

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	9
ВСТУП	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ	11
1.1 Поняття оцінки вартості майна	11
1.1.1 Вартість об'єктів оцінки. Види вартості	11
1.1.2 Фактори, які впливають на ціну об'єктів	13
1.2 Поняття експертної системи	14
1.2.1 Подання знань в експертних системах і моделі	16
1.2.2 Організація знань в базі даних	20
1.2.4 Методи пошуку рішень в експертних системах	29
1.2.4.1 Пошук рішень в одному просторі	32
1.2.4.2 Пошук в ієрархії просторів	40
1.2.4.3 Пошук в альтернативних просторах	47
1.2.4.4 Пошук з використанням декількох моделей	52
1.2.4.5 Вибір методу рішення задач	53
1.3 Постановка задачі роботи	55
2 МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ	56
2.1 Побудова моделей представлення знань	56
3 АНАЛІЗ ПО ДЛЯ РЕАЛІЗАЦІЇ БАЗИ ЗНАНЬ ЕКСПЕРТНОЇ СИСТЕМИ	63
3.1 Програмне середовище Visual Prolog	63
3.2 Програмне середовище CLIPS	64
4 ПРОГРАМНА РЕАЛІЗАЦІЯ БАЗИ ЗНАНЬ	67
ВИСНОВКИ	69
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	70

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення

ЕС – експертна система

БЗ – база знань

БД – база даних

ІС – інформаційна система

КС – комп'ютерні системи

СУБД – система управління базою даних

ВСТУП

Сьогодні ринок житла – одна зі сфер нашого повсякденного життя, що найбільш швидко розвивається. Купівля-продаж, оренда, іпотека, застава, страхування не відбудуться без оцінки майна.

В Україні з кожним роком продається все більше нерухомості. При цьому ціни на квадратні метри дуже різняться не тільки в залежності від регіону чи об'єкта, а й від часу укладання угоди, так як ціна часто прив'язана до курсу долара. Тому все більшої актуальності набуває оцінка вартості нерухомості.

Так, оцінка нерухомості і для банку, і для клієнта формує чітку оціночну вартість об'єкта, яка є вихідною. Оціночна вартість використовується для визначення ціни наступного продажу або передачі об'єкта в оренду. Крім того, оціночна вартість використовується для визначення доходу від продажу об'єкта нерухомості і, відповідно, для визначення розміру податку, що сплачується з такого доходу. Для самих банків наявність оцінки нерухомості також дуже важливо, тому що дозволяє знизити ризики неповернення кредиту.

При оцінці нерухомості враховуються багато показників, а в залежності від мети оцінювання, оцінки виконуються різними методами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Поняття оцінки вартості майна

Оцінка вартості нерухомості – процес розрахунку вартості об'єкта нерухомості або окремих прав користування . Оцінка вартості нерухомості включає: визначення вартості права власності або інших прав, наприклад, права оренди, права користування і т. д. по відношенню до різних об'єктів нерухомості.

1.1.1 Вартість об'єктів оцінки. Види вартості

В оцінній практиці існують кілька видів вартості.

Вартість нерухомості підрозділяється на наступні види:

- ринкова вартість;
- інвестиційна вартість;
- ліквідаційна вартість;
- кадастрова вартість.

При визначенні ринкової вартості об'єкта оцінки визначається найбільш ймовірна ціна, за якою об'єкт оцінки може бути відчужений на дату оцінки на відкритому ринку в умовах конкуренції, коли сторони угоди діють розумно, розташовуючи всією необхідною інформацією, а на величині ціни угоди не відбиваються які-небудь надзвичайні обставини, тобто коли:

- одна зі сторін угоди не зобов'язана відчужувати об'єкт оцінки, а інша сторона не зобов'язана приймати виконання;
- сторони угоди добре інформовані про предмет угоди і діють у своїх інтересах;
- об'єкт оцінки представлений на відкритому ринку за допомогою публічної оферти, типової для аналогічних об'єктів оцінки;

- ціна угоди являє собою розумну винагороду за об'єкт оцінки і примусу до здійснення угоди щодо сторін угоди з будь-якої сторони не було;
- платіж за об'єкт оцінки виражений у грошовій формі.

Інвестиційна вартість – це вартість об'єкта оцінки для конкретної особи або групи осіб при встановлених цією особою (особами) інвестиційних цілях використання об'єкта оцінки.

При визначенні інвестиційної вартості на відміну від визначення ринкової вартості облік можливості відчуження за інвестиційною вартістю на відкритому ринку не обов'язковий. Інвестиційна вартість може використовуватися для вимірювання ефективності інвестицій.

При визначенні інвестиційної вартості об'єкта оцінки визначається вартість для конкретної особи або групи осіб при встановлених цією особою (особами) інвестиційних цілях використання об'єкта оцінки – інвестиційної нерухомості для отримання доходу у вигляді орендної плати та приросту вартості. При визначенні інвестиційної вартості, на відміну від визначення ринкової вартості, облік можливості відчуження за інвестиційною вартістю на відкритому ринку не обов'язковий.

При визначенні ліквідаційної вартості об'єкта оцінки визначається розрахункова величина, що відображає найбільш ймовірну ціну, по якій даний об'єкт оцінки може бути відчужений за термін експозиції об'єкта оцінки, менший типового терміну експозиції для ринкових умов, в умовах, коли продавець змушений здійснити операцію з відчуження майна. При визначенні ліквідаційної вартості, на відміну від визначення ринкової вартості, враховується вплив надзвичайних обставин, які змушують продавця продавати об'єкт оцінки на умовах, які не відповідають ринковим .

При визначенні кадастрової вартості об'єкта оцінки визначається методами масової оцінки ринкова вартість, встановлена і затверджена відповідно

до законодавства, що регулює проведення кадастрової оцінки. Кадастрова вартість визначається оцінювачем, зокрема, для цілей оподаткування.

1.1.2 Фактори, які впливають на ціну об'єктів

Можна виділити основні групи факторів, що впливають на ринкову вартість майна.

Економічні чинники, що роблять істотний вплив на вартість майна, об'єднують специфічні властивості конкретного майна (прибутковість, ліквідність, корисність), а також загальні економічні характеристики (соціально-економічна і політична ситуація в світі або країні, економічна ситуація в регіоні, тенденції розвитку галузі, виду діяльності, до якого належить майно).

Фактори рівня і ступеня державного регулювання (загальні нормативно-правові умови і обмеження обороту нерухомості і способів землекористування; спеціальні правові норми, що впливають на вартість майна типу нормативного встановлення орендних ставок, обмеження прав власності, закони про охорону навколишнього середовища тощо).

Державна і місцева податкова політика (умови і пільги з оподаткування майна, інвестицій).

Вплив навколишнього середовища і територіального розташування, природних і штучних чинників, кліматичних умов місцевості (рівень опадів, температура, вологість, топографія ділянки, характеристика ґрунтів, вміст токсичних елементів, природні бар'єри на шляху подальшої забудови у вигляді річок, гірських хребтів, озер, а також схильність району стихійним лихам, близькість транспортних шляхів, ринків збуту і сировини, наявність трудових ресурсів).

Фактичні кількісні та якісні характеристики майна, його характерні риси, власні властивості і якості (тип і вид майна, його вік, фактичне фізичне і технічний стан, потужність, розміри, колір, тощо).

Співвідношення попиту і пропозиції. Попит залежить, в свою чергу, від числа потенційних покупців, їх смаків і уподобань, прибутковості і ліквідності, корисності майна, виставленого на продаж. На попит також впливає величина відкладеного попиту, тобто незадоволені інвестиційні очікування і ціни на аналогічні або взаємозамінні об'єкти, прогнози зміни цін і купівельної спроможності. Досить часто структура потенційного попиту не відповідає структурі пропозиції за видами, місцезнаходженням і конкретним характеристикам майна.

Рівень конкуренції, який залежить від наявності альтернативних варіантів пропозиції майна або інвестування. На рівень конкуренції впливає наявність нових потужностей, орендні ставки і ціни, що склалися на готові товари, собівартість альтернативних витрат.

1.2 Поняття експертної системи

Експертна система – це програма, яка поводить себе подібно експерту в деякій, звичайно вузькій прикладній області. Типові застосування експертних систем містять у собі такі завдання, як медична діагностика, локалізація несправностей в устаткуванні й інтерпретація результатів вимірів. Експертні системи повинні вирішувати задачі, що вимагають для свого рішення експертних знань в деякій конкретній області. В тій чи іншій формі експертні системи повинні володіти цими знаннями. Тому їх також називають системами, заснованими на знаннях (рис.1.1).

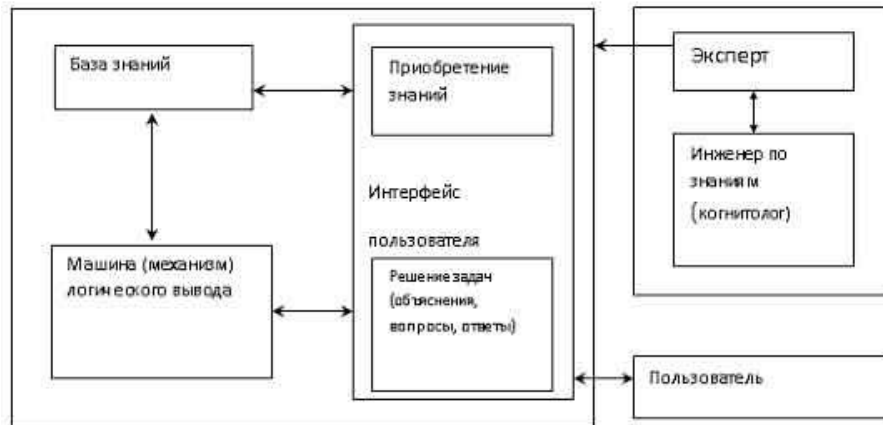


Рисунок 1.1 – Структура експертної системи

Однак не всяку систему, засновану на знаннях, можна розглядати як експертну. Експертна система повинна також уміти якимось чином пояснювати свою поведінку і свої рішення користувачу, так само, як це робить експерт-людина. Це особливо необхідно в областях, для яких характерна невизначеність, неточність інформації (наприклад, у медичній діагностиці). У цих випадках здатність до пояснення потрібна для того, щоб підвищити ступінь довіри користувача до рад системи, а також для того, щоб дати можливість користувачу знайти можливий дефект у міркуваннях системи. У зв'язку з цим в експертних системах варто передбачати дружню взаємодію з користувачем, що робить для користувача процес міркування системи "прозорим".

Часто до експертних систем висувають додаткову вимогу – здатність мати справу з невизначеністю і неповнотою. Інформація про поставлену задачу може бути неповною або ненадійною; відносини між об'єктами предметної області можуть бути наближеними.

У найзагальнішому випадку для того, щоб побудувати експертну систему, ми повинні розробити механізми виконання наступних функцій системи:

Рішення задач з використанням знань про конкретну предметну область можливо, при цьому виникне необхідності мати справу з невизначеністю;

Взаємодія з користувачем, включаючи пояснення намірів і рішень системи під час і після закінчення процесу рішення задачі.

Кожна з цих функцій може виявитися дуже складною і залежить від прикладної області, а також від різних практичних вимог. В процесі розробки і реалізації можуть виникати різноманітні важкі проблеми [1]¹⁾.

1.2.1 Подання знань в експертних системах і моделі

Перше і основне питання, яке треба вирішити при поданні знань – це питання визначення складу знань, тобто визначення того, "що представляти" в експертній системі. Важливість питання "що представляти" визначається тим, що рішення саме цієї проблеми забезпечує адекватне відображення модельованих сутностей в системі. Друге питання стосується того, "як представляти" знання. Необхідно відзначити, що ці дві проблеми не є незалежними. Дійсно, обраний спосіб представлення може виявитися непридатним в принципі або неефективним для вираження деяких знань.

На нашу думку, питання "як представляти" можна розділити на дві в значній мірі незалежні завдання: як організувати (структурувати) знання і як представити знання в обраному формалізмі.

Прагнення виділити організацію знань в самостійну задачу викликано, зокрема, тим, що ця задача виникає для будь-якої мови уявлення і способи вирішення цієї задачі є однаковими (або подібними) незалежно від використовуваного формалізму.

¹⁾ [1] П.Джексон. Введение в экспертные системы. Спб.: Питер. 2007. 384 с.

Отже, в коло питань, що вирішуються при поданні знань, будемо включати наступні:

- визначення складу знань, що подаються;
- організацію знань;
- уявлення знань, тобто визначення "моделі уявлення";

Склад знань ЕС визначається наступними факторами:

- проблемним середовищем;
- архітектурою експертної системи;
- потребами і цілями користувачів;
- мовою спілкування.

Відповідно до загальної схеми статичної експертної системи для її функціонування потрібні такі знання:

- знання про процес вирішення завдання (тобто керуючі знання), що використовуються інтерпретатором (вирішувачем);
- знання про мову спілкування і способах організації діалогу, використувані лінгвістичним процесором (діалоговим компонентом);
- знання про способи подання та модифікації знань, які використовуються компонентом придбання знань;
- підтримують структурні і керуючі знання, які використовуються пояснювальним компонентом.

Для динамічної ЕС, крім того, необхідні наступні знання:

- знання про методи взаємодії із зовнішнім оточенням;
- знання про моделі зовнішнього світу.

Залежність складу знань від вимог користувача проявляється в наступному:

- які завдання (із загального набору завдань) і з якими даними хоче вирішувати користувач;
- які бажані способи і методи вирішення;

- при яких обмеженнях на кількість результатів і способи їх отримання повинна бути вирішена задача;
- які вимоги до мови спілкування і організації діалогу;
- який ступінь спільності (конкретності) знань про проблемну область, доступна користувачу;
- які цілі користувачів.

Склад знань про мову спілкування залежить як від мови спілкування, так і від необхідного рівня розуміння.

З урахуванням архітектури експертної системи знання доцільно ділити на інтерпретовані і неінтерпретовані. До першого типу відносяться ті знання, які здатний інтерпретувати вирішувач (інтерпретатор). Всі інші знання відносяться до другого типу. Вирішувач не знає їх структури і змісту. Якщо ці знання використовуються будь-яким компонентом системи, то він не "усвідомлює" цих знань. Неінтерпретовані знання поділяються на допоміжні знання, що зберігають інформацію про лексику і граматику мови спілкування, інформацію про структуру діалогу, і підтримують знання. Допоміжні знання обробляються природно-мовної компонентою, але хід цієї обробки вирішувач не усвідомлює, так як цей етап обробки вхідних повідомлень є допоміжним для проведення експертизи. Підтримують знання використовуються при створенні системи і при виконанні пояснень. Підтримують знання виконують роль описів (обґрунтувань) як інтерпретованих знань, так і дій системи. Підтримують знання поділяються на технологічні і семантичні. Технологічні підтримуючі знання містять відомості про час створення описуваних ними знань, про автора знань і т.п. Семантичні підтримуючі знання містять смислове опис цих знань. Вони містять інформацію про причини введення знань, про призначення знань, описують спосіб використання знань і отриманий ефект. Підтримуючі знання мають описовий характер.

Інтерпретуючі знання можна розділити на предметні знання, керуючі знання і знання про подання. Знання про подання містять інформацію про те, яким чином (в яких структурах) в системі представлені інтерпретовані знання.

Предметні знання містять дані про предметну область і способи перетворення цих даних при вирішенні поставлених завдань. Відзначимо, що по відношенню до предметних знань знання про подання і знання про управління є метазнання. У предметних знаннях можна виділити описатели і власне предметні знання. Описувачі містять певну інформацію про предметні знання, таку, як коефіцієнт визначеності правил і даних, заходи важливості і складності. Власне предметні знання розбиваються на факти і виконувані твердження. Факти визначають можливі значення сутностей і характеристик предметної області. Виконувані твердження можуть містити інформацію про те, як можна змінювати опис предметної області в ході вирішення завдань. Говорячи іншими словами, виконувані твердження – це знання, що задають процедури обробки. Однак ми уникаємо використовувати термін "процедурні знання", так як хочемо підкреслити, що ці знання можуть бути задані не тільки в процедурній, але і в декларативній формі.

Керуючі знання можна розділити на фокусуючі і вирішальні. Фокусуючі знання описують, які знання слід використовувати в тій чи іншій ситуації. Зазвичай фокусують знання містять відомості про найбільш перспективних об'єктах або правилах, які доцільно використовувати при перевірці відповідних гіпотез. У першому випадку увага зосереджується на елементах робочої пам'яті, у другому – на правилах бази знань. Вирішальні знання містять інформацію, яка використовується для вибору способу інтерпретації знань, відповідного до поточної ситуації. Ці знання застосовуються для вибору стратегій або евристик, найбільш ефективних для вирішення даного завдання.

Якісні та кількісні показники експертної системи можуть бути значно поліпшені за рахунок використання метазнань, тобто знань про знання. Метазнання не уявляють деяку єдину сутність, вони можуть застосовуватися для досягнення різних цілей. Перерахуємо можливі призначення метазнань:

- метазнання у вигляді стратегічних метаправил використовуються для вибору релевантних правил;
- метазнання використовуються для обґрунтування доцільності застосування правил з області експертизи;
- метаправила використовуються для виявлення синтаксичних і семантичних помилок в предметних правилах;
- метаправила дозволяють системі адаптуватися до оточення шляхом перебудови предметних правил і функцій;
- метаправила дозволяють явно вказати можливості і обмеження системи, тобто визначити, що система знає, а що не знає.

Питання організації знань необхідно розглядати в будь-якому поданні, і їх рішення в значній мірі не залежить від обраного способу (моделі) уявлення. Виділимо наступні аспекти проблеми організації знань:

- організація знань за рівнями уявлення і за рівнями детальності;
- організація знань в робочій пам'яті;
- організація знань в базі знань.

1.2.2 Організація знань в базі даних

Показником інтелектуальності системи з точки зору представлення знань вважається здатність системи використовувати в потрібний момент необхідні (релевантні) знання. Системи, що не мають средств для визначення релевантних знань, неминуче стикаються з проблемою "комбінаторного вибуху". Можна стверджувати, що ця проблема є однією з основних причин,

що обмежують сферу застосування експертних систем. У проблемі доступу до знань можна виділити три аспекти: зв'язність знань і даних, механізм доступу до знань і спосіб зіставлення.

Можливості підключення знань є основним способом, що забезпечує прискорення пошуку релевантних знань. Більшість фахівців прийшли до переконання, що знання слід організовувати навколо найбільш важливих об'єктів (сутностей) предметної області. Всі знання, що характеризують деяку сутність, зв'язуються і представляються у вигляді окремого об'єкта. За такої організації знань, якщо системі потрібна була інформація про деякої сутності, то вона шукає об'єкт, що описує цю сутність, а потім вже всередині об'єкта відшукує інформацію про даної суті. В об'єктах доцільно виділяти два типи зв'язок між елементами: зовнішні і внутрішні. Внутрішні зв'язки об'єднують елементи в єдиний об'єкт і призначені для вираження структури об'єкта. Зовнішні зв'язки відображають взаємозалежності, існуючі між об'єктами в області експертизи. Багато дослідників класифікують зовнішні зв'язки на логічні і асоціативні. Логічні зв'язки виражають семантичні відносини між елементами знань. Асоціативні зв'язки призначені для забезпечення взаємозв'язків, сприяють прискоренню процесу пошуку релевантних знань.

Основною проблемою при роботі з великою базою знань є проблема пошуку знань, релевантних розв'язуваної задачі. У зв'язку з тим, що в оброблюваних даних може не міститися явних вказівок на значення, необхідні для їх обробки, необхідний більш загальний механізм доступу, ніж метод прямого доступу (метод явних посилань). Завдання цього механізму полягає в тому, щоб по деякому опису суті, наявного в робочій пам'яті, знайти, базі знань об'єкти, що задовольняють цій опису. Очевидно, що впорядкування і структурування знань можуть значно прискорити процес пошуку.

Знаходження бажаних об'єктів в загальному випадку доречно розглядати як двоетапний процес. На першому етапі, відповідному процесу вибору по асоціативним зв'язкам, відбувається попередній вибір в базі знань потенцій-

них кандидатів на роль бажаних об'єктів. На другому етапі шляхом виконання операції зіставлення потенційних кандидатів з описами кандидатів здійснюється остаточний вибір шуканих об'єктів. При організації подібного механізму доступу виникають певні труднощі: Як вибрати критерій придатності кандидата? Як організувати роботу в конфліктних ситуаціях? і т.п.

Операція зіставлення може використовуватися не тільки як засіб вибору потрібного об'єкту з безлічі кандидатів; вона може бути використана для класифікації, підтвердження, декомпозиції і корекції. Для ідентифікації невідомого об'єкта він може бути зіставлений з деякими відомими зразками. Це дозволить класифікувати невідомий об'єкт як такий відомий зразок, при зіставленні з якими було отримано кращі результати. При пошуку зіставлення використовується для підтвердження деяких кандидатів з безлічі можливих. Якщо здійснювати зіставлення деякого відомого об'єкта з невідомим описом, то в разі успішного зіставлення буде здійснена часткова декомпозиція опису.

Операції зіставлення дуже різноманітні. Зазвичай виділяють наступні їх форми: синтаксичне, параметричне, семантичне і примушуване зіставлення. У разі синтаксичного зіставлення співвідносять форми (зразки), а не зміст об'єктів. Успішним є зіставлення, в результаті якого зразки виявляються ідентичними. Зазвичай вважається, що змінна одного зразка може бути ідентична будь-якій константі (або виразу) іншого зразка. Іноді на змінні, що входять в зразок, накладають вимоги, що визначають тип констант, з якими вони можуть зіставлятися. Результат синтаксичного зіставлення є бінарним: зразки зіставляються або не пов'язані із . В параметричному зіставленні вводиться параметр, що визначає ступінь порівняння. У разі семантичного зіставлення співвідносять не зразки об'єктів, а їх функції. У разі примушуваного зіставлення один зіставляється зразок розглядається з точки зору іншого. На відміну від інших типів зіставлення тут завжди може бути отримано позитивний результат. Питання полягає в силі примусу. Примус можуть виконувати спеціа-

льні процедури, які пов'язують з об'єктами. Якщо ці процедури не в змозі здійснити зіставлення, то система повідомляє, що успіх може бути досягнутий тільки в тому випадку, якщо певні частини розглянутих сутностей можна вважати зіставляти.

1.2.3 Моделі і методи представлення знань

У багатьох випадках для прийняття рішень в тій чи іншій сфері людської діяльності невідомий алгоритм рішення, тобто відсутня чітка послідовність дій, явно призводять до необхідного результату. Наприклад:

- проектування розвитку важкої промисловості;
- оптимальне розміщення персоналу всередині будівлі;
- лікування хворої людини.

При прийнятті рішення в таких випадках необхідно мати деяку суму знань про саму цю галузі. Наприклад: при виборі найкращого ходу в конкретній шаховій позиції необхідні знання про правила гри, силі шахових фігур, стратегії і тактиці і багато іншого. Під знаннями розуміється то, що стало відомо після вивчення. Сукупність знань, потрібних для прийняття рішень, прийнято називати предметною областю або знаннями про предметну область.

У будь-якій предметній області є свої поняття і зв'язки між ними, своя термінологія, свої закони, що зв'язують між собою об'єкти даних предметної області, свої процеси і події. Крім того, кожна предметна область має свої методи вирішення завдань.

Вирішуючи завдання такого виду на ЕОМ використовують ІС, ядром яких є бази знань, що містять основні характеристики предметних областей.

При побудові баз знань традиційні мови, засновані на чисельному поданні даних, є неефективними. Для цього використовуються спеціальні мови представлення знань, засновані на символічному поданні даних. Вони ділять-

ся на типи за формальними моделям представлення знань. Різні автори по-різному ці моделі класифікують. Основні моделі знань представлені на рис.1.2.

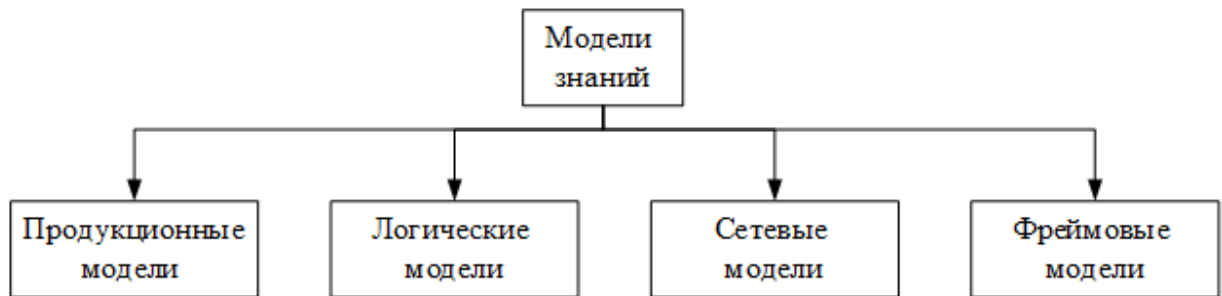


Рисунок 1.2 – Основні моделі знань

Продукційна модель знань. Продукційні моделі можна вважати найбільш поширеними моделями подання знань. Продукційна модель – це модель, заснована на правилах, що дозволяє уявити знання у вигляді пропозицій типу: «ЯКЩО умова, ТО дія».

Продукційна модель володіє тим недоліком, що при накопиченні досить великого числа (порядку декількох сотень) продукцій вони починають суперечити один одному.

Системи обробки знань, що використовують продукційну модель отримали назву «продукційних систем». До складу експертних систем продукційного типу входять база правил (знань), робоча пам'ять і інтерпретатор правил (вирішувач), який реалізує певний механізм логічного висновку. Будь-яке продукційне правило, яке міститься в базі знань, складається з двох частин: антецедент і консеквента. Антецедент є посилку правила (умовну частину) і складається з елементарних пропозицій, з'єднаних логічними зв'язками «і», «або». Консеквент (висновок) включає одне або кілька пропозицій, які висловлюють або деякий факт, або вказівку на певну дію, яка під-

лягає виконанню. Продукційні правила прийнято записувати у вигляді антецедент-консеквент.

Приклади продукційних правил:

ЯКЩО «двигун не заводиться» І «стартер двигуна не працює»
ТО «неполадки в системі електроживлення стартера»

Основні переваги систем, заснованих на продукційних моделях – простота подання знань і організації логічного висновку.

До недоліків таких систем можна віднести наступне:

- відміну від структур знань, властивих людині;
- неясність взаємних відносин правил;
- складність оцінки цілісного образу знань;
- низька ефективність обробки знань.

При розробці невеликих систем (десятки правил) проявляються в основному позитивні сторони продукційних моделей знань, проте при збільшенні обсягу знань більш помітними стають слабкі сторони.

Логічна модель знань. Основна ідея при побудові логічних моделей знань полягає в наступному: вся інформація, необхідна для вирішення прикладних завдань, розглядається як сукупність фактів і тверджень, які представляються як формули в деякій логіці. Знання відображаються сукупністю таких формул, а отримання нових знань зводиться до реалізації процедур логічного висновку.

Основні переваги логічних моделей знань:

- в якості «фундаменту» тут використовується класичний апарат математичної логіки, методи якої досить добре вивчені і формально обгрунтовані;
- існують досить ефективні процедури виведення, в тому числі реалізовані в мовах логічного програмування;

- в базах знань можна зберігати лише безліч аксіом, а всі інші знання отримувати з них за правилами виведення.

У логічних моделях знань слова, що описують сутність предметної області, називаються термами (константи, змінні, функції), а слова, що описують відносини сутностей – предикатами.

Предикат – логічна N-арная пропозиціональна функція, певна для предметної області і приймає значення або істинності, чи хибності. Пропозиціональною називається функція, яка ставить у відповідність об'єктів з області визначення одне з істинних значень («істина», «брехня»). Предикат приймає значення «істина» або «брехня» в залежності від значень назв термів [2]¹⁾.

Спосіб опису предметної області, який використовується в логічних моделях знань, призводить до втрати деяких нюансів, властивих природному сприйняттю людини, і тому знижує описову можливість таких моделей.

Складнощі виникають при описі «многосортності» світів, коли об'єкти не є однорідними. Так, висловлювання « $2 + 2 = 4$ » або «Київ – столиця України» мають одне і те ж значення «істина», але різний зміст. З метою подолання труднощів і розширення описових можливостей логічних моделей знань розробляються псевдофізичні логіки, тобто логіки, які оперують з нечіткими, що забезпечують індуктивні (від часткового до загального), дедуктивні (від загального до конкретного) і продуктивні (на одному рівні спільності) висновки. Такі розширені моделі, що поєднують можливості логічного і лінгвістичного підходів, прийнято називати логіко-лінгвістичними моделями предметної області.

Мережева модель знань. Однозначне визначення семантичної мережі (мережевий моделі знань) в даний час відсутня. В інженерії знань під нею

¹⁾ [2] Л. Н. Ясницкий. Введение в искусственный интеллект. М.: Академия. 2005. 420 с.

мається на увазі граф, що відображає зміст цілісного образу. Вузли графа відповідають поняттям і об'єктам, а дуги – відносинам між об'єктами.

Семантична мережа як модель найбільш часто використовується для представлення декларативних знань. За допомогою цієї моделі реалізуються такі властивості системи знань, як інтерпретируемість і зв'язність. За рахунок цих властивостей семантична мережа дозволяє знизити обсяг збережених даних, забезпечує висновок умовиводів по асоціативним зв'язкам.

Однією з перших відомих моделей, заснованих на семантичній мережі, є TLC-модель (Teachable Language Comprehender – доступний механізм розуміння мови), розроблена в 1968 році. Модель використовувалася для подання семантичних відносин між концептами (словами) з метою опису структури довготривалої пам'яті людини в психології.

Як правило, розрізняють екстенціональні і інтенціональні семантичні мережі. Екстенціональна семантична мережа описує конкретні відносини даної ситуації. Інтенціональна – імена класів об'єктів, а не індивідуальні імена об'єктів. Зв'язки в інтенціональній мережі відображають ті відносини, які завжди притаманні об'єктам даного класу.

Фреймова модель знань. Фреймова модель заснована на концепції Марвіна Мінскі (Marvin Minsky) – професора Массачусетського технологічного інституту, засновника лабораторії штучного інтелекту, автора низки фундаментальних робіт. Фреймова модель являє собою систематизовану психологічну модель пам'яті людини і його свідомості.

Фрейм – це мінімально можливе опис сутності якої-небудь події, ситуації, процесу або об'єкта. Існує й інше розуміння фрейму – це асоціативний список атрибутів. Поняття «мінімально можливе» означає, що при подальшому спрощенні опису втрачається його повнота, і воно перестає визначати ту одиницю знань, для якої було призначене. Подання знань за допомогою фреймів розуміється як один із способів представлення знань про ситуації. Фрейм має ім'я (назва) та складається з слотів.

Слоти – це незаповнені (нульові) позиції фрейма. Якщо у фрейма всі слоти заповнені – це опис конкретної ситуації. У перекладі з англійської слово «фрейм» означає «рамка», а слово «слот» – «щілину». На відміну від моделей інших типів у фреймових моделях фіксується жорстка структура інформаційних одиниць, яка називається протофрейма. У загальному вигляді структура інформаційних одиниць виглядає наступним чином:

$$f = [(r_1, v_1), \dots, (r_n, v_n)], \quad (1.1)$$

де f – ім'я фрейму;

v_i – значення слота.

Значенням слота може бути практично що завгодно (числа, математичні співвідношення, тексти на природній мові або на мові програм, посилання на інші слоти даного фрейму). Значенням слота може виступати і окремий фрейм, що є дуже зручним для упорядкування знань за ступенем спільності. Виняток з фрейма будь-якого слота робить його неповним, а іноді і безглуздим.

При конкретизації фрейму йому і слотам приписуються конкретні імена і відбувається заповнення слотів. Таким чином з протофрейма виходять фрейми – екземпляри. Перехід від вихідного протофрейма до кадру – екземпляру може бути багатокроковим, за рахунок поступового уточнення значень слотів [3]¹⁾.

Розглянемо деякий протофрейма:

(СПИСОК СПІВРОБІТНИКІВ :

¹⁾[3] Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. Статические и динамические экспертные системы: учеб. пособие для вузов. М.: Финансы и статистика. 1996. 320 с.

Прізвище (значення слота 1);
 Рік народження (значення слота 2);
 Спеціальність (значення слота 3);
 Стаж (значення слота 4)
).

Якщо в якості значень слотів використовувати конкретні дані, то отримаємо фрейм – екземпляр:

(СПИСОК СПІВРОБІТНИКІВ:
 Прізвище (Попов – Сидоров – Іванов – Петров);
 Рік народження (1965 – 1975 – 1980 – 1978);
 Спеціальність (директор – бухгалтер – технік – кур'єр);
 Стаж (15 – 7 – 3 – 4)
).

Зв'язки між фреймами задаються значеннями спеціального слота з ім'ям «зв'язок».

Фрейми поділяються на :

- фрейм-екземпляр – конкретна реалізація фрейму, що описує поточний стан в предметної області;
- фрейм-зразок – шаблон для опису об'єктів або допустимих ситуацій предметної області;
- фрейм-клас – фрейм верхнього рівня для представлення сукупності фреймів зразків.

1.2.4 Методи пошуку рішень в експертних системах

Методи вирішення завдань, засновані на зведенні їх до пошуку, залежать від особливостей предметної області, в якій вирішується завдання, і від вимог, що пред'являються користувачем до вирішення. Особливості предме-

тної області з точки зору методів вирішення можна характеризувати наступними параметрами:

- розмір, який визначає обсяг простору, в якому належить шукати рішення;
- змінність області, характеризує ступінь змінності області в часі і просторі (тут будемо виділяти статичні і динамічні області);
- повнота моделі, яка описує область, характеризує адекватність моделі, яка використовується для опису даної області. Зазвичай якщо модель не повна, то для опису області використовують кілька моделей, які доповнюють один одного за рахунок відображення різних;
- визначеність даних про розв'язуваної задачі, характеризує ступінь точності (хибності) і повноти (неповноти) даних. Точність (помилковість) є показником того, що предметна область з точки зору вирішуваних завдань описана точними або неточними даними; під повнотою (неповнотою) даних розуміється достатність (недостатність) вхідних даних для однозначного вирішення задачі.

Вимоги користувача до результату завдання, розв'язуваної за допомогою пошуку, можна характеризувати кількістю рішень і властивостями результату і (або) способом його отримання. Параметр "кількість рішень" може приймати такі основні значення: одне рішення, кілька рішень, всі рішення. Параметр "властивості" задає обмеження, яким повинен задовольняти отриманий результат або спосіб його отримання. Так, наприклад, для системи, яка видає рекомендації з лікування хворих, користувач може вказати вимога не використовувати деякий ліки (в зв'язку з його відсутністю або в зв'язку з тим, що воно протипоказане даному пацієнтові). Параметр "властивості" може визначати і такі особливості, як час вирішення ("не більше ніж", "діапазон часу" і т.п.), обсяг пам'яті, використовуваної для отримання резуль-

тату, вказівка про обов'язковість (неможливості) використання будь-яких знань (даних) і т.п.

Отже, складність завдання, що визначається вищенаведеним набором параметрів, варіюється від простих завдань малої розмірності з незмінними певними даними і відсутністю обмежень на результат і спосіб його отримання до складних задач великої розмірності із змінними, помилковими і неповними даними та довільними обмеженнями на результат і спосіб його отримання. Із загальних міркувань ясно, що будь-яким одним методом не можна вирішити всі завдання. Зазвичай одні методи перевершують інші тільки по деяким з перерахованих параметрів.

Розглянуті нижче методи можуть працювати в статичних і динамічних проблемних середовищах. Для того щоб вони працювали в умовах динаміки, необхідно враховувати час життя значень змінних, джерело даних для змінних, а також забезпечувати можливість зберігання історії значень змінних, моделювання зовнішнього оточення і оперування тимчасовими категоріями в правилах.

Існуючі методи вирішення завдань, які використовуються в експертних системах, можна класифікувати наступним чином:

- методи пошуку в одному просторі – методи, призначені для використання в наступних умовах: області невеликий розмірності, повнота моделі, точні і повні дані;
- методи пошуку в ієрархічних просторах – методи, призначені для роботи в областях великої розмірності;
- методи пошуку при неточних і неповних даних;
- методи пошуку, що використовують кілька моделей, призначені для роботи з областями, для адекватного опису яких однією моделі недостатньо.

Передбачається, що перераховані методи при необхідності повинні об'єднуватися для того, щоб дозволити вирішувати завдання, складність яких зростає одночасно за кількома параметрами.

1.2.4.1 Пошук рішень в одному просторі

Методи пошуку рішень в одному просторі зазвичай поділяються на пошук в іншій області станів, пошук методом редукції, евристичний пошук і пошук методом "генерація-перевірка" [4]¹⁾.

Пошук в просторі станів. Завдання пошуку в просторі станів звичайно формулюється в теоретико-графовій інтерпретації.

Нехай задана трійка (S_0, F, S_T) , де S_0 – безліч початкових станів (умови завдання), F – безліч операторів завдання, що відображають одні стани в інші; S_T – безліч кінцевих (цільових) станів (рішень задачі).

У цій постановці вирішити задачу – значить визначити таку послідовність операторів, яка перетворює початковий стану в кінцеві. Процес рішення можна представити у вигляді графа $G = (X, Y)$, де $X = \{x_0, x_1, \dots\}$ – безліч (в загальному випадку нескінченне) вершин графа, кожна з яких ототожнюється з одним із станів, а Y – безліч, що містить пари вершин (x_i, x_j) , (x_i, x_j) . Якщо кожна пара (x_i, x_j) неупорядкований, то її називають ребром, а граф – неорієнтованим. Якщо для кожної пари (x_i, x_j) поставлено порядок (напрямок), то пару (x_i, x_j) називають дугою (орієнтованим ребром), а граф називають орієнтованим (спрямованим). Вершини пари (x_i, x_j) називають кінцевими точками ребра (дуги).

Пошук в просторі станів природно представити у вигляді орієнтованого графа. Наявність пари (x_i, x_j) свідчить про існування деякого оператора, що

¹⁾[4] Д. Б. Гаскаров. Интеллектуальные информационные системы. М.: Высшая школа. 2003. 542 с.

перетворює стан, відповідне вершині x_i , в стан x_j . З точки зору пошуку в просторі станів для деякої вершини x_i доречно виділити безліч всіх спрямованих пар (x_i, x_j) , тобто безліч дуг, що виходять з вершини x_i (батьківської вершини), і безліч вершин (званих дочірніми вершинами), в які ці дуги призводять. Безліч дуг, що виходять з вершини x_i , відповідає безлічі операторів, які можуть бути застосовані до стану, відповідному вершині x_i .

У безлічі вершин X виділяють підмножина вершин відповідне безлічі початкових станів (S_0) , і підмножина вершин, відповідне безлічі кінцевих (цільових) станів (S_T) . Безліч X_T може бути задано як явно, так і неявно, тобто через властивості, якими повинні володіти цільові стану.

Відзначимо, що граф G може бути заданий явно і неявно. Неявне завдання графа G полягає у визначенні безлічі (відповідного безлічі початкових станів) і безлічі операторів, які, будучи застосовні до деякої вершини графа, дають всі її дочірні вершини.

Отже, граф G задає простір станів, тобто простір, в якому здійснюється пошук рішення. Побудова простору здійснюється за допомогою наступного процесу. Береться якась вершина, до неї застосовуються всі можливі оператори, які породжують все дочірні вершини. Цей процес називають процесом розкриття вершин. Якщо отримана цільова вершина, то це не розкрито. Процес побудови простору станів закінчується, коли всі нерозкриті вершини є цільовими, або термінальними (тобто вершинами, до яких не можна застосувати ніяких операторів). У зв'язку з тим, що простір станів може містити нескінченну кількість вершин, на практиці процес породження простору обмежують або часом, або об'ємом пам'яті.

У практичних додатках часто потрібно забезпечити повноту пошуку, тобто організувати пошук так, щоб всі цільові вершини були знайдені, якщо вони існують. Надійним способом забезпечення повноти є повний перебір всіх вершин. Для завдання процесу перебору необхідно визначити порядок, в якому будуть перебиратися вершини графа. Зазвичай виділяють два основ-

них способи пошуку: пошук в глибину і пошук в ширину. При пошуку в глибину спочатку розкривається та вершина, яка була побудована самої останньої. Глибина вершини в графі визначається так:

- глибина початкової вершини дорівнює нулю;
- глибина ненавчальної вершини дорівнює одиниці плюс глибина найбільш близькою батьківської вершини.

При практичній реалізації пошук в глибину в деякому напрямку завершується в наступних випадках:

- при досягненні цільової вершини;
- при досягненні термінальної вершини;
- при побудові в ході пошуку вершини, глибина якої перевищує деяку граничну глибину.

При пошуку в ширину вершини розкриваються в тому ж порядку, в якому вони породжуються.

Якщо в простір станів ввести оператори, що переводять стан S_i в попередній стан S_{i-1} , то пошук можна здійснювати не тільки в напрямку від початкового стану до цільовим, а й у зворотному напрямку. Пошук першого типу називають пошуком від даних, або прямим пошуком, а пошук другого типу – пошуком від мети, або зворотним пошуком. Можна організувати пошук в двох напрямках одночасно. Такий пошук називають двонаправленим (або бінаправленим).

На рис.1.3 наведено приклади розв'язання задач пошуком в глибину (рис.1.3, а) і в ширину (рис.1.3, б). Вершини пронумеровані в тому порядку, в якому вони розкриваються (а не породжуються), цільові вершини позначені чорними квадратами, а термінальні – білими квадратами. При використанні кожного із способів можуть бути знайдені всі рішення. При переборі всього простору обидва методи аналізуватимуть однакову кількість вершин, однак метод пошуку в ширину вимагатиме значно більше пам'яті, так як він запам'ятовує всі шляхи пошуку (а не один, як при пошуку в глибину).

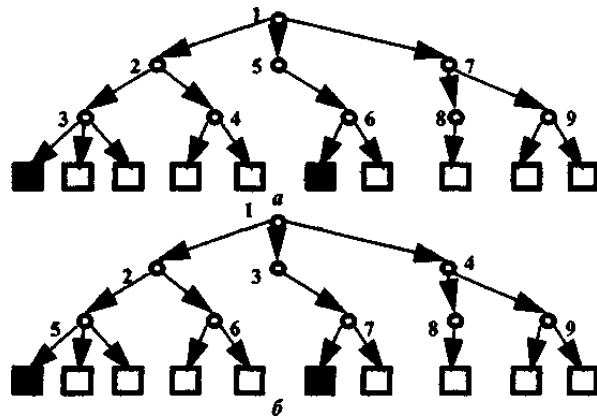


Рисунок 1.3 – Простір станів, побудованих пошуком в глибину (а) і пошуком в ширину (б)

Пошук методом редукції. При пошуку методом редукції вирішення завдання зводиться до вирішення сукупності утворюють її підзадач. Цей процес повторюється для кожної підзадачі до тих пір, поки кожна з отриманого набору підзадач, що утворюють рішення вихідної задачі, не матиме очевидне рішення. Підзадача вважається очевидною, якщо її рішення загальновідомо або отримано раніше. Процес рішення задачі розбиттям її на підзадачі можна представити у вигляді спеціального спрямованого графа G , званого І/АБО-графом. Кожній вершині цього графа ставиться у відповідність опис деякої задачі (підзадачі). У графі виділяють два типи вершин: кон'юнктивні вершини і диз'юнктивні вершини. Кон'юнктивні вершини, або вершини типу "І", разом зі своїми дочірніми вершинами інтерпретуються так: рішення задачі зводиться до вирішення всіх її підзадач, відповідних дочірнім вершин кон'юнктивній вершини. Диз'юнктивні вершини, або вершини типу "АБО", разом зі своїми дочірніми вершинами інтерпретуються так: рішення задачі зводиться до вирішення будь-якої з її підзадач, відповідних дочірнім вершин диз'юнктивній вершини. Відзначимо, що деякі автори визначають вершини І і АБО інакше.

У безлічі вершин І/АБО-графа виділяють підмножина початкових вершин, тобто задач, які слід вирішити, і підмножина кінцевих (цільових) вершин, тобто свідомо нерозв'язних завдань. Рішення завдання при пошуку методом редукції (при пошуку в І/АБО-графі) зводиться до знаходження в І/АБО-графі вирішального графа, визначення якого буде дано нижче. Зауважимо, що метод зведення задач до підзадач є в деякому роді узагальненням підходу з використанням простору станів. Дійсно, перебір в просторі станів можна розглядати як тривіальний випадок зведення задачі завжди до однієї підзадаче.

Графічно для розрізнення диз'юнктивній і кон'юнктивній вершин дуги, що виходять з кон'юнктивної вершини, з'єднуються дужкою при вершині. Приклад графічного представлення розбиття задачі на підзадачі наведено на рис. 1.4. Тут S_0 – початкова задача, для вирішення якої потрібно вирішити підзадачу S_3 або підзадачі S_1 і S_2 . Рішення завдання S_1 зводиться до вирішення або підзадачі S_4 , або підзадачі S_5 . Рішення підзадачі S_3 зводиться до вирішення підзадач S_6 і S_7 . Рішення задач S_2 , S_5 , S_7 передбачається відомим, рішення задач S_4 і S_6 невідомо. У наведеному прикладі завдання S_0 може бути вирішена або шляхом вирішення завдання S_3 , або шляхом вирішення завдань S_1 і S_2 . У зв'язку з тим, що в І/АБО-графі кожна вершина стоїть тільки одного типу (або І, або АБО), то для запису графа, зображеного на рис. 1.4 у вигляді І/АБО-графа, треба ввести додаткову вершину (вершина R_1 на рис.1.5). На рис.1.5 подвійними лініями виділено вирішальний граф завдання S_0 , а кінцеві вершини позначені зачорненими квадратами.

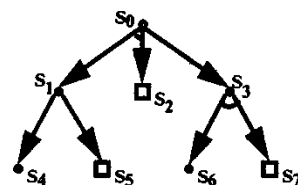


Рисунок 1.4 – Графічне представлення процесу розбиття задачі на підзадачі

Мета процесу пошуку в І/АБО-графі – показати, що початкова вершина розв'язна, тобто для цієї вершини існує вирішальний граф. Визначення можливо розв'язати вершини в І/АБО-графі можна сформулювати рекурсивно наступним чином:

- кінцеві (цільові) вершини можна розв'язати, так як їх рішення відоме по вихідному припущенню;
- вершина АБО можна вирішити тоді і тільки тоді, коли можна вирішити принаймні одна з її дочірніх вершин;
- вершина І можна вирішити тоді і тільки тоді, коли можна вирішити кожна з її дочірніх вершин.

Отже, вирішальний граф визначається як підграф з вирішуваних вершин, який показує, що початкова вершина розв'язна (відповідно до наведеного вище визначенням). На рис.1.5 розв'язні вершини зачорнені, а нерозв'язні залишені білими.

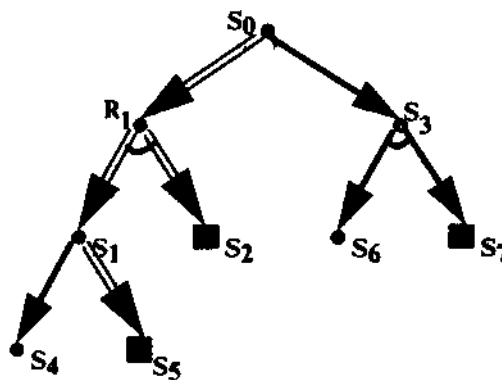


Рисунок 1.5 – Приклад І/АБО-графа

Для графа І/АБО, так само як для пошуку в просторі станів, можна визначити пошук в глибину і пошук в ширину як в прямому, так і в зворотному напрямку. На рис.1.6 наведені приклади пошуку в ширину (рис.1.6, а) і пошуку в глибину (рис.1.6, б). На малюнку вершини пронумеровані в тому порядку, в якому вони розкривалися; кінцеві вершини позначені квадратами,

розв'язні вершини зачорнені, дуги вирішального графа виділені подвійними лініями.

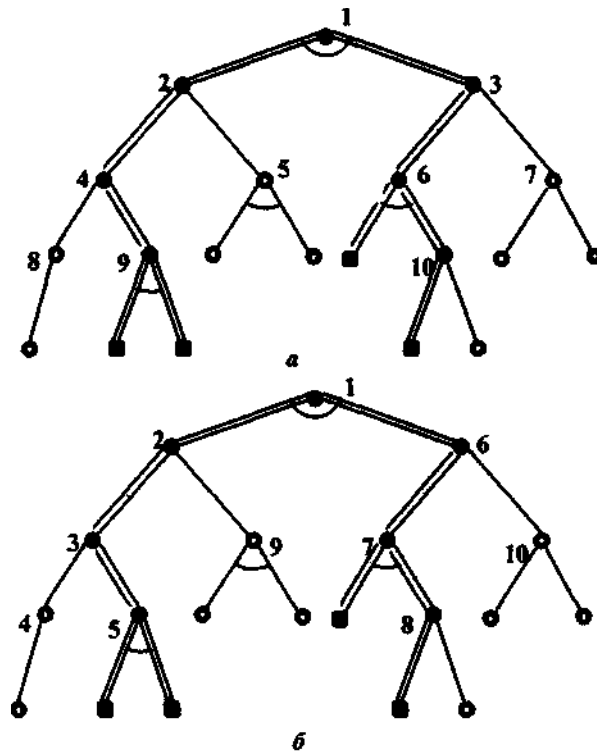


Рисунок 1.6 – Приклад розбиття задач на підзадачі при пошуку в ширину (а) і при пошуку в глибину (б)

Евристичний пошук. Методи пошуку в глибину і ширину називають сліпим пошуком, оскільки в цих методах порядок розкриття вершин зумовлений і ніяк не залежить від розташування цілі. При збільшенні простору пошуку методи сліпого пошуку вимагають надмірних витрат часу і (або) пам'яті. Прагнення скоротити час пошуку призвело до створення евристичних методів пошуку, тобто методів, які використовують деяку інформацію про предметну область для розгляду не тільки простору пошуку, а таких шляхів в ньому, які з найбільшою ймовірністю призводять до мети. Один спосіб скорочення перебору полягає у виборі більш "інформованої" оператора, який не буде так багато вершин, що не відносяться до справи. Інший спосіб полягає у використанні евристичної інформації для визначення на

кожному кроці подальшого направлення перебору. Для цього необхідно ввести міру "перспективності" вершини у вигляді деякої оціночної функції. У деяких випадках вдається ввести таку оціночну функцію, що вона, скорочуючи перебір, не втрачає властивості повноти. Найчастіше ж використовуються евристики, істотно скорочуючи перебір, тягнуть за собою втрату властивості повноти. Як правило, оціночні функції намагаються кількісно оцінити відстань від поточної вершини до кінцевої. З двох вершин при однаковій глибині перспективнішою та, від якої менше відстань до цілі. Для багатьох додатків, зокрема для експертних систем, застосування кількісних оцінок не дозволяє ефективно спрямовувати процес пошуку.

Пошук методом "генерація-перевірка". Процес пошуку може бути сформульований в термінах "генерація-перевірка". Дійсно, простір пошуку (простір станів або I/АБО-граф), як правило, явно не задано. Тому для здійснення процесу пошуку необхідно генерувати чергове можливе рішення (стан або підзадачу) і перевірити, чи не є воно результуючим. Розумно вимагати, щоб генератор задовольняв вимогам повноти і ненадлишкових. Кажуть, що генератор є повним, якщо він забезпечує генерацію всіх можливих рішень. Генератор є ненадлишковим, якщо він генерує кожне рішення тільки один раз. Забезпечення якості ненадлишкових є важливим, але важко виконуваним, так як відповідно до цієї вимоги не допускається генерація не тільки тотожних, але і синонімічних рішень. Наприклад, якщо завдання генератора – синтезувати всі фрази російської мови, то досить важко (якщо взагалі можливо) зробити такий генератор ненадлишковим.

При генерації поточного можливого рішення (стану або підзадачі) виникає проблема розподілу знань між генератором і пристроєм перевірки. При сліпому і евристичному пошуку генератор має мінімальні знання про область, достатні для генерації всіх можливих рішень (станів або підзадач), а пристрій перевірки визначає, чи не є чергове рішення цільовим. В принципі деякі знання можна перенести з пристрою перевірки в генератор, щоб він не

генерував рішення, які свідомо не можуть привести до успіху. Збільшення знань генератора про область призводить до скорочення простору, в якому здійснюється пошук. Однак при цьому підвищуються витрати на генерацію кожного чергового стану (підзадачі).

Можна виділити важливу форму методу "генерація-перевірка", яка називається "ієрархічна генерація-перевірка". У цьому випадку на верхньому рівні генератор виробляє не повне, а частково певне рішення (будемо для стислості називати такі рішення частковими). Кожне часткове рішення описує не всі стан, а тільки його певну частину, визначаючи таким чином клас можливих станів. Ідея полягає в тому, що пристрій перевірки може вже з вигляду часткового вирішення визначити, що воно (а отже, і всі повні рішення, які можуть бути отримані з нього) не веде до успіху. Якщо ж перевірка не відкидає часткове рішення, то на наступному рівні генератор продовжує виробляти з даного часткового вирішення все повні рішення, а пристрій перевірки визначає, чи є вони цільовими.

1.2.4.2 Пошук в ієрархії просторів

Методи пошуку в одному просторі не дозволяють вирішувати складні завдання, тому що зі збільшенням розміру простору час пошуку експоненціально зростає. При великому розмірі простору пошуку можна спробувати розбити загальний простір на підпростору і здійснювати пошук спочатку в них. Можна сказати, що в даному випадку простір пошуку представлено ієрархією просторів. Важливість ієрархічних методів при роботі з великими просторами зрозуміла давно. Ще в 1963 р М.Минская писала, що введення "острівців планування" зменшує час пошуку по експоненті: "У графі з 10 ребрами, що виходять із кожної вершини, 20-кроковий пошук може зажадати 1020 спроб, що нереально реалізувати, в той час як введення чотирьох лем або послідовних подцілей може зменшити пошук до 5×10^4 спроб, які машина мо-

же виконати. Тому має сенс докласти навіть величезні зусилля, щоб виявити такі "острівці" при вирішенні складних завдань" [5]¹⁾. Ідею М.Минская про ієрархії просторів можна розвинути, допустивши в ієрархії не тільки конкретні, але і абстрактні простори, тобто простору які мають опис тільки найбільш важливих сутностей. Як класичний приклад використання абстрактних просторів можна привести задачу визначення найкоротшого шляху на карті. Нехай потрібно переїхати з центру міста А в центр міста В. Якщо здійснювати пошук необхідного шляху на детальній карті, що містить всі вулиці у всіх містах, які зустрілися по дорозі, то завдання може стати практично нерозв'язною. При визначенні шляху з міста А в місто В доцільно спланувати маршрут по великомасштабній карті (тобто здійснити пошук в абстрактному просторі), а потім по детальній карті спланувати виїзд з міста А і в'їзд в місто В. В даному розділі будуть розглянуті методи, використовують загальну ідею ієрархії просторів, але відрізняються природою просторів.

Методи пошуку рішення в ієрархічних просторах зазвичай діляться на пошук в факторізованому просторі, пошук у фіксованому і змінюється безлічі просторів [6]²⁾.

Пошук в факторізованому просторі. У багатьох додатках потрібно знайти всі рішення. Прикладами таких областей є інтерпретація даних, постановка діагнозу і ін. Дійсно, в разі постановки діагнозу нас цікавлять всі, а не деякі хвороби пацієнта. Однак простір пошуку в практичних додатках буває настільки велике, що не дозволяє застосувати сліпі методи пошуку. Застосування евристичних методів в даному випадку, як правило, також виключено, так як вони не забезпечують отримання всіх можливих рішень. Якщо простір пошуку вдається факторизувати, то пошук навіть в дуже велико-

¹⁾ [5] Методы представления знаний: Метод. указ. / Сост. И. Л. Коробова. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2003.

²⁾ [6] А.Частиков, Т.Гаврилова, Д.Белов. Разработка экспертных систем. Среда CLIPS. Спб.: БХВ-Петербург. 2005. 608 с.

му просторі можна організувати ефективно. Простір називається факторизовано, якщо воно розбивається на непересічні підпростори (класи) частковими (неповними) рішеннями. Причому по виду часткового вирішення можна визначити, що воно не приведе до успіху, тобто що всі повні рішення, утворені з нього, не приведуть до цільових рішень. Пошук в факторизованому просторі здійснюється на основі методу "ієрархічна генерація-перевірка" (див. Вище). Генератор виробляє поточний часткове рішення, потім перевіряється, чи може це рішення привести до успіху. Якщо поточний часткове рішення відкидається, то з розгляду без генерації та перевірки усуваються всі повні рішення цього класу. Якщо поточний часткове рішення не відкидається, то генератор виробляє на його основі все повні рішення, а пристрій перевірки визначає, чи є ці рішення цільовими.

Пошук в фіксованому безлічі просторів. Застосування методу факторизації простору обмежена тим, що для ряду областей не вдається по частковому вирішенню зробити висновок про його непридатність. Прикладами таких областей є завдання планування і конструювання. Дійсно, як правило, за фрагментом плану або конструкції не можна сказати, що цей фрагмент не може бути частиною повного рішення. У цих випадках можуть бути застосовані методи пошуку, що використовують ідею абстрактного простору. Методи розрізняються припущеннями про природу цього простору. Абстракція повинна підкреслити важливі особливості розглянутої задачі, дозволити розбити задачу на простіші підзадачі і визначити послідовність підзадач (план рішення), що приводить до вирішення основного завдання. У найпростішому випадку простір пошуку розбивається на фіксовану послідовність підзадач (підпросторів), за допомогою яких можна вирішити будь-яку вихідну задачу.

Подібний метод пошуку використаний, наприклад, в експертній системі R1. На підставі замовлення покупця на необхідну йому кон-фігурації системи VAX система R1 визначає, чи не містить замовлення несумісних компонентів, виявляє відсутні компоненти і будує діаграми, що зображують про-

сторів взаємозв'язки компонентів VAX. Система R1 розбиває загальну задачу на шість підзадач Порядок, в якому викликаються ці завдання, залежить від замовленої конфігурації. Дії, що виконуються кожної підзадачі, залежать від комбінації замовлених компонентів і способу їх взаємозв'язку. В системі кожної підзадаче відповідає свій набір правил, т. е. Кожна підзадача вирішується в своєму підпросторі. Пошук в R1 здійснюється за допомогою безповоротної стратегії пошуку, т. е. Без використання процедури бектрекінга. Цей механізм відновлює стан, безпосередньо передую поточному, та потім вибирає чергову альтернативу.

Пошук в змінюваному безлічі ієрархічних просторів. У ряді програм не вдається все розв'язувати завдання звести до фіксованого набору підзадач. Прикладами таких додатків є завдання планування переміщень в просторі. План виконання завдання в даному випадку повинен мати змінну структуру і не може бути зведений до фіксованого набору підзадач. Для вирішення подібних завдань може бути використаний метод спадного уточнення (top-down refinement). Для того щоб спростити процес вирішення деякої задачі в складному просторі, доцільно отримати узагальнене простір (простір меншої розмірності) і спробувати отримати рішення в цьому просторі. Зазначений прийом можна повторювати багаторазово. При цьому повний процес вирішення завдання можна представити як спадний рух в ієрархії просторів від найбільш абстрактного до конкретного, в якому виходить остаточне рішення. Суттєвою характеристикою такого процесу є пошук рішення задачі в абстрактному просторі, перетворення цього рішення в рішення нижчого рівня і т. д. Причому на кожному рівні виробляється остаточне рішення і тільки потім здійснюється перехід на наступний, більш конкретний рівень. У середині кожного рівня підзадачі розглядаються як незалежні, що створює часткове впорядкування абстрактних станів. Формування більш абстрактного простору здійснюється шляхом ігнорування частини описів менш абстрактного простору (на першому кроці – конкретного простору). Ігнорування описів здійс-

нюється на основі ранжирування описів за ступенем важливості. Часто ранжування здійснюється на основі врахування ступеня незмінності фактів (найбільш абстрактні ті опису, які не можуть змінюватися). При цьому абстрактні простори, з одного боку, повинні для спрощення рішення задачі забезпечувати значне спрощення вихідного простору, а з іншого боку, повинні бути подібні один одному і конкретному простору, щоб процес низхідного перенесення рішення з більш абстрактних просторів в менш абстрактні не вимагав великих обчислювальних витрат.

Використання обмежень при пошуку рішення. Обмеження можна розглядати як спосіб часткового опису деяких сутностей. Обмеження можуть бути задані як в числовий, так і в символічній формі. Прикладом завдання обмеження в числовий формі є будь-яка формула, яка накладає обмеження на співвідношення входять до формулу змінних. Прикладом завдання обмеження в символічній формі є модель управління будь-якого дієслова, що задає семантичні категорії.

Обмеження можуть бути використані для подання цілей в методах пошуку в ієрархічних просторах. Наприклад, при конструюванні топології електричної схеми інвертор на верхньому рівні абстракції може бути описаний як дискретне комутаційне пристрій з одним входом і декількома виходами. На цьому рівні опису ігнорується така інформація, як геометрія інвертора, джерело живлення і земля. На нижчому рівні абстракції інвертор може бути описаний з урахуванням його геометрії. На цьому ж рівні можуть бути вказані два обмеження, що визначають, яка частина інвертора повинна бути пов'язана з харчуванням, а яка – з землею. Використання обмежень дозволяє відкласти вирішення питання про те, як саме виглядає маршрут, який з'єднує частини інвертора з харчуванням і землею. Ці обмеження можуть бути враховані при конструюванні інших частин схеми. Якщо обмеження не можуть бути враховані, то побудована схема інвертора повинна бути переглянута.

Використання обмежень замість отримання конкретного рішення дає можливість відкласти ухвалення рішення. Відкладати розв'язання може бути викликано рядом причин: немає достатньої інформації для того, щоб визначити місцезнаходження харчування і землі; інші міркування при конструюванні схеми можуть виявитися більш важливими, ніж розглянутий інвертор.

Друга із зазначених причин ілюструє важливий феномен процесу вирішення завдань – взаємодія підзадач. Якщо для вирішення підзадач потрібно їх незначна координація, то кажуть, що підзадачі майже незалежні. Зазвичай такі підзадачі мають більше одного рішення, якщо при отриманні рішення враховуються тільки локальні обмеження, тобто обмеження, що впливають із самої підзадачі, а не з інших підзадач, від яких дана підзадача майже незалежна. Якщо отримувати рішення таких підзадач як незалежних, то часто при об'єднанні підзадач виникають невідповідності. Введення обмежень дозволяє уникнути передчасного отримання рішень, що враховують не всі, а тільки локальні обмеження. Використання обмежень дозволяє застосовувати принцип найменших звершень. Цей принцип дозволяє перемикає увагу з одного підзадачі на іншу і уникати передчасних рішень.

Принцип найменших звершень. Основний недолік методу спадного уточнення полягає в тому, що він не має зворотного зв'язку. Метод передбачає, що одні й ті ж рішення повинні прийматися в однакових ситуаціях при вирішенні будь-якої задачі. При вирішенні ряду задач деталізація рішення, отриманого на абстрактному рівні, виявляється неможливою, тому що при побудові абстрактного плану були опущені деталі, що перешкоджають його уточнення, тобто потрібен перегляд абстрактного плану (рішення). У подібних ситуаціях доцільно застосування принципу найменших звершень. Відповідно до цього принципу рішення не будується відразу до кінця на верхніх рівнях абстракції. Часткове вирішення деталізується поступово, у міру появи інформації, яка б підтверджувала можливість вирішення і змушує прийняти

рішення. Міркування, заснований на використанні принципу найменших звершень, вимагає, щоб система була в стані зробити наступні дії:

- визначити, коли накопичилося достатньо інформації для прийняття рішення;
- припинити роботу над деякою підзадачею, коли для вирішення немає достатньої інформації;
- переходити з однієї підзадачі на іншу, відновлюючи виконання призупиненої підзадачі при появі якої бракує інформації;
- об'єднувати інформацію, отриману різними підзадачами.

Принцип найменших звершень вперше був використаний експертною системою MOLGEN, призначеної для планування експериментів з молекулярної генетики. MOLGEN представляє взаємодію між підзадачами у вигляді обмежень. Для міркувань про обмеження використовуються оператори метаязика (на противагу операторам предметної області). Система чергує використання принципу найменших звершень і використання евристичних стратегій. При використанні принципу найменших звершень вибір здійснюється тільки тоді, коли обмеження визначають досить вузький набір альтернатив. В іншому випадку процес вирішення завдання призупиняється (завдання переходить в стан "очікування обмежень"), і здійснюється перехід до іншої підзадачі.

Отже, принцип найменших звершень координує процес пошуку рішення з наявністю необхідної інформації і відповідно до доступною інформацією переміщує фокус активності за рішенням завдання від однієї підзадачі до іншої. Даний підхід непридатний, коли існує багато можливостей, але немає надійних підстав для вибору рішення. У цих випадках необхідно використовувати деякі форми правдоподібних міркувань або переходити на використання іншої моделі.

1.2.4.3 Пошук в альтернативних просторах

Розглянуті вище методи пошуку виходять з мовчазною передумови, що знання про предметну область і дані про розв'язуваної задачі є точними і повними і для них справедливо наступне:

- всі твердження, що описують стан, є істинними;
- застосування оператора до деякого станом формує деякий новий стан, опис якого складається тільки з істинних фактів.

Однак при вирішенні будь-яких практичних завдань і особливо при розв'язанні неформалізованих задач поширена зворотна ситуація. Експертові доводиться працювати в умовах неповноти і неточності знань (даних) і, як правило, в умовах дефіциту часу. Коли експерт вирішує задачу, він використовує методи, що відрізняються від формальних математичних міркувань. У математичних міркуваннях кожне висновок повинен строго слідувати з попередньої інформації. На противагу цьому в правдоподібних міркуваннях, заснованих на здоровому глузді, висновки повинні базуватися на частковій інформації. У цьому випадку експерт робить правдоподібні припущення, які він не може довести; тим самим питання про їх істинності залишається відкритим. Всі твердження, отримані на основі цих правдоподібних припущень, також не можуть бути доведені.

Один із способів обґрунтування припущень полягає в тому, щоб розглядати їх як можливі значення, що задаються за умовчанням. Наприклад, висловивши припущення, що зараз 14 год (подивившись на годинник), ми мовчазно припускаємо, що годинник йдуть і йдуть правильно. Зазвичай людина знає, що деякі припущення вірні тільки за певних умов. Якщо інформація, яка вказує на порушення цих умов; відсутня, то припущення може бути висловлене. Інше обґрунтування припущень базується на розгляді міркування як процесу з обмеженими ресурсами. Так, можна вважати, що припущення (X) має місце, якщо, використовуючи обмежені ресурси, не можна довести

істинність протилежного твердження. Припущення і висновки, зроблені на його основі, повинні усуватися, якщо з'явилася інформація, що показує хибність цього припущення. Цей аспект у побудові умовиводів з використанням припущень називається немонотонність.

Будь-яка формальна система є монотонною, тобто якщо A , B і C є деякі висловлювання, такі, що якщо B виводиться з A , то B виводитиметься і з C . Система немонотонна, якщо B виводиться з A [7]¹⁾. Немонотонні міркування особливо важливі при вирішенні задач планування і конструювання. У цих завданнях простір пошуку іноді дуже велике, і немає можливості передбачити всі наслідки зробленого вибору. Так, наприклад, конструктор знає, чого він хоче, але не знає, як це зробити. Тому при конструюванні припущення виступають у вигляді пробних рішень, наслідки яких потім аналізуються з точки зору їх придатності (непридатності). Якщо наслідки не суперечать тому, що хотів конструктор, то процес конструювання триває далі, можливо, з висуненням нових припущень. В іншому випадку необхідно усунути всі наслідки і зробити альтернативне припущення і т.д.

Отже, для того щоб система могла робити висновки, засновані на здоровому глузді, при роботі з неповними (неточними) даними і знаннями, вона повинна бути здатна робити припущення, а при отриманні нової інформації, яка б показала помилковість припущень, відмовлятися як від зроблених припущень, так і від висновків, отриманих на основі цих припущень. Думка системи про те, які факти мають місце, змінюється в ході міркування, тобто можна говорити про ревізію думок. Таким чином, навіть якщо розглядати проблемну область як статичну, неповнота (і неточність) знань і даних тягне за собою розгляд цієї області при різних (і навіть протилежних) припущеннях, що, в свою чергу, призводить до подання області у вигляді альтернативних

¹⁾ [7] В. А. Макушкин, К. А. Щербицкий. Экспертная система для контроля и диагностики цифромимологових устройств. М.: МДНТП. 1991. 598 с.

просторів, відповідних різним, можливо, суперечливим і (або) взаємодоповнюючих припущеннями і думок.

Думка В можна представляти у вигляді відношення, що складається з суб'єкта думки а (індивіда або системи, що мають розглядається думка), об'єкта думки р (те, про що суб'єкт має думка) і обґрунтування думки г (причини, за якими суб'єкт має дана думка), або в формальному вигляді – В (а, р, г) (а думає, що р, так як г).

Будемо говорити, що безліч думок, властивих деякого індивіду (системі), становить його систему думок. Ґрунтуючись на деякій системі думок, можна утворити простір пошуку, призначене для вирішення будь-яких завдань. В ході міркувань людина (система) може змінювати свої думки, утворюючи різні системи думок. Сукупність думок, якій система дотримується в поточний момент, будемо називати активною системою думок. Кожній з систем думок відповідає свій простір пошуку, а всі разом вони утворюють альтернативні простору. На рис. 1.7 зображені три альтернативних простору Р, Q і R. Простір Р утворено вихідними посилками С1, В1, припущеннями А1 і висновками, зробленими на їх основі; простір Q утворено з С1, В1, припущень А2, D1 і висновків, зроблених на їх основі; простір R утворено з С1, В1, припущень А2, D2 і висновків, зроблених на їх основі.

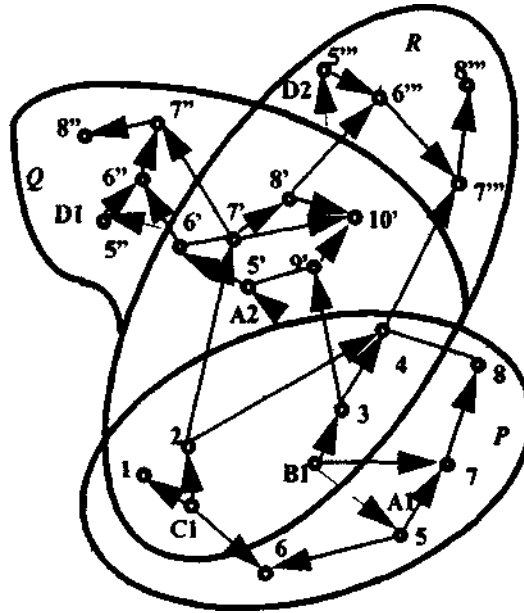


Рисунок 1.7 – Приклад альтернативних просторів

Для того щоб змінити думку, система повинна бути здатна міркувати про залежності, що існують в активній системі думок. Нові думки можуть бути наслідком нової інформації, отриманої ззовні, або виведеної. Залежності в системі думок повинні містити відомості про думки, правила виведення і обґрунтуваннях (підтримках) думок. Найпростішим видом обґрунтування (justification) може бути інформація про те, на яких думках ґрунтуються дані. Наприклад, припустимо, що система має наступні твердження:

- Думка 1: R бере участь в нараді в середу о 13 год.
- Думка 2: якщо R присутній на нараді, то він зайнятий.
- Правило 3: Modus ponens.
- З цієї інформації можна зробити висновок:
- Думка 4: R зайнятий в середу о 13 год.

Обґрунтуванням для цієї думки може бути такий запис:

- Обґрунтування 1:
- Підтримка для: Думка 4.

Правило виводу: Правило 3.

Посилки: (думка 1, думка 2).

Обґрунтування може використовуватися для підтримки або зміни поточної системи думок (ревізії думок).

Зупинимось на особливостях механізму пошуку в альтернативних просторах. У розглянутих раніше методах пошук в деякому напрямку переривався при досягненні цільового або термінального стану. Якщо досягалося цільове стан, то або робота завершувалася (при пошуку одного рішення), або тривав пошук наступного рішення. При досягненні термінального стану необхідно повернутися в якийсь попередній стан простору і продовжити пошук в новому напрямку. Зазвичай при реалізації пошуку використовувався механізм бектрекінга (backtracking). Бектрекінг працює за принципом "last-in, first-out" ("останнім увійшов, першим вийшов"), тобто спочатку усувається останній розглядається стан, для реалізації якого може бути застосований механізм стека.

Застосування механізму бектрекінга при пошуку в альтернативних світах буде приводити до зайвої неефективності, так як всі невдачі, що виникли при пошуку в одному напрямку, не запам'ятовуються при переході до пошуку в іншому напрямку, Та ж сама причина невдачі може заново виявлятися і на новому напрямку. Так, наприклад, якщо ми взяли стакан правою рукою і через те, що він гарячий, відпустили його, то недоцільно тут же намагатися брати склянку лівою рукою (що, образно кажучи, буде робити бектрекінг).

Таким чином, механізм традиційного бектрекінга відкидає занадто багато інформації. Здійснювати повернення доцільно не до стану, безпосередньо попереднього даному, а до того стану, який є причиною виникнення невдачі. У використовуваних нами термінах причиною невдач є припущення, тобто недоведені твердження. Тому при виявленні невдачі необхідно повертатися в стан, де це припущення було зроблено, і відчувати інше припущен-

ня. Так, наприклад, якщо при отриманні затвердження 8 (див. рис. 1.7) система встановила наявність невдачі (протиріччя), то повернення потрібно робити не до попереднього кроку (до утвердження 7), а до кроку 5, на якому було зроблено припущення A1, і замінювати A1 на деяке нове припущення A2. Таким чином здійснюється перехід з простору P в новий простір (R або Q). Для виконання описаних способів пошуку можна використовувати інформацію про залежність, представлену в тому чи іншому вигляді. З цієї причини даний метод пошуку називають пошуком, які направляються залежністю.

1.2.4.4 Пошук з використанням декількох моделей

Всі методи пошуку, розглянуті до сих пір, використовували при поданні проблемної області якусь одну модель, тобто розглядали область з якоїсь однієї точки зору. При вирішенні складних завдань в умовах обмежених ресурсів використання декількох моделей може значно підвищити потужність системи. Об'єднання в одній системі кількох моделей дає можливість подолати такі труднощі. По-перше, перехід з однієї моделі на іншу дозволяє обходити тупики, що виникають при пошуку в процесі поширення обмежень. По-друге, використання декількох моделей дозволяє в ряді випадків зменшити ймовірність втрати хорошого рішення (наслідок неповного пошуку, викликаного обмеженістю ресурсів) за рахунок конструювання повного рішення з обмеженого числа часткових кандидатів шляхом їх розширення і комбінації. По-третє, наявність декількох моделей дозволяє системі справлятися з неточністю (помилковістю) даних. Слід зазначити, що використання декількох моделей вимагає додаткових знань про те, як створювати і об'єднувати різні точки зору.

Розглянемо метод використання декількох моделей, вперше застосований в експертній системі SYN. SYN є програмою для синтезу електричних

схем. Система визначає значення компонент схеми, форму схеми і деякі особливості її роботи. Новизна SYN полягає у використанні декількох моделей, тобто SYN може розглядати схему з різних точок зору, що відповідає ідеї еквівалентних електричних схем. Наприклад, SYN може розглядати дільник напруги як що складається з двох послідовно з'єднаних опорів (R_1 ; R_2), а може розглядати ці опору як одне ($R = R_1 + R_2$). Так, при аналізі подільника напруги SYN використовує другу точку зору для обчислення струму, що проходить через дільник ($I = U/R$). Потім для обчислення напруги в середній точці дільника ($U_1 = I \times R_2$, $U_2 = I \times R_1$) SYN повертається до першої точки зору. Ідея почергового використання еквівалентних уявлень електричних схем дозволяє подолати тупики, що виникають при поширенні обмежень. Потужність використання декількох (в описаному випадку двох) моделей полягає в тому, що цей підхід забезпечує додаткові шляхи по поширенню пошуку (розрахунку). Використовуючи ідею еквівалентних електричних схем, система SYN здатна досліджувати складні схеми без трудомістких алгебраїчних обчислень.

1.2.4.5 Вибір методу рішення задач

Вибір методу розв'язання задачі залежить перш за все від складності завдання, яка визначається особливостями проблемної області та вимогами, що пред'являються користувачем до вирішення завдання. Прості завдання характеризуються невеликою розмірністю простору пошуку, точністю і повнотою даних, статичністю області, можливістю адекватного опису області за допомогою однієї моделі. На практиці зустрічається мало додатків, які відповідають перерахованим вимогам.

Складні завдання характеризуються тим, що значення хоча б одного з перерахованих параметрів виявляється в них більш складним. Для подолання труднощів, викликаних великим простором пошуку, використовуються ме-

тоди, засновані на введенні ієрархій просторів (конкретних, абстрактних і метапространств). Найпростіший з цих методів ґрунтується на факторизуємому простору рішень, що дозволяє виробляти раннє відсікання. Метод забезпечує отримання всіх рішень. Якщо простір пошуку не вдається факторизувати, але при цьому не потрібно отримувати всі рішення або вибирати краще, то можуть бути застосовані методи, які використовують ієрархію однорідних абстрактних просторів. Якщо простір пошуку таке, що будь-яке завдання може бути зведена до відомої заздалегідь послідовності підзадач, то використовується фіксований абстрактне простір.

Ефективність цього методу визначається можливістю використовувати безповоротну стратегію. У тих випадках, коли рішення задачі не може бути отримано без механізму бектрекінга, застосовуються більш складні методи. Метод спадного уточнення застосовуємо в тому випадку, коли всі завдання не можуть бути зведені до фіксованого набору підзадач, однак існує фіксована впорядкованість понять області та фіксований частковий порядок між підзадачами. У разі, якщо підзадачі взаємозалежні, тобто для вирішення деякої підзадачі може вимагатися інформація, одержувана Інший підзадачею, і підзадачі не можуть бути впорядковані, доцільно застосовувати принцип найменших звершень. Цей підхід дозволяє припиняти рішення підзадачі, для якої бракує інформації, переходити до вирішення іншої підзадачі і повертатися до вихідної задачі, коли відсутня інформація стане доступною.

Слід зазначити, що використання даного підходу вимагає більш різноманітних знань про рішення задачі, ніж в попередніх випадках. При використанні різноманітних знань про процес вирішення стає доцільним об'єднувати принцип найменших звершень з методами, які використовують метазнання. Принцип найменших звершень може призводити до утворення тупиків в процесі виконання завдання, що перешкоджає використанню цього принципу в чистому вигляді. Для подолання тупиків використовують припущення або застосовують метод декількох моделей. Для подолання

труднощів, викликаних неповнотою і (або) неточністю даних (знань), використовують імовірнісні, розмиті і точні методи. Всі ці методи ґрунтуються на ідеї збільшення надійності шляхом комбінування фактів і використання метазнаній про можливості комбінування фактів. Неточні підходи (ймовірні, псевдовероятностні, розмиті) використовують різноманітні апріорні оцінки, умовні ймовірності і розмиті безлічі; точні підходи використовують припущення і ревізію думок при немонотонності міркуваннях.

Для подолання неадекватності моделі проблемної області використовуються методи, орієнтовані на використання декількох моделей. Ці методи дозволяють об'єднати можливості різних моделей, що описують проблемну область з різних точок зору. Крім того, використання декількох моделей дозволяє зменшити ймовірність втрати хорошого рішення, незважаючи на неповноту пошуку, викликану обмеженістю обчислювальних ресурсів.

1.3 Постановка задачі роботи

У магістерській роботі необхідно спроектувати та реалізувати систему для оцінки нерухомості.

Необхідно розробити базу знань, використовуючи продукційну модель представлення знань для оцінки нерухомості.

База знань повинна включати в себе набір правил, що необхідні для оцінки нерухомості. Ці правила – це висновок експертів стосовно вартості нерухомості.

У роботі вирішено створювати систему для оцінки квартир у місті Одеса. Вартість квартир розбити на три категорії: бюджетний варіант, середній та елітне житло. Для кожної категорії створити правила.

2 МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ

2.1 Побудова моделей представлення знань

Опис проблемної інформації починається з подання всіх об'єктів, класів, об'єднаних об'єктів і понять, їх визначають.

Основні моделі подання знань: фрейми, семантичні мережі, логічні моделі.

Розглянемо основні моделі подання знань.

Фрейм – структура даних для представлення стереотипної ситуації. Фрейм визначається маєтком і сукупністю слотів. Слоти можуть мати різні значення і можуть залишатися незаповненими для конкретної ситуації. Слоти визначають об'єкти, класи і затвердження. Значним слота може бути ім'я іншого фрейму. Таким чином, фрейм може визначати вихідні стану, наприклад, стан А і перехід в новий стан В на основі вихідного опису за допомогою фрейму А в новий стан, що описується фреймом В. Слоти фрейма, крім декларативного і процедурного призначення, можуть визначатися примусово наступним чином:

- слоти, які призводять до виконання дій в залежності від отриманих фреймом значень при зіставленні, називаються слотами -слугами;
- слоти, які активізуються незалежно від значень інших слотів, називаються слотами-дияволами.

При використанні фреймового уявлення передбачається, що значення деяких слотів фрейма задані, а значення інших слотів фрейму заповнюються відповідно до ситуації. Використання фреймів можна порівняти з використанням процедур в традиційних мовах програмування. Опис процедури в програмі є (відповідає) фрейм-прототип. Дія, що виконується процедурою, відповідає поняттю фрейму-реалізація.

Використання фреймів передбачає реалізацію в інформаційній системі на основі об'єктно-орієнтованого підходу, де класи об'єднують в сукупності

однотипних об'єктів. Об'єкти мають властивості, характерні для даного класу. При побудові ієрархії класів встановлюється відповідність між класами і фреймами і характеристиками об'єктів зі слотами. Споріднені властивості об'єктів успадковуються, тобто об'єкт, що знаходиться на нижньому рівні класифікації, має всі властивості, притаманні даному класу об'єктів. Представимо фреймову модель (рис.2.1).



Рисунок 2.1 – Фреймова модель оцінки об'єкту

Семантична мережа. У традиційному розумінні семантична мережа – це конструкція двох основних компонентів-вузлів і дуг. Вузли моделюють поняття предметної області, дуги – відносини між парою понять. За допомогою семантичної мережі можна промоделювати досить складні відносини між різного роду поняттями, що описують досить багато станів проблемної області.

Відносини між об'єктами в семантичній мережі можна встановити як «все з усіма», тобто кожен вузол семантичної мережі може мати відношення з кожним її вузлом.

Семантичну мережу можна представити у вигляді бінарного предиката, або у вигляді відношення з двома аргументами, в загальному вигляді. Уявити цю інформацію можна в списковій структурі. Крім того, опис семантичної

мережі можна представити у вигляді n-аргументної відносини між поняттями. За умови, що ім'я відносини матиме одне і те ж значення (рис.2.2).

При реалізації семантичної мережі в базі даних інформація описується в вигляді трійок: об'єкт, атрибут, значення (те, що притаманне БД).

Недоліки використання семантичної мережі:

- важко реалізується властивість наслідування між поняттями;
- складність в реалізації тимчасових відносин між поняттями.

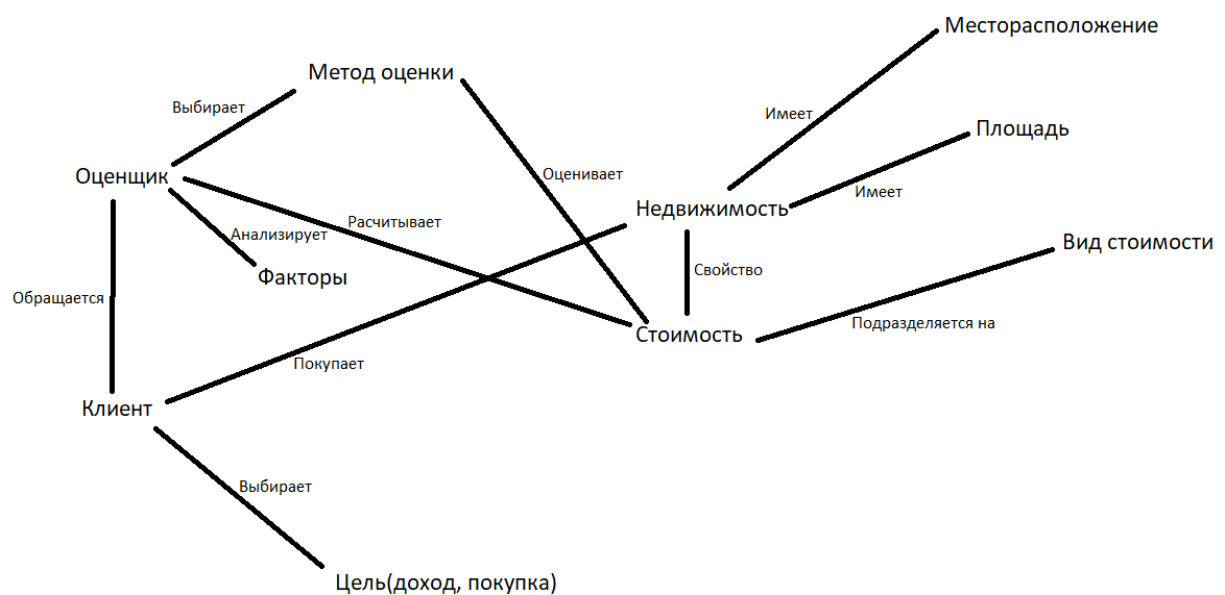


Рисунок 2.2 – Семантична модель оцінки об'єкта

Семантичні мережі використовуються в основному для подання декларативною інформації.

Логічні моделі. Розглянемо основні компоненти даної моделі.

Обробка фактів. Факт є основною одиницею моделі подання знань. Факт – це відношення, яке представляє об'єкт і сукупність властивостей об'єкта або процедур, які визначаються аргументами.

Ставлення – це предикат. Аргументи в предиката задаються константами або змінними.

Приклад: ϵ (Петро, X)
живе ($У$, Новгород, T)

де X , $У$, T – змінні, які визначають інтерпретацію факту в утвердженні.

Аналогом затвердження в численні предикатів ϵ поняття «висловлювання».

Висловлення – це елементарне твердження для використання в складній пропозиції.

Правила формування предикатів:

- кожен предикат має ім'я;
- кожен предикат обов'язково має аргументи;
- кількість аргументів предиката не обмежена;
- послідовність аргументів в завданні предиката визначається послідовністю їх інтерпретацій в заданій проблемній області;
- послідовність заздалегідь відома програмісту і не змінюється протягом усієї програми.

Правила. Найпростішим видом тверджень в моделі подання знань ϵ правила.

У загальному сенсі, правила – це складні умовиводи або висловлювання, які формуються з простих за допомогою відповідних зв'язок $\&$ (and), \cup (or), \sim (not).

Помилкові правила використовують для затвердження істинності на основі фактів, заданих у вихідних даних або на основі інтерпретації значень змінних при доказі. Мірою істинності змінних, використовуваних при доказі правила або складного висловлювання ϵ квантори.

Зауваження: між квантора не припустимо використання зв'язок. Зв'язки використовуються тільки між висловлюваннями.

Утвердження. Мовою логіки предикатів твердження можуть бути представлені:

- правилами, використовуваними для доказу;
- правилами виведення.

Складемо продукційні правила для системи оцінки об'єктів нерухомості (рис.2.3-2.5).

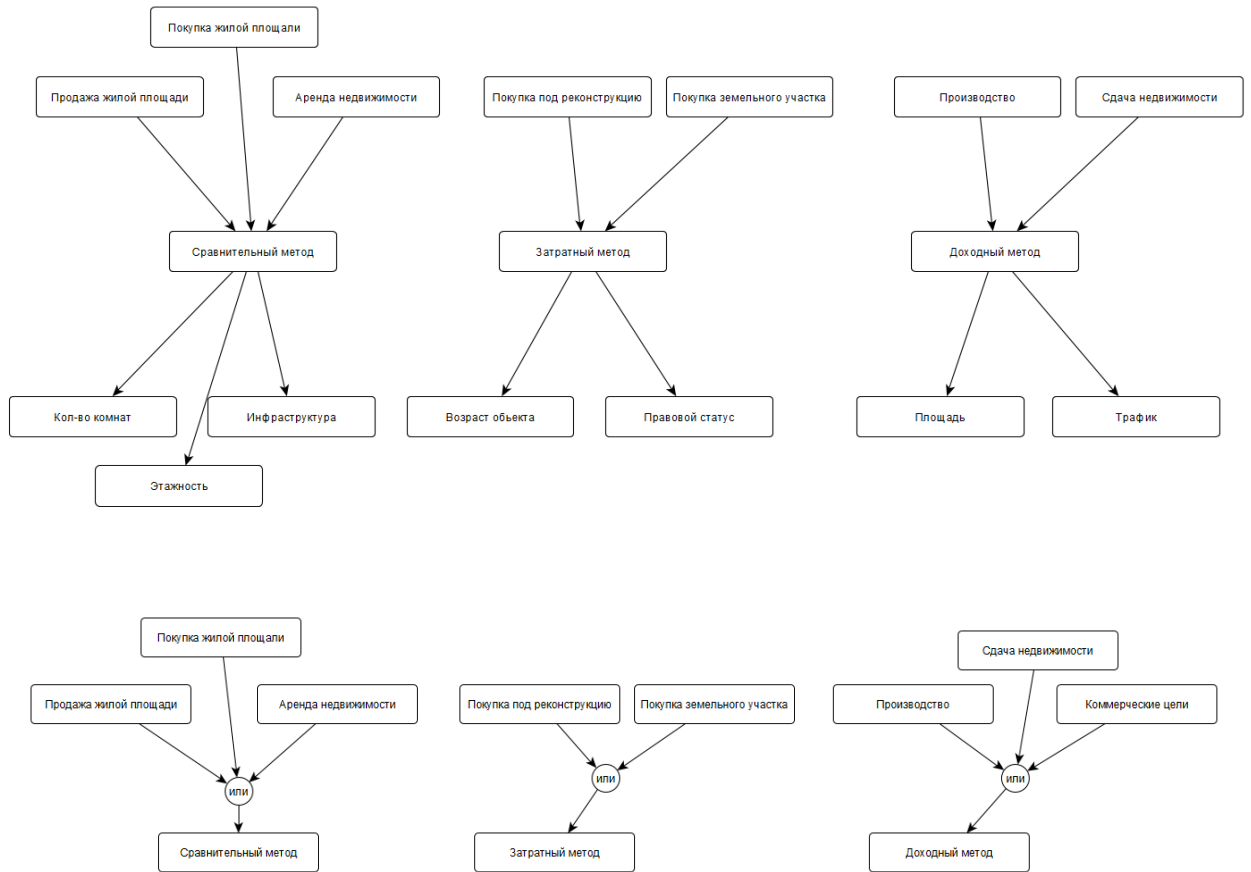


Рисунок 2.3 – Продукційна модель для вибору методу оцінки нерухомості

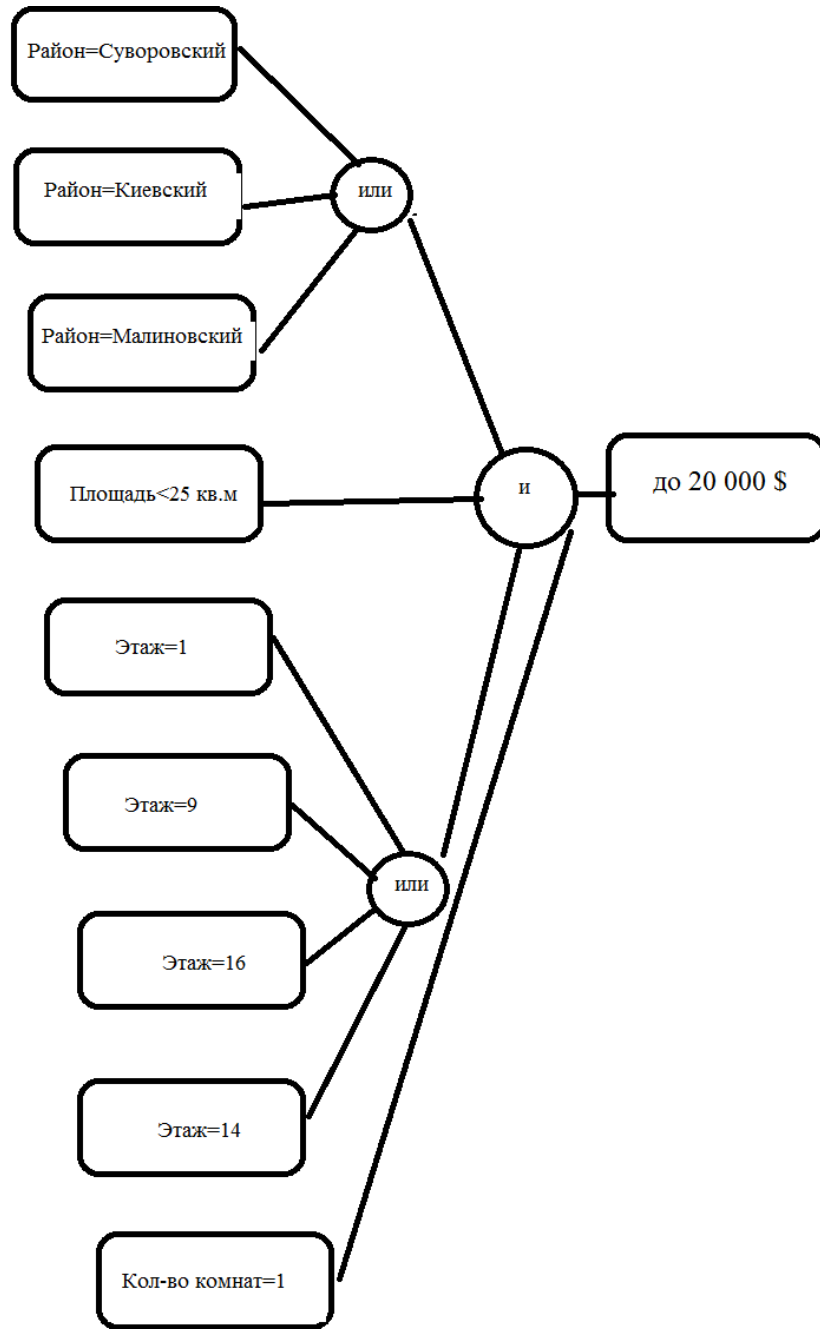


Рисунок 2.4 – Продукційна модель для вибору бюджетної цінової категорії об'єкта

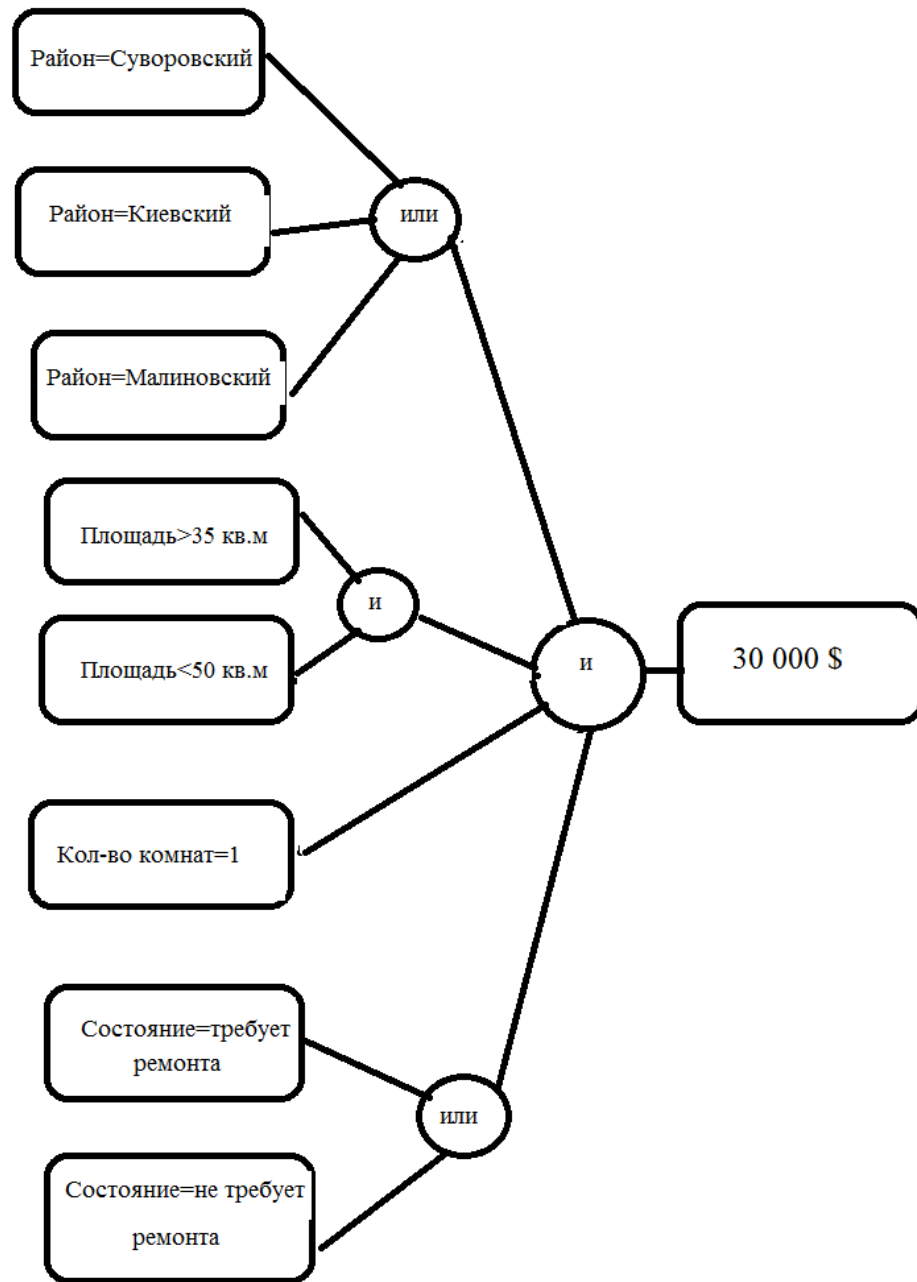


Рисунок 2.5 – Продукційна модель для вибору середньої цінової категорії об'єкта

3 АНАЛІЗ ПО ДЛЯ РЕАЛІЗАЦІЇ БАЗИ ЗНАНЬ ЕКСПЕРТНОЇ СИСТЕМИ

3.1 Програмне середовище Visual Prolog

Visual Prolog - продукт датської фірми Prolog Development Center. Раніше поширювався під назвою Turbo Prolog (Borland) і PDC Prolog (рис.3.1).

Prolog Development Center витратив більше трьох років на розробку системи Visual Prolog з поетапним бета-тестуванням, поставки комерційної версії якої почалися в лютому 1996. Visual Prolog є єдиною в світі Prolog-системою з 100% оболонкою виконаної в ідеології Visual густо високорівневих компонент, які спрощують розробку програм для систем Windows (версія 5.x підтримувала і платформу OS2).

Visual Prolog автоматизує побудову складних процедур і звільняє програміста від виконання тривіальних операцій. За допомогою Visual Prolog проектування призначеного для користувача інтерфейсу і пов'язаних з ним вікон, діалогів, меню, рядка повідомлень про станах і т.д. виробляється в графічному середовищі. З створеними об'єктами відразу ж можуть працювати різні Кодові Експерти (Code Experts), які використовуються для генерації базового і розширеного кодів на мові Prolog, необхідних для забезпечення їх функціонування.

Потужність мови Prolog в поєднанні з сучасною системою призначених для користувача інтерфейсів (GUI - Graphical User Interface) робить простий і інтуїтивно зрозумілою розробку систем, заснованих на знаннях, систем підтримай прийняття рішень, які планують програм, розвинених систем управління базами даних і т.д.

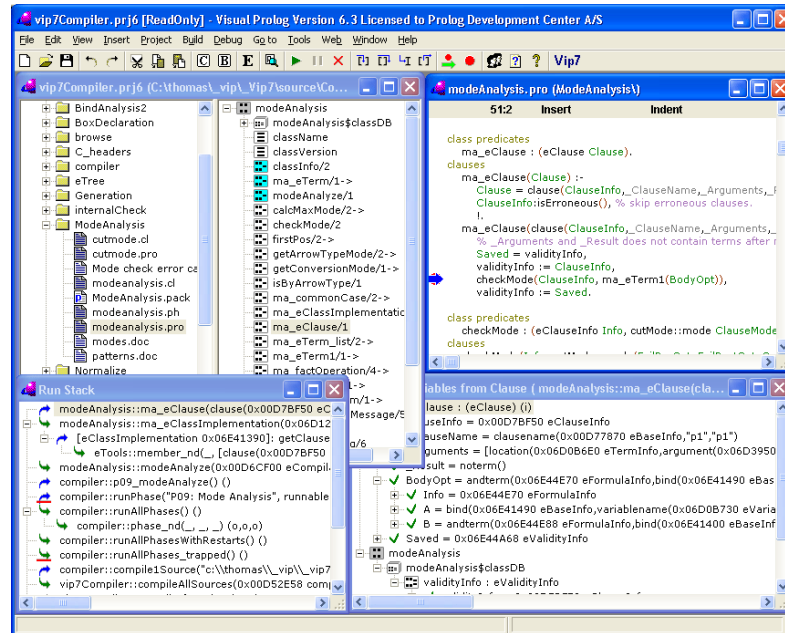


Рисунок 3.1 – Інтерфейс програмного забезпечення Visual Prolog

3.2 Програмне середовище CLIPS

CLIPS – програмне середовище для розробки експертних систем. Синтаксис і назва запропоновані Чарльзом Форґ (Charles Forgy) в OPS (Official Production System). Перші версії CLIPS розроблялися з 1984 року в Космічному центрі Джонсона NASA, як альтернатива існуючій тоді системі ART * Inference, поки на початку 1990-х не було припинено фінансування, і NASA змусили купити комерційні продукти (рис.3.2).

CLIPS є продукційною системою. Реалізація виведення використовує алгоритм Rete.

ефективності і безкоштовності. Будучи громадським надбанням, вона до цих пір оновлюється і підтримується своїм початковим автором, Гері Райлі (Gary Riley).

CLIPS включає повноцінний об'єктно-орієнтована мова COOL для написання експертних систем. Хоча вона написана на мові Сі, її інтерфейс набагато ближче до мови програмування LISP. Розширення можна створювати на мові Сі, крім того, можна інтегрувати CLIPS в програми на мові Сі [1].

CLIPS розроблений для застосування в якості мови прямого логічного висновку (forward chaining) і в своїй оригінальній версії не підтримує зворотного виведення (backward chaining).

Як і інші експертні системи, CLIPS має справу з правилами та фактами.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ БАЗИ ЗНАНЬ

Програмна реалізація набору правил, що були розроблені для системи здійснювалася у програмній середі CLIPS.

CLIPS є продукційною системою. Реалізація виведення використовує алгоритм Rete.

Основна ідея полягає в поданні знань у вигляді такої форми:

```

правило1:
  якщо
    (виконуються умовія1)
  ТОДІ
    (виконати дії1)
правило2:
  якщо
    (виконуються умовія2)
  ТОДІ
    (виконати діявїя2)
  
```

Представимо деякий набір правил, що були розроблені у роботі.

Набір правил для вибору методу оцінки нерухомості:

```

п1: ЕСЛИ
  (цель = Покупка) или (цель = Продажа) или (цель = Аренда)
  ТОГДА (Сравнительный метод)
п2: ЕСЛИ
  (цель = Покупка под. реконструкцию) или (цель = Покупка зе-
мельного участка под. строительство)
  ТОГДА (Затратный метод)
п3: ЕСЛИ
  (цель = Сдача объекта) или (цель = Покупка под производст-
во) или (цель = Покупка под другие коммерческие цели)
  ТОГДА (Доходный метод)
  
```

Набір правил для вибору бюджетної цінової категорії об'єкта:

ЕСЛИ

(Этаж = 1) и (Площадь = 20кв.м) и (Кол-во комнат = 1) и
(Санузел совмещен = да) и (Наличие бытовых условий = Нет)

ТОГДА (Бюджетная оценка)

Набір правил для вибору середньої цінової категорії об'єкта:

ЕСЛИ

(Этаж = 5) и (Площадь = 30кв.м) и (Кол-во комнат = 2) и
(Санузел совмещен = Нет) и (Наличие бытовых условий = Нет)

ТОГДА (Средняя оценка)

Набір правил для вибору високої цінової категорії об'єкта:

ЕСЛИ

(Этаж = 8) и (Площадь = 50кв.м) и (Кол-во комнат = 4) и
(Санузел совмещен = Нет) и (Наличие бытовых условий = да)

ТОГДА (Высокая оценка)

ВИСНОВКИ

В результаті виконання магістерської роботи було проаналізовано та вивчено:

- основні поняття та види оцінки вартості майна;
- експертні системи – їх склад, особливості архітектури;
- представлення знань у експертних системах;
- алгоритми пошуку відповіді у експертних системах;
- було проаналізовано програмне забезпечення, що використовується для розробки експертних систем. В якості системи розробки було вибрано мову CLIPS;
- була обрана продукційна модель представлення знань у експертній системі та проведене моделювання правил для оцінки вартості нерухомості для системи згідно з даною моделлю;
- розроблена база знань для експертної системи оцінки вартості нерухомості для подальшої інтеграції з експертною системою, що аналізує вид цінової категорії нерухомості та метод для кращої оцінки поставленої цілі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. П. Джексон. Введение в экспертные системы. Спб.: Питер. 2007. 384 с.
2. Л. Н. Ясницкий. Введение в искусственный интеллект. М.: Академия. 2005. 420 с.
3. Э. В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. Статические и динамические экспертные системы: учеб. пособие для вузов. М.: Финансы и статистика. 1996. 320 с.
4. Д. Б. Гаскаров. Интеллектуальные информационные системы. М.: Высшая школа. 2003. 542 с.
5. Методы представления знаний: Метод. указ. / Сост. И. Л. Коробова. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2003.
6. А. Частиков, Т.Гаврилова, Д.Белов. Разработка экспертных систем. Среда CLIPS. Спб.: БХВ-Петербург. 2005. 608 с.
7. В. А. Макушкин, К. А. Щербицкий. Экспертная система для контроля и диагностирования цифроаналоговых устройств. М.: МДНТП. 1991. 598 с.