах // ИИ. — 2006. — № 4. — С. 147–157.
- 6.19. Gettis J., Mogul J., Frystyk H., Bernes-Lee T. et al. Hyper Text Transfer Protokol (HTTP). — <http://www.3w.org/protocols/ffc2616.html>.
- 6.20. Web Accessibility Initiative (WAI) WebSite. — <http://www.w3.org/wai>.

Розділ 7. створення нових знань в інтелектуальних системах

- 7.1. Michie D. (eds.) Machine intelligence 3. — Edinburgh: Edinburgh University Press, 1968. — 412 p.
- 7.2. Bobrow D. G., Collins A. (eds.) Representation and understanding. — N.Y.: Academic Press, 1975. — 443 p.
- 7.3. Elcock E. W., Michie D. (eds.) Machine intelligence 8. — N.Y.: Wiley, 1977. — 510 p.
- 7.4. Schank R. C., Abelson R. Scripts, plans, goals and understanding. — Hillsdale, NJ: Lawrence Erlbaum, 1977. — 445 с.
- 7.5. Гладун В. П. Эвристический поиск в сложных средах. — К.: Наукова думка, 1977. — 166 с.
- 7.6. Кузнецов И. П. Механизмы обработки семантической информации. — М.: Наука, 1978. — 342 с.

- 7.7. Hayes J. E., Michie D., Pao Y. H. (eds.) Machine intelligence 10. — Chichester, UK: Ellis Horwood, 1982. — 564 p.
- 7.8. Bundy A., Silver B., Plummer D. An analytical comparison of some rule-learning programs // Artificial Intelligence. — 1985. — N. 27. — P. 137–181.
- 7.9. Brachman R. J., Levesque H. J. Readings in knowledge representation. — Los Altos CA: Morgan Kaufmann, 1985. — 427 p.
- 7.10. Ginsberg M. (eds.) Readings in nonmonotonic reasoning. — Los Altos, CA: Morgan Kaufmann, 1987. — 478 p.
- 7.11. Pearl J. Probabilistic reasoning for intelligent systems. — Los Altos, CA: Morgan Kaufmann, 1988. — 507 p.
- 7.12. Poundstone W. Labyrinths of reason. — N.Y.: Doubleday, 1988. — 520 p.
- 7.13. Поспелов Д. А. Моделирование рассуждений. Опыт анализа мыслительных актов. — М.: Радио и связь, 1989. — 184 с.
- 7.14. Осуеа Г. Обработка знаний. — М.: Мир, 1989. — 324 с.
- 7.15. Neapolitan R. E. Probabilistic reasoning in expert systems: Theory and Algorithms. — N.Y.: Wiley, 1990. — 387 p.
- 7.16. Гладун В. П. Процессы формирования новых знаний. — София, 1994. — 189 с.
- 7.17. Kolodner J. L. Case-based reasoning. — Los Altos, CA: Morgan Kaufmann, 1993. — 495 p.
- 7.18. Allen B. Case-based reasoning: business applications // Communications of the ACM. — 1994. — V. 37. — N. 3. — P. 40–42.
- 7.19. Allen J. F. Natural language understanding. — Menlo Park, CA: Benjamin-Cummings, 1995. — 378 p.
- 7.20. Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. — Los Altos, CA: Morgan Kaufmann, 1997. — 376 p.
- 7.21. Гладун В. П., Величко В. Ю., Киселева Н. Н. и др. Вывод гипотез о составе и свойствах объектов на основе аналогии // ИИ. — 2000. — № 1. — С. 44–52.
- 7.22. Люгер Дж. Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем. — М.: Вильямс, 2005. — 863 с.
- 7.23. Nilsson N. J. Problem solving methods in artificial intelligence. — N.Y.: McGraw-Hill, 1971. — 265 с.
- 7.24. Нильсон Н. Д. Искусственный интеллект. Методы поиска решений. — М.: Мир, 1973. — 478 с.
- 7.25. Nilsson N. J. Principles of artificial intelligence. — Palo Alto, CA: Tioga, 1985. — 376 p.
- 7.26. Allen B. Case-based reasoning: business applications // Communications of the ACM. — 1994. — V. 37. — N. 3. — P. 40–62.

7.27. *Golding A. R., Rosenbloom P. S.* Improving accuracy by combining rule-based and case-based reasoning // Artificial Intelligence. — 1996. — N. 87. — P. 215–254.

7.28. *Ефимов Е. И.* Решатели интеллектуальных задач. — М. : Наука, 1982. — 320 с.

Розділ 8. Методи створення нових знань інтелектуальних задач

8.1. *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов. — М. : Наука, 1974. — 415 с.

8.2. *Гренандер У.* Лекции по теории распознавания образов (в трех томах). — М. : Мир, 1979–1983.

8.3. *Васильев В. И.* Распознающие системы. — К. : Наук. думка, 1983. — 287 с.

8.4. *Верхаген К., Дейн Р. и др.* Распознавание образов: состояние и перспективы. — М. : Радио и связь, 1985. — 104 с.

8.5. *Васильев В. И., Шевченко А. И.* Искусственный интеллект: Проблема обучения опознаванию образов. — Донецк, 1997. — 223 с.

8.6. *Grabmeier J., Rudolph A.* Techniques of cluster algorithms in data mining // Data Mining and Knowledge Discovery. — October 2002. — Vol. 6. — № 4. — P. 303–360.

8.7. *Runkler T. A., Bezdek J. C.* Web mining with relation clustering // International Journal of Approximate Reasoning. — 2003. — V. 32. — N. 2–3. — P. 217–236.

8.8. *Hammouda K. M., Kamel M. S.* Efficient phrase-based document indexing for web document clustering // IEEE Transactions on Knowledge and Data Engineering. — 2004. — V. 16. — N. 10. — P. 1279–1296.

8.9. *Xu R., Wunsch D.* Survey of clustering algorithms // IEEE Transactions on Neural Networks. — May 2005. — Vol. 16, № 3. — P. 645–678.

8.10. *Jin H., Wong M. L., Leung K. S.* Scalable model-based clustering for large databases based on data summarisation // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2005. — V. 27. — N. 11. — P. 1710–1719.

8.11. *Алыгулиев Р. М.* Метод кластеризации коллекции документов и алгоритм для оценки оптимального числа классов // ИИ. — 2006. — № 4. — С. 651–659.

8.12. *Desai M., Spink A.* An algorithm to cluster documents based on relevance // Information Processing and Management, 2005, V.41, N.5. — P. 1035–1049.

8.13. *Li X., Ye N.* A Supervised clustering and classification algorithms for mining data with mixed variables // IEEE Transactions of Systems, Man and Cybernetics, 2006, V.36, N.2. — P. 396–406.

8.14. *Li T.* A unified view on clustering binary data // Machine Learning, 2006, V. 62, N. 3 — P. 199–215.

8.15. *Halkidi M., Baristakis Y., Vazigiannis M.* Cluster validity methods // ACM SIGMOD Record, 2002, V. 31 (Path I : N. 2 — P. 40–45) (Path II : N. 3 — P. 19–27).

8.16. *Goldberg D. E.* Genetic algorithms in search, optimization and machine learning. — Addison Wesley, 1998. — 287 p.

8.17. *Michalewicz Z.* Genetic algorithms + data structures = evolution programs. — Berlin: Springer Verlag, Addison Wesley, 1996. — 387 p.

8.18. *Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А.* Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. — Харьков: Основа, 1997. — 112 с.

8.19. *Курейчик В. М.* Методы генетического поиска. — Таганрог : ТГРУ, 1998. — 285 с.

8.20. *Роштейн А. П.* Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети. — Винница : Универсум, 1999. — 432 с.

8.21. *Kureichik V. M.* Genetic algorithms: state of the art, problems and perspectives // Journal of Computer and Systems Sciences International, 1999, V. 38, N. 1. — P. 137–152.

8.22. *Курейчик В. М.* Генетические алгоритмы и их применение. — Таганрог : ТРТУ, 2002. — 242 с.

8.23. *Кисляков А. В.* Генетические алгоритмы: математический анализ некоторых схем репродукции // ИТ. — 2000. — № 12. — С. 9–14.

8.24. *Ermolaev V., Plaksin S.* Cooperation layers in agent-enabled business process management // Проблемы программирования. — 2002. — № 1-2. — С. 354–367.

8.25. *Тененев В. А., Паклин Н. Б.* Гибридный генетический алгоритм с дополнительным обучением лидера // ИС в производстве. — 2003. — № 2. — С. 181–206.

8.26. *Чипига А. Ф., Петров Ю. Ю.* Модифицированная математическая модель простого генетического алгоритма // ИИ. — 2005. — № 4. — С. 314–317.

8.27. *Алыгулиев Р. М., Алыгулиев Р. М.* Быстрый генетический алгоритм решения задачи кластеризации текстовых документов // ИИ. — 2005. — № 3. — С. 698–707.

8.2. *Алыгулиев Р. М., Алыгулиев Р. М.* Генетический подход к оптимальному назначению заданий в распределенной системе // ИИ. — 2005. — № 4. — С. 79–88.

8.29. *Бабин Д. Б.* Генетический алгоритм решения задачи анализа рыночной корзины // ИИ. — 2006. — № 4. — С. 129–132.

8.30. Воловник А. Д., Тененев В. А. Применение генетического алгоритма с вещественным кодированием для настройки модели управления интеллектуальным капиталом банка // ИИ. — 2006. — № 4. — С. 383–391.

8.31. Волченко Е. В. Генетический алгоритм бисекции графов // ИИ. — 2007. — № 1. — С. 233–237.

8.32. Васильев Ф. П. Численные методы решения экстремальных задач. — М. : Наука, 1988. — 552 с.

8.33. Амтетков А. В., Галкин С. В., Зарубин В. С. Методы оптимизации. — М. : МГТУ, 2003. — 440 с.

8.34. Ивахненко А. Г. Системы эвристической самоорганизации в технической кибернетике. — К. : Техніка, 1971. — 372 с.

8.35. Ивахненко А. Г., Тодуа Н. Н., Фомичев А. А. Структурный синтез по обобщенному алгоритму МГУА фильтров для косвенного измерения переменных // Автоматика. — 1973. — № 1. — С. 49–61.

8.36. Ивахненко А. Г. Индуктивный метод самоорганизации моделей сложных систем. — К. : Наукова думка, 1982. — 296 с.

Розділ 9. Експертні системи

9.1. Shortliffe E. H. Computer-based medical consultations: Mycin. — N.Y. : Elsevier, 1976. — 358 p.

9.2. Buchanan B. G., Feigenbaum E. A. Dendral and Meta-Dendral: their applications dimension // Artificial Intelligence, 1978, N. 11. — P. 5–24.

9.3. Michie D. (eds.) Expert systems in the micro-electronic age. — Edinburgh : Edinburgh University Press, 1979. — 397 p.

9.4. Sleeman D., Brown J. S. (eds.) Intelligent tutoring systems. — London: Academic Press, 1982. — 420 p.

9.5. Buchanan B. G., Shortliffe E. H. (eds.) Rule-based expert systems. — Reading MA : Addison-Wesley, 1984. — 365 p.

9.6. Reitman W. (eds.) Artificial intelligence applications for business. — Norwood, NJ : Ablex, 1984. — 442 p.

9.7. Klahr P., Waterman D. (eds.) Expert systems: techniques, tools, and applications. — Reading, MA : Addison-Wesley, 1986. — 524 p.

9.8. Попов Э. В. Экспертные системы. Решение неформализованных задач в диалоге с ЭВМ. — М. : Наука, 1987. — 288 с.

9.9. Хейес-Рот Ф., Уотермен Д., Ленат Д. Построение экспертных систем. — М. : Мир, 1987. — 441 с.

9.10. Элти Д., Кумбс М. Экспертные системы. Концепции и примеры. — М. : Финансы и статистика, 1987. — 191 с.

9.11. Фортсайт Ф. (ред.) Экспертные системы. Принципы работы и примеры. — М. : Радио и связь, 1987. — 220 с.

9.12. Poeson M. C., Richardson J. Foundations of intelligent tutoring systems. — Hillsdale, NJ : Lawrence Erlbaum, 1988. — 425 p.

9.13. Поспелов Д. А. (ред.) Экспертные системы: состояние и перспективы: сб. тр. — М. : Наука, 1989. — 152 с.

9.14. Крисевич В. С. и др. Экспертные системы для персональных компьютеров: справочное пособие. — Минск : ВШ, 1990. — 200 с.

9.15. Harmon P., Sawyer B. Creating expert systems for business and industry. — N.Y. : Wiley, 1990. — 330 p.

9.16. Ashley K. D. Modeling legal argument: reasoning with cases and hypothetical. — Cambridge MA : MIT Press., 1990. — 290 p.

9.17. Бакаев А. А., Грищенко В. И., Козлов Д. Н. Экспертные системы и логическое программирование. — К. : Наукова думка, 1992. — 219 с.

9.18. Убейко В. М. и др. Экспертные системы. — М. : МАИ, 1992. — 83 с.

9.19. Петрушин В. А. Экспертно-обучающие системы. — К. : Наукова думка, 1992. — 196 с.

9.20. Feigenbaum E. A., Buchanan B. G. Dendral and Meta-Dendral: roots of knowledge systems and expert systems applications // Artificial Intelligence, 1993, N. 59. — P. 233–240.

9.21. McDermott J. RI (XCON) at age 12: lessons from an elementary school achiever // Artificial Intelligence, 1993, N. 59. — P. 241–247.

9.22. Башлыков А. А. и др. Экспертные системы поддержки принятия решений в энергетике. — М. : МЭИ, 1994. — 216 с.

9.23. Giarratano J., Riley G. Expert systems: principles and programming. — Boston, MA : PWS Publishing, 1994. — 327 p.

9.24. Герман О. В. Введение в теорию экспертных систем и обработку знаний. — Минск: ДизайнПРО, 1995. — 255 с.

9.25. Попов Э. В., Фоминых И. Б., Кисель Е. Б., Шанот М. Д. Статистические и динамические экспертные системы. — М. : Финансы и статистика, 1996. — 320 с.

9.26. Барвинский В. Л. Экспертные системы и банки данных и знаний. — К. : КМУГА, 1997. — 40 с.

9.27. Макурин Н. С. Технологические экспертные системы в производстве. — Х. : ХГТУР, 1998. — 160 с.

9.28. Продеус А. Н., Захрамова Е. Н. Экспертные системы в медицине. — К. : ВЕК+, 1998. — 320 с.

9.29. Головки А. П. Экспертные системы. Основные концепции. — Курган : КГУ, 2000. — 90 с.

9.30. Джексон П. Введение в экспертные системы. — М., Спб., К. : Вильямс, 2001. — 624 с.

9.31. *Гаврилов А. В.* Гибридные интеллектуальные системы. — Новосибирск : НГТУ, 2003.

9.32. *Кравець Б. О., Хавіна І. П., Комбін Ю. М. та ін.* Вступ до експертних систем. — Х. : НТУ ХПІ, 2006. — 232 с.

9.33. *Takata N. et al.* Genco: expert system building support tools // *Shimadzu Technical Review*, 1991, V. 48, N. 3. — P. 324–387.

9.34. *Mooge B. et al.* Questions and answers about G2 // *Gensym Corporation*, 1993. — P. 26–28.

9.35. *Kingston J. K. C., Doheny J. G., Filby I. M.* Evaluation of workbenches which support the CommonKADS methodology // *Knowledge Engineering Review*, 1995, V. 10, N. 3. — P. 285–344.

9.36. *Блинов А. Н., Воронов М. В.* Разработка экспертных систем в инструментальной среде. — СПб. : СПГУТД, 1998. — 322 с.

9.37. *Гаврилов А. В., Новицкая Ю. В.* Инструментальное программное обеспечение для создания гибридных экспертных систем. — Тр. Межд. Конф. ИСТ–2000. — Новосибирск. — НГТУ. — 2000. — Т. 3. — С. 488–490.

9.38. <http://vt.cs.nstu.ru/~expsystem/>

Розділ 10. Нейрокомп'ютерні системи

10.1. *McCulloch W. S., Pitts W.* A logical calculus of the imminent in nervous activity // *Bull. Math. Biophysics*, 1943, N. 5. — P. 115–137.

10.2. *Rosenblatt F.* Principles of neurodynamics. — N.Y. : Spartan Books, 1962. — 312 p.

10.3. *Розенблатт Ф.* Принципы нейродинамики. — М. : Мир, 1965. — 480 с.

10.4. *Minsky M., Papert S.* Perceptrons: an introduction to computational geometry. — Cambridge, Mass : MIT Press, 1969. — 214 p.

10.5. *Кохонен Т.* Ассоциативная память. — М. : Мир, 1980. — 324 с.

10.6. *Hopfield J. J.* Neural networks and physical systems with emergent collective computational abilities / *Proc. National Academy of Sciences USA*, 1982, N. 79. — P. 2554–2558.

10.7. *Lippmann R. P.* An introduction to computing with neural nets // *IEEE ASSP Magazine*, 1987, V. 4, N. 2. — P. 4–22.

10.8. *Anderson J. A., Rosenfeld E.* Neurocomputing: foundation of research. — Cambridge, Mass: MIT Press, 1988. — 356 p.

10.9. *Feldman J., Fanty M. A., Goddard N. H.* Computing with structured neural networks // *Computer*. — 1988. — V. 21. — N. 3. — P. 91–103.

10.10. *Kohonen T.* Self organization and associative memory. — N.Y. : Springer-Verlag, 1989. — 430 p.

10.11. *Brunak S., Lautrup B.* Neural networks, computers with intuition. — Singapore : World Scientific, 1990. — 411 p.

10.12. *Minsky M.* Logical versus analogical or symbolic versus connectionist or neat versus scruffy // *AI Magazine*, 1991, V. 65, N. 2. — P. 34–51.

10.13. *Горбань А. Н.* Обучение нейронных сетей. — М. : СП Параграф, 1990. — 278 с.

10.14. *Hertz J., Krogh A., Palmer R. G.* Introduction to the theory of neural computation, Addison-Wesley, Reading, Mass, 1991. — 456 p.

10.15. *Carpenter G. A., Grossberg S.* Pattern recognition by selforganizing neural networks. Cambridge, Mass : MIT Press, 1991. — 327 p.

10.16. *Куссуль Э. М., Амосов Н. М. и др.* Нейрокомпьютеры и интеллектуальные роботы. — К. : Наукова думка, 1991. — 310 с.

10.17. *Куссуль Э. М.* Ассоциативные нейроподобные структуры. — К. : Наукова думка, 1992. — 144 с.

10.18. *Горбань А. Н., Россиев Д. А.* Нейронные сети на персональном компьютере. — Новосибирск : Наука, 1996. — 285 с.

10.19. *Бэстэнс Д.-Э., Ван Ден Берг В. М., Вуд Д.* Нейронные сети и финансовые рынки. — М. : Изд-во ТВП, 1997. — 436 с.

10.20. *Шумский С. А.* Нейрокомпьютинг и его применение в экономике и бизнесе. — М. : МИФИ, 1998. — 383 с.

10.21. *Заенцев И. В.* Нейронные сети. Основные модели. — Воронеж, 1999. — 287 с.

10.22. *Галушкин А. И.* Теория нейронных сетей. — М. : ИПРЖР, 2000. — 378 с.

10.23. *Галушкин А.* Нейрокомпьютеры. — М. : ИПРЖР, 2000. — 298 с.

10.24. *Галушкин А. И.* Нейрокомпьютеры и их применение. — М. : Радиотехника, 2000. — 325 с.

10.25. *Кохонен Т., Дебок Г.* Анализ финансовых данных с помощью самоорганизующихся карт. — М. : Изд. дом Альпина, 2001. — 340 с.

10.26. *Байдык Т. Н.* Нейронные сети и задачи ИИ. — К. : Наук. думка, 2001. — 264 с.

10.27. *Головкин В. А.* Нейронные сети: обучение, организация и применение. — М. : Радиотехника, 2001. — 256 с.

10.28. *Круглов В. В., Борисов В. В.* Искусственные нейронные сети. Теория и практика. — М. : Горячая линия — Телеком, 2001. — 297 с.

10.29. *Круглов В. В., Борисов В. В.* Гибридные нейронные сети. — Смоленск : Русич, 2001. — 224 с.

10.30. *Круглов В. В., Дли М. И., Голунов Р. Ю.* Нечеткая логика и искусственные нейронные сети. — М. : Физматлит, 2001. — 224 с.

10.31. *Питолин А. В.* Основы проектирования искусственных нейронных сетей. — Воронеж : ВГТУ, 2001. — 108 с.

- 10.32. *Короткий С.* Нейронные сети: алгоритм обратного распространения. — СПб. : Питер, 2000. — 435 с.
- 10.33. *Короткий С.* Нейронные сети: обучение без учителя. — СПб. : Питер, 2000. — 545 с.
- 10.34. *Осовский С.* Нейронные сети для обработки информации. — М. : Финансы и статистика, 2002. — 344 с.
- 10.35. *Комарцова Л. Г., Максимов А. В.* Нейрокомпьютеры. — М. : МГТУ, 2002. — 254 с.
- 10.36. *Комашинский В. И., Смирнов В. И.* Нейронные сети и их применение в системах управления и связи. — М. : Горячая линия — Телеком, 2002. — 372 с.
- 10.37. *Медведев В. С., Потемкин В. Г.* Нейронные сети MATLAB 6. — М. : Диалог-МИФИ, 2002. — 195 с.
- 10.38. *Каллан Р.* Основные концепции нейронных сетей. — М. : Вильямс, 2003. — 287 с.
- 10.39. *Назаров А. В., Лоскутов А. И.* Нейросетевые алгоритмы прогнозирования и оптимизации систем. — СПб. : Наука и техника, 2003. — 345 с.
- 10.40. *Рутковская Д., Пилиньский М., Рутковский Л. и др.* Нейронные сети, генетические алгоритмы и нечеткие системы. — М. : Горячая линия — Телеком, 2004. — 383 с.
- 10.41. *Klan M. S., Khor S. W.* Web document clustering using a hybrid neural network // *Applied Soft Computing*, 2004, V. 4, N. 4. — P. 423–432.
- 10.42. *Новотарський М. А., Нестеренко Б. Б.* Штучні нейронні мережі: обчислення. — К. : Ін-т математики, 2004. — 408 с.
- 10.43. *Кондратенко Н. Р., Куземко С. М.* Основы нейронных сетей. Теория та практика. — Вінниця : ВНТУ, 2006. — 104 с.
- 10.44. *Кизим Н. А., Ястремская Е. Н.* Нейронные сети: теория и практика применения. — Х. : ИНЖЭК, 2006. — 234 с.
- 10.45. *Хайкин С.* Нейронные сети. — М. : Вильямс, 2006. — 1103 с.

Розділ 11. Робототехнічні інтелектуальні системи

- 11.1. *Тимофеев А. В.* Роботы и искусственный интеллект. — М. : Наука, 1978. — 192 с.
- 11.2. *Уинстон П.* (ред.) Психология машинного зрения. — М. : Мир, 1978. — 381 с.
- 11.3. *Янг Дж. Ф.* Робототехника. — Л. : Машиностроение, 1979. — 411 с.
- 11.4. *Гитис Э. И.* Аналогово-цифровые преобразователи. — М. : Наука, 1981. — 71 с.

- 11.5. Робототехника и гибкие автоматизированные производства : учеб. пособие в 9 кн. — М. : Высшая школа, 1986.
- 11.6. *Попов Э. В.* Общение с ЭВМ на естественном языке. — М. : Наука, 1986. — 282 с.
- 11.7. *Пью А.* (ред.) Техническое зрение роботов. — М. : Машиностроение, 1987. — 345 с.
- 11.8. *Винцук Т. К.* Анализ, распознавание и интерпретация речевых сигналов. — К. : Наукова думка, 1987. — 261 с.
- 11.9. *Тимофеев А. В.* Адаптивные робототехнические комплексы. — Л. : Машиностроение, 1988. — 234 с.
- 11.10. *Накано Э.* Введение в робототехнику. — М. : Мир, 1988. — 367 с.
- 11.11. *Фу К., Гонсалес Р., Ли К.* Робототехника. — М. : Мир, 1989. — 437 с.
- 11.12. *Потапова Р. К.* Речевое управление роботом. — М. : Радио и связь, 1989. — 293 с.
- 11.13. *Хорн Б. К. П.* Зрение роботов. — М. : Мир, 1989. — 370 с.
- 11.14. *Козлов Ю. М.* Адаптация и обучение в робототехнике. — М. : Наука, 1990. — 311 с.
- 11.15. *Тимофеев А. В.* Мультиагентное и интеллектуальное управление сложными робототехническими системами // *Теор. основы и прикл. задачи интел-х инф-х технологий*. — СПб. : СПИИРАН, 1999. — С. 71–81.
- 11.16. *Каляев И. А., Гайдук А. Р., Капустян С. Г.* Распределенные системы планирования действий коллективов микророботов. — М. : Янус-К, 2002. — 270 с.
- 11.17. *Градецкий В., Князьков М., Кулешов В., Чистяков В.* Микро-механика и микросистемы // *ИИ*. — 2004. — №4. — С. 701–708.
- 11.18. *Градецкий В. Г.* Международная программа по перспективной робототехнике // *Инф.-измерительные и управляющие системы*, 2006. — Т. 4. — № 1–3. — С. 16–21.
- 11.19. *Мащенко С. В.* Система управления маневренного робота с автоматической градуировкой // *ИИ*. — 2006. — № 1. — С. 540–547.
- 11.20. *Даринцев О. В.* Система управления коллективом микророботов // *ИИ*. — 2006. — № 4. — С. 391–399.
- 11.21. *Морговский Ю. Я.* Взаимодействие интеллектуальных роботов в неоднородных многоагентных системах // *Инф.-измерительные и управляющие системы*, 2006. — Т. 4. — № 1–3. — С. 52–58.
- 11.22. *Золкин С. Г.* Алгоритмы планирования и управления движением робота при статической и динамической оценке внешнего мира // *ИИ*. — 2007. — № 1. — С. 267–274.

Розділ 12. Інтелектуальні системи прийняття рішень

- 12.1. *Гладун В. П.* Планирование решений. — К. : Наукова думка, 1987. — 168 с.
- 12.2. *Борисов А. Н., Алексеев А. В. и др.* Обработка нечеткой информации в системах принятия решений. — М. : Радио и связь, 1989. — 304 с.
- 12.3. *Мушик Э., Мюллер П.* Методы принятия технических решений. — М. : Мир, 1990. — 208 с.
- 12.4. *Бондаренко М. Ф. та ін.* Оптимізаційні задачі в системах прийняття рішень. — Х. : ХДТУР, 1998. — 215 с.
- 12.5. *Чертовской В. Д.* Интеллектуальные системы поддержки решений. — Минск, 1995. — 93 с.
- 12.6. *Ларичев О. И.* Теория и методы принятия решений. — М. : Логос, 2000. — 487 с.
- 12.7. *Комарцова Л. Г.* Применение новых информационных технологий к созданию системы поддержки принятия решений при проектировании распределенных систем // ИТПП. — 2001. — № 4. — С. 16–20.
- 12.8. *Ситник В. Ф.* Системи підтримки прийняття рішень. — К. : КНЕУ, 2004. — 614 с.
- 12.9. *Сетлак Г.* Интеллектуальная система поддержки решений в оперативном управлении производством // ИИ. — 2005. — № 4. — С. 527–534.
- 12.10. *Пушкар О. І., Гіковатий В. М., Євсєєв О. С.* Системи підтримки прийняття рішень. — Х. : ВД Інжек, 2006. — 304 с.

Розділ 13. Мультиагентні інтелектуальні системи

- 13.1. *Maes P.* The Agent network architecture // SIGART Bulletin, 1991, V 2, N. 4. — P. 112–120.
- 13.2. *Genesereth M. R., Ketchpel S. P.* Software agents // Communications of the ACM, 1994, V. 37, N. 7. — P. 48–53.
- 13.3. *Gilbert D. J.* IBM Intelligent agent strategy / IBM Corporation, 1995. — 277 p.
- 13.4. *Wayner P.* Free agents // BYTE, 1995. — P. 105–114.
- 13.5. *Wooldridge M., Jennings N.* Intelligent agents: theory and practice // Knowledge Engineering Review, 1995, V. 10, N. 2. — P. 115–152.
- 13.6. *Cheong F.-C.* Internet agents: spiders, wanderers, brokers and bots. — Indianapolis: New Riders Publishing, 1996. — ISBN : 1-56205-463-5.
- 13.7. *Muller J. P.* The design of intelligent agents: a layered approach. — Springer, 1996. — 219 p.

- 13.8. *Nwana H.* Software agents. An overview // Knowledge Engineering Review, 1996, V. 11, N. 3. — P. 205–244.
- 13.9. *Ndumu D. T., Nwana H. S.* Research and development challenges for agent-based systems // Software Engineering, IEE Processing, 1997, V. 144. — P. 2–10.
- 13.10. *Petrie C. J.* Agent-based engineering, the web and intelligent // IEEE Expert, 1996, V. 11, N. 6. — P. 24–29.
- 13.11. *Поспелов Д. А.* Многоагентные системы — настоящее и будущее // Информационные технологии и вычислительные системы. — 1998. — № 1. — С. 23–31.
- 13.12. *Городецкий В. И., Грушинский М. С., Хабалов А. В.* Многоагентные системы (обзор) // Новости ИИ. — 1998. — № 2. — С. 64–116.
- 13.13. *Валькман Ю. Р., Золотаревский И. А., Квачев В. Г., Яковенко Л. П.* Распределенный искусственный интеллект и многоагентные системы проектирования сложных изделий // Проблемы программирования. — 1999. — № 2. — С. 75–85.
- 13.14. *Felber J.* Multi-Agent Systems. — Harlow England : Addison-Wesley, 1999. — 374 p.
- 13.15. *Dignum F.* Agents, markets, institutions and protocols. — Berlin : Springer Verlag, 2000. — 351 p.
- 13.16. *Атанасова Т. Б.* Агентная технология: концепции, модели, приложения. — Варна, 2000. — 287 с.
- 13.17. *Oliveira E., Rocha A.* Agents advanced features for negotitation in electronic commerce and virtual organization formation process. — Berlin : Springer Verlag, 2000. — 412 p.
- 13.18. *Козлов Д. Д., Смелянский Р. Л.* Использование интеллектуальных агентов для поиска информации в Интернет // ИИ. — 2002. — № 2. — С. 378–382.
- 13.19. *Плескач В. Л.* Мультиагентні системи в електронній комерції // Вісник КНТЕУ. — 2003. — № 6. — С. 35–41.
- 13.20. *Аграновский А. В. и др.* Мультиагентный подход к решению задач стегадно-графического анализа (сокрытия данных) // ИИ. — 2003. — № 3. — С. 428–431.
- 13.21. *Булаев В. В.* Мультиагентные системы защиты распределенных баз данных // ИИ. — 2003. — № 3. — С. 432–435.
- 13.22. *Jain L. C.* Intelligent Agents. — Berlin, Heidelberg : Springer Verlag, 2004.
- 13.23. *Gladun A., Rogushina J.* Multiagent ontology-based intelligent system of e-commerce // Вісник КНУ, Фіз.-мат. науки, 2004. — С. 118–124.
- 13.24. *Гриценко В. И., Гладун А. Я., Журавлев Е. Д., Несен М. В.* Модель мультиагентной системы для e-бизнеса и технология ее

программной реализации // Проблемы программирования. — 2004. — № 2–3. — С. 510–519.

13.25. *Плескач В. Л., Рогушина Ю. В.* Агентні технології. — К. : КНТЕУ, 2005. — 338 с.

13.26. *Коваленко С. Г.* Агентные технологии представления и обработки знаний в локальных и распределенных экспертных системах // ИИ. — 2005. — № 4. — С. 479–488.

13.27. *Жилин А. В., Ковалев С. Г.* Особенности поиска информации в открытых компьютерных информационных системах // ИИ. — 2007. — № 1. — С. 148–150.

13.28. *Wooldridge M.* Introduction in multiagent systems. — <http://www.csc.liv.ac.uk>.

13.29. *Sykara K.* Multiagent systems. — http://www.aaai.org/pathfinder/html/multiagent_systems.htm.

13.30. *Nwana H. S.* Software agent technologies. — <http://www.labs.bt.com/proj/agents/publish/papers>.

13.31. *Leppinen M., Rautiainen A.* Agents in commerce. — http://smartpush.cs.hut.fi/software_agents_in_commerce.

13.32. Agent Construction Tools. — http://www.agentbuilder.com/agent_tools/index.html.

13.33. Agent Software. — <http://www.agentlink.org>.

13.34. Agent Software Exampels. — <http://www.agents/tools/index.html>

13.35. Autonomy Technology Whitepaper. — <http://www.autonomy.com/tech/wp.html>.

Розділ 14. Нечіткі інтелектуальні системи

14.1. *Zadeh L.* Fuzzy logic and approximate reasoning // *Synthese*, 1975, N. 30. — P. 407–428.

14.2. *Zadeh L.* Fuzzy Sets // *Information and control*, 1965, V. 8, N. 3. — P. 338–353.

14.3. *Заде Л.* Понятие о лингвистической переменной и его применение к принятию решений. — М. : Мир, 1976. — 274 с.

14.4. *Zadeh L.* Fuzzy sets as a basis for a theory of possibility // *Fuzzy Sets and Systems*, 1978, N. 1. — P. 3–28.

14.5. *Mamdani E. H., Gaines B. R.* Fuzzy Reasoning and its Applications. — London : Academic Press, 1981. — 387 p.

14.6. *Аверкин А. Н., Батыршин И. З., Блишун А. Ф. и др.* Нечеткие множества в моделях управления и искусственного интеллекта. — М. : Наука, 1986. — 312 с.

14.7. *Мелихов А. Н., Бернштейн Л. С., Коровин С. Я.* Ситуационные советующие системы с нечеткой логикой. — М. : Наука, 1990. — 277 с.

14.8. *Малышев Н. Г., Берштейн Л. С., Боженик А. В.* Нечеткие модели для экспертных систем в САПР. — М. : Энергоатомиздат, 1991. — 290 с.

14.9. *Kosko B.* Neural networks and fuzzy systems. — Englewood Cliffs, NJ : Prentice-Hall, 1991. — 378 p.

14.10. *Kaufmann A., Gupta M.* Introduction to fuzzy arithmetic. — Thomson Computer Press, 1991.

14.11. *Тэрано Т., Ватада Д., Иваи С.* (ред.) Прикладные нечеткие системы. — М. : Мир, 1993. — 368 с.

14.12. *McNeill D., Freiburger P.* Fuzzy logic. — Touchstone Rockefeller Center, 1993. — 254 p.

14.13. *Kahlert J., Frank H.* Fuzzy logik und fuzzy control. — Braunschweig : Vieweg, 1994. — 314 p.

14.14. *Dubois D., Prade H., Yager R. R.* (eds.). Fuzzy information engineering: a guided tour of applications. — N.Y. : Wiley, 1996. — 472 p.

14.15. *Jamshidi M., Tilti A., Zadeh L., Boverie S.* (eds.). Applications of fuzzy logic: towards high machine intelligence quotient systems. — N.Y.: Prentice Hall, 1997. — 456 p.

14.16. *Герасимов Б. М., Грабовский Г. Т., Рюмишн Н. А.* Нечеткие множества в задачах проектирования, управления и обработки информации. — К. : Техніка, 2002. — 76 с.

14.17. *Ярушкина Н. Г.* Основы теории нечетких и гибридных систем. — М. : Финансы и статистика, 2004. — 289 с.

Навчальне видання

ДОЛЯ Віктор Гнатович

КОМП'ЮТЕРНІ СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ

ИБ ПНУС



768881

Літературний редактор
Технічний редактор
Художник-дизайнер
Комп'ютерна верстка
Коректор-редактор

С. І. Дудіна
І. Є. Гнатюк
І. О. Клименко
О. М. Стадник
І. Е. Стрельбицька

Університет «Україна» користується майновими правами на посібник
Долі В. Г. «Комп'ютерні системи штучного інтелекту» як такий

*Оригінал-макет виготовлено
у видавничо-друкарському комплексі Університету «Україна»
03115, м. Київ, вул. Львівська, 23, тел. (044) 424-40-69
Свідоцтво про державну реєстрацію ДК № 405 від 06.04.01*

*Віддруковано з оригінал-макета у видавничо-друкарському
комплексі Університету «Україна»*

Підписано до друку 02.03.2011. Формат 60×84/16.
Папір офсетний. Умовн. друк. арк. 17,21.
Обл. вид. арк. 20,65. Наклад 300 прим.